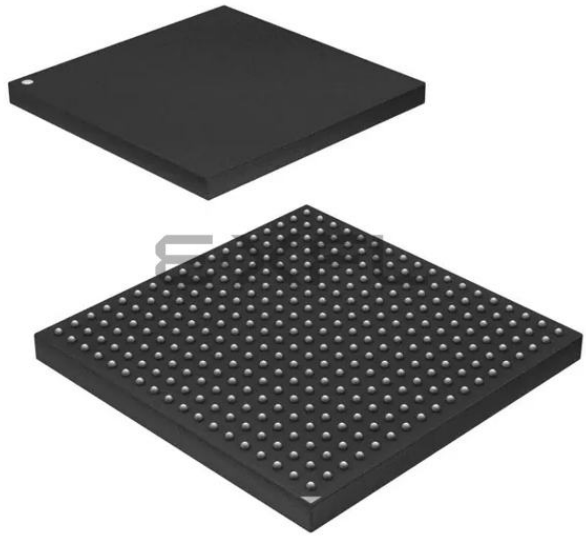


Welcome to E-XFL.COM

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	536MHz
Co-Processors/DSP	-
RAM Controllers	LPDDR, LPDDR2, DDR2
Graphics Acceleration	No
Display & Interface Controllers	Touchscreen
Ethernet	10/100/1000Mbps (1)
SATA	-
USB	USB 2.0 (3)
Voltage - I/O	1.2V, 1.8V, 3.3V
Operating Temperature	-40°C ~ 105°C (TA)
Security Features	AES, SHA, TDES, TRNG
Package / Case	324-LFBGA
Supplier Device Package	324-LFBGA (15x15)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsama5d35a-cnr

Table 4-2. SAMA5D3 Pinout for 324-ball TFBGA Package (Continued)

			Primary		Alternate		PIO Peripheral A		PIO Peripheral B		PIO Peripheral C		Reset State
Pin	Power Rail	I/O Type	Signal	Dir	Signal	Dir	Signal	Dir	Signal	Dir	Signal	Dir	Signal, Dir, PU, PD, HiZ, ST
W17	VDDIOM	EBI	PE0	I/O	—	—	A0/NBS0	O	—	—	—	—	A,I, PD, ST
Y18	VDDIOM	EBI	PE1	I/O	—	—	A1	O	—	—	—	—	A,I, PD, ST
W18	VDDIOM	EBI	PE2	I/O	—	—	A2	O	—	—	—	—	A,I, PD, ST
AA21	VDDIOM	EBI	PE3	I/O	—	—	A3	O	—	—	—	—	A,I, PD, ST
Y16	VDDIOM	EBI	PE4	I/O	—	—	A4	O	—	—	—	—	A,I, PD, ST
Y20	VDDIOM	EBI	PE5	I/O	—	—	A5	O	—	—	—	—	A,I, PD, ST
W19	VDDIOM	EBI	PE6	I/O	—	—	A6	O	—	—	—	—	A,I, PD, ST
Y22	VDDIOM	EBI	PE7	I/O	—	—	A7	O	—	—	—	—	A,I, PD, ST
Y21	VDDIOM	EBI	PE8	I/O	—	—	A8	O	—	—	—	—	A,I, PD, ST
W22	VDDIOM	EBI	PE9	I/O	—	—	A9	O	—	—	—	—	A,I, PD, ST
V19	VDDIOM	EBI	PE10	I/O	—	—	A10	O	—	—	—	—	A,I, PD, ST
W20	VDDIOM	EBI	PE11	I/O	—	—	A11	O	—	—	—	—	A,I, PD, ST
W21	VDDIOM	EBI	PE12	I/O	—	—	A12	O	—	—	—	—	A,I, PD, ST
T19	VDDIOM	EBI	PE13	I/O	—	—	A13	O	—	—	—	—	A,I, PD, ST
V22	VDDIOM	EBI	PE14	I/O	—	—	A14	O	—	—	—	—	A,I, PD, ST
V20	VDDIOM	EBI	PE15	I/O	—	—	A15	O	SCK3	I/O	—	—	A,I, PD, ST
V21	VDDIOM	EBI	PE16	I/O	—	—	A16	O	CTS3	I	—	—	A,I, PD, ST
T20	VDDIOM	EBI	PE17	I/O	—	—	A17	O	RTS3	O	—	—	A,I, PD, ST
U20	VDDIOM	EBI	PE18	I/O	—	—	A18	O	RXD3	I	—	—	A,I, PD, ST
U21	VDDIOM	EBI	PE19	I/O	—	—	A19	O	TXD3	O	—	—	A,I, PD, ST
U22	VDDIOM	EBI	PE20	I/O	—	—	A20	O	SCK2	I/O	—	—	A,I, PD, ST
R19	VDDIOM	EBI	PE21	I/O	—	—	A21/NANDALE	O	—	—	—	—	A,I, PD, ST
R20	VDDIOM	EBI	PE22	I/O	—	—	A22/NANDCLE	O	—	—	—	—	A,I, PD, ST
T21	VDDIOM	EBI	PE23	I/O	—	—	A23	O	CTS2	I	—	—	A,I, PD, ST
T22	VDDIOM	EBI	PE24	I/O	—	—	A24	O	RTS2	O	—	—	A,I, PD, ST
P19	VDDIOM	EBI	PE25	I/O	—	—	A25	O	RXD2	I	—	—	A,I, PD, ST
R22	VDDIOM	EBI	PE26	I/O	—	—	NCS0	O	TXD2	O	—	—	A,I, PD, ST
R21	VDDIOM	EBI	PE27	I/O	—	—	NCS1	O	TIOA2	I/O	LCDDAT22	O	PIO,I, PD, ST
P20	VDDIOM	EBI	PE28	I/O	—	—	NCS2	O	TIOB2	I/O	LCDDAT23	O	PIO, I, PD, ST
P21	VDDIOM	EBI	PE29	I/O	—	—	NWR1/NBS1	O	TCLK2	I	—	—	PIO, I, PD, ST
N19	VDDIOM	EBI	PE30	I/O	—	—	NWAIT	I	—	—	—	—	PIO, I, PD, ST
N21	VDDIOM	EBI	PE31	I/O	—	—	IRQ	I	PWML1	O	—	—	PIO,I, PD, ST
Y15	VDDBU	SYSC	TST	I	—	—	—	—	—	—	—	—	I, PD,
AB14	VDDIOP0	SYSC	BMS	I	—	—			—	—	—	—	I
AB11	VDDIOP0	CLOCK	XIN	I	—	—	—	—	—	—	—	—	I
AA11	VDDIOP0	CLOCK	XOUT	O	—	—			—	—	—	—	O
AB19	VDDBU	CLOCK	XIN32	I	—	—	—	—	—	—	—	—	I
AA19	VDDBU	CLOCK	XOUT32	O	—	—			—	—	—	—	O
W16	VDDBU	SYSC	SHDN	O	—	—	—	—	—	—	—	—	O
AB16	VDDBU	SYSC	WKUP	I	—	—			—	—	—	—	I, ST
Y13	VDDIOP0	RSTJTAG	NRST	I/O	—	—	—	—	—	—	—	—	I, PU, ST
AA14	VDDIOP0	RSTJTAG	NTRST	I	—	—	—	—	—	—	—	—	I, PU, ST
W13	VDDIOP0	RSTJTAG	TDI	I	—	—	—	—	—	—	—	—	I, ST

