Freescale Semiconductor
Engineering Bulletin

Document Number: EB740
Rev. 0, 04/2011

Detailed Information for Erratum SECF196

This engineering bulletin provides additional detail and clarification on erratum SECF196 affecting the ColdFire MCF532x and MCF537x devices. The objective of this document is to explain the cause and effects of the issue in greater detail and provide information on how systems might be impacted.

Contents

1 Description

During a power-on reset (POR), some power ramp and clock startup sequences can cause some pins to drive undefined values for a period of time during the reset sequence. Undefined signal states can be driven on FlexBus, SDRAMC, and BDM pins if the device exits POR with no internal clock present.

Initially the pins are tri-stated while IVDD and EVDD/SDVDD are not fully powered. As the voltage rails ramp to the operating levels, the processor releases the pins from tri-state and the pin states are determined by digital control logic. However, the digital logic needs

freescale semiconductor

several clock cycles to initialize, so the pins can be driven by uninitialized logic for a period of time while the digital logic initializes to the reset state.

1.1 Timing diagram

Figure 1 shows an example of power-on reset (POR) timing that causes undefined signal states on FlexBus, SDRAM, and BDM pins. For clarity the diagram includes POR and pad enable signals. These are internal signals of the processor, so they are not visible externally.

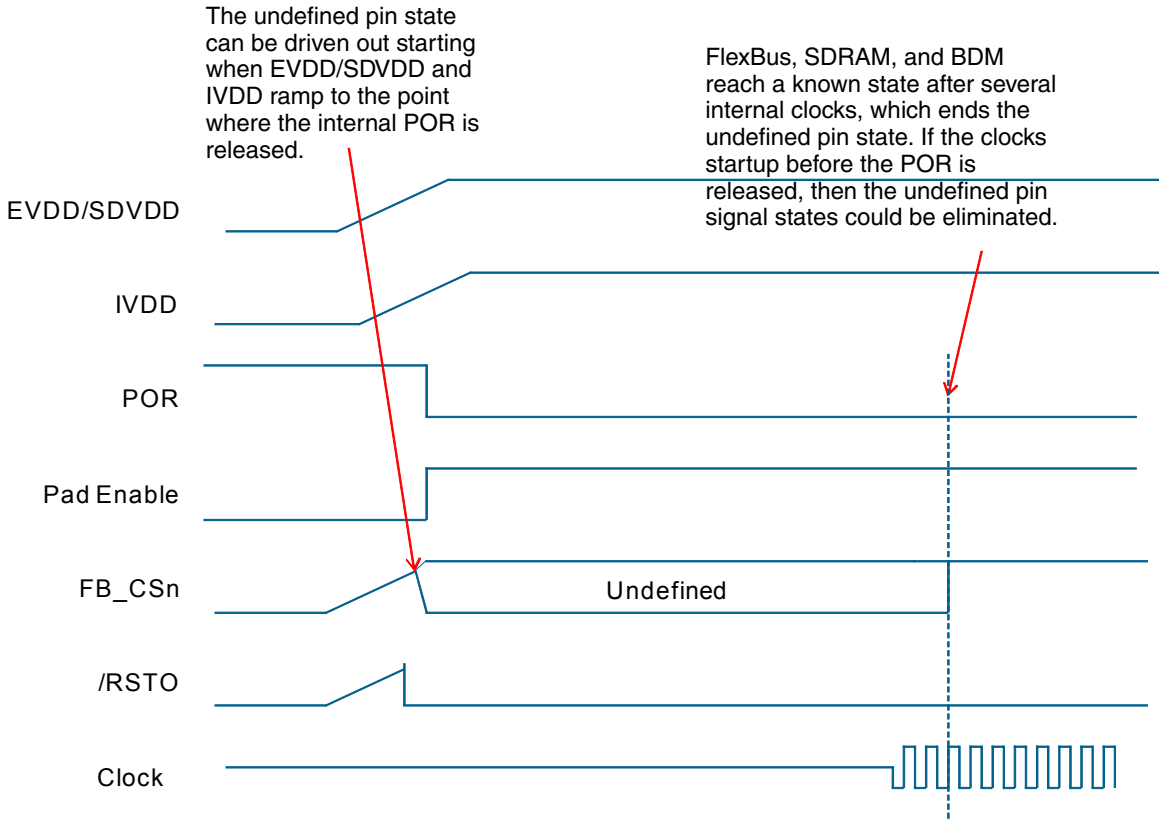

Figure 1. Power-on reset sequence causing undefined signal states

1.2 Affected signals

Table 1 lists the FlexBus, SDRAMC, and BDM signals and describes how they are impacted.

Table 1. MCF532x and MCF537x signals that can have undefined states during POR

Module	Signals	Undefined?	Description
FlexBus	A[23:0], D[31:0], BE/BWE[3:0], OE, R/W, TS, FB_CS[5:0]	Yes	Most FlexBus signals are in an undefined state any time between the release of the internal POR signal and clock startup.
FlexBus	TA	No	The TA signal is always an input and it will be configured as an input at POR; therefore, this signal is in a defined state.
SDRAMC	SD_A10, SD_CLK, SD_CS[1:0], SD_DQS[3:2], SD_SCAS, SD_SRAS, SD_SDR_DQS, SD_WE	Yes	Most SDRAMC signals are in an undefined state any time between the release of the internal POR signal and clock startup.
SDRAMC	SD_CKE	No	As part of the fix for an earlier erratum (SECF045) on any 3M29B and later devices, the SD_CKE signal is forced to drive high when the internal POR is released. **Note:** On 2M29B and earlier parts the state of SD_CKE can be undefined.
