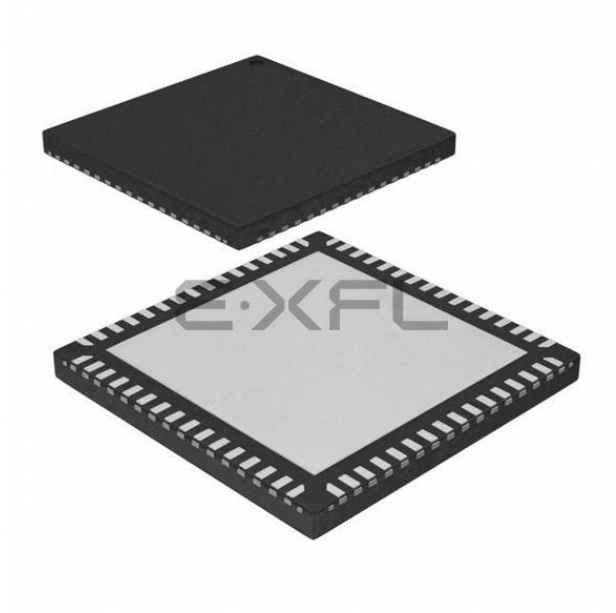


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	ARM® Cortex®-M7
Core Size	32-Bit Single-Core
Speed	300MHz
Connectivity	I²C, IrDA, LINbus, SPI, SSC, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, I²S, POR, PWM, WDT
Number of I/O	44
Program Memory Size	1MB (1M x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	384K x 8
Voltage - Supply (Vcc/Vdd)	1.7V ~ 3.6V
Data Converters	A/D 5x12b; D/A 1x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	64-VFQFN Exposed Pad
Supplier Device Package	64-QFN (9x9)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsams70j20b-mn

11. Memories

11.1 Embedded Memories

11.1.1 Internal SRAM

SAM E70/S70/V70/V71 devices embed 384 Kbytes or 256 Kbytes of high-speed SRAM.

The SRAM is accessible over the system Cortex-M bus at address 0x2040 0000.

SAM E70/S70/V70/V71 devices embed a Multi-Port SRAM with four ports to optimize the bandwidth and latency. The priorities, defined in the Bus Matrix for each SRAM port slave are propagated, for each request, up to the SRAM slaves.

The Bus Matrix supports four priority levels: Normal, Bandwidth-sensitive, Latency-sensitive and Latency-critical in order to increase the overall processor performance while securing the high-priority latency-critical requests from the peripherals.

The SRAM controller manages interleaved addressing of SRAM blocks to minimize access latencies. It uses Bus Matrix priorities to give the priority to the most urgent request. The less urgent request is performed no later than the next cycle.

Two SRAM slave ports are dedicated to the Cortex-M7 while two ports are shared by the AHB masters.

11.1.2 Tightly Coupled Memory (TCM) Interface

SAM E70/S70/V70/V71 devices embed Tightly Coupled Memory (TCM) running at processor speed.

- ITCM is a single 64-bit interface, based at 0x0000 0000 (code region).
- DTCM is composed of dual 32-bit interfaces interleaved, based at 0x2000 0000 (data region).

ITCM and DTCM are enabled/disabled in the ITCMR and DTCMR registers in ARM SCB.

DTCM is enabled by default at reset. ITCM is disabled by default at reset.

There are four TCM configurations controlled by software. When enabled, ITCM is located at 0x0000 0000, overlapping ROM or Flash depending on the general-purpose NVM bit 1 (GPNVM). The configuration is done with GPNVM bits [8:7].

Table 11-1. TCM Configurations in Kbytes

ITCM	DTCM	SRAM for 384K RAM-based	SRAM for 256K RAM-based	GPNVM Bits [8:7]
0	0	384	256	0
32	32	320	192	1
64	64	256	128	2
128	128	128	0	3

Accesses made to TCM regions when the relevant TCM is disabled and accesses made to the Code and SRAM region above the TCM size limit are performed on the AHB matrix, i.e., on internal Flash or on ROM depending on remap GPNVM bit.

Accesses made to the SRAM above the size limit will not generate aborts.

The Memory Protection Unit (MPU) can to be used to protect these areas.

19.4.2 Bus Matrix Slave Configuration Registers

Name: MATRIX_SCFGx
Offset: 0x40 + x*0x04 [x=0..8]
Reset: 0x000001FF
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			FIXED_DEFMSTR[3:0]				DEFMSTR_TYPE[1:0]	
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
							SLOT_CYCLE[8:7]	
Access							R/W	R/W
Reset							0	1

Bit	7	6	5	4	3	2	1	0
	SLOT_CYCLE[6:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	1	1	1	1	1	1	1	

Bits 21:18 – FIXED_DEFMSTR[3:0] Fixed Default Master
Number of the Default Master for this slave. Only used if DEFMSTR_TYPE is 2. Specifying the number of a master which is not connected to the selected slave is equivalent to setting DEFMSTR_TYPE to 0.

