Customer Training Workshop
Traveo™ II Body High and Cluster 2D SRAM Interface

Q4 2020

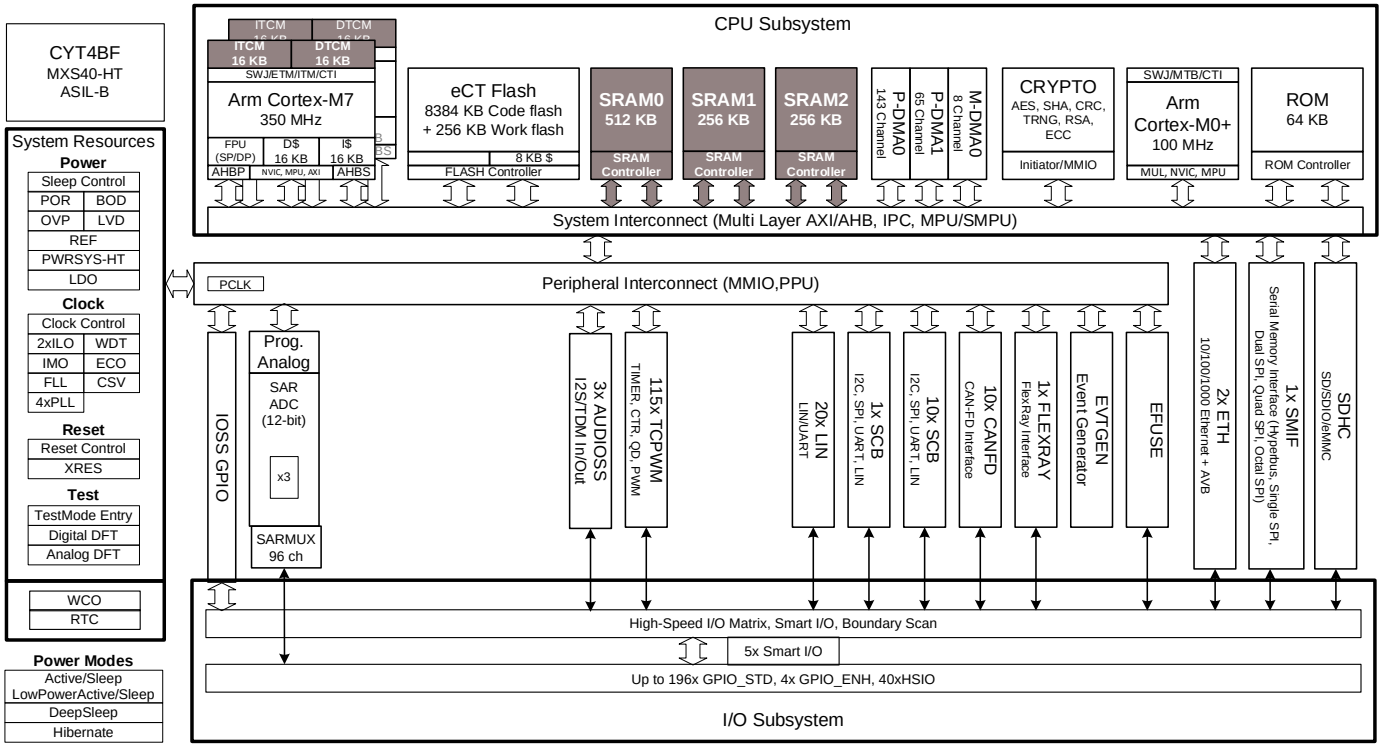
Target Products

› Target product list for this training material

Family Category	Series	Code Flash Memory Size
Traveo™ II Automotive Body Controller High	CYT3BB/CYT4BB	Up to 4160KB
Traveo II Automotive Body Controller High	CYT4BF	Up to 8384KB
Traveo II Automotive Cluster	CYT3DL	Up to 4160KB
Traveo II Automotive Cluster	CYT4DN	Up to 6336KB

Introduction to Traveo II Body Controller High

The SRAM interface is in the CPU subsystem



Hint Bar

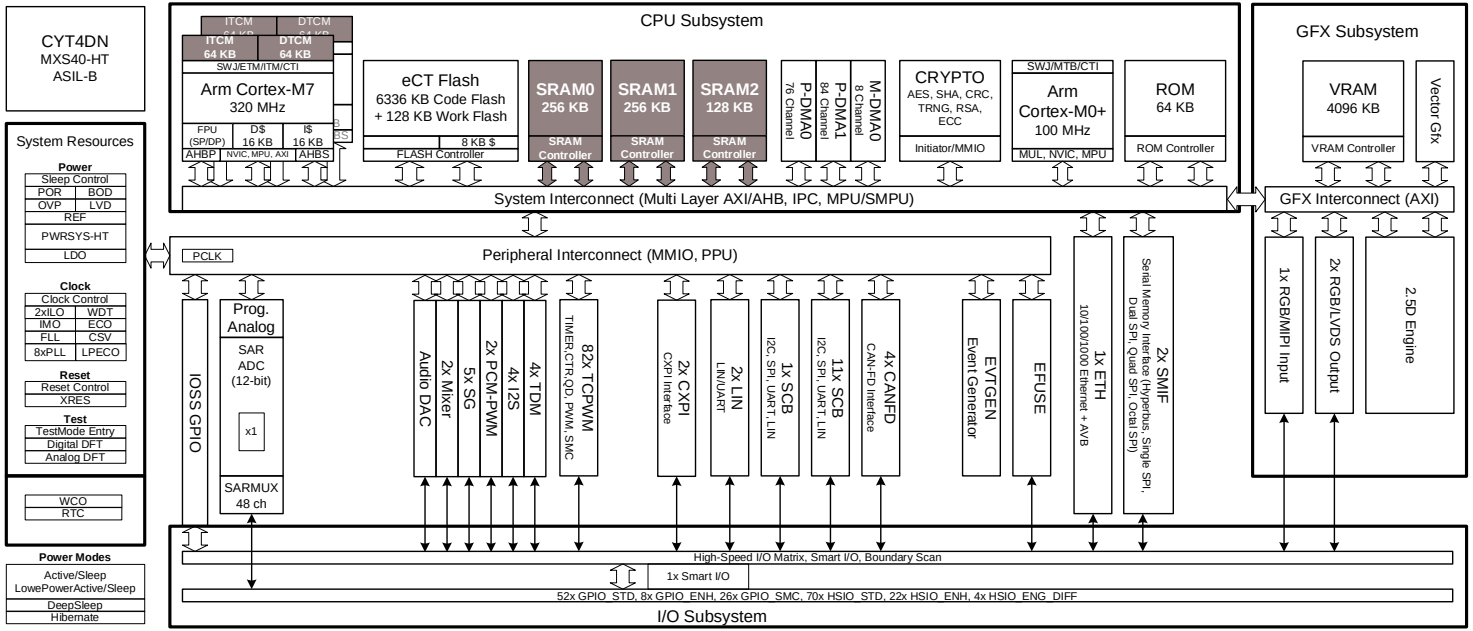
Review TRM chapter 10 for additional details

