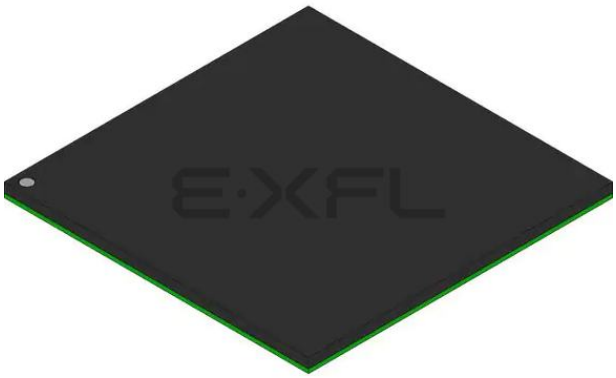


Welcome to E-XFL.COM

Understanding Embedded - Microprocessors

Embedded microprocessors are specialized computing chips designed to perform specific tasks within an embedded system. Unlike general-purpose microprocessors found in personal computers, embedded microprocessors are tailored for dedicated functions within larger systems, offering optimized performance, efficiency, and reliability. These microprocessors are integral to the operation of countless electronic devices, providing the computational power necessary for controlling processes, handling data, and managing communications.

Applications of Embedded - Microprocessors

Embedded microprocessors are utilized across a broad spectrum of applications, making them indispensable in

Details	
Product Status	Active
Core Processor	ARM® Cortex®-A5
Number of Cores/Bus Width	1 Core, 32-Bit
Speed	600MHz
Co-Processors/DSP	Multimedia; NEON™ SIMD
RAM Controllers	LPDDR, LPDDR2, DDR2
Graphics Acceleration	Yes
Display & Interface Controllers	LCD, Touchscreen
Ethernet	10/100Mbps (1)
SATA	-
USB	USB 2.0 (3)
Voltage - I/O	1.2V, 1.8V, 3.3V
Operating Temperature	-40°C ~ 85°C (TA)
Security Features	AES, SHA, TDES, TRNG
Package / Case	361-TFBGA
Supplier Device Package	361-TFBGA (16x16)
Purchase URL	https://www.e-xfl.com/product-detail/atmel/atsama5d44a-cu

Table 3-1: TFBGA361 Pin Description (Continued)

Pin	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral A		PIO Peripheral B		PIO Peripheral C		Reset State(1)
			Signal	Dir	Signal	Dir	Signal	Dir	Signal	Dir	Signal	Dir	Signal, Dir, PU, PD, HiZ, ST
U11	VDDANA	GPIO	PD22	I/O	–	–	–	–	–	–	–	–	PIO, I, PU, ST
R10	VDDANA	GPIO	PD23	I/O	–	–	–	–	–	–	–	–	PIO, I, PU, ST
U10	VDDANA	GPIO	PD24	I/O	–	–	–	–	–	–	–	–	PIO, I, PU, ST
R11	VDDANA	GPIO	PD25	I/O	–	–	–	–	–	–	–	–	PIO, I, PU, ST
U13	VDDANA	GPIO	PD26	I/O	–	–	–	–	–	–	–	–	PIO, I, PU, ST
T14	VDDANA	GPIO	PD27	I/O	–	–	–	–	–	–	–	–	PIO, I, PU, ST
R1	VDDIOP	GPIO_CLK	PD28	I/O	–	–	SCK0	I/O	–	–	–	–	PIO, I, PU, ST
P1	VDDIOP	GPIO_CLK	PD29	I/O	–	–	SCK1	I/O	–	–	–	–	PIO, I, PU, ST
N5	VDDIOP	GPIO	PD30	I/O	–	–	–	–	–	–	–	–	PIO, I, PU, ST
P5	VDDIOP	GPIO_CLK	PD31	I/O	–	–	SPI0_NPCS2	O	PCK1	O	–	–	PIO, I, PU, ST
W19	VDDIOM	MCI_CLK	PE0	I/O	–	–	A0/NBS0	O	MCI0_CDB	I/O	CTS4	I	O, High
U17	VDDIOM	EBI	PE1	I/O	–	–	A1	O	MCI0_DB0	I/O	–	–	O, High
T17	VDDIOM	EBI	PE2	I/O	–	–	A2	O	MCI0_DB1	I/O	–	–	A2, LOW
P16	VDDIOM	EBI	PE3	I/O	–	–	A3	O	MCI0_DB2	I/O	–	–	A3, LOW
U18	VDDIOM	EBI	PE4	I/O	–	–	A4	O	MCI0_DB3	I/O	–	–	A4, LOW
R17	VDDIOM	EBI	PE5	I/O	–	–	A5	O	CTS3	I	–	–	A5, LOW
V19	VDDIOM	EBI	PE6	I/O	–	–	A6	O	TIOA3	I/O	–	–	PIO, O, LOW
U19	VDDIOM	EBI	PE7	I/O	–	–	A7	O	TIOB3	I/O	PWMFI1	I	A7, LOW
T19	VDDIOM	EBI	PE8	I/O	–	–	A8	O	TCLK3	I	PWML3	O	A8, LOW
T18	VDDIOM	EBI	PE9	I/O	–	–	A9	O	TIOA2	I/O	–	–	A9, LOW
N14	VDDIOM	EBI	PE10	I/O	–	–	A10	O	TIOB2	I/O	–	–	A10, LOW
R18	VDDIOM	EBI	PE11	I/O	–	–	A11	O	TCLK2	I	–	–	A11, LOW
P17	VDDIOM	EBI	PE12	I/O	–	–	A12	O	TIOA1	I/O	PWMH2	O	A12, LOW
P18	VDDIOM	EBI	PE13	I/O	–	–	A13	O	TIOB1	I/O	PWML2	O	A13, LOW
N17	VDDIOM	EBI	PE14	I/O	–	–	A14	O	TCLK1	I	PWMH3	O	A14, LOW
N18	VDDIOM	EBI	PE15	I/O	–	–	A15	O	SCK3	I/O	TIOA0	I/O	A15, LOW
M15	VDDIOM	EBI	PE16	I/O	–	–	A16	O	RXD3	I	TIOB0	I/O	A16, LOW
N19	VDDIOM	EBI	PE17	I/O	–	–	A17	O	TXD3	O	TCLK0	I	A17, LOW
P19	VDDIOM	EBI	PE18	I/O	–	–	A18	O	TIOA5	I/O	MCI1_CK	I/O	A18, LOW
N16	VDDIOM	EBI	PE19	I/O	–	–	A19	O	TIOB5	I/O	MCI1_CDA	I/O	A19, LOW
M14	VDDIOM	EBI	PE20	I/O	–	–	A20	O	TCLK5	I	MCI1_DA0	I/O	A20, LOW
M18	VDDIOM	EBI	PE21	I/O	–	–	A23	O	TIOA4	I/O	MCI1_DA1	I/O	A23, LOW
M19	VDDIOM	EBI	PE22	I/O	–	–	A24	O	TIOB4	I/O	MCI1_DA2	I/O	A24, LOW
L18	VDDIOM	EBI	PE23	I/O	–	–	A25	O	TCLK4	I	MCI1_DA3	I/O	A25, LOW
L19	VDDIOM	EBI	PE24	I/O	–	–	NCS0	O	RTS3	O	–	–	NCS0, HIGH
M17	VDDIOM	EBI	PE25	I/O	–	–	NCS1	O	SCK4	I/O	IRQ	I	NCS1, HIGH
L15	VDDIOM	EBI	PE26	I/O	–	–	NCS2	O	RXD4	I	A18	O	NCS2, HIGH

