Freescale Semiconductor, Inc.
Application Note

Document Number: AN4989
Rev. 0, 08/2014

Implement IEC 60703 Class B Test for Kinetis E Family MCUs

by Jimmy Cheng

Contents

1. IEC 60730 B Requirements 2
2. CPU Registers Test 3
3. Program Counter Test 4
4. Interrupt Handing and Execution Test 4
5. Clock Test 4
6. Watchdog Test 4
7. Flash Test 5
8. RAM Test 6
9. Memory Address Test 8
10. Internal Data Path Test 8
11. Communications Tests 9
12. GPIO Test 9
13. Analog Peripherals Test 9
14. Analog Multiplexer Test 9
15. Freescale IEC 60730 Safety Library 10
16. Conclusion 10
17. References 10
18. Glossary 10
19. Appendix 11

The IEC 60730 safety standard defines the test and diagnostic methods that ensure the safe operation of embedded control hardware and software for household appliances.

The IEC 60730 classifies applicable equipment into three categories:

- Class A - Not intended to be relied upon for the safety of the equipment. Examples: Humidity controls, lighting controls, timers.
- Class B - Intended to prevent unsafe operation of the controlled equipment. Examples: Thermal cut-offs and door locks for laundry equipment.
- Class C - Intended to prevent special hazards, like explosion of the controlled equipment. Examples: Automatic burner controls, gas fired controlled dryer.

Kinetis E family devices are integrated with Cortex-M0+ processors and have safety modules like watchdog and CRC, along with rich internal modules like ICS, memory modules (Flash with EEPROM, RAM), timers (FTM, RTC, PIT), communications interface(SPI, I^2C and UART), human interface (GPIO, KBI) and analog modules (ADC, ACMP and 6-bit DAC, bandgap), etc. It is flexible in industry

© 2014 Freescale Semiconductor, Inc. All rights reserved.

freescale™

control and household appliances development. This application note describes how to implement the Kinetis E family IEC 60730 class B test introduction.

1 IEC 60730 B Requirements

The IEC 60730 specifies that the manufacturer of the automatic electronic control must design its software using one of the following structures:

- Single channel with functional test
- Single channel with periodic self-test
- Dual-channel without comparison

In a single channel with functional test structure, software is designed using a single CPU to execute functions as required. Prior to shipment, a functional test is performed to ensure that all critical features are functioning reliably.

In a single channel with periodic self-test structure, software is designed using a single CPU to execute functions as required, but periodic self tests occur while the electronic control is executing in its application. The CPU is expected to regularly check the various critical functions of the electronic control without conflicting with the end application's operation.

In a dual-channel without comparison structure, software is designed using two CPUs to execute on critical functions. Before executing a critical function, both CPUs are required to share that they have completed their corresponding task. For example, when a laundry door lock is released, one CPU stops the motor spinning the drum, and the other CPU checks the drum speed to verify it had stopped.

Dual-channel structure implementations will be more costly because two CPUs (or two MCUs) are required. The need for two devices to regularly communicate with each other also adds complexity. Single channel with functional test is the most popular structure implemented today, and most appliance manufacturers are moving to single channel with periodic self-test implementations.

Annex H. Table H.11.12.7 details the components that must be tested, depending on the software classification. Generally, each component offers optional measures to verify/test the corresponding components, providing the manufacturer flexibility. Throughout this application note, the intention is to employ the most cost-effective measure by using, where possible, on-board features of a single-chip microcontroller.

Table 1 summarizes Annex H Table H.11.12.7 in the IEC 60730-1:2010 standard. The table in Annex H lists all MCU internal modules that might need to be considered for IEC 60730 compliance.

Table 1. EC 60730 B components summary

	IEC60730 B Class Components	Fault/Error
1	1.1 CPU Registers	Stuck at
2	1.3 Program Counter	Stuck at
3	2. Interrupt handling & execution	No interrupt or too frequent interrupt
4	3. Clock	Wrong frequency (accuracy check)
5	4.1 Flash	All single bit faults

6	4.2 RAM	DC fault
7	4.3 Memory Address	Stuck at
8	5. Internal Data Path	Stuck at
9	5.2 Addressing(no need for single MCU)	Wrong address
10	6. Communications	Hamming distance (corruption check)
11	6.3 Timing	Wrong point in time/sequence
12	7. Input/output periphery(GPIO)	Fault conditions specified in H.27
13	7.2.1 ADC&DAC	Fault conditions specified in H.27
14	7.2.2 Analog multiplexer	Wrong addressing

2 CPU Registers Test

Table 2 shows the cortex-M0+ core register set summary. Each of these registers are 32 bits wide.

