AMD® 29K™ Family

Technical Bulletin

Synchronizing $\overline{\text{RESET}}$ and BOOTW to Avoid Misconfiguring ROM Bank 0 Width

EPD Systems Engineering

November 15, 1993

Purpose

This bulletin replaces the technical bulletin entitled *RESET-BOOTW Synchronization*, PID No. 18011, and its first revision, *RESET-BOOTW Correction*, PID No. 18011A. This revision includes the Revision B Am29200 microcontroller as an affected part, and incorporates minor, nontechnical changes throughout.

The 29K Family microcontrollers support 8-, 16-, and 32-bit-wide ROM banks. Configuring an 8-bit-wide boot bank (ROM Bank 0) requires that BOOTW and $\overline{\text{RESET}}$ be tied together. Any nonzero skew between these two signals can result in the possibility of misconfiguring the width of ROM Bank 0. This technical bulletin describes the cause and probability of such a misconfiguration and discusses possible solutions, including an example circuit.

Affected Parts

The information in this bulletin affects the following parts:

Device	Revision
Am29200™ microcontroller	A, B
Am29205™ microcontroller	A
Am29240™ microcontroller	A
Am29243™ microcontroller	A
Am29245™ microcontroller	A

The Problem

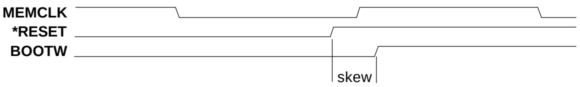
The state of BOOTW before and after the deassertion of $\overline{\text{RESET}}$ determines the width of ROM Bank 0.

State of BOOTW Signal		ROM Bank 0
Before Reset	After Reset	Width (Bits)
0	0	16
1	1	32
0	1	8
1	0	(reserved)

The only nonreserved selection where BOOTW has a different value before and after a reset is an 8-bit-wide boot bank. For this option, $\overline{\text{RESET}}$ and BOOTW are tied together. Any nonzero skew between these two signals can result in the possibility of misconfiguring the boot bank.

The processor samples $\overline{\text{RESET}}$ and BOOTW on approximately the rising edge of MEMCLK. It is difficult to be specific because during a processor reset, the clocks may not be established yet. In particular, MEMCLK is not guaranteed externally until after $\overline{\text{RESET}}$ is deasserted.

When the processor recognizes the deassertion of $\overline{\text{RESET}}$ (the first sample where $\overline{\text{RESET}}$ is High), it compares the new and previous samples of BOOTW. If a rising MEMCLK edge occurs during the skew region of the $\overline{\text{RESET}}$ and BOOTW signals, a misconfiguration may occur.



Again, it is difficult to know whether a misconfiguration will actually occur because of uncertainties in the signal-hardening logic and phase relationships of internal processor clocks. A misconfiguration is externally visible by an instruction fetch stream indicative of 16- or 32-bit-wide memory (0, 2, 4, ..., or 0, 4, 8, ...) instead of the intended 8-bit-wide bank (0, 1, 2, ...).

In a production environment, it is impossible to guarantee zero skew between the BOOTW and $\overline{\text{RESET}}$ pins of the processor, or account for varying processor internal delays due to process variations.

Assuming a random distribution of $\overline{\text{RESET}}$ deassertions relative to INCLK, the probability of misconfiguring ROM Bank 0 can be determined using the following equation:

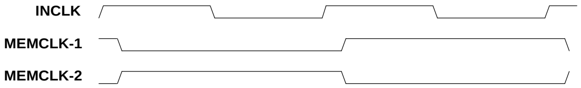
$$P_{\text{misconfigure}} = \frac{t_{\text{skew}}}{T_{\text{MEMCLK}}} = \frac{f_{\text{INCLK}} \bullet t_{\text{skew}}}{2}$$

A system with INCLK running at 32 megahertz and 3 nanoseconds of skew would misconfigure the width of ROM Bank 0 once in every 20 processor resets. Given the relative location of the $\overline{\text{RESET}}$ and BOOTW pins, and the chance of needing more than one layer to route the $\overline{\text{RESET}}$ signal, a skew of a few nanoseconds is likely.

The Solution

A misconfiguration of the boot-bank width occurs when the rising edge of MEMCLK takes place during the skew between $\overline{\text{RESET}}$ and BOOTW. Therefore, the solution is to synchronize $\overline{\text{RESET}}$ such that the rising edge of MEMCLK will never occur during the skew time.