Table 9-3: Processor Mode vs. Mode Field (Continued)

Mode	M[4:0]
ABT	10111
UND	11011
SYS	11111
Reserved	Other

9.4.3.1 CP15 Coprocessor

Coprocessor 15, or System Control Coprocessor CP15, is used to configure and control all the items in the list below:

- Cortex A5
- Caches (ICache, DCache and write buffer)
- MMU
- Security
- Other system options

To control these features, CP15 provides 16 additional registers. Refer to Table 9-4.

Table 9-4: CP15 Registers

Register	Name	Read/Write
0	ID Code(1)	Read/Unpredictable
0	Cache type(1)	Read/Unpredictable
1	Control(1)	Read/Write
1	Security(1)	Read/Write
2	Translation Table Base	Read/Write
3	Domain Access Control	Read/Write
4	Reserved	None
5	Data fault Status(1)	Read/Write
5	Instruction fault status	Read/Write
6	Fault Address	Read/Write
7	Cache and MMU Operations(1)	Read/Write
8	TLB operations	Unpredictable/Write
9	Cache lockdown(1)	Read/Write
10	TLB lockdown	Read/Write
11	Reserved	None
12	Interrupts management	Read/Write
12	Monitor vectors	Read-only
13	FCSE PID(1)	Read/Write
13	Context ID(1)	Read/Write
14	Reserved	None
15	Test configuration	Read/Write

Note 1: This register provides access to more than one register. The register accessed depends on the value of the CRm field or Opcode_2 field.

13.5.9 L2CC Event Counter 0 Configuration Register

Name:L2CC_ECFGR0

Address:0x00A00208

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–

23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–

15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–

7	6	5	4	3	2	1	0
–	–	ESRC				EIGEN	

EIGEN: Event Counter Interrupt Generation

Value	Name	Description
0x0	INT_DIS	Disables (default)
0x1	INT_EN_INCR	Enables with Increment condition
0x2	INT_EN_OVER	Enables with Overflow condition
0x3	INT_GEN_DIS	Disables Interrupt generation

ESRC: Event Counter Source

Value	Name	Description
0x0	CNT_DIS	Counter Disabled
0x1	SRC_CO	Source is CO
0x2	SRC_DRHIT	Source is DRHIT
0x3	SRC_DRREQ	Source is DRREQ
0x4	SRC_DWHIT	Source is DWHIT
0x5	SRC_DWREQ	Source is DWREQ
0x6	SRC_DWTREQ	Source is DWTREQ
0x7	SRC_IRHIT	Source is IRHIT
0x8	SRC_IRREQ	Source is IRREQ
0x9	SRC_WA	Source is WA
0xa	SRC_IPFALLOC	Source is IPFALLOC
0xb	SRC_EPFHIT	Source is EPFHIT
0xc	SRC_EPFALLOC	Source is EPFALLOC
0xd	SRC_SRRCVD	Source is SRRCVD
0xe	SRC_SRCONF	Source is SRCONF
0xf	SRC_EPFRCVD	Source is EPFRCVD

Table 15-3: Master to Slave Access on H64MX

		Master									
		0	1	2	3	4	5	6	7	8	9
Slave		Bridge from AXIMX (Core)	XDMAC0		XDMAC1		LCDC DMA		VDEC DMA	ISI DMA	Bridge from H32MX
			IF0	IF1	IF0	IF1					
0	Bridge from H64MX to AXIMX		X	X							
1	H64MX Peripheral Bridge	X		X		X					
2	VDEC	X									
3	DDR2 port0 - AESB	X									
4	DDR2 port1	X									
5	DDR2 port2						X				
6	DDR2 port3							X			
7	DDR2 port4								X	X	
8	DDR2 port5		X		X						
9	DDR2 port6			X		X					
10	DDR2 port7										X
11	Internal SRAM	X	X				X				X
12	Bridge from H64MX to H32MX	X	X	X	X	X					

15.4 MATRIX1 (H32MX)

15.4.1 Matrix Masters

The H32MX manages seven masters, which means that each master can perform an access concurrently with others, to an available slave.