8.1 Chip Identification

- Chip ID: 0x8A5C07C2
- Extended ID:

Table 8-1. Chip Identification of SAMA5D3 Devices

Device	Extended ID
SAMA5D31	**0x00444300**
SAMA5D33	**0x00414300**
SAMA5D34	**0x00414301**
SAMA5D35	**0x00584300**
SAMA5D36	**0x00004301**

- Boundary JTAG ID: 0x05B3103F
- Cortex-A5 JTAG IDCODE: 0x4BA00477
- Cortex-A5 Serial Wire IDCODE: 0x2BA01477

8.2 Backup Section

The SAMA5D3 features a Backup Section that embeds:

- RC Oscillator
- Slow Clock Oscillator
- SCKR register
- Real-time Clock (RTC)
- Shutdown Controller
- 4 Backup registers
- Part of the Reset Controller (RSTC)
- Boot Select Control Register

This section is powered by the VDDBU rail.

17.6 I/O Line Description

Table 17-1. I/O Line Description

Pin Name	Pin Description	Type
FIQ	Fast Interrupt	Input
IRQ0 - IRQn	Interrupt 0 - Interrupt n	Input

17.7 Product Dependencies

17.7.1 I/O Lines

The interrupt signals FIQ and IRQ0 to IRQn are normally multiplexed through the PIO controllers. Depending on the features of the PIO controller used in the product, the pins must be programmed in accordance with their assigned interrupt function. This is not applicable when the PIO controller used in the product is transparent on the input path.

Table 17-2. I/O Lines

Instance	Signal	I/O Line	Peripheral
AIC	FIQ	PC31	A
AIC	IRQ	PE31	A

17.7.2 Power Management

The Advanced Interrupt Controller is continuously clocked. The Power Management Controller has no effect on the Advanced Interrupt Controller behavior.

The assertion of the Advanced Interrupt Controller outputs, either nIRQ or nFIQ, wakes up the ARM processor while it is in Idle Mode. The General Interrupt Mask feature enables the AIC to wake up the processor without asserting the interrupt line of the processor, thus providing synchronization of the processor on an event.

17.7.3 Interrupt Sources

The Interrupt Source 0 is always located at FIQ. If the product does not feature a FIQ pin, the Interrupt Source 0 cannot be used.

The Interrupt Source 1 is always located at System Interrupt. This is the result of the OR-wiring of the system peripheral interrupt lines. When a system interrupt occurs, the service routine must first distinguish the cause of the interrupt. This is performed by reading successively the status registers of the above mentioned system peripherals.

The interrupt sources 2 to 127 can either be connected to the interrupt outputs of an embedded user peripheral or to external interrupt lines. The external interrupt lines can be connected directly, or through the PIO Controller.

The PIO Controllers are considered as user peripherals in the scope of interrupt handling. Accordingly, the PIO Controller interrupt lines are connected to the Interrupt Sources 2 to 127.

The peripheral identification defined at the product level corresponds to the interrupt source number (as well as the bit number controlling the clock of the peripheral). Consequently, to simplify the description of the functional operations and the user interface, the interrupt sources are named FIQ, SYS, and PID2 to PID127.

17.8 Functional Description

17.8.1 Interrupt Source Control

17.8.1.1 Interrupt Source Mode

The Advanced Interrupt Controller independently programs each interrupt source. The SRCTYPE field of the AIC_SMR (Source Mode Register) selects the interrupt condition of the interrupt source selected by the INTSEL field of the ”AIC Source Select Register”.

Fast Forcing is enabled or disabled by writing to the Fast Forcing Enable Register (AIC_FFER) and the Fast Forcing Disable Register (AIC_FFDR). Writing to these registers results in an update of the Fast Forcing Status Register (AIC_FFSR) that controls the feature for each internal or external interrupt source.

When Fast Forcing is disabled, the interrupt sources are handled as described in the previous pages.

When Fast Forcing is enabled, the edge/level programming and, in certain cases, edge detection of the interrupt source is still active but the source cannot trigger a normal interrupt to the processor and is not seen by the priority handler.

If the interrupt source is programmed in level-sensitive mode and an active level is sampled, Fast Forcing results in the assertion of the nFIQ line to the core.

If the interrupt source is programmed in edge-triggered mode and an active edge is detected, Fast Forcing results in the assertion of the nFIQ line to the core.

The Fast Forcing feature does not affect the Source 0 pending bit in the Interrupt Pending Register (AIC_IPR).