Training section reference: SRAM Interface for Traveo II Body Entry

CYT4BF series has two options of RAM size: 768KB and 1024KB

Introduction for Traveo II Cluster

The SRAM interface is in the CPU subsystem



Hint Bar

Review TRM chapter 10 for additional details

Training section reference: SRAM Interface for Traveo II Body Entry

SRAM Interface Overview

- SRAM controller for SRAM memory interface
- Features
  - One AXI bus interface in the fast clock domain
  - One AHB-Lite bus interface in the slow clock domain
  - Programmable wait states from 0 to 3
    - Wait states for slow clock domain
      - In 0 wait cycle, up to 100 MHz of CLK_MEM[1]
      - In 1 wait cycle, from 100 MHz to 200 MHz of CLK_MEM
    - Wait states for fast clock domain
      - In 0 wait cycle, up to 200 MHz of CLK_MEM
  - 64-bit wide interface to SRAM memory
  - Error-correcting code (ECC)
  - Optional retention of SRAM contents in DeepSleep mode

Hint Bar

Review TRM chapter 10 for additional details

Review the Clock System Training section for additional details about high-frequency clocks

[1] Divided version of high-frequency root clock (CLK_HF0)

TCM Interface Overview

- Used by the Arm® Cortex®-M7 core to enable low-latency access to the external memories
- Features
  - Programmable 0 wait states
    - Wait states for fast clock domain
      - In 0 wait cycle, up to 350 MHz of CLK_FAST[1]
  - TCM interfaces at CM7 core
    - ITCM[2] (64-bit data)
    - DTCM[3] (32-bit data)
  - Error-correcting code (ECC)

Hint Bar

Tight Coupled Memory (TCM)

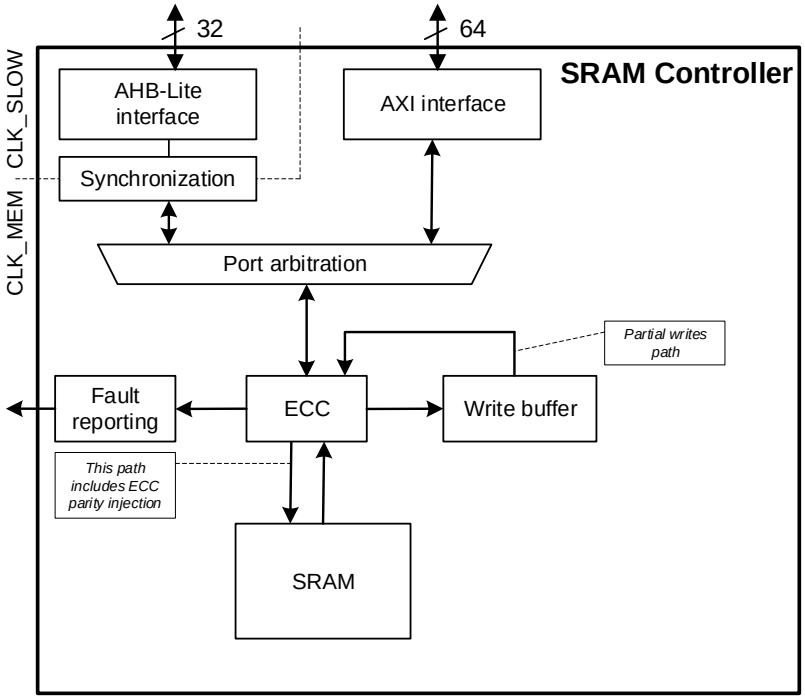
CM7: Cortex-M7

Review the Clock System Training section for additional details about high-frequency clocks

[1] Divided version of high-frequency root clock (CLK_HF1)
[2] Instruction TCM
[3] Data TCM

SRAM Controller Block Diagram

- The SRAM controller consists of:
  - AHB-Lite interface
  - Synchronization
  - AXI interface
  - Port arbitration
  - Error-correcting code (ECC)
  - SRAM memory
  - Write buffer
  - Fault reporting

