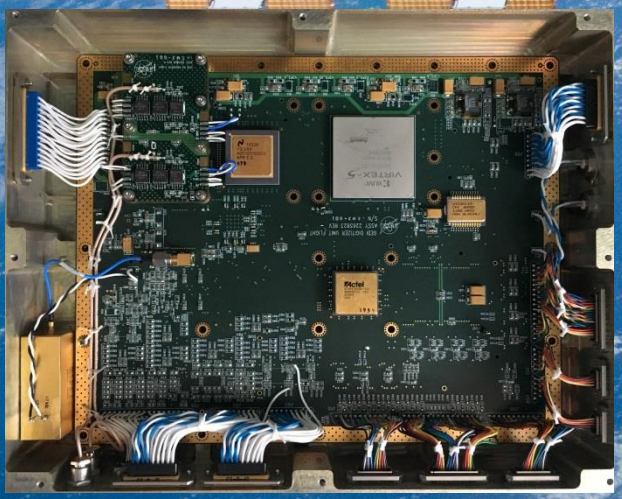
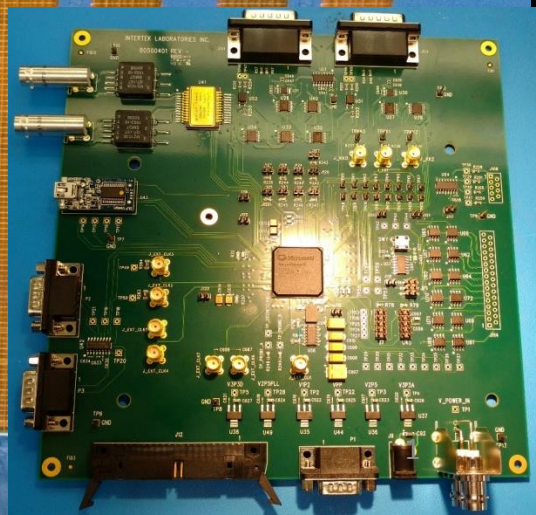
“Digital Device Architecture and the Safe Use of Flash Devices in Munitions”





60th Annual Fuze Conference, May 9-11, 2017, Cincinnati, Ohio.

“Digital Device Architecture and the Safe Use of Flash Devices in Munitions”

Rich Katz (NASA)
David Flowers (DMEA)
Keith Bergevin (DMEA)

Contact Information:
richard.b.katz@nasa.gov
david.flowers@dmea.osd.mil
keith.bergevin@dmea.osd.mil

Abstract

Flash technology is being utilized in fuzed munition applications and, based on the development of digital logic devices in the commercial world, usage of flash technology will increase. Digital devices of interest to designers include flash-based microcontrollers and field programmable gate arrays (FPGAs).

Almost a decade ago, a study was undertaken to determine if flash-based microcontrollers could be safely used in fuzes and, if so, how should such devices be applied. The results were documented in the "Technical Manual for the Use of Logic Devices in Safety Features."

This paper will first review the Technical Manual and discuss the rationale behind the suggested architectures for microcontrollers and a brief review of the concern about data retention in flash cells. An architectural feature in the microcontroller under study will be discussed and its use will show how to screen for weak or failed cells during manufacture, storage, or immediately prior to use. As was done for microcontrollers a decade ago, architectures for a flash-based FPGA will be discussed, showing how it can be safely used in fuzes. Additionally, architectures for using non-volatile (including flash-based) storage will be discussed for SRAM-based FPGAs.

Outline

- Fuze engineering working group discussions
  - Technologies for logic devices (antifuse, EEPROM, Flash, etc.)
- Example (non-DoD): EEPROMs and Single Board Computers
- Technical Manual criteria for use of Flash/EEPROM
- µController architectures
- FPGA Architecture and Capabilities for flash cell validation
- Performance characterization

Logic Device Technology

shall use the least complex logic device that can practically perform the required functionality.

2.2. While fixed-in-structure devices are acceptable and preferred, to avoid degradation of a safety feature, any logic device used in the implementation of that feature:

2.2.a. Shall not be re-programmable.

2.2.b. Shall not be alterable by credible environments..

2.2.c. Shall not have the SF logic configuration reside on volatile memory.

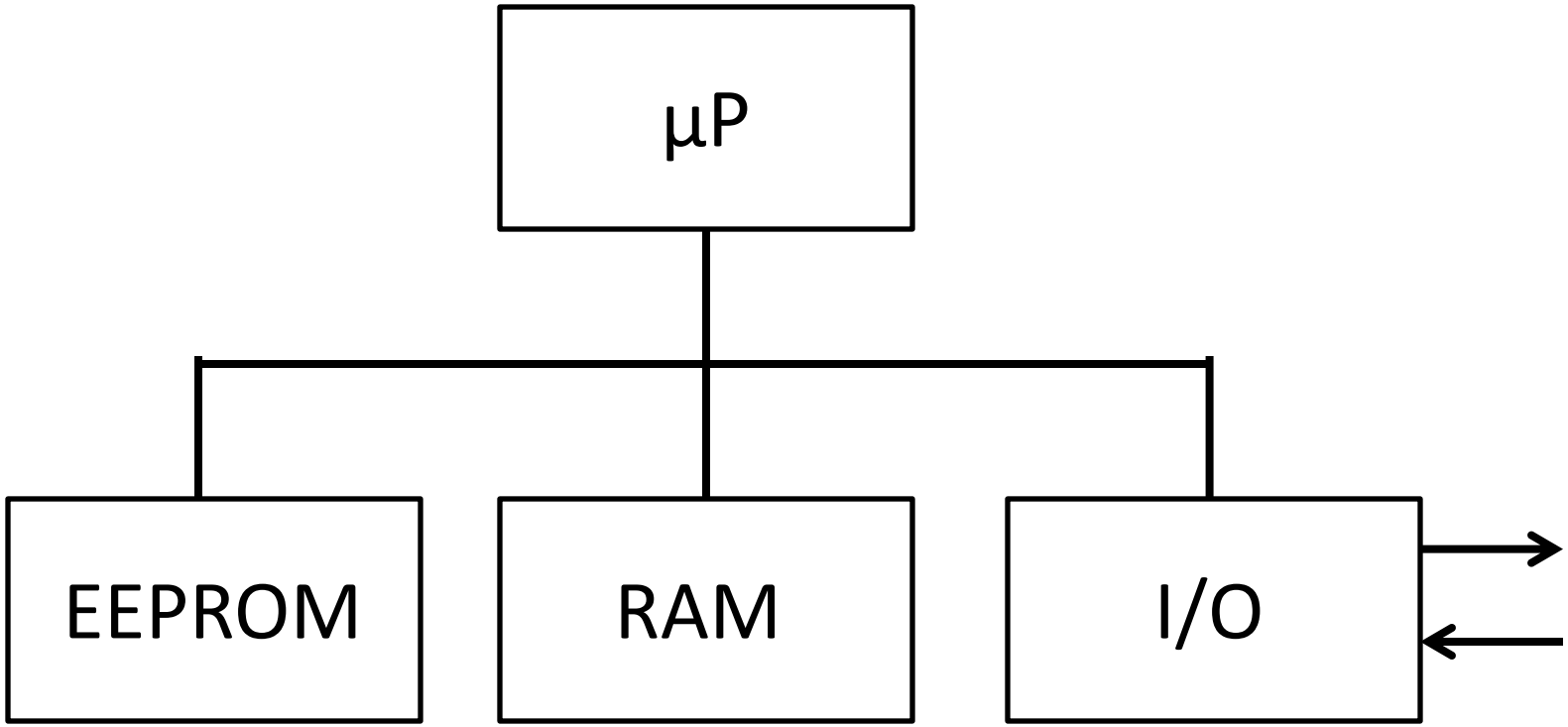
SF: Safety Feature

2.2.d. Should be rated to meet or exceed the lifecycle environments of the system. Shall have engineering rationale provided and associated risk(s) for logic devices not rated to meet or exceed the lifecycle of the system.

2.3. In order to minimize the potential for common cause failures, where all SFs are implemented with logic devices, at least two SFs shall be implemented with dissimilar logic devices. The degree of dissimilarity shall be sufficient to ensure that any credible common cause susceptibility will not result in unsafe operation

From DoD Fuze Engineering Standardization Working Group, "Technical Manual for the Use of Logic Devices in Safety Features," March 8, 2011.

A Typical µP-Based Computer



Not a recommended architecture.

Goddard Space Flight Center

GSFC NASA ADVISORY

1. Advisory Number	2. Subject
NA-GSFC-2005-04	Application of Hitachi 1-Mbit Die Based EEPROM Technology to Space Applications.

3. Manufacturer	4. Manufacturer CAGE Code	5. Federal Stock Code
Several manufacturers package this part and sell it under their brand name and part number.	N/A	N/A

6. Part/Material/Process Number	7. Lot Date Code/Serial Number	8. Controlling Spec/Document Number
Various	All	5962-38267 Various; Used on RAD 6000 & RAD 750.

