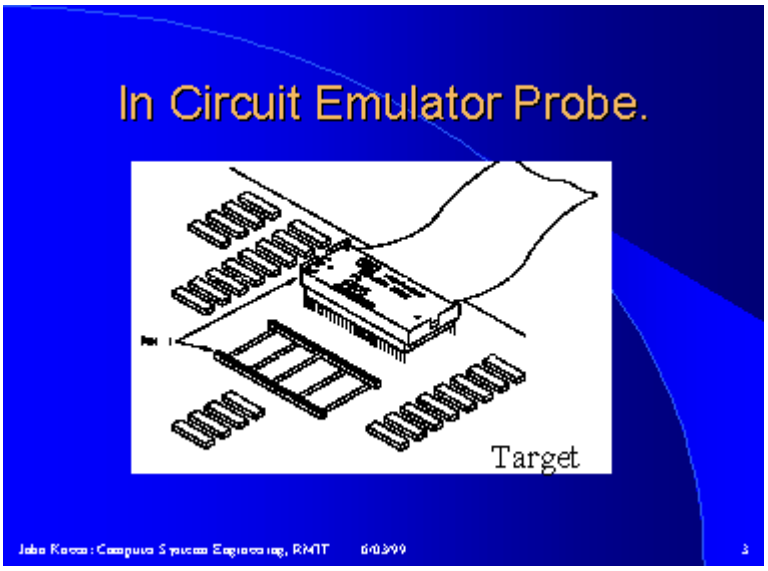
In Circuit Emulators

An ideal tool is one that provides visibility into the internal operation of the device or component being emulated. In circuit emulators are hardware tools that both provide that visibility and behave the same as the component being emulated. For microprocessor development there are two types of emulators; the in circuit emulator which is emulating the microprocessor, and, the ROM emulator which emulates the ROM. Historically these emulators were sold as part of a logic or microprocessor development system (LDS or MDS), today they are an extension pod for a PC or workstation, or even as a box on a network.

In-Circuit Emulators.

Introduction

An in-circuit emulator (ICE) consists of a small dual port pod. On one port is a probe that plugs into the microprocessor socket of the target system. The second port is interfaced to a workstation. A typical emulator probe is illustrated below:



The emulator probe ideally is physically and electrically identical to the target microprocessor. During system development the emulator probe will replace the target microprocessor. The operation of the target system will then be monitored and controlled via the interface to the workstation. Initially all the development will be on the workstation. As target resources become available then these will be utilized. Eventually the target hardware and software will be fully functional and the debugging capabilities of the in circuit emulator will be no longer required. The in-circuit emulator can then be removed and replaced with the actual microprocessor.

Features and capabilities available with an in-circuit emulator include:

- Ability to map resources between target and host. That is, programs may be run in either the target or the host. (The host memory is know as emulation or shadow memory). The mapping features provides the...
- Ability to run and test code in real time without target hardware.
- Ability to step or run programs from/to specified states or breakpoints (usually a specified address **but** it is the ability to specify other trigger states that gives the in-circuit emulator its capabilities over and above other debugging tools - the hardware

to provide a unique trigger point is similar to that required to trigger a logic state analyzer).

- Ability to observe and modify microprocessor registers.
- Ability to observe and modify memory contents. (Often in real time). This also allows exhaustive testing of the target hardware. See memory testing.
- Ability to trace program execution using internal logic analyzers. (see separate section on Logic Analyzers)

Note: Some of the ICE capabilities above are common to all debugging tools: single stepping, ability to display and modify both memory and registers, and to run programs. However it is only the ICE that allows testing of the software in real time prior to the target hardware becoming available.

Limitations of In Circuit Emulators

1. Availability and Cost

   Simple microprocessors are relatively easy to emulate, but there are several factors that reduce the potential markets and hence increase the cost

   a.For even simple microcontrollers there are many variants so different ICE hardware is required for each

   b.Modern cutting edge microprocessors are operating at very high speeds. To design emulators for these processors is very challenging and therefore extremely costly

   c.Other tools are available that handle many microprocessor development problems at much less cost so many large design teams may only need one ICE (reducing sales). The danger of this situation is that when the real difficult problems arise that require an ICE to solve no one has taken the opportunity to learn how to use its capabilities.