Table 2. Processor core register set summary

Name	Description
R0-R12	R0-R12 registers are general-purpose registers for data operations.
SP (R13)	The *Stack Pointer* (SP) is R13. In Thread mode, the CONTROL register indicates the stack pointer to use, *Main Stack Pointer* (MSP) or *Process Stack Pointer* (PSP).
LR (R14)	The *Link Register* (LR) is R14. It stores the return information for subroutines, function calls, and exceptions.
PC (R15)	The *Program Counter* (PC) is R15. It contains the current program address.
PSR	The *Program Status Register* (PSR) combines: • *Application Program Status Register* (APSR). • *Interrupt Program Status Register* (IPSR). • *Execution Program Status Register* (EPSR). These registers provide different views of the PSR.
PRIMASK	The PRIMASK register prevents activation of all exceptions with configurable priority. For information about the exception model the processor supports.
CONTROL	The CONTROL register controls the stack used, and optionally the code privilege level, when the processor is in Thread mode.

CPU general purpose registers R0~R12 are checked by writing pattern 0x55555555 to the register and the value is compared with an immediate value in another data register. The register value is then loaded with a "complemented" value of 0xAAAAAAAA and verified against with immediate value of 0xAAAAAAAA in another data register. The stack pointer register and special registers test are the same with 0x55555555 and 0xAAAAAAAA test pattern.

3 Program Counter Test

Due to Kinetis memory organization and Kinetis CPU behavior, it's not possible test all bits of PC register. There is an issue with linker file and flash memory allocation for small pieces of test code. The case, that Flash memory will be used for testing the PC register, causes the fragmentation of user application and problems in user code. For these reasons the test code can be placed only in RAM memory and thus 3 MSB bits of valid address range cannot be tested. There is a second issue for LSB bit of PC register, because the Kinetis core can't address odd addresses, so the LSB bit isn't tested. In summary, the 32 bits PC register is using only 29 bits to address complete address range of flash and RAM memories of CORTEX M0+. Kinetis core restrictions can only be tested at 29 bits.

4 Interrupt Handing and Execution Test

The purpose of the interrupt test is to check interrupts that can occur regularly. In a real-time embedded application, an MCU will almost always use interrupts to react to real-time events and to help prioritize the CPU on critical tasks or functions. IEC 60730 requires verification that the interrupt functionality for a critical function occurs as predicted. If no interrupt occurs, or too many occur, the electronic control functions safely.

The interrupt test function can be invoked at specified time intervals. It can be triggered by a timer interrupt to monitor and verify the interrupt operation. For example, the regular RTC interrupt occurs every 1s, set PIT interrupt to every 1ms, when the next RTC interrupt produces, PIT interrupt counter should add 1000.

5 Clock Test

The clock test is emphasized on system clock and bus clock test; the clocks should be neither too fast nor too slow. Furthermore, it is necessary to verify internal reference clock, external crystal, LPO and FLL function. Another way to perform this test is to use a communication bus like SPI to measure the SPI clock timing. The result will be compared to the theoretical value.

6 Watchdog Test

The watchdog test includes check watchdog reset possibility and its refresh mechanism.

Features of watchdog test in the Kinetis E family:

- Configurable input clock source from internal 32 KHz oscillator, 1 KHz (LPO) and external clock source.
- Programmable 16-bit timeout value
- Robust write sequence for counter refresh. Refresh sequence of writing 0x02A6 and then 0x80B4 within 16 bus clocks.
- Window mode option for the refresh mechanism. When Window mode is enabled, the watchdog must be refreshed after the counter has reached a minimum expected time value; otherwise, the watchdog resets the MCU.
- Programmable 16-bit window value.