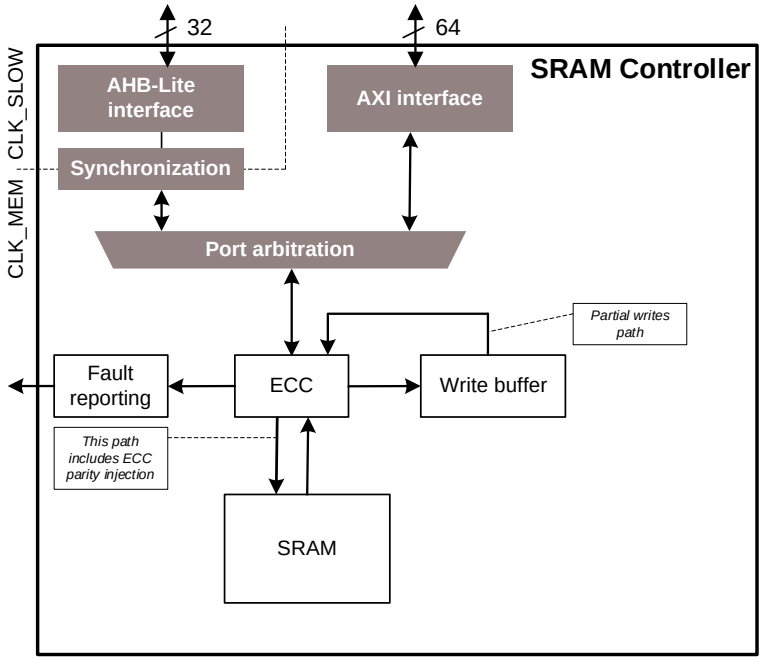


Hint Bar

Review TRM section 10.2 for additional details

AHB-Lite Interface and AXI Interface

- The AHB-Lite interface and AXI interface includes:
  - SRAM access
  - Arbitration priority



Hint Bar

Review TRM section 10.1 for additional details

SRAM Access

- SRAM has a 64-bit wide bus. The write access differs depending on the data size and ECC state
- Data read from SRAM
  - AHB-Lite and AXI read transfer
    - Translated into an SRAM read access and can be done by a single read access to SRAM
- Data write to SRAM without ECC (Disabled)
  - AHB-Lite and AXI write transfer
    - Translated into an SRAM write access and can be done by a single write access to SRAM
- Data write to SRAM with ECC (Enabled)
  - 64-bit AXI write transfers
    - Translated into an SRAM write access and can be done by a single write access to SRAM
  - 8-bit, 16-bit, 32-bit AXI, and AHB-Lite write transfer
    - Translated into an SRAM read access and an SRAM write access
    - Requires two accesses to SRAM

Hint Bar

Review TRM section 10.2.3 for additional details

Review the Error Correcting Code (ECC) section for additional details about ECC

Review the Write Buffer section for additional details about the write buffer

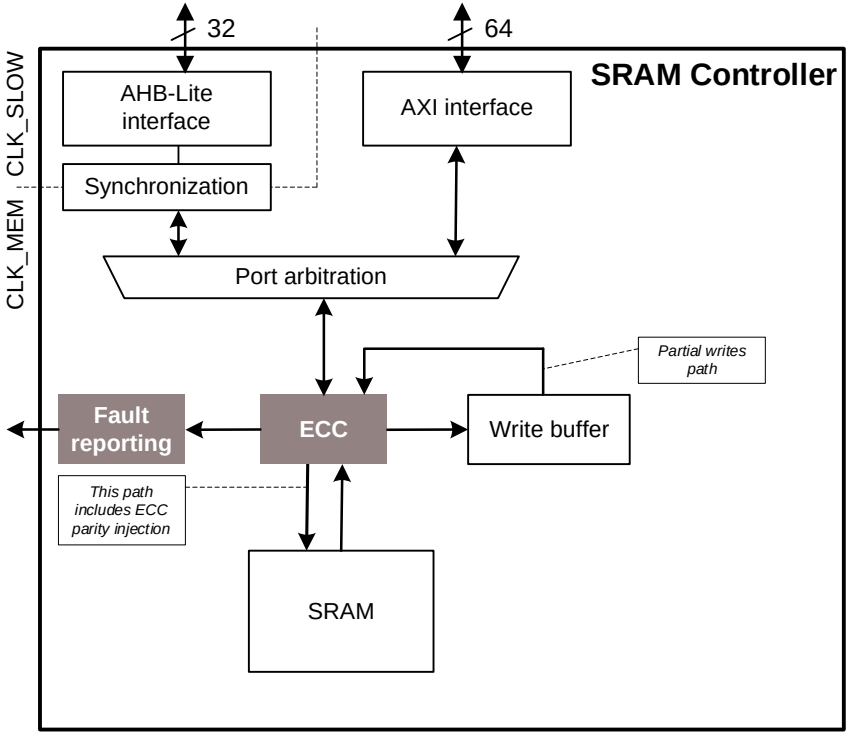
Arbitration Priority

- The arbiter component performs priority-based arbitration on the AHB-Lite interface and AXI interface ports
- Arbitration priority is set according to the PROT_SMPU_MS0_CTL.PRIO register of SMPU (0 is highest, 3 is lowest)
- If Masters have the same priority, round-robin arbitration is performed according to the bus master identifier (starting from 15 to a lower identifier number)

Master Identifier	Bus Master (CYT4BF)	Bus Master (CYT4DN)
0	Arm® Cortex®-M0+	Arm Cortex-M0+
1	Crypto	Crypto
2	P-DMA0	P-DMA0
3	P-DMA1	P-DMA1
4	M-DMA	M-DMA
5	SDHC	-
9	Ethernet 0	Ethernet 0
10	Ethernet 1	-
12	-	Video subsystem
13	Cortex-M7_1	Cortex-M7_1
14	Cortex-M7_0	Cortex-M7_0
15	Test controller	Test controller