Bits 17:16 – DEFMSTR_TYPE[1:0] Default Master Type

Value	Name	Description
0	NONE	No Default Master—At the end of the current slave access, if no other master request is pending, the slave is disconnected from all masters. This results in a one clock cycle latency for the first access of a burst transfer or for a single access.
1	LAST	Last Default Master—At the end of the current slave access, if no other master request is pending, the slave stays connected to the last master having accessed it. This results in not having one clock cycle latency when the last master tries to access the slave again.
2	FIXED	Fixed Default Master—At the end of the current slave access, if no other master request is pending, the slave connects to the fixed master the number that has been written in the FIXED_DEFMSTR field.

Read CKGR_MCFR until the MAINFRDY field is set, after which the user can read CKGR_MCFR.MAINF by performing an additional read. This provides the number of Main clock cycles that have been counted during a period of 16 SLCK cycles.

If MAINF = 0, switch MAINCK to the Main RC Oscillator by clearing CKGR_MOR.MOSCSEL. If MAINF ≠ 0, proceed to Step 6.

6. Set PLLA and Divider (if not required, proceed to Step 7.):
   All parameters needed to configure PLLA and the divider are located in CKGR_PLLAR.

   CKGR_PLLAR.DIVA is used to control the divider. This parameter can be programmed between 0 and 127. Divider output is divider input divided by DIVA parameter. By default, DIVA field is cleared which means that the divider and PLLA are turned off.

   CKGR_PLLAR.MULA is the PLLA multiplier factor. This parameter can be programmed between 0 and 62. If MULA is cleared, PLLA will be turned off, otherwise the PLLA output frequency is PLLA input frequency multiplied by (MULA + 1).

   CKGR_PLLAR.PLLACOUNT specifies the number of SLCK cycles before PMC_SR.LOCKA is set after CKGR_PLLAR has been written.

   Once CKGR_PLLAR has been written, the user must wait for PMC_SR.LOCKA to be set. This can be done either by polling PMC_SR.LOCKA or by waiting for the interrupt line to be raised if the associated interrupt source (LOCKA) has been enabled in PMC_IER. All fields in CKGR_PLLAR can be programmed in a single write operation. If MULA or DIVA is modified, the LOCKA bit goes low to indicate that PLLA is not yet ready. When PLLA is locked, LOCKA is set again. The user must wait for the LOCKA bit to be set before using the PLLA output clock.
7. Select MCK and HCLK:
   MCK and HCLK are configurable via PMC_MCKR.

   CSS is used to select the clock source of MCK and HCLK. By default, the selected clock source is MAINCK.

   PRES is used to define the HCLK and MCK prescaler.s The user can choose between different values (1, 2, 3, 4, 8, 16, 32, 64). Prescaler output is the selected clock source frequency divided by the PRES value.

   MDIV is used to define the MCK divider. It is possible to choose between different values (0, 1, 2, 3). MCK output is the HCLK frequency divided by 1, 2, 3 or 4, depending on the value programmed in MDIV.

   By default, MDIV is cleared, which indicates that the HCLK is equal to MCK.

   Once the PMC_MCKR has been written, the user must wait for PMC_SR.MCKRDY to be set. This can be done either by polling PMC_SR.MCKRDY or by waiting for the interrupt line to be raised if the associated interrupt source (MCKRDY) has been enabled in PMC_IER. PMC_MCKR must not be programmed in a single write operation. The programming sequence for PMC_MCKR is as follows:

   If a new value for PMC_MCKR.CSS corresponds to any of the available PLL clocks:

   a. Program PMC_MCKR.PRES.

   b. Wait for PMC_SR.MCKRDY to be set.

   c. Program PMC_MCKR.MDIV.

   d. Wait for PMC_SR.MCKRDY to be set.

31.20.4 PMC Peripheral Clock Enable Register 0

Name: PMC_PCER0
Offset: 0x0010
Property: Write-only

This register can only be written if the WPEN bit is cleared in the PMC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	PID31	PID30	PID29	PID28	PID27	PID26	PID25	PID24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	PID23	PID22	PID21	PID20	PID19	PID18	PID17	PID16
Access								
Reset								

