



Power Matters.



Designing with SmartFusion2 cSoCs Microcontroller Subsystem



Agenda

- SmartFusion2 Overview
- SmartFusion2 Microcontroller Subsystem (MSS)
- SmartFusion2 Design Environment
- SmartFusion2 Programming
- Firmware Development with SoftConsole
- SmartFusion2 Ecosystem
- Hands-on Labs

SmartFusion2 Customizable SoC

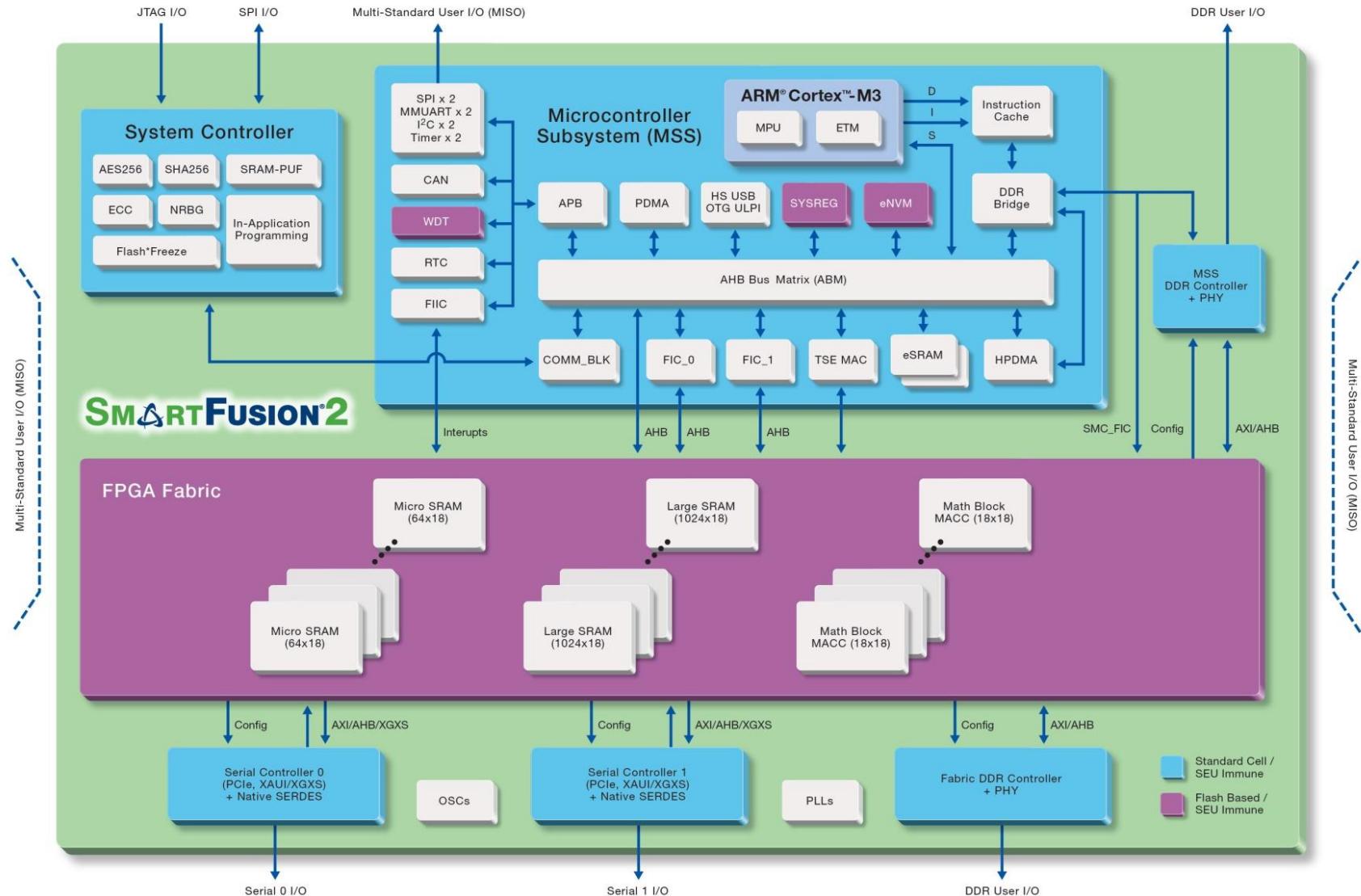
- Highest Reliability
 - Zero FIT Flash FPGA Configuration
 - SECDED on ASIC SRAM Blocks
 - SEU immune latches on Instruction Cache and Buffers
- Most Secure FPGA
 - Design Security - DPA Hardened
 - Data Security AES256, SHA256
 - Random Number Generator
- Lowest Power FPGA
 - Low static and dynamic power
 - < 1.0 mW in flash-freeze mode
 - 10 mW static power during operation
 - 65 nm nonvolatile, low power flash process



SmartFusion2 Highlights

- SEU immune FPGA configuration
 - 4 input LUTs with carry chains and DFF
 - Embedded SRAMS 18 Kbit and 1 Kbit
 - Fast Math Blocks (18x18 Multiplier and 44-Bit Accumulator)
- 166 MHz ARM Cortex-M3 microcontroller
 - 8 KB Instruction Cache, On Chip NVM and SRAM
 - Extensive peripherals CAN, TSE, USB
 - Embedded Trace Macrocell for debug
- High speed serial interfaces
 - 5Gbps SERDES, PCIe, XAUI / XGXS+ Native SERDES
- High Speed Memory Interfaces
 - 667 Mbps DDR2/3 controllers with SECDED protection

SmartFusion2 Architecture



SmartFusion2 Family

	Features	M2S005	M2S010	M2S025	M2S050	M2S060	M2S090	M2S150
Logic / DSP	Maximum Logic Elements (4LUT + DFF)*	6,060	12,084	27,696	56,340	56,340	86,316	146,124
	Math Blocks (18x18)	11	22	34	72	72	84	240
	PLLs and CCCs	2		6			8	
Security	AES256, SHA256, RNG			1 each				
	ECC, PUF			-			1 each	
MSS	Cortex-M3 + Instruction cache			Yes				
	eNVM (Kbytes)	128		256			512	
	eSRAM (Kbytes)			64				
	eSRAM (Kbytes) Non-SECDED			80				
	TSE, USB, CAN, Watchdog, RTC			1 each				
	SPI, I2C, Multi-Mode UART, Timer			2 each				
Fabric Memory	LSRAM 18K Blocks	10	21	31	69	69	109	236
	uSRAM 1K Blocks	11	22	34	72	72	112	240
	Total RAM (K bits)	191	400	592	1314	1314	2074	4488
High Speed	DDR Controllers (Count x Width)		1x18		2x36		1x18	2x36
	SERDES Lanes	0	4		8		4	16
	PCIe End Points	0	1		2			4
User I/Os	MSIO (3.3V)	115	123	157	139	271	309	292
	MSIOD (2.5V)	28	40	40	62	40	40	106
	DDRIO (2.5V)	66	70	70	176	76	76	176
	Total User I/O	209	233	267	377	387	425	574

* Total logic may vary based on utilization of DSP and memories in the design.

Package Options

I/Os Per Package

	Package Options																			
Package Type	FCS(G)325		VF(G)256		FCS(G)536		VF(G)400		FCV(G)484		TQ(G)144		FG(G)484		FG(G)676		FG(G)896		FC(G)1152	
Pitch (mm)	0.5		0.8		0.5		0.8		0.5		0.5		1.0		1.0		1.0		1.0	
Length x Width (mm)	11x11		14x14		16x16		17x17		19x19		20x20		23x23		27x27		31x31		35x35	
Device	I/O	Lanes	I/O	Lanes	I/O	Lanes	I/O	Lanes	I/O	Lanes	I/O	Lanes	I/O	Lanes	I/O	Lanes	I/O	Lanes	I/O	Lanes
M2S005 (S)	—	—	161	—	—	—	171	—	—	—	84	—	209	—	—	—	—	—	—	—
M2S010 (S/T/TS)	—	—	138	2	—	—	195	4	—	—	84	—	233	4	—	—	—	—	—	—
M2S025 (T/TS)	180	2	138	2	—	—	207	4	—	—	—	—	267	4	—	—	—	—	—	—
M2S050 (T/TS)	200	2	—	—	—	—	207	4	—	—	—	—	267	4	—	—	377	8	—	—
M2S060 (T/TS)	200	2	—	—	—	—	207	4	—	—	—	—	267	4	387	4	—	—	—	—
M2S090 (T/TS)	180	4	—	—	—	—	—	—	—	—	—	—	267	4	425	4	—	—	—	—
M2S150 (T/TS)	—	—	—	—	293	4	—	—	248	4	—	—	—	—	—	—	—	—	574	16

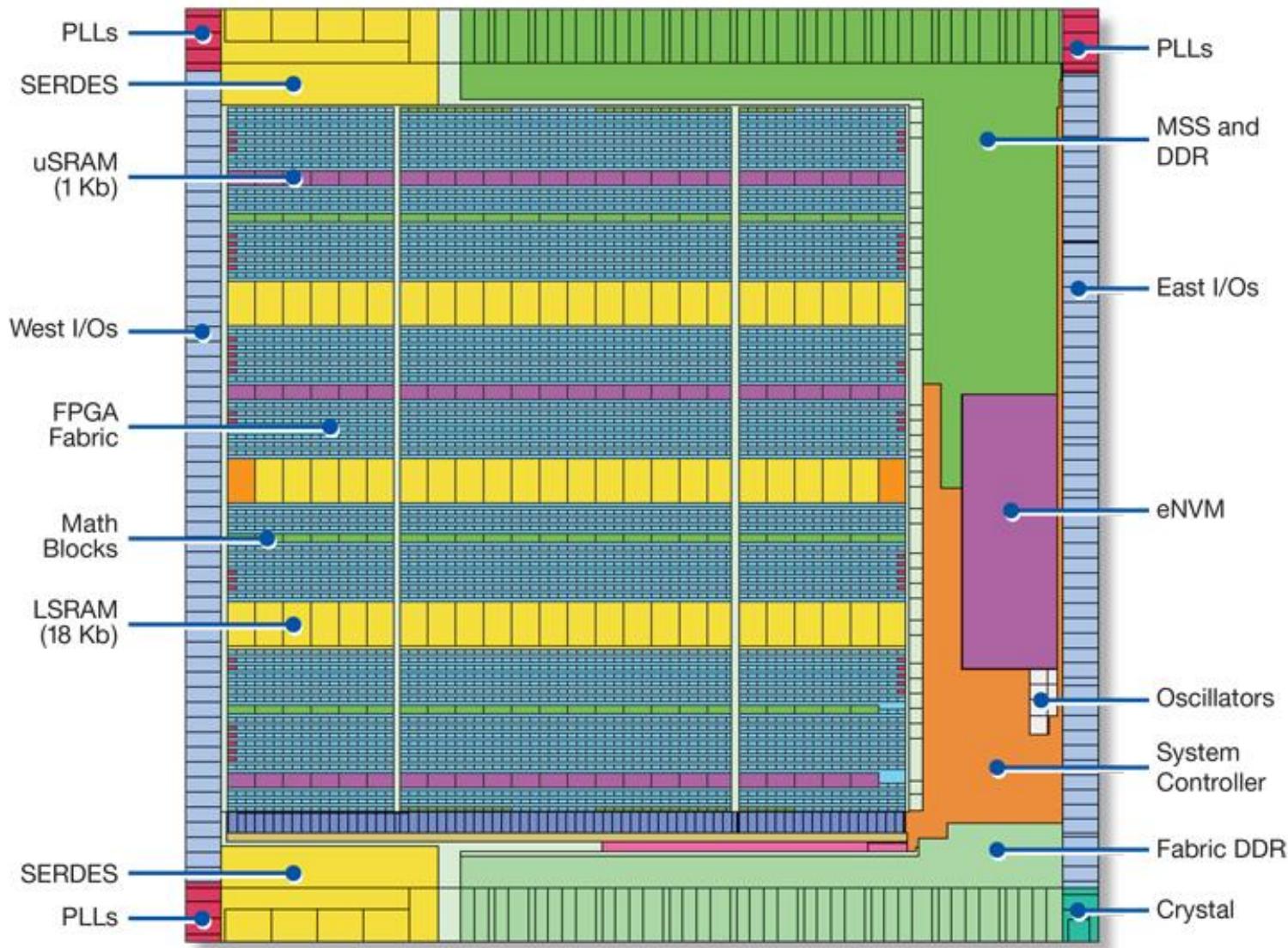
Note:

090 FCSQ325 is 11x13.5 package dimension

■ Highlighted devices can migrate vertically in the same package

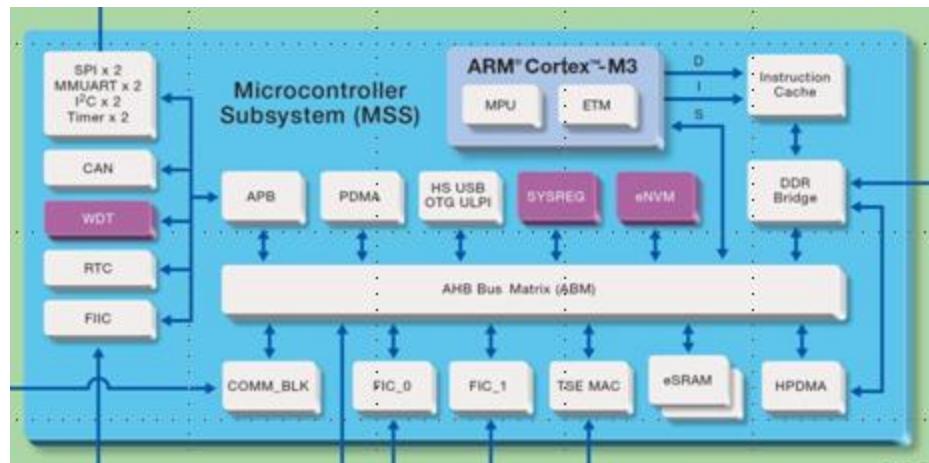
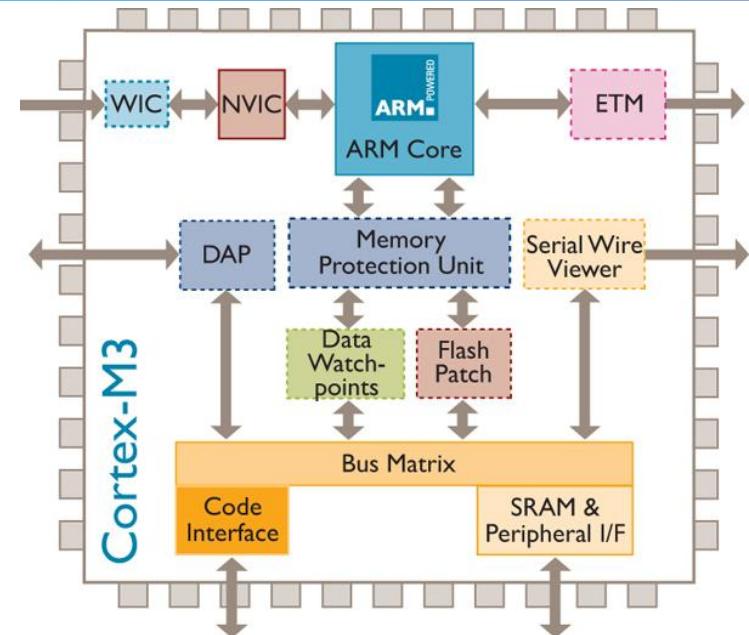
SmartFusion2 Microcontroller Subsystem (MSS)

SmartFusion2 Device Layout



Micro-Controller Subsystem (MSS)

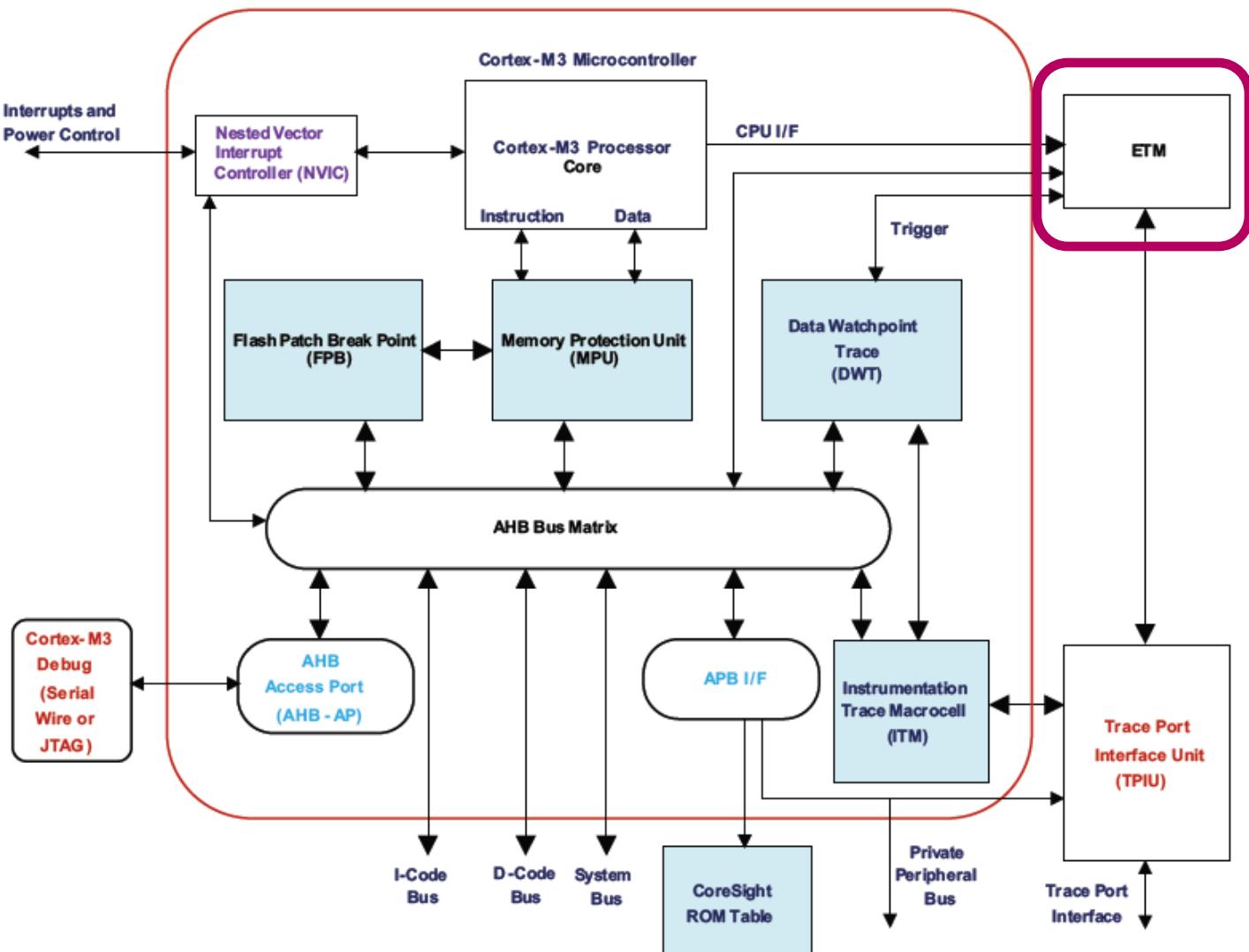
- 166 MHz Cortex M3 with Instruction Cache
- 4 Port DDR bridge for data caching
- High Performance Memory Subsystem
 - All memories SEU tolerant or SECDED
- Multi-layer AHB Bus Matrix with 10 masters and 7 slaves
- Peripherals
 - Triple Speed Ethernet (TSE) 10/100/1000 Mbps MAC
 - USB 2.0 High Speed On-The-Go (OTG) Controller
 - CAN Controller, 2.0B compliant, conforms to ISO11898-1
 - Two each of: SPI, I2C, Multi-Mode UARTs (MMUART)
 - Flash Configured Watchdog Timer
 - 1 General Purpose 64-Bit (or two 32-bit) Timer(s)
 - Real-Time Counter (RTC)
 - 32 GPIOs



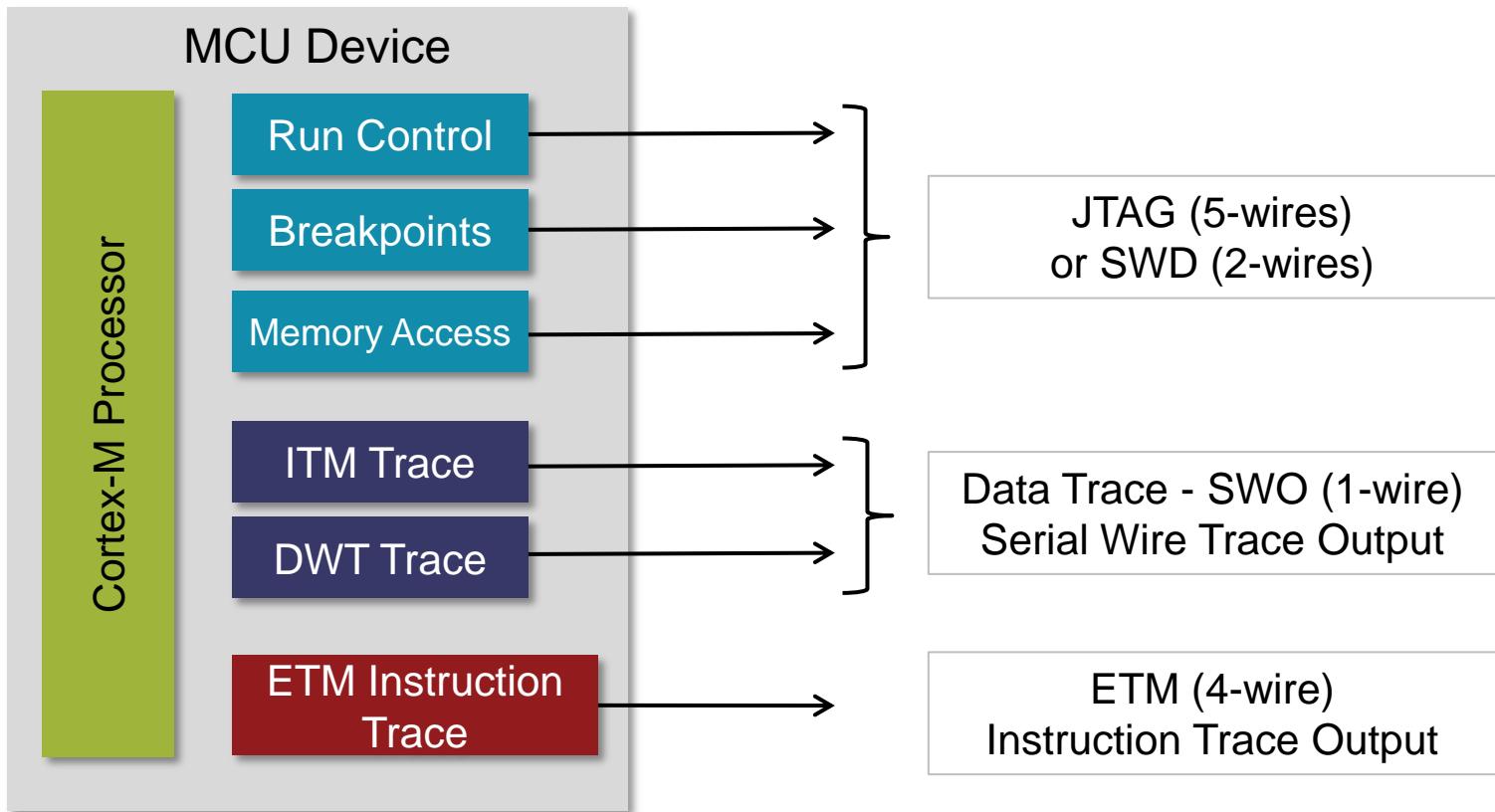
ARM Cortex-M3 Microcontroller

- 166 MHz ARM Cortex-M3
 - 1.25 DMIPS/MHz
 - 8 KB Instruction Cache
 - Memory Protection Unit (MPU)
 - Non-Blocking, Multi-Layer AHB Bus Matrix Allowing Multi-Master Scheme Supporting 10 Masters and 7 Slaves
 - 83 Interrupts (including NMI) with 16 levels of interrupt priority
 - On Chip Memory
 - 64 KB Embedded SRAM (eSRAM)
 - Up to 512 KB Embedded Nonvolatile Memory (eNVM)
 - Debug interfaces
 - JTAG Debug (4 wires)
 - Serial Wire Debug (SWD, 2wires)
 - Serial Wire Viewer (SWV)
 - Embedded Trace Macrocell
- 
- All Cortex-M3 debug interfaces supported!

Cortex M3 R2P1 Block Diagram in SF2



Cortex-M CoreSight Interfaces

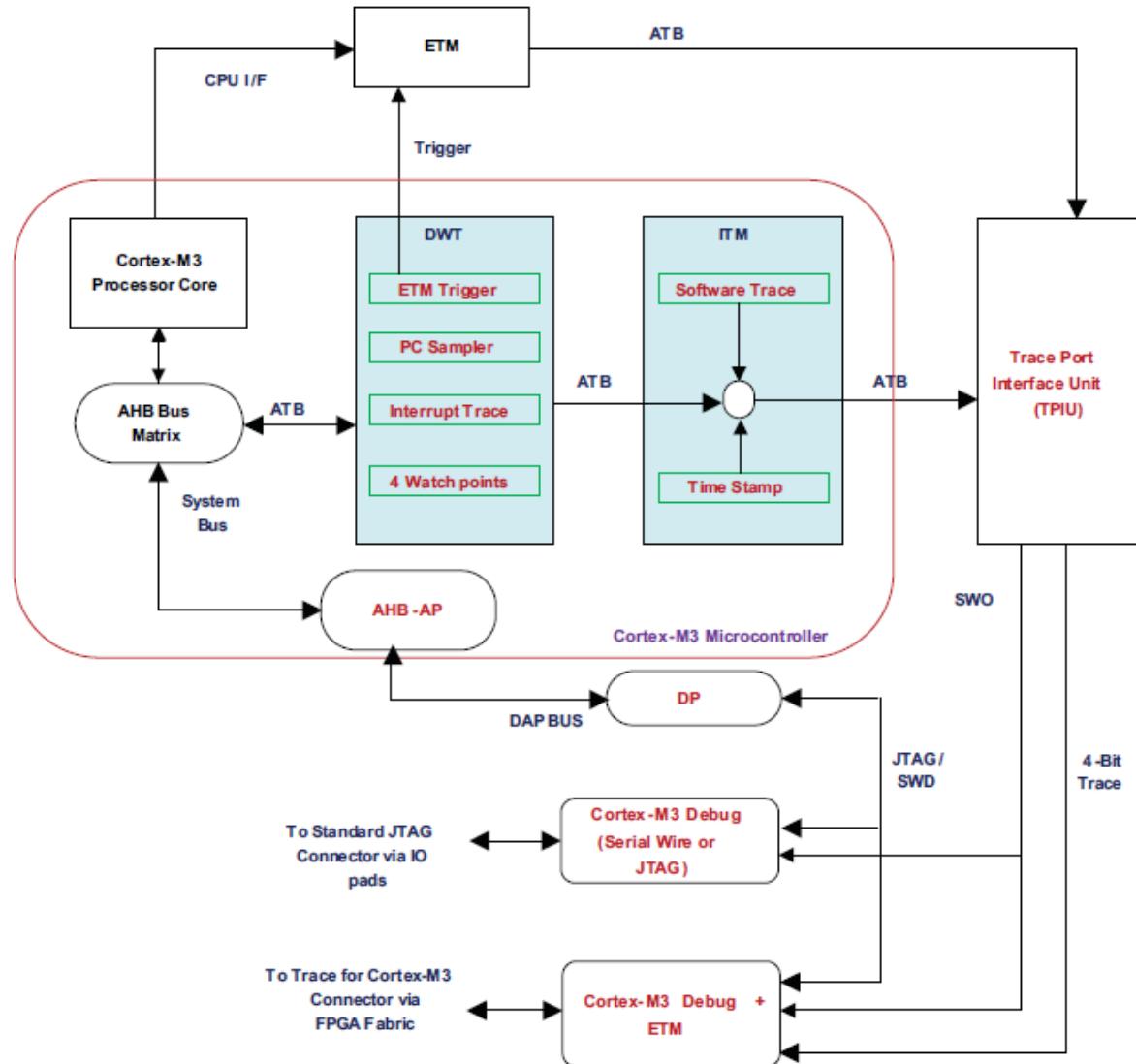


- SWD – Serial Wire Debug
- SWO – Serial Wire Output (data trace)
- ETM – Embedded Trace Macrocell

Cortex-M3 Debug Interfaces

- Embedded Trace Macrocell
 - The embedded trace macrocell provides high bandwidth Instruction Trace via four dedicated trace pins
- Serial Wire JTAG debug port (SWJ-DP)
 - Offers access to the Cortex-M3 CoreSight® debug capabilities
- Serial Wire debug port (SW-DP)
 - The serial wire debug (SWD) mode is an alternative to the standard JTAG interface
 - SWD uses 2-pins to provide the same debug functionality as JTAG with no performance penalty, and introduces data trace capabilities with the Serial Wire Viewer (SWV)
- Serial Wire Viewer (SWV)
 - Provides real-time data trace information from various sources within the Cortex-M3 processor device. This is output via the single SWO pin while your system processor continues running at full speed. SWV can only be used with the SWD Interface

Trace System Block Diagram



Embedded Trace Macrocell (ETM)

- The ETM is an optional debug component that enables reconstruction of program execution
- Features
 - Tracing of 16-bit and 32-bit Thumb instructions
 - Four EmbeddedICE watchpoint inputs
 - A Trace Start/Stop block with EmbeddedICE inputs
 - Two external inputs
 - A 24-byte FIFO queue
 - Global time stamping

Why ETM?

- Non-intrusive Validation of code execution.
 - You could “instrument” your code to facilitate validation. This however can introduce bugs and changes the behavior of the running program.
- Certain market segments require code coverage Validation.
- ETM allows non-intrusive code coverage analysis tools to assist the developer in compliance with standards bodies.
- For example,
 - G-Cover from Green Hills Software
 - G-Cover automates the structural coverage analysis of [DO-178B](#) Table A-7, Objectives 5, 6, and 7 (statement, decision, and MCDC coverage achievement).

Memory and Controllers

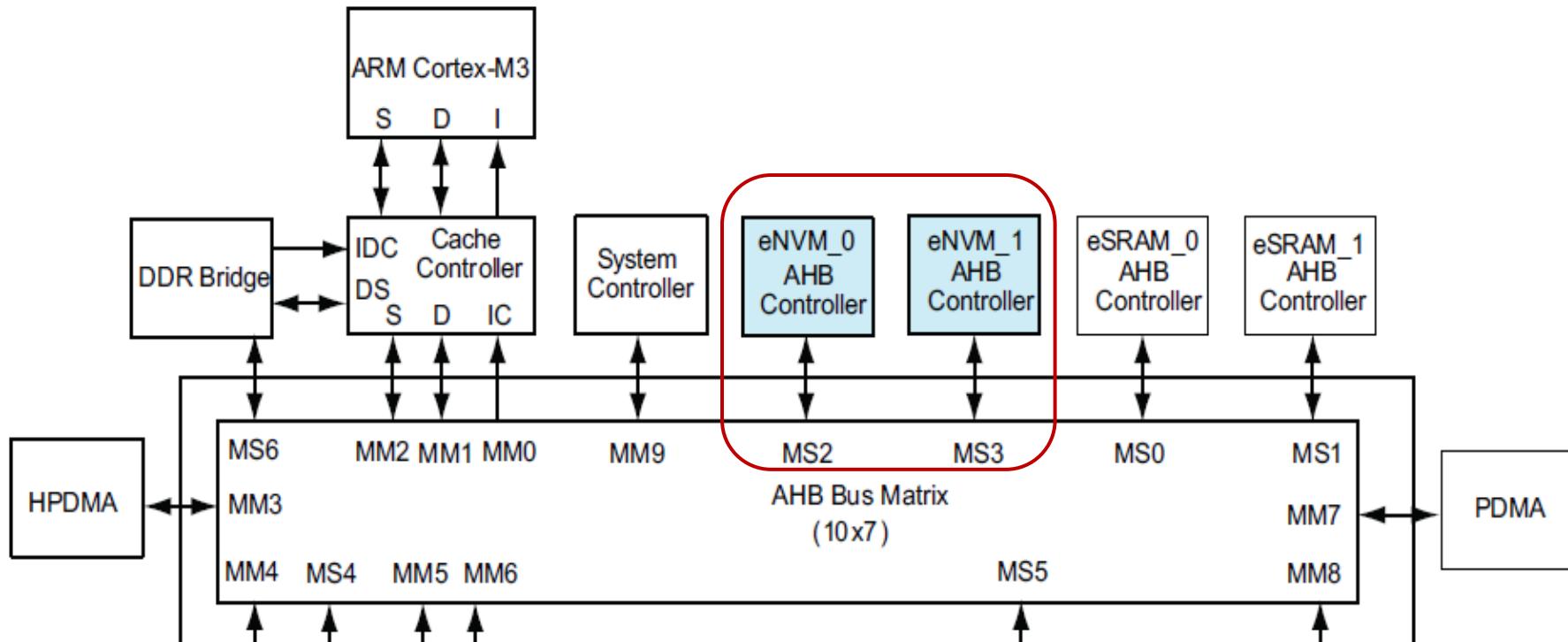
SECDED

- Single Error Correct Double Error Detect
- Blocks covered
 - eSRAM
 - eNVM
 - DDR Controller
 - CAN Message Buffers
 - Ethernet Buffers
 - USB Endpoint Buffers
 - PCIe Buffer
- Status for each block
 - 1 bit correct, 2 bit detect
 - Error signals to NVIC and Fabric Interface
 - Users can decide what to do in SW or HW

All other buffers in ASIC blocks constructed with latches
SPI, MMUART, I Cache, Comm_Blk

Embedded Non-volatile Memory (eNVM)

Embedded Non-Volatile Memory (eNVM)



Two eNVM slaves on the M2S090 and M2S150

eNVM Organization

Device	eNVM size	Number of Sectors	Pages per Sector	Bytes per Page
M2S/M2GL005	128 KB	32	32	128
M2S/M2GL010				
M2S/M2GL025				
M2S/M2GL050	256 KB	64	32	128
M2S/M2GL060				
M2S/M2GL090	512 KB ¹	64	32	128
M2S/M2GL150				

¹Two eNVMs, each 256 KB

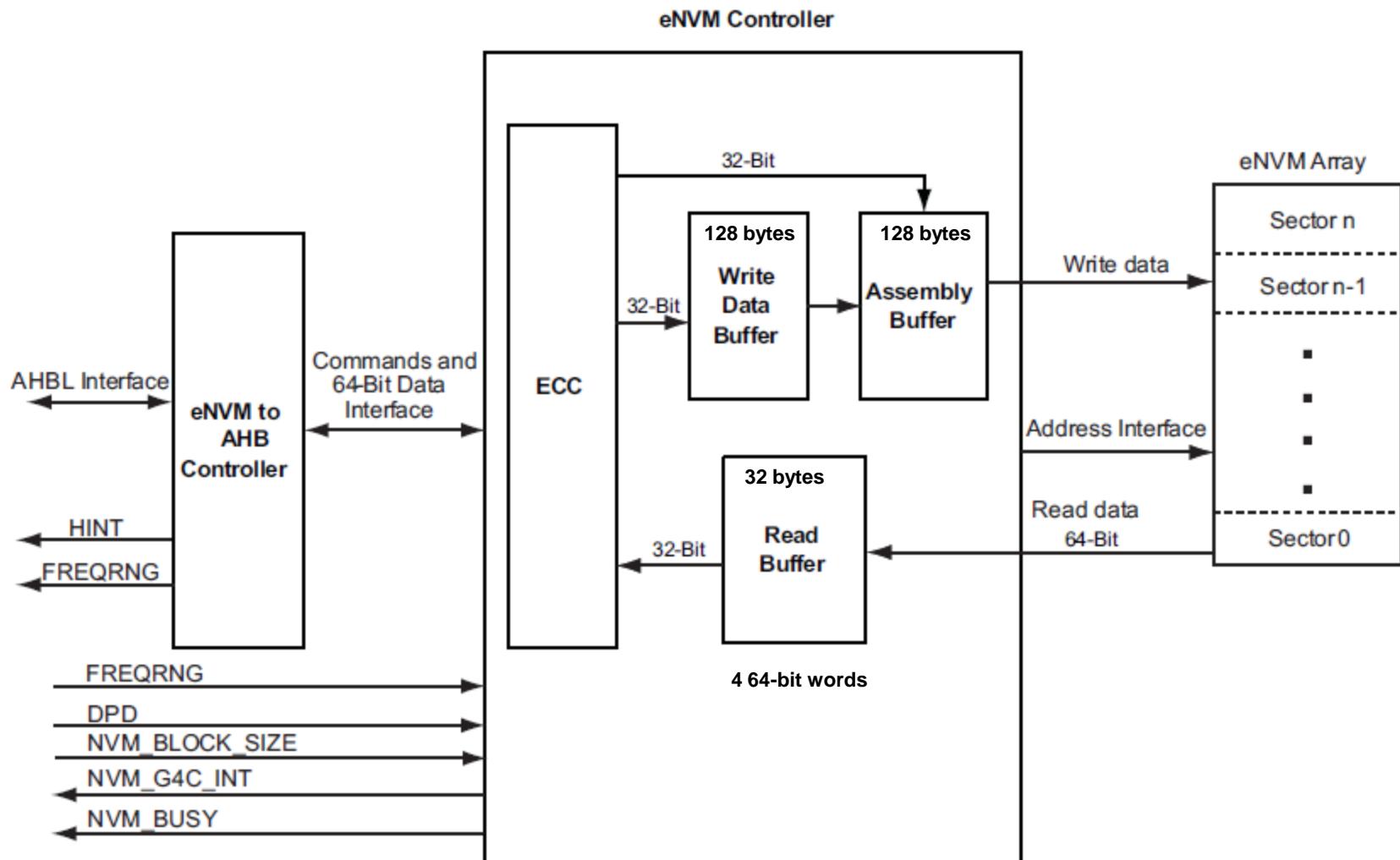
eNVM Addressing

Device	eNVM_0	eNVM_1	Total NVM
M2S005	0x6000_0000	None	128 Kbytes
M2S/M2GL010 M2S/M2GL025 M2S/M2GL050 M2S/M2GL060	0x6000_0000	None	256 Kbytes
M2S/M2GL090 M2G/M2GL150	0x6000_0000	0x6004_0000	512 Kbytes

- Features
 - 2M bit Flash Block, 256K Bytes, 64 bits wide
 - 1M bit (128K bytes) for M2S005
 - SECDED
- Write/Program/Erase granularity: Page
 - 128 byte page
- Erase granularity: Page erase
- Endurance – 10,000+ Page Erase/Program Cycles

Section	Size	
Page	128 bytes	
Sector	32 Pages	4 Kbytes
eNVM Array	64 Sectors	256 Kbytes

eNVM Block Diagram



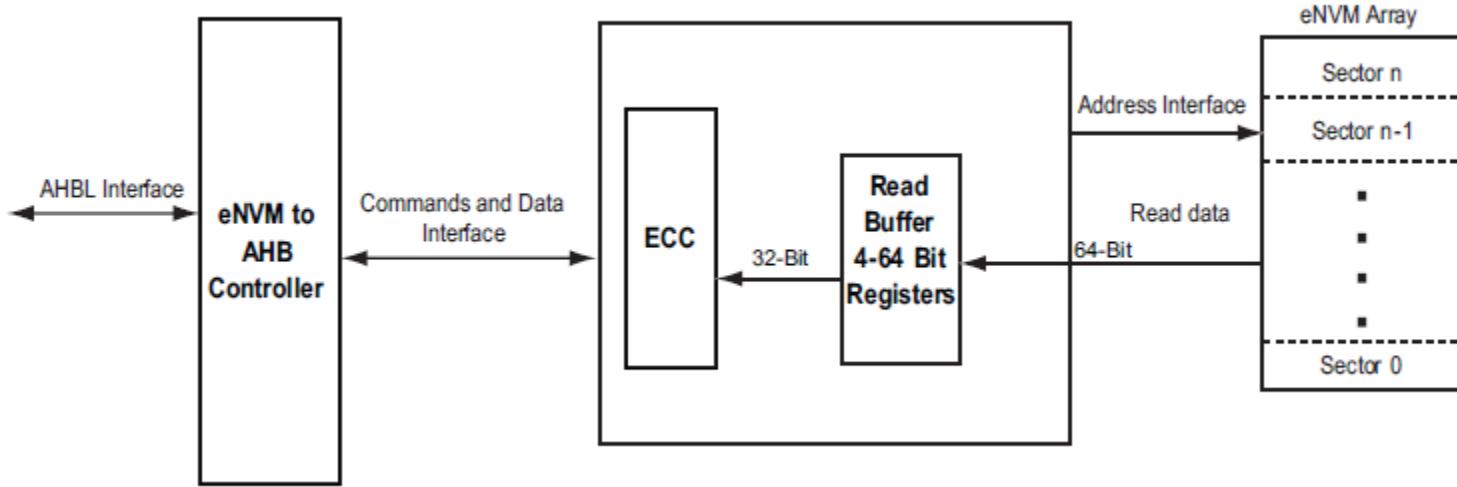
Components

1. eNVM Array
2. eNVM Controller - a wrapper around the eNVM Array
3. AHB to eNVM Controller – a wrapper around the eNVM Controller that interfaces to the switch

eNVM Controller Features

- Interface from AHBL for read/write/erase operations
- Issues all eNVM commands through AHBL read/write bus operations
 - Data width to/from AHBL bus is up to 32 bits
 - Data to/from eNVM is 64 bits
- Assembly buffer (AB) can be read directly from AHBL bus
- AB is loaded from the write data buffer (WDBUFF)
 - Data can be written to WDBUFF in byte, halfword or word AHB transfers
- eNVMs treated as ROM. AHBL write transactions to eNVM user data array will receive error on HRESP and write will be ignored.
- Write AB and Page Program or Page Write commands are used to write the NVM user data array
 - Data for Page Program or Page Write comes from WDBUFF or user data previously written into AB with the Write AB command
- Supports WRAP4 burst operations which are initiated by the cache controller

eNVM Read Path

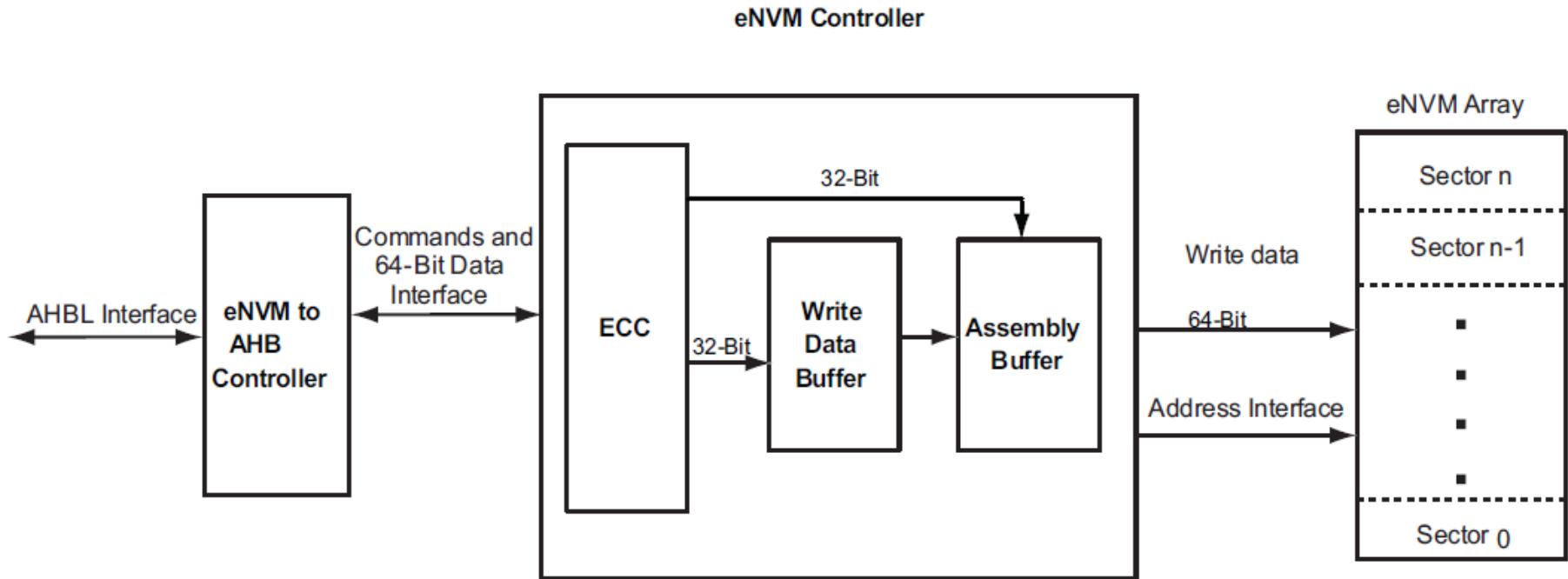


Read of program code is always from the Read Buffer

eNVM Read

- Reading from eNVM is accomplished through a read buffer
- Read Buffer acts like a simple-cache
- Read buffer is 256 Bits
 - 4 - 64 bit flash fetches
 - 8 - 32 bit instructions
 - 16 – 16 bit instructions
- A non-sequential read causes a 4 cycle read from the eNVM into the read buffer
- eNVM is mapped into cache-able region of M3 code space

eNVM Write Path



Data to be written to the eNVM must be written to the AB

eNVM Write/Erase

- eNVM commands via firmware driver:
 - Page (cmd, page number, protect)
 - Where cmd is
 - Erase
 - Program (with erase built in)
 - Number is page number
 - Protect is a boolean
- No Sector or Chip Erase commands exposed to the user

Power Down

- Each eNVM Block can be put into deep power down sleep mode.
 - Controlled by ENVM_CONFIG in SYSREG
 - Can be permanently on or off depending on flash bit(s) settings

eNVM Security

4 levels of security protection

■ All Devices

- eNVM Page Protect
 - Protect against inadvertent writes
 - Two levels: factory lock and user lock
- User defined region inaccessible by FPGA Fabric
 - SYSREG register to control this are of the RO-P type
 - M3 can read the size of the region and the base address of the region but can't change it. These registers are set by User Flash Bits.
 - SW_ERRORSTATUS[9:0] in SYSREG is updated to reflect port activity

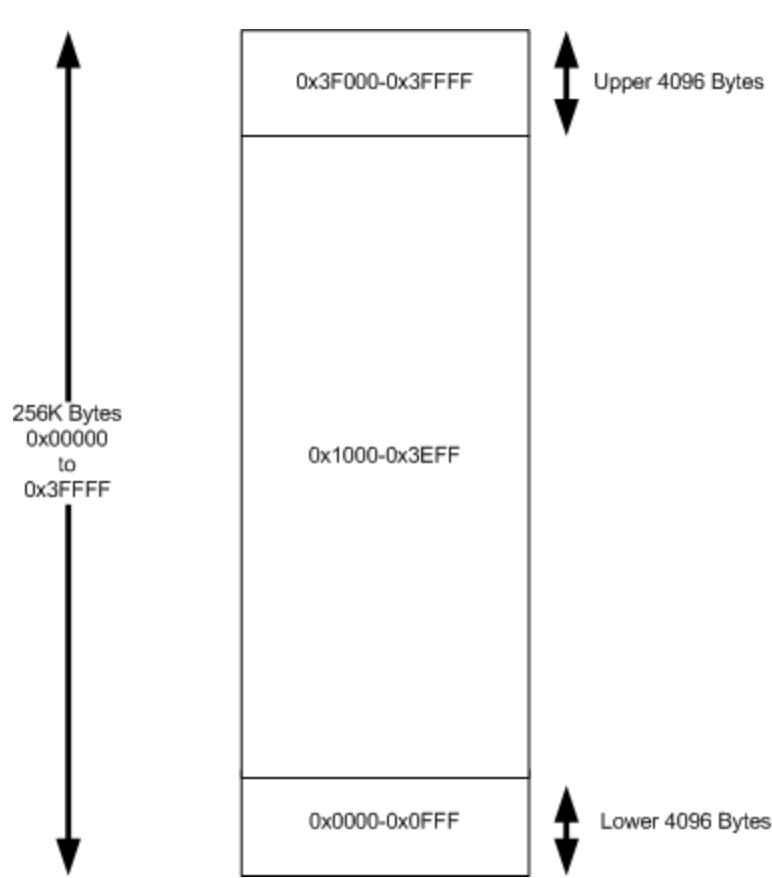
■ 'TS' Devices

- Top/Bottom Sector R/W Access Protection ("eNVM Protect")
 - Two 4 Kbyte regions that can be protected for read and write, depending on which entity is accessing the region
 - User defined as part of overall security architecture
- Switch Master access control
 - The eNVM can be protected from different masters connected on the AHB Bus matrix
 - User-defined as part of overall security architecture

“eNVM Protect” User Flash Bits

- M3 Read Access
 - When set ‘1’ M3 can read from this segment
- Fabric Read Access
 - When set ‘1’ FIC32 - 0 can read from this segment
- Others Read Access
 - When set ‘1’ all other masters can read from this segment
- Write Allowed
 - When set ‘1’ writes are allowed to those masters that have read access
- If M3 Access and Fabric Access and Others Access and Write Allowed == 1 then everyone has full read and write access to that section

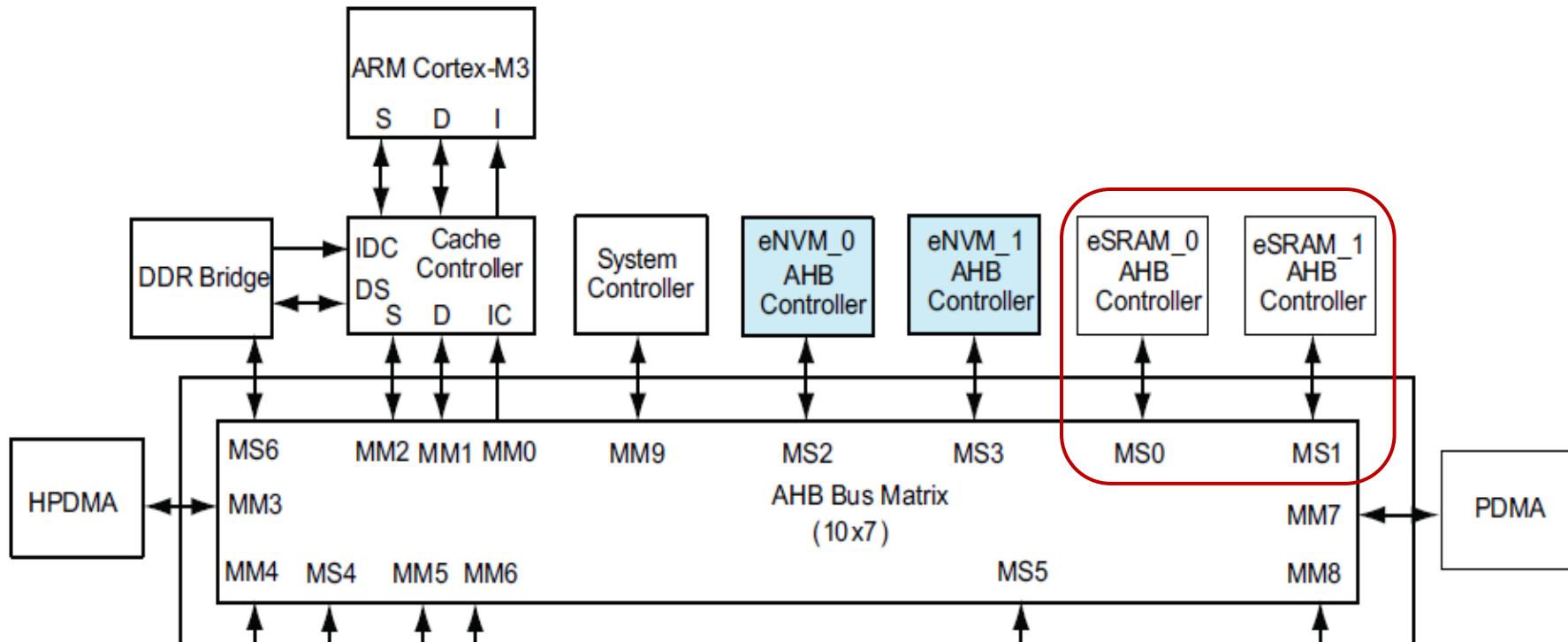
Top/Bottom Sector R/W Access Protection



User Flash Bits		
		M3 Read Access
eNVM0	Lower	Fabric Read Access
		Others Read Access
		Write Allowed
		M3 Read Access
eNVM0	Upper	Fabric Read Access
		Others Read Access
		Write Allowed
		M3 Read Access
eNVM1	Lower	Fabric Read Access
		Others Read Access
		Write Allowed
		M3 Read Access
eNVM1	Upper	Fabric Read Access
		Others Read Access
		Write Allowed
		M3 Read Access

eSRAM

Embedded SRAM (eSRAM)



Two blocks containing two banks each
One 32K Byte bank, second 8K Byte bank

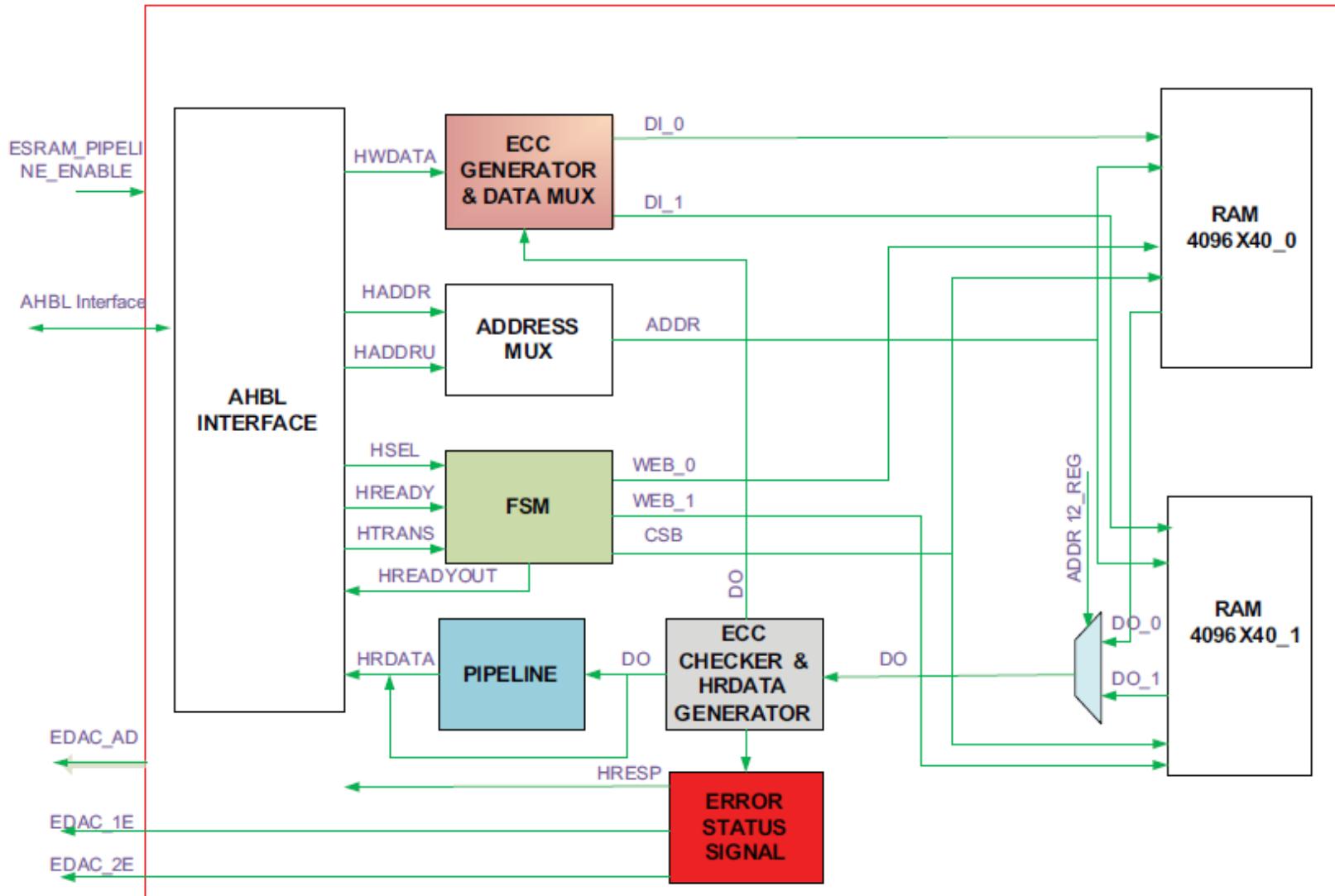
eSRAM Modes of Operation

- Four modes of operation:
 - SECDED-ON with PIPELINE enabled
 - SECDED-ON with PIPELINE disabled
 - SECDED-OFF PIPELINE enabled
 - SECDED-OFF PIPELINE disabled
- SECDED-ON
 - 2-bit error detect, 1-bit error correct
 - eSRAM controller generates 7 check bits for every 32 bits of data
 - Two counters provide Error statistics
 - Total available memory for each eSRAM is 32 Kbytes
- SECDED-OFF
 - Error correction not used
 - Total available memory for each eSRAM is 40 Kbytes
- PIPELINE
 - Must be enabled if FCLK > 100MHz
 - When the FCLK < 100MHz, the PIPELINE may be turned off

eSRAM Addressing

- eSRAM_1: 0x20008000 to 0x2000FFFF
- eSRAM_0: 0x20000000 to 0x20007FFF
- In SECDED-OFF mode two additional 8K memory regions exist
 - ECC eSRAM_1: 0x20012000 to 0x20013FFF
 - ECC eSRAM_0: 0x20010000 to 0x20011FFF
- Both eSRAMs and both eSRAM controllers are identical

eSRAM Controller



Performance 32 KB Block: Non-SECDED Mode

non - SECDED Mode eSRAM Performance
32KB Block

	Byte	Half Word	Word
Write	0	0	0
Read	0	0	0
Read*	1	1	1
Read**	1/2	1/2	1/2

* If FCLK > 100MHz

** If read immediately follows a write
(<100MHz / > 100MHz)

Performance 8 KB Block: Non-SECDED Mode

non - SECDED Mode eSRAM Performance
8KB block

	Byte	Half Word	Word
Write	0	1	3
Read	0	1	3
Read*	1	2	4
Read**	1/2	2/3	4/5

* If FCLK > 100MHz

** If read immediately follows a write
(<100MHz / > 100MHz)

SECDED Operation

- Word writes perform direct 32-data-bit plus 7-bit SECDED writes to the memory
- Byte and half word write operations are done using a read modify write operation.
 - The previous 32-bit data and SECDED value is read and correction will occur, if needed. The complete 32-bits plus SECDED is rewritten.
 - Incurs a wait state
- HRESP will be asserted
 - During a read cycle when a 2-bit error is detected
 - During a byte or half word write operation a 2-bit error is detected during the read part of the read modify write operation.

