
Using IGLOO and ProASIC3 FPGAs as a System Power Sequencer Design Example

Table of Contents

General Description	1
Design Description.	1
Implementation Details	3
Customizing the Design	3
Interface Description.	6
Utilization Details	6
Testing Scheme	6
Conclusion.	9

General Description

As system cost, power, and performance drive development of smaller semiconductor process geometries, the complexity of system-level power supply design increases greatly. Applications that incorporate the latest generations of components, which are powered by lower and lower supply voltages, must continue to interface with legacy components at legacy interface voltages. Not only are systems burdened with supplying many different power rails, but they must also follow the strict power sequencing requirements of all involved components. In many cases, power regulators do not offer enough flexibility to handle the unique requirements of each system, so power supply designs become complex. Microsemi® IGLOO® and ProASIC®3 FPGAs offer an alternative solution that provides system designers with a completely customizable, integrated solution to manage power supply sequencing. This solution is made possible by leveraging the embedded 1,024 bits FlashROM found on every re-programmable IGLOO and ProASIC3 FPGA.

This document describes an FPGA-based design that controls power sequencing for a system, or for any multiple-supply device. The FPGA stores power sequencing data in its internal FlashROM and generates enable signals upon power-up for the power regulators used on board, in the predefined time interval. Even Microsemi's smallest FPGAs can accommodate the extremely small logic. Since Microsemi's FPGAs are live at power-up (<http://www.microsemi.com/products/fpga-soc/technology-solutions>), power sequencing can begin as soon as the supplies are available; unlike SRAM-based FPGAs, there is no need to wait for configuration to complete.

Design files for this design example can be downloaded from the Microsemi website:
http://soc.microsemi.com/download/rsc/?f=System_Power_Sequencer_DF

Design Description

The top-level block diagram of the design is shown in [Figure 1 on page 2](#). Data patterns that include chip select timing information are stored in the FlashROM block inside the FPGA. The address generator

block generates the address for the FlashROM. The chip select generator block reads and decodes the data patterns from the FlashROM and enables the chip selects (CS) with the prescribed timing delay.

