EMUL51-PC™

User Guide

Edition 2

Contents

Product Notes

Warranty Information

The emulator board, trace board, pod board, and emulator cable are sold with a one-year warranty starting from the date of purchase. Components found to be defective under warranty will either be repaired or replaced at ICE Technology's discretion.

Pod boards that use a bondout processor are also warranted for one year from the date of purchase except for the processor. The bondout processor will be replaced once if Nohau determines that the failure in the bondout processor was not due to the user's actions. This replacement limit does not apply to the rest of the pod board.

Each optional adapter, cable, and extender is sold with a 90-day warranty, except that it may be subject to repair charges if damage was caused by the user's actions.

Seehau software is sold with no warranty, but upgrades can be obtained to all customers at the Nohua Web site: http://www.icetech.com.

Nohau makes no other warranties, express or implied, including, but not limited to the implied warranties of merchantability and fitness for a particular purpose. In no event will Nohau be liable for consequential damages. Third-party software sold by Nohau carries the manufacturer's warranty.

European CE Requirements

Nohau has included the following information in order to comply with European CE requirements.

User Responsibility

The in-circuit debugger application, as well as all other unprotected circuits need special mitigation to ensure Electromagnetic Compatibility (EMC).

The user has the responsibility to take required measures in the environment to prevent other activities from disturbances from the debugger application according to the user and installation manual.

If the debugger is used in an environment other than the intended (for example, field service applications), it is the user's responsibility to control that other activities cannot be disturbed in such a way that there may be risk for personal hazard/injuries.

Special Measures for Electromagnetic Emission Requirements

To reduce the disturbances to meet conducted emission requirements it is necessary to place a ground plane on the table under the pod cable and the connected processor board. The ground plane shall have a low impedance ground connection to the host computer frame. The insulation sheet between the ground plane and circuit boards shall not exceed 1mm of thickness.

System Requirements

CAUTION

Like all Windows applications, the Seehau software requires a minimum amount of free operating system resources. The recommended amount is at least 40%. (This is only a guideline. This percentage might vary depending on your PC.) If your resources are dangerously low, Seehau might become slow, unresponsive or even unstable. If you encounter any of these conditions, check your free resources. If they are below 40%, reboot and limit the number of concurrently running applications. If you are unable to free more than 40% operating system resources, contact your system administrator or Nohau Technical Support.

The following are minimum system requirements:

- Pentium 200 (Pentium II or faster is recommended)
- Single-Processor System
- Windows 95, 98, NT, 2000, or 2000 ME
- Random Access Memory (RAM)
  - For Windows 95/98: 64 MB
  - For Windows NT/2000/2000 ME: 128 MB

Warnings

To avoid damage to the pod or to your target, do not connect the pod to your target when the pod or target power is on

When powering up, always power up the emulator first followed by the target system. When powering down, power down the target system first followed by the emulator.
Failing to do so can cause damage to your target and/or emulator.

Do not apply power to your system unless you are absolutely sure the target adapter is correctly oriented. Failing to do so can cause damage to your target and/or emulator.

About This Guide

The *EMUL51–PC User Guide* describes how to use the EMUL51–PC emulation system with the Seehau graphical user interface.

This *EMUL51–PC User Guide* is intended for both novice and advanced users.

Downloading EMUL51–PC Product Documentation

To download an electronic version of this guide, do the following:

1. Open Nohau's home page at www.icetech.com.
2. Click **Publications/Documents**.
3. Click **Nohau Manuals**.
4. Scroll down to EMUL51–PC. Then select **EMUL51–PC User Guide** to download a PDF version of the guide.

1 Overview of the EMUL51–PC Emulator System

The basic hardware for the EMUL51–PC emulator system includes the following:

- Emulator board—plugs into a an ISA slot inside the PC, the High Speed Parallel (HSP) box, or the Universal Serial Bus (USB) box.
- Pod board—device-dependent board that replaces the MCU chip on the target system.
- Trace board (optional)—plugs into an ISA slot inside the PC, HSP, or USB and connects to the emulator board through a short ribbon cable.
- Five foot ribbon cable—connects the emulator and pod (see Figure 1).
- Target adapter—allows you to connect the pod board to your target system.

Three system configurations are available:

- High-Speed Parallel (HSP) box connects to the parallel printer port. See the following "High-Speed Parallel (HSP) Box" section.
- Universal Serial Bus (USB) box connects to the computer's USB port. See the following "Universal Serial Bus (USB) Box" section.
- PC Plug-In. See the following "PC Plug-In" section.

For information about configuring and installing your hardware, refer to Chapter 2, "Installing the Hardware" in this guide.

High-Speed Parallel (HSP) Box

In the HSP chassis, the emulator board and optional trace board are installed in an external box that connects to the PC's parallel printer port. The HSP option makes the emulator system portable, allowing you to move the system from one PC to another. You can also use this option with a portable laptop computer. The HSP eliminates the need for the two ISA bus slots.

Figure 1. HSP Box Connected to the Emulator, Trace, and Pod Board

Universal Serial Bus (USB) Box

When using a laptop computer, the USB box provides one of the most portable methods of connection and allows for full trace capability. A USB port is an external peripheral interface standard for communication between a computer and external peripheral over a cable that uses bi-serial transmission.

You can use the USB box to run the in-circuit emulator and optional trace board when ISA slots are unavailable in your computer.

Our personnel mount the emulator board, USB card, and optional trace board in the USB box chassis. The trace board connects to the emulator board through two ribbon cables. The pod board connects to the emulator board in the USB box with a five-foot ribbon cable. The USB card connects to the PC's USB port.

> **Note**
>
> When using the USB option, you must install the Seehau software first before connecting the Nohau hardware. This allows the computer to recognize the proper driver for the hardware.
>
> The USB option is not supported by Windows NT.

It is anticipated that the USB option will eventually replace the parallel port interface.

PC Plug-In

The emulator ISA board is plugged into an ISA slot in your PC and is connected with a five-foot cable to a device-dependent pod board. The optional ISA trace board is plugged into the PC and connects to the emulator board through a ribbon cable.

You can custom configure the emulator hardware to your requirements with various jumpers. See Chapter 2, “Installing the Hardware,” for details about jumper configuration.

User Interface

The emulator is configured and operated by the Seehau user interface.

Seehau is a high-level language user interface that allows you to perform many useful tasks, for example:

- Load, run, single-step and stop programs based on C or Assembly code.
- Set triggers and view trace (with optional trace board).
- Modify and view memory contents including Special Function Registers (SFRs).
- Set hardware breakpoints.
- Analyze code with Program Performance Analysis (PPA)

Quick Start for Installing Your Emulator System

The following illustration shows the major steps for installing and configuring the EMUL51–PC. For details, refer to the chapter referenced in each step.

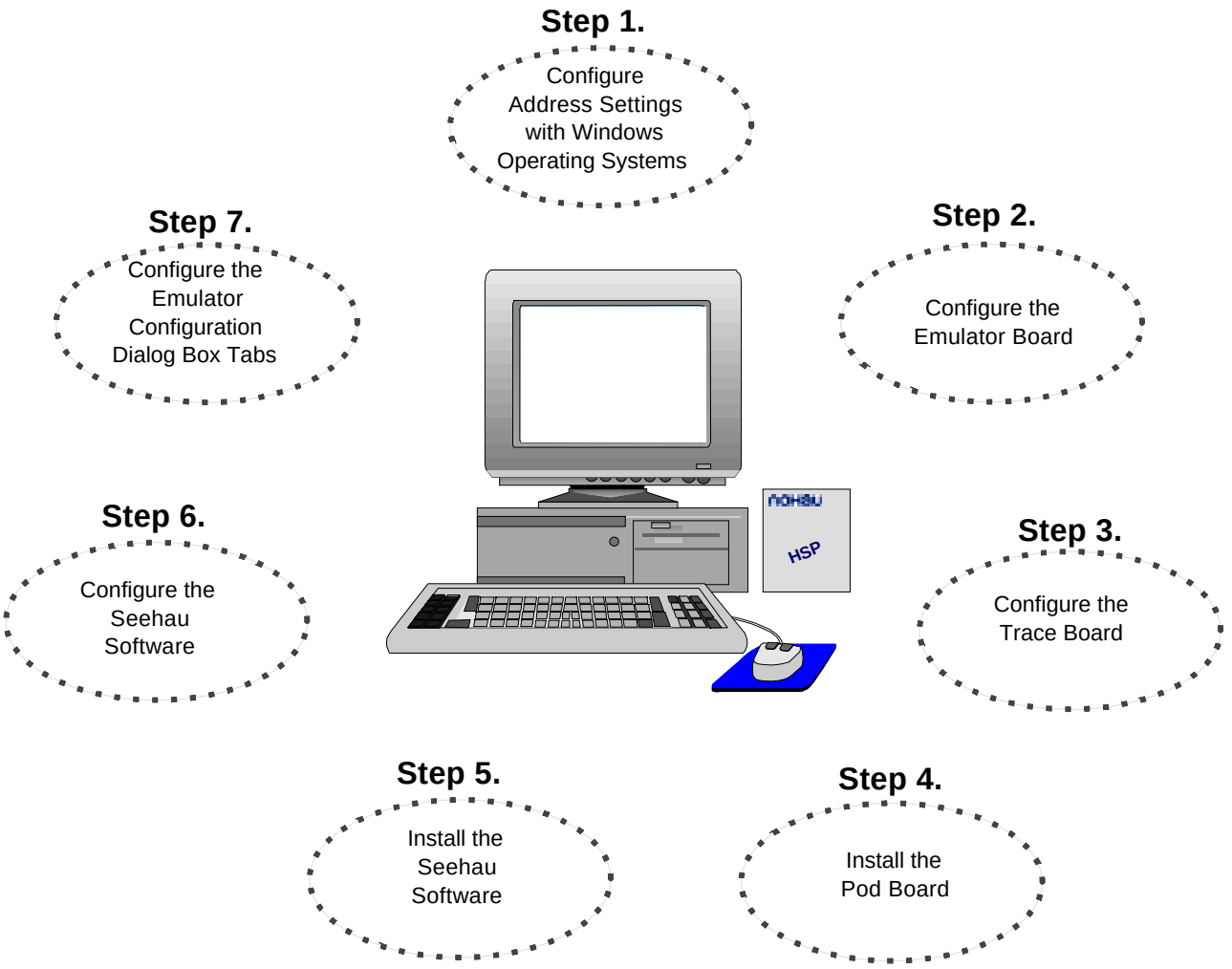


Figure 2. Steps for Installing and Configuring the EMUL51–PC and Seehau Software

2 Installing the Hardware

Quick Start for Installing the Hardware

Figure 3 shows the major steps for installing the EMUL51–PC hardware.

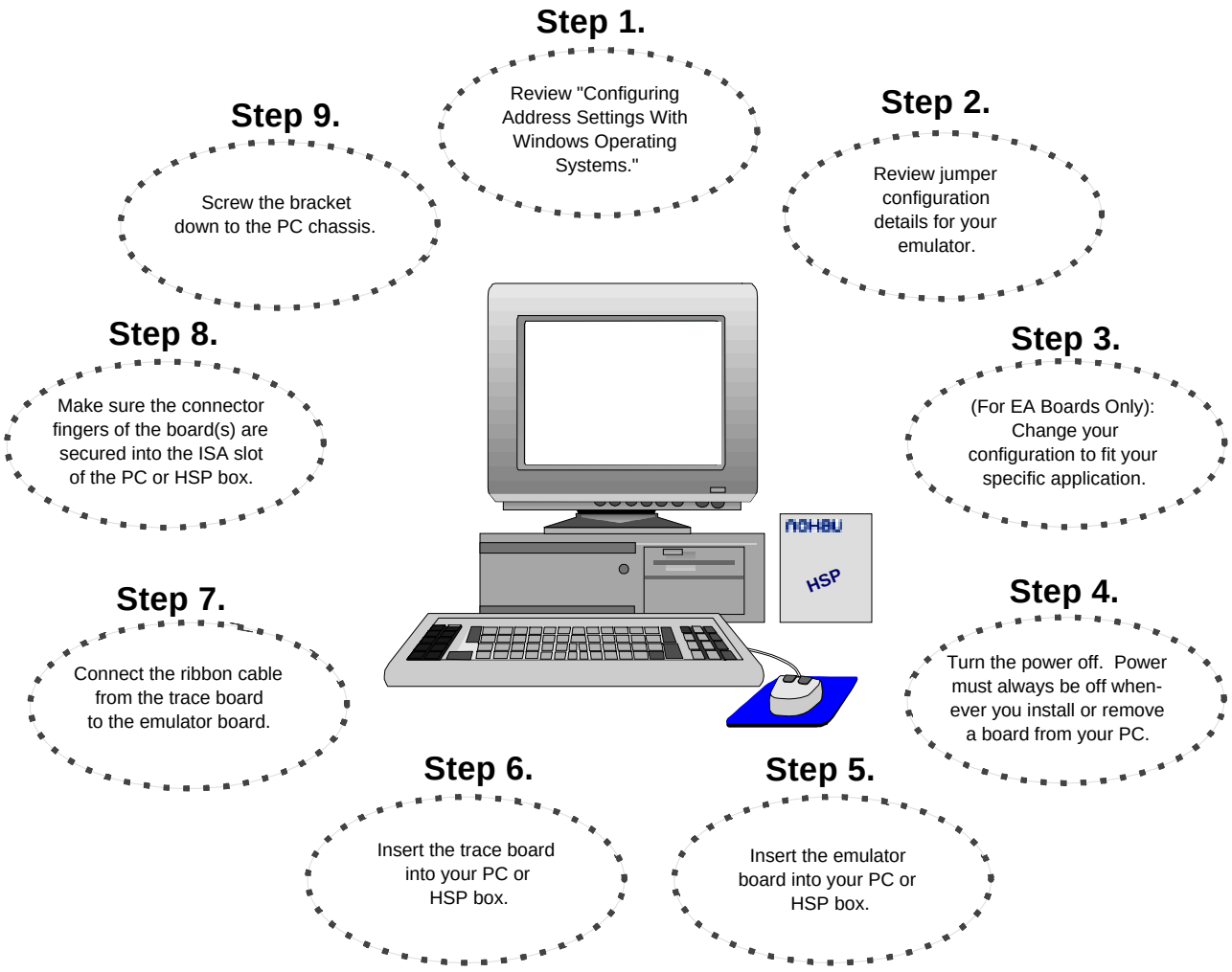


Figure 3. Steps for Installing the EMUL51–PC Hardware

Figure 4. HSP Box Connected to the Emulator, Trace, and Pod Boards

Configuring Address Settings With Windows Operating Systems

The following applies to all Windows operating systems:

- Default Address Ranges:
  - Emulator Board: 110H to 11FH
  - Standard Trace Board: 100H to 10FH
  - Enhanced Trace Board: 120H to 13FH
- Default Address Settings for the HSP Box:
  - No address conflict is possible when installing the HSP box with any Windows operating system. Use the default address ranges (previously listed).
  - Skip to the “Installing Emulator Boards” section later in this chapter.

Configuring Address Settings for the Emulator and Optional Trace Board

The following sections provide details about configuring address settings for the emulator and optional trace board for each Windows operating system. Refer to the section that covers your specific operating system.

Known Device Driver Conflicts

Nohau is aware of potential device driver conflicts with certain network cards running on Novell/ Netware networks. Most of these problems have been experienced when running Windows NT or Windows 2000 operating systems. Nohau Technical Support has also received some reports of hardware conflicts with both 3COM ISA network cards and some Novell network cards.

Possible Symptoms

- When starting Seehau, communication with the network stops. (You will be unable to access resources on the network.)
- Seehau will not start.

A possible solution might be to change your network card. Nohau Technical Support has not tested all network cards, although some customers have reported that the following network cards have resolved this conflict:

- Intel Ether Express Pro 10/100 ISA
- 3COM Etherlink III (905B or later) 10/100 PCI
- Bay Networks NetGear FA310TX 10/100 PCI

If you think the problem is caused by a software conflict, change the address of the emulator and trace boards (for example, use 310H and 300H respectively). Refer to the “Installing Emulator Boards” section in this chapter.

Configuring Address Settings With Windows 95/98/ME

Checking Your PC for Default Address Conflicts

1. Click the **Start** menu, and select **Settings**.
2. Click **Control Panel**.
3. Double-click **System**. The **Systems Properties** dialog box opens.
4. Click the **Device Manager** tab.
5. Click **Properties**.
6. Click **Input/Output**. Scroll the contents of the window to verify that no device is listed within that range.

Alternative Addressing

If you see a device present in the default address range for your emulator or trace board, do the following:

1. Beginning at the address 100H, scroll down to look for an unused address range:
   - 10H for any emulator board
   - 10H for the Standard Trace Board
   - 20H for the Enhanced Trace Board.
2. When you locate an unused address range, make a note of the base address of the range for use when configuring Seehau.
3. Refer to Appendix A, “Address Examples” to re-configure the base address of your board.

   The base address must be an even multiple of 10 (such as 200 or 210). If you have to change the address of the emulator or trace board, be sure to change both the board jumpers and the jumper settings in the software.

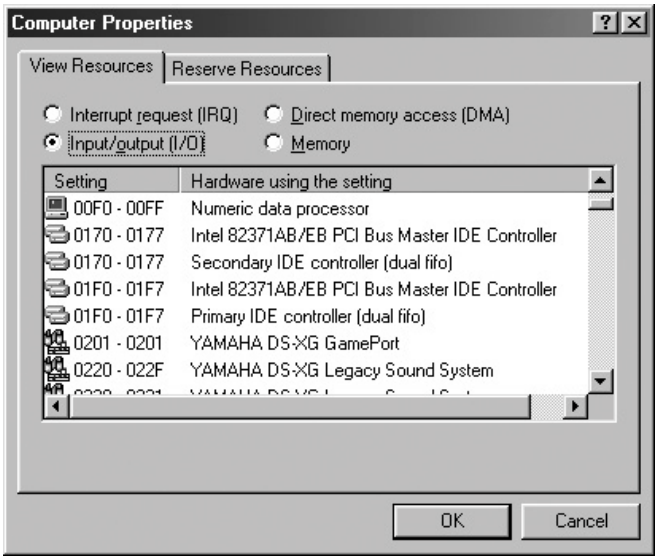


Figure 5. System I/O Resources

Configuring Address Settings With Windows NT

- First check your administrative privileges. This is required when installing the software.
- Then check your PC for default address conflicts.

Checking Administrative Privileges

1. Click the **Start** menu, and select **Programs**.
2. Select **Administrative Tools**, and click **User Manager**. The **User Manager** dialog box opens (Figure 6).
3. In the bottom half of the dialog box, double-click **Administrators**. The **Local Group Properties** dialog box opens displaying a list of login names (Figure 7).

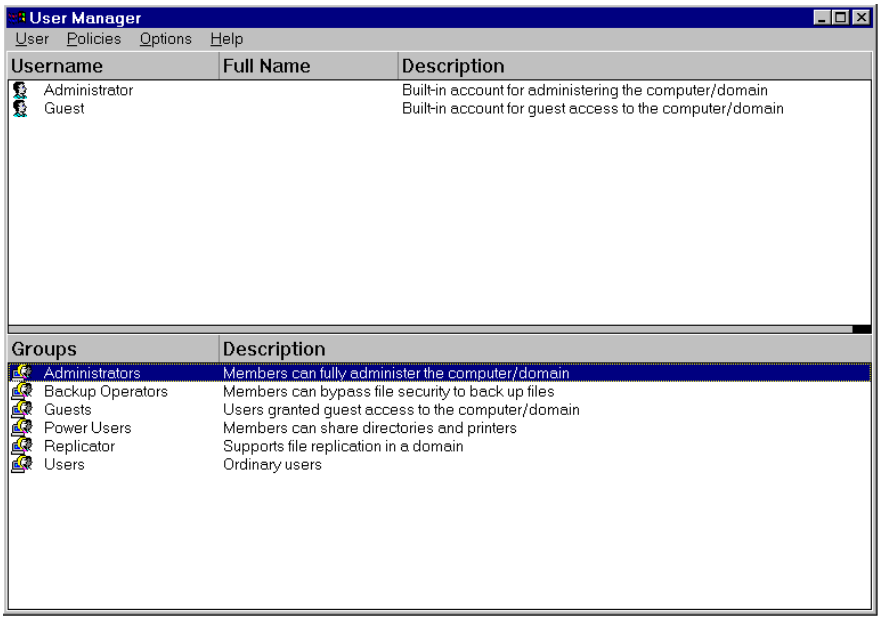


Figure 6. User Manager Dialog Box for Windows NT

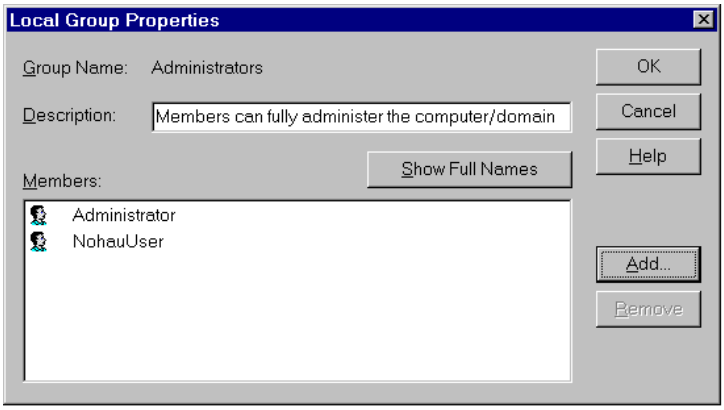


Figure 7. Local Group Properties Dialog Box for Windows NT

4. Look for your login name in the list of names. If your login name is not present, you are not set up with administrative privileges. Contact your System Administrator to update your privileges or give you the administrator's password.

Checking Your PC for Default Address Conflicts

1. Click the **Start** menu, and select **Programs**.
2. Select **Administrative Tools**, and click **Windows NT Diagnostics**. The Windows NT Diagnostics window opens (Figure 8).
3. Click the **Resources** tab.
4. Click **I/O Port**.
5. Check the I/O resources listed to verify that no device appears in a default address range.

Alternative Addressing

If you see a device present in the default range for your emulator or trace board, do the following:

1. Beginning at the address 100H, scroll down to look for an unused address range:
   - 10H for any emulator board
   - 10H for the Standard Trace Board
   - 20H for the Enhanced Trace Board.

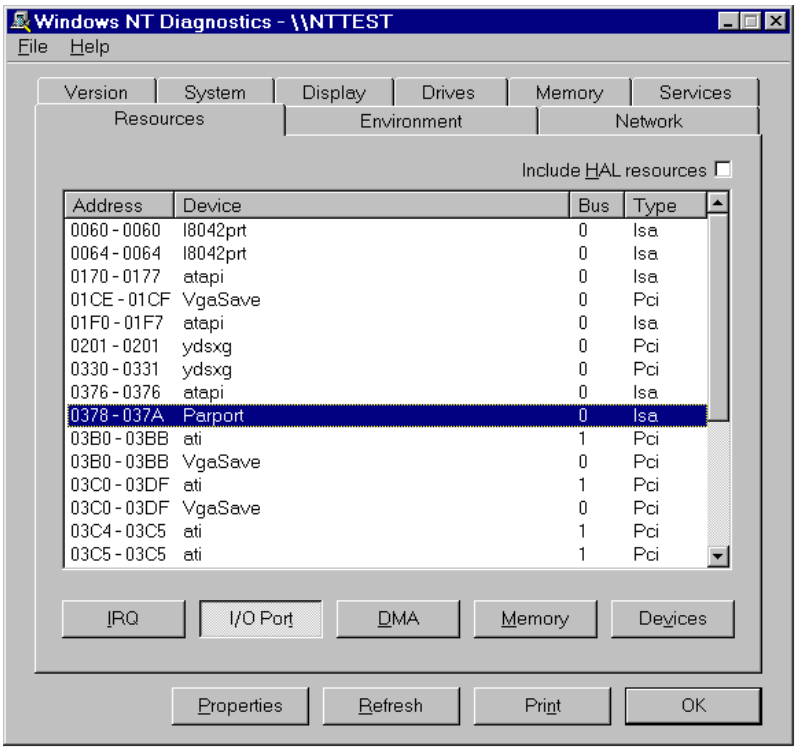


Figure 8. NT Diagnostics Window

2. When you locate an unused address range, make a note of the base address of the range for use when configuring Seehau.

3. Refer to Appendix A, “Address Examples” to re-configure the base address of your board.

Driver Troubleshooting

For details, see Chapter 8, “Seehau Startup Troubleshooting.”

- If you get a **Service or driver failed** error message when rebooting, you probably have a resource conflict.

- If you get a **create file failed** error message upon execution, the device driver did not properly start.

Nohau51 Device Driver With Windows NT

After installation, Windows NT Diagnostics will show the Nohau51 device driver present in the upper I/O range (FFxx). After launching Seehau, the driver is reassigned to the actual address ranges. In the Control Panel Devices window (Figure 9), you will see three columns: Device, Status, and Startup.

- Device: lists the Nohau device driver
- Status: displays **Started**
- Startup: displays **Automatic**

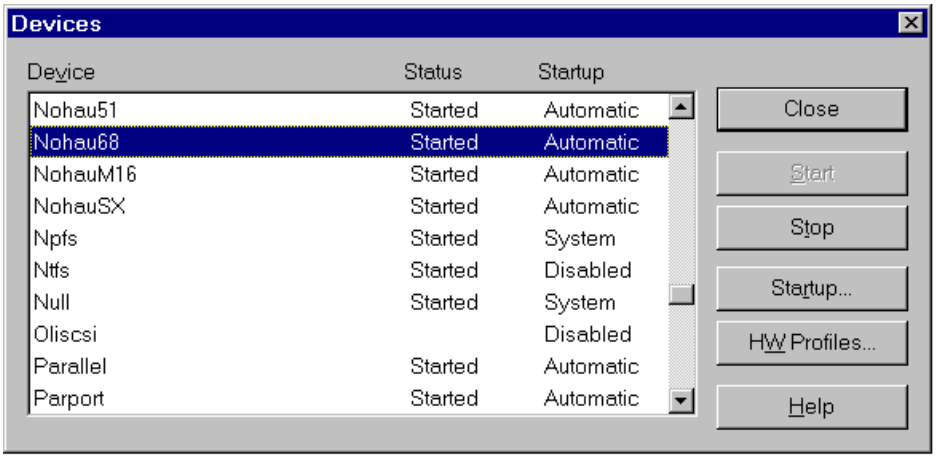


Figure 9. Control Panel Devices Window

Configuring Address Settings With Windows 2000

- First check your administrative privileges.
- Then check your PC for default address conflicts.

Checking Administrative Privileges

1. Click the **Start** menu, and select **Settings**. Click **Control Panel**.

2. From the **Control Panel**, double-click **Users and Passwords**. The Users and Passwords window opens (Figure 10).

3. Click the **Advanced** tab. Now click the **Advanced** button. The Local Users and Groups window opens (Figure 11).

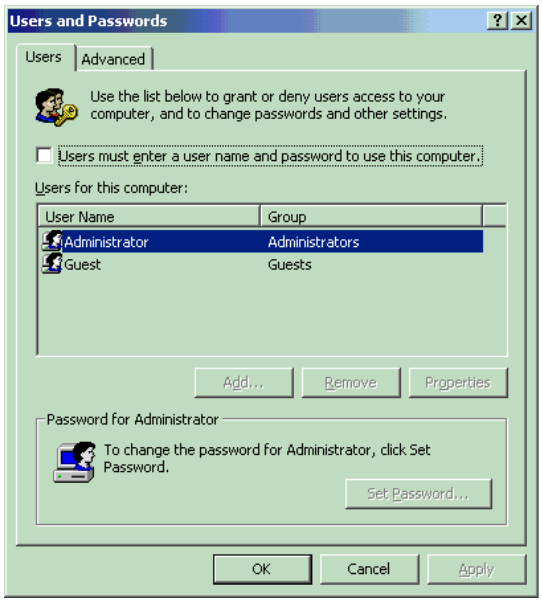


Figure 10. Users and Passwords Window

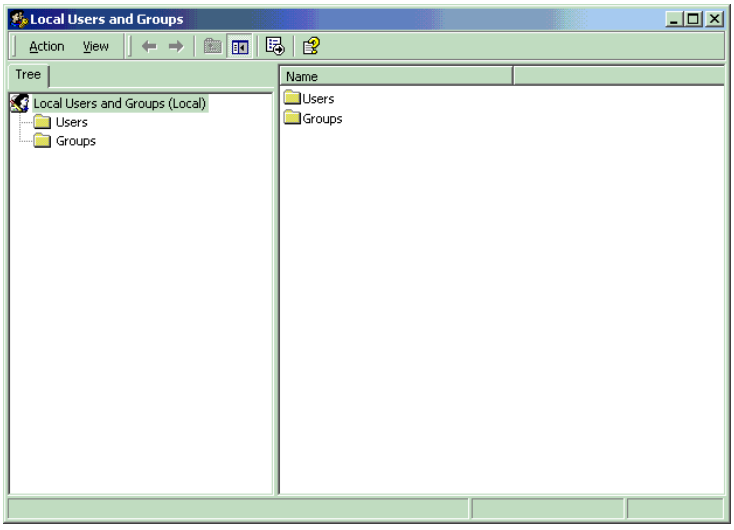


Figure 11. Local Users and Groups Window

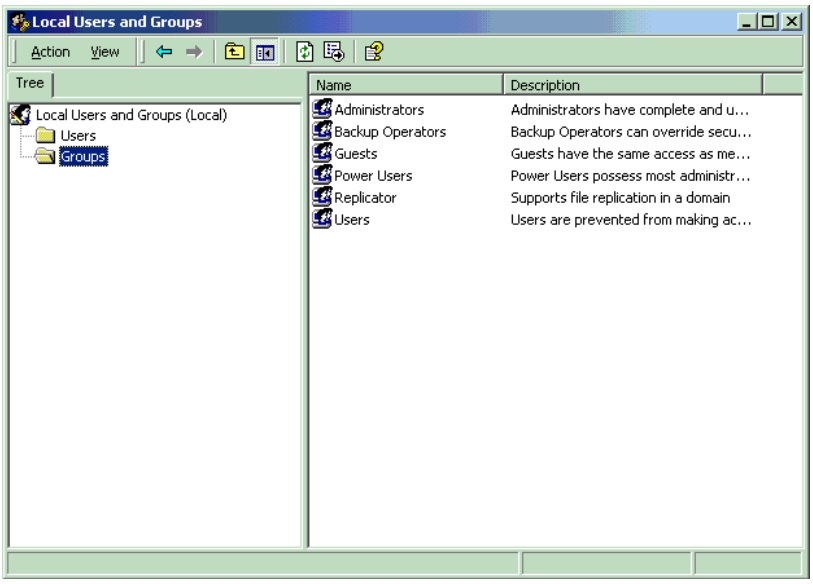


Figure 12. Local Users and Groups Window with Groups Folder

4. Click the Groups folder located in the left region of the window beneath Local Users and Groups.

5. Double-click the Groups folder. A list of groups appears in the right region of the window (Figure 12).

6. Double-click **Administrators**. Your user name should be listed.

> **Note**
>
> If you are not an administrator, ask your System Administrator to add you to this list.

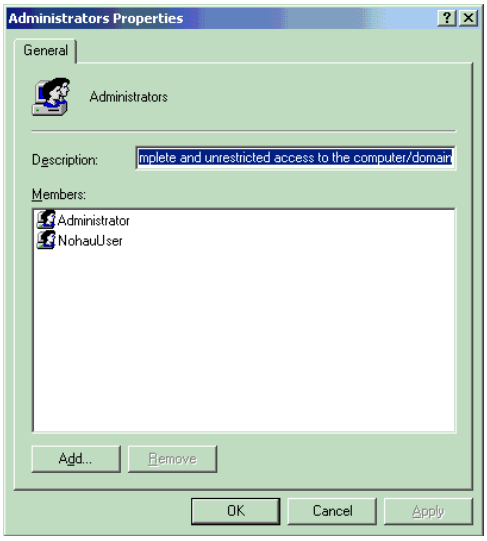


Figure 13. Administrator Dialog Box

Checking Your PC for Default Address Conflicts

1. Right-click the My Computer icon on your desktop, and select **Properties**. The System Properties window opens (Figure 14).

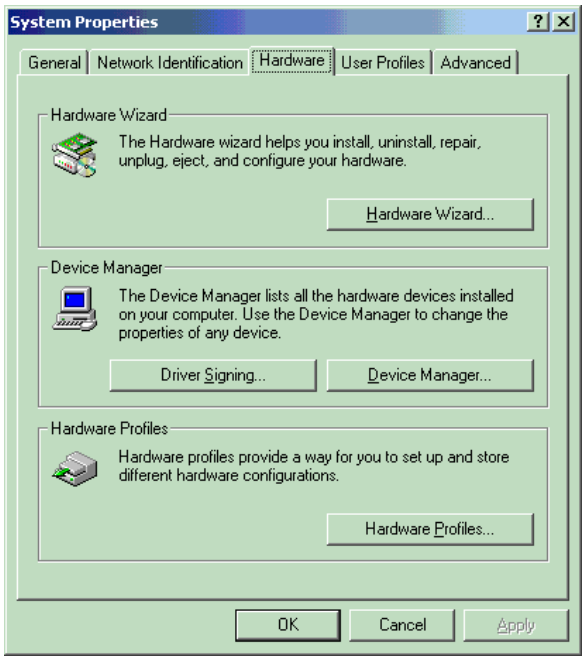


Figure 14. System Properties Window

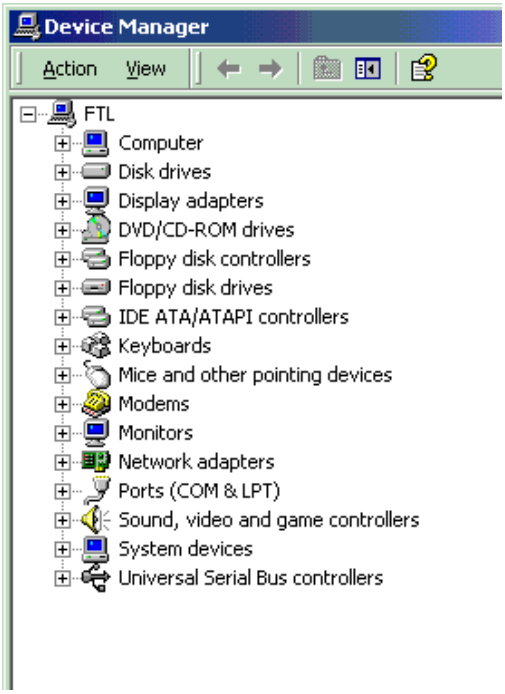


Figure 15. Device Manager Window

2. Click the **Hardware** tab. Then click **Device Manager**. The Device Manager window opens (Figure 15).

3. In the Device Manager window, select the **View** menu. Then click **Resources by Type**. A window opens that shows the system resources (Figure 16).

4. Double-click **Input/Output (I/O)**.

5. Check the I/O resources listed to verify that no device appears in the default address ranges.

> **Note**
>
> PCI bus items normally do not cause a conflict in that specific address range.

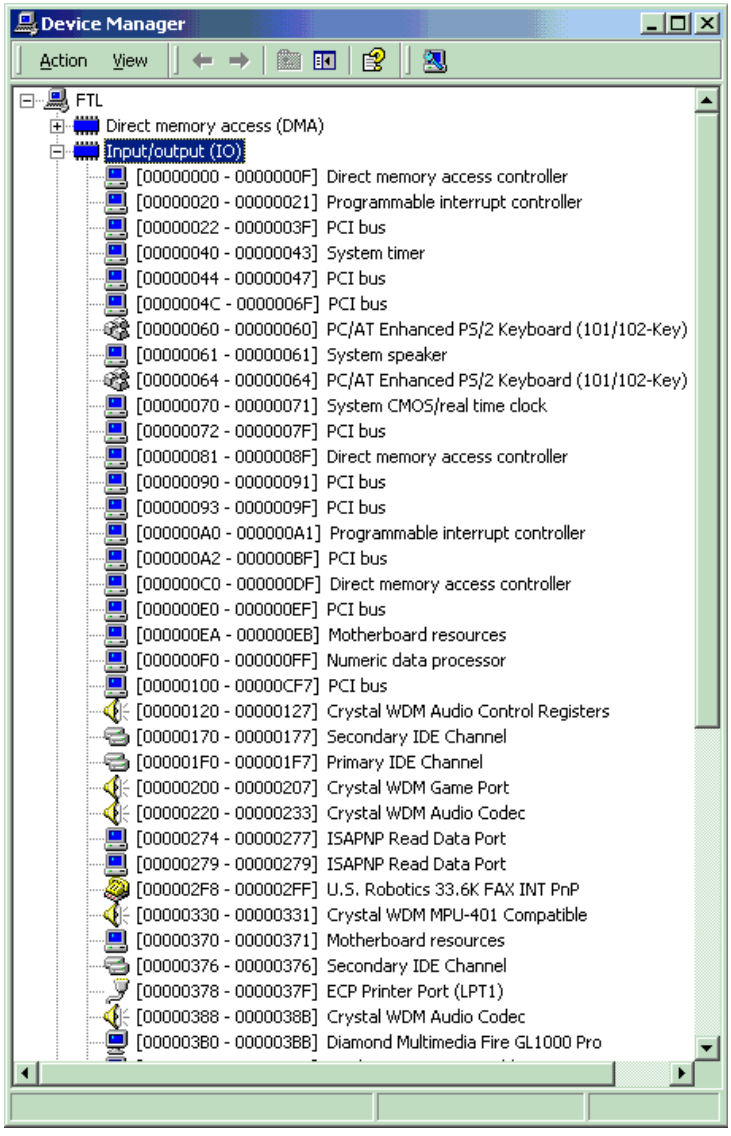


Figure 16. System Resources

Alternative Addressing

If you see a device present in the default address range for your emulator or trace board, do the following:

1. Beginning at the address 100H, scroll down to look for an unused address range:
   - 10H for any emulator board.
   - 10H for the Standard Trace Board.
   - 20H for the Enhanced Trace Board.
2. When you locate an unused address range, make a note of the base address of the range for use when configuring Seehau.
3. Refer to Appendix A, "Address Examples" to re-configure the base address of your board.

Driver Troubleshooting

For details, see Chapter 8, "Seehau Startup Troubleshooting."

- If you get a **Service or driver failed** error message when rebooting, you probably have a resource conflict.
- If you get a **create file failed** error message upon execution, the device driver did not properly start. Review the steps in this section again. You can use Windows 2000 System Properties to re-check that your port address has no conflicts.

Nohau51 Device Driver With Windows 2000

To verify that the Nohau51 device driver is properly installed, do the following:

1. From the **Start** menu, select **Programs**. Select **Accessories**, then **System Tools**.
2. Click **System Information**. The System Information window opens.
3. Double-click the Software Environment folder.
4. Double-click the Drivers folder. A list of active drivers appears (Figure 17). Refer to the **Name** column and scroll down to **nohau51**.
5. Verify the driver is running. In the **State** column, you should see the word **Running**. In the **Status** column, you should see **OK**.

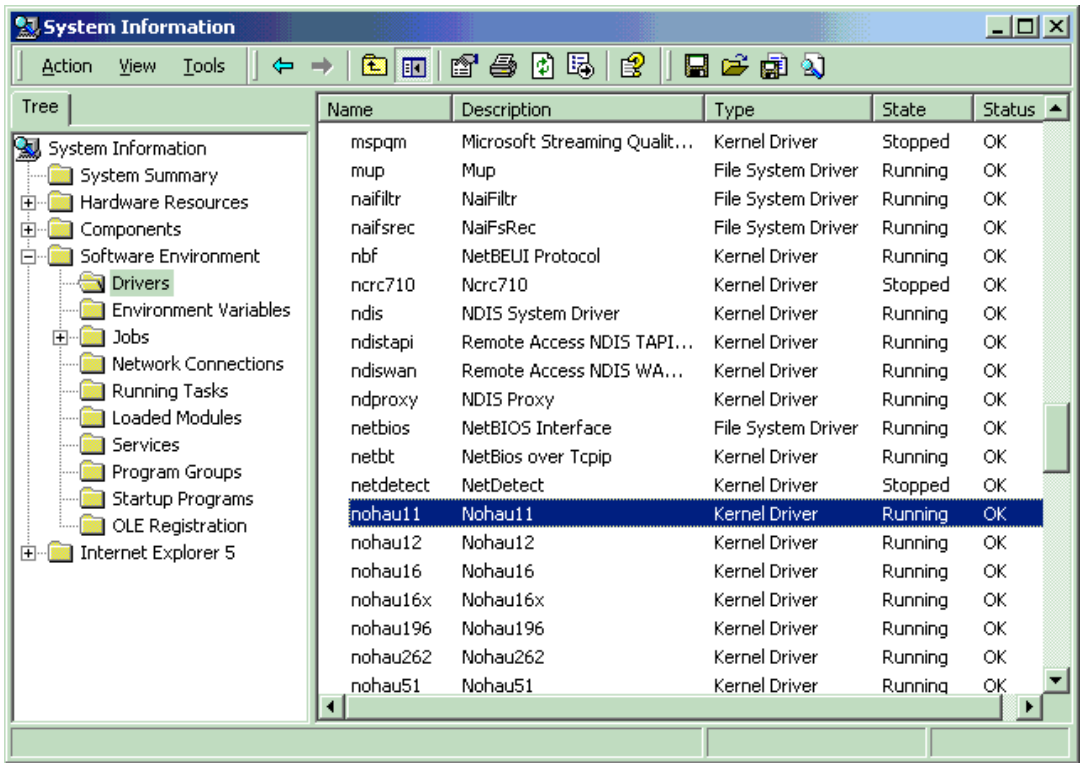
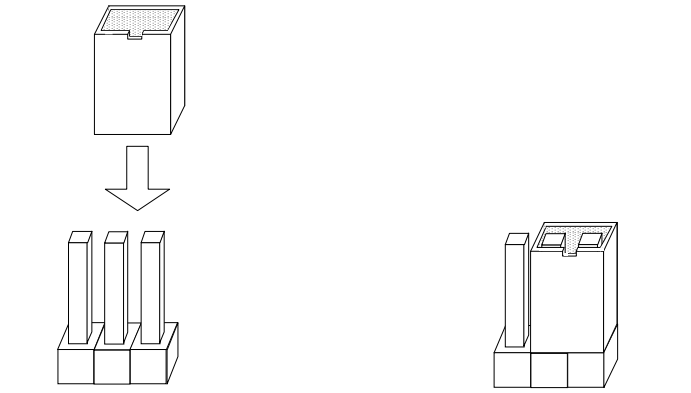


Figure 17. List of Active Drivers

Inspection of Jumpers

> **Note**
>
> Jumpers have been preset at the factory prior to shipment. On the boards you receive it is important to compare the jumper configurations against the configurations described in this guide. If a jumper is installed other than shown, refer to the jumper description information in Chapter 2, “Installing the Hardware,” or in Chapter 3, “Installing the Pod Board” before changing its position.

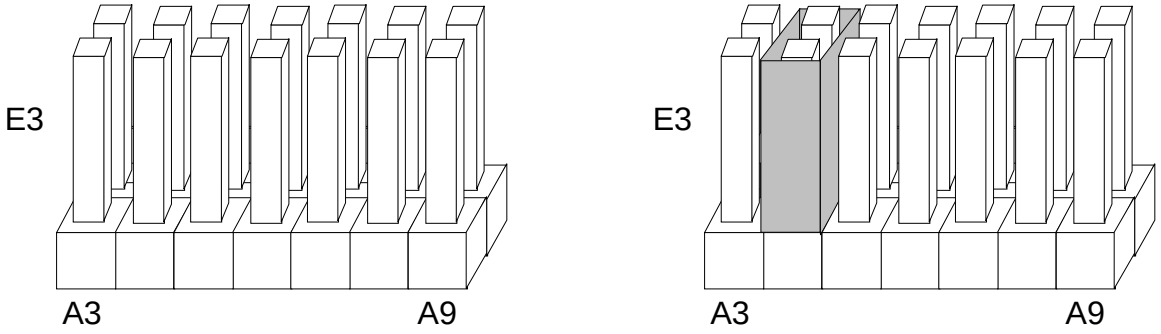
Jumpers for the emulator and trace boards are described in this section of the guide. For information on pod boards, refer to Chapter 3, “Installing the Pod Board.”



Pins Not Connected

Block Connecting Pins

Figure 18. Typical Set of Jumper Pins



No pins connected

Two pins connected (A4 pair)

Figure 19. Typical Set of Paired Jumper Pins

Several sets of jumper pins are provided on the boards for configuring on-board functions. To install a jumper, slide a jumper block over two adjacent pins. (See Figures 18 and 19 for details.)

Installing Emulator Boards

This section provides installation details for the following emulator boards: Standard, E32/E128, and the Advanced Emulator Board.

After you have inspected the boards for any damage, you can install the emulator board in your PC, HSP or USB box by doing the following:

1. Configure your emulator board jumpers. In this chapter, refer to the appropriate jumper description for your board.

2. Turn the power off. (Power must always be off when you plug in any PC board.)

3. Plug the emulator board into the ISA slot in your PC, HSP or USB box (if not already installed).

4. Screw the bracket down to the PC chassis.

5. Connect either end of the long ribbon cable to the connector on the emulator board located at the back of the PC, HSP or USB box. The connector has a notch on it, so you cannot insert it the wrong way.

6. Connect the other end of the ribbon cable from the emulator board to the pod. Close the locks on the connector over the cable.

Note

Although the cable and the connector are keyed, it is possible to plug them in backwards. Note the following illustration for proper connection.

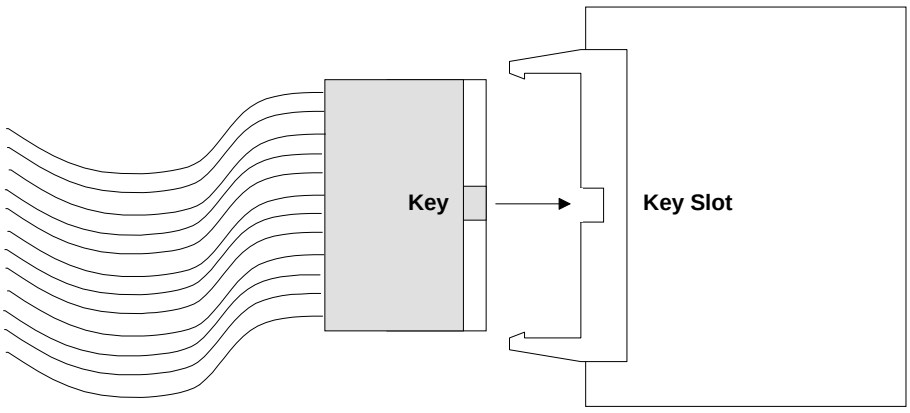


Figure 20. Connecting the Emulator to Your Pod Board

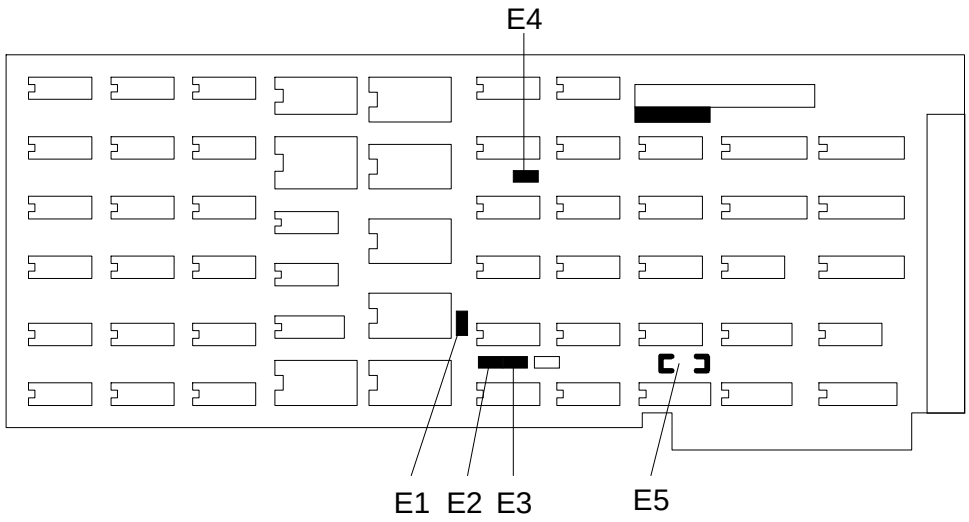


Figure 21. Locations of Jumpers on the Standard Emulator Board

Standard Emulator Board Description

On the emulator board the locations of jumper pins are designated on the board artwork as E1, E2, E3, E4 and E5 (pairs A3–A9). (See specified rectangles in Figure 21.) The Standard Emulator Board is available in two memory sizes, 32K (E32) or 128K (E128).

32K Emulator

When shipped from the factory, the E32 emulator board is normally configured for 32K RAM and with the I/O port address set for 110H. Figures 22 and 23 show details of the jumper connections for this configuration.

The 32K emulator board is manufactured in either of two ways, depending on memory availability at the time of manufacture, as follows: one large memory installed in the bottom socket; or, four small memories installed in the four sockets.

One-Memory E32 Board

Figure 22 shows the jumper configuration if there is a single memory in the bottom socket. Note the wire wrap connection between E2 and E3. The memory is a 32K by 8 static RAM, of the type 62256, 43256, 55256, or similar number.

Four-Memory E32 Board

Figure 23 shows the jumper configuration if there are four small memories in the sockets. The memories are 8K by 8 of the type 6264 or similar. (If there are four large memories in the sockets, the board is not a 32K emulator board.)

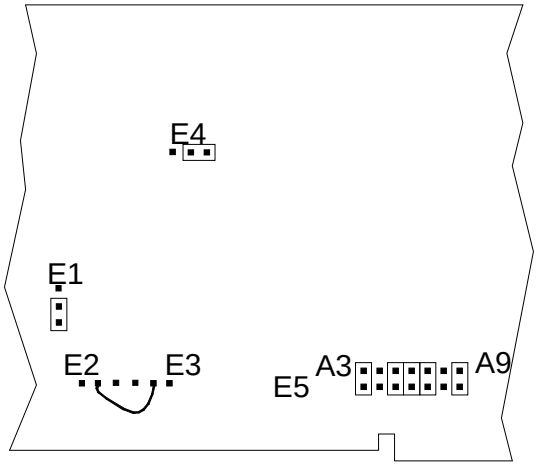


Figure 22. Emulator Board with ONE CHIP 32K Installed
(Address 110H. If mapped to the emulator, Code and Xdata are always overlaid.)

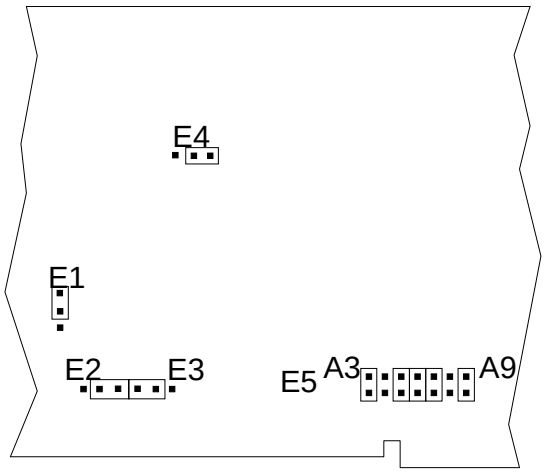


Figure 23. Emulator Board with FOUR CHIP 32K RAM.
(Address 110H. If mapped to the emulator, Code and Xdata are always overlaid.)

128K Emulator

If four 32K by 8 RAM chips are inserted in sockets U25, U26, U27 and U28, then the emulator board is configured as shown in Figure 24 or Figure 25. The differences between these configurations are based on whether the 64K code and 64K XDATA are in separate areas or are overlaid.

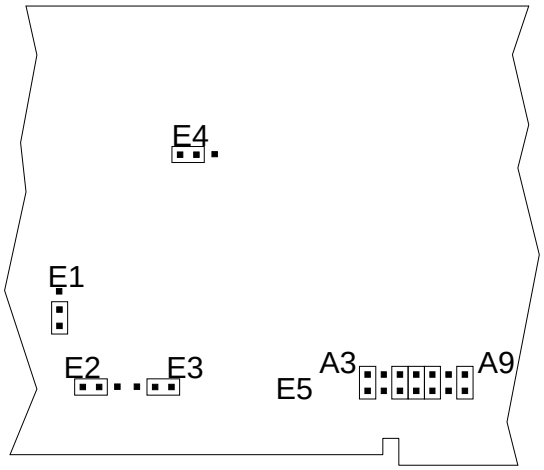


Figure 24. Emulator Board with 128K Installed
(64K Code and 64K XDATA in Separate Areas, Address 110H.)

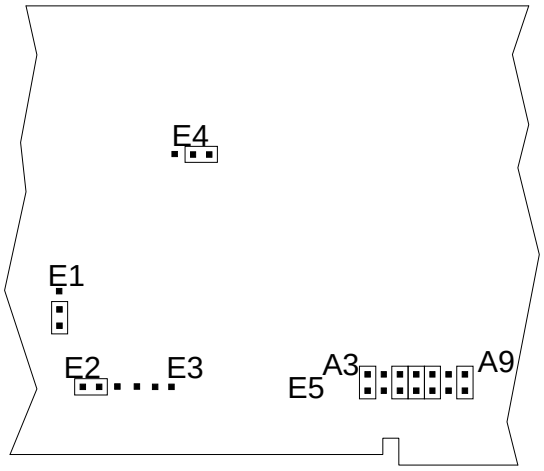


Figure 25. Emulator Board with 128K Installed
(64K Code and 64K XDATA in Overlaid Areas.)
Address 110H, must be invoked as a 32K emulator.

Setting Up Bank Switch Connections

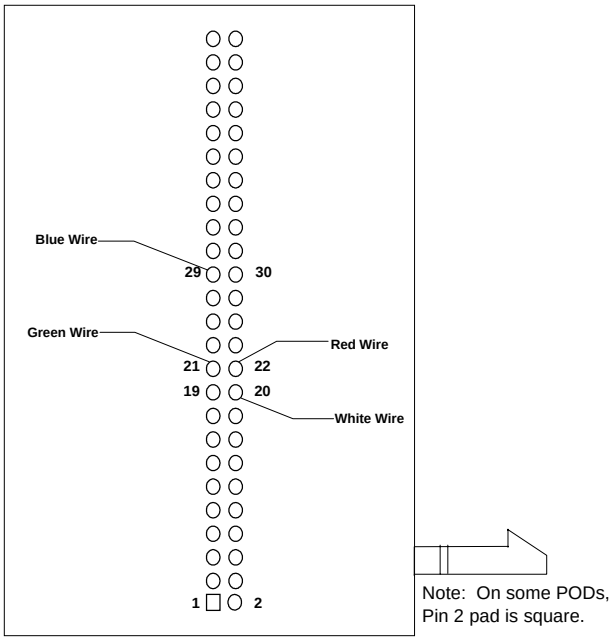


Figure 26. Pod Board Modification

Bank switch wire color	Bank switch input bit	Target chip select or extended address bit	Connections for trace inputs used for breakpoints and triggering on bank specific locations
Red	0	CS0 / A16	E0 *
Green	1	CS1 / A17	E1 *
White	2	CS2 / A18	SY0
Blue	3	CS3 / A19	SY1

Figure 27. Control Lines

EMUL51–PC / EA256 – 256K Bank Switch Emulator Configuration

The EA256 – 256K board comes with bank switch support for up to 256K memory and also supports the Dallas processors with speeded-up cycle times (DS80C320, DS87C520, andDS87C530) and other pods with non-standard bus timing requirements.

This board can be configured for various modes of operation. The following is a partial list of the standard EPROMs that are used. The EPROM labeled COM1.4 is used for most pod boards with Intel MCS51 architecture; COM1.46 is used specifically for the POD–C320; and COM1.47 is used for the POD–C520 and POD–C530**.** For a complete list of COM Proms and what they are used for, see Appendix I, “Emulator Board Communication EPROMs.”

To reconfigure for each of the different modes, refer to the following jumpers (Figure 28).

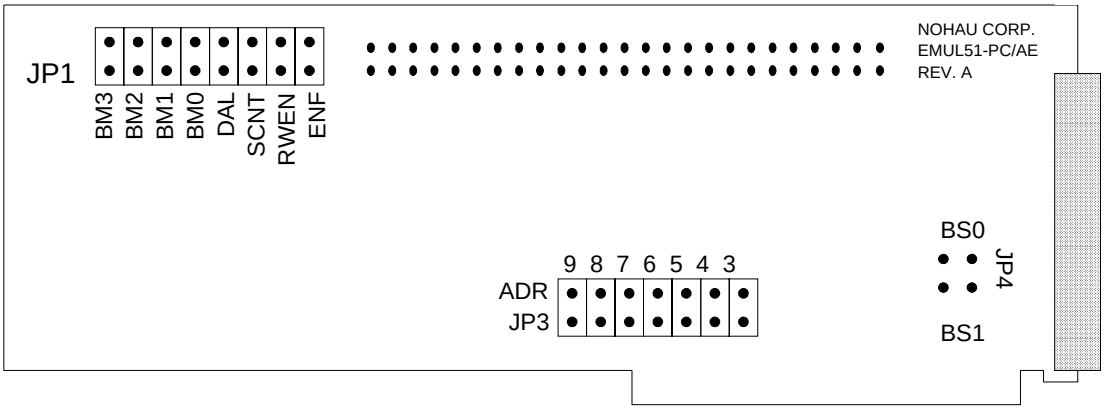


Figure 28. EMUL51–PC/EA256 Header Positions

Description of Jumpers

JP3	ADR	Used for board addressing and function the same as the standard emulator board.
JP4	BS0–BS1	These jumpers should not be installed when using bank switching.
JP1	BM0–BM3	Used to set the memory mode for the emulator. See the following “(BM Jumpers” section).
JP1	DAL	Selects the special mode for the Dallas MCUs: 320, 323, 520, and 530 only. Installed = Dallas, Removed = Normal Remember to change the EPROM to match the type of pod being used. **Note**: Currently the I/O feature is not supported for the Dallas 80C320.
JP1	SCNT	Install this jumper if you are using a pod board that supports DMA (80C152 and 80C452).
JP1	RWEN	When removed, allows you to use the R/W lines on the micro as general I/O. This mode will work with the POD–31A. This mode is not supported for the Dallas 80C320 (POD–C320). If you use this mode, remove the RE and WE jumpers on the pod board. Start the emulator, selecting one of the following pods: POD–C52, POD–C51FC, POD–C528. Select a pod based on the MCU installed in the pod. This allows the operation of R/W lines as I/O. If you are using the POD–C520, you must remove this jumper. Refer to Chapter 3, “Installing the Pod Board," for specific pod information.
JP1	ENF	Untested feature; leave jumper out.

Bank Switch / Memory Modes

X = Don’t care OUT = Jumper out IN = Jumper in

(BM) Jumpers

3	2	1	0	Description	Set Emulator Type As
IN	IN	IN	IN	No bank switching; 64K CODE + 64K XDATA in separate memory areas.	EA256
IN	IN	X	OUT	No bank switching; 64K CODE with 64K XDATA overlaid on code memory.	EA256 overlay

Bank Size = 32K

3	2	1	0	Description	Set Emulator Type As
OUT	IN	IN	IN	32K in root page (0000 – 7FFF) + three pages in upper memory (8000 – FFFF) + 64K of XDATA memory. First upper 32K page is the same as root page (Figure 29).	EA256
OUT	IN	IN	OUT	32K in root page (0000 – 7FFF) + three pages in upper memory (8000 – FFFF) + 64K of XDATA memory. Last upper 32K page is the same as root page (Figure 30).	EA256
OUT	IN	OUT	IN	32K in root page (0000 – 7FFF) + seven pages in upper memory (8000 – FFFF). First upper 32K page is the same as root page. Also, XDATA is mapped to the emulator. The last two upper pages contain the XDATA memory (Figure 31).	EA256
OUT	IN	OUT	OUT	32K in root page (0000 – 7FFF) + seven pages in upper memory (8000 – FFFF). Last upper 32K page is the same as root page. Also, XDATA is mapped to the emulator. The first two upper pages contain the XDATA memory (Figure 32).	EA256

Bank Size = 64K

3	2	1	0	Description	Set Emulator Type As
IN	OUT	IN	X	Two banks of 64K for code + 64K for XDATA memory; separate memory areas (Figure 33).	EA256
IN	OUT	OUT	IN	Four banks of 64K for code memory. If XDATA is mapped to emulator, XDATA memory appears in the first 64K bank of memory (Figure 34).	EA256
IN	OUT	OUT	OUT	Four banks of 64K for code memory. If XDATA is mapped to emulator, XDATA memory appears in the last 64K bank of memory (Figure 35).	EA256

Bank Size = 48K

3	2	1	0	Description	Set Emulator Type As
OUT	OUT	X	X	16K in root page (0000 – 3FFF) + three pages of 48K in upper memory (4000 – FFFF), + lower 48K (0000 – BFFF) of XDATA available in separate memory area, and the upper 16K of XDATA memory is the same as the root 16K of code memory (Figure 36).	EA256

Operation of the Bank Switch Pages

Refer to the following figures for information about bank switching memory for both types of pods, POD–31A, or other pod types. Modified bank switch pods use a different logic than POD–31A. Refer to the memory map information for each mode and pod type. (Figure 29 through Figure 36 for POD–31A; Figure 37 through Figure 44 for all other pods.)

Note

POD–31A refers to POD–C32HF–42, POD–C154HF–42, POD–C320–25, and any similar pod built with the POD–31A fabrication.

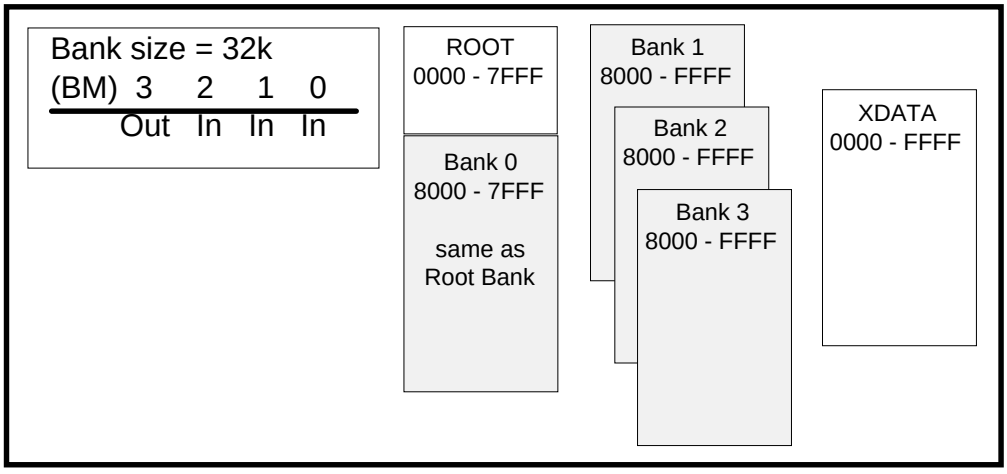


Figure 29. Bank Size 32K (1000b) for POD–31A-Type Pods

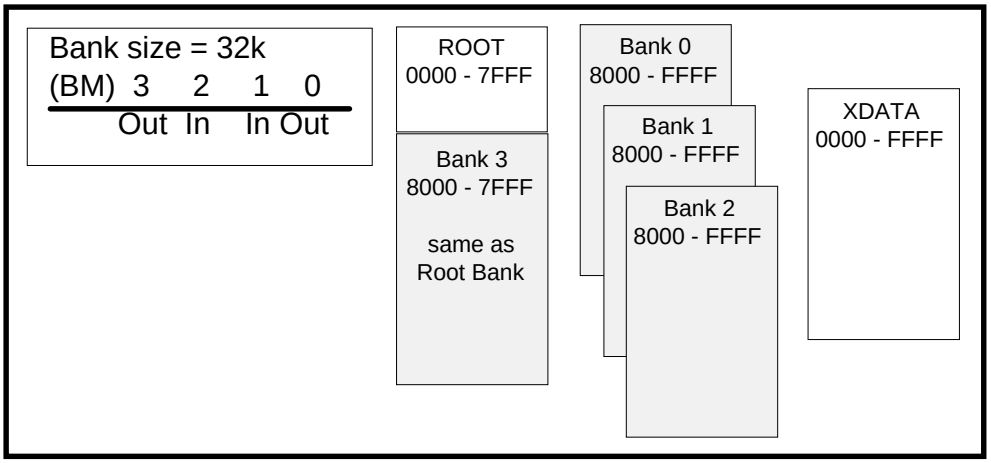


Figure 30. Bank Size 32K (1001b) for POD–31A-Type Pods

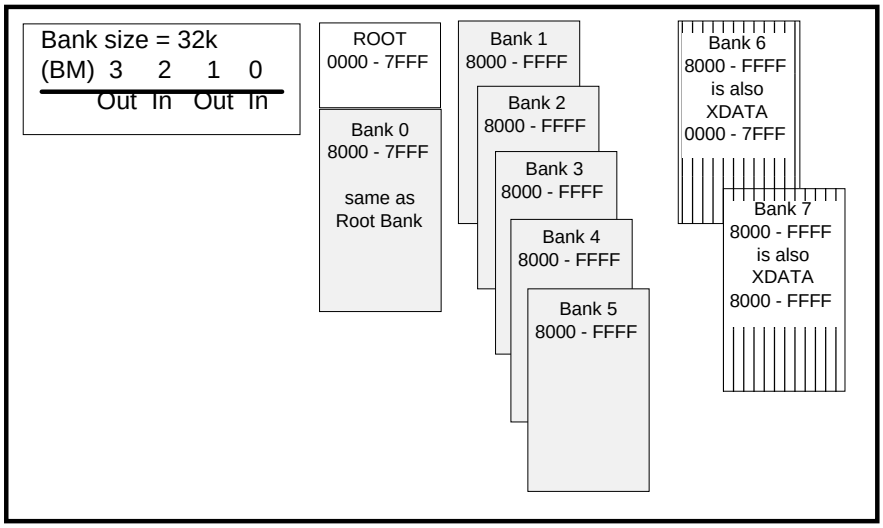


Figure 31. Bank Size 32K (1010b) for POD–31A-Type Pods

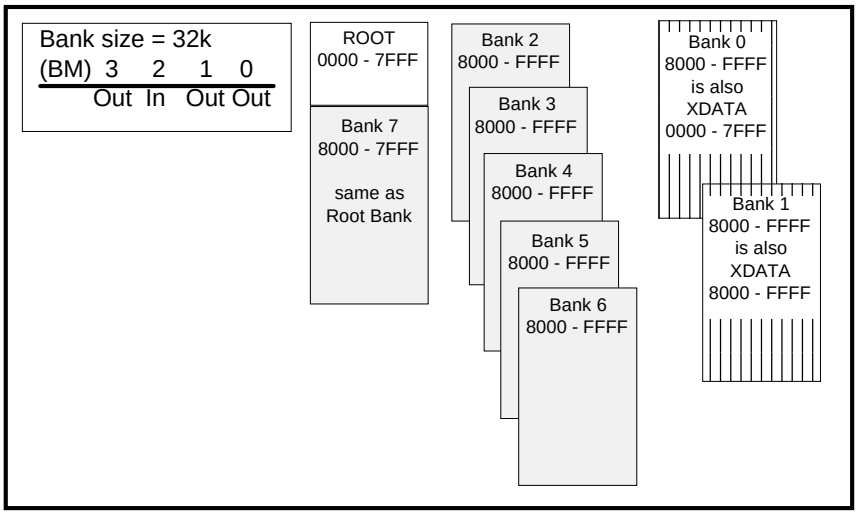


Figure 32. Bank Size 32K (1011b) for POD–31A-Type Pods

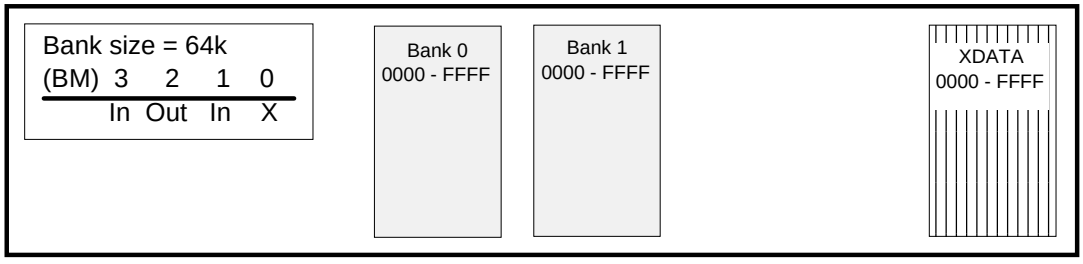


Figure 33. Bank Size 64K (010Xb) for POD–31A-Type Pods

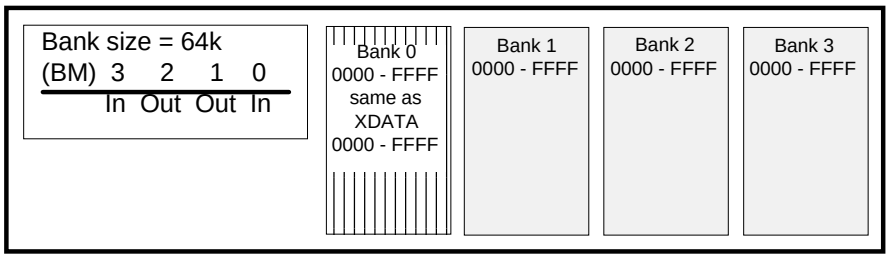


Figure 34. Bank Size 64K (0110b) for POD–31A-Type Pods

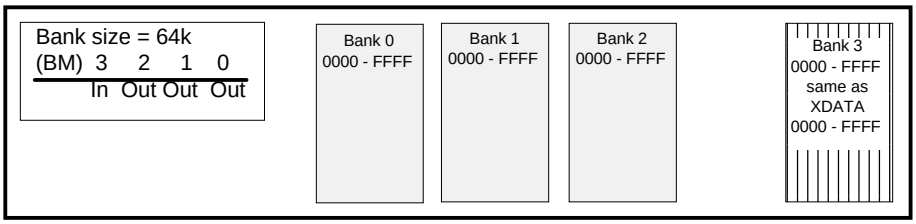


Figure 35. Bank Size 64K (0111b) for POD–31A-Type Pods

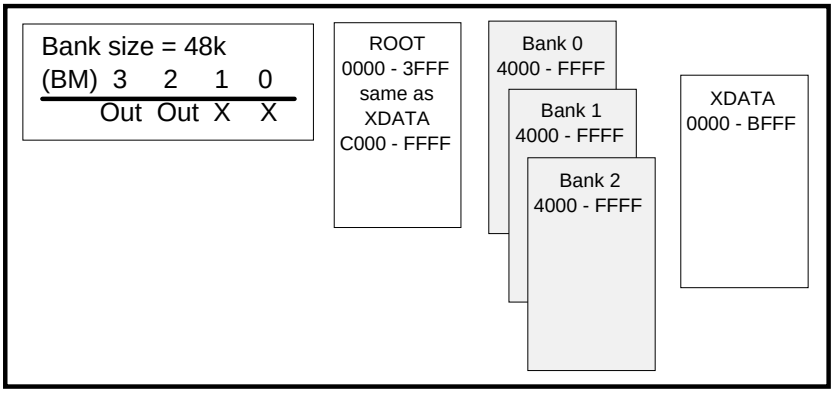


Figure 36. Bank Size 48K (11XXb) for POD–31A-Type Pods

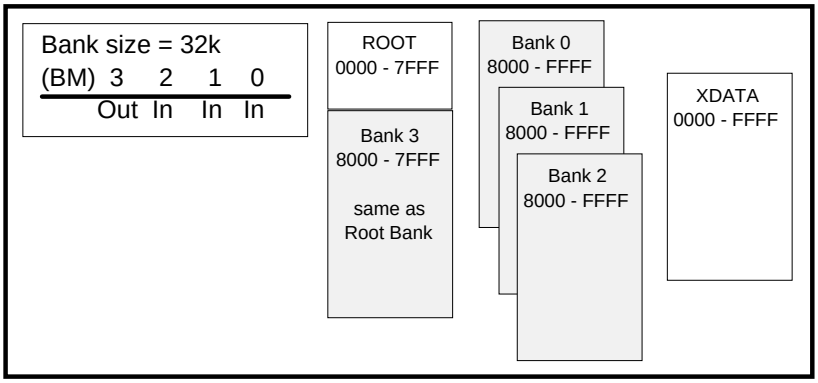


Figure 37. Bank Size 32K (1000b) for All Other Pod Types

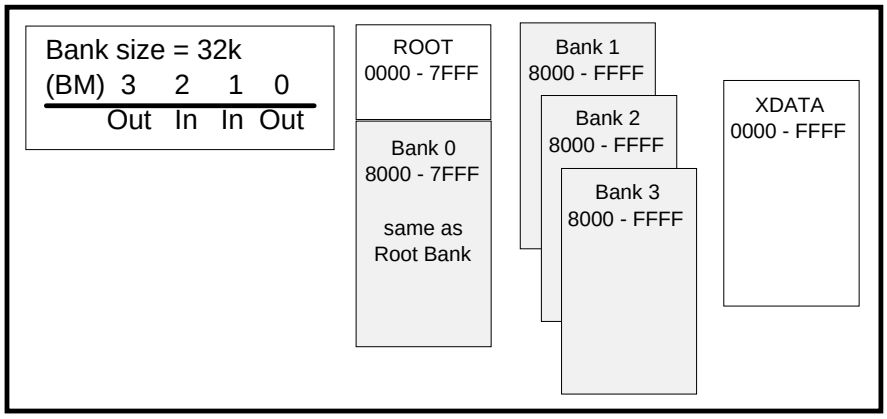


Figure 38. Bank Size 32K (1001b) for All Other Pod Types

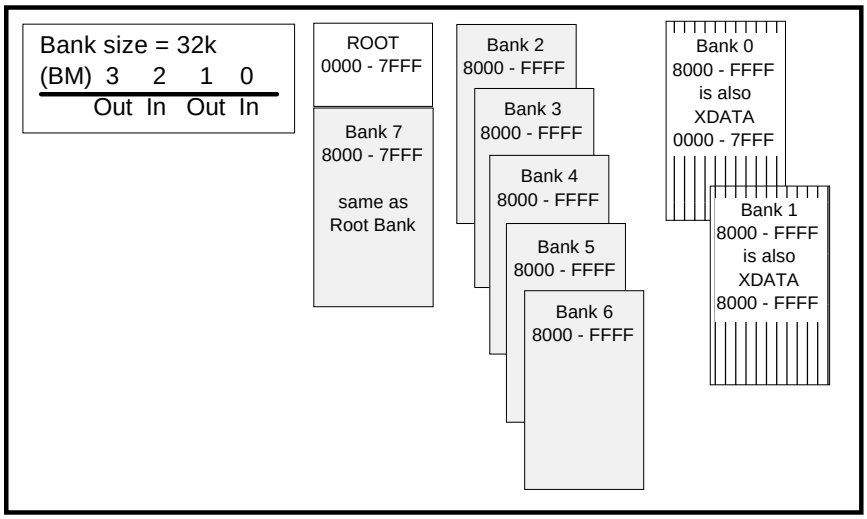


Figure 39. Bank Size 32K (1010b) for All Other Pod Types

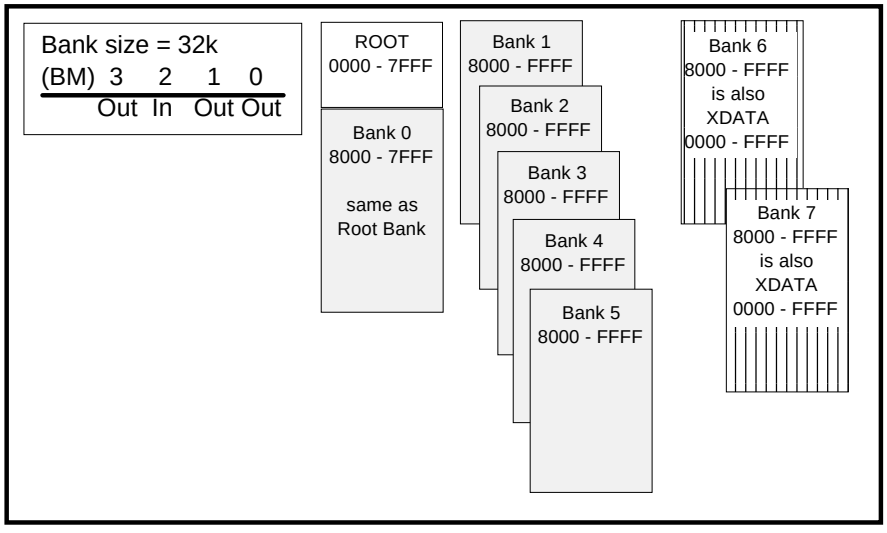


Figure 40. Bank Size 32K (1011b) for All Other Pod Types

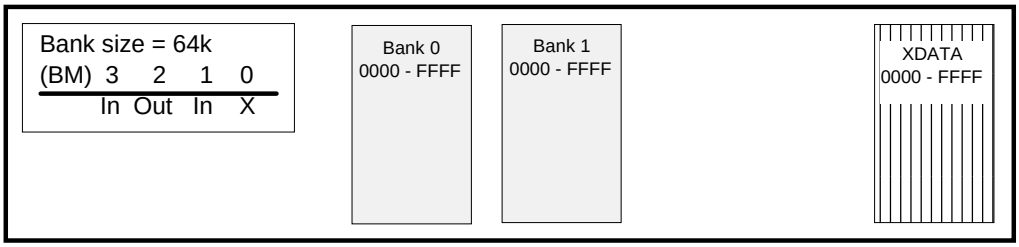


Figure 41. Bank Size 64K (010Xb) for All Other Pod Types

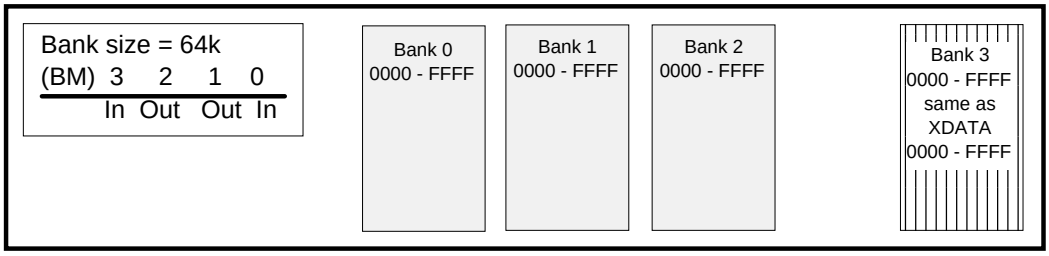


Figure 42. Bank Size 64K (0110b) for All Other Pod Types

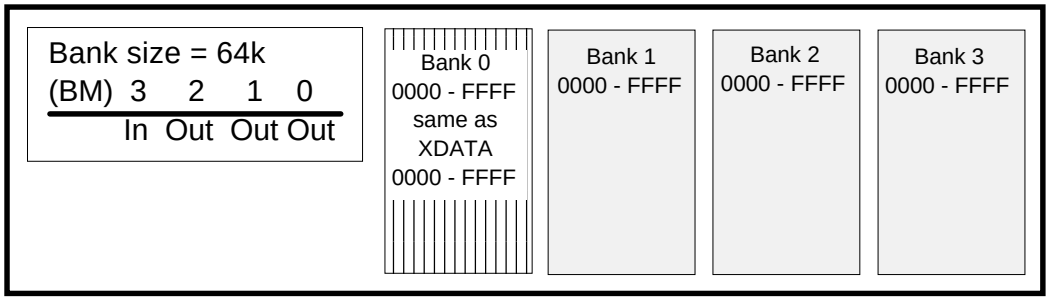


Figure 43. Bank Size 64K (0111b) for All Other Pod Types

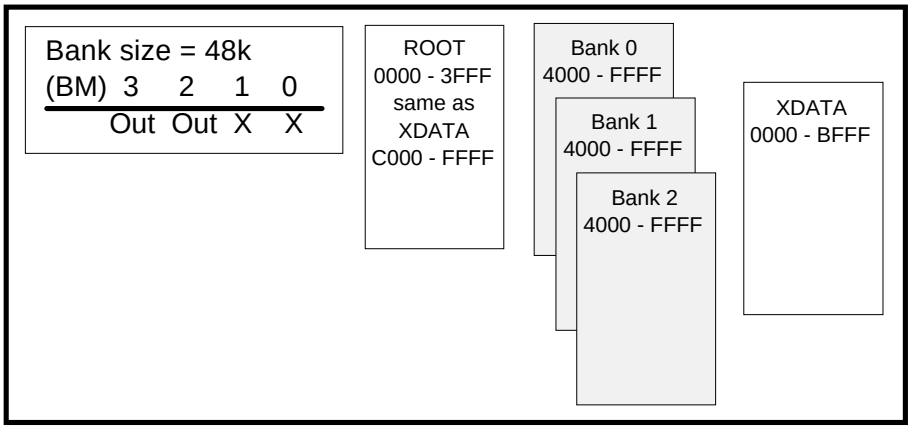


Figure 44. Bank Size 48K (11XXb) for All Other Pod Types

Reprogramming an EPROM for Another Pod Type

- The part required for this board is a Cypress part CY7C266-20WG. You must use this exact part due to the timing required by this board.
- The EPROM hex files are available in the EPROMS.zip file installed in the Seehau51 directory on your hard disk during the installation process.
- You can erase and re-program the EPROM that came in the board or program another part so the part supports both types. Refer to the readme.1st file regarding EPROMs for selecting the proper hex file for your pod type. Go to the following location to find this file: \Nohau\Seehau51\Logic. Open the file called **51eproms.zip**.
- If you do not have a way to program an EPROM, contact Nohau Customer Support to purchase a programmed part.

EMUL51–PC / EA768 - 768K Bank Switch Emulator Configuration

This board comes with bank switch support for up to 768K memory and also supports the Dallas processors with speeded-up cycle times (DS80C320, DS87C520, andDS87C530) and other pods with special requirements.

This board can be configured for particular modes of operation. The following is a partial list of the standard EPROMs that will be used. The EPROM labeled COM1.4 is used for all pod boards with Intel MC551 architecture; COM1.46 is used specifically for the POD–C320; and COM1.47 is used for the POD–C520 and POD–C530.

To reconfigure for each of the different modes, refer to the following jumpers (Figure 45).

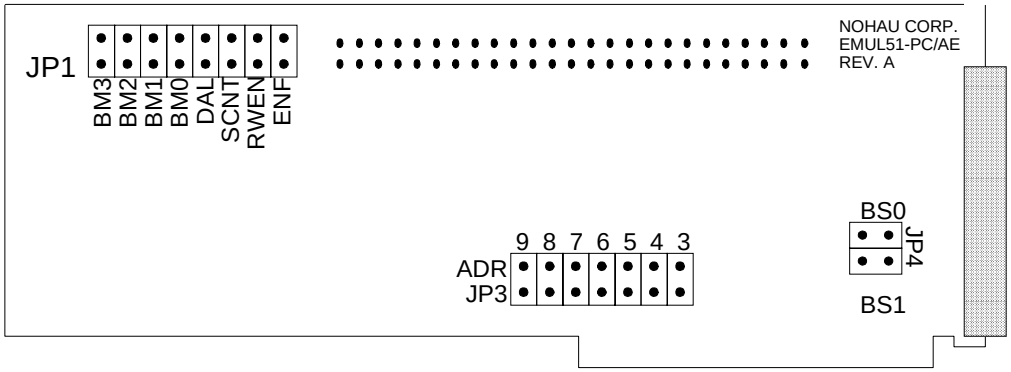


Figure 45. EMUL51–PC/EA768 Header Positions

Description of Jumpers

JP3	ADR	Used for addressing jumpers and functions the same as the Standard Emulator Board.
JP4	BS0–BS1	These jumpers should not be installed when using bank switching.
JP1	BM0–BM3	Used to set the memory mode for the emulator. (See the "(BM) Jumpers" section).
JP1	DAL	Selects the special mode for the Dallas MCUs: 320, 520, and 530 only. Installed = Dallas, Removed = Normal Also remember to change the EPROM to match the type of pod being used. **Note**: Currently the I/O feature is not supported for the Dallas 80C320.
JP1	SCNT	Install this jumper if you are using a pod board that supports DMA (80C152 and 80C452).
JP1	RWEN	When removed, allows you to use the R/W lines on the MCU as general I/O. This mode will work with the POD–31A only. This mode is not supported for the Dallas 80C320 (POD-C320). If you use this mode, remove the RE and WE jumpers on the pod board and start the emulator, selecting one of the following pods: POD–C52, POD–C51FC, or POD–C528. Select a pod based on the MCU that you have installed in the pod. This allows the operation of R/W lines as I/O. If you are using the POD–C520, you must remove this jumper. Refer to Chapter 3, "Installing the Pod Board" for specific information.
JP1	ENF	For factory use only. Leave jumper out.

Bank Switch / Memory Modes

X = Don't care OUT = Jumper out IN = Jumper in

(BM) Jumpers

3	2	1	0	Description	Set Emulator Type To
IN	IN	IN	IN	No bank switching; 64K CODE + 64K XDATA in separate memory areas.	EA768
IN	IN	X	OUT	No bank switching; 64K CODE with 64K XDATA overlaid on code memory.	EA768 overlay

Bank Size = 32K

3	2	1	0	Description	Set Emulator Type To
OUT	IN	IN	IN	32K in root page (0000 – 7FFF) + three pages in upper memory (8000 – FFFF) + 64K of XDATA memory. First upper 32K page is the same as root page (Figure 46).	EA768
OUT	IN	IN	OUT	32K in root page (0000 – 7FFF) + three pages in upper memory (8000 – FFFF) + 64K of XDATA memory. Last upper 32K page is the same as root page (Figure 47).	EA768
OUT	IN	OUT	IN	32K in root page (0000 – 7FFF) + seven pages in upper memory (8000 – FFFF). First upper 32K page is the same as root page. Also, 64K of XDATA memory (Figure 48).	EA768
OUT	IN	OUT	OUT	32K in root page (0000 – 7FFF) + seven pages in upper memory (8000 – FFFF). Last upper 32K page is the same as root page. Also, 64K of XDATA memory (Figure 49).	EA768
OUT	OUT	X	IN	32K in root page (0000 – 7FFF) + 15 pages in upper memory (8000 – FFFF). First upper 32K page is the same as root page. Also, 64K of XDATA memory (Figure 50).	EA768
OUT	OUT	X	OUT	32K in root page (0000 – 7FFF) + 15 pages in upper memory (8000 – FFFF). Last upper 32K page is the same as root page. Also, 64K of XDATA memory (Figure 51).	EA768

Bank Size = 64K

3	2	1	0	Description	Set Emulator Type To
IN	OUT	IN	X	Two banks of 64K for code + 64K for XDATA memory; separate memory areas (Figure 52).	EA768
IN	OUT	OUT	IN	Four banks of 64K for code memory + 64K for XDATA memory (Figure 53).	EA768
IN	OUT	OUT	OUT	Eight banks of 64K for code memory + 64K for XDATA memory (Figure 54).	EA768

Reprogramming an EPROM for Another Pod Type

- The part required for this board is a Cypress part CY7C266-20WG. You must use this exact part due to the timing required by this board.
- The EPROM hex files are available in the EPROMS.zip file installed in the Seehau51 directory on your hard disk during the installation process.
- You can erase and re-program the EPROM that came in the board or program a new Cypress part so that both pod types are supported. Refer to the readme.1st file regarding EPROMs for

selecting the proper hex file for your pod type. Go to the following location to find this file: \Nohau\Seehau51\Logic. Open the file called **51eproms.zip**.

- If you do not have a way to program an EPROM, contact Nohau Customer Support to purchase a programmed part.

Refer to the following figures and comments about bank switching memory for both types of pods, POD–31A, or other types. Modified bank switch pods use a different logic than the POD–31A type of pod. Refer to the memory map information for each mode and pod type.

Operation of the Bank Switch Pages

Figure 46 through Figure 54 reflect the logic based on the POD–31A board and Figure 55 through Figure 63 refer to other pods. As you can see, the logic is reversed between the two different types.

> **Note**
>
> POD–31A type refers to POD–C32HF–42, POD–C320–33, and any similar pod built with the POD–31A fabrication.

