VA108x0 SPI memory boot options application note

Dec 21, 2016 Version 1.0

VA10800/VA10820

Abstract

The VA108x0 family of MCUs boot from an external SPI memory then execute code from internal SRAM. This application note describes the programming software and hardware connections on four SPI memories proven with the VA108x0. This provides a platform for developing other memory options in the future.

Table of contents

1 Boot sequence and options for the SPI memory

The VA108x0 family of microcontrollers boot from an external SPI memory device. The power-on-reset boot sequence is shown in Figure 1. No scale is provided due to variable elapsed time as determined by input clock frequency and EF_CONFIG register contents. The yellow balloons show the order of the steps.

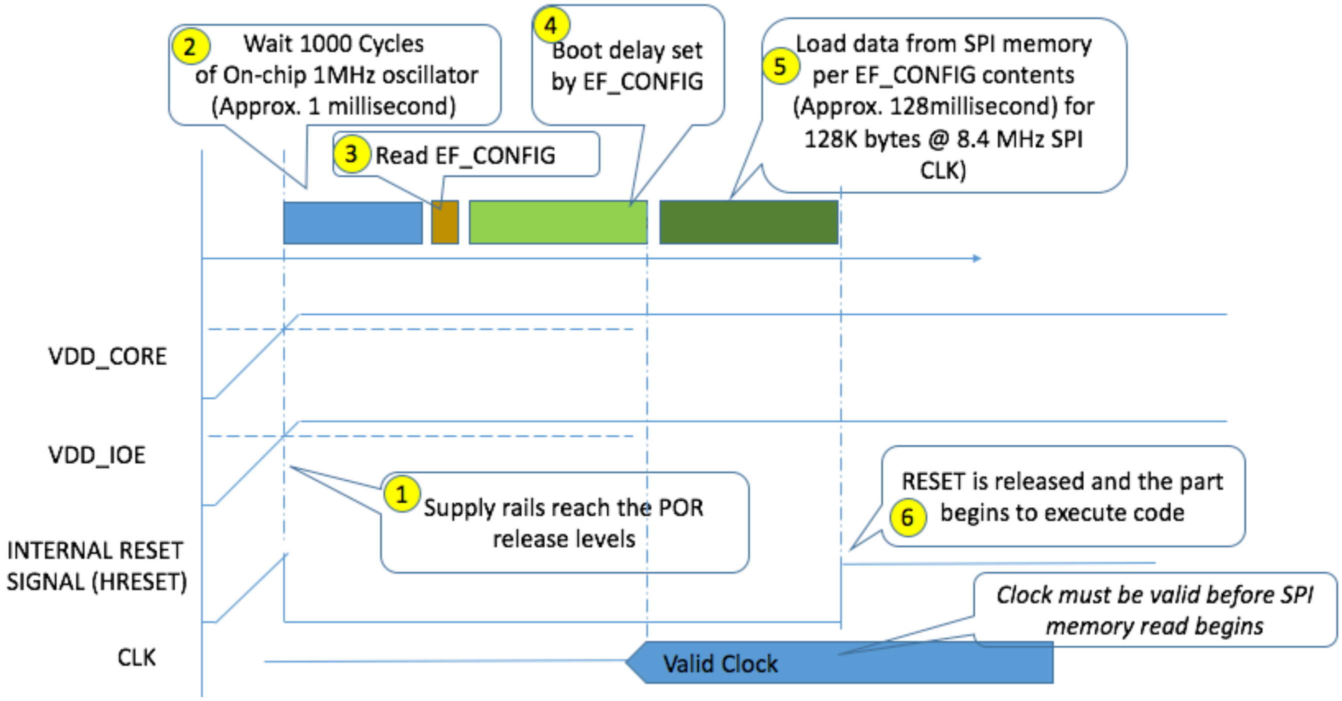
2 Wait 1000 Cycles of On-chip 1MHz oscillator (Approx. 1 millisecond)
4 Boot delay set by EF_CONFIG
5 Load data from SPI memory per EF_CONFIG contents (Approx. 128millisecond) for 128K bytes @ 8.4 MHz SPI CLK)
3 Read EF_CONFIG
VDD_CORE
VDD_IOE
1 Supply rails reach the POR release levels
RESET is released and the part 6 begins to execute code
INTERNAL RESET SIGNAL (HRESET)
Clock must be valid before SPI memory read begins
CLK
Valid Clock

Figure 1 – Timeline for POR sequence to boot MCU

For pin RESET and software RESET, the sequence begins at step 3, the read of the EF_CONFIG register.

Step 5 reads 128 bytes of data at a time. After each 128 bytes, a new READ command and address is supplied to the memory device.

1.1 Boot sequence options

Options for the boot sequence are set by EF_CONFIG field which is read from the on-chip eFuse memory prior to the SPI memory being loaded are shown in Table 1.