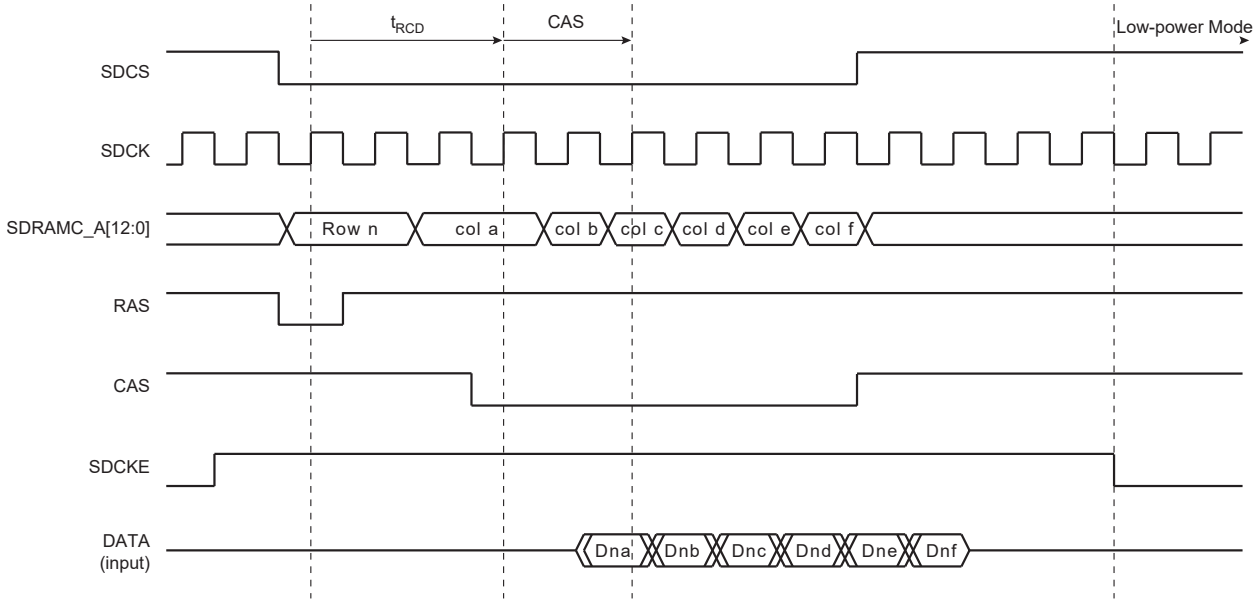
Bit	15	14	13	12	11	10	9	8
	PID15	PID14	PID13	PID12	PID11	PID10	PID9	PID8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	PID7							
Access								
Reset								

Bits 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – PIDx
Peripheral Clock x Enable

Value	Description
0	No effect.
1	Enables the corresponding peripheral clock. **Note:** 1. PIDx refers to identifiers defined in the section “Peripheral Identifiers”. Other peripherals can be enabled in PMC_PCER1 (see "PMC Peripheral Clock Enable Register 1"). 2. Programming the control bits of the Peripheral ID that are not implemented has no effect on the behavior of the PMC.

Figure 34-7. Low-power Mode Behavior



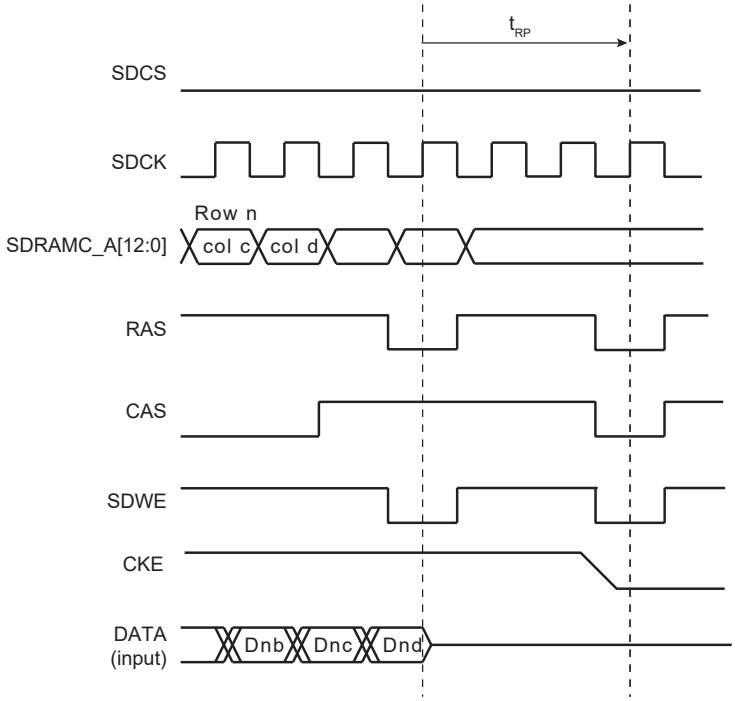
34.6.5.3 Deep Powerdown Mode

This mode is selected by configuring SDRAMC_LPR.LPCB to 3. When this mode is activated, all internal voltage generators inside the SDRAM are stopped and all data is lost.

When this mode is enabled, the application must not access the SDRAM until a new initialization sequence is done (see “SDRAM Device Initialization”).

Refer to the following figure.

Figure 34-8. Deep Powerdown Mode Behavior



Offset	Name	Bit Pos.								
		23:16								
		31:24								
0x58	ISI_DMA_C_DSCR	7:0	C_DSCR[5:0]							
		15:8	C_DSCR[13:6]							
		23:16	C_DSCR[21:14]							
		31:24	C_DSCR[29:22]							
0x5C ... 0xE3	Reserved									
0xE4	ISI_WPMR	7:0								WPEN
		15:8	WPKEY[7:0]							
		23:16	WPKEY[15:8]							
		31:24	WPKEY[23:16]							
0xE8	ISI_WPSR	7:0								WPVS
		15:8	WPVSRC[7:0]							
		23:16	WPVSRC[15:8]							
		31:24								

37.6.20 DMA Preview Descriptor Address Register

Name: ISI_DMA_P_DSCR
Offset: 0x4C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	P_DSCR[29:22]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	P_DSCR[21:14]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	P_DSCR[13:6]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	P_DSCR[5:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0		

Bits 31:2 – P_DSCR[29:0] Preview Descriptor Base Address
This address is word-aligned.