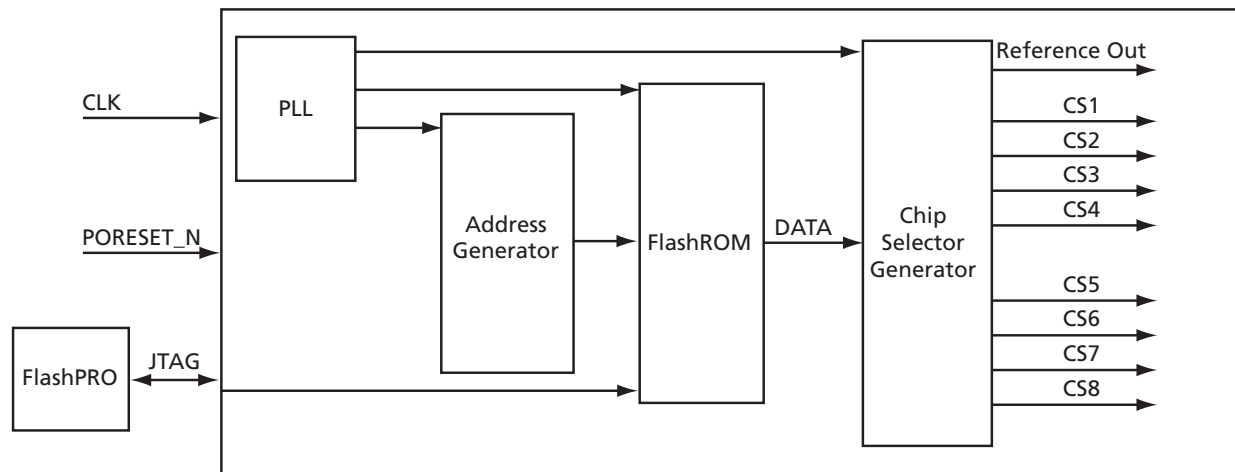


Figure 1 • Top-Level Block Diagram

The PLL block generates the basic frequencies required for the design. This design contains two clock frequencies that are used to control two timing domains. The first timing domain at 1 kHz is used to control chip selects in the millisecond timing range; the second, at 1 MHz, is used to control chip selects in the microsecond range. The time difference between the chip selects will be integer multiples of either 1 ms or 1 μ s. Refer to the examples described in "Customizing the Design" on page 3 for varying the time periods between the chip select outputs. The PLL is not a required element of the design, so this design can be modified to work on devices that do not contain a PLL. Simply build frequency dividers using logic tiles to create the timing derivatives.

The content of the FlashROM is used to define power sequencing for the system. This design contains eight chip select outputs: four with microsecond resolution and another four with millisecond resolution. In this design, the first four locations of the FlashROM are reserved for generating chip select signals in microseconds and the remaining four locations of the FlashROM are reserved for generating chip select signals in milliseconds. The REFERENCE_OUT signal acts as a reference for measuring the time of the chip select outputs. After power-up the core initializes and REFERENCE_OUT is immediately enabled.

All CS time delays make absolute reference to the REFERENCE_OUT signal rather than relative reference between the chip selects.

Assume the following CS values must be generated with respect to the REFERENCE_OUT signal: it is required to generate CS1 after 5 μ s, CS2 after 20 μ s, CS3 after 15 μ s, CS4 after 38 μ s, CS5 after 1 ms, CS6 after 6 ms, CS7 after 3 ms, and CS8 after 9 ms. The content of the FlashROM is shown in Table 1.

Table 1 • FlashROM Content and CS Generation

Address	Data (decimal)	CS Generation with Respect to REFERENCE_OUT
0x0	5	After 5 μ s CS1 goes High
0x1	20	After 20 μ s CS2 goes High
0x2	15	After 15 μ s CS3 goes High
0x3	38	After 38 μ s CS4 goes High
0x4	1	After 1 ms CS5 goes High
0x5	6	After 6 ms CS6 goes High
0x6	3	After 3 ms CS7 goes High
0x7	9	After 9 ms CS8 goes High

Implementation Details

This design contains mainly three components (RTL):

- The top-level, FlashROM control block: *power_sequencer.vhd*
- The PLL block: *pll_4_40.vhd*
- The FlashROM block: *FlashROM_cmp.vhd*

The top level integrates the lower blocks together and generates chip select signals. The PLL used within the FPGA generates 4 MHz and 40 MHz clocks from an external 48 MHz clock source. The output frequency of the PLL is further divided to generate clocks of 1 ms and 1 μ s. A clock of 20 MHz frequency is generated for the address generator block that generates addresses for reading the FlashROM contents, which are later stored in registers.

In the chip selector block, a 1 μ s clock is used to generate chip select signals CS1, CS2, CS3, and CS4. A 1 ms clock is used to generate chip select signals CS5, CS6, CS7, and CS8

Customizing the Design

This section explains how to fine tune the design parameters for the basic clock configuration and the FlashROM configuration.

Basic Clock Configuration

You can change the basic clock configuration if required.

The default basic clock frequency of the core is 1 MHz. With this frequency, the minimum time difference that can be achieved between two chip selects is 1 μ s. This output frequency of the PLL is further divided to generate a clock of 1 ms. To generate time differences between the CS, reconfigure the PLL block with different input and output frequencies.

1. Open the project files for this design example. Right-click the *pll_4_40* component on the left side window and select **Open Component**. This opens the Static PLL window (Figure 2).

