

Application Note:

What can a proper JTAG emulator do for me that a simple parallel-port cable cannot?

Q: When a development board is available, it is common practice to connect this to the development system using Ethernet or RS232. These services are not available until the board has booted, so a simple parallel-port JTAG interface is often used to download a bootloader first.

This set-up works perfectly well, so why spend money on a "JTAG emulator"? What additional benefits can such a device offer?

A: Although it's indeed of only limited use with a standard development board running an existing Board Support Package (BSP), a JTAG emulator really comes into its own when custom hardware is being built, or when a BSP, bootloader or drivers are to be developed.

The main advantages of using a JTAG emulator are:

You can bring up previously untested hardware

When faced with a new hardware prototype, and you would like to download test programs to the Flash and/or RAM, quite often design or manufacturing flaws will mean that when you attempt this, nothing happens.

A JTAG emulator is designed for this situation, and will always be able to communicate with and control a processor that is powered up, in order to begin the process of debugging a new board.

You can use it to build and debug a bootloader

Often, when faced with a new piece of hardware, or a standard piece of software onto which an OS is to be ported, you have no bootloader. At best you will have a bootloader which works for a similar system.

You will be able to download this bootloader to the target, but if there is any sort of problem, it will simply hang, and the target will not boot. At this point you have no good debug access to the target.

A JTAG emulator will give you powerful, fast and comprehensive source-code debugging during bootloader execution, accessing all the device's on-chip debug features. This contrasts with bootloader debugging via the parallel-port cable, which may be difficult to set up, slow, and limited in features.

You can use it before you have implemented an Ethernet, RS232 or other suitable interface

When developing new hardware, it is often desirable to run the prototype target hardware at an early stage. This target may not include an appropriate high-speed interface that can be hijacked to connect to the development system.

If there is a suitable interface, then it may not be working, or physically accessible in an early prototype.

Even if there is a suitable interface, which is accessible, that interface might need to be hooked up as intended in the product under development. For example, a handheld device will probably not have Ethernet at all, but might have an RS232 interface intended for connection to a GPRS modem. You can certainly borrow the RS232 interface to connect to the development system, but then your prototype will not be talking to its modem!

It can be used during driver development, when other interfaces don't work

Even if the prototype target has a suitable communications port which can be borrowed to connect to the development system, this will only work if a driver has been written.

For example, when it is time to test the driver for the Ethernet port on the prototype hardware, you cannot rely on that Ethernet port as the means of connection between development system and target.

The target is not "hanging" from the PC, but is instead connected via Ethernet (or USB)

This point also applies with development boards, and may seem trivial. However it is important to note that the need for a short ribbon cable connection between PC and board very often ends up with wasted time when connectors stress and fail.

Also, if the connection between target and PC is via Ethernet, debugging can be remote, and shared over multiple PCs.

The speed of communication is very high, so you don't need to switch to Ethernet once the bootloader is installed

You can look at the JTAG emulator as a fast Flash programmer. You can trash your Flash contents and make the system unbootable one minute, and with a couple of clicks reprogram the entire OS image via JTAG the next.

On-chip trace functions can be accessed, dramatically improving debugging options

Many 32-bit processors, including all Intel XScale devices, have an on-chip debug facility that includes internal trace capture. It is not possible to access this functionality via a simple JTAG cable. With access to the trace facility, your debugging options expand dramatically, allowing you powerful breakpoint capabilities.

Plug-in software is supplied which allows seamless debugging from within Platform Builder, GDB, XDB and other debuggers

Although it is possible to use most debuggers with a simple cable, JTAG emulators ship with software that allows you to access the full range of on-chip debug facilities, including trace, from your debugger of choice. For example, with the EPI MajicMX system, Platform Builder suddenly has a "Trace" window.

You can access professional tech. support, and will never be on your own with a non-working target that you can't bring up

You can spend hours or days scratching your head over a non-working board, and have to break out the logic analyzer. With a JTAG emulator, you have access to the on-chip debug functions whenever the device is powered up.

A real example:

A Direct Insight customer recently developed a PXA255 design that was very similar to one of our reference designs, and it seemed that there was no reason why some simple modifications to the bootloader should not allow it to boot. However, the bootloader crashed, offering no help as to what might be the cause.

An emulator was connected, and the debugger was set to break on the exception vector generated by the crash.

Immediately, it was possible to view the source code executed up to the point of the crash, as well as the register contents. Even though the user had not written the bootloader himself, it was very easy to see the problem, and make the required tweak.

Thus a potentially thorny problem was solved in minutes.

If you would like to talk to one of our embedded engineers about the issues raised in this note, please call **+44 1280 700262** and ask for embedded support, or email **support@directinsight.co.uk**