SECDED Mode Performance

RMW for SECDED
Circuitry

SECDED Mode eSRAM Performance 32KB Block			
	Byte	Half Word	Word
Write	1	1	0
Read	0	0	0
Read*	1	1	1
Read**	1/2	1/2	1/2

* If FCLK > 100MHz

** If read immediately follows a write (<100MHz / > 100MHz)

SECDED Statistics

- 1 bit error counter in SYSREG block
 - Counts the number of 1 bit errors
 - Processor can reset the counter
- 2 bit error counter in SYSREG block
 - Counts the number of 2 bit errors
 - Processor can reset the counter
- Error Address
 - 13 bit address of the last error location
- EDAC error signals can generate a single interrupt
 - Processor polls SYSREG to determine the source
 - All EDAC error signals are encoded and presented to FPGA Fabric for user logic usage

MSS Peripherals

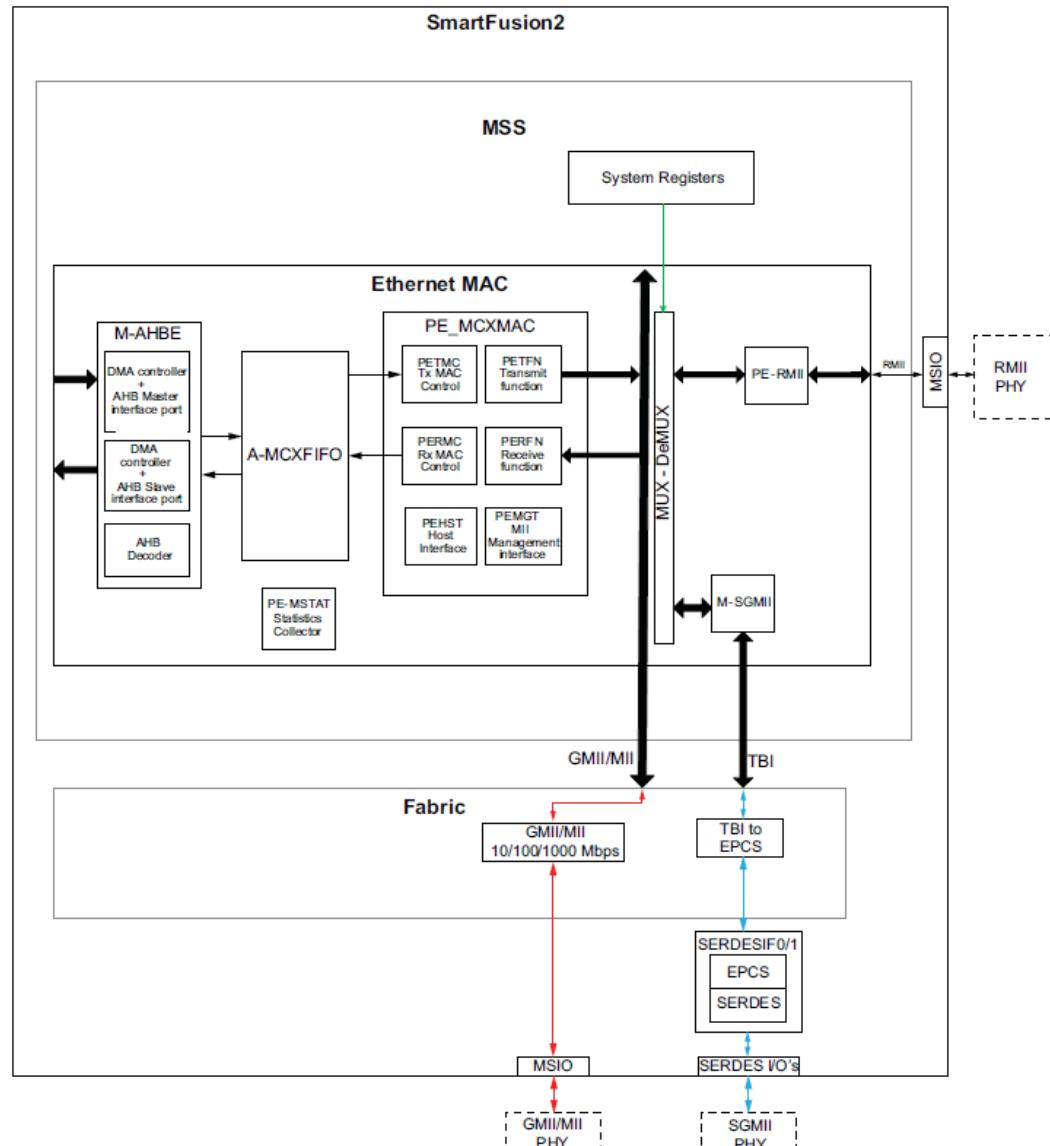
MSS Peripherals

■ Peripherals

- Triple Speed Ethernet (TSE) 10/100/1000 Mbps MAC
- USB 2.0 High Speed On-The-Go (OTG) Controller
- CAN Controller, 2.0B compliant, conforms to ISO11898-1
- Two each of: SPI, I2C, Multi-Mode UARTs (MMUART)
- Flash Configured Watchdog Timer
- 1 General Purpose 64-Bit (or two 32-bit) Timer(s)
- Real-Time Counter (RTC)
- 32 GPIOs

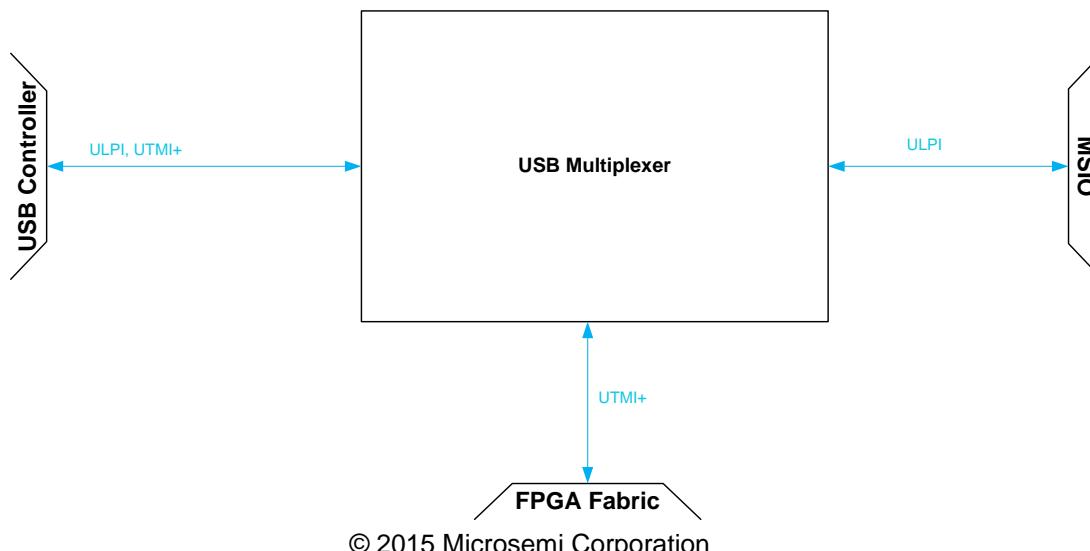
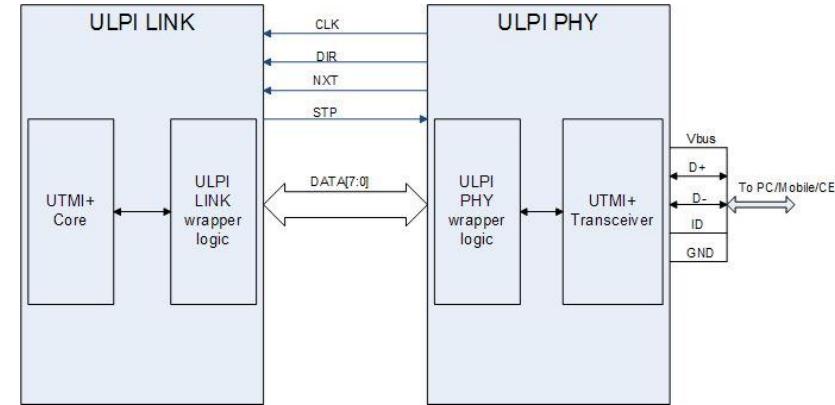
Triple Speed Ethernet

- 10,100,1000
- Built-in DMA, Performance Monitors
- Local Buffering (SECDED)
 - 8K Byte Receive FIFO
 - 4K Byte Transmit FIFO



USB

- High Speed USB OTG
 - Device mode
 - Mass Storage, HID, & CDC
 - Host Mode
 - Mass Storage
- Local buffers SECDED
- ULPI Interface
 - 12 pin LVCMOS interface to external PHY



Control Area Network 2.0 (CAN)

- Full CAN 2.0B compliant licensed from IniCore
- Conforms to ISO 11898-1
- Maximum baud rate of 1Mbps with 8MHz system clock
- **Receive path**
 - 32 receive buffers (SECDED)
 - Each buffer has its own message filter
 - Message filter covers ID, IDE, RTR, Data byte1 & Data byte2
 - Message buffers can be linked together to build a bigger message array
 - Automatic Remote Transmission Request(RTR) response handler with optional generation of RTR interrupt
- **Transmit path**
 - 32 transmit message holding registers with programmable priority arbitration (SECDED)
 - Message abort command
 - Single shot transmission (No automatic re-transmission upon error or arbitration loss)

SPI

- Motorola SPI
- National Microwire
 - SPI generates an 8 bit control word (taken as the first 8 bits of the first transmit FIFO word)
- TI Synchronous Serial
 - SPI generates a pulse on slave select before data transmission
- Support for Large SPI Serial Flash

I2C

- Conforms to the Philips Inter-Integrated Circuit (I2C) v2.1 Specification (7-bit addressing format at 100 kbps and 400 kbps data rates)
- Supports SMBus v2.0 Specification
- Supports PMBus v1.1 Specification
- Multi-master collision detection and arbitration
- Own Slave Address and General Call Address detection
- Second Slave address detection
- SMBus timeout and real-time idle condition counters
- Optional SMBus signals, SMBSUS_N and SMBALERT_N, controllable via APB IF
- 7 Bit addressing

MMUART

- Asynchronous and Synchronous operation
- 5 to 8 bits full duplex operation
- 9th bit Address Flag capability
- 16 byte FIFO
- Half duplex, single wire with error signaling per ISO7816-3, t=0 mode
- LSB or MSB First
- 1, 1.5, or 2 stop bits [1 for all, 1.5 for only 5 bit, 2 for 6-9 bit]
- Even, Odd, Mark, Space or No Parity
- 16x over-sampling receiver frequency
- Transmitter Time Guard
- Receiver Time Out Feature
- FIFO trigger levels of 1/16, 1/4, 1/2, 7/8
- 16550 Error Detection
 - Line Break detection
 - False Start bit detection
 - Framing Error detection
 - Parity Error detection (for Mark and Space too)
- Overrun Error detection
- FIFO Timeout Error detection
- Error detected on byte received, interrupt generated on byte received
- Fractional Baud rate capability
- Return to Zero Inverted (RZI) Mod/Demod
- Parameterize-able majority logic Glitch Filtering
- LIN Header Detection and Auto-Baud Rate Calculation
- Test Modes: Local Loop back, Remote Loop back, Automatic echo

Watchdog Timer

- 32 bit down counter clocked by 25/50 MHz RCOSCCLK
- Guards against system crashes
- Must refreshed at regular intervals by the Cortex-M3 or Fabric Master
- Starts up with 21.48 second timeout
- Generates reset or interrupt
- Watchdog Timer is enabled and configured by SEU immune Flash bits
 - Can only be reset by a Bus Master (Fabric or Cortex-M3)
 - Flash bits are programmed at program time
 - Content of flash bits generated by MSS configurator in Libero
- Watch Dog Timer in SmartFusion2 is hardware based



System Timer

- Two programmable, interrupt-generating, 32-bit decrementing counters
- Counter operation modes:
 - Periodic mode:
 - Counter generates interrupts at constant intervals. On reaching zero, the counter is reloaded and begins counting down again.
 - One-shot mode:
 - Counter generates a single interrupt. On reaching zero, the counter halts until reprogrammed.
- Both counters can be concatenated to implement a 64 bit counter
- Each 32 bit counter in the Timer is clocked with the PCLK input
 - With a PCLK frequency of 100 MHz, the maximum timeout period is approximately 42.9 seconds
 - For same frequency the 64 bit timeout period is 5849 years

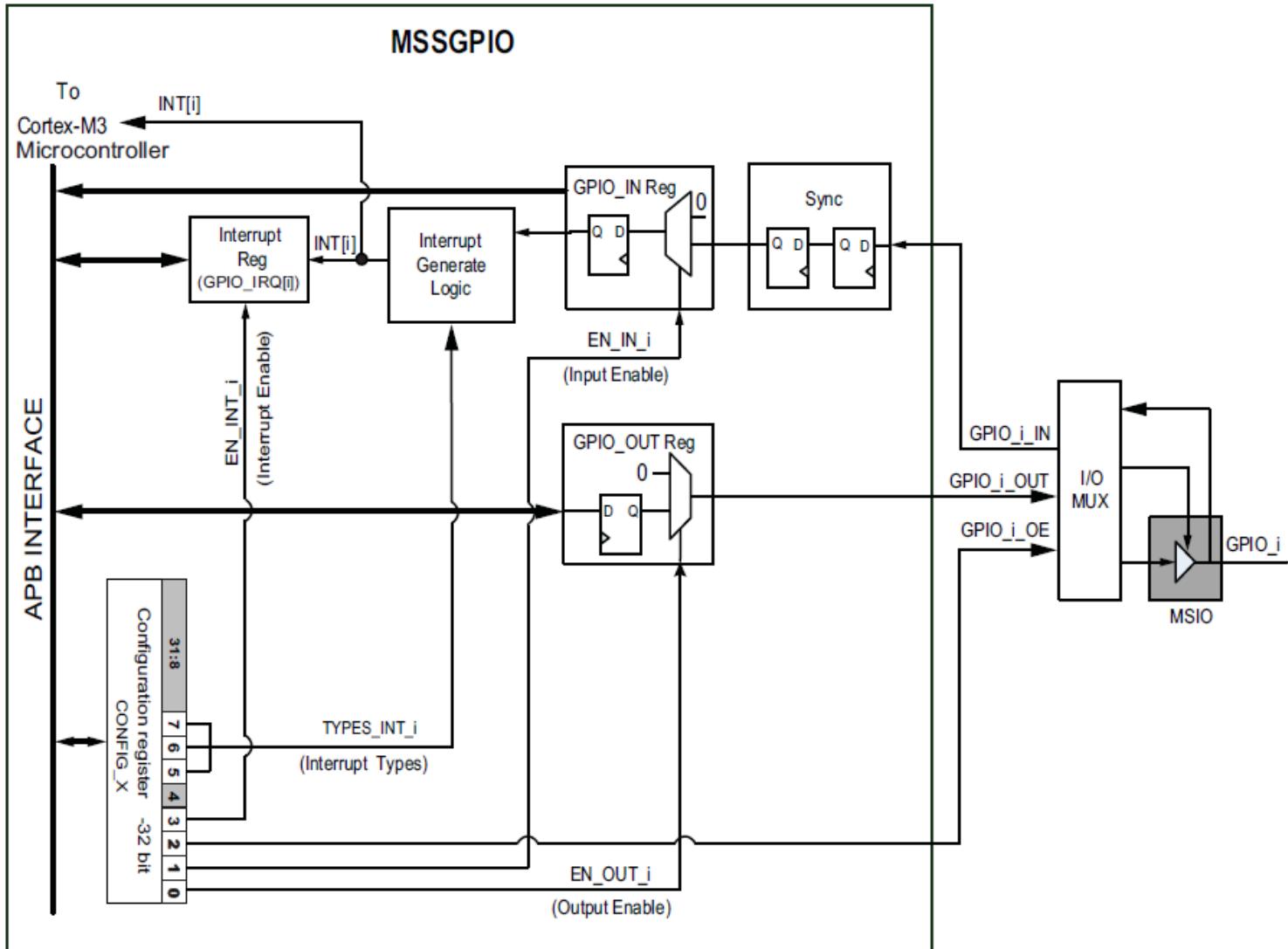
Real Time Counter (RTC)

- RTC system has two main modes of operation:
 - Real-time Calendar: counts seconds, minutes, hours, days, week, months, and years
 - Binary Counter: consecutively counts from 0 all the way to 2^{43}
- RTC Clock Source
 - Crystal Oscillator 32.767 KHz
 - Auxiliary Crystal Oscillator 32.767 KHz
 - 1-MHz Oscillator
 - 25/50 MHz Oscillator
- Programmable 26-bit prescaler used to generate RTC counter clock
- RTC_MATCH and RTC_WAKEUP signals routed to fabric

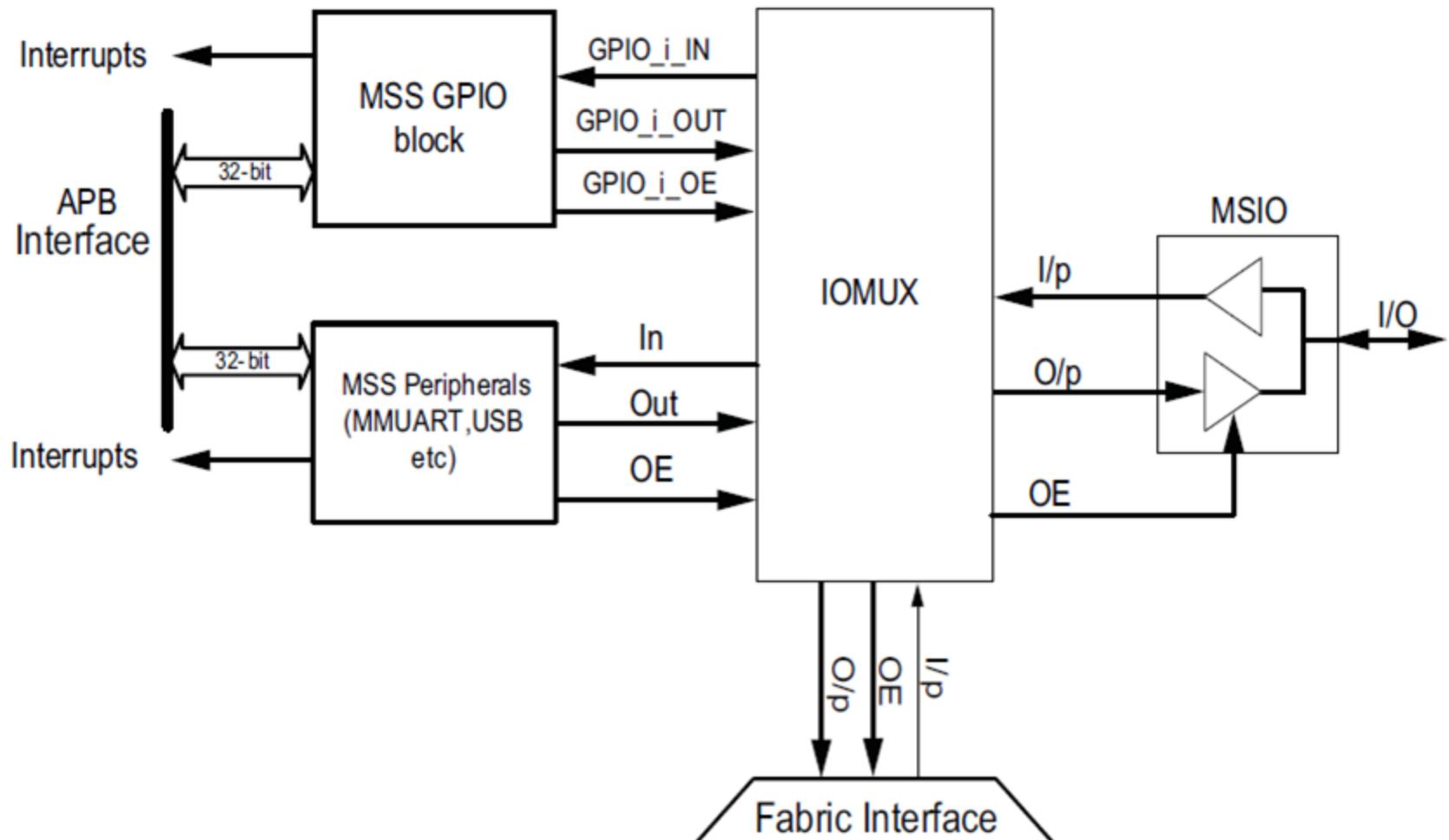
GPIO

- 32 Individually configurable GPIOs
 - Dynamically programmable as input, output or bi-directional
 - Can be configured to generate an interrupt in input mode
 - Reset state of GPIO is configurable
 - Inputs can come from the fabric or an MSIO buffer
 - Outputs can connect to both the fabric and an MSIO buffer

MSS GPIO



MSS GPIO, I/O Mux and MSIO



MSS Interfaces

MSS Interfaces

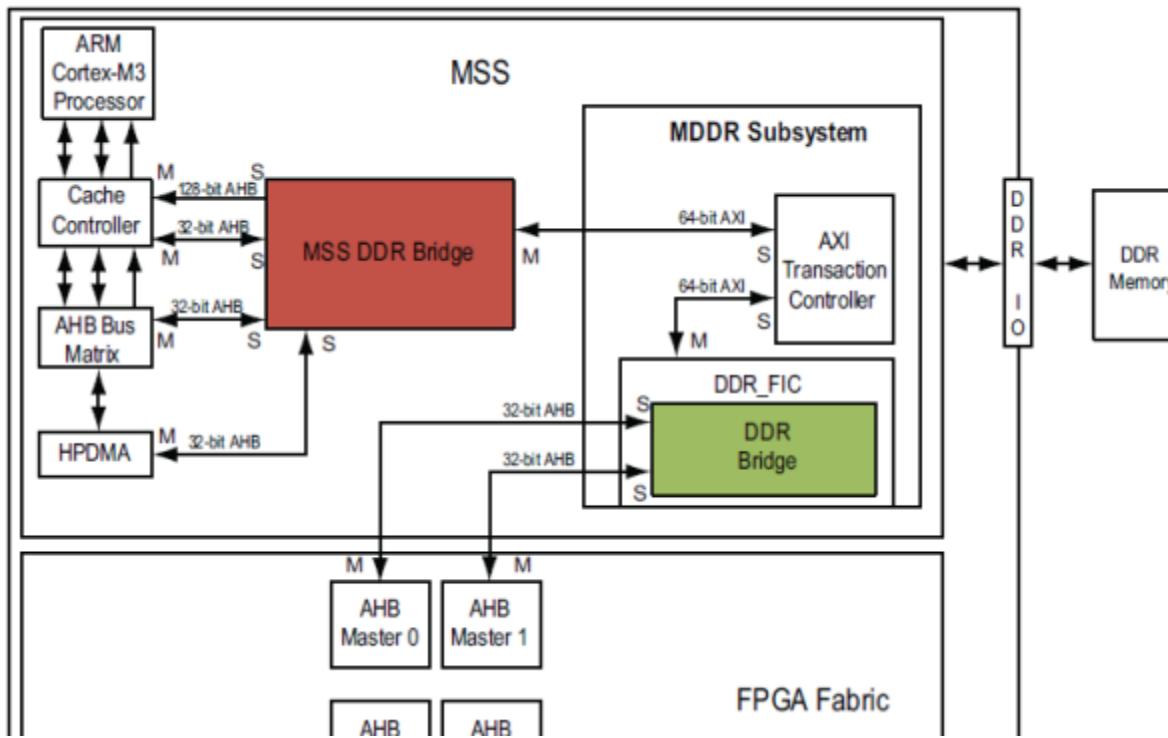
- DDR Bridge
 - High Performance Multi-Master Interface to MSS DDR Controller with 64-Bit AXI Interface
- Two DMA Controllers to Offload Data Transactions from the Cortex-M3 microcontroller
 - 8-Channel Peripheral DMA (PDMA) for Data Transfer Between MSS Peripherals and Memory
 - High Performance DMA (HPDMA) for Data Transfer Between eNVM, eSRAM and DDR Memories
- Two AHB/APB Interfaces to FPGA Fabric (master/slave capable)

Embedded DDR Memory Interfaces

- Up to 2 High Speed 667Mbps DDRx Memory Controllers plus PHY
 - MSS DDR available on all devices
 - Connects to embedded AHB and HPDMA Controller
 - Connects directly to FPGA Fabric
 - Fabric DDR available on 050 and 150
 - Connects directly to FPGA Fabric
- Supports LPDDR/DDR2/DDR3
 - Burst lengths of 2, 4, 8, or 16
- AMBA AXI or AHB interface
- SECDED Supported
- Supports Command and Data Reordering
- Supports Dynamic changing clock frequency in Self-Refresh
- External Memory Configurations
 - Memory densities up to 4 GB
 - Maximum of 8 memory banks
 - 1, 2, or 4 ranks of memory
 - Bus Width Modes of x16, x18, x32, x36

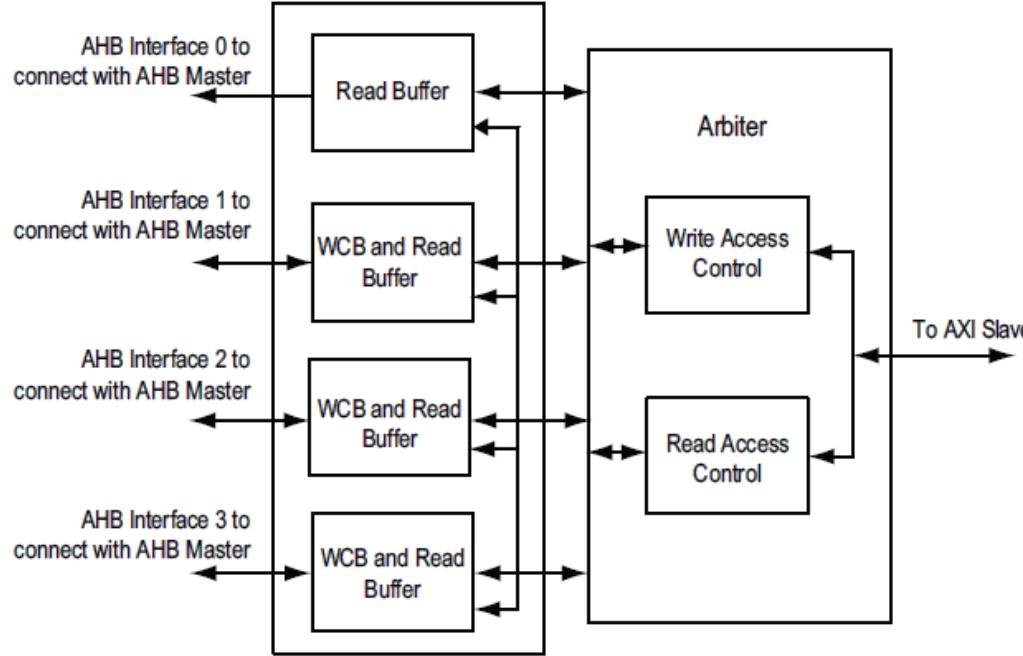


DDR Bridge



- Conceptually it's a 4 port Data Cache

DDR Bridge



- Combines single cycle accesses from MSS and bursts them out to DDR memory
 - Saves power, increases performance

DDR PHY

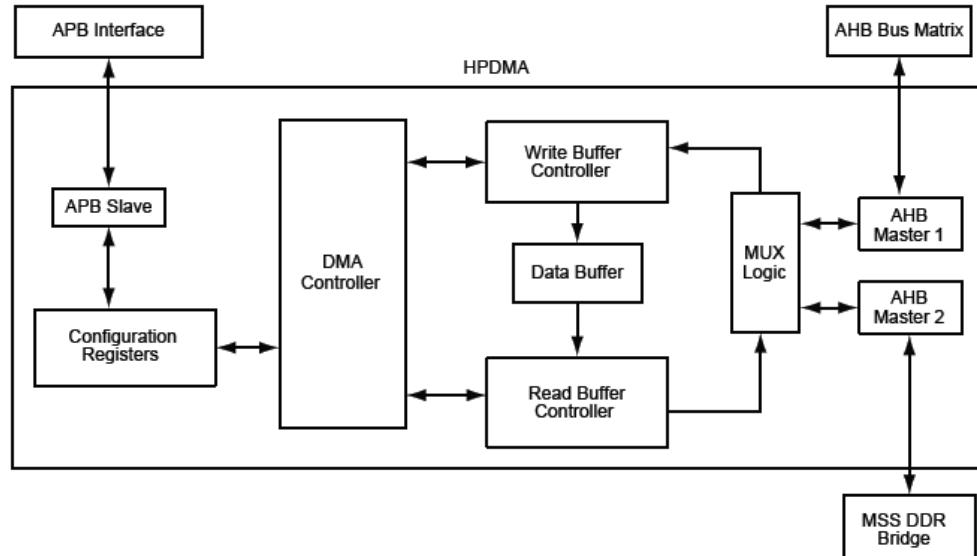
- Consists of 5 Data Slices & 1 Control Slice
- 4 Data Slices transfer the Actual Data and 5th Data slice is reserved for SECDED
 - Each Data slice is 8 bits wide
 - SECDED data is packed on the
 - First 4 data bits of the 5th Data slice in 32-bit width mode
 - First 2 data bits of the 5th Data slice in 16-bit width mode
- Unused slices can be disabled in smaller bus-widths to save power through Register Configuration

High Performance DMA (HPDMA) Controller

- Provides fast data transfer between any MSS memory connected to the AHB bus matrix and MSS DDR bridge
- All transfers by the HPDMA are full word transfers
- The HPDMA controller has two AHB masters which can operate simultaneously
- Can be used in paging access by the processor
 - The main data is stored in a large DDR space and relevant chunks of this data would be transferred as needed via the HPDMA to the eSRAM, where it can be processed faster

HPDMA Features

- Single Channel with 4 descriptor requests
 - Round Robin Priority
 - 32 bit source
 - 32 bit destination
 - Up to 64 Kbytes data transfer in single channel request



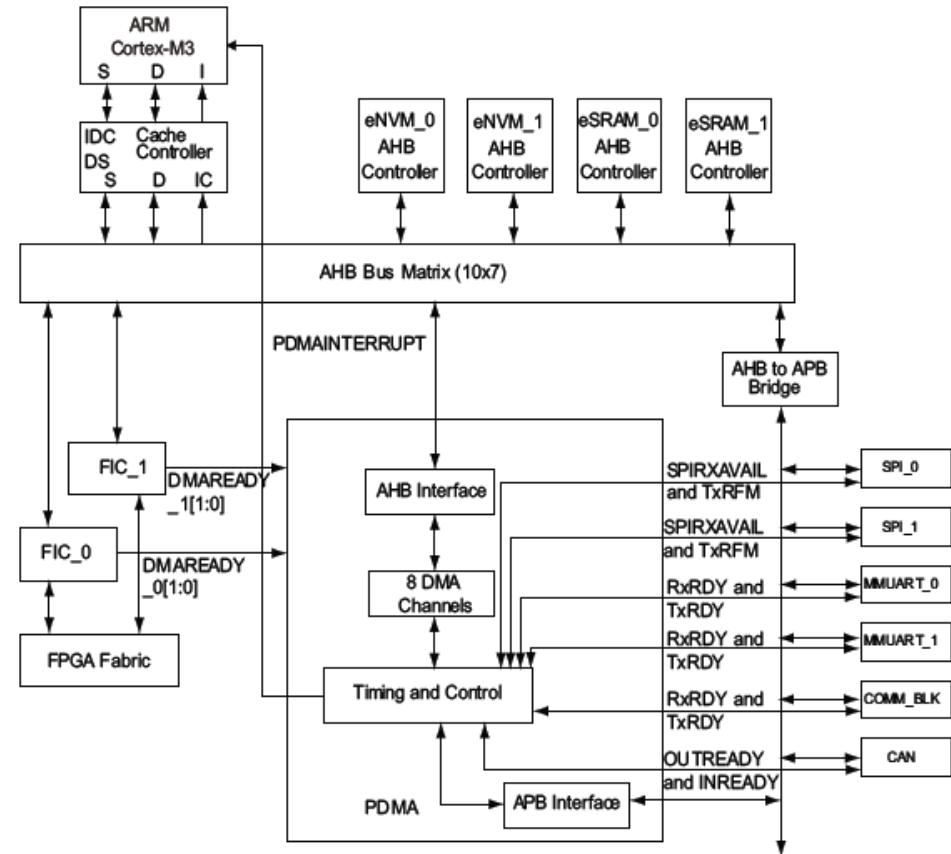
- DMA error and DMA Transfer complete interrupt
- Interrupt Masking

Peripheral DMA (PDMA)

- Implements 8 DMA channels, performs data transfers to/from peripherals, independent of the Cortex-M3 (after initialization)

- 16 available channels

- MMUART 0 tx
- MMUART 0 rx
- MMUART 1 tx
- MMUART 1 rx
- SPI 0 tx
- SPI 0 rx
- SPI 1 tx
- SPI 1 rx
- FPGA tx (FIC_0 or FIC_1)
- FPGA rx (FIC_0 or FIC_1)
- CAN tx
- CAN rx
- COMM_BLK tx
- COMM_BLK rx



- Example of a Channel:

- From MMUART_0 receive to any MSS memory-mapped location

PDMA Features

- Use any of the 8 channels for a memory to memory DMA
- Support for ping-pong mode
 - Continuous real-time DMA of any length with no hiccups
 - Limited by memory depth
- Prioritize channels

Fabric Interfaces

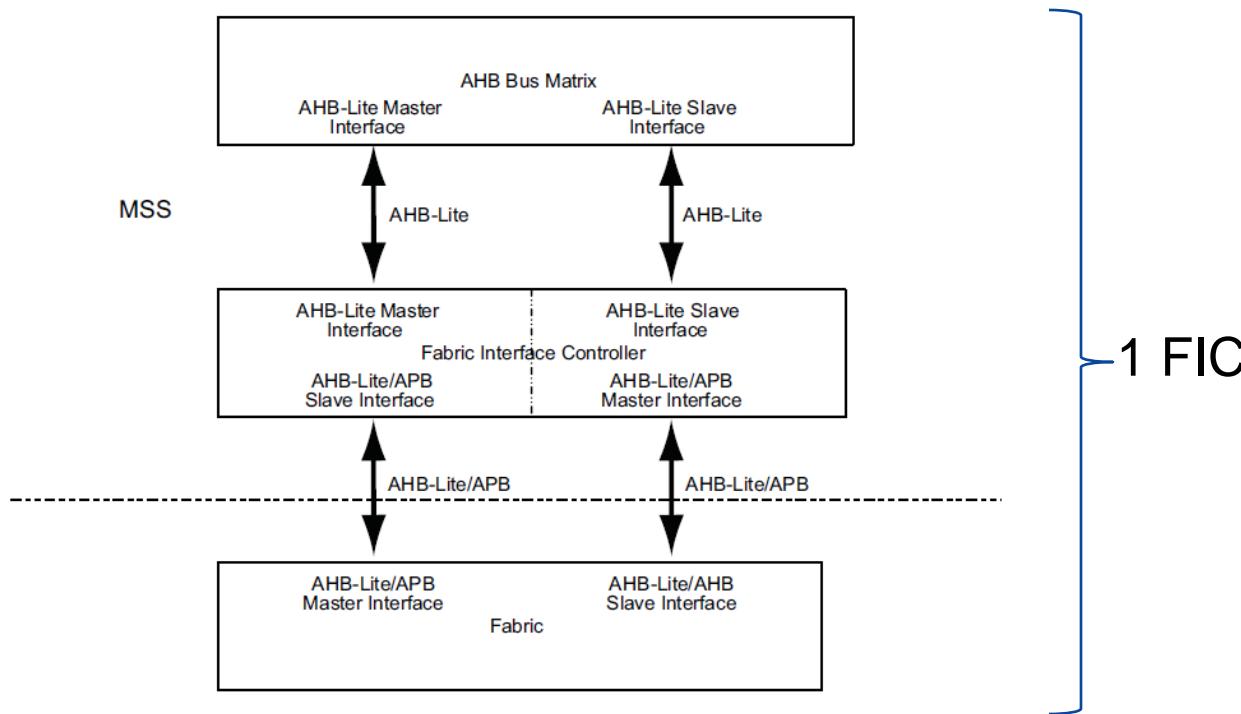
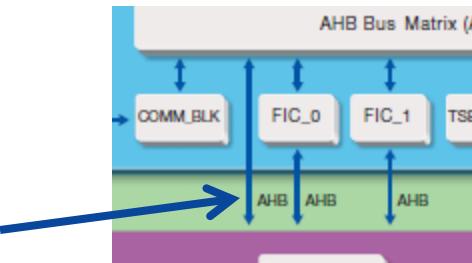
Fabric Interface Controller (FIC_x)

- The fabric interface controller (FIC) enables connectivity between the fabric and microcontroller subsystem (MSS)
 - There are up to two 32-bit FICs in SmartFusion2 - FIC_0 and FIC_1
 - The interfaces to the fabric can be 32-bit AHB-Lite or 32-bit APB
- Available options:
 - Configure the FIC for master interface (MSS master) and slave interface (fabric master)
 - Configure the FIC in Bypass mode or Synchronous Pipelined mode
 - Expose the master identity port to the fabric
 - Configure MSS master view for the FPGA fabric address

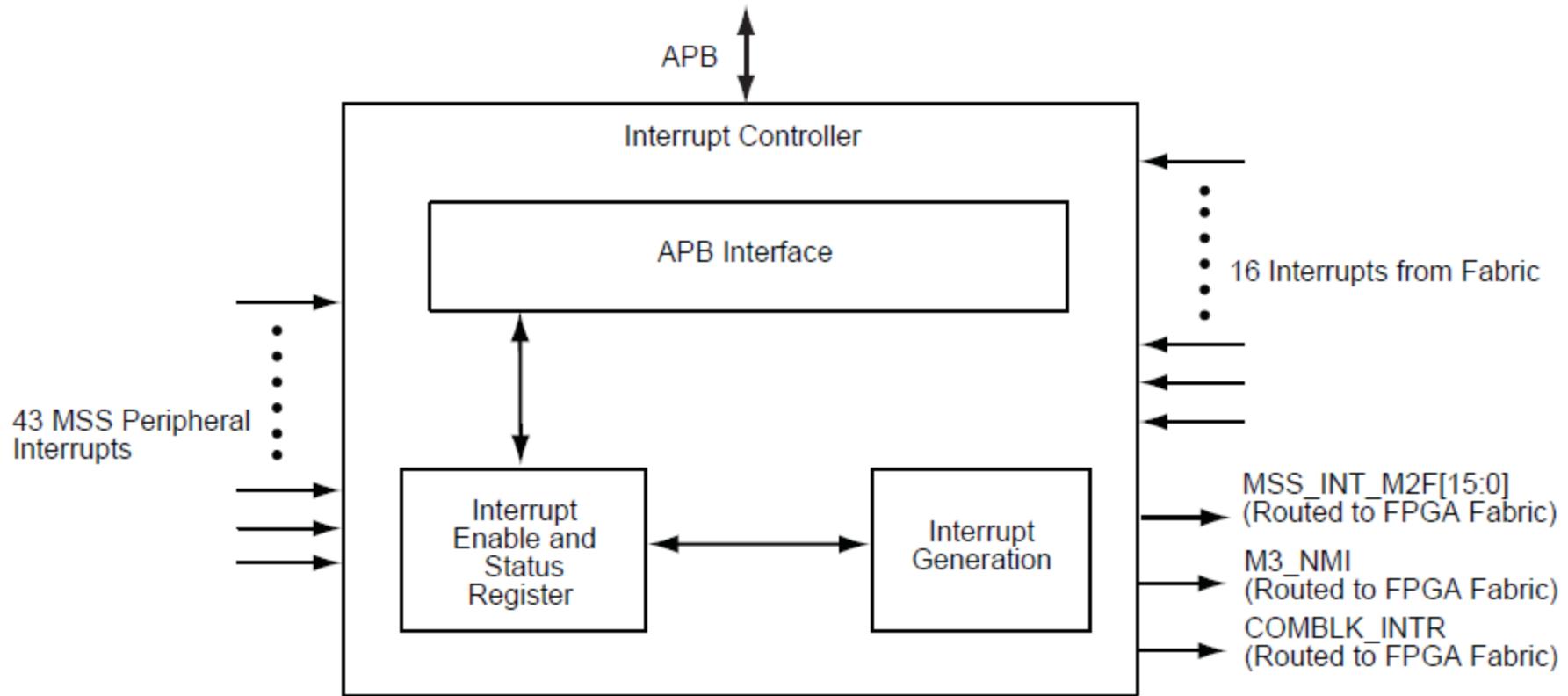
Fabric Interface Controller (FIC)

- 1 FIC on the 005, 010, 025, 060 and 090
- 2 FICs on the 050 and 150

Dedicated APB configuration bus in all devices



Fabric Interface Interrupt Controller



Control and Status to Fabric

- MSS to Fabric

- TXEV
- FPGA_RESET_N
- SPI0_CLK_OUT
- SPI1_CLK_OUT
- EDAC_ERROR (8)
- FACC_PLL_LOCK
- WDOGTIMETIMEOUT
- FIC32_0_MASTER (2)
- FIC32_1_MASTER (2)
- RTC_MATCH
- SLEEPING
- SLEEPDEEP
- SLEEPHOLDACK
- COMMS_INT

Control and Status from Fabric

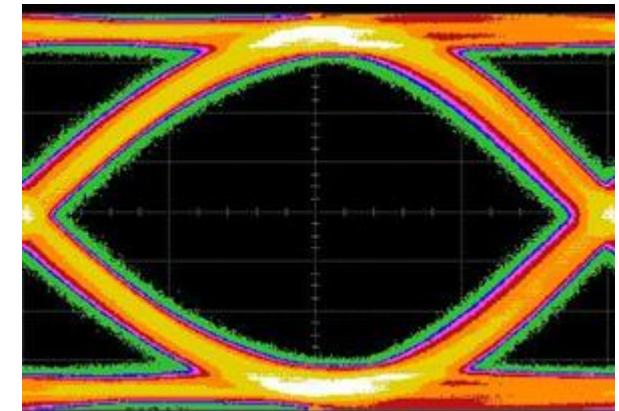
- Fabric to MSS

- RX_EV
- USER_MSS_RESET_N
- F_DMAREADY (2)
- F2_DMAREADY (2)
- USER_MSS_GPIO_RESET_N
- SPI0_CLK_IN
- SPI1_CLK_IN
- CLK_BASE
- FAB_M3_RESET_N
- FAB_PLL_LOCK
- SLEEPHOLDREQ

SERDES

Multi Protocol 5Gb/s SERDES

- Physical Media Attachment (PMA) Features
 - Up to 16 lanes at up to 5Gbps
 - Dual based reference clocks with single-lane rate granularity
 - Tx and Rx PLLs programmable for each lane
 - Reference clock is shared per groups of two lanes
 - Transmitter Features
 - Programmable Pre/Post-Emphasis
 - Programmable Impedance
 - Programmable Amplitude
 - Receiver Features
 - Programmable Termination
 - Programmable Linear Equalization
 - Built-In System Debug Features
 - PRBS Gen/Chk
 - Constant Patterns
 - Loopbacks

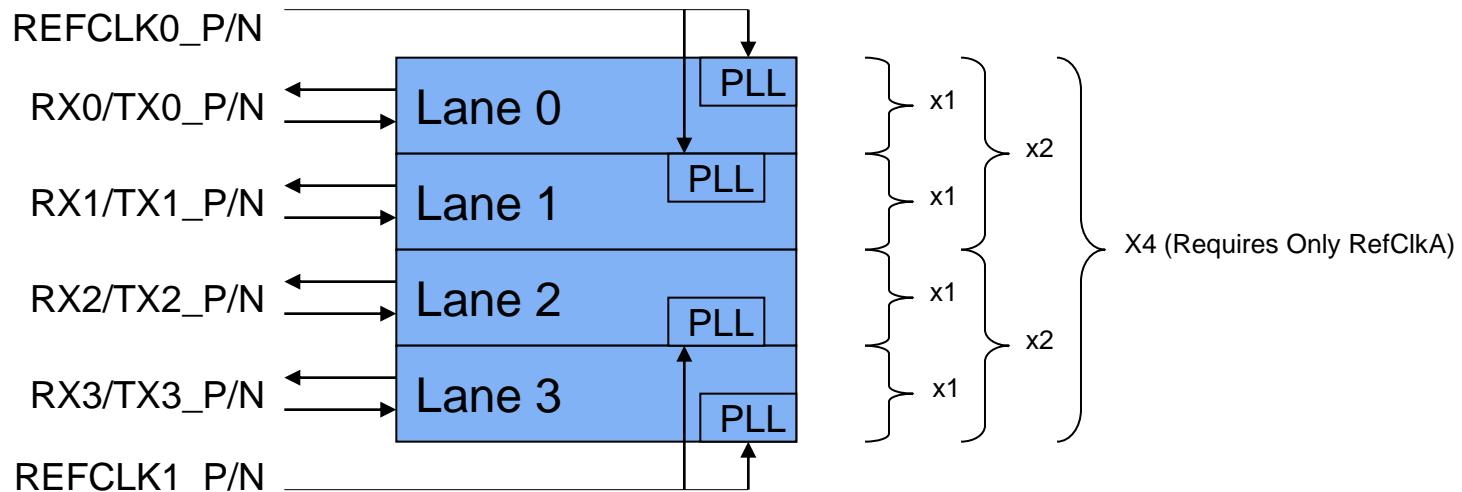


SmartFusion2 Eye Diagram

SERDES Rate Granularity

Dual based reference clocks with single-lane rate granularity

- Tx and Rx PLLs programmable for each channel
- Reference clock is shared per groups of two lanes



Example Scenario

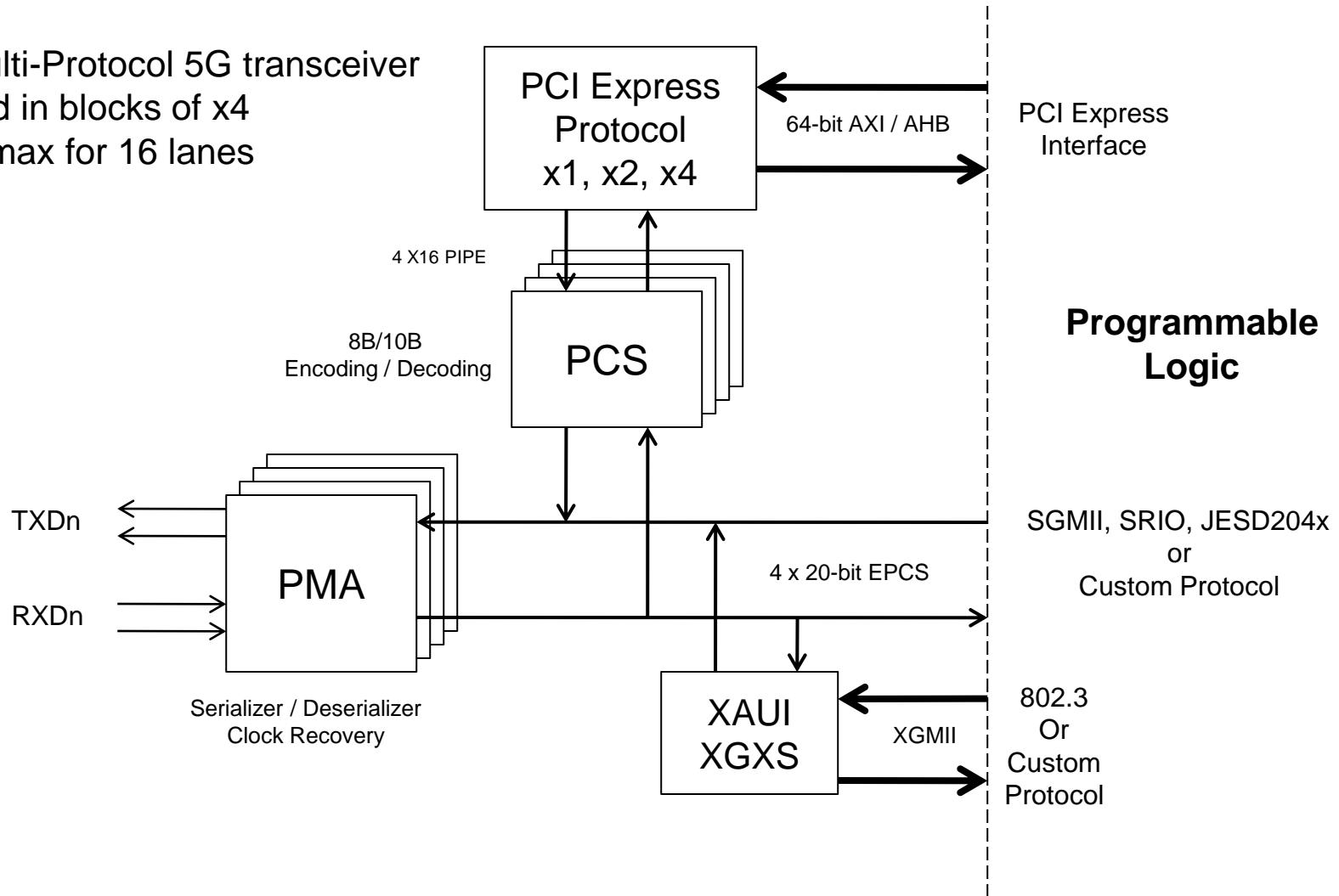
REFCLK0 at 100MHz
Lane0 at 5.0Gbps
Lane1 at 5.0Gbps

REFCLK1 at 125MHz
Lane2 at 2.5Gbps
Lane3 at 3.125Gbps

- Special connectivity for x4 usage using single reference clock.

SERDES / Hard IP architecture

PMA - Multi-Protocol 5G transceiver
Organized in blocks of x4
4 blocks max for 16 lanes



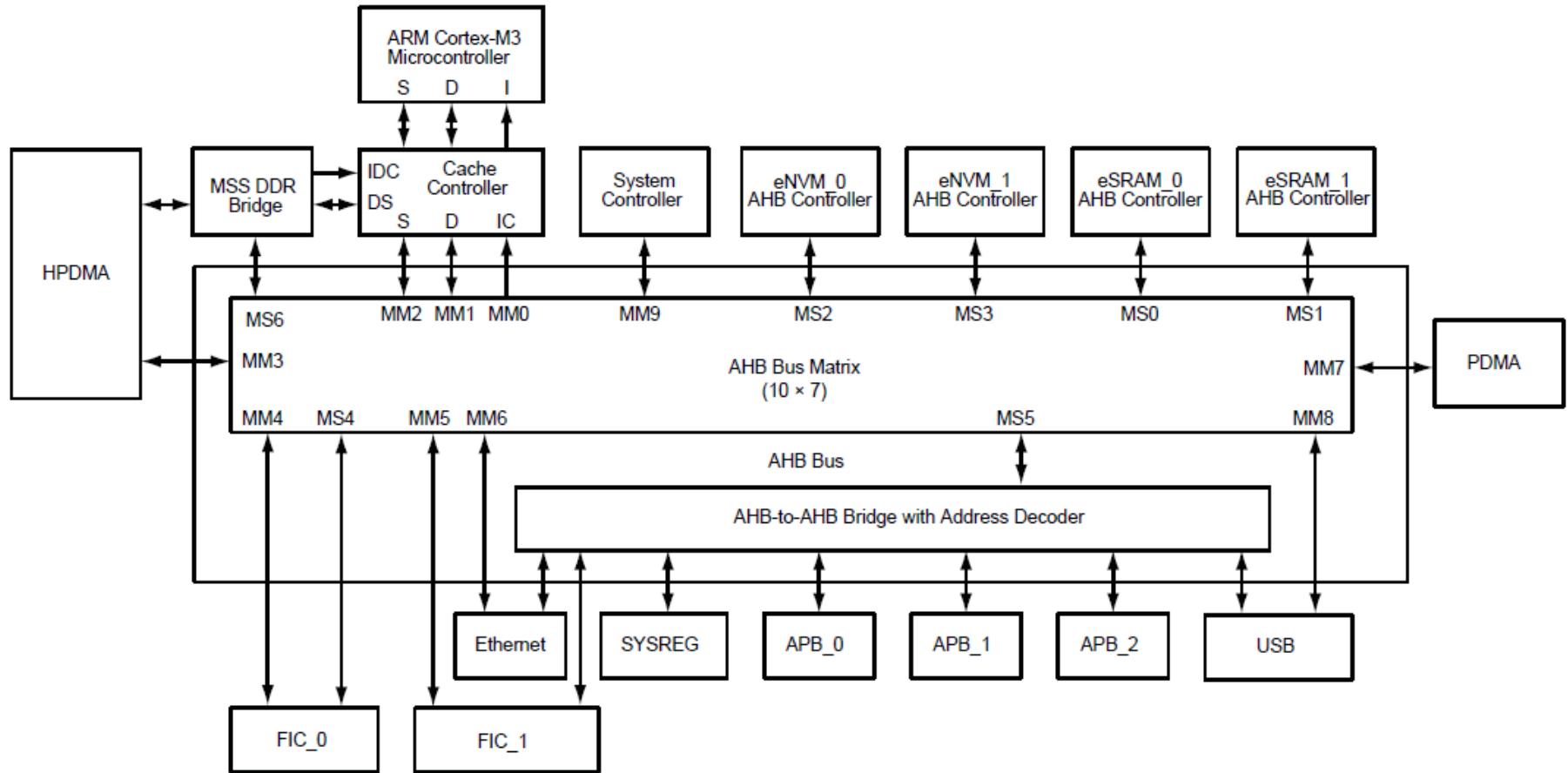
Hard IP - SERDES-Based Protocol Features

- PCI Express Gen1/Gen2 Controller
 - Single-Function Endpoint Configuration
 - x1, x2, x4 Link Widths
 - Static Lane Reversal support
 - 256 Bytes Maximum Payload Size
 - 64/32-Bit AXI/AHB Master/Slave Interfaces to FPGA
 - APB Interface for control/status
- XAUI/XGXS Physical Layer
 - Full compliance with IEEE 802.3
 - 64-Bit XGMII to FPGA at 156.25 MHz
 - MDIO Interface for status/control (802.3ae-clause 45)



AHB Bus Matrix

AHB Bus Matrix



AHB Bus Matrix Overview

- Multi-layer AHB matrix with 10 masters and 7 slaves
 - Customized subset of a full cross-point switch
- Significant Enhancement over SmartFusion
 - Arbitration Scheme
 - Fixed priority masters
 - WRR masters
 - User Programmable arbitration parameters
 - Support for burst transactions to eNVM
 - More layers, faster layers
 - More parallelism, 10 masters, 32 bits, 166MHz = 53 Gbps on Switch bus bandwidth
 - Security

AHB Bus Matrix Features

- Burst transaction support for AHB bursts from an ICode (IC) bus master accessing eNVM slaves
 - Other burst transactions converted to single cycle of type NONSEQ
 - Simplifies design of fabric slaves; fabric master can't dominate a slave
- Arbitration done at slave
 - Fixed Priority + Round robin – default
 - Fixed Priority + Weighted round robin option
- Locked transactions
 - Supports locked transactions from the M3 and System Controller to fabric peripherals
 - Arbiter indicates to the matrix that a locked transaction is occurring and another master can't get access to the slave
- Un-implemented Address
 - Returns Error and Generates an interrupt

Switch connectivity

Masters		Priority	Priority Type	Slaves													
				MS0	MS1	MS2	MS3	MS4	MS5							MS6	
				eSRAM0	eSRAM1	eNVM0	eNVM1	FIC32_0	MAC	FIC32_1	SYSREG	APB_0	APB_1	APB_2	USB	DDR Bridge	
M3 D-Code Bus	MM1	1	Fixed	RW	RW	RW*	RW*	-	-	-	-	-	-	-	-	-	-
M3 I-Code Bus, Cache	MM0	2	Fixed	RW	RW	R*	R*	-	-	-	-	-	-	-	-	-	-
M3 System Bus	MM2	3	Fixed	RW	RW	RW*	RW*	RW	RW	RW	RW	RW	RW	RW	RW	RW	
G Bus	MM9	4	Fixed	RW	RW	RW*	RW*	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
HPDMA	MM3	4	WRR	RW	RW	R*	R*	RW	-	RW	-	-	-	-	-	-	-
FIC32_0	MM4	4	WRR	RW	RW	RW*	RW*	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
FIC32_1	MM5	4	WRR	RW	RW	RW*	RW*	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
MAC	MM6	4	WRR	RW	RW	-	-	RW	-	RW	-	-	-	-	-	-	RW
PDMA	MM7	4	WRR	RW	RW	RW*	RW*	RW	-	RW	-	RW	RW	-	-	-	RW
USB	MM8	4	WRR	RW	RW	-	-	RW		RW	-	-	-	-	-	-	RW

* No hardware mechanisms to determine ownership of a previous write

Reads or writes to areas not allowed are consumed by the switch and HRESP is asserted. Really only applicable to the user when fabric logic is used to address the MSS.

Locked Transaction Support

- AHB bus matrix supports implementation of locked transactions for accesses by the Cortex-M3 to the memory controllers (eNVM AHB controller, and eSRAM AHB controller, by monitoring HMASTLOCK)
- The slaves to which HMASTLOCK is actually passed is the fabric slaves 0 and 1 and eNVM0 and 1; a circuit within the FPGA fabric may need to perform further locking
 - Use this signal if you want to perform a RMW on a memory location in a multi-master bus switch
 - Lock other masters out while the multi-cycle transaction occurs

Slave Arbitration

- Each Slave contains its own arbiter
- Fixed Priority Masters
 - Cortex-M3 I, D, S busses
 - System Controller
- Weighted Round Robin Masters
 - FIC32_0
 - FIC32_1
 - HPDMA
 - MAC
 - PDMA
 - USB
 - All have the same priority as the fixed bus System Controller Master

ESRAM_MAX_LAT

- ESRAM_MAX_LAT determines the peak wait time for a Fixed Priority master arbitrating for eSRAM access while the WRR master is currently accessing the same slave
 - ESRAM_MAX_LAT can be configurable from 1 to 8
 - Default = 8
- When ESRAM_MAX_LAT is set for the eSRAM, a WRR master can perform consecutive transactions equal to ESRAM_MAX_LAT value on that slave, and then re-arbitrate for eSRAM access

SW_WEIGHT_<master>

- Each master except D-Code M3 bus has a programmable weight value which can be configured from 1 to 32
- **SW_WEIGHT_<master>**
 - <master> = M3 IC bus, M3 system bus, System controller, HPDMA, FIC_0, FIC_1, MAC, PDMA and USB
 - D-Code bus does not need a programmable weight since it has the highest priority
- **SW_WEIGHT_<master>** is the number of consecutive transfers that the Weighted Master can perform without being interrupted by a Fixed Priority master, or before moving onto the next master in the weighted round robin cycle

SW_WEIGHT_<master> values

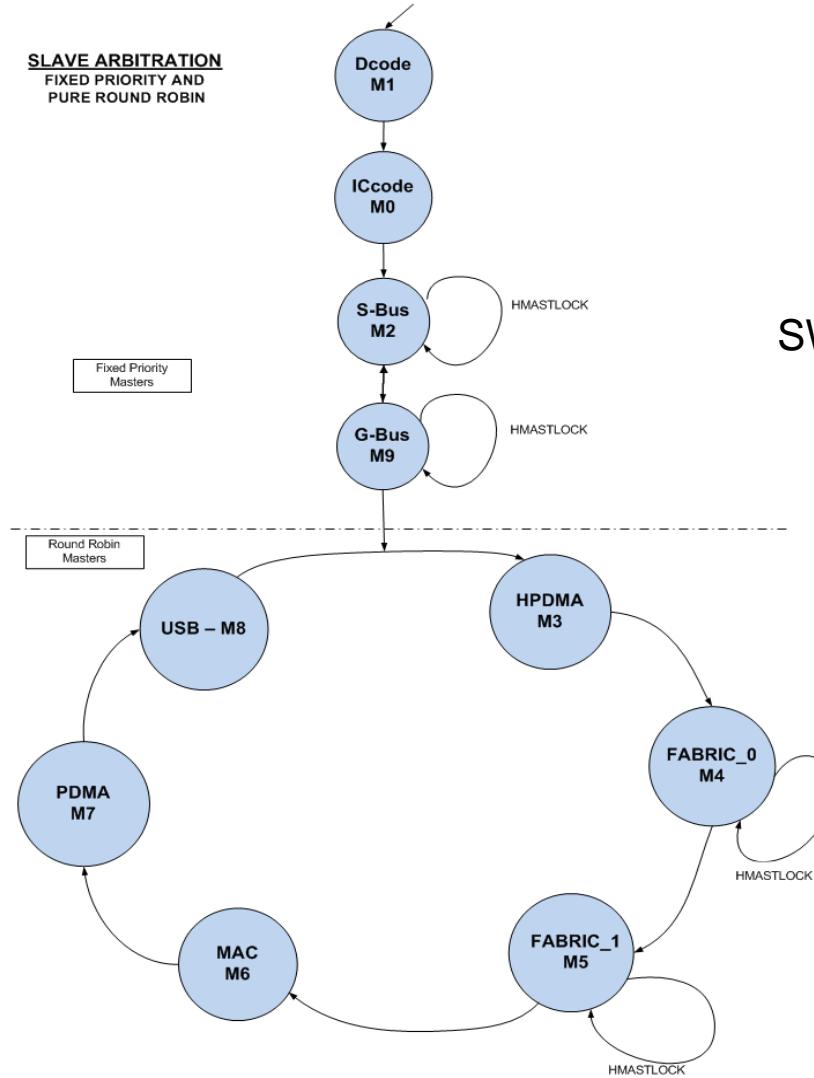
SW_WEIGHT_<master>	Weight
0	32
1	1
2	2
3	3
.	.
.	.
.	.
28	28
29	29
30	30
31	31

5-bit register for each master

MASTER_WEIGHT0_CR @ 0x4003803C

MASTER_WEIGHT1_CR @ 0x40038040

Pure Round Robin Arbitration (Default)



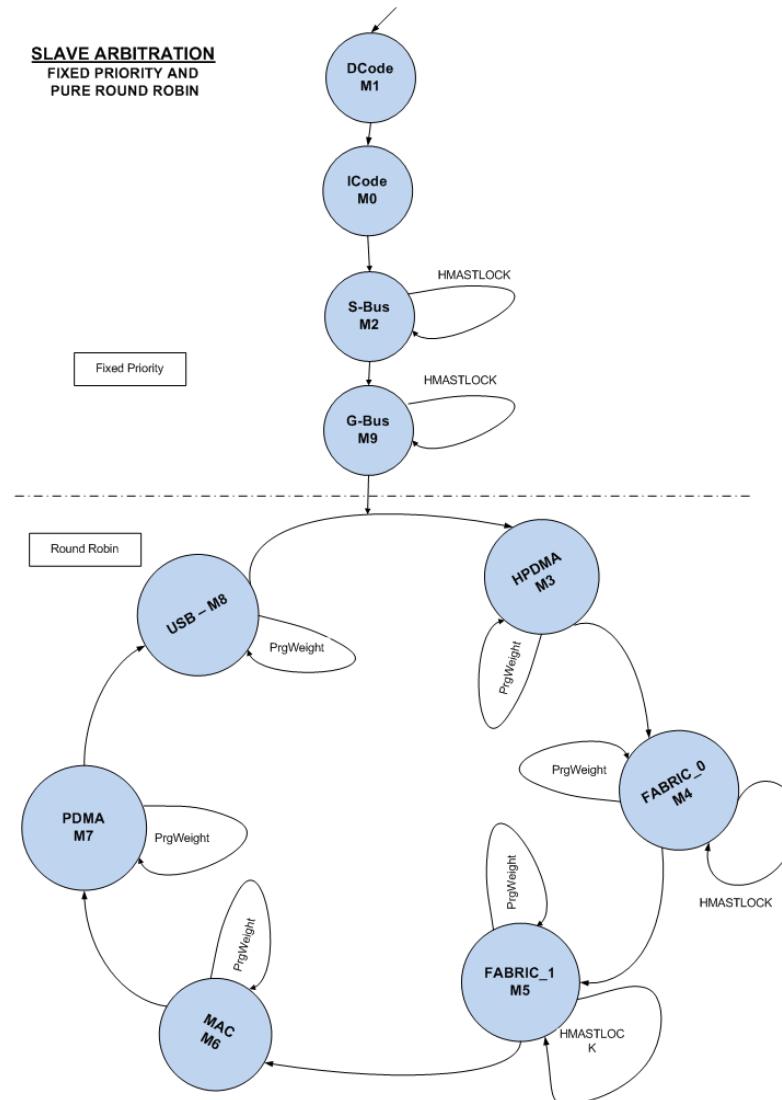
Example where
SW_WEIGHT_<master> = 1
ESRAM_MAX_LAT = 1

Round-Robin Arbitration for eSRAM1

		HCLK									
Master	Priority	1	2	3	4	5	6	7	8	9	
M3-I : M0	2				eSRAM1						
M3-D : M1	1				eSRAM1						
M3-S : M2	3		eSRAM1			eSRAM1					
HPDMA : M3	4	eSRAM1									
FIC32_0 : M4	4	eSRAM1									
FIC32_1 : M5	4	eSRAM1									
MAC : M6	4	eSRAM1									
PDMA : M7	4	eSRAM1									
eSRAM1 : S1		HPDMA M3	M3-S M2	FIC32_0 M4	M3-D M1	M3-I M0	M3-S M2	FIC32_1 M5	MAC M6	PDMA M7	

ESRAM1_MAX_LAT = 1
SW_WEIGHT_<master> = 1

Weighted Round Robin Arbitration Mode



HBURST Support for eNVM

- HBURST support for eNVM exists for IC bus only
 - I = M3 code space C = Cache
- The Switch handshakes correctly with masters performing AHB-Lite bursts to any slave
- The transactions are passed to slaves as single cycle accesses of type NONSEQ
 - This simplifies the design of the slaves, as they don't need to support AHB-Lite bursts
 - Avoids long latencies incurred by bursts of indeterminate length (e.g. from the FPGA fabric). The Switch doesn't connect to the HBURST bus of the master or slave for which HBURST is not supported
- SW_WEIGHT_IC should be programmed to support the maximum burst users expect from either the I or C bus
 - SW_WEIGHT_IC = 32 allows a cache line to be filled from eNVM without interruption

Arbitration for eNVM

	HCLK															
Master	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
M3-D M1		S3B4														
FIC32_0 M5	S3-B12															
eNVM	M5-B1	M5-B2	M5-B3	M5-B4	M5-B5	M5-B6	M5-B7	M5-B8	M1-B1	M1-B2	M1-B3	M1-B4	M5-B9	M5-B10	M5-B11	M5-B12

$$\text{SW_WEIGHT_FIC32_0} = 8$$

Burst Transactions supported for eNVM

Switch Access Security

- Designed to control access to memory slaves from masters
 - We have 10 Masters
 - M3 – (3) (IC, D, S)
 - HPDMA
 - FIC32_0
 - FIC32_1
 - PDMA
 - Ethernet MAC
 - USB
 - SII Master (G4 Control)
 - We have 5 memory Slaves
 - eSRAM0
 - eSRAM1
 - eNVM0
 - eNVM1
 - DDRBRIDGE

Switch Access Security

- Masters are “grouped”
 - M3
 - FIC32_0, FIC32_1
 - HPDMA, PDMA, Ethernet, USB
 - SII Master (G4 control)
- 4 Master Groups * 5 Memory Slaves = 20 User Flash Bits
- User defined Master Group(s) can be prevented from accessing the eNVM's
 - On Block
 - HRESP asserted, transaction cancelled, Reads return garbage, writes have no effect.