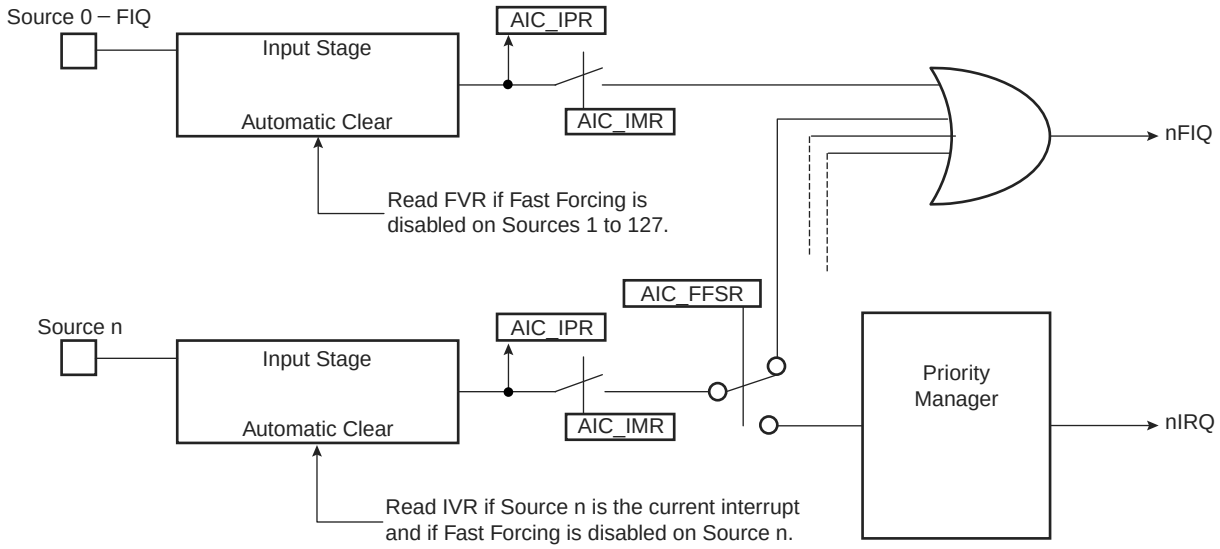
The FIQ Vector Register (AIC_FVR) reads the contents of the Source Vector Register 0 (AIC_SVR0), whatever the source of the fast interrupt may be. The read of the FVR does not clear the Source 0 when the fast forcing feature is used and the interrupt source should be cleared by writing to the Interrupt Clear Command Register (AIC_ICCR).

All enabled and pending interrupt sources that have the fast forcing feature enabled and that are programmed in edge-triggered mode must be cleared by writing to the Interrupt Clear Command Register. In doing so, they are cleared independently and thus lost interrupts are prevented.

The read of AIC_IVR does not clear the source that has the fast forcing feature enabled.

The source 0, reserved to the fast interrupt, continues operating normally and becomes one of the Fast Interrupt sources.

Figure 17-10.Fast Forcing



17.8.5 Protect Mode

The Protect Mode permits reading the Interrupt Vector Register without performing the associated automatic operations. This is necessary when working with a debug system. When a debugger, working either with a Debug Monitor or the ARM processor's ICE, stops the applications and updates the opened windows, it might read the AIC User Interface and thus the IVR. This has undesirable consequences:

- If an enabled interrupt with a higher priority than the current one is pending, it is stacked.
- If there is no enabled pending interrupt, the spurious vector is returned.

In either case, an End of Interrupt command is necessary to acknowledge and to restore the context of the AIC. This operation is generally not performed by the debug system as the debug system would become strongly intrusive and cause the application to enter an undesired state.