This matrix can operate at MCK if MCK is lower than 90 MHz, or at MCK/2 if MCK is higher than 90 MHz. Refer to Section 27. “Power Management Controller (PMC)” for more details.

Each master has its own decoder, which is defined specifically for each master. In order to simplify the addressing, all the masters have the same decodings.

Table 15-4: List of H32MX Masters

Master No.	Name
0	Bridge from H64MX to H32MX
1	Integrity Check Monitor (ICM)
2	UHP EHCI DMA
3	UHP OHCI DMA
4	UDPHS DMA
5	GMAC0 DMA
6	GMAC1 DMA

18.5 Watchdog Timer (WDT) User Interface

Table 18-1: Register Mapping

Offset	Register	Name	Access	Reset
0x00	Control Register	WDT_CR	Write-only	–
0x04	Mode Register	WDT_MR	Read/Write	0x3FFF_2FFF
0x08	Status Register	WDT_SR	Read-only	0x0000_0000

Figure 29-13: Self-refresh Mode Entry, TIMEOUT = 0

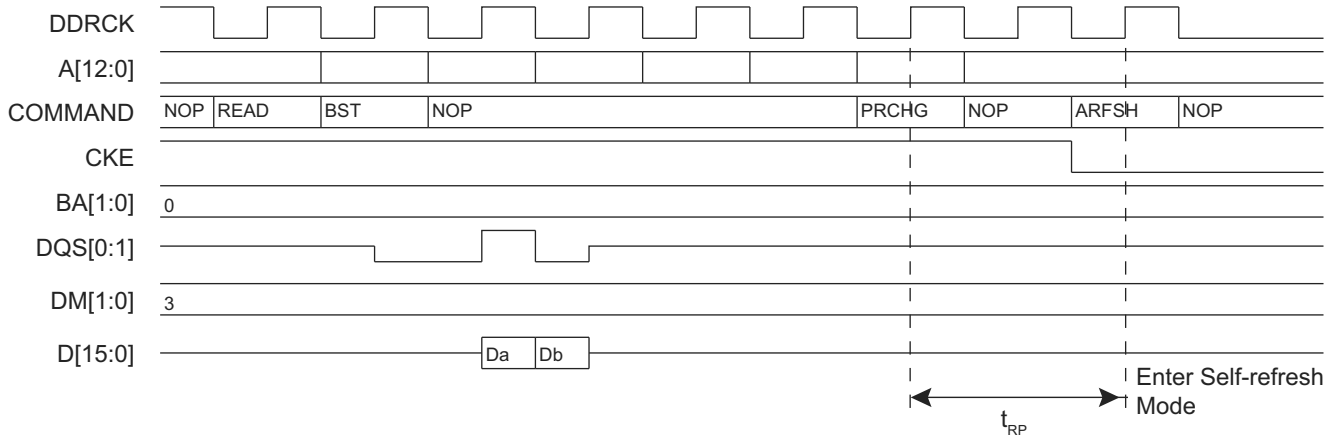


Figure 29-14: Self-refresh Mode Entry, TIMEOUT = 1 or 2

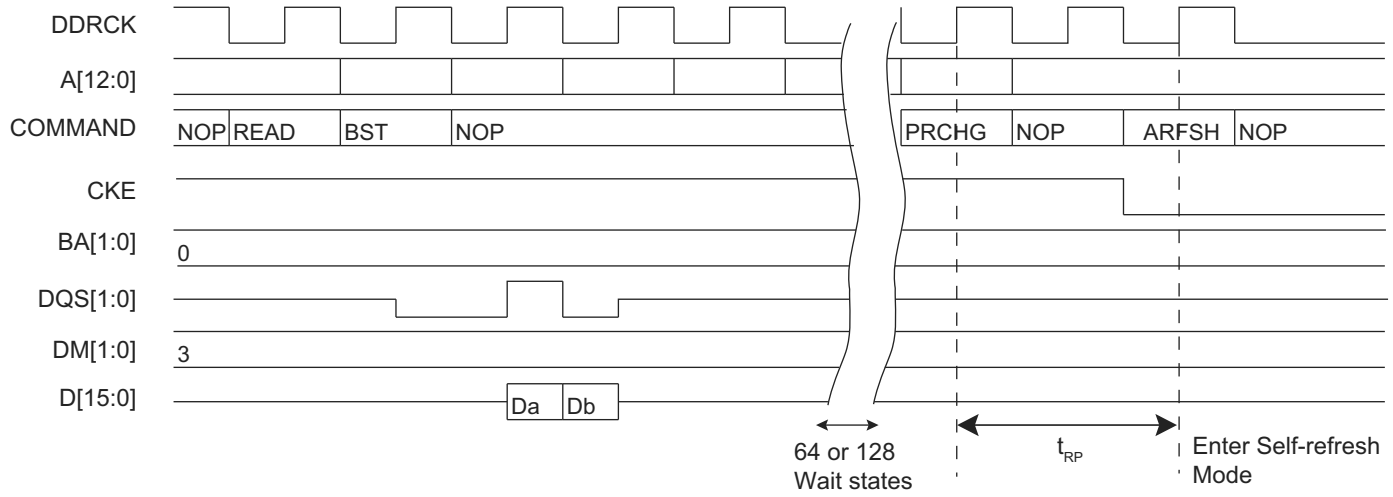
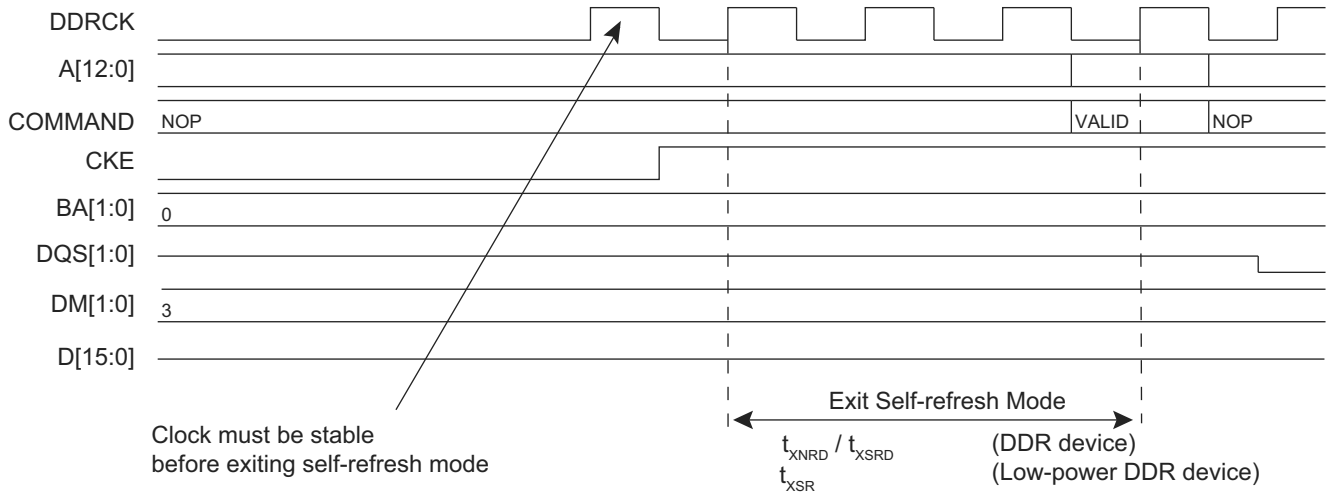


Figure 29-15: Self-refresh Mode Exit



TMRD: Load Mode Register Command to Activate or Refresh Command

Reset value is 2 DDRCK(1) clock cycles.

This field defines the delay between a Load mode register command and an Activate or Refresh command in number of DDRCK(1) clock cycles. The number of cycles is between 0 and 15. For low-power DDR2-SDRAM, this field is equivalent to TMRW timing.

Note 1: DDRCK is the clock that drives the SDRAM device.

30.4 I/O Lines Description

Table 30-1: I/O Line Description

Name	Description	Type	Active Level
NCS[3:0]	Static Memory Controller Chip Select Lines	Output	Low
NRD	Read Signal	Output	Low
NWR0/NWE	Write 0/Write Enable Signal	Output	Low
A0/NBS0	Address Bit 0/Byte 0 Select Signal	Output	Low
NWR1/NBS1	Write 1/Byte 1 Select Signal	Output	Low
A[25:1]	Address Bus	Output	–
D[15:0]	Data Bus	I/O	–
NWAIT	External Wait Signal	Input	Low
NANDRDY	NAND Flash Ready/Busy	Input	–