- Interrupt request to CPU with interrupt vector for an interrupt service routine (ISR). Forced reset occurs 128 bus clocks after the interrupt vector fetch.
- Configuration bits are write-once-after-reset to ensure watchdog configuration cannot be mistakenly altered.
- Robust write sequence for unlocking write-once configuration bits. Unlock sequence of writing 0x20C5 and then 0x28D9 within 16 bus clocks for allowing updates to write-once configuration bits. Software must make updates within 128 bus clocks after unlocking and before WDOG closing unlock window.

The SIM_SDID register [WDOG] bit records watchdog reset event after watchdog reset produces.

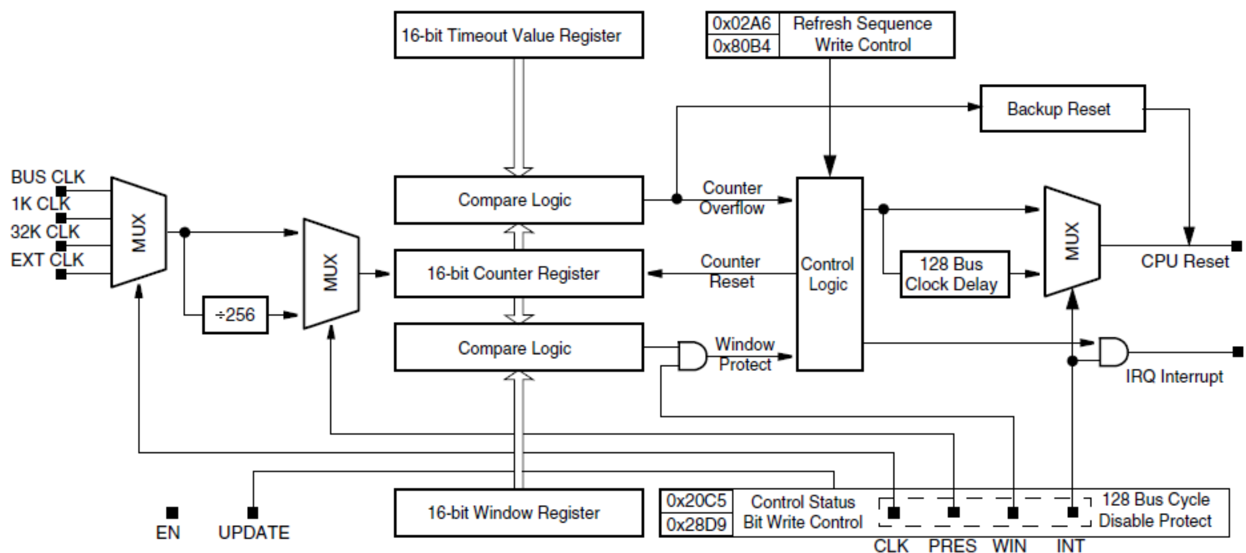

Figure 1. Kinetis E family watchdog diagram

7 Flash Test

Memory test functions, such as software-based CRC checks and March tests, are necessary. These functions can be used to check for memory integrity either periodically (as periodic self test) or upon power-up (as power-up software routines). In device hardware, memory access violation detection is supported to indicate any incorrect or unpredictable memory accesses.

Motorola S-records (.srec or .s19) file is usually used to Flash programming. The basic checksum has been filled in s record file. The S-Record format consists of 5 fields. These are the type field, length field, address field, data field, and the checksum. The lines always start with a capital S character.

S	Type	Record Length	Address	Data	Checksum

Type: The type field is a one-character field that specifies whether the record is an S0, S1, S2, S3, S5, S6, S7, S8 or S9 field.

Record Length: The record length field is a two-character (1 byte) field that specifies the number of character pairs (bytes) in the record, excluding the type and record length fields.

Address: This is a 2-, 3- or 4-byte address that specifies where the data in the S-Record is to be loaded into memory.

Data: The data field contains the executable code, memory-loadable data or descriptive information to be transferred.

Checksum: The checksum is an 8-bit field (one byte) that represents the least significant byte of the complement of the sum of the values represented by the pairs of characters making up the record's length, address, and data fields.

How to get checksum value on one S line, for example:

S11317C0C046C046C046C046FFF766FF1C1814104A

From the S format, we can get the checksum value is 0x4A.