9. References
See page 3.

GENERAL INFORMATION: This is a NASA Advisory issued by the Goddard Space Flight Center (GSFC) in accordance with the requirements of NASA Procedures and Requirements 8735.1A. For information concerning processing and actions required to be conducted in conjunction with this information, refer to your contract, or to NASA Procedures and Requirements 8735.1A. This information has been compiled and presented as accurately, completely, and objectively as possible consistent with the primary objective of alerting potentially affected projects as early as possible. This information may be altered, revised, or rescinded by subsequent developments or additional tests; and these changes could be communicated by other NASA documents. Neither NASA, the United States Government, nor any person acting on its behalf assumes any liability resulting from any distribution or use of this information.

10. DISTRIBUTION: While the primary distribution for this document is internal to NASA personnel and NASA contractor personnel, the information contained in this document is factual, and its release to, and use by, other entities is not restricted.

11. Problem Description and Details:

Reports of failures associated with the use of Electrically Erasable Programmable Read Only Memory (EEPROM) technology-based devices used by the aerospace industry have recently become widespread. Documented failures center around systems based on the Hitachi HCN58C1001 1-Mbit commercial die. The Hitachi die is packaged by various vendors into either single chip packages or multi-chip modules. This advisory is applicable for the use of Hitachi 1-Mbit die based components in both custom in-house designs as well as integrated into commercially available products as is the case with flight computer boards available from several vendors. Note: Hitachi no longer makes this die and the organization which previously made the die is now part of a different company.

Available evidence does not show conclusively whether this situation is a result of an inherent defect in the semiconductor die or may result from an anomaly during programming.

(See continuation on page 2)

12. Actions Recommended:

12. Actions Recommended: *(Continued from page 2)*

The implementation of redundancy can reduce the risk of mission failure due to a single point failure. For this approach to be effective, the determination of which side to power up can not be made by software executing from volatile memory. A hardware decoded command is the recommended approach.

The design can incorporate the capability to provide Direct Memory Access (DMA) to volatile memory independent of local controller or processor involvement. This allows for the EEPROM, or other type of memory, to be loaded directly and verified by an external source such as the S/C, off-board PROM, or command interface. In this way, failed portions of the memory space can be avoided by relocating the executable code as needed.

A less desirable architecture consists of multiple copies of the flight code to be stored in multiple EEPROM modules. A hardware decoded command determines from which EEPROM module the boot code will execute.

An even less desirable implementation consists of multiple copies of the flight code to be stored in multiple EEPROM modules, but with the decision from which EEPROM module the boot code will execute being made by a small block of decider code in the startup EEPROM. In this case the decider code is required to be already executing properly in order to verify and select the boot code EEPROM. This implementation is dependent on the decider code portion of the flight software not being corrupted. A failure in that region of the startup EEPROM will result in a fatal failure.

The least desirable and highest risk approach consists of a single EEPROM containing the boot code and a single copy of the executable code. A slight improvement would be to include multiple copies of the executable code to mitigate the risk of failure in certain parts of the EEPROM.

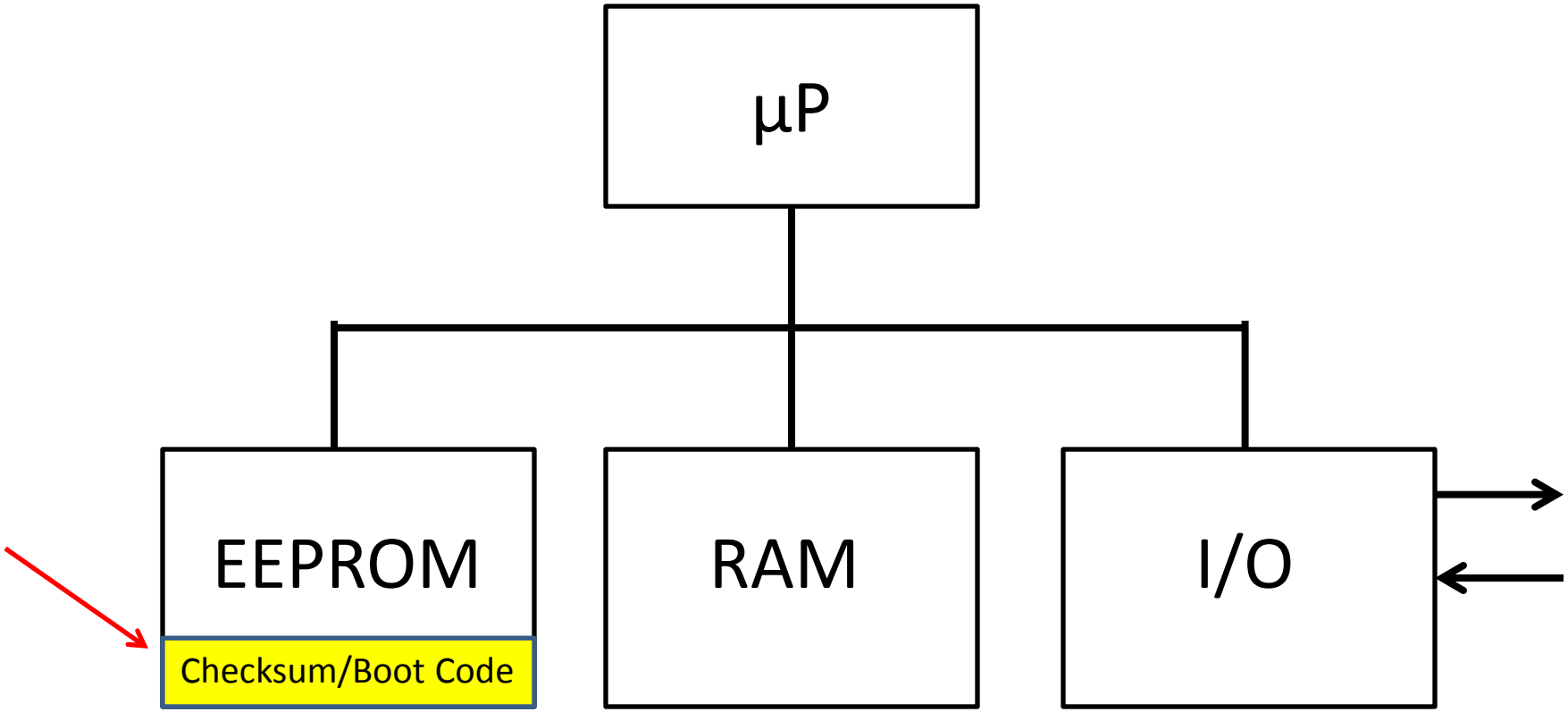
The use of Error Detection and Correction (EDAC) is recommended, but can not be relied upon totally to compensate for single “weak bits”. The reason being that in the cases of a slow acting or an oscillating bit failure, the output of the EDAC circuitry could be indeterminant due to the voltage transitions applied to its inputs while it performs its computations and the propagation delays inherent within the EDAC circuitry.

The memory access time of the EEPROM should be extended as much as feasible to provide added time for slow acting “weak bits” to reach the proper voltage level.

In addition to system architecture, there are mitigation strategies that can be applied at the component level. These are discussed next.

At the most basic level, designers must insure that all manufacturer component specifications and recommendations are properly

A Typical µP-Based Computer



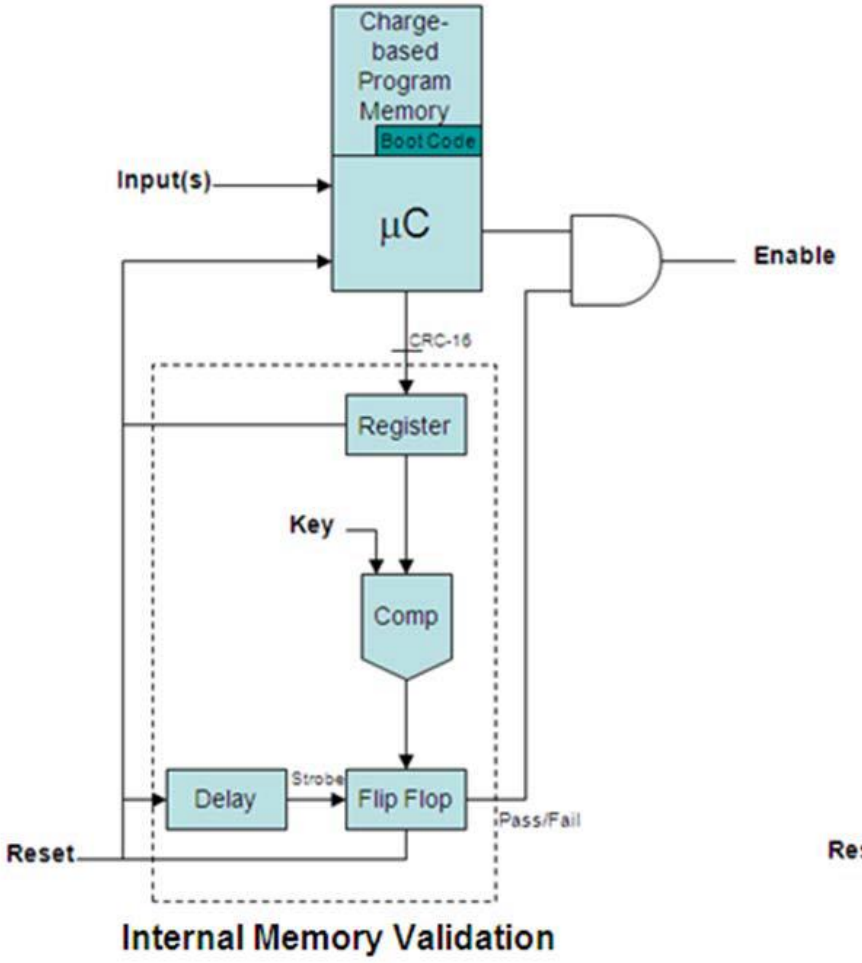
Not a recommended architecture.