NANDWE	NAND Flash Write Enable	Output	Low
NANDOE	NAND Flash Output Enable	Output	Low
NANDALE	NAND Flash Address Latch Enable	Output	–
NANDCLE	NAND Flash Command Latch Enable	Output	–

30.5 Multiplexed Signals

Table 30-2: Static Memory Controller (SMC) Multiplexed Signals

Multiplexed Signals		Related Function
NWR0	NWE	Byte-write or Byte-select access, refer to Section 30.9.2.1 “Byte Write Access” and Section 30.9.2.2 “Byte Select Access”
A0	NBS0	8-bit or 16-bit data bus, refer to Section 30.9.1 “Data Bus Width”
A22	NANDCLE	NAND Flash Command Latch Enable
A21	NANDALE	NAND Flash Address Latch Enable
NWR1	NBS1	Byte-write or Byte-select access, refer to Section 30.9.2.1 “Byte Write Access” and Section 30.9.2.2 “Byte Select Access”
A1	–	8-/16-bit data bus, refer to Section 30.9.1 “Data Bus Width” Byte-write or Byte-select access, refer to Section 30.9.2.1 “Byte Write Access” and Section 30.9.2.2 “Byte Select Access”

30.20.2 NFC Control Register

Name: HSMC_CTRL

Address:0xFC05C004

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	–	–	NFCDIS	NFCEN

NFCEN: NAND Flash Controller Enable

0: No effect

1: Enable the NAND Flash controller.

NFCDIS: NAND Flash Controller Disable

0: No effect

1: Disable the NAND Flash controller.

32.6.6.1 Chrominance Upsampling Algorithm

1. Read line n from chrominance cache and interpolate [x/2,0] chrominance component filling the 1 x 2 kernel with line n. If the chrominance cache is empty, then fetch the first line from external memory and interpolate from the external memory. Duplicate the last chrominance at the end of line.
2. Fetch line n+1 from external memory, write line n + 1 to chrominance cache, read line n from the chrominance cache. interpolate [0,y/2], [x/2,y/2] and [x, y/2] filling the 2x2 kernel with line n and n+1. Duplicate the last chrominance line to generate the last interpolated line.
3. Repeat step 1 and step 2.