Checksum = 0xFF-((0x13 + 0x17 + 0xC0 + 0xC0 + 0x46 +0xC0 + 0x46 + 0xC0 + 0x46 + 0xC0 + 0x46 + 0xFF +0xF7 + 0x66 + 0xFF 0x1C + 0x18 + 0x14 + 0x10)&0xFF) = 0x4A.

This checksum is too simple, it may not detect errors on multiple bytes in the code and it may not detect errors in the checksum itself. So it is not reliable for flash test. Cyclic redundancy check (CRC) algorithm has to be used in the flash single bit fault detect.

The Kinetis E family has a hardware CRC module to provide the fast mechanism for Flash, RAM and communications transfers and verification.

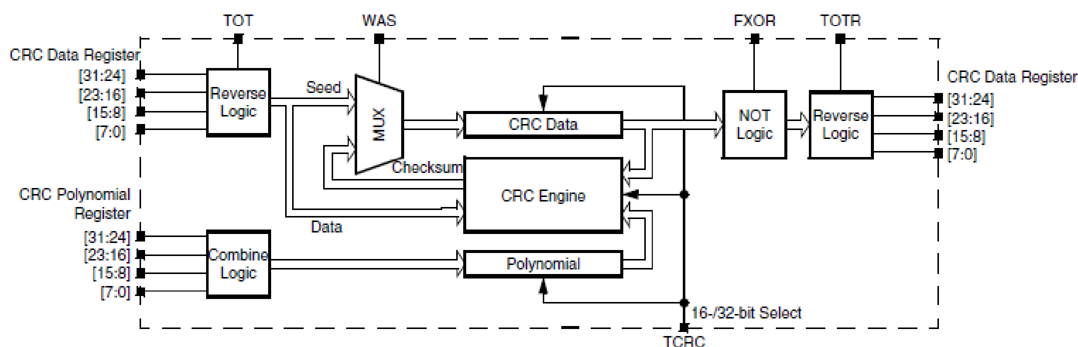

Figure 2. Kinetis E family CRC diagram

Some commonly used CRC divisors are as follows:

- CRC-16 = 1 1000 0000 0000 0101 = 8005(hex)
- CRC-CCITT = 1 0001 0000 0010 0001 = 1021(hex)
- CRC-32 = 1 0000 0100 1100 0001 0001 1101 1011 0111 = 04C11DB7 (hex)

8 RAM Test

The purpose of the RAM test is to check Control that no bit of the RAM memory is fault. Some RAM fault models are:

Stuck-At Fault (SAF) - The logic value of a cell or a line is always 0 or 1.

Transition Fault (TF) - A cell or a line that fails to undergo a 0?1 or a 1 ?0 transition.

Coupling Fault (CF) - A write operation to one cell changes the content of a second cell.

Neighborhood Pattern Sensitive Fault (NPSF) - The content of a cell, or the ability to change its content, is influenced by the contents of some other cells in the memory.

Address Decoding Fault (AF)- Any fault that affects address decoder:

- With a certain address, no cell will be accessed.
- A certain cell is never accessed.
- With a certain address, multiple cells are accessed simultaneously.
- A certain cell can be accessed by multiple addresses.

March tests are the simplest and most efficient tests for detecting AFs, SAFs, TFs and CFs. Table 3 lists some common March test algorithms.

Table 3. March algorithms

Name	Algorithm
March X	{⇕(w0);⇑(r0,w1);⇓(r1,w0);⇕(r0)}
March C-	{⇕(w0);⇑(r0,w1);⇑(r1,w0);⇓(r0,w1);⇓(r1,w0);⇕(r0)}
March C	{⇕(w0);⇑(r0,w1);⇑(r1,w0);⇕(r0)}⇓(r0,w1);⇓(r1,w0);⇕(r0)}
March B	{⇕(w0);⇑(r0,w1,r1,w0,r0,w1);⇑(r1,w0,w1);⇓(r1,w0,w1,w0);⇓(r0,w1,w0)}

KEY:

⇕ : indicates either ascending or descending order
⇑ : indicates address ascending order
⇓ : indicates address descending order
w0 : write 0 at current location
w1 : write 1 at current location
r0 : read current location, expecting a 0
r1 : read current location, expecting a 1
(….): algorithm element
{(…),(…),…,(…)}: full algorithm

