Texas Instruments

Application Report

SPNA080 – March 2005

Differences Between Start-Up Timing for the TMS470R1x Flash and ROM Controllers

Bernhard Rill

European Automotive Unit

ABSTRACT

This document describes the differences in the startup timing between flash and the ROM controller when using the same application software. The calculation of the difference in the startup timing behavior of a flash vs. a ROM controller is shown in an example. A method to minimize the startup behavior is described.

Contents

List of Figures

1 Introduction

This document explains the different startup behavior of flash and ROM controllers. It should be used as a guideline to optimize the startup behavior in applications and to calculate the exact start-up timing differences.

The start-up procedure of a TMS470 flash device differs slightly from the start-up procedure of a TMS470 ROM device. In some applications, it is important to know the absolute time a controller requires to begin program execution after a power-on reset (nPORRST).

2 Reasons for Start-Up Timing Differences

There are two reasons for the differences in start-up timing:

- Flash pump stabilization time
- Different default wait states settings for program memory access

2.1 Flash Pump Stabilization Time

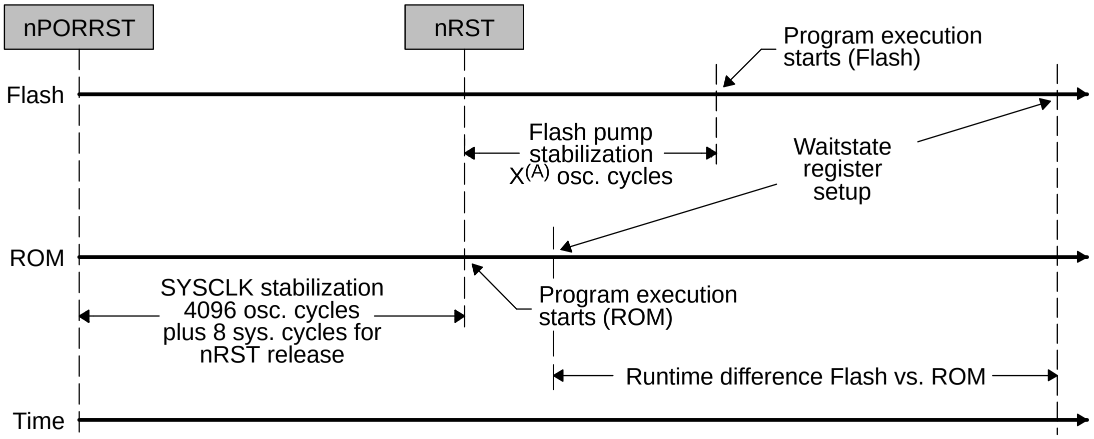
A counter in the flash wrapper counts down after nRST is released, allowing the flash pump to stabilize. This counter is clocked by the system clock (SYSCLK), which runs at a speed of OSCCLK/2, where OSCCLK is the oscillator clock. The ROM wrapper does not have this counter.

For detailed information on the exact number of cycles for a device with a ROM wrapper, please refer to the specific device data sheet.

2.2 Different Default Wait State Settings for Program Memory Access

Because of the default waitstates, the first instructions in a flash device are executed with 15 waitstates, whereas the first instructions on a ROM device are executed with only 1 waitstate. For more information, see the device-specific documentation, the *TMS470R1x C05 ROM Pipeline Wrapper Reference Guide* (SPNU009) and the *TMS470R1x F05 Flash Module Reference Guide* (SPNU213).

Figure 1 illustrates the startup sequence:


nPORRST
nRST
Program execution starts (Flash)
Flash
Flash pump stabilization X(A) osc. cycles
Waitstate register setup
ROM
SYSCLK stabilization 4096 osc. cycles plus 8 sys. cycles for nRST release
Program execution starts (ROM)
Runtime difference Flash vs. ROM
Time

(A) Number X of oscillator cycles is device specific.