32.6.7 Line and Pixel Striding

The LCD module includes a technique to increment the memory address by a programmable amount when the end of line has been reached. This offset is referred to as XSTRIDE and is defined on a per overlay basis. Additionally, the PSTRIDE field allows a programmable jump at the pixel level. Pixel stride is the value from one pixel to the next.

32.6.7.1 Line Striding

When the end of line has been reached, the DMA address counter points to the next pixel address. The channel DMA address register is added to the XSTRIDE field, and then updated. If XSTRIDE is set to '0', the DMA address register remains unchanged. The XSTRIDE field of the channel configuration register is aligned to the pixel size boundary. The XSTRIDE field is a two's complement number. The following formula applies at the line boundary and indicates how the DMA controller computes the next pixel address. The function Sizeof() returns the number of bytes required to store a pixel.

$$NextPixelAddress = CurrentPixelAddress + Sizeof(pixel) + XSTRIDE$$

32.6.7.2 Pixel Striding

The DMA channel engine may optionally fetch non-contiguous pixels. The channel DMA address register is added to the PSTRIDE field and then updated. If PSTRIDE is set to zero, the DMA address register remains unchanged and pixels are contiguous. The PSTRIDE field of the channel configuration register is aligned to the pixel size boundary. The PSTRIDE is a two's complement number. The following formula applies at the pixel boundary and indicates how the DMA controller computes the next pixel address. The function Sizeof() returns the number of bytes required to store a pixel.

$$NextPixelAddress = CurrentPixelAddress + Sizeof(pixel) + PSTRIDE$$

32.6.8 Color Space Conversion Unit

The color space conversion unit converts Luminance Chrominance color space into the Red Green Blue color space. The conversion matrix is defined below and is fully programmable through the LCD user interface.

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} CSCRY & CSCRU & CSCRV \\ CSCGY & CSCGU & CSCGV \\ CSCBY & CSCBU & CSCBV \end{bmatrix} \bullet \begin{bmatrix} Y - Yoff \\ Cb - Cboff \\ Cr - Croff \end{bmatrix}$$

Color space conversion coefficients are defined with the following equation:

$$CSC_{ij} = \frac{1}{2^7} \cdot \left[-2^9 \cdot c_9 + \sum_{n=0}^{8} c_n \cdot 2^n \right]$$

Color space conversion coefficients are defined with one sign bit, 2 integer bits and 7 fractional bits. The range of the CSCij coefficients is defined below with a step of 1/128.

$$-4 \leq CSC_{ij} \leq 3.9921875$$

Additionally, a set scaling factor {Yoff, Cboff, Croff} can be applied.

32.7.6 LCD Controller Configuration Register 5

Name: LCDC_LCDCFG5

Address:0xF0000014

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–

23	22	21	20	19	18	17	16
GUARDTIME							

15	14	13	12	11	10	9	8
–	–	VSPHO	VSPSU	–	–	MODE	

7	6	5	4	3	2	1	0
DISPDLY	DITHER	–	DISPPOL	VSPDLYE	VSPDLYS	VSPOL	HSPOL

HSPOL: Horizontal Synchronization Pulse Polarity

0: Active High

1: Active Low

VSPOL: Vertical Synchronization Pulse Polarity

0: Active High

1: Active Low

VSPDLYS: Vertical Synchronization Pulse Start

0: The first active edge of the Vertical synchronization pulse is synchronous with the second edge of the horizontal pulse.

1: The first active edge of the Vertical synchronization pulse is synchronous with the first edge of the horizontal pulse.

VSPDLYE: Vertical Synchronization Pulse End

0: The second active edge of the Vertical synchronization pulse is synchronous with the second edge of the horizontal pulse.

1: The second active edge of the Vertical synchronization pulse is synchronous with the first edge of the horizontal pulse.

DISPPOL: Display Signal Polarity

0: Active High

1: Active Low

DITHER: LCD Controller Dithering

0: Dithering logical unit is disabled

1: Dithering logical unit is activated

DISPDLY: LCD Controller Display Power Signal Synchronization

0: The LCD_DISP signal is asserted synchronously with the second active edge of the horizontal pulse.

1: The LCD_DISP signal is asserted asynchronously with both edges of the horizontal pulse.

34.6.4 ISI Preview Decimation Factor Register

Name: ISI_PDECF

Address:0xF000800C

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
DEC_FACTOR							

DEC_FACTOR: Decimation Factor

DEC_FACTOR is 8-bit width, range is from 16 to 255. Values from 0 to 16 do not perform any decimation.

(Refer to “DETACH: Detach Command”.)

DETACH	PULLD_DIS	DP	DM	Condition
0	0	Pull up	Pull down	Not recommended
0	1	Pull up	High impedance state	VBUS present
1	0	Pull down	Pull down	No VBUS
1	1	High impedance state	High impedance state	VBUS present & software disconnect

37.8.31 GMAC Type ID Match 4 Register

Name:GMAC_TIDM4

Address:0xF80200B4 (0), 0xFC0280B4 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
ENID4	–	–	–	–	–	–	–

23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–

15	14	13	12	11	10	9	8
TID							

7	6	5	4	3	2	1	0
TID							

TID: Type ID Match 4

For use in comparisons with received frames type ID/length frames.

ENID4: Enable Copying of TID Matched Frames

0: TID is not part of the comparison match.

1: TID is processed for the comparison match.

