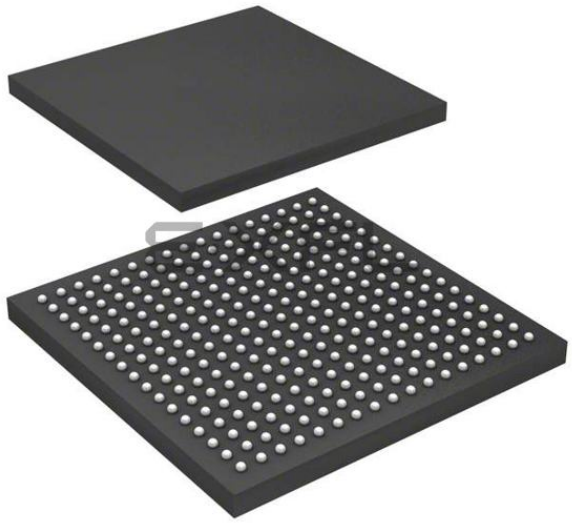


Welcome to E-XFL.COM

Understanding Embedded - Microprocessors

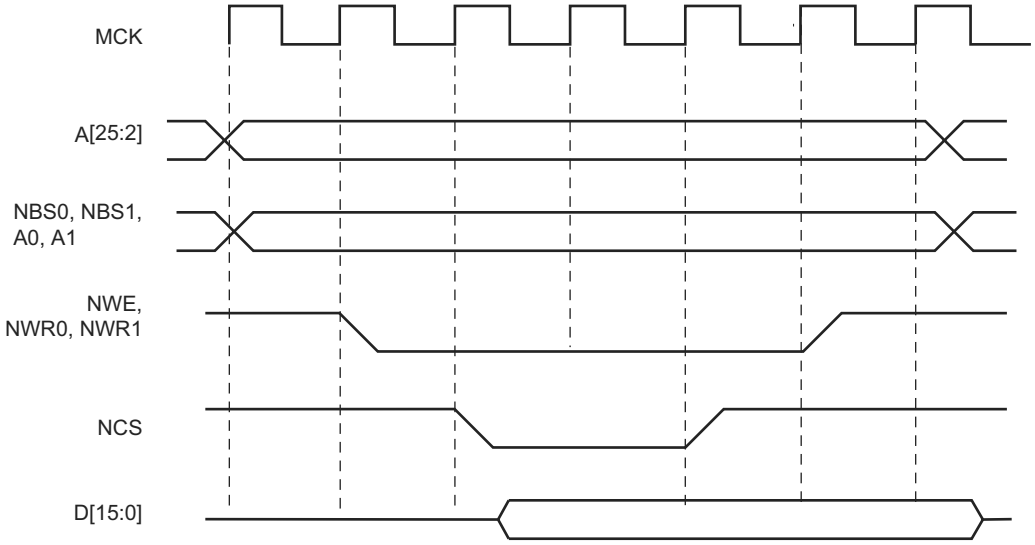
Embedded microprocessors are specialized computing chips designed to perform specific tasks within an embedded system. Unlike general-purpose microprocessors found in personal computers, embedded microprocessors are tailored for dedicated functions within larger systems, offering optimized performance, 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	Obsolete
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	289-LFBGA
Supplier Device Package	289-LFBGA (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsama5d41a-cur

Figure 30-12: WRITE_MODE = 0. The write operation is controlled by NCS



30.10.5 Coding Timing Parameters

All timing parameters are defined for one chip select and are grouped together in one register according to their type:

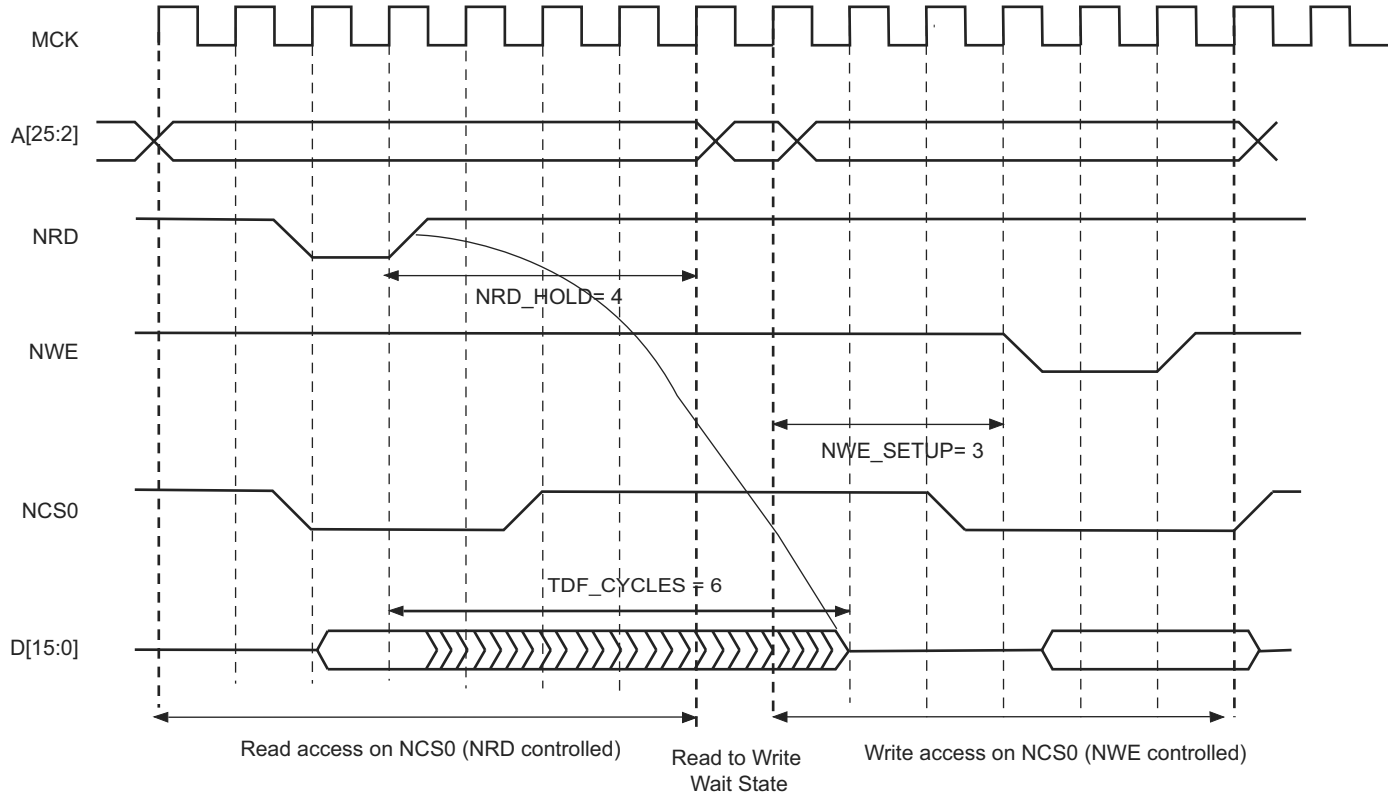
- The HSMC_SETUP register groups the definition of all setup parameters:
  NRD_SETUP, NCS_RD_SETUP, NWE_SETUP, NCS_WR_SETUP
- The HSMC_PULSE register groups the definition of all pulse parameters:
  NRD_PULSE, NCS_RD_PULSE, NWE_PULSE, NCS_WR_PULSE
- The HSMC_CYCLE register groups the definition of all cycle parameters:
  NRD_CYCLE, NWE_CYCLE

Table 30-5 shows how the timing parameters are coded and their permitted range.