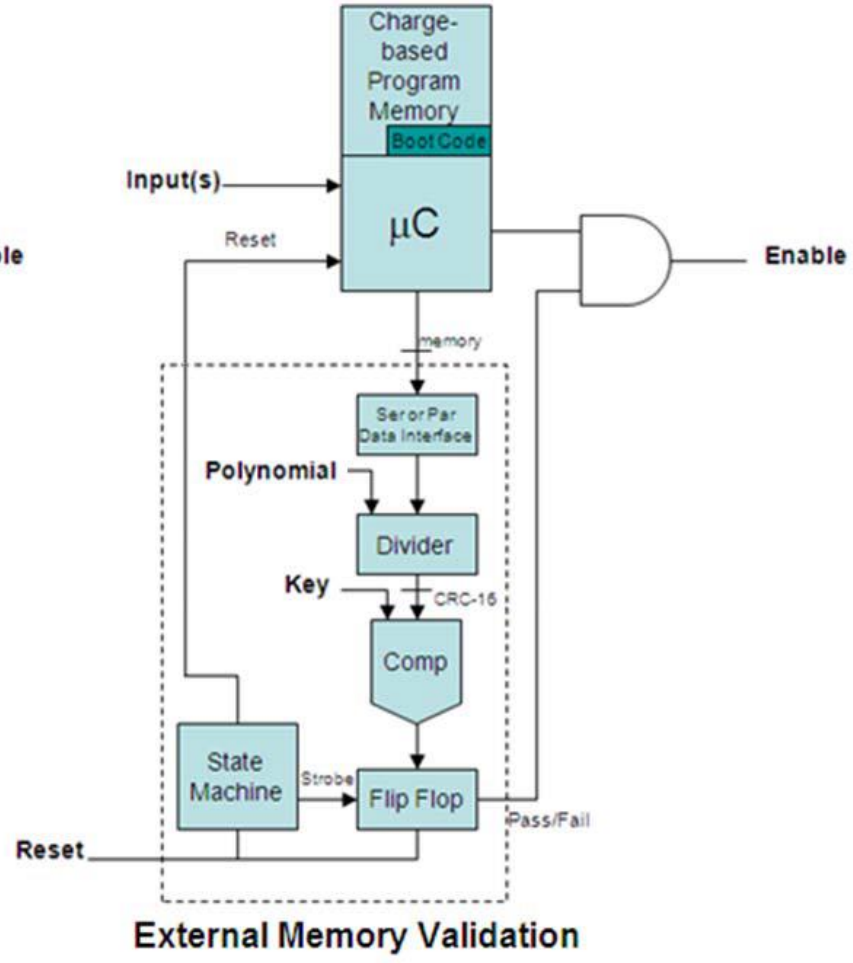
Memory Validation Required

A.2. Since, any changes to the SF hardware can adversely compromise the safety of the design, it is critical to establish and maintain a stable configuration throughout the lifecycle of the SF. A programmable logic device, including associated memory devices, may be considered as satisfying the requirements of paragraph 2.2 if, once programmed or configured during manufacturing, the configuration of its internal logic cannot be changed. Additionally, for devices relying on charged-based memory to implement a SF, a method of validating the integrity of the memory shall be performed prior to executing the safety function. The memory must be validated with the rigor equivalent to, or better than, that of a 16 bit Cyclic Redundant Check (CRC16). This computed result shall be externally compared against a known value that is stored externally. The external device(s) shall (1) be fixed-in-structure, (2) be dedicated circuitry, (3) not contain and be exclusive from any other functions, and, (4) when feasible, not be implemented as part of any other SF. Consult with the appropriate Service Safety Authority for guidance. A notional example of an internal and external memory validation is provided below:

From DoD Fuze Engineering Standardization Working Group, "Technical Manual for the Use of Logic Devices in Safety Features," March 8, 2011.

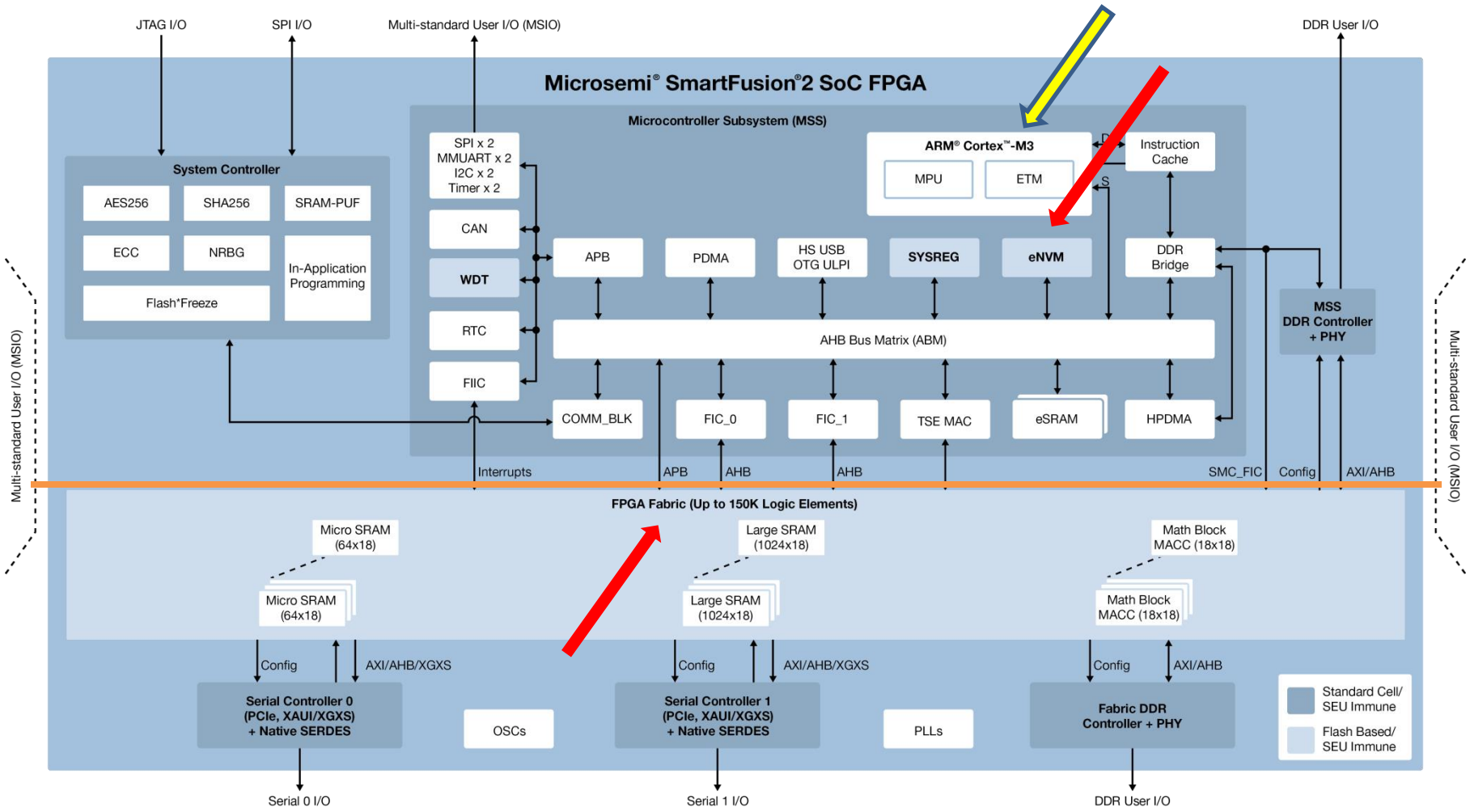
Memory Validation:µC Example





From DoD Fuze Engineering Standardization Working Group, "Technical Manual for the Use of Logic Devices in Safety Features," March 8, 2011.

SmartFusion2 FPGA Architecture



From Microsemi documentation.