38. High Speed Multimedia Card Interface (HSMCI)

38.1 Description

The High Speed Multimedia Card Interface (HSMCI) supports the MultiMedia Card (MMC) Specification V4.3, the SD Memory Card Specification V2.0, the SDIO V2.0 specification and CE-ATA V1.1.

The HSMCI includes a command register, response registers, data registers, timeout counters and error detection logic that automatically handle the transmission of commands and, when required, the reception of the associated responses and data with a limited processor overhead.

The HSMCI operates at a rate of up to Master Clock divided by 2 and supports the interfacing of 1 slot(s). Each slot may be used to interface with a High Speed MultiMedia Card bus (up to 30 Cards) or with an SD Memory Card. Only one slot can be selected at a time (slots are multiplexed). A bit field in the SD Card Register performs this selection.

The SD Memory Card communication is based on a 9-pin interface (clock, command, four data and three power lines) and the High Speed MultiMedia Card on a 7-pin interface (clock, command, one data, three power lines and one reserved for future use).

The SD Memory Card interface also supports High Speed MultiMedia Card operations. The main differences between SD and High Speed MultiMedia Cards are the initialization process and the bus topology.

HSMCI fully supports CE-ATA Revision 1.1, built on the MMC System Specification v4.0. The module includes dedicated hardware to issue the command completion signal and capture the host command completion signal disable.

38.2 Embedded Characteristics

- Compatible with MultiMedia Card Specification Version 4.3
- Compatible with SD Memory Card Specification Version 2.0
- Compatible with SDIO Specification Version 2.0
- Compatible with CE-ATA Specification 1.1
- Cards Clock Rate Up to Master Clock Divided by 2
- Boot Operation Mode Support
- High Speed Mode Support
- Embedded Power Management to Slow Down Clock Rate When Not Used
- Supports 1 Multiplexed Slot(s)
  - Each Slot for either a High Speed MultiMedia Card Bus (Up to 30 Cards) or an SD Memory Card
- Support for Stream, Block and Multi-block Data Read and Write
  - Minimizes Processor Intervention for Large Buffer Transfers
- Built in FIFO (from 16 to 256 bytes) with Large Memory Aperture Supporting Incremental Access
- Support for CE-ATA Completion Signal Disable Command
- Protection Against Unexpected Modification On-the-Fly of the Configuration Registers

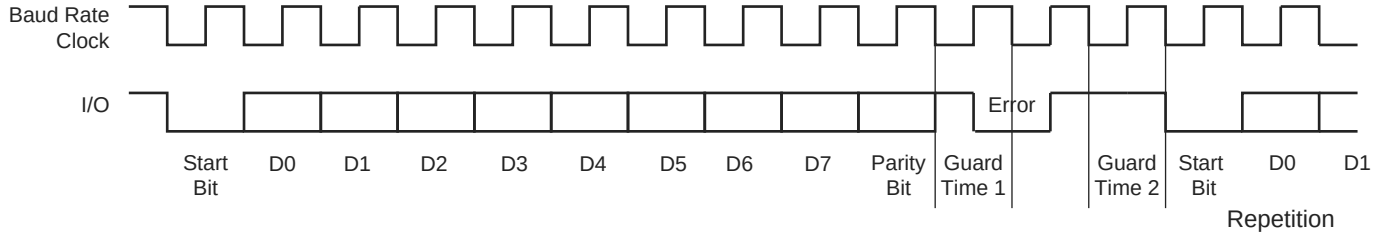
38.14 High Speed MultiMedia Card Interface (HSMCI) User Interface

Table 38-9: Register Mapping

Offset	Register	Name	Access	Reset
0x00	Control Register	HSMCI_CR	Write-only	–
0x04	Mode Register	HSMCI_MR	Read/Write	0x0
0x08	Data Timeout Register	HSMCI_DTOR	Read/Write	0x0
0x0C	SD/SDIO Card Register	HSMCI_SDCR	Read/Write	0x0
0x10	Argument Register	HSMCI_ARGR	Read/Write	0x0
0x14	Command Register	HSMCI_CMDR	Write-only	–
0x18	Block Register	HSMCI_BLKR	Read/Write	0x0
0x1C	Completion Signal Timeout Register	HSMCI_CSTOR	Read/Write	0x0
0x20	Response Register(1)	HSMCI_RSPR	Read-only	0x0
0x24	Response Register(1)	HSMCI_RSPR	Read-only	0x0
0x28	Response Register(1)	HSMCI_RSPR	Read-only	0x0
0x2C	Response Register(1)	HSMCI_RSPR	Read-only	0x0
0x30	Receive Data Register	HSMCI_RDR	Read-only	0x0
0x34	Transmit Data Register	HSMCI_TDR	Write-only	–
0x38–0x3C	Reserved	–	–	–
0x40	Status Register	HSMCI_SR	Read-only	0xC0E5
0x44	Interrupt Enable Register	HSMCI_IER	Write-only	–
0x48	Interrupt Disable Register	HSMCI_IDR	Write-only	–
0x4C	Interrupt Mask Register	HSMCI_IMR	Read-only	0x0
0x50	DMA Configuration Register	HSMCI_DMA	Read/Write	0x0
0x54	Configuration Register	HSMCI_CFG	Read/Write	0x0
0x58–0xE0	Reserved	–	–	–
0xE4	Write Protection Mode Register	HSMCI_WPMR	Read/Write	0x0
0xE8	Write Protection Status Register	HSMCI_WPSR	Read-only	0x0
0xEC–0xFC	Reserved	–	–	–
0x100–0x1FC	Reserved	–	–	–
0x200	FIFO Memory Aperture0	HSMCI_FIFO0	Read/Write	0x0
...	...	...	...	...
0x5FC	FIFO Memory Aperture255	HSMCI_FIFO255	Read/Write	0x0

Note 1: The Response Register can be read by N accesses at the same HSMCI_RSPR or at consecutive addresses (0x20 to 0x2C). N depends on the size of the response.