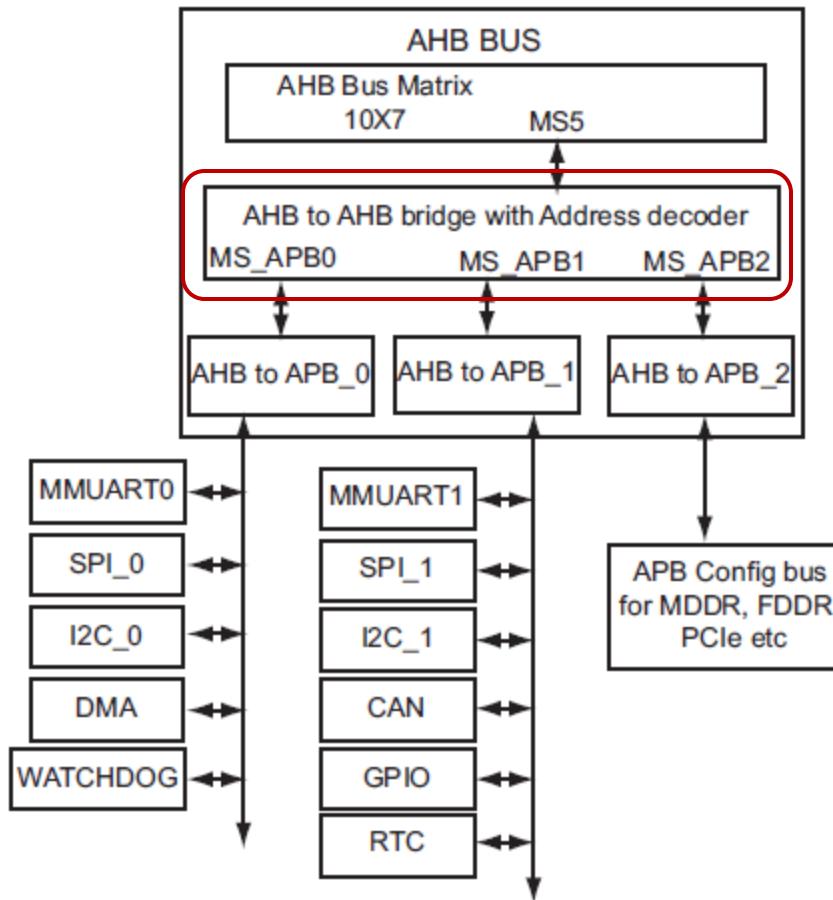
Memory Security: Port Blocking (TS Devices)

1 R and 1 W flash bit
for each accessible slave

Masters		Slaves				
		MS0	MS1	MS2	MS3	MS6
		eSRAM0	eSRAM1	eNVM0	eNVM1	DDR Bridge
M3 D-Code Bus	MM1	RW	RW	RW	RW	RW
	MM0					
	MM2					
	System Controller	MM9	RW	RW	RW	RW
	HPDMA	MM3	RW	RW	RW	RW
	MAC	MM6				
	PDMA	MM7				
	USB	MM8	RW	RW	RW	RW
	FIC32_0	MM4				
	FIC32_1	MM5				

AHB to AHB Bridge

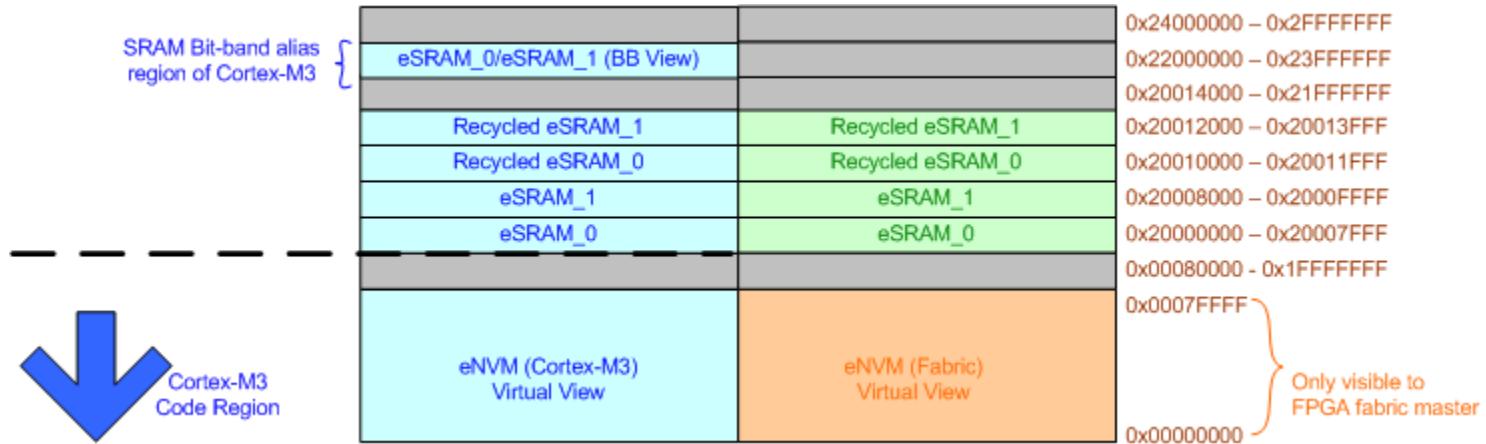
- Low performance busses go through the AHB to AHB bridge
 - Reduces the load on the AHB bus matrix
 - Inserts one cycle delay in each direction



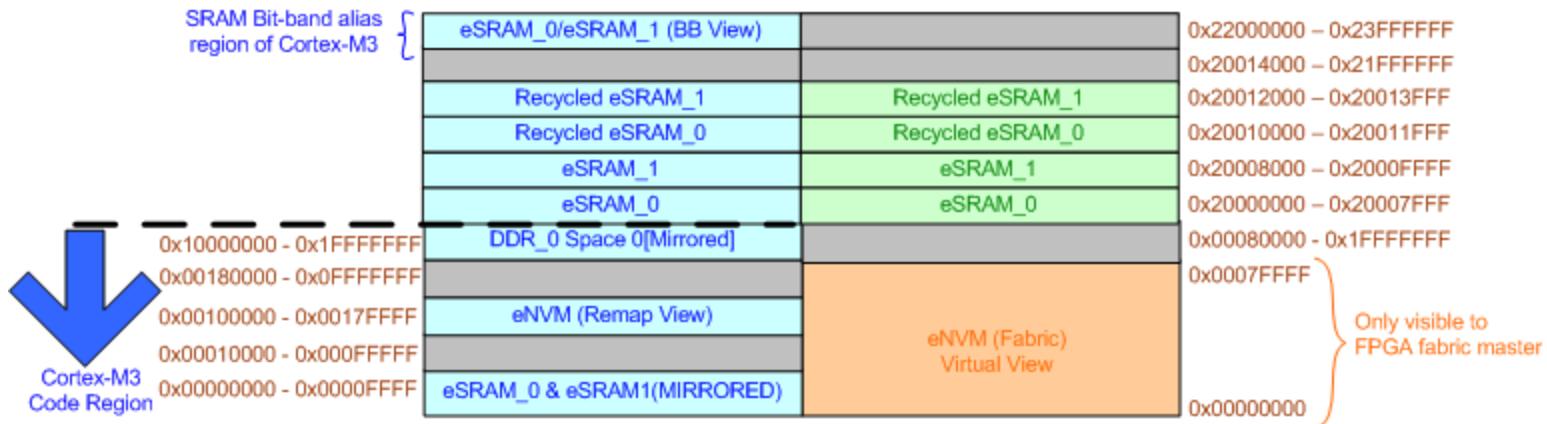
eSRAM Remap

- AHB bus matrix supports remapping the eSRAM blocks into code space with SECDED ON or OFF
 - When SECDED is ON, the two eSRAM blocks are re-mapped to appear at the bottom of Cortex-M3 processor code space, and the SECDED eSRAM is not remapped, and cannot be used by the Cortex-M3
 - When SECDED is OFF, the two eSRAM blocks are re-mapped to appear at the bottom of Cortex-M3 processor code space and the SECDED eSRAM can be used by the Cortex-M3

Memory Map - Default



Memory Map – eSRAM Remapped

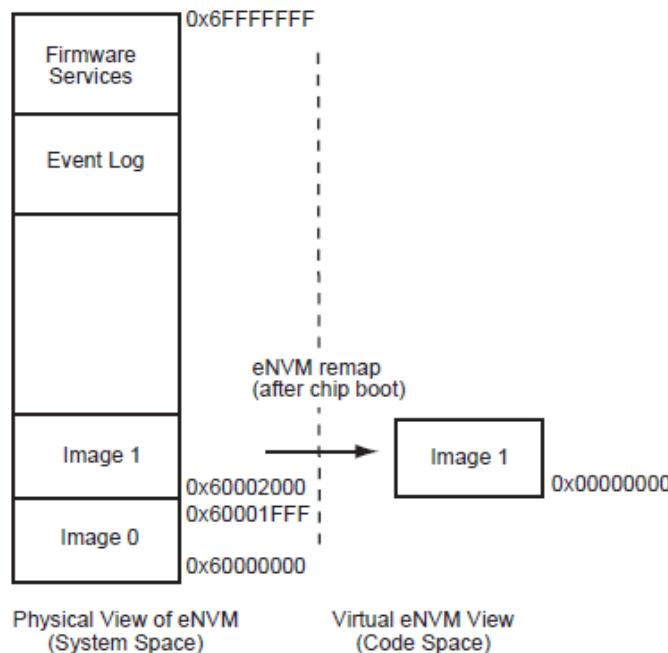


Useful for:

- Debugging
- Locating ISRs in on-chip SRAM for fast execution

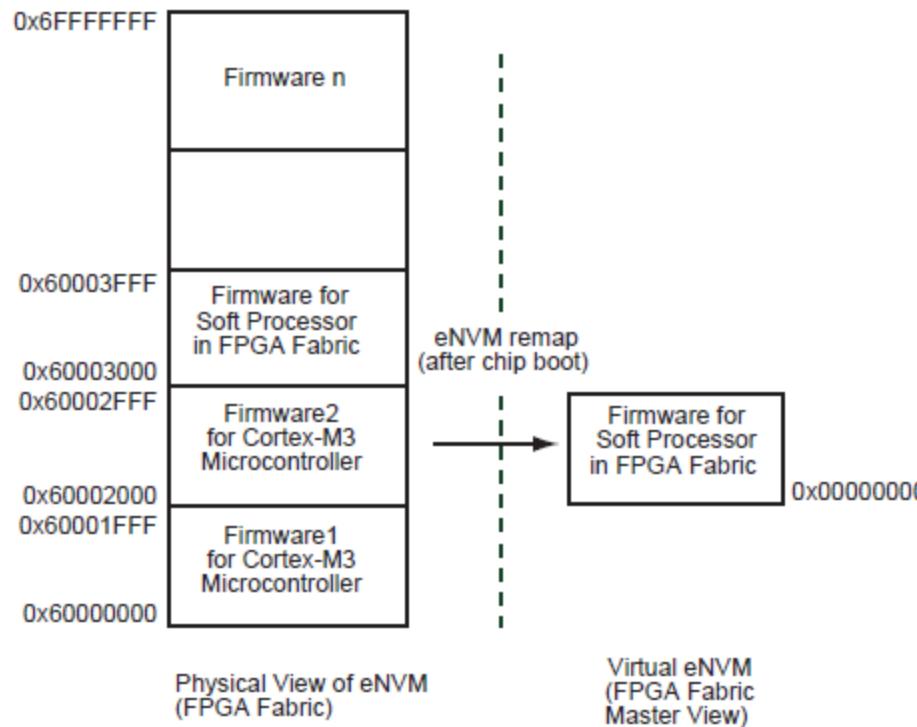
eNVM Remap

- eNVM may contain multiple firmware images
- AHB bus matrix handles the eNVM remapping from the base address to the range from 0x00000000 to 0x1FFFFFFF
 - Only the Cortex-M3 processor ICode and DCode AHB busses access the eNVM in this range
- ENVM_CR and ENVM_REMAP_BASE_CR registers in the SYSREG block control eNVM remapping



eNVM Remap for Soft Processor

- AHB bus matrix supports remapping an eNVM segment to address 0x00000000 as seen by Fabric masters
- ENVM_REMAP_FAB_CR register in the SYSREG block controls eNVM remapping for fabric masters



Cache Controller

Cache Controller

- 8 Kbyte instruction cache
 - No data cache because of data coherency concerns
- The cache controller can treat eSRAM, eNVM or DDR as main memory

Cache Controller Features

Cache Size	8KB
Associativity	4 ways
Cache Line Size	32 Bytes (Fixed)
Error Protection	Data RAM protected
Replacement Policy	LRU
Fill Mechanism	Full Cache line refill, critical word first
Write Mechanism	M3 has Read/Write Access through S-Bus
Performance	Zero-wait state on Cache Hit

Software Configurable Options

- Enable/Disable
 - Global
- Cacheable Region
 - 128MB, 256MB, or 512MB
- Remap Modes
 - Default (eNVM), eSRAM Remap, or DDR Remap
- Lockable
- Error Protection
 - SECDED Enable/Disable
- Management
 - Cache Line Invalidate, Full cache invalidate
- Pipelining
 - Programmable one stage pipeline for eSRAM

Memory Map – Default (eNVM)

Data/Code Region	Space	Address
CM3 Data Region	Reserved	0xE000_0000 to 0xFFFF_FFFF
	DDR _SPACE 3 (256 MB)	0xD000_0000 to 0xDFFF_FFFF
	DDR _SPACE 2 (256 MB)	0xC000_0000 to 0xCFFF_FFFF
	DDR _SPACE 1 (256 MB)	0xB000_0000 to 0xBFFF_FFFF
	DDR _SPACE 0 (256 MB)	0xA000_0000 to 0xAF00_FFFF
	eNVM SFR, Remap Area etc (1 GB)	0x6000_0000 to 0x9FFF_FFFF
	Peripheral [SPI, UART, CAN, Fabric etc.] (0.5 GB)	0x4000_0000 to 0x5FFF_FFFF
	Reserved	0x2001_0000 to 0x3FFF_FFFF
	eSRAM-1 (32 KB)	0x2000_8000 to 0x2000_FFFF
	eSRAM-0 (32 KB)	0x2000_0000 to 0x2000_7FFF
CM3 Code Region	Reserved	0x0008_0000 to 0x1FFF_FFFF
	eNVM (Virtual View) [512 KB]	0x0000_0000 to 0x0007_FFFF

Memory Maps – eSRAM Remap

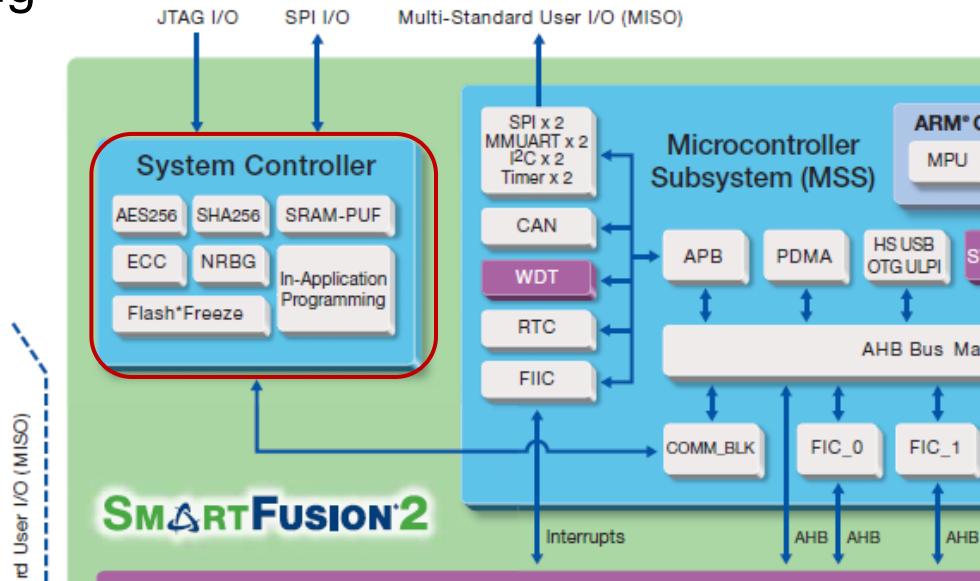
Data/Code Region	Space	Address
CM3 Data Region	Reserved	0xE000_0000 to 0xFFFF_FFFF
	DDR _SPACE 3 (256 MB)	0xD000_0000 to 0xDFFF_FFFF
	DDR _SPACE 2 (256 MB)	0xC000_0000 to 0xCFFF_FFFF
	DDR _SPACE 1 (256 MB)	0xB000_0000 to 0xBFFF_FFFF
	DDR _SPACE 0 (256 MB) [MIRRORED]	0xA000_0000 to 0xAF00_FFFF
	eNVM SFR, Remap Area etc (1 GB)	0x6000_0000 to 0x9FFF_FFFF
	Peripheral [SPI, UART, CAN, Fabric etc.] (0.5 GB)	0x4000_0000 to 0x5FFF_FFFF
	Reserved	0x2001_0000 to 0x3FFF_FFFF
	eSRAM-1 (32 KB) [MIRRORED]	0x2000_8000 to 0x2000_FFFF
	eSRAM-0 (32 KB) [MIRRORED]	0x2000_0000 to 0x2000_7FFF
CM3 Code Region	DDR _SPACE 0 (256 MB)	0x1000_0000 to 0x1FFF_FFFF
	Reserved	0x0018_0000 to 0x0FFF_FFFF
	eNVM (Virtual View) [512 KB]	0x0010_0000 to 0x0017_FFFF
	Reserved	0x0001_0000 to 0x000F_FFFF
	eSRAM0 & eSRAM1 [64 KB]	0x0000_0000 to 0x0000_FFFF

Memory Maps – DDR Remap

Data/Code Region	Space	Address
CM3 Data Region	Reserved	0xE000_0000 to 0xFFFF_FFFF
	DDR _SPACE 3 (256 MB)	0xD000_0000 to 0xDFFF_FFFF
	DDR _SPACE 2 (256 MB)	0xC000_0000 to 0xCFFF_FFFF
	DDR _SPACE 1 (256 MB)	0xB000_0000 to 0xBFFF_FFFF
	DDR _SPACE 0 (256 MB)	0xA000_0000 to 0xAF00_FFFF
	eNVM SFR, Remap Area etc (1 GB)	0x6000_0000 to 0x9FFF_FFFF
	Peripheral [SPI, UART, CAN, Fabric etc.] (0.5 GB)	0x4000_0000 to 0x5FFF_FFFF
	Reserved	0x2001_0000 to 0x3FFF_FFFF
	eSRAM-1 (32 KB)	0x2000_8000 to 0x2000_FFFF
	eSRAM-0 (32 KB)	0x2000_0000 to 0x2000_7FFF
CM3 Code Region	DDR _SPACE 1 (256 MB)	0x1000_0000 to 0x1FFF_FFFF
	DDR _SPACE 1 (256 MB)	0x0000_0000 to 0x0FFF_FFFF

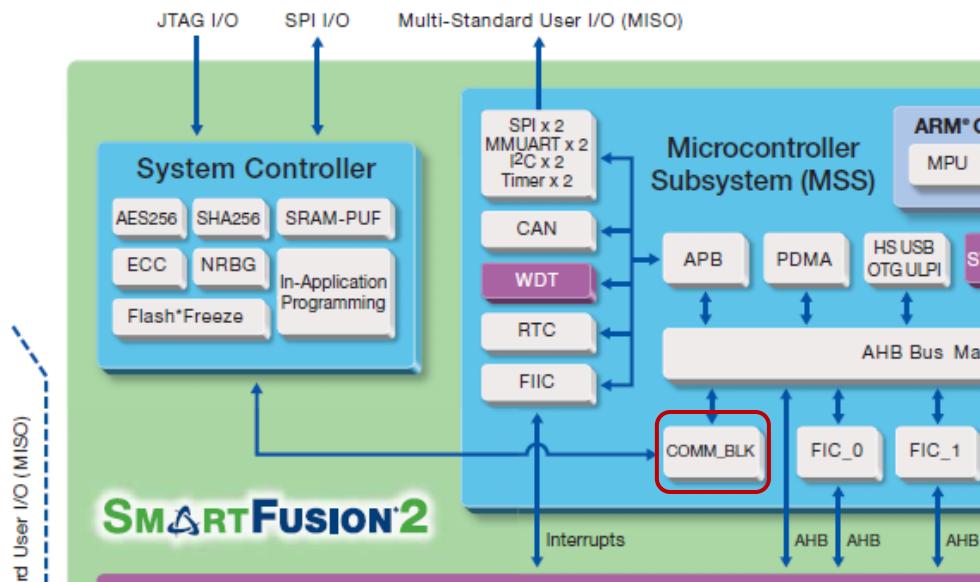
System Controller

- The System Controller runs System Services at the request of the user
- System Services Examples:
 - Enter Flash*Freeze
 - Get Random Number
 - Get device ID
 - Programming



COMM_BLK

- The COMM_BLK is the interface to the System Controller
 - TX and RX FIFO based command / message protocol
 - Accessible via M3 or user logic via the FIC



Inputs / Outputs

SmartFusion2 I/Os

- Multi Standard I/Os (MSIO)
 - LVTT/LVCMOS 3.3V
 - LVCMOS 1.2V, 1.5V, 1.8V, 2.5V
 - LVDS, MLVDS, mini-LVDS, RSRS differential standards
 - PCI
 - LVPECL (Receiver Only)
- MSIOD
 - Same as MSIO without PCI, 2.5V Max, no LVPECL
- DDR I/Os (DDRIO)
 - DDR, DDR2, DDR3, LPDDR, SSTL2, SSTL18, HSTL
 - LVCMOS 1.2V – 2.5V

I/O Standard Summary

I/O Standards	I/O Types		
	MSIO	MSIOD	DDRIO
Single-Ended I/O			
LV TTL 3.3V	Yes	-	-
LVC MOS 3.3V	Yes	-	-
PCI	Yes	-	-
LVC MOS 1.2V	Yes	Yes	Yes
LVC MOS 1.5V	Yes	Yes	Yes
LVC MOS 1.8V	Yes	Yes	Yes
LVC MOS 2.5V	Yes	Yes	Yes
Voltage-Referenced I/O			
HSTL 1.5V	-	-	Yes
SSTL 1.8V	-	-	Yes
SSTL 2.5V	Yes	Yes	Yes
SSTL 2.5V (DDR1)	Yes	Yes	Yes
SSTL 1.8V (DDR2)	-	-	Yes
SSTL 1.5V (DDR3)	-	-	Yes
Differential I/O			
LVPECL (input only)	Yes	-	-
LVDS 3.3V	Yes	-	-
LVDS 2.5V	Yes	Yes	-
RS DS	Yes	Yes	-
BLVDS	Yes	Yes	-
MLVDS	Yes	Yes	-

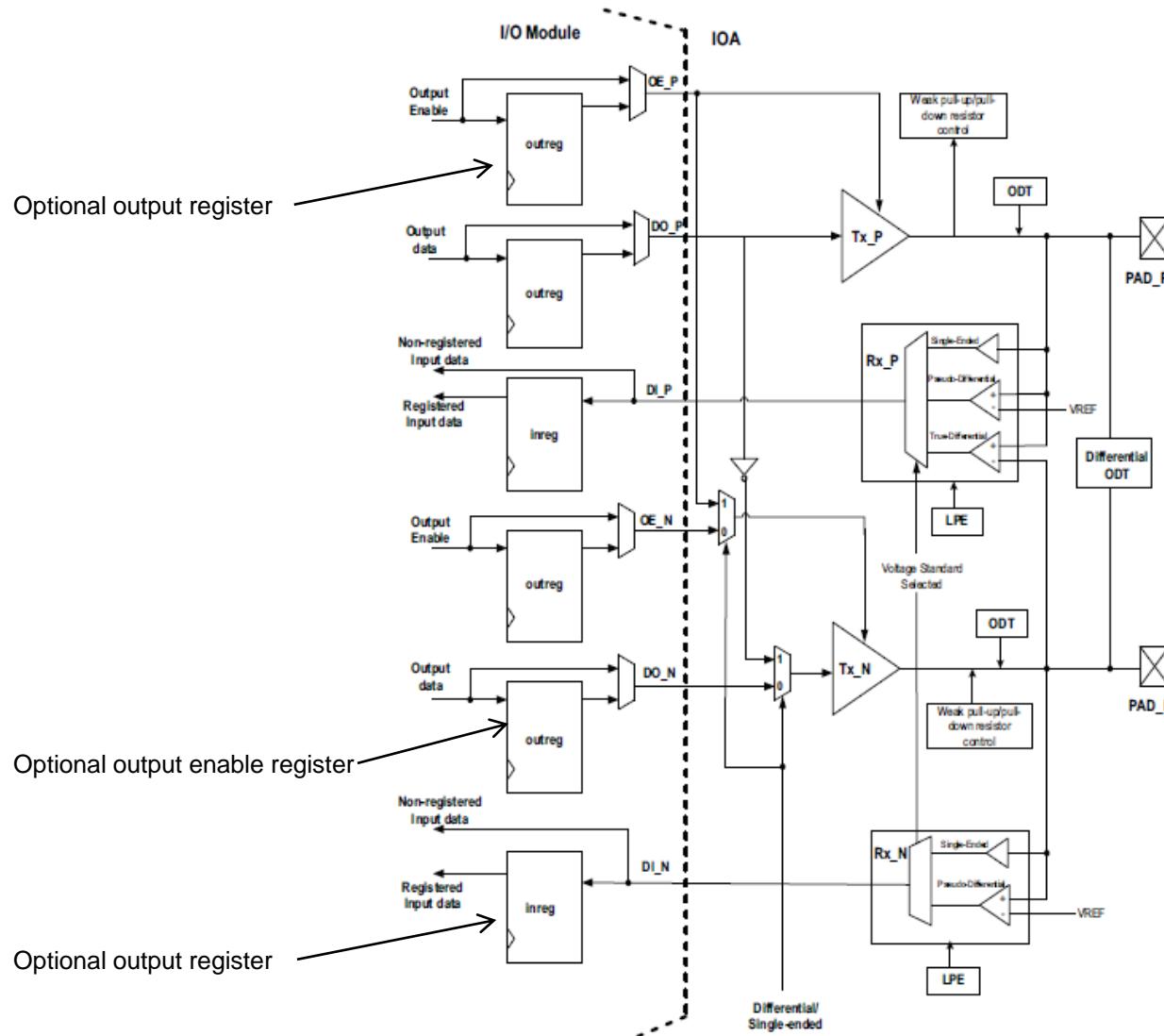
Programmable I/O Features

I/O Features	I/Os		
	MSIO	MSIOD	DDRIO
Single-ended transmitter			
Programmable drive strength	Yes	Yes	Yes
Programmable Weak pull-up and pull-down	Yes	Yes	Yes
Configurable ODT	Yes	Yes	Yes
Hot insertion capable	Yes	-	-
Bus keeper	Yes	Yes	Yes
I/O state control in Low power mode	Yes	Yes	Yes
LVTTL/LVC MOS 3.3 V outputs compatible with external 5 V TTL inputs	Yes	-	-
Pre-emphasis capability	-	Yes	-
Programmable Slew rate	-	-	Yes
Single-ended receiver			
5 V tolerant with minimal use of external circuitry	Yes	Yes	-
Schmitt receiver	Yes	Yes	Yes
LPE (Signature mode and Activity mode)	Yes	Yes	Yes
Programmable input delay	Yes	Yes	Yes
Programmable Slew rate	-	-	Yes
Differential transmitter			
Programmable Weak pull-up and pull-down	Yes	Yes	Yes
Configurable ODT	Yes	Yes	Yes
Programmable Slew rate	-	-	Yes
Differential receiver			
100 Ω Differential ODT	Yes	Yes	-
Schmitt receiver	Yes	Yes	Yes
LPE (Signature mode and Activity mode)	Yes	Yes	Yes
Programmable input delay	Yes	Yes	Yes
Programmable Slew rate	-	-	Yes

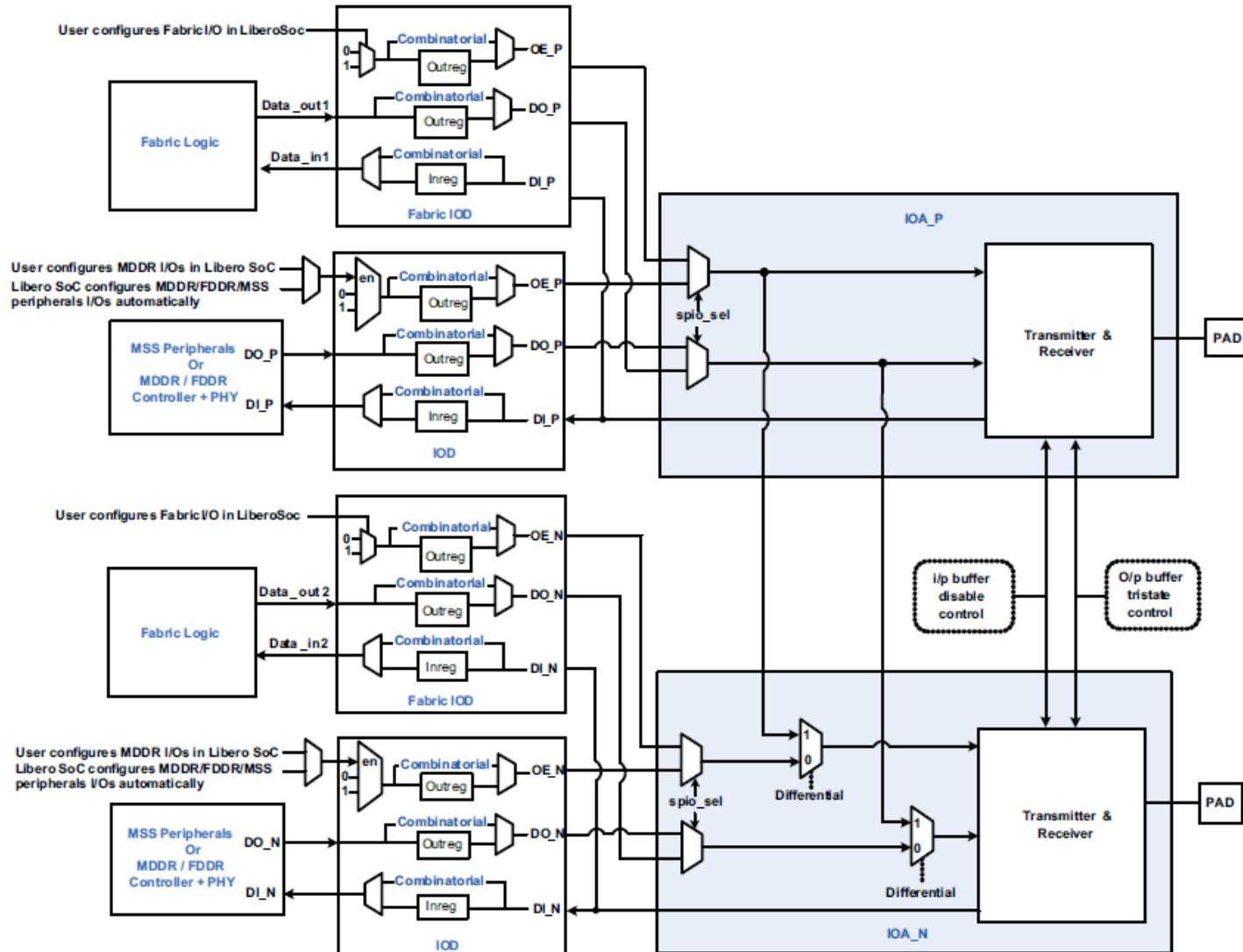
I/O Assignment

- DDRIOS with MDDR/FDDR
 - If you select MDDR/FDDR, Libero SoC automatically connects MDDR/FDDR signals to the DDRIOS.
 - Only the required DDRIOS are used by Libero SoC. The unused DDRIOS are available to connect to the FPGA fabric.
- DDRIOS with Fabric
 - If MDDR/FDDR is not used, DDRIOS are available to the FPGA fabric
 - Manually configure DDRIOS in Libero SoC
- MSIO/MSIODs with MSS Peripherals
 - If MSS peripherals are enabled, Libero SoC automatically connects MSS peripheral signals to either MSIOs or to the MSIODs. The unused MSIOs or MSIODs are available to the FPGA fabric.
- MSIO/MSIODs with Fabric
 - MSIO/MSIODs are available to the FPGA fabric when MSS peripherals are disabled.
 - Manually configure MSIO/MSIOD in Libero SoC.

Multi Standard I/O



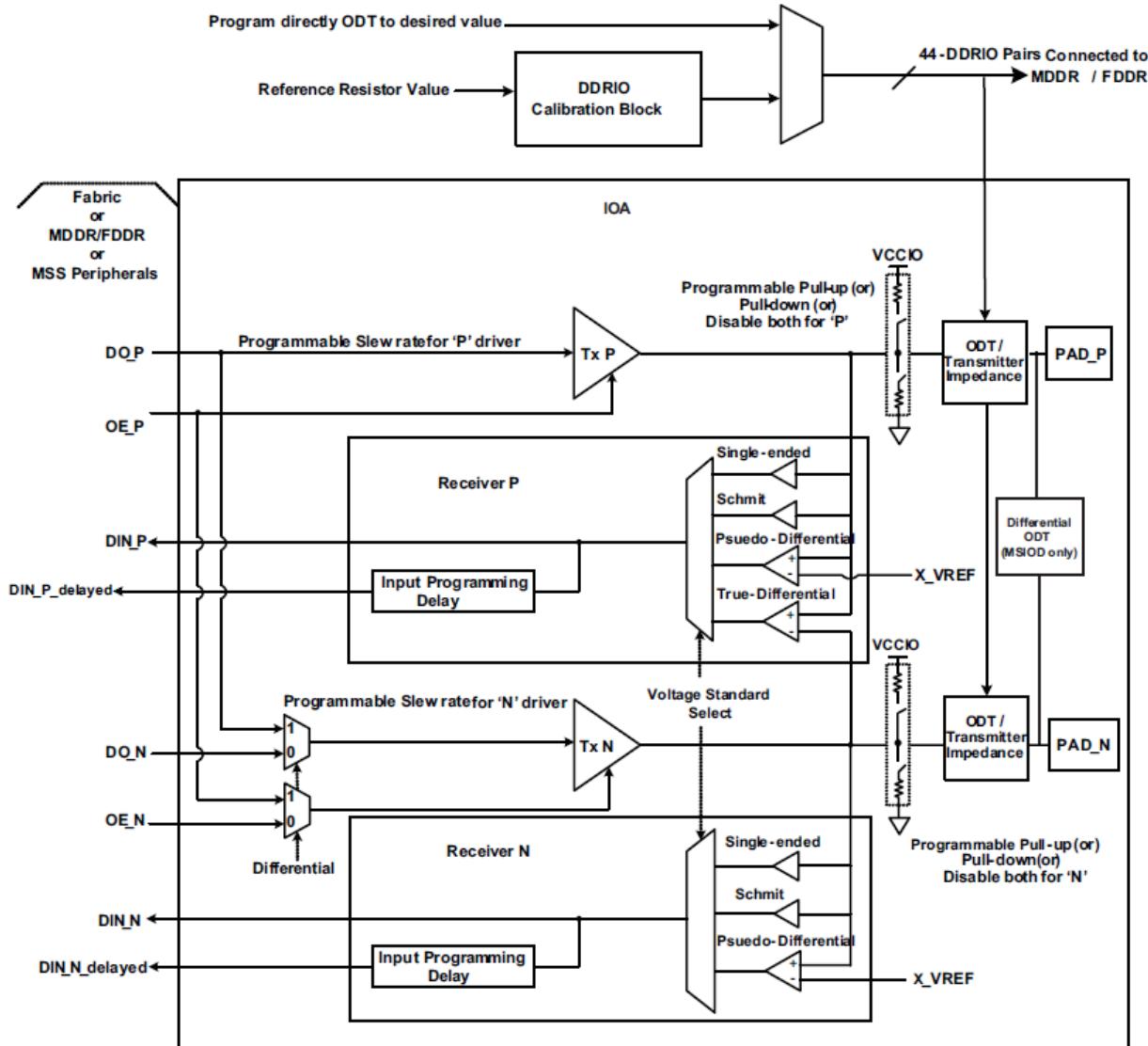
I/O Interconnection with Fabric and MDDR/FDDR/ MSS Peripherals



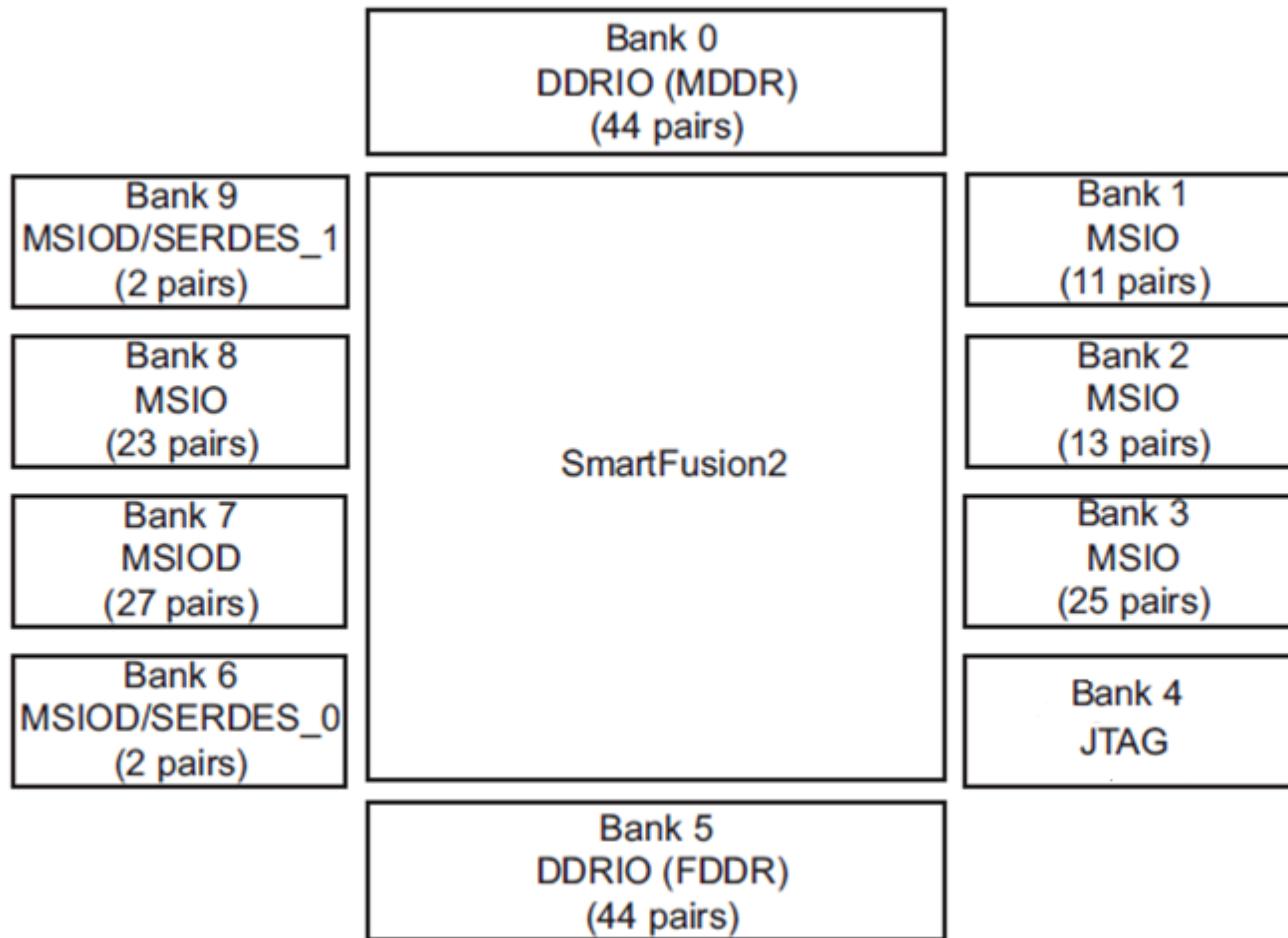
I/O Structure

- I/O consists of a highly featured bidirectional I/O buffer.
- The I/O is divided into two main sections:
 - Digital – IOD (fabric and MDDR/FDDR/ MSS Peripherals)
 - The digital (IOD) section generates output enable (OE), data out (DO), and data in (DIN) signals for both P and N IOA pairs.
 - Analog – IOA
 - The analog section (IOA) has transmitter and receiver buffers for the P and N pair.
- The main circuits in the IOA are transmitting and receiving buffers that support various I/O standards and contain the following modules:
 - Transmit buffer
 - Receive buffer
 - Low power exit (LPE) logic
 - On Die Termination

IOA Architecture



M2S050 I/O Bank Location



System Register Block

System Register Block

- The System Register (SYSREG) block is responsible for configuring the state of the MSS
- The SYSREG block is connected to the AHB bus matrix and can be accessed by all bus masters
- Users can define the power on reset state for the registers through the use of flash bits in the SYSREG

SYSREG Register Types

Type	Function
RW-P	Register is readable and writeable. P indicates that the register is initialized using flash bits which are part of FPGA configuration bits and are set as a part of the users design
RW	RW indicates that the register is readable and writeable and is not flash initialized
RO	RO is a read only register
RO-U	RO-U is a read only register type. U denotes that it can be initialized by user flash bits. These bits are device configuration bits and cannot be changed as part of the user design
RO-P	RO-P is a read only register type. P denotes that it can be initialized using flash bits which are a portion of the FPGA configuration bits and are set as a part of the user design
W1P	Writing 1 into a register of this type pulses the output
SW1C	SW1C is used for the MSS status and clear function. Set when input is asserted and is cleared when a '1' written.

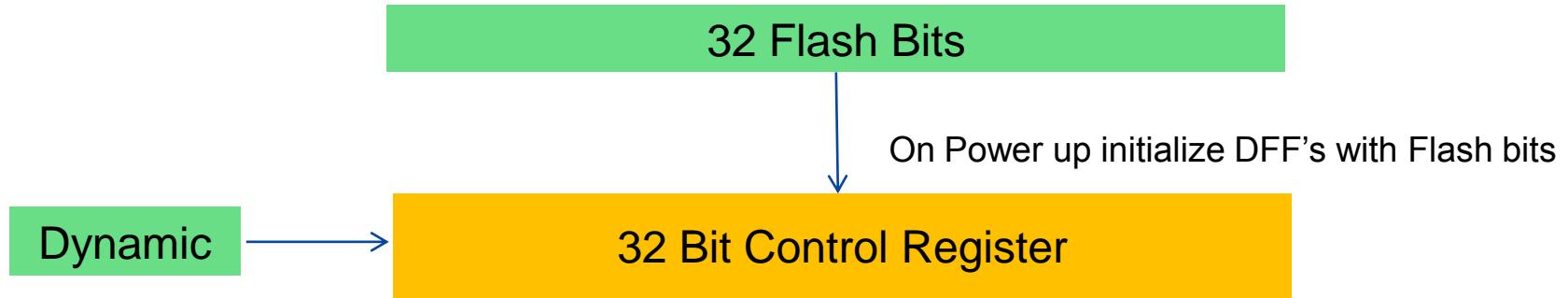
Flash Configuration Bits

- Liberal use of Flash configuration Bits to provide a static configuration of parts of the MSS
 - Watch Dog Timer
 - Cortex M3 or user logic can only reset it.
 - SYSREG Block (bank of control registers)
 - Users can re-define the default power up behavior
 - All registers configured by Flash bits
 - So users can prevent accidental re-configuration of portions of the MSS

These are not flash bits located in the eNVM, they are FPGA configuration bits

SYSREG Block

- Controls everything but the peripherals
 - AHB Bus Matrix
 - DDR Bridge
 - Cache Controller/Matrix
 - IOMUX'es
 - SECDED Enable/Disable
 - Soft Resets

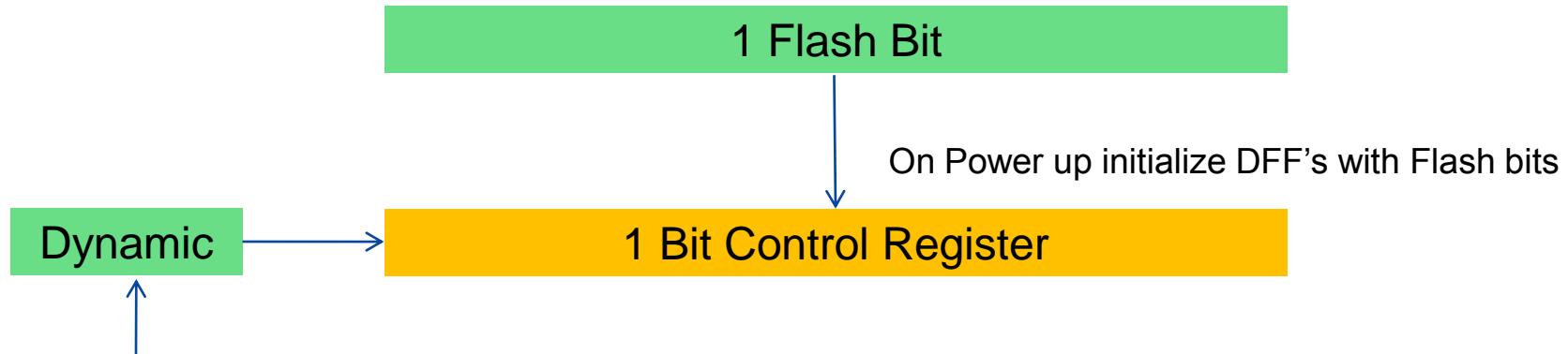


Dynamic bit keeps Bus Masters from changing contents of 32 Bit Control register

SYSREG Block

■ Reset Control

- Each peripheral has a reset register
- Allows high rel users the ability to permanently disable a peripheral
 - Kept in low power reset state.

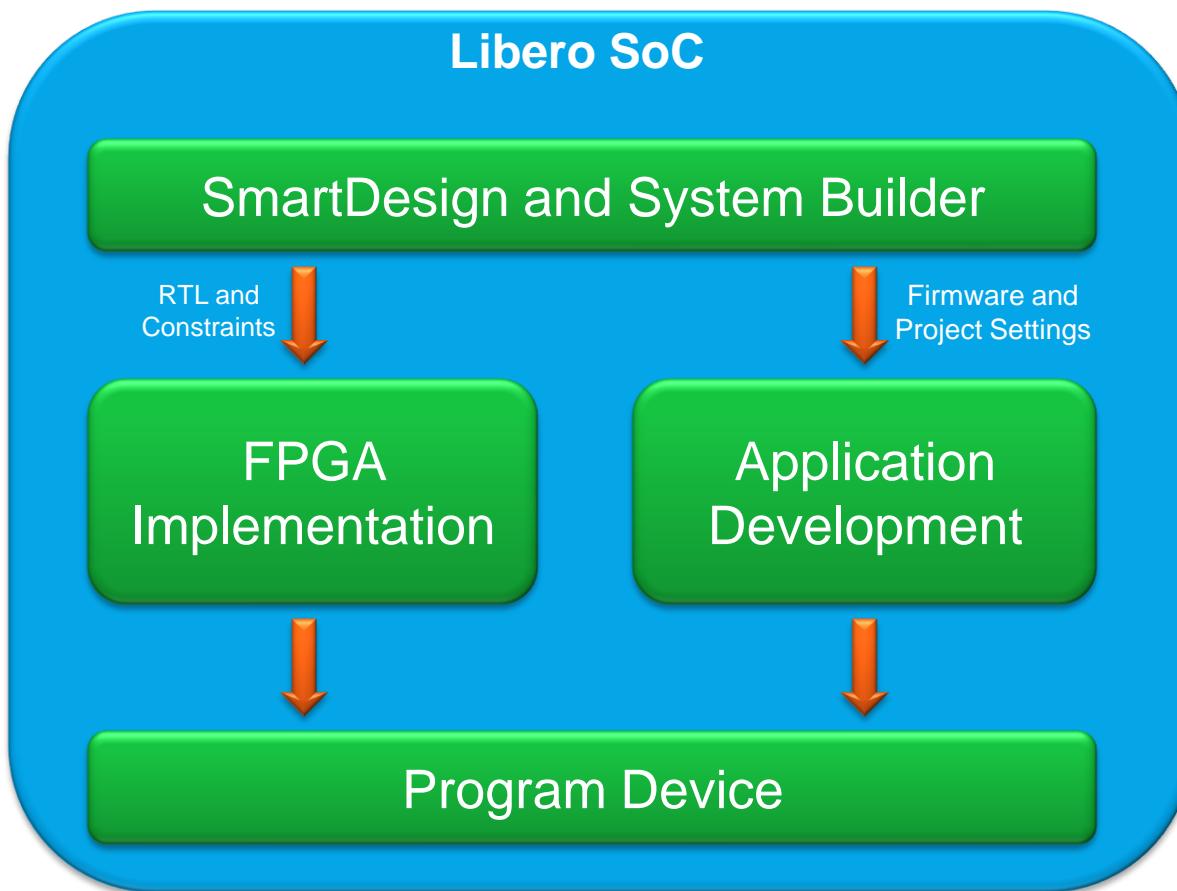


Dynamic bit keeps Bus Masters from changing contents of 1 Bit Control register

Design Tools

SmartFusion2 Design Creation

- Libero SoC v11.5
 - Integrated environment for System-on-Chip Design



Libero SoC v11.5

- **FPGA Design Tools**
 - System Builder
 - SmartDesign design entry
 - Rich IP library & user block support facilitates design reuse
 - Synplify Pro® synthesis
 - ModelSim® simulation
 - Power-driven place-and-route
 - SmartPower power analysis
 - SmartTime timing analysis
 - Flexible Programming Options
 - SmartDebug
- **Embedded Design Flow**
 - System Configuration
 - Project Creation
 - Firmware generation
 - Include file setup
 - SoftConsole Eclipse-based IDE
 - GNU GCC
 - GNU GDB debug

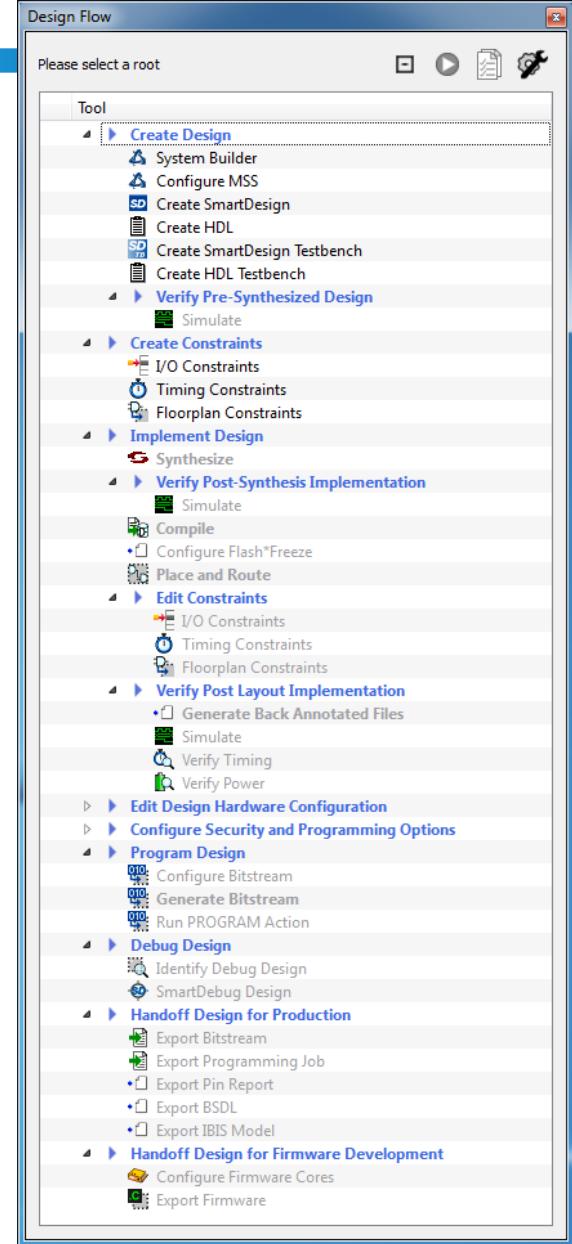


Libero SoC v11.5 Content

- Libero SoC v11.5 includes the following:
 - Microsemi Firmware Catalog 11.5
 - Microsemi FlashPro 11.5
 - ModelSim ME 10.3c
 - Synopsys Synplify Pro ME I-2014.03M-SP1
 - IP Catalog with approximately 50 IP Cores

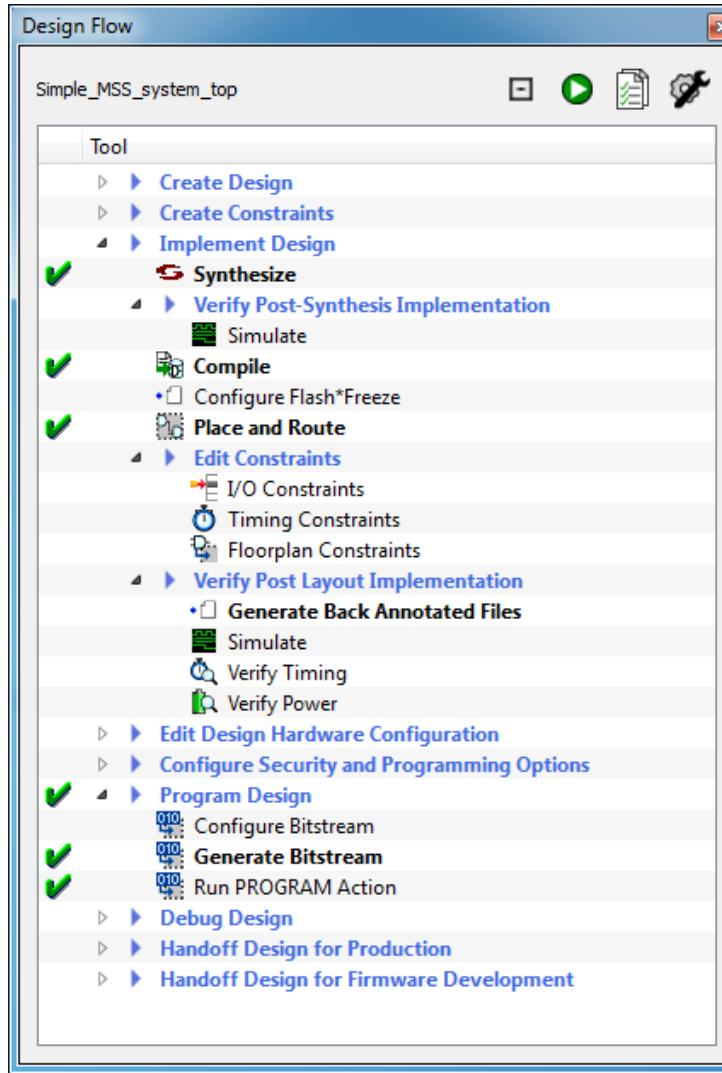
Libero SoC Design Flow Window

- Single Window Provides Access to tools and shows Design Status
- Tools Can be Run in Background or Launched from Design Flow Window or Menus
- Project Flow Window Displays:
 - Tools
 - Design Creation
 - Includes option to import files into the project
 - Stimulus Creation
 - Processing
 - Simulation
 - Synthesis
 - Place and Route
 - Transitions
 - Current State
 - Tool Tips

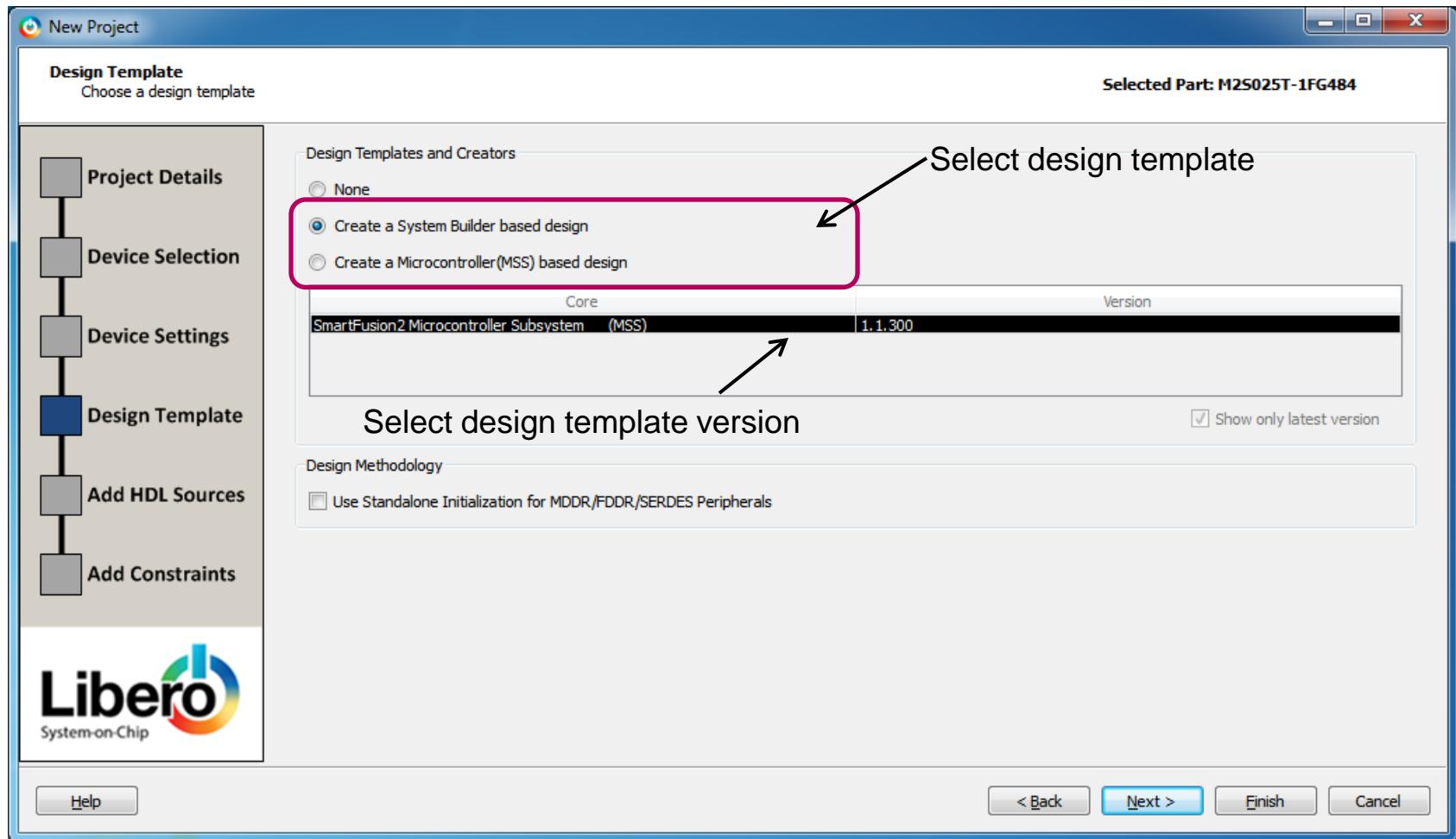


Design Flow Window: Design Status

- Check marks indicate completed steps in Design Flow



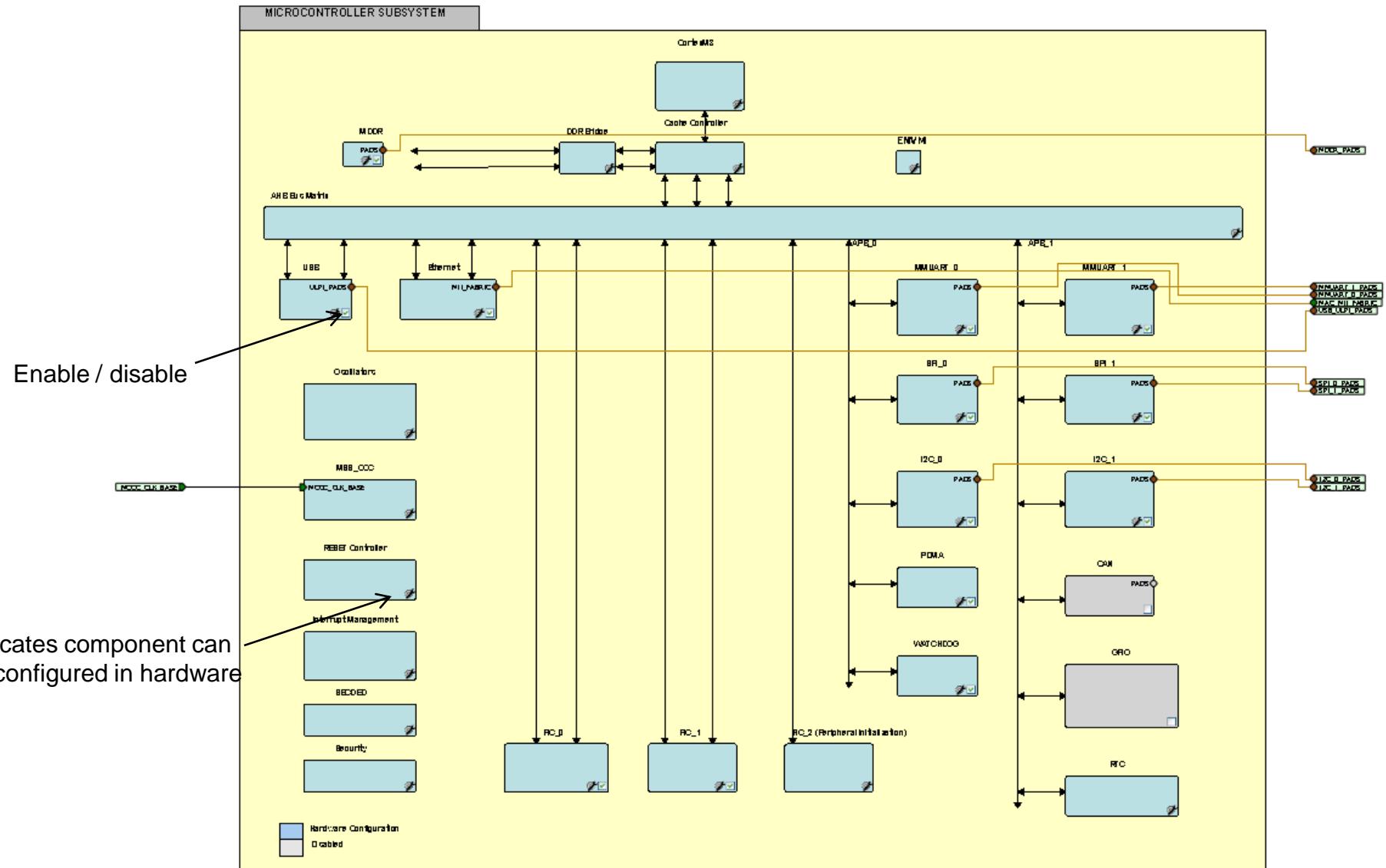
Implementing SmartFusion2 Designs



SmartFusion2 Design Templates

- MSS Configurator - a specialized SmartDesign that enables you to configure the SmartFusion2 MSS
 - The configurator presents a graphical block diagram of the SmartFusion2 Microcontroller Subsystem
 - Enable/disable and configure each MSS sub-block as per your application requirements
- System Builder - a powerful tool that enables you to specify your design based on high level specifications

SmartFusion2 MSS Configurator



MSS Configurator: Features

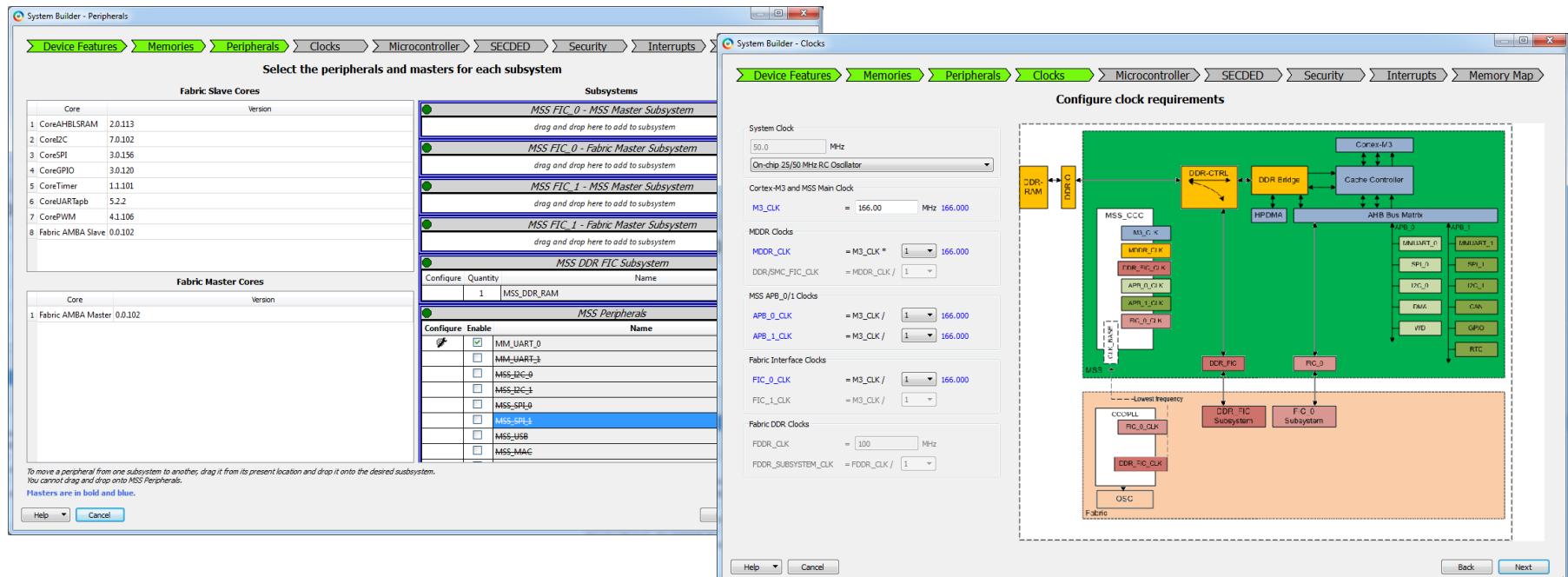
- Independent Application
 - Can be launched from Libero Project Flow window
- Only allows MSS Configuration
 - Must instantiate MSS component (in SmartDesign or RTL) for Fabric extension
- I/O Management
 - Automatically assigns all MSS I/Os
 - Automatic disabling/enabling of options based on user configurations
- Graphical widgets
 - Makes enable/disable and configure actions more obvious

System Builder

SmartFusion2 System Builder

■ System Builder Wizard

- Asks the user basic questions on system architecture
- Adds any additional peripherals in the fabric
- Walks through configuration options for each selected feature
- Builds complete base system and API – correct by design



Accelerates architecture design, so engineers can focus on their value add

System Builder – Device Features Page

- Select the SmartFusion2 features used in the design including:
 - MSS DDR
 - Fabric DDR
 - High-speed serial interfaces
 - Creating data storage clients in the Flash Memory

System Builder – Device Features Page

System Builder - Device Features

Device Features > Memories > Peripherals > Clocks > Microcontroller > SECDED > Security > Interrupts > Memory Map

Select the SmartFusion2 features you will be using in your design

Memory

- MSS External Memory
 - MDDR
 - Soft Memory Controller (SMC)
- MSS On-chip Flash Memory (eNVM)
- Fabric External DDR Memory (FDDR)

High Speed Serial Interfaces

- SERDESIF_0
- SERDESIF_1

Microcontroller Options

- Watchdog Timer
- Peripheral DMA
- Real Time Counter

The diagram illustrates the internal architecture of a SmartFusion2 device. It is divided into two main sections: the Memory section (green) and the Fabric section (orange).
The Memory section contains:

- MSS (Memory System Subsystem) with an MDDR component.
- DDR-RAM connected via a DDIO interface.
- SDR-RAM connected via a DDIO interface.
- MSS_CCC (Memory System Control Core).
- MSS_CLK.
- RESET CTRL.
- DDR Bridge.
- Cortex M3 microcontroller.
- eNVM (eNVM).
- SWITCH.
- FIC 1, FIC C, FIC 2 (Fabric Interface Components).
- CoreConfigP.
- CoreConfigR.
- DDR RAM.