Table 30-5: Coding and Range of Timing Parameters

Coded Value	Number of Bits	Effective Value	Permitted Range	
			Coded Value	Effective Value
setup [5:0]	6	128 x setup[5] + setup[4:0]	0 ≤ setup ≤ 31	0..31
			32 ≤ setup ≤ 63	128..(128 + 31)
pulse [6:0]	7	256 x pulse[6] + pulse[5:0]	0 ≤ pulse ≤ 63	0..63
			64 ≤ pulse ≤ 127	256..(256 + 63)
cycle[8:0]	9	256 x cycle[8:7] + cycle[6:0]	0 ≤ cycle ≤ 127	0..127
			128 ≤ cycle ≤ 255	256..(256 + 127)
			256 ≤ cycle ≤ 383	512..(512 + 127)
			384 ≤ cycle ≤ 511	768..(768 + 127)

Figure 30-19: TDF Optimization: No TDF wait states are inserted if the TDF period is over when the next access begins



30.13.3 TDF Optimization Disabled (TDF_MODE = 0)

When optimization is disabled, TDF wait states are inserted at the end of the read transfer, so that the data float period ends when the second access begins. If the hold period of the read1 controlling signal overlaps the data float period, no additional TDF wait states will be inserted.

Figure 30-20, Figure 30-21 and Figure 30-22 illustrate the cases:

- read access followed by a read access on another chip select,
- read access followed by a write access on another chip select,
- read access followed by a write access on the same chip select,

with no TDF optimization.

30.20.12 PMECC End Address Register

Name: HSMC_PMECCEADDR

Address:0xFC05C07C

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
–	–	–	–	–	–	–	ENDADDR
7	6	5	4	3	2	1	0
ENDADDR							

ENDADDR: ECC Area End Address

This register is programmed with the start ECC end address. When ENDADDR is equal to *N*, then the first ECC byte is located at byte *N* of the spare area.

32.7.50 Overlay 1 Configuration Register 5

Name: LCDC_OVR1CFG5

Address:0xF0000180

Access: Read/Write

31	30	29	28	27	26	25	24
PSTRIDE							

23	22	21	20	19	18	17	16
PSTRIDE							

15	14	13	12	11	10	9	8
PSTRIDE							

7	6	5	4	3	2	1	0
PSTRIDE							

PSTRIDE: Pixel Stride

PSTRIDE represents the memory offset, in bytes, between two pixels of the image.

32.7.52 Overlay 1 Configuration Register 7

Name: LCDC_OVR1CFG7

Address:0xF0000188

Access: Read/Write

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

23	22	21	20	19	18	17	16
RKEY							

15	14	13	12	11	10	9	8
GKEY							

7	6	5	4	3	2	1	0
BKEY							

RKEY: Red Color Component Chroma Key

Reference Red chroma key used to match the Red color of the current overlay.

GKEY: Green Color Component Chroma Key

Reference Green chroma key used to match the Green color of the current overlay.

BKEY: Blue Color Component Chroma Key

Reference Blue chroma key used to match the Blue color of the current overlay.

32.7.93 High End Overlay V DMA Control Register

Name: LCDC_HEOVCTRL

Address:0xF0000384

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
–	–	VDONEIEN	VADDIEN	VDSCRIEN	VDMAIEN	–	VDFETCH

VDFETCH: Transfer Descriptor Fetch Enable

0: Transfer Descriptor fetch is disabled.

1: Transfer Descriptor fetch is enabled.

VDMAIEN: End of DMA Transfer Interrupt Enable

0: DMA transfer completed interrupt is enabled.

1: DMA transfer completed interrupt is disabled.

VDSCRIEN: Descriptor Loaded Interrupt Enable

0: Transfer descriptor loaded interrupt is enabled.

1: Transfer descriptor loaded interrupt is disabled.

VADDIEN: Add Head Descriptor to Queue Interrupt Enable

0: Transfer descriptor added to queue interrupt is enabled.

1: Transfer descriptor added to queue interrupt is enabled.

VDONEIEN: End of List Interrupt Enable

0: End of list interrupt is disabled.

1: End of list interrupt is enabled.

32.7.127 High End Overlay Configuration Register 32

Name: LCDC_HEOCFG32

Address:0xF000040C

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
XPHI7COEFF4							

XPHI7COEFF4: Horizontal Coefficient for phase 7 tap 4

Coefficient format is 1 sign bit and 7 fractional bits.