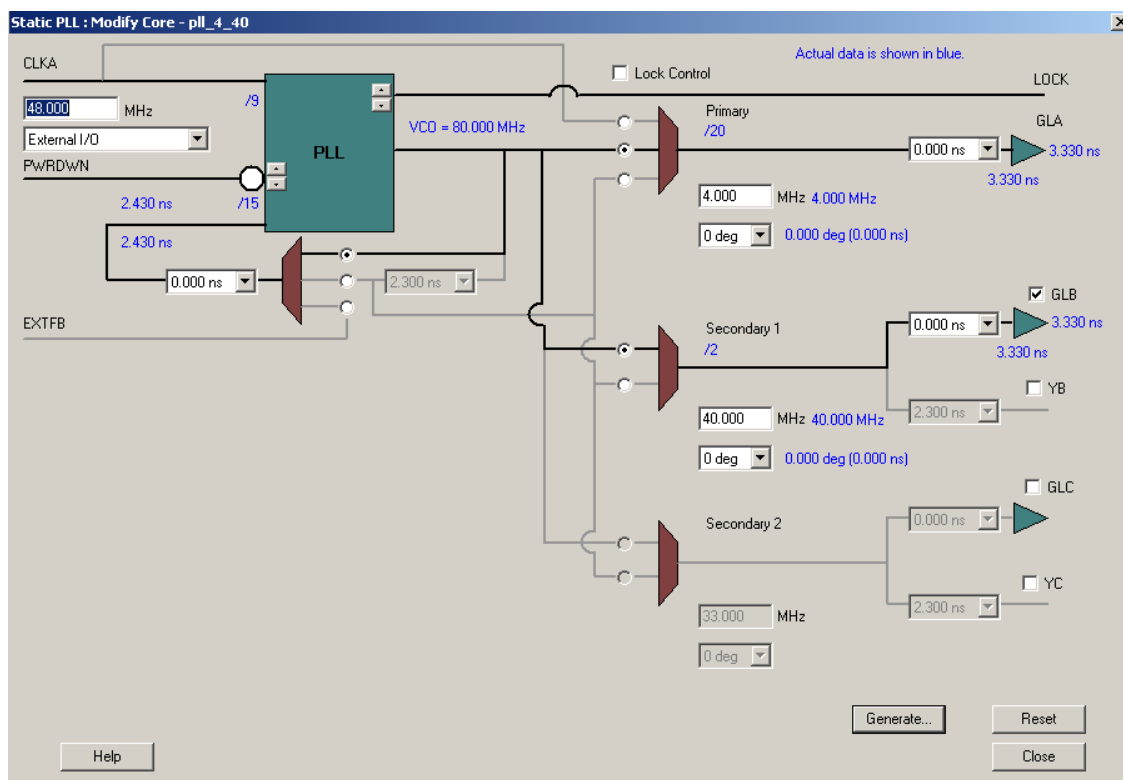


Figure 2 • PLL Configuration

2. Change the values as defined in your requirements.
3. Press the **Generate** button to update the PLL block with new values.
4. Press **Close** to close the Static PLL pop-up window.

For more details on configuring the PLL block, refer to the *SmartGen Cores Reference Guide* (www.microsemi.com/documents/gen_refguide_ug.pdf) from Microsemi.

FlashROM Configuration

You can change the FlashROM configuration if required.

This FlashROM is configured using Microsemi Libero® Integrated Design Environment (IDE).

1. Using Libero IDE, create a top-level module.
2. On the Cores tab, expand **Memory & Controllers**, right click **FlashROM**, and select **Configure core**. This opens the FlashROM: Create Core window.

The FlashROM region is divided into eight pages, each with 16 bytes of memory locations. Each memory location can be configured with different values, see [Figure 3](#).