Error-Correcting Code (ECC)

- ECC includes:
  - SEC/DED[1]
  - Fault reporting
  - Error injection



Hint Bar

Review TRM section 10.3 for additional details

[1] Single-error correction/double-error detection

ECC

- ECC supports SEC/DED, which:
  - Detects and corrects single-bit error and detects double-bit errors
  - Reports error to the fault reporting structure
  - Includes 8-bit ECC per 64 bits of data
- Fault reporting
  - Correctable and non-correctable ECC errors are reported to the fault structure in the same way
  - Use case
    - SEC report: Logging single-bit error count via the trigger that connects to TCPWM
    - DED report: Use for fail-safe operation, such as stop system control
- Error injection
  - Providing an error injection address and injection parity[1] can generate an ECC error
  - Use case
    - Use as initial diagnosis of ECC before running the application

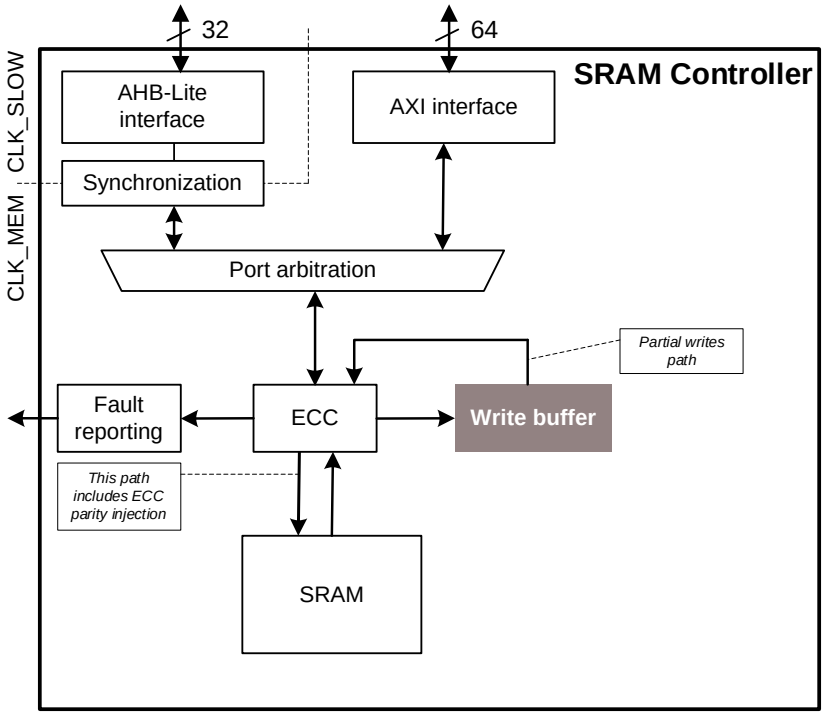
Hint Bar

Review TRM section 10.3 for additional details

[1] Any access size can be used to inject parity

Write Buffer

- Write buffer request



Hint Bar

Review TRM section 10.2.4 for additional details

Write Buffer

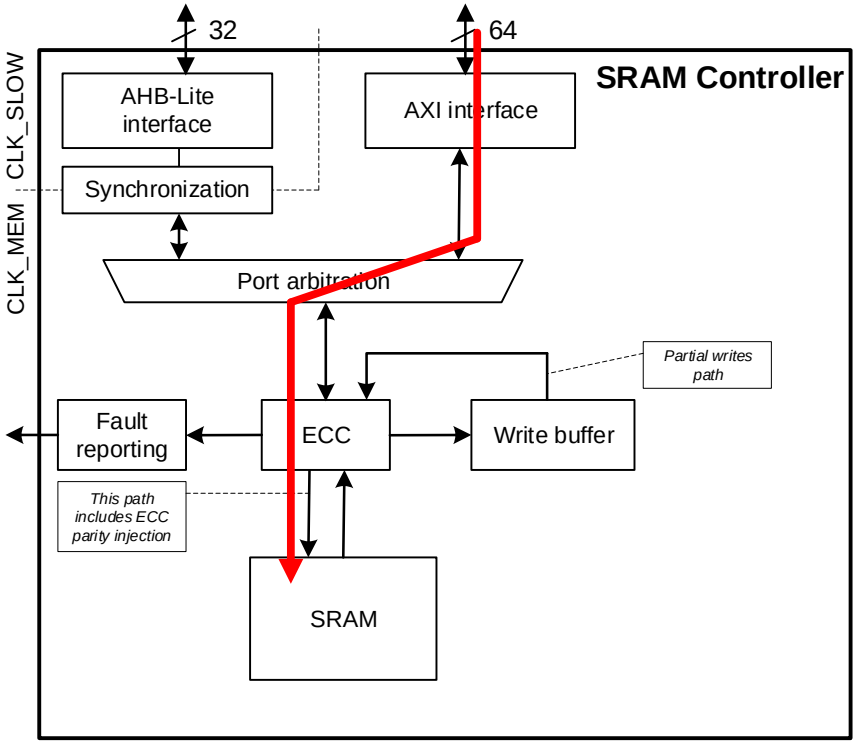
- The SRAM controller write buffer is used only when ECC is enabled
- Sequence of data merging involving write buffer
  - Requested read data is merged with partial write data to provide a complete 64-bit data word
  - Address and merged write data are written to the write buffer
  - A future write buffer request results in an SRAM write access with the merged write data
- For 8-bit, 16-bit, and 32-bit data size write transfers, an additional SRAM read access precedes the SRAM write access to retrieve the “missing” data bytes
- The write buffer is not retained in DeepSleep mode and should be emptied before entering the mode
- The state of the write buffer is shown by WB_EMPTY