2. On Chip Functions
   1. The ICE capability is tied to the being able to monitor the microprocessor address, data and control buses. With modern processors on chip cache and input-output these lines are buried deep in the internals of the microprocessor so are not available to the ICE (or a logic analyzer).
   2. To address the above problems some microprocessors route internal buses to output pins (often at a performance cost so losing some of the real time capability) or aternatively other microprocessors supply BDM (Background Debug Mode) capability.

3. Transparency

   Transparency implies that the target system does not know if it is interfaced to an ICE or to the real microprocessor. Transparency is

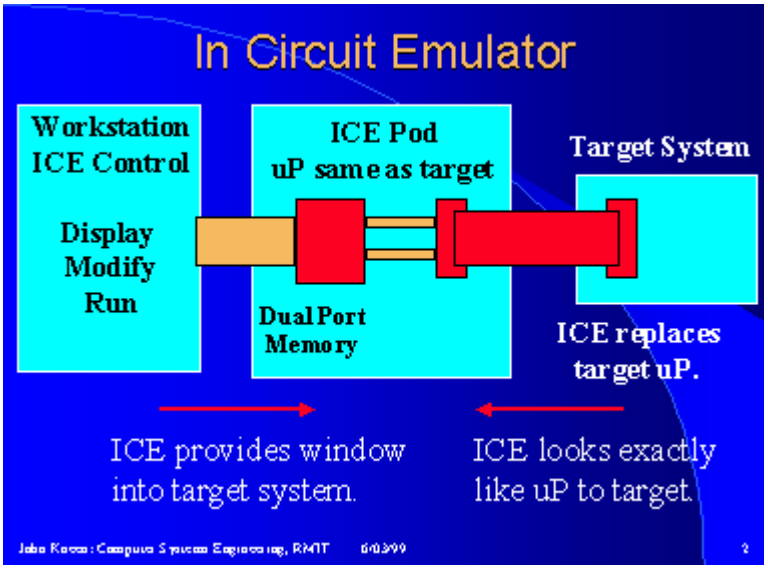
   a.Physical : basically this implies that the ICE has the same pin-out and fits into the same socket as the microprocessor being emulated. This is usually achieved using a remote pod (the emulator) and a short (shorter the better) cable connector.

   b.Electrical implies that the drive and timing capabilities of the ICE are identical to the microprocessor being emulated. The extra cababling and circuitry with the ICE will degrade both the loading and timing. Any target system operating at the margin could "fall over" with an ICE. If the hardware is developed using the ICE then the system can almost be guaranteed to work when the ICE is replaced with a real microprocessor. However, if the hardware has been developed without an ICE and

then an ICE is used late in the development to debug a software problem there may be problems with hardware that is operating at the limits of timing and loading. (Using faster memory for the prototype may give a platform where the ICE can be used for software debug.)

c.Functionality: This implies that all the pins are functionally the same. For most pins this is true but usually the ICE will use the Non- Maskable Interrupt for program control implying that it is not available to the target. Even the reset pin may be a compromise. To control the ICE from the workstation implies that the reset will come from the workstation and be an output pin at the ICE reset pin whereas in practice the reset is an input.