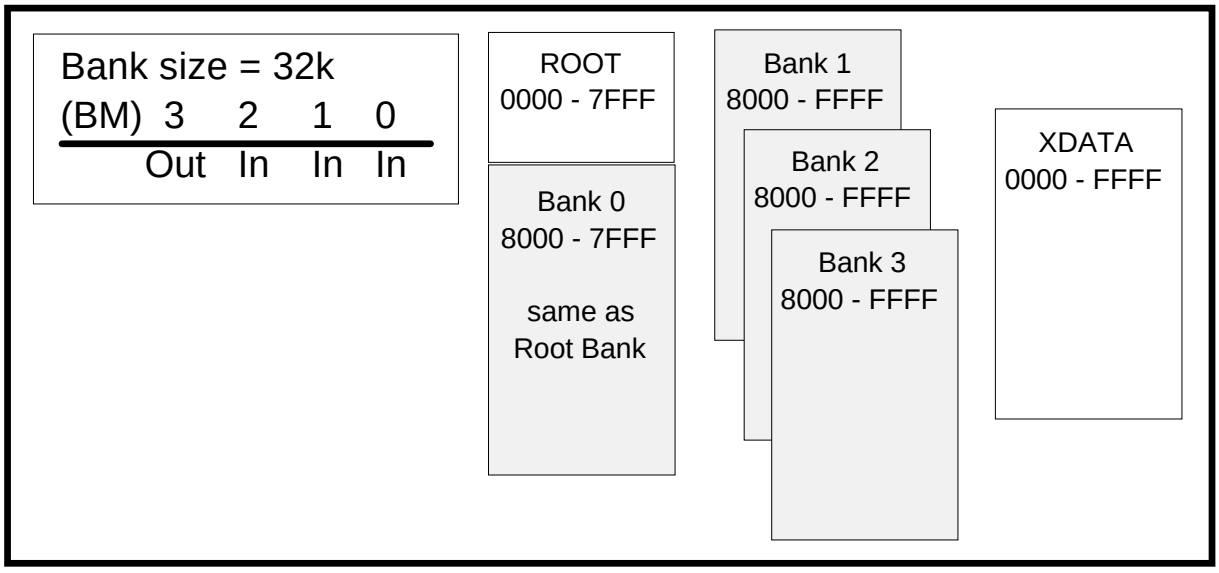


Figure 46. Bank Size 32K (1000b) for POD–31A-Type Pods

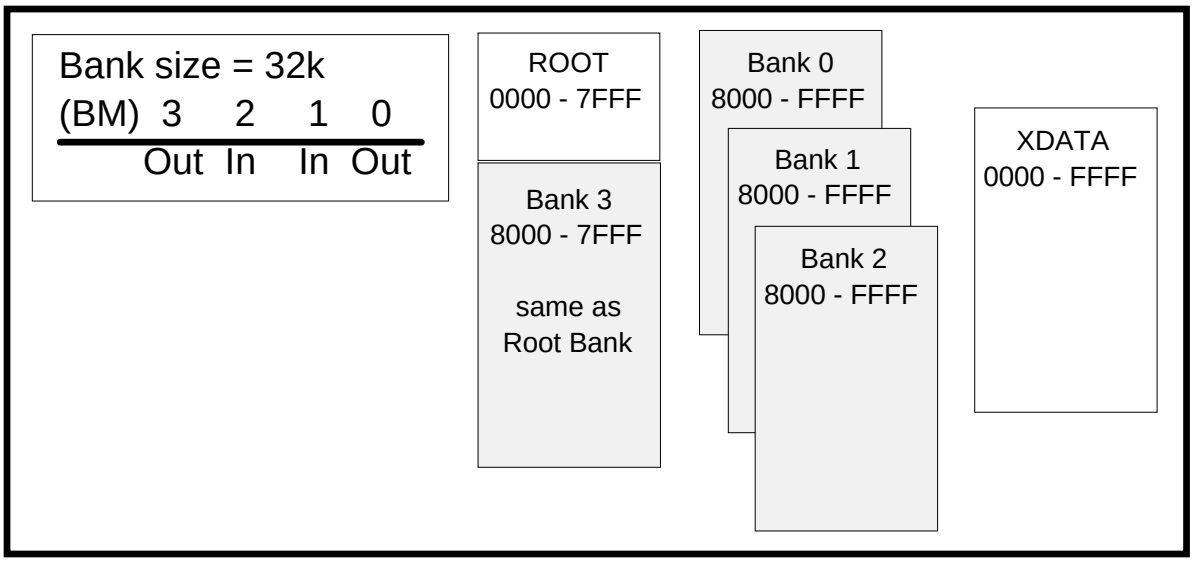


Figure 47. Bank Size 32K (1001b) for POD–31A-Type Pods

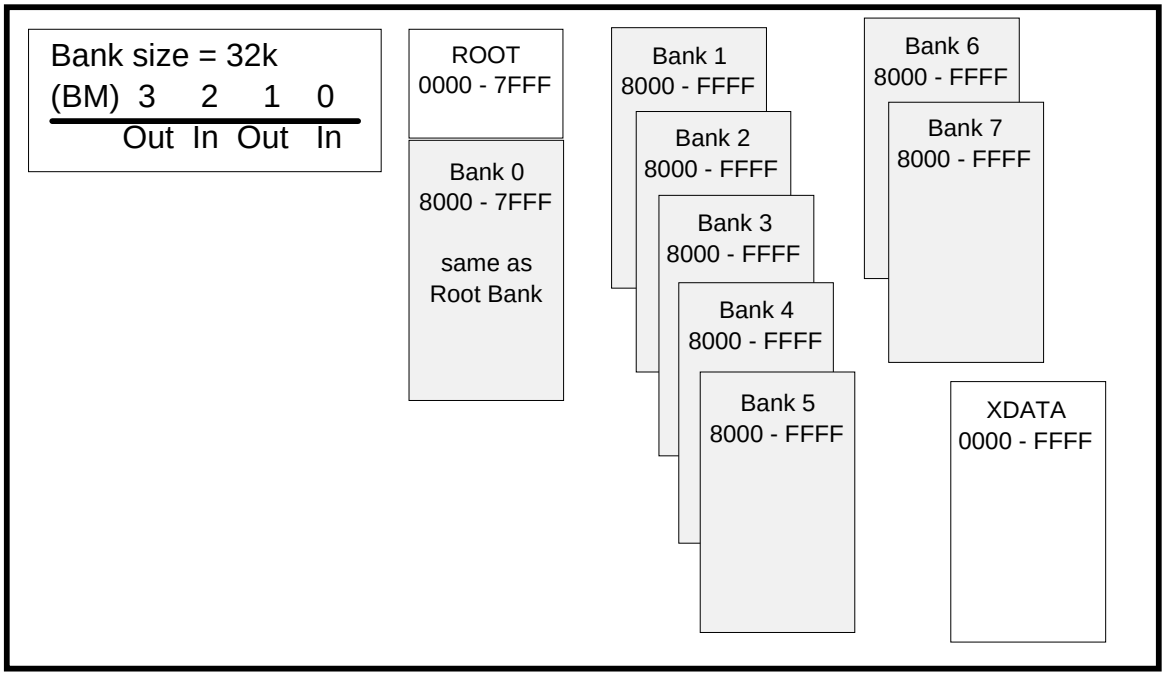


Figure 48. Bank Size 32K (1010b) for POD–31A-Type Pods

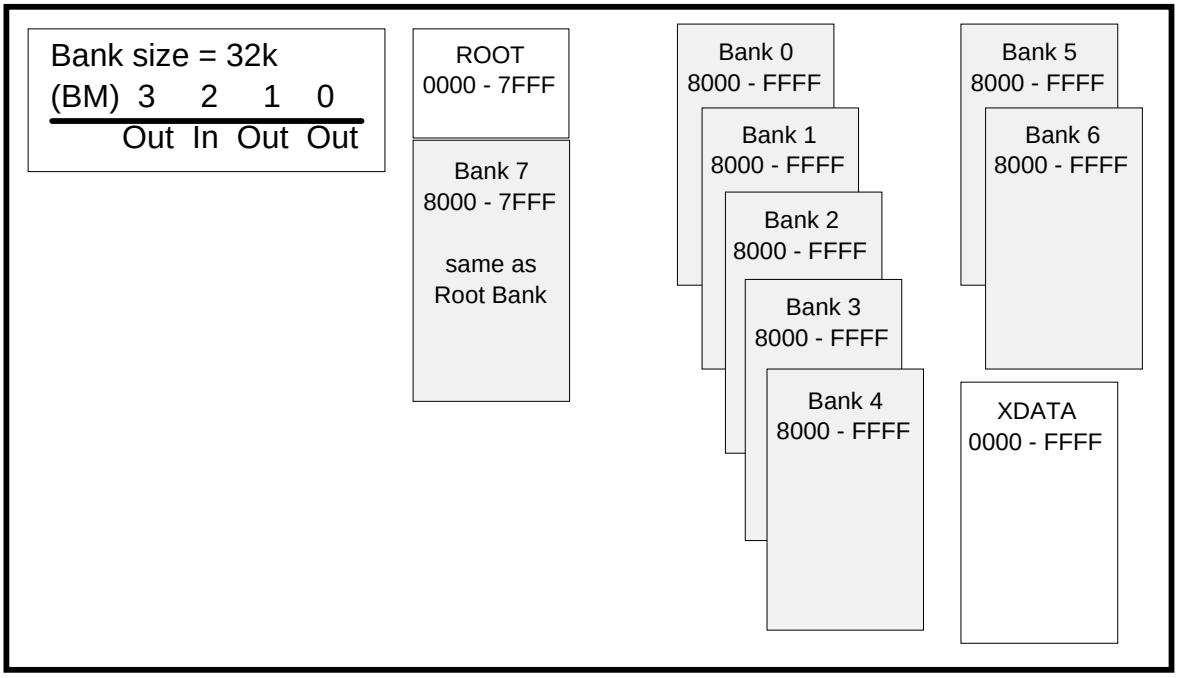


Figure 49. Bank Size 32K (1011b) for POD–31A-Type Pods

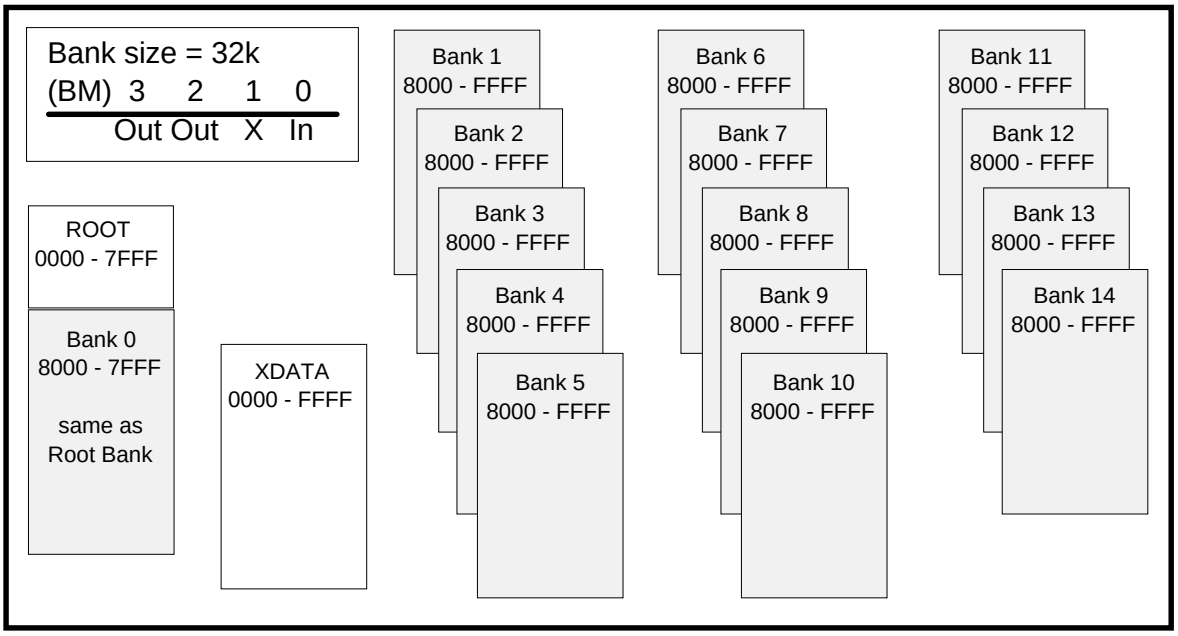


Figure 50. Bank Size 32K (11X0b) for POD–31A-Type Pods

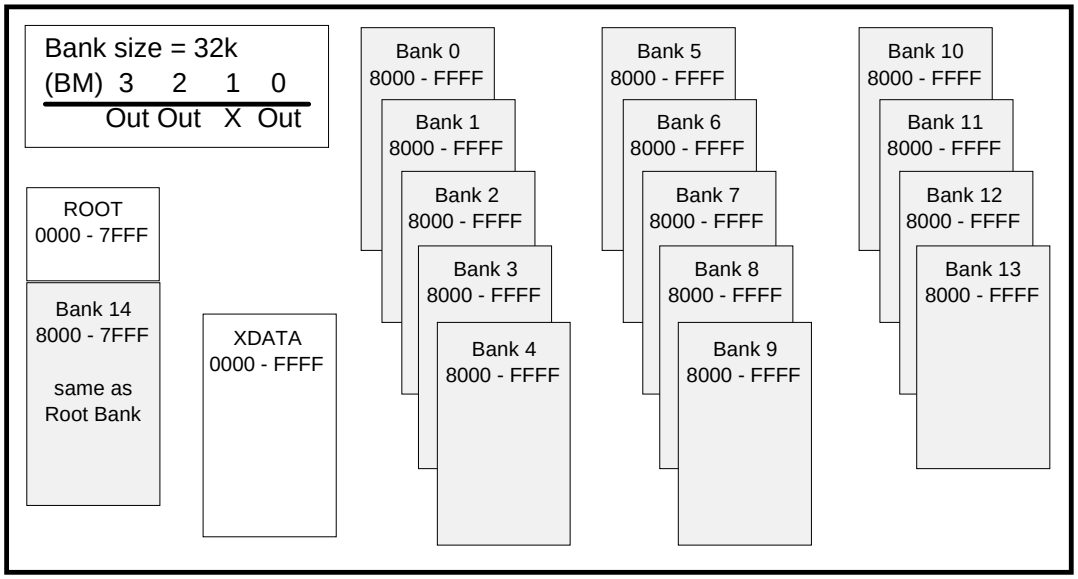


Figure 51. Bank Size 32K (11X1b) for POD–31A-Type Pods

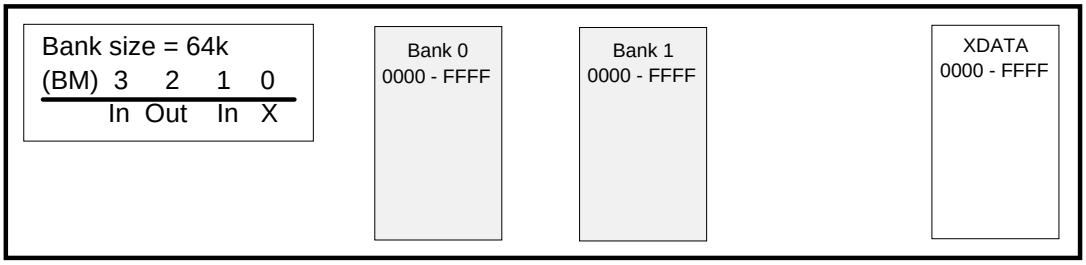


Figure 52. Bank Size 64K (010Xb) for POD–31A-Type Pods

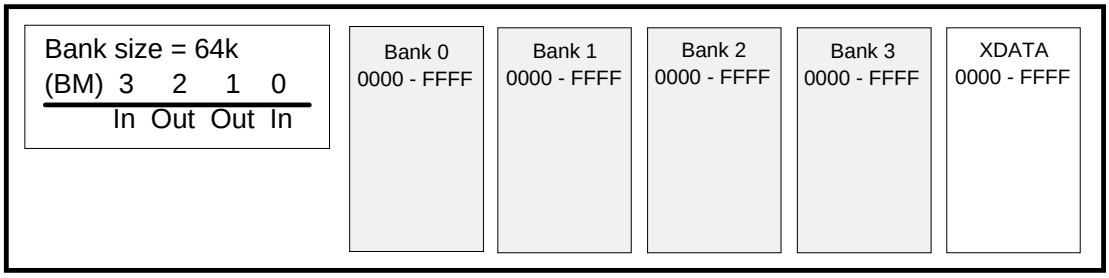


Figure 53. Bank Size 64K (0110b) for POD–31A-Type Pods

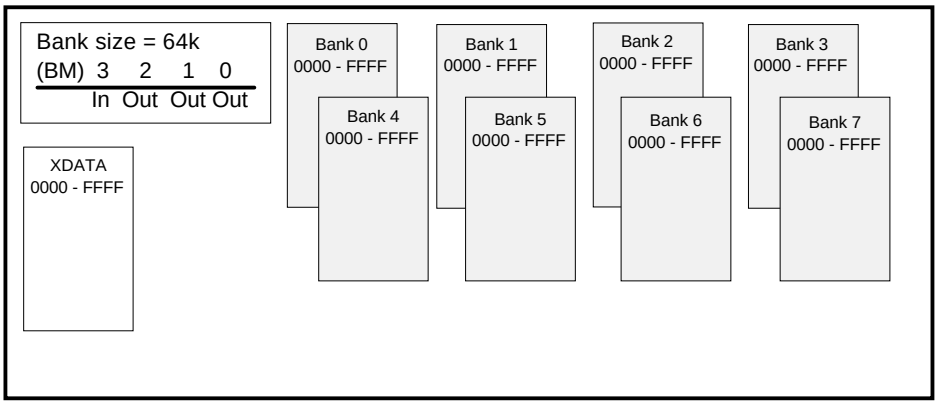


Figure 54. Bank Size 64K (0111b) for POD–31A-Type Pods

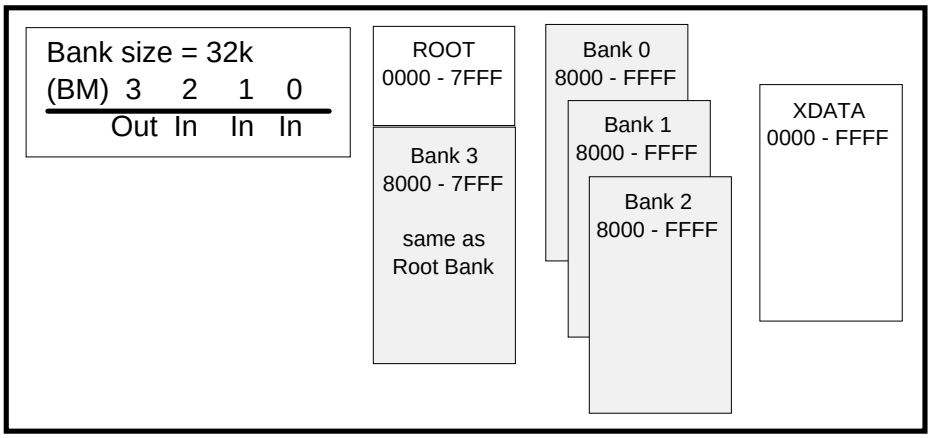


Figure 55. Bank Size 32K (1000b) for All Other Pod Types

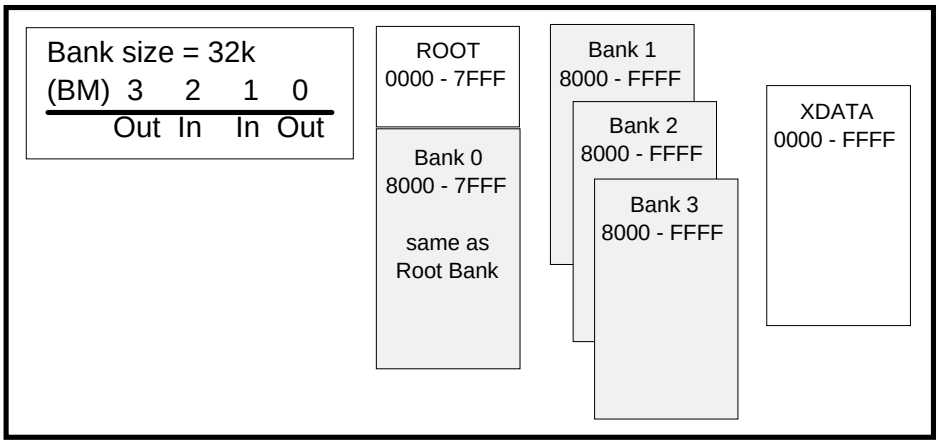


Figure 56. Bank Size 32K (1001b) for All Other Pod Types

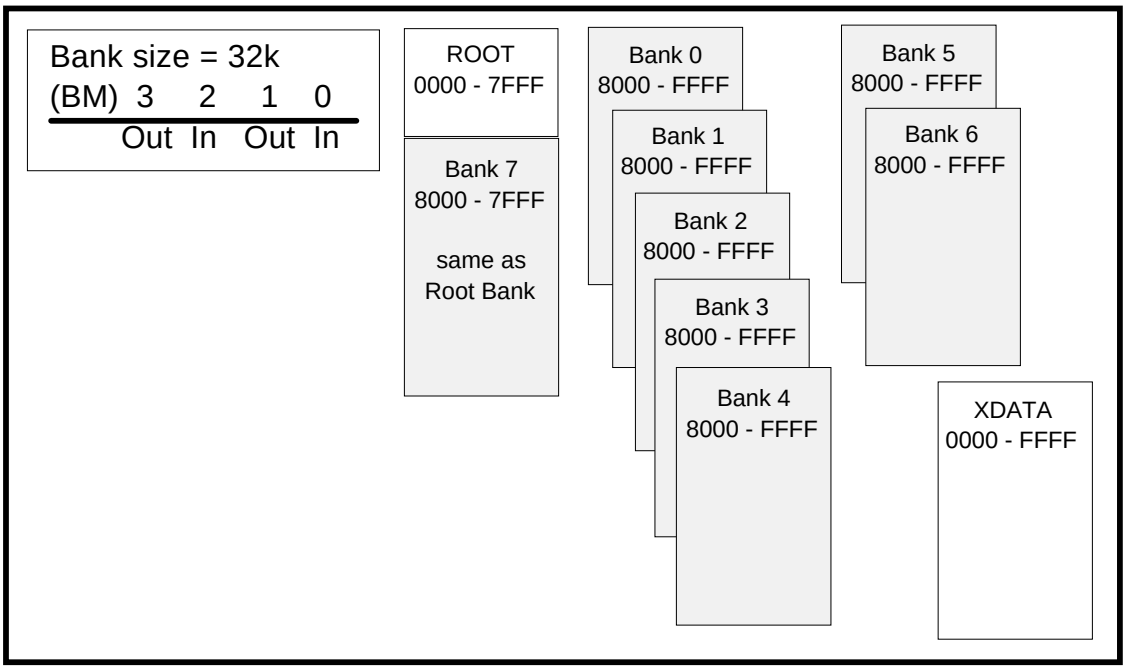


Figure 57. Bank Size 32K (1010b) for All Other Pod Types

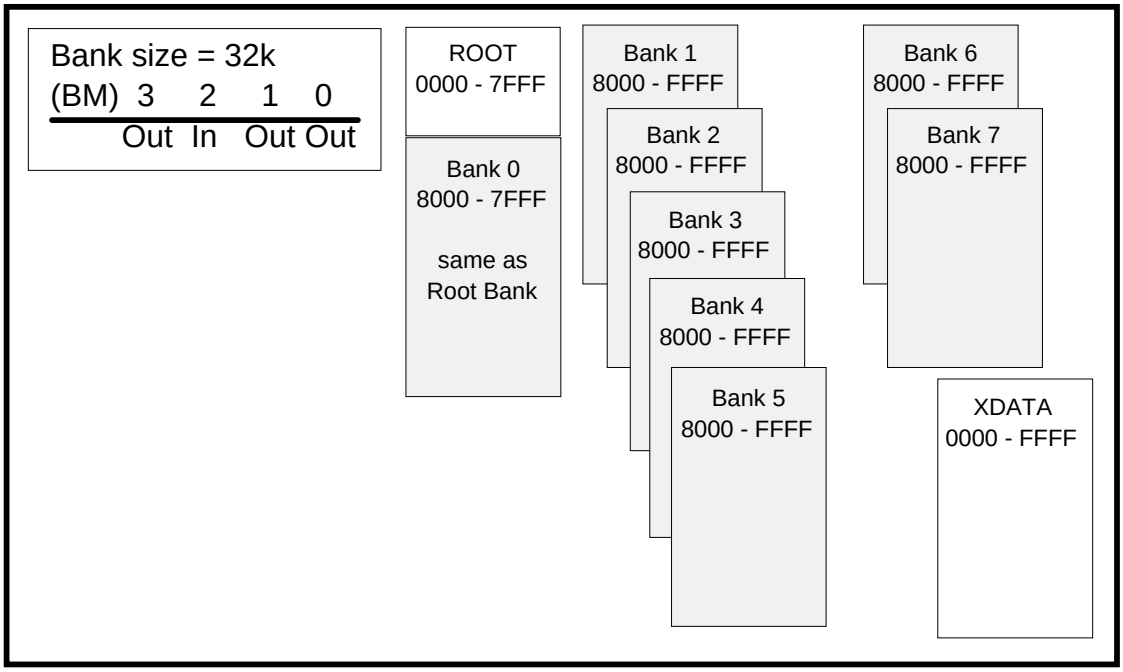


Figure 58. Bank Size 32K (1011b) for All Other Pod Types

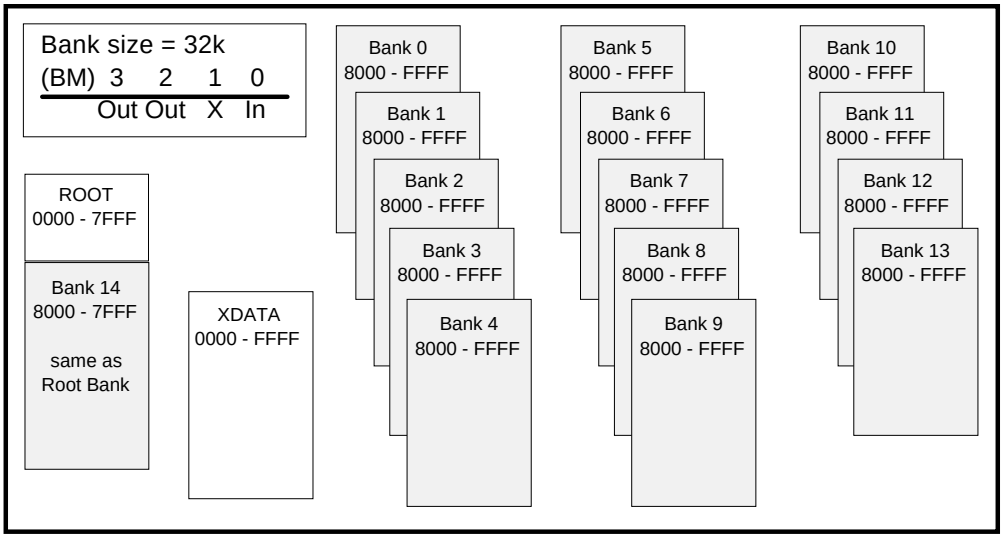


Figure 59. Bank Size 32K (11X0b) for All Other Pod Types

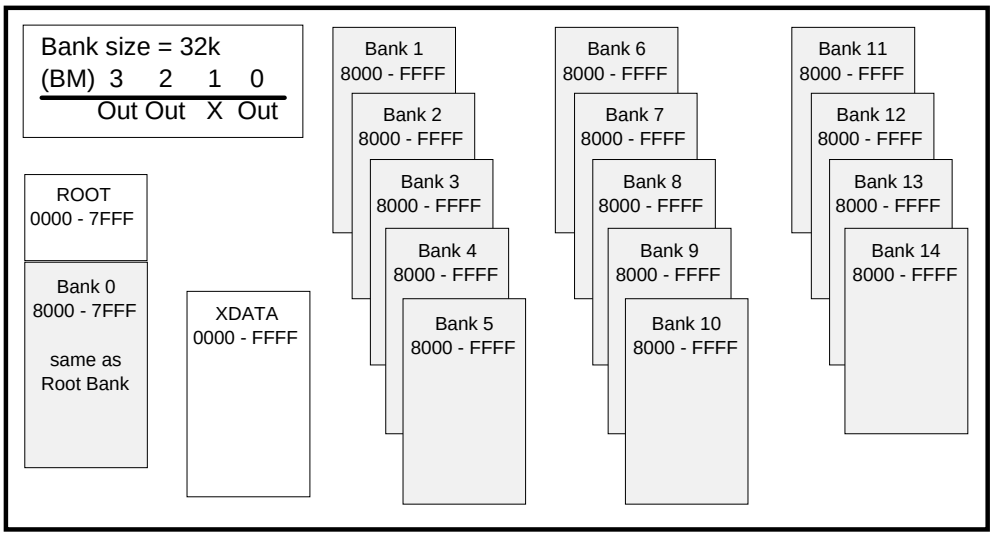


Figure 60. Bank Size 32K (11X1b) for All Other Pod Types

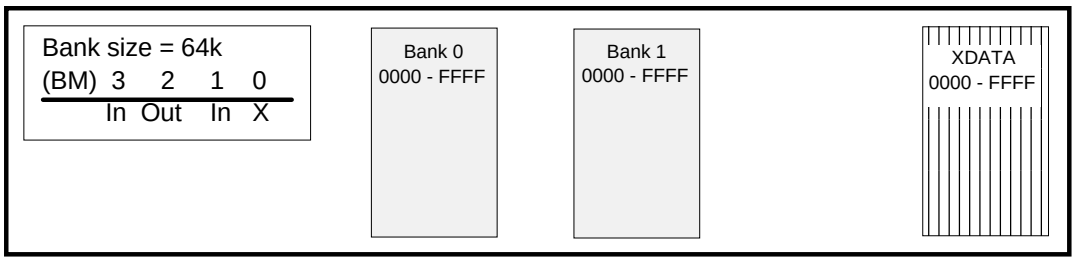


Figure 61. Bank Size 64K (010Xb) for All Other Pod Types

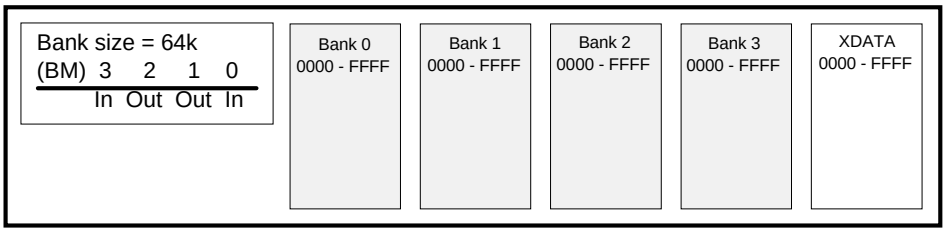


Figure 62. Bank Size 64K (0110b) for All Other Pod Types

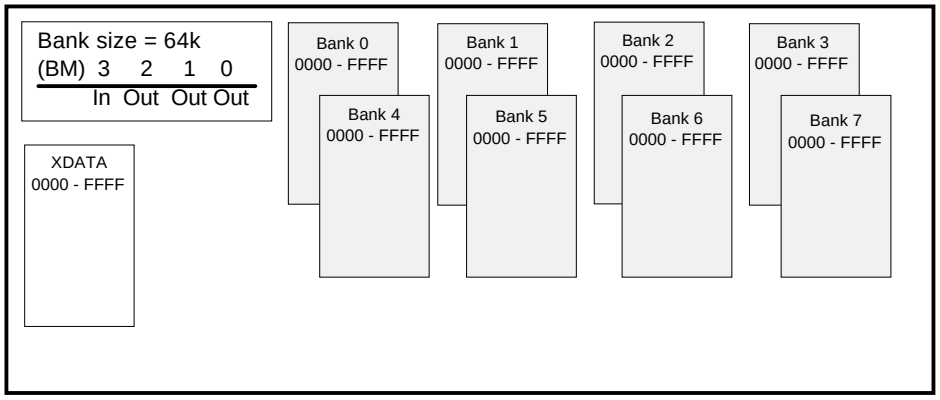


Figure 63. Bank Size 64K (0111b) for All Other Pod Types

Installing and Configuring Standard Trace Boards

Trace is an optional part of an emulator system that allows you to perform more advanced debugging. The standard trace board is a full-length card that can occupy any ISA slot as long as the ribbon cable can reach from the emulator card to the trace card.

The standard trace board includes 48 bits of RAM for each trace record and can hold either 4096 or 16384 records.

I/O Address

Installing the Standard Trace Board in an HSP or USB Box

If you purchased the HSP or USB box, you can use the default address (100H) regardless of the I/O addresses used by the computer. Skip to the “Steps for Installing the Standard Trace Board” section.

Installing the Standard Trace Board with a PC

The trace board address jumpers have been factory preset to 100H. If your system currently uses location 100H, you must find an alternate address location and make appropriate changes to the jumpers and software.

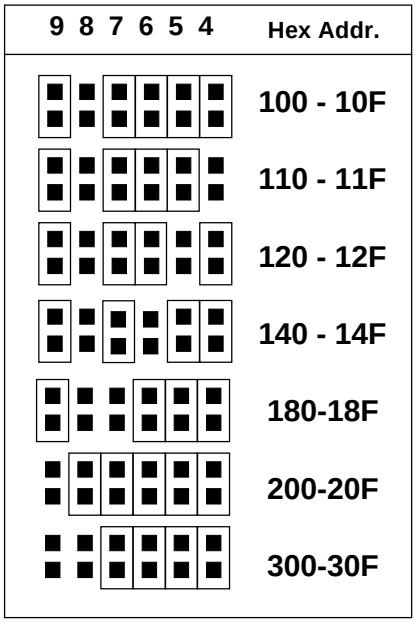


Figure 64. Alternate Standard Trace Addressing Examples

Alternate Addressing Examples

Figure 64 shows several examples of how to set the jumpers on header J2. The jumper columns are arranged the same as board jumpers. For a complete list of addressing examples, refer to Appendix A, “Address Examples.”

Steps for Installing the Standard Trace Board

Be sure to inspect the boards for any damage. Then with the address jumpers in place, you can install the trace board in your PC, HSP, or USB by doing the following:

1. Turn the power off. Power must always be off when you plug in any PC board.
2. Insert the trace board with the short ribbon cable connected to it.
3. Screw the bracket down to the PC chassis.
4. Connect the ribbon cable to the emulator board. Make sure that both rows of the trace cable are connected to and properly aligned with the trace header on the emulator board.
5. Check to ensure the connector fingers of the board(s) are secured into the female connector of the PC’s motherboard.

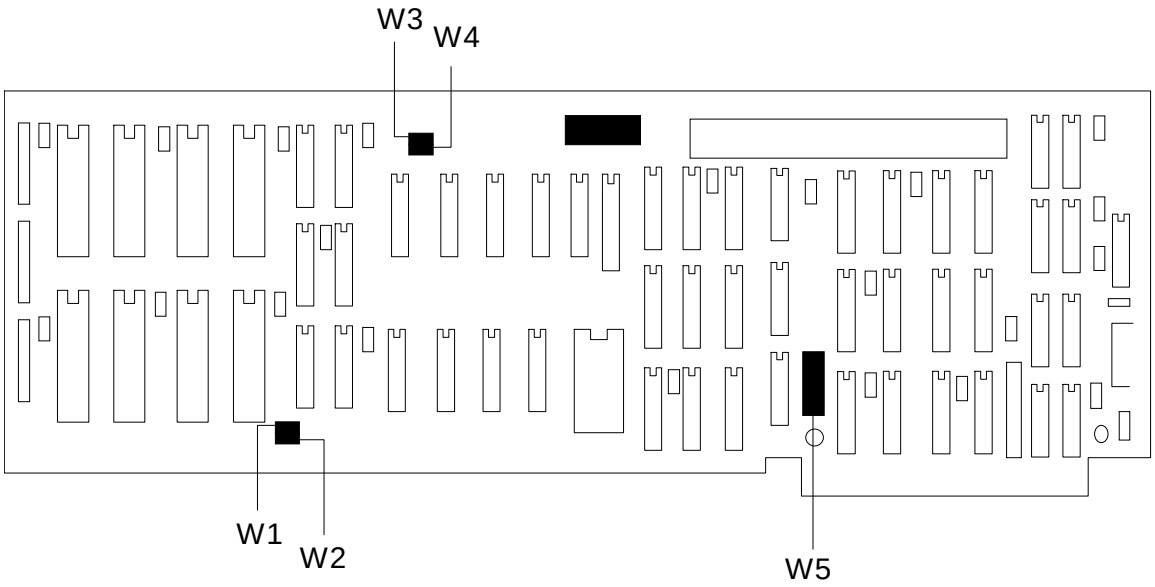


Figure 65. Locations of Jumpers on the Trace Board

Standard Trace Board Description

On the trace board the locations of jumper pins are designated on the board artwork as W1, W2, W3, W4 and W5. (See specified rectangles in Figure 65.)

Standard Trace Board Memory Configuration

Both the TR4 and TR16 trace boards are shipped in one of two memory configurations. There is either one large memory soldered into the bottom row at position U42, or there are four small memories soldered into the bottom row.

The one-memory version has the lower jumper, W2, connected. The four-memory version has the upper jumper, W1, connected. Both variations are functionally the same. Since the W1/W2 jumper depends on the memories soldered at the time of manufacture, the jumper does not get changed even if the trace frame size is changed.

W1/W2 Setting	U39 – U41	U42
W1	8K x 8	8K x 8
W2	Empty	32K x 8

These two factory built configurations both have the same function and size.

4K Trace

For a 4K trace board, 8K by 8 RAM chips are inserted in sockets U3, U4, and U5. The I/O port address is set for 100H (Figure 66).

16K Trace

For a 16K trace board, 32K by 8 RAM chips are inserted in sockets U3, U4 and U5. The I/O port address is set for 100H. (See Figure 67.)

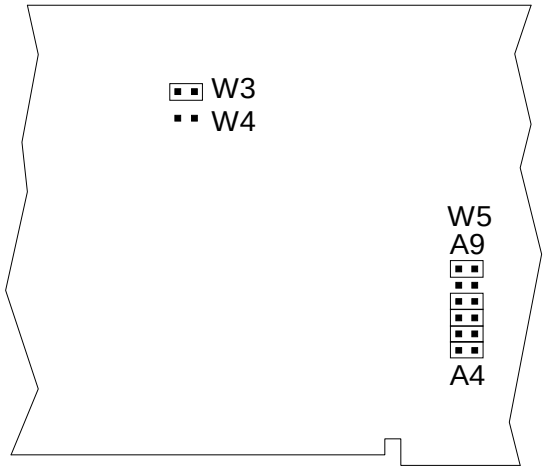


Figure 66. Trace Board with 4K Installed, Address 100H

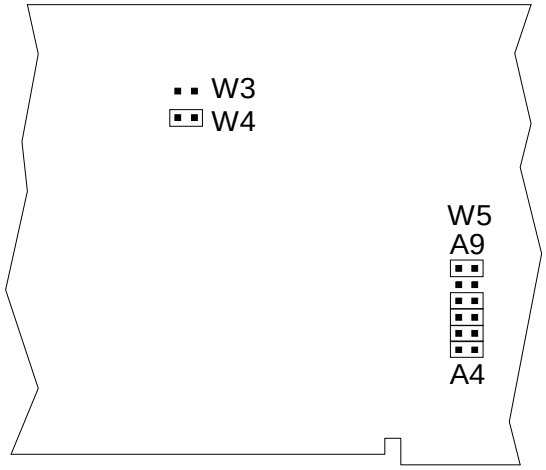


Figure 67. Trace Board with 16K Installed, Address 100H

Advanced Trace Board

- The ATR64 trace board has a memory configuration with the jumper top on the right (Figure 68).
- The ATR256 trace board has a memory configuration with the jumper top on the left (Figure 69).

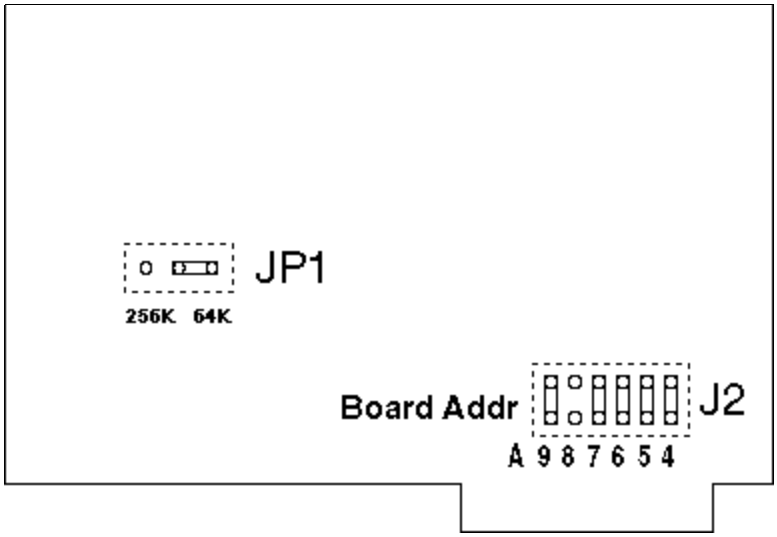


Figure 68. 64K Advanced Trace Board Jumpers and Headers

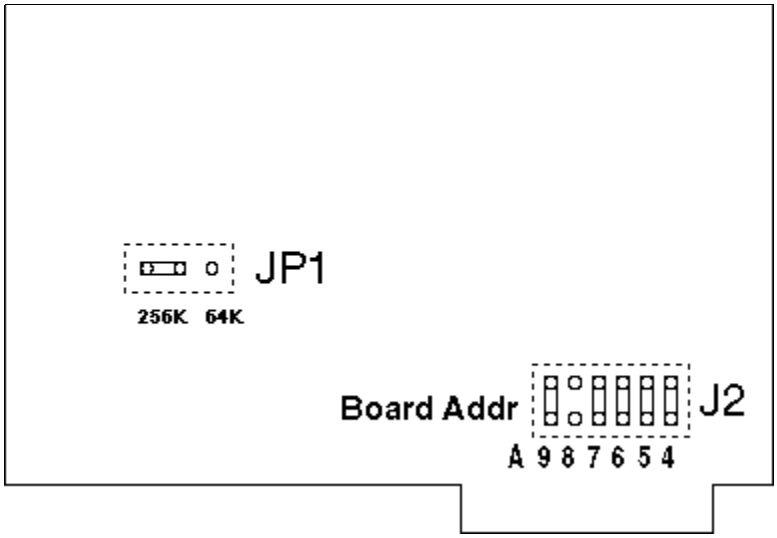


Figure 69. 256K Advanced Trace Board Jumpers and Headers

Installing and Configuring Enhanced Trace Boards

The enhanced trace board includes 64 bits of RAM for each trace record and can hold either 64K or 256K records.

Addressing Examples

Refer to Figure 70 for examples of how to set the jumpers on header JP1. For a complete list of addressing examples, refer to Appendix A, “Address Examples.”

Factory Settings

Board jumpers have been preset at the factory to 120H prior to shipment. Compare the jumper configurations on the boards you receive against the configurations described in Figure 71. If a jumper is installed other than as shown, refer to these figures before changing the jumper’s position.

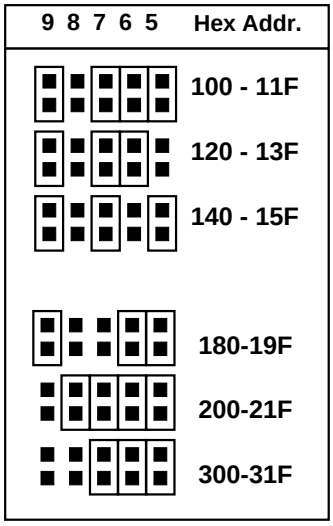


Figure 70. Enhanced Trace Addressing Example

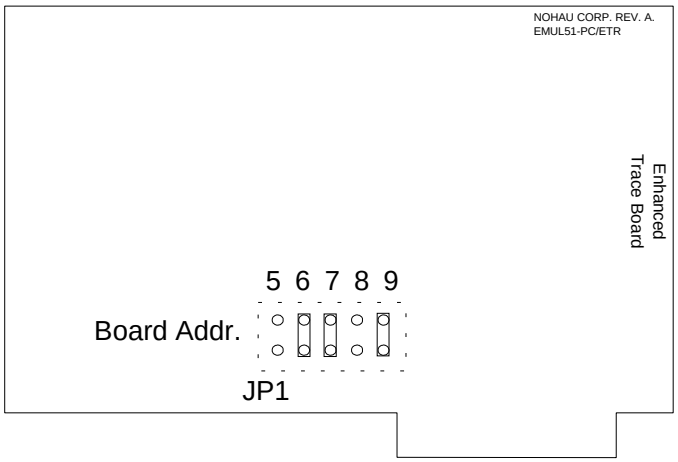


Figure 71. Enhanced Trace Board Jumpers and Headers

Steps for Installing the Enhanced Trace Board

Be sure to inspect the boards for any damage. Then with the address jumpers in place, you can install the trace board in your PC, HSP, or USB by doing the following:

1. Turn the power off. Power must always be off when you plug in any PC board.

1. Insert the trace board with the short ribbon cable connected to it.

2. Screw the bracket down to the PC chassis.

3. Connect the ribbon cable to the emulator board. Make sure that both rows of the trace cable are connected to and properly aligned with the trace header on the emulator board.

4. Check to ensure the connector fingers of the board(s) are secured into the female connector of the PC's motherboard.

3 Installing the Pod Board

Pod Boards

How this Chapter is Organized

Pod types are listed in alphabetical order. Each pod section presents information in the following format:

- **Configuration Options**: Describes configuration options.
- **Illustrations**: Shows various configurations for switches and jumpers, target power and internal crystal. The illustrations throughout this chapter are representative of the pod board layout. The notations used in the illustrations might not match the silk screens on the boards.
- **Special Considerations**: Provides specifics about the pod's features and functions.

Important Notes About Pod Boards

Read the following notes first before installing your pod board.

Remove Black Conducting Foam Before Using Your Pod

When using your pod in stand-alone mode, be sure to remove any black conducting foam. This foam is usually inserted at the factory to protect pins that mate with a target adapter or a socket on your target board. The foam covers pins which protrude from the bottom of the pod or from an adapter attached to the pod. The pod will not work with the conducting foam attached and might cause damage.

If you remove the pod from your target socket or target adapter and plan to store it, you should re-install the conducting foam to protect the exposed pins.

Limited Warranty on Bondout Chips

The bondout chip is used to emulate the MCU. Damage to expensive bondout chips is not covered under warranty if the proper power-up and power-down sequences are not followed. (Refer to the "Power-up / Power-down Sequences for External Mode Pods" section in Chapter 5, "Connecting the Emulator to Your Target Board.")

Memory Mapping Requirements for On-Chip XData

For those chips that have on-chip XDATA, you need to map that region of memory to the target. An exception to this is with the POD–C520. On this pod, the on-chip XDATA is disabled on the emulation chip which requires that you map that range to the emulator's memory.

Selecting Jumpers for On-Chip XData

When using a standard POD–31 or POD–31A, some users have reported that if they place a Philips RA/B/C/D+ MCU on the pod, the system does not operate.

Due to the nature of this MCU with the on-chip XDATA memory, use the following solution to make this configuration work.

Requirements

- Emulator Board: EA256/768 emulator board
- Pod: 31 or 31A
- Installed micro: Philips Rx+

Do the following steps:

1. Remove the RWEN jumper on the emulator board.

2. Start the emulator. Select one of the following, if available, in the current version of software you are using:

  – For POD type selections: in EMUL51-DOS or Windows software, select processor type.
  – For Seehau51: POD–C51RX

> **Note**
>
> On the POD–31A PCB, refer to the pod doumentation and adjust the jumpers to make P3.6/7 work as I/O. On the generic POD–31, P3.6 and P3.7 will not operate as I/O.

Generic Pod Boards: External Mode

This section covers the following pods:

POD–31	POD–C32
POD–C31	POD–C252/C51FA
POD–32	

Pod Identification

The pod has a 40-pin DIP socket (Figure 72). The silk-screen is labeled “Pod Board.”

Plug any of the following MCUs into a 40-pin socket on the generic board:

- **POD–31** for 8031
- **POD–C31** for 80C31
- **POD–32** for 8032
- **POD–C32** for 80C32
- **POD–C252/C51FA** for 80C51FA

These pods will all run up to 12 MHz. In addition, the higher frequency pods are guaranteed to run up to their rated frequencies.

Two jumpers make it possible to use an external crystal or clock. Power to the MCU chip can be taken either from the target system or from the pod. (The pod logic receives its power from the PC).

Since you can exchange MCUs on the generic pod board, you can use the same board to emulate any of these MCUs.

These pods emulate the MCUs only in their external mode: P0 is used as an address/data port, P2 is used as an address port, P3.6 is used as WR, and P3.7 is used as RD.

Generic Pods

Jumper Identification and Description

Designation	Function	Description
JB1	RST Selection	• With JB1 installed, pin 9 on the pod board MCU (RST) is connected to the target system. Pin 9 allows reset to be performed by S1 or the target system. • With JB1 removed, RST is connected only to the S1 switch and the PC controlled reset line. This is useful if you have a software-controlled watchdog that continues to reset the MCU.
JB2	Pod Board Power	This jumper determines the pod board's source of power. • If the jumper is set toward the 8031, power is received from the target. • If the jumper is set in the opposite position, power is taken from the PC (refer to Figure 73).
JB3	Port Connections	Ports P1-0 through P1-7 and P3-0 through P3-5 are connected to the trace board with the jumpers in their preset positions. If the jumpers are removed, the upper pins closest to the edge of the board can be connected to external signals with the optional EZ-hooks. These signals are then traced instead of the ports.
JB4	Standby External Signal	These four pins can also be connected to external signals with the optional EZ-hooks. All four signals are traced. Additionally, SY0 is used in breakpoint logic. No jumpers are used.
JB5, JB6	Crystal Selection	With JB6 jumpers installed, the crystal on the pod board is connected to the MCU. When the two jumpers are installled in JB5, the crystal of the target system is connected to the MCU.
JB7	PSEN Signal	In the factory-installed position, the PSEN signal only goes out on the target system in emulation mode. In the alternate position, PSEN will always go out to the target system. In that case, be sure that PSEN is not used to enable any device that drives the databus. (Remove the PROMs.)
DS1	Emulation Status LED	The DS1 LED lights when emulation is running. It is driven by the EM/ signal on TP.
GND Wire	Ground Wire	The GND ground wire connects the pod ground to the target ground before the pod is plugged into the target. When you connect this ground wire, you reduce the chance of electrostatic discharge.

Generic Pods

Board Layout

Figure 72 shows locations of jumpers and Figure 73 shows a diagram of the jumper pins.

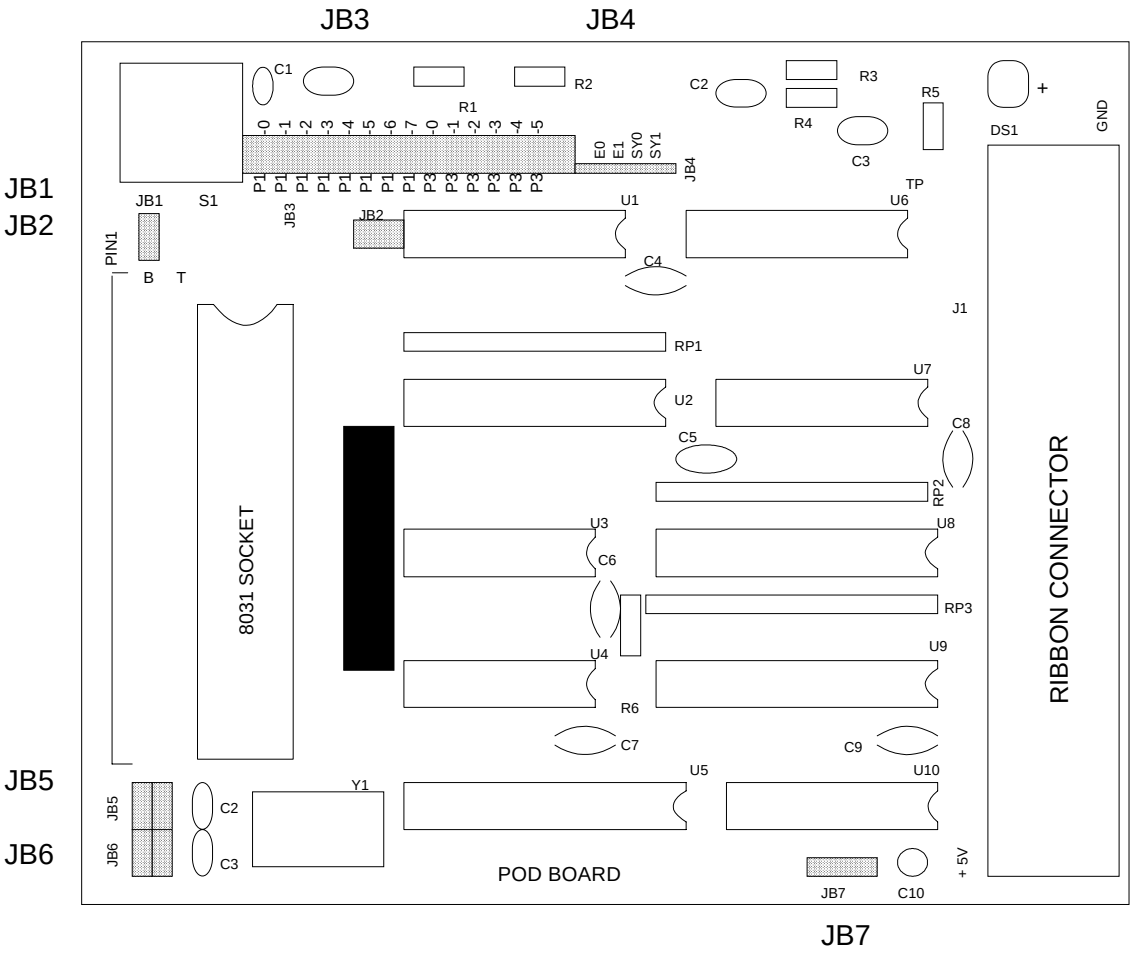


Figure 72. Locations of Jumpers on the POD–31 Board

Generic Pods

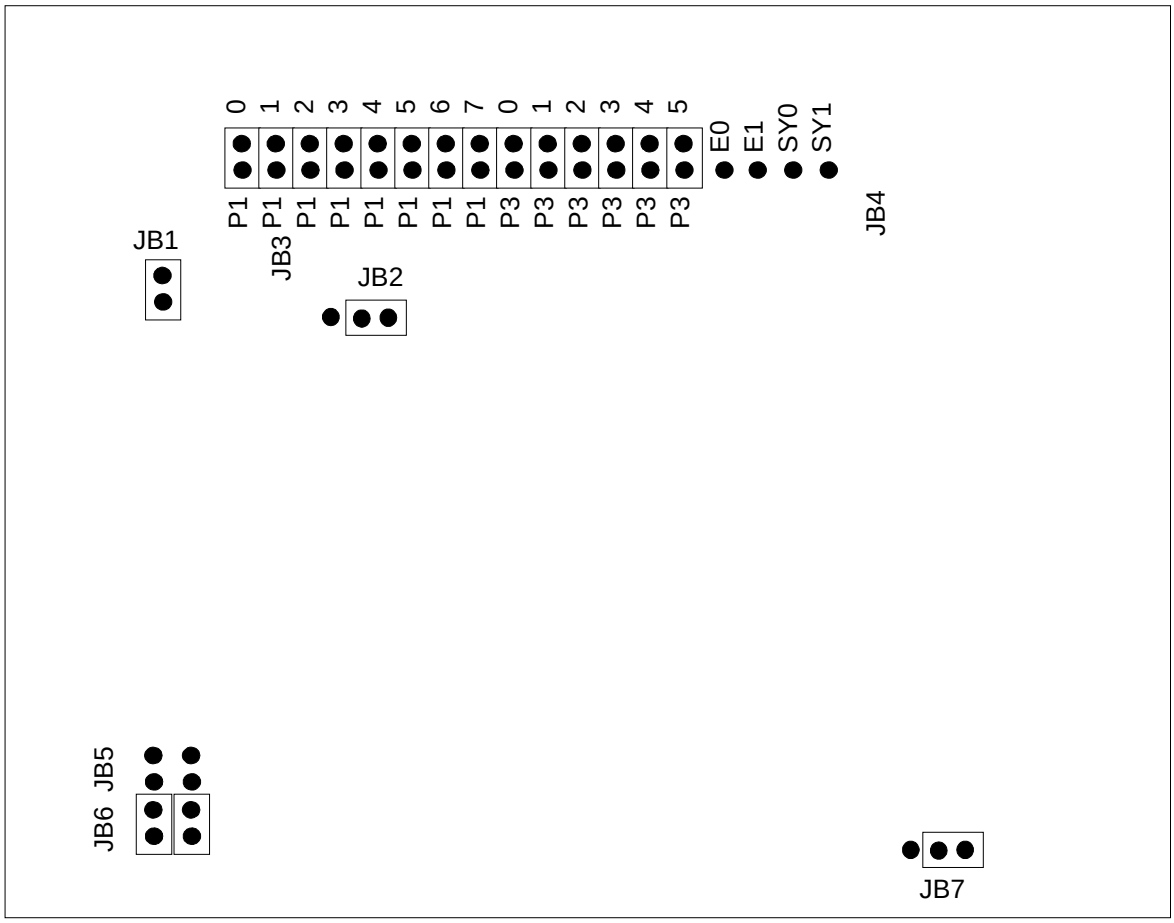


Figure 73. POD–31 Board Configured for Internal Power and Internal Crystal

POD–31S and Variations: External Mode

This pod is a superset of the non-S 40-pin external mode pod. One mode lets P3.6 and P3.7 each be used as either unidirectional input or output. Another mode makes P3.6 the WRITE strobe and P3.7 the READ strobe.

To work with this pod, you need to select **POD–C31** from the **Select Processor** drop-down list box (**Step 3** of the **Communications** tab in the Emulator Configuration window).

The EPROM on the emulator board must be version COM 1.4.

Jumper / Switch Identification and Description

Designation	Function	Description
JB1 or J1	Crystal Selection	• If the two switches are in position INT, the crystal on the pod is connected to the MCU. • In position EXT, the crystal on the target system is used.
JB2	Power Selection	• When this switch is in position INT, the power to the MCU comes from the PC. • In position EXT, power comes from the target system.
JB3, JB4	Port Selection	Ports P1.0 through P1.7 and P3.0 through P3.5 are connected to the trace board when these jumpers are installed. If the jumpers are removed, external signals can be connected to the pins closest to edge of the board with the optional EZ-hooks.
JB5	PORT 3 I/O Control	Move this switch to position 3 when P3.6 and P3.7 are used as WR/ and RD/. When P3.6 and P3.7 are used as I/O, JB5 needs to be in position 1. (See the “Setups” section.)
JB6, JB7	PORT 3 I/O Control	• These switches should be in position 1 when P3.7 is used as WR/ and RD/. • In position 3, P3.6 and P3.7 are used as I/O. (See the “Setups” section.)
JB8	PSEN Signal	• If this switch is in position 1, the PSEN/ signal only goes out to the system in emulation mode, and the PC is accessing an address mapped to target. • If the switch is in position 3, the PSEN signal always goes out to the target system. In that case, be sure that PSEN/ does not enable any device (such as a PROM) that drives Port 0, the data bus.
JB9	RST Selection	• With this jumper installed, pin 9 on the pod board MCU (RST) is connected to the target system, which allows reset to reach the MCU from your target system. • With JB9 removed, RST is connected only to the S1 switch and the emulator-controlled reset line. This is useful if you have a software controlled watchdog that keeps re-setting the MCU.

POD–31S

Jumper / Switch Identification and Description (continued)

Designation	Function	Description
JB10	PORT 3 I/O Control	• If the jumper is in the position nearest the edge of the PCB, then P3.6 is WR/. • If the jumper is in the opposite position, then P3.6 is an OUTPUT. • If no jumper is installed, P3.6 is an INPUT. (See the “Setups” section.)
JB11	PORT 3 I/O Control	• If the jumper is in the position nearest the edge of the PCB, then P3.7 is RD/. • If the jumper is in the opposite position, then P3.7 is an OUTPUT. • If no jumper is installed P3.7 is an INPUT. (See the “Setups” section.)
S1	RESET	This switch only resets the pod's MCU, not the target system.
J2	Emulator Status	This pin is at a low state during emulation, which is also indicated by the LED.
J4	GND	This ground wire is to connect the pod ground to the target ground before the pod is plugged into the target. Connecting this ground reduces the chance of electrostatic discharge.
J5	SY0, SY1, E0, E1	These pins can be used to trace external signals.
DS1	Emulator Status LED	The DS1 LED lights when emulation is running. It is driven by the EM/ signal on TP.

Setups

Possible Setups

P3.7	P3.6	JB5	JB6	JB7	JB10	JB11
RD/	WR/	Pos 3	Pos 1	Pos 1	Edge	Edge
Output	Output	Pos 1	Pos 3	Pos 3	Middle	Middle
Input	Input	Pos 1	Pos 3	Pos 3	Remove	Remove
Output	Input	Pos 1	Pos 3	Pos 3	Remove	Middle
Input	Output	Pos 1	Pos 3	Pos 3	Middle	Remove

Setups Not Possible

P3.7	P3.6
RD/	I/O
I/O	WR/
Bi-directional	

POD–31S

Board Layout

Figure 74 shows locations of jumpers and Figure75 shows a diagram of the jumper pins.

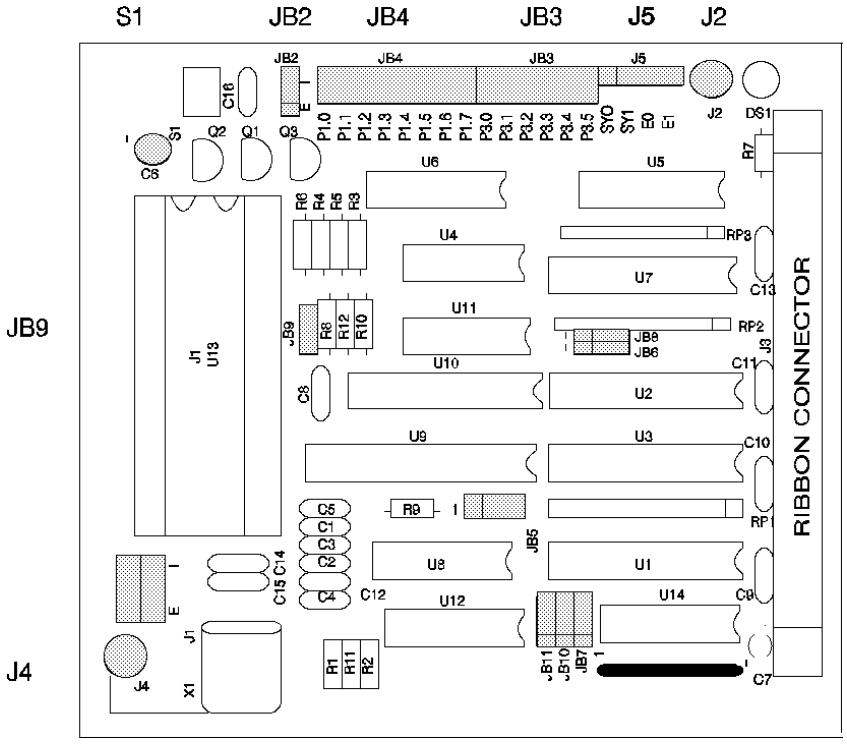


Figure 74. Locations of the Jumpers on the POD–31S Board

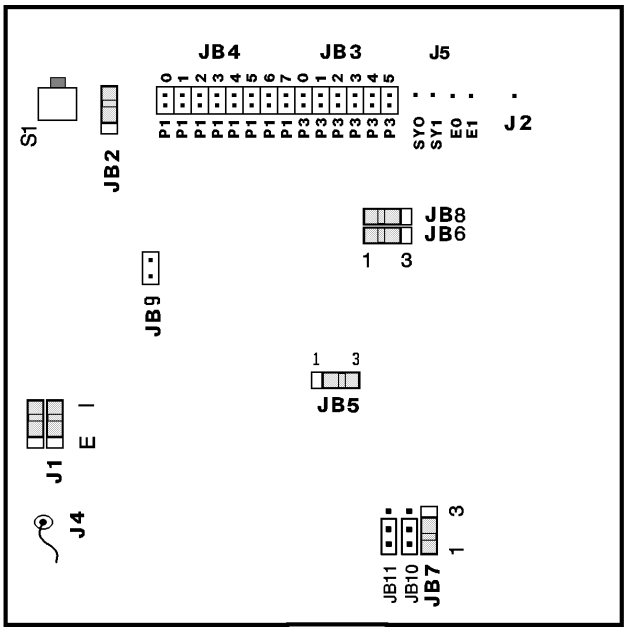


Figure 75. POD–31S Board Configured for Internal Power and Internal Crystal

POD–51EH–C504–40

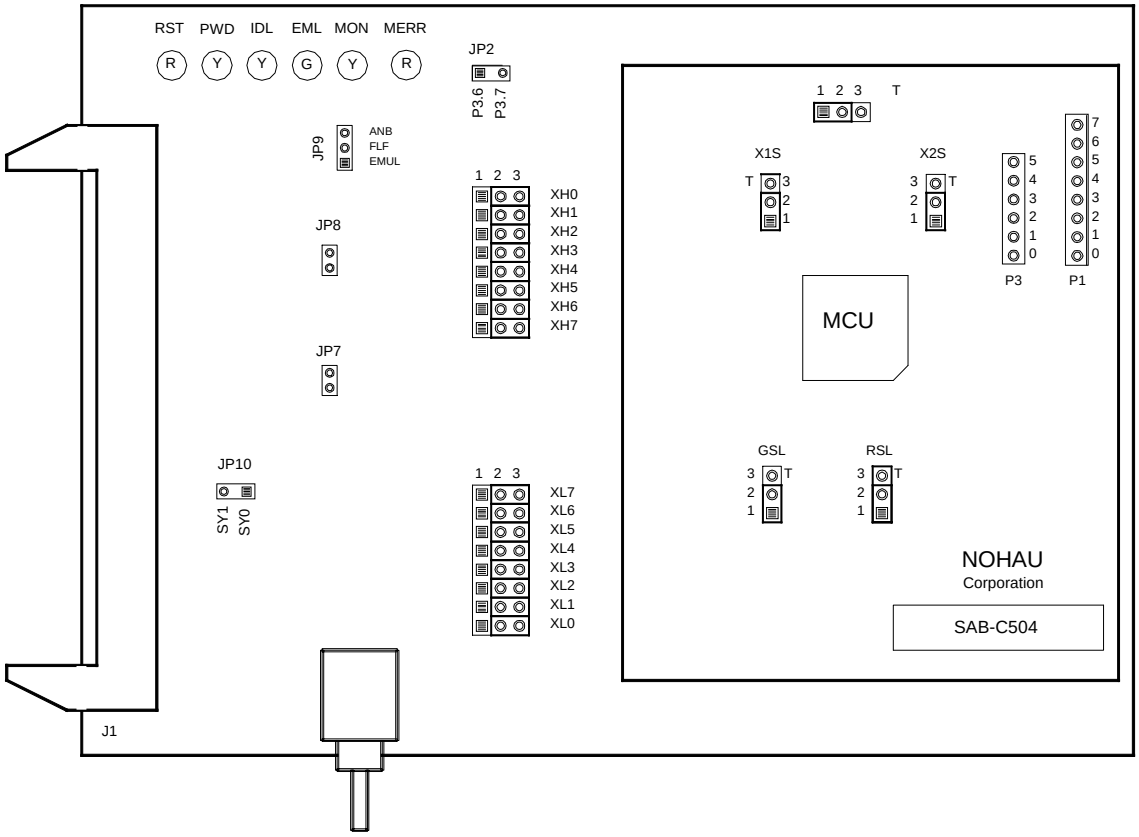


Figure 76. POD–51EH–C504–40

Operating Instructions

The POD–51EH–C504–40 supports the Siemens C504 MCU which is based on the C500 core.

To work with this pod, you need to select **POD–EH–C504** from the **Select Processor** drop-down list box (**Step 3** of the **Communications** tab in the Emulator Configuration window).

The EPROM on the emulator board must be version COM 1.4.

The pod assembly consists of two sub-assemblies: The larger universal Port Replacement Unit (PRU) and the smaller processor module daughterboard.

The PRU provides a 44-pin PGA socket on the bottom that you can connect to the target using an adapter. The PRU also has a 50-pin header J1 that connects through the ribbon cable to the emulator board.

POD–51EH–C504–40

The adapters plug into the target system, replacing the C504 MCU IC. The target adapter part number is EDI/44PG/QFS31-SD.

The POD–51EH–C504–40 runs up to a maximum frequency of 40 MHz.

The C504 MCU has on-chip XRAM memory located at addresses FF00H – FFFFH in the data (XRAM) space. If your application uses the on-chip XRAM, you must map the last memory range of the data memory (addresses F000H – FFFFH) to the target.

POD–51EH–C504–40 has two operating modes.

- Single-chip
- External

The active mode is selected by the EA pin (pin 29 of the target adapter) during reset signal rise (also sampled on internal watchdog timer reset). It is recommended that you connect this pin to either constant low or constant high. This pin defaults to high (selecting single-chip mode), which is implemented by a 22K pull-up resistor on the pin.

When using external mode, the code memory can be mapped either to the target or the emulator (with boundaries of 4K). In external mode, addresses are seen on the Port2 and Port0 adapter pins for every bus cycle regardless of where it is mapped. The data, however, is read or written to the Port0 pins only for addresses that are mapped to the target.

When using single-chip mode, you need to map all code memory to the emulator board. In this mode, Port2 and Port0 can be used as I/O ports.

POD–51EH–C504–40 also supports tracing of internal XRAM, both RD and WR cycles, using MOVX instructions. This is supported in both single-chip or external modes.

POD–51EH–C504–40

LED Indicators

Function	Description
RESET	A red LED that indicates the MCU is receiving a reset signal. There are five possible sources of the MCU reset. • Reset from the target starts whenever a low level is detected on the reset pin of the adapter. • Reset from the S1 button starts by pressing the S1 button on the pod. • Click the Reset button in the Seehau menu to reset the emulator. • Reset from the internal watchdog timer starts if the internal watchdog is enabled, and it is not fed by the user software for a long period. (**Note**: It is important to feed the watchdog periodically, otherwise, problems during the debug process might occur.) • When a map error occurs. During all of these reset sequences, the reset LED is turned on for a short or long period depending on the sequence).
Map Error	A red LED that indicates mapping is occurring in one of the following situations: • The internal C504 XRAM is enabled by bit XMAP in the SYSCON register. • The data memory address range F000H – FFFFH is mapped to the emulator. • There is a memory access (MOVX) to the internal XRAM. This state is illegal. When the C504 internal XRAM is enabled and accesses are made to it, the data memory address range F000H – FFFFH needs to be mapped to the target. If one of these illegal states occurs, the Map Error LED turns on simultaneously with the C504 MCU entering the reset state, signaling the illegal state. This state is maintained until you push the reset button. Leaving the map error state is done only by pressing the reset button S1 on the pod. This prevents repetitive occurrences of the illegal state until you fix the problem.
Emulation	A green LED that indicates when the user program is running, and turns off whenever the user program is stopped (emulation break). The user code can be fetched from RAM on the emulator board or from code memory on the target (usually PROM).
Monitor	A yellow LED that indicates when the monitor mode is active. In this mode, the MCU executes internal emulator code that communicates with the host PC. The emulator enters this mode upon power-up or after emulation break.

POD–51EH–C504–40

Jumper Identification and Description

Jumper Designation	Function	Description
Ground SeLect (GSL)	VAGND Grounding Select	This jumper selects how to ground the VAGND pin of the MCU. ● When the jumper top is across pins 1 and 2, VAGND is connected to emulator ground. ● When the jumper top is across pins 2 and 3, VAGND is grounded through pin 4 of the target adapter. ● Pin 3 of the jumper is marked by the letter T to show connection to the target. The default position of the jumper top is across pins 1 and 2 (emulator ground selected).
Reference SeLect (RSL)	VAREF Source Select	This jumper selects how to power the VAREF pin of the MCU. ● When the jumper top is across pins 1 and 2, VAREF is connected to emulator. ● When the jumper top is across pins 2 and 3, VAREF is connected to the target through pin 3 of the target adapter. ● Pin 3 of the jumper is marked by the letter T to show connection to the target. The default position of the jumper top is across pins 1 and 2 (emulator selected as the voltage source).
Vcc SeLect (VSL)	MCU Vcc Source Select	This jumper selects how to power the Vcc pin of the MCU. ● When the jumper top is on pins 1 and 2, the MCU Vcc pin is connected to the Vcc of the emulator. ● When the jumper top is on pins 2 and 3, the MCU Vcc pin is connected to the Vcc pin of the adapter. This allows the MCU Vcc pin to be powered by the target. The default position of the jumper top is across pins 1 and 2 (emulator selected as Vcc source). **Note**: Taking power from the target system is not recommended.
X1S, X2S	XTAL1 and XTAL2 Source Select	These two jumpers select the clock source for the MCU. ● With the jumper tops on pins 1 and 2, the MCU XTAL1 and XTAL2 pins are connected to a 24-MHz crystal on the pod. ● When these two jumpers have jumper tops on pins 2 and 3, the MCU XTAL1 and XTAL2 pins are connected to pins 15 and 14 of the adapter. This allows the target to supply the clock to the MCU XTAL1 and XTAL2 pins. The default position of the jumper top is across pins 1 and 2 (the pod crystal). **Note**: Both of the jumper tops need to be placed in the same position with either both selecting the pod crystal source, or both selecting the target clock source.

POD–51EH–C504–40

Jumper Identification and Description (continued)

Jumper Designation	Function	Description
JP8	Monitor Stop/Break Select	This jumper determines whether all or only part of the internal MCU peripherals stop in monitor mode after an emulation break occurs. • When the jumper top is out, the stop mode is selected. In this case, all of the internal MCU peripherals are stopped in monitor mode. • When the jumper top is in, the break mode is selected. In this case, only some of the internal peripherals are stopped. This includes the following MCU peripherals: interrupts, watchdog timer, and oscillator watchdog. Timer0, Timer1, and Timer2 Capture are not stopped in this mode. In stop and break mode, all of the internal peripherals start working from the same point they stopped before the break as soon as the user program starts running again when you press the Go or Step buttons. When changing this jumper, the new stop/break mode takes effect after the next reset sequence. The default setup of the jumper is stop mode selection (jumper top out). **Note**: It is recommended to work in stop mode only. The break mode needs to be selected only if your application has limitations that do not allow the timers to be stopped in monitor mode.
XL0..XL7	Trace Port Select Jumpers	These eight 3-pin jumpers select the signals that are routed to the trace board and are shown in the Trace window under the header P1. For any jumper in this group the jumper top position pin 2–3 means that the corresponding pin of the recreated Port0 is traced. Pin 1 of each jumper is left free and can be connected to any signal on the target or port headers on the pod. For example setting XL4 in position 2–3 means that pin P0.4 is traced. If pin XL4.1, for example, is connected to a signal P4.3 on the target, you can choose between tracing signal P0.4 or P4.3 by moving the jumper between positions 2–3 and 1–2.
XH0..XH7	Trace Port Select Jumpers	This group of jumpers is similar to XL0..XL7, except pin 3 of each jumper is connected to the recreated Port2 and pin 1 is left free.
JP7	External Code Enable	This jumper enables access to the external code if the single-chip mode is emulated and part of the code is mapped into the target code memory. The external code is enabled if the jumper top is out (default jumper configuration).
Reset Button		When pressed, this switch resets the pod (including the MCU). You can use this button to recover from a map error status.

POD–51EH–C504–40

Jumper Identification and Description (continued)

Jumper Designation	Function	Description
JP10	SY0, SY1	This header contains two pins: pin 1 (SY0), and pin 2 (SY1). These two signals are carried to the emulator and trace boards, and can be used in the trace and breakpoint logic.
JP9	EMUL, FLF, ANB	This header contains three pins: pin 1 (EMUL), pin 2 (FLF), and pin 3 (ANB). Pin 1 signals emulation or monitor mode. It is low when emulation is active (when the user program is running). Pins 2 and 3 are not used currently, but can be used in the future.
J2	P3.6, P3.7	This header carries the recreated signals; P3.6/WR (pin J2.1) and P3.7/RD (pin J2.2). These pins observe the signals or route them to the trace board through the headers XL0..XL7, XH0..XH7.
P1, P3, P4	Port Headers	These headers carry signals for the corresponding ports. The port pin numbers are marked on the silkscreen. For example, pin 5 of header P3 connects to pin P3.5 of the MCU. **Note**: Earlier revisions of the pod might have header JP15 installed on the board. This header is for factory testing only and is not for any user functions.

Differences Between the Emulator and the C504 MCU

EA Pin

The EA pin in the C504 MCU is an input, floating pin used to select between single-chip mode (the program in internal OTP–ROM) and expanded mode (the program in external memory). This pin does not have a pull-up or a pull-down resistor.

A 47K pull-up resistor on the pod is connected to this pin to select a default state of single-chip mode when this pin is not forced by the target.

This difference of the existence of the pull-up resistor must be considered: if the target pin is tied to a pull-down resistor, it should be a sufficiently low pull-down (3.3K or lower) in order to over-come the 47K pull-up resistor on the pod.

In the C504 MCU during level 0 protection, the EA pin is not sampled during Reset and can be switched dynamically during run-time without going through Reset. This mode is not supported by the pod. The EA pin from the target is always sampled during exit from Reset (Reset Signal Fall time), and the mode of operation is then set to either single-chip or expanded mode until the next Reset.

Port0 and Port2 Pins

These 16 port pins are emulated by the pod and are used both as the address/data bus and as port pins. These pins have slightly different AC and DC characteristics from the Port0 and Port2 pins of the C504 MCU. They sink up to 12mA and source up to 4mA, while maintaining valid TTL output logic levels. Port2 pins also have pull-up resistors of 22K, which might be different from the C504 MCU's internal pull-ups.

POD–51EH–C505C–20

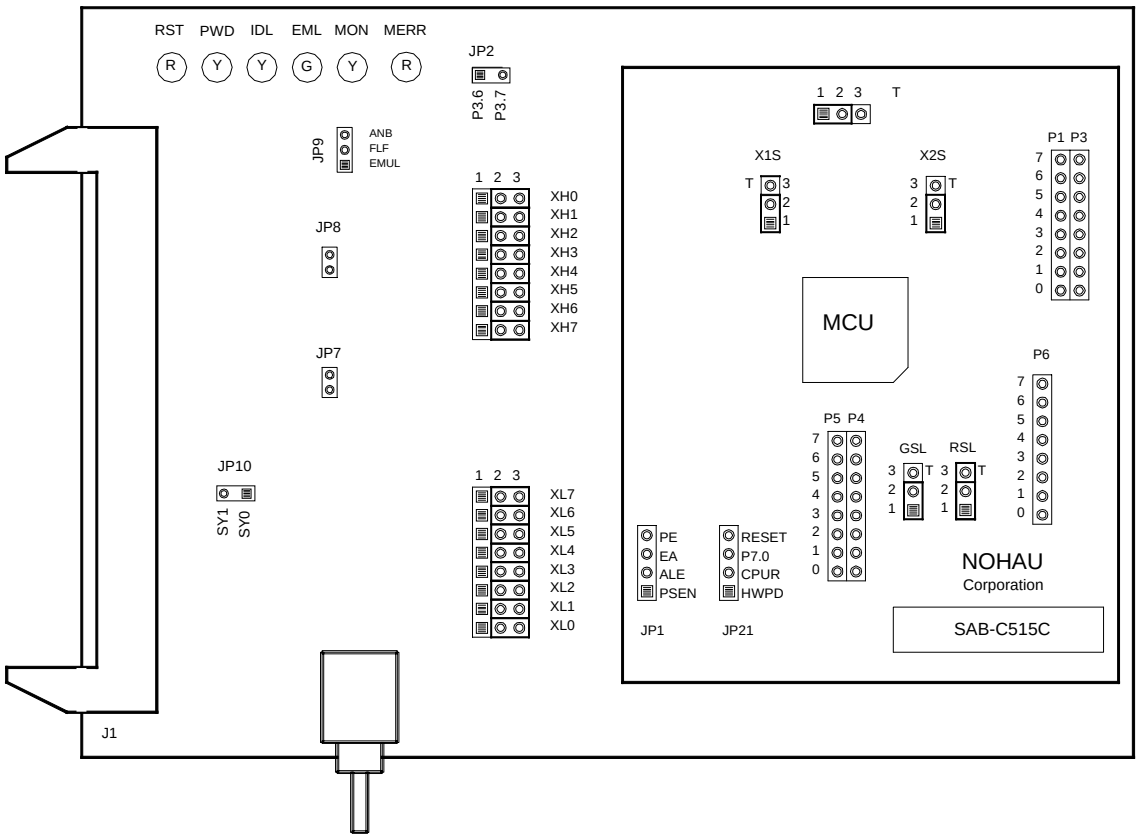


Figure 77. POD–51EH–C505C–20

Operating Instructions

The POD–51EH–C505C–20 supports the Siemens C505C and C505 MCUs.

To work with this pod, you need to select **POD–EH–C505C** from the **Select Processor** drop-down list box (**Step 3** of the **Communications** tab in the Emulator Configuration window).

The C505C MCU is a superset of the C505. The only difference between the two is the C505C has an on-chip CAN MCU which the C505 does not.

The pod uses a C505C MCU and thus the operation of the C505 is emulated by not using the on-chip CAN MCU of the C505C.

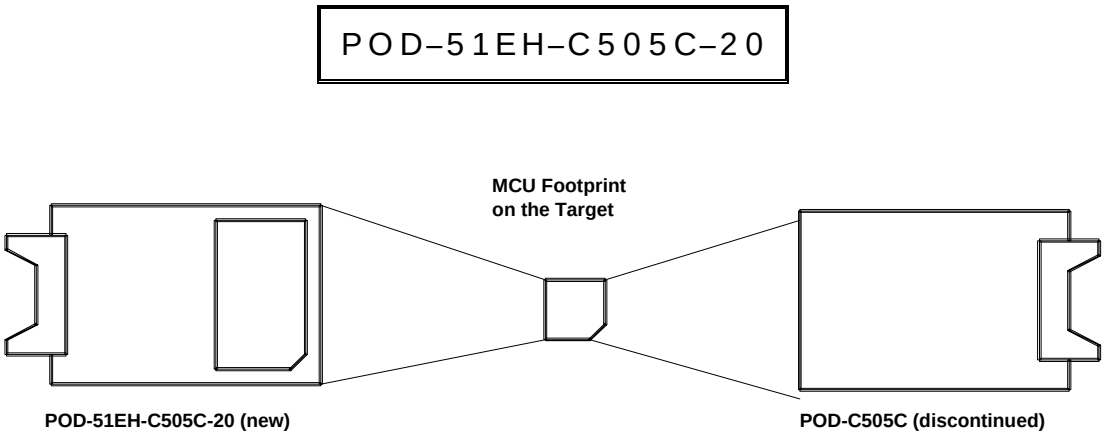


Figure 78. Pod Orientation Relative to the MCU Footprint on the Target

Note

POD–51EH–C505C–20 replaces POD–C505C–20, an older version of the pod for the Siemens SAB–C505C MCU. If you have the older version of the pod, you need to pay attention to the position of the pod relative to your target. Compared to the old pod, the new one needs to turn 180 degrees when you install it on the same target. (See Figure 78.)

The EPROM on the emulator board must be version COM 1.4.

The pod assembly consists of two sub-assemblies: The larger universal Port Replacement Unit (PRU) and the smaller processor module daughterboard.

The PRU provides a 44-pin PGA socket on the bottom that you can connect to the target using an adapter. The PRU also has a 50-pin header J1 that connects through the ribbon cable to the emulator board.

The PGA socket is designed to connect to the following PGA to QFP adapter:
Nohau P/N: EDI/44PG/QFS31–SD

This is a solder-down adapter that is soldered to the target system, replacing the C505C or C505 MCU IC. This adapter can be ordered from Nohau or directly from EDI.

Both sockets on the pod and the target have female contacts. To connect them to each other, you need to use a PGA–44 pin-to-pin plug supplied with the pod.

The POD–51EH–C505C–20 runs up to a maximum frequency of 20 MHz. It has an internal 16-MHz crystal on the board. In order to work at a frequency higher than 16 MHz, for example, 20 MHz, an external 50% duty cycle clock signal should be supplied from the target board and the jumpers should be configured accordingly. See the “Jumper Identification and Description” section.

POD–51EH–C505C–20

As an example, in order to generate a 20-MHz, 50 percent duty-cycle, the CMOS compatible clock signal (as required by the C505C and C505 User's Manual), uses a 40-MHz clock oscillator and a CMOS compatible TFF such as 74AC112.

Every operating frequency of the C505C and C505 is equivalent to twice the operating frequency of an original 8051 MCU with the same clock frequency. For example, when using a 16-MHz crystal the operating frequency is equivalent to an 8051 with a 32-MHz crystal. This is because the C505C and C505 don't divide the clock source by two as does the original 8051.

For this reason, the POD–51EH–C505C–20 might only function correctly with twice the higher frequency of the emulator and trace boards. For example, for 16-MHz operation, use an emulator and trace boards with 32-MHz frequency or higher. For 20-MHz operation, use an emulator and trace boards with a 40-MHz frequency or higher.

In addition, you need to input twice the MCU frequency in the CLK field in the Hardware Configuration window. For example, for a 20-MHz clock, input 40 MHz in the CLK field.

For details on how to configure the advanced emulator boards for high frequency operation with the pod, refer to the "Advanced Emulator Configuration for the POD–C505C" section.

The C505C and the C505 MCUs have 256 bytes of on-chip memory located at the XDATA addresses FF00H – FFFFH. The C505C also has the CAN MCU registers mapped to XDATA space at addresses F700H – F7FFH. If your application is using the on-chip XRAM or the CAN registers, you must map the last memory range of the XDATA memory (addresses F000H – FFFFH) to the target. This memory mapping is done in the EMUL51–PC Windows software under the Config – Memory map menu. Therefore, after the internal XRAM and CAN registers are enabled in the SYSCON register, accesses to address range F700H – FFFH will be used to access internal MCU memory, and accesses to address range F000H – F6FFH will be used to access target XDATA memory.

If your application is not using the on-chip XRAM, you can map this memory range (F000H – FFFFH) to either the emulator or to the target.

The POD–51EH–C505C–20 also supports tracing of the internal XRAM and CAN registers read and write cycles using the MOVX instructions. This is supported in both single-chip or external modes.

POD–51EH–C505C–20

POD–51EH–C505C–20 has two operating modes.

- Single-chip
- External

The active mode is selected by the EA pin (pin 29 of the POD–51EH–C505C–20 adapter) during the Reset signal rise (also sampled on internal watchdog timer Reset).

This pin is sampled by the emulator only during Exit from Reset, and therefore, it is recommended that you connect it to either Constant Low or Constant High. This pin defaults to High (selecting single-chip mode), which is implemented by a 47K pull-up resistor on the pin.

When using external mode, the code memory can be mapped either to the target or the emulator (with boundaries of 4K). In external mode, addresses are switched to Port2 and Port0 pins for every bus cycle regardless of where it is mapped. The data, however, is read or written to the Port0 pins only for addresses that are mapped to the target.

When using single-chip mode, you need to map the first 16K of the code memory (address range 0 –3FFFH) to the emulator board. In this mode, Port2 and Port0 can be used as I/O ports. These pins carry the address and data information only in the following cases.

- XDATA accesses, using MOVX instructions, to an address that is mapped to the target.
- XDATA accesses to the internal XRAM when the XMAP1 bit in the SYSCON register is set.
- Depending on JP7, for code accesses—see details in the “JP7: External Code Enable” section in the “LED Indicators” table.

In all other cases, depending on JP7, these pins carry I/O values of Port0 and Port2 either outputted from the MCU according to the Port0 and Port2 SFR registers value, or inputted from the target to the MCU.

POD–51EH–C505C–20

LED Indicators

Function	Description
Reset	A red LED that indicates when the MCU reset is active. There are five possible sources of the MCU reset. • Reset from the target starts whenever a low level is detected on the reset pin of the adapter. • Reset from the S1 button starts by pressing S1 on the pod. • Click the Reset button in the Seehau menu to reset the emulator. • Reset from the internal watchdog timer starts if the internal watchdog is enabled, and it is not fed by the user software for a long period. (**Note**: It is important to feed the watchdog periodically, otherwise, problems during the debug process might occur.) • When a map error occurs. During all of these reset sequences, the reset LED is turned on for a short or a long period of time depending on the sequence.
Map Error	This red LED signals a mapping error occurring in one of the following situations. • When the internal C505C XRAM is enabled by bit XMAP0 in the SYSCON register. • The data memory address range FF00H – FFFFH is mapped to the emulator. • There is a memory access (MOVX) to the internal XRAM. This state is illegal. When the C505C internal XRAM is enabled and accesses are made to it, the data memory address range FF00H – FFFFH needs to be mapped to the target. • When single-chip mode is selected by the EA pin from the target during the last Reset, a code memory address range in the first 16K (address 0 – 3FFFH) is mapped to the target and there is code access to one of the addresses in this range. This state is illegal. In single-chip mode, when JP7 is set to disable code accesses to the target, all code memory needs to be mapped to the emulator. If one of these illegal states occurs, the Map Error (ERR) LED turns on simultaneously with the C505C MCU on the pod entering the reset state, signaling the illegal state. This state is maintained until you push the reset button. Leaving the map error state is done only by pressing the reset button S1 on the pod. This prevents repetitive occurrences of the illegal state until you fix the problem.
Emulation	A green LED that indicates when the user program is running, and turns off whenever the user program is stopped (emulation break). The user code can be fetched from RAM on the emulator board or from code memory on the target (usually PROM).
Monitor	A yellow LED that indicates when the monitor mode is active. In this mode, the MCU executes internal emulator code that communicates with the host PC. The emulator enters this mode upon power-up or after emulation break.