For instance:

March X: {⇕(w0);⇑(r0,w1);⇓(r1,w0);⇕(r0)}
Step1: ⇕(w0)
Write 0 to all memory with either ascending or descending address order.
Step2: ⇑ (r0,w1)
Scan memory location from low to high address, check the read bit is 0 or not. If not, the test is failed on this bit. Or write 1 to the memory from low to high address.
Step3: ⇓(r1,w0)
Scan memory location from high to low address, check the read bit is 1 or not. If not, the test is failed

on this bit. Or write 0 to the memory from high to low address.

Step4: | (r0)

Read 0 to all memory with either ascending or descending address order. And check the read bit is 0 or not. If not, the test is failed on this bit.

For RAM testing, two different algorithms are implemented, as described in the following sections. The test routines are destructive in their nature. To make a transparent test pattern, the RAM was partitioned into four segments (0 - 3), where the segment 3 is used as a redundant area for copying other segments temporarily while the test pattern is being executed on them. The function first checks if SP is not located in the block 3 and if it is, the error code is returned and the function won't proceed. If SP is not in the block 3, the block 3 is checked. Then the function applies.

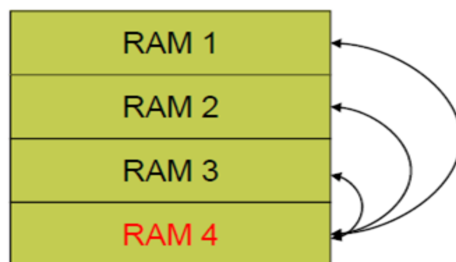

Figure 3. Transparent test pattern

Split RAM into four segments. The 4th segment is “shadow” RAM used to temporarily store other segment variables until the March test is completed. For example, complete the following in RAM1 test process:

- RAM 1 copy to RAM 4
- Verify copy is successful
- Deploy MARCH test on RAM 1
- Copy RAM 4 to RAM 1
- Verify copy is successful
- Deploy normal application code

9 Memory Address Test

Measures such as the periodic static memory test on RAM and the periodic checksum on Flash would highlight a stuck at fault on the internal address bus. For example, if the address bus had a stuck at fault, both of these tests would produce an error.

10 Internal Data Path Test

For single-chip microcontrollers, it is similar to the memory address test.

11 Communications Tests

External communication tests should be completed by following instructions on the items below.

11.1 Transfer redundancy test

The transfer redundancy is a fault/error control technique that protects against coincidental and/or systematic errors in the input and output information. It is achieved by transferring the data between the transmitter and receiver. The data is transferred at least twice in succession and then compared.

11.2 Protocol test

The protocol test is a fault/error control technique in which the data is transferred to and from the computer components to detect errors in the internal communication protocol.

11.3 CRC single word test

A CRC polynomial is used to calculate the CRC checksum of the transmitted message. At the transmitting end, this CRC checksum is appended to the message before transmitting it. At the receiving end, the receiver uses the same CRC polynomial to compute the CRC checksum, and compares the computed value with the received value.

12 GPIO Test

The GPIO pins on the Kinetis E family can be configured as inputs or outputs individually. They also support individual pullup function, and this feature can be used to prevent floating input condition of unused I/O pins. The user software can perform periodic known condition checks on GPIO port pins to monitor expected I/O states. Furthermore, Kinetis E family devices have set, clear and toggle registers to control GPIO pins output states.

It is necessary to detect the actual voltage level, high voltage level is “1” and low voltage is “0”, so that GPIO input can be accepted.

13 Analog Peripherals Test

The analog peripherals in the Kinetis E family include ADC, internal bandgap reference (about 1.2V), and ACMP with 6-bit DAC. Plausibility check for ADC is to ensure A/D results are within acceptable A/D drift count. Proper comparator operation can be checked by comparing known external voltages against internal 6-bit DAC.

14 Analog Multiplexer Test

The analog multiplexer ensures proper operation of selectable analog channels. For example, ADC has 23 analog input channels, to verify the operation of the analog multiplexer, known voltage values are applied to all external input channels. These values are read and compared with the applied voltage for verification.