Smartfusion2 Fabric Configuration and eNVM (non-volatile memory) Integrity Tests

SmartFusion2 and IGLOO2 FPGA devices have a novel NVM integrity check mechanism that can optionally be used to check the reliability and security of a device automatically upon power-up, or upon demand. The contents of all the nonvolatile configuration memory segments, including security keys, security settings, and the FPGA fabric configuration, plus any eNVM pages declared as ROM by the user (all the write-protected pages) are digested (hashed) using the SHA-256 algorithm. The results are compared to values stored in dedicated nonvolatile memory words located in each segment. If the contents are unchanged from when the digests were computed and stored during the original programming steps—that is, if the current and stored digests match—the test will pass; otherwise a failure is flagged. This test provides assurance against both natural and maliciously induced failures.

Note: The FPGA fabric is non-operational while the test is running on the fabric.

Ensure not to run the digest too often, as eventually the 20 year reliability of the flash cells will be affected. . Note that the digest verification of the fabric and each eNVM array can each be blocked with lock-bits. However, these can be overridden with a match of the FlashLock Passcode.

The digests can be verified on-demand by the user, either internally using a system service, or externally using a programming instruction. In addition, the user can run digests check automatically upon each power-up. The following section describes the various options to run digest check.

From Microsemi documentation.

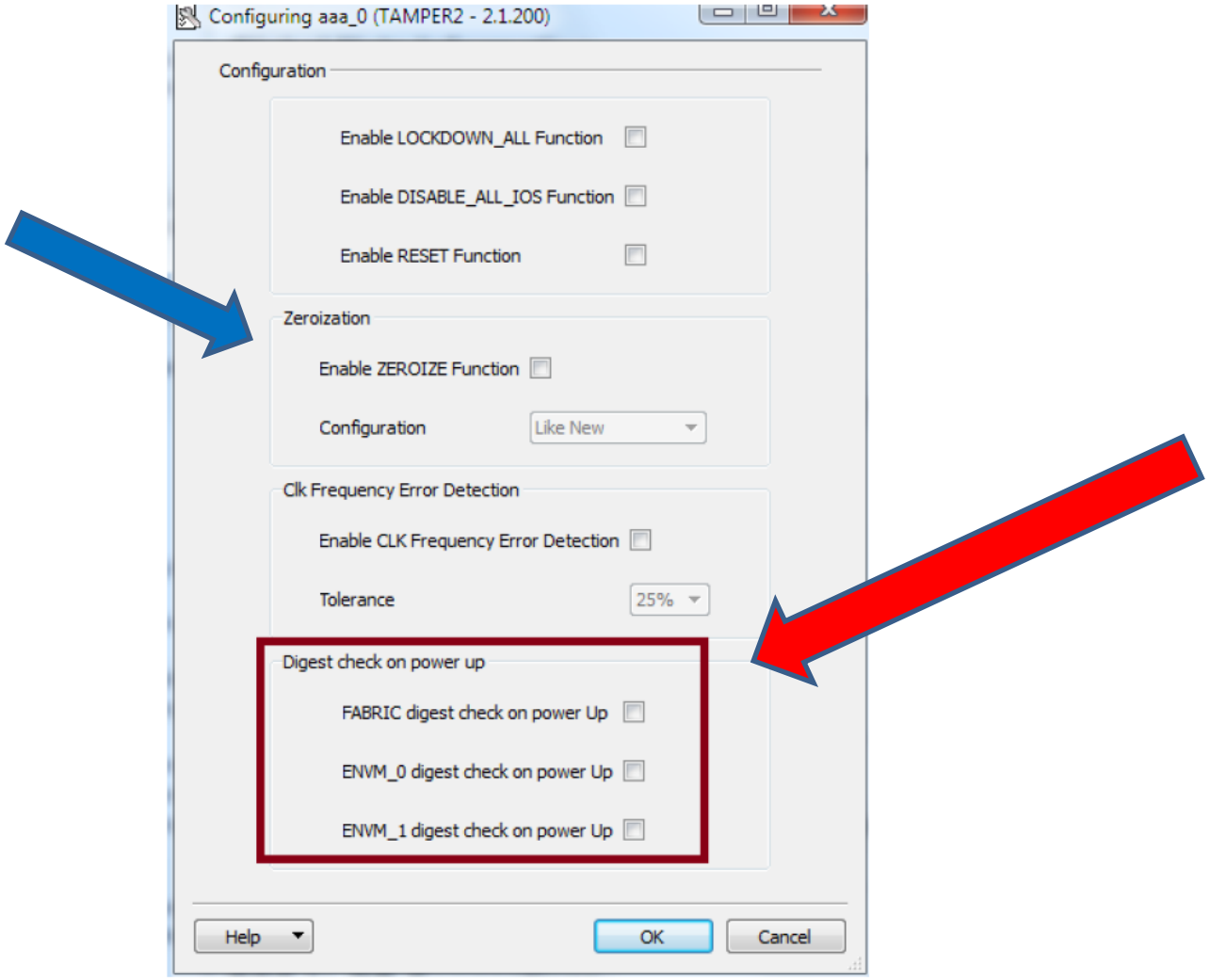
Smartfusion2 Integrity Check (Power-up Digest Check)

The Digest checks for the FPGA fabric and one or both eNVM arrays (if present) can be configured using lock-bits to run automatically upon each power-up. The Tamper macro in Libero SOC allows the user to set this option (refer to Figure 5-5) and the POWERUP_DIGEST_ERROR signal will allow the user to check the status of digest check..

If the FPGA is selected for power-on digest check, there will be a noticeable delay due to the time required to run the hash algorithm on millions of bits (for example, one or two seconds) before it boots into normal operation. Also, the user must be careful and not run the digest often.

From Microsemi documentation.

Smartfusion2 Integrity Check (Power-up Digest Check)



System Controller Checks Data Integrity

NVM Data Integrity Check Service

The NVM data integrity check service recalculates and compares cryptographic digests of the selected NVM component(s)—fabric, ENVM0, and ENVM1— to those previously computed and saved in NVM.

The contents of all the nonvolatile configuration memory segments are digested (hashed) using the SHA-256 algorithm. The results are compared to values stored in dedicated nonvolatile memory words located in each segment. If the contents are unchanged from when the digests were computed and stored during the original programming steps—that is, if the current and stored digests match—the test will pass; otherwise a failure is flagged.

The OPTIONS field in the NVM data integrity check service request selects the NVM components (fabric configuration, eNVM0, and eNVM1) for data integrity check, Table 2-93.

***Table 2-93* • NVM Data Integrity Check Service Request**

Offset	Length (bytes)	Field	Description
0	1	CMD = 23	Command
1	1	OPTIONS	Service options. Refer to Table 2-94.

***Table 2-94* • OPTIONS**

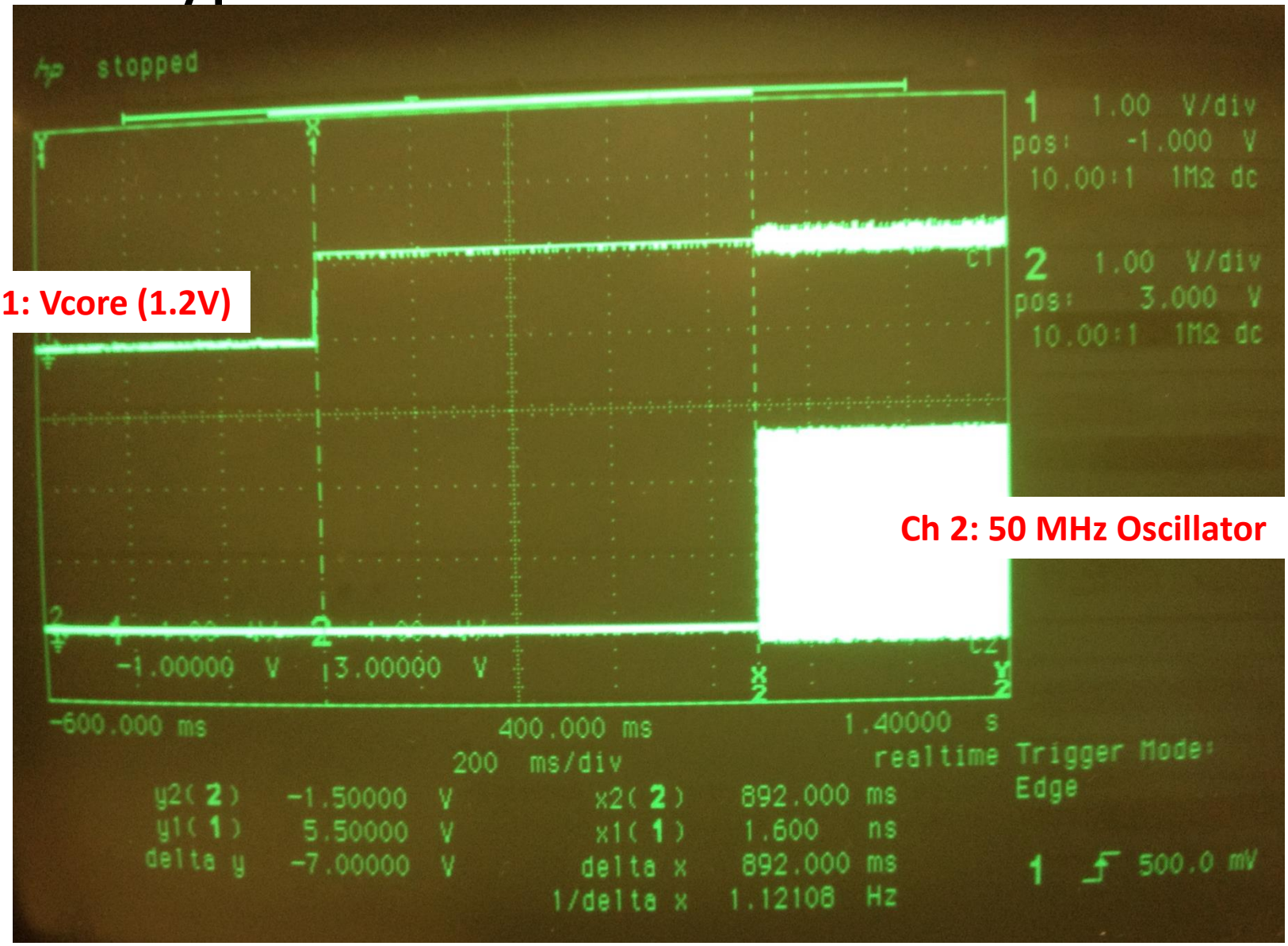
7	6	5	4	3	2	1	0
Reserved					ENVM1	ENVM0	FABRIC