THMASK: Threshold Mask

Value	Name	Description
0	BEATS_4	Only 4 beats AHB burst allowed
1	BEATS_8	Only 4 and 8 beats AHB burst allowed
2	BEATS_16	4, 8 and 16 beats AHB burst allowed

SLD: Start of Line Delay

SLD pixel clock periods to wait before the beginning of a line.

SFD: Start of Frame Delay

SFD lines are skipped at the beginning of the frame.

35.7.15 UDPHS Endpoint Set Status Register (Control, Bulk, Interrupt Endpoints)

Name: UDPHS_EPTSETSTAx [x=0..15]

Address:0xFC02C114 [0], 0xFC02C134 [1], 0xFC02C154 [2], 0xFC02C174 [3], 0xFC02C194 [4], 0xFC02C1B4 [5], 0xFC02C1D4 [6], 0xFC02C1F4 [7], 0xFC02C214 [8], 0xFC02C234 [9], 0xFC02C254 [10], 0xFC02C274 [11], 0xFC02C294 [12], 0xFC02C2B4 [13], 0xFC02C2D4 [14], 0xFC02C2F4 [15]

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
–	–	–	–	TXRDY	–	RXRDY_TXKL	–
7	6	5	4	3	2	1	0
–	–	FRCESTALL	–	–	–	–	–

This register view is relevant only if EPT_TYPE = 0x0, 0x2 or 0x3 in “UDPHS Endpoint Configuration Register”.

For additional information, refer to “UDPHS Endpoint Status Register (Control, Bulk, Interrupt Endpoints)”.

FRCESTALL: Stall Handshake Request Set

0: No effect.

1: Set this bit to request a STALL answer to the host for the next handshake

Refer to chapters 8.4.5 (Handshake Packets) and 9.4.5 (Get Status) of the *Universal Serial Bus Specification, Rev 2.0* for more information on the STALL handshake.

RXRDY_TXKL: KILL Bank Set (for IN Endpoint)

0: No effect.

1: Kill the last written bank.

TXRDY: TX Packet Ready Set

0: No effect.

1: Set this bit after a packet has been written into the endpoint FIFO for IN data transfers

- This flag is used to generate a Data IN transaction (device to host).
- Device firmware checks that it can write a data payload in the FIFO, checking that TXRDY is cleared.
- Transfer to the FIFO is done by writing in the “Buffer Address” register.
- Once the data payload has been transferred to the FIFO, the firmware notifies the UDPHS device setting TXRDY to one.
- UDPHS bus transactions can start.
- TXCOMP is set once the data payload has been received by the host.
- Data should be written into the endpoint FIFO only after this bit has been cleared.
- Set this bit without writing data to the endpoint FIFO to send a Zero Length Packet.

36.7 USB Host High Speed Port (UHPHS) User Interface

The Enhanced USB Host Controller contains two sets of software-accessible hardware registers: memory-mapped Host Controller Registers and optional PCI configuration registers. Note that the PCI configuration registers are only needed for PCI devices that implement the Host Controller.

- Memory-mapped USB Host Controller Registers—This block of registers is memory-mapped into non-cacheable memory. This memory space must begin on a DWord (32-bit) boundary. This register space is divided into two sections: a set of read-only capability registers and a set of read/write operational registers. Table 36-1 describes each register space.

  Note: Host controllers are not required to support exclusive-access mechanisms (such as PCI LOCK) for accesses to the memory-mapped register space. Therefore, if software attempts exclusive-access mechanisms to the host controller memory-mapped register space, the results are undefined.

- PCI Configuration Registers (for PCI devices)—In addition to the normal PCI header, power management, and device-specific registers, two registers are needed in the PCI configuration space to support USB. The normal PCI header and device-specific registers are beyond the scope of this document (the UHPHS_CLASSC register is shown in this document). Note that HCD does not interact with the PCI configuration space. This space is used only by the PCI enumerator to identify the USB Host Controller, and assign the appropriate system resources.