Furthermore, temperature sensor channel and bandgap channel is connected internally, the two channels can be checked by reading ADC conversion result.

15 Freescale IEC 60730 Safety Library

Freescale has developed safety features, including an IEC 60730 Safety Library, to help manufacturers of automatic controls in the large appliance and industrial control market meet the IEC 60730 class B regulation. These safety features consist of both hardware and software, and have been developed for S08, Kinetis (Cortex-M4 and Cortex-M0+ core) and DSC family of microcontrollers.

The Freescale IEC 60730 Safety Library test covers CPU registers, CPU instructions, RAM, Flash and watchdog test.

In line with IEC 60730 requirements Freescale has incorporated key hardware features. The independent clocked Watchdog Timer provides a safety mechanism to monitor:

- The flow of the software
- Interrupt handling & execution
- CPU clock (too fast, too slow and no clock)
- CRC Engine - this provides a fast mechanism for testing the Flash memory and check on serial communication protocols (UARTS, I^2C, SPI)

Get more Freescale IEC 60730 Safety Library information via this freescale.com (search “IEC 60730 Safety Standard for Household Appliances”).

16 Conclusion

This application note demonstrates how to use Kinetis E family devices to meet IEC 60730 Class B requirements.

17 References

MKE02Z64M20SF0RM Reference Manual

AN4873 IEC 60730B Safety Routines for Kinetis MCUs

AN3257 Meeting IEC 60730 Class B Compliance with the MC9S08AW60

18 Glossary

ISR: Interrupt Service Routine

WDOG: Watchdog

CRC: Cyclic Redundancy Check

ACMP: Analog Comparator

SAF: Stuck-At-Fault

19 Appendix

Table 4. IEC 60730-1 TABLE H.11.12.7

Component	Fault/error	Software Class		Acceptable measures		Definitions
		Class B	Class C			
1.CPU						
1.1 Registers	Stuck at	rq		Functional test, Periodic self-test using either -static memory test -word protection with single bit redundancy	or	H.2.16.5
						H.2.16.6
					or	H.2.19.6
						H.2.19.8.2
1.CPU	DC fault		rq	Comparison of redundant CPUs by either -reciprocal comparison -independent hardware comparator Internal error detection, redundant memory with comparison, Periodic self-test using either - walkpat memory test - Abraham test - transparent GALPAT test Word protection with multi-bit redundancy static memory test and word protection with single-bit redundancy		
1.1 Registers						H.2.18.15
					or	H.2.18.3
					or	H.2.18.9
					or	H.2.19.5
						H.2.19.7
						H.2.19.1
					or	H.2.19.2.1
					or	H.2.19.8.1
						H.2.19.6
						H.2.19.8.2
1.2 Instruction decoding and execution	Wrong decoding and execution		rq	Comparison of redundant CPUs by either -reciprocal comparison -independent hardware comparator Internal error detection, Periodic self-test using equivalence class test		
						H.2.18.15
					or	H.2.18.3
					or	H.2.18.9
					or	H.2.18.5

Table 4. IEC 60730-1 TABLE H.11.12.7 (continued)

1.3 Program counter	Stuck at	rq		Functional test, periodic self-test independent time-slot monitoring, logical monitoring of the program sequence Periodic self-test and monitoring using either -independent time-slot and logical monitoring - internal error detection Comparison of redundant functional channels by either -reciprocal comparison -independent hardware comparator	or	H.2.16.5
					or	H.2.16.6
					or	H.2.18.10.4
					or	H.2.18.10.2
						H.2.16.7
			rq		or	H.2.18.10.3
					or	H.2.18.9
					or	
					or	H.2.18.15
						H.2.18.3
1.4 Addressing	DC fault		rq	Comparison of redundant CPUs by either -reciprocal comparison -independent hardware comparator Internal error detection Periodic self-test using a testing pattern of: the address lines full bit bus parity including the address		
						H.2.18.15
					or	H.2.18.3
					or	H.2.18.9
						H.2.16.7
					or	H.2.18.22
						H.2.18.1.1
1.5 Data paths instruction decoding	DC fault and execution		rq	Comparison of redundant CPUs by either -reciprocal comparison -independent hardware comparator Internal error detection Periodic self-test using a testing pattern of: data redundancy multi-bit bus parity		
						H.2.18.15
					or	H.2.18.3
					or	H.2.18.9
						H.2.16.7
					or	H.2.18.22
						H.2.18.1.2
2.Interrupt handling and execution	or too frequent interrupt	rq		Functional test, time-slot monitoring	or	H.2.16.5
						H.2.18.10.4
	No interrupt or too frequent interrupt related to different sources		rq	Comparison of redundant functional channels by either: -reciprocal comparison -independent hardware comparator, -independent time-slot and logical monitoring		
					or	H.2.18.15
					or	H.2.18.3
						H.2.18.10.3
3. Clock		rq		Frequency monitoring time-slot monitoring	or	H.2.18.10.1
						H.2.18.10.4