From Microsemi documentation.

Summary of Test Results

- Testing smartfusion2 devices for start up time with flash cell checking on startup.
  - Using on-chip 50 MHz RC oscillator driven off-chip as the device start indicator.
  - Devices:
    - M2S005S: 6,060 Logic Elements, 128 kbytes ROM
    - M2S010TS: 12,084 Logic Elements, 256 kbytes ROM
    - M2S025TS: 27,696 Logic Elements, 256 kbytes ROM
    - M2S090TS: 86,315 Logic Elements, 512 kbytes ROM
  - No checks enabled: start time < 1 ms
    - Used 50 µs start time in device settings (this is a programmable value)
  - Fabric: flash cell checking enabled
  - NVM (non-volatile memories in on-chip computer):
    - Measured start times for 16k, 32k, 64k, 128k, 256k, and 512k memory sizes (size is programmable)
  - Preliminary Results for Fabric Checking (NVM Checking Disabled): Logic size vs. start time
    - M2S005S: 544 ms
    - M2S010TS: 897 ms
    - M2S025TS: 1,291 ms
    - M2S090TS: 2,530 ms

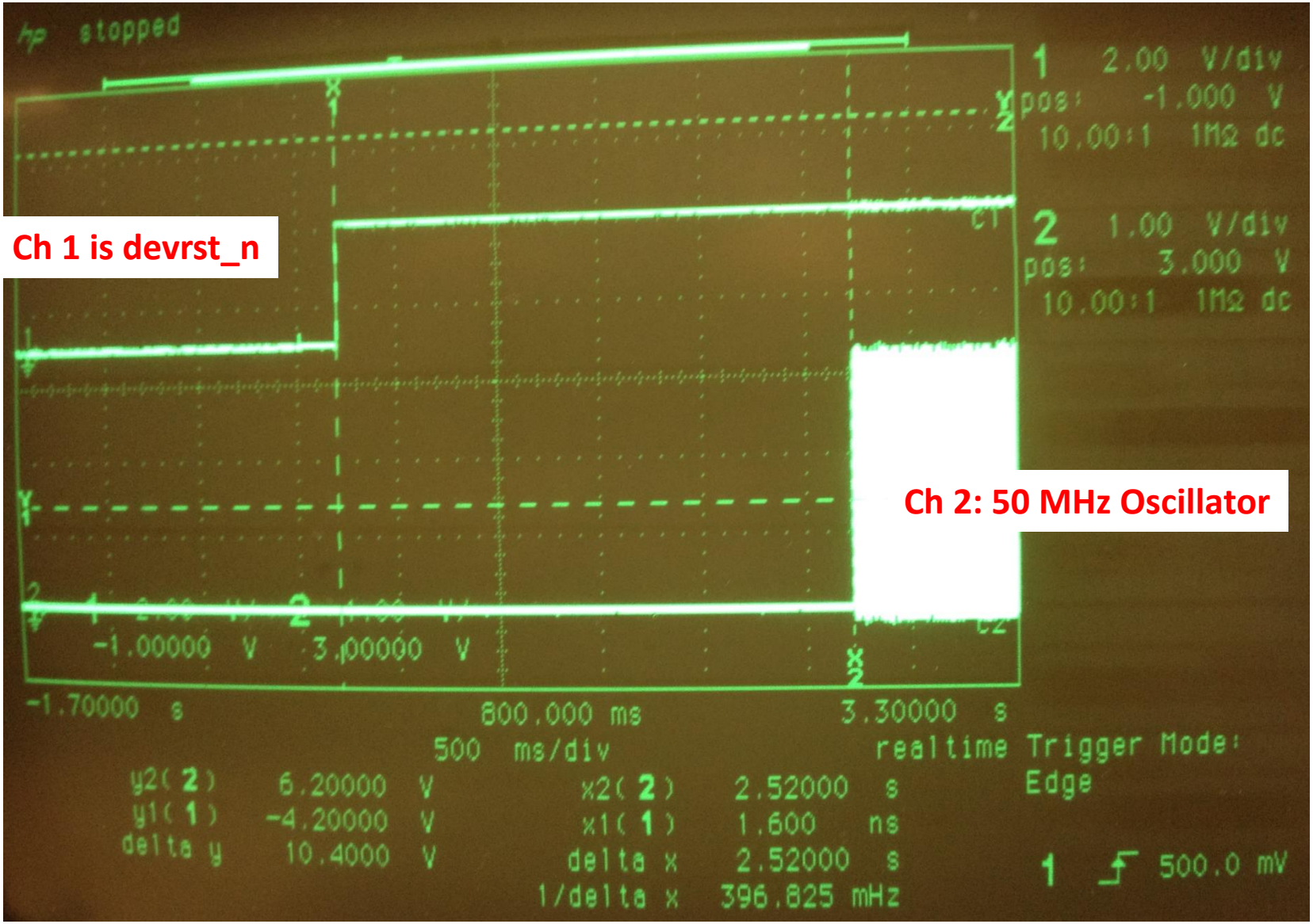
Typical Measurement Scheme



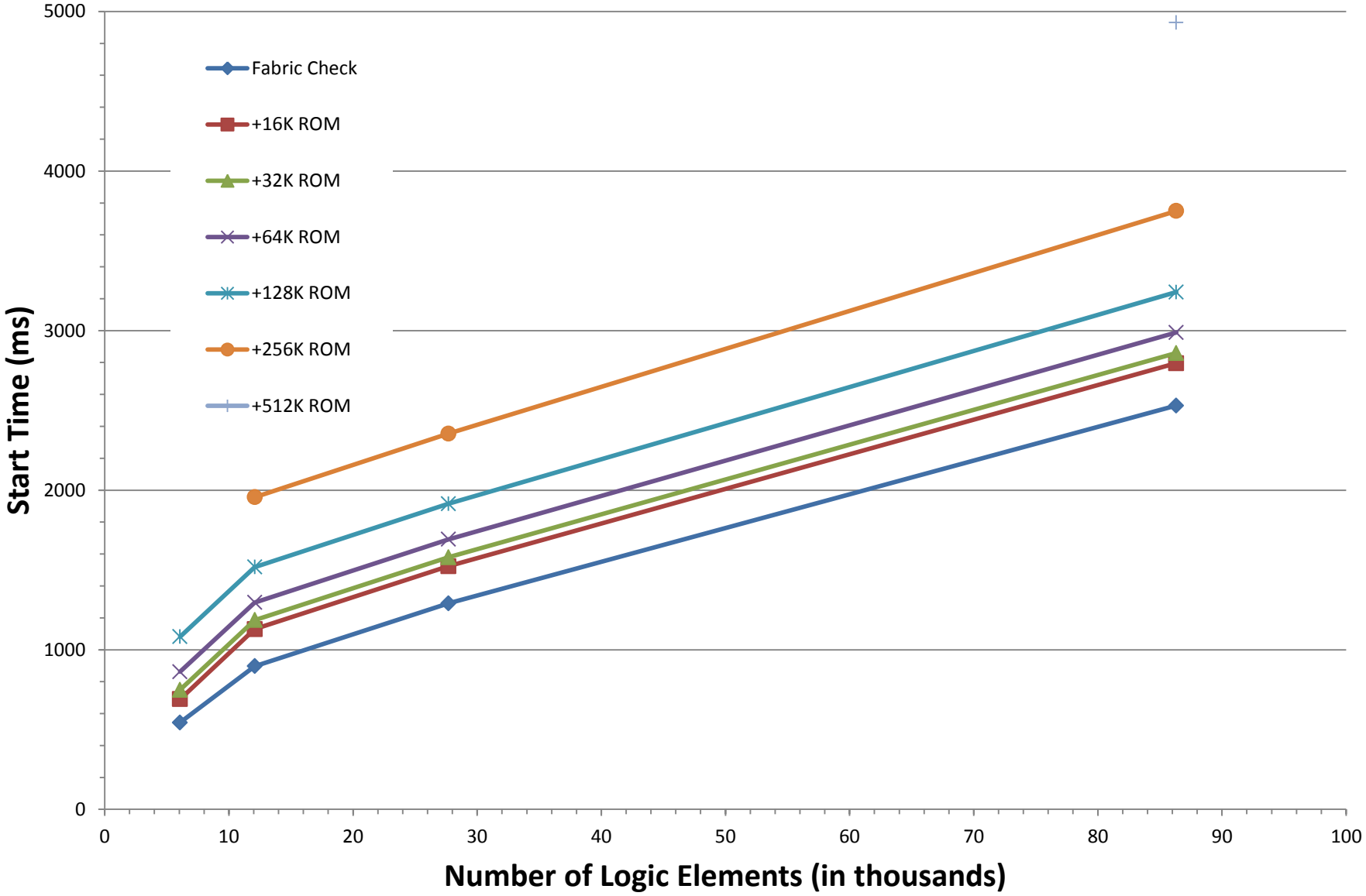
Ch 1: Vcore (1.2V)