Parameter	Options	Default
Boot SPI clock rate	bus clock divided by 2, 6, 12 or 52	6
Boot SPI memory size	4k, 8k,16k, 32k, 64k, or 128 kbytes	128 kbytes
Redundant boot	Reads every 128 byte block twice and compares the results. Repeats until they match.	Enabled
Boot delay	0, 1, 3, 10, 30, 100, 300, 500 mSec	30 mSec
ROM read instruction code	8 bits available	0x03 (READ instruction for most serial memory devices)
ROM address mode	16, 24 or 32 bits, 128 kbyte memories require 24-bit addresses	24-bit

Table 1 - Boot option summary

Programming the eFuse to set these options is described in AN1204.

2 Programming algorithm support

ARM® / Keil® define a specific format for programming algorithms so they may be used with their MDK IDE. The Keil tool downloads the programming algorithm to RAM on the MCU via the JTAG port then sends various commands such as erase, program and verify to the device. VORAGO has created and validated algorithms for 4 devices. The "C" files for these devices is included in this application note and can be used as a starting point for other IDEs or production programming tools.

Some restrictions are placed on the code so that it can run from RAM regardless of the position it is placed. No interrupts are allowed and the code must be position independent.

For each memory device, software is provided in "flash_prg.c" that performs functions as summarized in Table 2.

Subroutine	Description
Init:	This enables MCU peripheral clocks and configures SPIC (also referenced as SPI->BANK[2]) for operation. If the device has a block protect feature, it is disabled. The device is set to ignore the WP# pin. For most devices this consists of: 1) Write Enable Latch (WEL) command and 2) a Write Status Register (WRSR) command to clear the block protect bits (BP1 and BP0)
ProgramPage:	The SPI is used to provide the memory device with a series of commands and a string of data to be programmed. Memory is programmed using a block transfer. Prior to each block of data being sent to the EEPROM, the following are sent to the memory: - Write Enable Latch (WEL) command - Write (WRITE) command - a 24-bit starting address.
EraseChip:	The SPI is used to provide the memory device with a series of commands. For flash devices, the chip erase (CE) command is used. For EEPROM, FRAM and MRAM, a value of 0x00 is written to each address. Most flash devices have an erased value of 0xFF.
Verify:	The SPI is used to read the memory device. The IDE provides the program data to be verified. If an error is detected, the failing address is returned to the IDE.
UnInit:	If the device has a block protect feature, it is enabled. The write protect pin sensitivity is enable. The SPIC block is disabled.
Note: The Sector Erase and the Block Erase commands are not supported.	

Table 2 - Summary of subroutines in "flash_prg.c"

Parameters specific for each memory chip are defined in a file called “flash_dev.c”.

The software is provided “AS IS” and may be modified by anyone. VORAGO is not liable for any damage that altered software may impose on a system.

Notes:

1. *The evaluation version of the Keil tool, MDK-Lite, will not build the FLM files. The licensed version of the IDE is required.*

2. *If the software is customized, limit the page size to 256 bytes to avoid RAM overflow problems.*

3 Hardware considerations

The SPI interface has been proven to be a very robust communication link and is used throughout the electronics industry for chip to chip communications. While the interface is straight forward, there are several items to consider for robust operation.

First, keep the trace length as short as possible and use a ground trace to flank the signals as shown in Figure 2. This keeps the loop area very short and will isolate the high speed digital signals from the rest of the board.

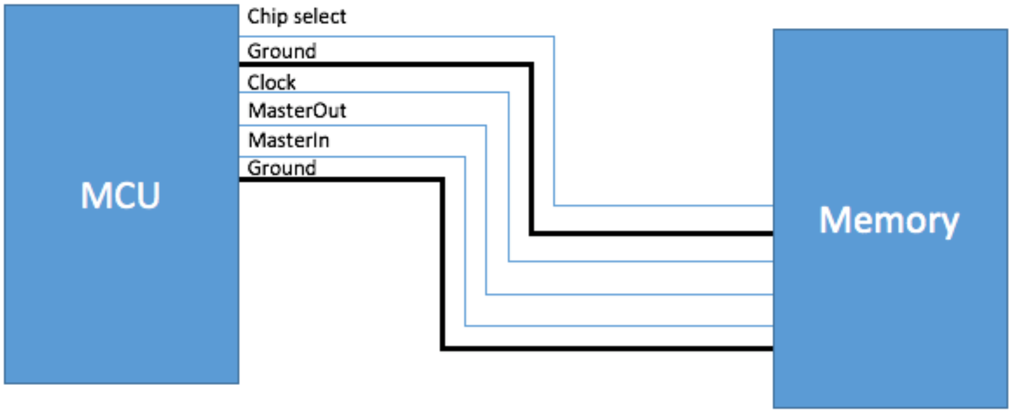

Figure 2 - Recommended trace routing from MCU to Memory device

Second, protect the device during power-up and power-down by using a pull-up resistor on chip select lines. While the processor is in reset, GPIO will be high impedance inputs and may float either high or low. The resistors will ensure no errant writes are made to the memory.

Section 4 has device specific information. A hardware section is provided for each that shows the signal interconnect between the MCU and the Memory.