38.8.78 GMAC Receive Resource Errors Register

Name: GMAC_RRE
Offset: 0x1A0
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							RXRER[17:16]	
Access							R	R
Reset							0	0

Bit	15	14	13	12	11	10	9	8
	RXRER[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

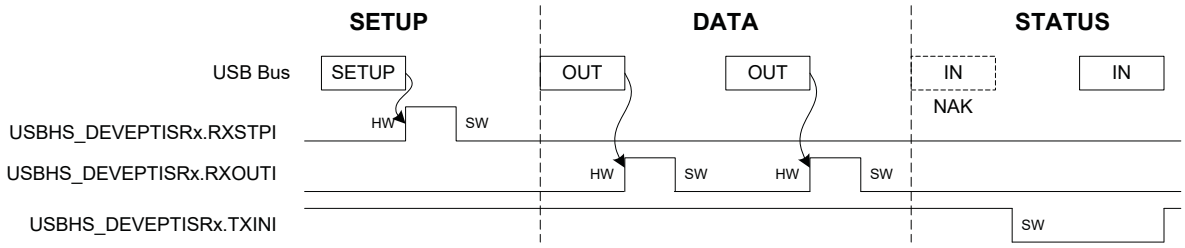
Bit	7	6	5	4	3	2	1	0
	RXRER[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 17:0 – RXRER[17:0] Receive Resource Errors
This bit field counts frames that are not an integral number of bytes long and have bad CRC when their length is truncated to an integral number of Bytes and are between 64 and 1518 Bytes in length (1536 if GMAC_NCFGR.MAXFS=1). This bit field is also incremented if a symbol error is detected and the frame is of valid length and does not have an integral number of Bytes.

Figure 39-9 shows a control write transaction. During the status stage, the controller does not necessarily send a NAK on the first IN token:

- if the user knows the exact number of descriptor bytes that must be read, it can then anticipate the status stage and send a zero-length packet after the next IN token, or
- it can read the bytes and wait for the NAKed IN Interrupt (USBHS_DEVEPTISRx.NAKINI), which acknowledges that all the bytes have been sent by the host and that the transaction is now in the status stage.

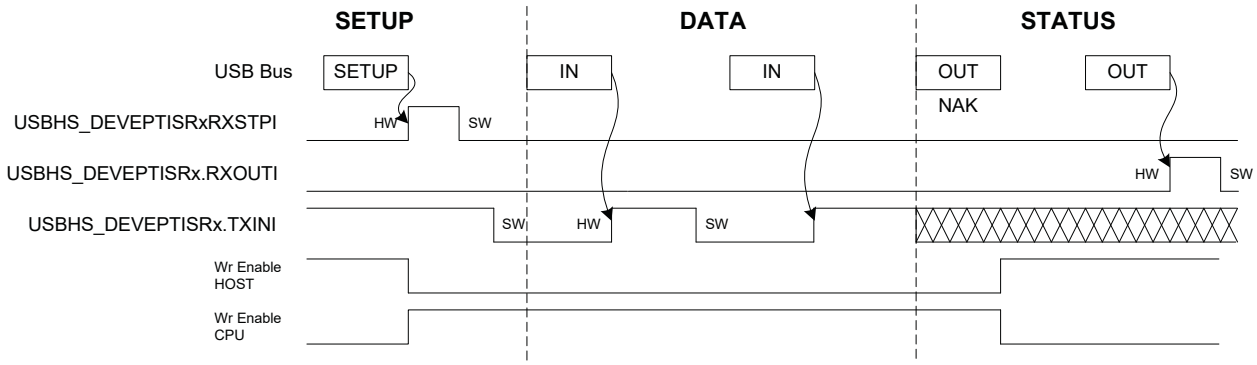
Figure 39-9. Control Write



Control Read

Figure 39-10 shows a control read transaction. The USBHS has to manage the simultaneous write requests from the CPU and the USB host.

Figure 39-10. Control Read



A NAK handshake is always generated on the first status stage command.

When the controller detects the status stage, all data written by the CPU is lost and clearing USBHS_DEVEPTISRx.TXINI has no effect.

The user checks if the transmission or the reception is complete.

The OUT retry is always ACKed. This reception sets USBHS_DEVEPTISRx.RXOUTI and USBHS_DEVEPTISRx.TXINI. Handle this with the following software algorithm:

set TXINI

wait for RXOUTI OR TXINI

if RXOUTI, then clear bit and return

if TXINI, then continue

Once the OUT status stage has been received, the USBHS waits for a SETUP request. The SETUP request has priority over any other request and has to be ACKed. This means that any other bit should be cleared and the FIFO reset when a SETUP is received.

The user has to consider that the byte counter is reset when a zero-length OUT packet is received.