Ch 2: 50 MHz Oscillator

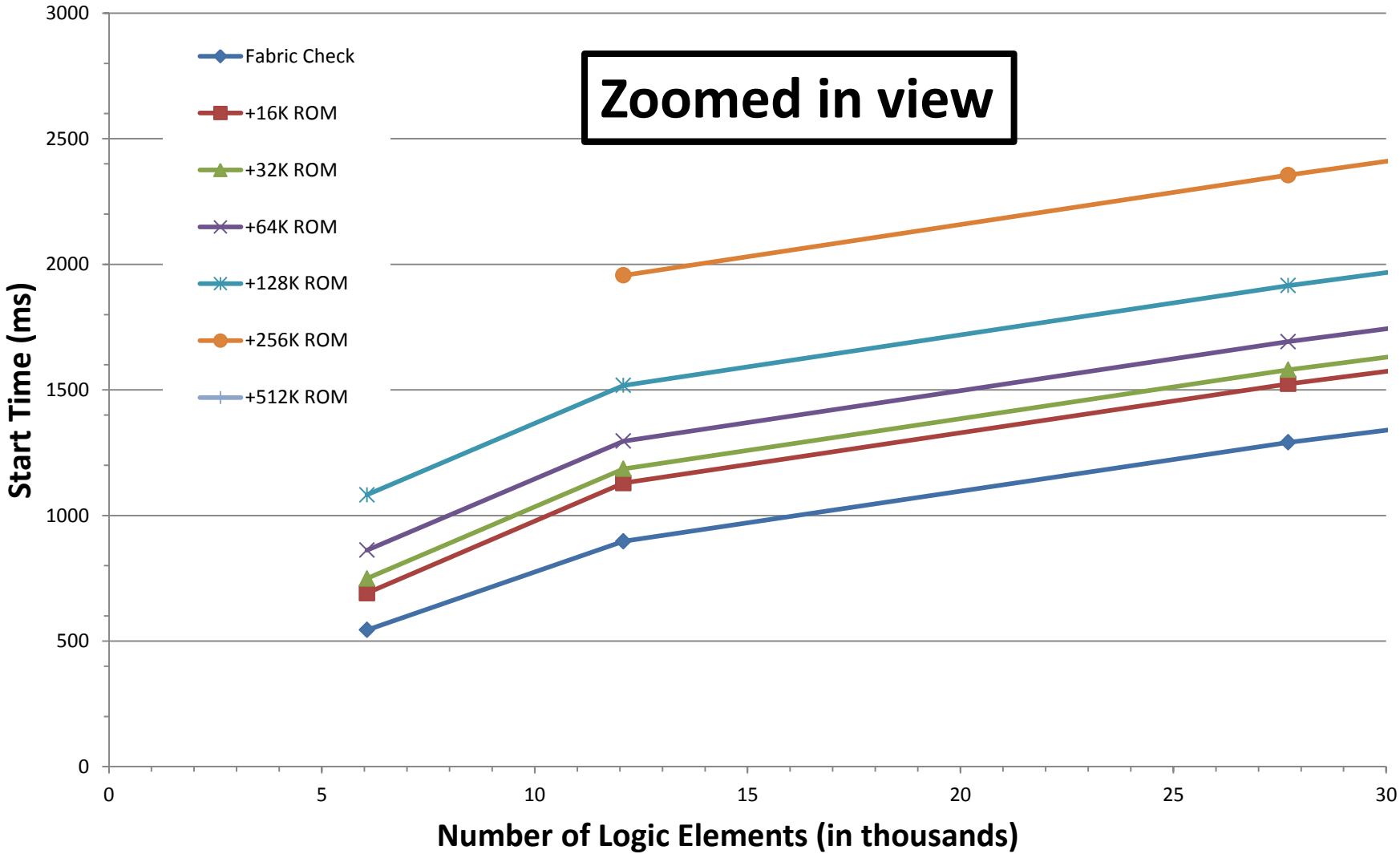
M2S090TS Measurement Scheme

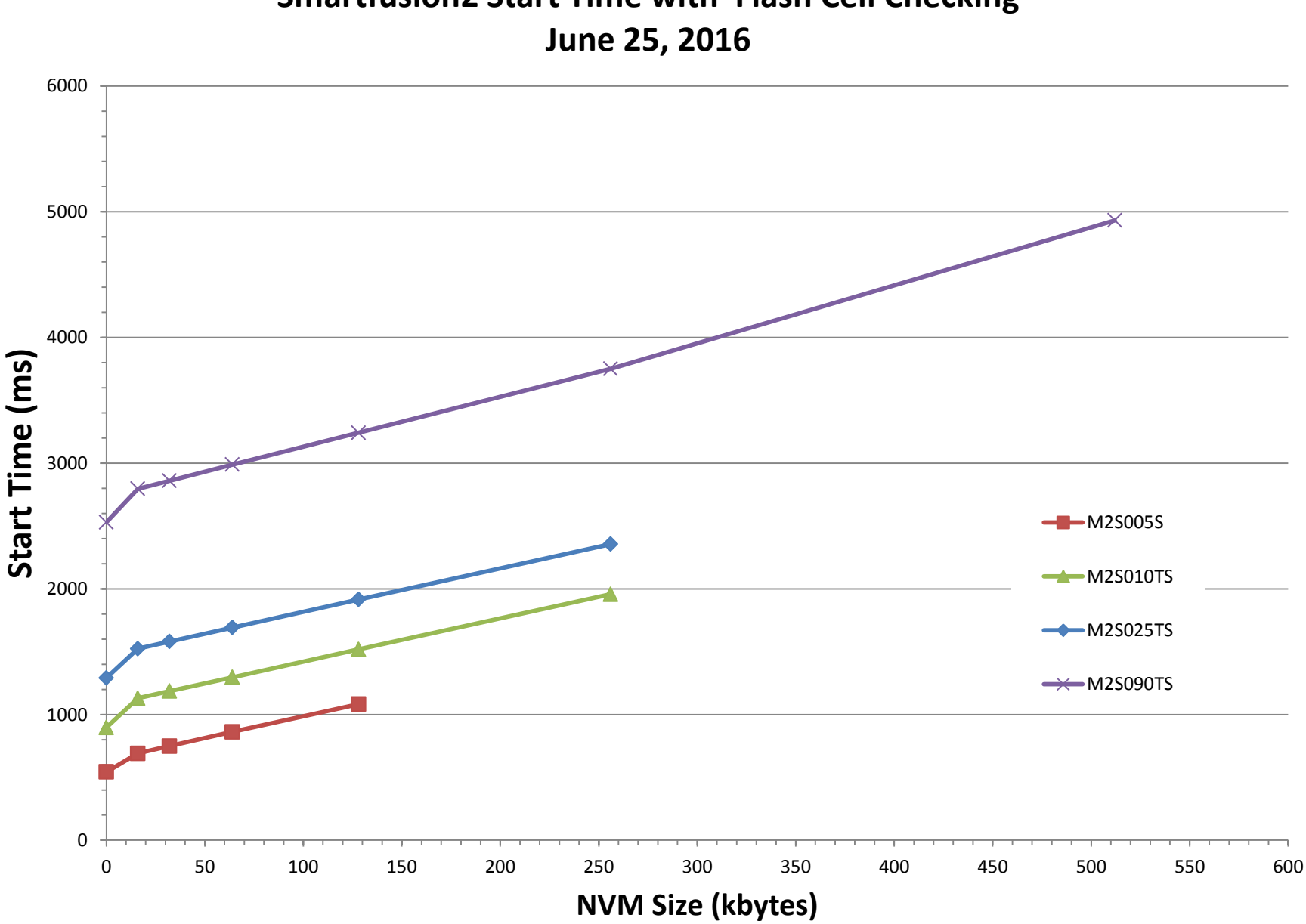


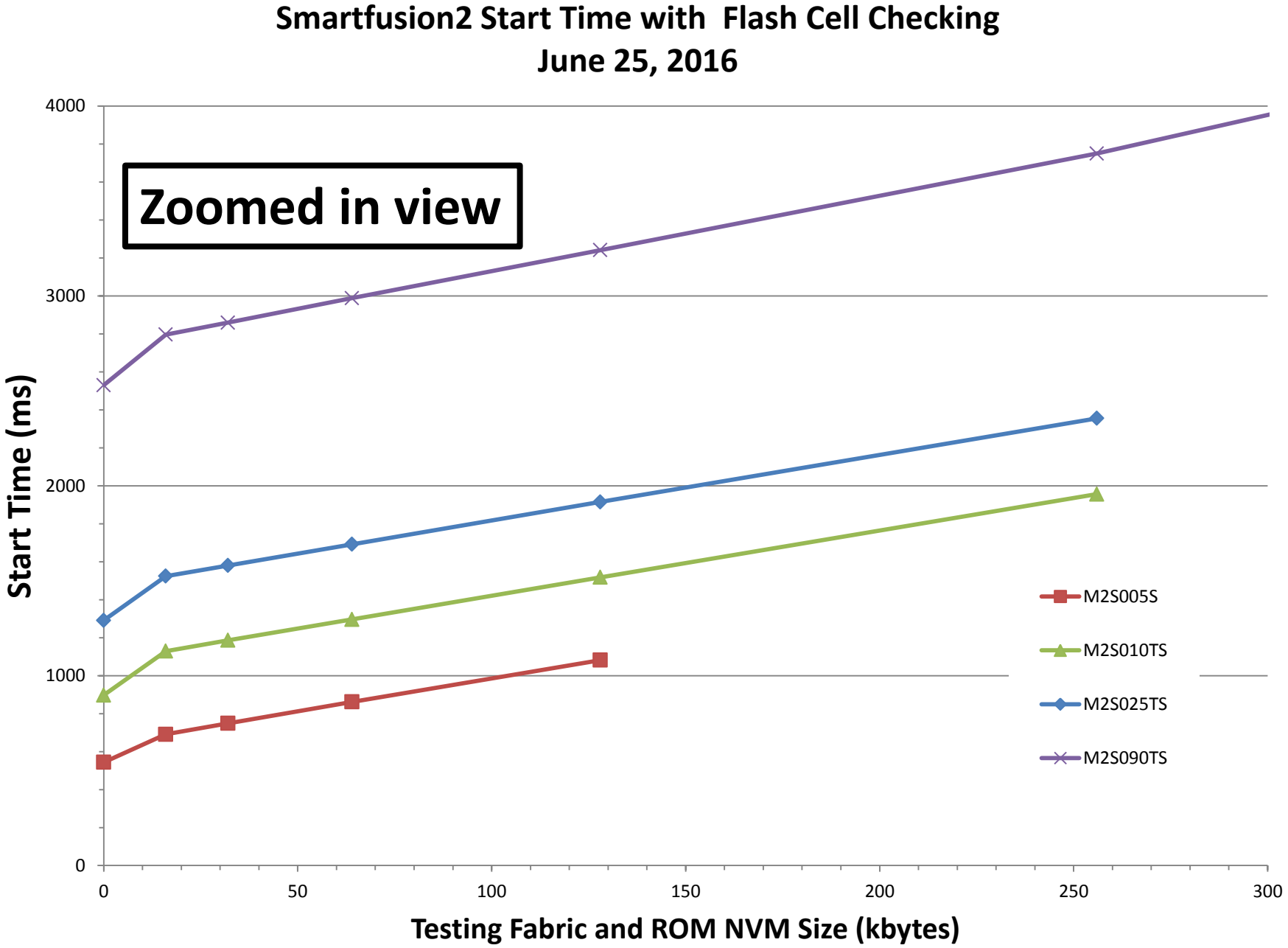
Smartfusion2 Start Time with Flash Cell Checking
June 25, 2016



Smartfusion2 Start Time with Flash Cell Checking
June 25, 2016




Smartfusion2 Start Time with Flash Cell Checking
June 25, 2016
Start Time (ms)
NVM Size (kbytes)
0
1000
2000
3000
4000
5000
6000
0
50
100
150
200
250
300
350
400
450
500
550
600
M2S005S
M2S010TS
M2S025TS
M2S090TS


Smartfusion2 Start Time with Flash Cell Checking
June 25, 2016
Zoomed in view
Start Time (ms)
4000
3000
2000
1000
0
0
50
100
150
200
250
300
Testing Fabric and ROM NVM Size (kbytes)
M2S005S
M2S010TS
M2S025TS
M2S090TS

System Controller Clocked by RC Oscillator

Clock Requirements

The System Controller is clocked by the on-chip 50 MHz RC oscillator.

It is powered by the VDD power pins and does not require external components for operation. The Chip Oscillators macro need not to be instantiated in the design for System Controller operation since it has a dedicated hardwired connection from the 50 MHz RC oscillator.

From Microsemi documentation.