18.3 Block Diagram

Figure 18-1. Watchdog Timer Block Diagram

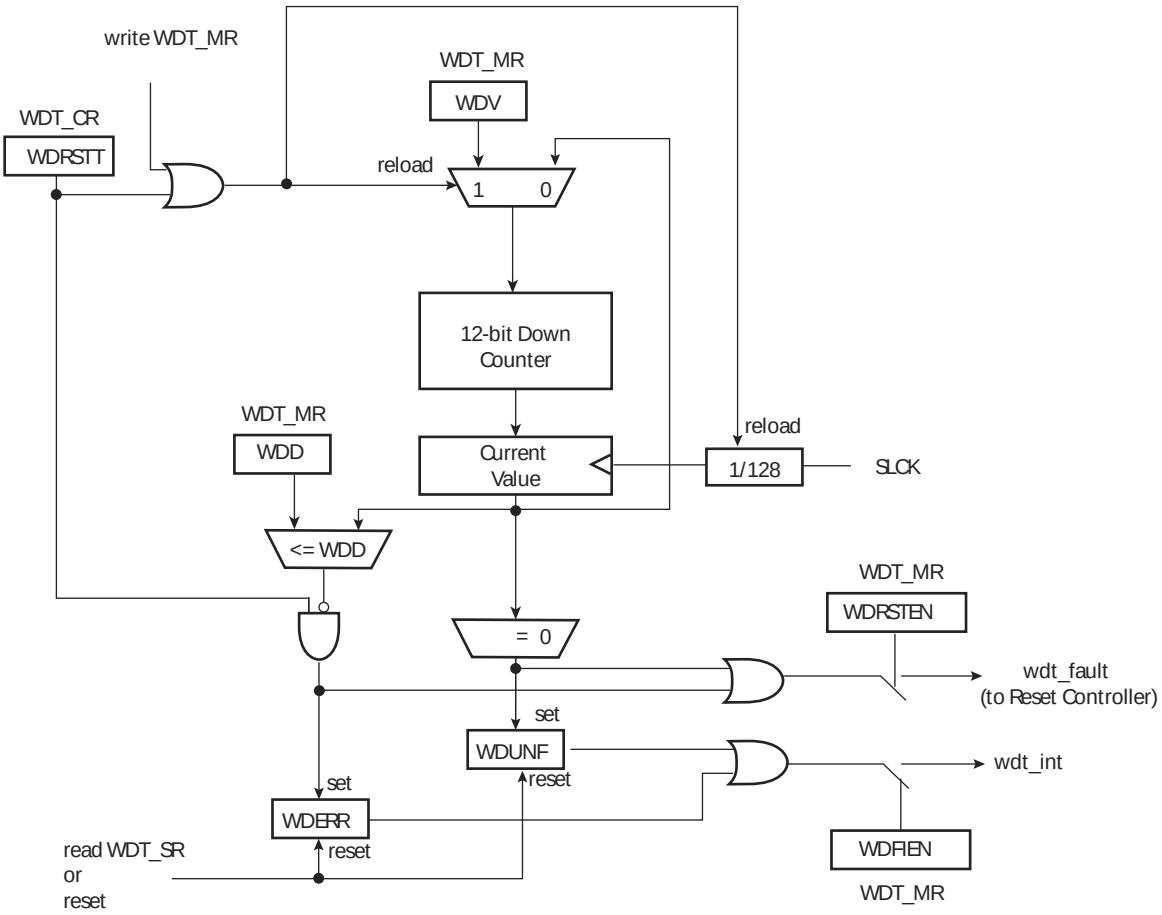
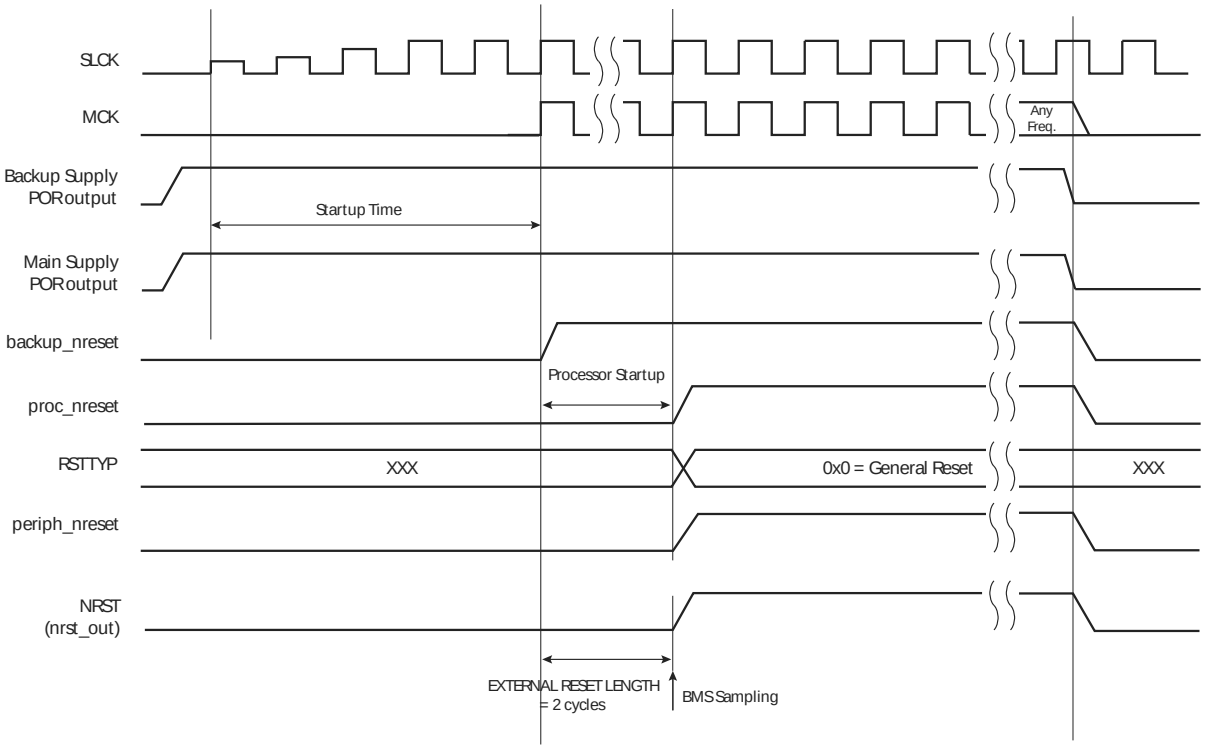


Figure 19-4. General Reset State



19.4.4.2 Wake-up Reset

The wake-up reset occurs when the main supply is down. When the main supply POR output is active, all the reset signals are asserted except backup_nreset. When the main supply powers up, the POR output is resynchronized on Slow Clock. The processor clock is then re-enabled during 3 Slow Clock cycles, depending on the requirements of the ARM processor.

At the end of this delay, the processor and other reset signals rise. The field RSTTYP in the RSTC_SR is updated to report a wake-up reset.

The “nrst_out” remains asserted for EXTERNAL_RESET_LENGTH cycles. As RSTC_MR is backed-up, the programmed number of cycles is applicable.

When the main supply is detected falling, the reset signals are immediately asserted. This transition is synchronous with the output of the main supply POR.

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, see Figure 30-4 "Memory Connection for an 8-bit Data Bus" and Figure 30-5 "Memory Connection for a 16-bit Data Bus"
A0	NBS0	8-bit or 16-bit data bus, see 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, see Figure 30-4 and Figure 30-5
A1	–	8-/16-bit data bus, see Section 30.9.1 “Data Bus Width” Byte-write or Byte-select access, see Figure 30-4 and Figure 30-5

30.10.6 Reset Values of Timing Parameters

Table 30-6 gives the default value of timing parameters at reset.

Table 30-6. Reset Values of Timing Parameters

Register	Reset Value	Description
HSMC_SETUP	–	All setup timings are set to 1
HSMC_PULSE	–	All pulse timings are set to 1
HSMC_CYCLE	–	The read and write operations last 3 Master Clock cycles and provide one hold cycle
WRITE_MODE	1	Write is controlled with NWE
READ_MODE	1	Read is controlled with NRD

30.10.7 Usage Restriction

The SMC does not check the validity of the user-programmed parameters. If the sum of SETUP and PULSE parameters is larger than the corresponding CYCLE parameter, this leads to an unpredictable behavior of the SMC.

30.10.7.1 For Read Operations

Null but positive setup and hold of address and NRD and/or NCS cannot be guaranteed at the memory interface because of the propagation delay of these signals through external logic and pads. When positive setup and hold values must be verified, then it is strictly recommended to program non-null values so as to cover possible skews between address, NCS and NRD signals.

30.10.7.2 For Write Operations

If a null hold value is programmed on NWE, the SMC can guarantee a positive hold of address, byte select lines, and NCS signal after the rising edge of NWE. This is true for WRITE_MODE = 1 only. See Section 30.12.2 “Early Read Wait State” on page 419.