Offset	Name	Bit Pos.								
0x057C	USBHS_HSTPIPICR7 (INTPIPES)	7:0	SHORTPACKETIC	RXSTALLDIC	OVERFIC	NAKEDIC		UNDERFIC	TXOUTIC	RXINIC
		15:8								
		23:16								
		31:24								
0x057C	USBHS_HSTPIPICR7 (ISOPIPES)	7:0	SHORTPACKETIC	CRCERRIC	OVERFIC	NAKEDIC		UNDERFIC	TXOUTIC	RXINIC
		15:8								
		23:16								
		31:24								
0x0580	USBHS_HSTPIPICR8	7:0	SHORTPACKETIC	RXSTALLDIC	OVERFIC	NAKEDIC		TXSTPIC	TXOUTIC	RXINIC
		15:8								
		23:16								
		31:24								
0x0580	USBHS_HSTPIPICR8 (INTPIPES)	7:0	SHORTPACKETIC	RXSTALLDIC	OVERFIC	NAKEDIC		UNDERFIC	TXOUTIC	RXINIC
		15:8								
		23:16								
		31:24								
0x0580	USBHS_HSTPIPICR8 (ISOPIPES)	7:0	SHORTPACKETIC	CRCERRIC	OVERFIC	NAKEDIC		UNDERFIC	TXOUTIC	RXINIC
		15:8								
		23:16								
		31:24								
0x0584	USBHS_HSTPIPICR9	7:0	SHORTPACKETIC	RXSTALLDIC	OVERFIC	NAKEDIC		TXSTPIC	TXOUTIC	RXINIC
		15:8								
		23:16								
		31:24								
0x0584	USBHS_HSTPIPICR9 (INTPIPES)	7:0	SHORTPACKETIC	RXSTALLDIC	OVERFIC	NAKEDIC		UNDERFIC	TXOUTIC	RXINIC
		15:8								
		23:16								
		31:24								
0x0584	USBHS_HSTPIPICR9 (ISOPIPES)	7:0	SHORTPACKETIC	CRCERRIC	OVERFIC	NAKEDIC		UNDERFIC	TXOUTIC	RXINIC
		15:8								
		23:16								
		31:24								
0x0588 ... 0x058F	Reserved									
0x0590	USBHS_HSTPIPIFRx	7:0	SHORTPACKETIS	RXSTALLDIS	OVERFIS	NAKEDIS	PERRIS	TXSTPIS	TXOUTIS	RXINIS
		15:8				NBUSYBKS				
		23:16								

Note: 1. bcr means broadcast command with response.

Table 40-5. Fields and Values for HSMCI_CMDR

Field	Value
CMDNB (command number)	2 (CMD2)
RSPTYP (response type)	2 (R2: 136 bits response)
SPCMD (special command)	0 (not a special command)
OPCMD (open drain command)	1
MAXLAT (max latency for command to response)	0 (NID cycles ==> 5 cycles)
TRCMD (transfer command)	0 (No transfer)
TRDIR (transfer direction)	X (available only in transfer command)
TRTYP (transfer type)	X (available only in transfer command)
IOSPCMD (SDIO special command)	0 (not a special command)

The HSMCI_ARGR contains the argument field of the command.

To send a command, the user must perform the following steps:

1. Fill the argument register (HSMCI_ARGR) with the command argument.
2. Set the command register (HSMCI_CMDR).

The command is sent immediately after writing the command register.

While the card maintains a busy indication (at the end of a STOP_TRANSMISSION command CMD12, for example), a new command shall not be sent. The NOTBUSY flag in the Status Register (HSMCI_SR) is asserted when the card releases the busy indication.

If the command requires a response, it can be read in the HSMCI Response Register (HSMCI_RSPR). The response size can be from 48 bits up to 136 bits depending on the command. The HSMCI embeds an error detection to prevent any corrupted data during the transfer.

The following flowchart shows how to send a command to the card and read the response if needed. In this example, the status register bits are polled but setting the appropriate bits in the HSMCI Interrupt Enable Register (HSMCI_IER) allows using an interrupt method.