The ICE in More detail

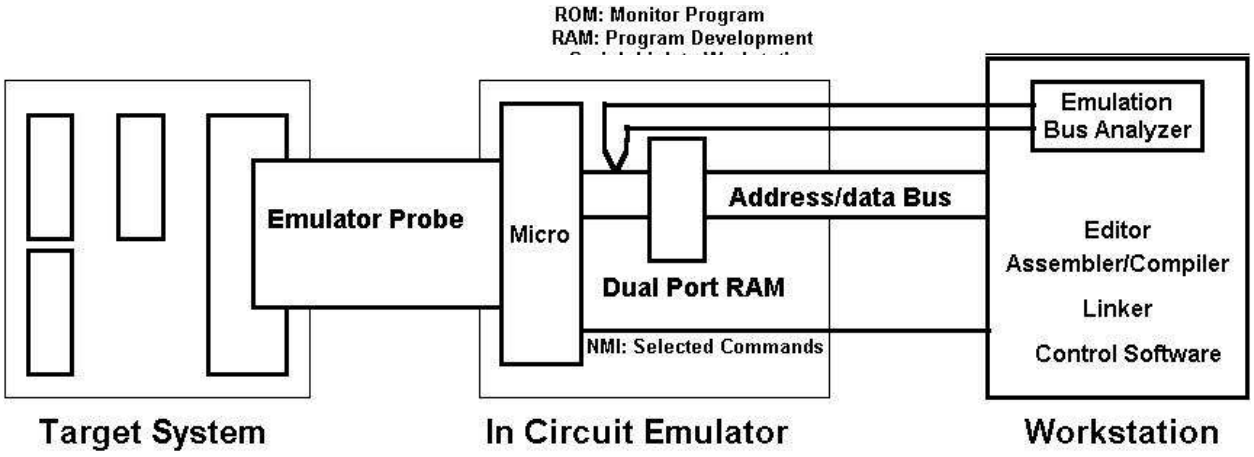


As noted one side the ICE interfaces to the target system and is ideally physically and electrically identical to the microprocessor being emulated. The second side provides an interface to a host computer or workstation allowing monitoring and control of the ICE. It is this link that provides the designer with a window into what is happening in the microprocessor system under development.

The layout of two in-circuit emulators is shown below.

Block Diagram of In-Circuit Emulators

The first design consists essentially of a microprocessor manufacturers evaluation board with a serial link back to a workstation. From the microprocessor there is a cable that plugs into the microprocessor socket on the target system. The "ICE" includes RAM for program development and ROM containing a monitor program to interface with the workstation. The monitor software contains routines that will modify/display memory or microprocessor registers and allow programs to be executed from an address to a break point. When ever a command is issued from the workstation the serial device will generate a NMI (non maskable interrupt) causing the microprocessor to break to the appropriate actions. All communications between the emulator and the workstation will be via the serial link.



(b) Advanced In Circuit Emulator.

The second design is typical of top of the range in circuit emulators. In this case all the emulator or overlay RAM is dual port so the microprocessor can continue with its normal operation while the workstation simultaneously accesses emulation or overlay memory. Since the microprocessor takes no part in the transfer of information between the host workstation and the emulation memory there is no need for a monitor program in ROM, the monitor program is loaded into RAM by the workstation at power up. When the microprocessor is required, for operations such as modify/display target memory, the workstation invokes the monitor by generating an NMI (or equivalent).

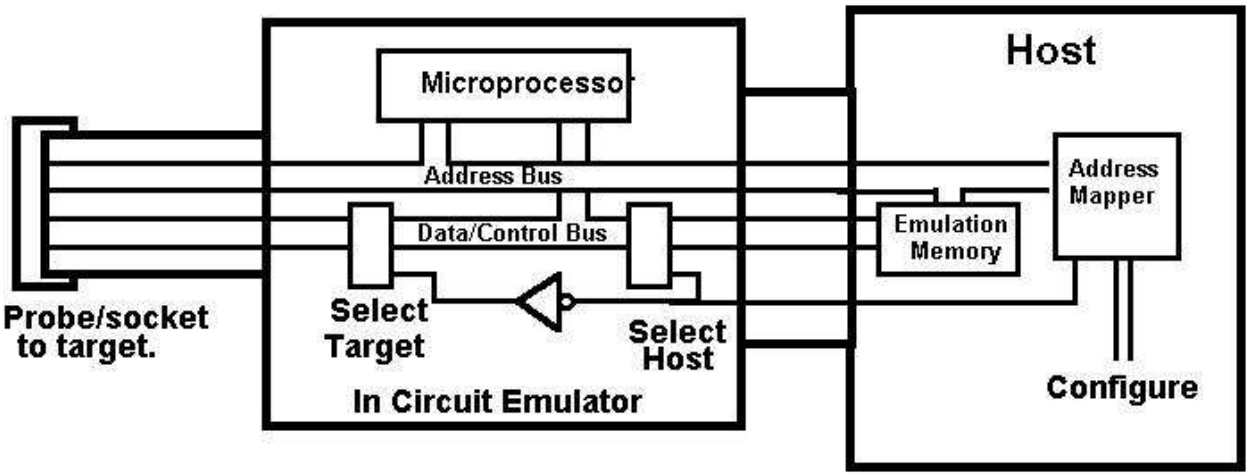
Most ICEs for general purpose microprocessors are constructed using the actual microprocessor with extra circuitry to provide the switching to/from the host workstation as required. For single chip microprocessors the chip manufacturer often provides a bond out version, where the internal address and data information is presented on additional pins. Alternatively, these features could be provided by emulating the microprocessor using ASIC devices.