POD–51EH–C505C–20

Jumper Identification and Description

Jumper Designation	Function	Description
Ground SeLect (GSL)	VAGND Grounding Select	This jumper selects how to ground the VAGND pin of the MCU. ● When the jumper top is across pins 1 and 2, VAGND is connected to emulator ground. ● When the jumper top is across pins 2 and 3, VAGND is grounded through pin 4 of the target adapter. Pin 3 of the jumper is marked by the letter T to show connection to the target. The default position of the jumper top is across pins 1 and 2 (emulator ground selected).
Reference SeLect (RSL)	VAREF Source Select	This jumper selects how to power the VAREF pin of the MCU. ● When the jumper top is across pins 1 and 2, VAREF is connected to emulator. ● When the jumper top is across pins 2 and 3, VAREF is connected to the target through pin 3 of the target adapter. Pin 3 of the jumper is marked by the letter T to show connection to the target. The default position of the jumper top is across pins 1 and 2 (emulator selected as the voltage source).
Vcc SeLect (VSL)	MCU Vcc Source Select	This jumper selects how to power the Vcc pin of the MCU. ● When the jumper top is on pins 1 and 2, the MCU Vcc pin is connected to the Vcc of the emulator. ● When the jumper top is on pins 2 and 3, the MCU Vcc pin is connected to the Vcc pin of the adapter. This allows the MCU Vcc pin to be powered by the target. The default position of the jumper top is across pins 1 and 2 (emulator selected as Vcc source). **Note**: Taking power from the target system is not recommended.
X1S, X2S	XTAL1 and XTAL2 Source Select	These two jumpers select the clock source for the MCU. ● With the jumper tops on pins 1 and 2, the MCU XTAL1 and XTAL2 pins are connected to a 16-MHz crystal on the pod. ● When these two jumpers have jumper tops on pins 2 and 3, the MCU XTAL1 and XTAL2 pins are connected to pins 15 and 14 of the adapter. This allows the target to supply the clock to the MCU XTAL1 and XTAL2 pins. The default position of the jumper top is across pins 1 and 2 (the pod crystal). **Note**: Both of the jumper tops need to be placed in the same position with either both selecting the pod crystal source, or both selecting the target clock source.

POD–51EH–C505C–20

Jumper Identification and Description (continued)

Jumper Designation	Function	Description
JP8	Monitor Stop/Break Select	This jumper determines whether all or only part of the internal MCU peripherals stop in monitor mode after an emulation break occurs. • When the jumper top is out, the stop mode is selected. In this case, all of the internal MCU peripherals are stopped in monitor mode. This includes the following MCU peripherals: interrupts, watchdog timer, timer0, timer1, timer2, and timer2 capture. • When the jumper top is on, the break mode is selected. In this case, only some of the internal peripherals are stopped. This includes the following MCU peripherals: interrupts, watchdog timer, and timer2. Timer0, timer1, and timer2 capture are not stopped in this mode. In stop and break mode, all of the internal peripherals start working from the same point they stopped before the break as soon as the user program starts running again when you press Go or Step. When changing this jumper, the new stop/break mode takes effect after the next reset sequence. The default setup of the jumper is stop mode selection (jumper top out). **Note**: It is recommended to work in stop mode only. The break mode needs to be selected only if your application has limitations that do not allow the timers to be stopped in monitor mode.
XL0..XL7	Trace Port Select Jumpers	These eight 3-pin jumpers select the signals that are routed to the trace board and are shown in the Trace window under the header P1. For any jumper in this group the jumper top position pin 2–pin 3 means that the corresponding pin of the recreated Port0 is traced. Pin 1 of each jumper is left free and can be connected to any signal on the target or port headers on the pod. For example setting XL4 in position 2–3 means that pin P0.4 is traced. If pin XL4.1, for example, is connected to a signal P4.3 on the target, you can choose between tracing signal P0.4 or P4.3 by moving the jumper between positions 2–3 and 1–2.
XH0..XH7	Trace Port Select Jumpers	This group of jumpers is similar to XL0..XL7, except pin 3 of each jumper is connected to the recreated Port2 and pin 1 is left free.
JP7	External Code Enable	This jumper enables access to the external code if the single-chip mode is emulated and part of the code is mapped into the target code memory. The external code is enabled if the jumper top is out (default jumper configuration).
Reset Button		When pressed, this switch resets the pod (including the MCU). You can use this button to recover from a map error status.
JP10	SY0, SY1	This header contains two pins; pin 1 (SY0), and pin 2 (SY1). These two signals are carried to the emulator and trace boards, and can be used in the trace and breakpoint logic.

POD–51EH–C505C–20

Jumper Identification and Description (continued)

Jumper Designation	Function	Description
JP9	EMUL, FLF, ANB	This header contains three pins: pin 1 (EMUL), pin 2 (FLF), and pin 3 (ANB). Pin 1 signals emulation or monitor mode. It is low when emulation is active (when the user program is running). Pins 2 and 3 are not used currently, but can be used in the future.
J2	P3.6, P3.7	This header carries the recreated signals; P3.6/WR (pin J2.1) and P3.7/RD (pin J2.2). These pins observe the signals or route them to the trace board through the headers XL0..XL7, XH0..XH7.
P1, P3, P4	Port Headers	These headers carry signals for the corresponding ports. The port pin numbers are marked on the silk-screen. For example, pin 5 of header P3 connects to pin P3.5 of the MCU. **Note**: Earlier revisions of the pod might have header JP15 installed on the board. This header is for factory testing only and is not for any user functions.

Advanced Emulator Configuration for the POD–C505C

When using an advanced emulator board (part numbers EMUL51–PC/EA256–BSW–50 or the EMUL51–PC/EA768–BSW–50), it is important to configure the advanced emulator board for the POD–C505C, especially when applying a high frequency clock signal above 12 MHz to the C505C MCU.

Make sure that the two following jumpers in the advanced emulator board are removed.

- The DAL jumper on JP1
- The RWEN jumper on JP1

These two jumpers affect the operation mode and maximum operating frequency of the advanced emulator board.

For details on how to configure the other jumpers on the advanced emulator board, refer to Chapter 2, “Installing the Hardware” (section “EMUL51–PC/EA256 – 256K Bank Switch Emulator Configuration,” or section “EMUL51–PC/EA768 – 768K Bank Switch Emulator Configuration”).

POD–51EH–C505C–20

The Differences Between the Emulator and the C505C and C505 MCUs

EA Pin

The EA pin in the C505C and the C505 MCUs is an input, floating pin used to select between single-chip mode (the program in internal OTP–ROM) and expanded mode (the program in external memory). This pin does not have a pull-up or a pull-down resistor.

A 47K pull-up resistor on the pod is connected to this pin to select a default state of single-chip mode when this pin is not forced by the target.

This difference of the existence of the pull-up resistor must be considered: if the target pin is tied to a pull-down resistor, it should be a sufficiently low pull-down (3.3K or lower) in order to overcome the 47K pull-up resistor on the pod.

In the C505C and the C505 MCUs during level 0 protection, the EA pin is not sampled during Reset and can be switched dynamically during run-time without going through Reset. This mode is not supported by the pod. The EA pin from the target is always sampled during exit from Reset (Reset Signal Fall time), and the mode of operation is then set to either single-chip or expanded mode until the next Reset.

Port0 and Port2 Pins

These 16 port pins are emulated by the pod and are used both as the address/data bus and as port pins. These pins have slightly different AC and DC characteristics from the Port0 and Port2 pins of the C505C and C505 microcontrollers. They sink up to 12mA and source up to 4mA, while maintaining valid TTL output logic levels. Port2 pins also have pull-up resistors of 22K, which might be different from the C505C and C505 MCUs' internal pull-ups.

Port3.6/WR and Port3.7/RD Pins

These two port pins are emulated by the pod and are used as the read and write signals or as port pins. They have slightly different AC and DC characteristics from the Port3.6/WR and Port3.7/RD pins of the C505C and C505 MCUs. These pins sink up to 12mA and source up to 4mA, (on transition from logic 0 to 1) while maintaining valid TTL output logic levels. They have pull-up resistors of 22K to maintain a logic 1 level and emulate the quasi-bi-directional operation. These features differ from the internal pull-ups of the C505C and C505 MCUs.POD–51EH–C505CA–20

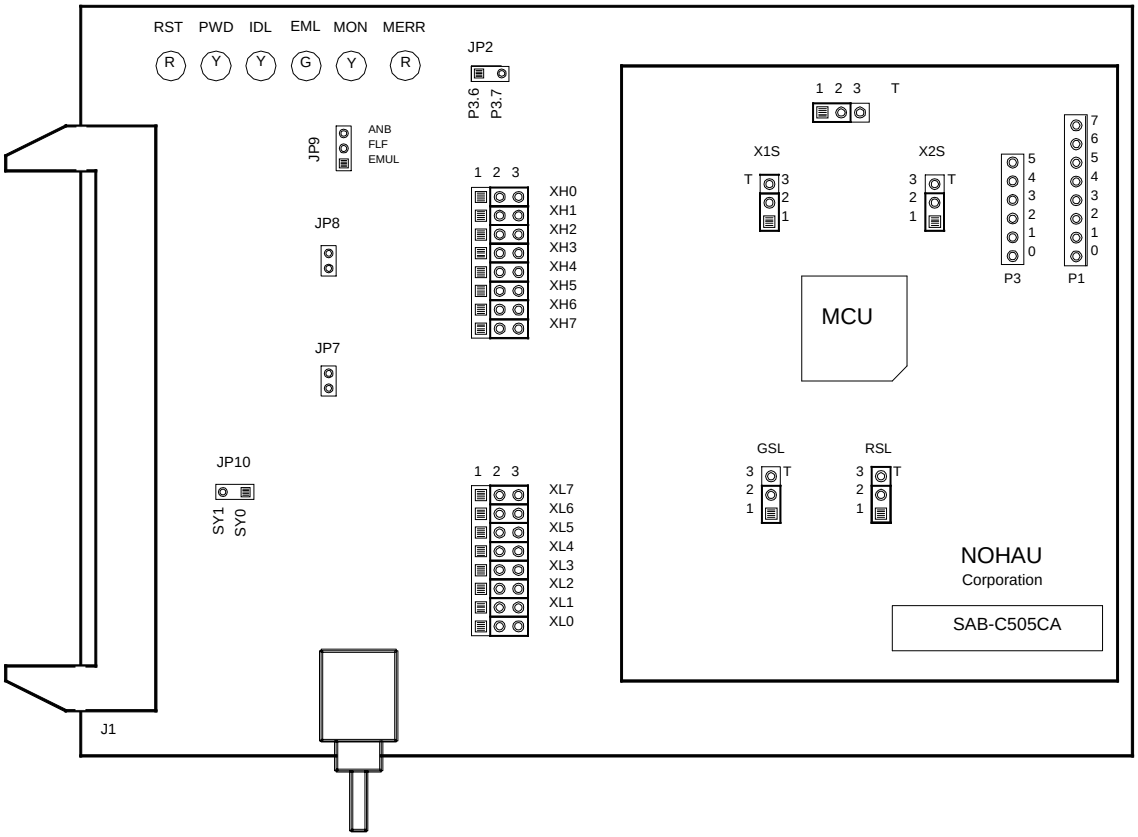


Figure 79. POD–51EH–C505CA–20

Operating Instructions

The POD–51EH–C505CA–20 supports the Siemens C505CA and C505A MCUs.

To work with this pod, you need to select **POD–EH–C505CA** from the **Select Processor** drop-down list box (**Step 3** of the **Communications** tab in the Emulator Configuration window).

The C505CA MCU is a superset of the C505A. The only difference between the two is the C505CA has an on-chip CAN MCU which the C505A does not.

The pod uses a C505CA MCU and thus the operation of the C505A is emulated by not using the on-chip CAN MCU of the C505CA

.POD–51EH–C505CA–20

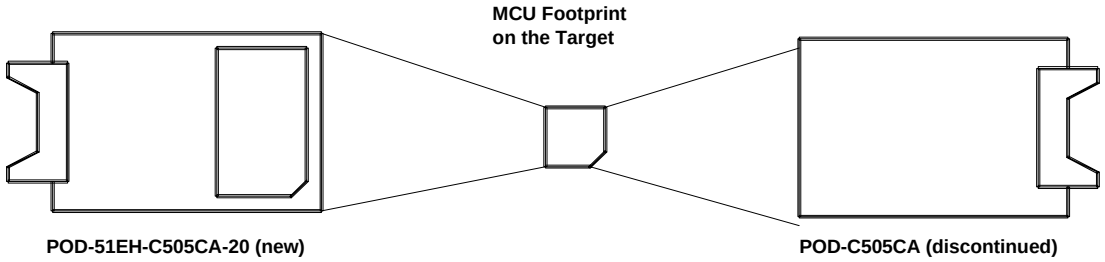


Figure 80. Pod Orientation Relative to the MCU Footprint on the Target

Note

POD–51EH–C505CA–20 replaces POD–C505CA–20, an older version of the pod for the Siemens SAB–C505CA MCU. If you have the older version of the pod, you need to pay attention to the position of the pod relative to your target. Compared to the old pod, the new one needs to turn 180 degrees when you install it on the same target. (See Figure 80.)

The EPROM on the emulator board must be version COM 1.4.

The pod assembly consists of two sub-assemblies: The larger universal Port Replacement Unit (PRU) and the smaller processor module daughterboard.

The PRU provides a 44-pin PGA socket on the bottom that you can connect to the target using an adapter. The PRU also has a 50-pin header J1 that connects through the ribbon cable to the emulator board.

The PGA socket is designed to connect to the following PGA to QFP adapter:
Nohau P/N: EDI/44PG/QFS31–SD

This is a solder-down adapter that is soldered to the target system, replacing the C505CA or C505A MCU IC. This adapter can be ordered from Nohau or directly from EDI.

Both sockets on the pod and the target have female contacts. To connect them to each other, you need to use a PGA–44 pin-to-pin plug supplied with the pod.

The POD–51EH–C505CA–20 runs up to a maximum frequency of 20 MHz. It has an internal 16-MHz crystal on the board. In order to work at a frequency higher than 16 MHz, for example, 20 MHz, an external 50% duty cycle clock signal should be supplied from the target board and the jumpers should be configured accordingly. See the “Jumper Identification and Description” table.

POD–51EH–C505CA–20

As an example, in order to generate a 20-MHz, 50 percent duty-cycle, the CMOS compatible clock signal (as required by the C505CA and C505A User's Manual), uses a 40-MHz clock oscillator and a CMOS compatible TFF such as 74AC112.

Every operating frequency of the C505CA and C505A is equivalent to twice the operating frequency of an original 8051 MCU with the same clock frequency. For example, when using a 16-MHz crystal the operating frequency is equivalent to an 8051 with a 32-MHz crystal. This is because the C505CA and C505A don't divide the clock source by two as does the original 8051.

For this reason, the POD–51EH–C505CA–20 might only function correctly with twice the higher frequency of the emulator and trace boards. For example, for 16-MHz operation, use an emulator and trace boards with 32 MHz frequency or higher. For 20-MHz operation, use an emulator and trace boards with 40 MHz frequency or higher.

In addition, you need to input twice the MCU frequency in the CLK field in the Hardware Configuration window. For example, for a 20-MHz clock, input 40 MHz in the CLK field.

For details on how to configure the advanced emulator boards for high frequency operation with the pod, refer to the "Advanced Emulator Configuration for the POD–C505CA" section.

The C505CA and the C505A MCUs have 1K on-chip memory located at XDATA addresses FC00H – FFFFH. The C505CA also has the CAN MCU registers mapped to XDATA space at addresses F700H – F7FFH. If your application is using the on-chip XRAM or the CAN registers, you must map the last memory range of the XDATA memory (addresses F000H – FFFFH) to the target. This memory mapping is done in the EMUL51–PC Windows software under the Config – Memory map menu. Therefore, after the internal XRAM and CAN registers are enabled in the SYSCON register, accesses to address range F700H – FFFH will be used to access internal MCU memory, and accesses to address range F000H – F6FFH will be used to access target XDATA memory.

If your application is not using the on-chip XRAM, you can map this memory range; (F000H – FFFFH) to either the emulator or to the target.

The POD–51EH–C505CA–20 also supports tracing of the internal XRAM and CAN registers read and write cycles using the MOVX instructions. This is supported in both single-chip or external modes.

POD–51EH–C505CA–20

POD–51EH–C505CA–20 has two operating modes.

- Single-chip
- External

The active mode is selected by the EA pin (pin 29 of the POD–51EH–C505CA–20 adapter) during the Reset signal rise (also sampled on internal watchdog timer Reset).

This pin is sampled by the emulator only during Exit from Reset, and therefore, it is recommended that you connect it to either Constant Low or Constant High. This pin defaults to High (selecting single-chip mode), which is implemented by a 47K pull-up resistor on the pin.

When using external mode, the code memory can be mapped either to the target or the emulator (with boundaries of 4K). In external mode, addresses are switched to Port2 and Port0 pins for every bus cycle regardless of where it is mapped. The data, however, is read or written to the Port0 pins only for addresses that are mapped to the target.

When using single-chip mode, you need to map the first 32K of the code memory (address range 0 – 7FFFH) to the emulator board. In this mode, Port2 and Port0 can be used as I/O ports. These pins carry the address and data information only in the following cases.

- XDATA accesses, using MOVX instructions, to an address that is mapped to the target.
- XDATA accesses to the internal XRAM when the XMAP1 bit in the SYSCON register is set.
- Depending on JP7, for code accesses—see details in the “JP7: External Code Enable” section in the “Jumper Identification and Description” table.

In all other cases, depending on JP7, these pins carry I/O values of Port0 and Port2 either outputted from the MCU according to the Port0 and Port2 SFR registers value, or Inputted from the target to the MCU.

POD–51EH–C505CA–20

LED Indicators

Function	Description
Reset	A red LED that indicates when the MCU reset is active. There are five possible sources of the MCU reset. • Reset from the target starts whenever a low level is detected on the reset pin of the adapter. • Reset from the S1 button starts by pressing S1 on the pod. • Click the Reset button in the Seehau menu to reset the emulator. • Reset from the internal watchdog timer starts if the internal watchdog is enabled, and it is not fed by the user software for a long period. (**Note**: It is important to feed the watchdog periodically, otherwise, problems during the debug process might occur.) • When a map error occurs. During all of these reset sequences, the reset LED is turned on for a short or a long period of time depending on the sequence.
Map Error	A red LED that indicates a mapping error occurring in one of the following situations. • When the internal C505C XRAM is enabled by bit XMAP0 in the SYSCON register. • The data memory address range FF00H – FFFFH is mapped to the emulator. • There is a memory access (MOVX) to the internal XRAM. This state is illegal. When the C505CA internal XRAM is enabled and accesses are made to it, the data memory address range FF00H – FFFFH needs to be mapped to the target. • When single-chip mode is selected by the EA pin from the target during the last Reset, a code memory address range in the first 32K (address 0 – 7FFFH) is mapped to the target and there is code access to one of the addresses in this range. This state is illegal. In single-chip mode, when JP7 is set to disable code accesses to the target, all code memory needs to be mapped to the emulator. If one of these illegal states occurs, the Map Error (ERR) LED turns on simultaneously with the C505CA MCU on the pod entering the reset state, signaling the illegal state. This state is maintained until you push the reset button. Leaving the map error state is done only by pressing the reset button S1 on the pod. This prevents repetitive occurrences of the illegal state until you fix the problem.
Emulation	A green LED that indicates when the user program is running, and turns off when the user program is stopped (emulation break). The user code can be fetched from RAM on the emulator board or from code memory on the target (usually PROM).
Monitor	A yellow LED that indicates when the monitor mode is active. In this mode, the MCU executes internal emulator code that communicates with the host PC. The emulator enters this mode upon power-up or after emulation break.

POD–51EH–C505CA–20

Jumper Identification and Description

Jumper Designation	Function	Description
Ground SeLect (GSL)	VAGND Grounding Select	This jumper selects how to ground the VAGND pin of the MCU. ● When the jumper top is across pins 1 and 2, VAGND is connected to emulator ground. ● When the jumper top is across pins 2 and 3, VAGND is grounded through pin 4 of the target adapter. Pin 3 of the jumper is marked by the letter T to show connection to the target. The default position of the jumper top is across pins 1 and 2 (emulator ground selected).
Reference SeLect (RSL)	VAREF Source Select	This jumper selects how to power the VAREF pin of the MCU. ● When the jumper top is across pins 1 and 2, VAREF is connected to emulator. ● When the jumper top is across pins 2 and 3, VAREF is connected to the target through pin 3 of the target adapter. Pin 3 of the jumper is marked by the letter T to show connection to the target. The default position of the jumper top is across pins 1 and 2 (emulator selected as the voltage source).
Vcc SeLect (VSL)	MCU Vcc Source Select	This jumper selects how to power the Vcc pin of the MCU. ● When the jumper top is on pins 1 and 2, the MCU Vcc pin is connected to the Vcc of the emulator. ● When the jumper top is on pins 2 and 3, the MCU Vcc pin is connected to the Vcc pin of the adapter. This allows the MCU Vcc pin to be powered by the target. The default position of the jumper top is across pins 1 and 2 (emulator selected as Vcc source). **Note**: Taking power from the target system is not recommended.
X1S, X2S	XTAL1 and XTAL2 Source Select	These two jumpers select the clock source for the MCU. ● With the jumper tops on pins 1 and 2, the MCU XTAL1 and XTAL2 pins are connected to a 16-MHz crystal on the pod. ● When these two jumpers have jumper tops on pins 2 and 3, the MCU XTAL1 and XTAL2 pins are connected to pins 15 and 14 of the adapter. This allows the target to supply the clock to the MCU XTAL1 and XTAL2 pins. The default position of the jumper top is across pins 1 and 2 (the pod crystal). **Note**: Both of the jumper tops need to be placed in the same position with either both selecting the pod crystal source, or both selecting the target clock source.

POD–51EH–C505CA–20

Jumper Identification and Description (continued)

Jumper Designation	Function	Description
JP8	Monitor Stop/Break Select	This jumper determines whether all or only part of the internal MCU peripherals stop in monitor mode after an emulation break occurs. • When the jumper top is out, the stop mode is selected. In this case, all of the internal MCU peripherals are stopped in monitor mode. This includes the following MCU peripherals: interrupts, watchdog timer, timer0, timer1, timer2, and timer2 capture. • When the jumper top is on, the break mode is selected. In this case, only some of the internal peripherals are stopped. This includes the following MCU peripherals: interrupts, watchdog timer, and timer2. timer0, timer1, and timer2 capture are not stopped in this mode. In stop and break mode, all of the internal peripherals start working from the same point they stopped before the break as soon as the user program starts running again when you press Go or Step. When changing this jumper, the new stop/break mode takes effect after the next reset sequence. The default setup of the jumper is stop mode selection (jumper top out). **Note**: It is recommended to work in stop mode only. The break mode needs to be selected only if your application has limitations that do not allow the timers to be stopped in monitor mode.
XL0..XL7	Trace Port Select Jumpers	These eight 3-pin jumpers select the signals that are routed to the trace board and are shown in the trace window under the header P1. For any jumper in this group the jumper top position pin 2–pin 3 means that the corresponding pin of the recreated Port0 is traced. pin 1 of each jumper is left free and can be connected to any signal on the target or port headers on the pod. For example setting XL4 in position 2–3 means that pin P0.4 is traced. If pin XL4.1, for example, is connected to a signal P4.3 on the target, you can choose between tracing signal P0.4 or P4.3 by moving the jumper between positions 2–3 and 1–2.
XH0..XH7	Trace Port Select Jumpers	This group of jumpers is similar to XL0..XL7, except pin 3 of each jumper is connected to the recreated Port2 and pin 1 is left free.
JP7	External Code Enable	This jumper enables access to the external code if the single-chip mode is emulated and part of the code is mapped into the target code memory. The external code is enabled if the jumper top is out (default jumper configuration).
Reset Button		When pressed, this switch resets the pod (including the MCU). You can use this button to recover from a map error status.
JP10	SY0, SY1	This header contains two pins; pin 1 (SY0), and pin 2 (SY1). These two signals are carried to the emulator and trace boards, and can be used in the trace and breakpoint logic.

POD–51EH–C505CA–20

Jumper Identification and Description (continued)

Jumper Designation	Function	Description
JP9	EMUL, FLF, ANB	This header contains three pins: pin 1 (EMUL), pin 2 (FLF), and pin 3 (ANB). Pin 1 signals emulation or monitor mode. It is low when emulation is active (when the user program is running). Pins 2 and 3 are not used currently, but can be used in the future.
J2	P3.6, P3.7	This header carries the recreated signals; P3.6/WR (pin J2.1) and P3.7/RD (pin J2.2). These pins observe the signals or route them to the trace board through the headers XL0..XL7, XH0..XH7.
P1, P3, P4	Port Headers	These headers carry signals for the corresponding ports. The port pin numbers are marked on the silk-screen. For example, pin 5 of header P3 connects to pin P3.5 of the MCU. **Note**: Earlier revisions of the pod might have header JP15 installed on the board. This header is for factory testing only and is not for any user functions.

Advanced Emulator Configuration for the POD–C505CA

When using an advanced emulator board (part numbers EMUL51–PC/EA256–BSW–50 or the EMUL51–PC/EA768–BSW–50), it is important to configure the advanced emulator board for the POD–C505CA, especially when applying a high frequency clock signal above 12 MHz to the C505CA MCU.

Make sure that the two following jumpers in the advanced emulator board are removed.

- The DAL jumper on JP1
- The RWEN jumper on JP1

These two jumpers affect the operation mode and maximum operating frequency of the advanced emulator board.

For details on how to configure the other jumpers on the advanced emulator board, refer to Chapter 2, “Installing the Hardware” (section “EMUL51–PC/EA256 – 256K Bank Switch Emulator Configuration,” or section “EMUL51–PC/EA768 – 768K Bank Switch Emulator Configuration”).

Differences Between the Emulator and the C505CA and C505A MCUs

EA Pin

The EA pin in the C505CA and the C505A MCUs is an input, floating pin used to select between single-chip mode (the program in internal OTP–ROM) and expanded mode (the program in external memory). This pin does not have a pull-up or a pull-down resistor.

A 47K pull-up resistor on the pod is connected to this pin to select a default state of single-chip mode when this pin is not forced by the target.

POD–51EH–C505CA–20

This difference of the existence of the pull-up resistor must be considered: if the target pin is tied to a pull-down resistor, it should be a sufficiently low pull-down (3.3K or lower) in order to overcome the 47K pull-up resistor on the pod.

In the C505CA and the C505A MCUs during level 0 protection, the EA pin is not sampled during Reset and can be switched dynamically during run-time without going through Reset. This mode is not supported by the pod. The EA pin from the target is always sampled during exit from Reset (Reset Signal Fall time), and the mode of operation is then set to either single-chip or expanded mode until the next Reset.

Port0 and Port2 Pins

These 16 port pins, used as the address/data bus and as port pins, are emulated by the pod. These pins have slightly different AC and DC characteristics from the Port0 and Port2 pins of the C505CA and C505A MCUs. These pins sink up to 12mA and source up to 4mA, while maintaining valid TTL output logic levels. Port2 pins also have pull-up resistors of 22K, which might be different from the C505CA and C505A MCUs' internal pull-ups.

Port3.6/WR and Port3.7/RD Pins

These two port pins are emulated by the pod and are used as the read and write signals or as port pins. They have slightly different AC and DC characteristics from the Port3.6/WR and Port3.7/RD pins of the C505CA and C505A MCUs. These pins sink up to 12mA and source up to 4mA, (on transition from logic 0 to 1) while maintaining valid TTL output logic levels. They have pull-up resistors of 22K to maintain a logic 1 level and emulate the quasi-bi-directional operation. These features differ from the internal pull-ups of the C505CA and C505A MCUs

POD–51EH–C505L–20

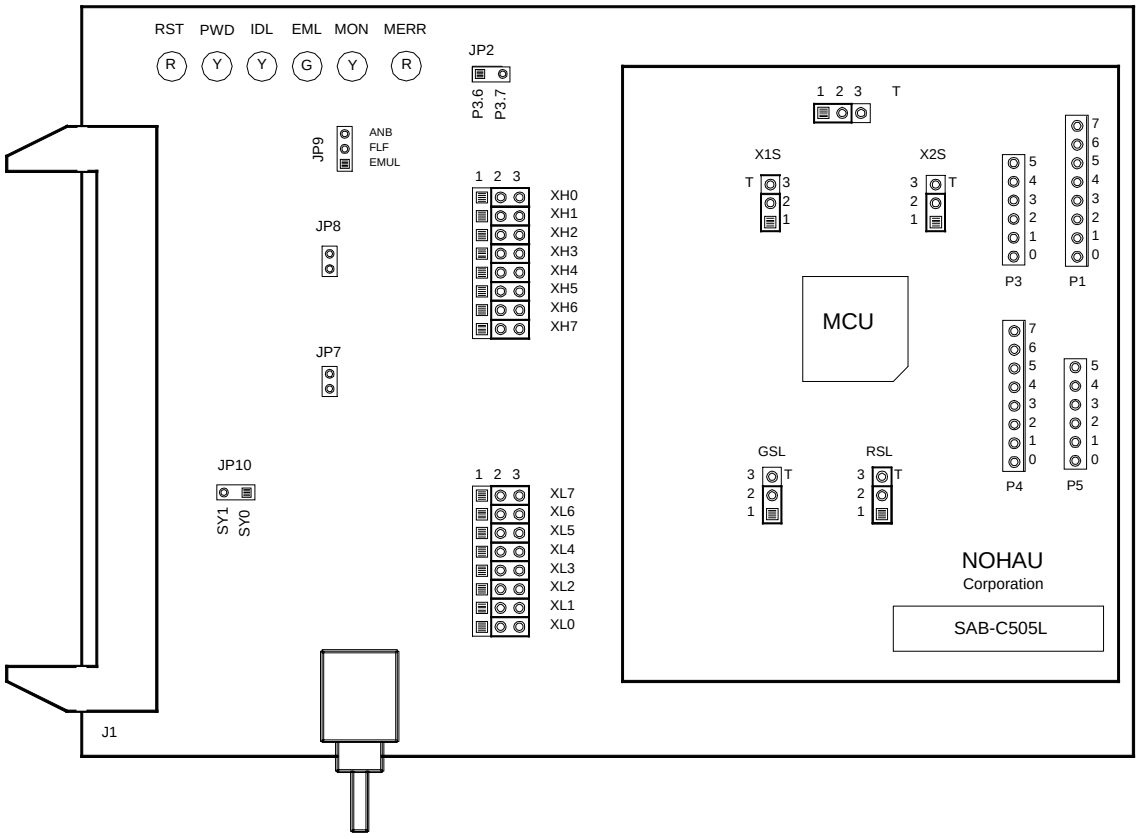


Figure 81. POD–51EH–C505L–20

Operating Instructions

The POD–51EH–C505L–20 supports the Siemens C505L MCU.

To work with this pod, you need to select **POD–EH–C505L** from the **Select Processor** drop-down list box (**Step 3** of the **Communications** tab in the Emulator Configuration window).

The EPROM on the emulator board must be version COM 1.4.

The pod assembly consists of two sub-assemblies: The larger universal Port Replacement Unit (PRU) and the smaller processor module daughterboard.

The target headers plug into an adapter that you select depending on the type of MCU. The C505L MCU uses the Emulation Solutions ES/180–5550–65 adapter.

POD–51EH–C505L–20

This is a QFP–80 solder-down adapter that is soldered to the target system, replacing the C505L MCU IC. This adapter can be ordered from www.icetech.com or directly from Emulation Solutions.

The POD–51EH–C505L–20 runs up to a maximum frequency of 20 MHz. It has an internal 16-MHz crystal on the board. In order to work at a frequency higher than 16 MHz, for example, 20 MHz, an external 50% duty cycle clock signal should be supplied from the target board and the jumpers should be configured accordingly. See the “Jumper Identification and Description” section.

Every operating frequency of the C505L is equivalent to twice the operating frequency of an original 8051 MCU with the same clock frequency. For example, when using a 16-MHz crystal the operating frequency is equivalent to an 8051 with a 32-MHz crystal. This is because the C505L doesn’t divide the clock source by two as does the original 8051.

For this reason, the POD–51EH–C505L–20 might only function correctly with twice the higher frequency of the emulator and trace boards. For example, for a 16-MHz operation, use an emulator and trace boards with 32 MHz frequency or higher. For a 20-MHz operation, use an emulator and trace boards with 40 MHz frequency or higher.

In addition, you need to input twice the MCU frequency in the CLK field in the Hardware Configuration window. For example, for a 20-MHz clock, input 40 MHz in the CLK field.

For details on how to configure the advanced emulator boards for high frequency operation with the pod, refer to the “Advanced Emulator Configuration for the Pod” section.

The C505L MCUs have 256 bytes of on-chip memory located at XDATA addresses FF00H – FFFFH. The C505L also has the LCD controller registers and real-time clock (RTC) registers mapped to XDATA space at addresses F3DCH – F3FFH. If your application is using the on-chip XRAM or the LCD/RTC registers, you must map the last memory range of the XDATA memory (addresses F000H – FFFFH) to the target. To map the memory, do the following:

1. From the Config menu in Seehau, select **Emulator**. This opens the **Emulator Configuration** dialog box.
2. Select the **Mem Map Config** tab.

POD–51EH–C505L–20 has two operating modes.

- Single-chip
- External

The active mode is selected by the EA pin (pin 29 of the POD–51EH–C505L–20 adapter) during the Reset signal rise (also sampled on internal watchdog timer Reset).

POD–51EH–C505L–20

This pin is sampled by the emulator only during Exit from Reset, and therefore, it is recommended that you connect it to either Constant Low or Constant High. This pin defaults to High (selecting single-chip mode), which is implemented by a 47K pull-up resistor on the pin.

When using external mode, the code memory can be mapped either to the target or to the emulator (with boundaries of 4K). In external mode, addresses are switched to Port2 and Port0 pins for every bus cycle regardless of where it is mapped. The data, however, is read or written to the Port0 pins only for addresses that are mapped to the target.

When using single-chip mode, you need to map the first 32K of the code memory (address range 0 –7FFFH) to the emulator board. In this mode, Port2 and Port0 can be used as I/O ports. These pins carry the address and data information only in the following cases.

- XDATA accesses, using MOVX instructions, to an address that is mapped to the target.
- XDATA accesses to the internal XRAM when the XMAP1 bit in the SYSCON register is set.
- Depending on JP7, for code accesses—see details in the “JP7: External Code Enable” section in the “Jumper Identification and Description” table.

In all other cases, depending on JP7, these pins carry I/O values of Port0 and Port2 either out-putted from the MCU according to the Port0 and Port2 SFR registers value, or in-putted from the target to the MCU.

LED Indicators

Function	Description
Reset	A red LED that indicates when the MCU reset is active. There are five possible sources of the MCU reset. • Reset from the target starts whenever a low level is detected on the reset pin of the adapter. • Reset from the S1 button starts by pressing S1 on the pod. • Click the Reset button in the Seehau menu to reset the emulator. • Reset from the internal watchdog timer starts if the internal watchdog is enabled, and it is not fed by the user software for a long period. (**Note**: It is important to feed the watchdog periodically, otherwise, problems during the debug process might occur.) • When a map error occurs. During all of these reset sequences, the reset LED is turned on for a short or a long period of time depending on the sequence.

POD–51EH–C505L–20

LED Indicators (continued)

Function	Description
Map Error	A red LED that indicates a mapping error occurring in one of the following situations. • When the internal C505L XRAM is enabled by bit XMAP0 in the SYSCON register. • The data memory address range FF00H – FFFFH is mapped to the emulator. • There is a memory access (MOVX) to the internal XRAM. This state is illegal. When the C505L internal XRAM is enabled and accesses are made to it, the data memory address range FF00H – FFFFH needs to be mapped to the target. • When single-chip mode is selected by the EA pin from the target during the last Reset, a code memory address range in the first 32K (address 0 – 7FFFH) is mapped to the target and there is code access to one of the addresses in this range. This state is illegal. In single-chip mode, when JP7 is set to disable code accesses to the target, all code memory needs to be mapped to the emulator. If one of these illegal states occurs, the Map Error (ERR) LED turns on simultaneously with the C505L MCU on the pod entering the reset state, signaling the illegal state. This state is maintained until you push the reset button. Leaving the map error state is done only by pressing the reset button S1 on the pod. This prevents repetitive occurrences of the illegal state until you fix the problem.
Emulation	A green LED that indicates when the user program is running, and turns off when the user program is stopped (emulation break). The user code can be fetched from RAM on the emulator board or from code memory on the target (usually PROM).
Monitor	A yellow LED that indicates when the monitor mode is active. In this mode, the MCU executes internal emulator code that communicates with the host PC. The emulator enters this mode upon power-up or after emulation break.

Jumper Identification and Description

Jumper Designation	Function	Description
Ground SeLect (GSL)	VAGND Grounding Select	This jumper selects how to ground the VAGND pin of the MCU. • When the jumper top is across pins 1 and 2, VAGND is connected to emulator ground. • When the jumper top is across pins 2 and 3, VAGND is grounded to the target. Pin 3 of the jumper is marked by the letter T to show connection to the target. The default position of the jumper top is across pins 1 and 2 (emulator ground selected).

POD–51EH–C505L–20

Jumper Identification and Description (continued)

Jumper Designation	Function	Description
Reference SeLect (RSL)	VAREF Source Select	This jumper selects how to power the VAREF pin of the MCU. • When the jumper top is across pins 1 and 2, VAREF is connected to emulator. • When the jumper top is across pins 2 and 3, VAREF is connected to the target. Pin 3 of the jumper is marked by the letter T to show connection to the target. The default position of the jumper top is across pins 1 and 2 (emulator selected as the voltage source).
Vcc SeLect (VSL)	MCU Vcc Source Select	This jumper selects how to power the Vcc pin of the MCU. • When the jumper top is on pins 1 and 2, the MCU Vcc pin is connected to the Vcc of the emulator. • When the jumper top is on pins 2 and 3, the MCU Vcc pin is connected to the Vcc pin of the adapter. This allows the MCU Vcc pin to be powered by the target. The default position of the jumper top is across pins 1 and 2 (emulator selected as Vcc source). **Note**: Taking power from the target system is not recommended.
X1S, X2S	XTAL1 and XTAL2 Source Select	These two jumpers select the clock source for the MCU. • With the jumper tops on pins 1 and 2, the MCU XTAL1 and XTAL2 pins are connected to a 16-MHz crystal on the pod. • When these two jumpers have jumper tops on pins 2 and 3, the MCU XTAL1 and XTAL2 pins are connected to the target. This allows the target to supply the clock to the MCU XTAL1 and XTAL2 pins. The default position of the jumper top is across pins 1 and 2 (the pod crystal). **Note**: Both of the jumper tops need to be placed in the same position with either both selecting the pod crystal source, or both selecting the target clock source.

POD–51EH–C505L–20

Jumper Identification and Description (continued)

Jumper Designation	Function	Description
JP8	Monitor Stop/Break Select	This jumper determines whether all or only part of the internal MCU peripherals stop in monitor mode after an emulation break occurs. • When the jumper top is out, the stop mode is selected. In this case, all of the internal MCU peripherals are stopped in monitor mode. This includes the following MCU peripherals: interrupts, oscillator watchdog, timer0, timer1, timer2, and USART. • When the jumper top is on, the break mode is selected. In this case, only some of the internal peripherals are stopped. This includes the following MCU peripherals: interrupts, oscillator watchdog, and timer2. Timer0, timer1, and USART are not stopped in this mode. In STOP and BREAK, all of the internal peripherals start working from the same point they stopped before the break as soon as the user program starts running again when you press GO or STEP. When changing this jumper, the new stop/break mode takes effect after the next reset sequence. The default setup of the jumper is stop mode selection (jumper top out). **Note**: It is recommended to work in stop mode only. The break mode needs to be selected only if your application has limitations that do not allow the timers to be stopped in monitor mode.
XL0..XL7	Trace Port Select Jumpers	These eight 3-pin jumpers select the signals that are routed to the trace board and are shown in the trace window under the header P1. For any jumper in this group the jumper top position pin 2–pin 3 means that the corresponding pin of the recreated Port0 is traced. pin 1 of each jumper is left free and can be connected to any signal on the target or port headers on the pod. For example setting XL4 in position 2–3 means that pin P0.4 is traced. If pin XL4.1, for example, is connected to a signal P4.3 on the target, you can choose between tracing signal P0.4 or P4.3 by moving the jumper between positions 2–3 and 1–2.
XH0..XH7	Trace Port Select Jumpers	This group of jumpers is similar to XL0..XL7, except pin 3 of each jumper is connected to the recreated Port2 and pin 1 is left free.
JP7	External Code Enable	This jumper enables access to the external code if the single-chip mode is emulated and part of the code is mapped into the target code memory. The external code is enabled if the jumper top is out (default jumper configuration).
Reset Button		When pressed, this switch resets the pod (including the MCU). You can use this button to recover from a map error status.
JP10	SY0, SY1	This header contains two pins; pin 1 (SY0), and pin 2 (SY1). These two signals are carried to the emulator and trace boards, and can be used in the trace and breakpoint logic.

POD–51EH–C505L–20

Jumper Identification and Description (continued)

Jumper Designation	Function	Description
JP9	EMUL, FLF, ANB	This header contains three pins: pin 1 (EMUL), pin 2 (FLF), and pin 3 (ANB). Pin 1 signals emulation or monitor mode. It is low when emulation is active (when the user program is running). Pins 2 and 3 are not used currently, but can be used in the future.
J2	P3.6, P3.7	This header carries the recreated signals; P3.6/WR (pin J2.1) and P3.7/RD (pin J2.2). These pins observe the signals or route them to the trace board through the headers XL0..XL7, XH0..XH7.
P1, P3, P4, P5	Port Headers	These headers carry signals for the corresponding ports. The port pin numbers are marked on the silk-screen. For example, pin 5 of header P3 connects to pin P3.5 of the MCU.

Advanced Emulator Configuration for the Pod

When using an advanced emulator board (part numbers EMUL51–PC/EA256–BSW–50 or the EMUL51–PC/EA768–BSW–50), it is important to configure the advanced emulator board for the POD–C505L, especially when applying a high frequency clock signal above 12 MHz to the C505L MCU.

Make sure that the two following jumpers in the advanced emulator board are removed.

- The DAL jumper on JP1
- The RWEN jumper on JP1

These two jumpers affect the operation mode and maximum operating frequency of the advanced emulator board.

For details on how to configure the other jumpers on the advanced emulator board, refer to Chapter 2, “Installing the Hardware” (section “EMUL51–PC/EA256 – 256K Bank Switch Emulator Configuration,” or section “EMUL51–PC/EA768 – 768K Bank Switch Emulator Configuration”).

Differences Between the Emulator and the C505L MCU

EA Pin

The EA pin in the C505L MCU is an input, floating pin used to select between single-chip mode (the program in internal OTP–ROM) and expanded mode (the program in external memory). This pin does not have a pull-up or a pull-down resistor.

A 47K pull-up resistor on the pod is connected to this pin to select a default state of single-chip mode when this pin is not forced by the target.

POD–51EH–C505L–20

This difference of the existence of the pull-up resistor must be considered: if the target pin is tied to a pull-down resistor, it should be a sufficiently low pull-down (3.3K or lower) in order to overcome the 47K pull-up resistor on the pod.

In the C505L MCU during level 0 protection, the EA pin is not sampled during Reset and can be switched dynamically during run-time without going through Reset. This mode is not supported by the pod. The EA pin from the target is always sampled during exit from Reset (Reset Signal Fall time), and the mode of operation is then set to either single-chip or expanded mode until the next Reset.

Port0 and Port2 Pins

These 16 port pins are emulated by the pod and are used both as the address/data bus and as port pins. These pins have slightly different AC and DC characteristics from the Port0 and Port2 pins of the C505L MCU. They sink up to 12mA and source up to 4mA, while maintaining valid TTL output logic levels. Port2 pins also have pull-up resistors of 22K, which might be different from the C505L MCU's internal pull-ups.

Port3.6/WR and Port3.7/RD Pins

These two port pins are emulated by the pod and are used as the read and write signals or as port pins. They have slightly different AC and DC characteristics from the Port3.6/WR and Port3.7/RD pins of the C505L MCU. These pins sink up to 12mA and source up to 4mA, (on transition from logic 0 to 1) while maintaining valid TTL output logic levels. They have pull-up resistors of 22K to maintain a logic 1 level and emulate the quasi-bi-directional operation. These features differ from the internal pull-ups of the C505L MCU.

POD–51EH–C508–10

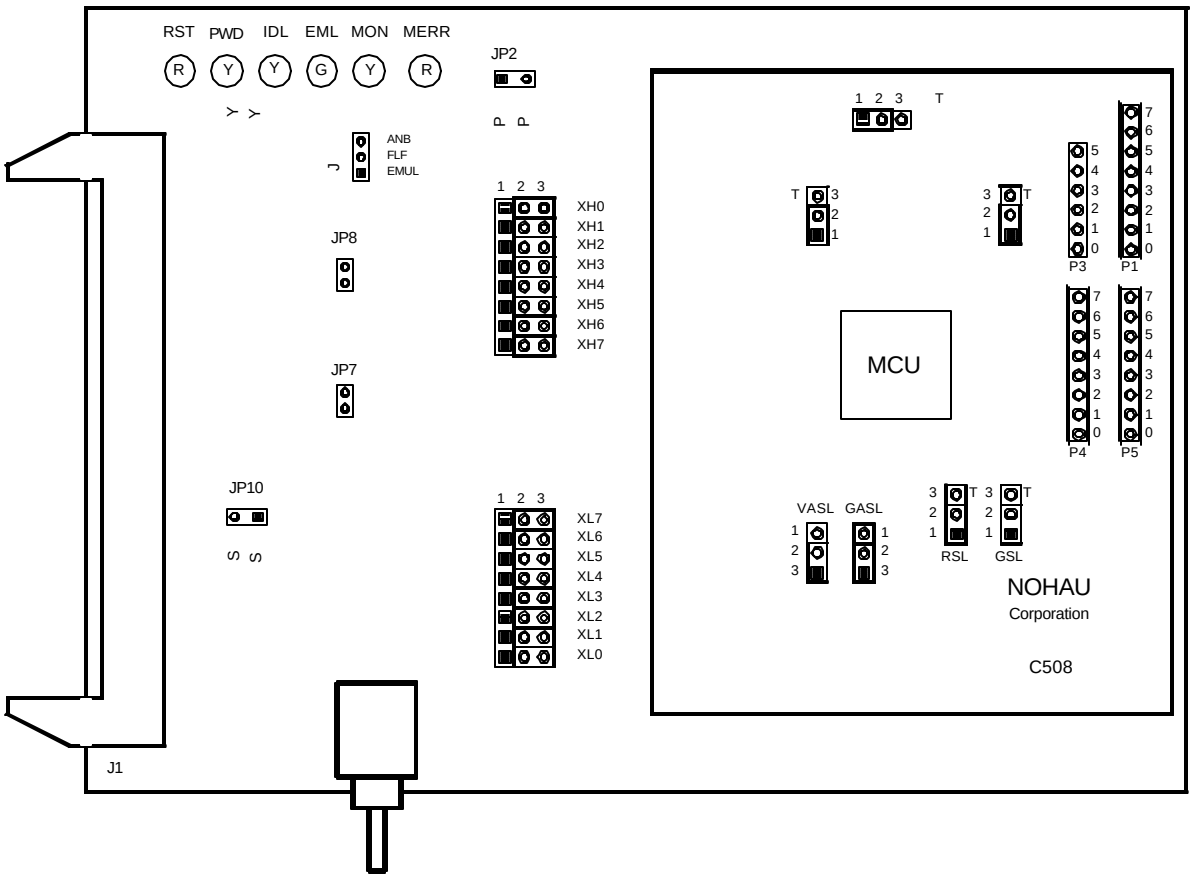


Figure 82. POD–51EH–C508–10

Operating Instructions

The POD–51EH–C508–10 supports the Siemens C508 MCU, which is based on the C500 core.

To work with this pod, you need to select **POD–EH–C508** from the **Select Processor** drop-down list box (**Step 3** of the **Communications** tab in the Emulator Configuration window).

The EPROM on the emulator board must be version COM 1.4.

The pod assembly consists of two sub-assemblies: The larger universal Port Replacement Unit (PRU) and the smaller processor module daughterboard.

The PRU has three headers on the bottom that carry the signals to the target. The PRU also has a 50-pin connector J1 that connects through the ribbon cable to the emulator board.

POD–51EH–C508–10

The target headers plug into an adapter that you select depending on the type of MCU. For example, the C508 MCU uses the following adapter:
Emulation Solutions ES/180–5545–00 (P–MQFP–64)

The adapter plugs into the target system, replacing the C508 MCU IC. This adapter can be ordered from www.icetech.com or directly from Emulation Solutions.

The POD–51EH–C508–10 runs up to a maximum frequency of 10 MHz.

The C508 MCU has on-chip XRAM memory located at addresses FC00H – FFFFH in the data space. If your application uses the on-chip XRAM, you must map the last memory range of the data memory (addresses F000H – FFFFH) to the target.

POD–51EH–C508–10 has two operating modes.

- Single-chip
- External

The EA pin during reset signal rise (also sampled on internal watchdog timer reset) selects the operating mode. It is recommended that you connect this pin to either constant low or constant high. This pin defaults to high (selecting single-chip mode), that is implemented by a 22K pull-up resistor on the pin.

The active mode is selected by the EA pin during reset signal rise (also sampled on internal reset).

When using external mode, code memory can be mapped to the target or the emulator (with boundaries of 4K). In external mode, addresses are seen on the Port2 and Port0 adapter pins for every bus cycle regardless of where it is mapped. The data, however, is read or written to the Port0 pins only for addresses that are mapped to the target.

When using single-chip mode, you need to map all code memory to the emulator board. In this mode, Port2 and Port0 can be used as I/O ports.

POD–51EH–C508–10 also supports tracing of internal XRAM read and write cycles using MOVX instructions. This is supported in both single-chip or external modes.

POD–51EH–C508–10

LED Indicators

Function	Description
Reset	A red LED that indicates the MCU reset is active. There are five possible sources of the MCU reset: • Reset from the target starts whenever a low level is detected on the reset pin of the adapter. • Reset from the S1 button starts by pressing the S1 button on the pod. • Click the Reset button in the Seehau menu to reset the emulator. • Reset from the internal watchdog timer starts if the internal watchdog is enabled, and the user software does not feed it for a long period. (**Note**: It is important to feed the watchdog periodically, otherwise, problems during the debug process might occur.) • When a map error occurs. During all of these reset sequences, the reset LED is turned on for a short or long period depending on the sequence
Map Error	A red LED that indicates a mapping error has occurred in one of the following situations. • The internal C508 XRAM is enabled by bit XMAP0 in the SYSCON register, the data memory address range F000H – FFFFH is mapped to the emulator, and there is a memory access (MOVX) to the internal XRAM. This state is illegal. When the C508 internal XRAM is enabled and accesses are made to it, the data memory address range F000H – FFFFH needs to be mapped to the target. • The EA pin from the target (during the last reset) selects single-chip mode, a code memory address range is mapped to the target, and there is code access to one of the addresses in this range. This state is illegal. In single-chip mode all of the code memory needs to be mapped to the emulator and not to the target, because the 64K code memory is internal to the C508 MCU. If one of these illegal states occurs, the Map Error LED turns on simultaneously with the C508 MCU entering the reset state, signaling the illegal state. This state is maintained until you push the reset button. You can only leave the map error state by pressing the reset button S1 on the pod. This prevents repetitive occurrences of the illegal state until you fix the problem.
Emulation	A green LED that indicates the user program is running, and turns off whenever the user program is stopped (emulation break). The user code can be fetched from RAM on the emulator board or from code memory on the target (usually PROM).
Monitor	A yellow LED that indicates the monitor mode is active. In this mode, the MCU executes internal emulator code that communicates with the host PC. The emulator enters this mode upon power-up or after emulation break.

POD–51EH–C508–10

Jumper Identification and Description

Jumper Designation	Function	Description
Ground SeLect (GSL)	VAGND Grounding Select	This jumper selects how to ground the VAGND pin of the MCU. ● When the jumper top is positioned across pins 1 and 2, VAGND is connected to emulator ground. ● When the jumper top is positioned across pins 2 and 3, VAGND is grounded to the target adapter. The default position of the jumper top is across pins 1 and 2 (emulator ground is selected).
Reference SeLect (RSL)	VAREF Source Select	This jumper selects how to power the VAREF pin of the MCU. ● When the jumper top is positioned across pins 1 and 2, VAREF is connected to emulator. ● When the jumper top is positioned across pins 2 and 3, VAREF is connected to the target. The default position of the jumper top is across pins 1 and 2 (emulator is selected as the voltage source).
Vcc SeLect (VSL)	MCU Vcc Source Select	This jumper selects how to power the Vcc pins of the MCU. ● When the jumper top is positioned on pins 1 and 2, the MCU Vcc pins are connected to the Vcc of the emulator. ● When the jumper top is positioned on pins 2 and 3, the MCU Vcc pins are connected to the Vcc pins of the adapter. This allows the MCU Vcc pins to be powered by the target. The default position of the jumper top is across pins 1 and 2 (emulator selected as Vcc source). **Note**: Taking power from the target system is not recommended.
Analog Ground Select (GASL)	VSSA Source Select	This jumper selects how to ground the VSSA pin of the MCU. ● When the jumper top is positioned across pins 1 and 2, VSSA is connected to the emulator ground. ● When the jumper top is positioned across pins 2 and 3, VSSA is grounded to the target adapter. The default position of the jumper top is across pins 1 and 2 (emulator ground is selected).
Analog Power Select (VASL)	VDDA Source Select	This jumper selects how to power the VDDA pin of the MCU. ● When the jumper top is positioned across pins 1 and 2, VDDA is connected to the emulator 5-V power supply. ● When the jumper top is positioned across pins 2 and 3, VDDA is connected to the target power supply through the target adapter. The default position of the jumper top is across pins 1 and 2 (emulator power is selected as the voltage source).

POD–51EH–C508–10

Jumper Identification and Description (continued)

Jumper Designation	Function	Description
X1S, X2S	XTAL1 and XTAL2 Source Select	These two jumpers select the clock source for the MCU. • With the jumper tops on pins 1 and 2, the MCU XTAL1 and XTAL2 pins are connected to a 24-MHz crystal on the pod. • When these two jumpers have jumper tops on pins 2 and 3, the MCU XTAL1 and XTAL2 pins are connected to the adapter. This allows the target to supply the clock to the MCU XTAL1 and XTAL2 pins. The default position of the jumper top is across pins 1 and 2 (the pod crystal). **Note**: Both of the jumper tops need to be placed in the same position with both selecting the pod crystal source or both selecting the target clock source.
JP8	Monitor Stop/Break Select	This jumper determines whether all or only part of the internal MCU peripherals stop in monitor mode after an emulation break occurs. • When the jumper top is out, the stop mode is selected. In this case, all of the internal MCU peripherals are stopped in monitor mode. This includes the following MCU peripherals: interrupts, watchdog timer, oscillator watchdog, timer0, timer1, timer2, and USART. • When the jumper top is on, the break mode is selected. In this case, only some of the internal peripherals are stopped. This includes the following MCU peripherals: interrupts, watchdog timer, oscillator timer, and timer2. Timer0, timer1, and USART are not stopped in this mode. • In stop and break mode, all of the internal peripherals start working from the same point they stopped before the break. This happens as soon as the user program starts running again when you press the Go or Step buttons. When changing this jumper, the new stop/break mode takes effect after the next reset sequence. The default setup of the jumper is stop mode selection (jumper top out). **Note**: It is recommended to work in stop mode only. The break mode needs to be selected only if your application has limitations that do not allow the timers to be stopped in monitor mode.
XL0..XL7	Trace Port Select Jumpers	These eight 3-pin jumpers select the signals that are routed to the trace board and are shown in the trace window under the header P1. For any jumper in this group, the jumper right position pin 2–pin 3 indicates that the corresponding pin of the re-created Port0 is traced. Pin 1 of each jumper is left free and can be connected to any signal on the target or port headers on the pod. For example, setting XL4 in position 2–3 indicates that pin P0.4 is traced. If pin XL4.1, for example, is connected to a signal P4.3 on the target, you can choose between tracing signal P0.4 or P4.3 by moving the jumper between positions 2–3 and 1–2.
XH0..XH7	Trace Port Select Jumpers	This group of jumpers is similar to XL0..XL7, except pin 3 of each jumper is connected to the recreated Port2 and pin 1 is left free.

POD–51EH–C508–10

Jumper Identification and Description (continued)

Jumper Designation	Function	Description
JP7	External Code Enable	This jumper enables access to the external code if the single-chip mode is emulated and part of the code is mapped into the target code memory. The external code is enabled if the jumper top is out (default jumper configuration).
Reset Button		When pressed, this switch resets the pod (including the MCU). You can use this button to recover from a map error status.
JP10	SY0, SY1	This header contains two pins: Pin 1 (SY0), and pin 2 (SY1). These two signals are carried to the emulator and trace boards, and can be used in the trace and breakpoint logic.
JP9	EMUL, FLF, ANB	This header contains three pins: Pin 1 (EMUL), pin 2 (FLF), and pin 3 (ANB). Pin1 signals emulation or monitor mode. It is low when emulation is active (when the user program is running). Pins 2 and 3 are not used currently, but can be used in the future.
J2	P3.6, P3.7	This header carries the recreated signals: P3.6/WR (pin J2.1) and P3.7/RD (pin J2.2). These pins observe the signals or route them to the trace board through the headers XL0..XL7, XH0..XH7.
P1, P3, P4, P5	Port Headers	These headers carry signals for the corresponding ports. The port pin numbers are marked on the silk-screen. For example, pin 5 of header P4 connects to pin P4.5 of the MCU.

Differences Between the Emulator and the C508 MCU

EA Pin

The EA pin in the C508 MCU is an input, floating pin used to select between single-chip mode (the program in internal OTP–ROM) and external mode (the program in external memory). This pin does not have a pull-up or a pull-down resistor.

A 47K pull-up resistor on the pod is connected to this pin to select a default state of single-chip mode when this pin is not forced by the target.

This difference of the existence of the pull-up resistor must be considered: if the target pin is tied to a pull-down resistor, it should be a sufficiently low pull-down (3.3K or lower) in order to overcome the 47K pull-up resistor on the pod.

In the C508 MCU during level 0 protection, the EA pin is not sampled during Reset and can be switched dynamically during run-time without going through Reset. This mode is not supported by the pod. The EA pin from the target is always sampled during exit from Reset (Reset Signal Fall time), and the mode of operation is then set to either single-chip or external mode until the next Reset.

POD–51EH–C508–10

Port0 and Port2 Pins

These 16 port pins are emulated by the pod and are used both as the address/data bus and as port pins. These pins have slightly different AC and DC characteristics from the Port0 and Port2 pins of the C508 MCU. They sink up to 12mA and source up to 4mA, while maintaining valid TTL output logic levels. Port2 pins also have pull-up resistors of 22K, which might be different from the C508 MCU's internal pull-ups.

POD–51EH–C513AO

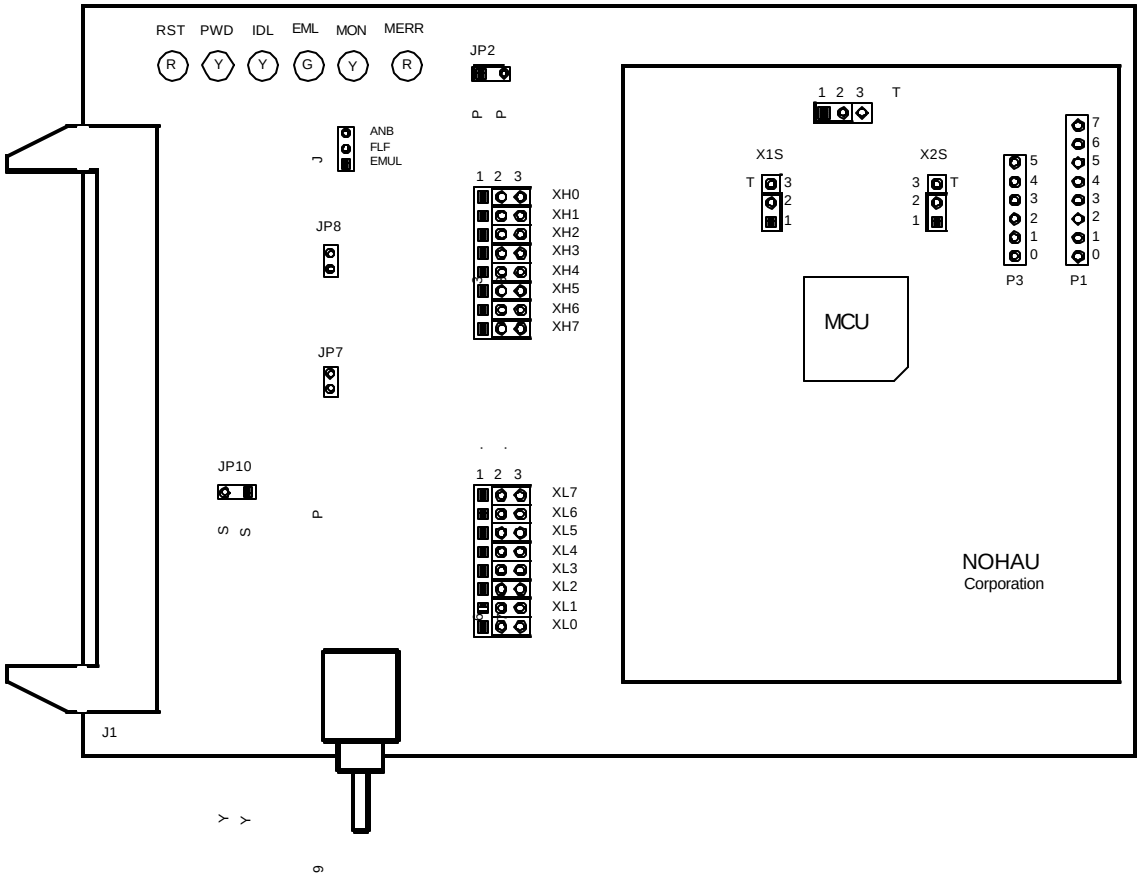


Figure 83. POD–51EH–C513AO

Operating Instructions

The POD–51EH–C513AO–16 supports the Siemens C513AO MCU which is based on the C500 core.

To work with this pod, you need to select **POD–EH–C513AO** from the **Select Processor** drop-down list box (**Step 3** of the **Communications** tab in the Emulator Configuration window).

The EPROM on the emulator board must be version COM 1.4.

The pod assembly consists of two sub-assemblies: The larger universal Port Replacement Unit (PRU) and the smaller processor module daughterboard.

The PRU provides a 44-pin PGA socket on the bottom that you can connect to the target using an adapter. The PRU also has a 50-pin header J1 that connects through the ribbon cable to the emulator board.

POD–51EH–C513AO

The adapters plug into the target system, replacing the C513AO MCU IC. The target adapter part numbers are
EDI/44PG/QFS31–SD
PGA44–PLCC44 (PLCC)

The POD–51EH–C513AO–16 runs up to a maximum frequency of 16 MHz.

The C513AO MCU has on-chip XRAM memory located at addresses FF00H – FFFFH in the data (XRAM) space. If your application uses the on-chip XRAM, you must map the last memory range of the data memory (addresses F000H – FFFFH) to the target.

The active mode is selected by the EA pin (pin 29 of the target adapter) during reset signal rise (also sampled on internal watchdog timer reset). It is recommended that you connect this pin to either constant low or constant high. This pin defaults to high (selecting single-chip mode), which is implemented by a 22K pull-up resistor on the pin.

POD–51EH–C513AO–16 has two operating modes.

- Single-chip
- External

When using external mode, the code memory can be mapped either to the target or the emulator (with boundaries of 4K). In external mode, addresses are seen on the Port2 and Port0 adapter pins for every bus cycle regardless of where it is mapped. The data, however, is read or written to the Port0 pins only for addresses that are mapped to the target.

When using single-chip mode, you need to map all code memory to the emulator board. In this mode, Port2 and Port0 can be used as I/O ports.

POD–51EH–C513AO–16 also supports tracing of internal XRAM read and write cycles using MOVX instructions. This is supported in both single-chip or external modes.

POD-51EH-C513AO

LED Indicators

Function	Description
Reset	A red LED that indicates when the MCU reset is active. There are five possible sources of the MCU reset. • Reset from the target starts whenever a low level is detected on the reset pin of the adapter. • Reset from the S1 button starts by pressing S1 on the pod. • Click the Reset button in the Seehau menu to reset the emulator. • Reset from the internal watchdog timer starts if the internal watchdog is enabled, and it is not fed by the user software for a long period. (**Note**: It is important to feed the watchdog periodically, otherwise, problems during the debug process might occur.) • When a map error occurs. During all of these reset sequences, the reset LED is turned on for a short or a long period of time depending on the sequence.
Map Error	A red LED that indicates a mapping error occurring in one of the following situations. • When the internal C513AO XRAM is enabled by bit XMAP in the SYSCON register. • The data memory address range F000H – FFFFH is mapped to the emulator. • There is a memory access (MOVX) to the internal XRAM. This state is illegal. When the C513AO internal XRAM is enabled and accesses are made to it, the data memory address range F000H – FFFFH needs to be mapped to the target. If one of these illegal states occurs, the Map Error (ERR) LED turns on simultaneously with the C513AO MCU entering the reset state, signaling the illegal state. This state is maintained until you push the reset button. Leaving the map error state is done only by pressing the reset button S1 on the pod. This prevents repetitive occurrences of the illegal state until you fix the problem.
Emulation	A green LED that indicates when the user program is running, and turns off when the user program is stopped (emulation break). The user code can be fetched from RAM on the emulator board or from code memory on the target (usually PROM).
Monitor	A yellow LED that indicates when the monitor mode is active. In this mode, the MCU executes internal emulator code that communicates with the host PC. The emulator enters this mode upon power-up or after emulation break.

POD–51EH–C513AO

Jumper Identification and Description

Jumper Designation	Function	Description
Vcc SeLect (VSL)	MCU Vcc Source Select	This jumper selects how to power the Vcc pin of the MCU. • When the jumper top is on pins 1 and 2, the MCU Vcc pin is connected to the Vcc of the emulator. • When the jumper top is on pins 2 and 3, the MCU Vcc pin is connected to the Vcc pin of the adapter. This allows the MCU Vcc pin to be powered by the target. The default position of the jumper top is across pins 1 and 2 (emulator selected as Vcc source). **Note**: Taking power from the target system is not recommended.
X1S, X2S	XTAL1 and XTAL2 Source Select	These two jumpers select the clock source for the MCU. • With the jumper tops on pins 1 and 2, the MCU XTAL1 and XTAL2 pins are connected to a 16-MHz crystal on the pod. • When these two jumpers have jumper tops on pins 2 and 3, the MCU XTAL1 and XTAL2 pins are connected to pins 15 and 14 of the adapter. This allows the target to supply the clock to the MCU XTAL1 and XTAL2 pins. The default position of the jumper top is across pins 1 and 2 (the pod crystal). **Note**: Both of the jumper tops need to be placed in the same position with either both selecting the pod crystal source, or both selecting the target clock source.
JP8	Monitor Stop/Break Select	This jumper determines whether all or only part of the internal MCU peripherals stop in monitor mode after an emulation break occurs. • When the jumper top is out, the stop mode is selected. In this case, all of the internal MCU peripherals are stopped in monitor mode. • When the jumper top is in, the break mode is selected. In this case, only some of the internal peripherals are stopped. This includes the following MCU peripherals: interrupts, watchdog timer, and oscillator watchdog. Timer0, timer1, and timer2 capture are not stopped in this mode. In stop and break mode, all of the internal peripherals start working from the same point they stopped before the break as soon as the user program starts running again when you press the Go or Step buttons. When changing this jumper, the new stop/break mode takes effect after the next reset sequence. The default setup of the jumper is stop mode selection (jumper top out). **Note**: It is recommended to work in stop mode only. The break mode needs to be selected only if your application has limitations that do not allow the timers to be stopped in monitor mode.

POD-51EH-C513AO

Jumper Identification and Description (continued)

Jumper Designation	Function	Description
XL0..XL7	Trace Port Select Jumpers	These eight 3-pin jumpers select the signals that are routed to the trace board and are shown in the Trace window under the header P1. For any jumper in this group the jumper top position pin 2–3 means that the corresponding pin of the recreated Port0 is traced. Pin 1 of each jumper is left free and can be connected to any signal on the target or port headers on the pod. For example setting XL4 in position 2–3 means that pin P0.4 is traced. If pin XL4.1, for example, is connected to a signal P4.3 on the target, you can choose between tracing signal P0.4 or P4.3 by moving the jumper between positions 2–3 and 1–2.
XH0..XH7	Trace Port Select Jumpers	This group of jumpers is similar to XL0..XL7, except pin 3 of each jumper is connected to the recreated Port2 and pin 1 is left free.
JP7	External Code Enable	This jumper enables access to the external code if the single-chip mode is emulated and part of the code is mapped into the target code memory
Reset Button		When pressed, this switch resets the pod (including the MCU). You can use this button to recover from a map error status.
JP10	SY0, SY1	This header contains two pins: pin 1 (SY0), and pin 2 (SY1). These two signals are carried to the emulator and trace boards, and can be used in the trace and breakpoint logic.
JP9	EMUL, FLF, ANB	This header contains three pins: pin 1 (EMUL), pin 2 (FLF), and pin 3 (ANB). Pin 1 signals emulation or monitor mode. It is low when emulation is active (when the user program is running). Pins 2 and 3 are not used currently, but can be used in the future.
J2	P3.6, P3.7	This header carries the recreated signals; P3.6/WR (pin J2.1) and P3.7/RD (pin J2.2). These pins observe the signals or route them to the trace board through the headers XL0..XL7, XH0..XH7.
P1, P3, P4	Port Headers	These headers carry signals for the corresponding ports. The port pin numbers are marked on the silk-screen. For example, pin 5 of header P3 connects to pin P3.5 of the MCU. **Note**: Earlier revisions of the pod might have header JP15 installed on the board. This header is for factory testing only and is not for any user functions.

Differences Between the Emulator and the C513AO MCU

EA Pin

The EA pin in the C513AO MCU is an input, floating pin used to select between single-chip mode (the program in internal OTP–ROM) and expanded mode (the program in external memory). This pin does not have a pull-up or a pull-down resistor.

A 47K pull-up resistor on the pod is connected to this pin to select a default state of single-chip mode when this pin is not forced by the target.

POD–51EH–C513AO

This difference of the existence of the pull-up resistor must be considered: if the target pin is tied to a pull-down resistor, it should be a sufficiently low pull-down (3.3K or lower) in order to overcome the 47K pull-up resistor on the pod.

In the C513AO MCU during level 0 protection, the EA pin is not sampled during Reset and can be switched dynamically during run-time without going through Reset. This mode is not supported by the pod. The EA pin from the target is always sampled during exit from Reset (Reset Signal Fall time), and the mode of operation is then set to either single-chip or expanded mode until the next Reset.

Port0 and Port2 Pins

These 16 port pins are emulated by the pod and are used both as the address/data bus and as port pins. These pins have slightly different AC and DC characteristics from the Port0 and Port2 pins of the C513AO MCU. They sink up to 12mA and source up to 4mA, while maintaining valid TTL output logic levels. Port2 pins also have pull-up resistors of 22K, which might be different from the C513AO MCU's internal pull-ups.

POD–51EH–C515C–10

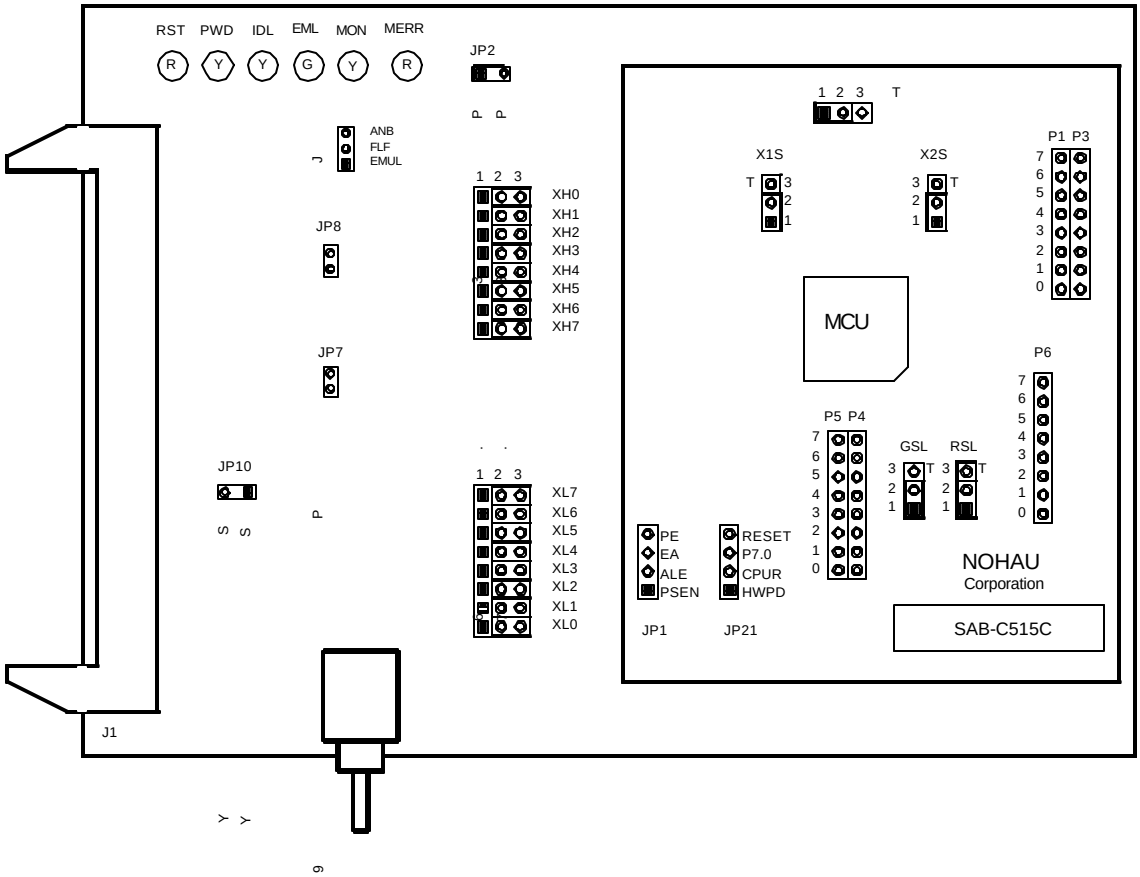


Figure 84. POD–51EH–C515C–10

Operating Instructions

The POD–51EH–C515C–10 supports the Siemens C515C MCU which is based on the C500 core.

To work with this pod, you need to select **POD–EH–C515C** from the **Select Processor** drop-down list box (**Step 3** of the **Communications** tab in the Emulator Configuration window).

> **Note**
>
> POD–51EH–C515C–10 replaces POD–C515C–10, an older version of the pod for the Siemens SAB–C515C MCU. If you have the older version of the pod, you need to pay attention to the position of the pod relative to your target. Compared to the old pod, the new one needs to turn 90 degrees clockwise when you install it on the same target. (See Figure 85.)

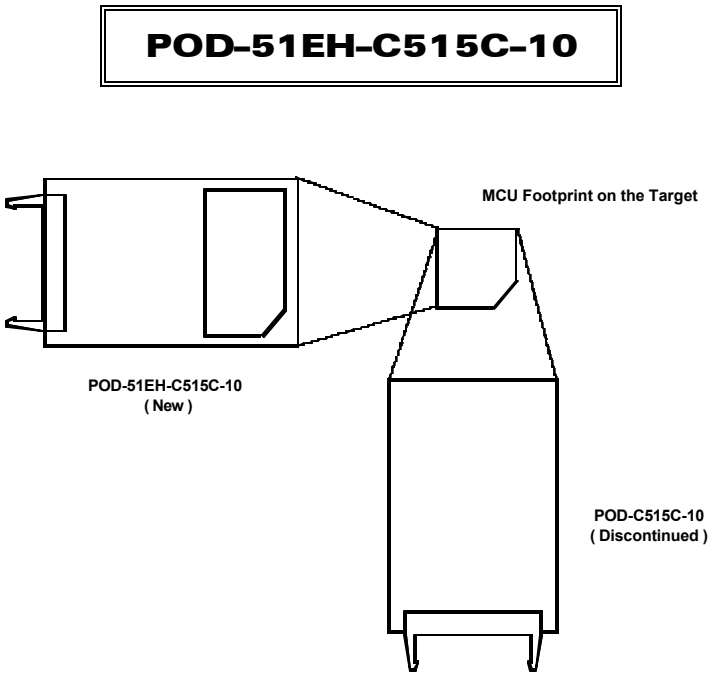


Figure 85. Pod Orientation Relative to the MCU Footprint on the Target

The EPROM on the emulator board must be version COM 1.4.

The pod assembly consists of two sub-assemblies: The larger universal Port Replacement Unit (PRU) and the smaller processor module daughterboard.

The PRU has three headers on the bottom that carry the signals to the target. For the packages with the small pinout (44-pin), it also provides a 44-pin PGA socket on the bottom that you can connect to the target using an adapter. The PRU also has a 50-pin header J1 that connects through the ribbon cable to the emulator board.

The target headers plug into an adapter that you select depending on the type of the MCU. For example, the SAB–C515C MCU uses the following adapters:

- Emulation Solutions ES/180–5550–50 (socketable)
- Emulation Solutions ES/180–5550–55 (non-socketable)

The adapters plug into the target system, replacing the C515C MCU IC. These adapters can be ordered from www.icetech.com or directly from Emulation Solutions.

The POD–51EH–C515C–10 runs up to a maximum frequency of 10 MHz. However, the internal MCU circuits are clocked with the frequency of the oscillator clock, unlike most 8051 derivatives that normally divide the oscillator frequency signal by two. Therefore, you need to input twice the operating oscillator frequency in the CLK field in the Hardware Configuration window. For example, input 20 MHz in the CLK field for a 10-MHz external crystal. In addition, the pod can only function correctly with 20 MHz or higher frequency emulator and trace boards.

POD–51EH–C515C–10

The C515C MCU has on-chip XRAM memory located at addresses F700H – FFFFH in the data (XRAM) space. If your application uses the on-chip XRAM, you must map the last memory range of the data memory (addresses F000H – FFFFH) to the target.

The active mode is selected by the EA pin (pin 49 of the target adapter) during reset signal rise (also sampled on internal watchdog timer reset). It is recommended that you connect this pin to either constant low or constant high. This pin defaults to high (selecting single-chip mode), which is implemented by a 22K pull-up resistor on the pin.

POD–51EH–C515C–10 has two operating modes.

- Single-chip
- External

The active mode is selected by the EA pin (pin 49 of the pod adapter) during reset signal rise (also sampled on internal reset).

When using external mode, the code memory can be mapped either to the target or the emulator (with boundaries of 4K). In external mode, addresses are seen on the Port2 and Port0 adapter pins for every bus cycle regardless of where it is mapped. The data, however, is read or written to the Port0 pins only for addresses that are mapped to the target.

When using single-chip mode, you need to map all code memory to the emulator board. In this mode, Port2 and Port0 can be used as I/O ports.

POD–51EH–C515C–10 also supports tracing of internal XRAM read and write cycles using MOVX instructions. This is supported in both single-chip or external modes.

POD-51EH-C515C-10

LED Indicators

Function	Description
Reset	A red LED that indicates the when the MCU reset is active. There are five possible sources of the MCU reset. • Reset from the target starts whenever a low level is detected on the reset pin of the adapter. • Reset from the S1 button starts by pressing S1 on the pod. • Click the Reset button in the Seehau menu to reset the emulator. • Reset from the internal watchdog timer starts if the internal watchdog is enabled, and it is not fed by the user software for a long period. (**Note**: It is important to feed the watchdog periodically, otherwise, problems during the debug process might occur.) • When a map error occurs. During all of these reset sequences, the reset LED is turned on for a short or a long period of time depending on the sequence.
Map Error	A red LED that indicates a mapping error occurring in one of the following situations: • The internal C515C XRAM is enabled by bit XMAP0 in the SYSCON register. • The data memory address range F000H – FFFFH is mapped to the emulator. • There is a memory access (MOVX) to the internal XRAM. This state is illegal. When the C515C internal XRAM is enabled and accesses are made to it, the data memory address range F000H – FFFFH needs to be mapped to the target. Single-chip mode is selected by the EA pin from the target (during the last reset), a code memory address range is mapped to the target, and there is code access to one of the addresses in this range. This state is illegal. In single-chip mode all of the code memory needs to be mapped to the emulator and not to the target, because the 64K code memory is internal to the C515C MCU. If one of these illegal states occurs, the Map Error LED turns on simultaneously with the C515C MCU entering the reset state, signaling the illegal state. This state is maintained until you push the reset button. Leaving the map error state is done only by pressing the reset button S1 on the pod. This prevents repetitive occurrences of the illegal state until you fix the problem.
Emulation	A green LED that indicates when the user program is running, and turns off when the user program is stopped (emulation break). The user code can be fetched from RAM on the emulator board or from code memory on the target (usually PROM).
Monitor	A yellow LED that indicates when the monitor mode is active. In this mode, the MCU executes internal emulator code that communicates with the host PC. The emulator enters this mode upon power-up or after emulation break.

POD–51EH–C515C–10

Jumper Identification and Description

Jumper Designation	Function	Description
Ground SeLect (GSL)	VAGND Grounding Select	This jumper selects how to ground the VAGND pin of the MCU. • When the jumper top is across pins 1 and 2, VAGND is connected to emulator ground. • When the jumper top is across pins 2 and 3, VAGND is grounded through pin 4 of the target adapter. Pin 3 of the jumper is marked by the letter T to show connection to the target. The default position of the jumper top is across pins 1 and 2 (emulator ground selected).
Reference SeLect (RSL)	VAREF Source Select	This jumper selects how to power the VAREF pin of the MCU. • When the jumper top is across pins 1 and 2, VAREF is connected to the emulator. • When the jumper top is across pins 2 and 3, VAREF is connected to the target through pin 3 of the target adapter. Pin 3 of the jumper is marked by the letter T to show connection to the target. The default position of the jumper top is across pins 1 and 2 (emulator selected as the voltage source).
Vcc SeLect (VSL)	MCU Vcc Source Select	This jumper selects how to power the Vcc pins of the MCU. • When the jumper top is on pins 1 and 2, the MCU Vcc pins are connected to the Vcc of the emulator. • When the jumper top is on pins 2 and 3, the MCU Vcc pins are connected to the Vcc pins of the adapter. This allows the MCU Vcc pins to be powered by the target. The default position of the jumper top is across pins 1 and 2 (emulator selected as Vcc source). **Note**: Taking power from the target system is not recommended. Both Vcc pins of the MCU are shorted together.
X1S, X2S	XTAL1 and XTAL2 Source Select	These two jumpers select the clock source for the MCU. • With the jumper tops on pins 1 and 2, the MCU XTAL1 and XTAL2 pins are connected to a 10-MHz crystal on the pod. • When these two jumpers have jumper tops on pins 2 and 3, the MCU XTAL1 and XTAL2 pins are connected to pins 37 and 36 of the adapter. This allows the target to supply the clock to the MCU XTAL1 and XTAL2 pins. The default position of the jumper top is across pins 1 and 2 (the pod crystal). **Note**: Both of the jumper tops need to be placed in the same position with either both selecting the pod crystal source, or both selecting the target clock source.

POD–51EH–C515C–10

Jumper Identification and Description (continued)

Jumper Designation	Function	Description
JP8	Monitor Stop/Break Select	This jumper determines whether all or only part of the internal MCU peripherals stop in monitor mode after an emulation break occurs. • When the jumper top is out, the stop mode is selected. In this case, all of the internal MCU peripherals are stopped in monitor mode. This includes the following MCU peripherals: interrupts, watchdog timer, timer0, timer1, timer2, and timer2 capture. • When the jumper top is on, the break mode is selected. In this case, only some of the internal peripherals are stopped. This includes the following MCU peripherals: interrupts, watchdog timer, and timer2. Timer0, timer1, and timer2 capture are not stopped in this mode. In stop and break mode, all of the internal peripherals start working from the same point they stopped before the break as soon as the user program starts running again when you press the Go or Step buttons. When changing this jumper, the new stop/break mode takes effect after the next reset sequence. The default setup of the jumper is stop mode selection (jumper top out). **Note**: It is recommended to work in stop mode only. The break mode needs to be selected only if your application has limitations that do not allow the timers to be stopped in monitor mode.
XL0..XL7	Trace Port Select Jumpers	These eight 3-pin jumpers select the signals that are routed to the trace board and are shown in the Trace window under the header P1. For any jumper in this group the jumper top position pins 2–3 means that the corresponding pin of the recreated Port0 is traced. Pin 1 of each jumper is left free and can be connected to any signal on the target or port headers on the pod. For example setting XL4 in position 2–3 means that pin P0.4 is traced. If pin XL4.1, for example, is connected to a signal P4.3 on the target, you can choose between tracing signal P0.4 or P4.3 by moving the jumper between positions 2–3 and 1–2.
XH0..XH7	Trace Port Select Jumpers	This group of jumpers is similar to XL0..XL7, except pin 3 of each jumper is connected to the recreated Port2 and pin 1 is left free.
JP7	External Code Enable	This jumper enables access to the external code if the single-chip mode is emulated and part of the code is mapped into the target code memory. The external code is enabled if the jumper top is out (default jumper configuration).
Reset Button		When pressed, this switch resets the pod (including the MCU). You can use this button to recover from a map error status.
JP10	SY0, SY1	This header contains two pins: pin 1 (SY0), and pin 2 (SY1). These two signals are carried to the emulator and trace boards, and can be used in the trace and breakpoint logic
JP9	EMUL, FLF, ANB	This header contains three pins: pin 1 (EMUL), pin 2 (FLF), and pin 3 (ANB). Pin 1 signals emulation or monitor mode. It is low when emulation is active (when the user program is running). Pins 2 and 3 are not used currently, but can be used in the future.

POD–51EH–C515C–10

Jumper Identification and Description (continued)

Jumper Designation	Function	Description
J2	P3.6, P3.7	This header carries the recreated signals; P3.6/WR (pin J2.1) and P3.7/RD (pin J2.2). These pins observe the signals or route them to the trace board through the headers XL0..XL7, XH0..XH7.
JP1	PE, EA, ALE, PSEN	This header carries the signal listed previously and is used for testing, or can be connected to the trace board through the headers XL0..XL7, XH0..XH7.
JP21	RESET, P7.0, CPUR, HWPD	This header carries the signal listed previously and is used for testing, or can be connected to the trace board through the headers XL0..XL7, XH0..XH7.
P1, P3, P4, P5, P6	Port Headers	These headers carry signals for the corresponding ports. The port pin numbers are marked on the silk-screen. For example, pin 5 of header P4 connects to pin P4.5 of the MCU. **Note**: Earlier revisions of the pod might have header JP15 installed on the board. This header is for factory testing only and is not for any user functions.

Differences Between the Emulator and the C515C MCU

EA Pin

The EA pin in the C515C MCU is an input, floating pin used to select between single-chip mode (the program in internal OTP–ROM) and expanded mode (the program in external memory). This pin does not have a pull-up or a pull-down resistor.

A 47K pull-up resistor on the pod is connected to this pin to select a default state of single-chip mode when this pin is not forced by the target.

This difference of the existence of the pull-up resistor must be considered: if the target pin is tied to a pull-down resistor, it should be a sufficiently low pull-down (3.3K or lower) in order to overcome the 47K pull-up resistor on the pod.

In the C515C MCU during level 0 protection, the EA pin is not sampled during Reset and can be switched dynamically during run-time without going through Reset. This mode is not supported by the pod. The EA pin from the target is always sampled during exit from Reset (Reset Signal Fall time), and the mode of operation is then set to either single-chip or expanded mode until the next Reset.

Port0 and Port2 Pins

These 16 port pins are emulated by the pod and are used both as the address/data bus and as port pins. These pins have slightly different AC and DC characteristics from the Port0 and Port2 pins of the C515C MCU. They sink up to 12mA and source up to 4mA, while maintaining valid TTL output logic levels. Port2 pins also have pull-up resistors of 22K, which might be different from the C515C MCU's internal pull-ups.

