

DESIGNER'S NOTEBOOK



Emulator Processor Access Timeout

Contributed by Rosemarie Piedra

Design Problem

What would cause a "processor access timeout" message in the XDS510 emulator command window? Is the address provided in the error message meaningful?

Solution

There are two basic causes:

1) The device is in reset.

Attempts to execute (single-step or run) a program when the DSP is in reset may cause the "processor access timeout" message to be displayed in the COMMAND window.

The device will always "time out" when using "run" or "runf." When single-stepping, the situation could be dependent on the DSP silicon version you are using. For example, in the case of the 'C4x PG2.x silicon, you may be able to step without the error message. That is not the case in PG1.x or PG3.x or above.

2) A current access that is being truncated.

The debugger will break any pending CPU/DMA access that is not completed within a timeout period (one second) during single-stepping or after an emulator halt. This could happen in the following situations:

A) Access to a device peripheral that is not ready.

For example, in the 'C4x, when DMA/CPU reads from an empty IFIFO, or when the DMA/CPU writes to a full OFIFO.

Solution: Check the level of the comm ports before accessing it. In the DMA case, you should use DMA synchronization.

Data in the comm port's input FIFOs can only be read once. The debugger displays memory values by reading them from the 'C4x. It is best, therefore, to avoid displaying the input FIFOs in any of the debugger windows as this will cause the data to be unavailable to your program. In the example emunit.cmd file shipped with the debugger, the memory map commands that define the comm ports' FIFOs are commented-out to avoid this problem, but still show how they might be defined if necessary.

B) During execution of a large “Repeat-Single” instruction.

The “repeat-single” instruction and the instruction that is being repeated is considered one single instruction from the emulator’s point of view. The emulator can only single-step between instruction fetches. In the case of the RPTS (‘C4x/’C3x repeat-single), the instruction that is being repeated is fetched only once.

C) During interlocked instructions or during any instruction that access memory is not ready.

WARNING: Different debugger versions may present a different behavior:

For example, in the ‘C4x XDS510, debugger versions 2.20 or higher will send a “processor access timeout addr=xxxx” message if a read or write access doesn’t complete. However, versions 2.01 and lower may not send any warning message if a read access is broken. This “incomplete read” can be misinterpreted as an access completion.

The addr=xxxx provided in the error message will approximately correspond to the address where the timeout occurs when this is the result of a “debugger” access timeout, for example, when displaying memory that is not ready. In the case of a CPU/DMA timeout, what you probably will receive is an address = 0xdeadc0de that is not meaningful. In this case, the previous 3–4 instructions (in the case of a DSP with a 4-deep pipeline) to the PC value should give you an indication to where the timeout occurs.

Sometimes a different message could be caused by the same problem. An “invalid operand” message coming from the debugger expression analyzer could be an indication that a timeout happens. For example, when you type “? *0xxxxxx” in the command window and you receive the “invalid operand” message, this may imply that the debugger is timed out when accessing that memory location.