Interconnections between components include: MSS_CCC to MDDR, MDDR to DDR Bridge, DDR Bridge to Cortex M3, Cortex M3 to eNVM, eNVM to CoreConfigP, CoreConfigP to CoreConfigR, CoreConfigR to DDR RAM, DDR RAM to SDR RAM, and SDR RAM back to CoreConfigR.
The Fabric section contains:

- FAB_CCC.
- Bus.
- CCRESDR.
- SERDES.

Interconnections include: Bus to CCRESDR, CCRESDR to SERDES, and SERDES to CoreConfigR.
A legend at the bottom right defines the colors and line styles:

- Red line: NODR_APB_S_PCLK
- Blue line: APB_S_PCLK
- Black line: APB_S_PRESET_N
- Green line: APB_G_PRESET_N
- Yellow line: APB_S_PCLK
- Grey line: APB_G_PRESET_N

Enabled interfaces are displayed

Help Cancel Next

System Builder – Memory Page

- Choose to access a double data rate memory (DDR) or single data rate memory (SDRAM)
 - Only appears if accessing external memory with Cortex-M3 or DDR from fabric
- DDR Options:
 - Memory standard: DDR2, DDR3, or LPDDR
 - Amount of time it will take for the DDR memory to be initialized (DDR memory settling time)
 - Import MDDR controller configuration file to match the external selected DDR memory standard selected
- Specify data storage clients in the Flash Memory (eNVM)

System Builder – Memory Page

System Builder - Memories

Device Features > Memories > Peripherals > Clocks > Microcontroller > SECDED > Security > Interrupts > Memory Map

Configure your external and embedded memories

Import or Export DDR register configuration file

Select memory type and enter configuration parameters

Memory Settings

Memory Type: DDR2

Data Width: 32

SECDED Enabled ECC:

Arbitration Scheme: Type-0

Highest Priority ID: 0

Address Mapping: {ROW,BANK,COLUMN}

Diagram:

Register Description

Help Cancel Back Next

System Builder – Select Peripherals Page

- Fabric Slave and Master cores panel lists the Fabric Slave and Fabric Master cores that you can add to a subsystem
- Subsystem Table lists the available subsystems based on previous configurations
 - Add/delete/move masters and peripherals around the subsystems to define how you want the various masters and peripherals to communicate
- Special cores in the Fabric Slave and Fabric Master list (Fabric AMBA Slave and Fabric AMBA Master) enable you to attach any custom peripherals onto the generated design
 - Drag from the available cores panel into a Subsystem in the subsystem panel and configuring these cores into a subsystem
- Click the Configure icon to configure the options of that core
- Specify how many cores with that configuration will be instantiated in the generated system

System Builder – Select Peripherals Page

System Builder - Peripherals

Device Features > Memories > Peripherals > Clocks > Microcontroller > SECDED > Security > Interrupts > Memory Map

Select the peripherals and masters for each subsystem

Fabric Slave Cores

Core	Version
1 CoreAHBLSRAM	2.0.113
2 CoreI2C	7.0.102
3 CoreSPI	3.0.156
4 CoreGPIO	3.0.120
5 CoreTimer	1.1.101
6 CoreUARTapb	5.2.2
7 CorePWM	4.1.106
8 Fabric AMBA Slave	0.0.102

Fabric Master Cores

Core	Version
1 Fabric AMBA Master	0.0.102

Subsystems

Fabric DDR Subsystem

Configure	Quantity	Name
	1	Fabric_DDR_RAM

MSS FIC_0 - MSS Master Subsystem
drag and drop here to add to subsystem

MSS FIC_0 - Fabric Master Subsystem
drag and drop here to add to subsystem

MSS FIC_1 - MSS Master Subsystem
drag and drop here to add to subsystem

MSS FIC_1 - Fabric Master Subsystem
drag and drop here to add to subsystem

MSS DDR FIC Subsystem

Configure	Quantity	Name
	1	MSS_DDR_RAM

MSS Peripherals

Configure	Enable	Name
	<input checked="" type="checkbox"/>	MM_UART_0
	<input checked="" type="checkbox"/>	MM_UART_1
	<input checked="" type="checkbox"/>	MSS_I2C_0
	<input checked="" type="checkbox"/>	MSS_I2C_1
	<input checked="" type="checkbox"/>	MSS_SPI_0
	<input checked="" type="checkbox"/>	MSS_SPI_1

To move a peripheral from one subsystem to another, drag it from its present location and drop it onto the desired subsystem.
You cannot drag and drop onto MSS Peripherals.

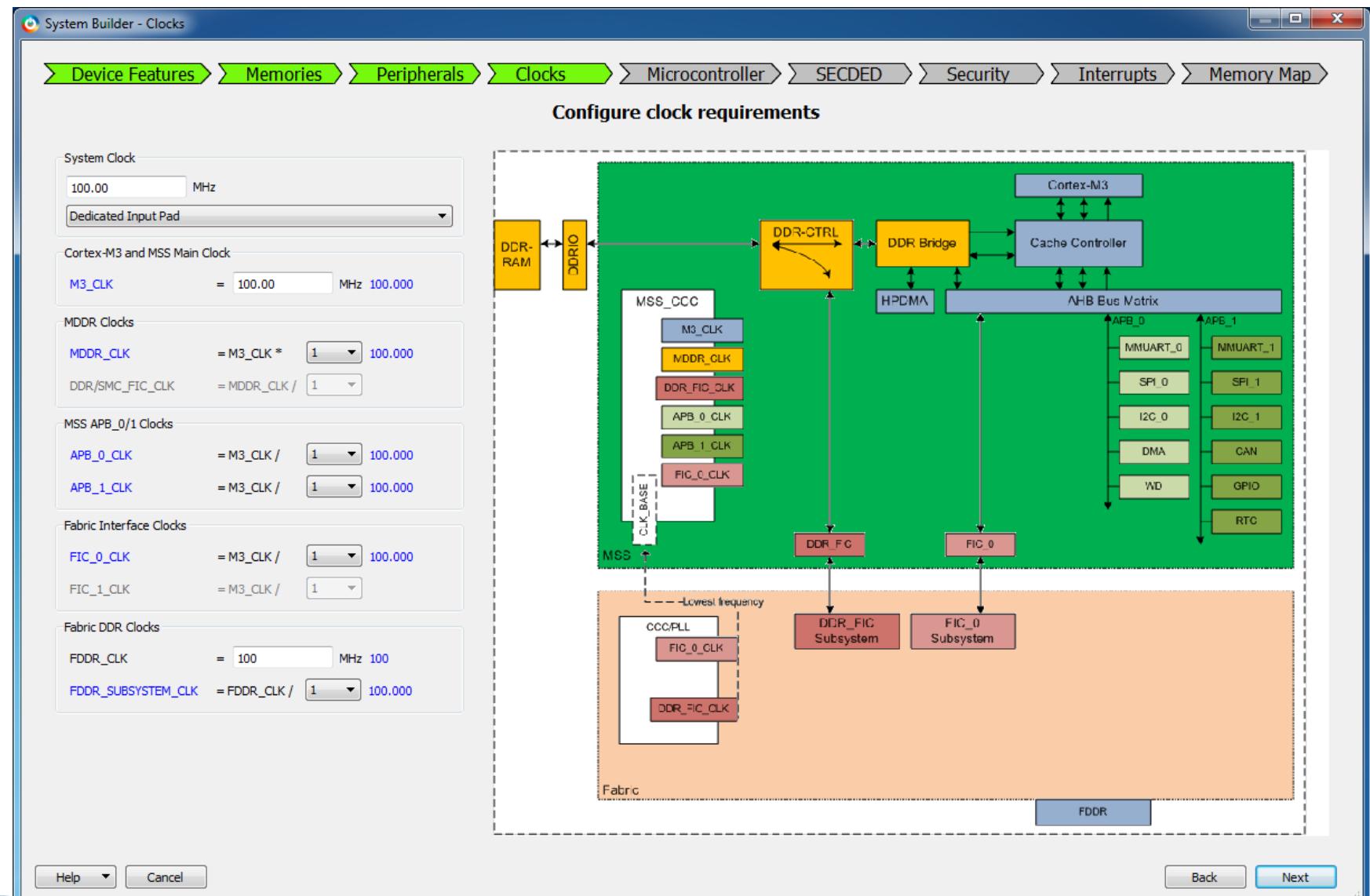
Masters are in bold and blue.

Help Cancel Back Next

System Builder – Clock Settings Page

- Specify the System Clock
 - System Builder automatically instantiates the required PLL (if needed)
- System Clock options include:
 - FPGA Fabric Input
 - Dedicated Input Pad
 - On-chip 1 MHz RC Oscillator
 - On-chip 25/50 MHz RC Oscillator
 - External Main Crystal Oscillator (Ceramic Resonator 0.5MHz to 20MHz)
 - External Main Crystal Oscillator (Crystal 32KHz - 20MHz)
 - External Main Crystal Oscillator (RC Network 32KHz - 4MHz)
- Specify system clock frequencies
 - Cortex-M3 and MSS Main Clock
 - MDDR Clocks
 - MSS APB_0/1
 - Fabric Interface Clocks (FIC_0_CLK and FIC_1_CLK)
 - Fabric DDR Clocks

System Builder – Clock Settings Page



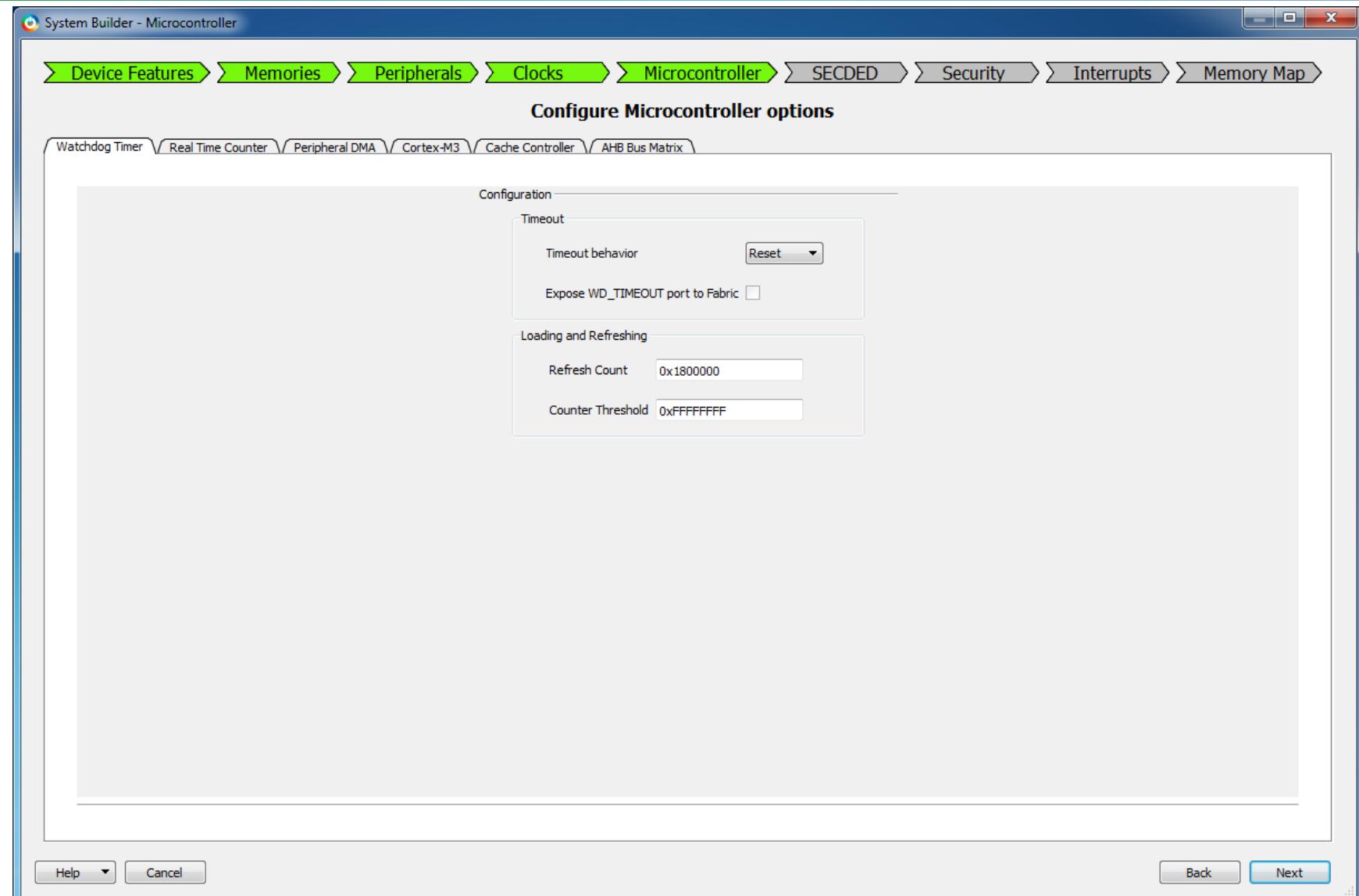
System Builder – Microcontroller Options Page

- Configuration options for:

- Watchdog Timer
- Real Time Counter
- Peripheral DMA
- Cortex-M3
- Cache Controller
- AHB Bus Matrix

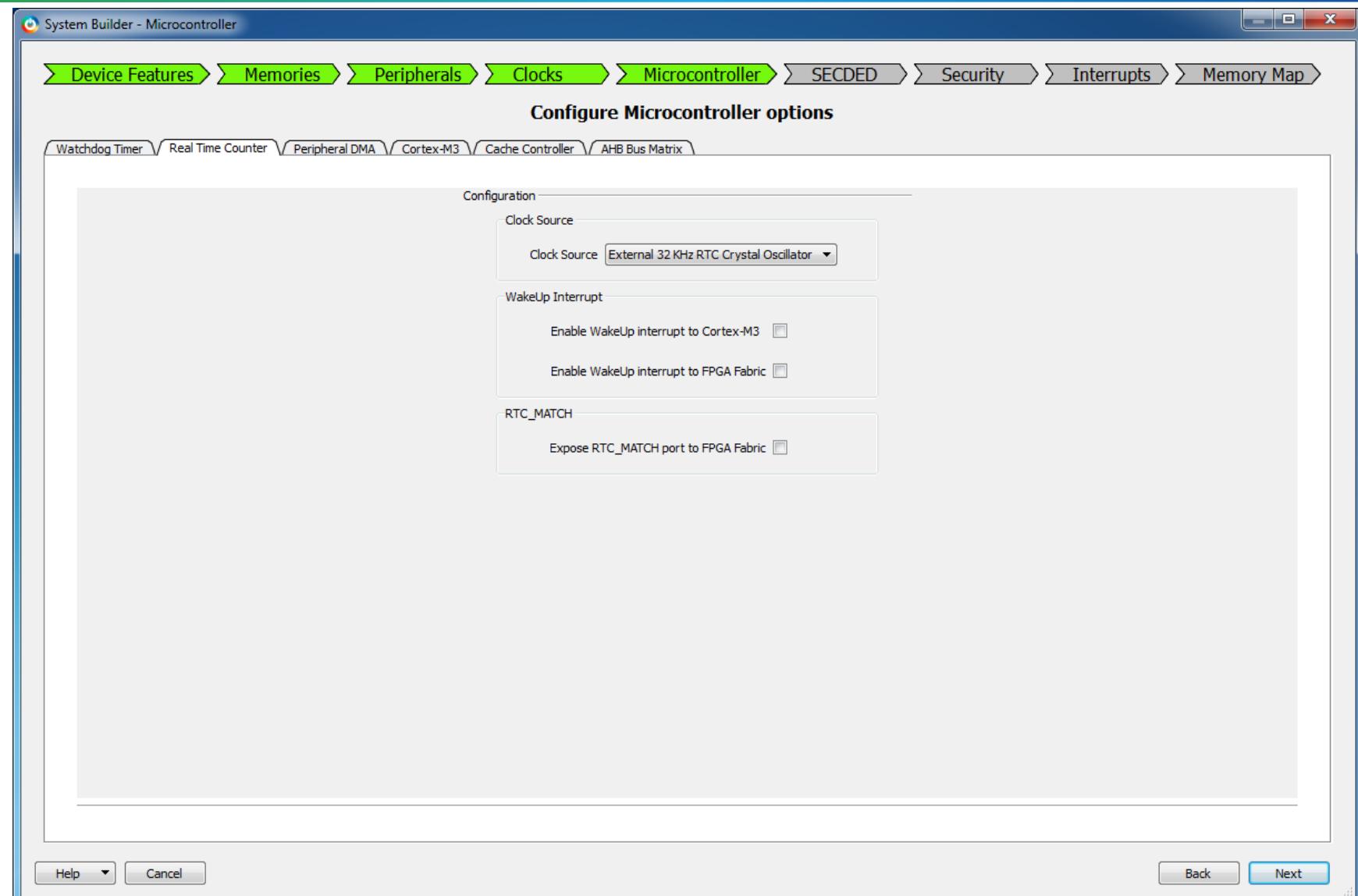
System Builder – Microcontroller Options

Watchdog Timer



System Builder – Microcontroller Options

Real Time Counter



System Builder – Microcontroller Options

Peripheral DMA

System Builder - Microcontroller

Device Features > Memories > Peripherals > Clocks > Microcontroller > SECDED > Security > Interrupts > Memory Map

Configure Microcontroller options

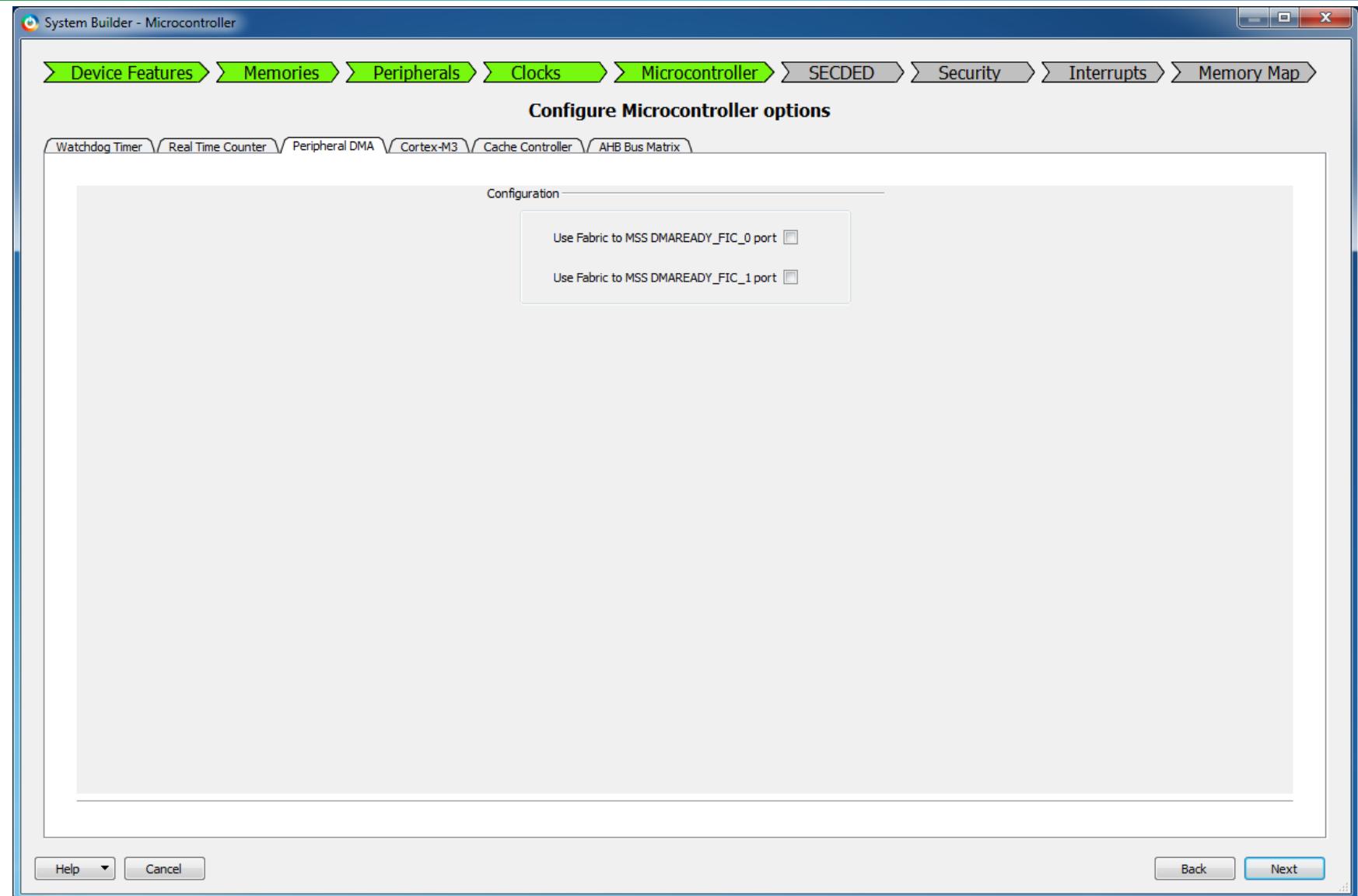
Watchdog Timer < Real Time Counter < Peripheral DMA < Cortex-M3 < Cache Controller < AHB Bus Matrix

Configuration

Use Fabric to MSS DMAREADY_FIC_0 port

Use Fabric to MSS DMAREADY_FIC_1 port

Help Back Next Cancel



System Builder – Microcontroller Options

Cortex-M3

System Builder - Microcontroller

Device Features > Memories > Peripherals > Clocks > Microcontroller > SECDED > Security > Interrupts > Memory Map

Configure Microcontroller options

Watchdog Timer / Real Time Counter / Peripheral DMA / Cortex-M3 / Cache Controller / AHB Bus Matrix

Configuration

Use Memory Protection Unit

Sys Tick Timer

Calibration Register 0x2000000

STCLK = M3_CLK /

Events

Expose RXEV port to the FPGA Fabric

Expose TXEV port to the FPGA Fabric

System Power Management

Expose SLEEPING port to the FPGA Fabric

Expose SLEEPDEEP port to the FPGA Fabric

Expose SLEEPHOLD* ports to the FPGA Fabric

Trace Port Interface Unit

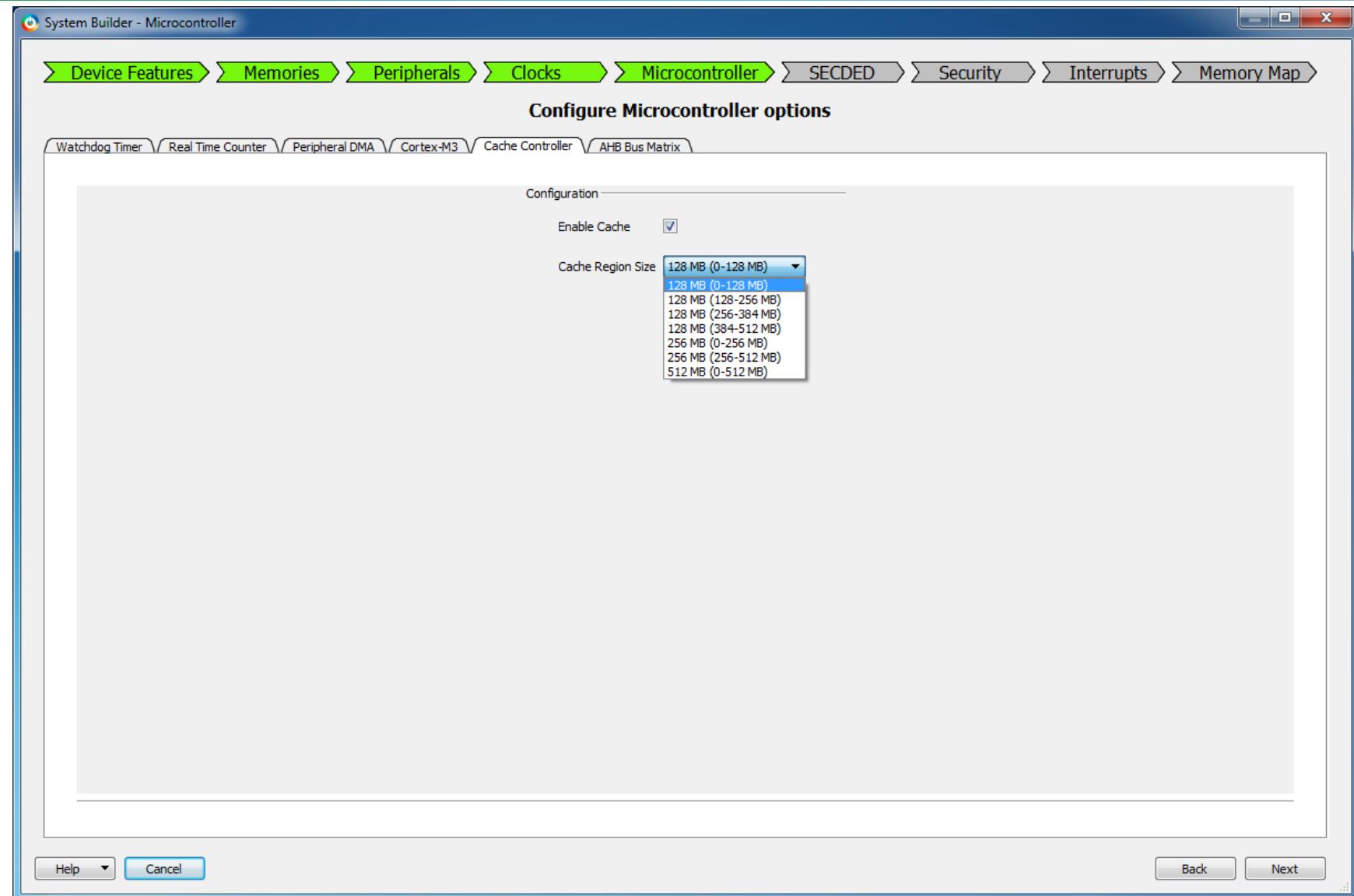
TRACECLK is M3_CLK divided by 4

Expose TRACE* ports to the FPGA Fabric

Help Back Next Cancel

System Builder – Microcontroller Options

Cache Controller



System Builder – Microcontroller Options

AHB Bus Matrix

System Builder - Microcontroller

Device Features > Memories > Peripherals > Clocks > Microcontroller > SECDED > Security > Interrupts > Memory Map

Configure Microcontroller options

Watchdog Timer / Real Time Counter / Peripheral DMA / Cortex-M3 / Cache Controller / AHB Bus Matrix

Configuration

Remapping

Remapped Region to location 0x00000000 of Cortex-M3 ID Code space

eNVM eSRAM MDDR

Remap eNVM to location 0x00000000 of Fabric Master space

eNVM Remap Region Size: 256KB

eNVM Remap Base Address (Cortex-M3): 0x00000000

eNVM Remap Base Address (Fabric Master): 0x00000000

Arbitration

Fixed Priority (2) Weight for Cortex-M3 IC Master: 1

Fixed Priority (3) Weight for Cortex-M3 S Master: 1

Fixed Priority (4) Weight for System Controller Master: 1

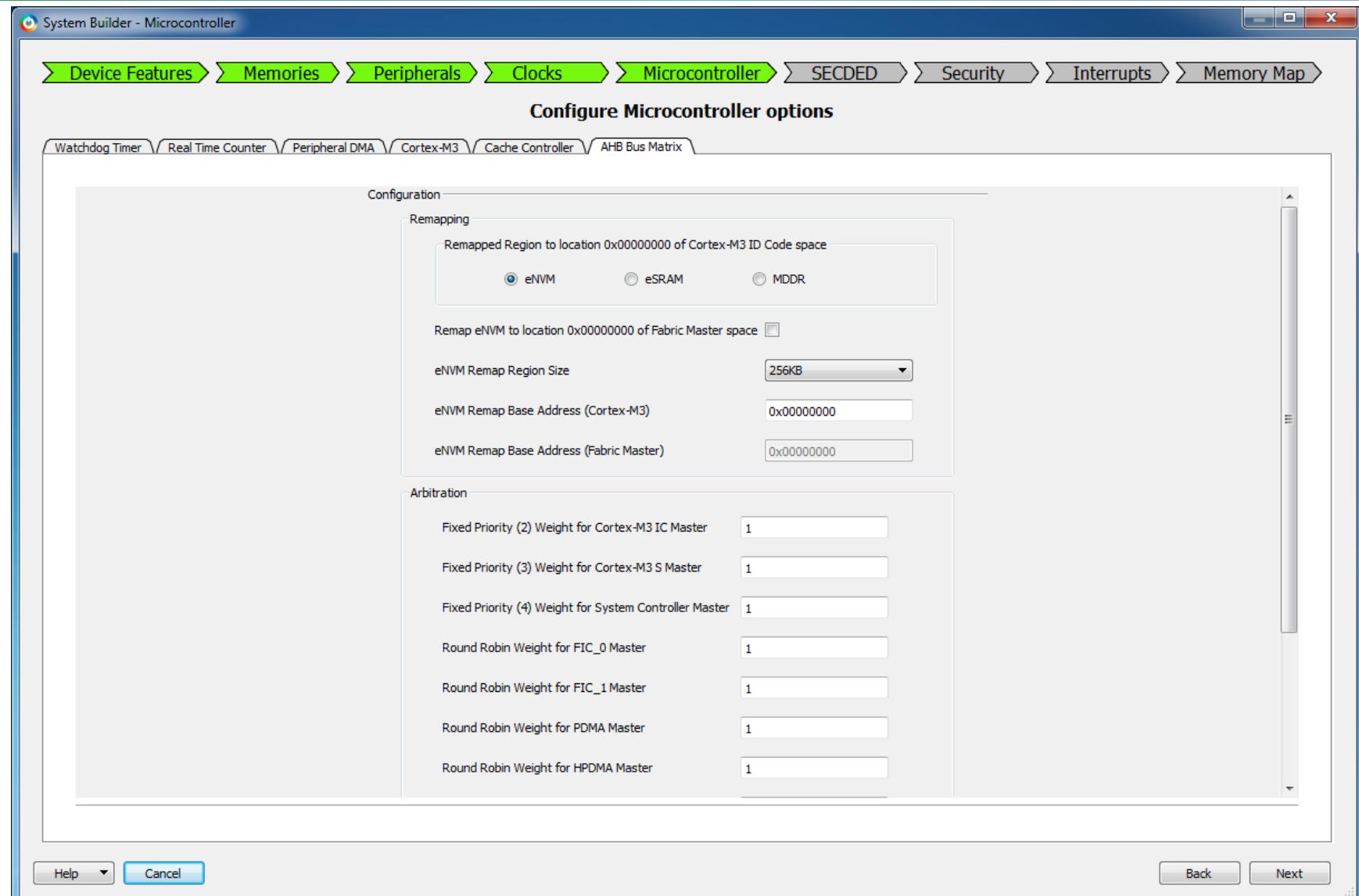
Round Robin Weight for FIC_0 Master: 1

Round Robin Weight for FIC_1 Master: 1

Round Robin Weight for PDMA Master: 1

Round Robin Weight for HPDMA Master: 1

Help ▾ Cancel Back Next



System Builder – SECDED Options Page

- Configure your Single Error Correction / Double Error Detection logic
 - Memories protected by SECDED are:
 - Cache
 - eSRAM_0 and eSRAM_1
 - USB internal memory
 - Ethernet MAC transmit and receive internal FIFOs
 - CAN controller internal RAMs
 - PCIe FIFOs
 - External DDR memories
 - Configure EDAC Interrupts
 - None, 1-bit error, 2-bit error, 1-bit and 2-bit errors
- Expose EDAC_ERROR bus signal to the fabric

System Builder – SECDED Options Page

System Builder - SECDED

Device Features > Memories > Peripherals > Clocks > Microcontroller > SECDED > Security > Interrupts > Memory Map

Configure Single Error Correct / Double Error Detect (SECDED) options

SECDED

Configuration

EDAC Errors

Expose EDAC_ERROR bus

eSRAM_0

Enable EDAC Enable EDAC Interrupt(s)

eSRAM_1

Enable EDAC Enable EDAC Interrupt(s)

Cache

Enable EDAC Enable EDAC Interrupt(s)

Ethernet TX RAM

Enable EDAC Enable EDAC Interrupt(s)

Ethernet RX RAM

Enable EDAC Enable EDAC Interrupt(s)

USB

Enable EDAC Enable EDAC Interrupt(s)

CAN

Enable EDAC Enable EDAC Interrupt(s)

Configure SECDED options

Help Cancel Back Next

The screenshot shows the 'SECDED' configuration page within the System Builder software. The main title is 'Configure Single Error Correct / Double Error Detect (SECDED) options'. The left sidebar lists components: eSRAM_0, eSRAM_1, Cache, Ethernet TX RAM, Ethernet RX RAM, USB, and CAN. Each component has two sections: 'Enable EDAC' (checkbox) and 'Enable EDAC Interrupt(s)' (dropdown menu). A dropdown menu for 'eSRAM_0' is open, showing options: None, 1-bit error, 2-bit error, and 1-bit and 2-bit errors. A large watermark 'Configure SECDED options' is overlaid across the center of the page. At the bottom, there are 'Help', 'Cancel', 'Back', and 'Next' buttons.

System Builder – Security Options Page

System Builder - Security

Device Features > Memories > Peripherals > Clocks > Microcontroller > SECDED > Security > Interrupts > Memory Map

Configure Security options

Master/Slave MSS to Fabric Memory Map Configuration Registers

Master to Slaves Read/Write Access

	eSRAM0 [MS0]	eSRAM1 [MS1]	eNVM0 [MS2]	FIC_0 [MS4]	AHB2AHB [MS5]	DDR Bridge [MS6]
IC Bus [MM0]	R <input checked="" type="checkbox"/> Read	R <input checked="" type="checkbox"/> Read	R <input checked="" type="checkbox"/> Read	--	--	<input checked="" type="checkbox"/> Read
D-BUS [MM1]	RW <input checked="" type="checkbox"/> Write	--	<input checked="" type="checkbox"/> Write			
S-BUS [MM2]	RW	RW	RW	RW	RW	--
FIC_0 [MM4]	RW <input checked="" type="checkbox"/> Read	RW <input checked="" type="checkbox"/> Read	RW <input checked="" type="checkbox"/> Read	RW	RW	<input checked="" type="checkbox"/> Read
FIC_1 [MM5]	RW <input checked="" type="checkbox"/> Write	RW <input checked="" type="checkbox"/> Write	RW <input checked="" type="checkbox"/> Write	RW	RW	<input checked="" type="checkbox"/> Write
HPDMA [MM3]	RW <input checked="" type="checkbox"/> Read	RW <input checked="" type="checkbox"/> Read	RW <input checked="" type="checkbox"/> Read	RW	--	<input checked="" type="checkbox"/> Read
MAC_M [MM6]	RW <input checked="" type="checkbox"/> Write	RW <input checked="" type="checkbox"/> Write	-- <input checked="" type="checkbox"/> Write	RW	--	<input checked="" type="checkbox"/> Write
PDMA [MM7]	RW	RW	RW	RW	RW	--
USB [MM8]	RW	RW	--	RW	RW	--

Show Special Sectors AHB2AHB [MS5]

Configure Security options

To protect these advanced security bits with user pass key 1, you must configure the Security Policy Manager, specify user key set 1, and program the security feature.
If the security programming feature is enabled for programming, then you must reprogram the security features if you modify the advanced security bits

Help Cancel Back Next

System Builder – Interrupts Page

The screenshot shows the 'System Builder - Interrupts' window. At the top, a breadcrumb navigation bar lists: Device Features > Memories > Peripherals > Clocks > Microcontroller > SECDED > Security > Interrupts > Memory Map. Below the navigation bar, a subtitle reads: 'Interrupt connections generated from attached peripherals and the processor'. A table is displayed with columns: Processor Interrupt, Instance Name, Trigger Signals, and Lock. The table is currently empty. At the bottom of the window, there are buttons for Help, Cancel, Back, Next, and a Microsemi logo.

Processor Interrupt Instance Name Trigger Signals Lock

Displays interrupts in system

System Builder – Memory Map Page

System Builder - Memory Map

Device Features > Memories > Peripherals > Clocks > Microcontroller > SECDED > Security > Interrupts > Memory Map

Memory Map for peripherals attached to the processor

Select Bus to View or Assign Peripheral(s)

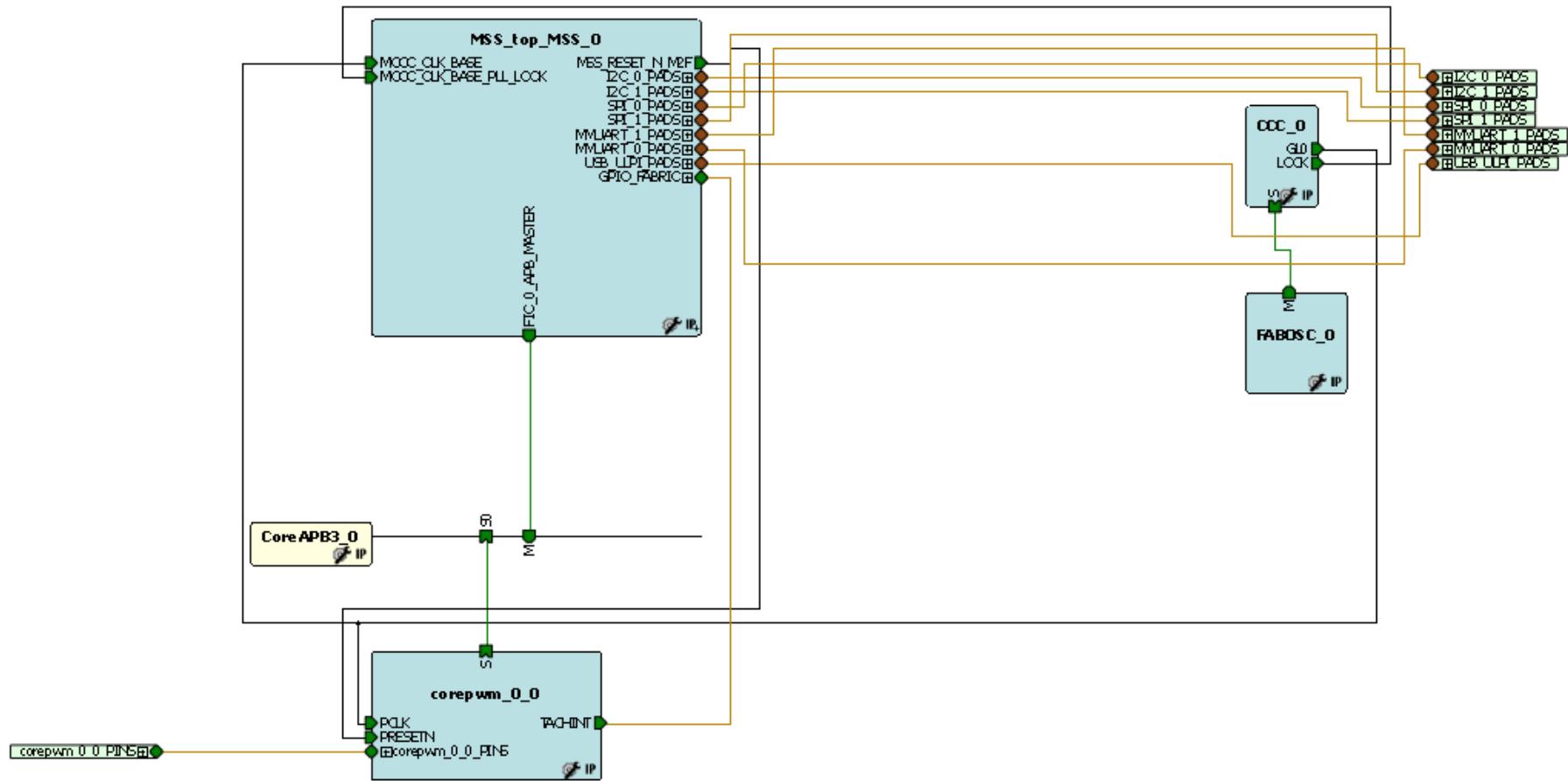
Assign peripherals to addresses on bus:

Address	Peripheral
0x40020800	M2S_system_MSS_0:MDDR_APB_SLAVE
0x40021000	FABDDR_0:APB_SLAVE
0x40028000	M2S_system:SDIF0_APBmslave

Displays system memory map

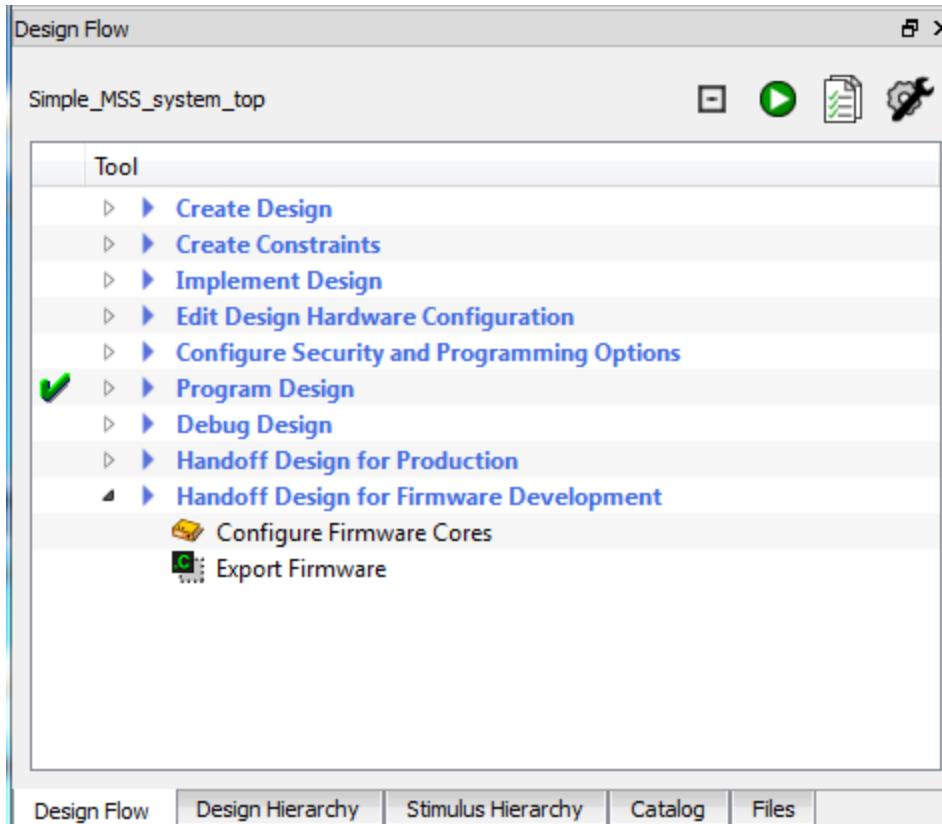
Help Back Cancel Finish

SmartFusion2 System



Firmware Drivers and Sample Projects

- Export Firmware drivers and Sample projects from the Libero Design Flow tab
 - Firmware and sample projects: right-click Configure Firmware cores
 - Template project: right-click Export Firmware



Firmware View Within Libero SoC

Configure Firmware Cores

- Generate firmware drivers for enabled peripherals

The screenshot shows the Libero SoC DESIGN_FIRMWARE interface with a table of configured firmware cores. The columns are: Generate, Instance Name, Core Type, Version, and Compatible Hardware Instance. The rows list various drivers for SmartFusion2_MSS hardware.

	Generate	Instance Name	Core Type	Version	Compatible Hardware Instance
1	<input checked="" type="checkbox"/>	SmartFusion2_CMSIS_0	SmartFusion2_CMSIS	1.3.2	SmartFusion2_Simple_MSS_MSS
2	<input checked="" type="checkbox"/>	SmartFusion2_MSS_GPIO_Driver_0	SmartFusion2_MSS_GPIO_Driver	1.2.2	SmartFusion2_Simple_MSS_MSS:GPIO
3	<input checked="" type="checkbox"/>	SmartFusion2_MSS_HPDMA_Driver_0	SmartFusion2_MSS_HPDMA_Driver	1.2.1	SmartFusion2_Simple_MSS_MSS
4	<input checked="" type="checkbox"/>	SmartFusion2_MSS_MMUART_Driver_0	SmartFusion2_MSS_MMUART_Driver	1.2.3	SmartFusion2_Simple_MSS_MSS:MMUART_0
5	<input checked="" type="checkbox"/>	SmartFusion2_MSS_NVM_Driver_0	SmartFusion2_MSS_NVM_Driver	1.3.100	SmartFusion2_Simple_MSS_MSS
6	<input checked="" type="checkbox"/>	SmartFusion2_MSS_System_Services_Driver_0	SmartFusion2_MSS_System_Services_Driver	1.3.2	SmartFusion2_Simple_MSS_MSS
7	<input checked="" type="checkbox"/>	SmartFusion2_MSS_Timer_Driver_0	SmartFusion2_MSS_Timer_Driver	1.2.2	SmartFusion2_Simple_MSS_MSS

- Create sample projects

The screenshot shows the Libero SoC DESIGN_FIRMWARE interface with a table of configured firmware cores. A context menu is open over the second row (SmartFusion2_MSS_GPIO_Driver_0). The menu items are: Disable Generation, Show Details..., Generate Sample Project, Cortex-M3, IAR Embedded Workbench, Keil-MDK, SoftConsole, and Simple Blink. The 'Simple Blink' option is highlighted.

	Generate	Instance Name	Core Type	Version	Compatible Hardware Instance
1	<input checked="" type="checkbox"/>	SmartFusion2_CMSIS_0	SmartFusion2_CMSIS	1.3.2	SmartFusion2_Simple_MSS_MSS
2	<input checked="" type="checkbox"/>	SmartFusion2_MSS_GPIO_Driver_0	SmartFusion2_MSS_GPIO_Driver	1.2.2	SmartFusion2_Simple_MSS_MSS:GPIO
3	<input checked="" type="checkbox"/>	SmartFusion2_MSS_HPDMA_Driver_0	SmartFusion2_MSS_HPDMA_Driver	1.2.1	SmartFusion2_Simple_MSS_MSS
4	<input checked="" type="checkbox"/>	SmartFusion2_MSS_MMUART_Driver_0	IAR Embedded Workbench > MMUART_Driver	1.2.3	SmartFusion2_Simple_MSS_MSS:MMUART_0
5	<input checked="" type="checkbox"/>	SmartFusion2_MSS_NVM_Driver_0	Keil-MDK > NVM_Driver	1.3.100	SmartFusion2_Simple_MSS_MSS
6	<input checked="" type="checkbox"/>	SmartFusion2_MSS_System_Services_Driver_0	SoftConsole > Simple Blink	1.3.2	SmartFusion2_Simple_MSS_MSS
7	<input checked="" type="checkbox"/>	SmartFusion2_MSS_Timer_Driver_0	SmartFusion2_MSS_Timer_Driver	1.2.2	SmartFusion2_Simple_MSS_MSS

SmartFusion2 IP & Solutions

- Leverage over 50 IP Cores in FPGA fabric
 - Peripherals: GPIO, I2C, SPI, Timers, UARTs, PWM
 - Cryptography: FIPS 140-2 compliant cores
 - Communications: LPC, AXI, AXI to AHB
 - DSP: RS Encode/Decode, FIR and FFT using Math Blocks
 - SERDES & Protocols: JESD204, EPICS, RGMII
- Next Generation Solutions
 - Micro Power Manager (MPM)
 - Motor Control

Libero SoC IP Core Catalog

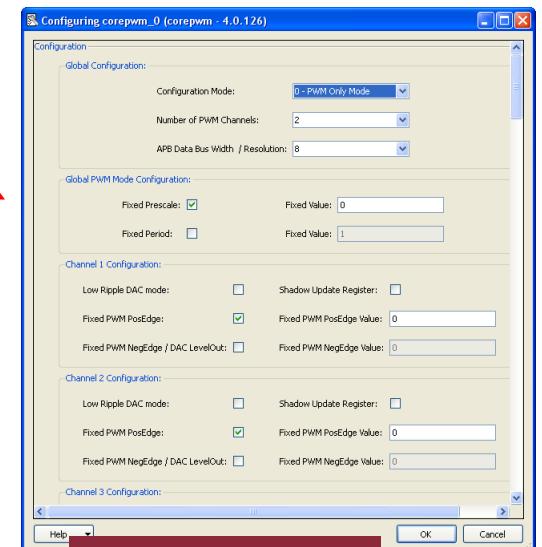
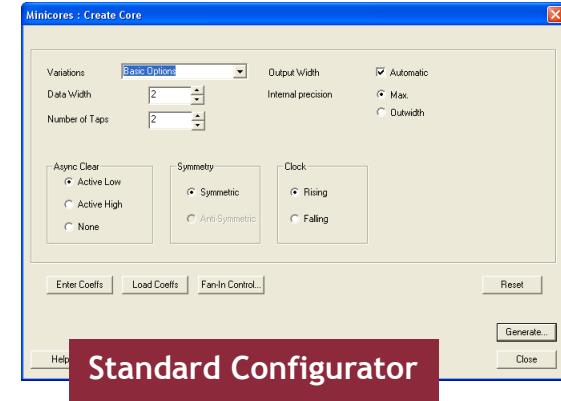
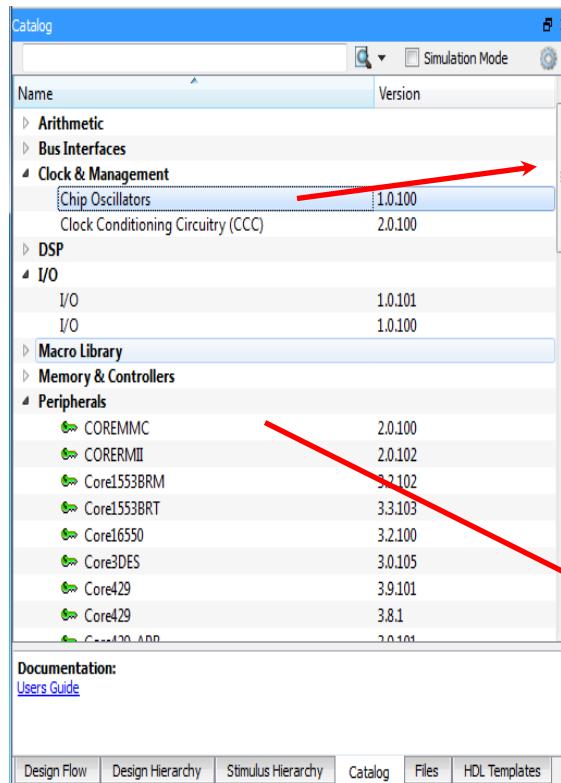
- Library of Proven Configurable Core Functions

- Microsemi Macros

- Quick-find

- Intuitive Configuration

- Drag and drop to Canvas

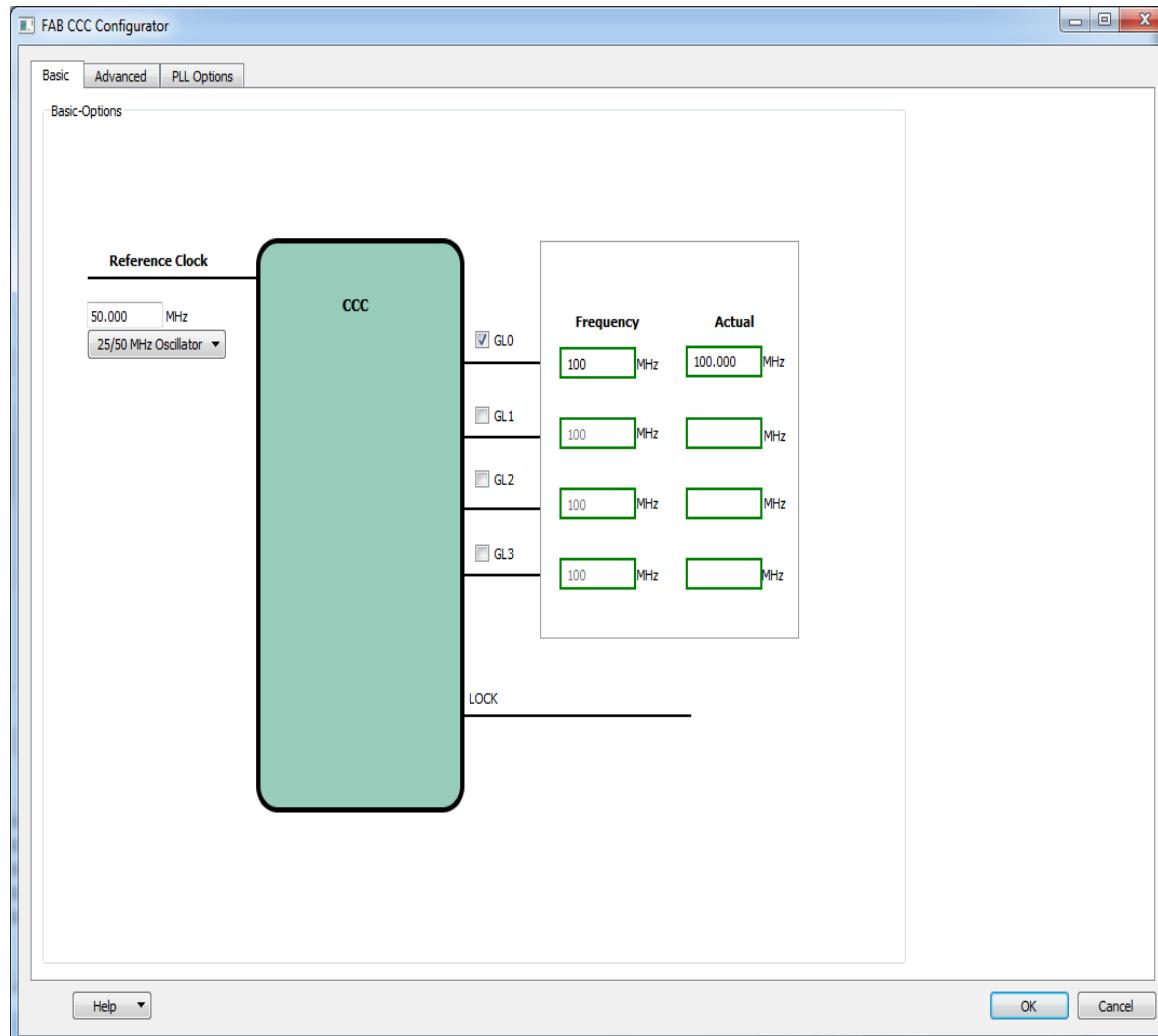


IP Cores for SmartFusion2 in Libero IP Catalog

- Processors
 - CoreABC
- AMBA Interfaces
 - CoreAXI
 - CoreAHBLTOAXI
 - CoreAXITOAHBL
 - CoreAHBLite
 - CoreAPB3
 - CoreAHB2APB3
- Subsystem Cores
 - CoreAhbSram
 - CoreAPBSRAM
 - CoreFIFO
 - CoreGPIO
 - CoreI2C
 - CoreInterrupt
 - CoreMemCtrl
 - CorePWM
 - CoreSDR
 - CoreSMBus
 - CoreTimer
 - CoreUART, CoreUARTapb
 - CoreWatchdog

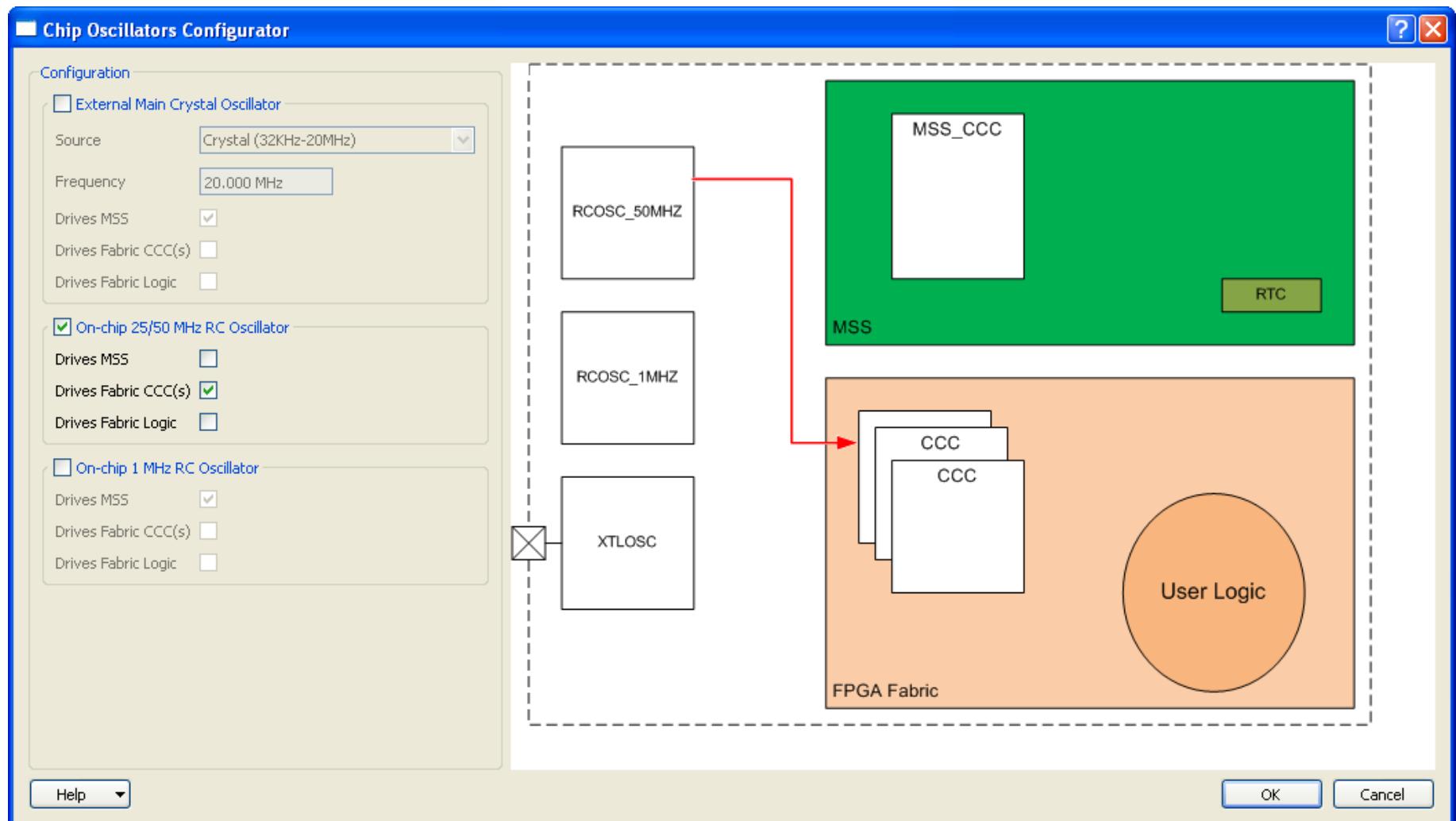


Fabric CCC Configurator



CCC = Clock Configuration Circuit

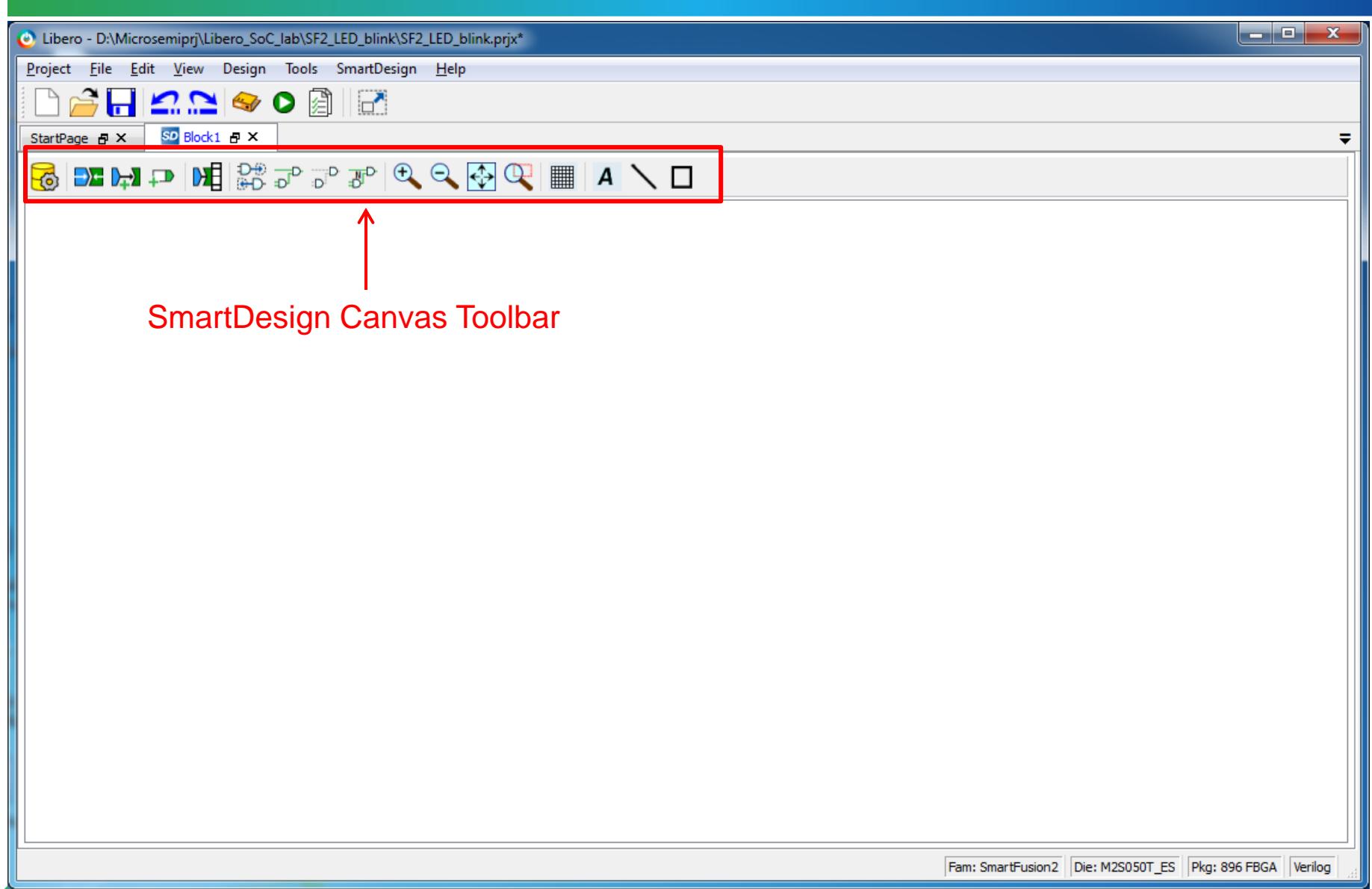
Chip Oscillators Configuration



What is SmartDesign?