POD–51EH–C517A–24

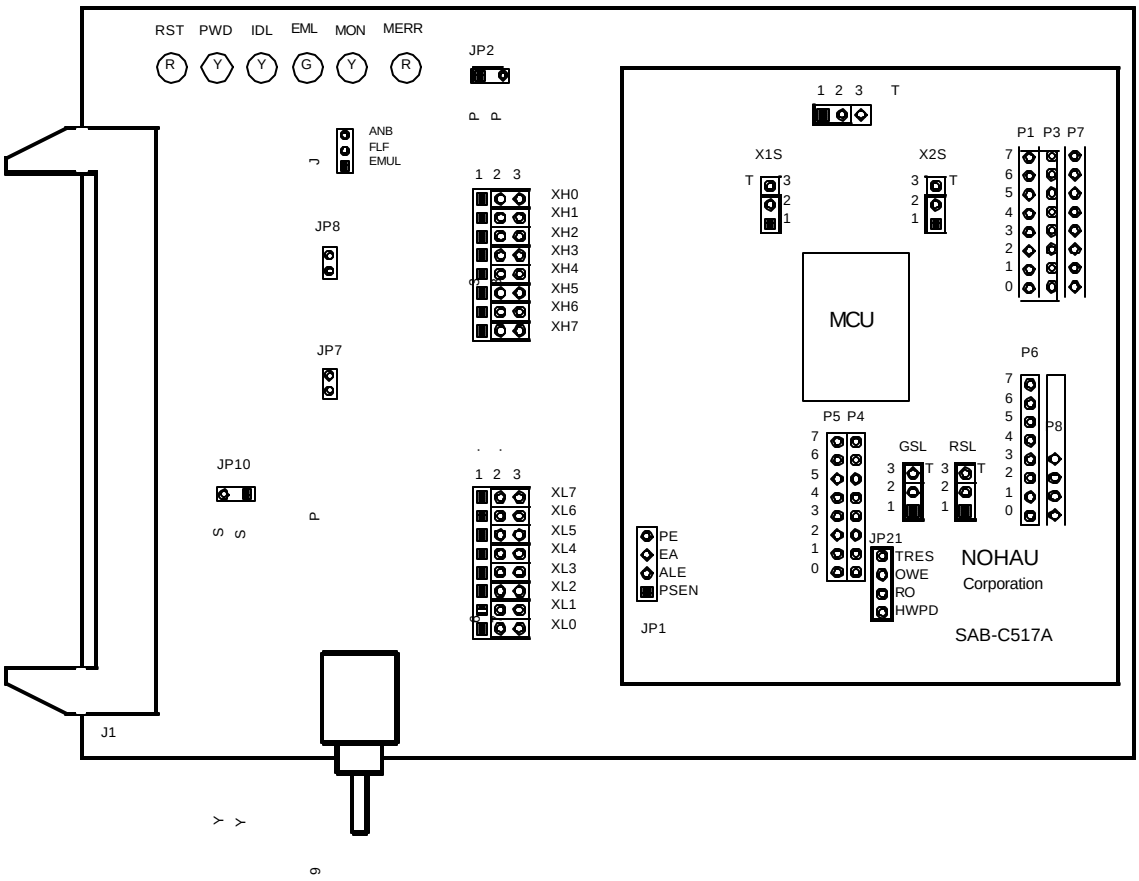


Figure 86. POD–51EH–C517A–24

Operating Instructions

The POD–51EH–C517A–24 supports the Siemens C517A MCU which is based on the C500 core.

To work with this pod, you need to select **POD–EH–C517A** from the **Select Processor** drop-down list box (**Step 3** of the **Communications** tab in the Emulator Configuration window).

> **Note**
>
> POD–51EH–C517A–24 replaces POD–C517A–18, an older version of the pod for the Siemens SAB–C517A MCU. If you have the older version of the pod, you need to pay attention to the position of the pod relative to your target. Compared to the old pod, the new one needs to turn 90 degrees clockwise when you install it on the same target. (See Figure 87.)

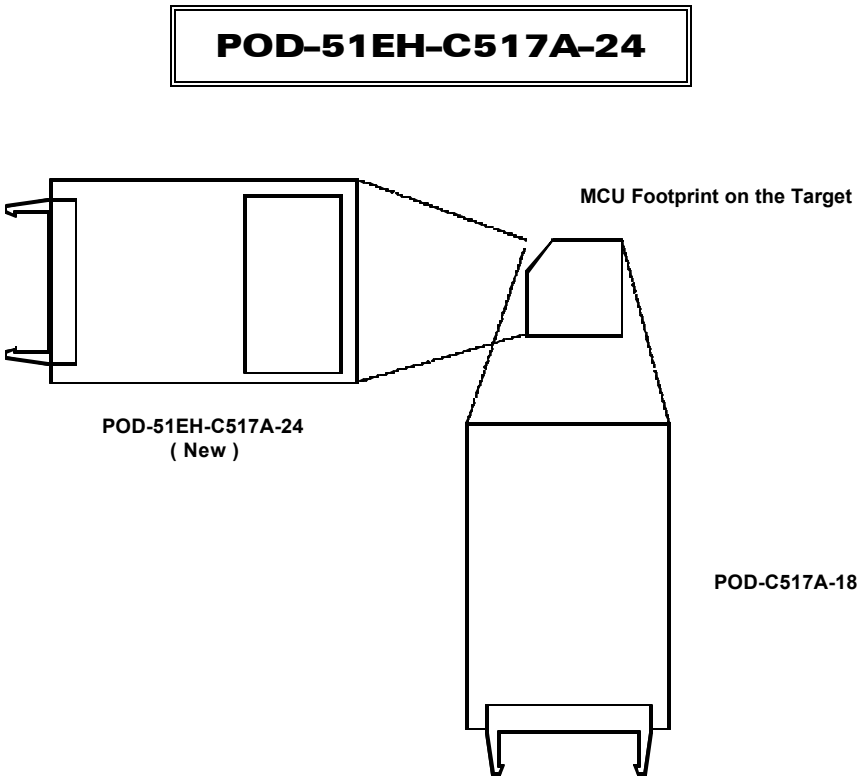


Figure 87. Pod Orientation Relative to the MCU Footprint on the Target

The EPROM on the emulator board must be version COM 1.4.

The pod assembly consists of two sub-assemblies: The larger universal Port Replacement Unit (PRU) and the smaller processor module daughterboard.

The PRU has three headers on the bottom that carry the signals to the target. The PRU also has a 50-pin header J1 that connects through the ribbon cable to the emulator board.

The target headers plug into an adapter that you select depending on the type of the MCU. For example, the SAB-C517A MCU uses the following adapters.

- Emulation Solutions ES/180–5690–10 (P–MQFP–100)
- Emulation Solutions ES/180–3975–10 (PLCC–84)

The adapters plug into the target system, replacing the C517A MCU IC. These adapters can be ordered from www.icetech.com or directly from Emulation Solutions.

The POD–51EH–C517A–24 runs up to a maximum frequency of 24 MHz.

The C517A MCU has on-chip XRAM memory located at addresses F800H – FFFFH in the data space. If your application uses the on-chip XRAM, you must map the last memory range of the data memory (addresses F000H – FFFFH) to the target.

POD–51EH–C517A–24

The EA pin during reset signal rise (also sampled on internal watchdog timer reset) selects the active mode. It is recommended that you connect this pin to either constant low or constant high. This pin defaults to high (selecting single-chip mode), that is implemented by a 22K pull-up resistor on the pin.

POD–51EH–C517A–24 has two operating modes.

- Single-chip
- External

The active mode is selected by the EA pin during reset signal rise (also sampled on internal reset).

When using external mode, the code memory can be mapped either to the target or the emulator (with boundaries of 4K). In external mode, addresses are seen on the Port2 and Port0 adapter pins for every bus cycle regardless of where it is mapped. The data, however, is read or written to the Port0 pins only for addresses that are mapped to the target.

When using single-chip mode, you need to map all code memory to the emulator board. In this mode, Port2 and Port0 can be used as I/O ports.

POD–51EH–C517A–24 also supports tracing of internal XRAM read and write cycles using MOVX instructions. This is supported in both single-chip or external modes.

It is important to note that:

- The C517A MCU has a default watchdog timer running at start-up. We recommend that you disable this watchdog timer by connecting pin PE/WDT to GND. Otherwise, an unexpected reset can occur while debugging is in process.

- The C517A MCU cannot wake up from a software power-down made by a low signal at the P3.2/INT0 pin.

POD–51EH–C517A–24

LED Indicators

Function	Description
Reset	A red LED that indicates when the MCU reset is active. There are six possible sources of the MCU reset. • Reset from the target starts whenever a low level is detected on the reset pin of the adapter. • Reset from the S1 button starts by pressing S1 on the pod. • Click the Reset button in the Seehau menu to reset the emulator. • Reset from the internal watchdog timer starts if the internal watchdog is enabled, and it is not fed by the user software for a long period. (**Note**: It is important to feed the watchdog periodically, otherwise, problems during the debug process might occur.) • During hardware power-down initiated by a low level detected on pin 69 (HWPD) of the target adapter/ • When a map error occurs. During all of these reset sequences, the reset LED is turned on for a short or a long period of time depending on the sequence.
Map Error	A red LED that indicates a mapping error occurring in one of the following situations: • The internal C517A XRAM is enabled by bit XMAP0 in the SYSCON register. • The data memory address range F000H – FFFFH is mapped to the emulator. • There is a memory access (MOVX) to the internal XRAM. This state is illegal. When the C517A internal XRAM is enabled and accesses are made to it, the data memory address range F000H – FFFFH needs to be mapped to the target. Single-chip mode is selected by the EA pin from the target (during the last reset), a code memory address range is mapped to the target, and there is code access to one of the addresses in this range. This state is illegal. In single-chip mode all of the code memory needs to be mapped to the emulator and not to the target, because the 64K code memory is internal to the C517A MCU. If one of these illegal states occurs, the Map Error LED turns on simultaneously with the C517A MCU entering the reset state, signaling the illegal state. This state is maintained until you push the reset button. Leaving the map error state is done only by pressing the reset button S1 on the pod. This prevents repetitive occurrences of the illegal state until you fix the problem.
Emulation	A green LED that indicates when the user program is running, and turns off when the user program is stopped (emulation break). The user code can be fetched from RAM on the emulator board or from code memory on the target (usually PROM).
Monitor	A yellow LED that indicates when the monitor mode is active. In this mode, the MCU executes internal emulator code that communicates with the host PC. The emulator enters this mode upon power-up or after emulation break.

POD–51EH–C517A–24

Jumper Identification and Description

Jumper Designation	Function	Description
Ground SeLect (GSL)	VAGND Grounding Select	This jumper selects how to ground the VAGND pin of the MCU. • When the jumper top is across pins 1 and 2, VAGND is connected to emulator ground. • When the jumper top is across pins 2 and 3, VAGND is grounded through pin 4 of the target adapter. Pin 3 of the jumper is marked by the letter T to show connection to the target. The default position of the jumper top is across pins 1 and 2 (emulator ground selected).
Reference SeLect (RSL)	VAREF Source Select	This jumper selects how to power the VAREF pin of the MCU. • When the jumper top is across pins 1 and 2, VAREF is connected to the emulator. • When the jumper top is across pins 2 and 3, VAREF is connected to the target through pin 3 of the target adapter. Pin 3 of the jumper is marked by the letter T to show connection to the target. The default position of the jumper top is across pins 1 and 2 (emulator selected as the voltage source).
Vcc SeLect (VSL)	MCU Vcc Source Select	This jumper selects how to power the Vcc pins of the MCU. • When the jumper top is on pins 1 and 2, the MCU Vcc pins are connected to the Vcc of the emulator. • When the jumper top is on pins 2 and 3, the MCU Vcc pins are connected to the Vcc pins of the adapter. This allows the MCU Vcc pins to be powered by the target. The default position of the jumper top is across pins 1 and 2 (emulator selected as Vcc source). **Note**: Taking power from the target system is not recommended. Both Vcc pins of the MCU are shorted together.
X1S, X2S	XTAL1 and XTAL2 Source Select	These two jumpers select the clock source for the MCU. • With the jumper tops on pins 1 and 2, the MCU XTAL1 and XTAL2 pins are connected to a 24-MHz crystal on the pod. • When these two jumpers have jumper tops on pins 2 and 3, the MCU XTAL1 and XTAL2 pins are connected to pins 37 and 36 of the adapter. This allows the target to supply the clock to the MCU XTAL1 and XTAL2 pins. The default position of the jumper top is across pins 1 and 2 (the pod crystal). **Note**: Both of the jumper tops need to be placed in the same position with either both selecting the pod crystal source, or both selecting the target clock source.

POD–51EH–C517A–24

Jumper Identification and Description (continued)

Jumper Designation	Function	Description
JP8	Monitor Stop/Break Select	This jumper determines whether all or only part of the internal MCU peripherals stop in monitor mode after an emulation break occurs. • When the jumper top is out, the stop mode is selected. In this case, all of the internal MCU peripherals are stopped in monitor mode. This includes the following MCU peripherals: interrupts, watchdog timer, oscillator watchdog, timer0, timer1, timer2, timer2 capture, CCU compare timer, USART0, and USART1. • When the jumper top is on, the break mode is selected. In this case, only some of the internal peripherals are stopped. This includes the following MCU peripherals: interrupts, watchdog timer, oscillator timer, and timer2. Timer0, timer1, timer2 capture, CCU compare timer, USART0, and USART1 are not stopped in this mode. In stop and break mode, all of the internal peripherals start working from the same point they stopped before the break as soon as the user program starts running again when you press the Go or Step buttons. When changing this jumper, the new stop/break mode takes effect after the next reset sequence. The default setup of the jumper is stop mode selection (jumper top out). **Note**: It is recommended to work in stop mode only. The break mode needs to be selected only if your application has limitations that do not allow the timers to be stopped in monitor mode.
XL0..XL7	Trace Port Select Jumpers	These eight 3-pin jumpers select the signals that are routed to the trace board and are shown in the Trace window under the header P1. For any jumper in this group the jumper top position pins 2–3 means that the corresponding pin of the recreated Port0 is traced. Pin 1 of each jumper is left free and can be connected to any signal on the target or port headers on the pod. For example setting XL4 in position 2–3 means that pin P0.4 is traced. If pin XL4.1, for example, is connected to a signal P4.3 on the target, you can choose between tracing signal P0.4 or P4.3 by moving the jumper between positions 2–3 and 1–2.
XH0..XH7	Trace Port Select Jumpers	This group of jumpers is similar to XL0..XL7, except pin 3 of each jumper is connected to the recreated Port2 and pin 1 is left free.
JP7	External Code Enable	This jumper enables access to the external code if the single-chip mode is emulated and part of the code is mapped into the target code memory. The external code is enabled if the jumper top is out (default jumper configuration).
Reset Button		When pressed, this switch resets the pod (including the MCU). You can use this button to recover from a map error status.
JP10	SY0, SY1	This header contains two pins: pin 1 (SY0), and pin 2 (SY1). These two signals are carried to the emulator and trace boards, and can be used in the trace and breakpoint logic.

POD–51EH–C517A–24

Jumper Identification and Description (continued)

Jumper Designation	Function	Description
JP9	EMUL, FLF, ANB	This header contains three pins: pin 1 (EMUL), pin 2 (FLF), and pin 3 (ANB). Pin 1 signals emulation or monitor mode. It is low when emulation is active (when the user program is running). Pins 2 and 3 are not used currently, but can be used in the future.
J2	P3.6, P3.7	This header carries the recreated signals; P3.6/WR (pin J2.1) and P3.7/RD (pin J2.2). These pins observe the signals or route them to the trace board through the headers XL0..XL7, XH0..XH7.
JP1	PE, EA, ALE, PSEN	This header carries the signal listed previously and is used for testing, or can be connected to the trace board through the headers XL0..XL7, XH0..XH7.
JP21	RESET, P7.0, CPUR, HWPD	This header carries the signal listed previously and is used for testing, or can be connected to the trace board through the headers XL0..XL7, XH0..XH7.
P1, P3, P4, P5, P6, P7, P8	Port Headers	These headers carry signals for the corresponding ports. The port pin numbers are marked on the silk-screen. For example, pin 5 of header P4 connects to pin P4.5 of the MCU.

Differences Between the Emulator and the C517A MCU

EA Pin

The EA pin in the C517A MCU is an input, floating pin used to select between single-chip mode (the program in internal OTP–ROM) and expanded mode (the program in external memory). This pin does not have a pull-up or a pull-down resistor.

A 47K pull-up resistor on the pod is connected to this pin to select a default state of single-chip mode when this pin is not forced by the target.

This difference of the existence of the pull-up resistor must be considered: if the target pin is tied to a pull-down resistor, it should be a sufficiently low pull-down (3.3K or lower) in order to overcome the 47K pull-up resistor on the pod.

In the C517A MCU during level 0 protection, the EA pin is not sampled during Reset and can be switched dynamically during run-time without going through Reset. This mode is not supported by the pod. The EA pin from the target is always sampled during exit from Reset (Reset Signal Fall time), and the mode of operation is then set to either single-chip or expanded mode until the next Reset.

Port0 and Port2 Pins

These 16 port pins are emulated by the pod and are used both as the address/data bus and as port pins. These pins have slightly different AC and DC characteristics from the Port0 and Port2 pins of the C517A MCU. They sink up to 12mA and source up to 4mA, while maintaining valid TTL output logic levels. Port2 pins also have pull-up resistors of 22K, which might be different from the C517A MCU's internal pull-ups.

POD–51EH–SAB–C515

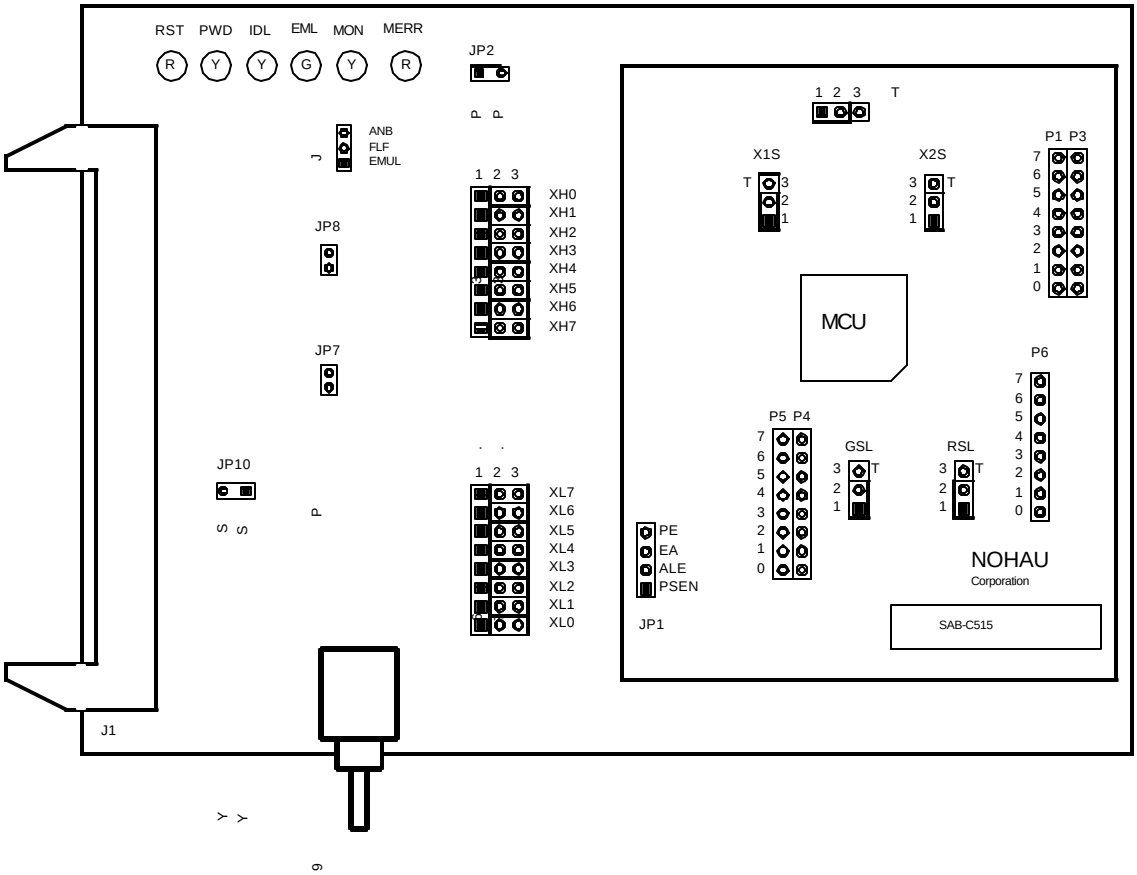


Figure 88. POD–51EH–SAB–C515

Operating Instructions

The POD–51EH–SAB–C515 supports the Siemens C515 MCU which is based on the C500 core.

To work with this pod, you need to select **POD–EH–C515** from the **Select Processor** drop-down list box (**Step 3** of the **Communications** tab in the Emulator Configuration window).

The EPROM on the emulator board must be version COM 1.4.

The pod assembly consists of two sub-assemblies: The larger universal Port Replacement Unit (PRU) and the smaller processor module daughterboard.

The PRU has three headers on the bottom that carry the signals to the target. For the packages with the small pinout (44-pin), it also provides a 44-pin PGA socket on the bottom that you can connect to the target using an adapter. The PRU also has a 50-pin header J1 that connects through the ribbon cable to the emulator board.

POD–51EH–SAB–C515

The target headers plug into an adapter that you select depending on the type of the MCU. For example, the SAB–C515 MCU uses the following adapters.

- Emulation Solutions ES/180–5550–40 (socketable)
- Emulation Solutions ES/180–5550–45 (non-socketable)

The adapters plug into the target system, replacing the C515 MCU IC. These adapters can be ordered from www.icetech.com or directly from Emulation Solutions.

POD–51EH–SAB–C515 runs to a maximum frequency of 24 MHz, and has two operating modes.

- Single-Chip
- External

The active mode is selected by the EA pin (pin 49 of the target adapter) during Reset signal rise (also sampled on internal watchdog timer reset). It is recommended that you connect this pin to either Constant Low or Constant High. This pin defaults to High (selecting single-chip mode), which is implemented by a 22K pull-up resistor on the pin.

When using the external mode, the code memory can be mapped either to the target or to the emulator (with boundaries of 4K). In external mode, addresses are seen on the Port2 and Port0 pins for every bus cycle, regardless of where it is mapped to. The data, however, reads or writes to the Port0 pins only for addresses that are mapped to the target.

When using single-chip mode, all code memory must be mapped to the emulator board. In this mode, Port2 and Port0 can be used as I/O ports.

POD-51EH-SAB-C515

LED Indicators

Function	Description
Reset	A red LED that indicates when the MCU reset is active. There are five possible sources of the MCU reset. • Reset from the target starts whenever a low level is detected on the reset pin of the adapter. • Reset from the S1 button starts by pressing S1 on the pod. • Click the Reset button in the Seehau menu to reset the emulator. • Reset from the internal watchdog timer starts if the internal watchdog is enabled, and it is not fed by the user software for a long period. (**Note**: It is important to feed the watchdog periodically, otherwise, problems during the debug process might occur.) • When a map error occurs. During all of these reset sequences, the reset LED is turned on for a short or a long period of time depending on the sequence.
Map Error	A red LED that indicates a mapping error occurring when the following conditions occur: • Single-chip mode is elected by the EA pin from the target. • All or part of the internal MCU code (address 0 to 8K) is mapped to the target. • Code is accessed to one of the addresses in this range. This state is illegal. In single-chip mode, the first 8K of the code memory needs to be mapped to the emulator and not to the target. If this illegal state occurs, the Map Error LED turns on simultaneously. The MCU enters the reset state, signaling to the user that this illegal state has occurred. This state is maintained until you push the reset button. Leaving the map error state is done only by pressing the reset button S1 on the pod. This prevents repetitive occurrences of the illegal state until you fix the problem.
Emulation	A green LED that indicates when the user program is running, and turns off when the user program is stopped (emulation break). The user code can be fetched from RAM on the emulator board or from code memory on the target (usually PROM).
Monitor	A yellow LED that indicates when the monitor mode is active. In this mode, the C515 MCU executes internal emulator code that communicates with the host PC. The emulator enters this mode upon power-up or after emulation break.

POD–51EH–SAB–C515

Jumper Identification and Description

Jumper Designation	Function	Description
Ground SeLect (GSL)	VAGND Grounding Select	This jumper selects how to ground the VAGND pin of the MCU. • When the jumper top is across pins 1 and 2, VAGND is connected to emulator ground. • When the jumper top is across pins 2 and 3, VAGND is grounded through pin 4 of the target adapter. Pin 3 of the jumper is marked by the letter T to show connection to the target. The default position of the jumper top is across pins 1 and 2 (emulator ground selected).
Reference SeLect (RSL)	VAREF Source Select	This jumper selects how to power the VAREF pin of the MCU. • When the jumper top is across pins 1 and 2, VAREF is connected to the emulator. • When the jumper top is across pins 2 and 3, VAREF is connected to the target through pin 3 of the target adapter. Pin 3 of the jumper is marked by the letter T to show connection to the target. The default position of the jumper top is across pins 1 and 2 (emulator selected as the voltage source).
Vcc SeLect (VSL)	MCU Vcc Source Select	This jumper selects how to power the Vcc pins of the MCU. • When the jumper top is on pins 1 and 2, the MCU Vcc pins are connected to the Vcc of the emulator. • When the jumper top is on pins 2 and 3, the MCU Vcc pins are connected to pins 33 and 69 of the adapter. This allows the MCU Vcc pins to be powered by the target. The default position of the jumper top is across pins 1 and 2 (emulator selected as Vcc source). **Note**: Taking power from the target system is not recommended. Both Vcc pins of the MCU are shorted together.
X1S, X2S	XTAL1 and XTAL2 Source Select	These two jumpers select the clock source for the MCU. • With the jumper tops on pins 1 and 2, the MCU XTAL1 and XTAL2 pins are connected to a 24-MHz crystal on the pod. • When these two jumpers have jumper tops on pins 2 and 3, the MCU XTAL1 and XTAL2 pins are connected to pins 37 and 36 of the adapter. This allows the target to supply the clock to the MCU XTAL1 and XTAL2 pins. The default position of the jumper top is across pins 1 and 2 (the pod crystal). **Note**: Both of the jumper tops need to be placed in the same position with either both selecting the pod crystal source, or both selecting the target clock source.

POD–51EH–SAB–C515

Jumper Identification and Description (continued)

Jumper Designation	Function	Description
JP8	Monitor Stop/Break Select	This jumper determines whether all or only part of the internal MCU peripherals stop in monitor mode after an emulation break occurs. • When the jumper top is out, the stop mode is selected. In this case, all of the internal MCU peripherals are stopped in monitor mode. This includes the following MCU peripherals: interrupts, watchdog timer, timer0, timer1, timer2, and timer2 capture. • When the jumper top is on, the break mode is selected. In this case, only some of the internal peripherals are stopped. This includes the following MCU peripherals: interrupts, watchdog timer, and timer2. Timer0, timer1, and timer2 capture are not stopped in this mode. In stop and break mode, all of the internal peripherals start working from the same point they stopped before the break as soon as the user program starts running again when you press the Go or Step buttons. When changing this jumper, the new stop/break mode takes effect after the next reset sequence. The default setup of the jumper is stop mode selection (jumper top out). **Note**: It is recommended to work in stop mode only. The break mode needs to be selected only if your application has limitations that do not allow the timers to be stopped in monitor mode.
XL0..XL7	Trace Port Select Jumpers	These eight 3-pin jumpers select the signals that are routed to the trace board and are shown in the Trace window under the header P1. For any jumper in this group the jumper top position pins 2–3 means that the corresponding pin of the recreated Port0 is traced. Pin 1 of each jumper is left free and can be connected to any signal on the target or port headers on the pod. For example setting XL4 in position 2–3 means that pin P0.4 is traced. If pin XL4.1, for example, is connected to a signal P4.3 on the target, you can choose between tracing signal P0.4 or P4.3 by moving the jumper between positions 2–3 and 1–2.
XH0..XH7	Trace Port Select Jumpers	This group of jumpers is similar to XL0..XL7, except pin 3 of each jumper is connected to the recreated Port2 and pin 1 is left free.
JP7	External Code Enable	This jumper enables access to the external code if the single-chip mode is emulated and part of the code is mapped into the target code memory. The external code is enabled if the jumper top is out (default jumper configuration).
Reset Button		When pressed, this switch resets the pod (including the MCU). You can use this button to recover from a map error status.
JP10	SY0, SY1	This header contains two pins: pin 1 (SY0), and pin 2 (SY1). These two signals are carried to the emulator and trace boards, and can be used in the trace and breakpoint logic

POD–51EH–SAB–C515

Jumper Identification and Description (continued)

Jumper Designation	Function	Description
JP9	EMUL, FLF, ANB	This header contains three pins: pin 1 (EMUL), pin 2 (FLF), and pin 3 (ANB). Pin 1 signals emulation or monitor mode. It is low when emulation is active (when the user program is running). Pins 2 and 3 are not used currently, but can be used in the future.
J2	P3.6, P3.7	This header carries the recreated signals; P3.6/WR (pin J2.1) and P3.7/RD (pin J2.2). These pins observe the signals or route them to the trace board through the headers XL0..XL7, XH0..XH7.
JP1	PE, EA, ALE, PSEN	This header carries the signal listed previously and is used for testing, or can be connected to the trace board through the headers XL0..XL7, XH0..XH7.
P1, P3, P4, P5, P6,	Port Headers	These headers carry signals for the corresponding ports. The port pin numbers are marked on the silk-screen. For example, pin 5 of header P4 connects to pin P4.5 of the MCU.

Differences Between the Emulator and the C515 MCU

EA Pin

The EA pin in the C515 MCU is an input, floating pin used to select between single-chip mode (the program in internal OTP–ROM) and expanded mode (the program in external memory). This pin does not have a pull-up or a pull-down resistor.

A 47K pull-up resistor on the pod is connected to this pin to select a default state of single-chip mode when this pin is not forced by the target.

This difference of the existence of the pull-up resistor must be considered: if the target pin is tied to a pull-down resistor, it should be a sufficiently low pull-down (3.3K or lower) in order to overcome the 47K pull-up resistor on the pod.

In the C515 MCU during level 0 protection, the EA pin is not sampled during Reset and can be switched dynamically during run-time without going through Reset. This mode is not supported by the pod. The EA pin from the target is always sampled during exit from Reset (Reset Signal Fall time), and the mode of operation is then set to either single-chip or expanded mode until the next Reset.

Port0 and Port2 Pins

These 16 port pins are emulated by the pod and are used both as the address/data bus and as port pins. These pins have slightly different AC and DC characteristics from the Port0 and Port2 pins of the C515 MCU. They sink up to 12mA and source up to 4mA, while maintaining valid TTL output logic levels. Port2 pins also have pull-up resistors of 22K, which might be different from the C515 MCU's internal pull-ups.

POD–51EH–TSC–51RX2–16

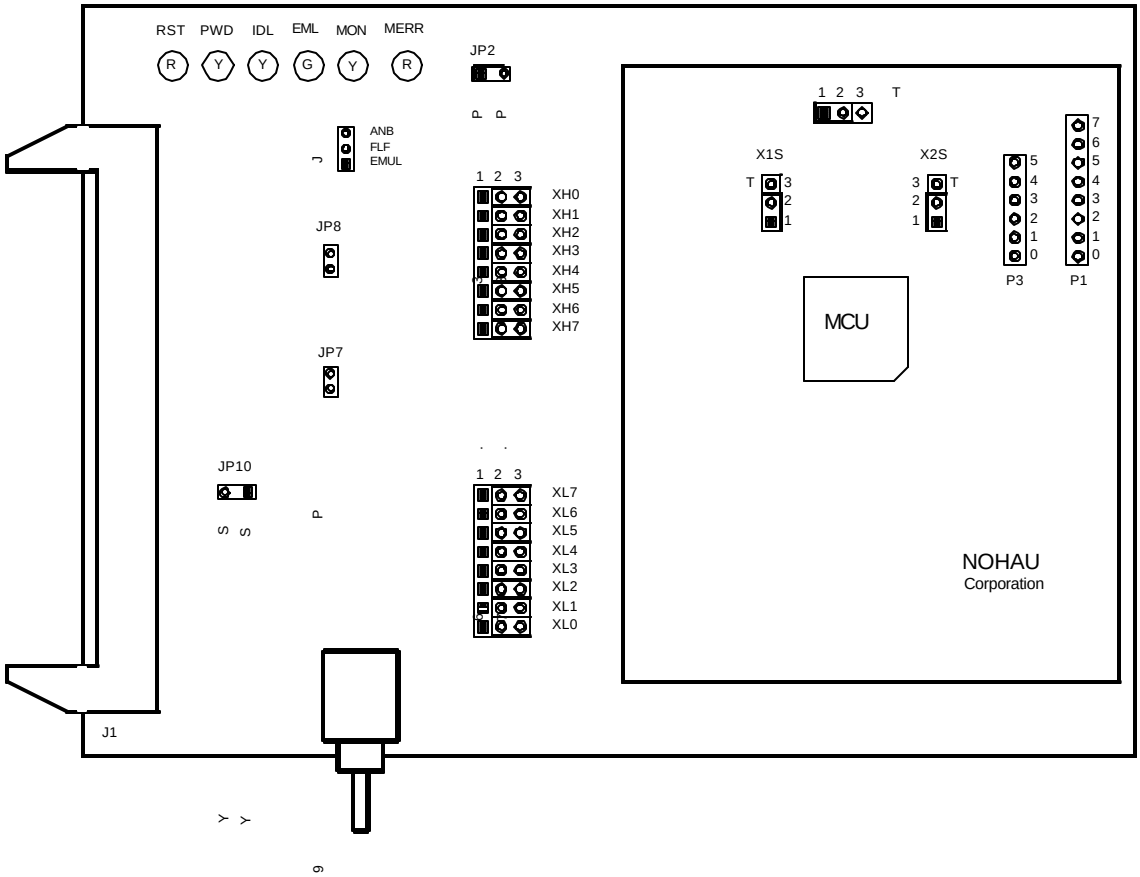


Figure 89. POD–51EH–TSC–51RX2–16

Operating Instructions

The POD–51EH–TSC–51RB2–16 supports the following Temic MCUs:
TS80C51RA2, TS83C51RB2, TS87C51RB2

The POD–51EH–TSC–51RC2–16 supports the following Temic MCUs:
TS83C51RC2, TS87C51RC2

The POD–51EH–TSC–51RD2–16 supports the following Temic MCUs:
TS80C51RD2, TS83C51RD2, TS87C51RD2

Note

The MCU used on the pod is a special emulation version. Contact support@icetech.com for a replacement MCU.

POD–51EH–TSC–51RX2–16

To work with this pod, you need to select **POD–EH–TSC51RD2** from the **Select Processor** drop-down list box (**Step 3** of the **Communications** tab in the Emulator Configuration window).

The EPROM on the emulator board must be version COM 1.4.

The pod assembly consists of two sub-assemblies: The larger universal Port Replacement Unit (PRU) and the smaller processor module daughterboard.

The PRU provides a 44-pin PGA socket on the bottom that you can connect to the target using an adapter. The PRU also has a 50-pin header J1 that connects through the ribbon cable to the emulator board.

The adapters plug into the target system, replacing the TS8XC51RX2 MCU IC. The target adapter part numbers are
EDI/44PG/QFS31–SD
PGA44–PLCC44 (PLCC)

The POD–51EH–TSC–51RX2–16 runs up to a maximum frequency of 16 MHz with the X2 option or 32 MHz without the X2 option.

The TS8XC51RX2 MCU has on-chip XRAM memory located at addresses 0000H – 02FFH(RD2) or 0000H-00FFH(RA2, RB2 and RC2) in the data (XRAM) space. If your application uses the on-chip XRAM, you must map the first memory range of the data memory (addresses 0000H – 0FFFH) to the target.

Operating Modes

POD–51EH–TSC–51RX2–16 has two operating modes:

- Single-chip
- External

The operating mode is selected by the EA pin during reset signal rise (also sampled on internal watch-dog timer reset). It is recommended that you connect this pin to either constant low or constant high. This pin defaults to high (selecting single-chip mode), which is implemented by a 22K pull-up resistor on the pin.

When using external mode, the code memory can be mapped either to the target or the emulator (with boundaries of 4K). In external mode, addresses are seen on the Port2 and Port0 adapter pins for every bus cycle regardless of where it is mapped.

When using single-chip mode, you need to map all code memory to the emulator board. In this mode, Port2 and Port0 can be used as I/O ports.

POD–51EH–TSC–51RX2–16 also supports tracing of internal XRAM read and write cycles using MOVX instructions. This is supported in both single-chip and external modes.

POD-51EH-TSC-51RX2-16

LED Indicators

Function	Description
Reset	A red LED that indicates when the MCU reset is active. There are five possible sources of the MCU reset. • Reset from the target starts whenever a low level is detected on the reset pin of the adapter. • Reset from the S1 button starts by pressing S1 on the pod. • Click the Reset button in the Seehau menu to reset the emulator. • Reset from the internal watchdog timer starts if the internal watchdog is enabled, and it is not fed by the user software for a long period. (**Note**: It is important to feed the watchdog periodically, otherwise, problems during the debug process might occur.) • When a map error occurs. During all of these reset sequences, the reset LED is turned on for a short or a long period of time depending on the sequence.
Map Error	A red LED that indicates a mapping error occurring in one of the following situations. • The internal TS8XC51RX2 XRAM is enabled, the data memory address range 0000H – 02FFH is mapped to the emulator, and there is a memory access (MOVX) to the internal XRAM. This state is illegal. When the TS8XC51RX2 internal XRAM is enabled and accesses are made to it, the data memory address range 0000H – 02FFH needs to be mapped to the target. • Single-chip mode is selected by the EA pin from the target, a code memory address range is mapped to the target, and there is code access to one of the addresses in this range. This state is illegal. In single-chip mode all of the code memory needs to be mapped to the emulator and not to the target. If one of these illegal states occurs, the Map Error (ERR) LED turns on simultaneously with the TS8XC51RX2 MCU entering the reset state, signaling the illegal state. This state is maintained until you push the reset button. Leaving the map error state is done only by pressing the reset button S1 on the pod. This prevents repetitive occurrences of the illegal state until you fix the problem.
Emulation	A green LED that indicates when the user program is running, and turns off when the user program is stopped (emulation break). The user code can be fetched from RAM on the emulator board or from code memory on the target (usually PROM).
Monitor	A yellow LED that indicates when the monitor mode is active. In this mode, the MCU executes internal emulator code that communicates with the host PC. The emulator enters this mode upon power-up or after emulation break.

POD–51EH–TSC–51RX2–16

Jumper Identification and Description

Jumper Designation	Function	Description
Vcc SeLect (VSL)	MCU Vcc Source Select	This jumper selects how to power the Vcc pin of the MCU. • When the jumper top is on pins 1 and 2, the MCU Vcc pin is connected to the Vcc of the emulator. • When the jumper top is on pins 2 and 3, the MCU Vcc pin is connected to the Vcc pin of the adapter. This allows the MCU Vcc pin to be powered by the target. The default position of the jumper top is across pins 1 and 2 (emulator selected as Vcc source). **Note**: Taking power from the target system is not recommended.
X1S, X2S	XTAL1 and XTAL2 Source Select	These two jumpers select the clock source for the MCU. • With the jumper tops on pins 1 and 2, the MCU XTAL1 and XTAL2 pins are connected to a 16-MHz crystal on the pod. • When these two jumpers have jumper tops on pins 2 and 3, the MCU XTAL1 and XTAL2 pins are connected to pins 15 and 14 of the adapter. This allows the target to supply the clock to the MCU XTAL1 and XTAL2 pins. The default position of the jumper top is across pins 1 and 2 (the pod crystal). **Note**: Both of the jumper tops need to be placed in the same position with either both selecting the pod crystal source, or both selecting the target clock source.
XL0..XL7	Trace Port Select Jumpers	These eight 3-pin jumpers select the signals that are routed to the trace board and are shown in the Trace window under the header P1. For any jumper in this group the jumper top position pin 2–3 means that the corresponding pin of the recreated Port0 is traced. Pin 1 of each jumper is left free and can be connected to any signal on the target or port headers on the pod. For example setting XL4 in position 2–3 means that pin P0.4 is traced. If pin XL4.1, for example, is connected to a signal P4.3 on the target, you can choose between tracing signal P0.4 or P4.3 by moving the jumper between positions 2–3 and 1–2.
XH0..XH7	Trace Port Select Jumpers	This group of jumpers is similar to XL0..XL7, except pin 3 of each jumper is connected to the recreated Port2 and pin 1 is left free.
JP7	External Code Enable	This jumper enables access to the external code if the single-chip mode is emulated and part of the code is mapped into the target code memory
Reset Button		When pressed, this switch resets the pod (including the MCU). You can use this button to recover from a map error status.
JP10	SY0, SY1	This header contains two pins: pin 1 (SY0), and pin 2 (SY1). These two signals are carried to the emulator and trace boards, and can be used in the trace and breakpoint logic.

POD–51EH–TSC–51RX2–16

Jumper Identification and Description (continued)

Jumper Designation	Function	Description
JP9	EMUL, FLF, ANB	This header contains three pins: pin 1 (EMUL), pin 2 (FLF), and pin 3 (ANB). Pin 1 signals emulation or monitor mode. It is low when emulation is active (when the user program is running). Pin 2 carries the signal from State Machine bit 5 if you have the ATR or ETR series trace boards. Pin 3 carries the signal from State Machine bit 4 if you have the ATR or ETR series trace boards.
J2	P3.6, P3.7	This header carries the recreated signals; P3.6/WR (pin J2.1) and P3.7/RD (pin J2.2). These pins observe the signals or route them to the trace board through the headers XL0..XL7, XH0..XH7.
P1, P3, P4	Port Headers	These headers carry signals for the corresponding ports. The port pin numbers are marked on the silk-screen. For example, pin 5 of header P3 connects to pin P3.5 of the MCU. **Note**: Earlier revisions of the pod might have header JP15 installed on the board. This header is for factory testing only and is not for any user functions.

Differences Between the Emulator and the TS8XC51RX2 MCU

EA Pin

The EA pin in the TS8XC51RX2 MCU is an input, floating pin used to select between single-chip mode (the program in internal OTP–ROM) and expanded mode (the program in external memory). This pin does not have a pull-up or a pull-down resistor.

A 47K pull-up resistor on the pod is connected to this pin to select a default state of single-chip mode when this pin is not forced by the target.

This difference of the existence of the pull-up resistor must be considered: if the target pin is tied to a pull-down resistor, it should be a sufficiently low pull-down (3.3K or lower) in order to overcome the 47K pull-up resistor on the pod.

In the TS8XC51RX2 MCU during level 0 protection, the EA pin is not sampled during Reset and can be switched dynamically during run-time without going through Reset. This mode is not supported by the pod. The EA pin from the target is always sampled during exit from Reset (Reset Signal Fall time), and the mode of operation is then set to either single-chip or expanded mode until the next Reset.

Port0 and Port2 Pins

These 16 port pins are emulated by the pod and are used both as the address/data bus and as port pins. These pins have slightly different AC and DC characteristics from the Port0 and Port2 pins of the TS8XC51RX2 MCU. They sink up to 12mA and source up to 4mA, while maintaining valid TTL output logic levels. Port2 pins also have pull-up resistors of 22K, which might be different from the TS8XC51RX2 MCU's internal pull-ups.

POD–51EH–TSC–51U2–16

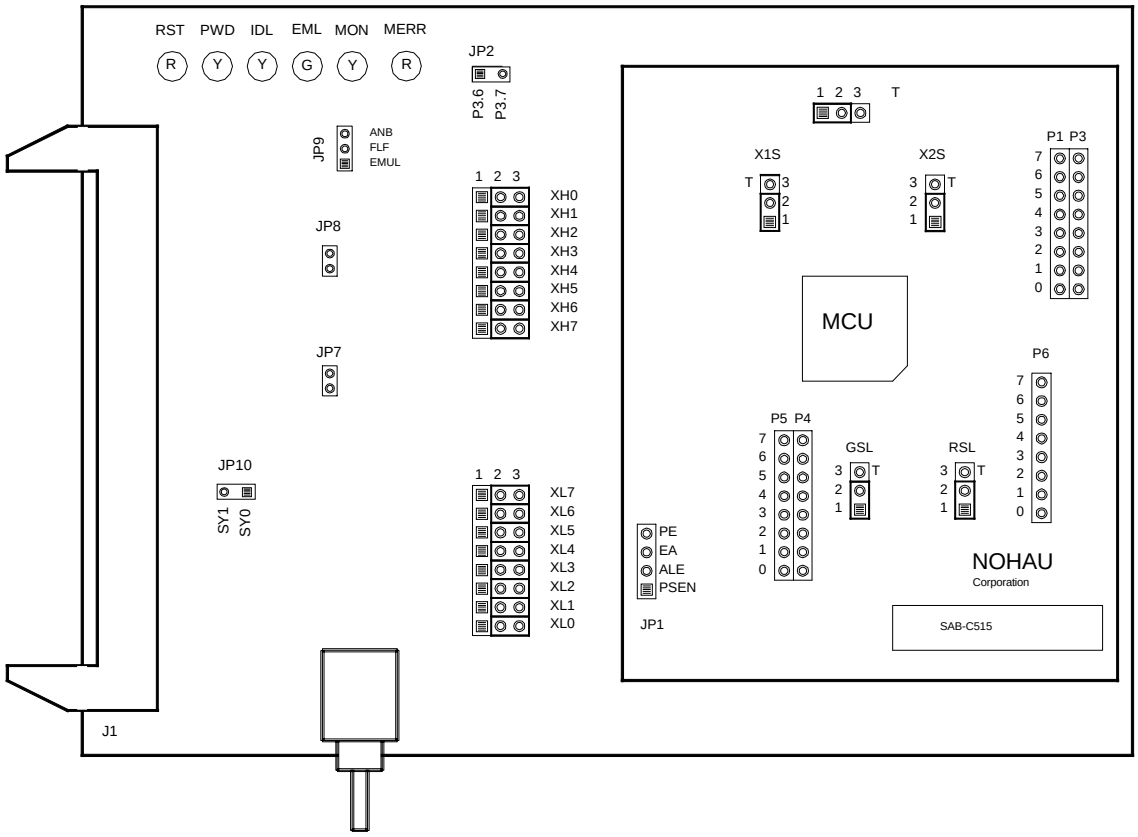


Figure 90. POD–51EH–TSC–51U2–16

Operating Instructions

> **Note**
>
> The Temic 89 series parts are not directly supported. The 87 series is typically used to emulate the 89 series. This results in the following limitations:
>
> - An 89 series part cannot be used in the pod.
> - Flash programming cannot be tested using the emulator.
> - 2K EEPROM is not supported; however, data tables can be copied into XDATA.
> - Only 1024 bytes of internal RAM are supported.

The POD–51EH–TSC–51U2–16 supports the following Temic MCUs:
TS80C51U2, TS83C51U2, TS87C51U2

To work with this pod, you need to select **POD–EH–TSC51U2** from the **Select Processor** drop-down list box (**Step 3** of the **Communications** tab in the Emulator Configuration window).

POD–51EH–TSC–51U2–16

The EPROM on the emulator board must be version COM 1.4.

Note

The MCU used on the pod is a special emulation version. Contact support@icetech.com for a replacement MCU.

The pod assembly consists of two sub-assemblies: The larger universal Port Replacement Unit (PRU) and the smaller processor module daughterboard.

The PRU provides a 44-pin PGA socket on the bottom that you can connect to the target using an adapter. The PRU also has a 50-pin header J1 that connects through the ribbon cable to the emulator board.

The adapters plug into the target system, replacing the TS8XC51U2 MCU IC. The target adapter part numbers are
EDI/44PG/QFS31–SD
PGA44–PLCC44 (PLCC)

The POD–51EH–TSC–51U2–16 runs up to a maximum frequency of 16 MHz with the X2 option or 32 MHz without the X2 option.

The active mode is selected by the EA pin during reset signal rise (also sampled on internal watchdog timer reset). It is recommended that you connect this pin to either constant low or constant high. This pin defaults to high (selecting single-chip mode), which is implemented by a 22K pull-up resistor on the pin.

Operating Modes

POD-51EH-TSC51U2-16 has two operating modes:

- Single-chip
- External

When using external mode, the code memory can be mapped either to the target or the emulator (with boundaries of 4K). In external mode, addresses are seen on the Port2 and Port0 adapter pins for every bus cycle regardless of where it is mapped. The data, however, is read or written to the Port0 pins only for addresses that are mapped to the target.

When using single-chip mode, you need to map all code memory to the emulator board. In this mode, Port2 and Port0 can be used as I/O ports.

POD–51EH–TSC–51U2–16

LED Indicators

Function	Description
Reset	A red LED that indicates when the MCU reset is active. There are five possible sources of the MCU reset. • Reset from the target starts whenever a low level is detected on the reset pin of the adapter. • Reset from the S1 button starts by pressing S1 on the pod. • Click the Reset button in the Seehau menu to reset the emulator. • Reset from the internal watchdog timer starts if the internal watchdog is enabled, and it is not fed by the user software for a long period. (**Note**: It is important to feed the watchdog periodically, otherwise, problems during the debug process might occur.) • When a map error occurs. During all of these reset sequences, the reset LED is turned on for a short or a long period of time depending on the sequence.
Map Error	A red LED that indicates a mapping error that happens when all of the following conditions occur. • Single-chip mode is selected by the EA pin from the target. • All or part of the 16K internal MCU code is mapped to the target. • Code is accessed to one of the addresses in this range. This state is illegal. In the single-chip mode, the first 16K of the code memory needs to be mapped to the emulator and not to the target. If this illegal state occurs, the Map Error (ERR) LED turns on simultaneously with the MCU entering the reset state, signaling the illegal state. This state is maintained until you push the reset button. Leaving the map error state is done only by pressing the reset button S1 on the pod. This prevents repetitive occurrences of the illegal state until you fix the problem.
Emulation	A green LED that indicates when the user program is running, and turns off when the user program is stopped (emulation break). The user code can be fetched from RAM on the emulator board or from code memory on the target (usually PROM).
Monitor	A yellow LED that indicates when the monitor mode is active. In this mode, the MCU executes internal emulator code that communicates with the host PC. The emulator enters this mode upon power-up or after emulation break.

POD–51EH–TSC–51U2–16

Jumper Identification and Description

Jumper Designation	Function	Description
Vcc SeLect (VSL)	MCU Vcc Source Select	This jumper selects how to power the Vcc pin of the MCU. • When the jumper top is on pins 1 and 2, the MCU Vcc pin is connected to the Vcc of the emulator. • When the jumper top is on pins 2 and 3, the MCU Vcc pin is connected to the Vcc pin of the adapter. This allows the MCU Vcc pin to be powered by the target. The default position of the jumper top is across pins 1 and 2 (emulator selected as Vcc source). **Note**: Taking power from the target system is not recommended.
X1S, X2S	XTAL1 and XTAL2 Source Select	These two jumpers select the clock source for the MCU. • With the jumper tops on pins 1 and 2, the MCU XTAL1 and XTAL2 pins are connected to a 16-MHz crystal on the pod. • When these two jumpers have jumper tops on pins 2 and 3, the MCU XTAL1 and XTAL2 pins are connected to pins 15 and 14 of the adapter. This allows the target to supply the clock to the MCU XTAL1 and XTAL2 pins. The default position of the jumper top is across pins 1 and 2 (the pod crystal). **Note**: Both of the jumper tops need to be placed in the same position with either both selecting the pod crystal source, or both selecting the target clock source.
XL0..XL7	Trace Port Select Jumpers	These eight 3-pin jumpers select the signals that are routed to the trace board and are shown in the Trace window under the header P1. For any jumper in this group the jumper top position pin 2–3 means that the corresponding pin of the recreated Port0 is traced. Pin 1 of each jumper is left free and can be connected to any signal on the target or port headers on the pod. For example setting XL4 in position 2–3 means that pin P0.4 is traced. If pin XL4.1, for example, is connected to a signal P4.3 on the target, you can choose between tracing signal P0.4 or P4.3 by moving the jumper between positions 2–3 and 1–2.
XH0..XH7	Trace Port Select Jumpers	This group of jumpers is similar to XL0..XL7, except pin 3 of each jumper is connected to the recreated Port2 and pin 1 is left free.
JP7	External Code Enable	This jumper enables access to the external code if the single-chip mode is emulated and part of the code is mapped into the target code memory
Reset Button		When pressed, this switch resets the pod (including the MCU). You can use this button to recover from a map error status.

POD–51EH–TSC–51U2–16

Jumper Identification and Description (continued)

Jumper Designation	Function	Description
JP10	SY0, SY1	This header contains two pins: pin 1 (SY0), and pin 2 (SY1). These two signals are carried to the emulator and trace boards, and can be used in the trace and breakpoint logic.
JP9	EMUL, FLF, ANB	This header contains three pins: pin 1 (EMUL), pin 2 (FLF), and pin 3 (ANB). Pin 1 signals emulation or monitor mode. It is low when emulation is active (when the user program is running). Pins 2 and 3 are not used currently, but can be used in the future.
J2	P3.6, P3.7	This header carries the recreated signals; P3.6/WR (pin J2.1) and P3.7/RD (pin J2.2). These pins observe the signals or route them to the trace board through the headers XL0..XL7, XH0..XH7.
P1, P3, P4	Port Headers	These headers carry signals for the corresponding ports. The port pin numbers are marked on the silk-screen. For example, pin 5 of header P3 connects to pin P3.5 of the MCU. **Note**: Earlier revisions of the pod might have header JP15 installed on the board. This header is for factory testing only and is not for any user functions.

Differences Between the Emulator and the TS8XC51U2 MCU

EA Pin

The EA pin in the TS8XC51U2 MCU is an input, floating pin used to select between single-chip mode (the program in internal OTP–ROM) and expanded mode (the program in external memory). This pin does not have a pull-up or a pull-down resistor.

A 47K pull-up resistor on the pod is connected to this pin to select a default state of single-chip mode when this pin is not forced by the target.

This difference of the existence of the pull-up resistor must be considered: if the target pin is tied to a pull-down resistor, it should be a sufficiently low pull-down (3.3K or lower) in order to overcome the 47K pull-up resistor on the pod.

In the TS8XC51U2 MCU during level 0 protection, the EA pin is not sampled during Reset and can be switched dynamically during run-time without going through Reset. This mode is not supported by the pod. The EA pin from the target is always sampled during exit from Reset (Reset Signal Fall time), and the mode of operation is then set to either single-chip or expanded mode until the next Reset.

Port0 and Port2 Pins

These 16 port pins are emulated by the pod and are used both as the address/data bus and as port pins. These pins have slightly different AC and DC characteristics from the Port0 and Port2 pins of the TS8XC51U2 MCU. They sink up to 12mA and source up to 4mA, while maintaining valid TTL output logic levels. Port2 pins also have pull-up resistors of 22K, which might be different from the TS8XC51U2 MCU's internal pull-ups.

POD–51EH–TSC–52/54/58X2–16

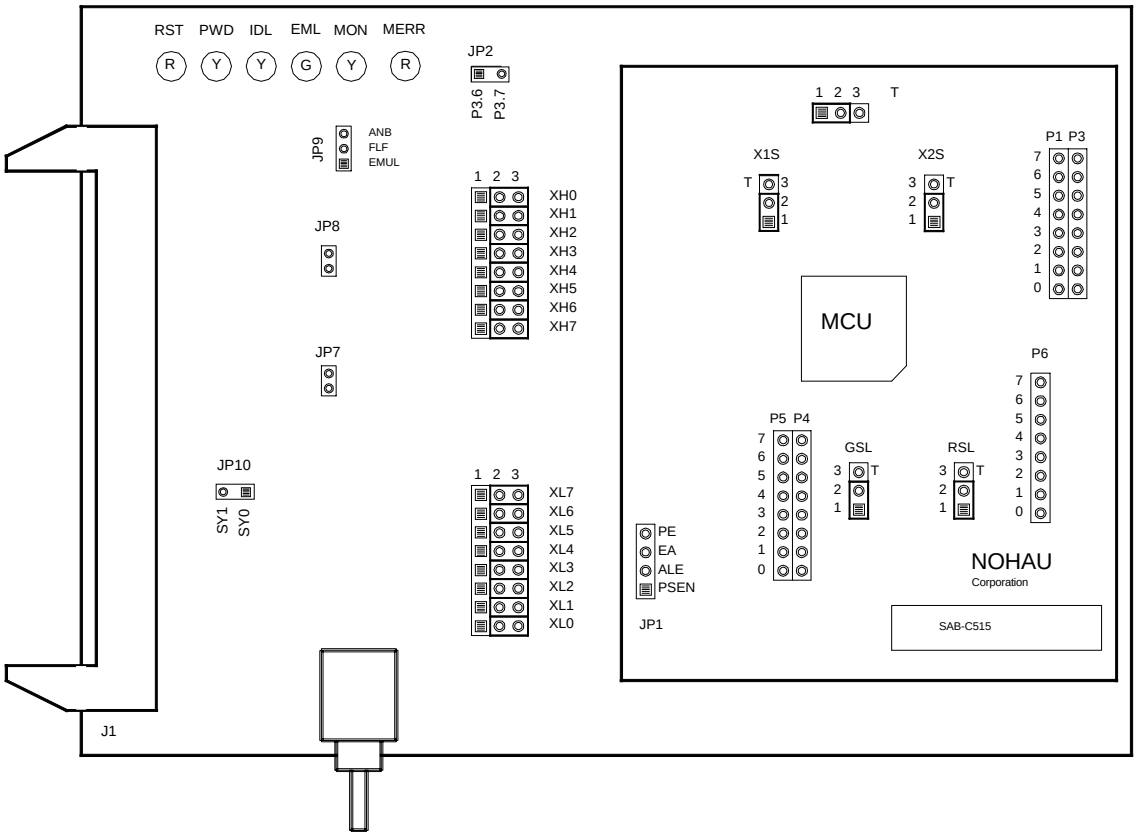


Figure 91. POD–51EH–TSC–52/54/58X2–16

Operating Instructions

The POD–51EH–TSC–52X2-16 supports the following Temic MCUs:
TS80C31X2, TS80C32X2, TS80C52X2, TS87C52X2

The POD–51EH–TSC–54X2-16 supports the following Temic MCUs:
TS80C31X2, TS80C32X2, TS80C54X2, TS87C54X2

The POD–51EH–TSC–58X2-16 supports the following Temic MCUs:
TS80C31X2, TS80C32X2, TS80C58X2, TS87C58X2

> **Note**
>
> The MCU used on the pod is a special emulation version. Contact support@icetech.com for a replacement MCU.

POD–51EH–TSC–52/54/58X2–16

To work with this pod, you need to select **POD–EH–TSC58X2** from the **Select Processor** drop-down list box (**Step 3** of the **Communications** tab in the Emulator Configuration window).

The EPROM on the emulator board must be version COM 1.4.

The pod assembly consists of two sub-assemblies: The larger universal Port Replacement Unit (PRU) and the smaller processor module daughterboard.

The PRU provides a 44-pin PGA socket on the bottom that you can connect to the target using an adapter. The PRU also has a 50-pin header J1 that connects through the ribbon cable to the emulator board.

The adapters plug into the target system, replacing the TS8XC31/32/52/54/58X2 MCU IC. The target adapter part numbers are
EDI/44PG/QFS31–SD
PGA44–PLCC44 (PLCC)

The POD–51EH–TSC52/54/58X2–16 runs up to a maximum frequency of 16 MHz with the X2 option or 32 MHz without the X2 option.

The active mode is selected by the EA pin during reset signal rise (also sampled on internal watchdog timer reset). It is recommended that you connect this pin to either constant low or constant high. This pin defaults to high (selecting single-chip mode), which is implemented by a 22K pull-up resistor on the pin.

Operating Modes

POD-51EH-TSC52/54/58X2-16 has two operating modes:

- Single-chip
- External

When using external mode, the code memory can be mapped either to the target or the emulator (with boundaries of 4K). In external mode, addresses are seen on the Port2 and Port0 adapter pins for every bus cycle regardless of where it is mapped. The data, however, is read or written to the Port0 pins only for addresses that are mapped to the target.

When using single-chip mode, you need to map all code memory to the emulator board. In this mode, Port2 and Port0 can be used as I/O ports.

POD–51EH–TSC–52/54/58X2–16

LED Indicators

Function	Description
Reset	A red LED that indicates when the MCU reset is active. There are five possible sources of the MCU reset. • Reset from the target starts whenever a low level is detected on the reset pin of the adapter. • Reset from the S1 button starts by pressing S1 on the pod. • Click the Reset button in the Seehau menu to reset the emulator. • Reset from the internal watchdog timer starts if the internal watchdog is enabled, and it is not fed by the user software for a long period. (**Note**: It is important to feed the watchdog periodically, otherwise, problems during the debug process might occur.) • When a map error occurs. During all of these reset sequences, the reset LED is turned on for a short or a long period of time depending on the sequence.
Map Error	This red LED signals a mapping error occurring in one of the following situations. • Single-chip mode is selected by the EA pin from the target. • All or part of the internal MCU code (for example, address 0 to 8K for 87C52X2) is mapped to the target. • Code is accessed to one of the addresses in this range. This state is illegal. In the single-chip mode, the first 8K (16K for 87C54X2, 32K for 87C58X2) of the code memory needs to be mapped to the emulator and not to the target. If this illegal state occurs, the Map Error LED turns on simultaneously with the MCU entering the reset state, signaling the illegal state. This state is maintained until you push the reset button. Leaving the map error state is done only by pressing the reset button S1 on the pod. This prevents repetitive occurrences of the illegal state until you fix the problem.
Emulation	A green LED that indicates when the user program is running, and turns off when the user program is stopped (emulation break). The user code can be fetched from RAM on the emulator board or from code memory on the target (usually PROM).
Monitor	A yellow LED that indicates when the monitor mode is active. In this mode, the MCU executes internal emulator code that communicates with the host PC. The emulator enters this mode upon power-up or after emulation break.

POD–51EH–TSC–52/54/58X2–16

Jumper Identification and Description

Jumper Designation	Function	Description
Vcc SeLect (VSL)	MCU Vcc Source Select	This jumper selects how to power the Vcc pin of the MCU. • When the jumper top is on pins 1 and 2, the MCU Vcc pin is connected to the Vcc of the emulator. • When the jumper top is on pins 2 and 3, the MCU Vcc pin is connected to the Vcc pin of the adapter. This allows the MCU Vcc pin to be powered by the target. The default position of the jumper top is across pins 1 and 2 (emulator selected as Vcc source). **Note**: Taking power from the target system is not recommended.
X1S, X2S	XTAL1 and XTAL2 Source Select	These two jumpers select the clock source for the MCU. • With the jumper tops on pins 1 and 2, the MCU XTAL1 and XTAL2 pins are connected to a 16-MHz crystal on the pod. • When these two jumpers have jumper tops on pins 2 and 3, the MCU XTAL1 and XTAL2 pins are connected to pins 15 and 14 of the adapter. This allows the target to supply the clock to the MCU XTAL1 and XTAL2 pins. The default position of the jumper top is across pins 1 and 2 (the pod crystal). **Note**: Both of the jumper tops need to be placed in the same position with either both selecting the pod crystal source, or both selecting the target clock source.
XL0..XL7	Trace Port Select Jumpers	These eight 3-pin jumpers select the signals that are routed to the trace board and are shown in the Trace window under the header P1. For any jumper in this group the jumper top position pin 2–3 means that the corresponding pin of the recreated Port0 is traced. Pin 1 of each jumper is left free and can be connected to any signal on the target or port headers on the pod. For example setting XL4 in position 2–3 means that pin P0.4 is traced. If pin XL4.1, for example, is connected to a signal P4.3 on the target, you can choose between tracing signal P0.4 or P4.3 by moving the jumper between positions 2–3 and 1–2.
XH0..XH7	Trace Port Select Jumpers	This group of jumpers is similar to XL0..XL7, except pin 3 of each jumper is connected to the recreated Port2 and pin 1 is left free.
JP7	External Code Enable	This jumper enables access to the external code if the single-chip mode is emulated and part of the code is mapped into the target code memory
Reset Button		When pressed, this switch resets the pod (including the MCU). You can use this button to recover from a map error status.

POD–51EH–TSC–52/54/58X2–16

Jumper Identification and Description (continued)

Jumper Designation	Function	Description
JP10	SY0, SY1	This header contains two pins: pin 1 (SY0), and pin 2 (SY1). These two signals are carried to the emulator and trace boards, and can be used in the trace and breakpoint logic.
JP9	EMUL, FLF, ANB	This header contains three pins: pin 1 (EMUL), pin 2 (FLF), and pin 3 (ANB). Pin 1 signals emulation or monitor mode. It is low when emulation is active (when the user program is running). Pins 2 and 3 are not used currently, but can be used in the future.
J2	P3.6, P3.7	This header carries the recreated signals; P3.6/WR (pin J2.1) and P3.7/RD (pin J2.2). These pins observe the signals or route them to the trace board through the headers XL0..XL7, XH0..XH7.
P1, P3, P4	Port Headers	These headers carry signals for the corresponding ports. The port pin numbers are marked on the silk-screen. For example, pin 5 of header P3 connects to pin P3.5 of the MCU. **Note**: Earlier revisions of the pod might have header JP15 installed on the board. This header is for factory testing only and is not for any user functions.

Differences Between the Emulator and the TS8XC52/54/58X2 MCU

EA Pin

The EA pin in the TS8XC52/54/58X2 MCU is an input, floating pin used to select between single-chip mode (the program in internal OTP–ROM) and expanded mode (the program in external memory). This pin does not have a pull-up or a pull-down resistor.

A 47K pull-up resistor on the pod is connected to this pin to select a default state of single-chip mode when this pin is not forced by the target.

This difference of the existence of the pull-up resistor must be considered: if the target pin is tied to a pull-down resistor, it should be a sufficiently low pull-down (3.3K or lower) in order to overcome the 47K pull-up resistor on the pod.

In the TS8XC52/54/58X2 MCU during level 0 protection, the EA pin is not sampled during Reset and can be switched dynamically during run-time without going through Reset. This mode is not supported by the pod. The EA pin from the target is always sampled during exit from Reset (Reset Signal Fall time), and the mode of operation is then set to either single-chip or expanded mode until the next Reset.

Port0 and Port2 Pins

These 16 port pins are emulated by the pod and are used both as the address/data bus and as port pins. These pins have slightly different AC and DC characteristics from the Port0 and Port2 pins of the TS8XC52/54/58X2 MCU. They sink up to 12mA and source up to 4mA, while maintaining valid TTL output logic levels. Port2 pins also have pull-up resistors of 22K, which might be different from the TS8XC52/54/58X2 MCU's internal pull-ups.

POD–51HB–C51FX

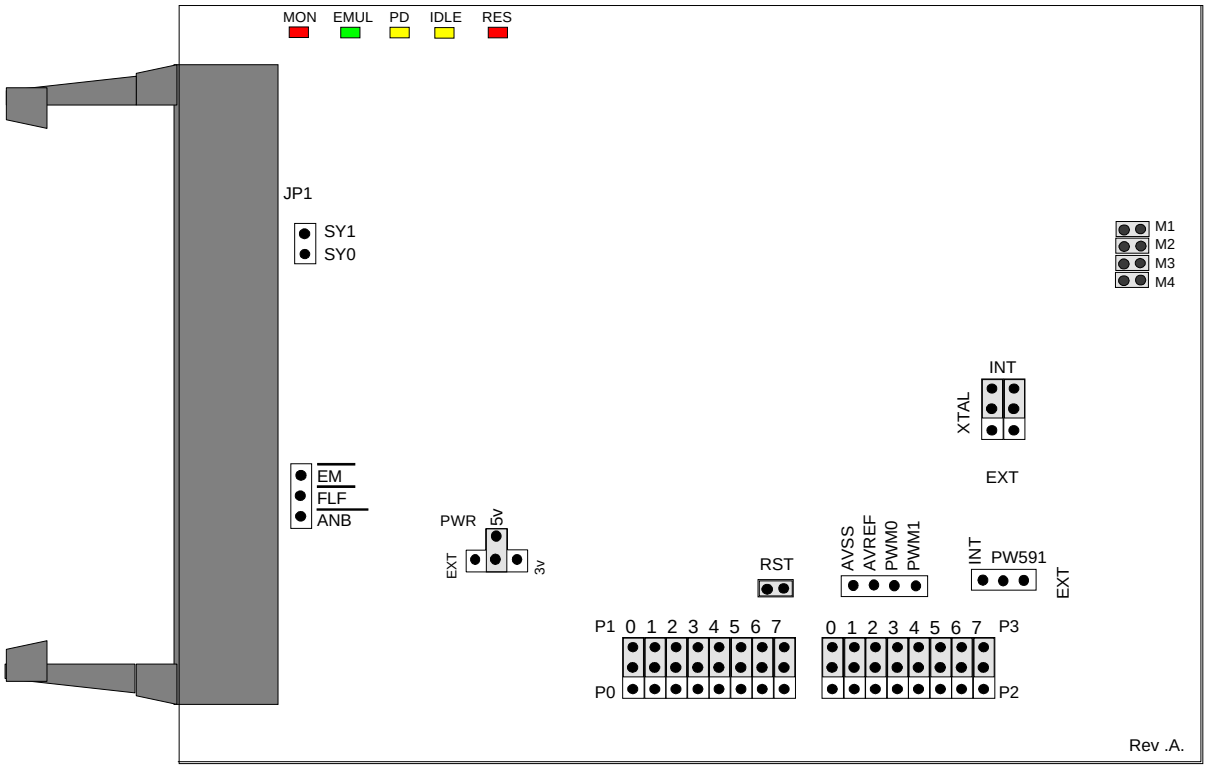


Figure 92. POD–51HB–C51FX Jumpers and Headers

Operating Instructions

To work with this pod, you need to select **POD–C51FX** from the **Select Processor** drop-down list box (**Step 3** of the **Communications** tab in the Emulator Configuration window).

The EPROM on the emulator board must be version COM 1.4.

The solder side of the board has a 44-pin PGA plug. The plug goes into the target system through a 44-pin PGA to 44-pin PLCC or 44-pin QFP adapter. (You can order this separately.)

The pod board can have any of the following Philips MCUs installed: 8xC51RD+, 8xC51RD2, 8Xc51FA/B/C, 8xC52, or 8xC66x. These chips are using a special emulation mode. This emulation mode also uses a programmable gate array from XILINX (LCA).

POD–51HB–C51FX

LED Indicators and Reset Switch

Function	Description
MON	A red LED that indicates when the system is in monitor mode. MON turns off during emulation mode (Run or Go) and turns back on when a break occurs and the system returns to monitor mode. In monitor mode, the MCU is executing code that is internal to the emulator. This code is not user code and is used to communicate with the host PC, to set up breakpoints, and so on.
EMUL	A green LED that indicates when the system is in emulation mode. (It performs the inverse function of MON.) In emulation mode, the MCU is executing the user's code from the emulation RAM or the targets PROM, depending on the mapping in the emulator software.
PD	A yellow LED that indicates when the MCU is in powder-down mode.
IDLE	A yellow LED that indicates when the MCU is in idle mode. If you break in idle or power-down mode, you must issue "reset chip" to regain communication with the MCU.
RES	A red LED that indicates the MCU RESET pin is in the active state.

Jumper Identification and Description

Jumper Designation	Function	Description
JP1	SY0, SY1	This header contains two pins: SY1 (top pin), and SY0 (bottom pin). These two signals are carried to the emulator and trace boards, and can be used in the trace and breakpoint logic.
JP4	M1, M2	These jumpers determine the mode of operation of P0, P2, P3.6 and P3.7. If only internal memory is to be emulated, both jumpers (M1 and M2) should be in. This enables P0, P2, P3.6 and P3.7 to be used as I/O ports. **M1 Jumper Removed** • If this jumper is out, P0 has the lower address and data. P2 has the upper address for external code fetches. • PSEN is active. • The emulator mapping determines whether the code is read from emulation memory or target memory. **M2 Jumper Removed** • If this jumper is out, P0 has the lower address and data. P2 has the upper address for MOVX instructions. • P3.6 and P3.7 is used for the RD and WR strobes. • The mapping determines if the RD and WR strobes go to emulation memory or target memory. • If only internal RAM is used, the emulator board has 128K RAM, map data MAPX to target, and the M2 jumper should be in. **Note**: You can change the mode jumpers M1and M2 at any time; however, you must issue a reset chip operation for the LCA and MCU to properly change modes.

POD–51HB–C51FX

Jumper Identification and Description (continued)

Jumper Designation	Function	Description
JP4	M3	This jumper should be installed. It configures the pod to operate in the 12 clocks per cycle mode. (This pod does not support the 6 clocks per cycle mode.)
JP4	M4	This jumper is factory set and should be left installed.
EM/, FLF/, ANB		This header carries signals from the emulator. • The EM/ is high when the emulator is in monitor mode, and low if in emulation mode. • The ANB/FLF/ pins are used with the enhanced trace boards as a signal output from a state machine logic.
AVSS/AVREF/ PWM0/PWM1		This header provides convenient test points for the analog reference voltages AVREF (+) and AVSS (-), PWM1 and PWM0 output strobes.
PWR	MCU Power Selection	This jumper selects the source of power for the MCU on the pod. • A jumper installed on the 5V or 3V position (depending on the type of MCU) selects power from the emulator. • A jumper installed in the EXT position selects power from the target system. **Note**: All jumpers should be in the INT position when a target is not connected to the pod board.
RST		This jumper connects the target RESET pin to the emulator. If the target system has a watchdog, it might interfere with the emulator in monitor mode. If this occurs, remove RST.
P1, P0		• The upper pins carry the MCU Port 1 pins: the P1.0 pin is on the left, and the P1.7 pin is on the right. Port 1 is connected directly to the MCU. • The lower pins carry Port 0: P0.0 is on the left, and P0.7 is on the right. Port 0 is emulated in the LCA which enables P0 to have better sink capability than the MCU. • Output signals also appear three clock cycles later than the real part. • The middle row of the jumper P1/P0 is normally connected to P1 using the jumpers. The signal level on the middle pins will be reflected on the trace display. • The board is delivered with the jumpers set, so that P1 will be traced. • You can connect external signals to the middle pins if the jumpers are removed. • When a jumper is connected to the trace, an extra load ALS TTL load is added (74ALS258).

POD–51HB–C51FX

Jumper Identification and Description (continued)

Jumper Designation	Function	Description
P2, P3		• The upper pins carry the MCU Port 3 pins: the P3.0 pin is on the left, and the P3.7 pin is on the right. Port 3 is connected directly to the MCU except for P3.6 and P3.7. These pins go through the high speed switch which adds approximately 100 Ohms to their output impedance. • The lower pins carry Port 2: P2.0 is on the left, and P2.7 is on the right. Port 2 is emulated in the LCA which enables P2 to have better sink capability than the MCU. • Port 2 output signals also appear three clock cycles later than the real part. • The middle row of the jumper P2/P3 is normally connected to P3 using the jumpers. The signal level on the middle pins will be reflected on the trace display. • The board is delivered with the jumpers set, so that P3 will be traced. • You can connect external signals to the middle pins if the jumpers are removed. • When a jumper is connected to the trace, an extra load ALS TTL load is added (74ALS258).
XTAL	Crystal Selection	Jumper pins for selecting the source of the crystal or clock • Jumpers installed in the top two positions (INT position) select the crystal on the pod. • Jumpers installed in the lower positions (EXT position) select the crystal on the target. **Notes:** • If you use an external clock, note that XTAL1 is an input and XTAL2 is left open. • All jumpers should be in the INT position when a target is not connected to the pod board.
S1	Reset	Resets the MCU and can be used instead of the target system reset.

Notes

- Ports 0 and 2 are emulated, and have a slightly different AC and DC characteristics:
  - Emulated I/O pins sink a minimum of 4 mA, and source (P2) minimum 4 mA for two clock cycles after an output low to high transition.
  - As high outputs or inputs, P2 ports have 22K-Ohm pull-up resistors.

POD–51HB–C51FX

- The trace of accesses to the on-chip AUX-RAM contains correct addresses and data for writes. However, if you have XDATA mapped to the emulator, the data will be written to the emulator RAM. For reads, the Data Display window in Seehau might contain the emulation RAM data if data is mapped to the emulator. This data might not be the same as the AUX-RAM unless you map the XDATA to the target. It is recommended that you always map the AUX-RAM region to the target.

- When accessing target memory, P0, P2, and ALE will be delayed approximately 25 ns.

- The clock circuitry is emulated, but is very close to the MCU clock.

- The timer/counters are stopped at breakpoints. This usually indicates that the serial port also stops at breakpoints. If a character is received or sent at that moment, the character will be distorted.

POD–51HB–C51RX2

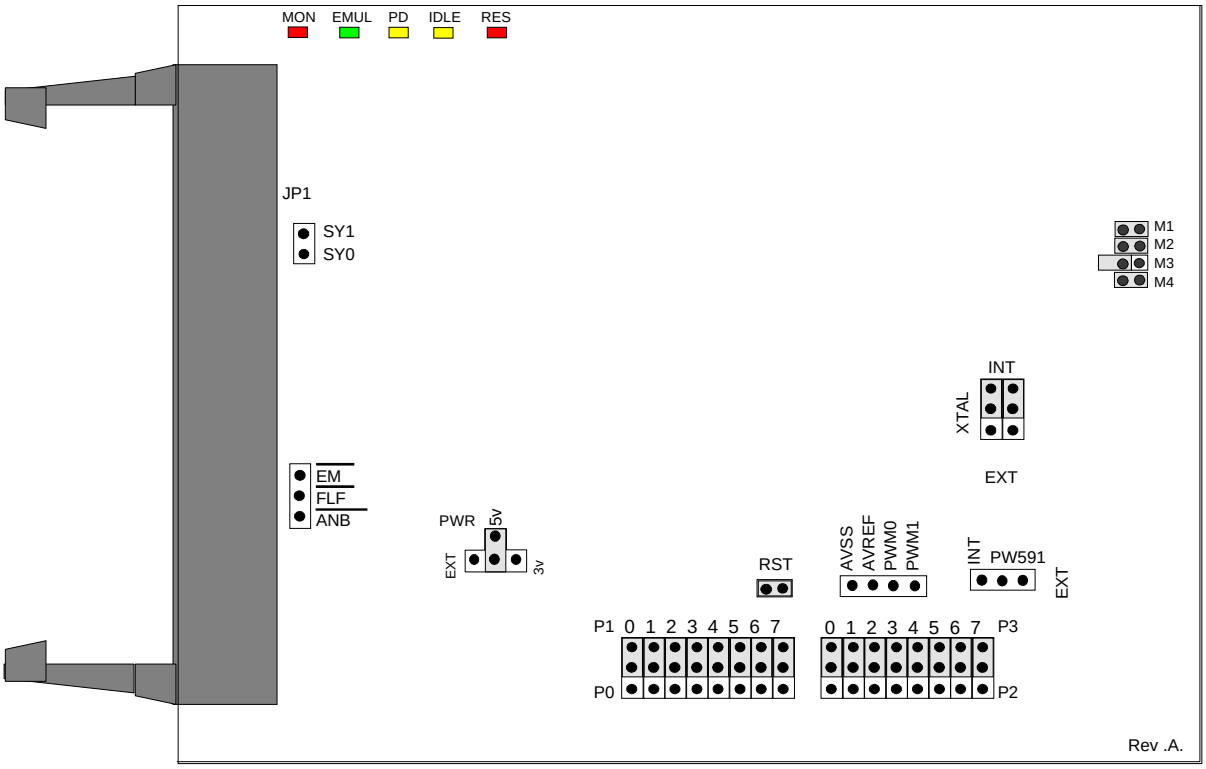


Figure 93. POD–51HB–C51RX2 Jumpers and Headers

Operating Instructions

To work with this pod, you need to select **POD–HB–C51RX2** from the **Select Processor** drop-down list box (**Step 3** of the **Communications** tab in the Emulator Configuration window).

The EPROM on the emulator board must be version COM 1.4.

The solder side of the board has a 44-pin PGA plug. The plug goes into the target system through a 44-pin PGA to 44-pin PLCC or 44-pin QFP adapter. (You can order this separately.)

The pod board can have any of the following Philips MCUs installed: 8xC51RD+, 8xC51RD2, 8Xc51FA/B/C, 8xC52, or 8xC66x. These chips are using a special emulation mode. This emulation mode also uses a programmable gate array from XILINX (LCA).

POD–51HB–C51RX2

LED Indicators and Reset Switch

Function	Description
MON	A red LED that indicates when the system is in monitor mode. MON turns off during emulation mode (Run or Go) and turns back on when a break occurs and the system returns to monitor mode. In monitor mode, the MCU is executing code that is internal to the emulator. This code is not user code and is used to communicate with the host PC, to set up breakpoints, and so on.
EMUL	A green LED that indicates when the system is in emulation mode. (It performs the inverse function of MON.) In emulation mode, the MCU is executing the user's code from the emulation RAM or the targets PROM, depending on the mapping in the emulator software.
PD	A yellow LED that indicates when the MCU is in powder-down mode.
IDLE	A yellow LED that indicates when the MCU is in idle mode. If you break in idle or power-down mode, you must issue "reset chip" to regain communication with the MCU.
RES	A red LED that indicates the MCU RESET pin is in the active state.

Jumper Identification and Description

Jumper Designation	Function	Description
JP1	SY0, SY1	This header contains two pins: SY1 (top pin), and SY0 (bottom pin). These two signals are carried to the emulator and trace boards, and can be used in the trace and breakpoint logic.
JP4	M1, M2	These jumpers determine the mode of operation of P0, P2, P3.6 and P3.7. If only internal memory is to be emulated, both jumpers (M1 and M2) should be in. This enables P0, P2, P3.6 and P3.7 to be used as I/O ports. **M1 Jumper Removed** • If this jumper is out, P0 has the lower address and data. P2 has the upper address for external code fetches. • PSEN is active. • The emulator mapping determines whether the code is read from emulation memory or target memory. **M2 Jumper Removed** • If this jumper is out, P0 has the lower address and data. P2 has the upper address for MOVX instructions. • P3.6 and P3.7 is used for the RD and WR strobes. • The mapping determines if the RD and WR strobes go to emulation memory or target memory. • If only internal RAM is used, the emulator board has 128K RAM, map data MAPX to target, and the M2 jumper should be in. **Note**: You can change the mode jumpers M1and M2 at any time; however, you must issue a reset chip operation for the LCA and MCU to properly change modes.

POD–51HB–C51RX2

Jumper Identification and Description (continued)

Jumper Designation	Function	Description
JP4	M3	This jumper should be installed. It configures the pod to operate in the 12 clocks per cycle mode. (This pod does not support the 6 clocks per cycle mode.)
JP4	M4	This jumper is factory set and should be left installed.
EM/, FLF/, ANB		This header carries signals from the emulator. • The EM/ is high when the emulator is in monitor mode, and low if in emulation mode. • The ANB/FLF/ pins are used with the enhanced trace boards as a signal output from a state machine logic.
AVSS/AVREF/ PWM0/PWM1		This header provides convenient test points for the analog reference voltages AVREF (+) and AVSS (-), PWM1 and PWM0 output strobes.
PWR	MCU Power Selection	This jumper selects the source of power for the MCU on the pod. • A jumper installed on the 5V or 3V position (depending on the type of MCU) selects power from the emulator. • A jumper installed in the EXT position selects power from the target system. **Note**: All jumpers should be in the INT position when a target is not connected to the pod board.
RST		This jumper connects the target RESET pin to the emulator. If the target system has a watchdog, it might interfere with the emulator in monitor mode. If this occurs, remove RST.
P1, P0		• The upper pins carry the MCU Port 1 pins: the P1.0 pin is on the left, and the P1.7 pin is on the right. Port 1 is connected directly to the MCU. • The lower pins carry Port 0: P0.0 is on the left, and P0.7 is on the right. Port 0 is emulated in the LCA which enables P0 to have better sink capability than the MCU. • Output signals also appear three clock cycles later than the real part. • The middle row of the jumper P1/P0 is normally connected to P1 using the jumpers. The signal level on the middle pins will be reflected on the trace display. • The board is delivered with the jumpers set, so that P1 will be traced. • You can connect external signals to the middle pins if the jumpers are removed. • When a jumper is connected to the trace, an extra load ALS TTL load is added (74ALS258).

POD–51HB–C51RX2

Jumper Identification and Description (continued)

Jumper Designation	Function	Description
P2, P3		• The upper pins carry the MCU Port 3 pins: the P3.0 pin is on the left, and the P3.7 pin is on the right. Port 3 is connected directly to the MCU except for P3.6 and P3.7. These pins go through the high speed switch which adds approximately 100 Ohms to their output impedance. • The lower pins carry Port 2: P2.0 is on the left, and P2.7 is on the right. Port 2 is emulated in the LCA which enables P2 to have better sink capability than the MCU. • Port 2 output signals also appear three clock cycles later than the real part. • The middle row of the jumper P2/P3 is normally connected to P3 using the jumpers. The signal level on the middle pins will be reflected on the trace display. • The board is delivered with the jumpers set, so that P3 will be traced. • You can connect external signals to the middle pins if the jumpers are removed. • When a jumper is connected to the trace, an extra load ALS TTL load is added (74ALS258).
XTAL	Crystal Selection	Jumper pins for selecting the source of the crystal or clock • Jumpers installed in the top two positions (INT position) select the crystal on the pod. • Jumpers installed in the lower positions (EXT position) select the crystal on the target is to be used. **Notes:** • If you use an external clock, note that XTAL1 is an input and XTAL2 is left open. • All jumpers should be in the INT position when a target is not connected to the pod board.
S1	Reset	Resets the MCU and can be used instead of the target system reset.

Notes

- Ports 0 and 2 are emulated, and have a slightly different AC and DC characteristics:
  - Emulated I/O pins sink a minimum of 4 mA, and source (P2) minimum 4 mA for two clock cycles after an output low to high transition.
  - As high outputs or inputs, P2 ports have 22K-Ohm pull-up resistors.

POD–51HB–C51RX2

- The trace of accesses to the on-chip AUX-RAM contains correct addresses and data for writes. However, if you have XDATA mapped to the emulator, the data will be written to the emulator RAM. For reads, the Data Display window in Seehau might contain the emulation RAM data if data is mapped to the emulator. This data might not be the same as the AUX-RAM unless you map the XDATA to the target. It is recommended that you always map the AUX-RAM region to the target.

- When accessing target memory, P0, P2, and ALE will be delayed approximately 25 ns.

- The clock circuitry is emulated, but is very close to the MCU clock.

- The timer/counters are stopped at breakpoints. This usually indicates that the serial port also stops at breakpoints. If a character is received or sent at that moment, the character will be distorted.

POD–51HB–C51RX

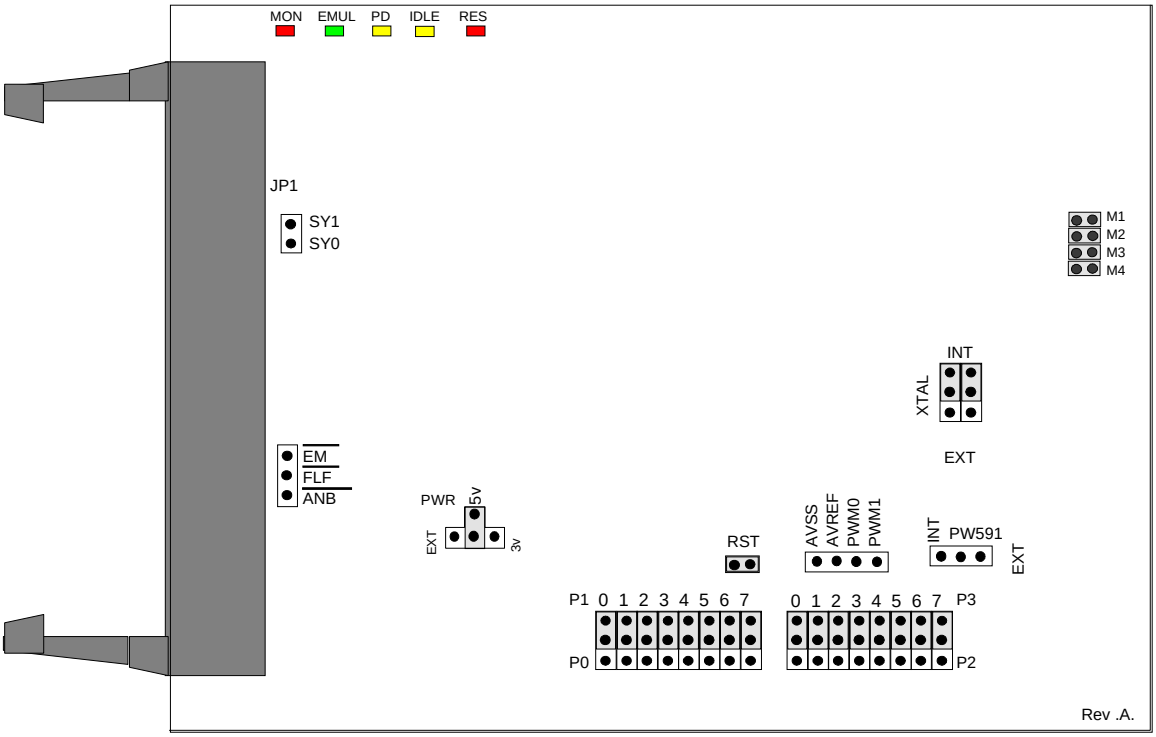


Figure 94. POD–51HB–C51RX Jumpers and Headers

Operating Instructions

To work with this pod, you need to select **POD–C51RX** from the **Select Processor** drop-down list box (**Step 3** of the **Communications** tab in the Emulator Configuration window).