Hint Bar

Review TRM section 10.2.4 for additional details

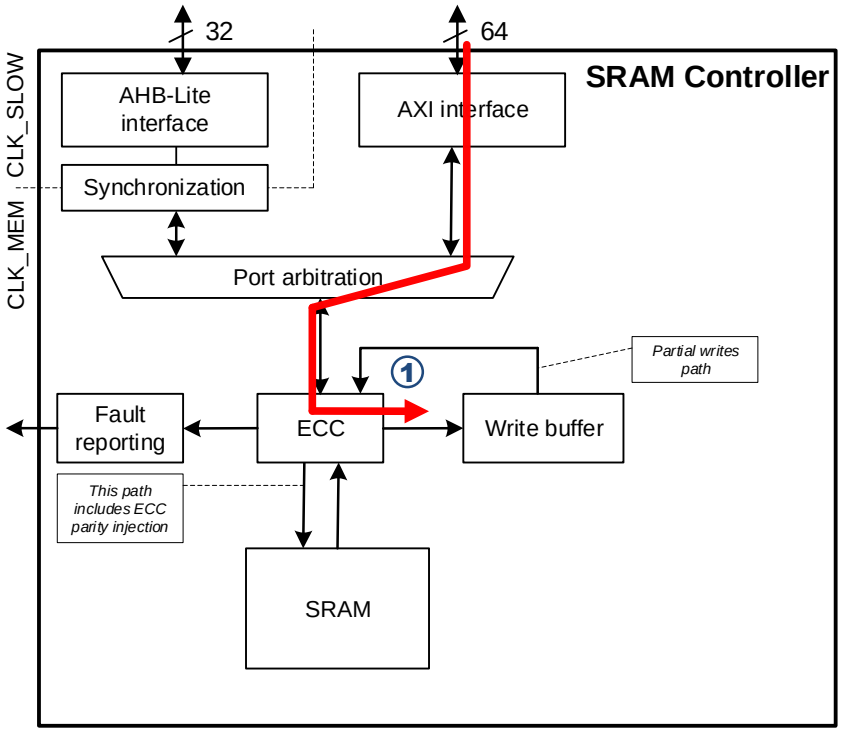
Write 64 Bits

› Only a single SRAM write access is required



Write 8/16/32 Bits, Partial Write with ECC (1/3)

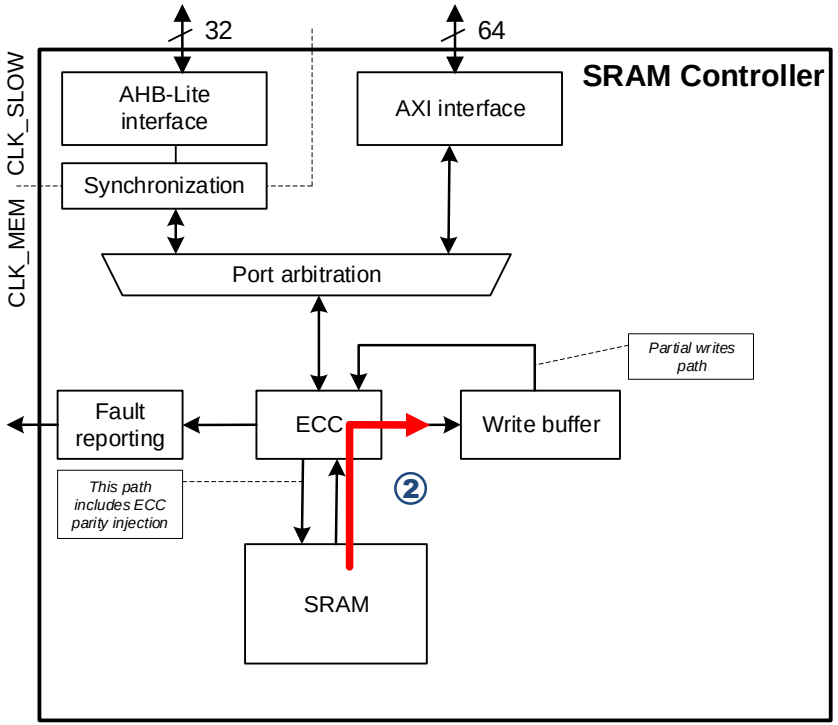
① Data from AXI interface (or AHB-Lite interface) is stored to the write buffer



Write 8/16/32 Bits, Partial Write with ECC (2/3)

① Data from the AXI interface (or AHB-Lite interface) is stored to the write buffer

② Missing data are read from SRAM and stored to the write buffer to complete the 64-bit word