- Powerful Block-based Visual Design Creation Tool
 - Instantiate blocks from a variety of sources
 - DirectCore IP, SmartGen, User HDL, Companion Cores, Microsemi library cells, etc.
 - Supported for all platforms
- Simple and Intuitive Design Creation
 - Auto Connect
 - Fast manual connectivity between blocks
 - Hierarchical design support
- Design Rule Checking (DRC)
 - Checks rules to guarantee correct by construction design
 - Connectivity errors
 - Configuration errors
 - Special silicon rules
- SoC Features
 - Memory Map Configuration Dialog
 - Testbench and Bus Functional Model (BFM) script generation

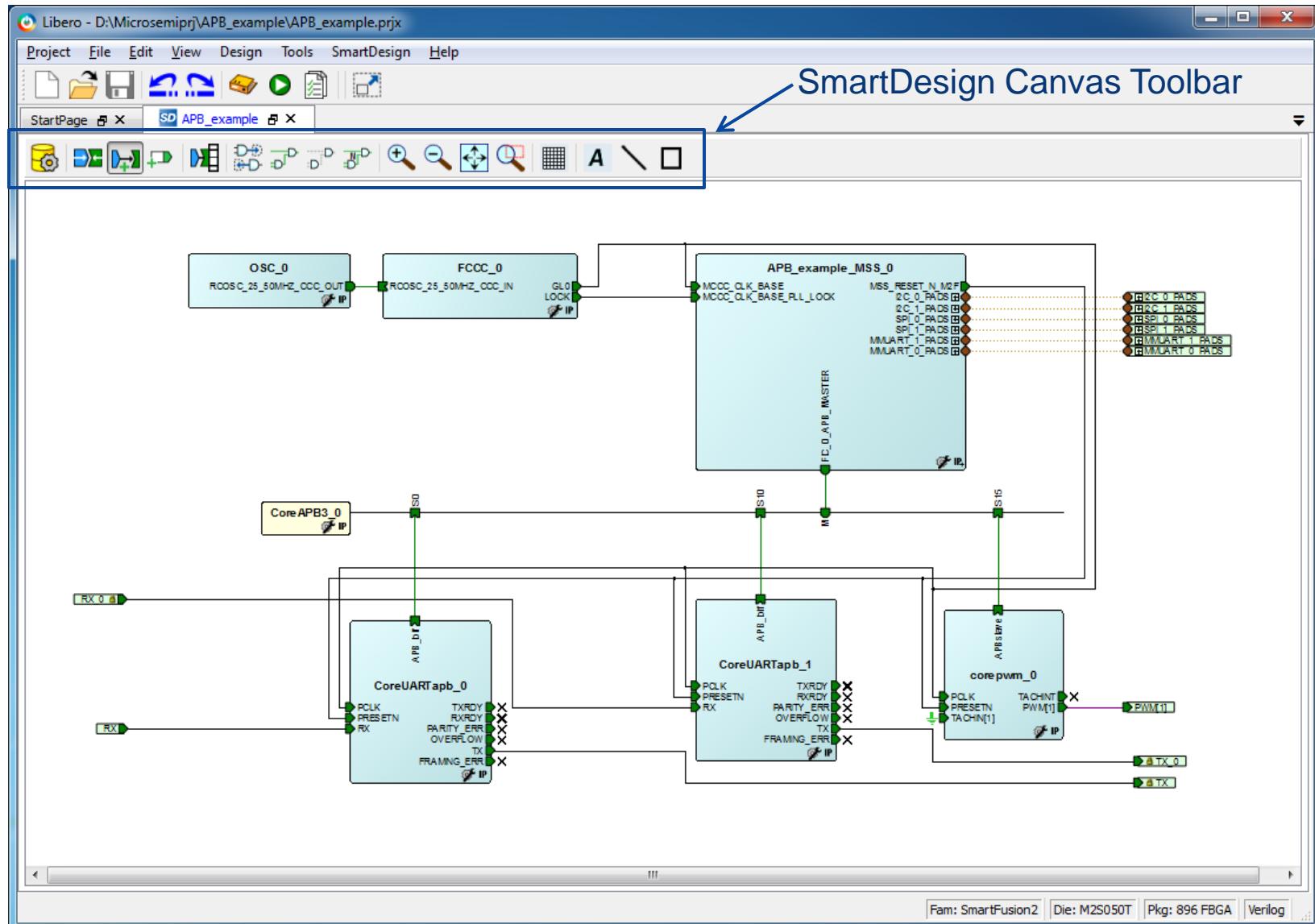
SmartDesign Canvas



SmartDesign Canvas Features

- Instance pins are displayed on canvas
- Connections are shown using nets
 - Displaying of Nets is optional
 - Selective enabling / disabling of showing nets
- Drag and Drop directly from the Catalog into the Canvas
- All Design Operations Available in the Canvas
 - Connect / Disconnect
 - Promote To Top
 - Tie Low / Tie High / Tie Constant / Inversion
 - Float
 - Split (if bus)
 - Group

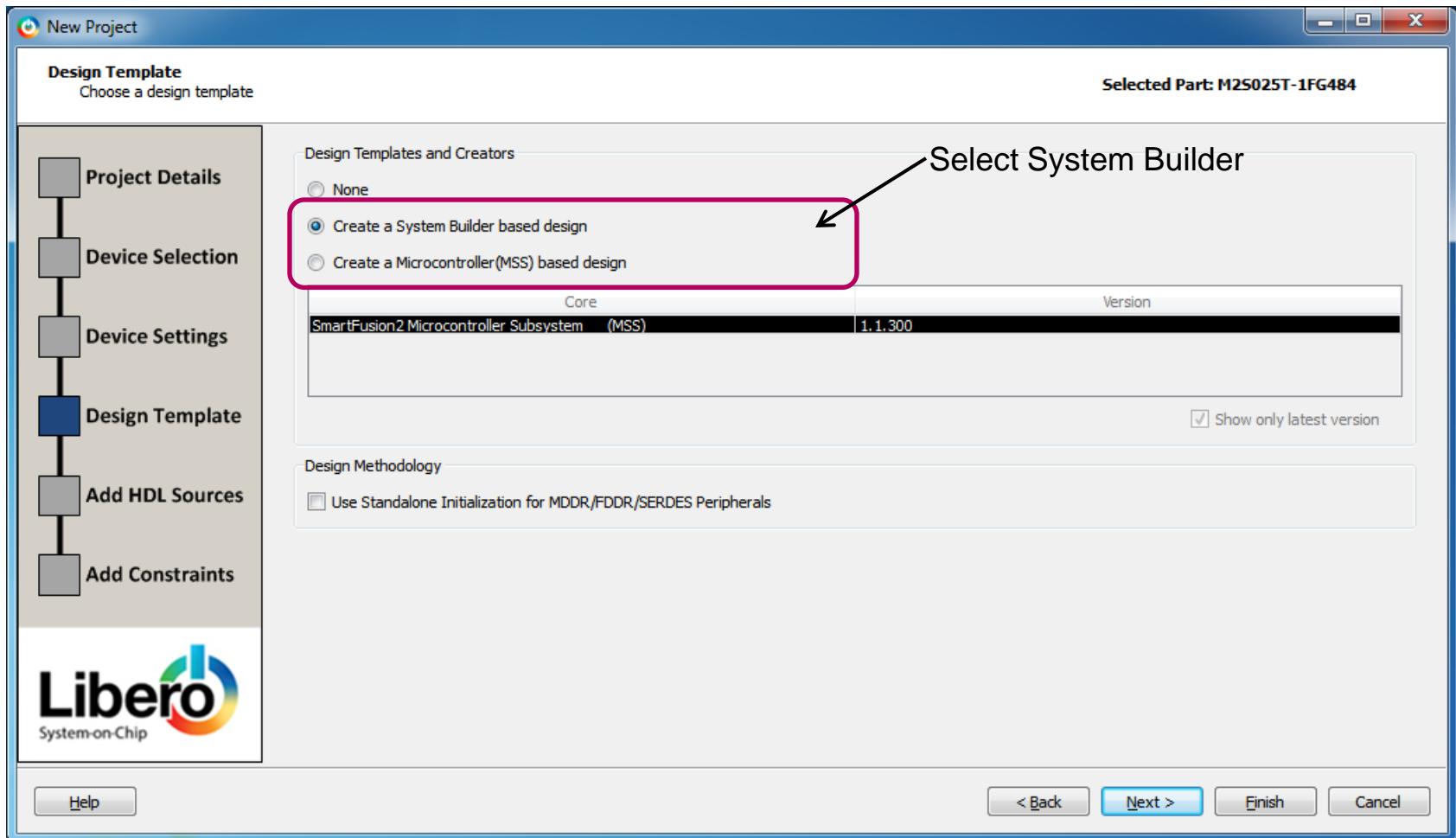
Complete Design



ARM Cortex-M3 Lab

- This lab demonstrates how to implement a basic SmartFusion2 MSS configuration with the GPIO block and MMUART_0 enabled using the SmartFusion2 MSS configurator.
 - The SmartFusion2 GPIO[8] will be configured as an input and GPIO[1:0] will be configured as outputs
 - The outputs drive LEDs on one of the SmartFusion2 kits; GPIO[8] will be connected to one of the switches on the kit
- Complete steps 1 - 5:
 - Create a Libero SoC project
 - Use System Builder to configure the SmartFusion2 MSS and generate the design
 - Generating sample firmware projects from the Firmware catalog
 - Synthesizing the SmartFusion2 design with Synplify Pro ME
 - Import a PDC pin constraint file and run layout
 - Generate a programming file

This Lab Uses System Builder



GPIO Configuration

MSS GPIO Configurator

Configuration

Set/Reset Definition

GPIO_31_24 Reset Source	SYSREG (MSS_GPIO_31_24_SOFT_RESET)	Reset State	1
GPIO_23_16 Reset Source	SYSREG (MSS_GPIO_23_16_SOFT_RESET)	Reset State	1
GPIO_15_8 Reset Source	SYSREG (MSS_GPIO_15_8_SOFT_RESET)	Reset State	1
GPIO_7_0 Reset Source	SYSREG (MSS_GPIO_7_0_SOFT_RESET)	Reset State	1

GPIO Assignment

GPIO ID	Direction	Package Pin	Connectivity
GPIO_0	Output		FABRIC_A
GPIO_1	Output		FABRIC_A
GPIO_2	Not Used		IO_A
GPIO_3	Not Used		IO_A
GPIO_4	Not Used		IO_A
GPIO_5	Not Used		IO_A
GPIO_6	Not Used		IO_A
GPIO_7	Not Used		IO_A
GPIO_8	Input		FABRIC_A
GPIO_9	Not Used		IO_A
...			

Connectivity Preview

GPIO_31

MSS

FPGA Fabric

Click on a signal row to see the preview

OK Cancel

MSS CCC Configurator

MSS Clock Conditioning Circuitry Configurator

System Clocks Advanced Options

Clock Source

CLK_BASE: 100 MHz
 Monitor FPGA Fabric PLL Lock (CLK_BASE_PLL_LOCK)

Cortex-M3 and MSS Main Clock

M3_CLK: 100.000 MHz 100.000 MHz

MDDR Clocks

MDDR_CLK = M3_CLK * 2
DDR_SMC_FIC_CLK = MDDR_CLK / 1

MSS APB_0/1 Sub-busses Clocks

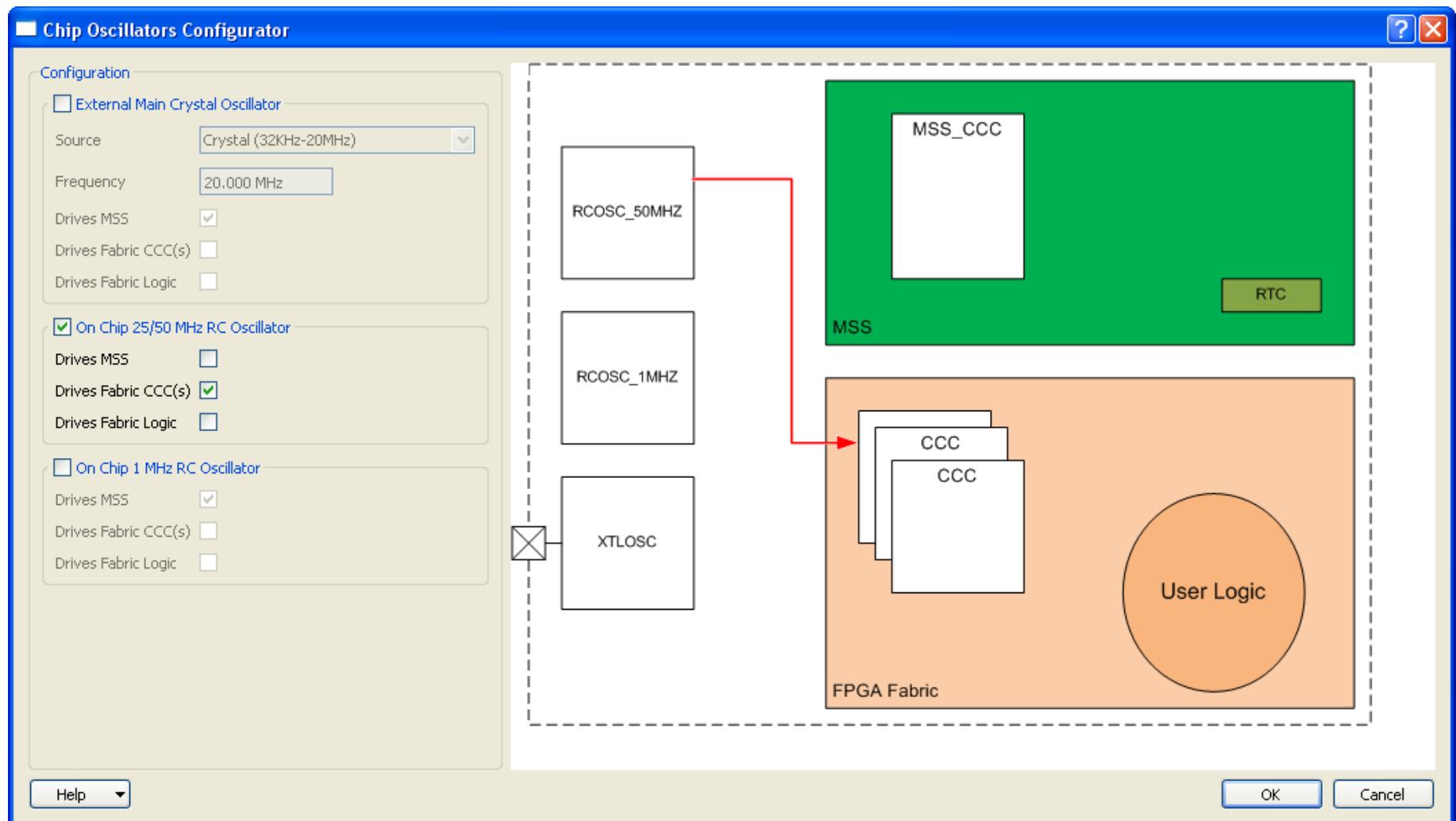
APB_0_CLK = M3_CLK / 4 25.000 MHz
 APB_1_CLK = M3_CLK / 4 25.000 MHz

FPGA Fabric Interface Clocks

FIC_0_CLK = M3_CLK / 1
FIC_1_CLK = M3_CLK / 1

OK Cancel

OSC Setup



Hands-on Lab

Complete Steps 1 - 5

Embedded Design

Embedded Software IDE Options

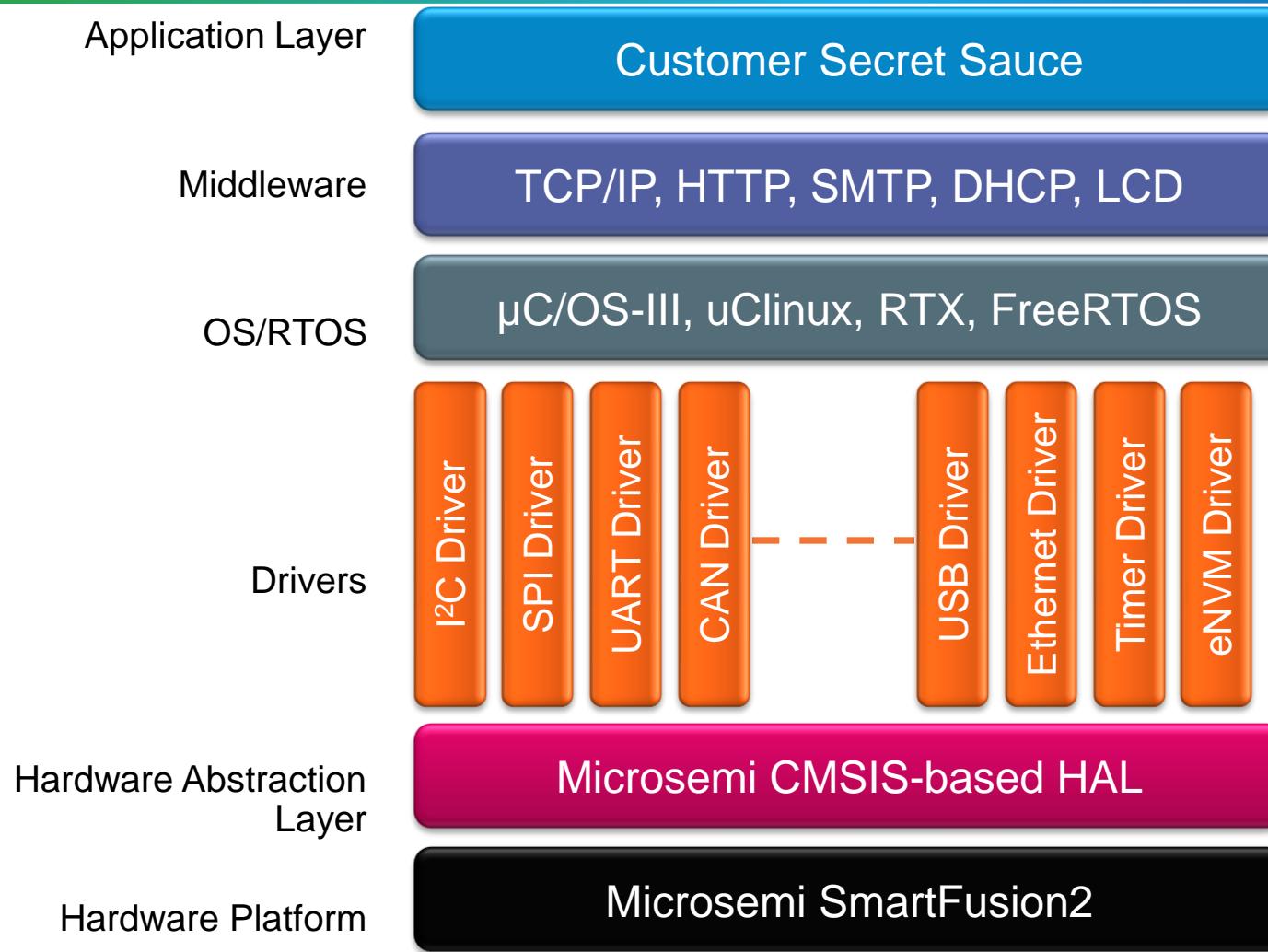
- SoftConsole Eclipse-based IDE
 - GNU C/C++ compiler, GDB Debugger
 - Free download from Microsemi SoC website
- Keil MDK – Microcontroller Development Kit
 - Combines the ARM C/C++ Compiler
 - With popular µVision4 debug environment
- IAR Embedded Workbench
 - First standard compiler for industrial and 8 bit processors
- Lauterbach
 - Low cost µTrace system specifically targets the Cortex-M family

Embedded Design Options

 CMSIS COMPLIANT <small>ARM® Cortex® Microcontroller Software Interface Standard</small>	 Microsemi.	 KEIL™ An ARM® Company	 IAR SYSTEMS
Software IDE	SoftConsole	Keil MDK	IAR Embedded Workbench®
Website	www.microsemi.com/soc	www.keil.com	www.iar.com
Free versions from SoC Products Group	Free with Libero SoC	32 K code limited	32 K code limited
Available from Vendor	N/A	Full version	Full version
Compiler	GNU GCC	RealView C/C++	IAR ARM Compiler
Debugger	GDB debug	Vision Debugger	C-SPY® Debugger
Instruction Set Simulator	No	Vision Simulator	Yes
Debug Hardware	FlashPro4	ULINK2® or ULINK-ME	J-LINK™ or J-LINK Lite

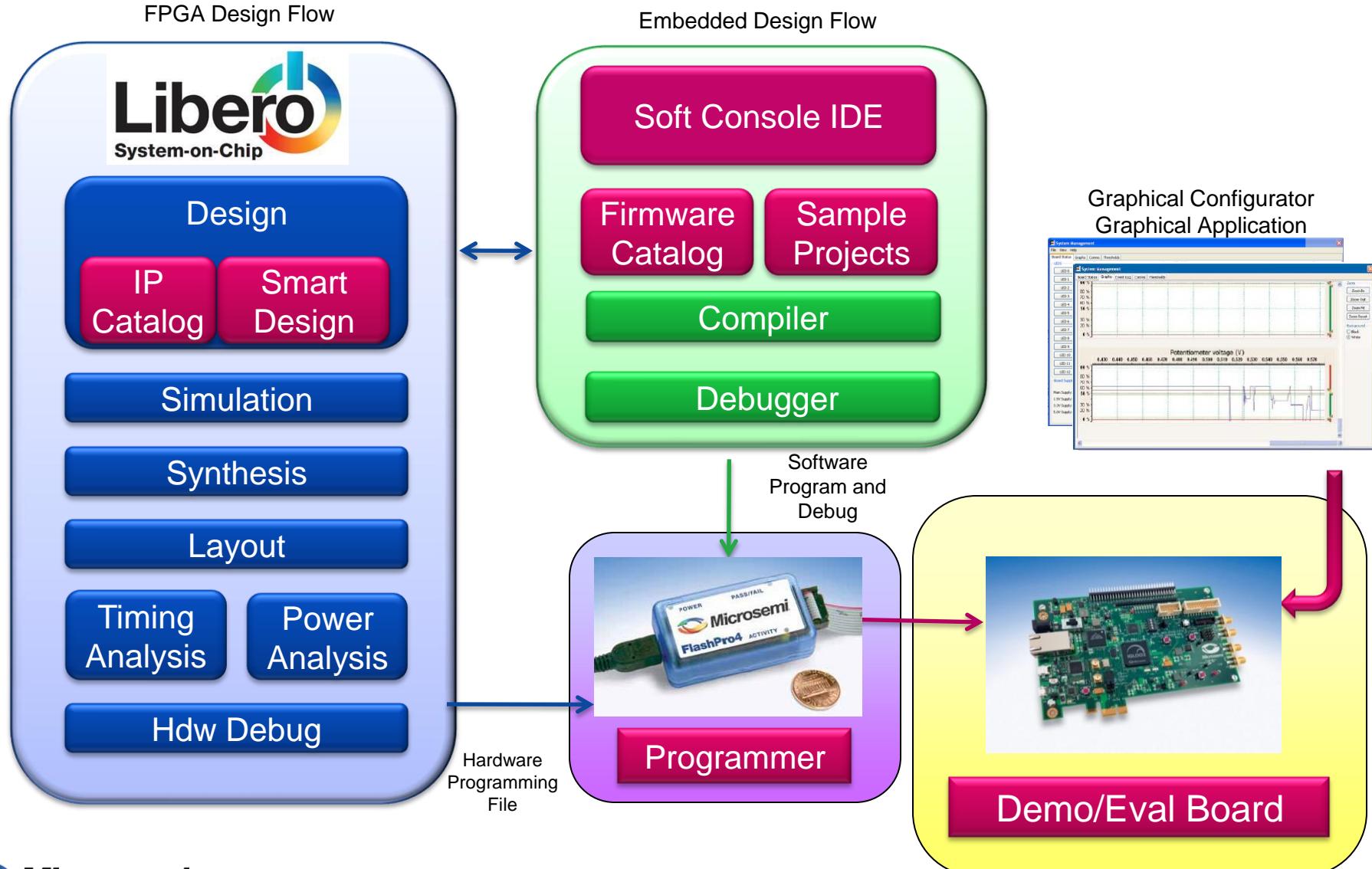
- SmartFusion2 hardware kit software options include
 - Free Libero SoC license
 - Free SoftConsole support
 - Optional Embedded Linux design environment

SmartFusion2 Software Stack



SmartFusion2 stack accelerates application development

Eco-System Supporting Silicon

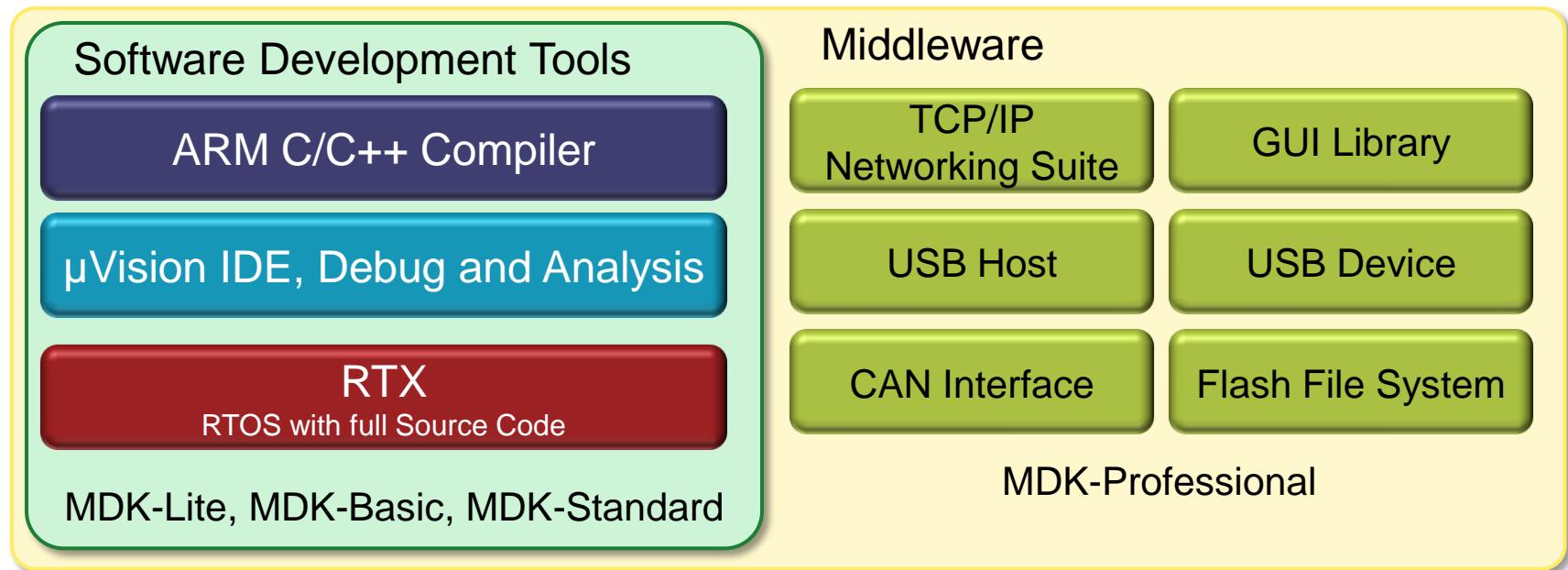


Ecosystem of Industry Leaders



Trusted Ecosystem Partners accelerate design in time with IP and Ease of Use

MDK-ARM Development System



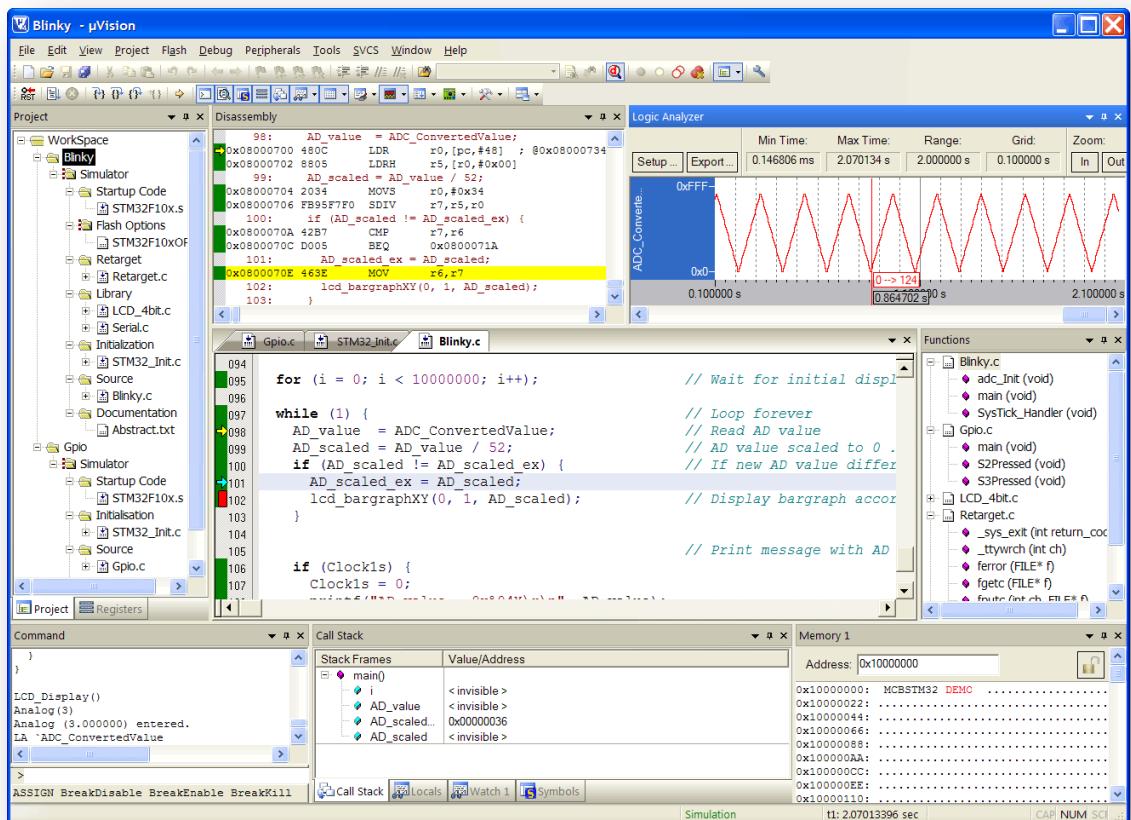
www.keil.com/arm

MDK-ARM Editions

	Professional	Standard	Cortex-M	Lite
μVision				
IDE	✓	✓	✓	✓
Debugger	✓	✓	✓	
Simulator	✓	✓	✓	32KB 32KB
ARM Compiler				
C/C++ Compilation Tools	✓	✓	✓	32KB
Device Support				
Cortex-M	✓	✓	✓	✓
SecurCores (SC000, SC300)	✓	✓	✓	✓
ARM7, ARM9, Cortex-R4	✓	✓	✓	✓
RTOS and Middleware				
RTX RTOS with full source	✓	✓	✓	✓
Middleware libraries	✓			
Licensing				
PSN Required	✓	✓	✓	
Pricing	\$9,500 €7,540	\$4,895 €3,885	\$3,200 €2,540	free

Keil µVision IDE

- Project Management
 - Source Code Editing
 - Programm Debugging
 - Trace Viewing
-
- Well-known environment
 - Optimized for MCU development



Keil Trace Support

- Supports ARM7, ARM9, Cortex-M0, Cortex-M1, Cortex-M3, and Cortex-M4 devices
- JTAG support for ARM7, ARM9, and Cortex-M
- Serial Wire Debug (SWD) support for Cortex-M
- Serial Wire Viewer (SWV) [Data and Event](#) Trace for Cortex-M up to 100Mbit/s (Manchester mode)
- Instruction Trace (ETM) for Cortex-M3 and Cortex-M4 up to 800Mbit/s
- Unique [Streaming Trace](#) direct to your PC, provides unlimited trace buffer
- JTAG Clock Speed up to 50MHz
- *Supports Cortex-M devices running at up to 200MHz*
- High-Speed Memory Read/Write up to 1MBytes/sec
- Seamless integration with the Keil [uVision](#) IDE & Debugger
- Wide target voltage range: 1.2V - 3.3V, 5V tolerant
- Support for 5V only devices using optional [5V Adapter](#)
- Optional [Isolation Adapter](#) provides electrical isolation from the target system
- USB 2.0 High-Speed connection
- USB powered (no power supply required)
- Target [Connectors](#)
 - 10-pin (0.05") - Cortex Debug Connector
 - 20-pin (0.10") - ARM Standard JTAG Connector
 - 20-pin (0.05") - Cortex Debug+ETM Connector



Keil Debug and Trace Adapters



ULINK2:

- Programming + Run-Control
- Memory + Breakpoint Access
- Serial Wire Trace Capturing (SWO)
 - 1Mbit/sec (UART mode)



ULINKpro:

- ULINK2 +
- Serial Wire Trace (SWO)
 - 100Mbit/sec (Manchester Mode)
- ETM Streaming Trace
 - Up to 800Mbit/sec
 - 100% Code Coverage and Performance Analysis

IAR Trace Support

■ General features

- USB-driven JTAG/SWD and Trace (as defined by ARM) interface
- Supports all Cortex-M3/M4 devices with ETM
- No power supply required, powered through USB
- SWD/JTAG speed up to 20 MHz
- USB 2.0 full speed (12 Mbit/sec)
- Target voltage can be measured
- High download speed
- Fully J-Link compatible
- 20-pin standard JTAG connector, 19 pin standard ETM v3 trace connector
- USB A-B, 20-pin flat JTAG cable and 19 pin flat trace cable included
- Full integration with C-SPY Debugger; advanced debugging features available from C-SPY
- Integration into IAR Embedded Workbench for ARM makes setup and use easy

■ Trace features

- *Trace supports up to 200 MHz*
- Trace based on ARM ETM v3/CoreSight
- 4Mbyte trace memory buffer
- 4 bit trace port support



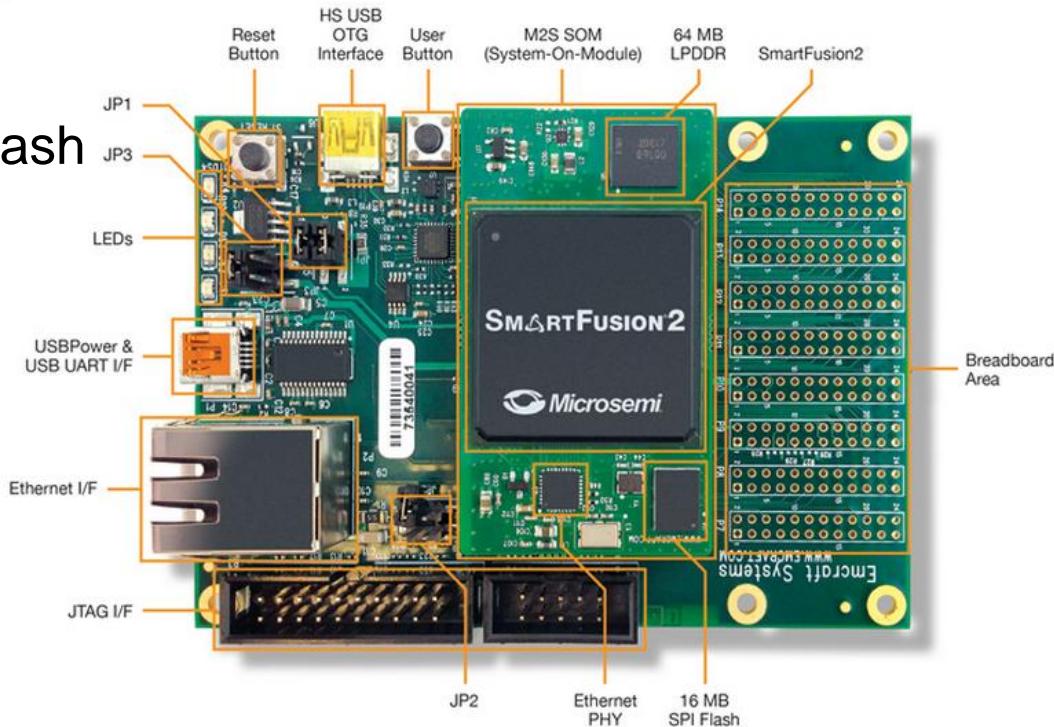
Development Boards

SmartFusion2 Development Board Overview

Development Kit/Boards	Device	Description
SmartFusion2 Starter Kit (SF2-484-STARTER-KIT or SF2-STARTER-KIT)	M2S050-FGG484 or M2S010-FGG484	uClinux based Low cost, Embedded design platform
SmartFusion2 Evaluation Kit (M2S-EVAL-KIT)	M2S025T-1FGG484	Low cost, SERDES, PCIe system development platform
SmartFusion2 Security Evaluation Kit (M2S090TS-EVAL-KIT)	M2S090TS-1FGG484	Low cost, Security, SERDES, PCIe system development platform
SmartFusion2 Advanced Development Kit (M2S150-ADV-DEV-KIT-ES)	M2S150TS-1FCG1152ES	Development platform for 150K LE devices focused on Security and FMC Expansion
IGLOO2 Evaluation Kit (M2GL-EVAL-KIT)	M2GL010T-1FGG484	Low cost, SERDES, PCIe system development platform

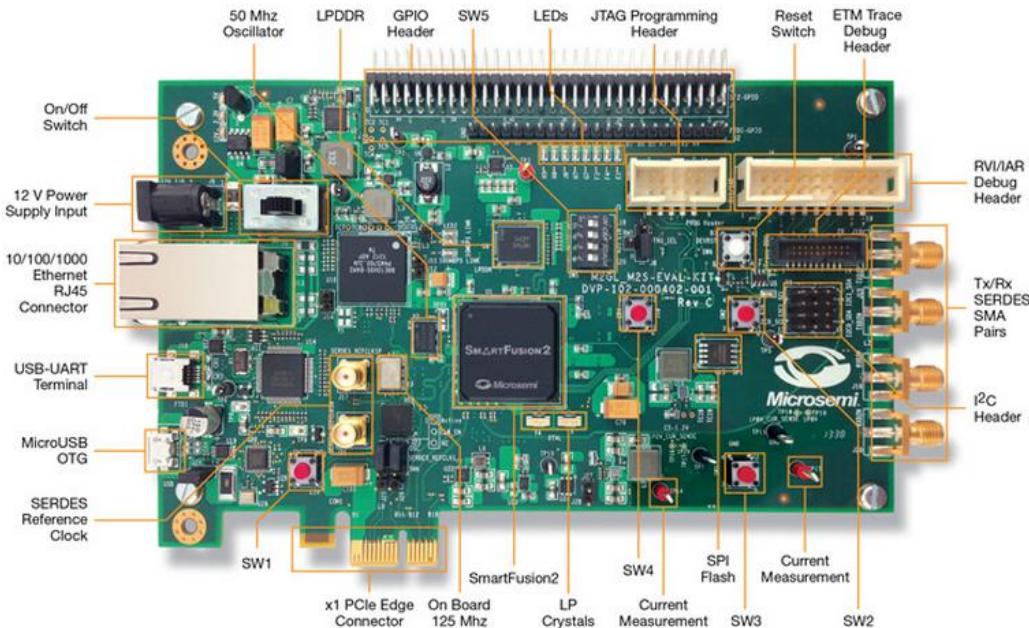
SmartFusion2 Starter Kit

- Two variants –
 - SF2-STARTER-KIT with **M2S050-FGG484**
 - SF2-484-STARTER-KIT with **M2S010-FGG484**
- Features –
 - 10/100 Ethernet
 - USB OTG interface
 - 64MB LPDDR, 16MB SPI flash
 - USB based Wi-Fi module
 - Breadboard expansion
- Price
 - List - \$299
- Part number
 - SF2-STARTER-KIT
 - SF2-484-STARTER-KIT



SmartFusion2 Evaluation Kit

- Applications
 - Develop and test PCI Express Gen2 x1
 - Evaluate SerDes transceiver using SMA Pairs
 - Power measurement of the SmartFusion2 SoC FPGA
 - Create a working PCIe link quickly with the PCIe Control Plane Demo
- Features
 - 512MB LPDDR, 64MB SPI flash
 - x1 PCIe Edge connector
 - Four SMA connector
 - 10/100/1000 Ethernet
 - GPIO expansion
- Price
 - List - \$399
- Part number
 - M2S-EVAL-KIT



SmartFusion2 Advanced Development Kit

■ Applications

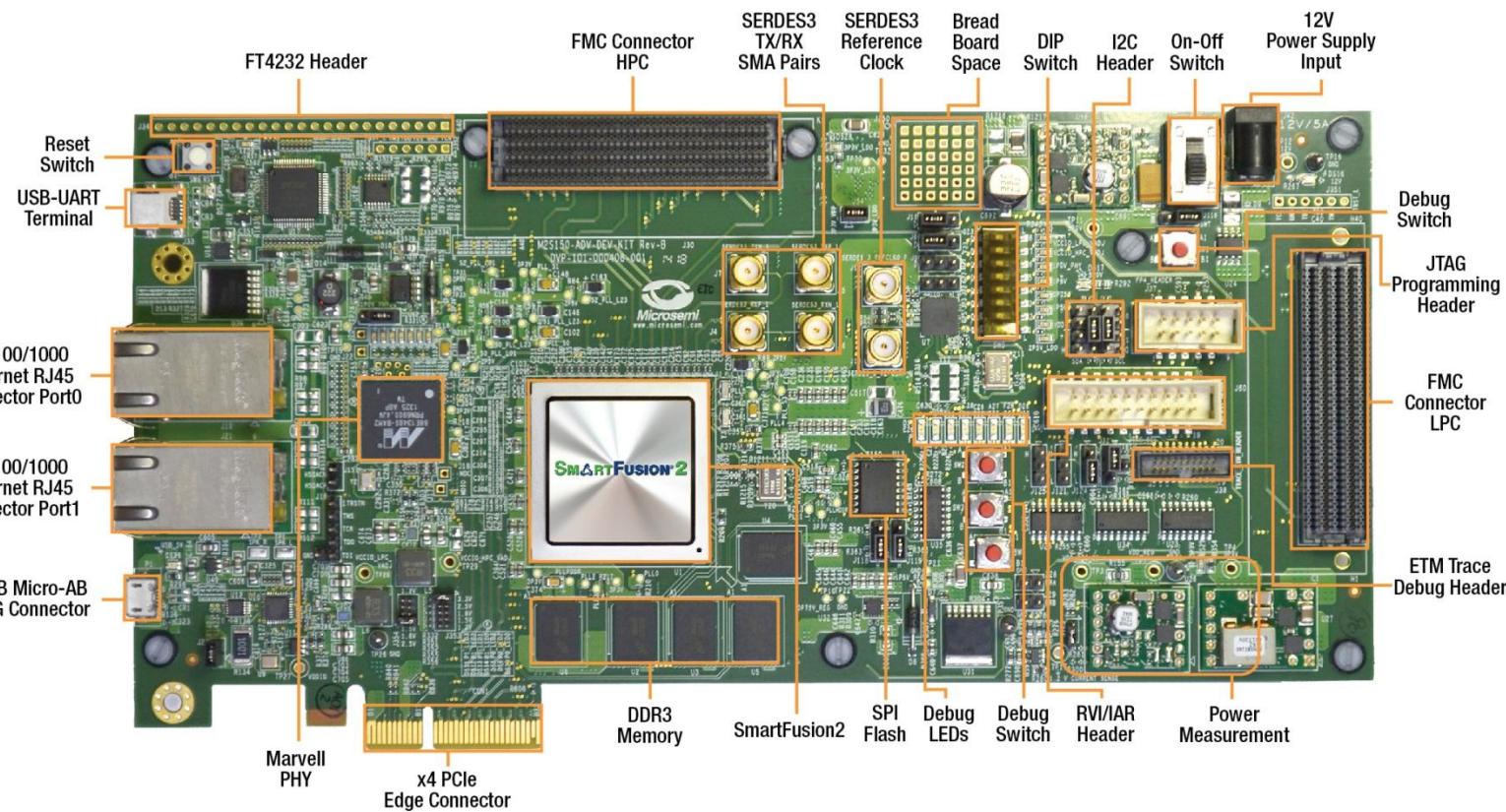
- Embedded ARM® Cortex™-M3 processor based systems
- PCIe endpoint
- Motor control
- Industrial automation
- Power measurement
- Security
- FMC expansion
- High speed I/O
- Universal serial bus (USB) applications (OTG support)

■ Features

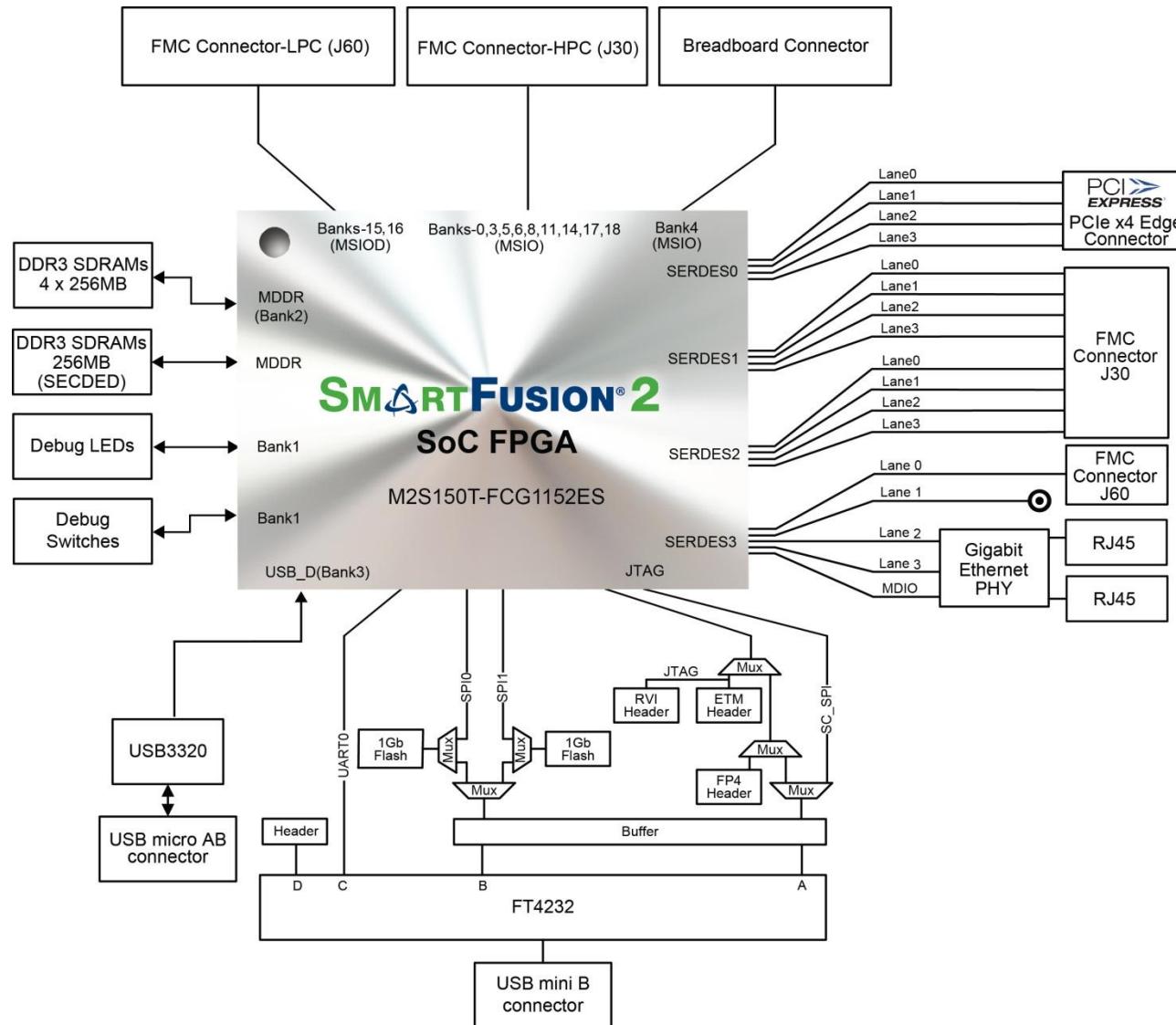
- Dual Gigabit Ethernet, USB 2.0, SPI, I2Cs, UARTs
- 2 Giga Bytes (GB) SPI flash - 1GB connected to MSS and other 1GB connected to FPGA fabric
- Two FMC connector with HPC/LPC pinout for expansion
- x4 PCIe edge connector
- One pair SMA connector
- Core current measurement test points

SmartFusion2 Advanced Development Kit

- Part number - M2S150-ADV-DEV-KIT-ES
- List Price – \$999
- DC - \$900



SmartFusion2 Advanced Development Kit



SoftConsole Overview

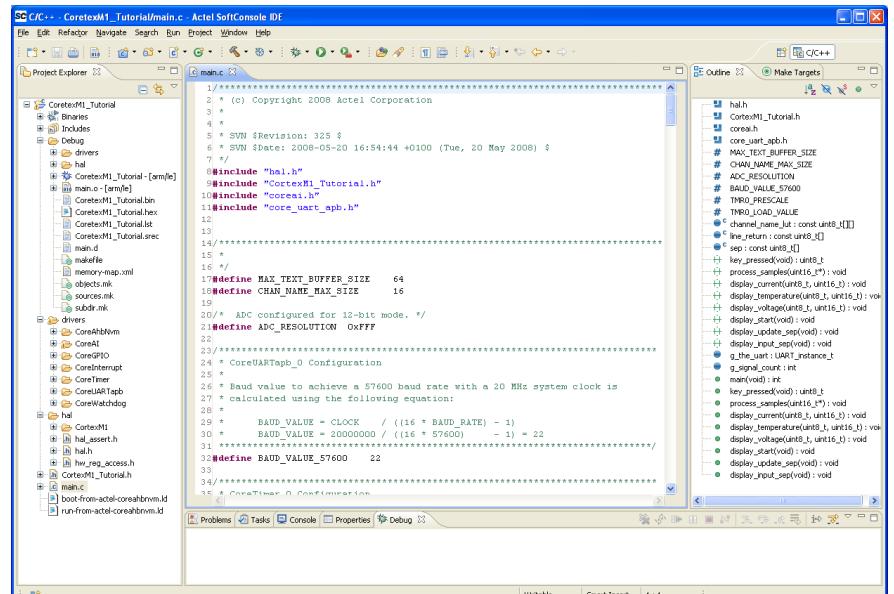
SoftConsole Software Tools

■ FREE Software Development Environment

- Eclipse-based IDE
- Free GNU Tools with FS2 Debug
- Customized for Microsemi Processors
- Alternative to Professional-grade Tools

■ V3.4 SP1 Available Now

- Info/Download on Website
- Installs with Libero SoC v11.5
 - SP1 must be installed standalone
- Supports Microsemi Processors
 - Cortex-M3
 - Cortex-M1
 - CoreMP7
 - Core8051s



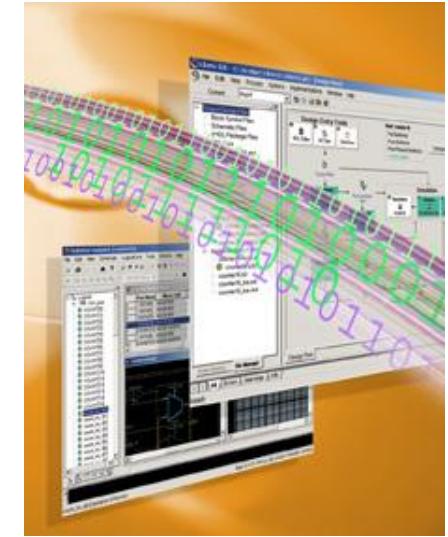
The screenshot shows the Actel SoftConsole IDE interface. The Project Explorer on the left lists the project structure for "CoretexM1_Tutorial". The main window displays the "main.c" file with C code. The code includes definitions for buffer sizes, channel names, and ADC resolution, along with comments about baud rate calculations. The right side of the interface contains toolbars for File, Edit, Refactor, Navigate, Search, Run, Project, Window, Help, and various debugging and configuration options.

```
1 /**************************************************************************  
2 * (c) Copyright 2008 Actel Corporation  
3 *  
4 * SVN $Revision: 325 $  
5 * SVN $Date: 2008-05-20 16:54:44 +0100 (Tue, 20 May 2008) $  
6 */  
7 #include "hal.h"  
8 #include "CoretexM1_Tutorial.h"  
9 #include "coremi.h"  
10 #include "core_wart_epb.h"  
11  
12  
13  
14 //*****  
15 //*****  
16 //*****  
17 #define MAX_TEXT_BUFFER_SIZE 64  
18 #define CHAN_NAME_MAX_SIZE 16  
19  
20 /* ADC configured for 12-bit mode. */  
21 #define ADC_RESOLUTION 0xFF  
22  
23 //*****  
24 //*****  
25 //*****  
26 // CoreUARTapb_0 Configuration  
27 //*****  
28 // Baud value to achieve a 57600 baud rate with a 20 MHz system clock is  
29 // calculated using the following equation:  
30 // Baud_Value = CLOCK / ((16 * Baud_Pate) - 1)  
31 // Baud_Value = 2000000 / ((16 * 57600) - 1) = 22  
32 #define BAUD_VALUE_57600 22  
33  
34 //*****  
35 // CoreUART Configuration  
36  
37 //*****
```

SoftConsole Features

- Tools
 - C/C++ Programming
 - Debugging
 - Disassembly
 - Signals
 - Evaluation of Expressions at Runtime

- Intelligent Software Analysis
 - Citing Points of Declaration and Definition
 - Source Code Outlining
 - Syntax Highlighting and Comment Toggling
 - Code History for Tracking Changes
 - Code Assist Mode
 - Source Code Completion to Avoid Syntax Errors
 - Code Templates to Auto-insert Standard Code
 - Examples – `switch` Statement and `try/catch` Block