SDRAMC	SD_CLK	No	As part of the fix for an earlier erratum (SECF045) on any 3M29B and later devices, the SD_CLK signal is driven with a digital version of the input clock on EXTAL when the internal POR is released.
BDM/JTAG	PSTCLK, DSO, DDATA[3:0], PST[3:0], ALLPST	Depends on JTAG_EN	If JTAG functions are selected (JTAG_EN = 1), then pin states are defined. If BDM functions are selected (JTAG_EN = 0), the BDM outputs are undefined until clock start.
BDM/JTAG	DSI, BKPT	No	These signals are inputs in both BDM and JTAG mode, and they will default to inputs when the internal POR is released.
BDM/JTAG	JTAG_EN	No	This input signal is used to select between BDM and JTAG functions. It is configured as an input and sampled when the internal POR is released.

2 System impact

Depending on how the processor signals are used in the rest of the system, an undefined pin state could cause unexpected behavior for a device external to the processor. Because the undefined pin states are temporary (pins will enter a defined state before the processor exits reset), in many cases the undefined pin states will not cause a functional issue for the system.

2.1 FlexBus impact

The possibility of impact to devices on the FlexBus depends on the types of memory used in the system. Table 2 lists some common devices that might be used in a system and how they could be impacted by undefined signal states.

Table 2. Potential impact of undefined signal states on common FlexBus devices

Device type	Impacted?	Comments
NOR flash	No	Because NOR flash devices require a specific sequence with multiple steps to erase or write memory, the undefined signal states cannot corrupt NOR flash contents.
SRAM	No	Because SRAM contents are undefined at power-on reset, any accidental writes to RAM caused by the undefined signal states should not impact the system. **Note:** If the SRAM has a battery backup or is powered independently of the processor such that its contents are defined at the time the processor goes through a power-on reset, then data corruption is possible.
MRAM/FRAM	Yes	Any nonvolatile memory device with an asynchronous bus interface could be impacted. Because the memory contents are retained, accidental writes to the memory caused by the undefined signal states could corrupt the non-volatile data.
CPLDs, FPGAs, and external peripherals	Depends	Because the use of these devices can vary and in some cases is user-defined, we are unable to make an accurate assessment of how these types of devices could be impacted.