30.10.7.3 For Read and Write Operations

A null value for pulse parameters is forbidden and may lead to an unpredictable behavior.

In read and write cycles, the setup and hold time parameters are defined in reference to the address bus. For external devices that require setup and hold time between NCS and NRD signals (read), or between NCS and NWE signals (write), these setup and hold times must be converted into setup and hold times in reference to the address bus.

30.11 Scrambling/Unscrambling Function

The external data bus D[15:0] can be scrambled in order to prevent intellectual property data located in off-chip memories from being easily recovered by analyzing data at the package pin level of either the microcontroller or the memory device.

The scrambling and unscrambling are performed on-the-fly without additional wait states.

The scrambling method depends on two user-configurable key registers, HSMC_KEY1 and HSMC_KEY2. These key registers are only accessible in write mode.

The key must be securely stored in a reliable non-volatile memory in order to recover data from the off-chip memory. Any data scrambled with a given key cannot be recovered if the key is lost.

The scrambling/unscrambling function can be enabled or disabled by programming the HSMC_OCMS register.

One bit is dedicated to enabling/disabling the NAND Flash scrambling, and one bit is dedicated to enabling/disabling the off chip SRAM scrambling. When at least one external SRAM is scrambled, the SMSC field must be set in the HSMC_OCMS register.

When multiple chip selects (external SRAM) are handled, it is possible to configure the scrambling function per chip select using the OCMS field in the HSMC_TIMINGS registers.

32.7.60 Overlay 2 Layer Interrupt Mask Register

Name: LCDC_OVRIMR2

Address: 0xF0030254

Access: Read-only

Reset: 0x00000000

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
–	OVR	DONE	ADD	DSCR	DMA	–	–

- **DMA: End of DMA Transfer Interrupt Mask Register**

0: Interrupt source is disabled.

1: Interrupt source is enabled.

- **DSCR: Descriptor Loaded Interrupt Mask Register**

0: Interrupt source is disabled.

1: Interrupt source is enabled.

- **ADD: Head Descriptor Loaded Interrupt Mask Register**

0: Interrupt source is disabled.

1: Interrupt source is enabled.

- **DONE: End of List Interrupt Mask Register**

0: Interrupt source is disabled.

1: Interrupt source is enabled.

- **OVR: Overflow Interrupt Mask Register**

0: Interrupt source is disabled.

1: Interrupt source is enabled.

34.7.12 UDPHS Endpoint Control Disable Register (Isochronous Endpoint)

Name: UDPHS_EPTCTLDISx [x=0..15] (ISOENDPT)

Address: 0xF8030108 [0], 0xF8030128 [1], 0xF8030148 [2], 0xF8030168 [3], 0xF8030188 [4], 0xF80301A8 [5], 0xF80301C8 [6], 0xF80301E8 [7], 0xF8030208 [8], 0xF8030228 [9], 0xF8030248 [10], 0xF8030268 [11], 0xF8030288 [12], 0xF80302A8 [13], 0xF80302C8 [14], 0xF80302E8 [15]

Access: Write-only

31	30	29	28	27	26	25	24
SHRT_PCKT	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	BUSY_BANK	–	–
15	14	13	12	11	10	9	8
–	ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
7	6	5	4	3	2	1	0
MDATA_RX	DATAX_RX	–	–	INTDIS_DMA	–	AUTO_VALID	EPT_DISABL

This register view is relevant only if EPT_TYPE = 0x1 in “UDPHS Endpoint Configuration Register” on page 888.

For additional information, see “UDPHS Endpoint Control Register (Isochronous Endpoint)” on page 902.

- **EPT_DISABL: Endpoint Disable**

0: No effect.

1: Disable endpoint.

- **AUTO_VALID: Packet Auto-Valid Disable**

0: No effect.

1: Disable this bit to not automatically validate the current packet.

- **INTDIS_DMA: Interrupts Disable DMA**

0: No effect.

1: Disable the “Interrupts Disable DMA”.

- **DATAX_RX: DATAx Interrupt Disable (Only for High Bandwidth Isochronous OUT endpoints)**

0: No effect.

1: Disable DATAx Interrupt.

- **MDATA_RX: MDATA Interrupt Disable (Only for High Bandwidth Isochronous OUT endpoints)**

0: No effect.

1: Disable MDATA Interrupt.

- **ERR_OVFLW: Overflow Error Interrupt Disable**

0: No effect.

1: Disable Overflow Error Interrupt.

- **ROVR: Receive Overrun**

Set when the receive overrun status bit is set. Cleared on read.

- **HRESP: HRESP Not OK**

Set when the DMA block sees HRESP not OK. Cleared on read.

- **PFNZ: Pause Frame with Non-zero Pause Quantum Received**

Indicates a valid pause has been received that has a non-zero pause quantum field. Cleared on read.

- **PTZ: Pause Time Zero**

Set when either the Pause Time Register at address 0x38 decrements to zero, or when a valid pause frame is received with a zero pause quantum field. Cleared on read.

- **PFTR: Pause Frame Transmitted**

Indicates a pause frame has been successfully transmitted after being initiated from the Network Control Register. Cleared on read.

- **DRQFR: PTP Delay Request Frame Received**

Indicates a PTP delay_req frame has been received. Cleared on read.

- **SFR: PTP Sync Frame Received**

Indicates a PTP sync frame has been received. Cleared on read.

- **DRQFT: PTP Delay Request Frame Transmitted**

Indicates a PTP delay_req frame has been transmitted. Cleared on read.

- **SFT: PTP Sync Frame Transmitted**

Indicates a PTP sync frame has been transmitted. Cleared on read.

- **PDRQFR: PDelay Request Frame Received**

Indicates a PTP pdelay_req frame has been received. Cleared on read.

- **PDRSFR: PDelay Response Frame Received**

Indicates a PTP pdelay_resp frame has been received. Cleared on read.

- **PDRQFT: PDelay Request Frame Transmitted**

Indicates a PTP pdelay_req frame has been transmitted. Cleared on read.

- **PDRSFT: PDelay Response Frame Transmitted**

Indicates a PTP pdelay_resp frame has been transmitted. Cleared on read.