Table 4. IEC 60730-1 TABLE H.11.12.7 (continued)

	Wrong freq. for quartz sync'd clock; harmonics/ subharmonics only		rq	Frequency monitoring time-slot monitoring Comparison of redundant functional channels by either: -reciprocal comparison -independent hardware comparator,	or or or 	H.2.18.10.1 H.2.18.10.4 H.2.18.15 H.2.18.3
4. Memory						
4.1 Invariable memory	All single bit faults	rq		Periodic modified checksum multiple checksum word protection with single bit redundancy	or or 	H.2.19.3.1 H.2.19.3.2 H.2.19.8.2
	99.60% coverage of all information errors		rq	Comparison of redundant CPUs by either -reciprocal comparison -independent hardware comparator, Redundant memory with comparison periodic cyclic redundancy check, either: -single word -double word word protection with multi-bit redundancy	 or or or or or 	 H.2.18.15 H.2.18.3 H.2.19.5 H.2.19.4.1 H.2.19.4.2 H.2.19.8.1
4.2 Variable memory	DC fault	rq		Periodic static memory test, word protection with single bit redundancy	or 	H.2.19.6 H.2.19.8.2
	DC fault and dynamic cross links		rq	Comparison of redundant CPUs by either -reciprocal comparison -independent hardware comparator, Redundant memory with comparison Periodic self-test using either - walkpat memory test - Abraham test - transparent GALPAT test Word protection with multi-bit redundancy	 or or or or or	 H.2.18.15 H.2.18.3 H.2.19.5 H.2.19.7 H.2.19.1 H.2.19.2.1 H.2.19.8.1
4.3 Addressing	Stuck at	rq		Word protection with single bit parity incl. the address	or	H.2.19.18.2

Table 4. IEC 60730-1 TABLE H.11.12.7 (continued)

(relevant to variable and invariable memory)						
	DC fault		rq	Comparison of redundant CPUs by either -reciprocal comparison -independent hardware comparator, full bus redundancy Testing pattern periodic cyclic redundancy check, either: -single word -double word Word protection with multi-bit redundancy incl. address		
					or	H.2.18.15
					or	H.2.18.3
						H.2.18.1.1
						H.2.18.22
					or	H.2.19.4.1
					or	H.2.19.4.2
						H.2.19.8.1
5. Internal data Path	Stuck at DC fault	rq		Word protection with single bit parity incl. the address		H.2.19.18.2
			rq	Comparison of redundant CPUs by either -reciprocal comparison -independent hardware comparator,		
					or	H.2.18.15
					or	H.2.18.3
5.1 Data				Word protection with multi-bit redundancy including the address, or data redundancy, testing pattern protocol test	or	H.2.19.8.1
					or	H.2.18.2.1
					or	H.2.18.22
						H.2.18.14
5.2 Addressing	Wrong address	rq		Word protection with multi-bit redundancy incl. address		H.2.19.8.1
			rq	Comparison of redundant CPUs by either -reciprocal comparison -independent hardware comparator, Word protection with multi-bit redundancy including the address, or data redundancy, testing pattern		
					or	H.2.18.15
					or	H.2.18.3
					or	H.2.19.8.1
					or	H.2.18.2.1
					or	H.2.18.22
6 External communications				Word protection with multi-bit redundancy, CRC - single word Transfer redundancy protocol test	or	H.2.19.8.1
	Hamming distance 3	rq			or	H.2.19.4.1
					or	H.2.18.2.2
						H.2.18.4