Table 36-1: Enhanced Interface Register Sets

Offset	Register Set	Explanation
0 to N-1	Capability Registers	The capability registers specify the limits, restrictions, and capabilities of a host controller implementation. These values are used as parameters to the host controller driver.
N to N+M-1	Operational Registers	The operational registers are used by system software to control and monitor the operational state of the host controller.

Table 36-2: Register Mapping

Offset	Register	Name	Access	Reset
Host Controller Capability Registers				
0x00	UHPHS Host Controller Capability Register	UHPHS_HCCAPBASE	Read-only	0x0100 0010
0x04	UHPHS Host Controller Structural Parameters Register	UHPHS_HCSPARAMS	Read-only	0x0000 1116
0x08	UHPHS Host Controller Capability Parameters Register	UHPHS_HCCPARAMS	Read-only	0x0000 A010
0x0C	Reserved	–	–	–
Host Controller Operational Registers				
0x10	UHPHS USB Command Register	UHPHS_USBCMD	Read/ Write(1)	0x0008 0000 or 0x0008 0B00(2)
0x14	UHPHS USB Status Register	UHPHS_USBSTS	Read/ Write(1)	0x0000 1000
0x18	UHPHS USB Interrupt Enable Register	UHPHS_USBINTR	Read/Write	0x0000 0000
0x1C	UHPHS USB Frame Index Register	UHPHS_FRINDEX	Read/Write	0x0000 0000
0x20	UHPHS Control Data Structure Segment Register	UHPHS_CTRLDSSEGMENT	Read/Write	0x0000 0000
0x24	UHPHS Periodic Frame List Base Address Register	UHPHS_PERIODICLISTBASE	Read/Write	0x0000 0000

37.8.61 GMAC Excessive Collisions Register

Name:GMAC_EC

Address:0xF8020140 (0), 0xFC028140 (1)

Access: Read-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
–	–	–	–	–	–	XCOL	

7	6	5	4	3	2	1	0
XCOL							

XCOL: Excessive Collisions

This register counts the number of frames that failed to be transmitted because they experienced 16 collisions.

37.8.104 GMAC Received LPI Transitions

Name:GMAC_RXLPI

Address:0xF8020270 (0), 0xFC028270 (1)

Access: Read-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
COUNT							

7	6	5	4	3	2	1	0
COUNT							

COUNT: Count of RX LPI transitions (cleared on read)

A count of the number of times there is a transition from receiving normal idle to receiving low power idle.

39.7.3.10 Mode Fault Detection

The SPI has the capability to operate in multimaster environment. Consequently, the NPCS0/NSS line must be monitored. If one of the masters on the SPI bus is currently transmitting, the NPCS0/NSS line is low and the SPI must not transmit any data. A mode fault is detected when the SPI is programmed in Master mode and a low level is driven by an external master on the NPCS0/NSS signal. In multimaster environment, NPCS0, MOSI, MISO and SPCK pins must be configured in open drain (through the PIO controller). When a mode fault is detected, the SPI_SR.MODF bit is set until SPI_SR is read and the SPI is automatically disabled until it is reenabled by setting the SPI_CR.SPIEN bit.

By default, the mode fault detection is enabled. The user can disable it by setting the SPI_MR.MODFDIS bit.

39.7.4 SPI Slave Mode

When operating in Slave mode, the SPI processes data bits on the clock provided on the SPI clock pin (SPCK).

The SPI waits until NSS goes active before receiving the serial clock from an external master. When NSS falls, the clock is validated and the data is loaded in the SPI_RDR depending on the BITS field configured in SPI_CSR0. These bits are processed following a phase and a polarity defined respectively by the NCPHA and CPOL bits in SPI_CSR0. Note that the fields BITS, CPOL and NCPHA of the other chip select registers (SPI_CSR1...SPI_CSR3) have no effect when the SPI is programmed in Slave mode.

The bits are shifted out on the MISO line and sampled on the MOSI line.