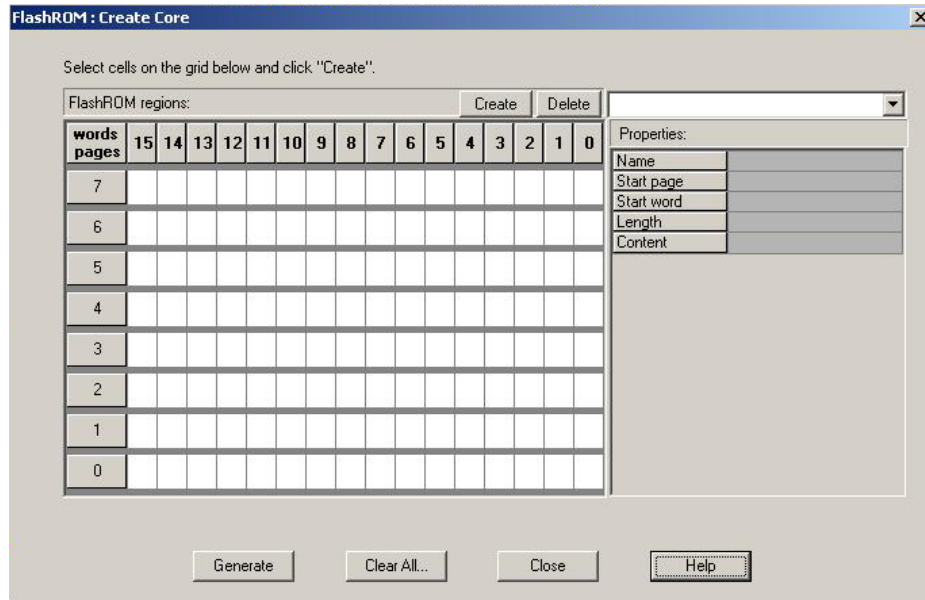


Figure 3 • Creating the FlashROM Core

3. Press **Generate** to generate a netlist (output format should match the HDL type you specified when you created your project). This opens the Generate Core dialog box (Figure 4). Specify a name and click **OK**.



Figure 4 • Generating a FlashROM Core

4. Close the FlashROM: Create Core window.

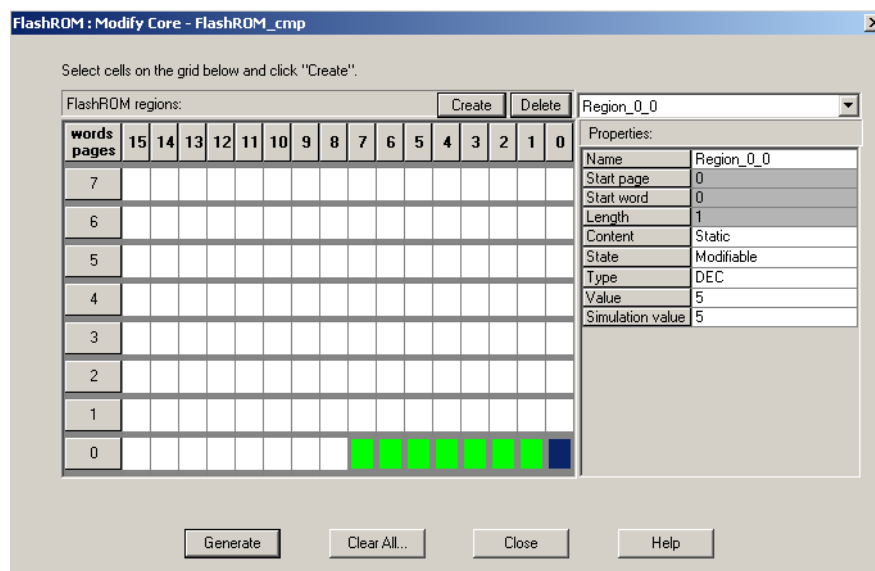


Figure 5 • Configuration Final Window

5. Press **Close**.
The next step is to instantiate the FlashROM component at the top-level module.

6. On the Hierarchy tab, right-click the <FlashROM_name>, and select **Open HDL File** to view HDL source code of the component.
7. Instantiate the FlashROM component in the top-level module. Assign a pin constraint, and run Compile and Layout.
8. Download the *.stp file into the FPGA using FlashPro.