41.7.3.2 Master Mode Flow Diagram

Figure 41-7. Master Mode Flow Diagram

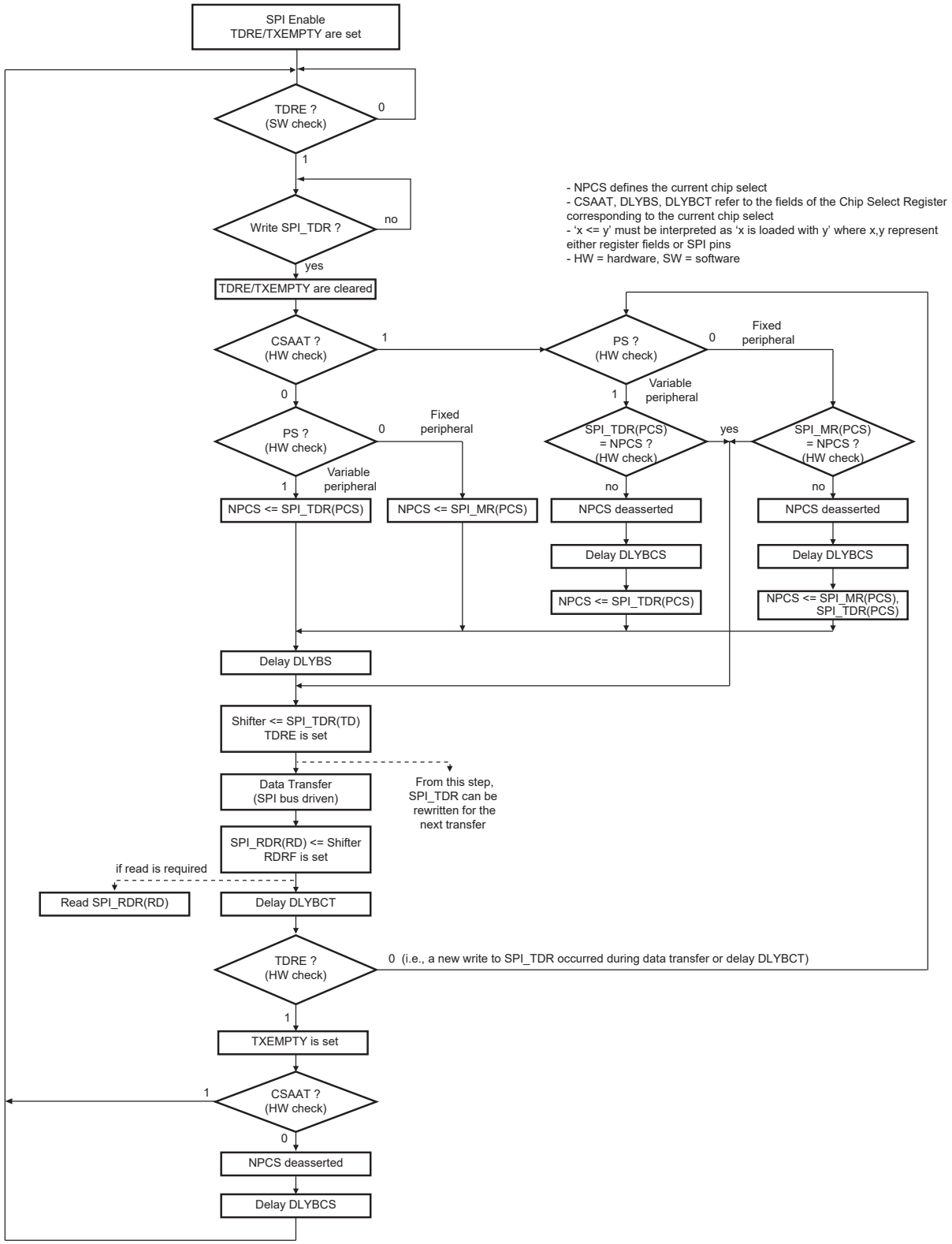
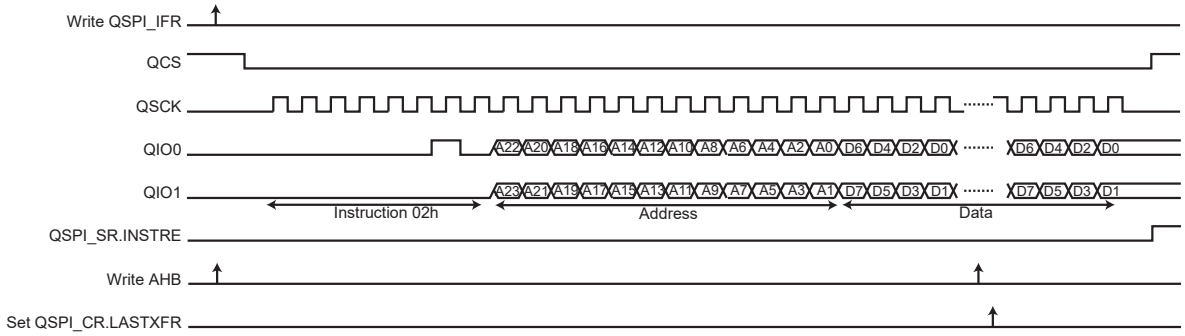


Figure 42-15. Instruction Transmission Waveform 5



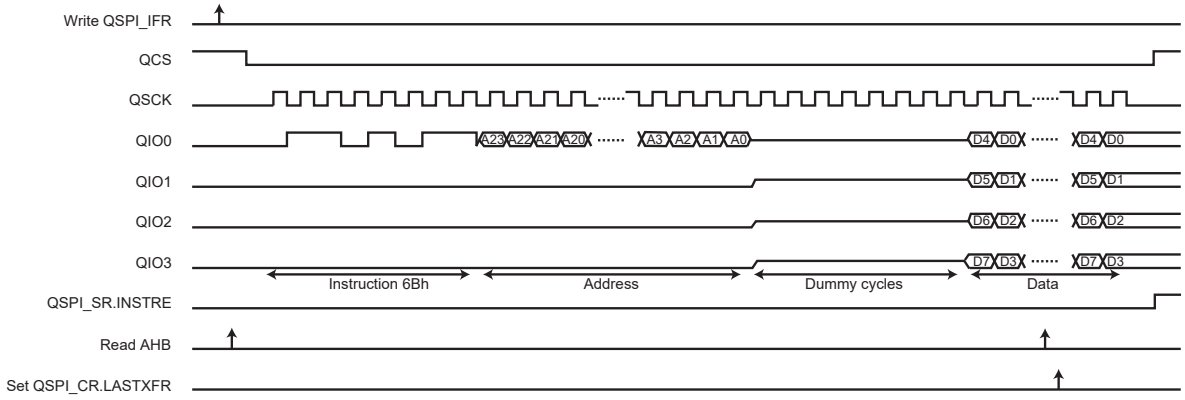
Example 6:

Instruction in Single-bit SPI, with address in Single-bit SPI, without option, with data read in Quad SPI, with eight dummy cycles.

