Order this document
by EB305/D

Motorola Semiconductor Engineering Bulletin

EB305

Startup Problems When Using a Software Background Mode Debugger and Booting from RAM or an Empty ROM Socket

By Charles Melear
Austin, Texas

General Information

When booting from RAM using a background mode debugger, several concerns must be taken into account.

The method used by many in-circuit debuggers for M68300 and M68HC16 MCUs is to invoke the MCU's background mode. But there are several other methods for entering background mode. For instance:

- One method is to purposefully create a double bus fault. This can be done by asserting bus error (BERR) during two consecutive fetches.
- Another method is to deliberately cause the MCU to jump to an odd address for a program fetch. This will cause an address error. Then, BERR can be asserted during the stacking operation for the address error exception. This causes the double bus fault, sending the MCU into background mode.
- Another method is to execute a BGND mode instruction.

MOTOROLA

- Background mode can also be entered by holding the BKPT (breakpoint) pin low at the release of RESET. When this is done, the MCU responds by fetching the first four long words and the first word of the instruction pointed to by the program counter. If an 8-bit bus is being used, the MCU will fetch the first eight bytes and the first byte of the instruction pointed to by the program counter.

At this point, the background debugger causes the MCU to fetch a few instructions, starting at the address pointed to by the program counter, and then to display these instructions on the computer's display. If a valid program is resident, along with valid values for the stack pointer and program counter starting at address $00000000, everything should work fine.

When some engineers begin debugging a circuit using RAM (as opposed to ROM) for the memory device selected by CSBOOT or there is not a program in the ROM, erratic behavior almost always results when the background mode is entered.

This happens because random values are fetched for the stack pointer and the program counter. It is very likely that this value will be $FFFFFFFF as the data bus has weak pullup devices. As both of these values result in odd addresses, the program fetches will be from odd addresses and cause address errors. Also, the value fetched from a blank RAM or empty ROM socket almost certainly will be outside the first megabyte of memory.

When these fetches are made, no DSACK is generated to terminate the bus cycle, and the debugger software will force a bus error. Thus, when the debugger starts trying to read memory at these bad addresses, there probably will be a series of messages on the computer screen, executing the debug software, that say: Memory implementation error: debugger supplied DSACK.

If no background mode software is used to control the MCU and the first program fetch is outside the first megabyte of memory, no internal DSACK will be generated to terminate the bus cycle and the software watchdog timer will time out and cause a reset.

After the last fetch, the error messages will stop.

After the debugger software has finished making the scheduled number of program fetches, the MCU will be in background mode waiting for the next command from the debugger software. However, errors probably will result if the next command is an external memory access because there is no valid program counter value and no valid stack pointer value.

When using a software background mode debugger to boot an M683xx level device from RAM or an empty boot ROM socket, it is imperative that these two actions are taken immediately after starting the background mode software:

1. Load a value into address register A7 to serve as a stack pointer value. It must point to a modifiable memory address.
2. Load the address of the first instruction to be executed into the program counter.

When using a software background mode debugger to boot an M68HC16 device from RAM or an empty boot ROM socket, it is imperative that these three actions are taken immediately after starting the background mode software:

1. Load the stack pointer with a value that points to a modifiable memory address.
2. Load the SK register so that the proper bank of memory is selected for the stack pointer.
3. Load the instruction pointer (not the program counter) with the address of the first instruction to be executed.

After the appropriate registers have been loaded for the selected MCU, the software background mode debugger should work reliably. That is, programs can be downloaded into the RAM and executed.

Engineering Bulletin

How to reach us:

USA/EUROPE/Locations Not Listed: Motorola Literature Distribution, P.O. Box 5405, Denver, Colorado 80217, 1-800-441-2447 or 1-303-675-2140. Customer Focus Center, 1-800-521-6274

JAPAN: Motorola Japan Ltd.: SPD, Strategic Planning Office, 141, 4-32-1 Nishi-Gotanda, Shinagawa-ku, Tokyo, Japan, 03-5487-8488

ASIA/PACIFIC: Motorola Semiconductors H.K. Ltd., Silicon Harbour Centre, 2 Dai King Street, Tai Po Industrial Estate, Tai Po, New Territories, Hong Kong, 852-26629298

Mfax™, Motorola Fax Back System: RMFAX0@email.sps.mot.com; http://sps.motorola.com/mfax/; TOUCHTONE, 1-602-244-6609; US and Canada ONLY, 1-800-774-1848

HOME PAGE: http://motorola.com/sps/

MOTOROLA