Interface Description

Table 2 gives port descriptions.

Table 2 • Signal Descriptions

Port	Direction	Description
CLK	Input	48 MHz input reference clock to the FPGA
PORESET_N	Input	Active Low reset signal from SW1 push-button present on board.
REFERENCE_OUT	Output	Active High output reference signal
CS1, CS2, CS3, CS4,	Output	Active High output chip select signals with microsecond time interval
CS5, CS6, CS7, CS8	Output	Active High output chip select signals with millisecond time interval

Utilization Details

This design was verified in the M1 IGLOO M1AGL600V2-484FBGA device. Table 3 lists the utilization results for the targeted device.

Table 3 • Logic Utilization

Resource	Used	Total	Percentage (%)
Core	422	13,824	3.05%
I/Os	11	235	4.68%
Differential I/O	0	60	0.00%
Global (chip + quadrant)	5	18	27.78 %
PLL	1	1	100.00 %
RAM/FIFO	0	24	0.00 %
Low Static ICC	0	1	0.00 %
FlashROM	1	1	100.00 %
User JTAG	0	1	0.00 %

Testing Scheme

Simulation Flow – Best case and worst case timing simulation is completed for this design. Testbench and waveform files are included in the simulation project folder. To run the testbench, click the **Simulation** icon in the Libero IDE design flow window. This invokes ModelSim[®], where best case and worst case simulation results can be verified.

Figure 6 shows the FlashROM read cycles.

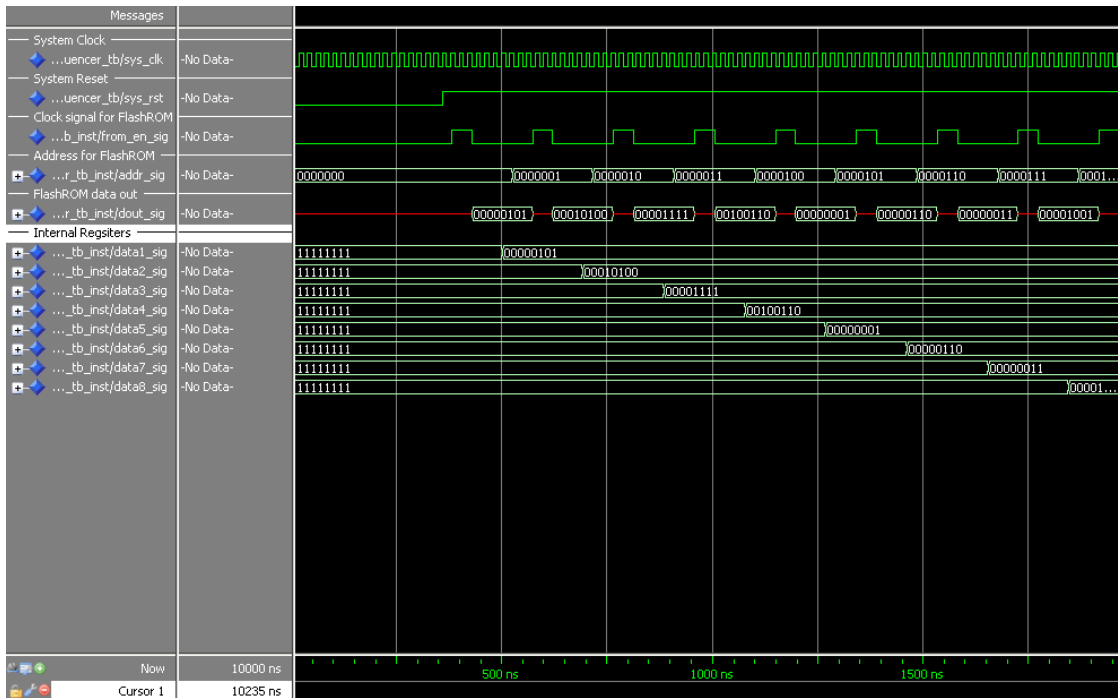


Figure 6 • FlashROM Read Cycle

Figure 7 shows the timing generation for CS1, CS2, CS3, and CS4.