- **SRI: TSU Seconds Register Increment**

Indicates the register has incremented. Cleared on read.

- **WOL: Wake On LAN**

WOL interrupt. Indicates a WOL event has been received.

36.7.71 1519 to Maximum Byte Frames Received Register

Name: GMAC_TMXBFR

Address: 0xF0028180

Access: Read-only

31	30	29	28	27	26	25	24
NFRX							

23	22	21	20	19	18	17	16
NFRX							

15	14	13	12	11	10	9	8
NFRX							

7	6	5	4	3	2	1	0
NFRX							

- **NFRX: 1519 to Maximum Byte Frames Received without Error**

This register counts the number of 1519 byte or above frames successfully received without error. Maximum frame size is determined by the Network Configuration Register bit 8 (1536 maximum frame size) or bit 3 (jumbo frame size). Excludes pause frames, and is only incremented if the frame is successfully filtered and copied to memory. See: Section 36.7.2 “Network Configuration Register”.

37.6.7 Receive Status Register

Name: EMAC_RSR

Address: 0xF802C020

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
–	–	–	–	–	OVR	REC	BNA

This register, when read, provides details of the status of a receive. Once read, individual bits may be cleared by writing 1 to them. It is not possible to set a bit to 1 by writing to the register.

- **BNA: Buffer Not Available**

An attempt was made to get a new buffer and the pointer indicated that it was owned by the processor. The DMA rereads the pointer each time a new frame starts until a valid pointer is found. This bit is set at each attempt that fails even if it has not had a successful pointer read since it has been cleared.

Cleared by writing a one to this bit.

- **REC: Frame Received**

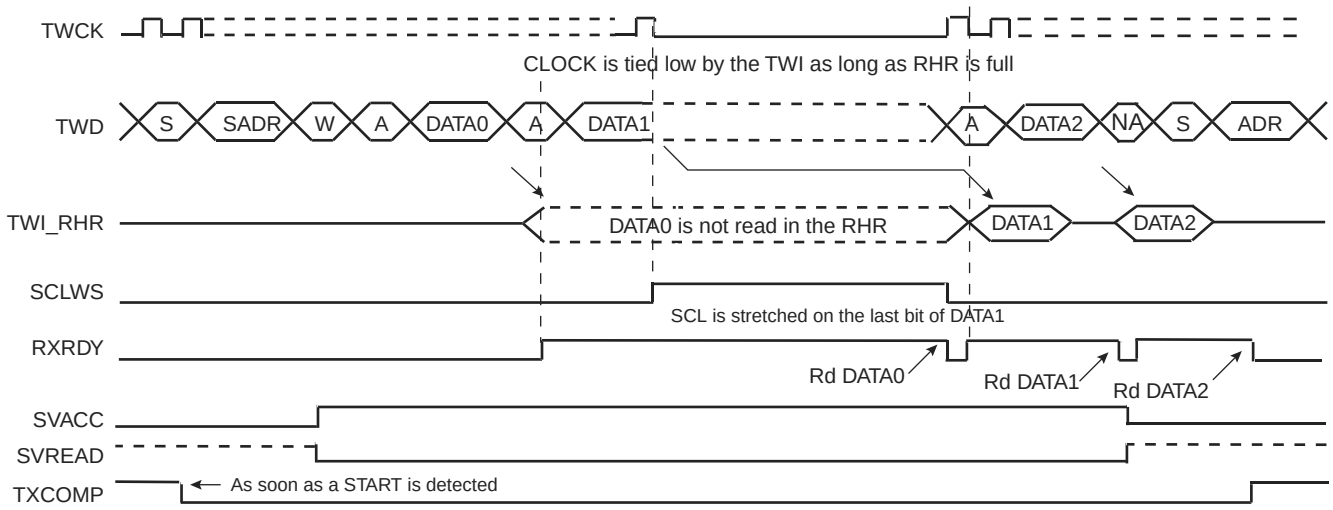
One or more frames have been received and placed in memory. Cleared by writing a one to this bit.

- **OVR: Receive Overrun**

The DMA block was unable to store the receive frame to memory, either because the bus was not granted in time or because a not OK `hresp(bus error)` was returned. The buffer is recovered if this happens.

Cleared by writing a one to this bit.

Figure 40-30.Clock Synchronization in Write Mode



Notes: 1. At the end of the read sequence, TXCOMP is set after a STOP or after a REPEATED_START + an address different from SADR.

2. SCLWS is automatically set when the clock synchronization mechanism is started and automatically reset when the mechanism is finished.

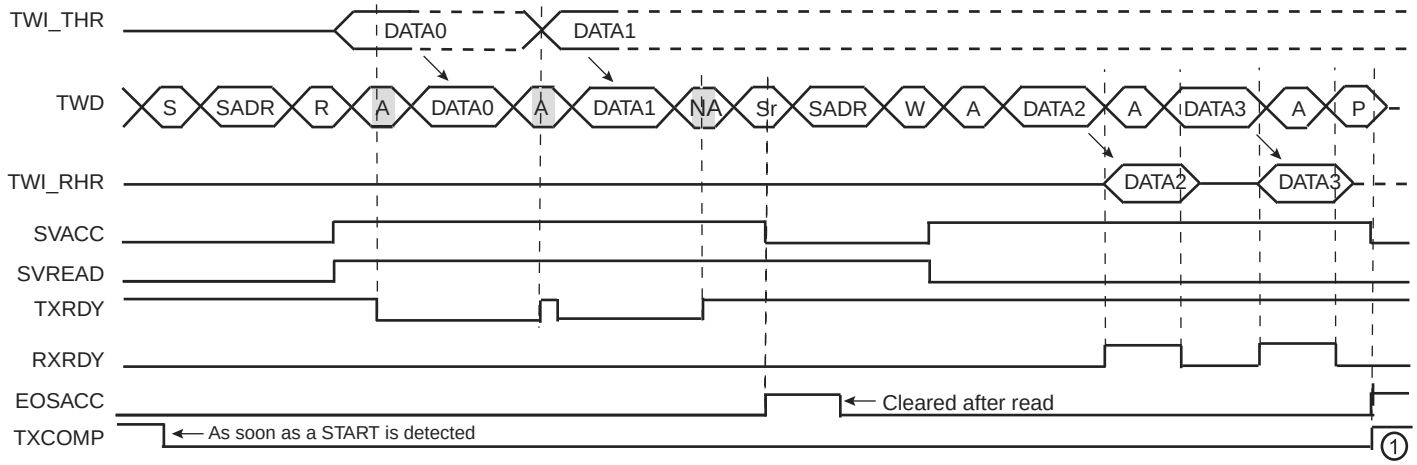
40.10.5.5 Reversal after a Repeated Start

Reversal of Read to Write

The master initiates the communication by a read command and finishes it by a write command.