Note: For more information on the BITS field, refer to the note below the SPI_CSRx bitmap (Section 39.8.9 “SPI Chip Select Register”).

When all bits are processed, the received data is transferred in the SPI_RDR and the RDRF bit rises. If the SPI_RDR has not been read before new data is received, the Overrun Error Status (OVRES) bit in the SPI_SR is set. As long as this flag is set, data is loaded in the SPI_RDR. The user must read SPI_SR to clear the OVRES bit.

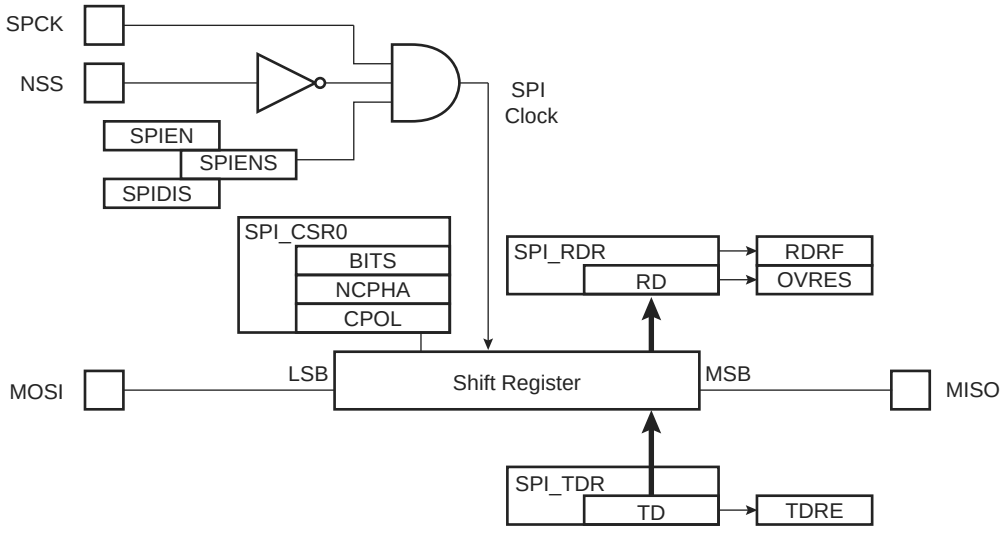
When a transfer starts, the data shifted out is the data present in the Shift register. If no data has been written in the SPI_TDR, the last data received is transferred. If no data has been received since the last reset, all bits are transmitted low, as the Shift register resets to 0.

When a first data is written in the SPI_TDR, it is transferred immediately in the Shift register and the TDRE flag rises. If new data is written, it remains in the SPI_TDR until a transfer occurs, i.e., NSS falls and there is a valid clock on the SPCK pin. When the transfer occurs, the last data written in the SPI_TDR is transferred in the Shift register and the TDRE flag rises. This enables frequent updates of critical variables with single transfers.

Then, new data is loaded in the Shift register from the SPI_TDR. If no character is ready to be transmitted, i.e., no character has been written in the SPI_TDR since the last load from the SPI_TDR to the Shift register, the SPI_TDR is retransmitted. In this case the Underrun Error Status Flag (UNDES) is set in the SPI_SR.

Figure 39-12 shows a block diagram of the SPI when operating in Slave mode.

Figure 39-12: Slave Mode Functional Block Diagram



46.6.14 Output Controller

The output controller defines the output level changes on TIOAx and TIOBx following an event. TIOBx Control is used only if TIOBx is defined as output (not as an external event).

The following events control TIOAx and TIOBx:

- Software Trigger
- External Event
- RC Compare

RA Compare controls TIOAx, and RB Compare controls TIOBx. Each of these events can be programmed to set, clear or toggle the output as defined in the corresponding parameter in TC_CMR.

46.6.15 Quadrature Decoder

46.6.15.1 Description

The quadrature decoder (QDEC) is driven by TIOA0, TIOB0, TIOB1 input pins and drives the timer/counter of channel 0 and 1. Channel 2 can be used as a time base in case of speed measurement requirements (refer to Figure 46-16).