The EPROM on the emulator board must be version COM 1.4.

The solder side of the board has a 44-pin PGA plug. The plug goes into the target system through a 44-pin PGA to 44-pin PLCC or 44-pin QFP adapter. (You can order this separately.)

The pod board can have any of the following Philips MCUs installed: 8xC51RD+, 8xC51RD2, 8Xc51FA/B/C, 8xC52, or 8xC66x. These chips are using a special emulation mode. This emulation mode also uses a programmable gate array from XILINX (LCA).

POD–51HB–C51RX

LED Indicators and Reset Switch

Function	Description
MON	A red LED that indicates when the system is in monitor mode. MON turns off during emulation mode (Run or Go) and turns back on when a break occurs and the system returns to monitor mode. In monitor mode, the MCU is executing code that is internal to the emulator. This code is not user code and is used to communicate with the host PC, to set up breakpoints, and so on.
EMUL	A green LED that indicates when the system is in emulation mode. (It performs the inverse function of MON.) In emulation mode, the MCU is executing the user's code from the emulation RAM or the targets PROM, depending on the mapping in the emulator software.
PD	A yellow LED that indicates when the MCU is in powder-down mode.
IDLE	A yellow LED that indicates when the MCU is in idle mode. If you break in idle or power-down mode, you must issue "reset chip" to regain communication with the MCU.
RES	A red LED that indicates the MCU RESET pin is in the active state.

Jumper Identification and Description

Jumper Designation	Function	Description
JP1	SY0, SY1	This header contains two pins: SY1 (top pin), and SY0 (bottom pin). These two signals are carried to the emulator and trace boards, and can be used in the trace and breakpoint logic.
JP4	M1, M2	These jumpers determine the mode of operation of P0, P2, P3.6 and P3.7. If only internal memory is to be emulated, both jumpers (M1 and M2) should be in. This enables P0, P2, P3.6 and P3.7 to be used as I/O ports. **M1 Jumper Removed** • If this jumper is out, P0 has the lower address and data. P2 has the upper address for external code fetches. • PSEN is active. • The emulator mapping determines whether the code is read from emulation memory or target memory. **M2 Jumper Removed** • If this jumper is out, P0 has the lower address and data. P2 has the upper address for MOVX instructions. • P3.6 and P3.7 is used for the RD and WR strobes. • The mapping determines if the RD and WR strobes go to emulation memory or target memory. • If only internal RAM is used, the emulator board has 128K RAM, map data MAPX to target, and the M2 jumper should be in. **Note**: You can change the mode jumpers M1and M2 at any time; however, you must issue a reset chip operation for the LCA and MCU to properly change modes.

POD–51HB–C51RX

Jumper Identification and Description (continued)

Jumper Designation	Function	Description
JP4	M3	This jumper should be installed. It configures the pod to operate in the 12 clocks per cycle mode. (This pod does not support the 6 clocks per cycle mode.)
JP4	M4	This jumper is factory set and should be left installed.
EM/, FLF/, ANB		This header carries signals from the emulator. • The EM/ is high when the emulator is in monitor mode, and low if in emulation mode. • The ANB/FLF/ pins are used with the enhanced trace boards as a signal output from a state machine logic.
AVSS/AVREF/ PWM0/PWM1		This header provides convenient test points for the analog reference voltages AVREF (+) and AVSS (-), PWM1 and PWM0 output strobes.
PWR	MCU Power Selection	This jumper selects the source of power for the MCU on the pod. • A jumper installed on the 5V or 3V position (depending on the type of MCU) selects power from the emulator. • A jumper installed in the EXT position selects power from the target system. **Note**: All jumpers should be in the INT position when a target is not connected to the pod board.
RST		This jumper connects the target RESET pin to the emulator. If the target system has a watchdog, it might interfere with the emulator in monitor mode. If this occurs, remove RST.
P1, P0		• The upper pins carry the MCU Port 1 pins: the P1.0 pin is on the left, and the P1.7 pin is on the right. Port 1 is connected directly to the MCU. • The lower pins carry Port 0: P0.0 is on the left, and P0.7 is on the right. Port 0 is emulated in the LCA which enables P0 to have better sink capability than the MCU. • Output signals also appear three clock cycles later than the real part. • The middle row of the jumper P1/P0 is normally connected to P1 using the jumpers. The signal level on the middle pins will be reflected on the trace display. • The board is delivered with the jumpers set, so that P1 will be traced. • You can connect external signals to the middle pins if the jumpers are removed. • When a jumper is connected to the trace, an extra load ALS TTL load is added (74ALS258).

POD–51HB–C51 RX

Jumper Identification and Description (continued)

Jumper Designation	Function	Description
P2, P3		• The upper pins carry the MCU Port 3 pins: the P3.0 pin is on the left, and the P3.7 pin is on the right. Port 3 is connected directly to the MCU except for P3.6 and P3.7. These pins go through the high speed switch which adds approximately 100 Ohms to their output impedance. • The lower pins carry Port 2: P2.0 is on the left, and P2.7 is on the right. Port 2 is emulated in the LCA which enables P2 to have better sink capability than the MCU. • Port 2 output signals also appear three clock cycles later than the real part. • The middle row of the jumper P2/P3 is normally connected to P3 using the jumpers. The signal level on the middle pins will be reflected on the trace display. • The board is delivered with the jumpers set, so that P3 will be traced. • You can connect external signals to the middle pins if the jumpers are removed. • When a jumper is connected to the trace, an extra load ALS TTL load is added (74ALS258).
XTAL	Crystal Selection	Jumper pins for selecting the source of the crystal or clock • Jumpers installed in the top two positions (INT position) select the crystal on the pod. • Jumpers installed in the lower positions (EXT position) select the crystal on the target. **Notes:** • If you use an external clock, note that XTAL1 is an input and XTAL2 is left open. • All jumpers should be in the INT position when a target is not connected to the pod board.
S1	Reset	Resets the MCU and can be used instead of the target system reset.

Notes

- Ports 0 and 2 are emulated, and have a slightly different AC and DC characteristics:
  - Emulated I/O pins sink a minimum of 4 mA, and source (P2) minimum 4 mA for two clock cycles after an output low to high transition.
  - As high outputs or inputs, P2 ports have 22K-Ohm pull-up resistors.

POD–51HB–C51RX

- The trace of accesses to the on-chip AUX-RAM contains correct addresses and data for writes. However, if you have XDATA mapped to the emulator, the data will be written to the emulator RAM. For reads, the Data Display window in Seehau might contain the emulation RAM data if data is mapped to the emulator. This data might not be the same as the AUX-RAM unless you map the XDATA to the target. It is recommended that you always map the AUX-RAM region to the target

- When accessing target memory, P0, P2, and ALE will be delayed approximately 25 ns.

- The clock circuitry is emulated, but is very close to the MCU clock.

- The timer/counters are stopped at breakpoints. This usually indicates that the serial port also stops at breakpoints. If a character is received or sent at that moment, the character will be distorted.

POD–51HB–C52

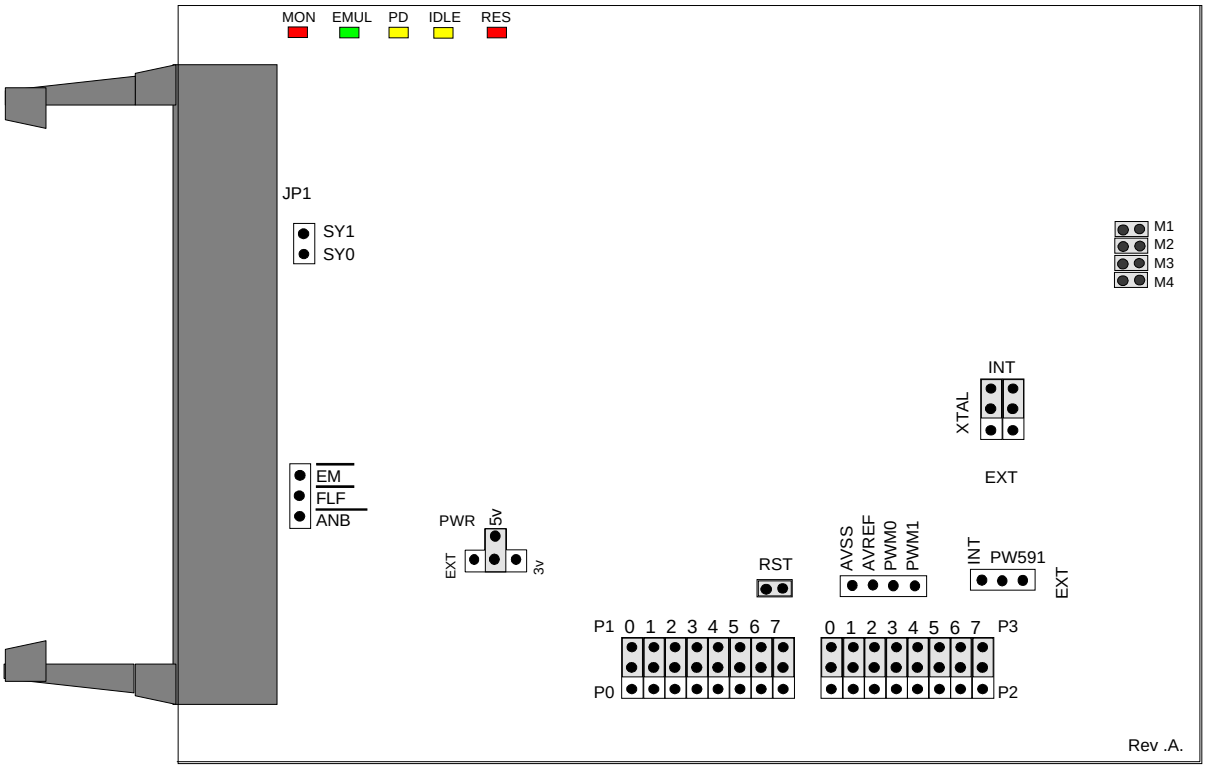


Figure 95. POD–51HB–C52 Jumpers and Headers

Operating Instructions

To work with this pod, you need to select **POD–C52** from the **Select Processor** drop-down list box (**Step 3** of the **Communications** tab in the Emulator Configuration window).

The EPROM on the emulator board must be version COM 1.4.

The solder side of the board has a 44-pin PGA plug. The plug goes into the target system through a 44-pin PGA to 44-pin PLCC or 44-pin QFP adapter. (You can order this separately.)

The pod board can have any of the following Philips MCUs installed: 8xC51RD+, 8xC51RD2, 8Xc51FA/B/C, 8xC52, or 8xC66x. These chips are using a special emulation mode. This emulation mode also uses a programmable gate array from XILINX (LCA).

POD–51HB–C52

LED Indicators and Reset Switch

Function	Description
MON	A red LED that indicates when the system is in monitor mode. MON turns off during emulation mode (Run or Go) and turns back on when a break occurs and the system returns to monitor mode. In monitor mode, the MCU is executing code that is internal to the emulator. This code is not user code and is used to communicate with the host PC, to set up breakpoints, and so on.
EMUL	A green LED that indicates when the system is in emulation mode. (It performs the inverse function of MON.) In emulation mode, the MCU is executing the user's code from the emulation RAM or the targets PROM, depending on the mapping in the emulator software.
PD	A yellow LED that indicates when the MCU is in powder-down mode.
IDLE	A yellow LED that indicates when the MCU is in idle mode. If you break in idle or power-down mode, you must issue "reset chip" to regain communication with the MCU.
RES	A red LED that indicates the MCU RESET pin is in the active state.

Jumper Identification and Description

Jumper Designation	Function	Description
JP1	SY0, SY1	This header contains two pins: SY1 (top pin), and SY0 (bottom pin). These two signals are carried to the emulator and trace boards, and can be used in the trace and breakpoint logic.
JP4	M1, M2	These jumpers determine the mode of operation of P0, P2, P3.6 and P3.7. If only internal memory is to be emulated, both jumpers (M1 and M2) should be in. This enables P0, P2, P3.6 and P3.7 to be used as I/O ports. **M1 Jumper Removed** • If this jumper is out, P0 has the lower address and data. P2 has the upper address for external code fetches. • PSEN is active. • The emulator mapping determines whether the code is read from emulation memory or target memory. **M2 Jumper Removed** • If this jumper is out, P0 has the lower address and data. P2 has the upper address for MOVX instructions. • P3.6 and P3.7 is used for the RD and WR strobes. • The mapping determines if the RD and WR strobes go to emulation memory or target memory. • If only internal RAM is used, the emulator board has 128K RAM, map data MAPX to target, and the M2 jumper should be in. **Note**: You can change the mode jumpers M1and M2 at any time; however, you must issue a reset chip operation for the LCA and MCU to properly change modes.

POD–51HB–C52

Jumper Identification and Description (continued)

Jumper Designation	Function	Description
JP4	M3	This jumper should be installed. It configures the pod to operate in the 12 clocks per cycle mode. (This pod does not support the 6 clocks per cycle mode.)
JP4	M4	This jumper is factory set and should be left installed.
EM/, FLF/, ANB		This header carries signals from the emulator. • The EM/ is high when the emulator is in monitor mode, and low if in emulation mode. • The ANB/FLF/ pins are used with the enhanced trace boards as a signal output from a state machine logic.
AVSS/AVREF/ PWM0/PWM1		This header provides convenient test points for the analog reference voltages AVREF (+) and AVSS (-), PWM1 and PWM0 output strobes.
PWR	MCU Power Selection	This jumper selects the source of power for the MCU on the pod. • A jumper installed on the 5V or 3V position (depending on the type of MCU) selects power from the emulator. • A jumper installed in the EXT position selects power from the target system. **Note**: All jumpers should be in the INT position when a target is not connected to the pod board.
RST		This jumper connects the target RESET pin to the emulator. If the target system has a watchdog, it might interfere with the emulator in monitor mode. If this occurs, remove RST.
P1, P0		• The upper pins carry the MCU Port 1 pins: the P1.0 pin is on the left, and the P1.7 pin is on the right. Port 1 is connected directly to the MCU. • The lower pins carry Port 0: P0.0 is on the left, and P0.7 is on the right. Port 0 is emulated in the LCA which enables P0 to have better sink capability than the MCU. • Output signals also appear three clock cycles later than the real part. • The middle row of the jumper P1/P0 is normally connected to P1 using the jumpers. The signal level on the middle pins will be reflected on the trace display. • The board is delivered with the jumpers set, so that P1 will be traced. • You can connect external signals to the middle pins if the jumpers are removed. • When a jumper is connected to the trace, an extra load ALS TTL load is added (74ALS258).

POD–51HB–C52

Jumper Identification and Description (continued)

Jumper Designation	Function	Description
P2, P3		• The upper pins carry the MCU Port 3 pins: the P3.0 pin is on the left, and the P3.7 pin is on the right. Port 3 is connected directly to the MCU except for P3.6 and P3.7. These pins go through the high speed switch which adds approximately 100 Ohms to their output impedance. • The lower pins carry Port 2: P2.0 is on the left, and P2.7 is on the right. Port 2 is emulated in the LCA which enables P2 to have better sink capability than the MCU. • Port 2 output signals also appear three clock cycles later than the real part. • The middle row of the jumper P2/P3 is normally connected to P3 using the jumpers. The signal level on the middle pins will be reflected on the trace display. • The board is delivered with the jumpers set, so that P3 will be traced. • You can connect external signals to the middle pins if the jumpers are removed. • When a jumper is connected to the trace, an extra load ALS TTL load is added (74ALS258).
XTAL	Crystal Selection	Jumper pins for selecting the source of the crystal or clock • Jumpers installed in the top two positions (INT position) select the crystal on the pod. • Jumpers installed in the lower positions (EXT position) select the crystal on the target. **Notes:** • If you use an external clock, note that XTAL1 is an input and XTAL2 is left open. • All jumpers should be in the INT position when a target is not connected to the pod board.
S1	Reset	Resets the MCU and can be used instead of the target system reset.

Notes

- Ports 0 and 2 are emulated, and have a slightly different AC and DC characteristics:
  - Emulated I/O pins sink a minimum of 4 mA, and source (P2) minimum 4 mA for two clock cycles after an output low to high transition.
  - As high outputs or inputs, P2 ports have 22K-Ohm pull-up resistors.

POD–51HB–C52

- The trace of accesses to the on-chip AUX-RAM contains correct addresses and data for writes. However, if you have XDATA mapped to the emulator, the data will be written to the emulator RAM. For reads, the Data Display window in Seehau might contain the emulation RAM data if data is mapped to the emulator. This data might not be the same as the AUX-RAM unless you map the XDATA to the target. It is recommended that you always map the AUX-RAM region to the target.

- When accessing target memory, P0, P2, and ALE will be delayed approximately 25 ns.

- The clock circuitry is emulated, but is very close to the MCU clock.

- The timer/counters are stopped at breakpoints. This usually indicates that the serial port also stops at breakpoints. If a character is received or sent at that moment, the character will be distorted.

POD–51HB–C591

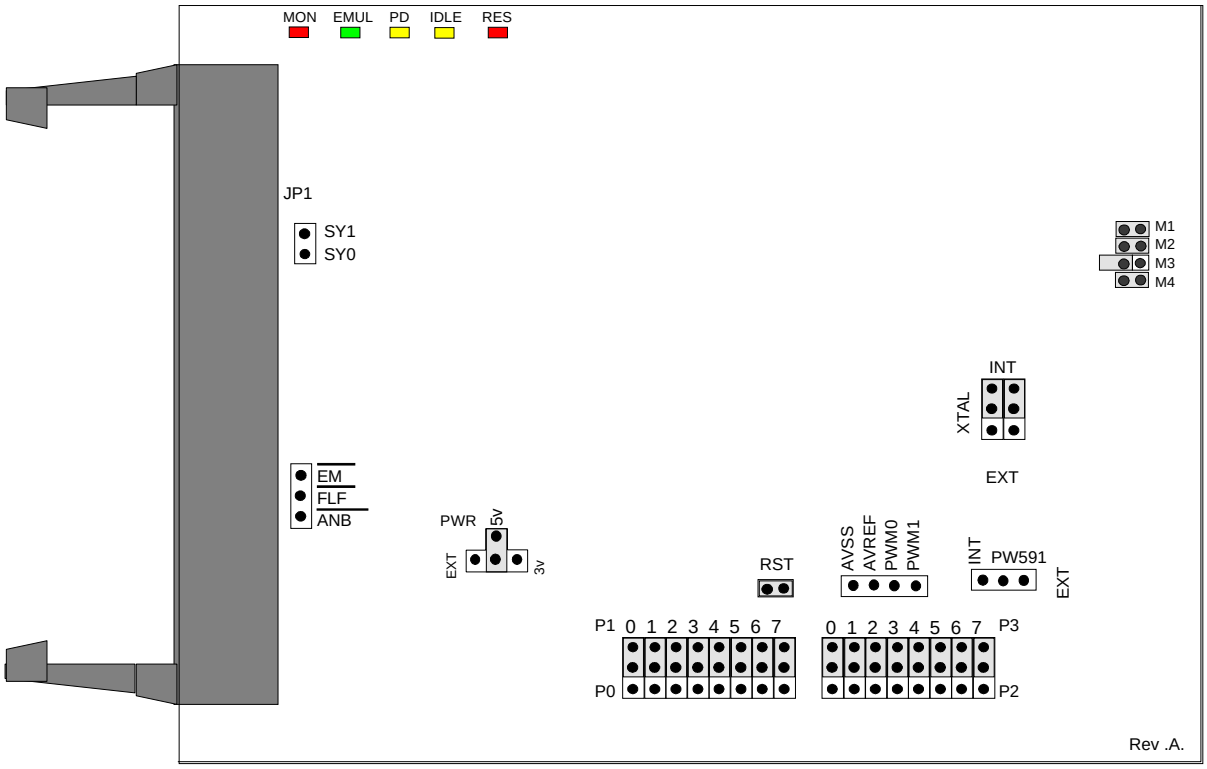


Figure 96. POD–51HB–C591 Jumpers and Headers

Operating Instructions

To work with this pod, you need to select **POD–C591** from the **Select Processor** drop-down list box (**Step 3** of the **Communications** tab in the Emulator Configuration window).

The EPROM on the emulator board must be version COM 1.4.

The solder side of the board has a 44-pin PGA plug. The plug goes into the target system through a 44-pin PGA to 44-pin PLCC or 44-pin QFP adapter. (You can order this separately.)

These chips are using a special emulation mode. This emulation mode also uses a programmable gate array from XILINX (LCA).

> **Note**
>
> This pod only supports the Philips 8xC591 MCU.

POD–51HB–C591

LED Indicators and Reset Switch

Function	Description
MON	A red LED that indicates when the system is in monitor mode. MON turns off during emulation mode (Run or Go) and turns back on when a break occurs and the system returns to monitor mode. In monitor mode, the MCU is executing code that is internal to the emulator. This code is not user code and is used to communicate with the host PC, to set up breakpoints, and so on.
EMUL	A green LED that indicates when the system is in emulation mode. (It performs the inverse function of MON.) In emulation mode, the MCU is executing the user's code from the emulation RAM or the targets PROM, depending on the mapping in the emulator software.
PD	A yellow LED that indicates when the MCU is in powder-down mode.
IDLE	A yellow LED that indicates when the MCU is in idle mode. If you break in idle or power-down mode, you must issue "reset chip" to regain communication with the MCU.
RES	A red LED that indicates the MCU RESET pin is in the active state.

Jumper Identification and Description

Jumper Designation	Function	Description
JP1	SY0, SY1	This header contains two pins: SY1 (top pin), and SY0 (bottom pin). These two signals are carried to the emulator and trace boards, and can be used in the trace and breakpoint logic.
JP4	M1, M2	These jumpers determine the mode of operation of P0, P2, P3.6 and P3.7. If only internal memory is to be emulated, both jumpers (M1 and M2) should be in. This enables P0, P2, P3.6 and P3.7 to be used as general purpose ports. **M1 Jumper Removed** • If this jumper is out, P0 has the lower address and data. P2 has the upper address for external code fetches. • PSEN is active. • The emulator mapping determines whether the code is read from emulation memory or target memory. **M2 Jumper Removed** • If this jumper is out, P0 has the lower address and data. P2 has the upper address for MOVX instructions. • P3.6 and P3.7 is used for the RD and WR strobes. • The mapping determines if the RD and WR strobes go to emulation memory or target memory. • If only internal RAM is used, the emulator board has 128K RAM, map data MAPX to target, and the M2 jumper should be in. **Note**: You can change the mode jumpers M1and M2 at any time; however, you must issue a reset chip operation for the LCA and MCU to properly change modes.

POD–51HB–C591

Jumper Identification and Description (continued)

Jumper Designation	Function	Description
JP4	M3	This jumper should be installed. It configures the pod to operate in the 12 clocks per cycle mode. (This pod does not support the 6 clocks per cycle mode.)
JP4	M4	This jumper is factory set and should be left installed.
EM/, FLF/, ANB		This header carries signals from the emulator. • The EM/ is high when the emulator is in monitor mode, and low if in emulation mode. • The ANB/FLF/ pins are used with the enhanced trace boards as a signal output from a state machine logic.
AVSS/AVREF/ PWM0/PWM1		This header provides convenient test points for the analog reference voltages AVREF (+) and AVSS (-), PWM1 and PWM0 output strobes.
PWR	MCU Power Selection	This jumper selects the source of power for the MCU on the pod. • A jumper installed on the 5V or 3V position (depending on the type of MCU) selects power from the emulator. • A jumper installed in the EXT position selects power from the target system. **Note**: All jumpers should be in the INT position when a target is not connected to the pod board.
RST		This jumper connects the target RESET pin to the emulator. If the target system has a watchdog, it might interfere with the emulator in monitor mode. If this occurs, remove RST.
P1, P0		• The upper pins carry the MCU Port 1 pins: the P1.0 pin is on the left, and the P1.7 pin is on the right. Port 1 is connected directly to the MCU. • The lower pins carry Port 0: P0.0 is on the left, and P0.7 is on the right. Port 0 is emulated in the LCA which enables P0 to have better sink capability than the MCU. • Output signals also appear three clock cycles later than the real part. • The middle row of the jumper P1/P0 is normally connected to P1 using the jumpers. The signal level on the middle pins will be reflected on the trace display. • The board is delivered with the jumpers set, so that P1 will be traced. • You can connect external signals to the middle pins if the jumpers are removed. • When a jumper is connected to the trace, an extra load ALS TTL load is added (74ALS258).

POD–51HB–C591

Jumper Identification and Description (continued)

Jumper Designation	Function	Description
P2, P3		• The upper pins carry the MCU Port 3 pins: the P3.0 pin is on the left, and the P3.7 pin is on the right. Port 3 is connected directly to the MCU except for P3.6 and P3.7. These pins go through the high speed switch which adds approximately 100 Ohms to their output impedance. • The lower pins carry Port 2: P2.0 is on the left, and P2.7 is on the right. Port 2 is emulated in the LCA which enables P2 to have better sink capability than the MCU. • Port 2 output signals also appear three clock cycles later than the real part. • The middle row of the jumper P2/P3 is normally connected to P3 using the jumpers. The signal level on the middle pins will be reflected on the trace display. • The board is delivered with the jumpers set, so that P3 will be traced. • You can connect external signals to the middle pins if the jumpers are removed. • When a jumper is connected to the trace, an extra load ALS TTL load is added (74ALS258).
XTAL	Crystal Selection	Jumper pins for selecting the source of the crystal or clock • Jumpers installed in the top two positions (INT position) select the crystal on the pod. • Jumpers installed in the lower positions (EXT position) select the crystal on the target. **Notes:** • If you use an external clock, note that XTAL1 is an input and XTAL2 is left open. • All jumpers should be in the INT position when a target is not connected to the pod board.
S1	Reset	Resets the MCU and can be used instead of the target system reset.

Notes

- Ports 0 and 2 are emulated, and have a slightly different AC and DC characteristics:
  - Emulated I/O pins sink a minimum of 4 mA, and source (P2) minimum 4 mA for two clock cycles after an output low to high transition.
  - As high outputs or inputs, P2 ports have 22K-Ohm pull-up resistors.

POD–51HB–C591

- The trace of accesses to the on-chip AUX-RAM contains correct addresses and data for writes. However, if you have XDATA mapped to the emulator, the data will be written to the emulator RAM. For reads, the Data Display window in Seehau might contain the emulation RAM data if data is mapped to the emulator. This data might not be the same as the AUX-RAM unless you map the XDATA to the target. It is recommended that you always map the AUX-RAM region to the target.

- When accessing target memory, P0, P2, and ALE will be delayed approximately 25 ns.

- The clock circuitry is emulated, but is very close to the MCU clock.

- The timer/counters are stopped at breakpoints. This usually indicates that the serial port also stops at breakpoints. If a character is received or sent at that moment, the character will be distorted.

POD–51HB–C66X

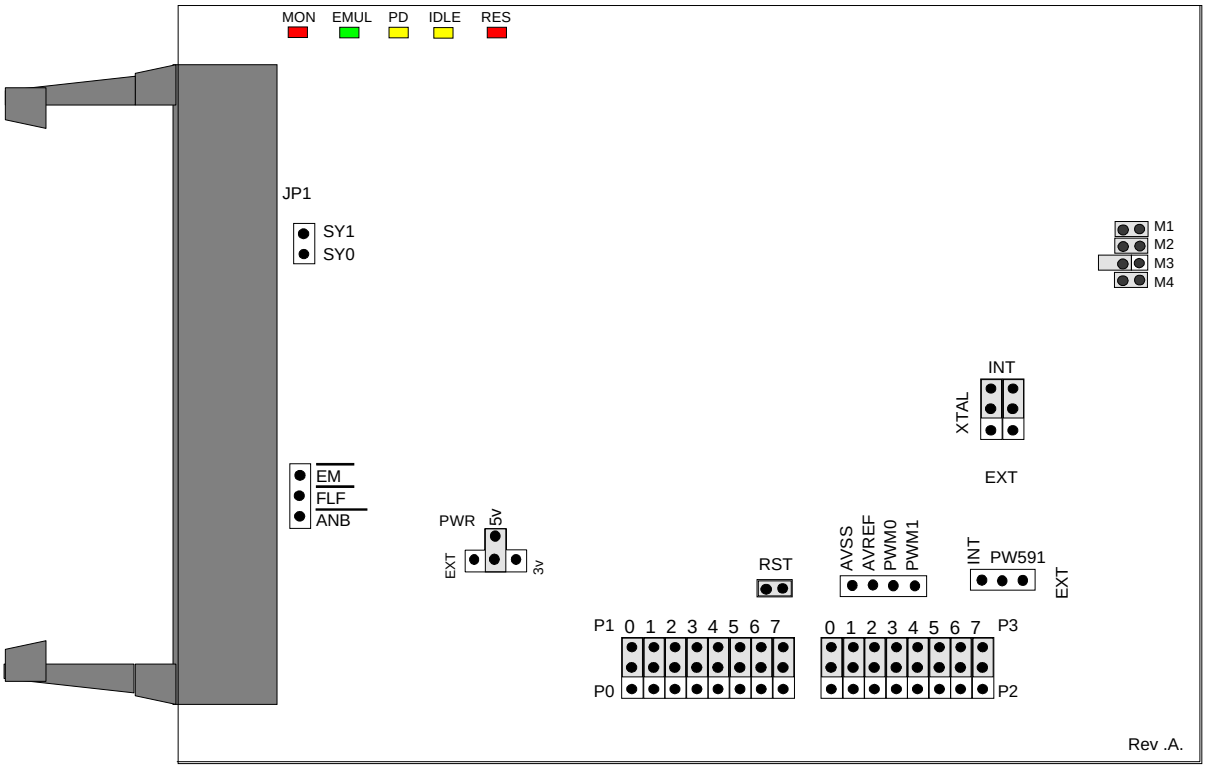


Figure 97. POD–51HB–C66X Jumpers and Headers

Operating Instructions

To work with this pod, you need to select **POD–HB–C66X** from the **Select Processor** drop-down list box (**Step 3** of the **Communications** tab in the Emulator Configuration window).

The EPROM on the emulator board must be version COM 1.4.

The solder side of the board has a 44-pin PGA plug. The plug goes into the target system through a 44-pin PGA to 44-pin PLCC or 44-pin QFP adapter. (You can order this separately.)

The pod board can have any of the following Philips MCUs installed: 8xC51RD+, 8xC51RD2, 8Xc51FA/B/C, 8xC52, or 8xC66x. These chips are using a special emulation mode. This emulation mode also uses a programmable gate array from XILINX (LCA).

POD–51HB–C66X

LED Indicators and Reset Switch

Function	Description
MON	A red LED that indicates when the system is in monitor mode. MON turns off during emulation mode (Run or Go) and turns back on when a break occurs and the system returns to monitor mode. In monitor mode, the MCU is executing code that is internal to the emulator. This code is not user code and is used to communicate with the host PC, to set up breakpoints, and so on.
EMUL	A green LED that indicates when the system is in emulation mode. (It performs the inverse function of MON.) In emulation mode, the MCU is executing the user's code from the emulation RAM or the targets PROM, depending on the mapping in the emulator software.
PD	A yellow LED that indicates when the MCU is in powder-down mode.
IDLE	A yellow LED that indicates when the MCU is in idle mode. If you break in idle or power-down mode, you must issue "reset chip" to regain communication with the MCU.
RES	A red LED that indicates the MCU RESET pin is in the active state.

Jumper Identification and Description

Jumper Designation	Function	Description
JP1	SY0, SY1	This header contains two pins: SY1 (top pin), and SY0 (bottom pin). These two signals are carried to the emulator and trace boards, and can be used in the trace and breakpoint logic.
JP4	M1, M2	These jumpers determine the mode of operation of P0, P2, P3.6 and P3.7. If only internal memory is to be emulated, both jumpers (M1 and M2) should be in. This enables P0, P2, P3.6 and P3.7 to be used as general purpose ports. **M1 Jumper Removed** • If this jumper is out, P0 has the lower address and data. P2 has the upper address for external code fetches. • PSEN is active. • The emulator mapping determines whether the code is read from emulation memory or target memory. **M2 Jumper Removed** • If this jumper is out, P0 has the lower address and data. P2 has the upper address for MOVX instructions. • P3.6 and P3.7 is used for the RD and WR strobes. • The mapping determines if the RD and WR strobes go to emulation memory or target memory. • If only internal RAM is used, the emulator board has 128K RAM, map data MAPX to target, and the M2 jumper should be in. **Note**: You can change the mode jumpers M1and M2 at any time; however, you must issue a reset chip operation for the LCA and MCU to properly change modes.

POD–51HB–C66X

Jumper Identification and Description (continued)

Jumper Designation	Function	Description
JP4	M3	This jumper should be installed. It configures the pod to operate in the 12 clocks per cycle mode. (This pod does not support the 6 clocks per cycle mode.)
JP4	M4	This jumper is factory set and should be left installed.
EM/, FLF/, ANB		This header carries signals from the emulator. • The EM/ is high when the emulator is in monitor mode, and low if in emulation mode. • The ANB/FLF/ pins are used with the enhanced trace boards as a signal output from a state machine logic.
AVSS/AVREF/ PWM0/PWM1		This header provides convenient test points for the analog reference voltages AVREF (+) and AVSS (-), PWM1 and PWM0 output strobes.
PWR	MCU Power Selection	This jumper selects the source of power for the MCU on the pod. • A jumper installed on the 5V or 3V position (depending on the type of MCU) selects power from the emulator. • A jumper installed in the EXT position selects power from the target system. **Note**: All jumpers should be in the INT position when a target is not connected to the pod board.
RST		This jumper connects the target RESET pin to the emulator. If the target system has a watchdog, it might interfere with the emulator in monitor mode. If this occurs, remove RST.
P1, P0		• The upper pins carry the MCU Port 1 pins: the P1.0 pin is on the left, and the P1.7 pin is on the right. Port 1 is connected directly to the MCU. • The lower pins carry Port 0: P0.0 is on the left, and P0.7 is on the right. Port 0 is emulated in the LCA which enables P0 to have better sink capability than the MCU. • Output signals also appear three clock cycles later than the real part. • The middle row of the jumper P1/P0 is normally connected to P1 using the jumpers. The signal level on the middle pins will be reflected on the trace display. • The board is delivered with the jumpers set, so that P1 will be traced. • You can connect external signals to the middle pins if the jumpers are removed. • When a jumper is connected to the trace, an extra load ALS TTL load is added (74ALS258).

POD–51HB–C66X

Jumper Identification and Description (continued)

Jumper Designation	Function	Description
P2, P3		• The upper pins carry the MCU Port 3 pins: the P3.0 pin is on the left, and the P3.7 pin is on the right. Port 3 is connected directly to the MCU except for P3.6 and P3.7. These pins go through the high speed switch which adds approximately 100 Ohms to their output impedance. • The lower pins carry Port 2: P2.0 is on the left, and P2.7 is on the right. Port 2 is emulated in the LCA which enables P2 to have better sink capability than the MCU. • Port 2 output signals also appear three clock cycles later than the real part. • The middle row of the jumper P2/P3 is normally connected to P3 using the jumpers. The signal level on the middle pins will be reflected on the trace display. • The board is delivered with the jumpers set, so that P3 will be traced. • You can connect external signals to the middle pins if the jumpers are removed. • When a jumper is connected to the trace, an extra load ALS TTL load is added (74ALS258).
XTAL	Crystal Selection	Jumper pins for selecting the source of the crystal or clock • Jumpers installed in the top two positions (INT position) select the crystal on the pod. • Jumpers installed in the lower positions (EXT position) select the crystal on the target. **Notes:** • If you use an external clock, note that XTAL1 is an input and XTAL2 is left open. • All jumpers should be in the INT position when a target is not connected to the pod board.
S1	Reset	Resets the MCU and can be used instead of the target system reset.

Notes

- Ports 0 and 2 are emulated, and have a slightly different AC and DC characteristics:
  - Emulated I/O pins sink a minimum of 4 mA, and source (P2) minimum 4 mA for two clock cycles after an output low to high transition.
  - As high outputs or inputs, P2 ports have 22K-Ohm pull-up resistors.

POD–51HB–C66X

- The trace of accesses to the on-chip AUX-RAM contains correct addresses and data for writes. However, if you have XDATA mapped to the emulator, the data will be written to the emulator RAM. For reads, the Data Display window in Seehau might contain the emulation RAM data if data is mapped to the emulator. This data might not be the same as the AUX-RAM unless you map the XDATA to the target. It is recommended that you always map the AUX-RAM region to the target.

- When accessing target memory, P0, P2, and ALE will be delayed approximately 25 ns.

- The clock circuitry is emulated, but is very close to the MCU clock.

- The timer/counters are stopped at breakpoints. This usually indicates that the serial port also stops at breakpoints. If a character is received or sent at that moment, the character will be distorted.

POD–51HB–L51FX

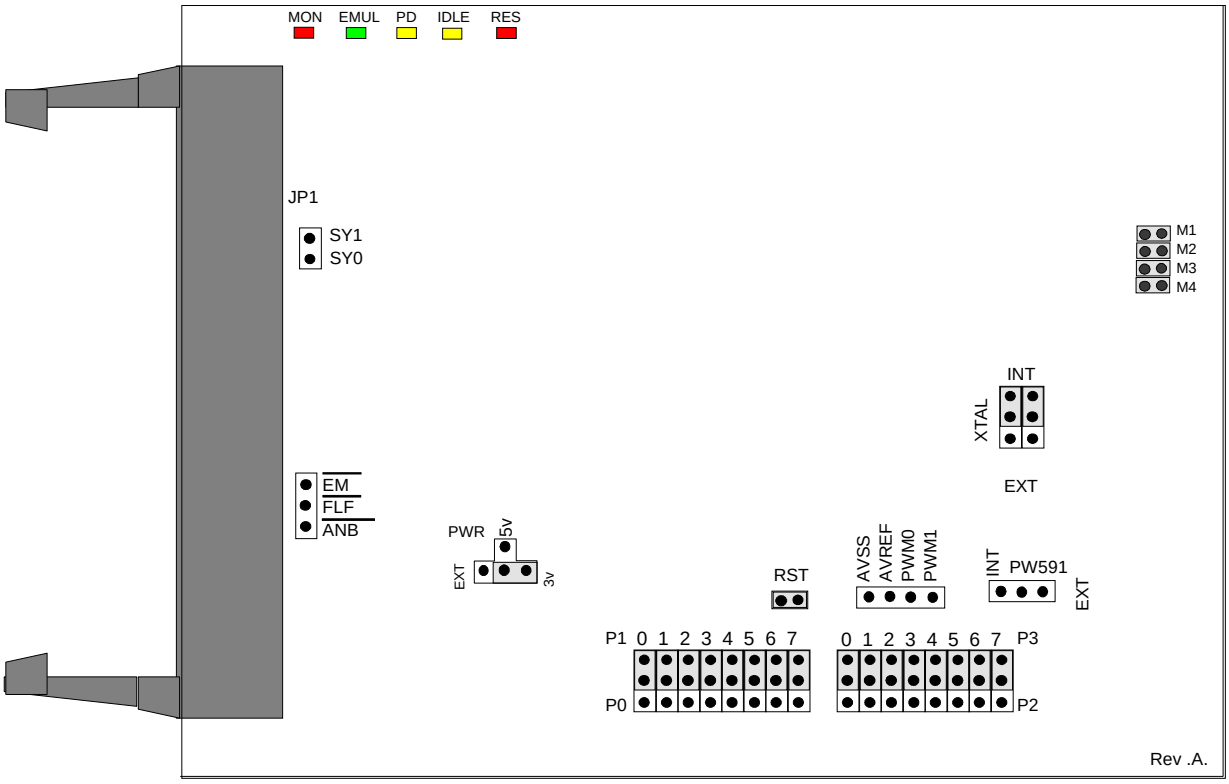


Figure 98. POD–51HB–L51FX Jumpers and Headers

Operating Instructions

To work with this pod, you need to select **POD–HB–C51FX** from the **Select Processor** drop-down list box (**Step 3** of the **Communications** tab in the Emulator Configuration window).

The EPROM on the emulator board must be version COM 1.4.

The solder side of the board has a 44-pin PGA plug. The plug goes into the target system through a 44-pin PGA to 44-pin PLCC or 44-pin QFP adapter. (You can order this separately.)

The pod board can have any of the following Philips MCUs installed: 8xC51RD+, 8xC51RD2, 8Xc51FA/B/C, 8xC52, or 8xC66x. These chips are using a special emulation mode. This emulation mode also uses a programmable gate array from XILINX (LCA).

POD–51HB–L51FX

LED Indicators and Reset Switch

Function	Description
MON	A red LED that indicates when the system is in monitor mode. MON turns off during emulation mode (Run or Go) and turns back on when a break occurs and the system returns to monitor mode. In monitor mode, the MCU is executing code that is internal to the emulator. This code is not user code and is used to communicate with the host PC, to set up breakpoints, and so on.
EMUL	A green LED that indicates when the system is in emulation mode. (It performs the inverse function of MON.) In emulation mode, the MCU is executing the user's code from the emulation RAM or the targets PROM, depending on the mapping in the emulator software.
PD	A yellow LED that indicates when the MCU is in powder-down mode.
IDLE	A yellow LED that indicates when the MCU is in idle mode. If you break in idle or power-down mode, you must issue "reset chip" to regain communication with the MCU.
RES	A red LED that indicates the MCU RESET pin is in the active state.

Jumper Identification and Description

Jumper Designation	Function	Description
JP1	SY0, SY1	This header contains two pins: SY1 (top pin), and SY0 (bottom pin). These two signals are carried to the emulator and trace boards, and can be used in the trace and breakpoint logic.
JP4	M1, M2	These jumpers determine the mode of operation of P0, P2, P3.6 and P3.7. If only internal memory is to be emulated, both jumpers (M1 and M2) should be in. This enables P0, P2, P3.6 and P3.7 to be used as general purpose ports. **M1 Jumper Removed** • If this jumper is out, P0 has the lower address and data. P2 has the upper address for external code fetches. • PSEN is active. • The emulator mapping determines whether the code is read from emulation memory or target memory. **M2 Jumper Removed** • If this jumper is out, P0 has the lower address and data. P2 has the upper address for MOVX instructions. • P3.6 and P3.7 is used for the RD and WR strobes. • The mapping determines if the RD and WR strobes go to emulation memory or target memory. • If only internal RAM is used, the emulator board has 128K RAM, map data MAPX to target, and the M2 jumper should be in. **Note**: You can change the mode jumpers M1and M2 at any time; however, you must issue a reset chip operation for the LCA and MCU to properly change modes.

POD–51HB–L51FX

Jumper Identification and Description (continued)

Jumper Designation	Function	Description
JP4	M3	This jumper should be installed. It configures the pod to operate in the 12 clocks per cycle mode. (This pod does not support the 6 clocks per cycle mode.)
JP4	M4	This jumper is factory set and should be left installed.
EM/, FLF/, ANB		This header carries signals from the emulator. • The EM/ is high when the emulator is in monitor mode, and low if in emulation mode. • The ANB/FLF/ pins are used with the enhanced trace boards as a signal output from a state machine logic.
AVSS/AVREF/ PWM0/PWM1		This header provides convenient test points for the analog reference voltages AVREF (+) and AVSS (-), PWM1 and PWM0 output strobes.
PWR	MCU Power Selection	This jumper selects the source of power for the MCU on the pod. • A jumper installed on the 5V or 3V position (depending on the type of MCU) selects power from the emulator. • A jumper installed in the EXT position selects power from the target system. **Note**: All jumpers should be in the INT position when a target is not connected to the pod board.
RST		This jumper connects the target RESET pin to the emulator. If the target system has a watchdog, it might interfere with the emulator in monitor mode. If this occurs, remove RST.
P1, P0		• The upper pins carry the MCU Port 1 pins: the P1.0 pin is on the left, and the P1.7 pin is on the right. Port 1 is connected directly to the MCU. • The lower pins carry Port 0: P0.0 is on the left, and P0.7 is on the right. Port 0 is emulated in the LCA which enables P0 to have better sink capability than the MCU. • Output signals also appear three clock cycles later than the real part. • The middle row of the jumper P1/P0 is normally connected to P1 using the jumpers. The signal level on the middle pins will be reflected on the trace display. • The board is delivered with the jumpers set, so that P1 will be traced. • You can connect external signals to the middle pins if the jumpers are removed. • When a jumper is connected to the trace, an extra load ALS TTL load is added (74ALS258).

POD–51HB–L51FX

Jumper Identification and Description (continued)

Jumper Designation	Function	Description
P2, P3		• The upper pins carry the MCU Port 3 pins: the P3.0 pin is on the left, and the P3.7 pin is on the right. Port 3 is connected directly to the MCU except for P3.6 and P3.7. These pins go through the high speed switch which adds approximately 100 Ohms to their output impedance. • The lower pins carry Port 2: P2.0 is on the left, and P2.7 is on the right. Port 2 is emulated in the LCA which enables P2 to have better sink capability than the MCU. • Port 2 output signals also appear three clock cycles later than the real part. • The middle row of the jumper P2/P3 is normally connected to P3 using the jumpers. The signal level on the middle pins will be reflected on the trace display. • The board is delivered with the jumpers set, so that P3 will be traced. • You can connect external signals to the middle pins if the jumpers are removed. • When a jumper is connected to the trace, an extra load ALS TTL load is added (74ALS258).
XTAL	Crystal Selection	Jumper pins for selecting the source of the crystal or clock • Jumpers installed in the top two positions (INT position) select the crystal on the pod. • Jumpers installed in the lower positions (EXT position) select the crystal on the target. **Notes:** • If you use an external clock, note that XTAL1 is an input and XTAL2 is left open. • All jumpers should be in the INT position when a target is not connected to the pod board.
S1	Reset	Resets the MCU and can be used instead of the target system reset.

Notes

- Ports 0 and 2 are emulated, and have a slightly different AC and DC characteristics:
  - Emulated I/O pins sink a minimum of 4 mA, and source (P2) minimum 4 mA for two clock cycles after an output low to high transition.
  - As high outputs or inputs, P2 ports have 22K-Ohm pull-up resistors.

POD–51HB–L51FX

- The trace of accesses to the on-chip AUX-RAM contains correct addresses and data for writes. However, if you have XDATA mapped to the emulator, the data will be written to the emulator RAM. For reads, the Data Display window in Seehau might contain the emulation RAM data if data is mapped to the emulator. This data might not be the same as the AUX-RAM unless you map the XDATA to the target. It is recommended that you always map the AUX-RAM region to the target.

- When accessing target memory, P0, P2, and ALE will be delayed approximately 25 ns.

- The clock circuitry is emulated, but is very close to the MCU clock.

- The timer/counters are stopped at breakpoints. This usually indicates that the serial port also stops at breakpoints. If a character is received or sent at that moment, the character will be distorted.

POD–51HB–L51RX

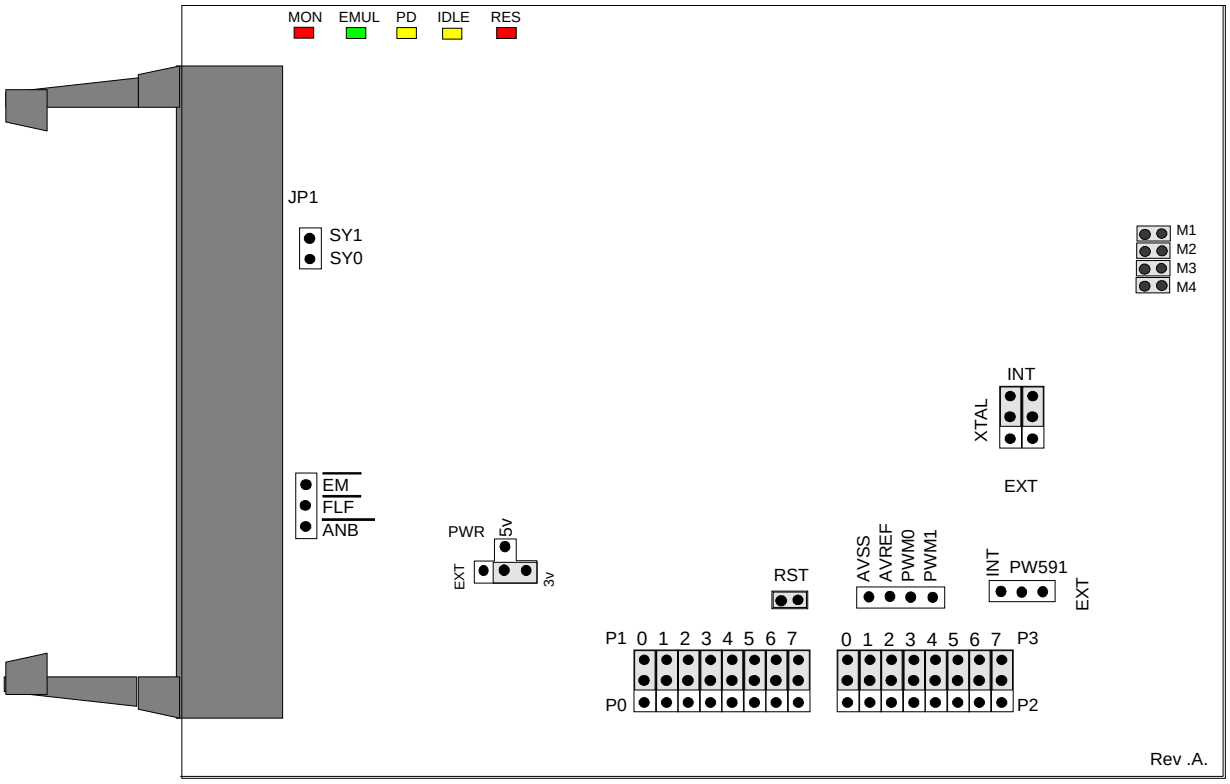


Figure 99. POD–51HB–L51RX Jumpers and Headers

Operating Instructions

To work with this pod, you need to select **POD–HB–C51RX** from the **Select Processor** drop-down list box (**Step 3** of the **Communications** tab in the Emulator Configuration window).

The EPROM on the emulator board must be version COM 1.4.

The solder side of the board has a 44-pin PGA plug. The plug goes into the target system through a 44-pin PGA to 44-pin PLCC or 44-pin QFP adapter. (You can order this separately.)

The pod board can have any of the following Philips MCUs installed: 8xC51RD+, 8xC51RD2, 8Xc51FA/B/C, 8xC52, or 8xC66x. These chips are using a special emulation mode. This emulation mode also uses a programmable gate array from XILINX (LCA).

POD–51HB–L51RX

LED Indicators and Reset Switch

Function	Description
MON	A red LED that indicates when the system is in monitor mode. MON turns off during emulation mode (Run or Go) and turns back on when a break occurs and the system returns to monitor mode. In monitor mode, the MCU is executing code that is internal to the emulator. This code is not user code and is used to communicate with the host PC, to set up breakpoints, and so on.
EMUL	A green LED that indicates when the system is in emulation mode. (It performs the inverse function of MON.) In emulation mode, the MCU is executing the user's code from the emulation RAM or the targets PROM, depending on the mapping in the emulator software.
PD	A yellow LED that indicates when the MCU is in powder-down mode.
IDLE	A yellow LED that indicates when the MCU is in idle mode. If you break in idle or power-down mode, you must issue "reset chip" to regain communication with the MCU.
RES	A red LED that indicates the MCU RESET pin is in the active state.

Jumper Identification and Description

Jumper Designation	Function	Description
JP1	SY0, SY1	This header contains two pins: SY1 (top pin), and SY0 (bottom pin). These two signals are carried to the emulator and trace boards, and can be used in the trace and breakpoint logic.
JP4	M1, M2	These jumpers determine the mode of operation of P0, P2, P3.6 and P3.7. If only internal memory is to be emulated, both jumpers (M1 and M2) should be in. This enables P0, P2, P3.6 and P3.7 to be used as general purpose ports. **M1 Jumper Removed** • If this jumper is out, P0 has the lower address and data. P2 has the upper address for external code fetches. • PSEN is active. • The emulator mapping determines whether the code is read from emulation memory or target memory. **M2 Jumper Removed** • If this jumper is out, P0 has the lower address and data. P2 has the upper address for MOVX instructions. • P3.6 and P3.7 is used for the RD and WR strobes. • The mapping determines if the RD and WR strobes go to emulation memory or target memory. • If only internal RAM is used, the emulator board has 128K RAM, map data MAPX to target, and the M2 jumper should be in. **Note**: You can change the mode jumpers M1and M2 at any time; however, you must issue a reset chip operation for the LCA and MCU to properly change modes.

POD–51HB–L51RX

Jumper Identification and Description (continued)

Jumper Designation	Function	Description
JP4	M3	This jumper should be installed. It configures the pod to operate in the 12 clocks per cycle mode. (This pod does not support the 6 clocks per cycle mode.)
JP4	M4	This jumper is factory set and should be left installed.
EM/, FLF/, ANB		This header carries signals from the emulator. • The EM/ is high when the emulator is in monitor mode, and low if in emulation mode. • The ANB/FLF/ pins are used with the enhanced trace boards as a signal output from a state machine logic.
AVSS/AVREF/ PWM0/PWM1		This header provides convenient test points for the analog reference voltages AVREF (+) and AVSS (-), PWM1 and PWM0 output strobes.
PWR	MCU Power Selection	This jumper selects the source of power for the MCU on the pod. • A jumper installed on the 5V or 3V position (depending on the type of MCU) selects power from the emulator. • A jumper installed in the EXT position selects power comes from the target system. **Note**: All jumpers should be in the INT position when a target is not connected to the pod board.
RST		This jumper connects the target RESET pin to the emulator. If the target system has a watchdog, it might interfere with the emulator in monitor mode. If this occurs, remove RST.
P1, P0		• The upper pins carry the MCU Port 1 pins: the P1.0 pin is on the left, and the P1.7 pin is on the right. Port 1 is connected directly to the MCU. • The lower pins carry Port 0: P0.0 is on the left, and P0.7 is on the right. Port 0 is emulated in the LCA which enables P0 to have better sink capability than the MCU. • Output signals also appear three clock cycles later than the real part. • The middle row of the jumper P1/P0 is normally connected to P1 using the jumpers. The signal level on the middle pins will be reflected on the trace display. • The board is delivered with the jumpers set, so that P1 will be traced. • You can connect external signals to the middle pins if the jumpers are removed. • When a jumper is connected to the trace, an extra load ALS TTL load is added (74ALS258).

POD–51HB–L51RX

Jumper Identification and Description (continued)

Jumper Designation	Function	Description
P2, P3		• The upper pins carry the MCU Port 3 pins: the P3.0 pin is on the left, and the P3.7 pin is on the right. Port 3 is connected directly to the MCU except for P3.6 and P3.7. These pins go through the high speed switch which adds approximately 100 Ohms to their output impedance. • The lower pins carry Port 2: P2.0 is on the left, and P2.7 is on the right. Port 2 is emulated in the LCA which enables P2 to have better sink capability than the MCU. • Port 2 output signals also appear three clock cycles later than the real part. • The middle row of the jumper P2/P3 is normally connected to P3 using the jumpers. The signal level on the middle pins will be reflected on the trace display. • The board is delivered with the jumpers set, so that P3 will be traced. • You can connect external signals to the middle pins if the jumpers are removed. • When a jumper is connected to the trace, an extra load ALS TTL load is added (74ALS258).
XTAL	Crystal Selection	Jumper pins for selecting the source of the crystal or clock • Jumpers installed in the top two positions (INT position) select the crystal on the pod. • Jumpers installed in the lower positions (EXT position) select the crystal on the target. **Notes:** • If you use an external clock, note that XTAL1 is an input and XTAL2 is left open. • All jumpers should be in the INT position when a target is not connected to the pod board.
S1	Reset	Resets the MCU and can be used instead of the target system reset.

Notes

- Ports 0 and 2 are emulated, and have a slightly different AC and DC characteristics:
  - Emulated I/O pins sink a minimum of 4 mA, and source (P2) minimum 4 mA for two clock cycles after an output low to high transition.
  - As high outputs or inputs, P2 ports have 22K-Ohm pull-up resistors.

POD–51HB–L51RX

- The trace of accesses to the on-chip AUX-RAM contains correct addresses and data for writes. However, if you have XDATA mapped to the emulator, the data will be written to the emulator RAM. For reads, the Data Display window in Seehau might contain the emulation RAM data if data is mapped to the emulator. This data might not be the same as the AUX-RAM unless you map the XDATA to the target. It is recommended that you always map the AUX-RAM region to the target.

- When accessing target memory, P0, P2, and ALE will be delayed approximately 25 ns.

- The clock circuitry is emulated, but is very close to the MCU clock.

- The timer/counters are stopped at breakpoints. This usually indicates that the serial port also stops at breakpoints. If a character is received or sent at that moment, the character will be distorted.

POD–C320–33/323–33: External Mode

Operating Instructions

The POD–31A is the base for advanced versions of the POD–31. It is intended to be used for the Dallas 80C320, Dallas 80C323, or other 40-pin DIP parts in external mode which have high clock rates. This pod can also be used to support bank switching applications. It also has the following features: three LED indicators, gated reset, and support for the Advanced/Enhanced Trace Board signals. It supports negative reset polarity and 3-volt operation (for example, Dallas 80C323).

To work with this pod, you need to select **POD–C320** from the **Select Processor** drop-down list box (**Step 3** of the **Communications** tab in the Emulator Configuration window).

The DS80C320 and DS80C323 MCUs on the pod are a special emulation version of the chip. These MCUs can be replaced with a standard chip; however, the timers will not stop in monitor mode.

For the POD–C320–33 or the POD–C323–33 to work properly, the emulator board EMUL51–PC/EA256–C320–BSW–50 or EMUL–PC/EA768–C320–BSW–50 must be used. These emulator boards have special jumper settings and come with COM 1.46 chips installed.

LED Indicators

Function	Description
MON	A red LED that indicates when the system is in monitor mode. MON turns off during emulation mode (Run or Go) and turns back on when a break occurs and the system returns to monitor mode.
EMUL	A green LED that indicates when the system is in emulation mode. (It performs the inverse function of MON.)
RST	A red LED that indicates the MCU is receiving a reset signal.

Jumper Identification and Description

Jumper Designation	Function	Description
XTAL	Crystal Selection	Jumper pins for selecting the source of the crystal or clock • Jumpers installed in the upper two positions (INT position) select the crystal on the pod. • Jumpers installed in the lower two positions (EXT position) select the crystal on the target. **Note**: This jumper should be in the INT position when a target is not connected to the pod board. This is also the default position.
POWER	Power Selection	• When jumpers are installed across the two lower pins (INT position), the power to the MCU comes from the emulator. • When jumpers are installed across the two upper pins (EXT position), the power to the MCU comes from the target. **Note**: This jumper should be in the INT position when a target is not connected to the pod board. This is also the default position.

POD–C323–33

Jumper Identification and Description (continued)

Jumper Designation	Function	Description
RES		This jumper connects the target RESET pin to the emulator. If the target system has a watchdog, it might interfere with the emulator in monitor mode. If this occurs, remove RES.
EM/		The EM/ pin is an output pin that indicates the emulator's status. This pin is high when the emulator is in monitor mode, and low when the emulator is in emulation mode.
SY0, SY1, E0, E1		These pins are used to connect the external signals to the emulator. They are also used for the trace function. The SY0 pin is used in the breakpoint logic.
BS0, BS1, BS2, BS3		These pins are used for bank switching. • BS0 (LSB) corresponds to the red wire. (Refer to the "Setting Up Bank Switch Connections" section in Chapter 2, "Installing the Hardware.") • BS1 corresponds to the green wire. • BS2 corresponds to the white wire. • BS3 (MSB) is normally used for tracing the RXD signal. If BS3 is used for bank switching, you must remove the jumper from RXD to BS3. • If BS0 and BS1 are used for bank switching, the jumpers at B0/ and B1/ must be moved up so that the jumpers B0/ and B1/ are between the + sign and the B#/ pins.
FLF/, ANB/		These pins are used with the Advanced Trace Board (ATR) and Enhanced Trace Board (ETR). They can not be used with bank switching. Refer to the "Configuring the Enhanced Trace Board" section in Chapter 7, "Configuring Trace Boards."
SPEED		This jumper should normally be in the upper SLOW position except when the DS80C320 runs at 20 MHz and above, when it needs to be in the lower FAST position.
PSEN/, RD/, WR/		• Signals PSEN/, RD/, and WR/ can be routed directly (non-gated) to the target system, or go through a gate, so that the target system is not accessed in monitor mode (gated). The board is delivered with the corresponding jumpers in the upper, gated position (GP/, GR/ and GW/). • The jumper for PSEN/ is at the left upper corner of the crystal Y1. The jumpers for RD/ and WR/ are to the left of the jumper block with PSEN/ and ALE. **CAUTION** Use caution if you connect the pod to a target when the jumpers PSEN/, RD/, or WR/ are in the non-gated position. The following can occur: • If a target device drives the data bus when the MCU is fetching code or reading XDATA memory mapped to the emulator, a bus contention can occur. This condition can cause the emulator to malfunction. • During XDATA writes, target memory is written to, even when accessing an address mapped to the emulator.

POD–C323–33

Jumper Identification and Description (continued)

Jumper Designation	Function	Description
RES-FUNC		If the RES-FUNC jumper is in, reset from the target system is active only in emulation mode. If the jumper is out, a reset from the target system resets the emulator in monitor mode also.
RESPOL		This jumper needs to be in the upper + position. For processors with active low reset, this jumper needs to be in the lower - position.
WR-ENABLE, RD-ENABLE		These jumpers are factory set and should not be changed.
P1		• The right row of pins is connected to Port 1: the P1.7 pin at the top, and the P1.0 pin at the bottom. • The left row of pins is reflected on the trace display of P1. • The board is delivered with the jumpers set, so that P1 will be traced. • You can connect external signals or signals from other ports to the left row of pins if the jumpers are removed. • When a jumper is connected to the trace, an extra load ALS TTL load is added (74ALS258).
P3		• The right row of pins is connected to Port 3: the P3.5 pin at the top, and the P3.0 pin at the bottom. • The left row of pins is reflected on the trace display of P3. • The board is delivered with the jumpers set, so that P3 will be traced. • You can connect external signals or signals from other ports to the left row of pins if the jumpers are removed. • When a jumper is connected to the trace, an extra load ALS TTL load is added (74ALS258).
S1	RESET	Resets the MCU and can be used instead of the target system reset.

Notes

- For Dallas DS80C320, jumper ALE must be in the upper (D) position. Also jumper TXD/FMC must be in the lower (FMC) position. Jumper SPEED needs to be in the upper S position for below 20 MHz. It needs to be in the lower F position at clock rates 20 MHz and above.

- The POD–C320–33 is designed to run at 5 volts.

- The POD–C323–33 is designed to run from 3 to 5 volts. When the POWER jumper is in the INT position, the MCU will run at 5 volts. For the MCU to run down to 3 volts, the POWER jumper must be in the EXT position. This can only be done when the pod is plugged into the target. Some current might flow from the emulator into the target system power (pin 40). The pod also might drive the P0 lines up to approximately 3.5 volts. The gate for PSEN/, RD/ and WR/ is powered from the same power as the MCU. All other signals to the target driven from the pod come directly from the MCU on the pod.

POD–C323–33

- SY0, SY1, E0, E1, BS0, BS1, BS2, BS3 have each a 100K-Ohm pull-up resistor.
- Signals PSEN/, RD/ and WR/ go through a 74ACT32 if they are gated. This adds approximately 5 ns to the timing of these signals. For DS80C320, ALE goes through a 5 ns PAL.

Board Layout

Figure 100 shows locations of jumpers and Figure 101 shows a diagram of the jumper pins.

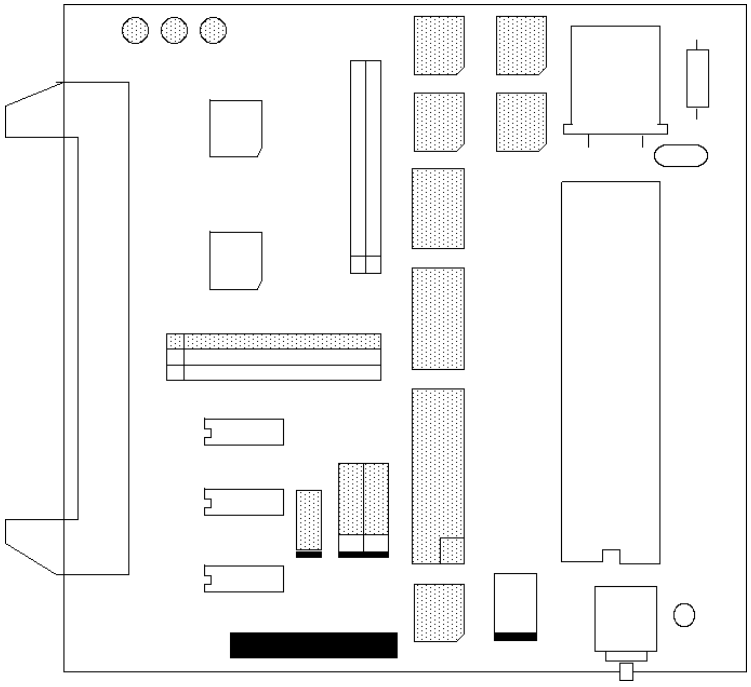


Figure 100. Locations of Jumpers on POD–C323–33

POD–C323–33

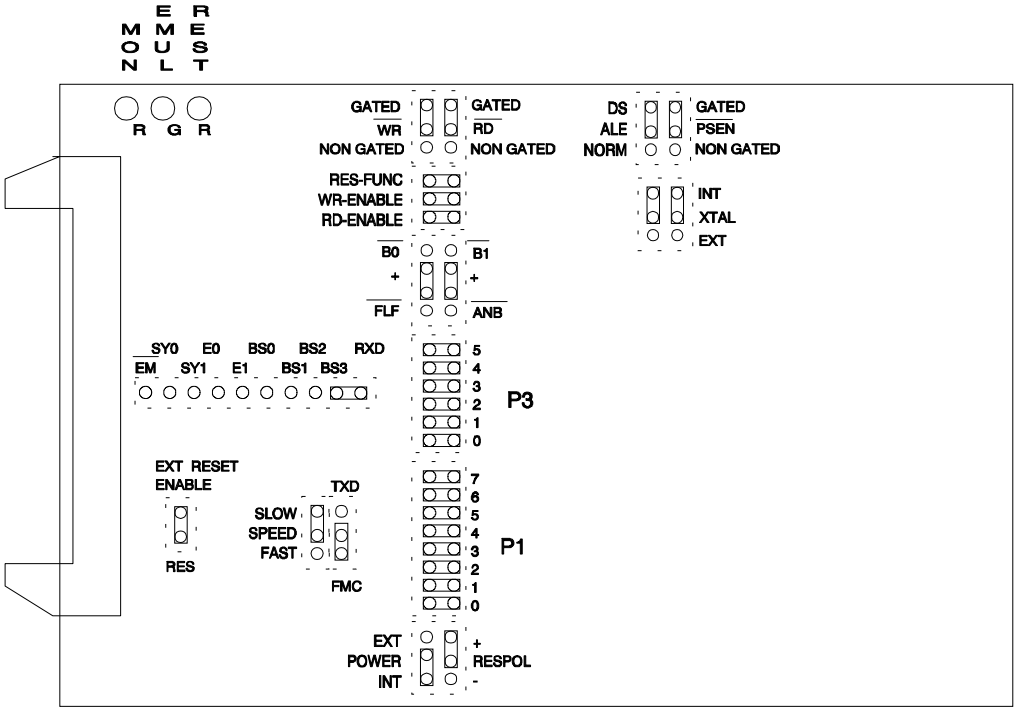


Figure 101. POD–C323-33 Jumper Diagram

POD–C32HF–42: External Mode

Operating Instructions

This section covers the following pods:

- POD–C32HF–42
- POD–TS80C31X2
- POD–TS80C32X2
- POD–TS80C51RA2
- POD–TS80C51RD2
- POD–TS80C51U2

The POD–31A is the base for advanced versions of the POD–31, including POD–C32HF–42. It is intended to be used for 40-pin DIP parts in external mode at high clock rates, and supports bank switching. It also has these features: Three LEDs, gated reset, support for the advanced trace board signals, negative reset polarity support, and support for 3-volt operation with a user-installed 3-volt MCU.

To work with this pod, you need to select the right processor from the **Select Processor** drop-down list box (**Step 3** of the **Communications** tab in the Emulator Configuration window).

LED Indicators

Function	Description
MON	A red LED that indicates when the system is in monitor mode. MON turns off during emulation mode (Run or Go) and turns back on when a break occurs and the system returns to monitor mode.
EMUL	A green LED that indicates when the system is in emulation mode. (It performs the inverse function of MON.)
RESET	A red LED that indicates the MCU is receiving a reset signal.

Jumper Identification and Description

Jumper Designation	Function	Description
XTAL	Crystal Selection	Jumper pins for selecting the source of the crystal or clock. • Jumpers installed in the upper two positions (INT position) select the crystal on the pod. • Jumpers installed in the lower two positions (EXT position) select the crystal on the target. **Note**: This jumper should be in the INT position when a target is not connected to the pod board. This is also the default position.

POD–C32HF–42

Jumper Identification and Description (continued)

Jumper Designation	Function	Description
POWER	Power Selection	• When jumpers are installed across the two lower pins (INT position), the power to the MCU comes from the emulator. • When jumpers are installed across the two upper pins (EXT position), the power to the MCU comes from the target. **Note**: This jumper should be in the INT position when a target is not connected to the pod board. This is also the default position.
RES		This jumper connects the target RESET pin to the emulator. If the target system has a watchdog, it might interfere with the emulator in monitor mode. If this occurs, remove RES.
EM/		The EM/ pin is an output pin that indicates the emulator's status. This pin is high when the emulator is in monitor mode, and low when the emulator is in emulation mode.
SY0, SY1, E0, E1		These pins are used to connect the external signals to the emulator. They are also used for the trace function. The SY0 pin is used in the breakpoint logic.
BS0, BS1, BS2, BS3		These pins are used for bank switching. • BS0 corresponds to the red wire. (Refer to the "Setting Up Bank Switching Connections" section in Chapter 2, "Installing the Hardware.") • BS1 corresponds to the green wire. • BS2 corresponds to the white wire. • BS0, BS1, and BS2 go through an inverter on the pod board. • BS3 (MSB) is normally used for RXD in the trace board. If BS3 is used for bank switching, you must remove the jumper from RXD to BS3. • If BS0 and BS1 are used for bank switching, the jumpers at B0/ and B1/ must be moved up so that the jumpers B0/ and B1/ are between the + sign and the B#/ pins. • Do not use the ENABLE SIG command if BS0 and BS1 are used.
FLF/, ANB/		These pins are used with the Advanced Trace Board (ATR) and Enhanced Trace Board (ETR). They can not be used with bank switching. Refer to the "Configuring the Enhanced Trace Board" section in Chapter 7, "Configuring Trace Boards."
SPEED		• This jumper needs to be in the FAST position when operating above 20 MHz. • This jumper needs to be in the SLOW position when operating at frequencies below 20 MHz.
RESPOL		This jumper needs to be in the upper + position. For MCUs with active low reset, it needs to be in the lower - position.

POD–C32HF–42

Jumper Identification and Description (continued)

Jumper Designation	Function	Description
PSEN/, RD/, WR/		• Signals PSEN/, RD/, and WR/ can be routed directly (non-gated) to the target system, or go through a gate, so that the target system is not accessed in monitor mode (gated). The board is delivered with the corresponding jumpers in the upper, gated position (GP/, GR/ and GW/). • The jumper for PSEN/ is at the left upper corner of the crystal Y1. The jumpers for RD/ and WR/ are to the left of the jumper block with PSEN/ and ALE. **CAUTION** Use caution if you connect the pod to a target when the jumpers PSEN/, RD/, or WR/ are in the non-gated position. The following can occur: • If a target device drives the data bus when the MCU is fetching code or reading XDATA memory mapped to the emulator, a bus contention can occur. This condition can cause the emulator to malfunction. • During XDATA writes, target memory is written to, even when accessing an address mapped to the emulator.
RES-FUNC		If the RES-FUNC jumper is in, reset from the target system is active only in emulation mode. If the jumper is out, a reset from the target system resets the emulator in monitor mode also.
WE, RE		These jumpers are factory set and should not be changed.
P1		• The right row of pins is connected to Port 1: the P1.7 pin at the top, and the P1.0 pin at the bottom. • The left row of pins is reflected on the trace display of P1. • The board is delivered with the jumpers set, so that P1 will be traced. • You can connect external signals or signals from other ports to the left row of pins if the jumpers are removed. • When a jumper is connected to the trace, an extra load ALS TTL load is added (74ALS258).
P3		• The right row of pins is connected to Port 3: the P3.5 pin at the top, and the P3.0 pin at the bottom. • The left row of pins is reflected on the trace display of P3. • The board is delivered with the jumpers set, so that P3 will be traced. • You can connect external signals or signals from other ports to the left row of pins if the jumpers are removed. • When a jumper is connected to the trace, an extra load ALS TTL load is added (74ALS258)..
S1	RESET	Resets the MCU and can be used instead of the target system reset.

POD–C32HF–42

Notes

- When a 3-volt MCU is installed, this pod is capable of working with targets that operate at 3 volts. The POWER jumper must be in the EXT position, when the pod is connected to a target system that runs 3 volts. Some current can flow from the emulator into the target systems power (pin 40). The pod can drive the P0 lines up to approximately 3.5 volts. The gate for PSEN/, RD/ and WR/ is powered from the same power as the processor. All other signals to the target driven from the pod come directly from the processor on the pod.

- SY0, SY1, E0, E1, BS0, BS1, BS2, BS3 have each a 100K-Ohm pull-up resistor.

- Signals PSEN/, RD/ and WR/ go through a 74ACT32 if they are gated. This adds approximately 5 ns to the timing of these signals.

Board Layout

Figure 102 shows locations of jumpers and Figure 103 shows a diagram of the jumper pins.

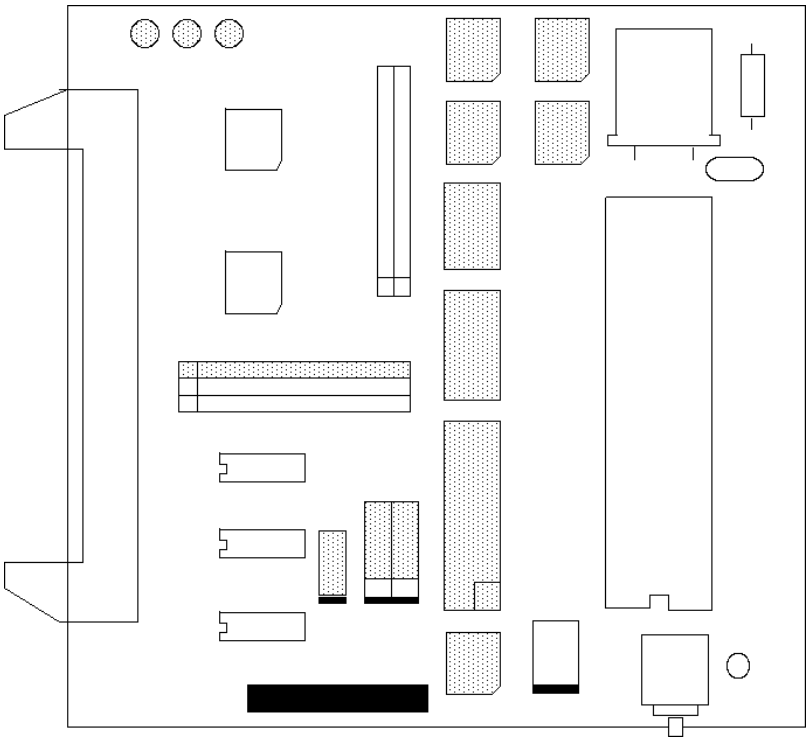


Figure 102. Locations of the Jumpers on POD–C32HF–42

POD–C32HF–42

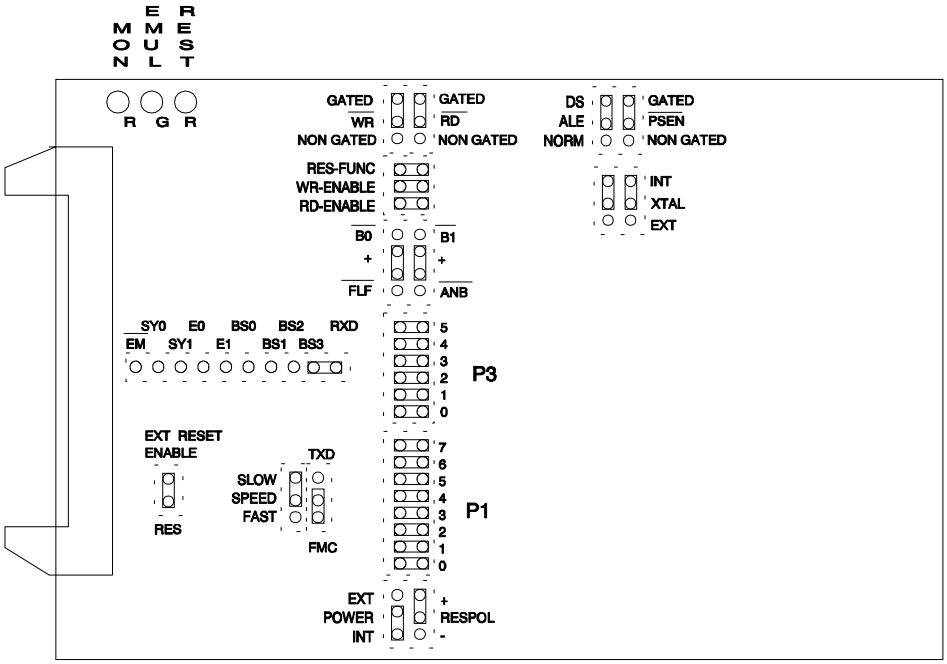


Figure 103. POD–C32HF–42 Jumper Diagram

POD–C51FX: Hooks Mode

Operating Instructions

To work with this pod, you need to select **POD–C51FX** from the **Select Processor** drop-down list box (**Step 3** of the **Communications** tab in the Emulator Configuration window).

The EPROM on the emulator board must be version COM 1.4.

The solder side of the board has a 40-pin DIP plug. The plug goes into the target system replacing the MCU IC.

If a 44-pin PLCC is used, an adapter can be purchased.

The pod board has an Intel 87C51FC chip using a special emulation mode. This emulation mode also uses a programmable gate array from XILINX (LCA).

LED Indicators

Function	Description
MON	A red LED that indicates when the system is in monitor mode. MON turns off during emulation mode (Run or Go) and turns back on when a break occurs and the system returns to monitor mode.
EMUL	A green LED that indicates when the system is in emulation mode. (It performs the inverse function of MON.)
PWD	A yellow LED that indicates when the MCU is in power-down mode.
IDLE	A yellow LED that indicates when the MCU is in idle mode.
RES	A red LED that indicates the MCU is receiving a reset signal.

POD–C51FX

Jumper Identification and Description

Jumper Designation	Function	Description
JP5		This jumper determines the mode of operation of P0, P2, P3.6 and P3.7. If only internal memory is to be emulated, both jumpers (M1 and M2) should be in. This enables P0, P2, P3.6 and P3.7 to be used as I/O ports. If the lower jumper, M1, is out, P0 has address and data and P2 has address data for external code fetches. PSEN is active. The emulator mapping determines whether code is read from the emulator or from the target system. If the upper jumper, M2, is out, P2 has address and P0 has address and data for MOVX instructions. P3.6 and P3.7 is active as RD and WR if XDATA is mapped to the target system. If only internal 8xC52 RAM is used and your emulator board has 128K RAM, the XDATA needs to be mapped to the emulator. M2 needs to be in. **Note**: The mode jumpers M1, M2 can be changed at any time but you have to issue the command RES CHIP so that the LCA can properly switch operational modes.
JP6 (PWR)	MCU Power	• Power should be supplied from the emulator. • The jumper should be across the two left pins (INT position). • Taking power from the target system, with the jumper across the two right pins (EXT position) is not recommended. • Power should be on the INT position, even if the pod is connected to a user target system. (This selection is the opposite of that for the bondout pods.) The MCU should be powered from the emulator since its operation is closely connected with the LCA. **Note**: This jumper should be in the INT position when a target is not connected to the pod board. This is also the default position.
JB3, JB4 (XTAL)	Crystal Selection	Jumper pins for selecting the source of the crystal or clock. • Jumpers installed in the right two positions (INT position) select the crystal on the pod. • Jumpers installed in the left two positions (EXT position) select the crystal on the target. • The oscillator is on the LCA, so do not drive the clock inputs when the emulator is powered off. • If you use an external clock, note that XTAL1 (pin 19) is input, and XTAL2 (pin 18) needs to be left open. **Note**: These jumpers should be in the INT position when a target is not connected to the pod board. This is also the default position.
JP7	RST	This jumper connects the target RESET pin to the emulator. If the target system has a watchdog, it might interfere with the emulator in monitor mode. If this occurs, remove JP7.

POD–C51FX

Jumper Identification and Description (continued)

Jumper Designation	Function	Description
JP2	SY0, SY1	This header connects external signals to the emulator. SY1 and SY0 are used for the trace function. SY0 (the right pin) can also be used in the breakpoint logic. The left pin is SY1. The silk screen might be wrong on early boards.
JP1	EM/FLF/ ANB	This header contains three pins: pin 1 (EM), pin 2 (FLF), and pin 3 (ANB). Pin 1 signals emulation or monitor mode. It is low when emulation is active (when the user program is running). Pins 2 and 3 are not used currently, but can be used in the future.
P1, P0		• The upper pins carry the MCU Port 1 pins: the P1.0 pin is on the left, and the P1.7 pin is on the right. Port 1 is connected directly to the MCU. • The lower pins carry Port 0: P0.0 is on the left, and P0.7 is on the right. Port 0 is emulated in the LCA which enables P0 to have better sink capability than the MCU. • P0 output signals also appear three clock cycles later than the real part. • The middle row of the jumper P1/P0 is normally connected to P1 using the jumpers. The signal level on the middle pins will be reflected on the trace display. • The board is delivered with the jumpers set, so that P1 will be traced. • You can connect external signals to the middle pins if the jumpers to P1 are removed. • When a jumper is connected to the trace, an extra load ALS TTL load is added (74ALS258).
P3, P2		• The upper pins carry the MCU Port 3 pins: P3.0 is on the left, and P3.7 is on the right. Port 3 is connected directly to the MCU except for P3.6 and P3.7. These pins go through a 74HC4066 which adds approximately 100 Ohms to their output impedance. • The lower pins carry Port 2: P2.0 is on the left, and P2.7 is on the right. Port 2 is emulated in the LCA which enables P2 to have better sink capability than the MCU. • P2 output signals also appear three clock cycles later than the real part. • The middle row of the jumper P3/P2 is normally connected to P3 using the jumpers. It can be used for P2 or other external signals the same way as P1/P0. • When a jumper is connected to the trace, an extra load ALS TTL load is added (74ALS258).
S1	Reset	Resets the MCU and can be used instead of the target system reset.

POD–C51FX

Notes

- Ports P0, and P2 are emulated, and have slightly different AC and DC characteristics.
  - Emulated I/O pins sink a minimum of 4 mA, and source (P2) minimum 4 mA for two clock cycles after an output low to high transition.
  - As high outputs or inputs, P2 ports have 22K-Ohm pull-up resistors.
- If your emulator board has only 32K RAM, normally, data (MAPX) needs to be mapped to target.
- P3.6 and P3.7 go through a 74HC4066 that adds about 100 Ohms to their output impedance.
- When accessing target memory, P2, P0, and ALE are delayed approximately 25 ns.
- The clock circuitry is emulated, but is very close to the MCU clock.
- The silk screen might be reversed for SY0 and SY1. SY1 is on the left.
- The timer/counters are stopped at breakpoints. This means the serial port also stops at breakpoints. If a character is received or sent at this moment it is distorted.
- If you break emulation and IDLE or PWD is lit, a number of timeout error messages show. You must issue the RESET CHIP command before you do anything else.
- If you want to change the MCU, you have to use an Intel 87C51FC. The 87C51FC has to have lock bit one programmed. Other bits do not affect the emulation mode.

POD–C51FX

Board Layout

Figure 104 shows locations of jumpers and Figure 105 shows the jumper pins in detail.

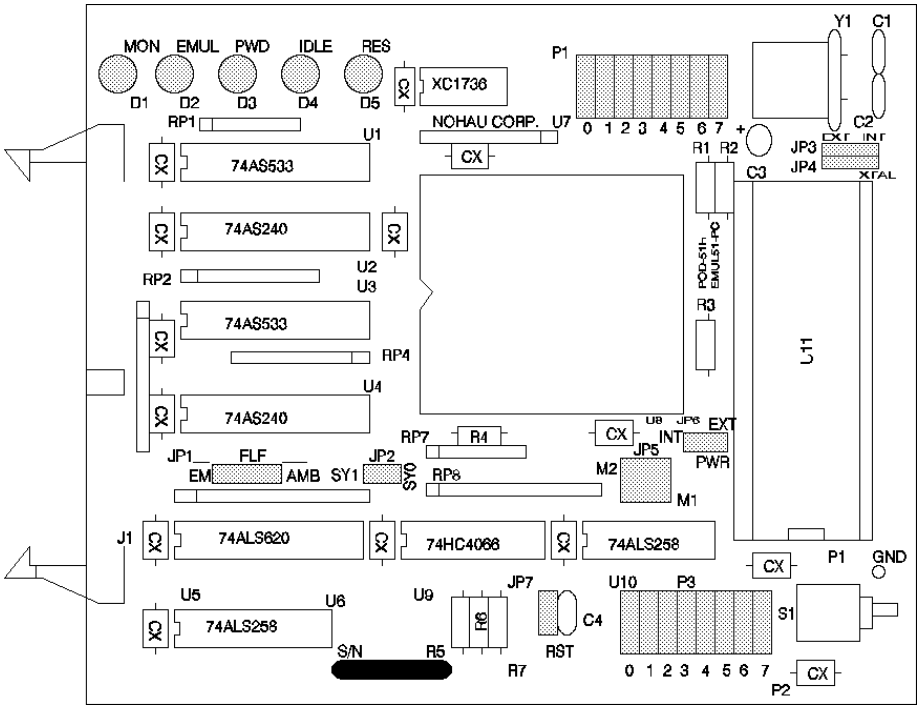


Figure 104. Locations of Jumpers on POD–C51FX

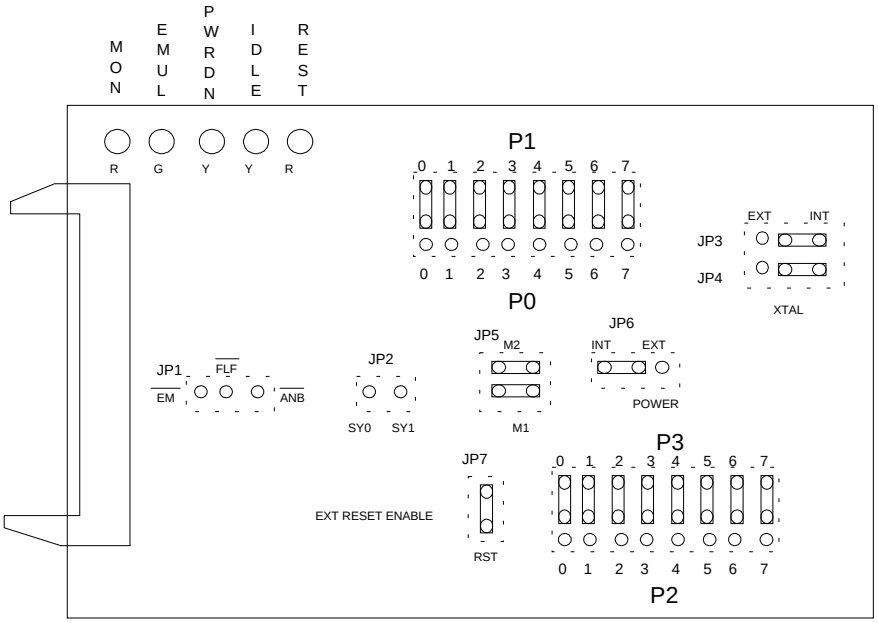


Figure 105. POD–C51FX Jumper Diagram

POD–C51MX

Overview

This section covers both the POD–C51MX and the EA256–51MX–BSW emulator board. This system is designed to emulate the Philips 8xC51MB/C enhanced 8051 microcontroller from Philips Semiconductors.