Command: QUAD_OUTPUT READ ARRAY (6Bh)

- Write 0x0000_006B in QSPI_ICR.
- Write 0x0008_10B2 in QSPI_IFR.
- Read QSPI_IR (dummy read) to synchronize system bus accesses.
- Read data in the QSPI system bus memory space (0x80000000).
  The address of the first system bus read access is sent in the instruction frame.
  The address of the next system bus read accesses is not used.
- Write a ‘1’ to QSPI_CR.LASTXFR.
- Wait for QSPI_SR.INSTRE to rise.

Figure 42-16. Instruction Transmission Waveform 6



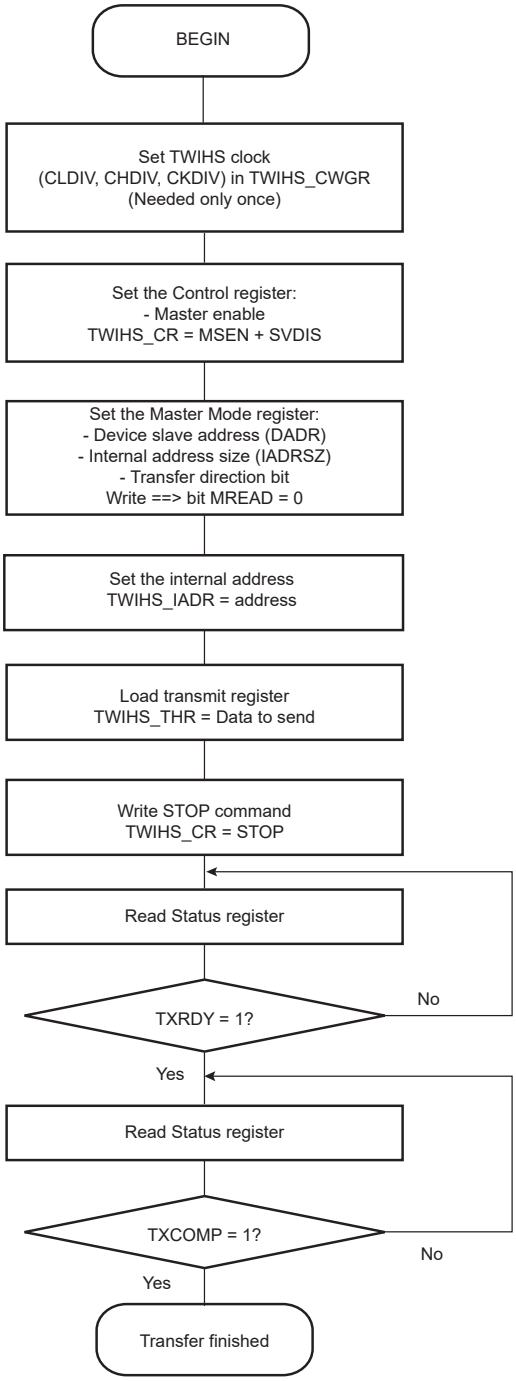
Example 7:

Instruction in Single-bit SPI, with address and option in Quad SPI, with data read in Quad SPI, with four dummy cycles, with fetch and continuous read.

Command: FAST READ QUAD I/O (EBh) - 8-BIT OPTION (0x30h)

- Write 0x0030_00EB in QSPI_ICR.
- Write 0x0004_33F4 in QSPI_IFR.
- Read QSPI_IFR (dummy read) to synchronize system bus accesses.
- Read data in the QSPI system bus memory space (0x80000000).
  Fetch is enabled, the address of the system bus read accesses is always used.

Figure 43-15. TWIHS Write Operation with Single Data Byte and Internal Address



44.9.3 SSC Receive Clock Mode Register

Name: SSC_RCMR
Offset: 0x10
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the SSC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	PERIOD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	STTDLY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
				STOP	START[3:0]			
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CKG[1:0]		CKI	CKO[2:0]			CKS[1:0]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:24 – PERIOD[7:0] Receive Period Divider Selection
This field selects the divider to apply to the selected Receive Clock in order to generate a new Frame Sync signal. If 0, no PERIOD signal is generated. If not 0, a PERIOD signal is generated each 2 x (PERIOD + 1) Receive Clock.

Bits 23:16 – STTDLY[7:0] Receive Start Delay
If STTDLY is not 0, a delay of STTDLY clock cycles is inserted between the start event and the current start of reception. When the receiver is programmed to start synchronously with the transmitter, the delay is also applied.

Note:
STTDLY must be configured in relation to the receive synchronization data to be stored in SSC_RSHR.

