

*TMS320 DSP  
DESIGNER'S NOTEBOOK*

# ***Emulator Processor Access Timeout***

---

---

---

*APPLICATION BRIEF: SPRA248*

*Rosemarie Piedra  
Digital Signal Processing Products  
Semiconductor Group*

*Texas Instruments  
November 1994*



## **IMPORTANT NOTICE**

Texas Instruments (TI) reserves the right to make changes to its products or to discontinue any semiconductor product or service without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current.

TI warrants performance of its semiconductor products and related software to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Certain application using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications").

**TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS.**

Inclusion of TI products in such applications is understood to be fully at the risk of the customer. Use of TI products in such applications requires the written approval of an appropriate TI officer. Questions concerning potential risk applications should be directed to TI through a local SC sales office.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does TI warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used.

## **TRADEMARKS**

TI is a trademark of Texas Instruments Incorporated.

Other brands and names are the property of their respective owners.

## CONTACT INFORMATION

US TMS320 HOTLINE	(281) 274-2320
US TMS320 FAX	(281) 274-2324
US TMS320 BBS	(281) 274-2323
US TMS320 email	<a href="mailto:dsph@ti.com">dsph@ti.com</a>

## Contents

<b>Abstract.....</b>	<b>7</b>
<b>Design Problem.....</b>	<b>8</b>
<b>Solution.....</b>	<b>8</b>

# Emulator Processor Access Timeout



## Abstract

This document discusses what would cause a “processor access timeout” message in the XDS510 emulator command window. Also discussed are issues related to the validity of the error message information.



## 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 PGI.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 emuinit.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.

**NOTE:**

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 = 0xdeadcode 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.