Figure 1. Startup Timing Behavior

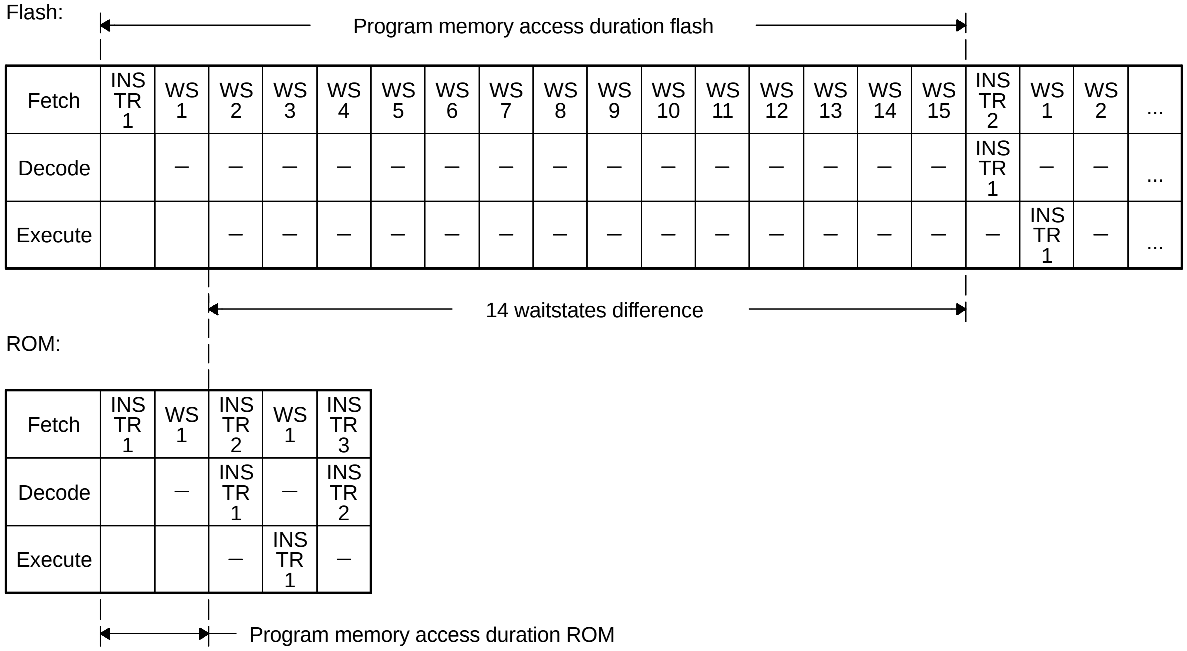
Note:
This drawing is not to scale.

The SYSCLK stabilization time is 4096 oscillator cycles for Flash and ROM devices. After the stabilization, nRST will be released:

- Program execution on ROM devices can start immediately.
- Flash devices need additional cycles for flash pump stabilization; therefore, the flash program execution starts x oscillator cycles later (x is device dependent).

Note:
For detailed information on the exact number of flash pump stabilization cycles, please refer to the specific device data sheet.

For example, Figure 2 illustrates the difference in the execution of one instruction. After reset for each program memory access, a default number of waitstates is executed: 15 for Flash and 1 for ROM. Therefore, for each program memory access the difference between Flash and ROM devices is 14 cycles, as shown in Figure 2.


Flash:
Program memory access duration flash
14 waitstates difference
ROM:
Program memory access duration ROM

Flash:

Fetch	INSTR 1	WS 1	WS 2	WS 3	WS 4	WS 5	WS 6	WS 7	WS 8	WS 9	WS 10	WS 11	WS 12	WS 13	WS 14	WS 15	INSTR 2	WS 1	WS 2	...
Decode		–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	INSTR 1	–	–	...
Execute			–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	INSTR 1	–	...

ROM:

Fetch	INSTR 1	WS 1	INSTR 2	WS 1	INSTR 3
Decode		–	INSTR 1	–	INSTR 2
Execute			–	INSTR 1	–

Figure 2. Comparison of the Default Flash versus ROM Waitstates

The runtime difference of 14 cycles is valid until the waitstate in the bank access control register is set up.

3 Minimizing the Differences in Start-Up Timing

It is not possible to have exactly the same timing between flash and ROM devices, but it is possible to minimize the difference. The flash pump stabilization time exists only on flash devices. Figure 2 shows the timing difference between running the same code on a flash device and a ROM device, in reference to the release of nRST. After setting up the waitstate register, the code execution will have the same timing behavior.

Therefore, to minimize the timing difference between flash and ROM devices, it is recommended to set up the waitstate registers to the same values as soon as possible in the startup code. The following section provides an example for implementing this recommendation.

3.1 Start-Up Code Example

For the following example of startup code, it is assumed that the only flash bank, Bank 0, is active and that VREAD is set to 5V by default. The main goal of this startup code is to set up the waitstate registers for program memory access to the same values as soon as possible.