Figure 44-31: T = 0 Protocol with Parity Error



- Receive Error Counter

The USART receiver also records the total number of errors. This can be read in the Number of Errors (US_NER) register. The NB_ERRORS field can record up to 255 errors. Reading US_NER automatically clears the NB_ERRORS field.

- Receive NACK Inhibit

The USART can be configured to inhibit an error. This is done by writing a ‘1’ to US_MR.INACK. In this case, no error signal is driven on the I/O line even if a parity bit is detected.

Moreover, if INACK = 1, the erroneous received character is stored in the Receive Holding register as if no error occurred, and the RXRDY bit rises.

- Transmit Character Repetition

When the USART is transmitting a character and gets a NACK, it can automatically repeat the character before moving on to the next one. Repetition is enabled by writing US_MR.MAX_ITERATION to a value greater than 0. Each character can be transmitted up to eight times; the first transmission plus seven repetitions.

If MAX_ITERATION does not equal zero, the USART repeats the character as many times as the value loaded in MAX_ITERATION.

When the USART repetition number reaches MAX_ITERATION and the last repeated character is not acknowledged, the US_CSR.ITER is set. If the repetition of the character is acknowledged by the receiver, the repetitions are stopped and the iteration counter is cleared.

US_CSR.ITER can be cleared by writing a ‘1’ to US_CR.RSTIT.

- Disable Successive Receive NACK

The receiver can limit the number of successive NACKs sent back to the remote transmitter. This is programmed by setting US_MR.DSNACK. The maximum number of NACKs transmitted is configured in US_MR.MAX_ITERATION. As soon as MAX_ITERATION is reached, no error signal is driven on the I/O line and US_CSR.ITER is set.

44.6.4.3 Protocol T = 1

When operating in ISO7816 protocol T = 1, the transmission is similar to an asynchronous format with only one stop bit. The parity is generated when transmitting and checked when receiving. Parity error detection sets US_CSR.PARE.

44.6.5 IrDA Mode

The USART features an IrDA mode supplying half-duplex point-to-point wireless communication. It embeds the modulator and demodulator which allows a glueless connection to the infrared transceivers, as shown in Figure 44-32. The modulator and demodulator are compliant with the IrDA specification version 1.1 and support data transfer speeds ranging from 2.4 kbit/s to 115.2 kbit/s.

The IrDA mode is enabled by writing the value 0x8 to US_MR.USART_MODE. The IrDA Filter register (US_IF) is used to configure the demodulator filter. The USART transmitter and receiver operate in a normal Asynchronous mode and all parameters are accessible. Note that the modulator and the demodulator are activated.

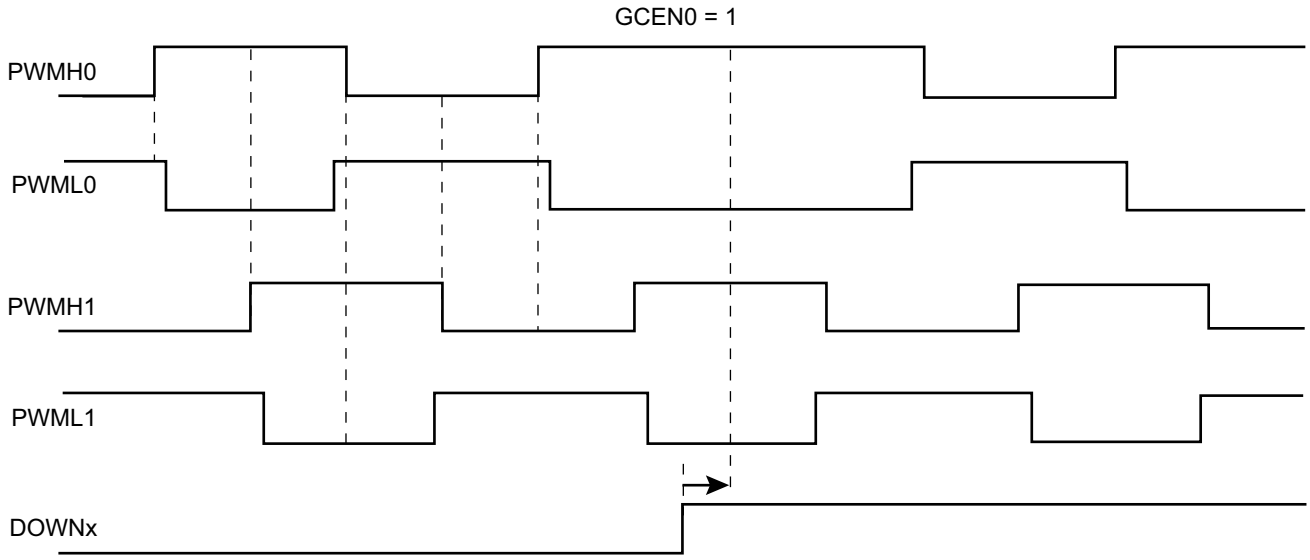
47.6.2.3 2-bit Gray Up/Down Counter for Stepper Motor

A pair of channels may provide a 2-bit gray count waveform on two outputs. Dead-time generator and other downstream logic can be configured on these channels.

Up or Down Count mode can be configured on-the-fly by means of PWM_SMMR configuration registers.