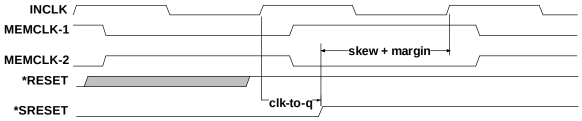
It would be desirable to use MEMCLK to achieve synchronization; however, MEMCLK is not guaranteed until after the processor resets, so INCLK must be used instead. Which edge of INCLK should be used? That depends on what the synchronization logic looks like. There are two INCLK cycles for every MEMCLK cycle, so a rising edge of INCLK could appear at either a rising or falling edge of MEMCLK; a falling edge of INCLK will appear in the middle of MEMCLK high- or low-time.



This circular dilemma can be solved by exploring both alternatives.

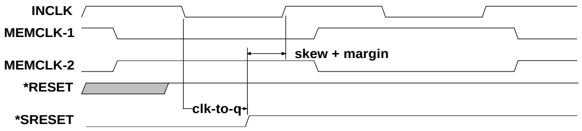
Option 1—The Rising Edge of INCLK

From the data sheet we know that MEMCLK rises or falls within 5 nanoseconds of the rising edge of INCLK. If our synchronization circuitry has a clock-to-q delay on the order of 7–8 nanoseconds, we will be past the critical region. There is nothing magic about the 7–8 nanoseconds; it is one quarter of the INCLK period, or 5 nanoseconds plus some margin. The amount of skew such a system could tolerate is approximately three quarters of an INCLK period minus some margin.



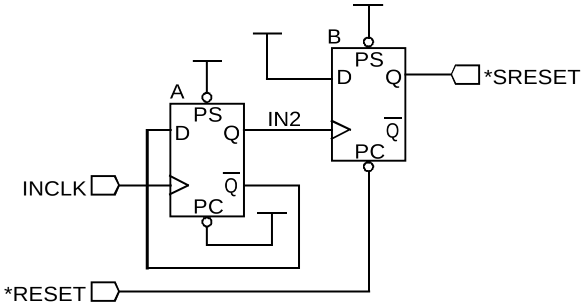
Option 2—The Falling Edge of INCLK

What about using the negative edge of INCLK? INCLK falls halfway through the MEMCLK high- and low-times. With this option, the clock-to-q delay is not an issue. However, the amount of skew the system can tolerate is limited by the next rising edge of INCLK.

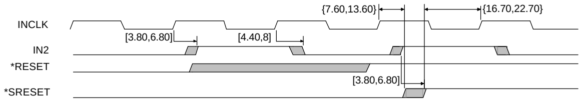


Which option is the best alternative? Option 1 has the advantage of more available time for skew and margin, but requires a minimum delay that can be difficult to guarantee. Option 2 has no such minimum-delay requirement, but will require faster logic to accommodate the skew and margin within the available time.

The implementation of the synchronization logic depends on the technologies used in the system (ASIC, PLD, TTL, etc.). To implement Option 1, a single 74F74 dual flip-flop can be used.



The circuit uses Flip-Flop A to delay the clock to Flip-Flop B. It divides INCLK, but that isn't important to the operation of the circuit. Depending on the circuitry used to generate a reset, a Schmidt-trigger gate may be necessary. Using the specified min-max propagation delays, the delay characteristics of the circuit can be calculated.



The sample circuit provides a guaranteed minimum propagation delay of 7.6 nanoseconds and can accommodate 16.7 nanoseconds of skew and margin.

If You Need Assistance

Product support for the 29K Family processors is available from our Embedded Processor Division (EPD) Technical Support Hotlines located in the U.S. and in the U.K.

Assistance is available in the U.S. from 9:00 A.M. to 6:00 P.M. central time, Monday through Friday (except major holidays). In Europe assistance is available during U.K. business hours. Contact us at one of the following numbers:

To reach the U.S. hotline

From	Call
U.S.	1-800-2929-AMD
U.K.	0-800-89-1455
Japan	0031-11-1163
Any other location	+1-512-462-4118†

†Toll applies.

To reach the U.K. hotline

From	Call
U.K.	(0)256-811101
France	0590-8621
Germany	0130-813875
Italy	1678-77224
Any other location	+44-(0)256-811101†

†Toll applies.