2.2 SDRAM impact

The SDRAMC on the MCF537x/MCF532x family is a special case. To fix an earlier erratum (SECF045), the SD_CLK signal is driven as soon as clocks are detected and SD_CKE is driven high. Because of this fix, there will be a short overlap between undefined pin states on the SDRAM control signals and SD_CLK starting up. This can allow an SDRAM to latch an erroneous SDRAM command.

During initial startup, the SDRAM doesn't allow any commands except NOP and COMMAND INHIBIT. Undefined signal states on the SDRAMC signals could potentially be recognized as other commands and cause undefined operation of the SDRAM. If the undefined operation of the SDRAM causes the SDRAM to drive the data bus, it could potentially cause interference with chip configuration and code fetches.

2.2.1 Non-split bus configurations

If a non-split bus configuration is used, then the SDRAM shares data lines with the FlexBus. Data line contention between the SDRAM, FlexBus devices, and/or the MCU is possible. If a 32-bit wide SDRAM is used, then it could potentially cause interference with latching of chip configuration values on the FB_D[9:1]/RCON[9:1] signals. Contention while latching chip configuration values and/or boot code could result in boot failures.

2.2.2 Split bus configuration

If a split bus configuration is used, then the SDRAM bus has a 16-bit wide dedicated data bus. Because the data lines are not shared with the FlexBus or chip configuration pins in this configuration, no issues are expected as a result of the undefined signal states on the SDRAMC signals.

2.3 BDM impact

If the BDM signals are enabled (JTAG_EN = 0), then BDM outputs could also drive undefined signal states during reset. Because the BDM signals are not multiplexed with any module pins, in most systems these signals would route only to a BDM/JTAG header. Therefore, undefined pin states on BDM signals are not expected to cause functional problems for systems.

3 Workarounds

The undefined pin states occur only during a power-on reset (POR) — the /RESET and/or /RSTOUT signals can be used to qualify FlexBus and SDRAMC signals. In most cases, the only signals that will need to be qualified are the FlexBus chip select and SDRAM chip select signals. By using qualified versions of the chip selects to connect to external devices, you can prevent those external devices from sampling the undefined pin states on address, data, and other control signals.

3.1 Workaround step 1: Generate a qualifier signal

The first step to the workaround is to generate a signal that can be used to qualify or mask the undefined signal states on the chip select signals (or other control signals as needed). Determining the qualifier to use can require some characterization of power and reset signal operation in the system. The idea is to find a signal or combination of signals that remains high throughout the time when the undefined signal states can occur.

Theoretically the $\overline{\text{RSTOUT}}$ signal could be used to qualify signals; however, because $\overline{\text{RSTOUT}}$ asserts low at the same time that the undefined signal states start to drive on other pins, if this signal were used as a qualifier alone, there could be small glitches while waiting for $\overline{\text{RSTOUT}}$ to propagate through logic. For this reason Freescale doesn't recommend using $\overline{\text{RSTOUT}}$ on its own as a qualifier.

If the system keeps the $\overline{\text{RESET}}$ input signal to the processor asserted during system power-up and through the start of clocks, then the $\overline{\text{RESET}}$ signal could be used to generate the qualifier signal as shown in Figure 2.

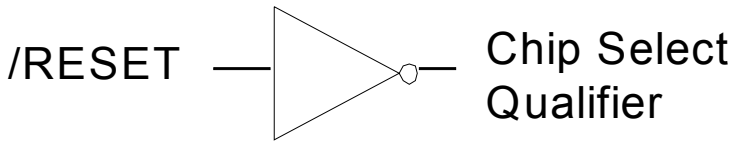

Figure 2. Generating a qualifier signal using $\overline{\text{RESET}}$

If the system is not guaranteed to keep the $\overline{\text{RESET}}$ input signal asserted through the start of clocks, then a combination of the $\overline{\text{RESET}}$ and $\overline{\text{RSTOUT}}$ signals could be used instead as shown in Figure 3. It is important that there be a period of time during which $\overline{\text{RESET}}$ and $\overline{\text{RSTOUT}}$ are asserted simultaneously, to avoid possible glitches.