Table 4. IEC 60730-1 TABLE H.11.12.7 (continued)

6.1 Data	Hamming distance 4		rq	CRC - double word data redundance or comparison of redundant functional channels be either -reciprocal comparison -independent hardware comparator	or	H.2.19.4.2
					or	H.2.18.2.1
					or	H.2.18.15
						H.2.18.3
6.2 Addressing	Wrong address		rq	Word protection with multi-bit redundancy incl address CRC singl word inc address transfer redundancy protocol test	or	H.2.19.8.1
					or	H.2.19.4.1
					or	H.2.18.2.2
						H.2.18.14
	Wrong & multiple addressing		rq	CRC double word incl address full bus redundancy of data & address comparison of redundant communication channels by either: -reciprocal comparison -independent hardware comparator	or	H.2.19.4.2
					or	H.2.18.1.1
					or	H.2.18.15
						H.2.18.3
6.3 Timing	Wrong point in time	rq	rq	Time-slot monitoring scheduled transmission independent time-slot and logical monitoring comparison of redundant communication channels by either: -reciprocal comparison -independent hardware comparator,	or	H.2.18.10.4
					and	H.2.18.18
		rq	rq		or	H.2.18.10.3
					or	H.2.18.15
						H.2.18.3
	Wrong Sequence	rq		Logical monitoring time-slot monitoring scheduled transmission	or	H.2.18.10.2
					or	H.2.18.10.4
						H.2.18.18

Table 4. IEC 60730-1 TABLE H.11.12.7 (continued)

7 I/O Periphery						
	Fault conditions specified H.27	rq		Plausibility check		H.2.18.13
			rq	Comparison of redundant CPUs by either		
				-reciprocal comparison		
				-independent hardware comparator,	or	H.2.18.15
				input comparison	or	H.2.18.3
				multiple parallel outputs	or	H.2.18.8
				output verification	or	H.2.18.11
				testing pattern	or	H.2.18.12
				code safety	or	H.2.18.22
						H.2.18.2
7.2 Analog I/O						
7.2.1 A/D & D/A Convertor	Fault conditions specified H.27	rq		Plausibility check		H.2.18.13
			rq	Comparison of redundant CPUs by either		
				-reciprocal comparison		
				-independent hardware comparator,	or	H.2.18.15
				input comparison	or	H.2.18.3
				multiple parallel outputs	or	H.2.18.8
				output verification	or	H.2.18.11
				testing pattern	or	H.2.18.12
						H.2.18.22
7.2.2 Analog multiplexor	Wrong addressing	rq		Plausibility check		H.2.18.13
			rq	Comparison of redundant CPUs by either		
				-reciprocal comparison		
				-independent hardware comparator,	or	H.2.18.15
				input comparison	or	H.2.18.3
				testing pattern	or	H.2.18.8
						H.2.18.22

Table 4. IEC 60730-1 TABLE H.11.12.7 (continued)

8. Monitoring device and comparators	Any output outside the static and dynamic functional specification		rq	Tested monitoring redundant monitoring and comparison error recognizing means	or	H.2.18.21
					or	H.2.18.17
						H.2.18.6
9. Custom chips (eg ASIC,GAL gate array)	Any output outside the static and dynamic functional specification	rq		Periodic self-test		H.2.16.6
			rq	Periodic Self-test and monitoring dual channel(diverse) with comparison, error recognizing means	or	H.2.16.7
					or	H.2.16.2
						H.2.18.6

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Information in this document is provided solely to enable system and software implementers to use Freescale products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document.

Freescale reserves the right to make changes without further notice to any products herein. Freescale makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," must be validated for each customer application by customer's technical experts. Freescale does not convey any license under its patent rights nor the rights of others. Freescale sells products pursuant to standard terms and conditions of sale, which can be found at the following address: freescale.com/SalesTermsandConditions.

Freescale, the Freescale logo, the Energy Efficient Solutions logo and Kinetis are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. ARM and Cortex are the registered trademarks of ARM Limited.

All other product or service names are the property of their respective owners.

© 2014 Freescale Semiconductor, Inc.

Document Number: AN4989
Rev. 0
08/2014