4 Memory devices already validated

4.1 STM EEPROM

The M95M01 device is available in several temperature ranges with identical hardware connections and serial protocols. The high temperature version is the M95M01-A145. See st.com for the latest datasheet.

This device is used on the VORAGO evaluation board.

4.1.1 Hardware connections

Chip to Chip connections for M95M01

Signal function	Master VA108xx pin name	Slave M95M01 pin name	Recommended board level connection
SPI-MOSI	ROM_SO	D	
SPI-MISO	ROM_SI	Q	
SPI-SCK	ROM_SCK	C	
SPI-SS or Chip Select	ROM_CSn	S#	pull-up
Write protect	Port pin	W#	pull-down
Memory HOLD - Suspends serial operation	No connect	HOLD#	pull-up

Note: Signals with # are active low.

4.1.2 Programming software

The project for this device can be found in AN1205_SW.7z file in the st_eeprom folder.

4.2 TT Semiconductor High Temperature Flash

TT Semiconductor offers a high temperature SPI flash device (TTZ2564) rated for 200° C. Contact TT Semiconductor (http://www.ttsemiconductor.com/home.asp?uri=1000) for more information on this device.

4.2.1 Hardware connections

Chip to Chip connections for TTZ2564

Signal function	Master VA108xx pin name	Slave TTZ2564 pin name	Recommended board level connection
SPI-MOSI	ROM_SO	SI	
SPI-MISO	ROM_SI	SO	
SPI-SCK	ROM_SCK	SCK	
SPI-SS or Chip Select	ROM_CSn	CS#	pull-up
Memory Write protect	Port pin	WP#	pull-down
Memory HOLD - Suspends serial operation	No connect	HOLD#	pull-up

Note: Signals with # are active low.

Note: This device has a very high speed clock input buffer and is prone to double clocking if the slew rate on the clock is not very fast. It is advised to adhere to the chip specification and keep the rise and fall time of the clock below 5 nsec.

4.2.2 Programming software

The erase chip function sends two commands to the SPI flash:

- Write enable latch (WEL)
- Chip Erase (CE)

The Write-In-Progress (WIP) bit in the status register is polled until the memory device indicates a successful erase has completed. Before programming the flash, it must be erased.

The project for this device can be found in AN1205_SW.7z file in the tt_flash folder.

4.3 Everspin MRAM

MRAM has storage characteristics that makes data retention immune to many forms of radiation. The data sheet for this device can be found at: https://www.everspin.com.

4.3.1 Hardware connections

Chip to Chip connections for MR25H40VDF

Signal function	Master VA108xx pin name	Slave MR25H40VDF pin name	Recommended board level connection
SPI-MOSI	ROM_SO	SI	
SPI-MISO	ROM_SI	SO	
SPI-SCK	ROM_SCK	SCK	
SPI-SS or Chip Select	ROM_CSn	CS#	pull-up
Memory HOLD - Suspends serial operation	No connect	HOLD#	pull-up
Write Protect	Optional Port Pin	WP#	pull-up if no MCU connection, pull-down if tied to MCU port pin

Note: Signals with # are active low.

4.3.2 Programming software

The project for this device can be found in AN1205_SW.7z file in the everspin_mram folder.

4.4 Cypress FRAM

Cypress will be offering a 2Mb radiation hardened FRAM device (CYRS15x102) in 2017. The hardware connection and SPI protocol will be like the CY14E101 which has been proven out with the VA10820 device.

4.4.1 Hardware connections

Chip to Chip connections for CY14C101Q

Function	Master VA108xx pin name	Slave CY14C101Q pin name	Recommended board level connection
SPI-MOSI	ROM_SO	SI	
SPI-MISO	ROM_SI	SO	
SPI-SCK	ROM_SCK	SCK	
SPI-SS or Chip Select	ROM_CSn	CS#	pull-up
Memory Write protect	Port pin	WP#	pull-down
Memory HOLD - Suspends serial operation	No connect	HOLD#	pull-up
Hardware Store Busy	No connect	HSB#	
AutoStore Cap	No connect	VCAP	capacitor or Open circuit

Note: Signals with # are active low.

4.4.2 Programming software

Even though FRAM does not require an erase operation before programming, it is sometimes required to set all locations of a memory device to a known condition. This can allow checksums to be used over the entire memory space.

The project for this device can be found in AN1205_SW.7z file in the cypress_fram folder.

5 Conclusions

Four suitable companion SPI memory devices have been proven to operate with the VA108xx family of devices. This application note along with the associated software provides all the information necessary to program, erase and verify those four memory devices.

6 Other Resources

VORAGO VA108x0 programmers guide:
http://www.voragotech.com/sites/default/files/VA10800_VA10820_PG_July2016revision1.16%5B4%5D.pdf

VORAGO MCU products: http://www.voragotech.com/VORAGO-products

VORAGO Application notes: http://www.voragotech.com/resources

VORAGO VA108xx REB1board user guide: Part of Board support package
http://www.voragotech.com/products/reb1

ARM support answering question on how to modify a flash algorithm:
http://www.keil.com/support/docs/3656.htm