Figure 40-31 describes the repeated start + reversal from Read to Write mode.

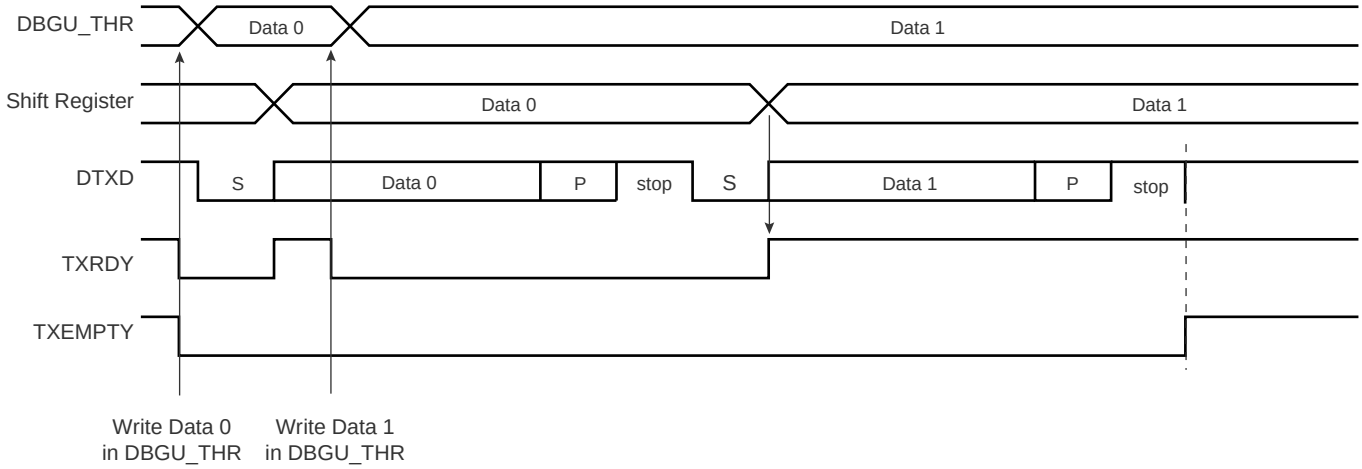
Figure 40-31.Repeated Start + Reversal from Read to Write Mode



Note: 1. TXCOMP is only set at the end of the transmission because after the repeated start, SADR is detected again.

When both the Shift Register and the DBGU_THR are empty, i.e., all the characters written in DBGU_THR have been processed, the bit TXEMPTY rises after the last stop bit has been completed.

Figure 42-11.Transmitter Control



42.5.4 DMA Support

Both the receiver and the transmitter of the Debug Unit's UART are connected to a DMA Controller (DMAC) channel.

The DMA Controller channels are programmed via registers that are mapped within the DMAC user interface.

42.5.5 Test Modes

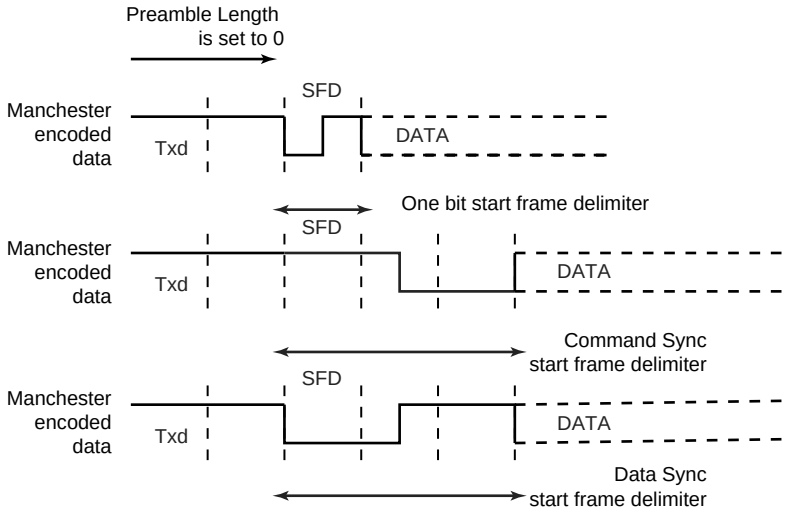
The Debug Unit supports three tests modes. These modes of operation are programmed by using the field CHMODE (Channel Mode) in the mode register DBGU_MR.

The Automatic Echo mode allows bit-by-bit retransmission. When a bit is received on the DRXD line, it is sent to the DTXD line. The transmitter operates normally, but has no effect on the DTXD line.

The Local Loopback mode allows the transmitted characters to be received. DTXD and DRXD pins are not used and the output of the transmitter is internally connected to the input of the receiver. The DRXD pin level has no effect and the DTXD line is held high, as in idle state.

The Remote Loopback mode directly connects the DRXD pin to the DTXD line. The transmitter and the receiver are disabled and have no effect. This mode allows a bit-by-bit retransmission.