> **Note**
>
> The emulator board has been specially modified and will work **only** with the POD–C51MX.
>
> The EPROM on the emulator board must be version COM 1.6. This board is also labeled **51MX USE ONLY**.
>
> (Refer to Appendix J, "POD–C51MX Emulator Board Configurations.")

The solder side of the board has a 44-pin PGA plug. The plug goes into the target system through a 44-pin PGA to 44-pin PLCC or 44-pin QFP adapter. (You can order this plug separately.)

LED Indicators

Function	Description
MON	A red LED that indicates when the system is in Monitor mode. In Monitor mode, the processor is executing code that is internal to the emulator. This code is not user code, and is used to communicate with the host PC, to set up breakpoints, etc.
EMUL	A green LED that indicates when the system is in Emulation mode. In Emulation mode, the processor is executing the user's code from the emulation RAM or the targets PROM depending on the mapping in the emulator software.
PD	A yellow LED that indicates when the MCU is in power-down mode.
IDLE	A yellow LED that indicates when the MCU is in idle mode. If you break emulation during power-down or idle states the emulator will loose control of the emulation system and the emulation software will yield an error condition. RESET is the only way to regain control of the MCU and emulator again.
RES	A red LED that indicates the MCU's Reset pin is low (active state).

POD–C51MX

Jumper Identification and Description

Jumper	Function	Description
JP5	M1, M2, M3, M4	This jumper determines the mode(s) of operation. (Refer to Appendix J, "POD–C51MX Emulator Board Configurations.") **M1** • If this jumper is In, you can perform normal tracing. • If this jumper is Out, you can trace the special function register (SFR writes. • The default position is In. **Note**: 16 bits of either external probes, or 8 bits probe with A16-A23, or writes to SFRs. Currently the operation of the writes to SFRs not disassembled in human readable format you must hand decipher this information as it is displayed in binary form **M2** • If this jumper is In, you can trace the normal P1 and P3 header signals on the pod board. • If this jumper is Out, you can trace the A16 through the A23 address lines. • The default position is Out. **M3** • If this jumper is In, PSEN is not gated. • If this jumper is Out, PSEN is gated. • The default position is In. **M4** • If this jumper is In, Port 0 and Port 2 are used for general I/O. • If this jumper is Out, Port 0 and Port 2 are used for address/data bus. • The default position is In.
PWR	MCU Power Selection	This jumper selects the source of power for the MCU on the pod. • A jumper installed in the 5-V or 3-V position (depending on the type of MCU) selects power from the emulator. • A jumper installed in the EXT position (on the left tow pins) selects power from the target system.
XTAL	Crystal Selection	Jumper pins for selecting the source of the crystal or clock. • Jumpers installed in the top two positions (INT position) select the crystal on the pod. • Jumpers installed in the lower positions (EXT position) select the crystal on the target. • If you use an external clock, note that XTAL1 is input, and XTAL2 needs to be left open. **Note**: These jumpers should be in the INT position when a target is not connected to the pod board.
JP7	RST	This jumper connects the target RESET pin to the emulator. If the target system has a watchdog, it might interfere with the emulator in monitor mode. If this occurs, remove JP7.

POD–C51MX

Jumper Identification and Description (continued)

Jumper Designation	Function	Description
RST		This jumper connects the target Reset pin to the emulator. If the target system has a watchdog, it might interfere with the emulator in Monitor mode. If this occurs, remove RST.
JP2	SY0, SY1	This header contains two pins: pin 1 (SY0, center pin), and pin 2 (SY1, pin to the right away from the pod cable connector). These two signals are carried to the emulator and trace boards, and can be used in the trace and breakpoint logic. The pin on the left (closest to the pod cable connector) is the EM/ signal. This signal goes low during the Run mode and goes high when you enter monitor mode of the emulator.
EM/FLF/ ANB		The header labeled FLF/, ANB/ carries signals from the emulator. The ANB/ pin is used in conjunction with the enhanced trace boards as a signal output from a state machine logic.
P1, P0		• The upper pins carry the MCU Port 1 pins: the P1.0 pin is on the left, and the P1.7 pin is on the right. Port 1 is connected directly to the MCU. • The lower pins carry Port 0: P0.0 is on the left, and P0.7 is on the right. Port 0 is emulated in the LCA which enables P0 to have better sink capability than the MCU. • P0 output signals also appear three clock cycles later than the real part. • The middle row of the jumper P1/P0 is normally connected to P1 using the jumpers. The signal level on the middle pins will be reflected on the trace display. • The board is delivered with the jumpers set, so that P1 will be traced. • You can connect external signals to the middle pins if the jumpers are removed. The input load is a 100K pull-up.
P3, P2		• The upper pins carry the MCU Port 3 pins: P3.0 is on the left, and P3.7 is on the right. Port 3 is connected directly to the MCU except for P3.6 and P3.7. These pins go through a 74HC4066 which adds approximately 100 Ohms of series resistance. • The lower pins carry Port 2: P2.0 is on the left, and P2.7 is on the right. Port 2 is emulated in the LCA which enables P2 to have better sink capability than the MCU. • P2 output signals also appear three clock cycles later than the real part. • The middle row of the jumper P3/P2 is normally connected to P3 using the jumpers. The signal level on the middle pins will be reflected on the trace display. • The board is delivered with the jumpers setso that P3 will be traced. • You can connect external signals to the middle pins if the jumpers are removed. • The input load is one ALS input 74ALS258).

POD–C51MX

Jumper Identification and Description (continued)

Jumper Designation	Function	Description
S1	Reset	Resets the MCU and can be used instead of the target system reset.
JP1		Connects the upper address lines, A16-A23, to the emulator boards bank switching memory logic. There are only four input banking lines that can be used, they are labeled on the J1 connectors top row as 0, 1, 2, and 3. The bottom row is the address lines A16 through A23. The B0/ and B1/ jumpers are used for the connection of the A16 and A17 lines to the emulators bank logic.

Notes

- Ports 0 and 2 are emulated, and have slightly different AC and DC characteristics:
  - Emulated I/O pins sink a minimum of 4 mA, and source (P2) minimum 4 mA for two clock cycles after an output low to high transition.
  - As high outputs or inputs, P2 ports have 22K-Ohm pull-up resistors.
  - Emulated output will change state three clock cycles later than a normal output would.
- P3.6 and P3.7 go through a high speed switch logic that adds approximately. 100 Ohms to their output impedance.
- When accessing target memory, P0, and P2 will be delayed approximately 5 to 10 ns.
- The clock circuitry is emulated, but is very close to the processor clock.
- Because the POC–C51MX is a six-clock per instruction micro, you need to enter two times the external frequency in the **Clock (MHz)** field in the **Hdw Config** tab.
- The timer/counters are stopped at breakpoints. This usually means that the serial port also stops at breakpoints. If a character is received or sent at this moment it will be distorted.
- If you break emulation during the time that the PD (power-down) or IDLE LEDs are lit, a number of communication errors will cause the software to loose control of the emulation system. You will need to restart the system at this point.

See Appendix J, “POD–C51MX Emulator Board Configuration.”

How Nohau Supports the Philips 8051MX Microcontroller

There are two options for the emulation of the 8xC51MB2 and 8xC51MC2 MCUs:

- An emulator with 768K of emulation memory that has been configured to support up to 512K of code and 256K of external data memory.
- A 256K emulator which is configured for 128K for code memory and 128K for external data memory.

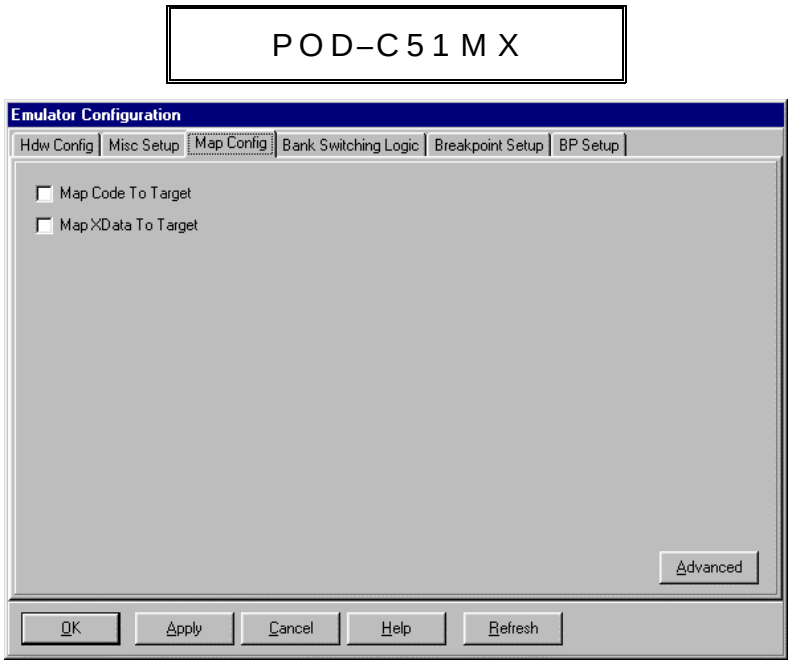


Figure 106. Regular Map Config Tab for the C51MX

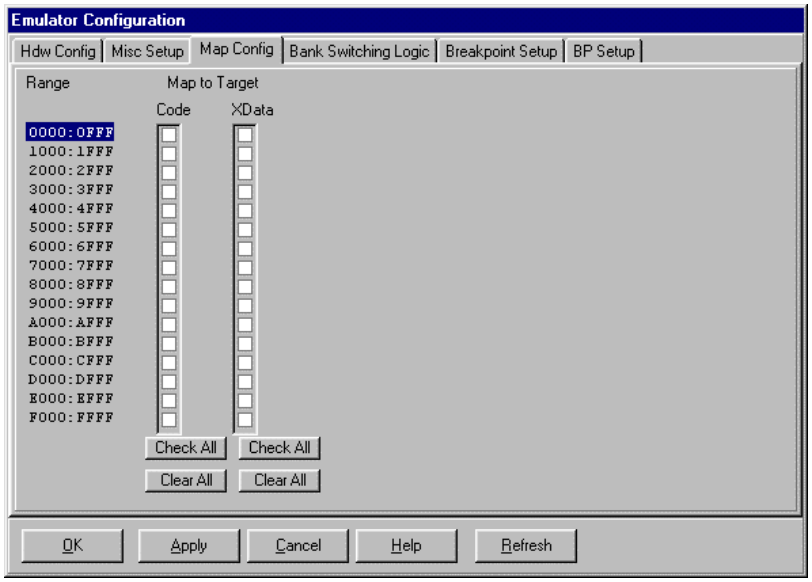


Figure 107. Map Config Tab for the Advanced Setup of the C51MX

In most cases you can typically map all of the external data to the target and leave the code memory mapped to emulation RAM to emulate the on chip / off-chip ROM memory. The current software is preliminary for supporting the 51MX family. Nohau has changed the **Map Config** tab for the MX family to have the following options. This will be the normal interface displayed (Figure 106). If you use the advanced setup and then re-enter the mapping configuration then the **Map Config** tab will display as in Figure 107. (Refer to Appendix J, “POD–C51MX Emulator Board Configurations.”)

POD–C51MX

Memory Block Layout Table

Code Memory block mapping layout	256k emulator board boundary ends here			768K emulator board boundary ends here		
80:0000 - 80:0FFF	81:0000 - 81:0FFF	82:0000 - 82:0FFF	83:0000 - 83:0FFF	84:0000 - 84:0FFF	85:0000 - 85:0FFF	86:0000 - 86:0FFF	87:0000 - 87:0FFF
80:1000 - 80:1FFF	81:1000 - 81:1FFF	82:1000 - 82:1FFF	83:1000 - 83:1FFF	84:1000 - 84:1FFF	85:1000 - 85:1FFF	86:1000 - 86:1FFF	87:1000 - 87:1FFF
80:2000 - 80:2FFF	81:2000 - 81:2FFF	82:2000 - 82:2FFF	83:2000 - 83:2FFF	84:2000 - 84:2FFF	85:2000 - 85:2FFF	86:2000 - 86:2FFF	87:2000 - 87:2FFF
80:3000 - 80:3FFF	81:3000 - 81:3FFF	82:3000 - 82:3FFF	83:3000 - 83:3FFF	84:3000 - 84:3FFF	85:3000 - 85:3FFF	86:3000 - 86:3FFF	87:3000 - 87:3FFF
80:4000 - 80:4FFF	81:4000 - 81:4FFF	82:4000 - 82:4FFF	83:4000 - 83:4FFF	84:4000 - 84:4FFF	85:4000 - 85:4FFF	86:4000 - 86:4FFF	87:4000 - 87:4FFF
80:5000 - 80:5FFF	81:5000 - 81:5FFF	82:5000 - 82:5FFF	83:5000 - 83:5FFF	84:5000 - 84:5FFF	85:5000 - 85:5FFF	86:5000 - 86:5FFF	87:5000 - 87:5FFF
80:6000 - 80:6FFF	81:6000 - 81:6FFF	82:6000 - 82:6FFF	83:6000 - 83:6FFF	84:6000 - 84:6FFF	85:6000 - 85:6FFF	86:6000 - 86:6FFF	87:6000 - 87:6FFF
80:7000 - 80:7FFF	81:7000 - 81:7FFF	82:7000 - 82:7FFF	83:7000 - 83:7FFF	84:7000 - 84:7FFF	85:7000 - 85:7FFF	86:7000 - 86:7FFF	87:7000 - 87:7FFF
80:8000 - 80:8FFF	81:8000 - 81:8FFF	82:8000 - 82:8FFF	83:8000 - 83:8FFF	84:8000 - 84:8FFF	85:8000 - 85:8FFF	86:8000 - 86:8FFF	87:8000 - 87:8FFF
80:9000 - 80:9FFF	81:9000 - 81:9FFF	82:9000 - 82:9FFF	83:9000 - 83:9FFF	84:9000 - 84:9FFF	85:9000 - 85:9FFF	86:9000 - 86:9FFF	87:9000 - 87:9FFF
80:A000 - 80:AFFF	81:A000 - 81:AFFF	82:A000 - 82:AFFF	83:A000 - 83:AFFF	84:A000 - 84:AFFF	85:A000 - 85:AFFF	86:A000 - 86:AFFF	87:A000 - 87:AFFF
80:B000 - 80:BFFF	81:B000 - 81:BFFF	82:B000 - 82:BFFF	83:B000 - 83:BFFF	84:B000 - 84:BFFF	85:B000 - 85:BFFF	86:B000 - 86:BFFF	87:B000 - 87:BFFF
80:C000 - 80:CFFF	81:C000 - 81:CFFF	82:C000 - 82:CFFF	83:C000 - 83:CFFF	84:C000 - 84:CFFF	85:C000 - 85:CFFF	86:C000 - 86:CFFF	87:C000 - 87:CFFF
80:D000 - 80:DFFF	81:D000 - 81:DFFF	82:D000 - 82:DFFF	83:D000 - 83:DFFF	84:D000 - 84:DFFF	85:D000 - 85:DFFF	86:D000 - 86:DFFF	87:D000 - 87:DFFF
80:E000 - 80:EFFF	81:E000 - 81:EFFF	82:E000 - 82:EFFF	83:E000 - 83:EFFF	84:E000 - 84:EFFF	85:E000 - 85:EFFF	86:E000 - 86:EFFF	87:E000 - 87:EFFF
80:F000 - 80:FFFF	81:F000 - 81:FFFF	82:F000 - 82:FFFF	83:F000 - 83:FFFF	84:F000 - 84:FFFF	85:F000 - 85:FFFF	86:F000 - 86:FFFF	87:F000 - 87:FFFF

XDATA Memory block mapping layout	256k emulator boundary ends here.		768k emulator boundary ends here
00:0000 - 00:0FFF	01:0000 - 01:0FFF	02:0000 - 02:0FFF	03:0000 - 03:0FFF
00:1000 - 00:1FFF	01:1000 - 01:1FFF	02:1000 - 02:1FFF	03:1000 - 03:1FFF
00:2000 - 00:2FFF	01:2000 - 01:2FFF	02:2000 - 02:2FFF	03:2000 - 03:2FFF
00:3000 - 00:3FFF	01:3000 - 01:3FFF	02:3000 - 02:3FFF	03:3000 - 03:3FFF
00:4000 - 00:4FFF	01:4000 - 01:4FFF	02:4000 - 02:4FFF	03:4000 - 03:4FFF
00:5000 - 00:5FFF	01:5000 - 01:5FFF	02:5000 - 02:5FFF	03:5000 - 03:5FFF
00:6000 - 00:6FFF	01:6000 - 01:6FFF	02:6000 - 02:6FFF	03:6000 - 03:6FFF
00:7000 - 00:7FFF	01:7000 - 01:7FFF	02:7000 - 02:7FFF	03:7000 - 03:7FFF
00:8000 - 00:8FFF	01:8000 - 01:8FFF	02:8000 - 02:8FFF	03:8000 - 03:8FFF
00:9000 - 00:9FFF	01:9000 - 01:9FFF	02:9000 - 02:9FFF	03:9000 - 03:9FFF
00:A000 - 00:AFFF	01:A000 - 01:AFFF	02:A000 - 02:AFFF	03:A000 - 03:AFFF
00:B000 - 00:BFFF	01:B000 - 01:BFFF	02:B000 - 02:BFFF	03:B000 - 03:BFFF
00:C000 - 00:CFFF	01:C000 - 01:CFFF	02:C000 - 02:CFFF	03:C000 - 03:CFFF
00:D000 - 00:DFFF	01:D000 - 01:DFFF	02:D000 - 02:DFFF	03:D000 - 03:DFFF
00:E000 - 00:EFFF	01:E000 - 01:EFFF	02:E000 - 02:EFFF	03:E000 - 03:EFFF
00:F000 - 00:FFFF	01:F000 - 01:FFFF	02:F000 - 02:FFFF	03:F000 - 03:FFFF

POD-C51MX

(This page intentionally left blank)

POD–C51MX

With the XDTATA all mapped to target the on-chip XDATA memory will be visible to you and also to the emulator's trace memory. The trace will read the correct data regardless of the mapping of the XDATA for the on-chip XDATA memory.

In the advanced setup the advanced **Map Config** tab will appear (Figure 107). This is also the current **Mag Config** tab for the EMUL51–PC emulator family, and we will be documenting how the operation of this works in both the manuals and the on-line help. Refer to the following "Code Memory" and XDATA Memory" tables for information on memory mapping with the advanced setup option and how it reflects on the available memory.

The memory mapping controls operate on a basic 64K address range. Each of the 4K mapping regions will mirror in the same address range for all of the pages. Mapping can be set for either the emulator's memory or the target system's memory.

Data RAM

(This memory is completely handled by the emulation MCU without interference from the emulator.)

The P87C51MB2 and P87C51MC2 have 2K and 3K of on-chip RAM respectively. Refer to Philips documentation for usages of the different data segments under the 51MX Architecture Reference section.

On-Chip Data Memory Usage

Data Memory		Size (in Bytes)	
Type	**Description**	**P87C51MB2**	**P87C51MC2**
DATA	Memory that can be addressed directly and indirectly.	128	128
IDATA	Memory that can be addressed indirectly (where direct address is for SFRs only).	128	128
EDATA	Memory that can only be addressed indirectly.	1024	1024
XDATA	Memory (on-chip external data) that is accessed using the MOVX instructions.	768	1792
	Total Memory	2048	3072

Extended SFRs in Seehau

Nohau has incorporated a flag in the symbol table for all special function registers that are accessed through the A5H extended operation. This is by telling the software that the SFR's address is 0xFF + the normal address.

POD-C51MX

This means that the following SFRs will appear with these in Seehau and the BIT symbols for each of these are tagged as BIT and not EBIT:

P4	S1CON	S1BUF	S1ADDR	S1ADEN	S1STAT
BRGCON	BRGR0	BRGR1	S0STAT	WDCON	SPE
EPL	EPM	EPH	MXCON		

Shadow RAM

Nohau currently supports one 64K page range of Shadow RAM that will see writes to the external data area. There is an unimplemented feature that would give the ability to have a second 64K page based on a specified extended address line giving the total shadow memory to be 128K.

Currently if a write to XDATA at address 00:0000h and then a write to XDATA to address 04:0000h, this would result in the same cell in the Data window to be displayed.

Memory / Code Coverage

Nohau's current setup is to have the ability to cover 1 MB of memory range. The hardware and software work together to set the mode, cycle type, for the coverage analysis. The available modes are the shown in the following table.

Code Coverage Operational Modes

Mode	Description
Code Access	Any access to code memory be it a fetch, pre-fetch or read.
Opcode Fetch	First true fetch of the Opcode byte for an instruction.
XDATA Write	Any write cycle to the XDATA memory space.
XDATA Read	Any read cycle to the XDATA memory space

Refer to Appendix L "Code Coverage" for general information.

POD–C51MX

Board Layout

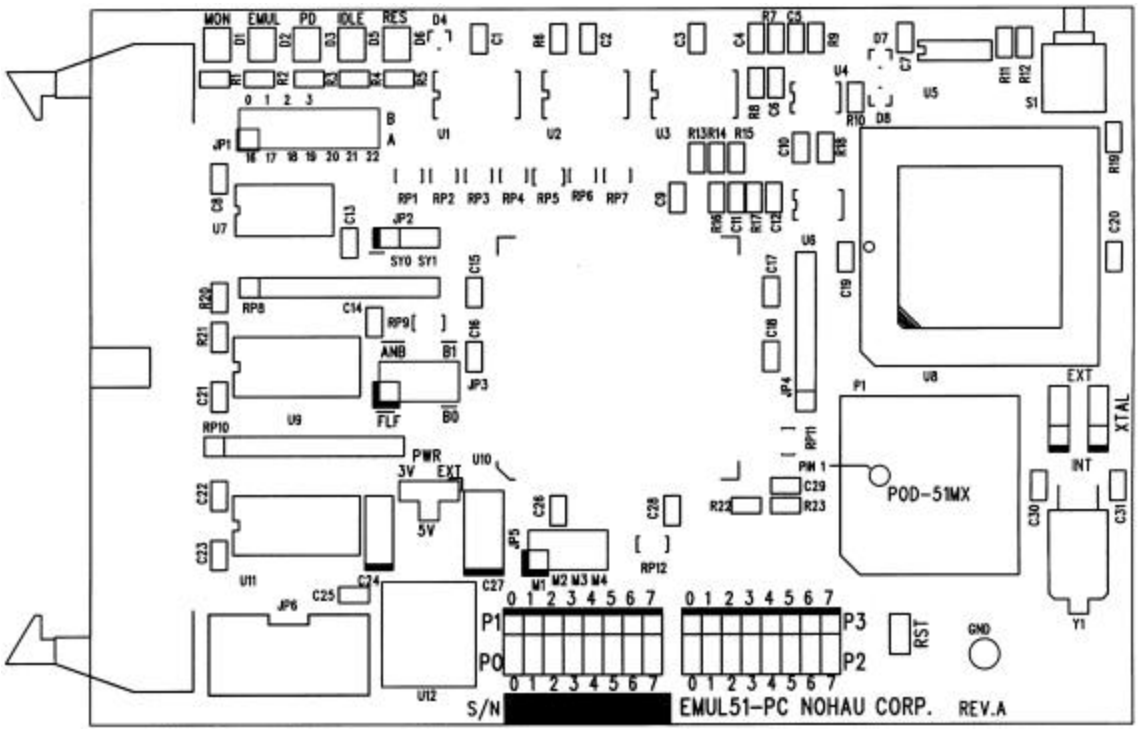


Figure 108. Jumpers and Headers for the POD–C51MX

POD–C52: Hooks Mode

Operating Instructions

This pod supports the following MCUs:

- 8xC51/52/54/58
- 8xC51/52
- 80C31/32
- 8031/32

To work with this pod, you need to select **POD–C52** from the **Select Processor** drop-down list box (**Step 3** of the **Communications** tab in the Emulator Configuration window).

The EPROM on the emulator board must be version COM 1.4.

The solder side of the board has a 40-pin DIP plug. The plug goes into the target system replacing the MCU IC.

If a 44-pin PLCC is used, an adapter can be purchased.

The pod board has a Philips 87C52 chip using a special emulation mode. This emulation mode also uses a programmable gate array from XILINX (LCA).

LED Indicators

Function	Description
MON	A red LED that indicates when the system is in monitor mode. MON turns off during emulation mode (Run or Go) and turns back on when a break occurs and the system returns to monitor mode.
EMUL	A green LED that indicates when the system is in emulation mode. (It performs the inverse function of MON.)
PWD	A yellow LED that indicates when the MCU is in power-down mode.
IDLE	A yellow LED that indicates when the MCU is in idle mode.
RES	A red LED that indicates the MCU is receiving a reset signal.

POD–C52

Jumper Identification and Description

Jumper Designation	Function	Description
JP5		This jumper determines the mode of operation of P0, P2, P3.6 and P3.7. If only internal memory is to be emulated, both jumpers (M1 and M2) should be in. This enables P0, P2, P3.6 and P3.7 to be used as I/O ports. If the lower jumper, M1, is out, P0 has address and data and P2 has address data for external code fetches. PSEN is active. The emulator mapping determines whether code is read from the emulator or from the target system. If the upper jumper, M2, is out, P2 has address and P0 has address and data for MOVX instructions. P3.6 and P3.7 is active as RD and WR if XDATA is mapped to the target system. If only internal 8xC52 RAM is used and your emulator board has 128K RAM, the XDATA needs to be mapped to the emulator. M2 needs to be in. **Note**: The mode jumpers M1, M2 can be changed at any time but you have to issue the command RES CHIP so that the LCA can properly switch operational modes.
JP6 (PWR)	MCU Power	• Power should be supplied from the emulator. • The jumper should be across the two left pins (INT position). • Taking power from the target system, with the jumper across the two right pins (EXT position) is not recommended. • Power should be on the INT position, even if the pod is connected to a user target system. (This selection is the opposite of that for the bondout pods.) The MCU should be powered from the emulator since its operation is closely connected with the LCA. **Note**: This jumper should be in the INT position when a target is not connected to the pod board. This is also the default position.
JB3, JB4 (XTAL)	Crystal Selection	Jumper pins for selecting the source of the crystal or clock. • Jumpers installed in the right two positions (INT position) select the crystal on the pod. • Jumpers installed in the left two positions (EXT position) select the crystal on the target. • The oscillator is on the LCA, so do not drive the clock inputs when the emulator is powered off. • If you use an external clock, note that XTAL1 (pin 19) is input, and XTAL2 (pin 18) needs to be left open. **Note**: These jumpers should be in the INT position when a target is not connected to the pod board. This is also the default position.
JP7	RST	This jumper connects the target RESET pin to the emulator. If the target system has a watchdog, it might interfere with the emulator in monitor mode. If this occurs, remove JP7.

POD–C52

Jumper Identification and Description (continued)

Jumper Designation	Function	Description
JP2	SY0, SY1	This header connects external signals to the emulator. SY1 and SY0 are used for the trace function. SY0 (the right pin) can also be used in the breakpoint logic. The left pin is SY1. The silk screen might be wrong on early boards.
JP1	EM/FLF/ ANB	This header contains three pins: pin 1 (EM), pin 2 (FLF), and pin 3 (ANB). Pin 1 signals emulation or monitor mode. It is low when emulation is active (when the user program is running). Pins 2 and 3 are not used currently, but can be used in the future.
P1, P0		• The upper pins carry the MCU Port 1 pins: the P1.0 pin is on the left, and the P1.7 pin is on the right. Port 1 is connected directly to the MCU. • The lower pins carry Port 0: P0.0 is on the left, and P0.7 is on the right. Port 0 is emulated in the LCA which enables P0 to have better sink capability than the MCU. • P0 output signals also appear three clock cycles later than the real part. • The middle row of the jumper P1/P0 is normally connected to P1 using the jumpers. The signal level on the middle pins will be reflected on the trace display. • The board is delivered with the jumpers set, so that P1 will be traced. • You can connect external signals to the middle pins if the jumpers to P1 are removed. • When a jumper is connected to the trace, an extra load ALS TTL load is added (74ALS258).
P3, P2		• The upper pins carry the MCU Port 3 pins: P3.0 is on the left, and P3.7 is on the right. Port 3 is connected directly to the MCU except for P3.6 and P3.7. These pins go through a 74HC4066 which adds approximately 100 Ohms to their output impedance. • The lower pins carry Port 2: P2.0 is on the left, and P2.7 is on the right. Port 2 is emulated in the LCA which enables P2 to have better sink capability than the MCU. • P2 output signals also appear three clock cycles later than the real part. • The middle row of the jumper P3/P2 is normally connected to P3 using the jumpers. It can be used for P2 or other external signals the same way as P1/P0. • When a jumper is connected to the trace, an extra load ALS TTL load is added (74ALS258).
S1	Reset	Resets the MCU and can be used instead of the target system reset.

POD–C52

Notes

- Ports P0, and P2 are emulated, and have slightly different AC and DC characteristics.
  - Emulated I/O pins sink a minimum of 4 mA, and source (P2) minimum 4 mA for two clock cycles after an output low to high transition.
  - As high outputs or inputs, P2 ports have 22K-Ohm pull-up resistors.
- If your emulator board has only 32K RAM, normally, data (MAPX) needs to be mapped to target.
- P3.6 and P3.7 go through a 74HC4066 that adds about 100 Ohms to their output impedance.
- When accessing target memory, P2, P0, and ALE are delayed approximately 25 ns.
- The clock circuitry is emulated, but is very close to the MCU clock.
- The silk screen might be reversed for SY0 and SY1. SY1 is on the left.
- The timer/counters are stopped at breakpoints. This means the serial port also stops at breakpoints. If a character is received or sent at this moment it is distorted.
- If you break emulation and IDLE or PWD is lit, a number of timeout error messages show. You must issue the RESET CHIP command before you do anything else.
- If you want to change the MCU, you have to use a Philips 87C52. The 87C52 has to have lock bit one programmed. Other bits do not affect the emulation mode.

POD–C52

Board Layout

Figure 109 shows locations of jumpers and Figure 110 shows the jumper pins in detail.

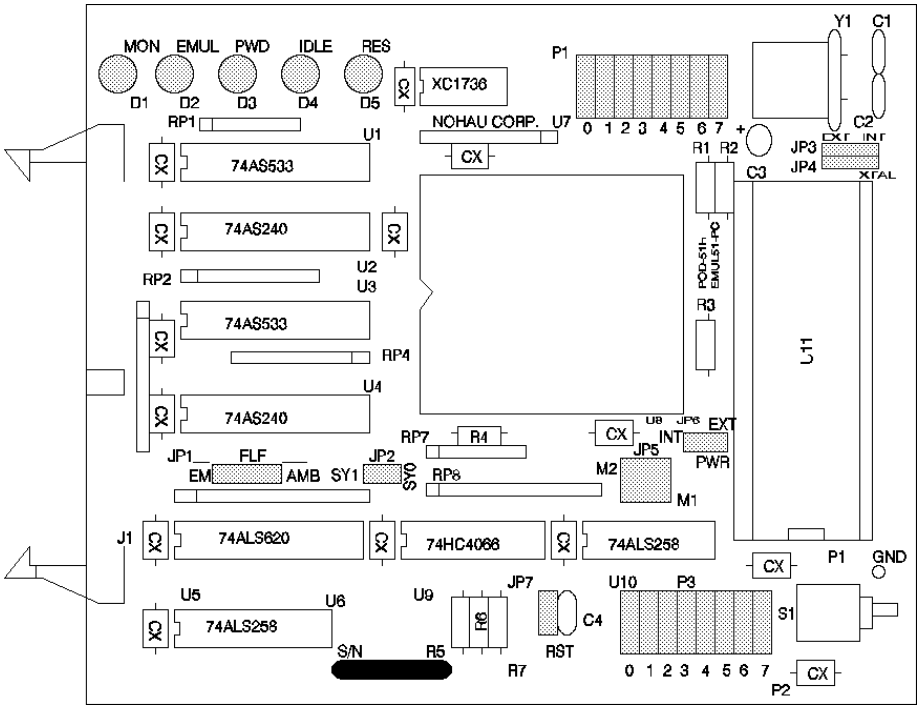


Figure 109. Locations of Jumpers on POD–C52

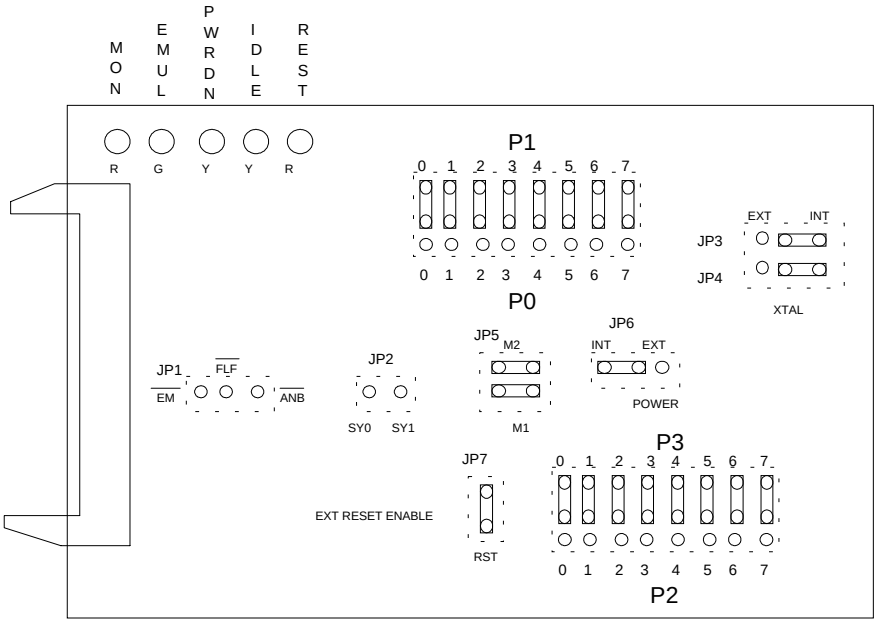


Figure 110. POD–C52 Jumper Diagram

POD–C520–PGA–33: External Mode

Operating Instructions

The POD–C530/C520 can be used to emulate the 8xC520/530 MCUs, and the 8xC320/310 MCUs. Use the following adapters for emulation:

DS520 EDI/52PG/40D–530/520-S (for 40-pin DIP), or EDI/52PG/44PL–530/520 (for 44-pin PLCC), EDI/52PG/44QFS31-SD-530/520 (44-pin QFP)

DS530 EMUL51–PC/PGA52–PLCC52

To work with this pod, you need to select **POD–C520/530** from the **Select Processor** drop-down list box (**Step 3** of the **Communications** tab in the Emulator Configuration window).

The EPROM on the emulator board must be version COM 1.4. This EPROM is a special version for the POD–C530/520 and does not work well with the other Nohau pods. This pod requires the use of the EA256–C530.

The pod board has a special Dallas Semiconductor bondout chip to emulator the MCUs.

Note

If emulating the 310 MCU, see Appendix F, "Using the POD–C520–PGA–33 to Emulate the DS80C310 MCU."

LED Indicators

Function	Description
MON	A red LED that indicates when the system is in monitor mode. MON turns off during emulation mode (Run or Go) and turns back on when a break occurs and the system returns to monitor mode.
EMUL	A green LED that indicates when the system is in emulation mode. (It performs the inverse function of MON.)
RESET	A red LED that indicates the MCU is receiving a reset signal.

Jumper Identification and Description

Jumper Designation	Function	Description
XTAL	Crystal Selection	Jumper pins for selecting the source of the crystal or clock. • Jumpers installed in the side closest to the ribbon connector (INT position) select the crystal on the pod. • Jumpers installed toward the MCU (EXT position) select the crystal on the target. **Note**: These jumpers should be in the INT position when a target is not connected to the pod board. This is also the default position.

POD–C520–PGA–33

Jumper Identification and Description (continued)

Jumper Designation	Function	Description
POWER	Power Selection	Jumper pins for selecting the source of the power. • When jumpers are installed across the left two pins (INT), the power to the MCU comes from the emulator. • When jumpers are installed across the two right pins (EXT), the power to the MCU comes from the target. **Note**: This jumper should be in the INT position when a target is not connected to the pod board. This is also the default position.
RES FUN		Jumper RES FUN is used to determine if the MCUs reset input is to be (G) gated, controlled by emulator board, or (NG) non-gated, not controlled by emulator board. The gated position causes a small delay on the reset line before the MCU receives the signal.
RES	RST	This jumper connects the target RESET pin to the emulator. If the target system has a watchdog, it might interfere with the emulator in monitor mode. If this occurs, remove RES.
JP2	ALE	This jumper is for the ALE signal. • With the jumper in the right position, the two pins away from the ribbon connector, the ALE is gated. • If the jumper is in the opposite position the ALE signal comes directly from the bondout MCU. • The normal position for this jumper is in the gated position (to the right).
JP3, JP4		These jumpers connect pod Vcc and ground, respectively, to the on-chip real-time clock. • The real-time clock XTAL pins are connected to a 32-kHz crystal on the pod. • The jumper JP1 (labeled VXXG), connects to the target pins for the real-time clock Vcc (V), RTXC1&2 (X,X), and ground (G). • If you want to supply Vcc and ground for the real-time clock from the target, remove jumpers on JP3 and JP4. Then connect pin V of JP1 to the pin of JP3 closest to the MCU and pin G of JP1 to the pin of JP4 closest to the MCU.
SY0, SY1		This header connects external signals to the emulator. SY1 and SY0 are used for the trace function. SY0 (the upper pin) can also be used in the breakpoint logic.
P2, P3		• The upper pins carry the MCU Port 3 pins: the P3.0 pin is on the left, and the P3.7 pin is on the right. • The lower pins carry Port 2: P2.0 is on the left, and P2.7 is on the right. • The middle row of pins will be reflected on the trace display. • The board is delivered with the jumpers set, so that P3 will be traced. • If you move the jumpers to the bottom row, P2 will be traced. • You can connect external signals to the middle pins if the jumpers are removed. • When a jumper is connected to the trace, an extra load ALS TTL load is added (74ALS258).

POD–C520–PGA–33

Jumper Identification and Description (continued)

Jumper Designation	Function	Description
P0, P1		• The top row of pins carry the MCU Port 1 pins: the P1.0 pin is on the left, and the P1.7 pin is on the right. • The bottom row of pins carry Port 0: P0.0 is on the left, and P0.7 is on the right. • The middle row of pins will be reflected on the trace display of Port 1. • The board is delivered with the jumpers set so that P1 will be traced. • If you move the jumpers to the bottom row, P0 will be traced. • You can connect external signals to the middle pins if the jumpers are removed. • The input load is one ALS input (74ALS258).
EM/, FLF/, ANB/		The jumper pins next to U2 carry signals from the emulator and trace boards. • The EM/ (top pin) is high when the emulator is in monitor mode, and low if in emulation mode. • The FLF/ (middle pin) and ANB/ (bottom pin) are used with the Advance Trace Board (ATR).
S1	Reset	Resets the MCU and can be used instead of the target system reset.

Board Layout

Figure 111 shows locations of jumpers and Figure 112 shows the jumper pins in detail.

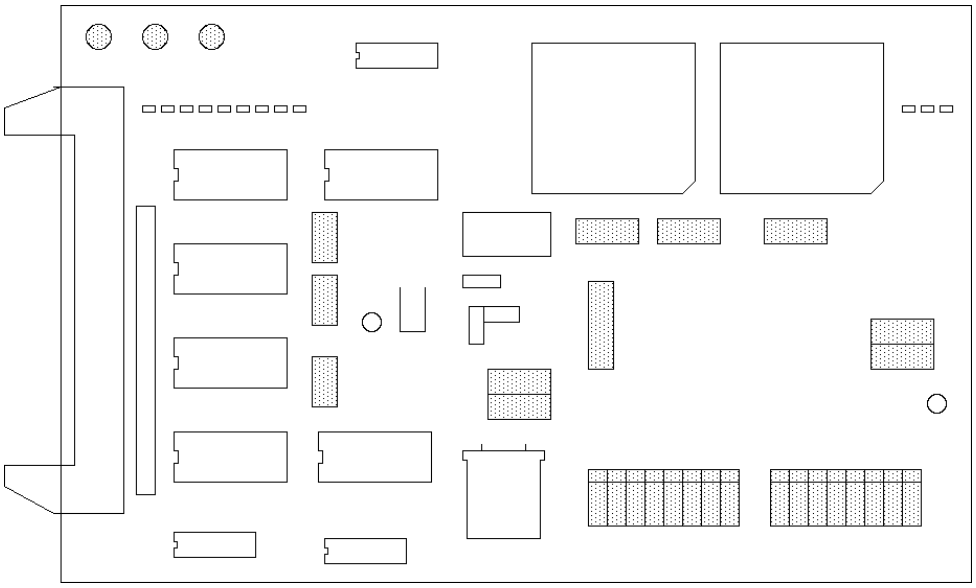


Figure 111. Locations of Jumpers on POD–C520–PGA-33

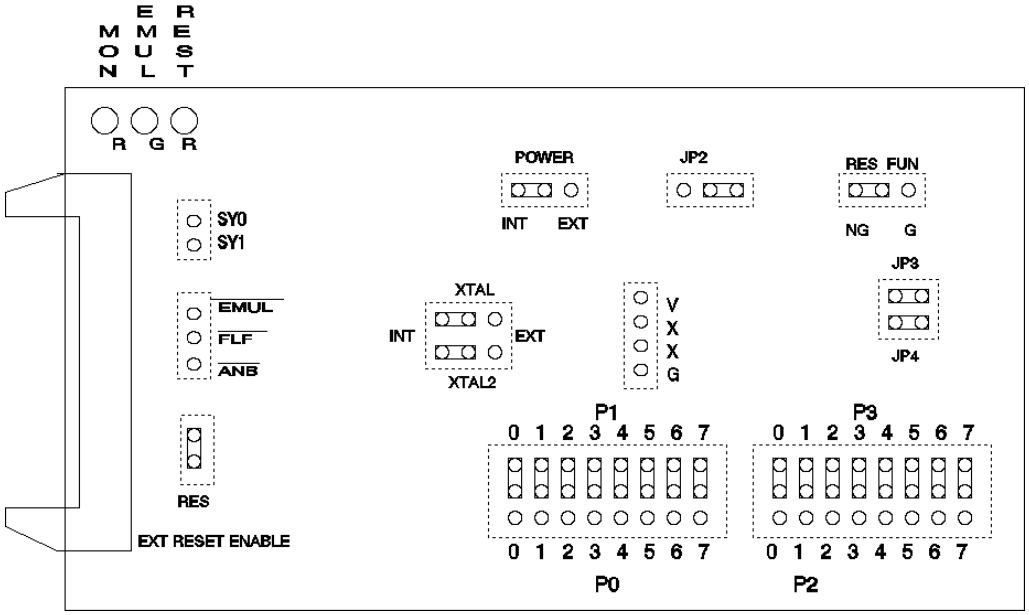


Figure 112. POD–C520–PGA-33 Jumper Diagram

POD–C528: Hooks Mode

Operating Instructions

The POD–528 is used to emulate Philips MCUs 83C528.

To work with this pod, you need to select **POD–C528** from the **Select Processor** drop-down list box (**Step 3** of the **Communications** tab in the Emulator Configuration window).

The EPROM on the emulator board must be version COM 1.4.

The solder side of the board has a 40-pin DIP plug. The plug goes into the target system replacing the MCU IC.

If a 44-pin PLCC is used, you can purchase an adapter (EMUL51–PC/DIP40/PLCC44).

The pod board has a Philips 83C528 chip using a special emulation mode. This emulation mode also uses a programmable gate array from XILINX (LCA).

LED Indicators

Function	Description
MON	A red LED that indicates when the system is in monitor mode. MON turns off during emulation mode (Run or Go) and turns back on when a break occurs and the system returns to monitor mode.
EMUL	A green LED that indicates when the system is in emulation mode. (It performs the inverse function of MON.)
PWD	A yellow LED that indicates when the MCU is in power-down mode.
IDLE	A yellow LED that indicates when the MCU is in idle mode.
RES	A red LED that indicates the MCU is receiving a reset signal.

POD–C528

Jumper Identification and Description

Jumper Designation	Function	Description
JP5	M1, M2	These jumpers determine the mode of operation of P0, P2, P3.6 and P3.7. If only internal memory is to be emulated, both jumpers (M1 and M2) should be in. This enables P0, P2, P3.6 and P3.7 to be used as I/O ports. **M1 Jumper Removed** ● If this jumper is out, P0 has the lower address and data. P2 has the upper address for external code fetches. ● PSEN is active. ● The emulator mapping determines whether the code is read from emulation memory or target memory. **M2 Jumper Removed** ● If this jumper is out, P0 has the lower address and data. P2 has the upper address for MOVX instructions. ● P3.6 and P3.7 is used for the RD and WR strobes. ● The mapping determines if the RD and WR strobes go to emulation memory or target memory. ● If only internal RAM is used, the emulator board has 128K RAM, map data MAPX to target, and the M2 jumper should be in. **Note**: You can change the mode jumpers M1and M2 at any time; however, you must issue a reset chip operation for the LCA and MCU to properly change modes.
JP6 (PWR)	MCU Power	● Power should be supplied from the emulator. ● The jumper should be across the two left pins (INT position). ● Taking power from the target system, with the jumper across the two right pins (EXT position) is not recommended. ● Power should be on the INT position, even if the pod is connected to a user target system. (This selection is the opposite of that for the bondout pods.) The MCU should be powered from the emulator since its operation is closely connected with the LCA. **Note**: This jumper should be in the INT position when a target is not connected to the pod board. This is also the default position.
B3, JB4 (XTAL)	Crystal Selection	Jumper pins for selecting the source of the crystal or clock. ● Jumpers installed in the right two positions (INT position) select the crystal on the pod. ● Jumpers installed in the left two positions (EXT position) select the crystal on the target is to be used. ● The oscillator is on the LCA, so do not drive the clock inputs when the emulator is powered off. ● If you use an external clock, note that XTAL1 (pin 19) is input, and XTAL2 (pin 18) needs to be left open. **Note**: These jumpers should be in the INT position when a target is not connected to the pod board. This is also the default position.

POD–C528

Jumper Identification and Description (continued)

Jumper Designation	Function	Description
JP7	RST	This jumper connects the target RESET pin to the emulator. If the target system has a watchdog, it might interfere with the emulator in monitor mode. If this occurs, remove RST.
JP2	SY0, SY1	This header connects external signals to the emulator. SY1 and SY0 are used for the trace function. SY0 (the right pin) can also be used in the breakpoint logic. The left pin is SY1. The silk screen might be wrong on early boards.
JP1	EM/FLF/ ANB	This header contains three pins: pin 1 (EM), pin 2 (FLF), and pin 3 (ANB). Pin 1 signals emulation or monitor mode. It is low when emulation is active (when the user program is running). Pins 2 and 3 are not used currently, but can be used in the future.
P1, P0		• The upper pins carry the MCU Port 1 pins: the P1.0 pin is on the left, and the P1.7 pin is on the right. Port 1 is connected directly to the MCU. • The lower pins carry Port 0: P0.0 is on the left, and P0.7 is on the right. Port 0 is emulated in the LCA which enables P0 to have better sink capability than the MCU. • P0 output signals also appear three clock cycles later than the real part. • The middle row of the jumper P1/P0 is normally connected to P1 using the jumpers. The signal level on the middle pins will be reflected on the trace display. • The board is delivered with the jumpers set so that P1 will be traced. • You can connect external signals to the middle pins if the jumpers to P1 are removed. • When a jumper is connected to the trace, an extra load ALS TTL load is added (74ALS258).
P3, P2		• The upper pins carry the MCU Port 3 pins: P3.0 is on the left, and P3.7 is on the right. Port 3 is connected directly to the MCU except for P3.6 and P3.7. These pins go through a 74HC4066 which adds approximately 100 Ohms to their output impedance. • The lower pins carry Port 2: P2.0 is on the left, and P2.7 is on the right. Port 2 is emulated in the LCA which enables P2 to have better sink capability than the MCU. • P2 output signals also appear three clock cycles later than the real part. • The middle row of the jumper P3/P2 is normally connected to P3 using the jumpers. It can be used for P2 or other external signals the same way as P1/P0. • When a jumper is connected to the trace, an extra load ALS TTL load is added (74ALS258).
S1	Reset	Resets the MCU and can be used instead of the target system reset.

POD-C528

Notes

- Ports P0, and P2 are emulated, and have slightly different AC and DC characteristics.
  - Emulated I/O pins sink a minimum of 4 mA, and source (P2) minimum 4 mA for two clock cycles after an output low to high transition.
  - As high outputs or inputs, P2 ports have 22K-Ohm pull-up resistors.
- The trace of accesses to the on-chip AUX-RAM contains correct addresses and data for writes. However, if you have XDATA mapped to the emulator, the data will be written to the emulator RAM. For reads, the Data Display window in Seehau might contain the emulation RAM data if data is mapped to the emulator. This data might not be the same as the AUX-RAM unless you map the XDATA to the target. It is recommended that you always map the AUX-RAM region to the target.
- If your emulator board only has 32K RAM, XDATA should normally be mapped to target.
- P3.6 and P3.7 go through a 74HC4066 that adds about 100 ohms to their output impedance.
- When accessing target memory, P2, P0, and ALE are delayed approximately 25 ns.
- The clock circuitry is emulated, but is very close to the MCU clock.
- The silk-screen can be reversed for SY0 and SY1. SY1 is on the left.
- The timer/counters are stopped at breakpoints. This means that the serial port also stops at breakpoints. If a character is received or sent at this moment it is distorted.
- If you break emulation and IDLE or PWD is lit, a number of time-out error messages shows. Issue the RESET CHIP command before you do anything else.
- If you want to change the MCU, you need to use a Philips 8XC528. Other bits do not affect the emulation mode.

POD–C528

Board Layout

Figure 113 shows locations of jumpers and Figure 114 shows the jumper pins in detail.

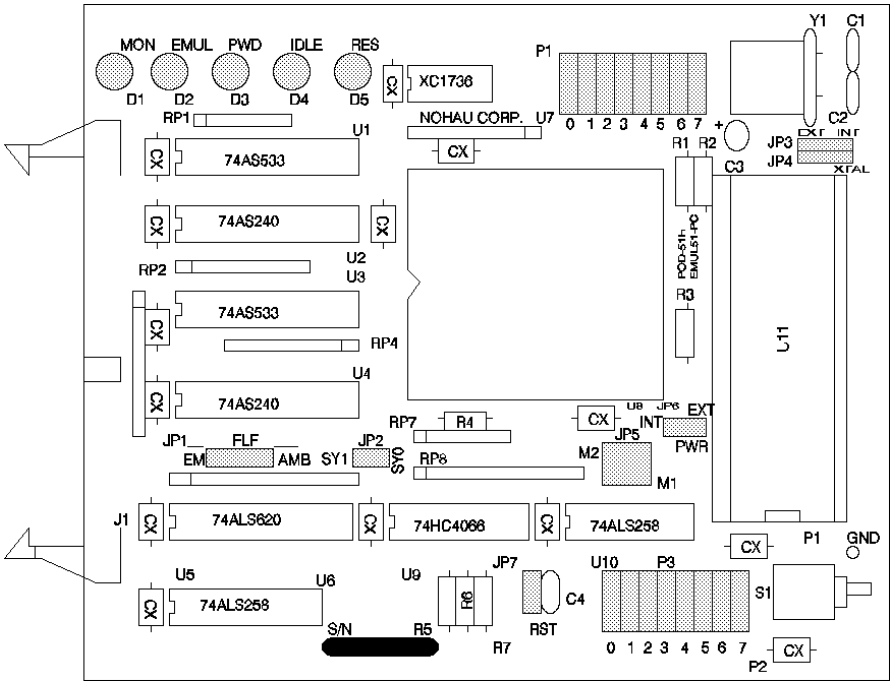


Figure 113. Locations of Jumpers on POD–C528

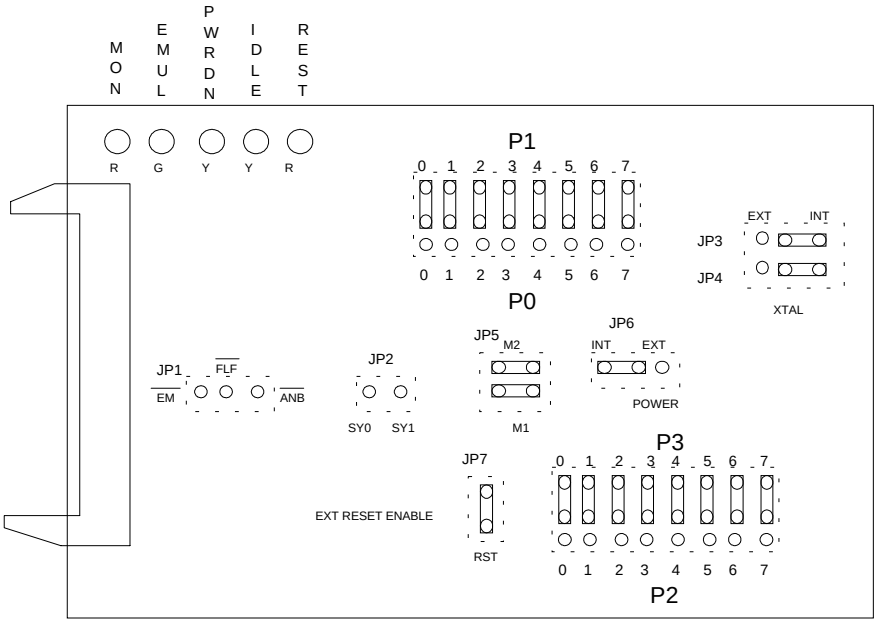


Figure 114. POD–C528 Jumper Diagram

POD–C550–PGA: Hooks Mode

Operating Instructions

The POD–550 is used to emulate Philips MCUs 80C550, 83C550, and 87C550.

To work with this pod, you need to select **POD–C550** from the **Select Processor** drop-down list box (**Step 3** of the **Communications** tab in the Emulator Configuration window).

The EPROM on the emulator board must be version COM 1.4.

The solder side of the board has a 44-pin PGA plug. It is recommended to use a PGA socket in your target system for mechanical stability. If a 44-pin PLCC is used, an adapter can be purchased.

The pod boards use a Philips 80C550 chip in a special emulation mode. This emulation mode also uses a programmable gate array from XILINX (LCA).

LED Indicators

Function	Description
MON	A red LED that indicates when the system is in monitor mode. MON turns off during emulation mode (Run or Go) and turns back on when a break occurs and the system returns to monitor mode.
EMUL	A green LED that indicates when the system is in emulation mode. (It performs the inverse function of MON.)
PWD	A yellow LED that indicates when the MCU is in power-down mode.
IDLE	A yellow LED that indicates when the MCU is in idle mode.
RES	A red LED that indicates the MCU is receiving a reset signal.

POD–C550–PGA

Jumper Identification and Description

Jumper Designation	Function	Description
JP4	M1, M2	These jumpers determine the mode of operation of P0, P2, P3.6 and P3.7. If only internal memory is to be emulated, both jumpers (M1 and M2) should be in. This enables P0, P2, P3.6 and P3.7 to be used as I/O ports. **M1 Jumper Removed** • If this jumper is out, P0 has the lower address and data. P2 has the upper address for external code fetches. • PSEN is active. • The emulator mapping determines whether the code is read from emulation memory or target memory. **M2 Jumper Removed** • If this jumper is out, P0 has the lower address and data. P2 has the upper address for MOVX instructions. • P3.6 and P3.7 is used for the RD and WR strobes. • The mapping determines if the RD and WR strobes go to emulation memory or target memory. • If only internal RAM is used, the emulator board has 128K RAM, map data MAPX to target, and the M2 jumper should be in. **Note**: You can change the mode jumpers M1and M2 at any time; however, you must issue a reset chip operation for the LCA and MCU to properly change modes.
JP5 (PWR)	MCU Power	• Power should be supplied from the emulator. • The jumper should be across the two right pins (INT). • Taking power from the target system, with the jumper across the two left pins (EXT) is not recommended. • Power should be on the INT position, even if the pod is connected to a user target system. (This selection is the opposite of that for the bondout pods.) The MCU should be powered from the emulator since its operation is closely connected with the LCA. **Note**: This jumper should be in the INT position when a target is not connected to the pod board. This is also the default position.
XTAL	Crystal Selection	Jumper pins for selecting the source of the crystal or clock. • Jumpers installed in the right two positions (INT position) select the crystal on the pod. • Jumpers installed in the left two positions (EXT position) select the crystal on the target. • The oscillator is on the LCA, so do not drive the clock inputs when the emulator is powered off. **Note**: These jumpers should be in the INT position when a target is not connected to the pod board. This is also the default position.

POD-C550-PGA

Jumper Identification and Description (continued)

Jumper Designation	Function	Description
JP2	RST	This jumper connects the target RESET pin to the emulator. If the target system has a watchdog, it might interfere with the emulator in monitor mode. If this occurs, remove RST.
JP3	SY0, SY1	This header connects external signals to the emulator. SY1 and SY0 are used for the trace function. SY0 (the right pin) can also be used in the breakpoint logic. The left pin is SY1. The silk screen might be wrong on early boards.
JP1	EM/FLF/ ANB	This header contains three pins: pin 1 (EM), pin 2 (FLF), and pin 3 (ANB). Pin 1 signals emulation or monitor mode. It is low when emulation is active (when the user program is running). Pins 2 and 3 are not used currently, but can be used in the future.
P1, P0		• The upper pins carry the MCU Port 1 pins: the P1.0 pin is on the left, and the P1.7 pin is on the right. Port 1 is connected directly to the MCU. • The lower pins carry Port 0: P0.0 is on the left, and P0.7 is on the right. Port 0 is emulated in the LCA which enables P0 to have better sink capability than the MCU. • P0 output signals also appear three clock cycles later than the real part. • The middle row of the jumper P1/P0 is normally connected to P1 using the jumpers. The signal level on the middle pins will be reflected on the trace display. • The board is delivered with the jumpers set, so that P1 will be traced. • You can connect external signals to the middle pins if the jumpers to P1 are removed. • When a jumper is connected to the trace, an extra load ALS TTL load is added (74ALS258).
P3, P2		• The upper pins carry the MCU Port 3 pins: P3.0 is on the left, and P3.7 is on the right. Port 3 is connected directly to the MCU except for P3.6 and P3.7. These pins go through a 74HC4066 which adds approximately 100 Ohms to their output impedance. • The lower pins carry Port 2: P2.0 is on the left, and P2.7 is on the right. Port 2 is emulated in the LCA which enables P2 to have better sink capability than the MCU. • P2 output signals also appear three clock cycles later than the real part. • The middle row of the jumper P3/P2 is normally connected to P3 using the jumpers. It can be used for P2 or other external signals the same way as P1/P0. • When a jumper is connected to the trace, an extra load ALS TTL load is added (74ALS258).
S1	Reset	Resets the MCU and can be used instead of the target system reset.

POD–C550–PGA

Notes

- Ports P0, and P2 are emulated, and have slightly different AC and DC characteristics.
  - Emulated I/O pins sink a minimum of 4 mA, and source (P2) minimum 4 mA for two clock cycles after an output low to high transition.
  - As high outputs or inputs, P2 ports have 22K-Ohm pull-up resistors.
- P3.6 and P3.7 go through a 74HC4066 which adds about 100 ohms to their output impedance.
- When accessing target memory, P2, P0, and ALE will be delayed approximately 25 ns.
- The clock circuitry is emulated, but is very close to the MCU clock.
- The silk screen might be reversed for P3/P2 and P1/P0. P3/P2 is in the upper right corner under the S/N block. P1/P0 is below the target PGA plug.
- The timer/counters are stopped at breakpoints. This means that the serial port also stops at breakpoints. If a character is received or sent at this moment it is distorted.
- If you break emulation and IDLE or PWD is lit, a number of time-out error messages shows. Issue the RESET CHIP command before you do anything else.
- If you use Port 1 pins as digital inputs, do the following:
  1. Connect +5V to AVCC (pin 1).
  2. Connect ground to AVSS (pin 2 on the DIP package, pin 4 on the PLCC package).

POD–C550–PGA

Board Layout

Figure 115 shows locations of jumpers and Figure 116 shows the jumper pins in detail.

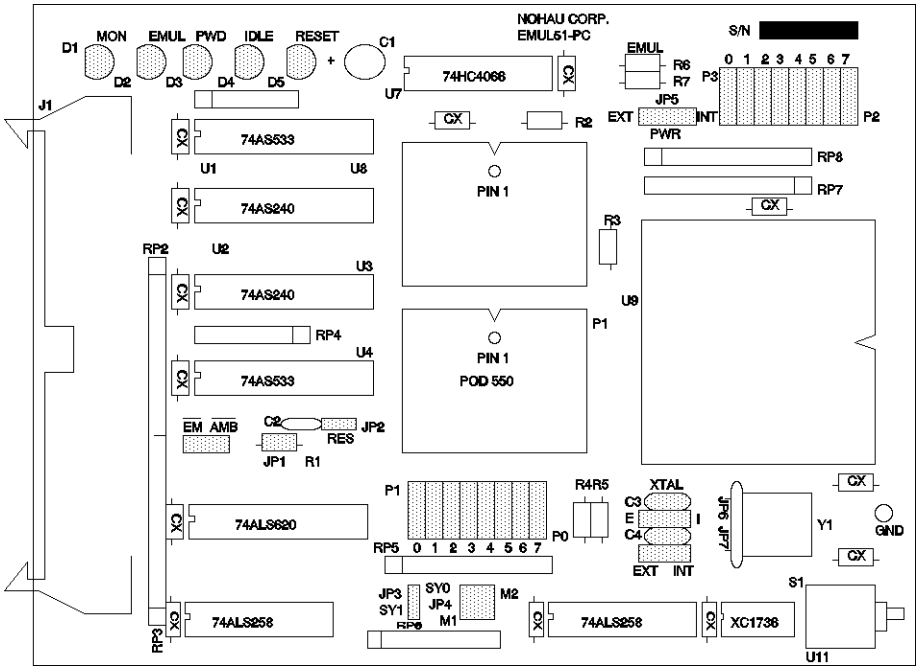


Figure 115. Locations of Jumpers on POD–C550–PGA

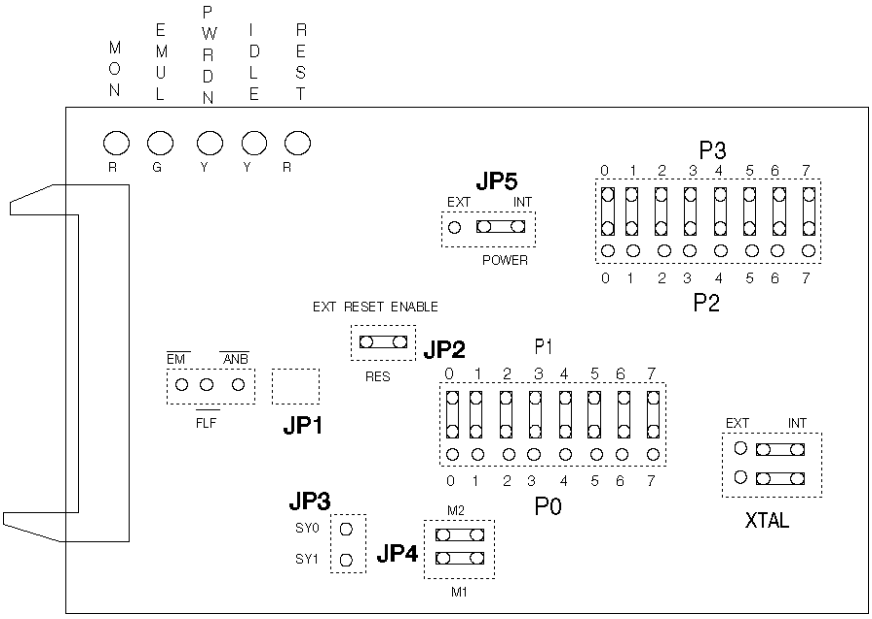


Figure 116. Jumper Diagram for POD–C550–PGA

POD–C554

Operating Instructions

For this pod, you must use Seehau 1.0512A or later.

To work with this pod, you need to select **POD–C554** from the **Select Processor** drop-down list box (**Step 3** of the **Communications** tab in the Emulator Configuration window).

The POD–C554 is used to emulate the Philips MCU 8xC554.

The EPROM on the emulator board must be version COM 1.4.

The solder side of the board has a 68-pin PGA plug. The PGA pins can be used if the target board has a PGA socket. If the target has a PLCC socket an adapter must be used.

(EMUL51–PC/PGA68–PLCC68) The plug goes into the target system replacing the MCU IC.

The pod board has a special Philips bondout 83C554 chip. This emulation mode also uses a programmable gate array from XILINX (LCA).

LED Indicators

Function	Description
MON	A red LED that indicates when the system is in monitor mode. MON turns off during emulation mode (Run or Go) and turns back on when a break occurs and the system returns to monitor mode.
EMUL	A green LED that indicates when the system is in emulation mode. (It performs the inverse function of MON.)
PDWN	A yellow LED that indicates when the MCU is in power-down mode.
IDLE	A yellow LED that indicates when the MCU is in idle mode.
RES	A red LED that indicates the MCU is receiving a reset signal.

Jumper Identification and Description

Jumper Designation	Function	Description
POWER	Power Selection	Jumper pins for selecting the source of the power. • When jumpers are installed towards the INT position, the power to the MCU comes from the emulator. • When jumpers are installed towards theEXT position, the power to the MCU comes from the target. • The MCU should be powered from the emulator since its operation is closely connected with the LCA. **Note**: This jumper should be in the INT position when a target is not connected to the pod board. This is also the default position.

POD–C554

Jumper Identification and Description (continued)

Jumper Designation	Function	Description
XTAL	Crystal Selection	Jumper pins for selecting the source of the crystal or clock. • Jumpers installed in the INT position select the crystal on the pod. • Jumpers installed in the EXT position select the crystal on the target. **Note**: These jumpers should be in the INT position when a target is not connected to the pod board. This is also the default position.
RST		This jumper (located close to R21) connects the target RESET pin to the emulator. If the target system has a watchdog, it might interfere with the emulator in monitor mode. If this occurs, remove RST.
SY0, SY1		This header connects external signals to the emulator. They are also used for the trace function. SY0 can also be used in the breakpoint logic.
EM/, FLF/, ANB/		These jumper pins, located between U2 and U3 carry signals from the emulator. • The EM/ (left pin) is high when the emulator is in monitor mode, and low if in emulation mode. • The FLF/ and ANB/ pins are used only with the ATR or ETR tace options.
P1, P0		These jumper blocks will be traced under the P1 identifier in both the trace setup and trace display screens. • The bit order of the signals is that bit 0 is on the left and bit 7 is on the right of the row of eight jumper connections. • When a jumper is connected to the trace, an extra load ALS TTL load is added (74ALS258).
P3, P2		These jumper blocks will be traced under the P3 identifier in both the trace setup and trace display screens. • The bit order of the signals is that bit 0 is on the left and bit 7 is on the right of the row of eight jumper connections. • When a jumper is connected to the trace, an extra load ALS TTL load is added (74ALS258).
S1	Reset	Resets the MCU and can be used instead of the target system reset.

Notes

- Many of the port pins are emulated, and have slightly different AC and DC characteristics. Port 0, 2, and P3.6/7 (when in single-chip mode) are emulated and will reflect the following characteristics:
  - Emulated I/O pins sink a minimum of 4 mA, and source (P2) minimum 4 mA for two clock cycles after an output low to high transition.
  - As high outputs or inputs, P2 ports have 22K-Ohm pull-up resistors.

POD–C554

- The clock circuitry is emulated, but is very close to the 8xC554 clock.
- The delay from reset low to MCU start is longer than in the normal part. The delay is about 80 clock cycles.
- The timer/counter is stopped at breakpoints, but counts one step extra when used with the internal clock.
- A reset during emulation will generate about five invalid frames in the trace buffer. Idle mode will generate about 16 invalid frames.

Board Layout

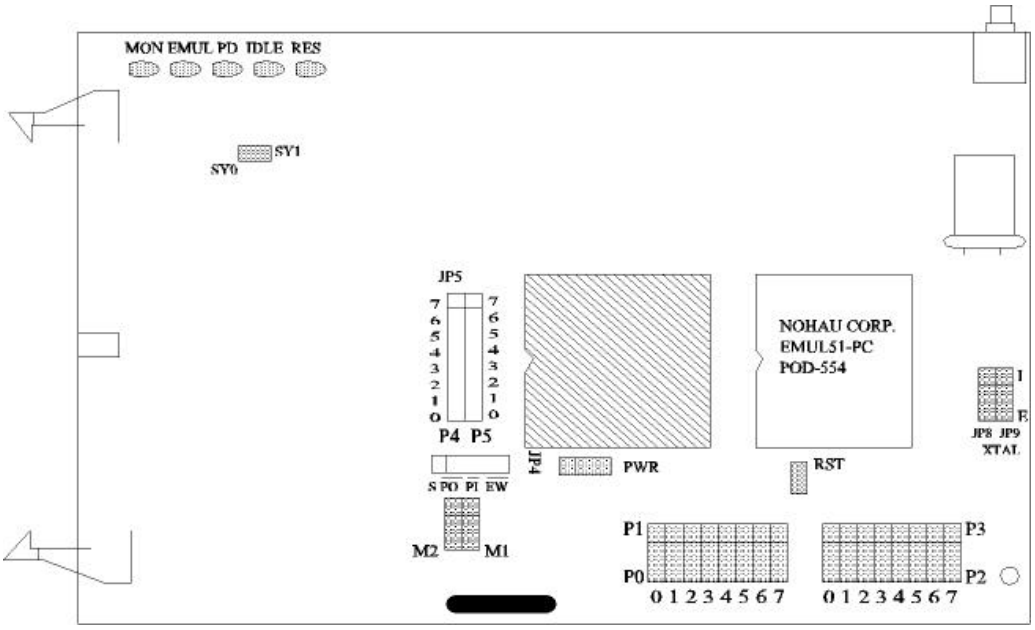


Figure 117. POD–C554 Jumper Layout

POD–C558–16: Hooks Mode

Operating Instructions

The POD–C558 adapter is used to emulate Philips MCUs 8xC557, 83CE558, 89CE558, 8xCE559, and 8xCE560.

To work with this pod, you need to select **POD–C558** from the **Select Processor** drop-down list box (**Step 3** of the **Communications** tab in the Emulator Configuration window).

The EPROM on the emulator board must be version COM 1.4.

The solder side of the board has two 16-pin headers and two 24-pin headers. They are designed to plug into this Emulation Technology adapter for surface mount pads (PQFP package): ET/EPP–080–QF08–LG.

The Emulation Technology adapters can be ordered from Nohau, or directly from Emulation Technology. The adapter goes into the target system replacing the MCU IC.

The pod board uses a Philips 85C558 chip in a special emulation mode. This emulation mode also uses a programmable gate array from XILINX (LCA).

LED Indicators

Function	Description
MON	A red LED that indicates when the system is in monitor mode. MON turns off during emulation mode (Run or Go) and turns back on when a break occurs and the system returns to monitor mode.
EMUL	A green LED that indicates when the system is in emulation mode. (It performs the inverse function of MON.)
PWRDN	A yellow LED that indicates when the MCU is in power-down mode.
IDLE	A yellow LED that indicates when the MCU is in idle mode.
RST	A red LED that indicates the MCU is receiving a reset signal.

POD–C558

Jumper Identification and Description

Jumper Designation	Function	Description
M1, M2		These jumpers determine the mode of operation of P0, P2, P3.6 and P3.7. If only internal memory is to be emulated, both jumpers (M1 and M2) should be in. This enables P0, P2, P3.6 and P3.7 to be used as I/O ports. **M1 Jumper Removed** ● If this jumper is out, P0 has the lower address and data. P2 has the upper address for external code fetches. ● PSEN is active. ● The emulator mapping determines whether the code is read from emulation memory or target memory. **M2 Jumper Removed** ● If this jumper is out, P0 has the lower address and data. P2 has the upper address for MOVX instructions. ● P3.6 and P3.7 is used for the RD and WR strobes. ● The mapping determines if the RD and WR strobes go to emulation memory or target memory. ● If only internal RAM is used, the emulator board has 128K RAM, map data MAPX to target, and the M2 jumper should be in. **Note**: You can change the mode jumpers M1and M2 at any time; however, you must issue a reset chip operation for the LCA and MCU to properly change modes.
JP6 (PWR)	MCU Power	● Power should be supplied from the emulator. ● The jumper should be across the two upper pins (INT). ● Taking power from the target system, with the jumper across the two lower pins (EXT) is not recommended. ● Power should be on the INT position, even if the pod is connected to a user target system. (This selection is the opposite of that for the bondout pods.) The MCU should be powered from the emulator since its operation is closely connected with the LCA. **Note**: This jumper should be in the INT position when a target is not connected to the pod board. This is also the default position.
XT/X1		This jumper determines the source of the pin SELXTAL1. ● If the jumper is in the top position (X1) SELXTAL1 is forced to select XTAL1. ● If it is in the lower (XT) position, SELXTAL1 comes from the target system.
XTAL1	Crystal Selection	Jumper pins for selecting the source of the crystal or clock. ● Jumpers installed in the upper two positions (INT position) select the crystal on the pod. ● Jumpers installed in the lower two positions (EXT position) select the crystal on the target. **Note**: These jumpers should be in the INT position when a target is not connected to the pod board. This is also the default position.

POD–C558

Jumper Identification and Description (continued)

Jumper Designation	Function	Description
X4GND		Rev. A boards have a jumper marked X4GND. With this jumper installed pin X4 is grounded. Remove the jumper top if X4 is used.
RST		This jumper connects the target RESET pin to the emulator. If the target system has a watchdog, it might interfere with the emulator in monitor mode. If this occurs, remove RST.
SY0, SY1		This header connects external signals to the emulator. SY1 and SY0 are used for the trace function. SY0 (the right pin) can also be used in the breakpoint logic.
EM/, FLF/, ANB/		This header carries signals from the emulator. • EM/ (left pin) is high when the emulator is in monitor mode, and low if in emulation mode. • FLF/ (middle pin) and ANB/ (right pin) are used with the Advanced Trace Board (ATR).
P1, P0		• The upper pins carry the MCU Port 1 pins: the P1.0 pin is on the left, and the P1.7 pin is on the right. Port 1 is connected directly to the MCU. • The lower pins carry Port 0: P0.0 is on the left, and P0.7 is on the right. Port 0 is emulated in the LCA which enables P0 to have better sink capability than the MCU. • P0 output signals also appear three clock cycles later than the real part. • The middle row of the jumper P1/P0 is normally connected to P1 using the jumpers. The signal level on the middle pins will be reflected on the trace display. • The board is delivered with the jumpers set so that P1 will be traced. • You can connect external signals to the middle pins if the jumpers to P1 are removed. • When a jumper is connected to the trace, an extra load ALS TTL load is added (74ALS258).
P3, P2		• The upper pins carry the MCU Port 3 pins: P3.0 is on the left, and P3.7 is on the right. Port 3 is connected directly to the MCU except for P3.6 and P3.7. These pins go through a 74HC4066 which adds approximately 100 Ohms to their output impedance. • The lower pins carry Port 2: P2.0 is on the left, and P2.7 is on the right. Port 2 is emulated in the LCA which enables P2 to have better sink capability than the MCU. • P2 output signals also appear three clock cycles later than the real part. • The middle row of the jumper P3/P2 is normally connected to P3 using the jumpers. It can be used for P2 or other external signals the same way as P1/P0. • When a jumper is connected to the trace, an extra load ALS TTL load is added (74ALS258).
S1	Reset	Resets the MCU and can be used instead of the target system reset.

POD-C558

Notes

- Ports P0, and P2 are emulated, and have slightly different AC and DC characteristics:
  - Emulated I/O pins sink a minimum of 4 mA, and source (P2) minimum 4 mA for two clock cycles after an output low to high transition.
  - As high outputs or inputs, P2 ports have 22K-Ohm pull-up resistors.
- P3.6 and P3.7 go through a 74HC4066 which adds about 100 Ohms to their output impedance.
- When accessing target memory, addresses on P2 and P0 are delayed approximately one clock cycle. The falling edge of ALE is also delayed about one clock cycle. This is caused by the special emulation mode of the 85CE558 chip.
- Signals ADEXS and EW/ signals have a 220K-Ohm pull-up resistor on the pod.
- The silk screen might say STADC instead of ADEXS on some pods.
- The timer/counters are stopped at breakpoints. This means that the serial port also stops at breakpoints. If a character is received or sent at this moment it will be distorted.
- The trace of accesses to the 83C558 AUX-RAM will contain correct addresses when using DPTR, but only the low address will be correct using RI. The high address will always show zero for any value of XRAM. Data will be correct for writes but not for reads. If your emulator has 128K RAM and data (MAPX) is mapped to emulator, the data will be written to the emulator RAM. For reads, the data displayed is the emulator RAM data if data is mapped to emulator. This data might not be the same as in the AUX-RAM unless it has already been written to the emulator RAM. This method will not work if XRAM is used since the high address might be incorrect.
- If your emulator board only has 32K RAM, XDATA should normally be mapped to target.
- If you break emulation and IDLE or PWRDN is lit, a number of time-out error messages show. Issue the RESET CHIP command before you do anything else.

POD–C558

Board Layout

Figure 118 shows locations of jumpers and Figure 119 shows the jumper pins in detail.

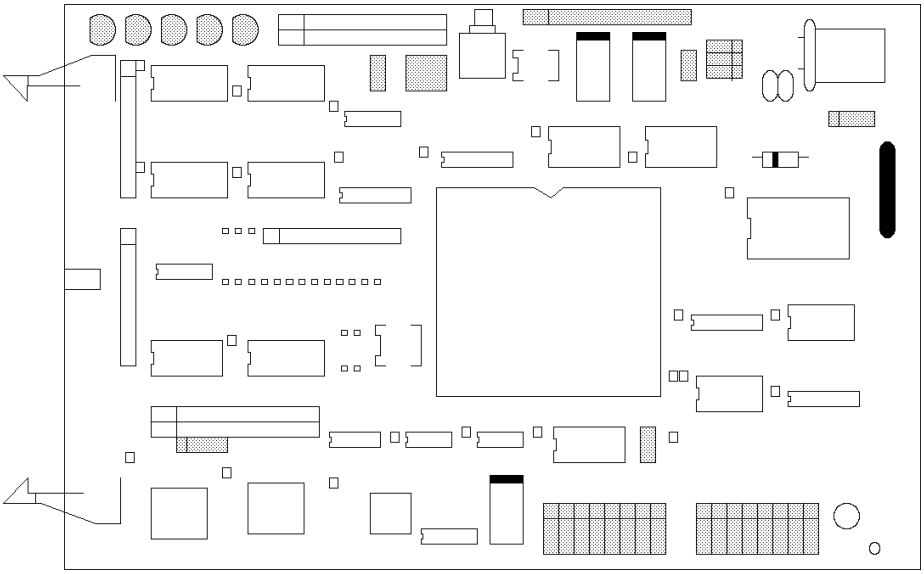


Figure 118. Locations of Jumpers on POD–C558–16

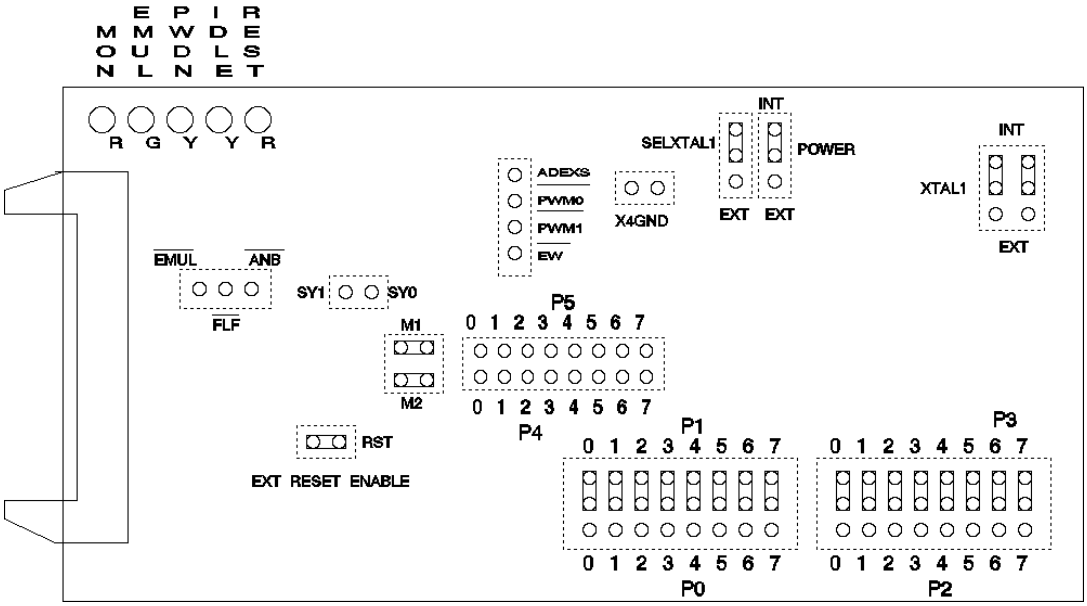


Figure 119. POD–C558–16 Jumper Diagram

Installing and Configuring the Seehau Software

Installing Seehau Software From the CD

To install the Seehau software, do the following:

1. Locate your Seehau CD and insert the CD into your CD ROM drive. The installation process will start automatically.
2. Follow the instructions that appear on your screen.

> **Note**
>
> If the installation does not start automatically, you probably have your Windows Auto-run feature disabled. You will then need to use Windows Explorer and navigate to the CD root directory. Double-click **Install**.

Downloading and Installing Seehau From the Internet

1. Go to the Nohau web site (http://www.icetech.com/).
2. In the menu bar, click **Downloads**. The Downloads page opens.
3. Click **Current Software Versions**. The Current Software Versions page opens displaying a table with listings of current software versions.
4. In the **Family** column, scroll to EMUL51–PC, and click **Seehau**.
5. Review the "Known Issues" section.
6. Click **Yes I Want to Download**. The Software Request Form page opens. Complete this form, then click **Proceed**.
7. Click **Go to Download**. Click either option for a download site. The Nohau Software Updates page opens.
8. Select the **Download Site** option, then click **EMUL51–PC**.
9. Click **S8051.exe**. The application will start downloading.
10. After downloading the application, click **S8051.exe** and follow the installation instructions.

Note

After installing the Seehau software, the **Setup Complete** dialog box opens where you can review the Readme.txt file and/or launch the Seehau 51 configuration.

If you are installing Seehau for the first time, you must launch the Seehau 51 configuration before running the Seehau software.

Selecting to Automatically Start the Seehau Configuration Program

After installing Seehau, it is recommended that you automatically start the Seehau Configuration program. Do the following steps before starting Seehau:

1. From the **Setup Complete** dialog box, select **Launch Seehau51 Configuration**.
2. Click **Finish**.

If you do not select to automatically start the Seehau Configuration Program, do the following:

1. From the **Start** menu, select **Programs**.
2. Select **Seehau51**. Then click **Config** to open the **Emulator Configuration Connect** dialog box (Figure 120).

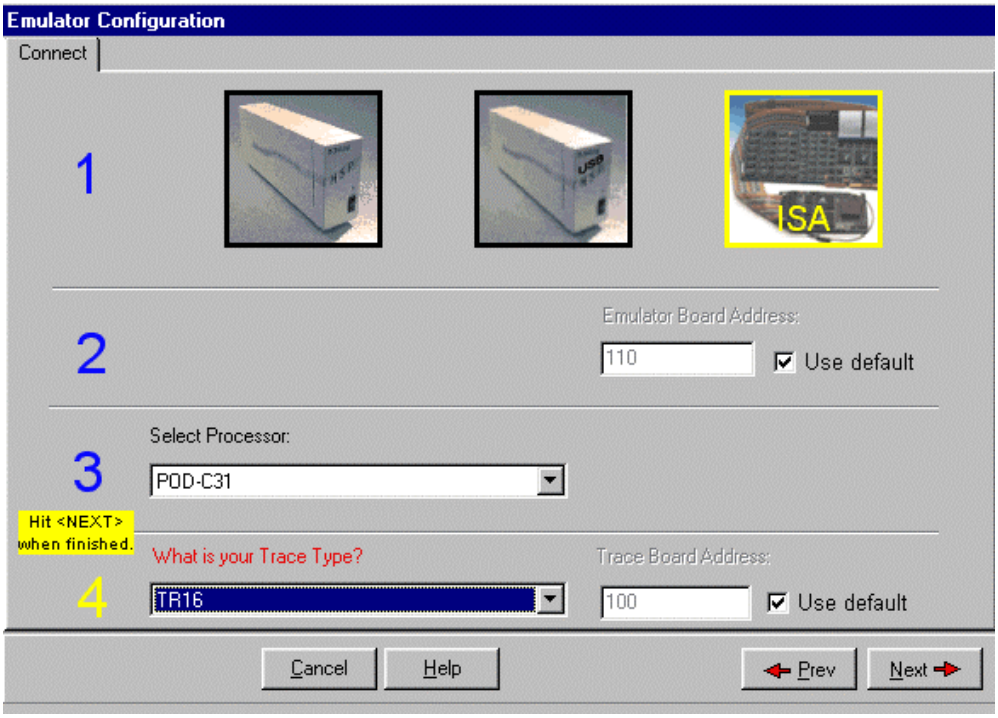


Figure 120. Emulator Configuration Connect Dialog Box

Note

You do not need the hardware connected at this time.

Configuring Seehau

First select one of the following communication interfaces:

- **High Speed Parallel Box (HSP)**: The emulator is placed in an external box and communicates with the PC through a standard PC parallel port (LPTx).
- **USB**: The emulator is placed in an external box and communicates with the PC through a USB port.
- **ISA Board**: Communicates with the emulator system through the PC's ISA bus.

Note

The serial expansion box called BOX–S is now obsolete and is not supported by Seehau. If you have a BOX–S, you can purchase a conversion kit to upgrade to an HSP box.

To configure your software, do the following steps:

1. From the **Start** menu, select **Programs**.
2. Select **Seehau51**, and click **Config**. The **Emulator Configuration Connect** dialog box opens.

As an example of setting up your configuration, the ISA communications interface is shown in Figure 120. The HSP communications interface is very similar, and the steps you do when using the HSP communications interface are almost identical.

The graphical user interface for this dialog box is divided into four regions. Do the following in each region:

1. Region 1—**Communications Interface**:
   Select either the HSP, USB, or ISA communications interface.
2. Region 2—**Emulator Board Address**:
   Contains the address of the internal communication link from your computer. For the ISA card, the default address is 110. To disable this default, clear the **Default** option and insert the appropriate address for the emulator board. For the HSB/USB Box, the emulator and trace boards should always be set to their defaults.
3. Region 3—**Select Processor**:
   Click the down arrow and select the pod type you are using.
4. Region 4—**Trace Board**:
   Click the down arrow and select your trace type. If you do not have a trace board, select **None**.
5. Click **Next**. The **Emulator Configuration, HdwCfg51** tab opens.