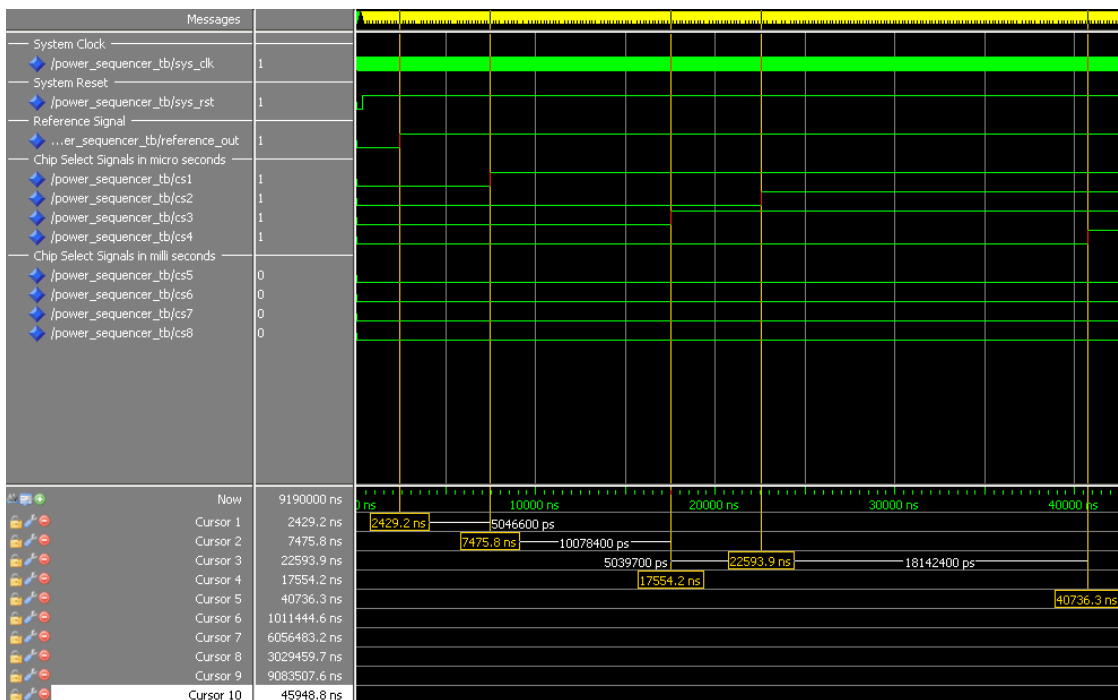
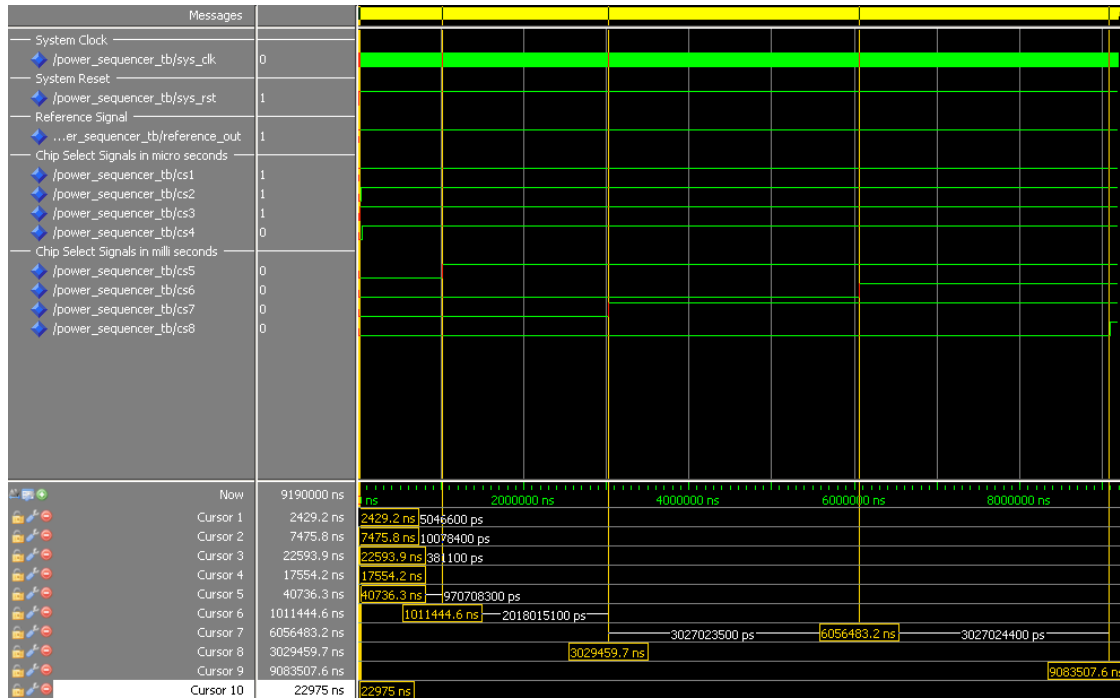