When GCEN0 is set to ‘1’, channels 0 and 1 outputs are driven with gray counter.

Figure 47-6: 2-bit Gray Up/Down Counter



47.6.2.4 Dead-Time Generator

The dead-time generator uses the comparator output OCx to provide the two complementary outputs DTOHx and DTOLx, which allows the PWM macrocell to drive external power control switches safely. When the dead-time generator is enabled by setting the bit DTE to 1 or 0 in the PWM Channel Mode Register (PWM_CMRx), dead-times (also called dead-bands or non-overlapping times) are inserted between the edges of the two complementary outputs DTOHx and DTOLx. Note that enabling or disabling the dead-time generator is allowed only if the channel is disabled.

The dead-time is adjustable by the PWM Channel Dead Time Register (PWM_DTx). Each output of the dead-time generator can be adjusted separately by DTH and DTL. The dead-time values can be updated synchronously to the PWM period by using the PWM Channel Dead Time Update Register (PWM_DTUPDx).

The dead-time is based on a specific counter which uses the same selected clock that feeds the channel counter of the comparator. Depending on the edge and the configuration of the dead-time, DTOHx and DTOLx are delayed until the counter has reached the value defined by DTH or DTL. An inverted configuration bit (DTHI and DTLI bit in PWM_CMRx) is provided for each output to invert the dead-time outputs. The following figure shows the waveform of the dead-time generator.

47.7.35 PWM Comparison x Value Register

Name:PWM_CMPVx

Address:0xF800C130 [0], 0xF800C140 [1], 0xF800C150 [2], 0xF800C160 [3], 0xF800C170 [4], 0xF800C180 [5], 0xF800C190 [6], 0xF800C1A0 [7]

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	CVM

23	22	21	20	19	18	17	16
CV							

15	14	13	12	11	10	9	8
CV							

7	6	5	4	3	2	1	0
CV							

Only the first 16 bits (channel counter size) of field CV are significant.

CV: Comparison x Value

Define the comparison x value to be compared with the counter of the channel 0.

CVM: Comparison x Value Mode

0: The comparison x between the counter of the channel 0 and the comparison x value is performed when this counter is incrementing.

1: The comparison x between the counter of the channel 0 and the comparison x value is performed when this counter is decrementing.

Note: This bit is not relevant if the counter of the channel 0 is left-aligned (CALG = 0 in PWM Channel Mode Register)

The input to the encryption processes of the CBC mode includes, in addition to the plaintext, a 128-bit data block called the initialization vector (IV), which must be set in the Initialization Vector Registers (AESB_IVRx). The initialization vector is used in an initial step in the encryption of a message and in the corresponding decryption of the message. The Initialization Vector Registers are also used by the CTR mode to set the counter value.

53.4.1 Operating Modes

The AESB supports the following modes of operation:

- ECB—Electronic Code Book
- CBC—Cipher Block Chaining
- CTR—Counter

The data pre-processing, post-processing and data chaining for the operating modes are performed automatically. Refer to the *NIST Special Publication 800-38A Recommendation* for more complete information.

The modes are selected by the OPMOD field in AESB_MR.

In CTR mode, the size of the block counter embedded in the module is 16 bits. Therefore, there is a rollover after processing 1 megabyte of data. If the file to be processed is greater than 1 megabyte, this file must be split into fragments of 1 megabyte or less for the first fragment if the initial value of the counter is greater than 0. Prior to loading the first fragment into AESB_IDATARx registers, the AESB_IVRx registers must be cleared. For any fragment, after the transfer is completed and prior to transferring the next fragment, AESB_IVR0 must be programmed so that the fragment number (0 for the first fragment, 1 for the second one, and so on) is written in the 16 MSB of AESB_IVR0.

If the initial value of the counter is greater than 0 and the data buffer size to be processed is greater than 1 megabyte, the size of the first fragment to be processed must be 1 megabyte minus 16x(initial value) to prevent a rollover of the internal 1-bit counter.

53.4.2 Double Input Buffer

The input data register can be double-buffered to reduce the runtime of large files.

This mode allows writing a new message block when the previous message block is being processed.

The DUALBUFF bit in register AESB_MR must be set to 1 to access the double buffer.

53.4.3 Start Modes

The SMOD field in register AESB_MR allows selection of the Encryption (or Decryption) Start mode.

53.4.3.1 Manual Mode

The sequence is as follows:

1. Write AESB_MR with all required fields, including but not limited to SMOD and OPMOD.
2. Write the 128-bit key in the Key Registers (AESB_KEYWRx).
3. Write the initialization vector (or counter) in the Initialization Vector Registers (AESB_IVRx).

   Note: The Initialization Vector Registers concern all modes except ECB.
4. Set the DATRDY (Data Ready) bit in the AESB Interrupt Enable Register (AESB_IER) depending on whether an interrupt is required, or not, at the end of processing.
5. Write the data to be encrypted/decrypted in the authorized Input Data Registers (refer to Table 53-2).

Table 53-2: Authorized Input Data Registers

Operating Mode	Input Data Registers to Write
ECB	All
CBC	All
CTR	All

6. Set the START bit in the AESB Control Register (AESB_CR) to begin the encryption or decryption process.
7. When processing is complete, the DATRDY bit in the AESB Interrupt Status Register (AESB_ISR) raises. If an interrupt has been enabled by setting the DATRDY bit in AESB_IER, the interrupt line of the AESB is activated.
8. When the software reads one of the Output Data Registers (AESB_ODATARx), the AESB_ISR.DATRDY bit is automatically cleared.