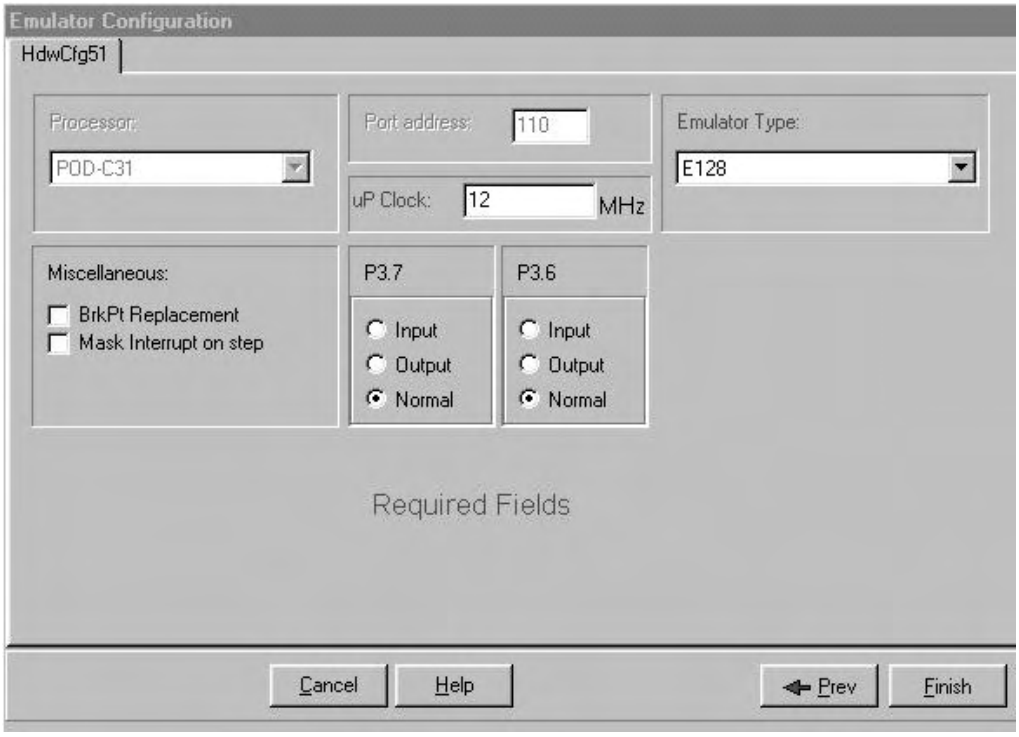


Figure 121. Emulator Configuration, HdwCfg51 Tab

Complete the following fields in the **HdwCfg51** tab:

- **Processor**: Shown for reference only. If you need to change the pod type, click **Prev**.
- **Port address**: Shown for reference only. If you need to change the port address, click **Prev**.
- **Emulator Type**: displays the type of emulator you are using. This information is available from the top of the board. The emulator type is configured during the initial configuration and does not need to be changed.
- **uP Clock** This field contains the time base reference for the trace timestamp only. The timestamp calculation assumes standard 8051 timing with 12 clocks per bus cycle. Enter the following:
  - If you are using a standard part, enter the crystal/oscillator frequency.
  - If you are using a Dallas 520/530/320/323 part, enter 1.5x the crystal/oscillator frequency.
  - If you are using a Philips or Temic X2 part, enter 2x the crystal/oscillator frequency.
  - If you are using a Philips or Temic X2 part, enter two times the external frequency.
- **Miscellaneous**
  - **Brkpt Replacement**: This (no skid) option prevents the current instruction at a breakpoint from being executed. When not enabled, the instruction at a breakpoint will be executed.
    Note: When using this option with the Dallas pods (320, 323, 520, 530) or the Temic/Philips X2 parts (6-clock parts), you must use the current revision of the EAxxx emulator board.

- **Mask interrupt on step**: Clears the interrupt-enable flag bit to allow single-stepping through your non-interrupt source code without servicing frequent or time-based interrupts. It then re-enables interrupts after the single step.

- **P3.7**: configures the reset state of Port 3 bit 7 which is normally -RD.

> **Note**
>
> This option should only be used with POD–31S (discontinued) and POD–51 (discontinued), or POD–C32HF–42.

  - **Input**: Configures P3.7 as input.
  - **Output**: Configures P3.7 as output.
  - **Normal**: Configures P3.7 as -RD.

- **P3.6**: configures the functionality of Port 3 bit 6 which is normally -WR.

> **Note**
>
> This option should only be used with POD–31S (discontinued) and POD–51 (discontinued), or POD–C32HF–42.

  - **Input**: Configures P3.6 as input.
  - **Output**: Configures P3.6 as output.
  - **Normal**: Configures P3.6 as -WR.

- **Cancel**: Exits without saving the settings for the dialog box.

- **Help**: Displays the Seehau Help file.

- **Finish**: Click to save the configuration and exit the dialog box. A window appears asking whether you want to start the emulator. Select **Yes** to launch Seehau. Select **No** to exit Seehau Configuration.

Configuring the Emulator Options From Within Seehau

From Seehau open the Emulator Configuration window. Select the **Config** menu and click **Emulator**. The **Emulator Configuration** dialog box opens (Figure 122).

There are six tabs across the top of the main **Emulator Configuration** window. When selected, each tab allows you to access the following dialog boxes:

Hdw Config:	Set up emulator hardware options.
Misc. Setup:	Select reset options and tab size.
Map Config:	Map address ranges to the emulator or target with or without write protection.
Bank Switching Logic:	Configure the bank switching logic.
Breakpoint Setup	Set up special breakpoint options.
BP Setup	Configure hardware breakpoint range.

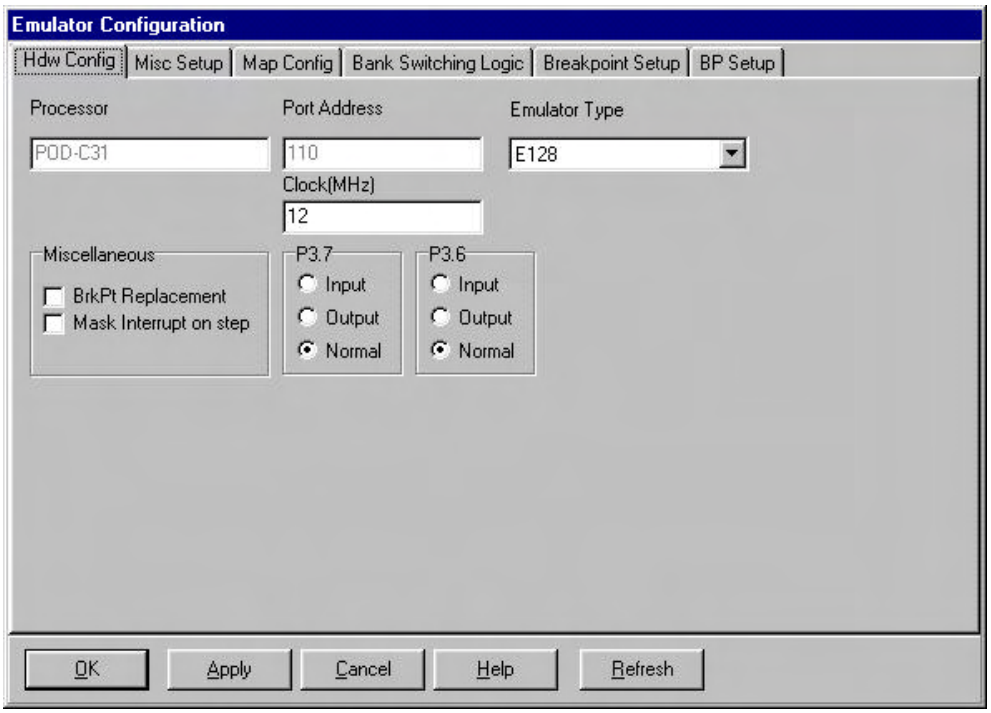


Figure 122. Hdw Config Tab

Hdw Config Tab

The **Hdw Config** tab displays the MCU that is being emulated, sets the clock speed for the time tag, and the P3.6 and P3.7 operation. Using the **Hdw Config** tab you can set the following:

- **Processor**: Displays the pod that you selected during the initial configuration. To change this, run Seehau Config. From the **Start** menu, select **Programs**. Then select **Seehau** and click **Config** from the sub menu. The **Emulator Configuration Connect** dialog box appears. For more information about this dialog box, refer to the "Configuring Seehau"section.

- **Port Address**: Shows the I/O address the ISA emulator card is set for. To change this, set the emulator jumpers and run Configure Seehau.

- **Clock (MHz)**: This field contains the time base reference for the trace timestamp only. The timestamp calculation assumes standard 8051 timing with 12 clocks per bus cycle. Enter the following:

  - If you are using a standard part, enter the crystal/oscillator frequency.
  - If you are using a Dallas 520/530/320/323 part, enter 1.5x the crystal/oscillator frequency.
  - If you are using an MX pod, enter 2xfor the external frequency.
  - If you are using a Philips or Temic X2 part, enter two times the external frequency.

- **Emulator Type**: Select the type of emulator you are using. This information is available from the top of the board. The emulator type is configured during the initial configuration and does not need to be changed.

- **Miscellaneous**
  - **Brkpt Replacement**: This (no skid) option prevents the current instruction at a breakpoint from being executed. When not enabled, the instruction at a breakpoint will be executed.
    Note: When using this option with the Dallas pods (320, 323, 520, 530) or the Temic/Philips X2 parts (6-clock parts), you must use the current revision of the EAxxx emulator board
  - **Mask interrupt on step**: Clears the interrupt-enable flag bit to allow single-stepping through your non-interrupt source code without servicing frequent or time-based interrupts. It then re-enables interrupts after the single step.
- **P3.7**: configures the reset state of Port 3 bit 7 which is normally -RD.

> **Note**
>
> This option should only be used with POD–31S (discontinued) and POD–51 (discontinued), or POD–C32HF–42.

  - **Input**: Configures P3.7 as input.
  - **Output**: Configures P3.7 as output.
  - **Normal**: Configures P3.7 as -RD.
- **P3.6**: configures the functionality of Port 3 bit 6 which is normally -WR.

> **Note**
>
> This option should only be used with POD–31S (discontinued) and POD–51 (discontinued), or POD–C32HF–42.

  - **Input**: Configures P3.6 as input.
  - **Output**: Configures P3.6 as output.
  - **Normal**: Configures P3.6 as -WR.
- **OK**: Saves the settings for the tabs and exits the dialog box.
- **Apply**: Saves the settings for this tab.
- **Cancel**: Exits without saving the settings for the dialog box.
- **Help**: Displays the Seehau Help file.
- **Refresh**: Allows you to retrieve and view the current trace and emulator hardware configuration settings.

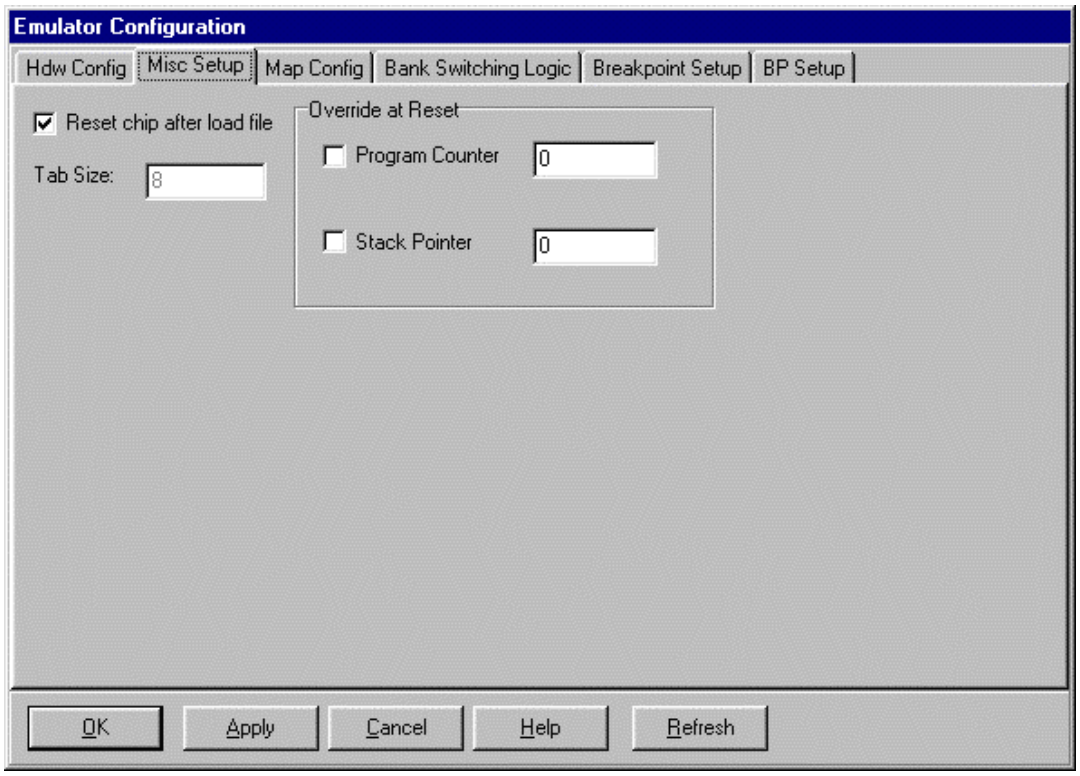


Figure 123. Misc Setup Tab

Misc Setup Tab

The **Misc Setup** tab (Figure 123) is used to set if the emulator is reset after loading the code, and what value the **Program Counter** and **Stack Pointer** are set to after reset.

- **Reset chip after load file**: Sets the emulator to issue a reset after the code is loaded. The result of this is that the program counter will contain 0 and the Source window will show the assembly startup code.
- **Tab Size**: Sets the number of spaces that a tab advances in the Source window.
- **Override at Reset**
  - The **Program Counter** option selects the override. Enter the program counter value in the text box.
  - The **Stack Pointer** option selects the override. Enter the stack pointer value in the text box.
- **OK**: Saves the settings for the tabs and exits the dialog box.
- **Apply**: Saves the settings for this tab.
- **Cancel**: Exits without saving the settings for the dialog box.
- **Help**: Displays the Seehau Help file.
- **Refresh**: Allows you to retrieve and view the current trace and emulator hardware configuration settings.

Map Config Tab

The **Map Config** tab (Figure 124) is used to map memory. The default configuration maps both code and XData memory to the emulator. Note that the code is read by PSEN, and the XData is read/write accessed through the MOVX instruction.

- **Range**: The 64K address space is divided into 4K blocks.
- **Code**: To map a block to the target, select the row that corresponds with the range you want to set. To map all the blocks to the emulator, click **Clear All** at the bottom of the **Code** column.
- **XData**: To map a block to the target, select the row that corresponds with the range you want to set. To map all the blocks to the emulator, click **Clear All** at the bottom of the **XData** column.
- **OK**: saves the settings for the tabs and exits the dialog box.
- **Apply**: saves the settings for this tab.
- **Cancel**: exits without saving the settings for the dialog box.
- **Help**: displays the Seehau Help file.
- **Refresh**: Allows you to retrieve and view the current trace and emulator hardware configuration settings.

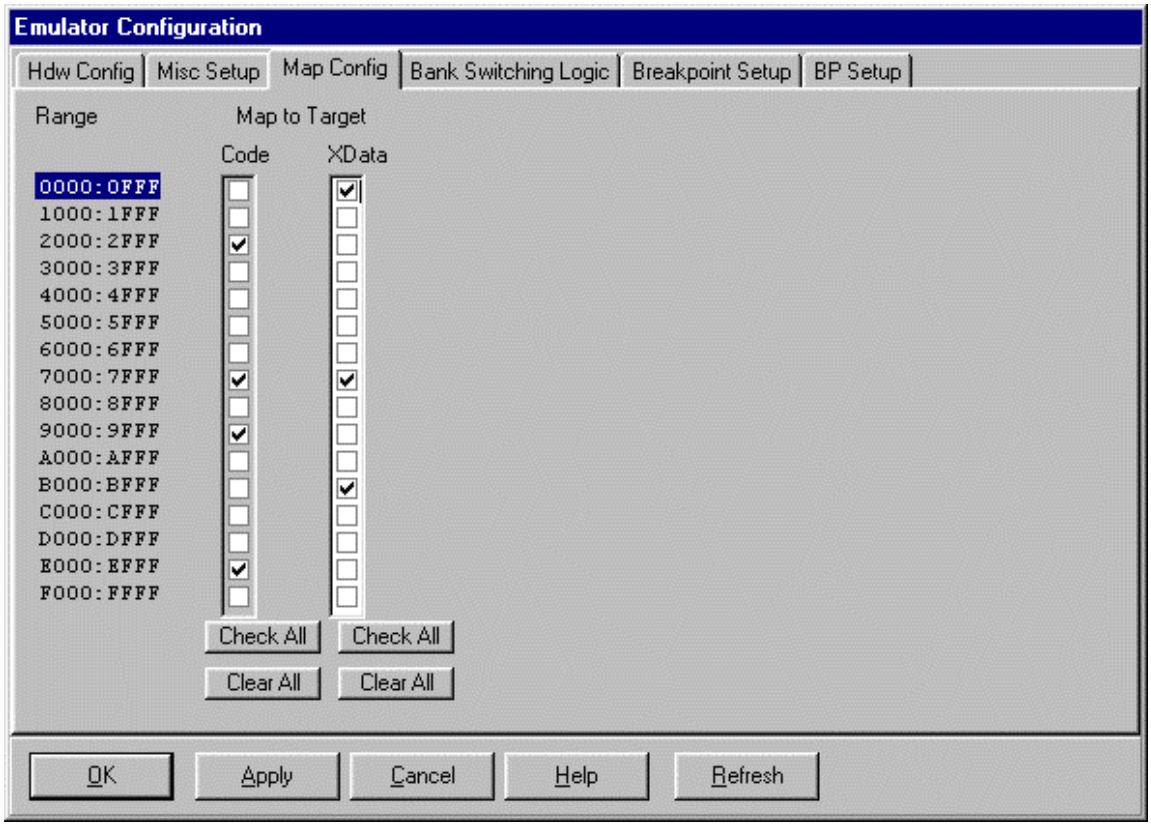


Figure 124. Map Config Tab

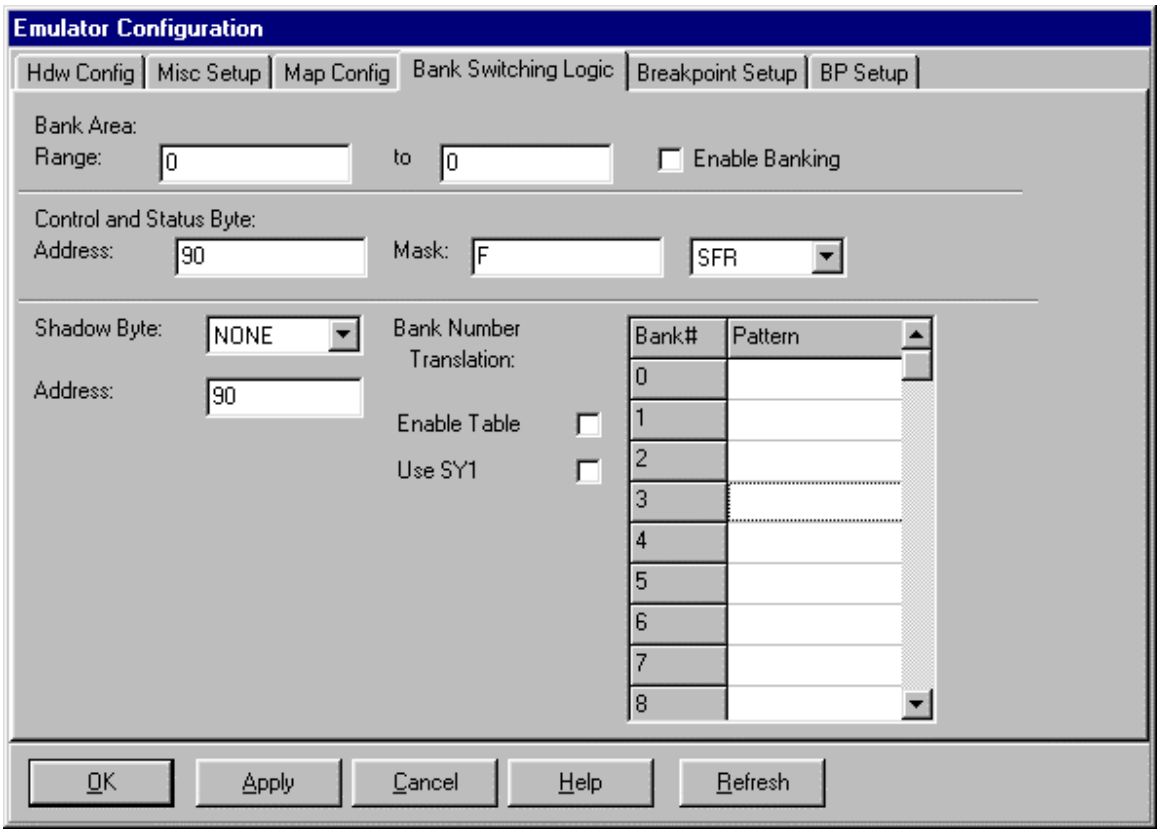


Figure 125. Bank Switching Logic Tab

Bank Switching Logic Tab

The **Bank Switching Logic** tab (Figure 125) configures emulation banking. You can also set up a banked code range, the control mechanism and a translation table in this tab.

- **Bank Area**
  - Enter the address of the region you want to bank in the **Range** text boxes.
  - Selecting the **Enable Banking** option enables the banking at the specified address.

- **Control and Status Byte**
  - The address in the **Address** text box controls the banking. When using a port, select **SFR**. For a latch in XData space, select **XData**.
  - The **Mask** filters the bits that are used. A 1 (one) indicates that bit is used.

Example

For an SFR address equal to 90H (Port 1) if the **Mask** equals 1CH then bits P1.2, P1.3 and P1.4 are used. (P1.2 being the LSB and P1.4 being the MSB for the binary counter of the banks.)

Use the drop-down list box to select an SFR or XData address to control the banking. Select **XData** if your target has a latch in XData space.

- **Shadow Byte**: If your target hardware uses a latch or write-only register in XDATA space for bank control, your code must maintain a shadow byte so the emulator can read the code in monitor mode to determine which bank is currently active. The shadow byte is written to immediately before the actual control address is written.

  Use the drop-down list to select where the shadow byte is mapped:

  - **None**: Indicates the shadow byte is not used.
  - **Data**: Indicates the shadow byte is in internal data space.
  - **XDATA**: Indicates the shadow byte is in external data space.

- **Bank Number Translation**: The emulator assumes a 1 to 1 correlation of bank number-to-bit pattern, for example:
  0000b = Bank 0, 0001b = Bank 1, 0111 = Bank 7

  If your configuration uses any other correlation, such as Grey code, this can be defined in the bank number translation table. This table defines the bit patterns that are used in the control byte for that bank number:

  - The **Enable Table** option acitvates the bank table.
  - The **Use SY1** option enables the banking logic to use the SY1 pin on the pod board as the fifth bank input for support of up to 32 banks.

- **OK**: Saves the settings for the tabs and exits the dialog box.
- **Apply**: Saves the settings for this tab.
- **Cancel**: Exits without saving the settings for the dialog box.
- **Help**: Displays the Seehau Help file.
- **Refresh**: Allows you to retrieve and view the current trace and emulator hardware configuration settings.

Breakpoint Setup Tab

The **Breakpoint Setup** tab (Figure 126) configures how breakpoints operate. You can also select the cycle type.

- **Break On**
  - **Address Only**: Selecting this option configures breakpoints to be an address only.
  - **SY0 and Address**: Selecting this option causes a breakpoint when SY0 and address are *true*.
  - **SY0 or Address**: Selecting this option causes a breakpoint when SY0 or address are *true*.
  - **SY0 Only**: Selecting this option causes a breakpoint when the SY0 signal is *true*.

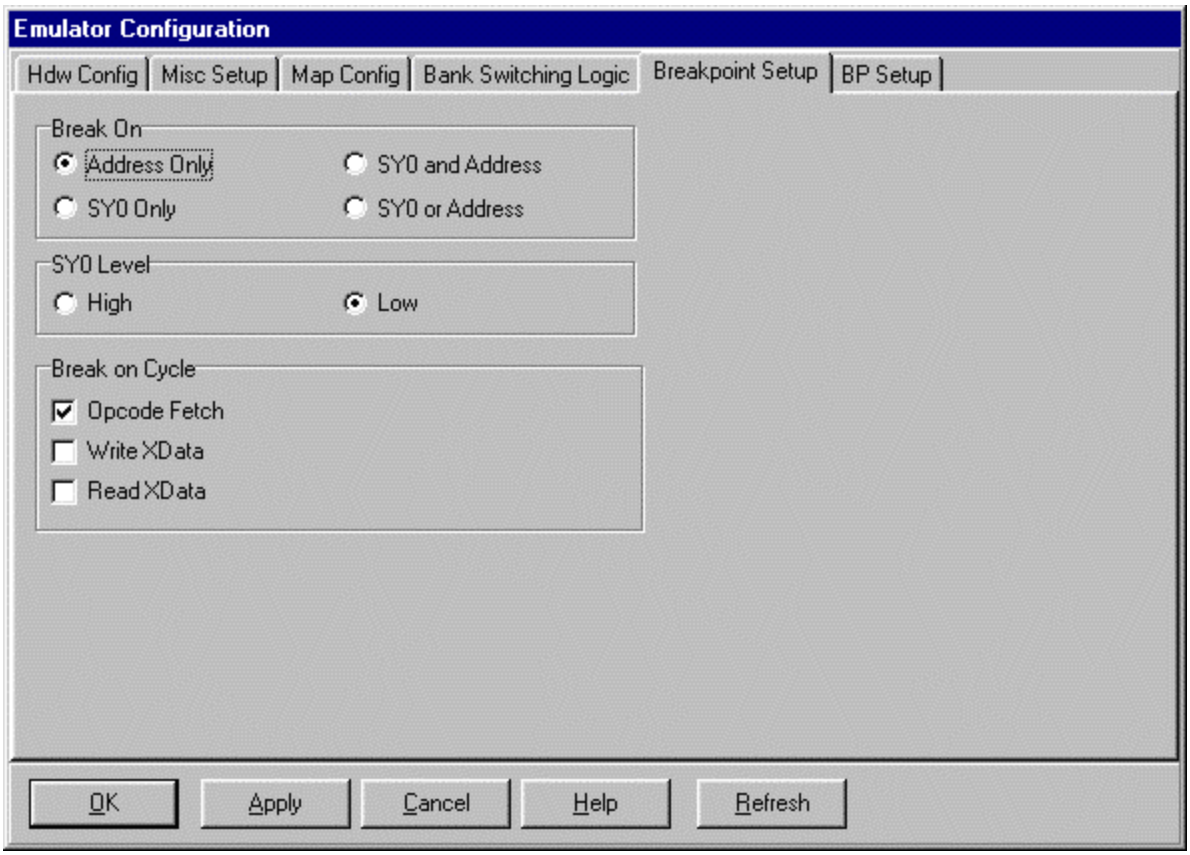


Figure 126. Breakpoint Setup Tab

- **SY0 Level**: sets the true state of the SY0 signal.
- **Break On Cycle**: These qualifiers are ORd together allowing any or all of them to be selected.
  - **Opcode Fetch**: Selecting this option qualifies the breakpoint to be a fetch.
  - **Write XData**: Selecting this option qualifies the breakpoint to be a write to XData.
  - **Read XData**: Selecting this option qualifies the breakpoint to be a read of XData.
- **OK**: Saves the settings for the tabs and exits the dialog box.
- **Apply**: Saves the settings for this tab.
- **Cancel**: Exits without saving the settings for the dialog box.
- **Help**: Displays the Seehau Help file.
- **Refresh**: Allows you to retrieve and view the current trace and emulator hardware configuration settings.

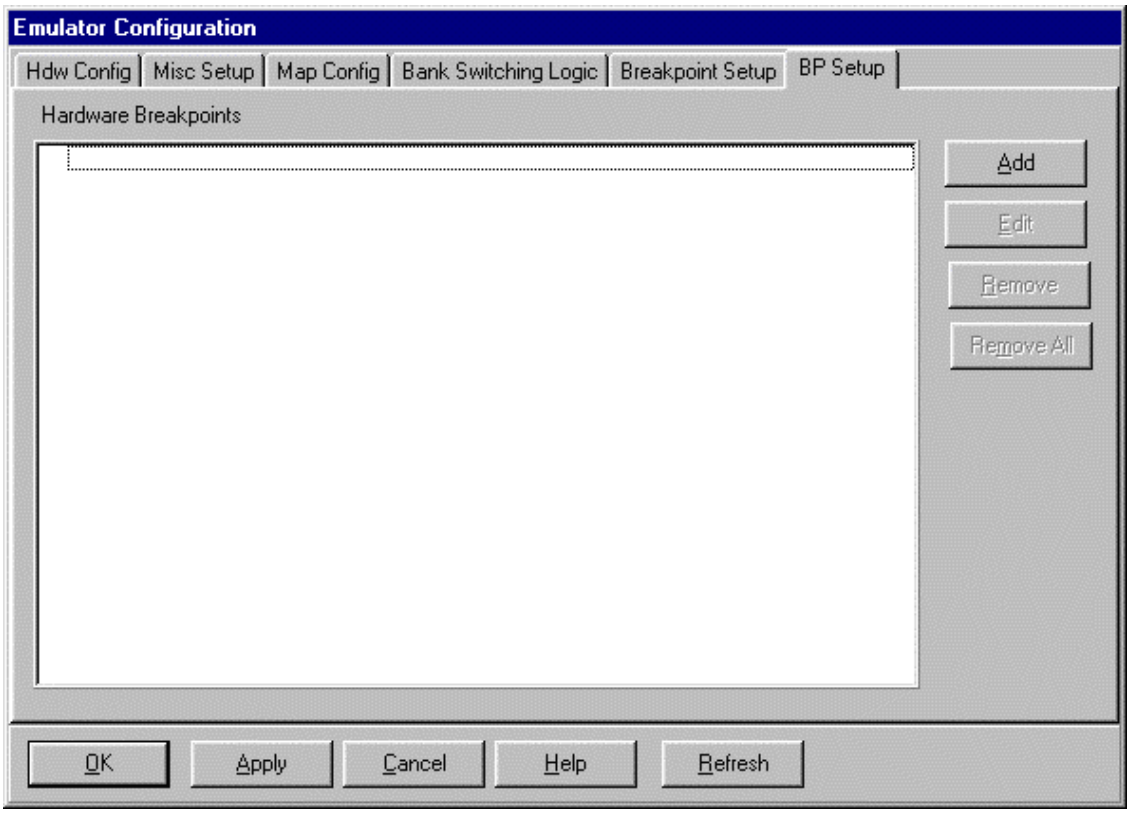


Figure 127. BP Setup Tab

BP Setup Tab

The **BP Setup** tab (Figure 127) configures hardware breakpoint ranges.

- **Hardware Breakpoints** display area: displays the address ranges that are enabled for hardware breakpoints. These breakpoints use the cycle types that are declared under the **Breakpoint Setup** tab.
- **Add**: Opens the **Add Address Range** dialog box (Figure 128). Entries can be done with hex address or symbolically.
- **Edit**: Select an address range in the **Hardware Breakpoints** display area, then click **Edit**. The **Add Address Range** dialog box opens. Make any changes to the values, then click **OK**.
- **Remove**: To remove an address range, select the address range in the **Hardware Breakpoints** display area, then click **Remove**.
- **OK**: Saves the settings for the tabs and exits the dialog box.
- **Apply**: Saves the settings for this tab.
- **Cancel**: Exits without saving the settings for the **BP Setup** dialog box.
- **Help**: Displays the Seehau Help file.
- **Refresh**: Allows you to retrieve and view the current trace and emulator hardware configuration settings.

Add Address Range Dialog Box

- **Begin**: Enter the first address of the range.
- **End**: Enter the last address of the range.
- **OK**: Saves the settings for the tabs and exits the dialog box.
- **Cancel**: Exits without saving the settings for the **Add Address Range** dialog box.
- **Help**: Displays the Seehau Help file.

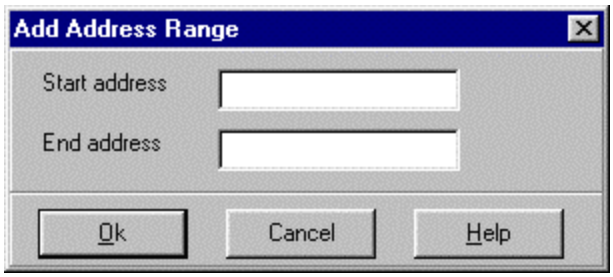


Figure 128. Add Address Range Dialog Box

5 Connecting the Emulator to Your Target Board

WARNING

To avoid damage to the pod or to your target, do not connect the pod to your target when the pod or target power is on.

Make sure you always power up the emulator first followed by the target system. When powering down, power down the target system first followed by the emulator. Failing to do so can cause damage to your target and/or emulator.

Power-up / Power-down Sequences

Use the internal position as your first choice for power selection even if the pod is connected to your target system. With the power jumpers set to the internal position, use the following recommended power sequence:

1. Power ON the PC (or HSP/USB box option).
2. Power ON the target.
3. Use emulator for the session.
4. Power OFF the target.
5. Power OFF the PC (or HSP/USB box option).

Selecting external (target) power is usually not necessary, unless you are emulating a target in which the voltage is less than 5 volts. If you want to determine how much current the MCU draws, you can use an actual part in the circuit instead of measuring with the pod. If you require external power, use the following power-up/power-down sequence:

1. Power ON the target.
2. Power ON the PC (or HSP/USB box option).
3. Use the emulator for session.
4. Power OFF the PC (or HSP/USB box option).
5. Power OFF the target.

Note

The ideal solution when selecting external power is to have both the PC or the HSP/USB box and the target system on the same power switch so that the power to both is turned on and off at the same time.

Plugging the Emulator into Your Target Board

Installing the DIP / PLCC Socket Adapters

Do the following steps:

1. Make sure the power to the target system and the emulator host PC or HSP/USB is turned off.
2. Connect the pod ground clip to the target ground.
3. Make sure the adapter is correctly oriented (pin 1 of the target processor to pin 1 of the pod adapter). For details about correct adapter orientation, see the following “Verifying Adapter Orientation” section.
4. Plug the adapter into the target socket.

Installing PGA Socket Adapters

Since most targets use PLCC sockets that are soldered into a PGA footprint, you can install a PGA socket that allows PGA pods to mate directly to the target. A PLCC socket plugs directly into the PGA socket for running the target with the MCU instead of the emulator.

Installing Surface Mount Adapters

First make sure you have a qualified technician solder the surface mount portion of the adapter to your target. Make sure you check the soldering for opens and shorts under a microscope, as these conditions are hard to detect otherwise.

Then do the following steps:

1. Make sure the power to the target system and the emulator host PC or HSP/USB is turned off.
2. Connect the pod ground clip to the target ground.
3. Verify that pin 1 of the target processor is connected to pin 1 of the pod adapter. For details about correct adapter orientation, see the following “Verifying Adapter Orientation” section.
4. Plug the adapter into the surface mount portion of the target side adapter.

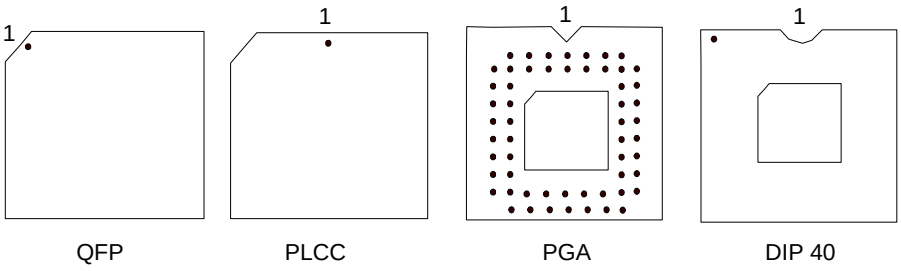


Figure 129. Pin Variations

Verifying Adapter Orientation

WARNING

Do not apply power to your system unless you are absolutely sure the target adapter is correctly oriented. Failure to do so can cause damage to your target and/or emulator.

Usually pin 1 is clearly marked on the pod, target adapter, and target. If not, you need to manually verify that the pod adapter and target are correctly oriented. To verify proper connection, you will need an Ohm meter to check the status of one or two signals.

To verify proper connection, do the following steps:

1. Make sure all power is off.
2. Connect the adapter to the pod. Pin 1 can be marked in various ways. For example, you might see a beveled corner, the number one, or a V-shaped notch (Figure 129).
3. Connect the adapter to the target.
4. Choose an easily accessible signal such as the XTAL signal. Then connect one side of the Ohm meter to the target signal and one side to the equivalent pin on the pod target header.

Tips for Avoiding Common Adapter Problems

- Select the correct socket. Select a quality socket that can withstand repeated use. Some inexpensive sockets tend to wear out quickly causing potential intermittant problems.
- Check the condition of the adapter and socket.
  - Make sure the adapter and socket are free of oxidation and corrosion.
  - Make sure the connector pins are not bent out of place. They should appear parallel to one another (Figure 130).
  - If you are using a PLCC socket, make sure the pins have not flattened out. This tends to occur after repeated adapter insertions (Figure 131).

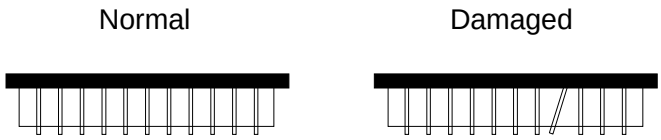


Figure 130. Connector Pins

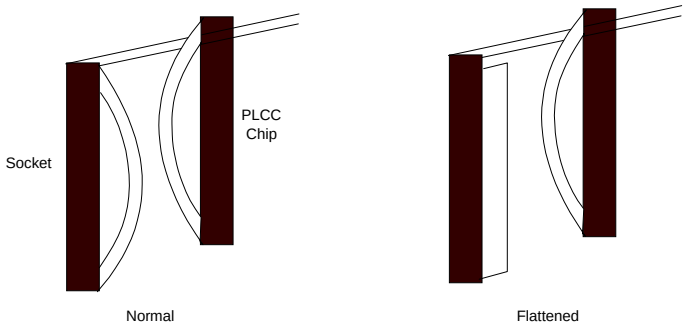


Figure 131. PLCC Socket Pins

> **Note**
>
> Usually surface mount PLCC sockets tend to wear out more quickly. Therefore, you need to inspect these sockets more frequently.

- Use the minimal amount of space required to connect to the target system. For example, avoid stacking sockets to meet height requirements. When necessary, use one piece extender adapters or short, flexible ribbon cables. (Refer to the *EMUL51–PC Price List* for further information about these items.)

Setting Up Proper Pod Support

In most cases, the pod will require additional support on the ribbon cable side to prevent the adapter from being pulled out of the socket.

Configuring Jumpers for Target Operation

Clock Jumpers

Each pod contains two clock jumpers or switches. These should be set to the external or target side position. Setting the jumpers or switches to this position ensures the following:

- The emulator will run at the target clock speed.
- The baud rate generator will have the same crystal reference.
- The periodic time-based interrupt will occur at the correct intervals.

Power Jumper

Set this jumper to the external position if you are connecting to a 3-volt target.

Control Signal Jumpers

Most pods have a gated PSEN signal; however, some pods might have a gated RD/WR signal also. These gates prevent the signals from going out to the target unless an access to target memory occurs. In some cases, target logic might require one or more of these signals to be present at all times. When this occurs, move the appropriate jumpers to the non-gated position.

CAUTION

When a control signal is placed in a non-gated condition, that signal will always be sent to the target and might cause bus collisions or unintentional writes to target data memory. To avoid these problems, make sure you review your control logic carefully.

Configuring Seehau for Target Operation

Emulator Configuration Tabs

After you connect the emulator to your target, review the following **Emulator Configuration** tabs and enter changes as appropriate. From the Seehau main menu, click the **Config** menu. Then click **Emulator**. The **Emulator Configuration** dialog box opens.

Hdw Config Tab

Enter your target clock frequency in the **Clock (MHz)** field. This field contains the time base reference for the trace timestamp only. The timestamp calculation assumes standard 8051 timing with 12 clocks per bus cycle. Enter the following:

- If you are using a standard part, enter the crystal/oscillator frequency.
- If you are using a Dallas 520/530/320/323 part, enter 1.5x the crystal/oscillator frequency.
- If you are using a Philips or Temic X2 part, enter 2x the crystal/oscillator frequency.
- If you are using an MX pod, enter two times the external frequency.

Misc Setup Tab

- Select the **Program Counter** option if you want to override the reset vector. Enter the value that you want to preload into the program counter. (optional)
- Select the **Stack Pointer** option if you want to preload the stack pointer with a designated value. (optional)

Map Config Tab

By default, all memory is mapped to the emulator. Select memory ranges to access target resources such as ROM, RAM, or memory mapped I/O.

Bank Switching Logic Tab

If your target uses bank switching, make sure you configure this tab prior to emulation. (Refer to Chapter 4, “Installing and Configuring the Seehau Software” in this guide for details about the **Bank Switching Logic** tab.) Also verify the following:

- Set the jumpers on the emulator board correctly.
- Check that bank switch wires on the pod are set up correctly.

Executing Code from Target ROM

If you are executing from target ROM, you can perform source level debug by loading symbols only. Do the following:

1. From the Seehau main menu bar, click the **Config** menu.
2. Click **Environment**.
3. From the **Preferences** tab, clear the **Code** option. Then click **OK**.
4. From the Seehau main menu bar, click the **File** menu. Then click **Load Code**. Seehau will now load only the debug symbol information from the selected file.

6 Introduction to Tracing

Overview of Tracing

Trace is a comprehensive tool used to analyze the MCU environment. While using the trace feature, you can passively analyze user code in real time as the code continues to execute. Without the trace tool, the emulator only provides a snapshot of the MCU environment.

The trace feature provides the same basic capabilities as a logic analyzer. Each trace frame is time stamped and records the following fields: address and data values, cycle type, and up to 18 external inputs. Based on various combinations of these fields, you can set up trigger and/or filter conditions to control the trace recording.

By default, trace automatically starts recording when you begin user code execution. When you stop code execution, the trace history is displayed automatically.

Tracing allows you to perform many tasks, including:

- Detecting an error condition
- Analyzing a history of the sequence of events leading to an error
- Characterizing code behavior using tools such as Program Performance Analysis (PPA) and Code Coverage (available only with Enhanced Trace)
- Sampling time measurements (ATR/ERT only)
- Analyzing peripheral I/O

The Seehau software provides a symbolic interface for both the trace setup and display.

Basic Features of Tracing

- **Trigger**: An event that stops trace buffer recording by setting the zero frame reference in the trace buffer.
- **Delay**: The number of trace frames collected after a trigger event occurs.
- **Filter**: A set of conditions that determine which frames are allowed into the trace buffer.
- **Time Stamp**: A feature that displays the number of machine cycles that have elapsed since the beginning of program execution. (ATR/ERT only)

Trace Window

The contents of the trace buffer are displayed in the Trace window (Figures 132 and 133). To open a Trace window, use the TR button on the toolbar, or select the **New** menu and click **Trace**.

The **Trace** menu controls most of the Trace window features.

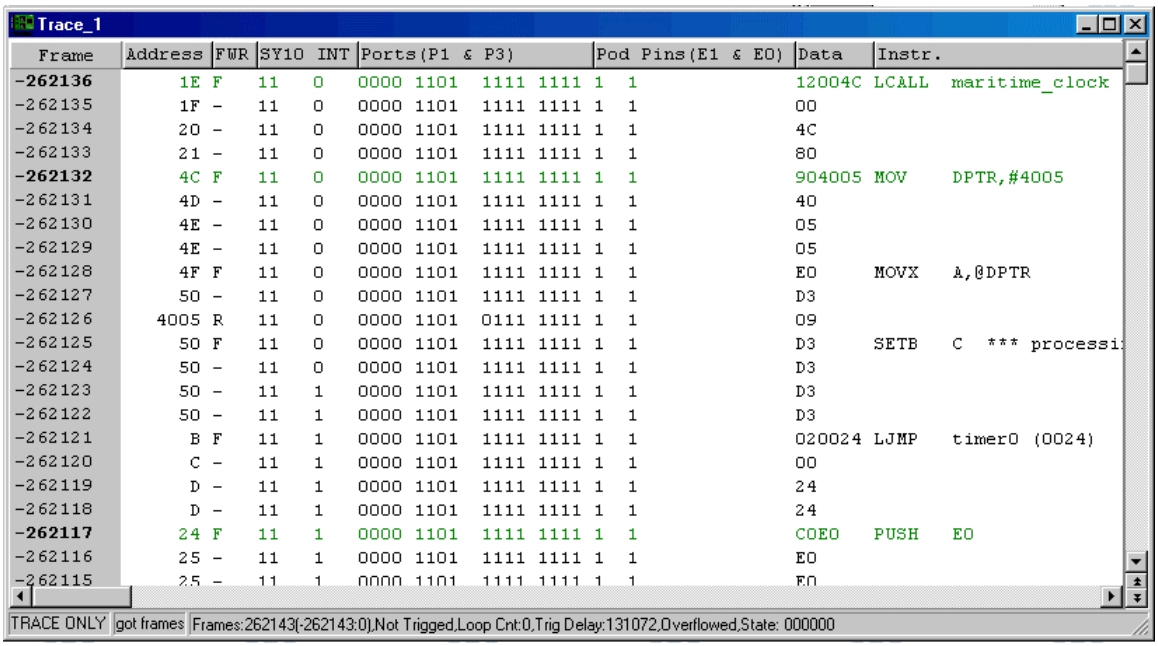


Figure 132. Trace Window (non-compressed)

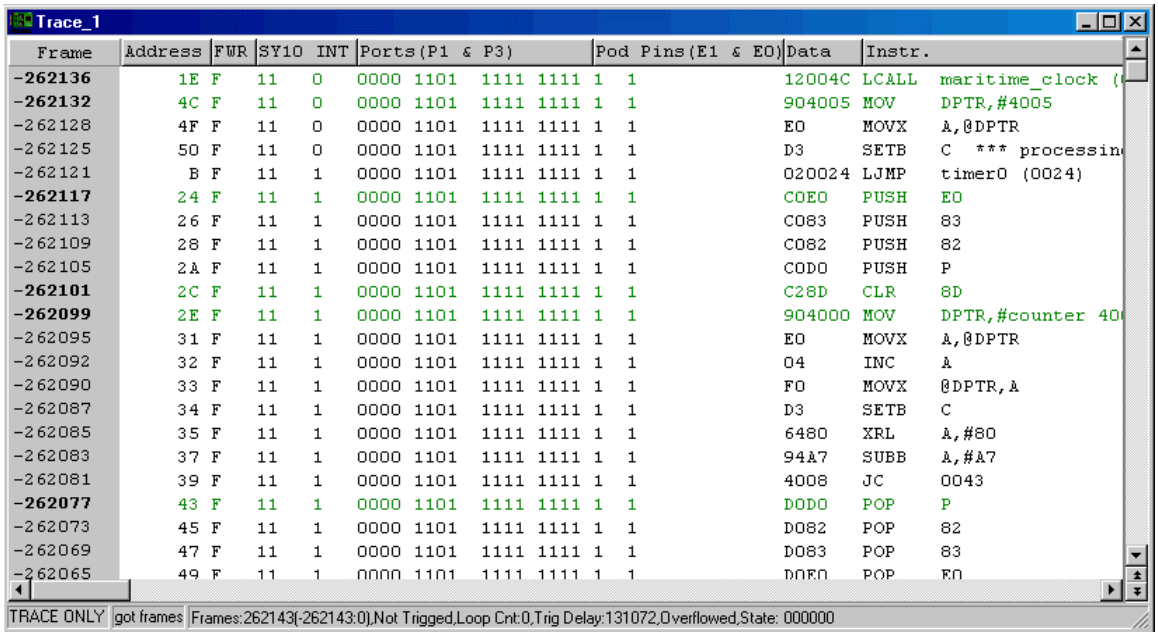


Figure 133. Trace Window (compressed)

Trace Window Columns

- **Frame Number** is on the far left of the window. Frame 0 always represents the trigger frame. If there is no trigger, frame 0 is the last frame in the buffer.
- **Address** displays the address of the bus cycle in hexadecimal notation.
- **FWR** shows the cycle type: first byte of Fetch, Write, Read, or other frames (shown with a dash).
- **SY0 & SY1** display status of the SY0 and SY1 inputs from the pod board.
- **Port1 & Port3** display the user-selected ports in binary format.
- **Time Stamp** displays one of the following options based on what you choose in the **Trace Display** options local menu:
  - **Relative Time**: Amount of elapsed time for the current instruction.
  - **Absolute Time**: Amount of elapsed time since the beginning of execution.
  - **Relative Cycle**: Number of CPU cycles for the current instruction.
  - **Absolute Cycle**: Total number of CPU cycles since the beginning of execution.
- **Data** displays bytes of data from the displayed bus cycle.
- **Instr.** shows the instruction disassembly.
- **Symbol** shows any symbolic label that refers to a specific address.

Local Trace Menu

To access the **Local Trace** menu, right click the Trace window or select the **Trace** menu which appears in the main menu bar only when the Trace window is active (Figure 134).

- **Go to Frame number**: Opens a dialog box where you can enter a specific frame number for display.
- **Find Trigger Point**: Displays the trigger point (frame zero).
- **Zero Time at Cursor**: Changes the timestamp at the selected frame to zero and makes all other timestamps relative to the selected frame.
- **Synchronize Source Window**: Automatically aligns the display of code in the Source window as you scroll through the Assembly code in the trace buffer. You must use the up/down arrow keys in conjunction with this feature.

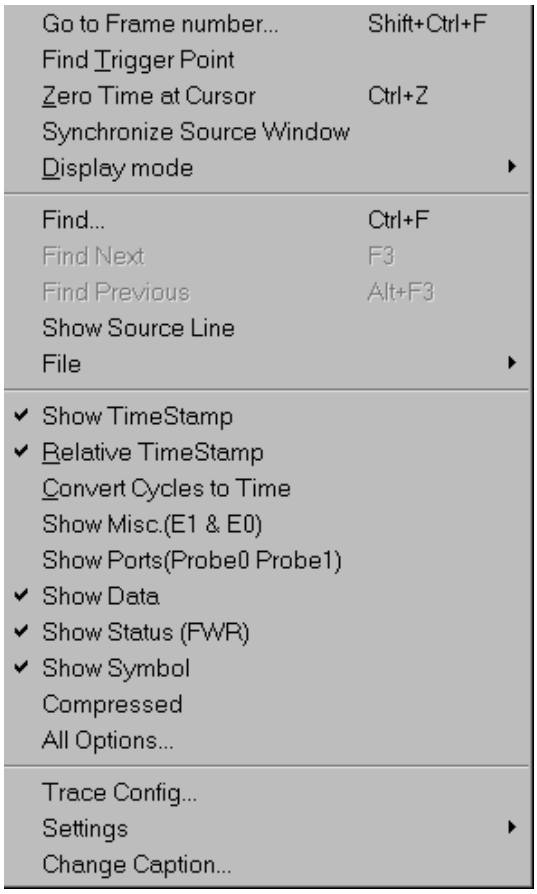


Figure 134. Local Trace Menu

- **Display mode**: Opens a submenu that allows you to select which code to display:
  - **Trace Only**: Displays Assembly code only in the trace buffer.
  - **Mixed**: Displays both C and Assembly code in the trace buffer.
  - **Source Only**: Displays C source code only in the trace buffer.

- **Find**: Opens the **Find Address** dialog box where you can search the trace buffer for an address or a range of addresses and a cycle type.

- **Find Next**: Click to find the next frame match.

- **Find Previous**: Click to find the previous frame match.

- **Show Source Line**: Displays the associated source code for the current frame. The frame address must match a source line address.

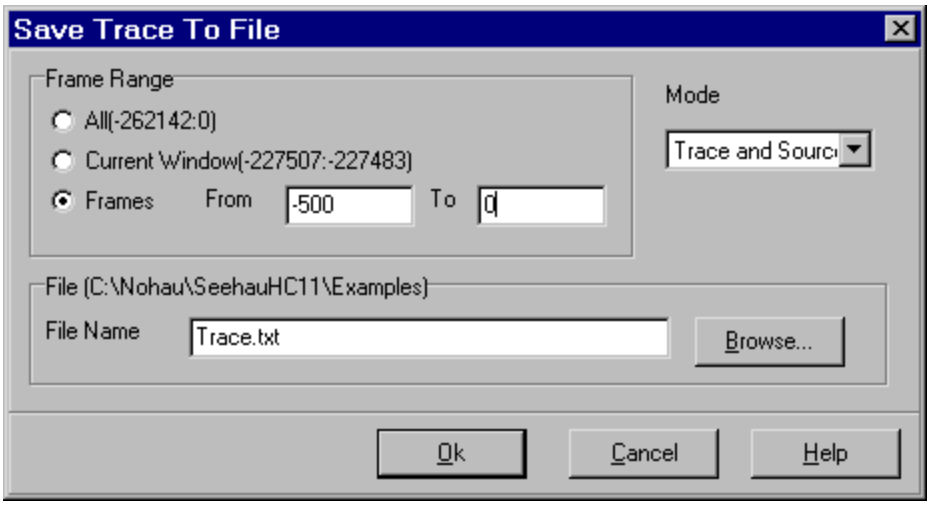


Figure 135. Save Trace to File Dialog Box

- **File**:
  - **Save to File**: Opens the **Save Trace To File** dialog box where you can save the contents of the trace buffer as text to a file (Figure 135).
    Frame Range: Select one of the following:
    All: Saves the entire trace buffer
    Current Window: Saves only the contents of the current Trace window
    Frames: Allows you to specify a range of frames
  - **Print**: Allows you to send the trace buffer to a printer.

> **Note**
>
> The following menu items can be toggled individually or can be configured all at once. Click **All Options** to open the **Display Options** dialog box.

- **Show TimeStamp**: Displays the timestamp which represents the number of machine cycles that have elapsed since the beginning of program execution.
- **Relative TimeStamp**: Displays the timestamp as the number of machine cycles that have elapsed since the execution of the previous instruction.
- **Convert Cycles to Time**: Converts the timestamp from machine cycles to actual time based on the MCU clock (uP clock).
- **Show Misc.(SY0 & SY1)**: Displays status of the SY0 and SY1 inputs from the pod board.
- **Show Ports (Port1 Port3)**: Displays the user-selected ports in binary format.
- **Show Data**: Displays the data field.

- **Show Status (FWR)**: Displays bus cycle type (Fetch, Write, or Read).
- **Show Symbol**: Displays symbolic labels associated with the address field.
- **Compressed**: Displays Assembly code only. (**Note**: RD/WR cycles are not shown.)
- **All Options**: Opens the Display Options window where you can select or clear trace options in a single update.
- **Trace Config**: Opens the **Trace Configuration** dialog box. Refer to Chapter 7, "Configuring Trace Boards" in this guide.
- **Settings**: Opens a submenu that allows you to set up Trace window attributes.
- **Change Caption**: Allows you to change the Trace window caption in the title bar.

7 Configuring Trace Boards

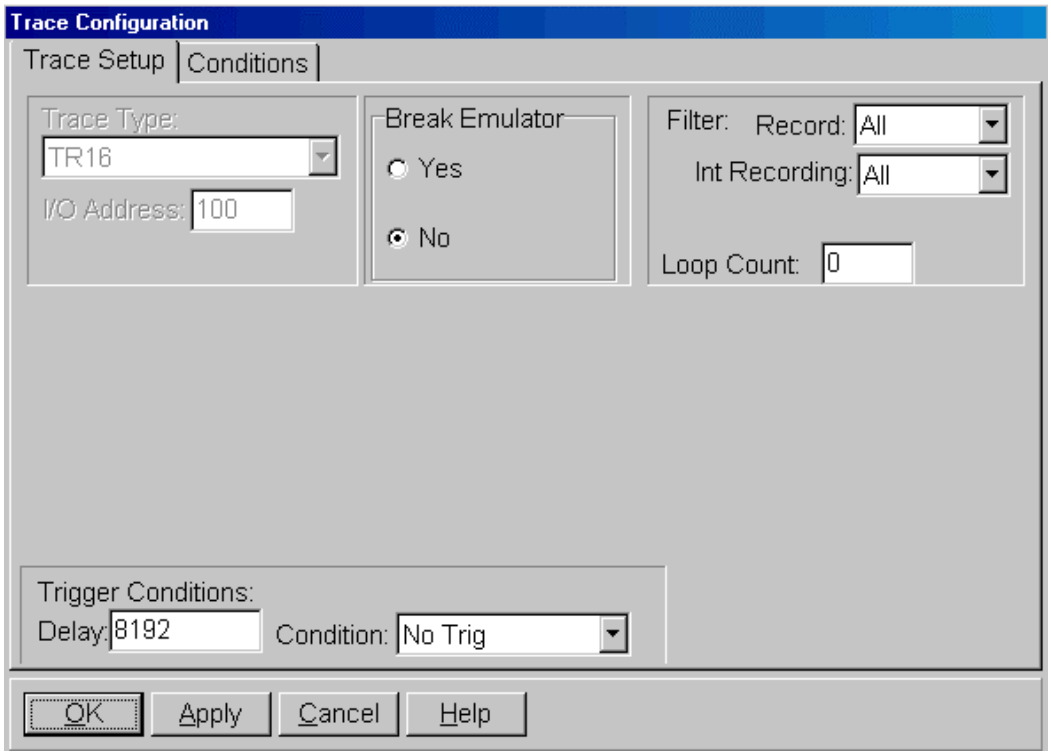


Figure 136. Standard Trace Configuration Dialog Box (Trace Setup)

To program the trace board for specific tasks, use the **Trace Configuration** dialog box (Figure 136). To open the dialog box, do the one of the following:

- From the **Trace** menu, click **Trace Config**
- From the **Config** menu, click **Trace**

Configuring the Standard Trace Board

After the **Trace Configuration** dialog box opens, click **Trace Setup**. Enter or select appropriate data for the following **Trace Setup** fields:

Trace Setup Tab

- **Trace Type**: Displayed for reference only. Exit Seehau and run Seehau Config to change this setting.
- **I/O Address**: Displayed for reference only. Exit Seehau and run Seehau Config to change this setting.

- **Trigger Conditions**
  - **Loop Count**: The number of times the trigger condition must be met before the trigger action is taken.
  - **Delay**: The number of frames recorded after the trigger condition is met.
  - **Condition**:
    - **No Trig**: Disables triggering. Trace collects data continuously in a FIFO buffer.
    - **Trig On A**: Triggers when the A condition is met.
    - **Trig On B**: Triggers when the B condition is met.
    - **A then B**: A sequential trigger where A condition must be met first, followed by the B condition.
    - **A Loop**: Triggers after A condition is met ***n*** times, where ***n*** is the loop count field.
    - **B Loop**: Triggers after B condition is met ***n*** times, where ***n*** is the loop count field.
    - **A Loop then B**: A sequential trigger, where A condition must be met ***n*** times, followed by the B condition.
    - **(A then B) loop**: A sequential trigger, where A condition must be met first, followed by the B condition ***n*** times.

- **Break Emulator**
  - **No**: Emulation does not stop as a result of trace trigger conditions.
  - **On Trig**: Emulation stops when the trigger condition is met.
  - **On B**: Emulation stops when the B condition is met.

- **Filter**: The **Filter** field controls which frames are recorded by the trace board.
  - **Record**: Click the down arrow and select one of the following options:
    - **All**: All frames are captured. There is no filtering.
    - **A**: Only frames which match the A condition are recorded.
    - **B**: Only frames which match the B condition are recorded.
    - **A & B**: Frames which meet both A and the B conditions are captured.
    - **A On, B Off**: When the A condition is met, the trace is turned on and all frames are recorded. When the B condition is met, the trace is turned off.

  - **Int Recording**: Controls which frames are recorded by the trace board. Click the down arrow and select one of the following options:
    - **INT**: Causes the trace to record the code that the interrupt level is greater than 0. Code that is part of any interrupt service routine will be recorded.
    - **Main**: Causes the trace to record the code that the interrupt level is equal to 0. Code that is not part of any interrupt service routine will be recorded.

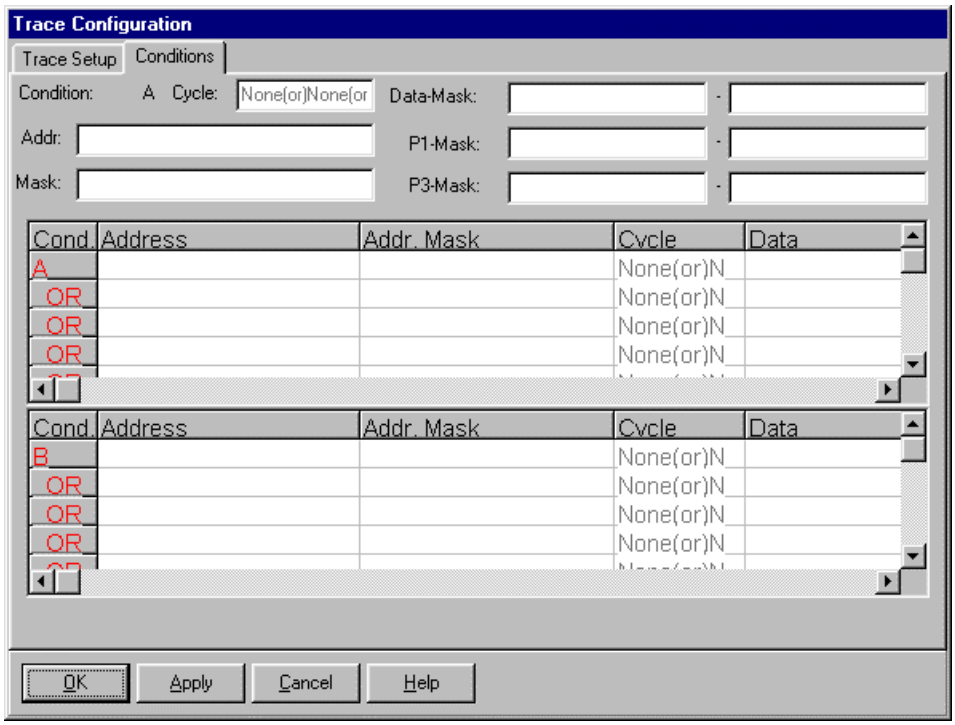


Figure 137. Standard Trace Configuration Dialog Box (Conditions Tab)

Conditions Tab

Right-click in the Conditions window to access the **Add**, **Edit**, or **Remove** menu. Select **Edit** from this menu, and the **Edit Data Condition** dialog box opens. (Figure 138).

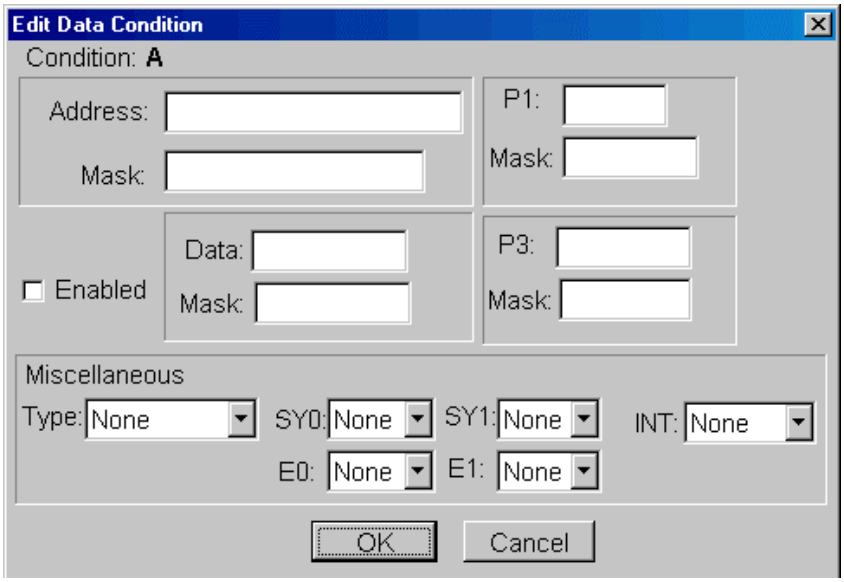


Figure 138. Edit Data Condition Dialog Box

Edit Data Condition Dialog Box

- **Enabled**: Select to enable condition.

> **Note**
>
> If left blank, the following fields are evaluated as Don't Cares.

- **Address**: Enter an address or range of addresses.
  - 8000 (Single address)
  - 8000:8FFF (Range from 8000 to 8FFF)
  - timer (Value of variable timer)
  - &timer (Sddress of variable timer)
  - timer:timer+10 (Range from the value of timer to value of timer +10)

- **Address Mask**: Enter an address mask. The mask is used as an alternate method to create address ranges. The address mask value is logically ANDed with the address value. A zero in the result corresponds to a Don't Care in the address value. For example:
  Address=ABCDH
  Mask=FF0FH
  Resulting range=ABxDH

- **P1**: Enter an 8-bit value or range corresponding to the port that is compared to the TTL level data being traced by P1. On most pods P1 is connected to Port 1 by jumpers. By removing the jumpers and using wires, any eight external signals can be traced as P1. Purchase optional E-Z hooks from Nohau (Part #EMUL-PC/E-Z).

- **P1 Mask**: Enter a P1 mask. The mask is used as an alternate method to create ranges. The mask value is logically ANDed with the P1 value. A zero in the result corresponds to a Don't Care in the value. For example:
  P1=A5H
  Mask=F0H
  Resulting range=AxH

- **Data**: Enter an 8-bit (data bus) value or range.

- **Data Mask**: Enter a data mask. The mask is used as an alternate method to create ranges. The mask value is logically ANDed with the data value. A zero in the result corresponds to a Don't Care in the value. For example:
  Data=A5H
  Mask=F0H
  Resulting range=AxH

- **P3**: Enter an 8-bit value or range corresponding to the port that is compared to the TTL level data being traced by P3. On most pods P3 is connected to Port 3 by jumpers. By using wires, any eight external signals can be traced as P3.

- **P3 Mask**: Enter a P3 mask. The mask is used as an alternate method to create ranges. The mask value is logically ANDed with the P1 value. A zero in the result corresponds to a Don't Care in the value. For example:
  P3=A5H
  Mask=F0H
  Resulting range=AxH

- **Miscellaneous**

  — **Type**: Select one of the following bus cycle types:
    - **None**: Don't Care
    - **Read**: A data read
    - **Write**: A data write
    - **R/W**: A data read or write
    - **Fetch**: A code fetch.

  — **E0**: The E0 external input pin on the pod. Select one of the following levels:
    - **High**
    - **Low**
    - **None**: Don't Care

  — **E1**: The E1 external input pin on the pod. Select one of the following levels:
    - **High**
    - **Low**
    - **None**: Don't Care

> **Note**
>
> Signals E0 and E1 are level triggered.

  — **SY0**: The SY0 external input pin on the pod. Select one of the following levels:
    - **High**
    - **Low**
    - **None**: Don't Care

  — **SY1**: The SY1 external input pin on the pod. Select one of the following levels:
    - **High**
    - **Low**
    - **None**: Don't Care

Configuring a Trigger Condition

The following example will trigger when the value of 0FH is written to the symbol **timer.min**. The trace will then force a break when the trigger occurs.

1. Open the Trace Configuration window by doing one of the following: from the **Config** menu, click **Trace**. Or from the **Trace** menu, click **Trace Config**.

2. In the **Trig Exp** text box, enter A for the condition. Then select the check box to enable the trigger expression.

3. In the Break Emulator area, select **On Trig**. (**On Trig** breaks immediately. Alternatively, you can choose **When Done** which breaks after the number of delay cycles are collected.)

4. From the Condition A line, right-click and select **Edit**. The **Edit Data Condition** dialog box opens (Figure 139).

5. Complete the following fields:

   - **Enabled check box**: Automatically selected when you enter an address in the **Address** text box.
   - **Address**: Enter &timer.min. (The **&** denotes the address of the symbol rather than the value of the symbol.)
   - **Mask**: Enter FFFF. (A one in any bit position denotes that the value specified in the address field must match the address field in the trace frame. Whereas, a zero denotes a Don't Care bit.)
   - **Data**: Enter FH.
   - **Mask**: Enter FF. (A one in any bit position denotes that the value specified in the address field must match the address field in the trace frame. Whereas, a zero denotes a Don't Care bit.)
   - **Miscellaneous Type**: Select **Write** from the pull-down menu.

6. Click **OK**.

7. Click **OK** in the Trace Configuration window (Figure 139).

8. Click the Reset and Go button on the Speedbar. The Trace Display window will now update.

(For addition trace examples, refer to the "Trace Configuration Examples" section.)

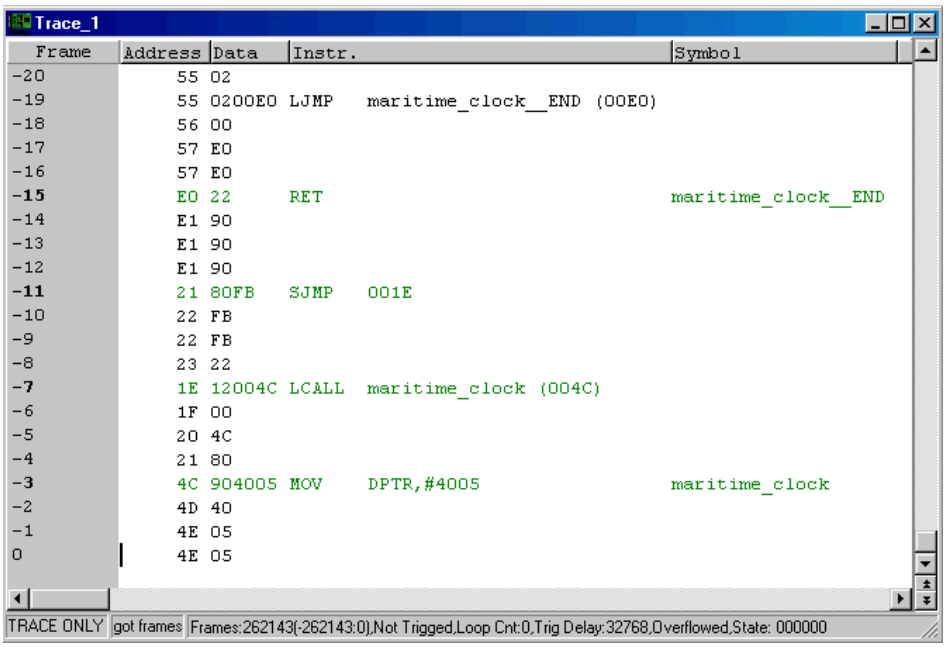


Figure 139. Trace Display Window

Configuring the Enhanced Trace Board

After the **Trace Configuration** dialog box opens, click **Trace Setup**. Enter or select appropriate data for the following fields:

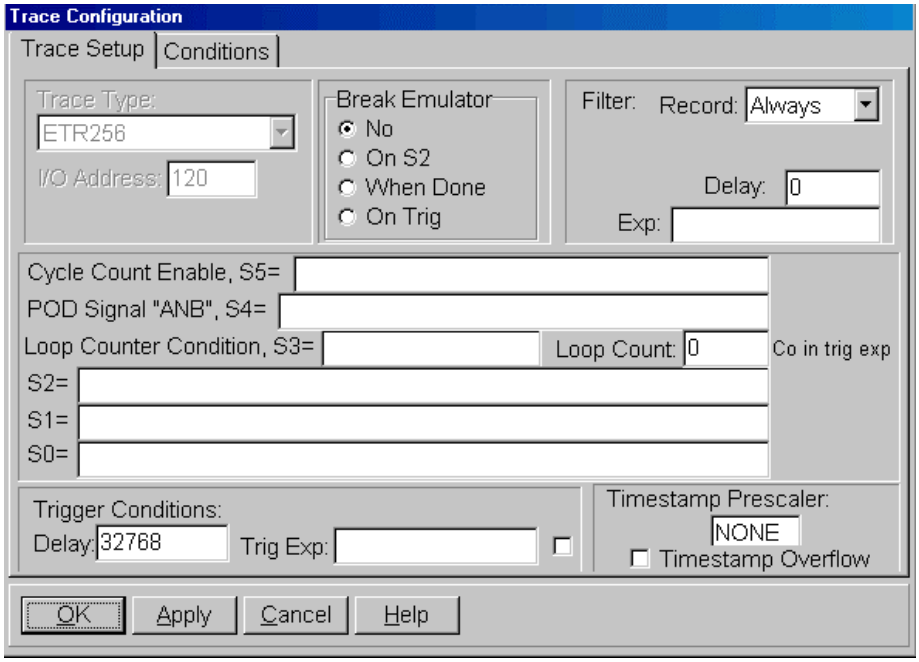


Figure 140. Enhanced Trace Configuration Dialog Box (Trace Setup)

Trace Setup Tab

- **Trace Type**: Displayed for reference only. Exit Seehau. From the Start menu, run Seehau Config to change this setting.

- **I/O Address**: Displayed for reference only. Exit Seehau. From the Start menu, run Seehau Config to change this setting.

- **Break Emulator**
  - **No**: Emulation does not stop as a result of trace conditions.
  - **On S2**: Emulation stops when S2 is *true*.
  - **When Done**: Emulation stops after the trigger condition is met and the delay counter reaches zero.
  - **On Trig**: Emulation stops when the trigger condition is met.

- **Filter**: The **Filter** field controls which frames are recorded by the trace board.
  - **Record**: Click the down arrow and select one of the following options:
    - **Always**: If **Always** is selected, there is no filtering of the trace capture; therefore, all frames will be recorded.
    - **IF**: If a trace frame matches **Exp**, then the frame is recorded.
  - **Delay**: The number of frames collected after each match of the filter condition. This feature is most useful when filtering data/port reads and writes. Allows you to determine where in your code the read or write occurred.
  - **Exp**: (See Trig Exp)

- **Cycle Count Enable, S5=**: Consists of a Boolean expression of events, S0 to S5 and C0. When the expression is *true*, a 32-bit counter is enabled to count cycles. This is used to measure cycles or time without stopping emulation. These numbers are displayed on the toolbar of the Seehau user interface. On pods that have the pin labeled FLF, the S5 bit is logically out-put here. This signal can also be found on pin 22 of the 50-pin ribbon cable connector.

- **Pod Signal "ANB", S4=**: Consists of a Boolean expression of events, S0 to S5 and/or qualifiers A - H that are used to describe how the trace output signal will operate. When the condition goes from *false* to *true*, the loop counter will shut down. This output is available on the pod board with the pin labled ***ANB***.

- **Loop Counter Condition, S3=**: Consists of a Boolean expression of events, S0 to S5 and/ or qualifiers A – H that are used to describe how the loop counter will operate. When the condition goes from *false* to *true*, the loop counter will shut down.

- **Loop Count**: The number of times the trigger condition must be met before the trigger action is taken.

- **S0, S1, S2**: Used only if the THEN operator is not being used in the **Trig** or **Record** fields. Otherwise, these fields can be used freely for Boolean equations.

- **Trigger Conditions**
  - **Delay**: The number of frames recorded after the trigger condition is met.
  - **Trig Exp**: Select this option to enable the trigger. Clear this option to disable the trigger.

  In the **Trig Exp** text box, enter a Boolean expression that uses the qualifying registers A – H, S0 – S5 and the output of the loop counter CO. The Boolean express can also contain the THEN operator.

  If the THEN operator is used the state bits S0, S1 and S2 can not be used. These bits are being used to generate the THEN condition.

  Examples
  A
  NOT A
  A OR B
  A AND B
  A THEN B
  A AND NOT B
  A THEN B THEN C THEN D THEN E THEN F THEN G THEN H
  A OR B THEN C AND NOT D OR E
  A OR S3
  A AND CO
  A THEN B OR S3
  A THEN B THEN CO AND C

- **Timestamp Prescaler**
  - **Advanced Trace**: Consists of one 32-bit hardware counter with a 16-bit prescaler. Only the 32-bit counter is recorded in the trace buffer. If the prescaler is none, the result is full resolution in which each frame represents a full bus cycle. Therefore, 4,294,967,294 cycles can be counted before the counter rolls over to 0.
  - **Enhanced Trace**: Consists of one 64-bit hardware counter with a 16-bit prescaler. Only the 64-bit counter is recorded in the trace buffer. If the prescaler is none, the result is full resolution in which each frame represents a full bus cycle. Therefore, 8,589,934,588 cycles can be counted before the counter rolls over to 0.
  - With the prescaler set to 2, the time is doubled. By programming the prescaler to a high enough number, the time of overflow time can be in minutes, hours, days, and so on.

- **Timestamp Overflow**: By checking the box to the left of **Timestamp Overflow**, the roll-over of the counter can be forced in the trace buffer.

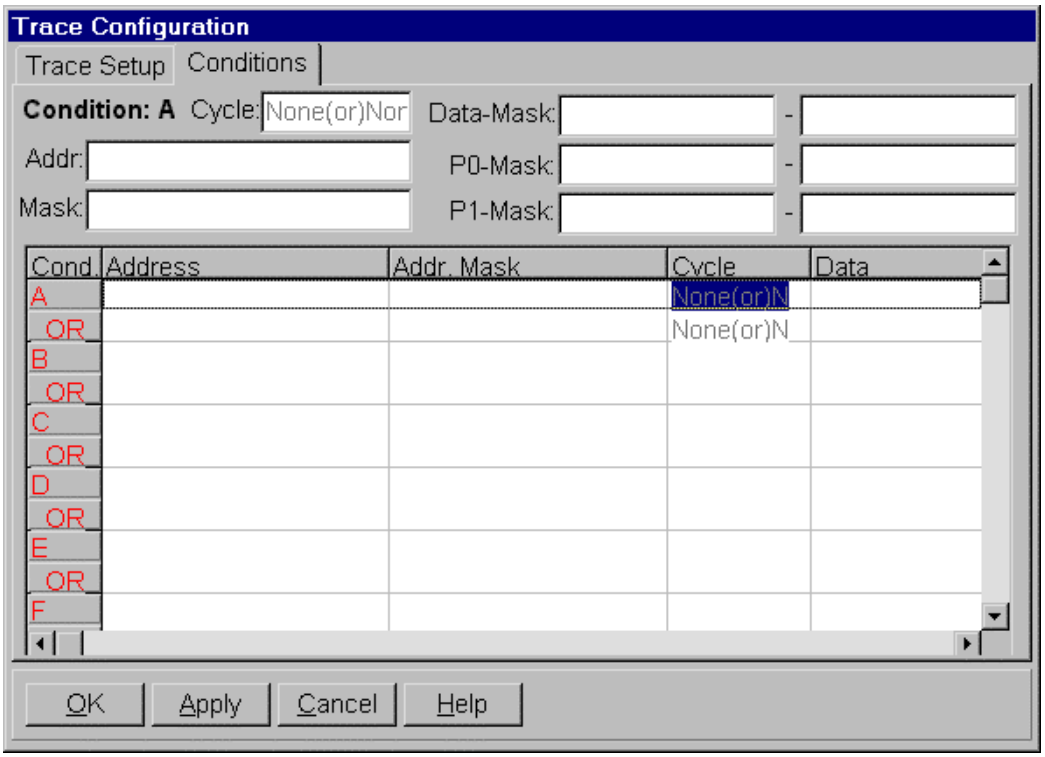


Figure 141. Enhanced Trace Configuration Dialog Box (Conditions Tab)

Conditions Tab (ETR)

Right click in the Conditions window to access the **Edit** or **Remove** menu. Select **Edit** from this menu, and the **Edit Data Conditions** dialog box opens.

- **Enabled**: Select to enable condition.

> **Note**
>
> If left blank, the following fields are evaluated as Don't Cares.

- **Address**: Enter an address or range of addresses. (See Appendix A for correct syntax.)
- **Address Mask**: Enter an address mask. The mask is used as an alternate method to create address ranges. The address mask value is logically ANDed with the address value. A zero in the result corresponds to a Don't Care in the address value. For example:
  Address=ABCDH
  Mask=FF0FH
  Resulting range=AbxDH
- **P1 Mask**: Enter a P1 mask. The mask is used as an alternate method to create ranges. The mask value is logically ANDed with the P1 value. A zero in the result corresponds to a Don't Care in the value. For example:
  P1=A5H
  Mask=F0H
  Resulting range=AxH

- **Data**: Enter an 8-bit (data bus) value or range.
- **Data Mask**: Enter a data mask. The mask is used as an alternate method to create ranges. The mask value is logically ANDed with the data value. A zero in the result corresponds to a Don't Care in the value. For example:
  Data=A5H
  Mask=F0H
  Resulting range=AxH
- **P3**: Enter an 8-bit value or range corresponding to the port that is compared to the TTL level data being traced by P3. On most pods P3 is connected to Port 3 by jumpers. By using wires, any eight external signals can be traced as P3.
- **P3 Mask**: Enter a P3 mask. The mask is used as an alternate method to create ranges. The mask value is logically ANDed with the P3 value. A zero in the result corresponds to a Don't Care in the value. For example:
  P3=A5H
  Mask=F0H
  Resulting range=AxH
- **Miscellaneous (ETR)**
  - **Type**: Select one of the following bus cycle types:
    - **None**: Don't Care.
    - **Read**: A data read.
    - **Write**: A data write.
    - **R/W**: A data read or write.
    - **Fetch**: A code fetch.
  - **E0**: External input pin on the pod. Select one of the following levels:
    - **High**
    - **Low**
    - **None**: Don't Care
  - **E1**: External input pin on the pod. Select one of the following levels:
    - **High**
    - **Low**
    - **None**: Don't Care
  - **SY0**: External input pin on the pod. Select one of the following levels:
    - **High**
    - **Low**
    - **None**: Don't Care
  - **SY1**: External input pin on the pod. Select one of the following levels:
    - **High**
    - **Low**
    - **None**: Don't Care

Trace Configuration Examples

Trigger Example

This section describes how to set up the emulator trace to trigger on a specific condition with the following parameters:

- Set the trace to trigger on the function Main.
- Collect 8K bus cycles after the trigger.
- Allow the emulator to continue running after the trigger.

From the **File** menu, select **Load Code**. Double-click the file Timer.d07. Then do the following steps:

1. From the **Config** menu, open the **Trace Configuration** dialog box.
2. Set the **Delay** field to 8,192 frames. This setting will allow the trace to collect 8K bus cycles after the trigger condition is met.
3. Set the **Trigger Condition** field to A.
4. In the **Break Emulator** area of the dialog box, click **No** to allow the emulator to continue running after the trigger.
5. Click the **Conditions** tab. This will allow you to configure condition A for the entry to function Main.
6. Right-click on the A condition line, and select **Edit**. This opens the **Condition A** dialog box.

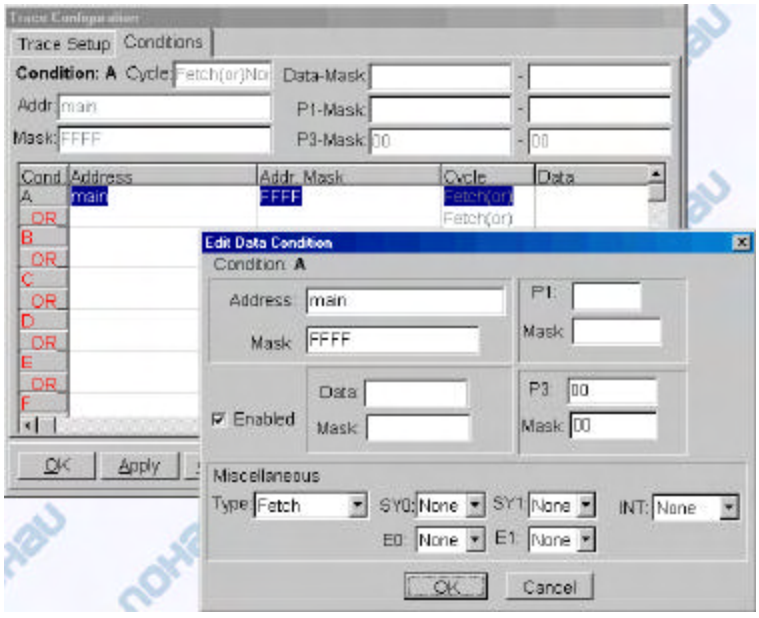


Figure 142. Trace Conditions Setup

7. In the **Address** field, type `Main`

8. From the **Type** pull down list, select **Fetch**.

9. Click **OK** to complete the trace setup.

To capture the trace, on the Seehau Speedbar click the Reset button  . Then click the Go button 

.

As soon as the trigger condition is met, the Trace Display window automatically updates. (See Figure 143.) Frame 0 is the trigger point which is the entry to function Main.

Filter Example

This section describes how to set up the emulator trace to filter data writes with the following parameters:

- Capture only data writes to a structure field called Timer.sec.
- Stop the trace after collecting 5,000 frames.

From the **File** menu, select **Load Code**. Double-click the file Timer.d07.
Then do the following steps:

1. From the **Config** menu, open the **Trace Configuration** dialog box.

2. Set the **Delay** field to 5,000 frames. This setting will allow the trace to collect 5,000 bus cycles after the trigger condition is met.

Trace_1

Frame	Address	SY10	INT	Ports(P1 & P3)	Pod Pins(E1 & E0)	Data	Instr.		Symbol
0	E	11	0	1111 1111	1111 1111 1 1	758C06	MOV	TH0,#06	main
1	F	11	0	1111 1111	1111 1111 1 1	8C			
2	10	11	0	1111 1111	1111 1111 1 1	06			
3	10	11	0	1111 1111	1111 1111 1 1	06			
4	11	11	0	1111 1111	1111 1111 1 1	E4	CLR	A	
5	12	11	0	1111 1111	1111 1111 1 1	90			
6	12	11	0	1111 1111	1111 1111 1 1	904000	MOV	DPTR,#counter 4000	
7	13	11	0	1111 1111	1111 1111 1 1	40			
8	14	11	0	1111 1111	1111 1111 1 1	00			
9	14	11	0	1111 1111	1111 1111 1 1	00			
10	15	11	0	1111 1111	1111 1111 1 1	F0	MOVX	@DPTR,A	
11	16	11	0	1111 1111	1111 1111 1 1	75			
12	4000	11	0	1111 1111	1011 1111 1 1	00			
13	16	11	0	1111 1111	1111 1111 1 1	75A882	MOV	IE,#82	
14	17	11	0	1111 1111	1111 1111 1 1	A8			
15	18	11	0	1111 1111	1111 1111 1 1	82			
16	18	11	0	1111 1111	1111 1111 1 1	82			
17	19	11	0	1111 1111	1111 1111 1 1	758902	MOV	TMOD,#02	

TRACE ONLY | got frames | Frames 9676(-1483:8192)

Figure 143. Trace Window with Trigger Example

3. Set the **Trigger Condition** field to A.

4. Set the **Record** field to A.

5. In the **Break Emulator** area of the dialog box, click **No** to allow the emulator to continue running after the trigger.

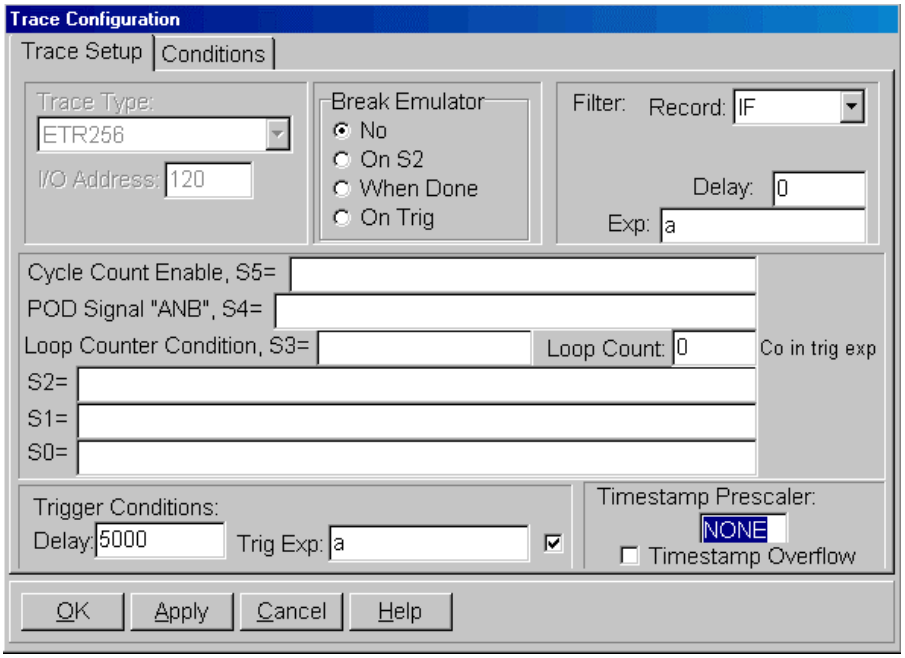


Figure 144. Trace Setup Tab

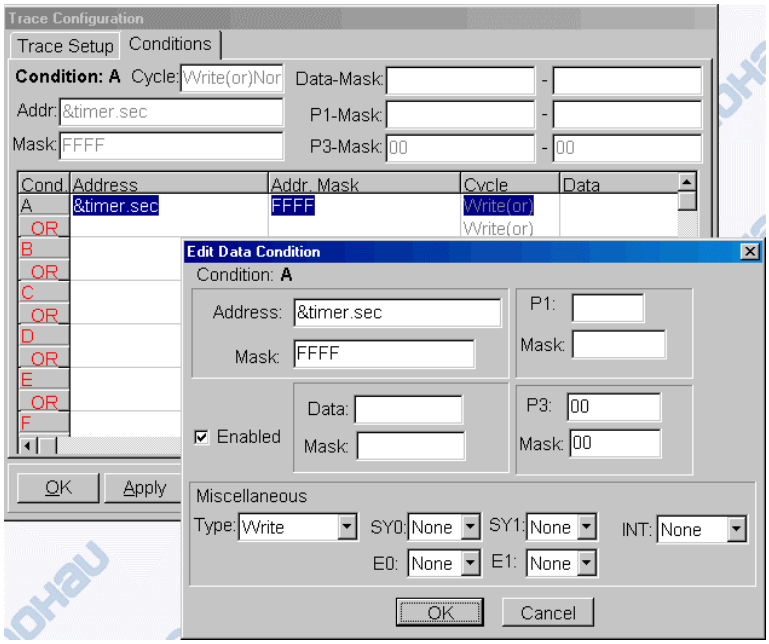


Figure 145. Trace Filter Setup

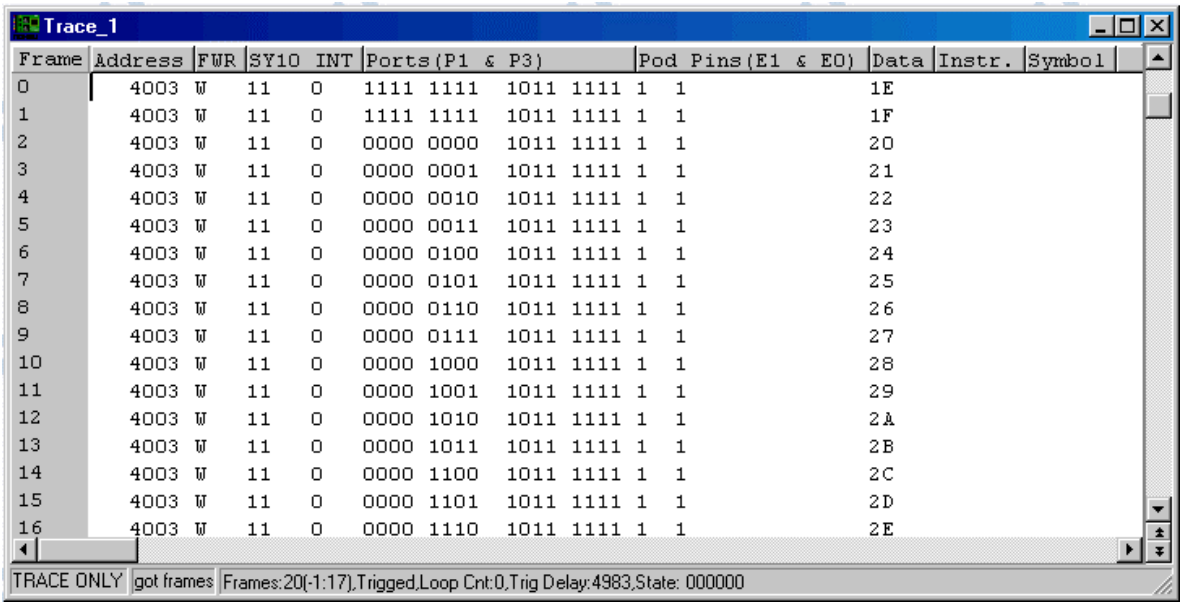


Figure 146. Trace Window with Filter Example

6. Click the **Conditions** tab. This will allow you to configure condition A for writes to the Timer.sec structure field.