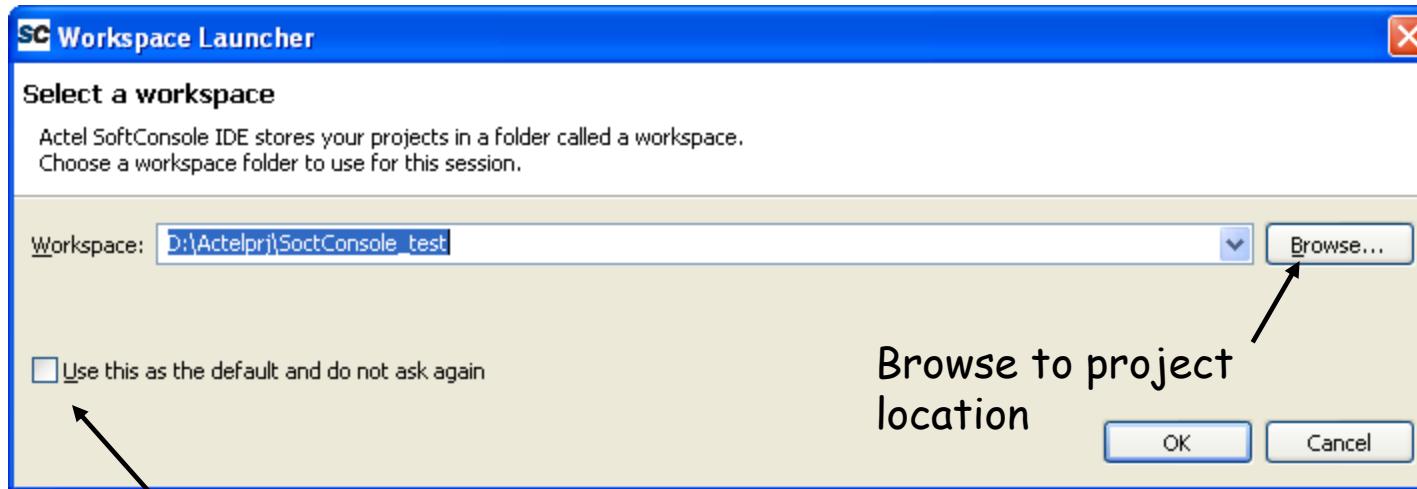
SoftConsole Perspectives and Views

- Views
 - Different Ways of Displaying Information
 - Can Be Opened, Closed, Docked, Undocked, and Moved within a Perspective
- Perspectives
 - Pre-defined Sets of Views aimed at Accomplishing a Specific Task
 - C/C++
 - Debugging
 - Open in Existing Workspace Window or New Window
 - Available Perspectives:
 - C/C++ (Default)
 - Resource
 - Debug
 - SVN Repository Exploring
 - Team Synchronizing
 - Perspectives can be Customized

SoftConsole Workspace Selection

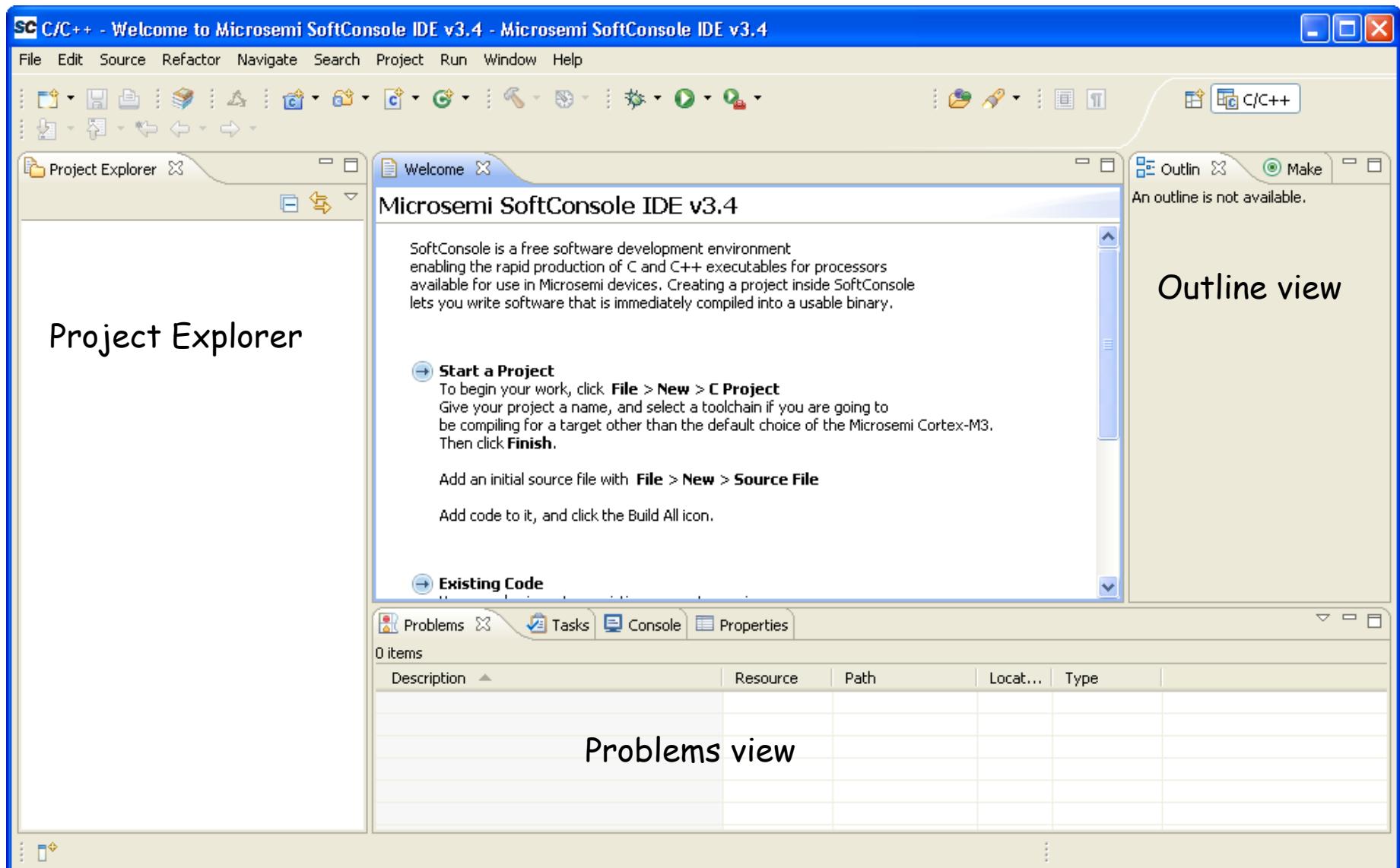
File > Switch Workspace

- Specify Default Location for Projects
 - Specify location of projects, folders and files created with the Workbench



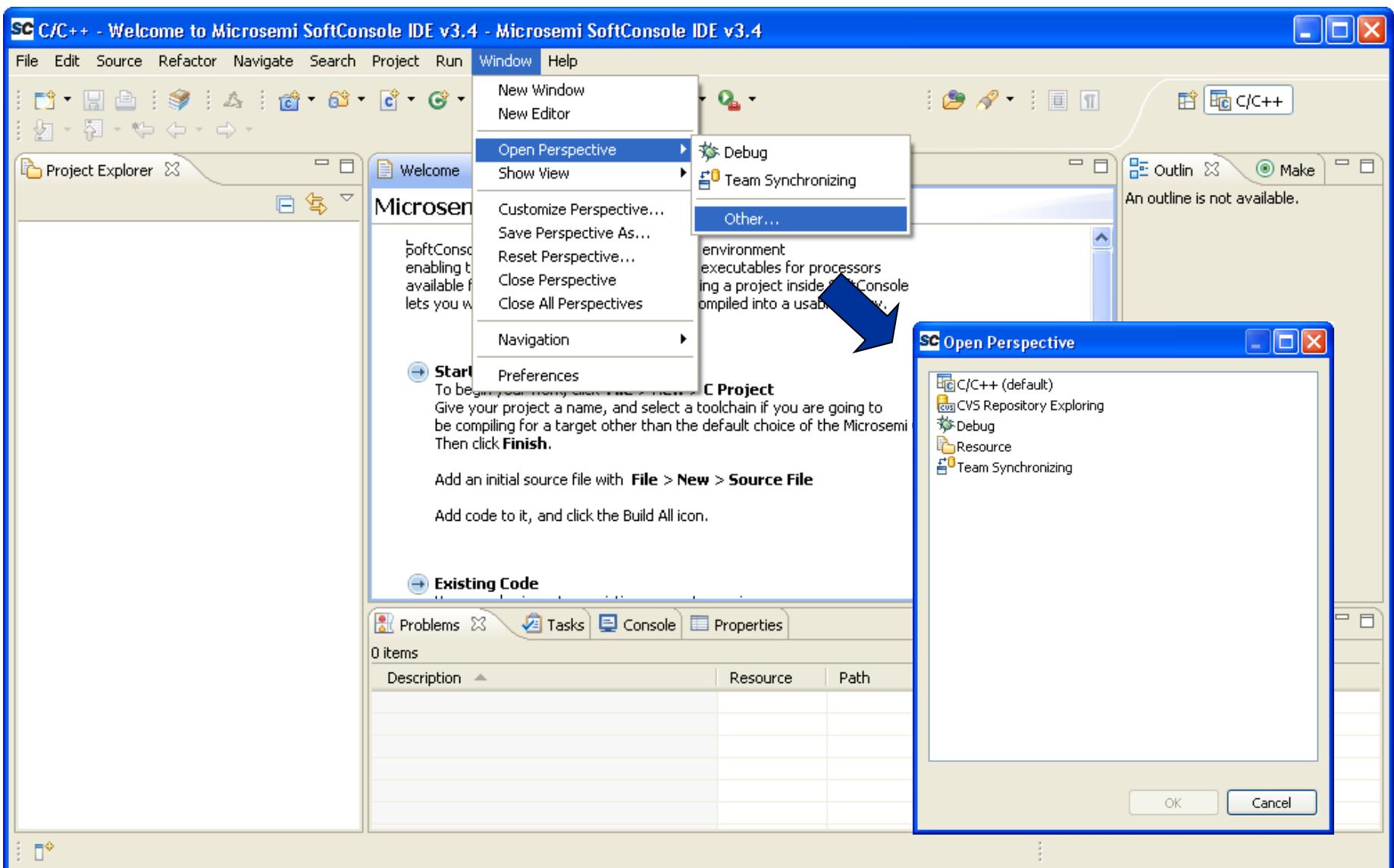
Check to prevent Workspace Launcher from opening each time SoftConsole is launched

SoftConsole C/C++ Perspective (Default)

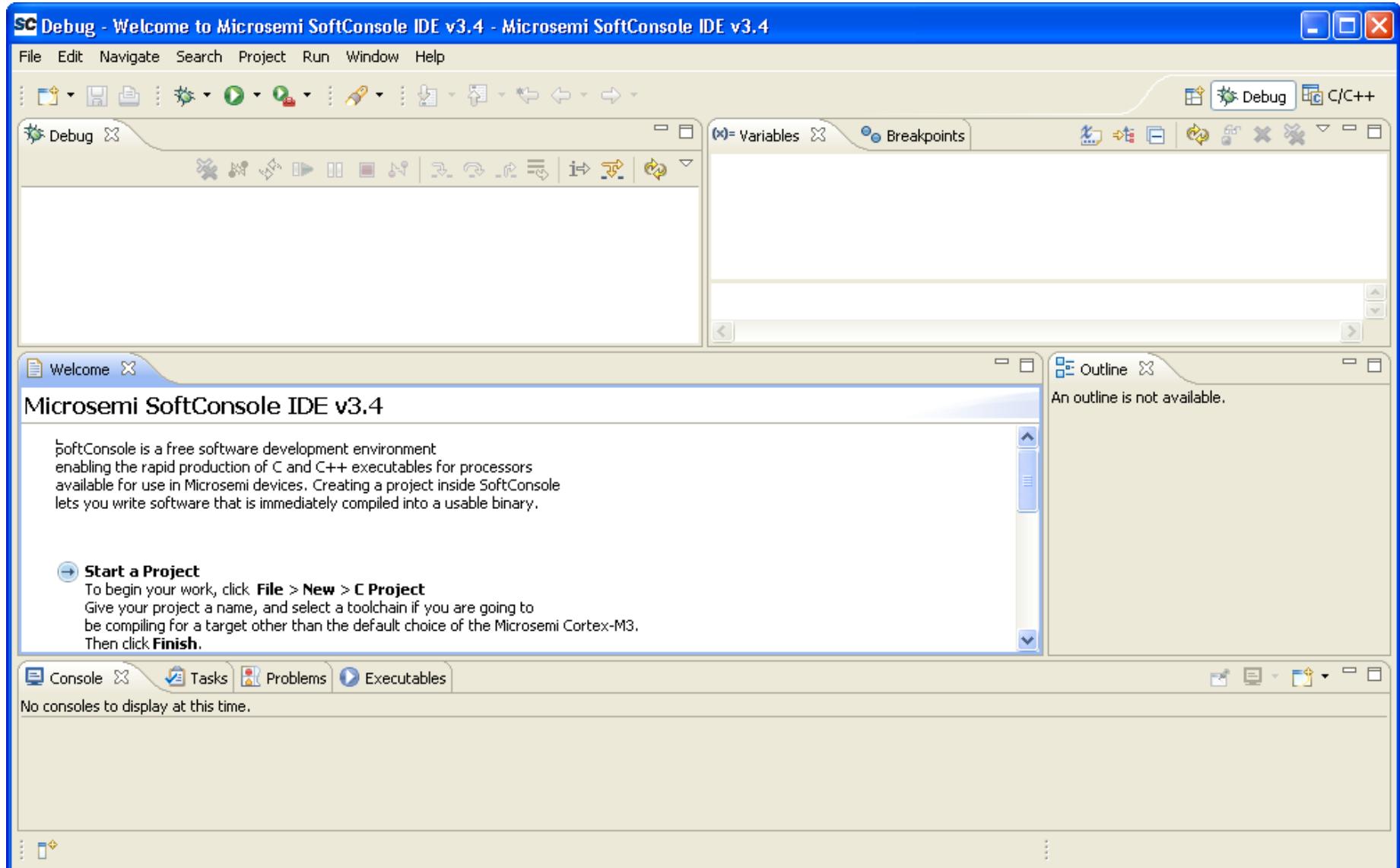


Opening Perspectives

Window > Open Perspective

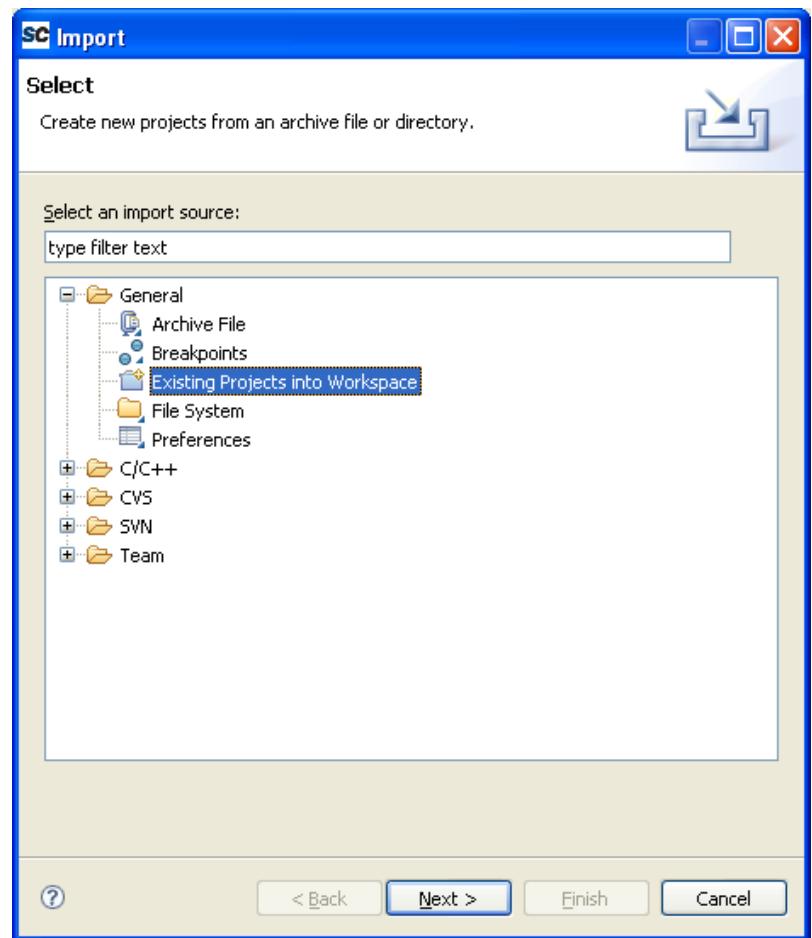
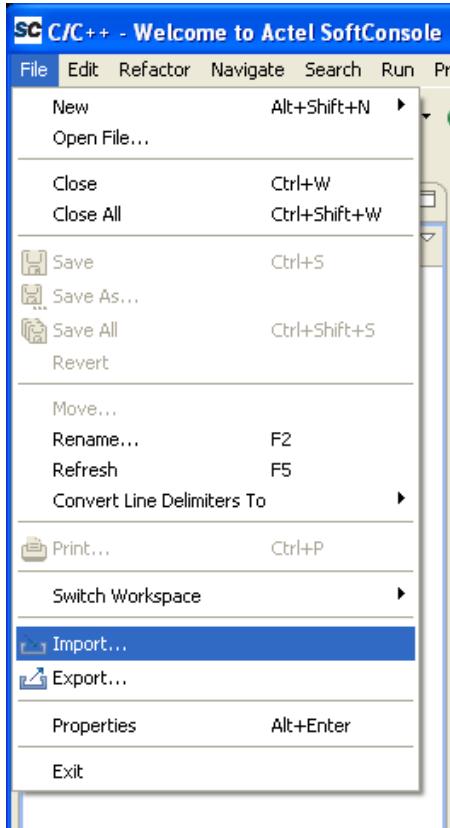


Debug Perspective User Interface

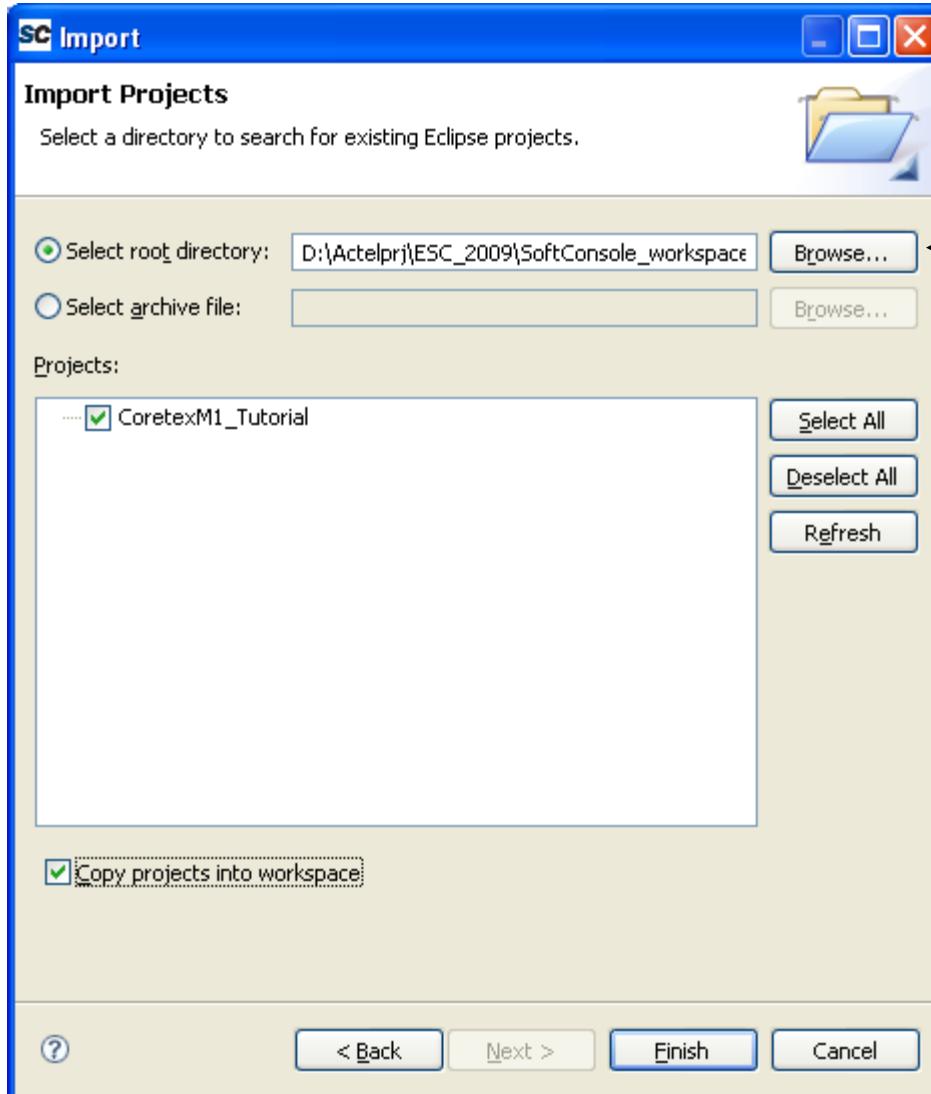


Importing an Existing Project

- Select the Workspace
- Select File > Import
 - Choose Existing Projects

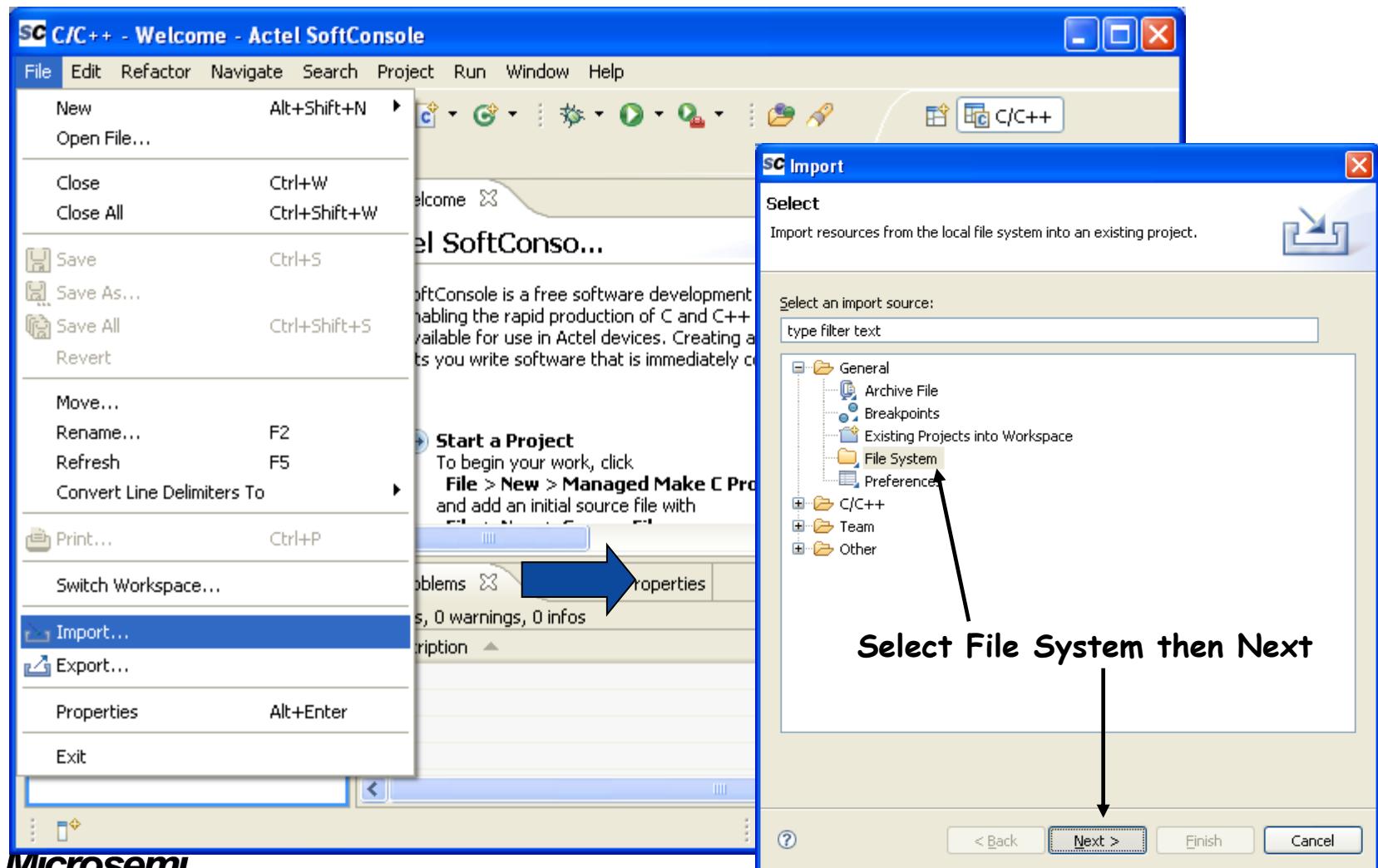


Importing an Existing Project



Importing Files to Project

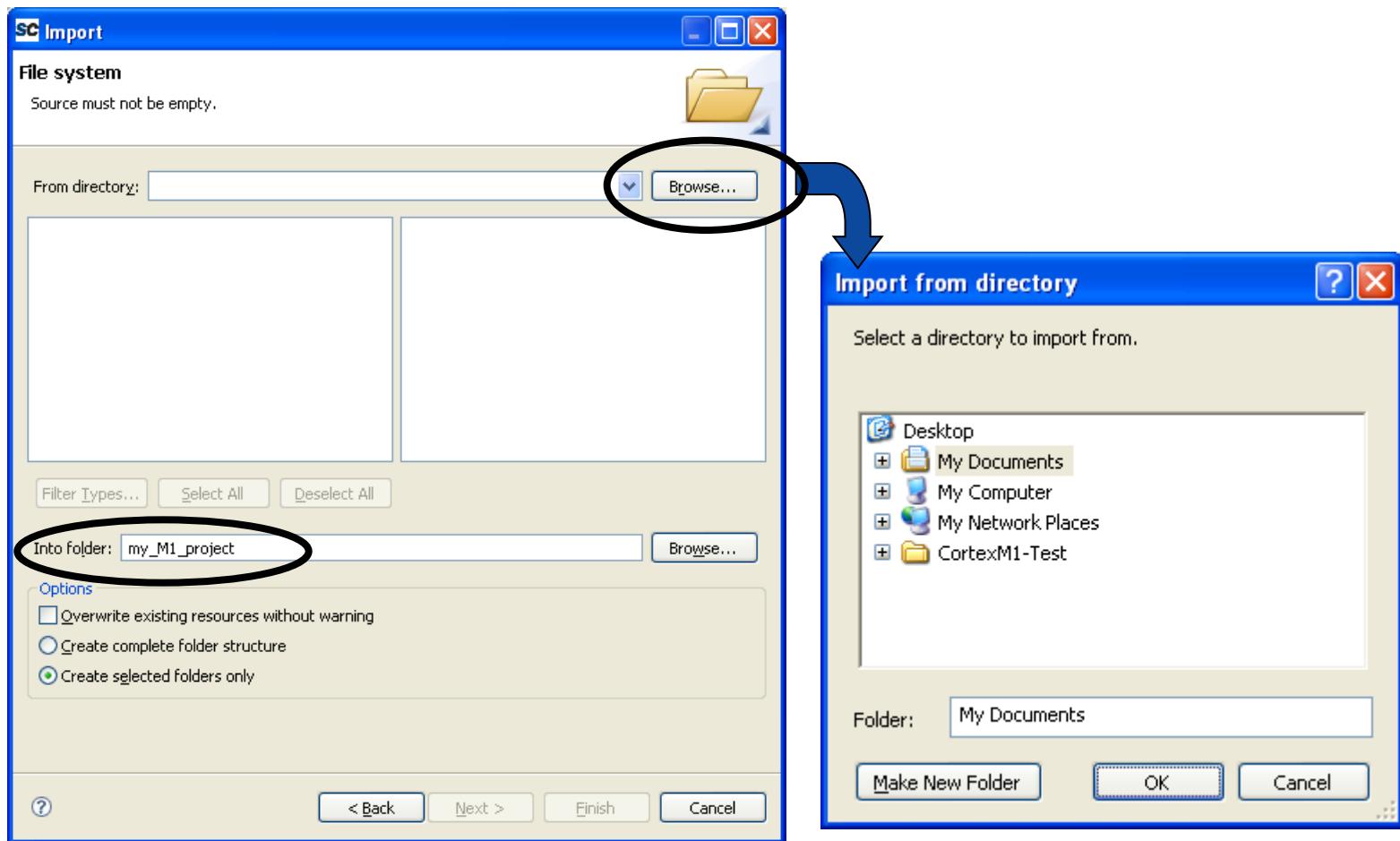
- File > Import



Importing Files to Project

- Click Browse and Navigate to Files

Files
Imported
Here

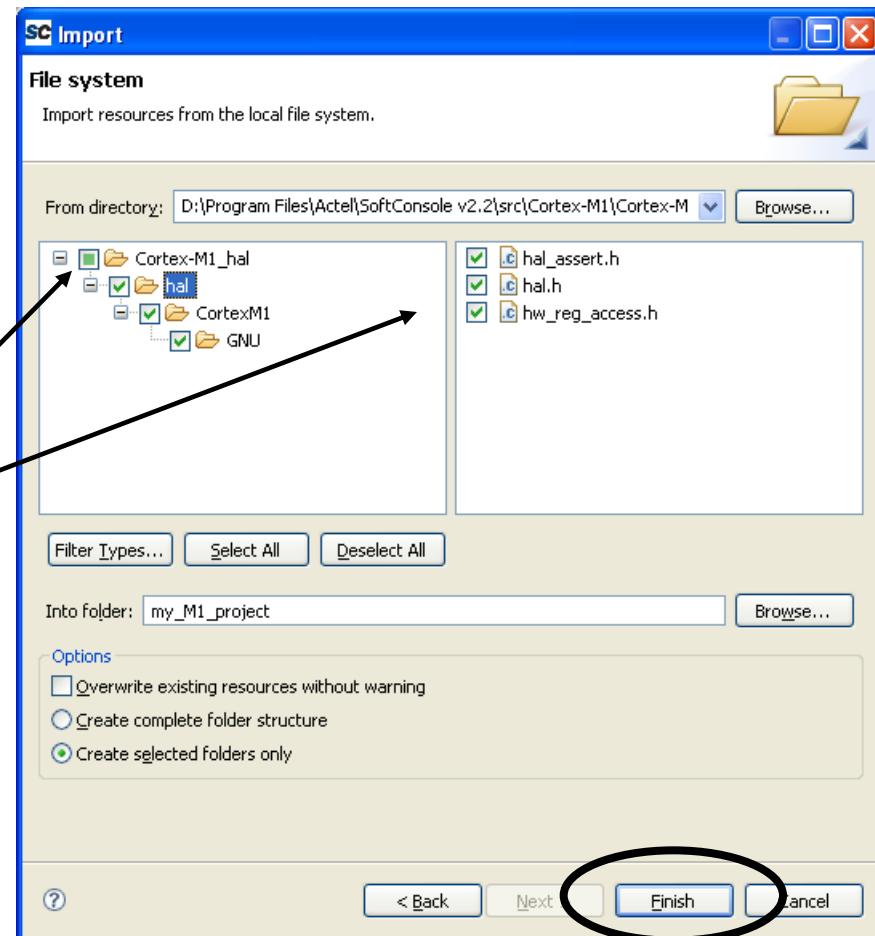


Importing Files to Project

- Select File(s) to Import
- Specify Destination Folder and Import Options
- Click Finish

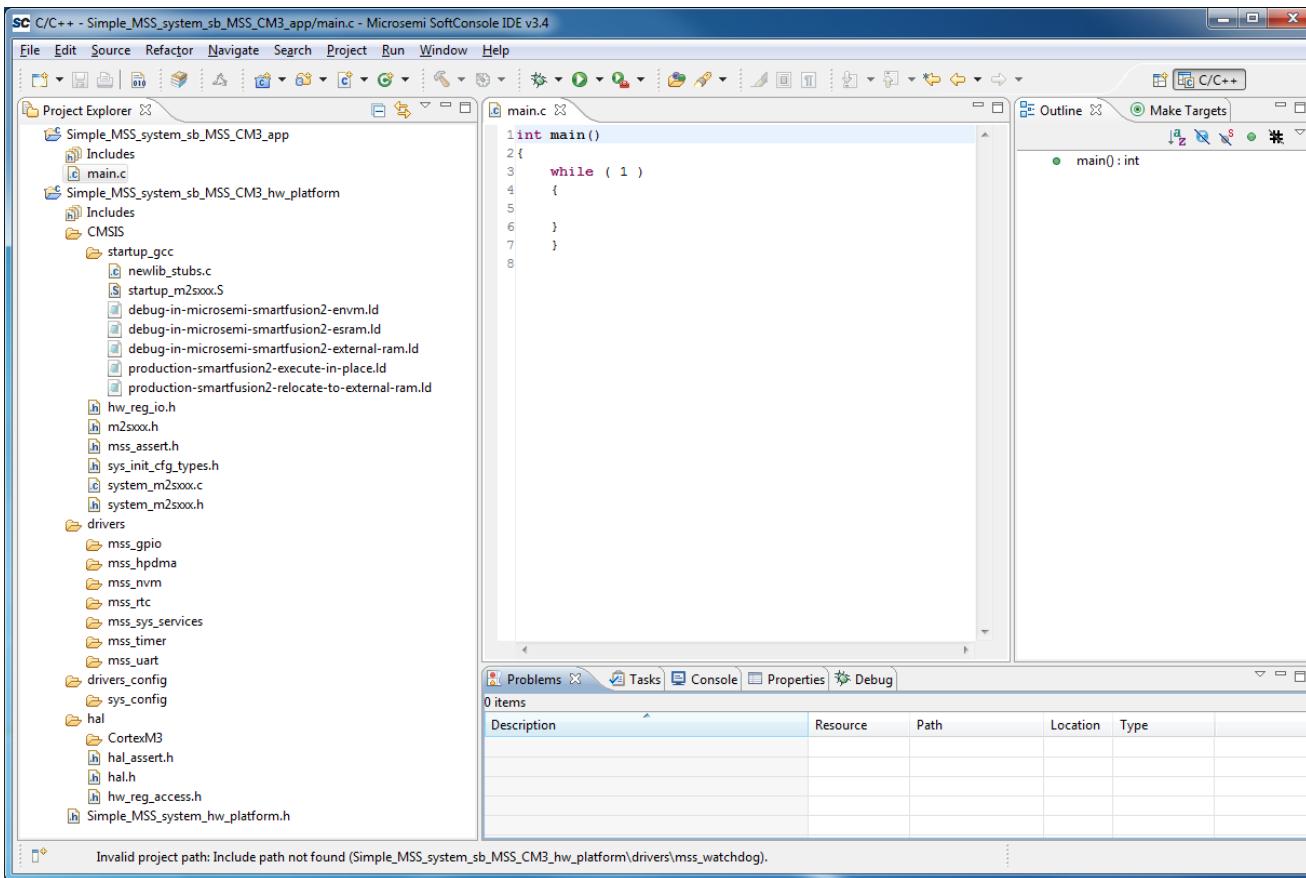
Check to select all files
OR
Select individual files

File import options

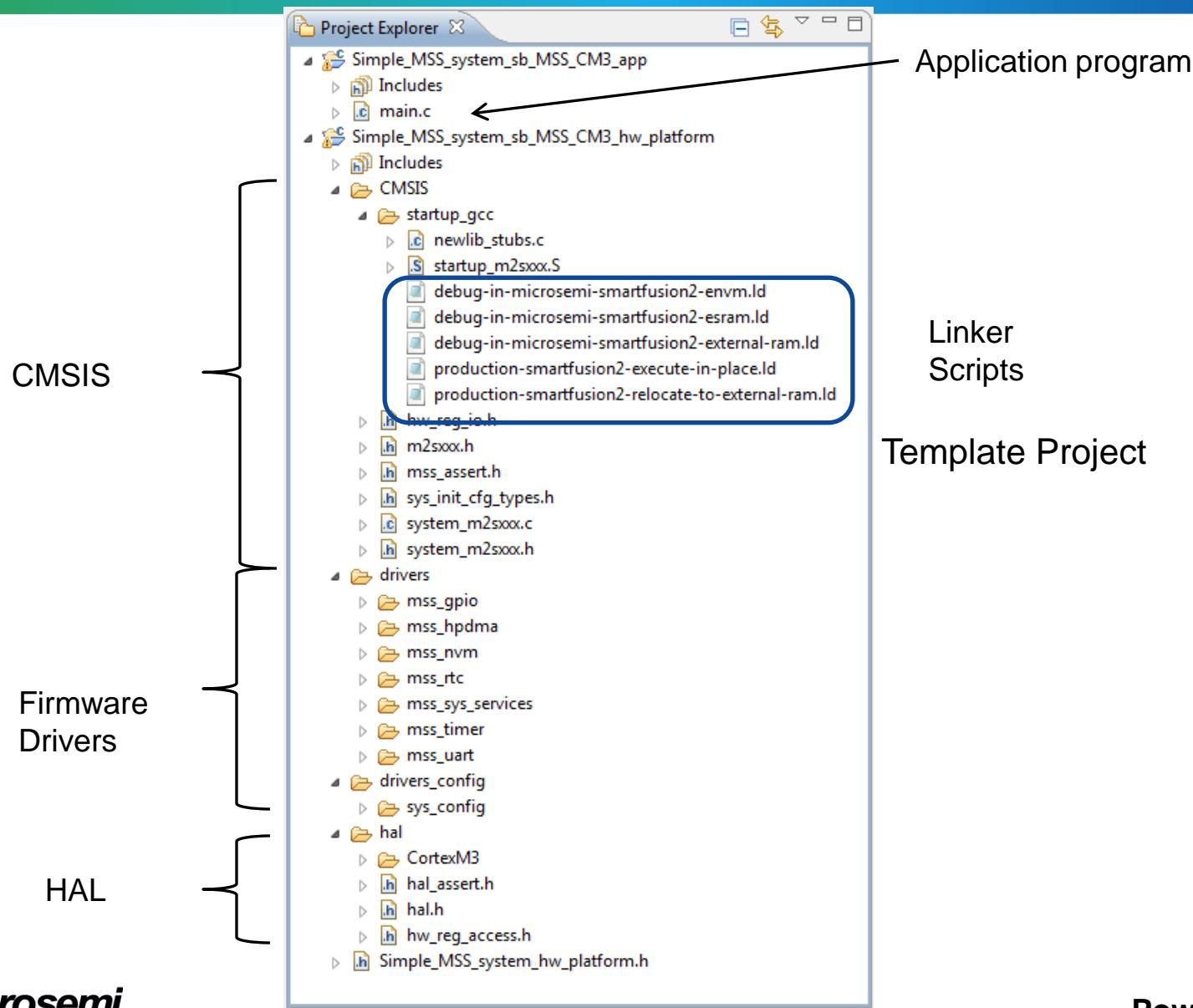


Libero SoC SoftConsole Project

- Libero generated software project:
 - Includes all hardware driver files, CMSIS, HAL
 - Contains all necessary project settings
 - Starting point for code development



SmartFusion2 Template Project in SoftConsole



Linker Scripts

- The GCC linker is controlled using a linker script
- Linker scripts are used to specify the load and execution memory locations

```
MEMORY
{
    flash (rx) : org = 0x00000000, len = 0x00040000
    sram  (rw) : org = 0x10000000, len = 0x00080000
}

SECTIONS
{
    . = 0x00000000;
    _image_start = . ;
    .boot 0x00000000 : AT( 0x00000000 )
    {
        *vectors.o (.text)
        *crt0.o (.text)
        _boot_end = . ;
    } > flash

    .init : AT( _boot_end )
    {
        _init_start = . ;
        KEEP (*(.init))
        *(.init_array)
        *(.fini_array)
        *(.fini)
        *(.eh_frame)
        *(.jcr)
        _init_end = .;
    } > flash

    __start_code_lma__ = . ;
    .text : AT( _init_end )
    {
        __start_read_only__ = . ;
        __code_start_vma__ = . ;
        __code_vma_start__ = . ;
        _text_start = . ;
        code = .;
        *(.text)
        *(.glue_7t) *(.glue_7)
        . = ALIGN(4);
        _end_text_lma = . ;
        __code_end_vma__ = . ;
        __code_vma_end__ = . ;
    } > sram
    __end_code_lma__ = __start_code_lma__ + (__code_vma_end__ - __code_vma_start__)

    __end_text__ = .;

    __start_rodata_lma__ = __end_code_lma__ ;
```

SmartFusion2 Linker Scripts

- Linker Scripts in CMSIS-PAL firmware :
 - `debug-in-microsemi-smartfusion2-envm.1d`
 - `debug-in-microsemi-smartfusion2-esram.1d`
 - `debug-in-microsemi-smartfusion2-external-ram.1d`
 - `production-smartfusion2-execute-in-place.1d`
 - `production-smartfusion2-relocate-to-external-ram.1d`

`debug-in-*` linker scripts are used to build/link programs for downloading to and debugging from eNVM, eSRAM and external RAM using SoftConsole

Programs built using the `debug-in-microsemi-smartfusion2-envm.1d` linker script can be downloaded to and debugged in SmartFusion2 eNVM. After downloading to eNVM they can also run out of eNVM from power on reset

`production-*` linker scripts are for building/linking “production” programs that are stored in eNVM and execute out of reset in a live system

The Intel `<project-name>.hex` file generated by SoftConsole in the project’s Debug or Release folder is stored in the SmartFusion eNVM Data Storage Client using FlashPro4 or some other method

Linker Scripts – More Information

- **GNU Linker Manual (ld.pdf)**

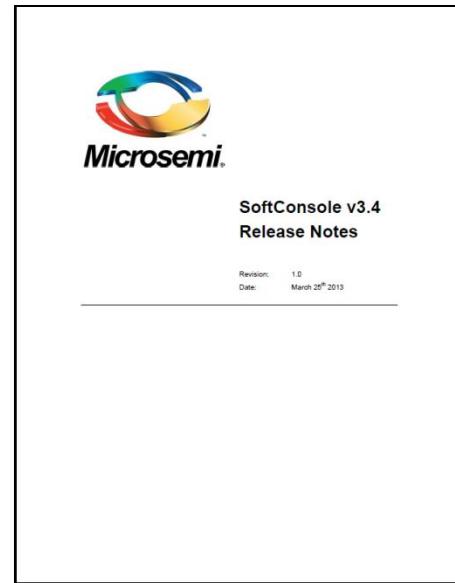
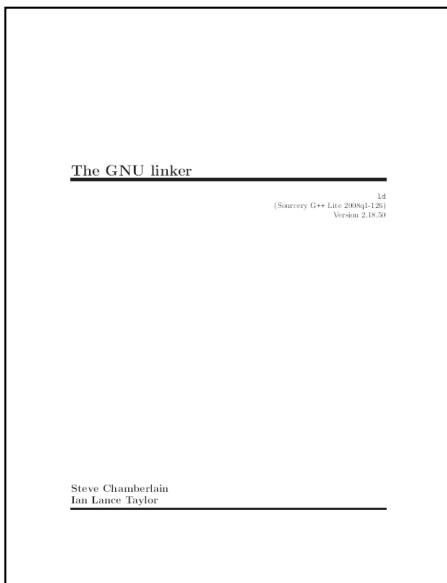
Contains a detailed description of Linker scripts

(<SoftConsole v3.4 install>\Sourcery-G++\share\doc\arm-none-eabi\pdf)

- **SoftConsole 3.4 Release Notes**

- ***“Using ld, the GNU linker” document***

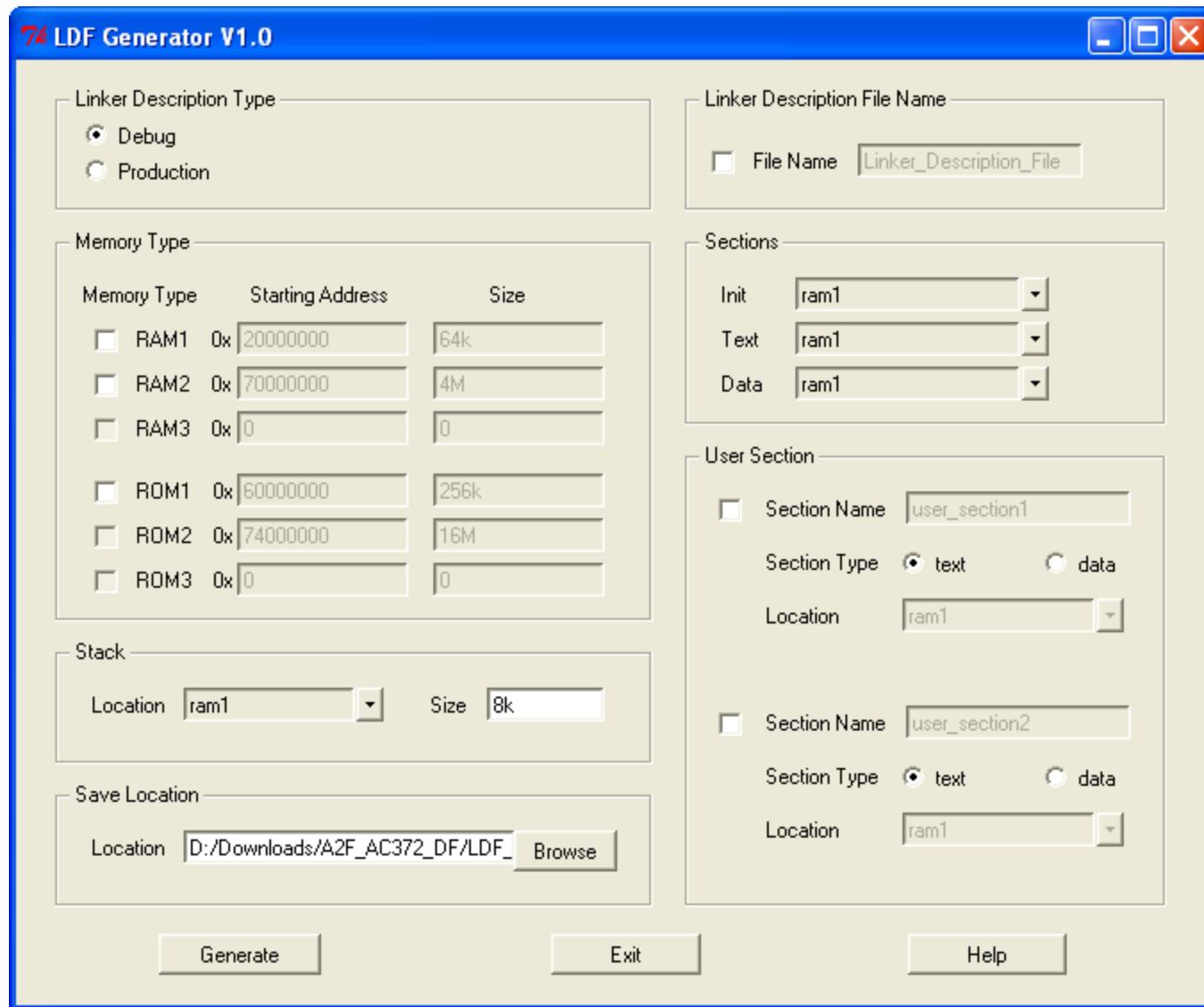
http://ftp.gnu.org/old-gnu/Manuals/ld-2.9.1/html_mono/ld.html



Linker Description File (LDF) Generator Utility

- Utility for creating Linker Scripts for SmartFusion2 cSoC systems
 - LDF Generator only supports SoftConsole
- Refer to Application Note AC390 – “SmartFusion2 SoC FPGA – Remapping eNVM, eSRAM, and DDR/SDR SDRAM Memories” for more information:
 - http://www.microsemi.com/document-portal/doc_download/129976-ac390-smartfusion2-soc-fpga-remapping-envm-esram-and-ddr-sdr-sdram-memories-app-note
- LDF Generator utility is included in design files for AC-372 “SmartFusion cSoC: Basic Bootloader and Field Upgrade eNVM Through IAP Interface”
 - http://www.microsemi.com/document-portal/doc_download/129823-ac372-smartfusion-csoc-basic-bootloader-and-field-upgrade-envm-through-iap-interface-app-note

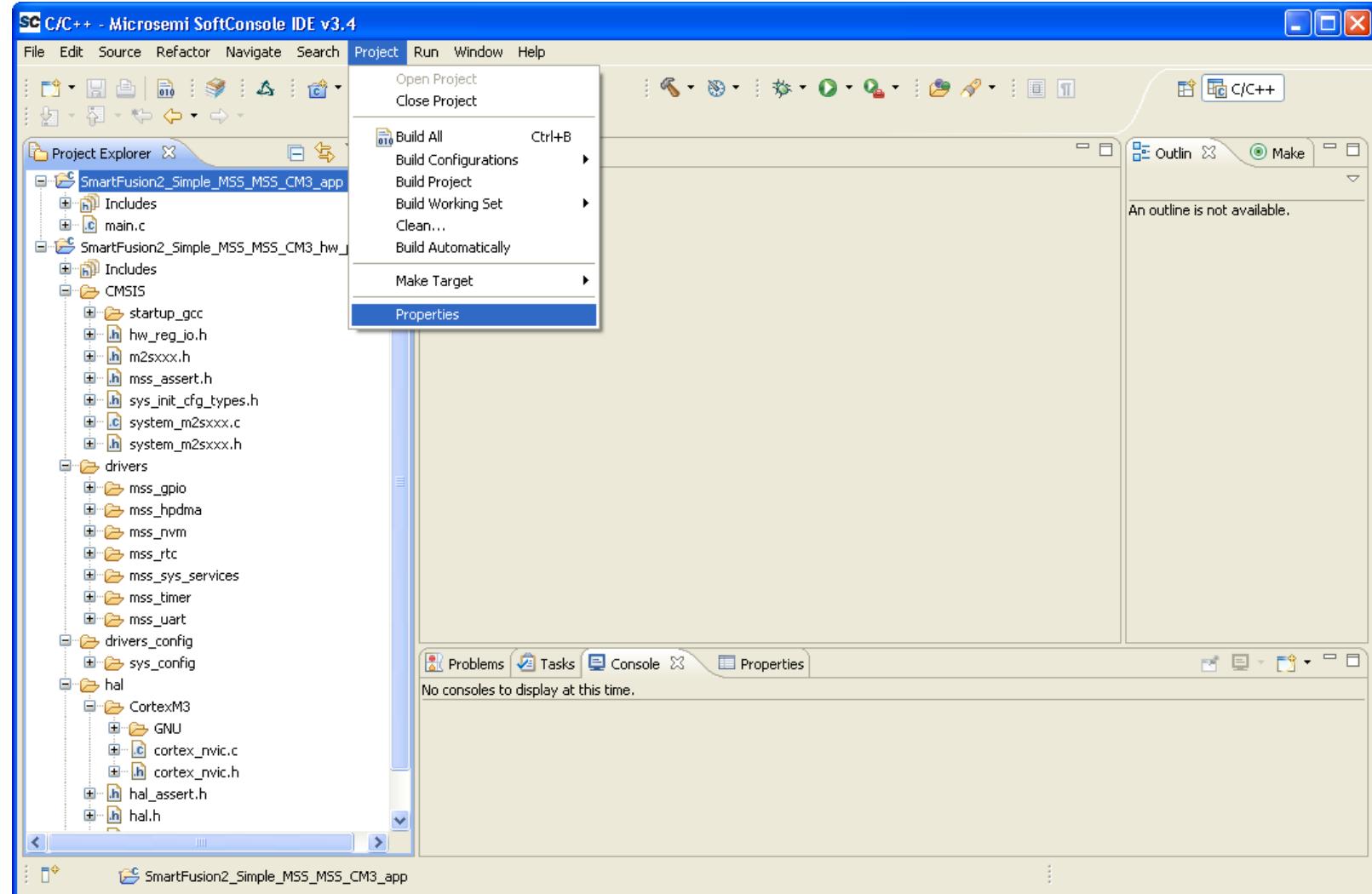
LDF Generator GUI



Specifying the Linker Script

SoftConsole Project Configuration

■ Project > Properties



GNU C Linker - Set Linker Script

- Specify Linker Script for the Project in the Linker Flags field
 - Sample projects use debug in eSRAM linker script

Select Settings

Settings

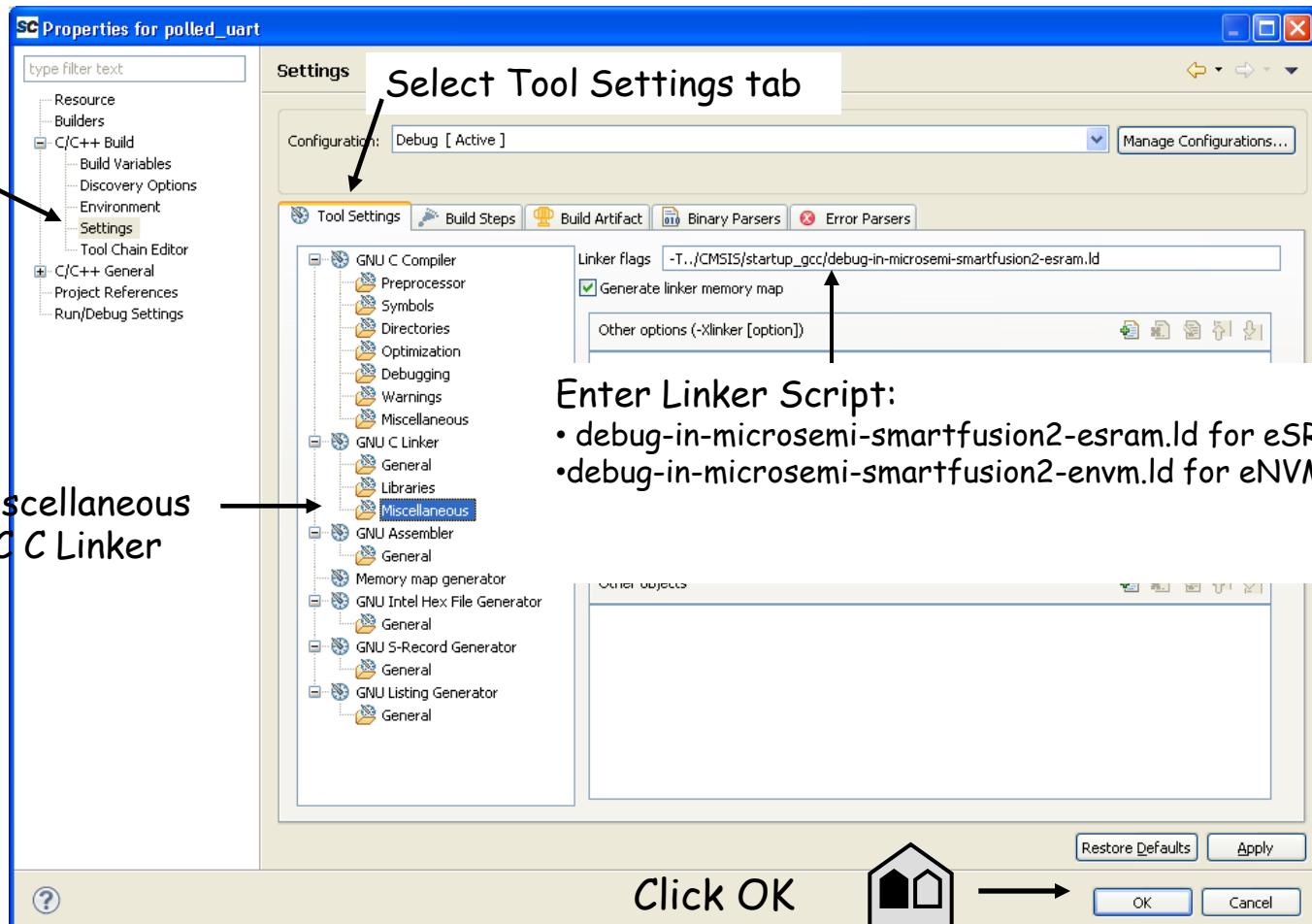
Select Tool Settings tab

Select Miscellaneous under GCC C Linker

Click OK

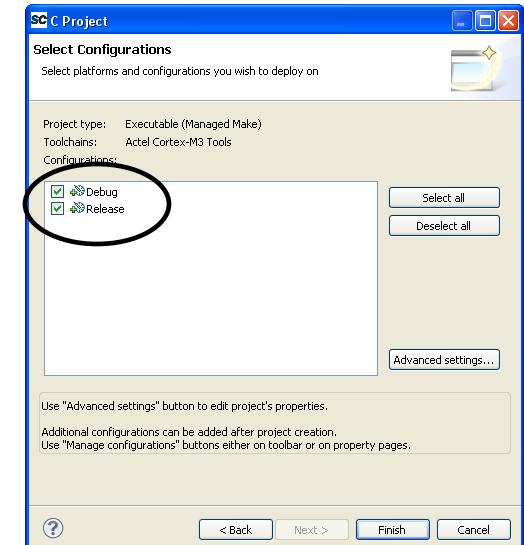


OK Cancel



Project Build Configurations

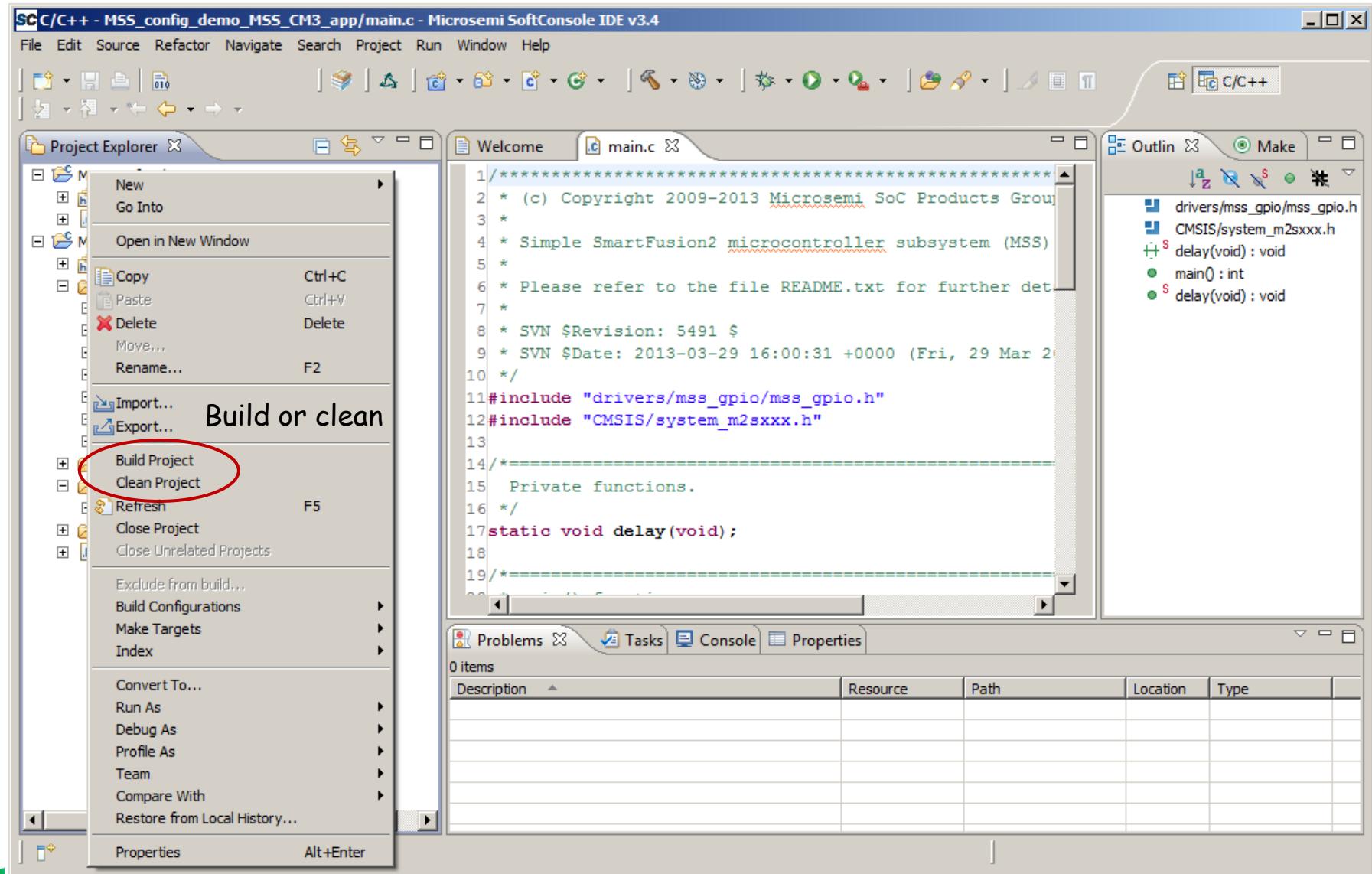
- SoftConsole Supports Two Build Configurations:
 - Debug Configuration
 - Builds an executable that can be run from eSRAM
 - Uses Debug Linker script
 - Example: "debug-in-microsemi-smartfusion2-esram.ld"
 - Release Configuration
 - Builds an executable that can be run from non-volatile memory (eNVM)
 - Creates Intel Hex file
 - Uses Release Linker script
 - Example:
"debug-in-microsemi-smartfusion2-envm.ld"
- Configurations for Project can be Selected in New Project Wizard