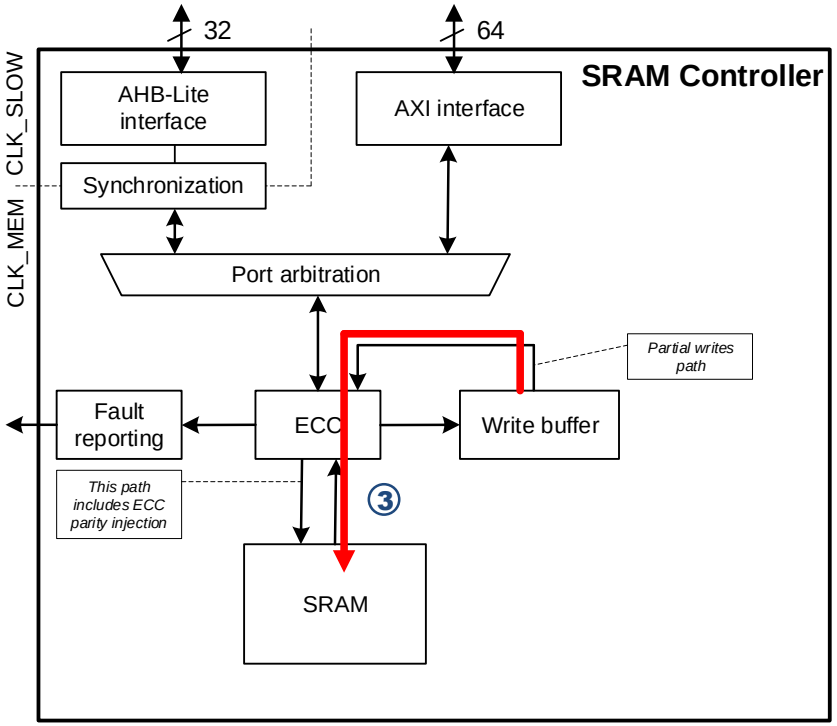


Write 8/16/32 Bits, Partial Write with ECC (3/3)

① Data from the AXI interface (or AHB-Lite interface) is stored to the write buffer

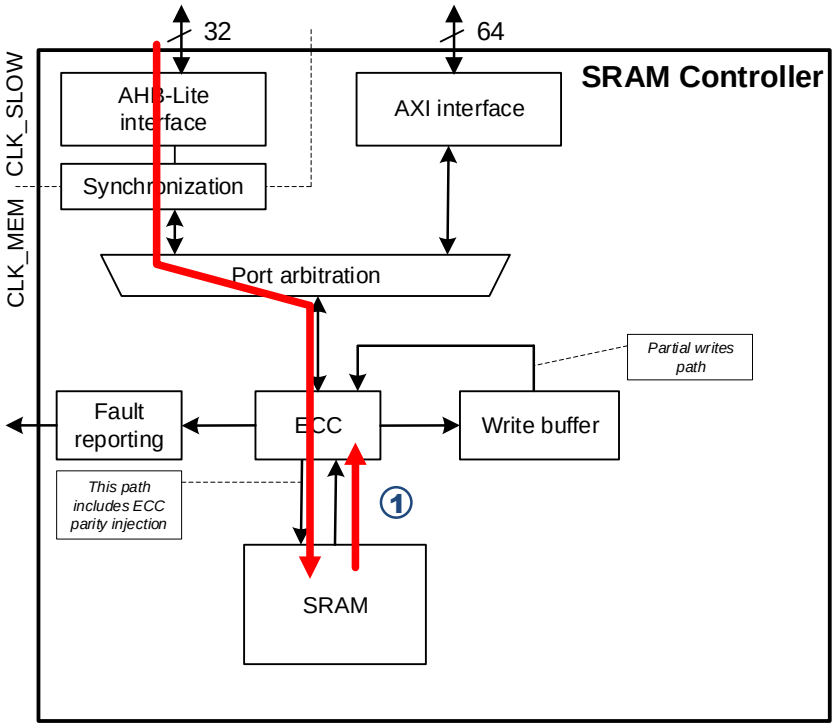
② Missing data are read from SRAM and stored to the write buffer to complete the 64-bit word

③ Complete 64-bit word with ECC is written to the SRAM



Read With ECC with 1-bit Correctable Error (1/3)

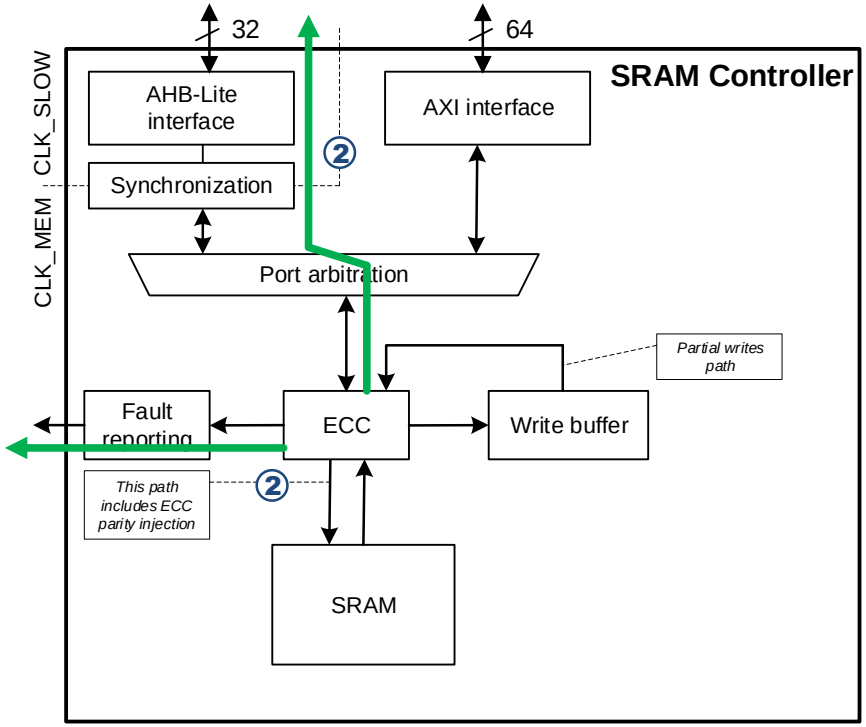
① 1-bit error is detected in AHB-Lite (or AXI interface) bus read data



Read With ECC with 1-bit Correctable Error (2/3)