7. Right-click on the A condition line, and select **Edit**. This opens the **Condition A** dialog box.

8. In the **Address** field, type `&timer.sec`.

9. From the **Type** pull down list, select **Write**. Then click **OK** to complete the trace setup.

To capture the trace, on the Seehau Speedbar click the Reset button  .Then click the Go button 

.

As soon as the trigger condition is met, the Trace window automatically updates. (See Figure 146.) Cycle 0 is the trigger point which is the first write to Timer.sec.

Combination of Conditions

When setting up conditions, it is possible to enter multiple qualifiers under a single condition. A qualifier is displayed on a single line under the Trace Condition Display. For example, under Condition A, a qualifier preceded by the word or indicates that condition has multiple qualifiers.

When a trace condition has multiple qualifiers, the fields in the qualifier are logically ORed together. For example, if you define Condition A as a Write to address 81H “OR” a Read from address 4003H, the logically combined condition results in a Read “OR” Write to or from address 4002H “OR” 4003H.

Alternatively, setting condition B to a Write to address 4002 and setting condition C to a Read from 4003, then triggering or filtering on “B or C” results in the entire condition being ORed rather than the individual fields. The resulting condition is a Write to 4002H OR a Read from 4003H.

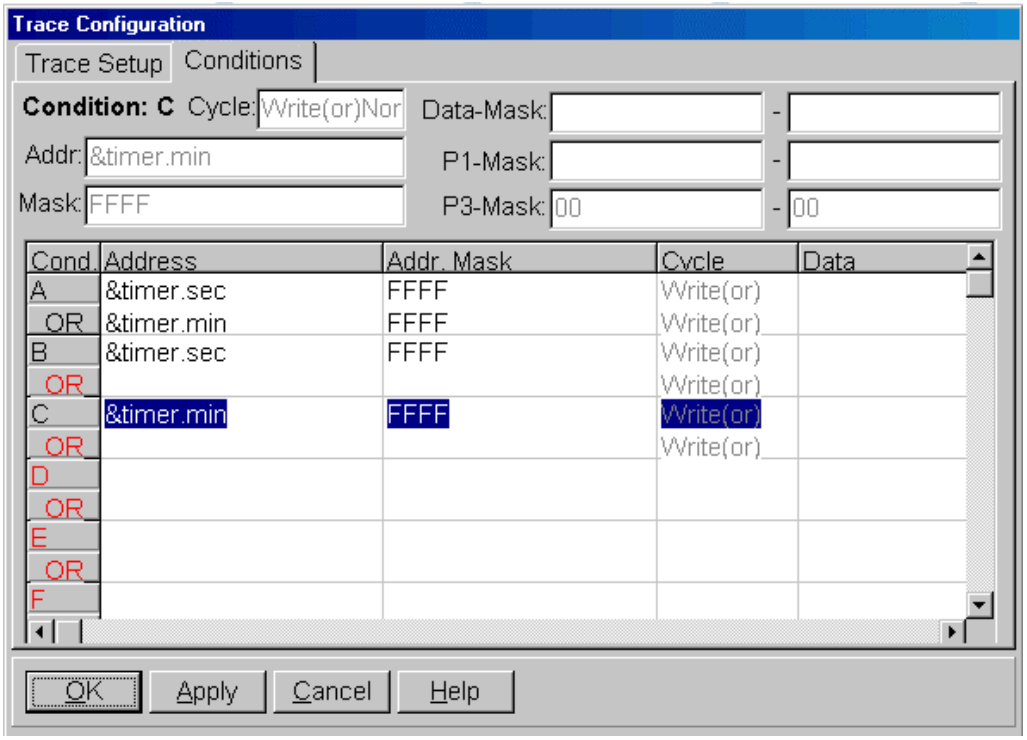


Figure 147. Combination of Conditions

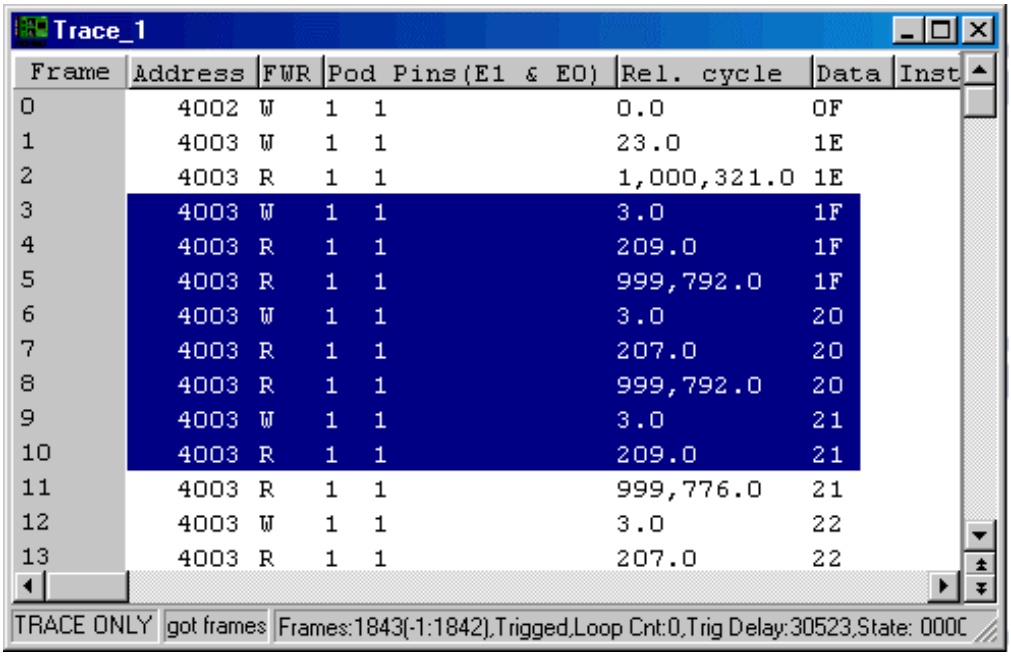


Figure 148. Trace Display Filter A

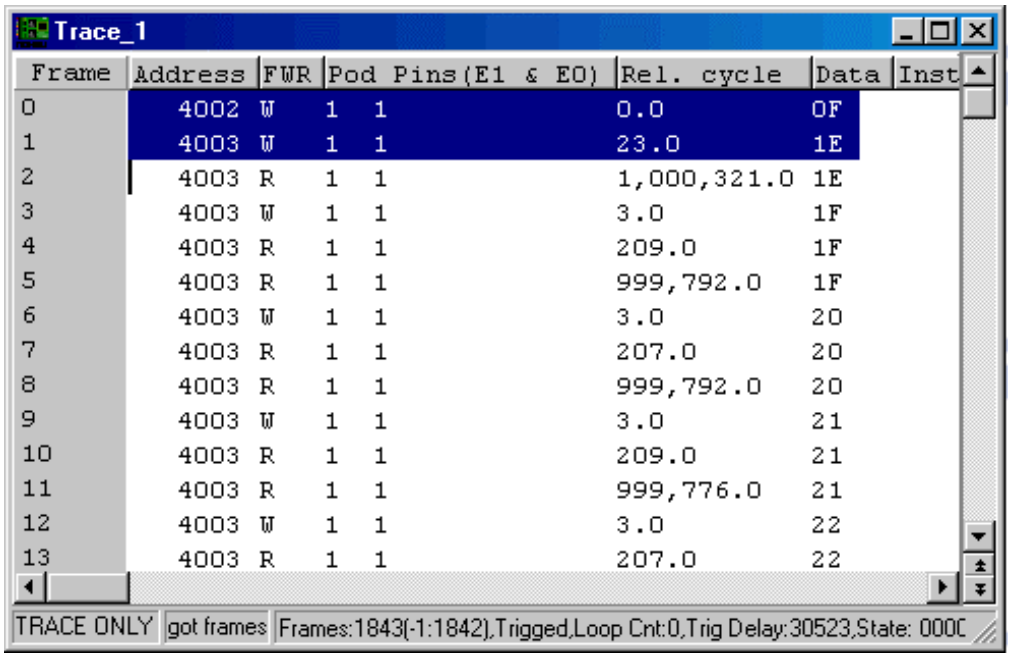


Figure 149. Trace Display Filter B or C

Building a State Machine Using S(x) Functions

The following provides an example using S2 – S0 as a 3-bit counter (eight states) for each time the A event equals the First Opcode fetch at address 24 hex. In this case, the result of S0 – S2 triggered the trace when S2, S1and S0 are all *true*. This example can be expanded by using more events (A-H) in the equations.

Using the program Timer.d07 the address 24 hex is the address of the timer( ) function. All trace fields should be at their default values except for the following:

- Trig = (S0 and S1 and S2)
- S2 = S2 AND /(S0 and S1 and A) or /S2 and (S0 and S1 and A)
- S1 = S1 and /(S0 and A) or /S1 and (S0 and A)
- S0 = (S0 and /A) or (/S0 and A)

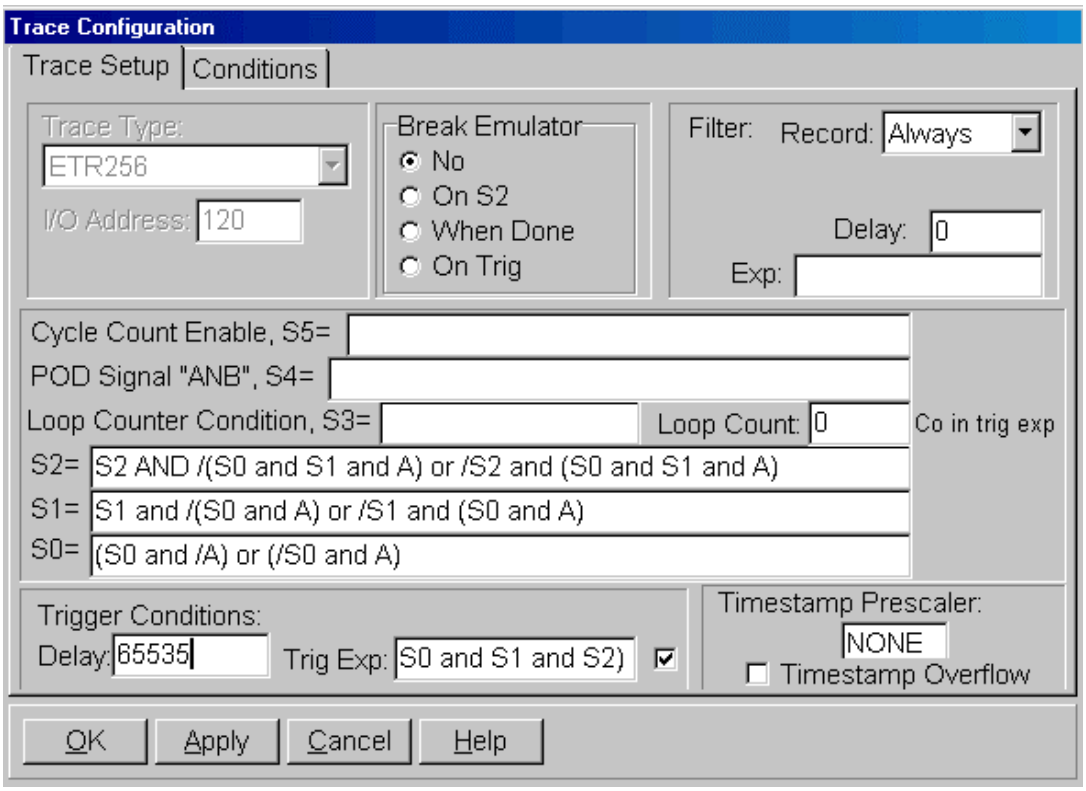


Figure 150. 6-Bit State Machine

The expression in S0, S1 and S2 could have been written using the Boolean XOR; however, the XOR is not implemented. The equivalent expressions are:

- S2 = S2 xor (S0 and S1 and A)
- S1 = S1 xor(S0 and A)
- S0 = S0 xor A

Example of How to Use the Set-Clear with the Sx Functions

S0 = set A clear B

S0 is set to *true* when A becomes *true*. It remains *true* until B becomes *true* (even if A is *false* before B is *true*). A and B in the previous example can be Boolean expressions. This can be used in a number of ways. The following example shows how to create a window for recording.

Record Expression = IF
Exp=S0 or A
S0 = set A clear B

The A must be in the RECORD expression in order for the result of the SET A to appear in S0 delayed by one frame. Otherwise, the address 24 hex will be missed.

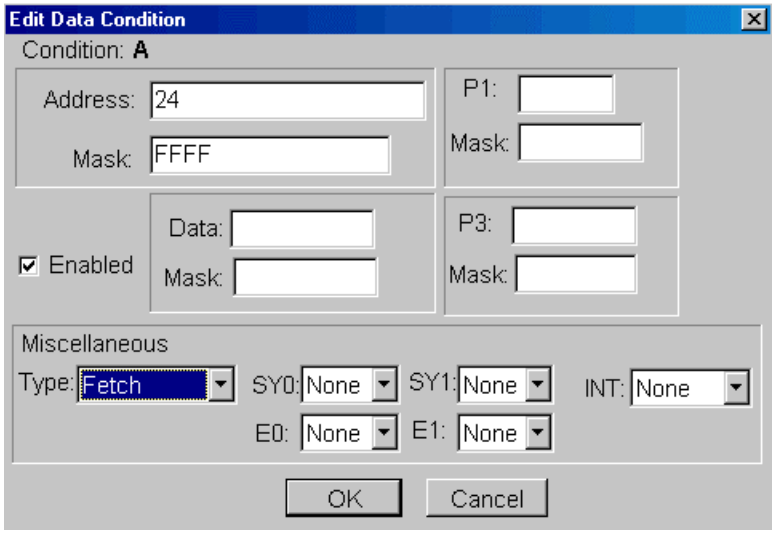


Figure 151. Event Field (A)

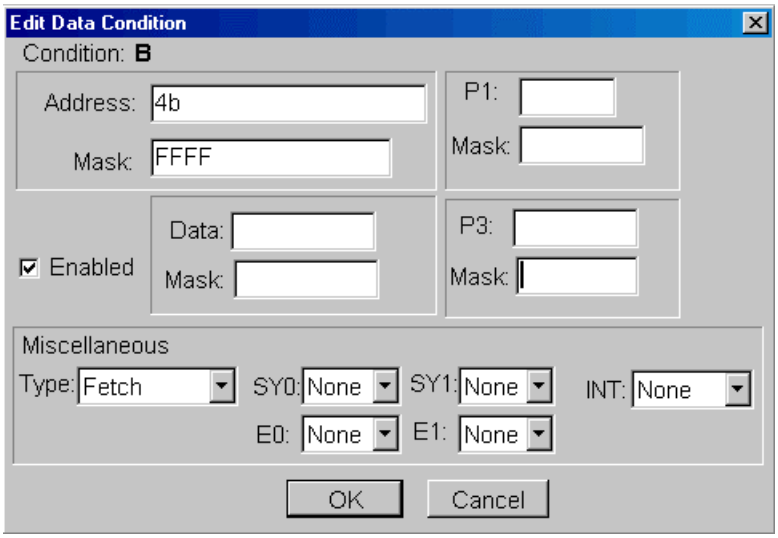


Figure 152. Event Field (B)

All activities taking place between the execution of the two specified addresses are recorded into the trace buffer. Any calls to subroutines that would be outside of the address range between 24 and 4B are included.

Figure 153 is an example of a 6-bit state machine using the Set-Clear and demonstrating how the loop counter and cycle counter work.

The program **Timer** located in the Examples directory will write to the symbolic address of Timer.sec (address 4003h) at an approximate rate of one second. To see the 6-bit state machine bits being updated in the Trace window status bar, first click the Reset button on the Speedbar, then click the Go button.

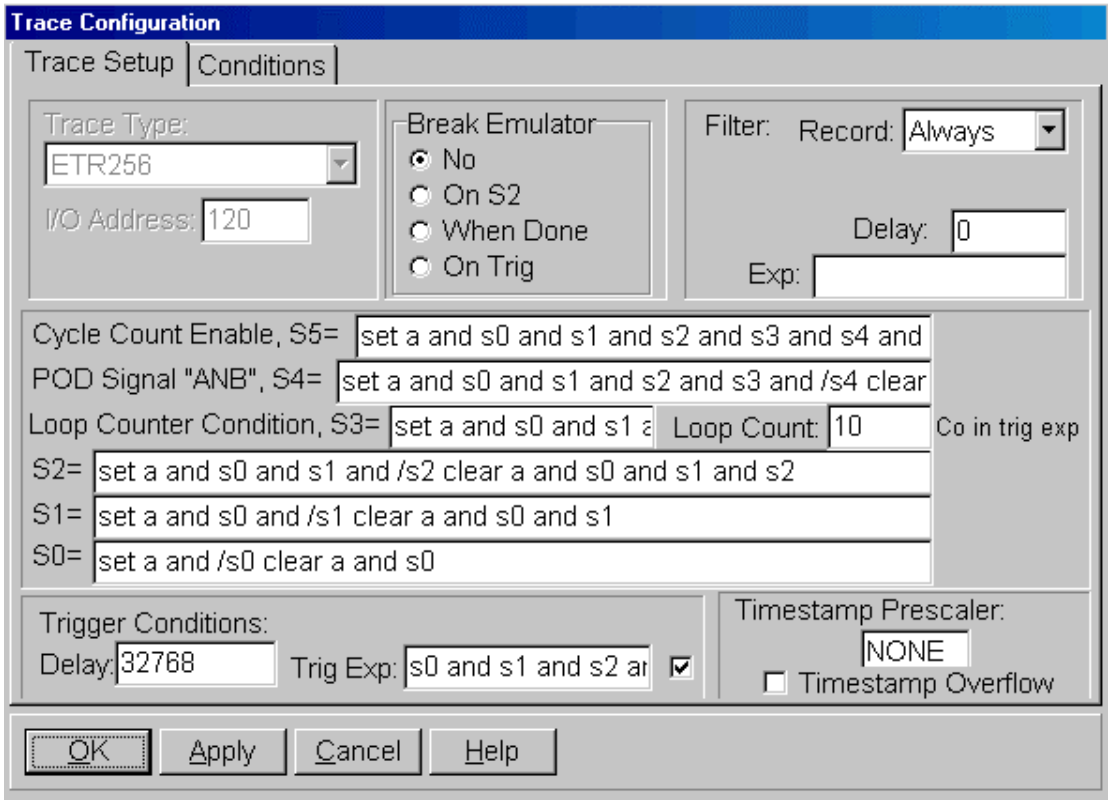


Figure 153. Six-Bit State Machine

With this example the loop counter, state machine bit 3, is set for 10. The counter will decrement every time the loop counter condition changes from *false* to *true* (0 to 1). The status of the state machine bits, value of cycle count, and the triggered or not triggered status are all displayed on the status bar in the Trace window.

Note

The MSB of the state machine is on the left and the LSB in on the right of the binary display.

8 Seehau Startup Troubleshooting

Before you start troubleshooting, first check the following items:

- Are the cables connected properly?
- Is the pod connected to the emulator board?
- If you are using an HSP/USB box, is the HSP/USB power turned on?
- Did you remove any foam that might be present on the bottom pins of the pod?
- Did you configure Seehau correctly for your MCU and pod?
- If the pod is not connected to your target, are the power and crystal jumpers/switches in the INT position? If not, move the jumpers/switches to the INT position.
- If the pod is connected to your target, is the target power turned on?

HSP Box

Step 1. When you start Seehau, does the HSP card LED flash?

- Yes. Go to Step 2.
- No. Make sure the power is on. Make sure the following are connected:
  - HSP box is connected to computer.
  - Power supply is connected to HSP.
  - Pod is connected to emulator board

If the HSP card LED is still not working, refer to the “Debugging the Parallel Port” section.

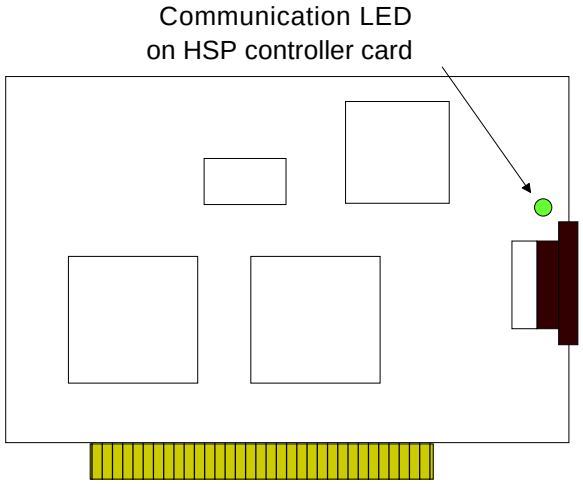


Figure 154. HSP Card LED

Step 2. If your pod has a reset LED, does it flash when you start Seehau?

- Yes. Contact Nohau Technical Support.
- No. Go to Step 3.

Step 3. Do board I/O addresses match the values in the Seehau configuration?

If your reset LED does not flash or your pod is not equipped with a reset LED, verify that the board I/O addresses (for emulator and trace boards) match the values in the Seehau Configuration:

- Yes. The I/O addresses match the values:
  1. From the **Start** menu, select **Programs**.
  2. Select **Seehau51**, then click **Config**. If the board I/O addresses match the values in the Seehau configuration, go to the "Configuring Address Settings with Windows Operating Systems" section in Chapter 2. Pay specific attention to alternate addressing.

     If you still encounter problems, contact Nohau Technical Support, support@icetech.com.

- No. The I/O addresses do not match the values:
  1. From the **Start** menu, select **Programs**.
  2. Select **Seehau51** and click **Config**.
  3. Enter the appropriate values.

Now does the reset LED flash?

- Yes. The reset LED flashes.

Does Seehau start?

  - Yes. Troubleshooting is complete!
  - No. Contact Nohau Technical Support.

Debugging the Parallel Port

Step 1. Disconnect other devices that might be sharing this parallel port (such as printers, zip, or jazz drives, parallel CD ROM drives, or software dongle keys).

Now is it working?

- Yes. You're done. You might opt to purchase an additional parallel port card.
- No. Do the following:

Windows NT Users

Check the Nohauxx driver status by doing the following:

- To check the status, go to the **Start** menu. Select **Control Panel**. Then double-click **Devices**.
  - If the status shows **Started**, go to Step 2.
  - If the status shows **Stopped**, check the ParPort driver for **Started** status.
  - If the ParPort driver shows **Stopped** click **Start**.

- Now re-check the driver status.
  - If the driver shows **Started**, try restarting Seehau.
  - If the ParPort driver still shows **Stopped**, go to NT Diagnostics:
    1. From the **Start** menu, select **Programs**.
    2. Then select **Administrative Tools**, and click **Windows NT Diagnostics**. The Windows NT Diagnostics window opens.
    3. Click the **Resources** tab.
    4. Click **I/O Port**. Scroll down to address 378 (LPT1) and look for a device at this address.
    5. From the Control Panel, double-click **Devices**. Disable the device located at 378.
    6. Attempt to restart Seehau. If this fails, go to Step 2.

Windows 9x/ME Users

Check the parallel port mode. Go to Step 2.

Windows 2000 Users

Verify that the Nohau51 device driver is properly installed. Do the following:

1. From the **Start** menu, select **Programs**. Select **Accessories**, then click **System Tools**.

2. Double-click **System Information**. The System Information window opens (Figure 155).

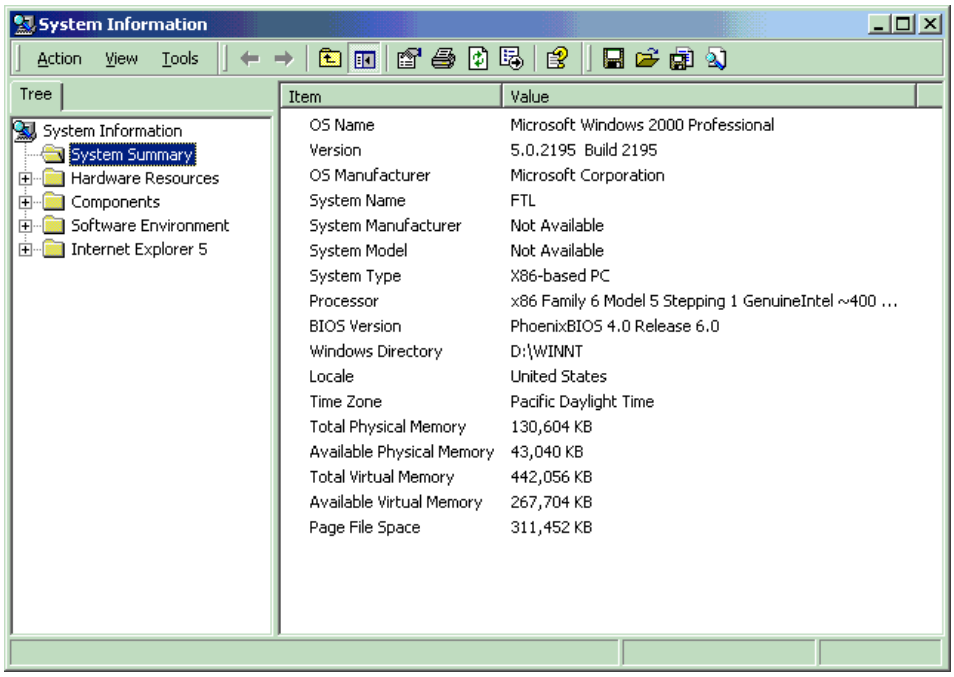


Figure 155. System Information Window

3. Click **Software Environment**.

4. Click **Drivers** to display a list of active drivers. Refer to the **Name** column and scroll down to Nohau51 (Figure 156).

5. In the **State** column, verify the driver is running. In the **Status** column, you should see OK.

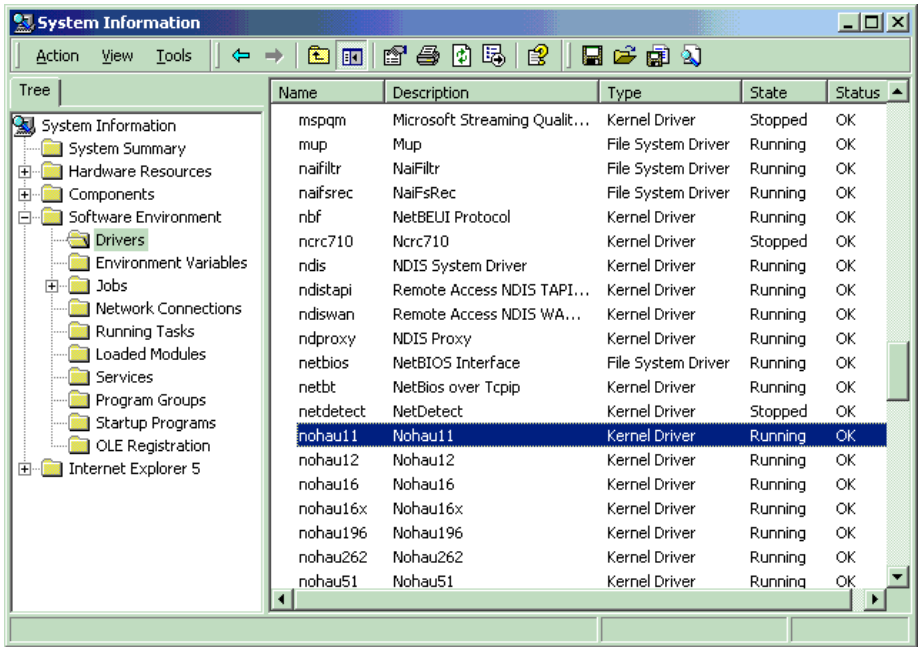


Figure 156. List of Active Drivers

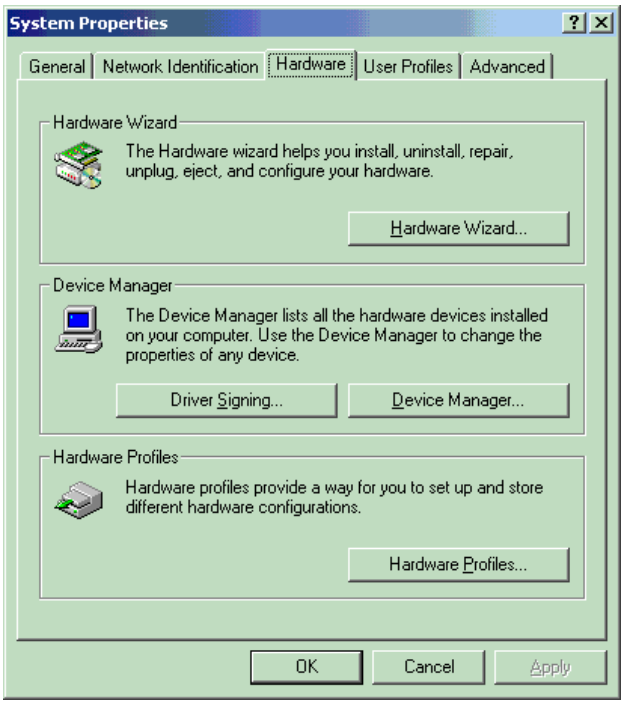


Figure 157. System Properties Window

If the ParPort driver still shows “Stopped,” do the following:

1. Right-click the My Computer icon on your desktop, and select **Properties**. The System Properties window opens (Figure 157).

2. Click the **Hardware** tab. Then click **Device Manager**. The Device Manager window opens (Figure 158).

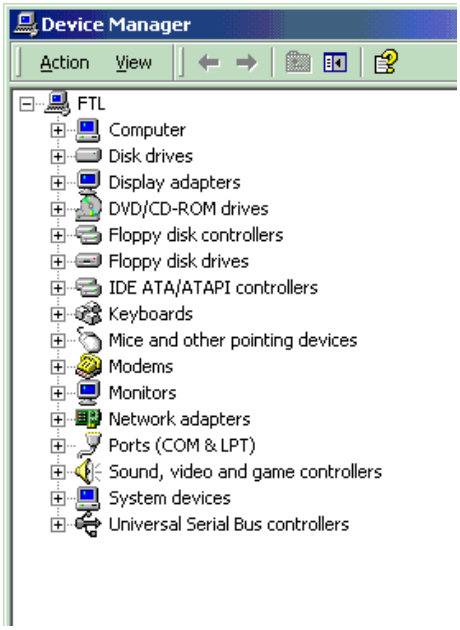


Figure 158. Device Manager Window

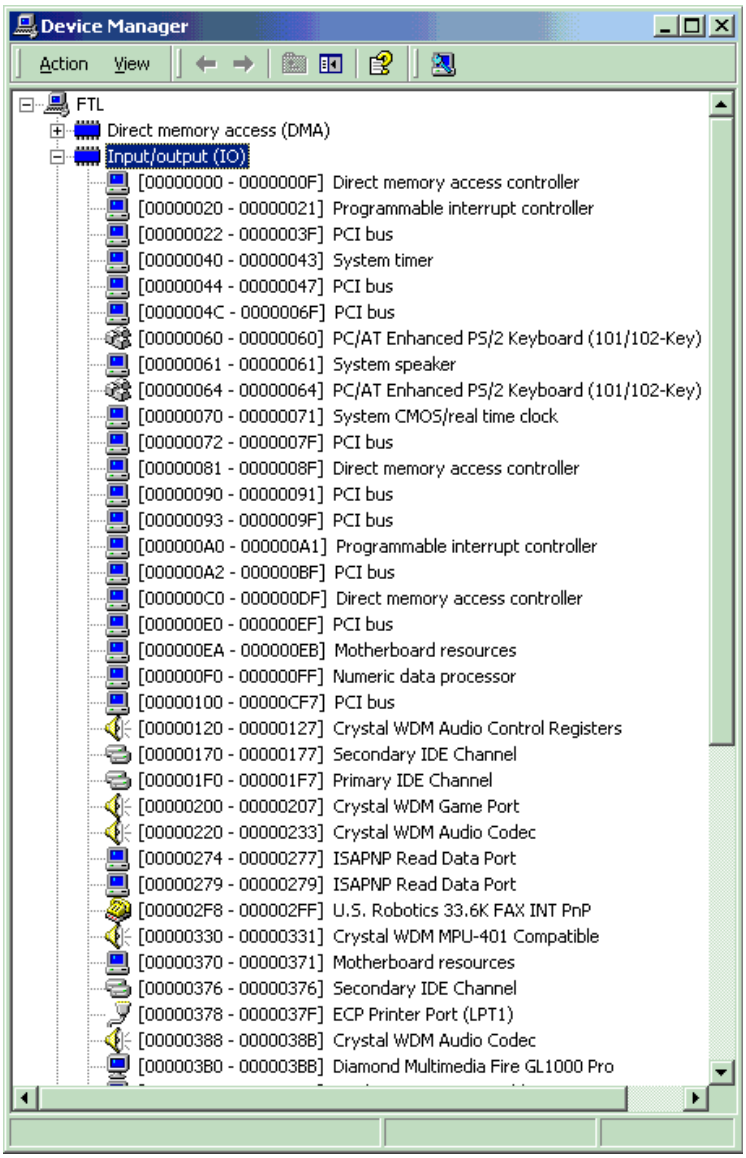


Figure 159. Device Manager Window Displaying the System Resources

3. In the Device Manager window, select the **View** menu. Then click **Resources by Type**. A window appears that shows system resources (Figure 159).

4. Double-click **Input/Output (I/O)**.

5. Scroll down to address 378 (LPT1) and look for a device at this address. Go back to the Control Panel and double-click **Devices**. Disable the device located at address 378. Attempt to restart Seehau. If this fails, proceed to Step 2.

Step 2. Check the parallel port mode.

1. Reboot and enter BIOS setup. From BIOS setup, check for one of the following parallel port modes:
   - Normal
   - Standard
   - Compatible
   - Output only
   - Bi-directional
2. Ensure that one of these modes is selected.
3. Then try selecting another mode.
4. Save your settings and reboot.

> **Note**
>
> The following modes have been known to cause problems: ECP, EPP, or ECP + EPP.

ISA

Step 1. If your pod has a reset LED, does it flash when you start Seehau?

- Yes. Contact Nohau Technical Support.
- No. Go to Step 2.

Step 2. Do board I/O addresses match the values in the Seehau configuration?

If your reset LED does not flash or your pod is not equipped with a reset LED, verify that the board I/O addresses (for emulator and trace boards) match the values in the Seehau Configuration:

- Yes. The I/O addresses match the values:
  1. From the **Start** menu, select **Programs**.
  2. Select **Seehau51**, then click **Config**. If the board I/O addresses match the values in the Seehau configuration, go to the "Configuring Address Settings with Windows Operating Systems" section in Chapter 2. Pay specific attention to alternate addressing.

If you still encounter problems, contact Nohau Technical Support.

- No. The I/O addresses do not match the values:
  1. From the **Start** menu, select **Programs**.
  2. Select **Seehau51** and click **Config**.
  3. Enter the appropriate values.

Now does the reset LED flash?

- Yes. The reset LED flashes.

Does Seehau start?

  - Yes. Troubleshooting is complete!
  - No. Seehau does not start. Contact Nohau Technical Support.

Known Device Driver Conflicts

Nohau is aware of potential device driver conflicts with certain network cards running on Novell/ Netware networks. Problems have been reported with both 3COM ISA network cards and some Novell network cards. Most of these problems have been experienced when running Windows NT or Windows 2000 operating systems.

Possible Symptoms

- When starting Seehau, communication with the network stops. (You will be unable to access resources on the network.)
- Seehau will not start.

A possible solution might be to change you network card. Nohau Technical Support has not tested all network cards, although some customers have reported that the following network cards have resolved this conflict:

- Intel Ether Express Pro 10/100 ISA
- 3COM Etherlink III (905B or later) 10/100 PCI
- Bay Networks NetGear FA310TX 10/100 PCI

If the Emulator Does Not Start When Connected to the Target System

- Make sure power is applied to the target system.
- If the target has an external watchdog timer, remove the Reset jumper on the pod.
- Check the ALE for a clean signal.
- Try switching the crystal jumpers/switches to the INT position.
- Disconnect the target. Make sure you change the crystal and power jumpers/switches to the INT position. Then try starting Seehau.
- Check the orientation of the target adapter. Confirm that the adapter is inserted properly. For more information, refer to Chapter 5, “Connecting the Emulator to Your Target Board.”
- Check for grounding problems. The emulator and target should have a solid common ground. Targets that are improperly grounded or designed with a *floating* ground might experience improper operation. A closer examination of control signals might reveal excessive over / undershoot or ground noise.
- If you are able to start the emulator, the problem is with one or more of the following critical target signals:
  - address and data bus
  - R/W signal
  - ALE
  - PSEN
- Check the target for any device that is enabled by address qualification only and does not use ALE, R/W, or PSEN signals.

- Check the address/data bus loading. If your target design approaches maximum fanout for CPU drive capability, the emulator might not function correctly. This is caused by additional loads (approximately two TTL loads) from the pod board.
- Use an isolator to debug the target. (Refer to Appendix E, “Troubleshooting With an Isolator.”)

Target Does Not Operate Correctly

Problem:	**Serial port is not operating correctly.**
Cause/ Solution:	• Check jumpers/switches on pod board for external crystal selection. • Check signal and ground connections between pod and target.

Problem:	**Cannot access target memory or memory mapped I/O**
Cause/ Solution:	• Check the **Map Config** tab to ensure that the memory address range is mapped to target. (For details, refer to Chapter 4, "Installing and Configuring the Seehau Software." Go to the “Starting Seehau” section. • Check RD, WR and ALE signals on the target. • Make sure you have an external mode pod.

Appendix A. Address Examples

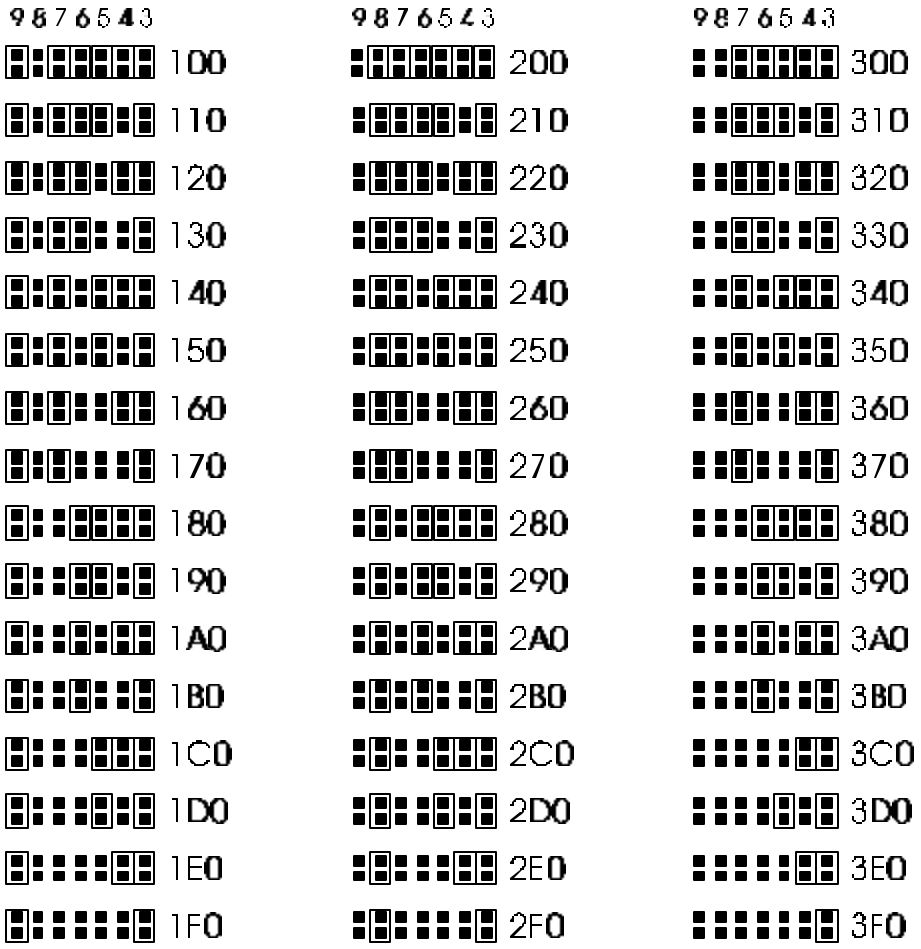


Figure 160. Emulator Addressing Example

Note

Be aware that the standard emulator boards reverse the order of A9 through A3.

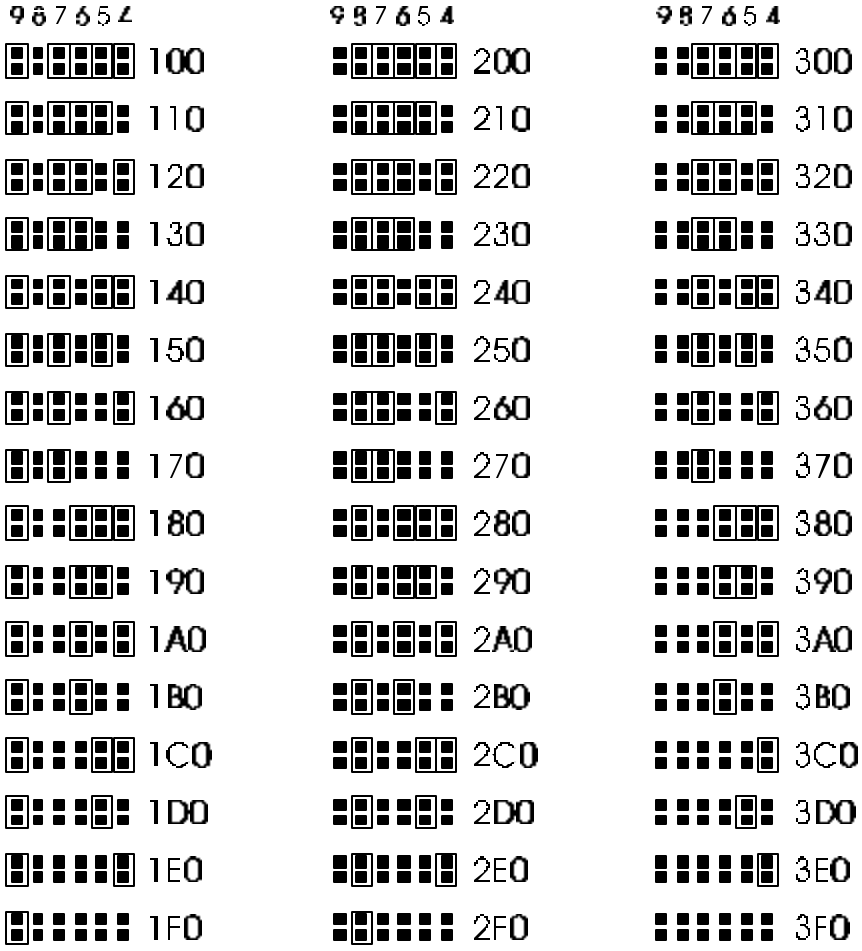


Figure 161. Standard Trace Addressing Example

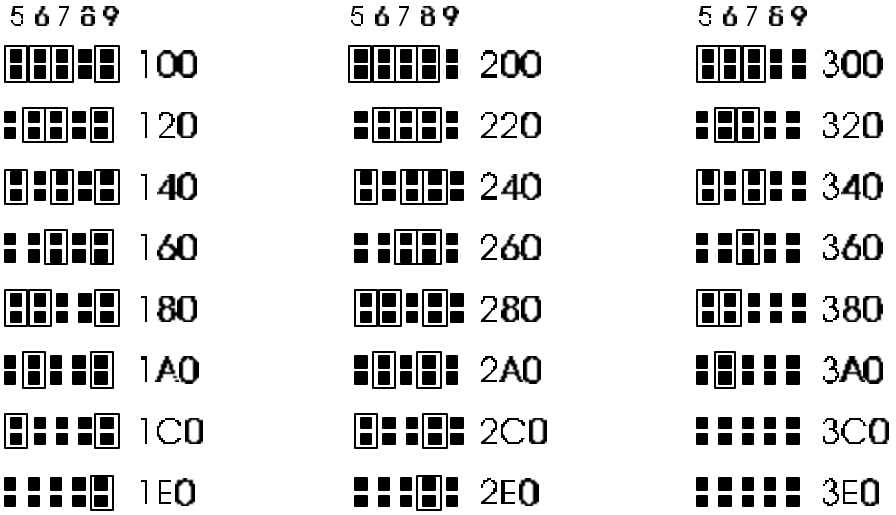


Figure 162. Enhanced Trace Addressing Example

Appendix B. EXT–DIP40 Extender Cable

This is a six-inch long, 40-pin DIP extender cable. As shown in Figure 163, the EXT–DIP40 adapter could be a good solution if the target system is in a box where the pod will not fit. However, a potential problem is that the emulator gets more sensitive to noise.

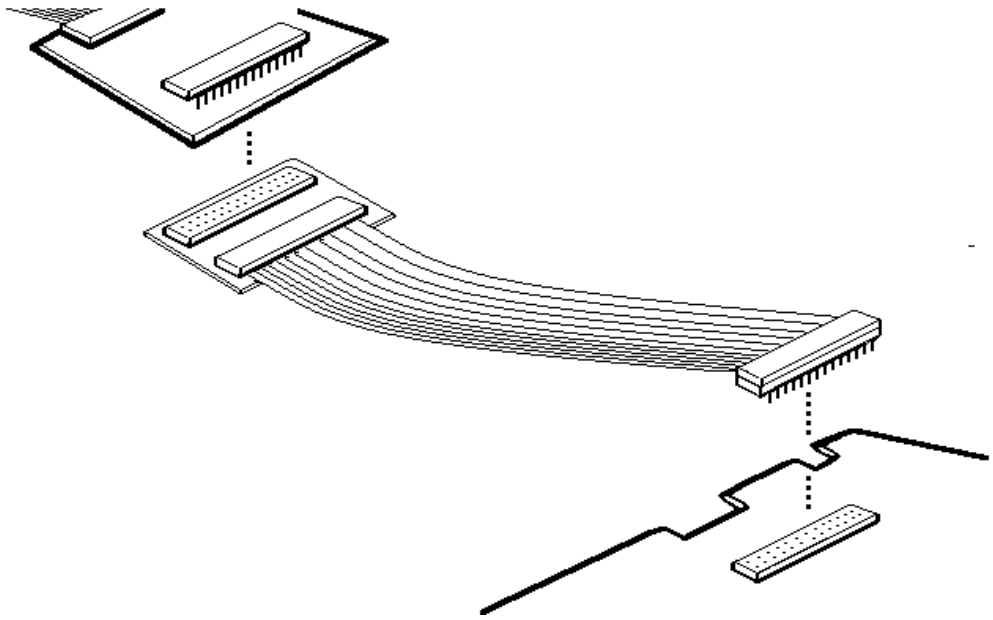


Figure 163. EXT40 Adapter Inserted Between the Pod and the Target

Appendix C. Configuring the M3 Setting for Your HB Pod

Note

You must use a Philips-manufactured part on the HB fab.

Pod Name	M3 Setting	Processor Interchangeable
POD–51HB–C591–16	Out	No
POD–51HB–C51FX–33	In	Yes (except for 591)
POD–51HB–C51RX–33	In	Yes (except for 591)
POD–51HB–-C51RX2–16	Out	Yes (except for 591)
POD–51HB–L51FX–16	In	Yes (except for 591)
POD–51HB–L51Rx–16	In	Yes (except for 591)
POD–51HB–C66X	Out	Yes (except for 591)
POD–-51HB–C52–33	In	Yes (except for 591)
POD–-51HB–C52–24	In	Yes (except for 591)

Appendix D. Seehau Startup Options

Parameter	Function
-b	Allows Seehau to start up with macro other than startup.bas This parameter is followed by a macro name.
-2	Allows you to run a second macro. This parameter is followed by a macro name.
-l	Allows you to connect to another machine. This parameter is followed by an 'ip' address
-r	Starts the Seehau reconfiguration program.
-c	Starts the Seehau configuration program.
-d	Starts Seehau in "demo" mode. Use demo mode to familiarize yourself with the Seehau interface without hardware.
-l	Lengthens the time it takes Seehau to startup.
-u	Brings up a dialog box during startup which asks you to select a startup macro.

Appendix E. Troubleshooting With an Isolator

Pin isolators are available to troubleshoot target-related problems.

Each isolator contains one DIP switch for each of the MCU's pins. With all pins disconnected, the emulator basically runs stand-alone. You can then connect pins from the target individually or in groups to determine which pin or group of pins is causing a problem. After the pin(s) is identified, the cause of the problem can be resolved.

Appendix F. Using the POD–C520–PGA–33 to Emulate the DS80C310 MCU

DS80C310 Emulation

The DS80C310 MCU is a lower-featured version of Dallas Semiconductor's popular DS80C320 High-Speed MCU for more cost-sensitive applications. Although this chip is functionally comparable to a DS80C320, its timing parameters more closely match that of the DS87C520. In an effort to reduce the amount of work required by emulator manufacturers, Dallas Semiconductor has allowed this device to be emulated by DS87C520 hardware (pod) using an emulation bondout chip. No hardware modifications are necessary, and only minor software updates are required to allow a POD–C520–PGA–33 and emulator to support the DS80C310.

The DS80C310 is available in 40-pin DIP, 44-pin PLCC, and 44-pin TQFP package types.

Feature Differences

The following features present on the DS87C520 MCU and in the POD–C520–PGA–33 are not supported on the DS80C310:

- Serial Port 1 (second serial port)
- Power-fail reset
- 1K MOVX SRAM
- Watchdog timer
- Ring oscillator

No Hardware Modifications are Necessary to Support the DS80C310 MCU

Because the DS80C310 and DS87C520 are both pin-compatible with the 8051, no hardware modifications are necessary to use a POD–C520–PGA–33 and emulator as a DS80C310 emulator.

Modifying Existing Software to Support the DS80C310 MCU

In terms of software (SFR) support, the following registers or bits are absent or have a modified function in the DS80C310:

Port 0	80h	(register not implemented)
CKCON	8Eh	(bits CKCON.7-6 not implemented)
EXIF	91h	(bits EXIF.3-0 not implemented)
IE	A8h	(bit IE.6 not implemented)
SADDR1	AAh	(register not implemented)
IP	B8h	(bit IP.6 not implemented)
SADEN1	BAh	(register not implemented)
ROMSIZE	C2h	(register not implemented)
PMR	C4h	(register not implemented)
TA	C7h	(register not implemented)
WDCON	D8h	(bits WDCON.7,5-0 not implemented)
EIE	E8h	(bits EIE.5-4, not implemented)
EIP	F8h	(bits EIP.5-4, not implemented)

User Application Software Precautions

The DS87C520/DS87C530 implements a few SFRs differently than the DS80C310 MCU. When using the POD–C520–PGA–33 to emulate a DS80C310, note the following software precautions for user application software.

- Clearing the POR bit: Use the CLR bit command to clear the POR bit (WDCON.6). This step must be preceded by a Timed Access procedure. Use the following instructions to clear the POR:
  - MOV 0C7h, #0AAh
  - MOV 0C7h, #55h
  - CLR POR

  These instructions are not required when reading the POR bit, only when clearing it. These instructions can be included in your source program and enabled or disabled through a SWITCH option when compiling for the emulator or DS80C310 application. The Timed Access procedure will not affect the DS80C310 if it is inadvertently left enabled in your final application code.

Note

The next two items are necessary to prevent the enabling of DS87C520/DS87C530 features that are not present in the DS80C310. These are a minor and will not affect operation in your final application code:

- Bits EXIF.3-0 must remain 1000b whenever the EXIF register is modified.
- Only modify bits in the EIE and WDCON register when using the SETB bit, CLR bit, or CPL bit commands.

1996 SEP 04

Adapted from information from Dallas Semiconductor

Appendix G. Using an 80C51RX MCU in an External Mode Pod

Some users have reported that if they place a Philips RA/B/C/D+ MCU on the pod, the system will not operate while using a standard POD–31 or POD–31A.

Due to the specifics of this MCU with the on-chip XDATA memory, the following solution is the only way to make this configuration work:

Requirements:	**Emulator Board**	EA256/768 emulator board
	Pod	31 or 31A
	Installed MCU	Philips Rx+

1. Remove the RWEN jumper on the emulator board.

2. Start the emulator. Select one of the following if available in the current version of software you are using:

Pod type selections:	**EMUL51–DOS or WIN**	80528
	Seehau51	POD–C51RX

> **Note**
>
> This will not allow P3.6 or P3.7 to operate as I/O on the POD BOARD PCB. On the POD-31A PCB (see the “POD-C32HF-42” section in Chapter 3 for details.)

Appendix H: Bank Switching

Overview

In the past the MCS-51 architecture was limited to 64K addressable space for CODE and XDATA. Users started running into the boundaries of this micro family. In this case the C-compiler vendors started to struggle to optimize their compilers so that users could stay within the constraints of the family's architecture and could not. Henceforth the birth of bank switching (paging a section of the code memory) to allow much larger applications to run on the very popular MCS-51 family of parts.

Nohau was one of the first emulator companies to work with some of these customers to develop a way of handling this need. Since then there has been a quasi-standard way of working with banked applications that has been industry wide. The most common is to have the upper 32K page of memory swap between different code pages thus leaving the lower 32K open for the critical functions that need to common to overall operation of an application.

Hardware Identification

Do to this migration the emulator equipment has evolved from the earliest version of the boards, which had many hand-done modifications to the E128 emulator board (full length 8-bit ISA style of PCB). To the PCB sold today that uses surface mount technology and the PCB is a 2/3^{rds} length 8-bit ISA board.

Older Hand-Modified Bank Switch Emulators

The following are a few types of boards available from Nohau:

- E128–BSW
- E128–BSW–DMA
- E256–BSW
- E256–BSW–DMA
- E320–BSW–DMA (only sold internationally)

Today's Generation Emulators

- EA256–BSW
- EA256–C320–BSW
- EA256–C530–BSW
- EA256–MX
- EA768–BSW
- EA768–C320–BSW
- EA768–C530–BSW
- EA768–MX

Refer to Chapter 2, "Installing the Hardware" regarding the various memory configurations available for the emulator board you have.

In the case of the option for memory size in the emulator startup most of the older generation bank switch boards are started with the size option set to 32K, except for the E320–BSW which is set for 128K. When using Seehau you will be selecting either the E32 (32K) or E128 (128Kk) option depending on your board type. With the newer generation of emulators the memory configuration uses the 128K-memory option or the board type option from the Seehau configuration.

Setting the memory operation is critical for proper operation of the emulator. An incorrect operation will cause various problems with code loading and execution.

Pod Boards

Most all of the 8051 family pods can be modified to work with bank switching with an exception of a few where the micro itself is not designed for external memory access anyway. The modification includes the addition of wires that are soldered to the bottom side of the pod. These wires are used to connect to the source that drives the extra address lines: A16, A17, A18, and A19.

In some cases, for those users of the newer generation emulators that did not purchase the bank switch option to the pod at the time of purchase, it might require an additional modification. (Contact Nohau Technical Support for assistance. support@icetech.com)

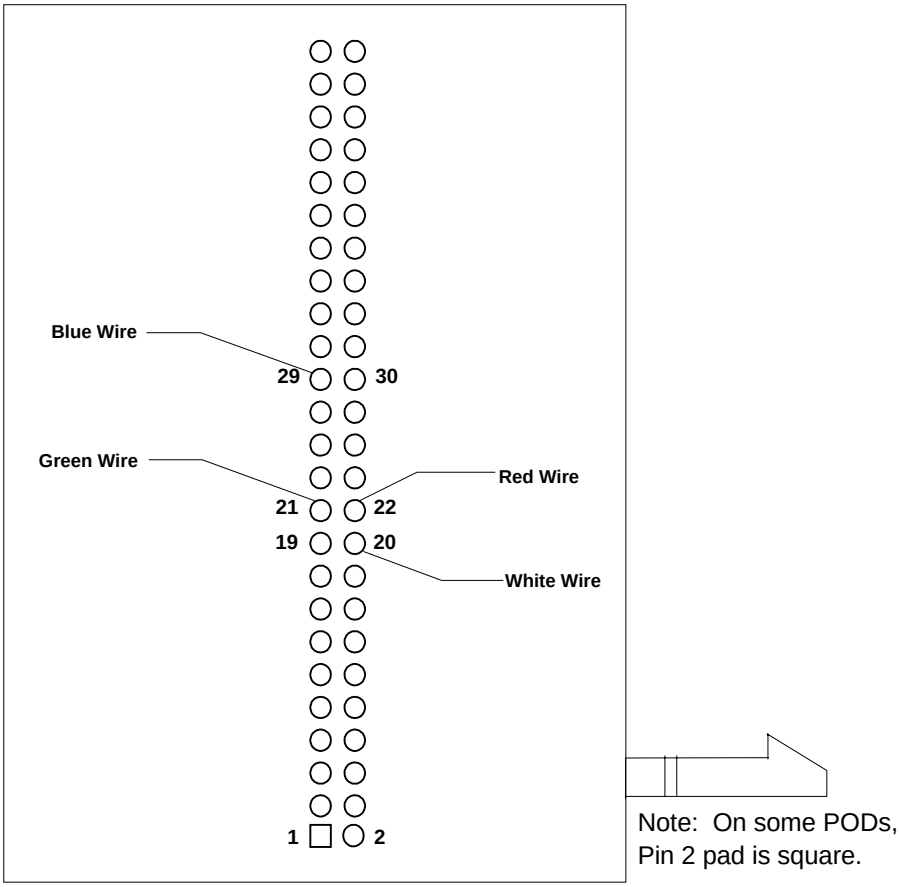


Figure 164. Pod Board Modification

Definitions of Bank Switch Connections

Bank switch wire color	Bank switch input bit	Target chip select or extended address bit	Connections for trace inputs used for breakpoints and triggering on bank specific locations
Red	0	CS0 / A16	E0 *
Green	1	CS1 / A17	E1 *
White	2	CS2 / A18	SY0
Blue	3	CS3 / A19	SY1

* Some pods don't have a labeled connection for E0 and E1, in most cases these two inputs can be found on the trace input jumpers labeled P3. In these cases remove the jumpers for P3 bits 6 and 7 and connect E0 to P3.6 and E1 to P3.7.

C Compilers

Most of the compilers today have some form of support for the bank switching. Some support both banked code and XDATA memory. Nohau only supports the banked code portion of the application and the target system must support the banked XDATA memory.

When building a banked application it might require linking the application differently from programming an EPROM or FLASH memory to be placed on your target. This means for each bank you will increment the base address by 0x10000.

There in most cases is an area of code memory known as COMMON. This is memory normally located in a non-banked area or is common and in the same location in every bank when dealing with 64K paging.

The COMMON code always includes the vector table and any other functions that need to be quickly accessed from within any bank without bank switching.

Emulator Memory for Banked Memory Regions

32K Banks	48K Banks	64K Banks	Bank Number	Select Bits Ex/Syx Bit Pattern
0000-7fff Non-Banked	**0000-3fff Non-Banked**			**S S E E Y Y 1 0 1 0 _ _**
08000–0ffff	04000–0ffff	00000–0ffff	0	0 0 0 0
18000–1ffff	14000–1ffff	10000–1ffff	1	0 0 0 1
28000–2ffff	24000–2ffff	20000–2ffff	2	0 0 1 0
38000–3ffff		30000–3ffff	3	0 0 1 1
48000–4ffff		40000–4ffff	4	0 1 0 0
58000–5ffff		50000–5ffff	5	0 1 0 1
68000–6ffff		60000–6ffff	6	0 1 1 0
78000–7ffff		70000–7ffff	7	0 1 1 1
88000–8ffff			8	1 0 0 0
98000–9ffff			9	1 0 0 1
A8000–Affff			10	1 0 1 0
B8000–Bffff			11	1 0 1 1
C8000–Cffff			12	1 1 0 0
D8000–Dffff			13	1 1 0 1
E8000–Effff			14	1 1 1 0
F8000–Fffff			15	1 1 1 1

General Operations

To successfully work with a banked application do the following:

1. Set the emulator board jumpers correctly.
2. Start up the emulator with all the associated parameters correctly.
3. Set up the banking configuration in the emulator software.
4. Set the memory mapping configuration correctly for the emulator.
5. Have all associated wire connections for the bank switching set up. (If you are using a setup that bridges the numbers of available banks but you are not using all the banks that are available in an emulator configuration, ground the extra bank switch wires so that you do not get spurious switching of banks.)

Important Considerations for Bank Switch Control

- Bank switch control through the MCU's port does not require a Shadow byte.
- Bank switch control through an external data memory mapped I/O: Can the memory location that is written to for controlling the banks be read back with the same value on the associated bits to confirm the bank number?
  - Yes: No Shadow byte is required.
  - No: A shadow byte is required.

Q & A

Q—**What is a Shadow byte?**

A—A Shadow byte is a location in memory that is in either internal chip data memory or off-chip XDATA memory that your program would also have to write to at the time you write to the control register for the bank selection. This would give us the ability to read the current bank.

Q—**Do I need to use the Bank Translation table?**

A—If you have consecutive bits in the byte location and then count up in a normal binary incrementing pattern, then **no**. If you are not using consecutive bits or you have a special bit pattern for banks then **yes**, you will need to enable the table and place the hex value for each bank into the table.

Q—**Do I need a trace board to work with a banked application?**

A—If you don't have a trace then you will not be able to set a breakpoint to break in the proper bank that you want. The emulator will break in every bank at the same logical address. At that point you would have to go again until you reach the bank you want to stop at. (You can automate this procedure using a Seehau macro.)

Q—**Do the bank switch signals need to remain at a constant state while inside a bank?**

A—Yes, if the signals change states while in a bank then the emulators memory will also change banks and you will not be able to execute your code correctly. Therefore, you need to latch the banking control lines to a solid state.

Q—**When placing an address in the trigger condition address field what address do I use?**

A—The Seehau interface will use full symbolic operation. This means that you can place the symbolic function name, or you can use the physical address. (This includes the upper nibble of the extended address.) For example, if you have a function at logical address 87F2h that is in bank 3, then the physical address would be 387F2h. When you exit the trace configuration we will automatically set up the Ex and SYx bits correctly base on you bank mask byte.

Q—**How do I set the control and status byte information in the configuration setup?**

A

- **Address**: The address that your program writes to control the banking lines.
- **Mask**: The byte mask that filters all un-used bits for banking at the byte address. Setting the bit to a 1 indicates that the bit is used for the banking control line. For example, if you are using bits 3 through 7 for controlling your bank selects, the mask would be 38 hex.
- **Memory Selection Pull-down**: Allows you to select either SFR (special function register) or XDATA based on your method of switching banks.

Troubleshooting

Problem:	**I load my banked application and it does not appear to have loaded correctly.**
Cause/ Solution:	• Check to make sure you have connected all the bank control wires from the pod to your target systems device that is controlling the banks. • Check to make sure you have enabled the bank switch option in the emulators configuration. • If a special bit pattern or shadow byte is required make sure that you have configured these correctly.

Problem:	**Some banks have the wrong data. They appear to have the same information that is in other banks.**
Cause/ Solution:	• Connecting the bank switch wires to the device that is driving the wires. • Under the configuration option in the emulator software, disable the bank switching option. • Now write a value through the DATA window or the Register window to the address that controls the bank switch wires. • Check the wires to see if they have the proper values (high or low) for each bit. • Continue checking all bits with different patterns for the output driver.

Problem:	Some bits do not go to a full high or low.
Cause/ Solution:	Disconnect the wires from the device that is controlling these lines and retest. If these signals work correctly then the following might be true: • You have an EA-series emulator and the BS0 or BS1 jumpers are installed on the board. You should remove them. • You have an EA-series emulator and it has not been modified correctly so that the termination resister pack has been removed from the circuit. • Some pods require additional modification due to some signal conflicts. You need to get this correctly through Nohau. (If you had the pod purchased from Nohau with the bank switch modification option, then it should have been done already.) • If you experience hardware failure, contact Nohau Technical Support for assistance.

Problem:	Some or all of the bits don't move.
Cause/ Solution:	Disconnect the wires from the device that is controlling these lines and retest. If these signals work correctly then it might be hardware failure. Contact Nohau Technical Support for assistance. If these signal still don't move (either stuck high or low) then the following might be true: • You are using one of the ports on the micro that is being traced and the trace buffer is damaged causing the signal to not be outputted correctly from the micro. Try removing the trace jumper for that port bit(s). At this point if the signal works or still does not work then contact Nohau Technical Support for assistance. • You are using a memory mapped I/O device and the programming of this device is either not correct, or the output from the device is damaged.

Problem:	I set a breakpoint by clicking in the Source/Program window and my code does not stop at this address.
Cause/ Solution:	• Check to make sure that you have connected the E0, E1, SY0 and SY1 trace input bits to each bit that is used for banking. Refer to the "Definitions of Bank Switch Connections" table. If these bits are not connected then the trace can not trigger correctly and thus stop the emulator. • You must have a trace to break correctly in a specific bank, otherwise your program will break at the normal 64K logical address for every bank. • The code might have never reached this point.

Problem:	I set a breakpoint and the emulation stopped a little bit further down the code than I had expected.
Cause/ Solution:	Do to the slight delay for the trace to compare the trigger condition and issue the break signal back to the emulator board, the execution of the program slides an instruction or two.

Appendix I. Emulator Board Communication EPROMs

Current Versions

Version #	Pod Types
1.4	Used for most all pods

Speciality EPROMs

Version #	Pod Types
1.41	POD–C517A
1.42	POD–C51SL
1.43	POD–5001
1.44	POD–407
1.45	POD–C575
1.46	POD–C320
1.47	POD–C520/530
1.48	POD–C576
1.49	POD–C542
1.60	POD–C51MX

Watchdog EPROMs

Version #	Pod Types
1.31	POD–C51GB
1.32	POD–C535, POD–C515A
1.33	POD–C552
1.44	POD–407
1.50	POD–C51RX, POD–51HB–C51RX / RX2

Appendix J. POD–C51MX Emulator Board Configurations

Emulation Setup

The EA series emulator board has two configurations as Shown in Figure 165 and Figure 166.

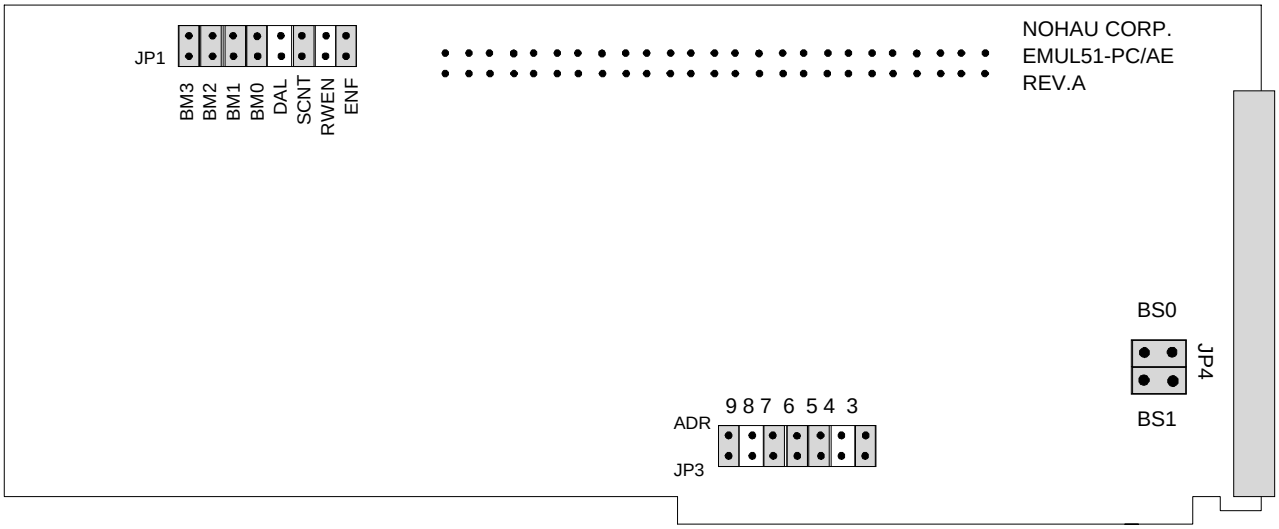


Figure 165. Configuration for the 256K Memory Model

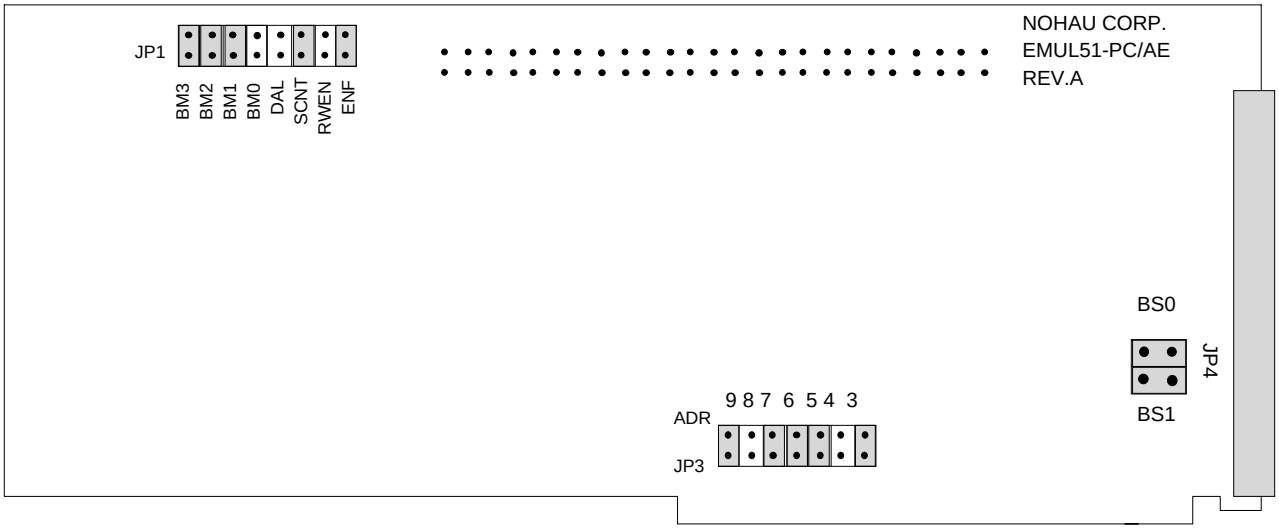


Figure 166. Configuration for the 768K Memory Model

BM3 and BM2 are not currently used with this emulation setup. BM1 determines if the memory spaces of code and data are overlaid or separate. BM2 recognizes if the board has 256K or 786K of emulation RAM. The positions for ENF, SCNT, RWEN, and DAL are factory set and should be left in their default positions.

BMO	BM1	Description of Function
In	**In**	256K emulation memory with separate CODE (128K) and DATA (128K).
In	**Out**	256K emulation memory with overlaid CODE and DATA all 256K.
Out	**In**	768K emulation memory with separate CODE (512K) and DATA (256K).
Out	**Out**	768K emulation memory with overlaid CODE and DATA all 768K.

Jumpers BS0 and BS1 should always be In so that the A16 and A17 lines are properly terminated.

Special Hardware Considerations

Due to current design information Nohau has found a difference in the operation of the P2 in reference to using the 8-bit MOVX operations (MOVX @Rx,A & MOVX @A,Rx). Do this issue if you have XDATA memory region mapped to the emulator and you attempt to use these operations they will fail and the data will be incorrectly written and read. Nohau has been able, with the current hardware, to make this operation work if the XDATA is mapped to the target.

Special Software Considerations

Currently, if you choose to use the option to trace the SFRs, the information is no decoded in the trace buffer. You will have to manually decode this information, as it will appear under the P1 and P3 columns.

The extended address information for operations, such as the EMOV instruction for the 51MX, will appear under the P3 column in the trace buffer display window.

Nohau does not support loading code to the target memory.

Support for the 51MX family is currently looked at as banked or paged memory to the emulator hardware. The extended address lines A16, A17, A18, and A19 support up to 512K of code memory; EA768 (eight banks of 64K code and 256K data) or EA256 (two banks of 64K code and 128K data).

The Shadow RAM on the EMUL51 is only 64K. This means that writes to addresses above 64K (FFFFh) will wrap on the 64K boundaries. For example, if you have a variable at address 0x0400 and another at address 0x20400, then they share the same Shadow RAM memory cell.

Currently problems displaying and watching variables classed as FAR full decoding of watch for proper memory space is under development.

Breakpoint and Trace Setup

- Breaks are handled by hardware on an array basis thusly you can have up to 64K breakpoints available. Additionally, with the trace you can set up a trigger and break on trigger for a write or

read access to XDATA with a specific value. There are eight additional qualifiers for these operations.

- Currently only the base 64K classic MOVX operations work the EMOV is not working for triggering on this transfer to/from the external data areas. (See the following “Configuring the FAR Variable” section.)

- Currently only the base 64K classic MOVX operations work the EMOV is not working for triggering on this transfer to/from the external data areas. (See the following “Configuring the FAR Variable” section.)

- Currently only the base 64K classic MOVX operations work the EMOV is not working for triggering on this transfer to/from the external data areas. Also you can not trigger on access to the classic DATA and IDATA locations or the SFR regions.

Configuring the FAR Variable

Currently the software does not automatically sets up the extended address of a FAR variable type. To set up the trace to trigger or capture information for this remove the M2 jumper on the pod. This will allow the emulator to trace the extended address lines. These lines appear under the P3 setup in the trace configuration and the P3 column of the trace display.

First find the address of the variable in question. This can be done by either using the symbol browser or the Inspect window. Figure 167 shows the variable address displayed in the Inspect/Watch_1 window.

In this example the variable class is HDATA and is at address 0x10000. Now set up the trace configuration to trigger on a write cycle to this address. First select **Break Emulator – On Trig**, and enable the trigger and in this case use Condition A (Figure 168).

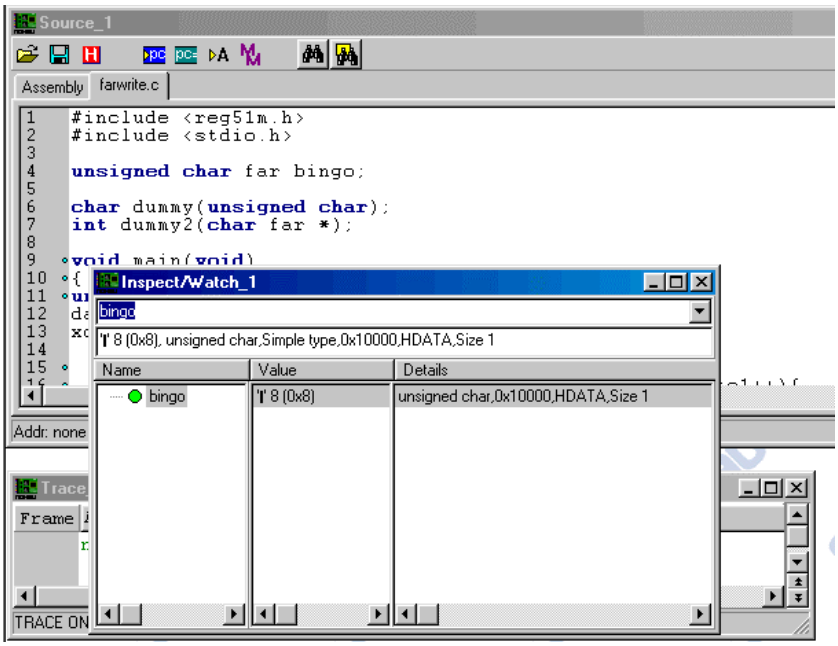


Figure 167. Variable Address Displayed in the Inspect/Watch_1 Window.

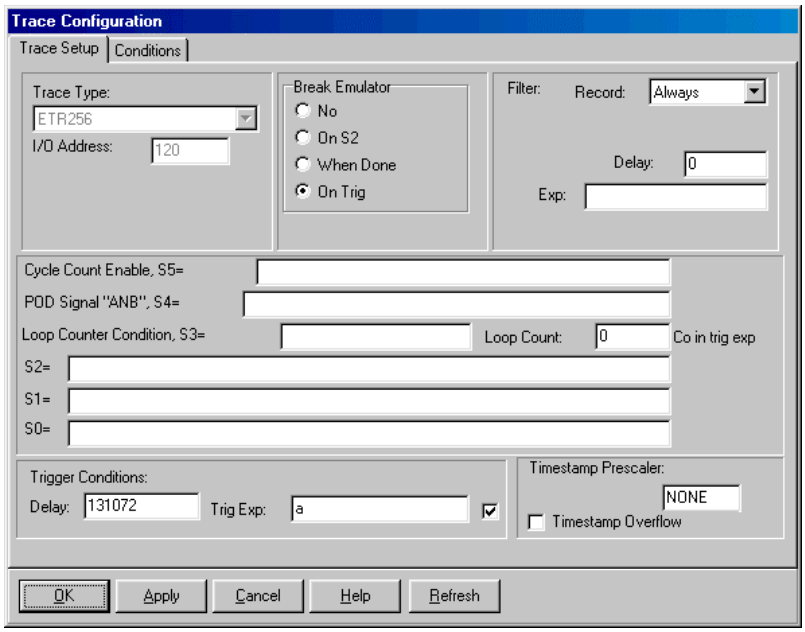


Figure 168. Trace Setup Tab

To configure Condition A, do the following:

1. Select the **Conditions** tab from the Trace Configuration window (Figure 169).
2. Double-click on line **A** to open the **Edit** dialog box for this condition (Figure 170).
3. Use the symbolic reference to enter the **Address** text box.
4. Set up the extended address for comparison in the **P3** and **P3 Mask** text boxes.
5. Select **Write** in the **Type** drop-down list in the **Miscellaneous** group. This specifies the variable at address 0x10000 on a write cycle to trigger the trace.

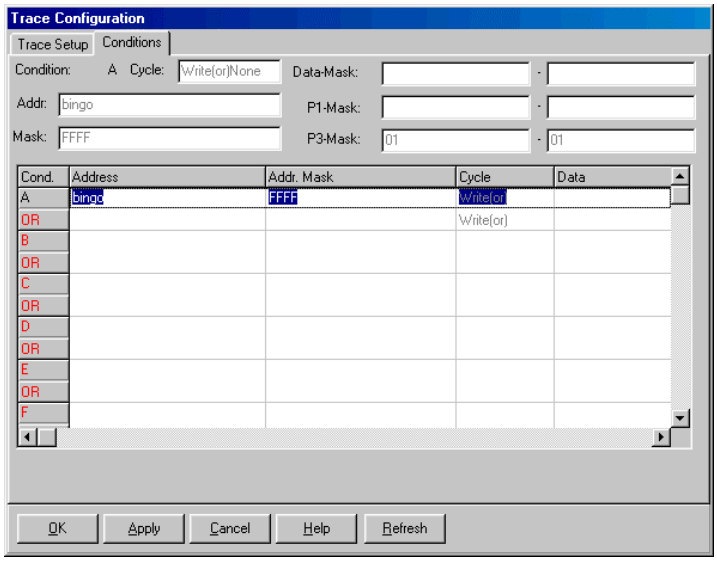


Figure 169. Conditions Tab

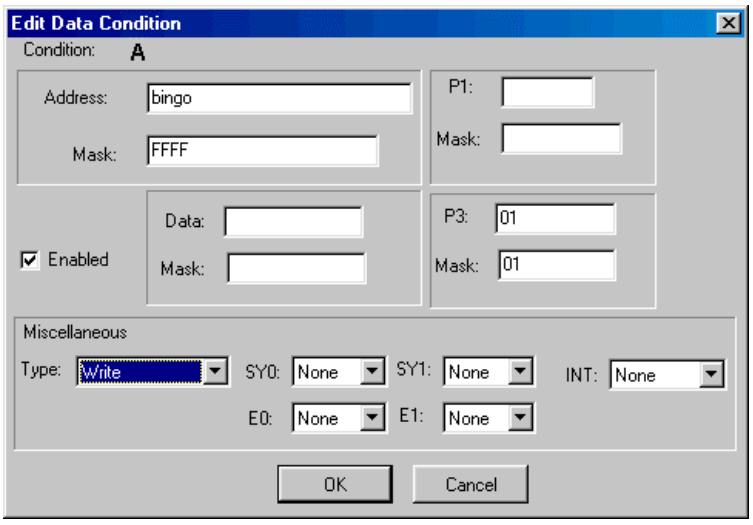


Figure 170. Condition A

Address	P3 Bit #
A16	0
A17	1
A18	2
A19	3
A20	4
A21	5
A22	6

Appendix K. Adding and Modifying Symbol Information

Functionality

The Src_AddSymInfo command works with the with symbol table information so you can create a new module by adding source lines and symbols information.

Parameter

String with symbol information that you can add or modify. This string can be in one of the following formats:

Syntax Example	Description of Function
module_name	Deletes all symbols from an existing module module_name.
module_name#file_name#symbol# symbol_type#address	Adds a symbol with an address to the module with module_name or modifies the address of the existing symbol. (Symbol name can not begin with a number.)
module_name#file_name###	Adds a module with module_name to the symbol table.
module_name#file_name#line#address	Adds a new line with an address to the module with module_name or modifies the address of the existing line. (Line number should be bigger than 0.)

Where:

- Module_name is the name of the module.
- File_name is the source file name with the full path associated with the module.
- Symbol is the symbol name.
- Symbol_type is the type of the symbol (for example, byte, word and long). If the type is not specified, the default is set to long.
- Line is the line number starting at 1.
- Address is the address of the symbol or line.

If the string has an additional pound sign (#) at the end of the string when adding a symbol, SymbolTableEvent will not be issued.

Return Type

None

Wait?

No

Remarks

After issuing a Src_AddSymInfo command, several windows, such as the Source window, and the Symbol Browser window will be updated. To prevent an update, add an additional "#" at the end of the parameter string (except in the case of the module cleaning). If a new symbol/line is added to the module that doesn't exist, this module gets created.

Examples

C:\Seehau\Examples\Nops.asm:

```
Start:      NOP
  NOP
  NOP
  JMP Start
```

To add symbolic information to the symbol table and write a program to the memory, use the following macro (it is like loading code, but without compiled program):

```
Sub Main
  Addr = 5000
  Src_AddSymboInfo
  ("nops#c:seehau\examples\nops.asm#Start#"
  + Hex$(Addr))  ' Adds lines with NOPs
MsgBox Src_AddrToLine("5000")          ' Message box displays "None"
End Sub
Sub Main
  Addr = 5000
```

```
  Src_AddSymInfo ("nops#c:\seehau\examples\nops.asm#Start#" + Hex$(Addr))   ' Add symbol Start
  ' Add lines with NOPs
  SLine = 1
  While SLine <= 3
Src_AddSymInfo "nops#c:\seehau\examples\nops.asm#" + Str$(SLine) + "#"+ Hex$(Addr) + "#"
        Addr = Addr + Src_Assemble (Hex$(Addr) + " nop")      ' Increase address
        SLine = SLine +1
  Wend
  ' Add last line
Src_AddSymInfo ("nops#c:\seehau\examples\nops.asm#"+ Str$(SLine) + "#" + Hex$(Addr))

Src_Assemble (Hex$(Addr) + " jmp Start")
End Sub
```

Appendix L. Code Coverage

Overview

The Code Coverage utility that is built-in to the EMUL51–PC product family requires the system to be configured with the enhanced trace board option. The hardware is equipped with different modes of operation for this feature.

- Code fetch only
- Any code memory access
- Any Write to external data (XDATA) memory space
- Any Read to external data (XDATA) memory space

Code Fetch Only

The first Opcode of an instruction or instruction sequence is the only valid capture to denote a covered instruction. In this mode, the operands to the opcode (if any) will not be flagged as executed. Given the detailed operation of the microcontroller the operands will always be executed when an instruction is being fetched for real execution. The hardware can tell if the difference to a pre-fetched instruction verses the true execution code fetch. This will cause the software to show that the instruction at an address that is greater than one (1) byte to be displayed as only partially executed. Although because of the operation of the microcontroller the full instruction has to be executed and will be executed even if an interrupt request appears during the time frame between bytes. In the case of the interrupt the micro will execute the vector after the current instruction sequence is completed.

Code Memory Access

This will ignore the opcode fetch flag and any read access to code memory will be displayed as a fully executed address. The only draw back to this mode is that if your code reads all of the code memory to do a Checksum validity test it will show all addresses as executed even if it was not.

Read / Write XDATA

These two modes are used to validate what addresses have been accesses in either a READ or WRITE operation in the off-chip XDATA memory space of the microcontroller.

Accessing Options

To access these options, do the following:

1. From the **Tools** menu select **Code Coverage** (Figure 171). The Code Coverage window opens.
2. Right-click to open the submenu and select **Options** (Figure 172.) The Code Coverage Options window opens (Figure xxx).
3. Select the **Memory Access** tab (Figure 173).
4. Select the cycle type on which you want to qualify your code coverage operation to process.

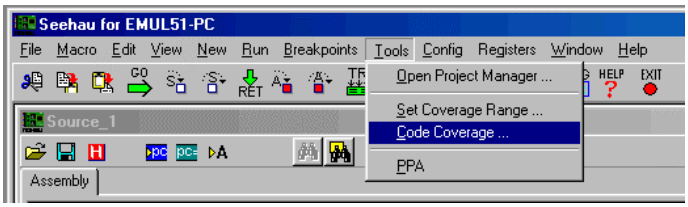


Figure 171. Selecting the Code Coverage Menu Item

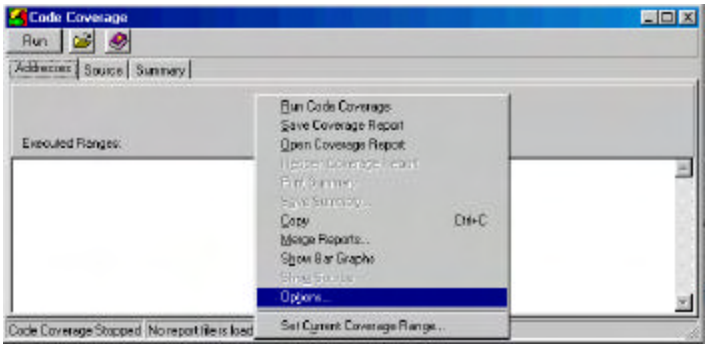


Figure 172. Selecting the Options Menu Item

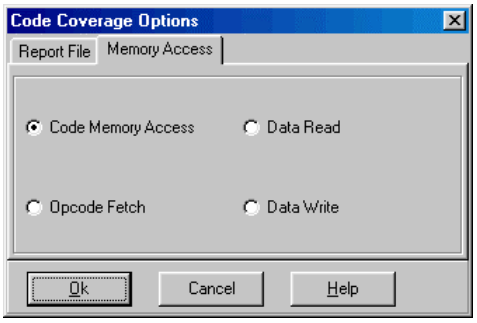


Figure 173. Code Coverage Options Memory Access Tab

Index

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

Z