When writing a 0 to bit QDEN of the TC_BMR, the QDEC is bypassed and the IO pins are directly routed to the timer counter function.

TIOA0 and TIOB0 are to be driven by the two dedicated quadrature signals from a rotary sensor mounted on the shaft of the off-chip motor.

A third signal from the rotary sensor can be processed through pin TIOB1 and is typically dedicated to be driven by an index signal if it is provided by the sensor. This signal is not required to decode the quadrature signals PHA, PHB.

Field TCCLKS of TC_CMRx must be configured to select XC0 input (i.e., 0x101). Field TC0XC0S has no effect as soon as the QDEC is enabled.

Either speed or position/revolution can be measured. Position channel 0 accumulates the edges of PHA, PHB input signals giving a high accuracy on motor position whereas channel 1 accumulates the index pulses of the sensor, therefore the number of rotations. Concatenation of both values provides a high level of precision on motion system position.

In Speed mode, position cannot be measured but revolution can be measured.

Inputs from the rotary sensor can be filtered prior to down-stream processing. Accommodation of input polarity, phase definition and other factors are configurable.

Interruptions can be generated on different events.

A compare function (using TC_RC) is available on channel 0 (speed/position) or channel 1 (rotation) and can generate an interrupt by means of the CPCS flag in the TC_SRx.

47.7.5 PWM Interrupt Enable Register 1

Name:PWM_IER1

Address:0xF800C010

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	FCHID3	FCHID2	FCHID1	FCHID0
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	CHID3	CHID2	CHID1	CHID0

CHIDx: Counter Event on Channel x Interrupt Enable

FCHIDx: Fault Protection Trigger on Channel x Interrupt Enable

47.7.13 PWM Interrupt Enable Register 2

Name:PWM_IER2

Address:0xF800C034

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
CMPU7	CMPU6	CMPU5	CMPU4	CMPU3	CMPU2	CMPU1	CMPU0
15	14	13	12	11	10	9	8
CMPM7	CMPM6	CMPM5	CMPM4	CMPM3	CMPM2	CMPM1	CMPM0
7	6	5	4	3	2	1	0
–	–	–	–	UNRE			WRDY

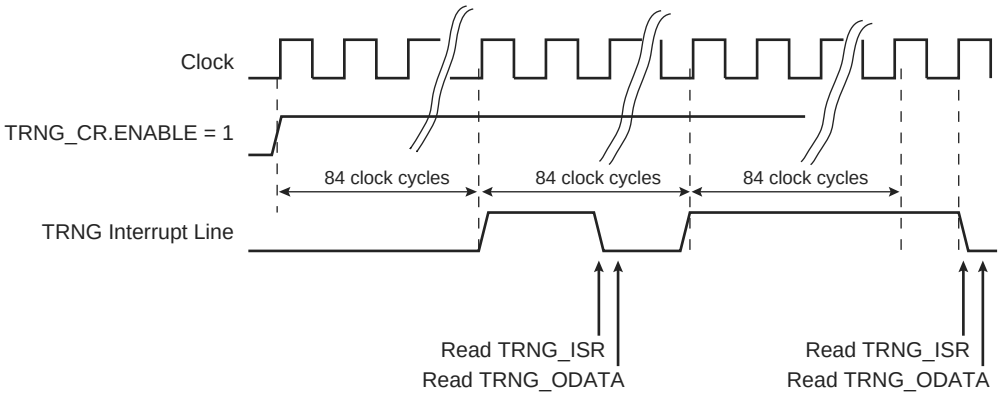
WRDY: Write Ready for Synchronous Channels Update Interrupt Enable

UNRE: Synchronous Channels Update Underrun Error Interrupt Enable

CMPMx: Comparison x Match Interrupt Enable

CMPUx: Comparison x Update Interrupt Enable

Figure 49-2: TRNG Data Generation Sequence



KEYSIZE: Key Size

Value	Name	Description
0	AES128	AES Key Size is 128 bits
1	AES192	AES Key Size is 192 bits
2	AES256	AES Key Size is 256 bits

OPMOD: Operating Mode

Value	Name	Description
0	ECB	ECB: Electronic Codebook mode
1	CBC	CBC: Cipher Block Chaining mode
2	OFB	OFB: Output Feedback mode
3	CFB	CFB: Cipher Feedback mode
4	CTR	CTR: Counter mode (16-bit internal counter)
5	GCM	GCM: Galois/Counter mode

For CBC-MAC operating mode, set OPMOD to CBC and LOD to 1.