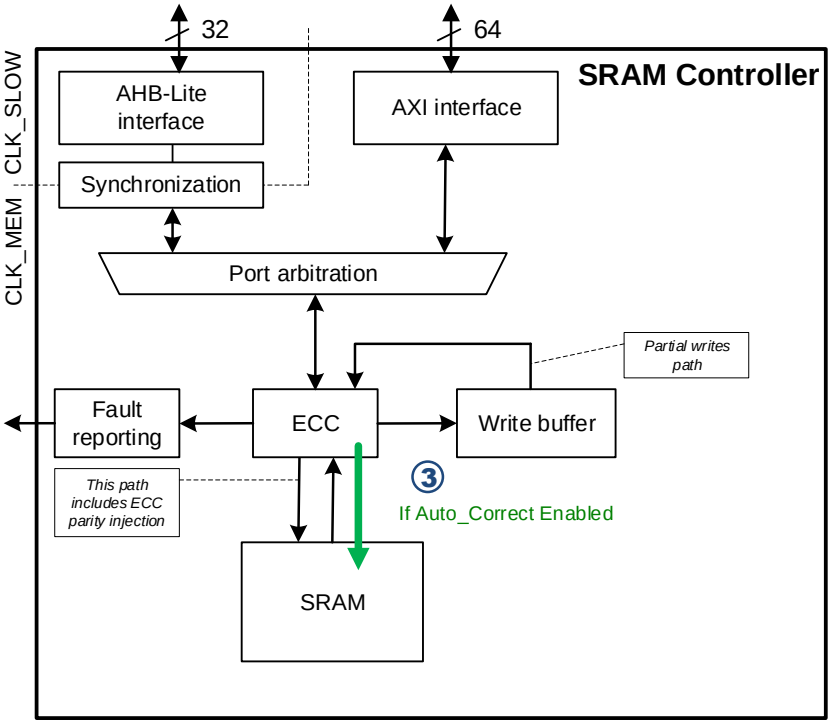
① 1-bit error is detected in AHB-Lite (or AXI interface) bus read data

② The error is notified to fault reporting and the corrected data is sent to the AHB-Lite master requester



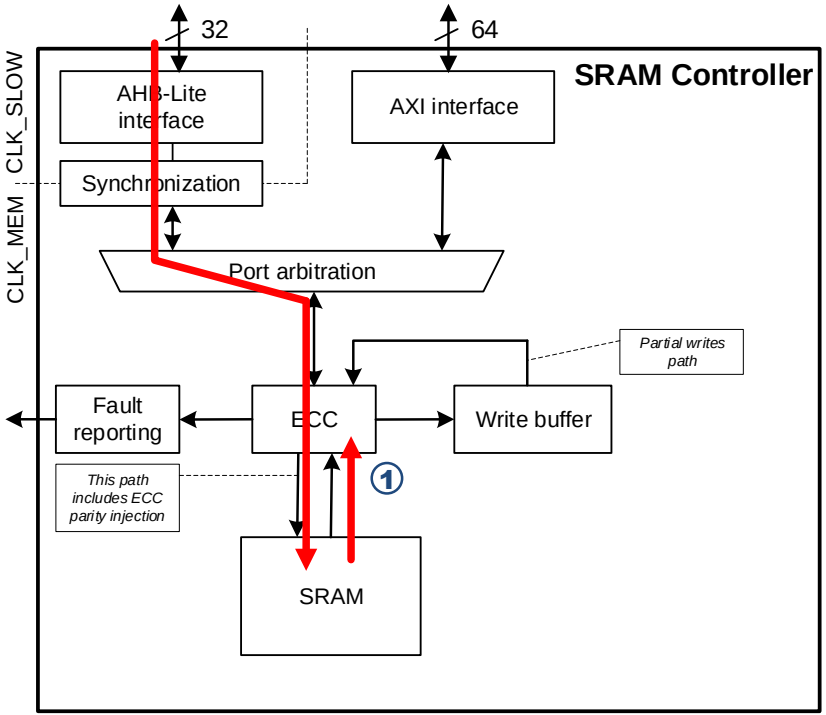
Read With ECC with 1-bit Correctable Error (3/3)

1. 1-bit error is detected in AHB-Lite (or AXI interface) bus read data
2. The error is notified to fault reporting and the corrected data is sent to the AHB-Lite master requester
3. If ECC_AUTO_CORRECT is enabled, the corrected data is written to the SRAM



Read With ECC with Non-Correctable Error (1/2)

① Non-correctable error is detected in AHB-Lite (or AXI interface) bus read data

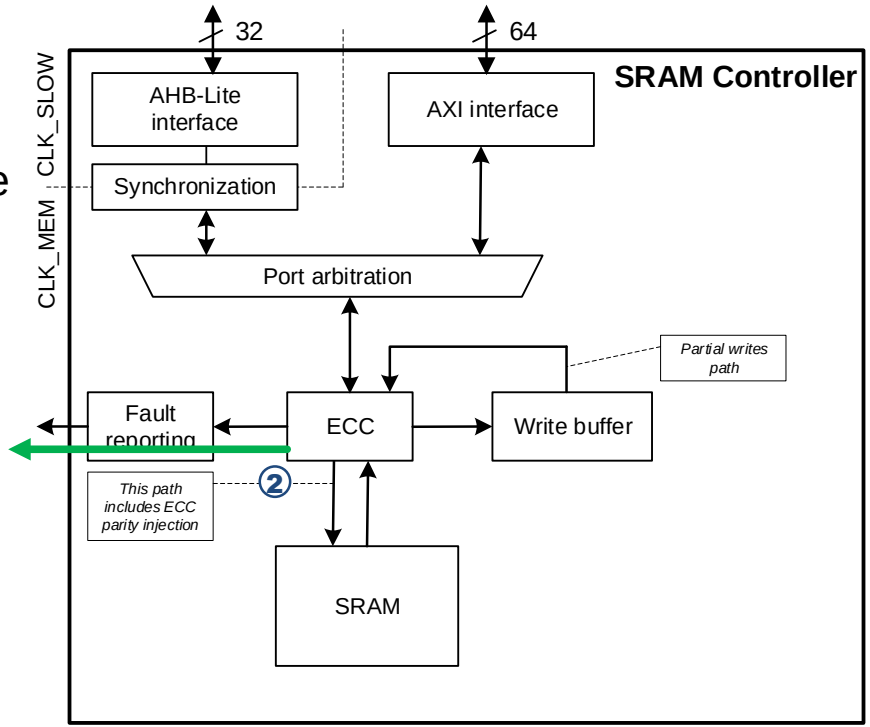


Read With ECC with Non-Correctable Error (2/2)