Figure 7 • CS1, CS2, CS3, and CS4 Generation (microseconds)

Figure 8 gives the timing generation for CS5, CS6, CS7, and CS8.


Figure 8 • CS5, CS6, CS7, and CS8 Generation (milliseconds)

Hardware Verification – This design is tested and verified on the ARM[®] Cortex[®]-M1-Enabled IGLOO FPGAs Development Kit. Output chip select signals are connected to the general purpose outputs

available with Bank0, which are connected to P1 connector on the board. The oscilloscope screen shot is shown in Figure 9.

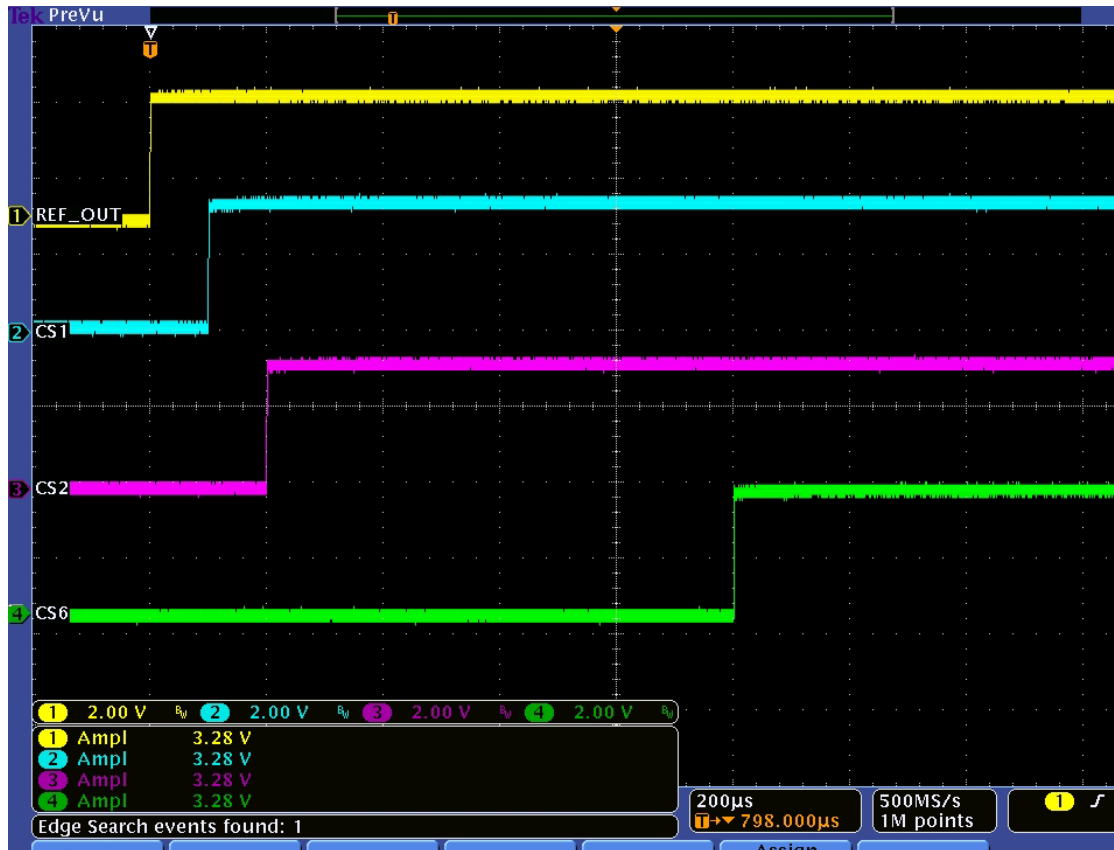


Figure 9 • Oscilloscope Waveform

Only REFERENCE_OUT (in yellow) and three chip select signals (CS1 in blue, CS2 in pink, and CS6 in green) are shown. CS1 is generated after 100 μs from the reference out. CS2 is generated after 200 μs, and CS6 is generated after 1 ms.

Conclusion

Microsemi’s embedded FlashROM enables a true single-chip, programmable, low-cost, and low-power solution for system-level power sequence control. This design can be easily modified to support various application requirements.

List of changes

Date	Changes	Page
Revision 2 (July 2015)	Non-technical Updates.	NA
Revision 1 (April 2009)	Initial Release.	NA

Note: *The revision number is located in the part number after the hyphen. The part number is displayed at the bottom of the last page of the document. The digits following the slash indicate the month and year of publication.



Microsemi Corporate Headquarters
One Enterprise, Aliso Viejo,
CA 92656 USA

Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996

E-mail: sales.support@microsemi.com

© 2015 Microsemi Corporation. All rights reserved. Microsemi and the Microsemi logo are trademarks of Microsemi Corporation. All other trademarks and service marks are the property of their respective owners.