LOD: Last Output Data Mode

0: No effect.

After each end of encryption/decryption, the output data are available either on the output data registers (Manual and Auto modes) or at the address specified in the Channel Buffer Transfer Descriptor for DMA mode.

In Manual and Auto modes, the DATRDY flag is cleared when at least one of the Output Data registers is read.

1: The DATRDY flag is cleared when at least one of the Input Data Registers is written.

No more Output Data Register reads is necessary between consecutive encryptions/decryptions (refer to Section 50.4.5 “Last Output Data Mode”).

Warning: In DMA mode, reading to the Output Data registers before the last data encryption/decryption process may lead to unpredictable results.

CFBS: Cipher Feedback Data Size

Value	Name	Description
0	SIZE_128BIT	128-bit
1	SIZE_64BIT	64-bit
2	SIZE_32BIT	32-bit
3	SIZE_16BIT	16-bit
4	SIZE_8BIT	8-bit

CKEY: Key

Value	Name	Description
0xE	PASSWD	This field must be written with 0xE the first time AES_MR is programmed. For subsequent programming of AES_MR, any value can be written, including that of 0xE. Always reads as 0.

51.5.1 TDES Control Register

Name: TDES_CR

Address:0xFC04C000

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
–	–	–	–	–	–	–	SWRST
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	START

- **START: Start Processing**

0: No effect

1: Starts Manual encryption/decryption process.

SWRST: Software Reset

0: No effect

1: Resets the TDES. A software triggered hardware reset of the TDES interface is performed.

5. Generate a STOP condition.
6. Reconfigure SDA/SCL PIOs as peripheral.

61.2.9 Serial Synchronous Controller (SSC)

61.2.9.1 Inverted Left/Right Channels

When the SSC is in Slave mode, the TF signal is derived from the codec and not controlled by the SSC. The SSC transmits the data when detecting the falling edge on the TF signal after the SSC transmission is enabled. In some

cases of overflow, a left/right channel inversion may occur. In this case, the SSC must be re-initialized.

Problem Fix/Workaround

Using the SSC in Master mode will ensure that TF is controlled by the SSC. No error occurs. If the SSC must be used in TF Slave mode, the SSC must be started by writing TXEN and RXEN synchronously with TXSYN flag rising in the SSC_SR.

61.2.9.2 Unexpected Delay on TD Output

When SSC is configured with the following conditions:

- RCMR.START = Start on falling edge/Start on Rising edge/Start on any edge
- RFMR.FSOS = None (input)
- TCMR.START = Receive Start

an unexpected delay of 2 or 3 system clock cycles is added to TD output.

Problem Fix/Workaround

None.

61.2.10 Universal Synchronous Asynchronous Receiver Transceiver (USART)

61.2.10.1 USART Framing error not detected if last data bit is 1

If a bad frame is received (incorrect baud rate) with the last data bit being sampled at 1, there is no detection of frame error.

Problem Fix/Workaround

There is no general fix. When performing baud rate detection with receive part, the transmit frame must be sent with a parity bit set to 0.

61.2.11 Ethernet MAC (GMAC)

61.2.11.1 Bad Association of Timestamps and PTP packets

An issue in the association mechanism between event registers and queued PTP packets may lead to timestamps incorrectly associated with these packets.

Even if it is highly unlikely to queue consecutive packets of the same type, there is no way to know to which frame the content of the PTP event registers refers.

Problem Fix/Workaround

None.