① Non-correctable error is detected in AHB-Lite (or AXI interface) bus read data

② The error is notified to fault reporting

- Fault reporting captures the non-correctable error
  - Single-bit error detected in the word address
  - Double-bit error detected in the data access
- No bus error is generated while a fault is generated



SRAM retention in DeepSleep

- In DeepSleep mode, SRAM0 can be fully retained or retained in increments of 32-KB sectors
- Advantage
  - By setting the size of backup RAM according to the system, it is possible to optimize the current consumption during DeepSleep mode

Hint Bar

SRAM units, other than SRAM0, can be fully retained

Review TRM section 10.1 for additional details on SRAM region that is not guaranteed to be retained across resets

TCM Interface (1/2)

- Used by the CM7 core to enable low-latency access to the external memories
- Functionality
  - No read and write wait states
  - Can fetch instructions from any TCM interface
  - ECC with SECDED and fault reporting
    - ITCM with 64-bit data needs 8 ECC bits, which require 72-bit wide SRAMs
    - DTCMs with 32-bit data need 7 ECC bits, which require 39-bit wide SRAMs
    - Supports read-modify-write for smaller[1] byte write
- Advantage
  - Can achieve high-performance operation by placing code in TCM RAM

Hint Bar

Review the Error Correcting Code (ECC) section for additional details about ECC

[1] 8, 16, 32-bit in ITCM
8, 16-bit in D0/1TCM

TCM Interface (2/2)

- A CM7 core can access its own TCMs and other CM7's TCMs via the following specific address

Address	Target TCM	Description
0x0000:0000	CM7_0_ITCM/CM7_1_ITCM	CM7_0 can only access to CM7_0_I/DTCM through this space CM7_1 can only access to CM7_1_I/DTCM
0x2000:0000	CM7_0_DTCM/CM7_1_DTCM	
0xA000:0000	CM7_0_ITCM	CM7_0 can access to CM7_1_I/DTCM through this space CM7_1 can access to CM7_0_I/DTCM The CM7 cannot access their own TCM using this space; it will result in an address decode failure and will be returned as a bus error
0xA001:0000	CM7_0_DTCM	
0xA010:0000	CM7_1_ITCM	
0xA011:0000	CM7_1_DTCM	

Hint Bar

No other device should be mapped between the ITCM and DTCM of a particular CM7

Appendix

Comparison between CYT2BL, CYT4BF, and CYT4DN

Features		CYT2BL	CYT4BF	CYT4DN
SRAM	Memory size option	512KB	1024KB	640KB
	Interface width to SRAM memory	32-bit	64-bit	
	Bus interface	AHB-Lite	AXI, AHB-Lite	
	Wait states	Slow clock domain: 0 wait cycle for CLK_HF <= 100 MHz 1 wait cycle for 100 MHz < CLK_HF <= 160 MHz Fast clock domain: 0 wait cycle for CLK_HF <= 160 MHz	Slow clock domain: 0 wait cycle for CLK_MEM <= 100 MHz 1 wait cycle for 100 MHz < CLK_MEM <= 200 MHz Fast clock domain: 0 wait cycle for CLK_MEM <= 200 MHz	
	Bus master priority of arbiter	0: Cortex-M0+ CPU 1: Cryptography Component 2: P-DMA0 3: P-DMA1 4: M-DMA 14: Cortex-M4 CPU 15: Test Controller	0: Cortex-M0+ CPU 1: Cryptography Component 2: P-DMA0 3: P-DMA1 4: M-DMA 5: SDHC 9: Ethernet 0 10: Ethernet 1 13: Cortex-M7_1 CPU 14: Cortex-M7_0 CPU 15: Test Controller	0: Cortex-M0+ CPU 1: Cryptography Component 2: P-DMA0 3: P-DMA1 4: M-DMA 9: Ethernet 0 12: Video Subsystem 13: Cortex-M7_1 CPU 14: Cortex-M7_0 CPU 15: Test Controller
	ECC (SED/DED)	Same		
	RAM retention in DeepSleep	Same		

Comparison between CYT2BL, CYT4BF, and CYT4DN

Features		CYT2BL	CYT4BF	CYT4DN
TCM	Memory size	N/A	16KB ITCM, 16KB DTCM	64KB ITCM, 64KB DTCM
	Interface width		ITCM: 64-bit D0TCM: 32-bit D1TCM: 32-bit	
	Bus interface		TCM interface	
	Wait states		Fast clock domain: 0 wait cycle for CLK_FAST <= 350 MHz	
	Priority		Cortex-M7 CPU only	
	ECC (SED/DED)		Supported	

Part of your life. Part of tomorrow.

Revision History

Revision	ECN	Submission Date	Description of Change
**	6402537	12/05/2018	Initial release
*A	6649384	08/07/2019	Updated page 2, 3, 4, and 28.
*B	7073093	01/22/2021	Updated page 1, 2, 3, 10, 24, and 28.