Microsemi Corporation (MSCC) offers a comprehensive portfolio of semiconductor and system solutions for communications, defense & security, aerospace and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; security technologies and scalable anti-tamper products; Ethernet solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Microsemi is headquartered in Aliso Viejo, Calif., and has approximately 3,600 employees globally. Learn more at www.microsemi.com.

Microsemi makes no warranty, representation, or guarantee regarding the information contained herein or the suitability of its products and services for any particular purpose, nor does Microsemi assume any liability whatsoever arising out of the application or use of any product or circuit. The products sold hereunder and any other products sold by Microsemi have been subject to limited testing and should not be used in conjunction with mission-critical equipment or applications. Any performance specifications are believed to be reliable but are not verified, and Buyer must conduct and complete all performance and other testing of the products, alone and together with, or installed in, any end-products. Buyer shall not rely on any data and performance specifications or parameters provided by Microsemi. It is the Buyer's responsibility to independently determine suitability of any products and to test and verify the same. The information provided by Microsemi hereunder is provided "as is, where is" and with all faults, and the entire risk associated with such information is entirely with the Buyer. Microsemi does not grant, explicitly or implicitly, to any party any patent rights, licenses, or any other IP rights, whether with regard to such information itself or anything described by such information. Information provided in this document is proprietary to Microsemi, and Microsemi reserves the right to make any changes to the information in this document or to any products and services at any time without notice.