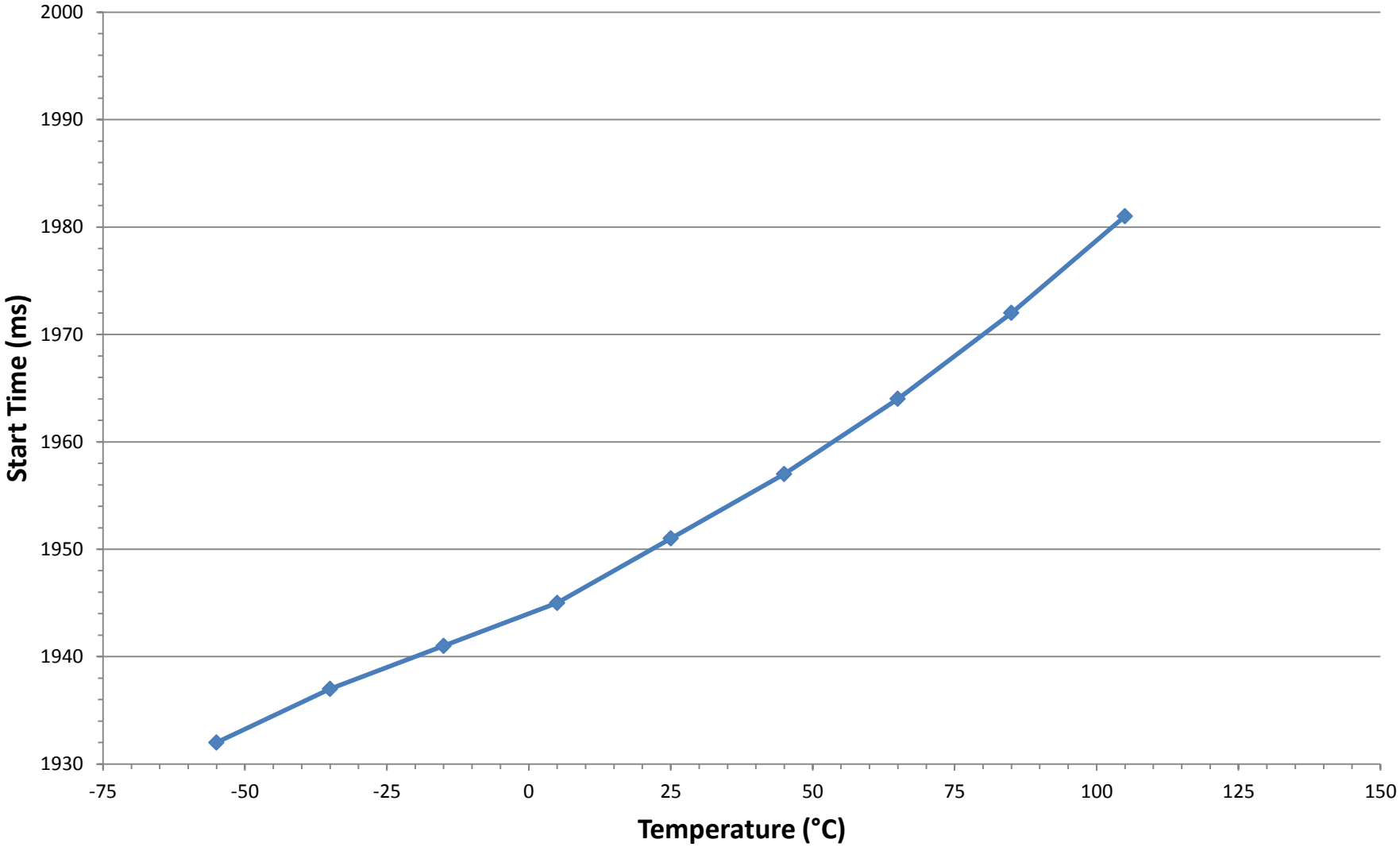
RC Oscillator Specification

***Table* • Electrical Characteristics of the 50 MHz RC Oscillator**

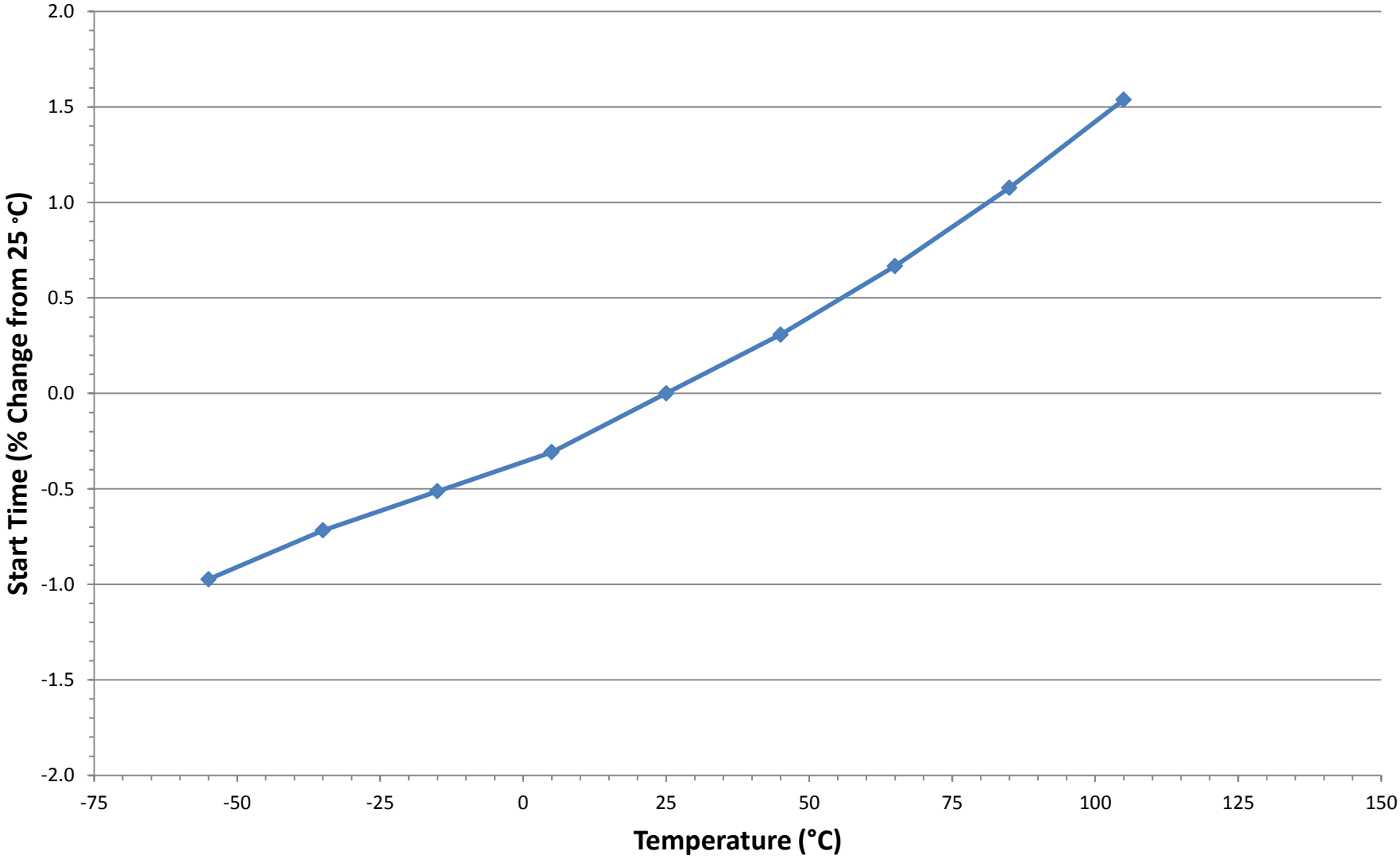
Parameter	Description	Condition	Min	Typ	Max	Units
F50RC	Operating frequency	–	–	50	–	MHz
ACC50RC	Accuracy	050 Devices	–	1	4	%
		005, 025, and 060 Devices	–	1	5	%
		090 Devices	–	1	6.3	%
		010 and 150 Devices	–	1	7.1	%
CYC50RC	Output duty cycle	–	–	49–51	46.5–53.5	%
JIT50RC	Output jitter (peak to peak)	Period Jitter				
		005, 010, 050, and 060 Devices	–	200	300	ps
		150 Devices	–	200	400	ps
		025 and 090 Devices	–	300	500	ps
		Cycle-to-Cycle Jitter				
		005 and 050Devices	–	200	300	ps
		010, 060, and 150 Devices	–	320	420	ps
		025 and 090 Devices	–	320	850	ps
IDYN50RC	Operating current	–	–	6.5	–	mA

From Microsemi documentation.

Start Time for M2S010TS
Flash Cell Check of Fabric and 256 kbytes of ROM
June 30, 2016



Start Time for M2S010TS
Flash Cell Check of Fabric and 256 kbytes of ROM
June 30, 2016



Upcoming Work

Table 24 • **Integrity Check Function**

Function	JTAG/SPI command	System services
Export digest and C-of-C (during bitstream programming)	Yes[1]	–
Export digest stored during programming (on demand)	Yes[2]	–
Compute/Export Fresh Digest (on demand)	Yes[3]	–
Compute/Report Fresh Flag (on demand)	Yes[4]	Yes[4]
Compute/Report Fresh Flag (after Power-on-Reset)	–	Yes[5]

1. As part of bitstream programming (FRAME_DATA JTAG/SPI commands): The digests and C-of-Cs (MACs) of the newly programmed data are exported for each included bitstream segment (BITS, KEYS, FPGA, eNVM, EOB) as they are loaded. Not available as a system service with IAP.
2. READ_DIGESTS JTAG/SPI commands: On demand, exports digests that were previously computed and stored during programming for the Fabric, and for the eNVM0 and eNVM1 ROM'd pages.
3. CHECK_DIGESTS JTAG/SPI commands: Fresh flags for all the security segments, the FPGA fabric, the eNVM0 & eNVM1 ROM'd pages, and the private eNVM. Immediately after running the CHECK_DIGESTS command, the fresh digests for the FPGA fabric, eNVM0 and eNVM1 can be exported.
4. DIGEST CHECK system service: Computes fresh flags on demand for the FPGA Fabric, the eNVM0 and eNVM1 ROMVM0 & eNVM1 ROM'd pages, and the system controller ROM.