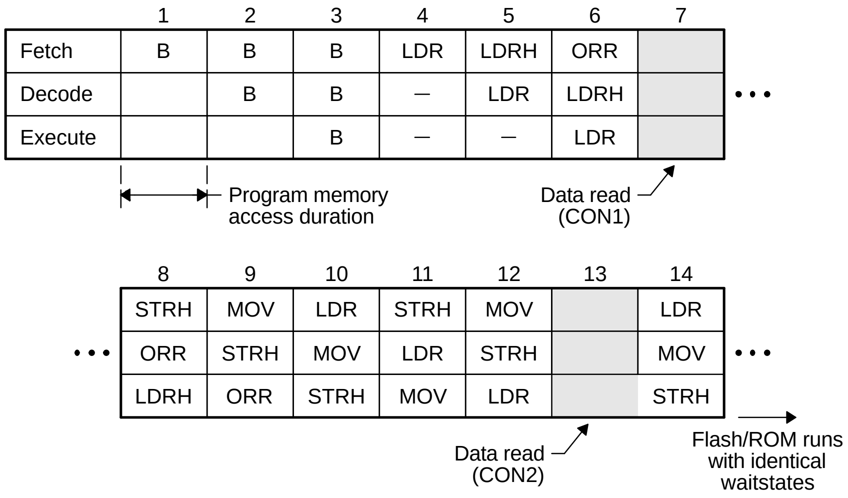
This assembler code example is generated by the absolute listing tool, which generates the absolute address and the opcode in addition to the assembly code. This information is necessary to identify the location of the data read accesses (e.g., flash access and register).

```
;Abs. Adresse, Opcode, Assemblercode, Comment

;          CONSTANT TABLE
000001a0                   .sect   ".text"
                           .align  4
000001a0   00000000  CON1:   .field          _e_SARSYS_ST,32
                                       ; _e_SARSYS_ST origin=0xFFFFFFD0
                           .align  4
000001a4   FFE88006  CON2:   .field          -1540090,32
                           .align  4
000001a8   FFE89C00  CON3:   .field          -1532928,32

;          INTVECS.ASM  RESET INTERRUPT
00000000   EAFFFFFE           b     _c_int00


                      _c_int00:
;          ENABLE FLASH/ROM WRAPPER ACCESS
00000018   E59FC180          LDR       V9, CON1
0000001c   E1DC00BE          LDRH      A1, [V9, #14]
                                       ;Load contence of GLBCTRL in CPU
                                       ;Register
00000020   E3800010          ORR       A1, A1, #16
00000024   E1CC00BE          STRH      A1, [V9, #14]
                                       ;Set GLBCTRL[4:0] to 1

;          SET FLASH/ROM WAITSTATES
00000028   E3A00C7F          MOV       A1, #0x7F00
                                       ;Mask for the Bank Access Control
                                       ;Register 2 (FMBAC2 (see Flash
                                       ;Wrapper Spec) or BAC2 (see ROM Wrapper Spec))

0000002c   E59F1170          LDR       A2, CON2
                                       ;Address of the Bank Access Control
                                       ;Register 2 (FMBAC2 or BAC2)
00000030   E1C100B0          STRH      A1, [A2, #0]
                                       ;Set waitstates to 0
; After the execution of the STRH instruction Flash and ROM run with
; an equal number of waitstates
;          DISABLE PIPELINE MODE
00000034   E3A01000          MOV       A2, #0
00000038   E59F0168          LDR       A1, CON3
0000003c   E5801000          STR       A2, [A1, #0]

;          DISABLE FLASH/ROM WRAPPER ACCESS
00000040   E1DC00BE          LDRH      A1, [V9, #14]
00000044   E20020EF          AND       A3, A1, #0x00EF
00000048   E2000CFF          AND       A1, A1, #0x0FF00
0000004c   E1820000          ORR       A1, A3, A1
00000050   E1CC00BE          STRH      A1, [V9, #14]

;          START WITH SYSTEM MODULE SETUP
...
```

Note:
The start up behavior is device dependent. For detailed information on the start-up behavior, please refer to the specific device data sheet.

4 Calculating the Differences in Timing for the Start-Up Code Example

This section explains how to calculate the differences in the start-up timing for the example code in the previous section. Only the accesses (instruction fetch and data read) to the program memory need to be considered for calculating the runtime difference. Figure 3 shows the pipeline of the startup code example.



	1	2	3	4	5	6	7
Fetch	B	B	B	LDR	LDRH	ORR	
Decode		B	B	–	LDR	LDRH	
Execute			B	–	–	LDR	

8	9	10	11	12	13	14
STRH	MOV	LDR	STRH	MOV		LDR
ORR	STRH	MOV	LDR	STRH		MOV
LDRH	ORR	STRH	MOV	LDR		STRH

• • •
Program memory access duration
Data read (CON1)
Data read (CON2)
Flash/ROM runs with identical waitstates

Figure 3. Pipeline for the Startup Code Example

4.1 Explanation of the Data Read

If data is loaded from the program memory, the following data read needs to be considered to calculate the runtime difference. For example, in Figure 3, see that No. 7, CON1, is loaded from the program memory and leads to an additional 14 waitstates difference.

On No. 8, the LDRH reads from the GLBCTRL register (no access to program memory). The following data read is equal for flash and ROM and therefore has no influence on the calculation of the runtime difference.

4.2 Assumptions for the Calculation

The following assumptions are necessary for the calculation:

- An oscillator frequency of 6MHz results in a SYSCLK frequency of *3MHz*, since SYSCLK = OSCCLK / 2 after reset.
- Table 2 shows the requirement for *14 accesses* to the program memory to set the waitstate register.
- There is a difference of *14 cycles* for every program memory read access.

4.3 Calculation

The total number of SYSCLK cycles from program start to the waitstate register set up is calculated as follows:

14 accesses * 14 cycles waitstate difference = 196 SYSCLK cycles

The runtime difference caused by default waitstate differences on flash and ROM devices is calculated as:

196 SYSCLK cycles / 3MHz = 65.3µs

The complete difference between flash and ROM devices is the result of the waitstates differences and the flash pump stabilization time.

This example is using a flash pump stabilization time of 336 oscillator cycles. The flash pump stabilization time is calculated as:

336 OSCCLK cycles / 6MHz = 56µs

The total timing difference is:

65.3µs + 56µs = 121.3µs

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments
Post Office Box 655303 Dallas, Texas 75265