The quality, and therefore the cost of an in-circuit emulator, depends very heavily on how closely the in-circuit emulator matches the target processor. Ideally to the target system it should be transparent whether there is an in-circuit emulator or the actual microprocessor that is plugged into its processor socket. In practice perfect emulation will be impossible, compromises in the design will be necessary.

Memory Mapping

Included with the ICE is emulation or overlay memory, so except for the I/O functions, users can develop and verify all of their code on the ICE prior to any hardware becoming available. An important part of the ICE is the address mapping circuit. As shown in the figure below the address mapper is used to select what resources the ICE will use. Examples of how the mapping may be used are:

- Before any target becomes available the mapper will point to emulation memory. As some of the target becomes available the memory mapping circuit will be configured to select between emulation or target resources. For example, when the microprocessor accesses address location 1000H the mapping circuit might enable target resources, while for address 2000H resources within the emulator are utilized. As more of the target becomes available, resources are moved from the emulator until when the design is complete the ICE will be utilizing only target system resources. The ICE is then removed from the target and replaced with the actual microprocessor. The target will then operate as a stand alone system.
- For system maintenance the ICE would be plugged into the target. The program area would be mapped to emulation or overlay memory. So instead of executing code from the target ROM(s) the program would be executed from emulation memory. It could then be upgraded and tested as required, utilizing all the capabilities of the ICE, before burning replacement ROMs.



Address Mapper and the In Circuit Emulator.

Note in the figure emulation memory and the address mapper are shown located in the host. For state of the art microprocessors, to enable high speed operation, minimum signal paths are essential so these components are located in the probe as close to the microprocessor as possible.

Conclusion

The major advantage of using an in circuit emulator are:

- Software development, including testing in real time, can occur prior to any hardware becoming available;
- The ICE provides facilities to modify/display memory and registers. These facilities can be used to more confidently verify the operation of the system hardware and software.

The quality of an in circuit emulator depends to a large degree on how closely it matches the microprocessor being emulated. Usually the better the match the higher the cost. With some of the more modern microprocessors it has become prohibitively expensive to produce high quality in circuit emulators so designers sometimes have been forced to do their development by other means.

ROM Emulators.

Similar to the in circuit emulator the ROM emulator replaces some of the target resources, in this

case the ROM. During program development the code is loaded into the ROM emulator from the host station and the target microprocessor reset line released. The microprocessor should then execute the code that is in the "ROM". The program execution can be captured and verified using a logic analyzer. If the analyzer is integrated into the host machine it will monitor all activity that occurs on the pins of the ROM.

With the ROM emulator flying leads are necessary to both control the microprocessor and to capture information on the remaining address and control lines. For example, the reset line will be necessary to force the microprocessor into its reset state while an NMI would be used to force the microprocessor to break to a monitor program. Since the microprocessor read-write control line is not normally wired to a socket designed for ROM a flying connection is necessary to capture the status for use by any logic analyzer.