5. POWER-ON-RESET DIGEST system service: Immediately after power-up, fresh flags for the same data as DIGEST CHECK can be read (only available if PoR digest option is selected in security policy).

From Microsemi documentation. UG0443, Rev. 8, July 2016.

Conclusion

- Appears Technical Manual criteria for validating configuration memory in flash-based FPGAs can now mostly be met.
  - FPGA includes a µP, flash and RAM memories, and peripherals
  - Comparison done in a separate internal section but not an external device.
  - Next year's work to investigate exporting checksum for external comparison.

- Stronger validation check then notional µController architecture
  - µController architecture relies on small amount of flash being functional for small memory scanning/CRC calculating software routine.
  - Time for validation is not specified by manufacturer.

- Validation time dependent on:
  - Size of FPGA "fabric" (logic area)
  - Amount of non-volatile memory **used** in the application.
  - Frequency of on-chip ring oscillator.

References

- "Long Term Data Retention of Flash Cells Used in Critical Applications," K. Bergevin, R. Katz, and D. Flowers, 58th Annual Fuze Conference, July 7-9, 2015, Baltimore, MD.
- "Viability of New COTS Technologies in Future Weapon Systems," J. Marchiondo, et. al, Sandia National Labs, September 2010.
- "High Reliability FPGAs in Fuze and Fuze Safety Applications," O'Neill, K., 59th Annual NDIA Fuze Conference, May 3-6, 2016, Charleston, South Carolina.
- "An Evaluation of Flash Cells Used in Critical Applications," R. Katz, D. Flowers, and K. Bergevin, 59th Annual Fuze Conference, May 3-6, 2016, Charleston, South Carolina.
- "Analysis & Recommendations for the Implementation of Flash Devices in Safety-Critical Applications," D. Flowers, and K. Bergevin, 59th Annual Fuze Conference, May 3-6, 2016, Charleston, South Carolina.
- "Environmental Effects on Data Retention in Flash Cells," R. Katz, D. Flowers, and K. Bergevin, 60th Annual Fuze Conference, May 9-11, 2017, Cincinnati, Ohio.
- "Advanced Analysis Techniques for the Implementation of Flash Devices in Safety-Critical Applications," D. Flowers, K. Bergevin, K. Islam, and M. Demmick, 60th Annual Fuze Conference, May 9-11, 2017, Cincinnati, Ohio.
- DoD Fuze Engineering Standardization Working Group, "Technical Manual for the Use of Logic Devices in Safety Features," March 8, 2011.