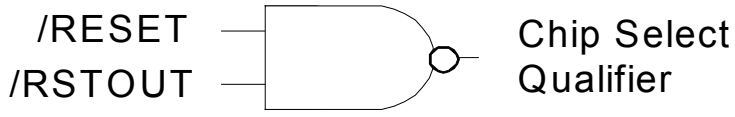

Figure 3. Generating a qualifier signal Using $\overline{\text{RESET}}$ and $\overline{\text{RSTOUT}}$

3.2 Workaround step 2: Qualify signals

For many systems there might be only a small number of chip selects or other control signals that need to be qualified to prevent the undefined signal states from reaching external devices. If there are only a small number of signals that need to be qualified, then adding discrete logic gates might be the best solution. Figure 4 shows an example of how a chip select signal could be qualified using an OR gate.

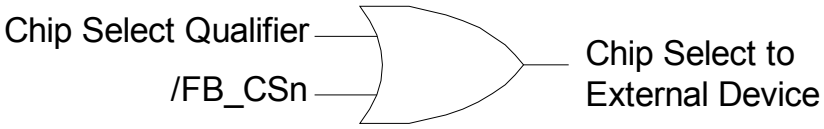

Figure 4. Qualifying a chip select using an OR gate

There are up to eight chip select signals available on the MCF532x/7x family devices — six FlexBus chip selects and two SDRAM chip selects. This means that all of the chip select signals on the processor could be qualified using a single 8-bit tri-statable buffer as shown in Figure 5.

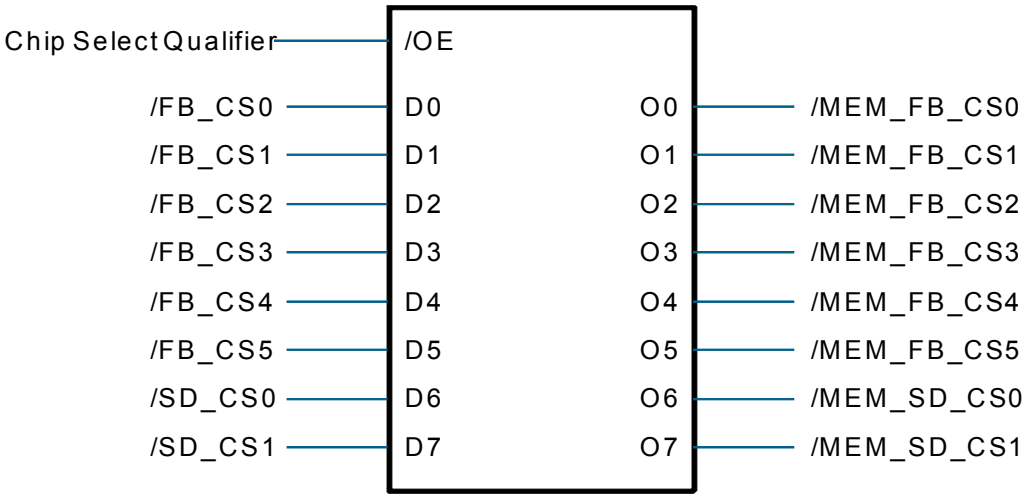

Figure 5. Qualifying all chip selects using a buffer

In Figure 5, “MEM_” is used to differentiate the chip select signals from the processor and the qualified versions that would be connected to external memories. Each of the “MEM_” signals should include a pullup resistor to force the signal high while the buffer is disabled.

THIS PAGE IS INTENTIONALLY BLANK

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor, Inc.
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Document Number: EB740
Rev. 0
04/2011

freescale™
semiconductor