Bit 12 – STOP Receive Stop Selection

Value	Description
0	After completion of a data transfer when starting with a Compare 0, the receiver stops the data transfer and waits for a new compare 0.
1	After starting a receive with a Compare 0, the receiver operates in a continuous mode until a Compare 1 is detected.

Bit 18 – TOOL Timeout Occurred Interrupt Line

Bit 17 – MRAFL Message RAM Access Failure Interrupt Line

Bit 16 – TSWL Timestamp Wraparound Interrupt Line

Bit 15 – TEFLL Tx Event FIFO Event Lost Interrupt Line

Bit 14 – TEFFL Tx Event FIFO Full Interrupt Line

Bit 13 – TEFWL Tx Event FIFO Watermark Reached Interrupt Line

Bit 12 – TEFNL Tx Event FIFO New Entry Interrupt Line

Bit 11 – TFEL Tx FIFO Empty Interrupt Line

Bit 10 – TCFL Transmission Cancellation Finished Interrupt Line

Bit 9 – TCL Transmission Completed Interrupt Line

Bit 8 – HPML High Priority Message Interrupt Line

Bit 7 – RF1LL Receive FIFO 1 Message Lost Interrupt Line

Bit 6 – RF1FL Receive FIFO 1 Full Interrupt Line

Bit 5 – RF1WL Receive FIFO 1 Watermark Reached Interrupt Line

Bit 4 – RF1NL Receive FIFO 1 New Message Interrupt Line

Bit 3 – RF0LL Receive FIFO 0 Message Lost Interrupt Line

Bit 2 – RF0FL Receive FIFO 0 Full Interrupt Line

Bit 1 – RF0WL Receive FIFO 0 Watermark Reached Interrupt Line

Bit 0 – RF0NL Receive FIFO 0 New Message Interrupt Line

51.7.12 PWM Sync Channels Update Period Register

Name: PWM_SCUP
Offset: 0x2C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	UPRCNT[3:0]				UPR[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:4 – UPRCNT[3:0] Update Period Counter
Reports the value of the update period counter.

Bits 3:0 – UPR[3:0] Update Period
Defines the time between each update of the synchronous channels if automatic trigger of the update is activated (UPDM = 1 or UPDM = 2 in PWM Sync Channels Mode Register). This time is equal to UPR+1 periods of the synchronous channels.

55.6.1 ICM Configuration Register

Name: ICM_CFG
Offset: 0x00
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	UALGO[2:0]			UIHASH			DUALBUFF	ASCD
Access	R/W	R/W	R/W	R/W			R/W	R/W
Reset	0	0	0	0			0	0

Bit	7	6	5	4	3	2	1	0
	BBC[3:0]					SLBDIS	EOMDIS	WBDIS
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	0	0		0	0	0

Bits 15:13 – UALGO[2:0] User SHA Algorithm

Value	Name	Description
0	SHA1	SHA1 algorithm processed
1	SHA256	SHA256 algorithm processed
4	SHA224	SHA224 algorithm processed

Bit 12 – UIHASH User Initial Hash Value

Value	Description
0	The secure hash standard provides the initial hash value.
1	The initial hash value is programmable. Field UALGO provides the SHA algorithm. The ALGO field of the ICM_RCFG structure member has no effect.

Bit 9 – DUALBUFF Dual Input Buffer

Value	Description
0	Dual Input Buffer mode is disabled.
1	Dual Input Buffer mode is enabled (better performances, higher bandwidth required on system bus).

Bit 8 – ASCD Automatic Switch To Compare Digest

55.6.7 ICM Interrupt Status Register

Name: ICM_ISR
Offset: 0x1C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
								URAD
Access								R
Reset								0

Bit	23	22	21	20	19	18	17	16
	RSU[3:0]				REC[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RWC[3:0]				RBE[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RDM[3:0]				RHC[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 24 – URAD Undefined Register Access Detection Status
The URAD bit is only reset by the SWRST bit in ICM_CTRL.

The URAT field in ICM_UASR indicates the unspecified access type.

Value	Description
0	No undefined register access has been detected since the last SWRST.
1	At least one undefined register access has been detected since the last SWRST.

Bits 23:20 – RSU[3:0] Region Status Updated Detected
When RSU[i] is set, it indicates that a region status updated condition has been detected.

Bits 19:16 – REC[3:0] Region End Bit Condition Detected
When REC[i] is set, it indicates that an end bit condition has been detected.

Bits 15:12 – RWC[3:0] Region Wrap Condition Detected
When RWC[i] is set, it indicates that a wrap condition has been detected.

Bits 11:8 – RBE[3:0] Region Bus Error
When RBE[i] is set, it indicates that a bus error has been detected while hashing memory region i.

57.5.8 AES Input Data Register x

Name: AES_IDATARx
Offset: 0x40 + x*0x04 [x=0..3]
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	IDATA[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	IDATA[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	IDATA[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	IDATA[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	–

Bits 31:0 – IDATA[31:0] Input Data Word
The four 32-bit Input Data registers set the 128-bit data block used for encryption/decryption.

AES_IDATAR0 corresponds to the first word of the data to be encrypted/decrypted, and AES_IDATAR3 to the last one.

These registers are write-only to prevent the input data from being read by another application.