Building and Cleaning

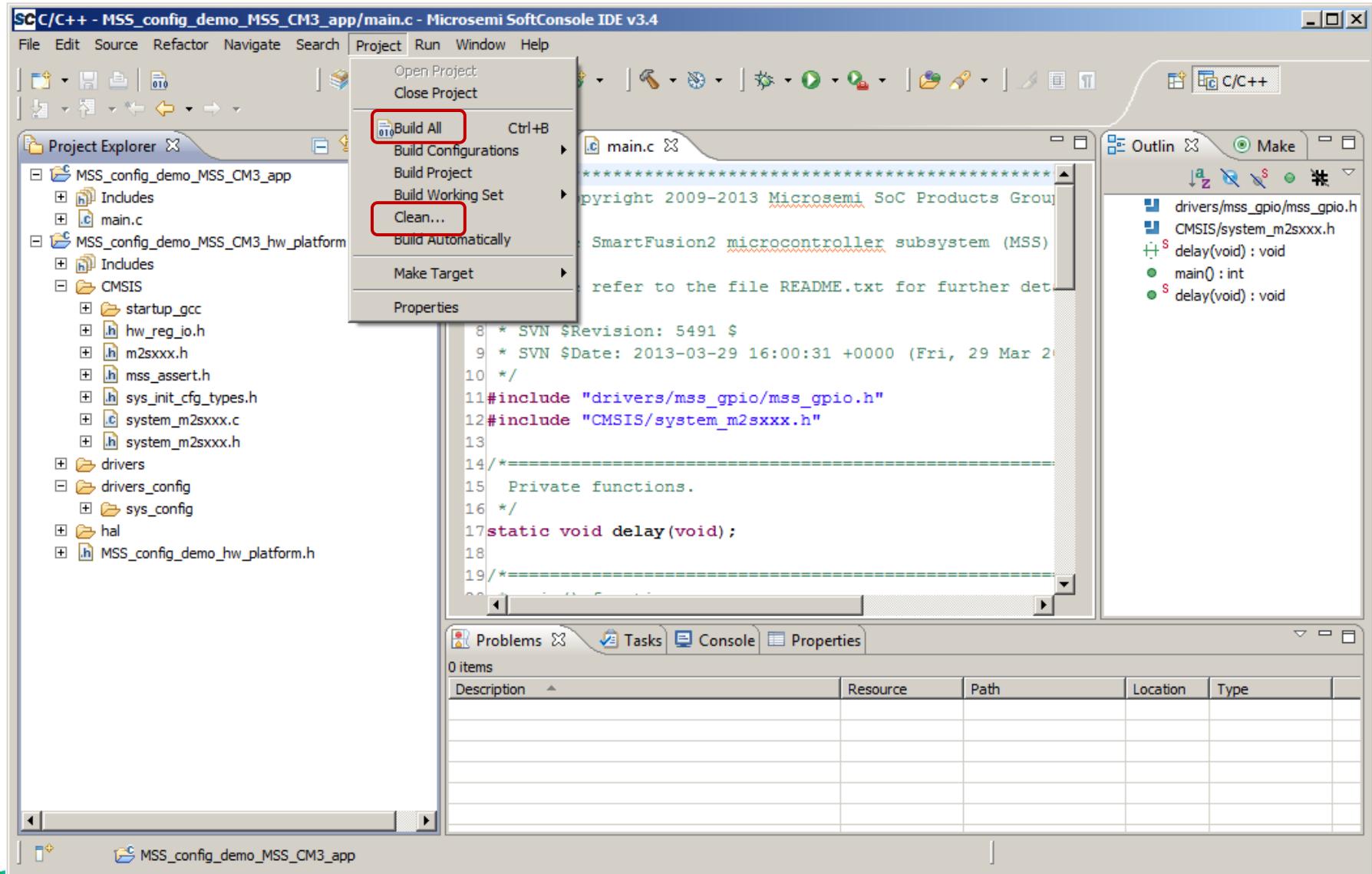
- Builds
 - Work incrementally based on a previous built state
 - Auto-building always uses incremental building for efficiency
- Clean Build (Project > Clean)
 - Discards any existing built state
- Build and Clean Can be Done Over a Specific Set of Projects or the Workspace as a Whole

Building the Project



Building the Project

SoftConsole Menu

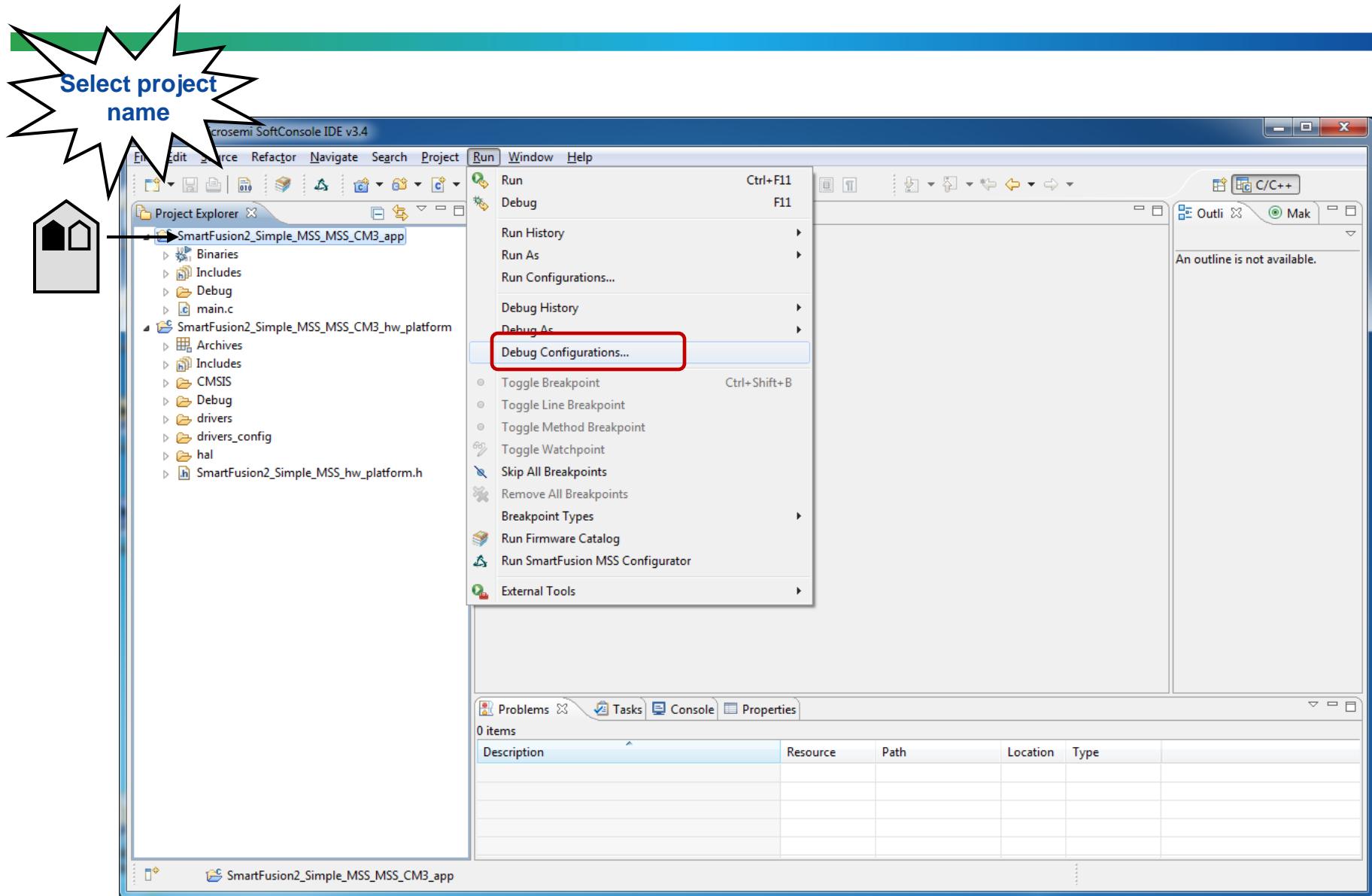


Debugging Applications

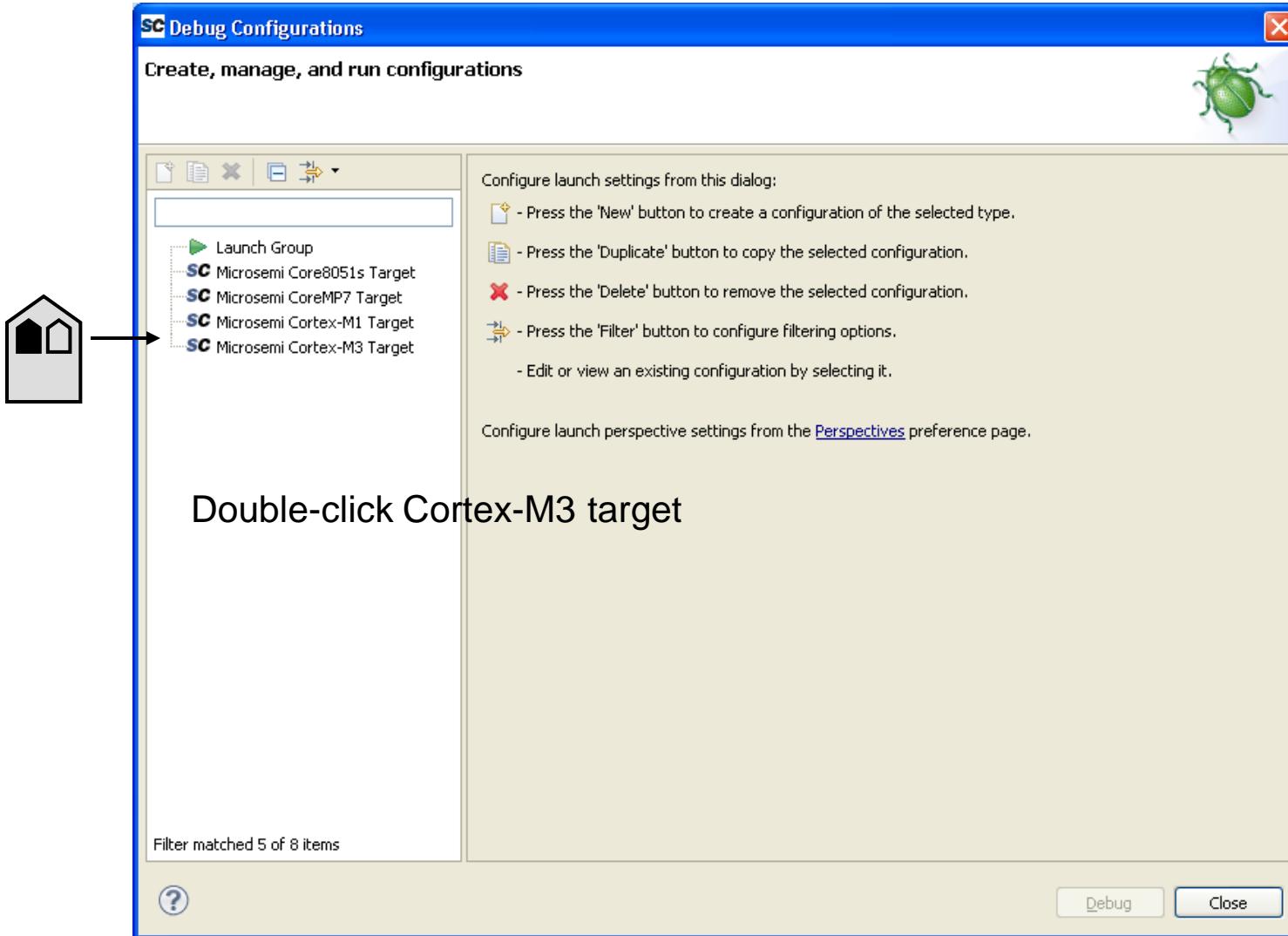
On-Chip Debugging via FlashPro4

- Download and Debug Executable Programs to Cortex-M3 using FlashPro4
 - No ULINK or JLINK required
 - Utilizes Dedicated FPGA JTAG Pins via UJTAG (10 Pins)
- Full Debugging of Code on Remote Target
 - View Internal Registers, Memory Locations, Variables, etc.
- Microsemi FlashPro Programming Software and Drivers Must be Installed

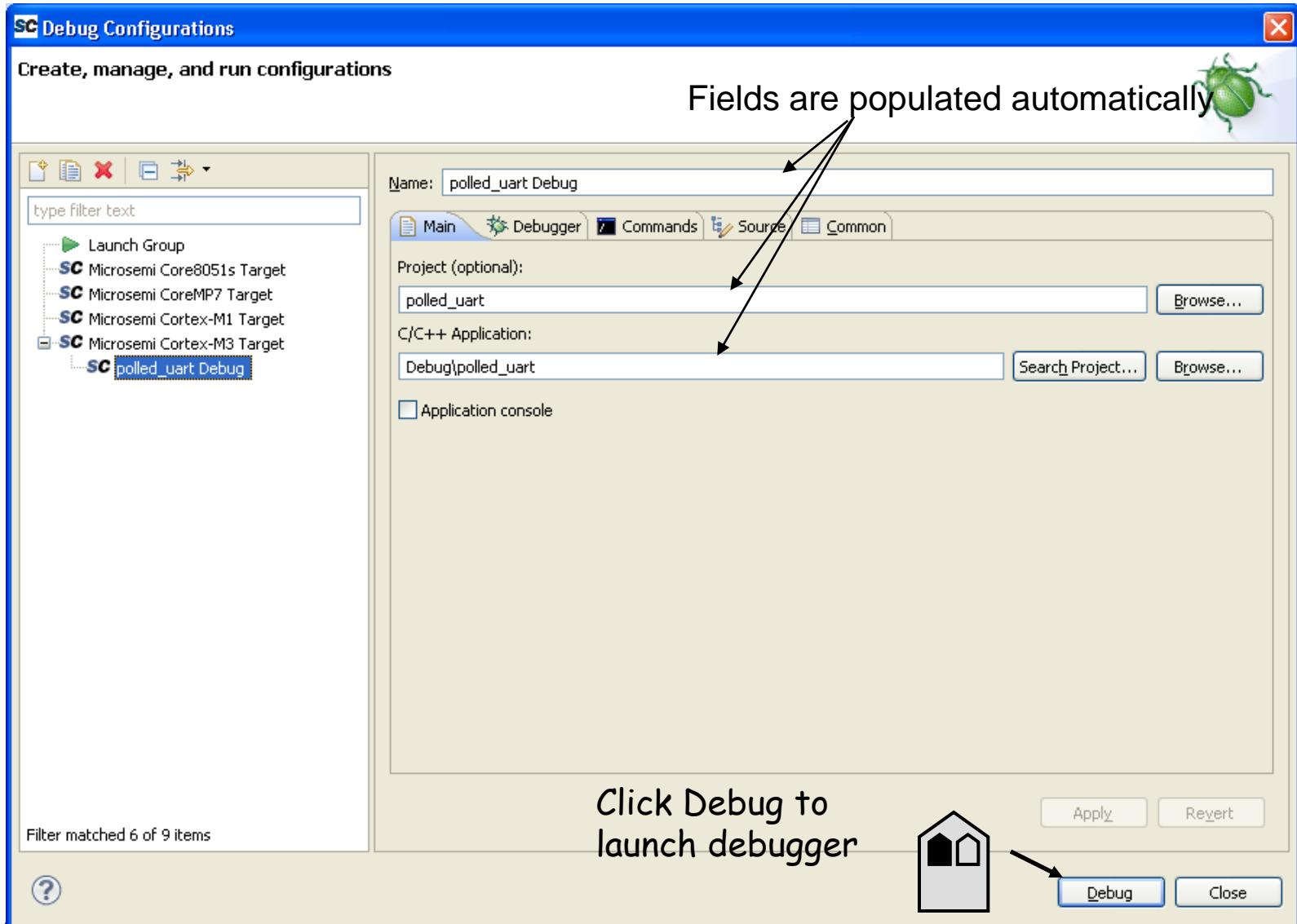
SoftConsole Debug Configuration



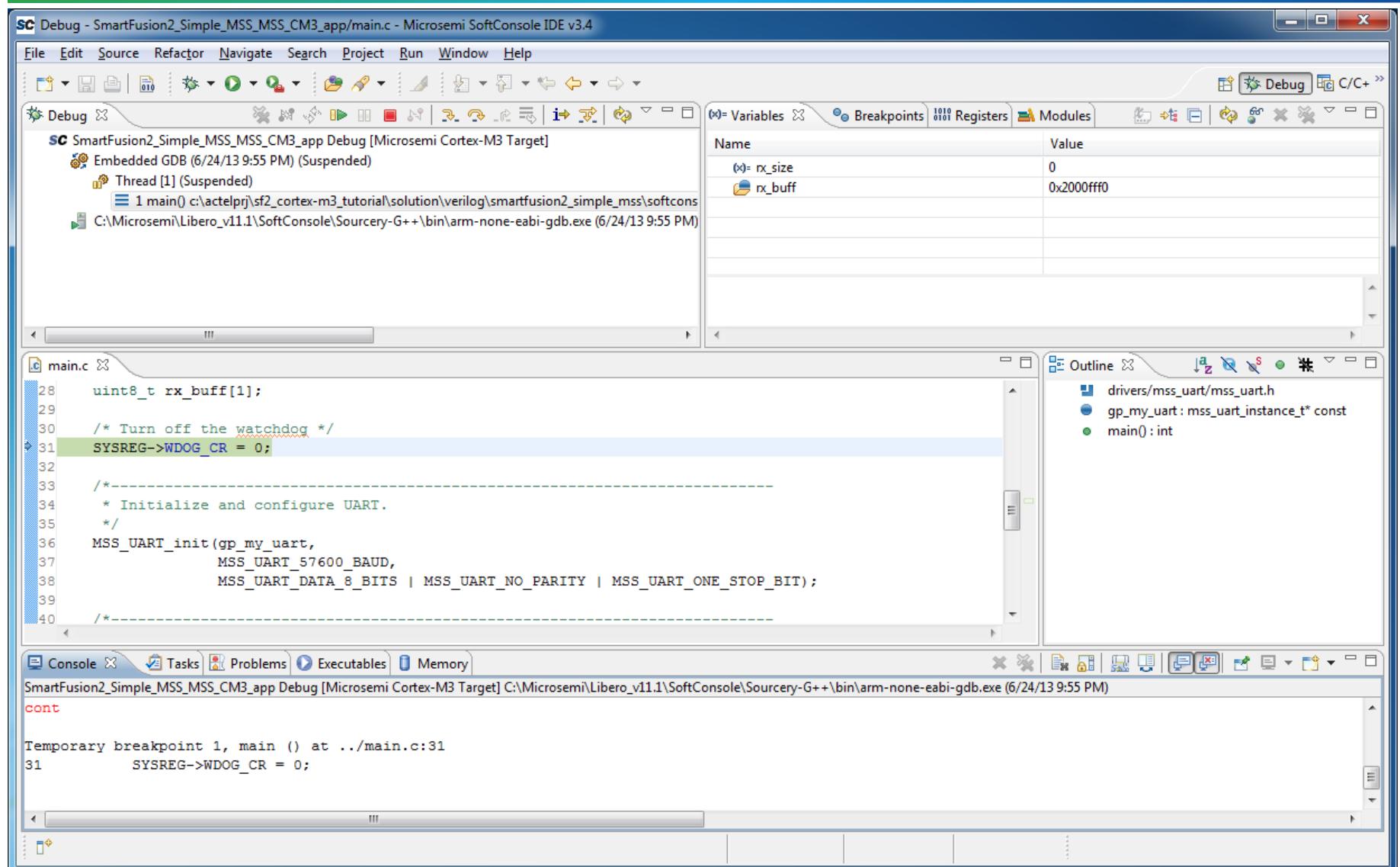
Creating an Embedded Debug Target



New Embedded Debug Target



User Interface: Debug Perspective



Completion of Hands-on Lab1

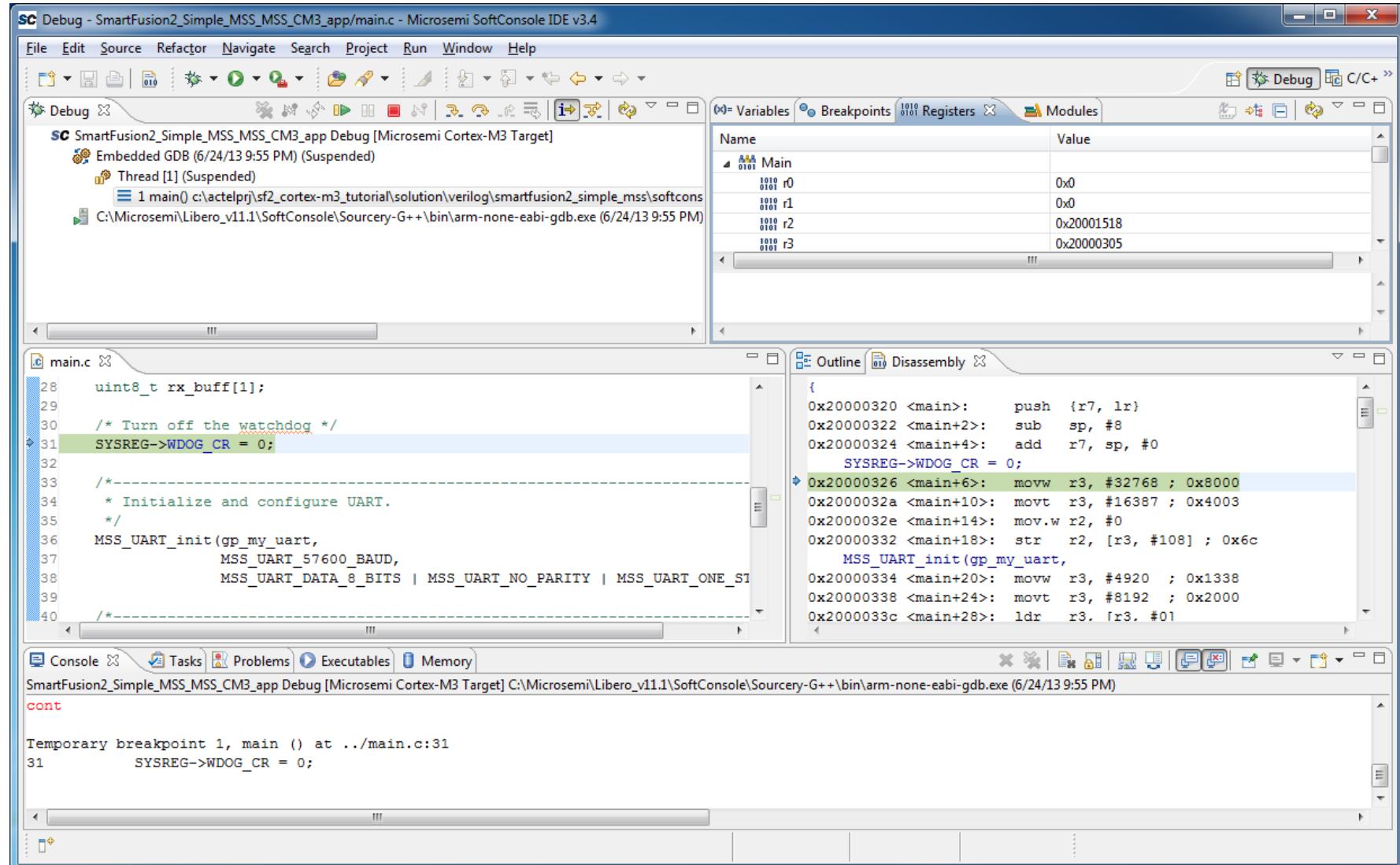
Steps 6 to 9

ARM Cortex-M3 Lab

- Complete steps 6 to 9:
 - Program the SmartFusion2 cSoC on the Starter Kit board
 - Use SoftConsole to download Cortex-M3 applications to the SmartFusion2 target board and debug the applications

NOTE: Cycle power on the board after programming if you are using ES silicon

Running Example in SoftConsole



Code Relocation with SoftConsole

Debug Linker Scripts

- **debug-in-microsemi-smartfusion2-envm.ld**
 - Build/link programs for downloading to and debugging from SmartFusion2 eNVM at 0x60000000 mirrored to 0x00000000 at runtime
- **debug-in-microsemi-smartfusion2-esram.ld**
 - Build/link programs for downloading to and debugging from SmartFusion2 eSRAM at 0x20000000
- **debug-in-microsemi-smartfusion2-external-ram.ld**
 - Build/link programs for downloading to and debugging from external MDDR RAM

debug-in-* linker scripts are used to build/link programs for downloading to and debugging from eNVM, eSRAM and external RAM using SoftConsole

Production Linker Scripts

- **production-smartfusion2-execute-in-place.ld**
 - Build/link programs for execution from eNVM @ 0x00000000 out of reset. This is for production programs which are stored in and execute directly from eNVM
- **production-smartfusion2-relocate-to-external-ram.ld**
 - Build/link programs for relocating executable from internal eNVM to external RAM before starting execution.

production-* linker scripts are for building/linking “production” programs that are stored in eNVM and execute out of reset in a live system

The Intel <project-name>.hex file generated by SoftConsole in the project’s Debug or Release folder is stored in the SmartFusion eNVM Data Storage Client using FlashPro4 or some other method

Programs linked using the **production-*** linker scripts cannot be downloaded or debugged using SoftConsole

Flash Programming with SoftConsole

- SoftConsole can load an executable file into flash program memory for debugging or executing the program in SmartFusion2
- The flash program memory of the target system can external or embedded:
 - SmartFusion2 embedded flash memory (eNVM) @ 0x60000000
 - External flash memory @ 0x70000000
 - No debug support for code which is executing in External Flash since its starting location of 0x70000000 is beyond the Cortex-M3 requirement that hardware breakpoints be placed beneath 0x20000000

Flash Programming Steps

- Build program with appropriate linker script
- Configure debug target to match linker script used
- Launch a debug session to load the executable file to flash program memory
- Messages in the Console view will indicate that flash programming is in progress
 - Flash memory blocks are erased and rewritten only if there is any change to the contents

Suggested SoftConsole Flow

1. Configure the SmartFusion2 MSS and FPGA fabric as required
2. Generate a sample project from the Firmware Catalog as a starting point for development of your application code
 - Alternatively generate the individual firmware cores - SmartFusion CMSIS-PAL, SmartFusion MSS peripheral drivers and the SmartFusion HAL and DirectCore peripheral drivers (for FPGA fabric based DirectCore peripherals) to create your program from scratch
3. (Re)configure the SoftConsole project properties as required
4. Develop your application code while using the **debug-in-microsemi-smartfusion2-esram.1d** or **debug-in-external-ram.1d** linker scripts to simplify recompile/download/debug cycles
5. When application code is nearing completion switch to the **debug-in-microsemi-smartfusion2-envm.1d** linker script to test/debug your code in eNVM
6. Once the application is complete create a production image using the **production-execute-in-place.1d** or **production-relocate-executable.1d** linker scripts, store the Intel Hex file generated by SoftConsole in eNVM using a SmartFusion2 eNVM Data Storage Client and verify that the program executes correctly

Microsemi SmartFusion2 SoC FPGA

**SmartFusion2 SoC FPGAs extend our leadership in
security, reliability and low power into
mainstream applications**

- Leadership in Low Power FPGAs
 - 100X lower static power with same performance
- Leadership in Secure FPGAs
 - State of the art security enables root-of-trust applications
 - Radically transforms the usefulness of FPGAs in security applications
- Leadership in Reliable FPGAs
 - Only SoC FPGA with SEU immune fabric and processor
 - Reliability designed for safety critical and mission critical systems
- Leadership in Real-Time FPGAs
 - ARM® Cortex™-M3 real-time microcontroller
 - Flash*Freeze real-time power management
 - Instant on real-time availability





SMARTFUSION[®]2



Microsemi.

WHEN FAILURE IS
NOT AN OPTION

Most Secure, Highest Reliability, Lowest Power

Back-up Slides

SmartFusion2 Packaging Options

Type	FCS325		VF256		FCS536		VF400		FCV484		TQ144		FG484		FG676		FG896		FC1152	
Pitch (mm)	0.5		0.8		0.5		0.8		0.8		0.5		1.0		1.0		1.0		1.0	
L x W (mm)	11x11		14x14		16x16		17x17		19x19		20x20		23x23		27x27		31x31		35x35	
Device	I/O	Lanes	I/O	Lanes	I/O	Lanes	I/O	Lanes	I/O	Lanes	I/O	Lanes	I/O	Lanes	I/O	Lanes	I/O	Lanes	I/O	Lanes
M2S005		161	-		171	-			84	-	209	-								
M2S010		138	2		195	4			84	-	233	4								
M2S025	180	2	138	2		207	4				267	4								
M2S050	200	2				207	4				267	4					377	8		
M2S090 ²	180	4									267	4	425	4						
M2S150					293	4			273 ¹	4 ¹									574	16

Notes:

1 -Preliminary

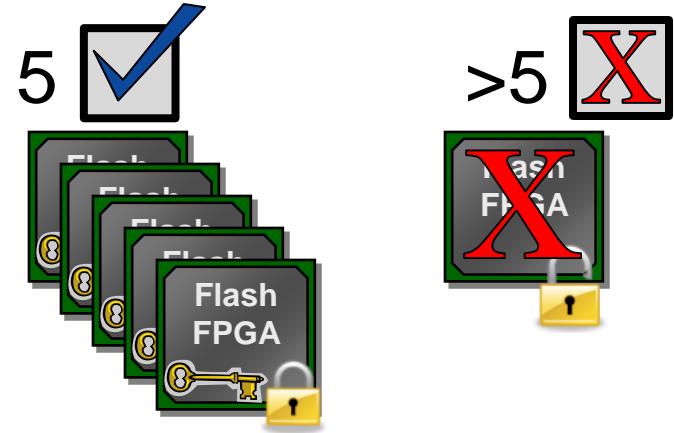
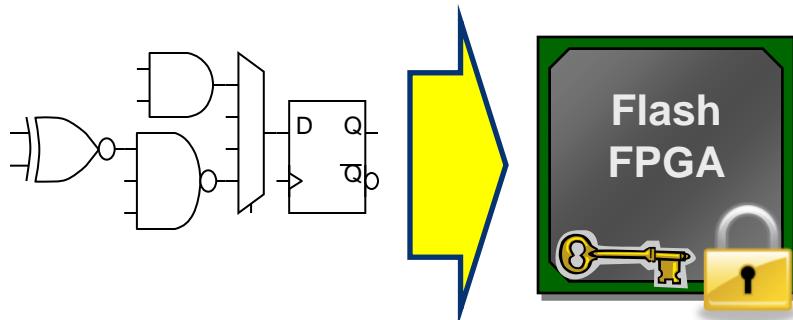
2 - 090 FCS325 is 11x13.5 package dim

Most Secure, Highest Reliability, Lowest power

Security: Design Security vs. Data Security

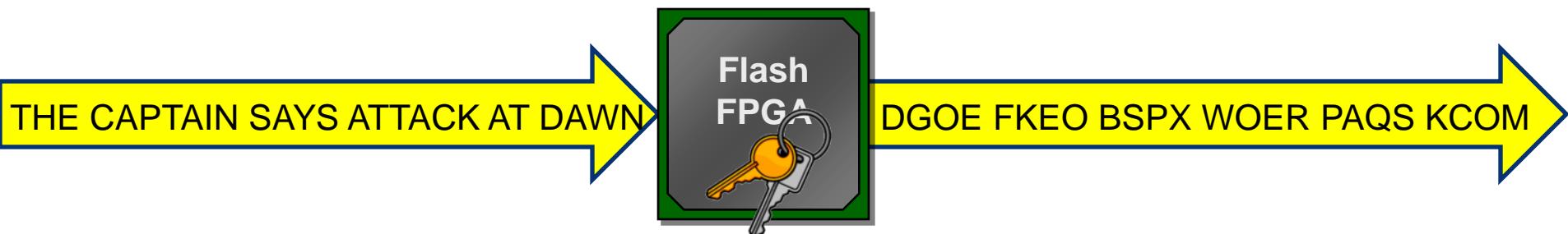
■ Design Security

- Making sure that the *FPGA Design* is protected and the IP owner's security intent is respected



■ Data Security

- The *Application* programmed into the device meets its security objectives (authenticity, confidentiality, integrity, etc.)



Security: Why is FPGA Design Security Important?

- **Design Security**
 - Cloning
 - Someone copies your design without even necessarily having to understand how it works
 - Overbuilding
 - Your contract manufacturer fills your order... then makes a few for himself. After all, he has all the data!
 - Reverse engineering
 - Someone figures out how your design works, then uses or improves on what he learned
 - Counterfeiting
 - Illegal use of your brand name on a work-alike or cloned product
 - Tampering
 - Changing the design for malicious intent
- **Data Security**
 - The data being managed by the device stays secure
 - Without design/device security, it is virtually impossible to provide good data security

Security: Programmable Design Security

- Built-in Design Security on all Devices
 - Protection against tampering, cloning, overbuilding, reverse engineering and counterfeiting
 - No design or manufacturing overhead to build secure devices
 - Supply-chain assurance with digital Certificate of Conformance
- Design Protection
 - Anti-tamper detection with active zeroization
 - Bitstream is always encrypted with AES-256
 - Secure programming with SHA-256 bitstream authentication
 - On-demand BIST for all non-volatile memories
- Security Key Protection
 - Cryptographic Research Incorporated (CRI) DPA resistant technology
 - Intrinsic-ID physically un-clonable function (PUF) for state of the art key protection
 - Non-deterministic random bit generator (NRBG) for key generation
 - Non-volatile key storage in encrypted form

State of the art security enables root-of-trust applications

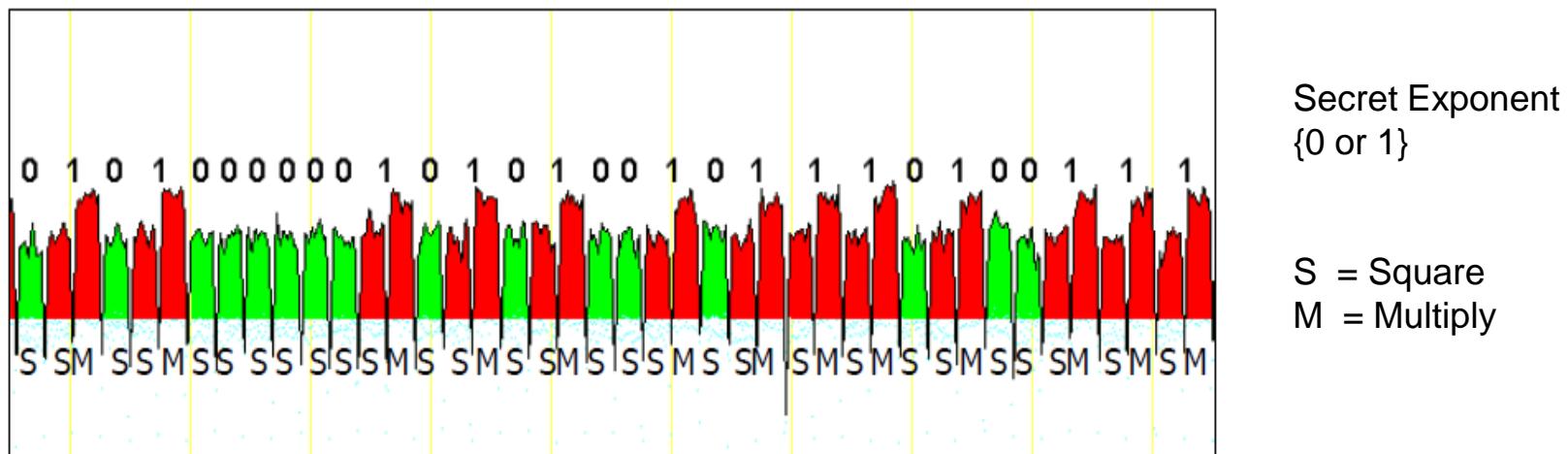
Security: Breakthrough Data Security

- Security Processing Accelerators for Advanced Cryptographic Applications
 - Cryptographic services built into the device accessible to users
 - AES-256, SHA-256, HMAC
 - 384 bit Elliptical Curve Cryptographic (ECC) Engine
 - Pseudo-PUF challenge-response service
 - Non-Deterministic Random Bit Generator with DPA countermeasures
 - CRI-patented key-tree protocol-level construct for DPA-safe designs
 - Hardware firewalls available in the ARM AHB bus matrix
- Protection of Application-level Cryptographic Keys
 - Cryptographic Research Incorporated (CRI) DPA resistant technology
 - Intrinsic-ID's physically unclonable function (PUF) technology

*Radically transforms the usefulness of FPGAs
in security applications*

Simple Power Analysis / Differential Power Analysis

- Extraction of encryption keys through measuring power of cryptographic operations
 - IC power consumption depends on activity of transistors
 - Measurements of device operation can directly reveal keys and other secrets



Crypto algorithms can be functionally correct but susceptible to attack!

They need to be SPA / DPA resistant!

Reliability: Our Devices Deployed at the Highest Levels

Example: Safety Integrity Level (SIL) - IEC 61508

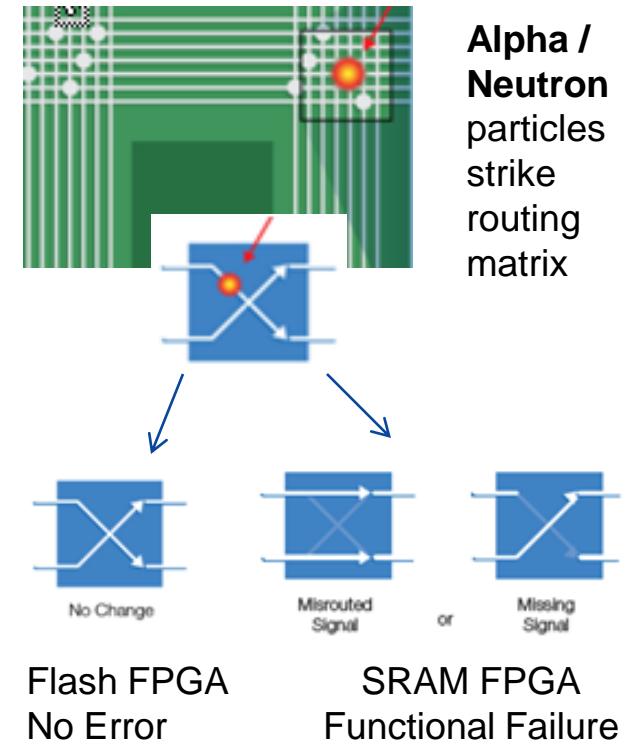
SIL Level	Availability	Probability of Failure	FIT Rate	Consequence	Application Example
4	>99.99%	1 failure in 110,000 yrs	~1	Potential for fatalities in the community	Nuclear Power Plant Control
3	99.9%	1 failure in 11,100 yrs	~10	Potential for multiple on-site fatalities	Hazardous area laser curtain sensors
2	99-99%	1 failure in 1,100 yrs	~100	Potential for major on-site injuries or fatalities	Hazardous liquid flow meter
1	90-90%	1 failure in 110 yrs	~1000	Potential for minor on-site injuries	Thermal Meter

Example: Design Assurance Level (DAL) – DO 254

DAL Level	Occurrence/hour	Classification	Mitigation	Application Examples
A	10^{-9} per flight hour	Catastrophic	Need Redundancy and Dissimilar Technologies	Primary flight controls, navigation
B	10^{-7} per flight hour	Hazardous	Need Redundancy	Secondary flight controls
C	10^{-5} per flight hour	Major	May Need Redundancy	Backup systems
D	10^{-3} per flight hour	Minor	No Redundancy	Announcement systems, maps
E	N/A	No Effect	No Redundancy	In flight entertainment

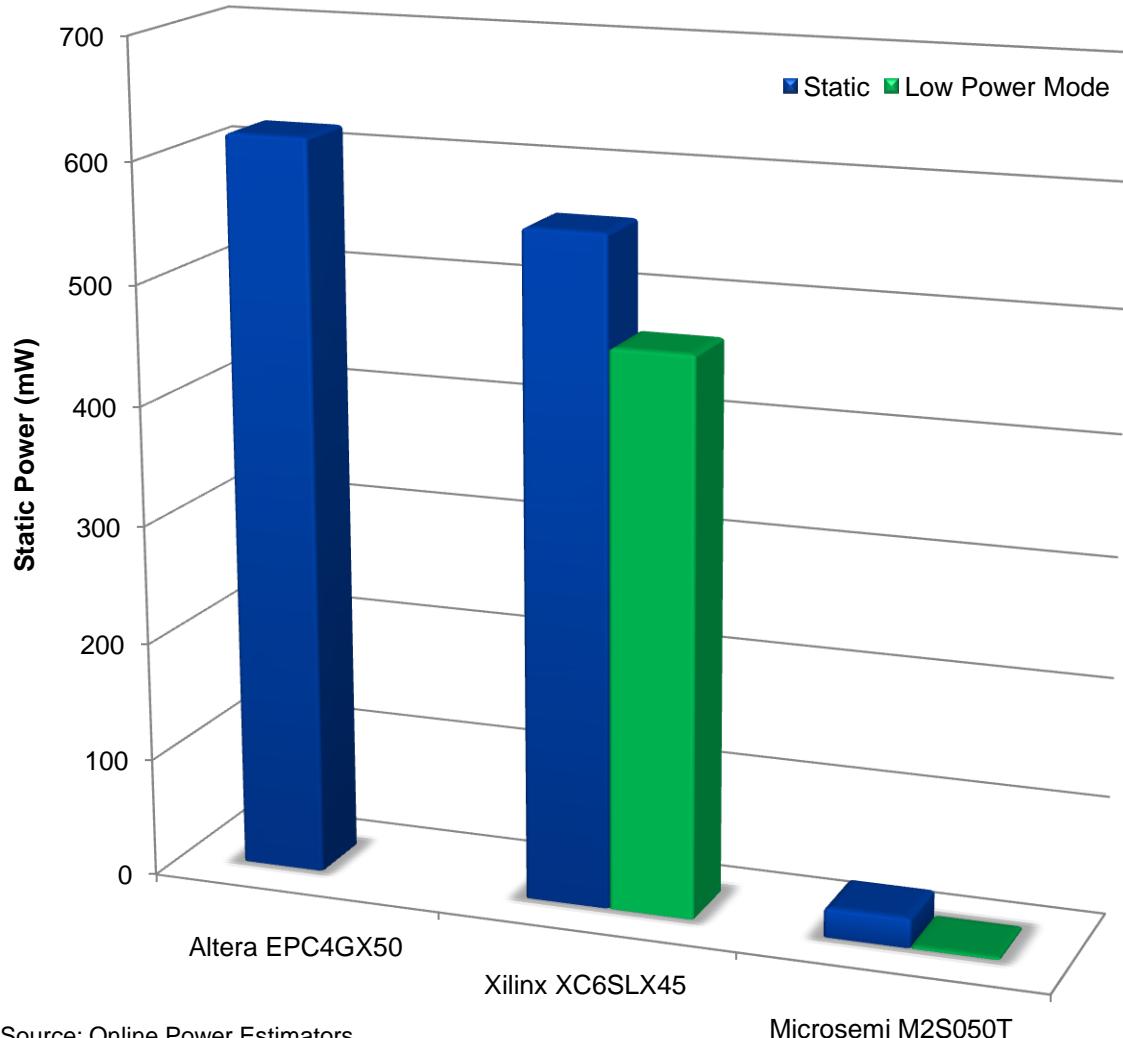
Reliability: The Most Reliable FPGA in the Industry

- Flash FPGA Fabric
 - SEU immune Zero FIT rate configuration
- All SoC Memory SEU Tolerant
 - Thorough built in error detection and correction techniques
 - Cortex-M3 Embedded Scratch Pad Memory, Ethernet, CAN and USB Buffers, PCIe FIFOs
 - Or SEU tolerant implementation
 - DDR Bridges (MSS, MDDR, FDDR), Instruction Cache, MMUART and SPI FIFOs



Reliability for safety critical or mission critical systems

Low Power: Competitive Comparison



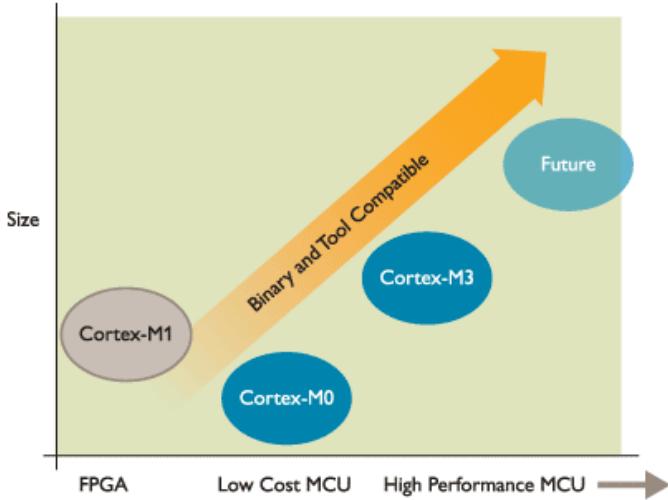
Source: Online Power Estimators

- Conditions
 - Equivalent device size
 - Worst case
 - Industrial

Cortex-M3 Overview

ARM Cortex-M Series Processors

- Cortex-M processors are designed and optimized for cost and gate count sensitive applications
- Typical Applications are MCU, wireless networking (Bluetooth, ZigBee and others), automotive and industrial control systems, white goods and medical instrumentation
- These processors support the Thumb-2 instruction set only



FEATURE SUMMARY							
	Cache Size (Inst/Data)	Tightly Coupled Memory	Memory Mgt	Bus Interface	Thumb	DSP	Jazelle
ARM Cortex-M0					Yes	No	No
ARM Cortex-M1		Yes	-	AMBA AHB-Lite + APB	Yes	No	No
ARM Cortex-M3	-	-	MPU (optional)	3x AHB-Lite + APB	Yes	No	No
ARM Cortex-R4 (F)	0K-64k	Variable	MPU	AMBA 3 AXI	Yes	Yes	No
ARM1156T2(F)-S	Variable	Yes	MPU	3xAXI	Yes	Yes	No
ARM7EJ-S	-	-	-	Yes	Yes	Yes	Yes
ARM7TDMI	-	-	-	Yes**	Yes	No	No
ARM7TDMI-S	-	-	-	Yes	Yes	No	No
ARM946E-S	Variable	Yes	MPU	AHB	Yes	Yes	No
ARM966E-S	-	Yes	-	AHB	Yes	Yes	No
ARM968E-S	n/a	Yes	DMA	AHB-Lite	Yes	Yes	No

Cortex-M3 Highlights

- Harvard architecture – separate code and data busses
- Interrupt controller
- SYSTIC timer for RTOSs
 - Don't need to give up a timer
- New Thumb2 instruction set
- 3 stage pipeline with branch speculation
- Support for Bit manipulation
- Designed to be written in C – no assembly required

ARM Cortex-M3: Designed for Ease of Use

- Ease of use - Coding
 - Everything can be written in C
 - No need for assembler for startup code and interrupt handlers
 - Easy to use atomic bit manipulation
- Excellent real-time performance
 - Lowest interrupt latency ARM
 - Integrated Nested Vectored Interrupt Controller (NVIC)



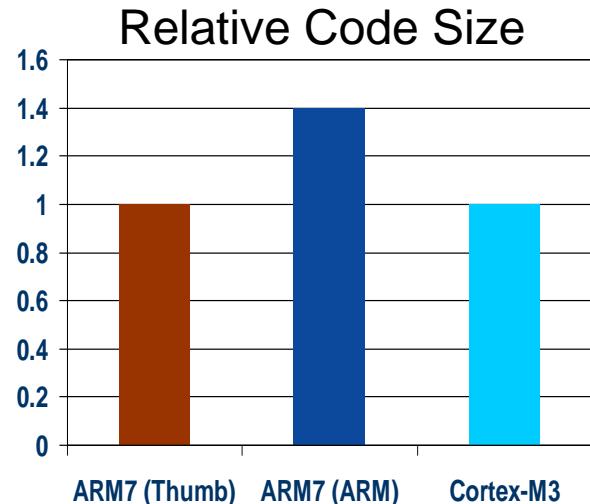
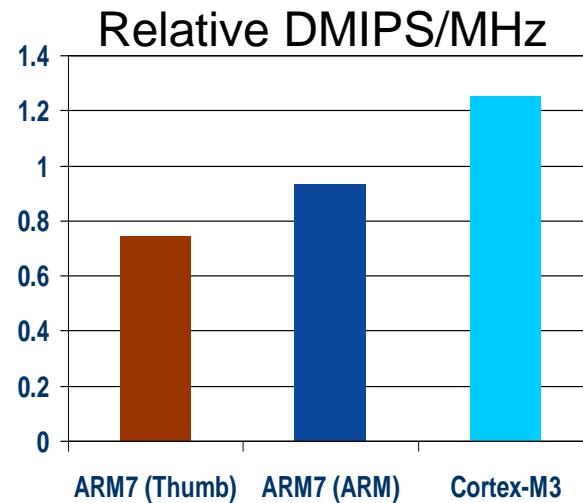
The screenshot shows a software interface for a Microsemi device. At the top, there's a menu bar with options like File, Edit, View, Object, Debug, Run, Preferences, Tools, gDB, Window, Help. Below the menu is a toolbar with icons for various functions. The main window is divided into several panes:

- Assembly Editor:** Shows assembly code for a file named 'main.s'. The code includes instructions like LDR, ADD, SUB, and MOV, along with comments explaining the implementation of patches used by printf functions.
- Memory Dump:** A large pane showing memory contents at address 0x00000000. It displays data in hex, decimal, and ASCII formats.
- EBI: External Bus Interface:** A configuration window with tabs for Control, Memory Control, and Analog Input. It shows memory mapping details like Base Address, PAGES, DBW, NWS, TDF, BAT, and CSEN. It also includes settings for External Memory (EBI_CSR0) and Memory Control (EBI_MCR).
- DACO: Digital/Analog Convert...**: A configuration window for the DAC module. It includes tabs for Control, Mode, Data Holding & Output, Interrupt Mask & Status, and Analog Output. It shows DAC0_CR, DAC0_MR, DAC0_DHR, DAC0_DDR, DAC0_IMR, DAC0_SR, and DAC0_DR.
- ADC: Analog/Digital Converter 1**: A configuration window for the ADC module. It includes tabs for Control, Mode, Data Holding & Output, Interrupt Mask & Status, and Analog Output. It shows ADC1_CR, ADC1_MR, ADC1_DHR, ADC1_DDR, ADC1_IMR, ADC1_SR, and ADC1_DR.

ARM Cortex-M3: Designed for Performance

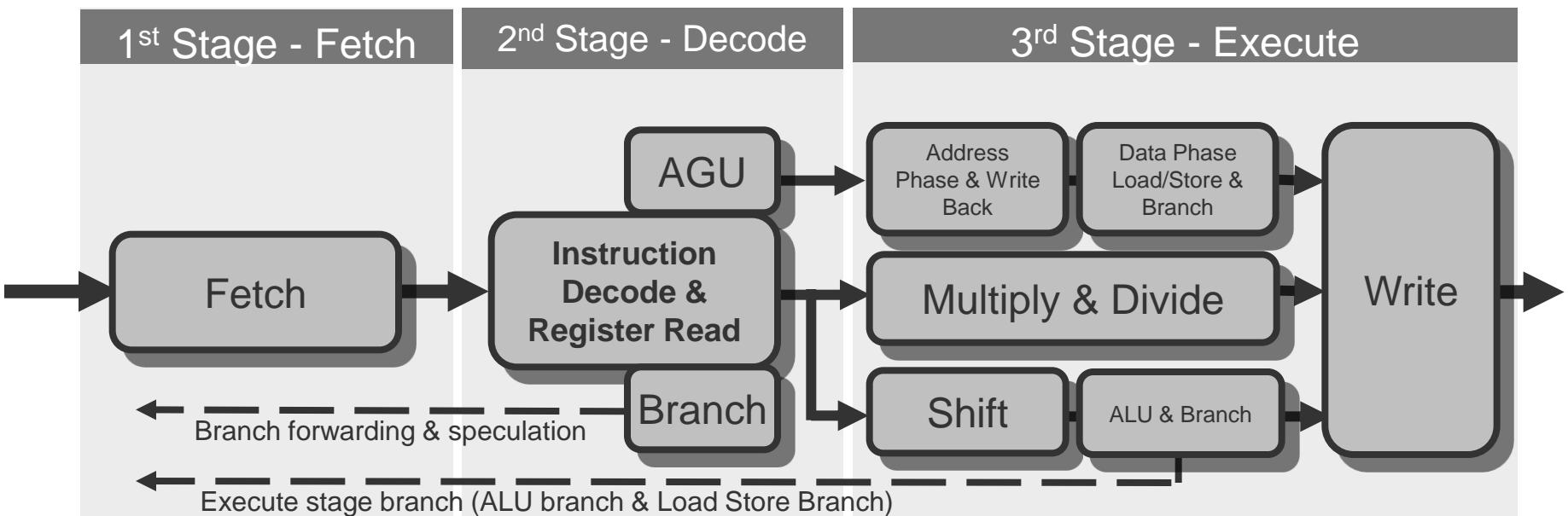
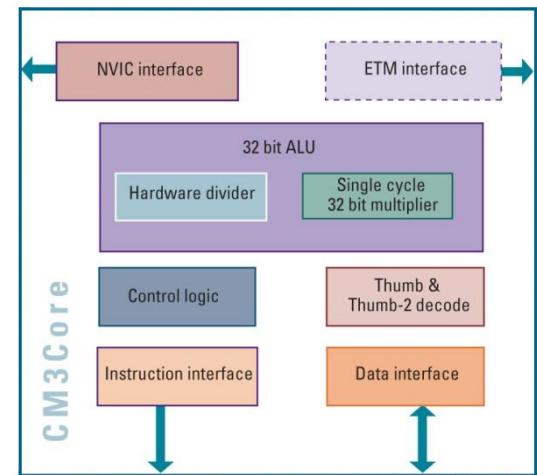
- High Performance
 - High efficiency processor core – 1.25 DMIPS/MHz
 - Advanced instructions for data manipulation
 - Single Cycle Multiply
 - Hardware Division
 - Bit Field Manipulation
 - Exceptional performance at low frequency
- Excellent Code Density
 - Thumb-2 Instruction Set Architecture (ISA)
 - Compiler-chosen mix of 16-bit and 32-bit instructions
 - Performance of ARM (32-Bit) code
 - Code density of Thumb (16-Bit) Code

DMIPS – Dhrystone performance normalized to 1MHz Vax11/780



Central Core Overview

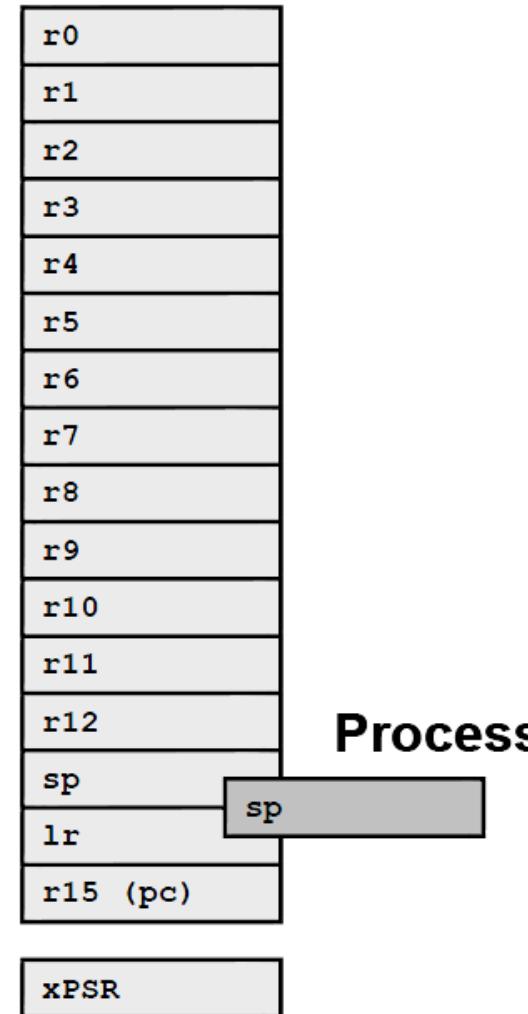
- Harvard architecture
 - Separate Instruction & Data buses enable parallel fetch & store
- Advanced 3-Stage Pipeline
 - Includes Branch Forwarding & Speculation



Cortex-M3 Register Set

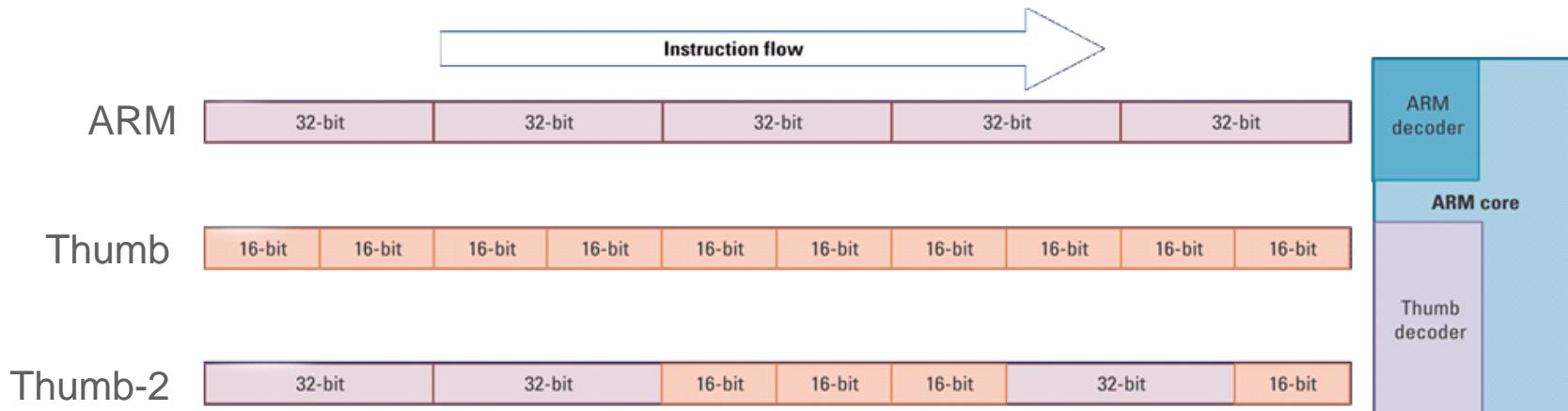
- Very compiler friendly
 - Simpler
 - Load/Store Architecture
 - 32-bit registers
 - Linear 32-bit address space
 - No data or data pages
- Cortex-M3 can run in Supervisor and User modes
 - Or all code can run with the same privilege
 - Exceptions always execute in supervisor mode
- Supervisor and User can have separate stacks
 - Or just one stack

Main



Thumb-2 Instruction Set

- **Thumb-2 ISA was introduced in ARMv7 architecture**
 - Original 16-bit Thumb instructions maintain full compatibility with existing code
 - New 16-bit Thumb instructions for improved program flow
 - New 32-bit Thumb instructions for improved performance and code size
 - One 32-bit instruction replaces multiple 16-bit opcodes
 - 32-bit instructions are handled in the same mode ~ no interworking required

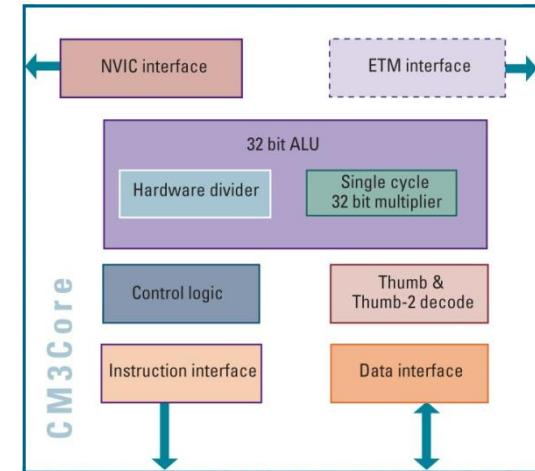


Advanced Instruction Capability

- Single Cycle Multiply
 - Significantly improves performance in intensive math functions

Source	Destination	Cycles
16b x 16b	32b	1
32b x 16b	32b	1
32b x 32b	32b	1
32b x 32b	64b	3-7*

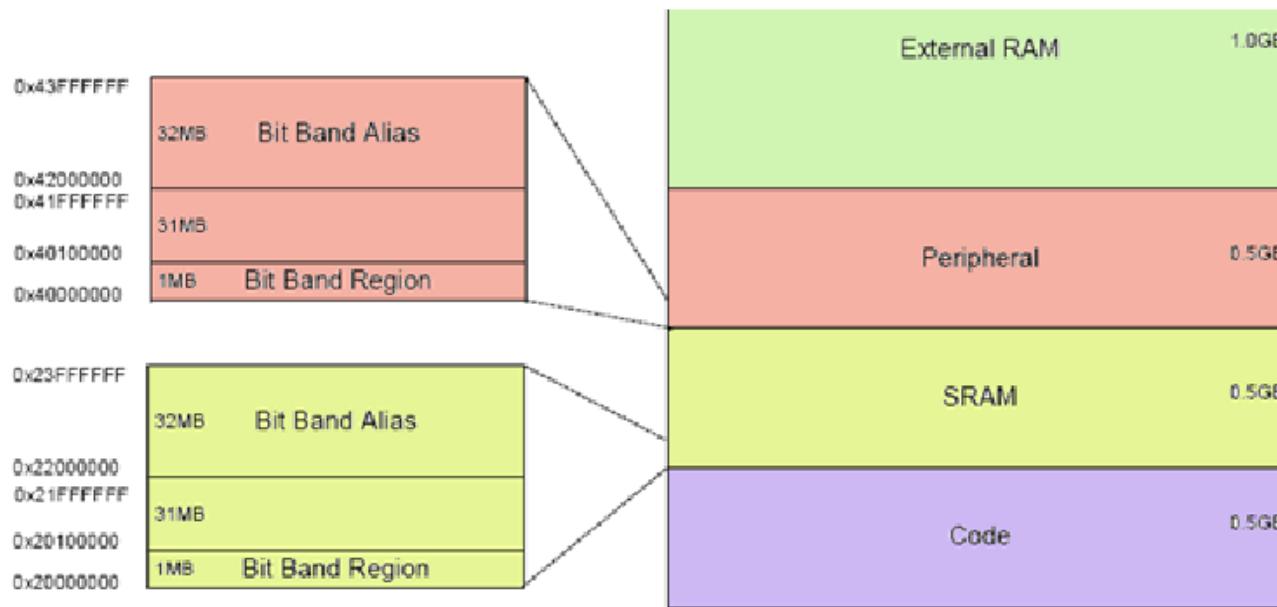
*Long Multiplies are interruptible



- Hardware Division
 - New Thumb-2 instructions UDIV & SDIV (Unsigned or Signed divide)
 - Instructions take between 2 & 12 cycles depending on dividend and divisor
 - The closer the dividend and divisor, the faster the instruction completes
 - Instruction is interruptible (abandoned/restarted)

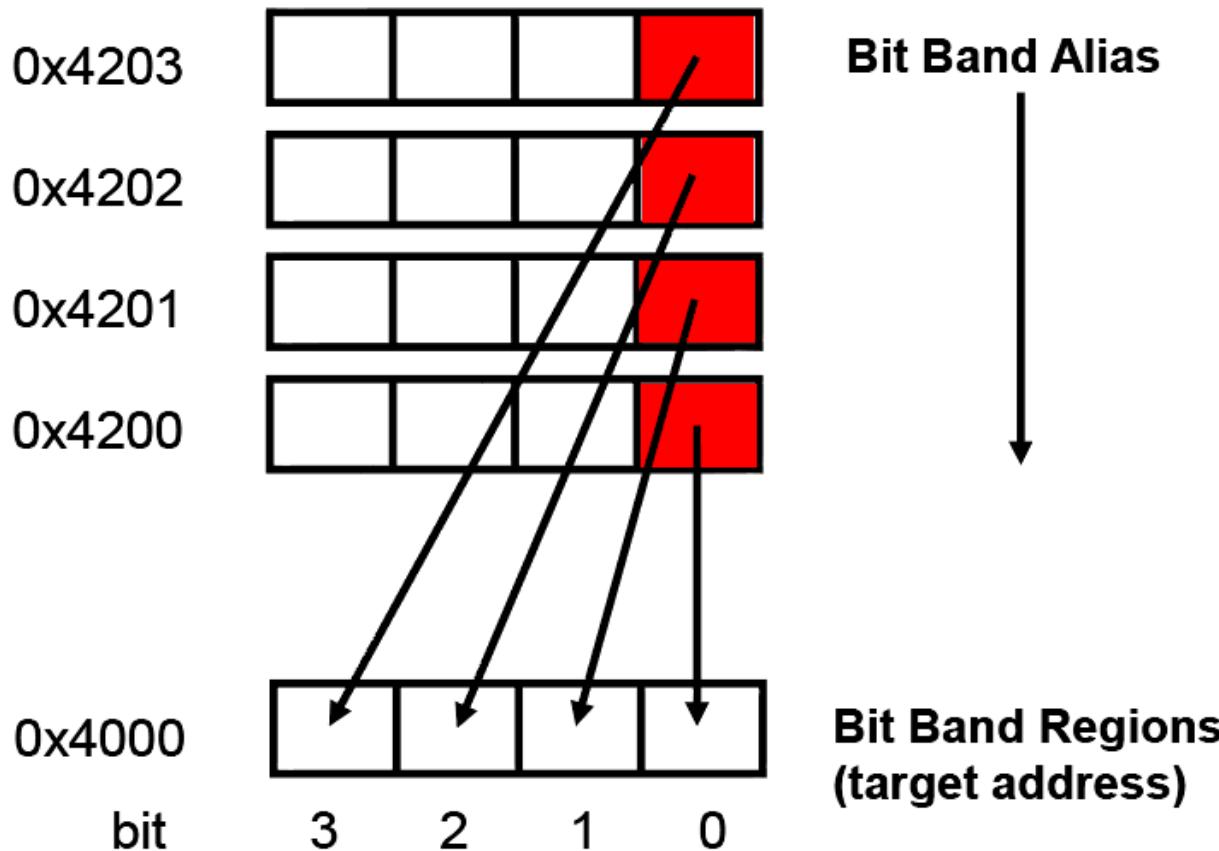
Bit Manipulation

- Uses the LSBit at a 32 bit address to set/clear a bit
 - All other 31 bits are don't care/not used
 - We now define Bit band Alias and Bit Band Regions
 - Alias addresses “point” into the 32 bit region addresses
 - 1 of these 32 bit alias addresses for each bit at one band address
 - This means for every address in the band region, are 32 aliases



4 bit Example ...

- Uses the LSBit to set/clear a bit in a target address:



Exception Handling

- Optimized for low latency and good interrupt performance
- Automatic save and restore of processor registers
 - {PC, xPSR, R0-R3, R12, R14}
 - Allows handler to be written entirely in ‘C’
- Multi-cycle instructions interruptible for low interrupt latency
 - LDM/STM continued on return from interrupt
 - Other instructions restarted
- Exceptions processed in Handler mode
 - Supervisor privilege

Cortex-M3 Exceptions

- Exception handling order is defined by programmable priority
- Cortex-M3 supports up to 239 external interrupts

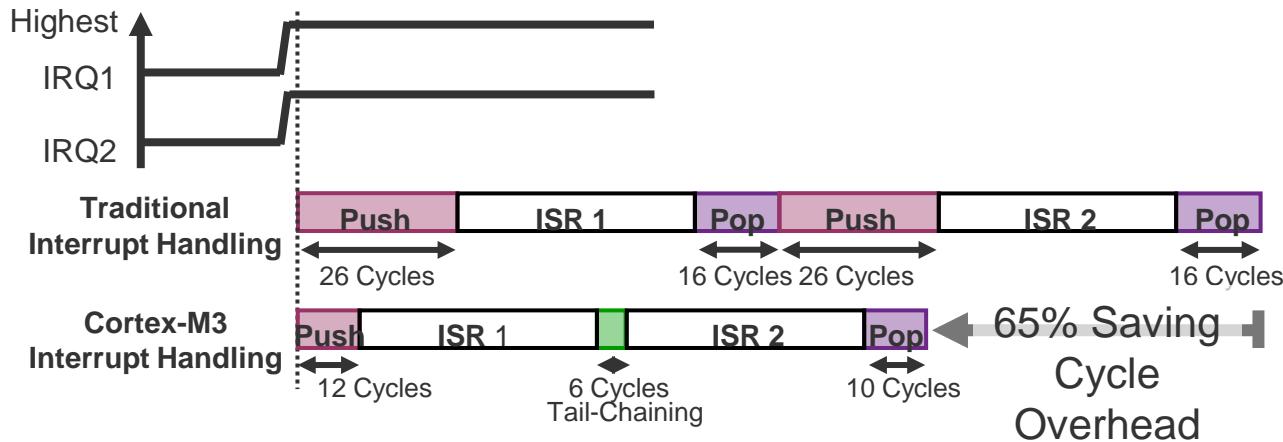
	Exception	Name	Priority	Descriptions
Fault Mode & Start-up Handlers	1	Reset	-3 (Highest)	Reset
	2	NMI	-2	Non-Maskable Interrupt
	3	Hard Fault	-1	Default fault if other handler not implemented
	4	MemManage Fault	Programmable	MPU violation or access to illegal locations
	5	Bus Fault	Programmable	Fault if AHB interface receives error
	6	Usage Fault	Programmable	Exceptions due to program errors
System Handlers	11	SVCall	Programmable	Supervisor Call
	12	Debug Monitor	Programmable	Break points, watch points, external debug
	14	PendSV	Programmable	Pendable Supervisor Call for System Devices
	15	Systick	Programmable	System Tick Timer
Custom Handlers	16	Interrupt #0	Programmable	External Interrupt #0

	47	Interrupt #31	Programmable	External Interrupt #31

Interrupts

■ Interrupt latency for Cortex-M3

- 12 cycles
- 6 cycles if tail-chaining implemented
 - Tail-chaining is back-to-back processing of exceptions without the overhead of state saving and restoration between interrupts. The processor skips the pop of eight registers and push of eight registers when exiting one ISR and entering another because this has no effect on the stack contents.

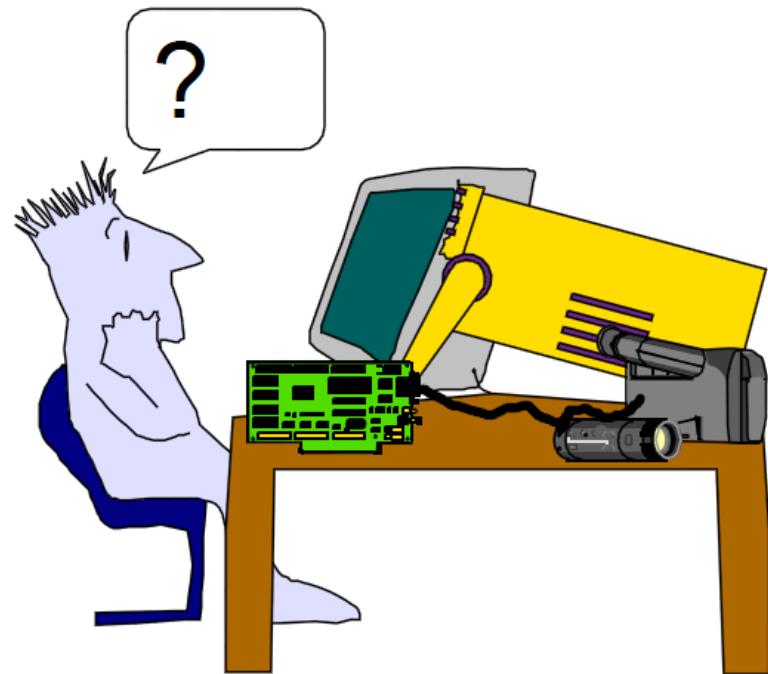


System Timer - SysTick

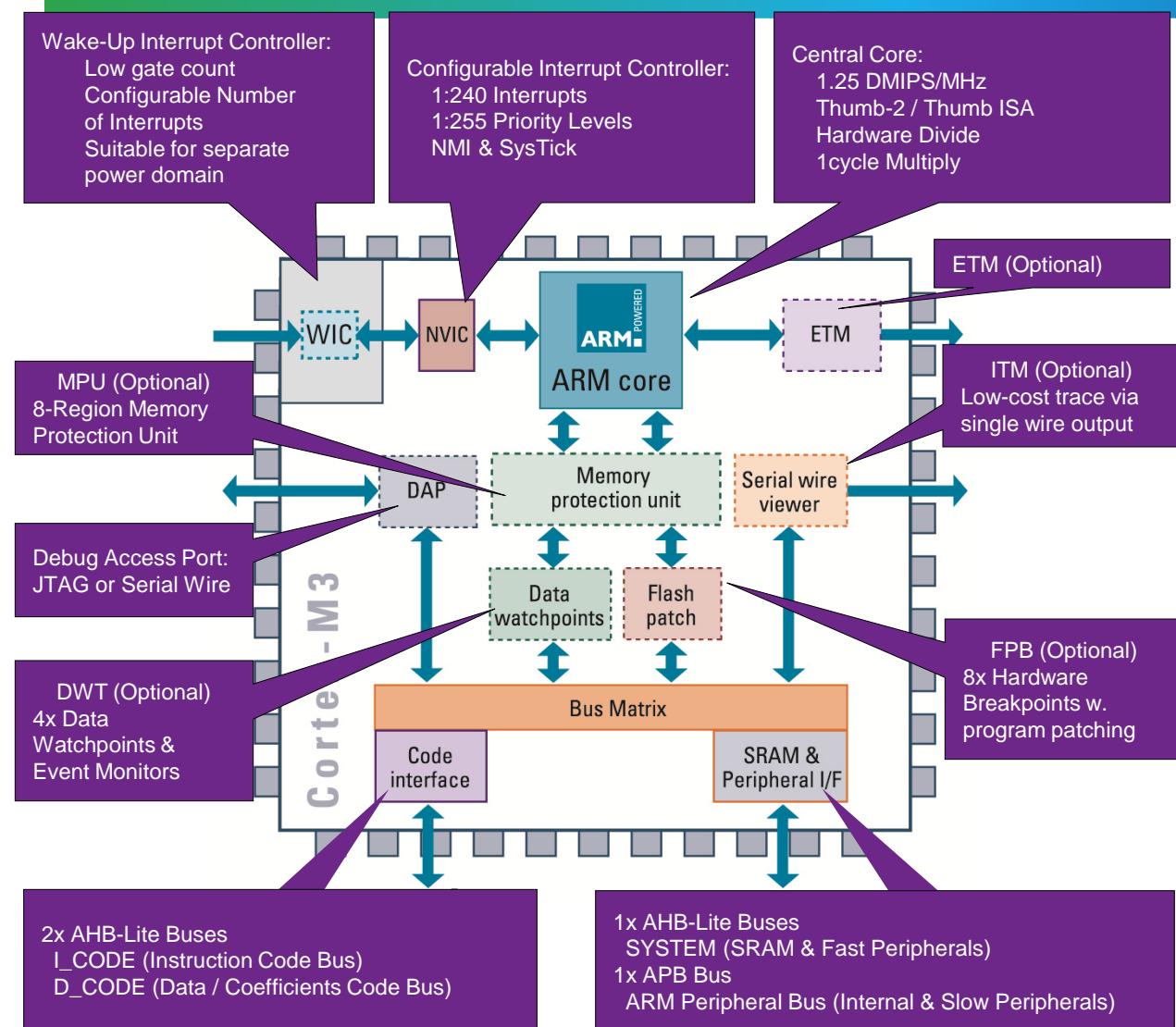
- Dedicated Timer for RTOS clock inside Cortex-M3
 - Don't have to use up a regular timer
 - Really easy to use. See MDK examples.
 - Many RTOSs have already ported to SmartFusion

Debug and Trace

- Cortex-M3 has Serial Wire Debug, Serial Wire Viewer
- Displays:
 - Read, write operations
 - PC Samples
 - Exceptions (including interrupts)
 - CPU counters
 - All timestamped! ALL in real-time.



Cortex-M3 Processor Feature Summary



Processor Gate Count

Block	Gates
CM3Core	32K
Bus Matrix	3.7K
NVIC (1 Int.)	4K
Minimal Processor	<45Kgates*

Optional Components Gate Count

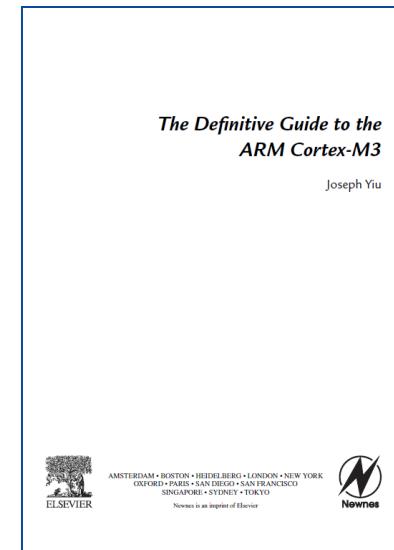
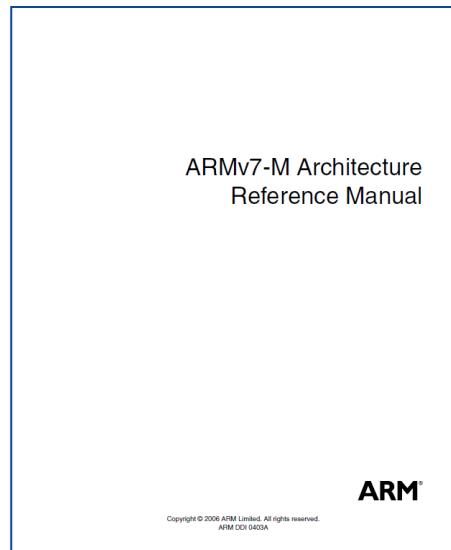
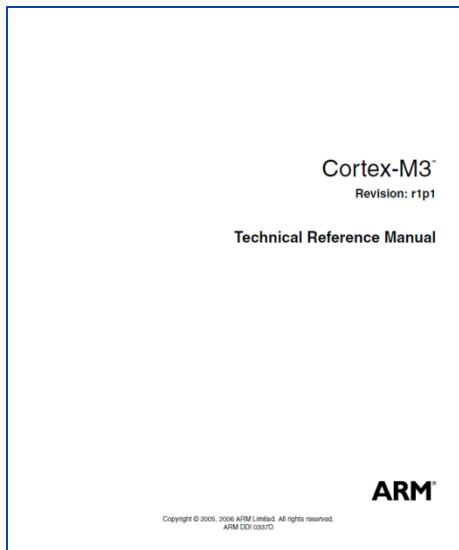
Block	Gates
NVIC (16 Int.)	4K**
DAP / SWJ	1.9K
DWT	2K (1of)* 6.8K (4of)
FPB	1K (2of)* 2.9K (8of)
ITM	3.4K
MPU	17K
ETM	7.5K

*Approximate Gate Count with r2p0 release

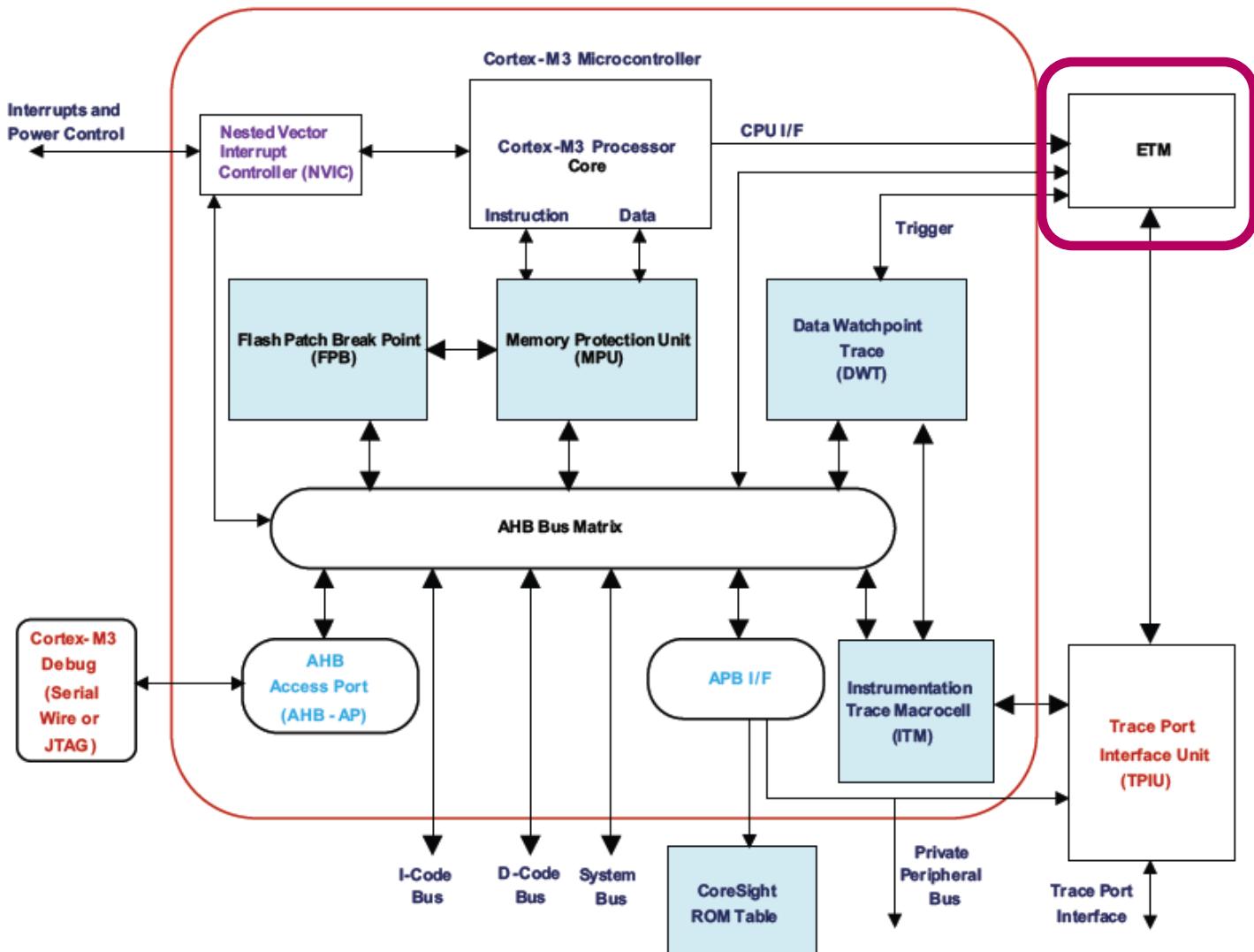
** Additional to basic NVIC implementation

Cortex-M3 Recommended Reading

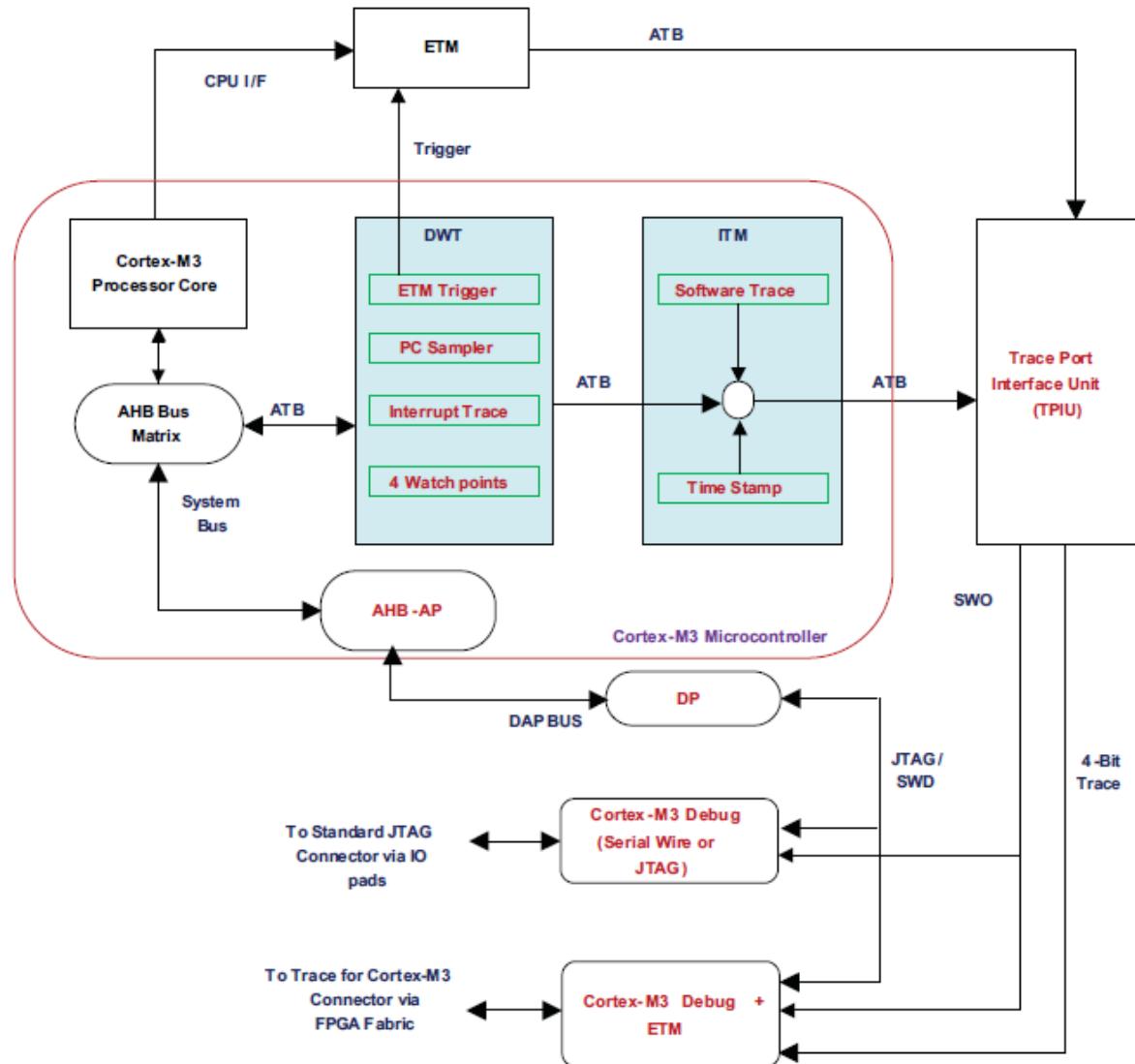
- Available from the ARM Information Center (<http://infocenter.arm.com>)
 - Cortex-M3 Technical Reference Manual
 - ARMv7-M Architecture Reference Manual
 - ARMv7-M Architecture Application Level Reference Manual
- *The Definitive Guide to the ARM Cortex-M3* by Joseph Yiu



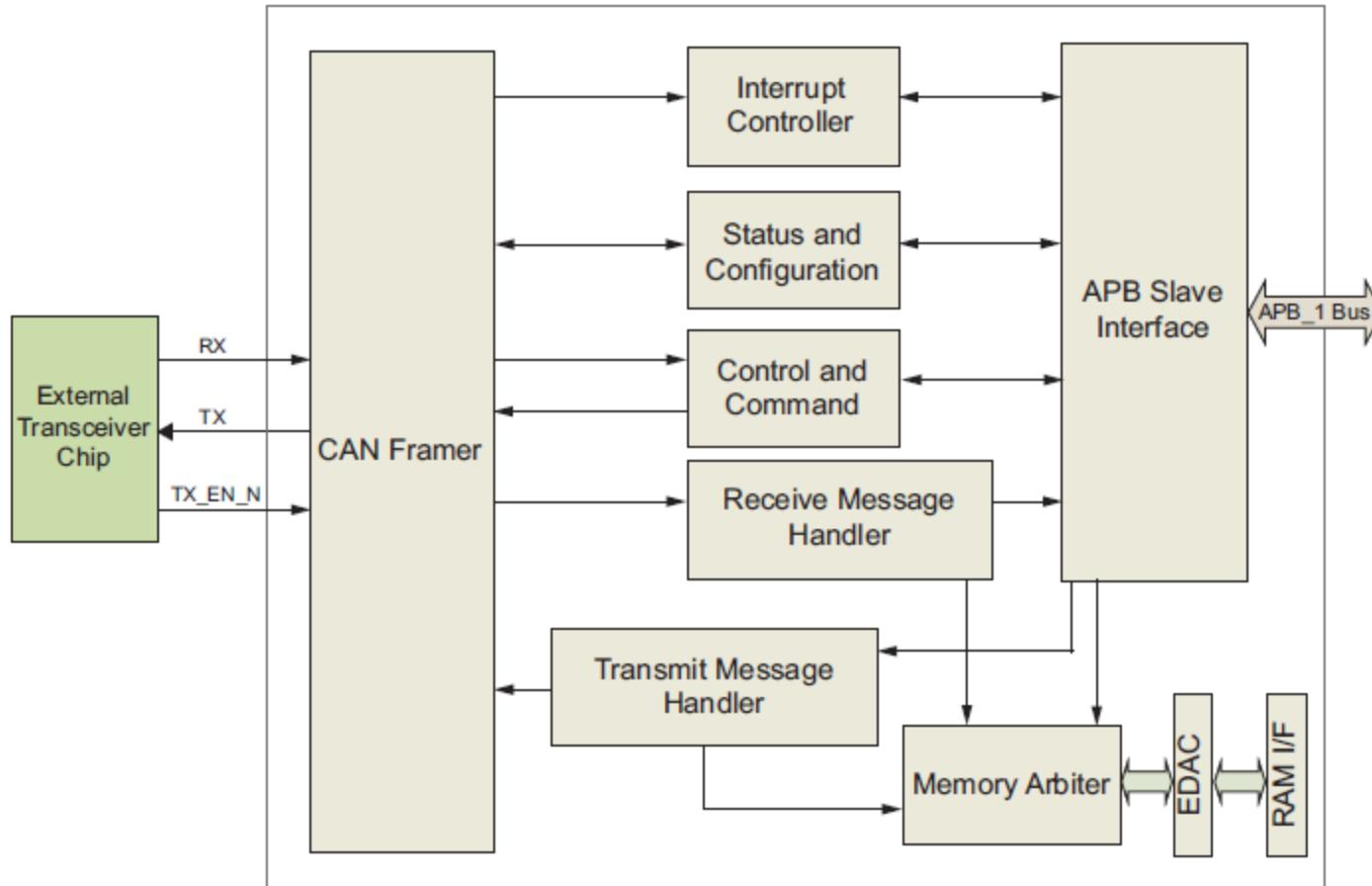
Cortex M3 R2P1 Block Diagram in SF2



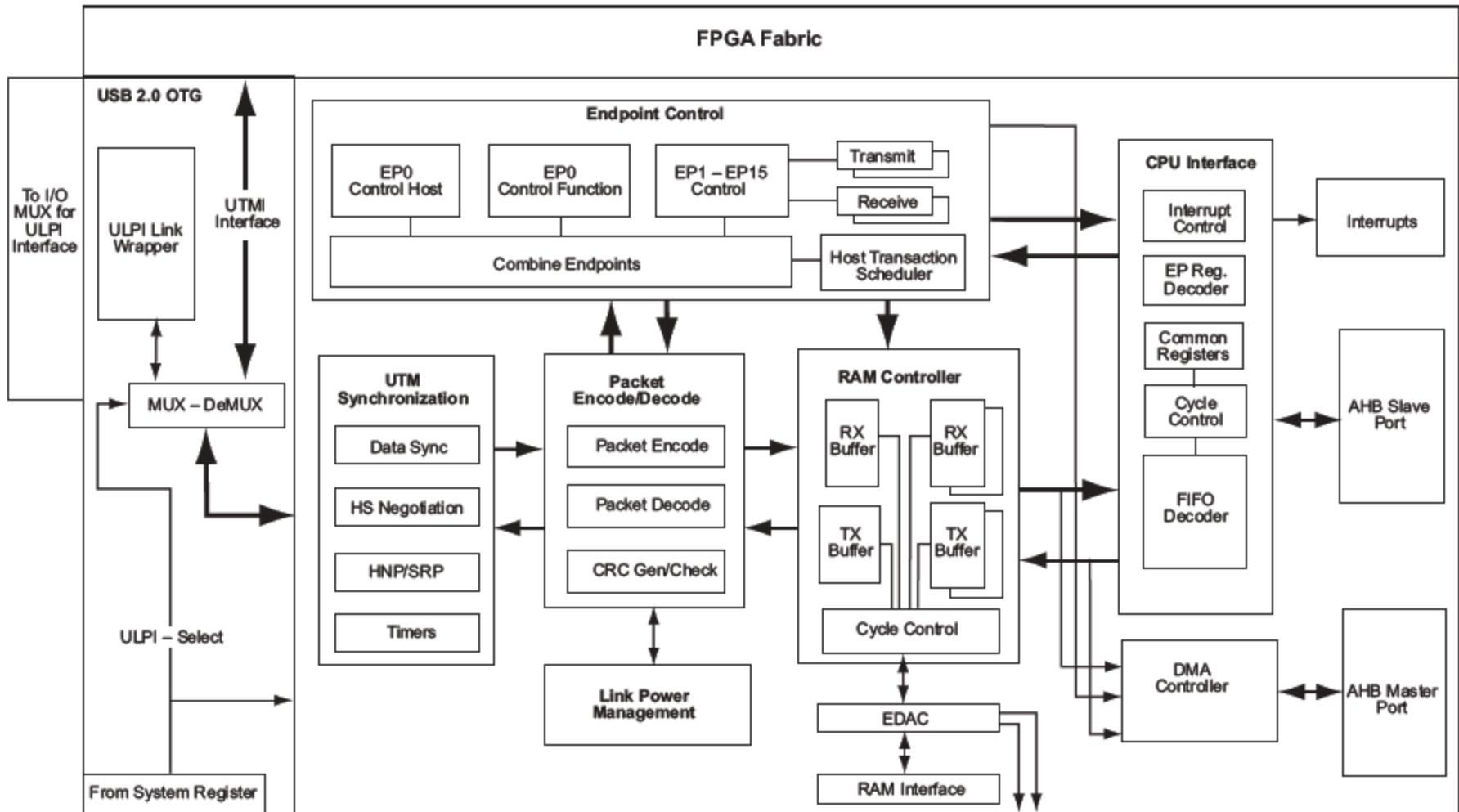
Trace System Block Diagram



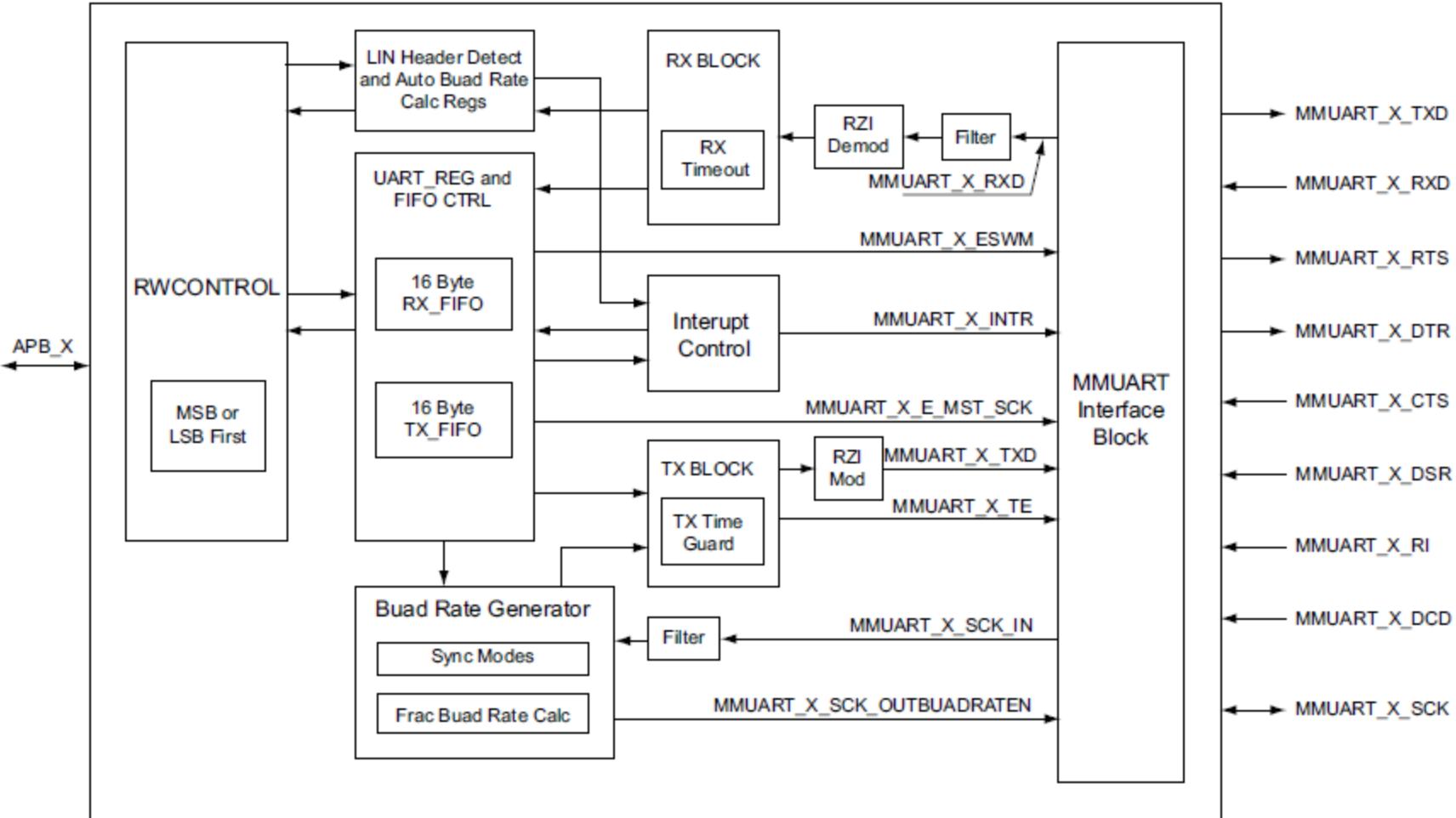
CAN Controller



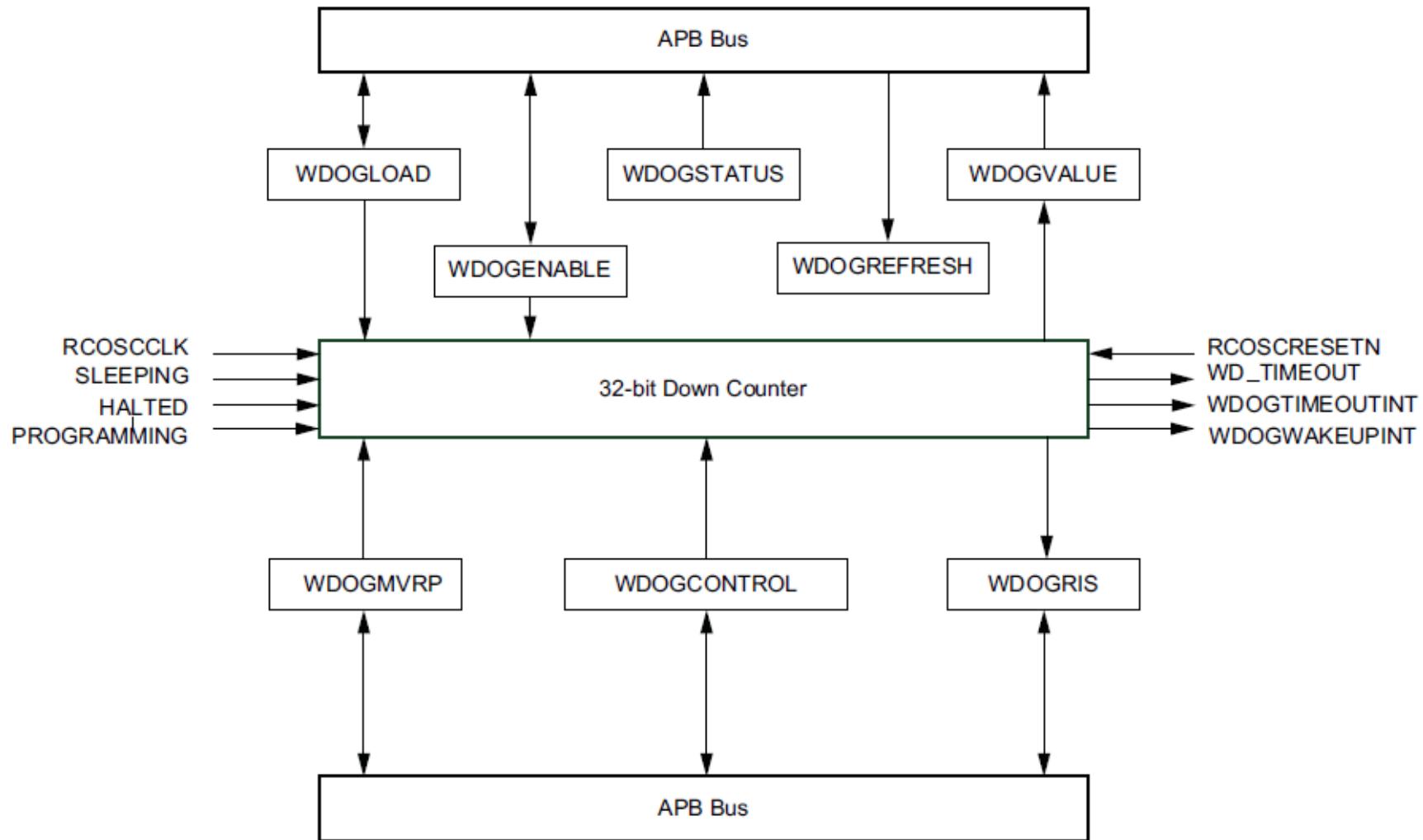
USB Controller



MMUART Block Diagram

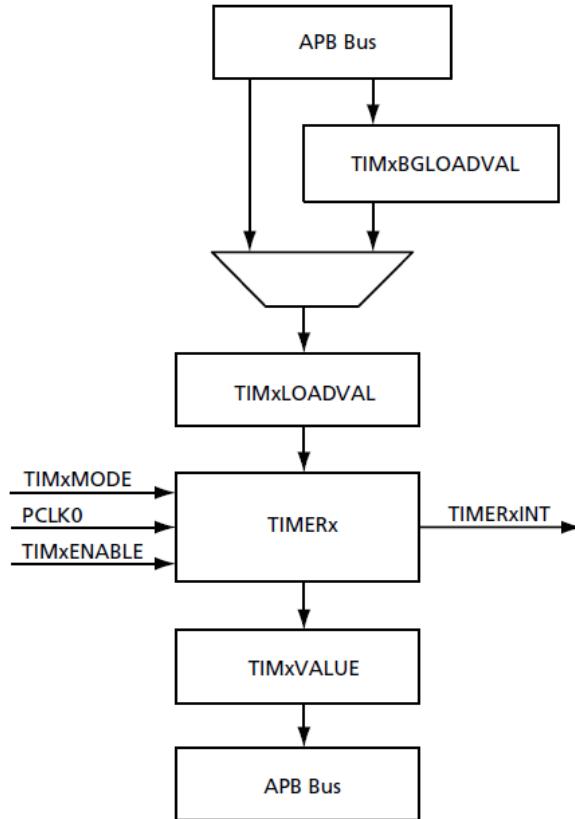


Watchdog Timer Block Diagram

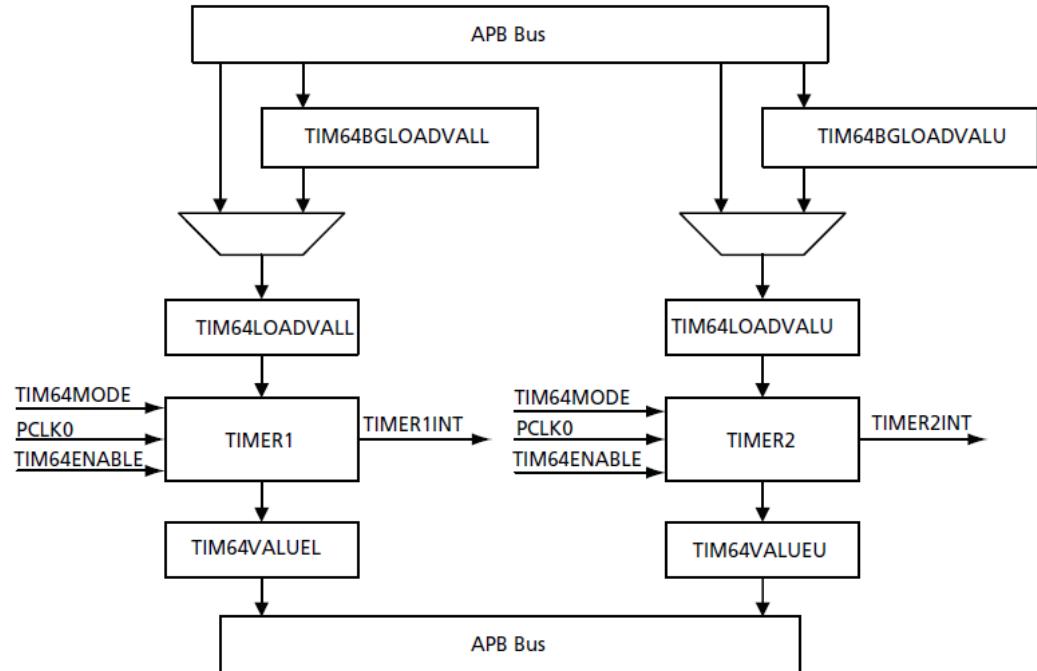


System Timer Block Diagram

32 and 64 bit Modes

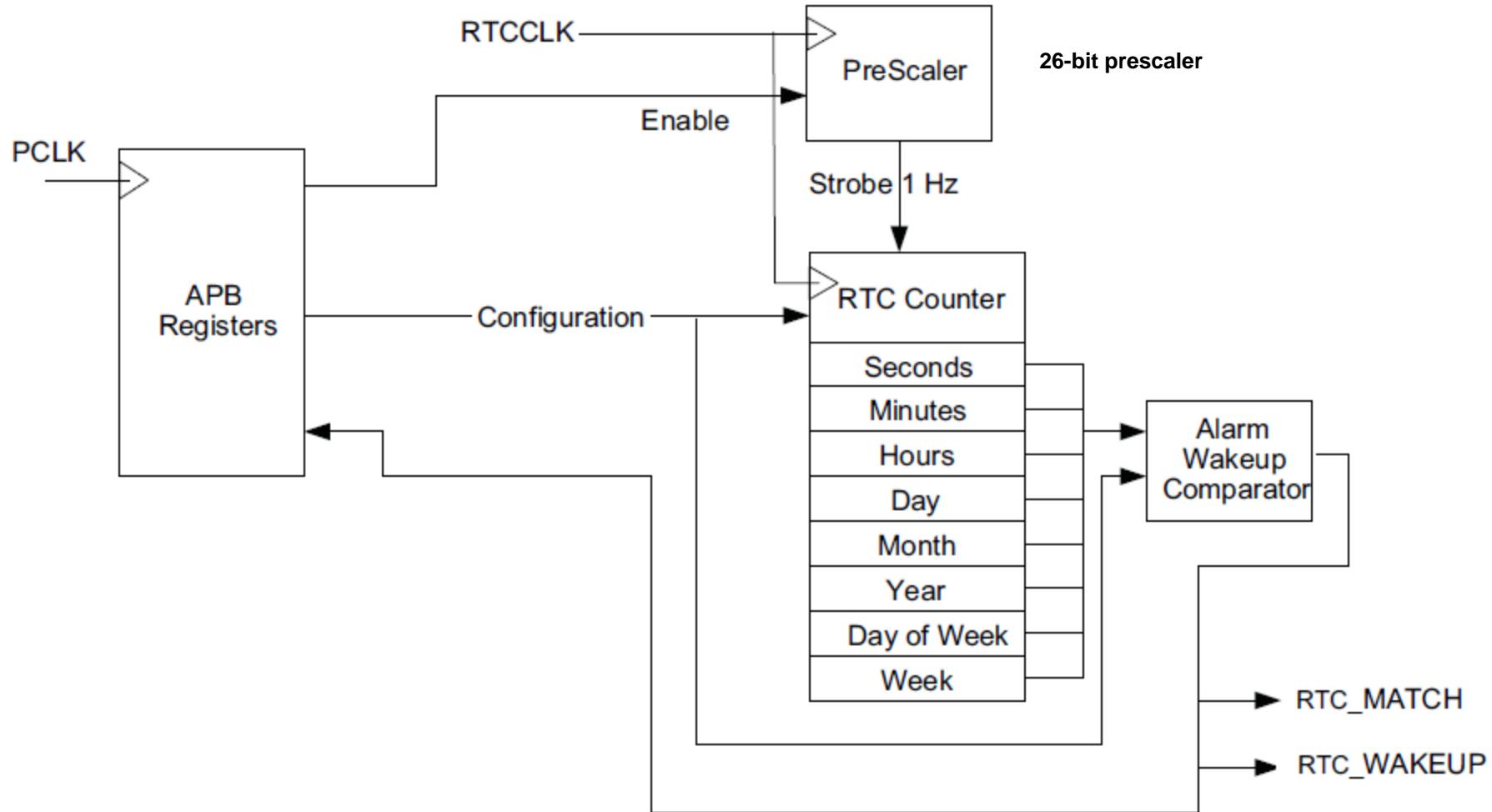


32-bit mode



64-bit mode

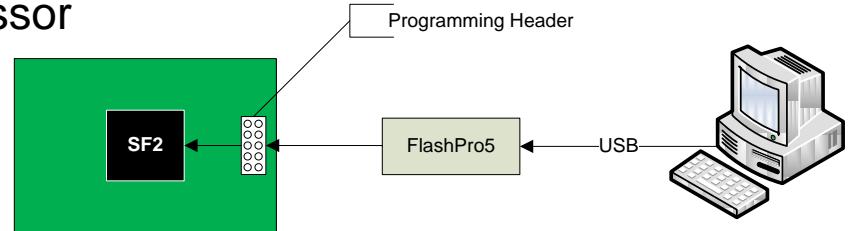
RTC Block Diagram



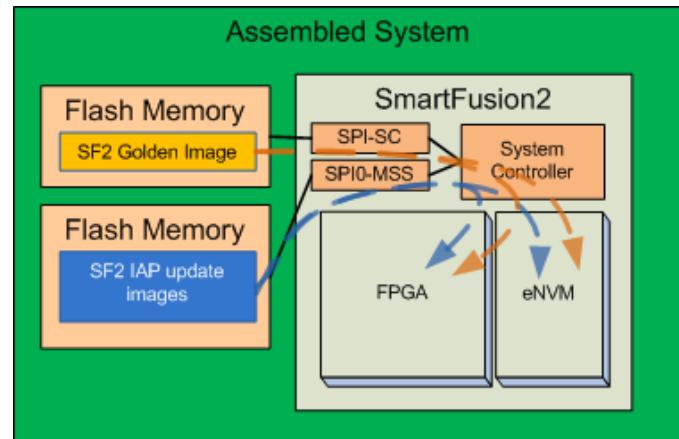
Programming and Debug

SF2 Programming Terminology

- In-System Programming (ISP)
 - **External intelligent** source acting as master
 - FlashPro, Libero or embedded processor
 - Programming interfaces
 - JTAG, **SPI-Slave**



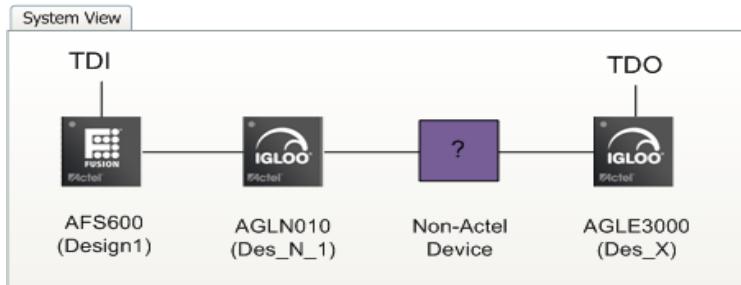
- In-Application-Programming (IAP)
 - **Internal intelligent** acting as master
 - User Application (in FPGA or Cortex®M3)
 - Programming data received from **USB**, Ethernet, etc.
 - **Programming Interface**
 - SPI0-MSS, as SPI-Master



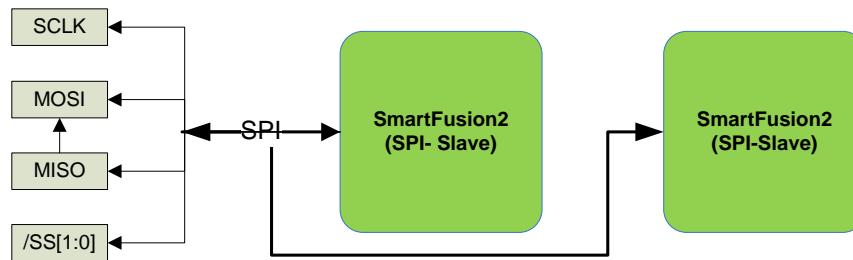
- Auto-programming with Golden Image
 - System Controller acting as master

In-System Programming (ISP)

- JTAG Chain (FlashPro & Embedded Solution)

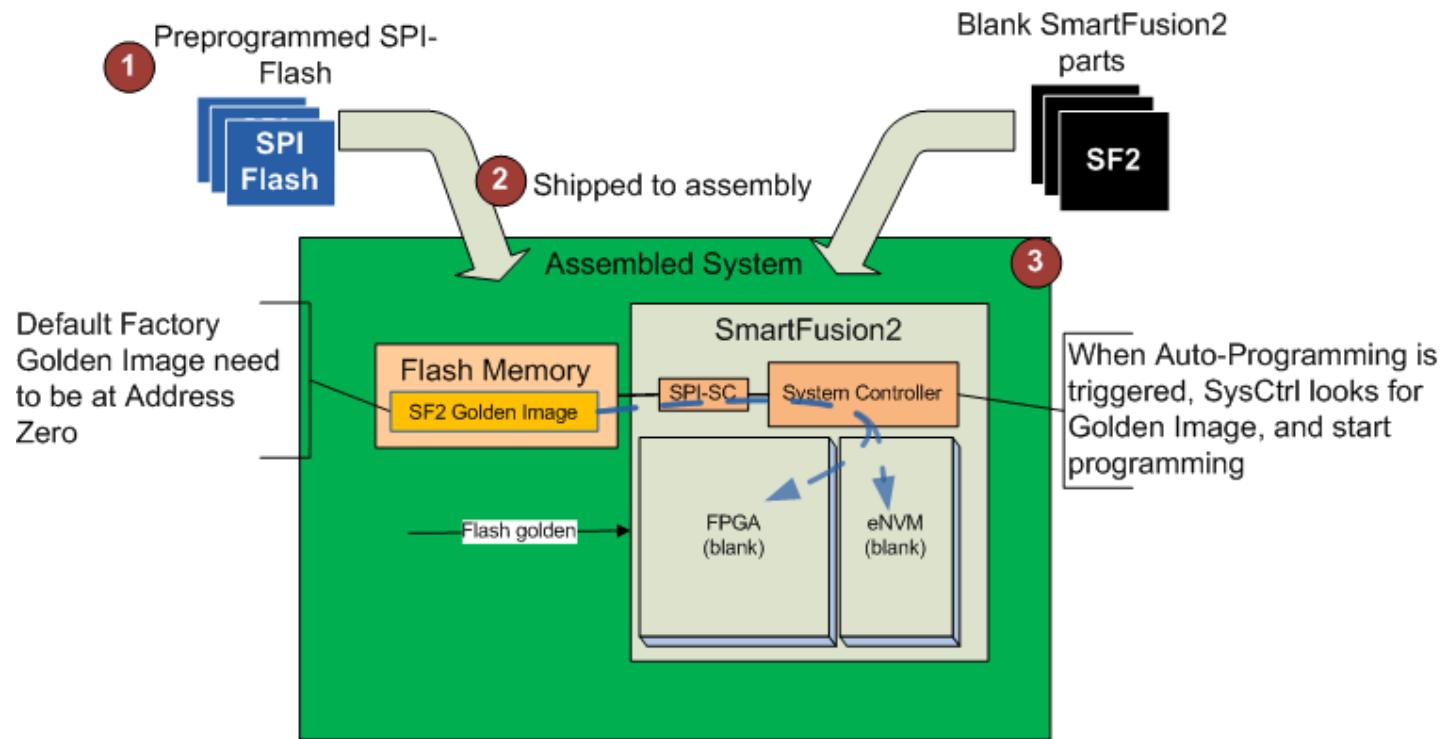


- SPI-Slave (FlashPro & Embedded Solution)



- New programmer (FlashPro5) will support programming one SF2 device only

Auto-programming

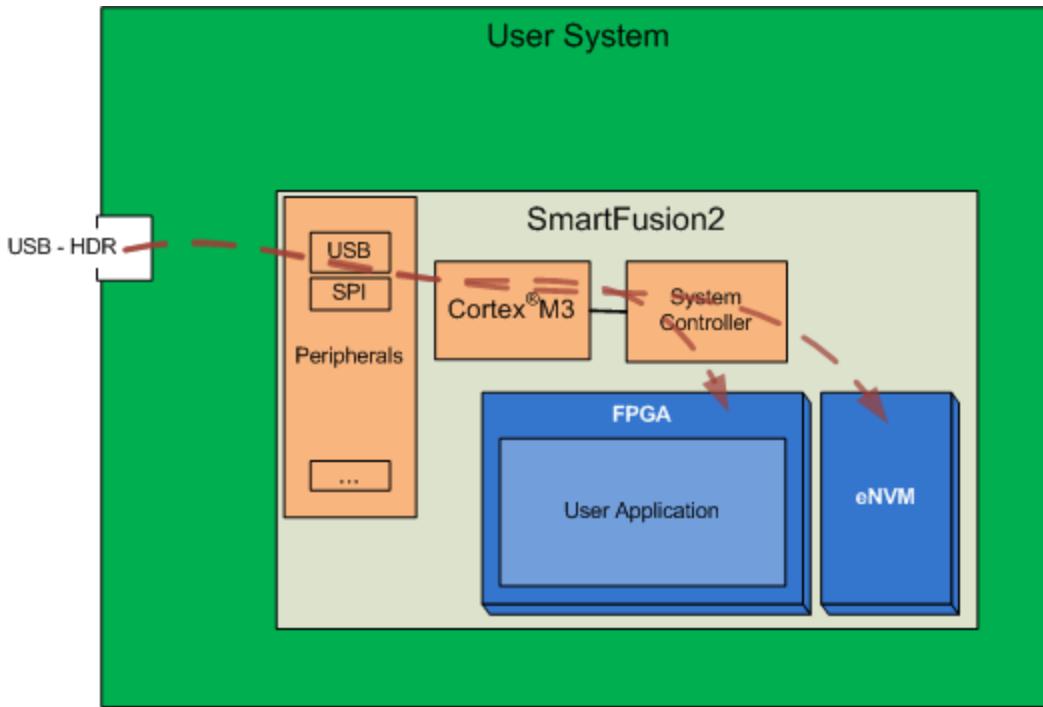


1. External SPI-Flash are pre-programmed with encrypted golden bitstream
2. SPI-Flash and blank SF2 parts are assembled on the system board
3. When flash_golden is asserted at power on reset, System Controller (SysCtrl) reads the golden bitstream in the external SPI-Flash, and programs the FPGA/eNVM

IAP: Cortex®M3 Update

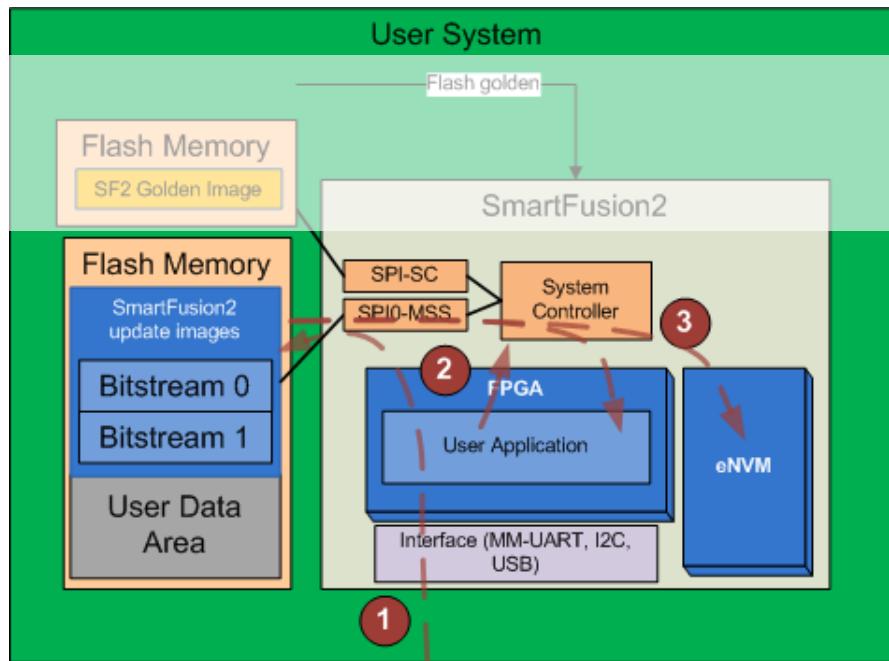


Ferrari
USB Flash Drive



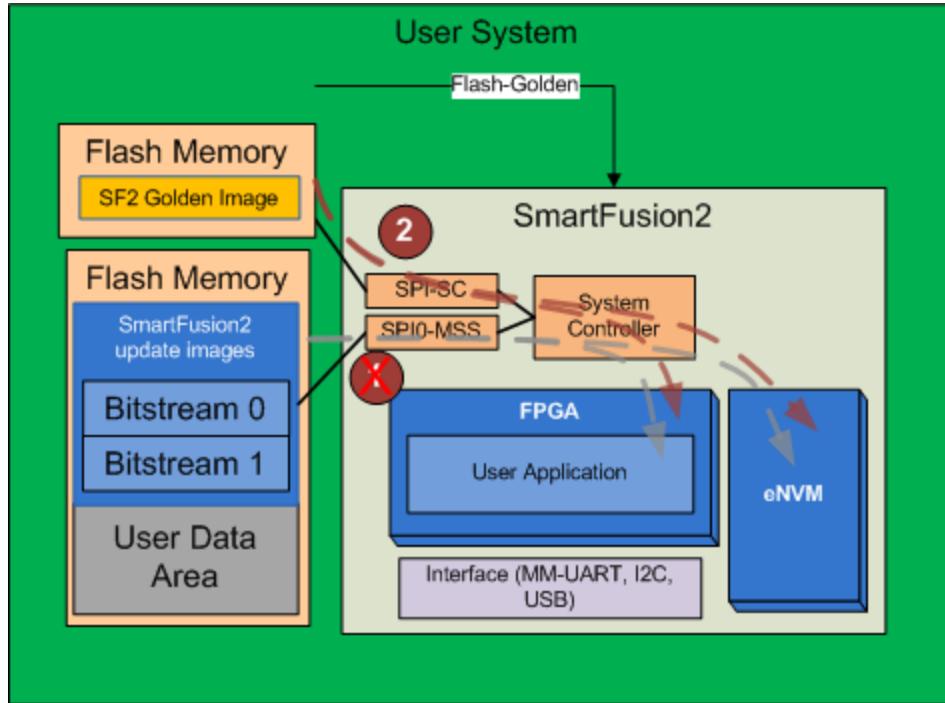
- User firmware receives programming bitstream through the available peripherals and programs the FPGA and/or eNVM
 - Recommendation : authenticate bitstream before programming

IAP: Auto-Update



1. User application receives and programs updated bitstream(s) into SPI-Flash
 - Recommendation: authenticate bitstream before committing to SPI-Flash
2. User application initiates programming services request, and pass the starting memory location of the updated bitstream to SysCtrl
 - Recommendation : authenticate bitstream from SPI-Flash before programming
3. SysCtrl reads and programs the bitstream into FPGA/eNVM

Auto-Update Failure Recovery



1. When programming fails, system controller performs 1 retry
2. The user can recover from the failure by asserting the flash golden pin

I/O During Programming

- ISP
 - JTAG – I/O controlled by Boundary Scan Register (BSR)
 - SPI-Slave
 - Tri-state (if device is blank)
 - Bus hold, using user F*F configuration (if device is pre-programmed with the configuration)
- IAP (Auto-Update or Cortex®M3-Update)
 - I/O depends on user option
 - Tri-state, or
 - Bus hold, using user F*F configuration (if device is pre-programmed with the configuration)
- Auto-Programming
 - Tri-state

Peripheral During Programming

- JTAG & SPI-Slave (ISP)
 - MSS are held in reset, and the peripheral reverts to its hardware default
 - Cortex®M3 are held in reset
- Auto-Update
 - The peripheral initialized data will persist - until the new configurations are loaded into the peripheral registers when the Fabric powers up
- Cortex®M3-Update
 - Peripherals are active during programming, as in the case of SmartFusion

SmartFusion2 Hardware Comparison

	SF2 Starter Kit	SF2 Evaluation Kit	SF2 Development Kit
Price (US)	\$299	\$399	\$1800
Device	M2S050T-FGG896	M2S025T-1FGG484	M2S050T-FGG896
Memory	64MB LPDDR, 16MB SPI Flash	512 MB LPDDR, 64 MB SPI Flash	512MB DDR3, 16MB SDRAM, 8MB SPI Flash
USB 2.0 On-The-Go Controller	Yes	Yes	Yes
Power over Ethernet	No	No	Yes; Up to 48W of Power
SERDES/PCIe	No	PCI Express Gen2 x1 interface Four SMA connector for testing of full-duplex SERDES channel	X4 PCIe Gen1/Gen2 4 Tx/Rx High Speed SMP Connectors
I2C, SPI, GPIO Headers	Via breadboard area	Yes	Yes
Precision ADC	No	No	16-bit, 1MSPS, 8-channel
eMMC	No	No	4GB NAND Flash Memory
Ethernet	10/100 in MII mode	RJ45 interface to 10/100/1000 Ethernet	10/100/1000 in SGMII mode
Expansion interfaces	Via breadboard area	GPIO header	FMC header for daughter card support
IEEE 1588 Timing and Synchronization	No	No	Yes; ZL30362
User Controlled LEDs	2	8	8
USB WiFi Module	Yes included	No	Support but not included
CAN 2.0	No	No	Yes; (2) DB9 connectors