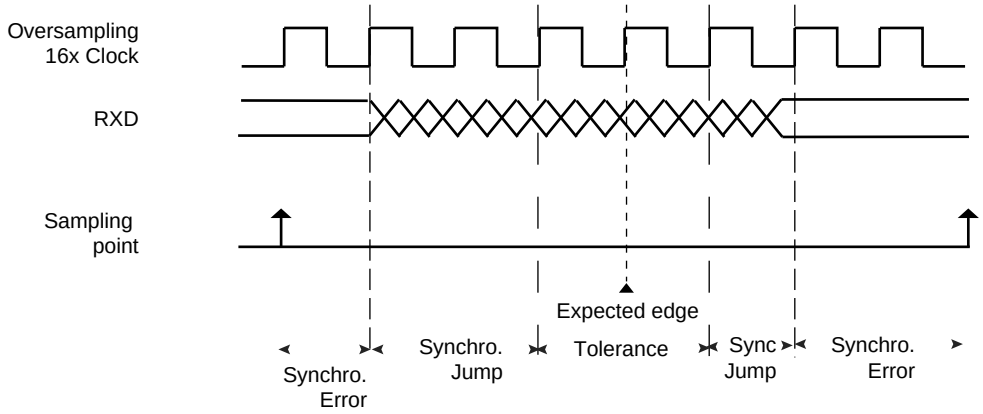
Figure 44-10.Start Frame Delimiter



Drift Compensation

Drift compensation is available only in 16X oversampling mode. An hardware recovery system allows a larger clock drift. To enable the hardware system, the bit in the USART_MAN register must be set. If the RXD edge is one 16X clock cycle from the expected edge, this is considered as normal jitter and no corrective actions is taken. If the RXD event is between 4 and 2 clock cycles before the expected edge, then the current period is shortened by one clock cycle. If the RXD event is between 2 and 3 clock cycles after the expected edge, then the current period is lengthened by one clock cycle. These intervals are considered to be drift and so corrective actions are automatically taken.

Figure 44-11.Bit Resynchronization



44.7.3.3 Asynchronous Receiver

If the USART is programmed in asynchronous operating mode (SYNC = 0), the receiver oversamples the RXD input line. The oversampling is either 16 or 8 times the baud rate clock, depending on the OVER bit in the US_MR.

The receiver samples the RXD line. If the line is sampled during one half of a bit time to 0, a start bit is detected and data, parity and stop bits are successively sampled on the bit rate clock.

If the oversampling is 16, (OVER to 0), a start is detected at the eighth sample to 0. Then, data bits, parity bit and stop bit are sampled on each 16 sampling clock cycle. If the oversampling is 8 (OVER to 1), a start bit is detected at the fourth sample to 0. Then, data bits, parity bit and stop bit are sampled on each 8 sampling clock cycle.

The number of data bits, first bit sent and parity mode are selected by the same fields and bits as the transmitter, i.e., respectively CHRL, MODE9, MSBF and PAR. For the synchronization mechanism **only**, the number of stop bits has no

44.7.7 SPI Mode

The Serial Peripheral Interface (SPI) Mode is a synchronous serial data link that provides communication with external devices in Master or Slave Mode. It also enables communication between processors if an external processor is connected to the system.

The Serial Peripheral Interface is essentially a shift register that serially transmits data bits to other SPIs. During a data transfer, one SPI system acts as the “master” which controls the data flow, while the other devices act as “slaves" which have data shifted into and out by the master. Different CPUs can take turns being masters and one master may simultaneously shift data into multiple slaves. (Multiple Master Protocol is the opposite of Single Master Protocol, where one CPU is always the master while all of the others are always slaves.) However, only one slave may drive its output to write data back to the master at any given time.

A slave device is selected when its NSS signal is asserted by the master. The USART in SPI Master mode can address only one SPI Slave because it can generate only one NSS signal.

The SPI system consists of two data lines and two control lines:

- Master Out Slave In (MOSI): This data line supplies the output data from the master shifted into the input of the slave.
- Master In Slave Out (MISO): This data line supplies the output data from a slave to the input of the master.
- Serial Clock (SCK): This control line is driven by the master and regulates the flow of the data bits. The master may transmit data at a variety of baud rates. The SCK line cycles once for each bit that is transmitted.
- Slave Select (NSS): This control line allows the master to select or deselect the slave.

44.7.7.1 Modes of Operation

The USART can operate in SPI Master Mode or in SPI Slave Mode.

Operation in SPI Master Mode is programmed by writing 0xE to the USART_MODE field in the Mode Register (US_MR). In this case the SPI lines must be connected as described below:

- The MOSI line is driven by the output pin TXD
- The MISO line drives the input pin RXD
- The SCK line is driven by the output pin SCK
- The NSS line is driven by the output pin RTS

Operation in SPI Slave Mode is programmed by writing to 0xF the USART_MODE field in the Mode Register. In this case the SPI lines must be connected as described below:

- The MOSI line drives the input pin RXD
- The MISO line is driven by the output pin TXD
- The SCK line drives the input pin SCK
- The NSS line drives the input pin CTS

In order to avoid unpredicted behavior, any change of the SPI Mode must be followed by a software reset of the transmitter and of the receiver (except the initial configuration after a hardware reset). (See Section 44.7.7.4).

44.7.7.2 Baud Rate

In SPI Mode, the baud rate generator operates in the same way as in USART synchronous mode: See “Baud Rate in Synchronous Mode or SPI Mode” on page 1374. However, there are some restrictions:

In SPI Master Mode:

- The external clock SCK must not be selected (USCLKS ≠ 0x3), and the bit CLKO must be set to ‘1’ in the US_MR, in order to generate correctly the serial clock on the SCK pin.
- To obtain correct behavior of the receiver and the transmitter, the value programmed in CD must be superior or equal to 6.
- If the internal clock divided (MCK/DIV) is selected, the value programmed in CD must be even to ensure a 50:50 mark/space ratio on the SCK pin, this value can be odd if the internal clock is selected (MCK).

48.7.23 PWM Fault Mode Register

Name: PWM_FMR

Address: 0xF002C05C

Access: Read/Write

31	30	29	28	27	26	25	24
–							

23	22	21	20	19	18	17	16
FFIL							

15	14	13	12	11	10	9	8
FMOD							

7	6	5	4	3	2	1	0
FPOL							

This register can only be written if bits WPSWS5 and WPHWS5 are cleared in the “PWM Write Protection Status Register” .

- **FPOL: Fault Polarity**

For each field bit y (fault input number):

0: The fault y becomes active when the fault input y is at 0.

1: The fault y becomes active when the fault input y is at 1.

- **FMOD: Fault Activation Mode**

For each field bit y (fault input number):

0: The fault y is active until the Fault condition is removed at the peripheral(1) level.

1: The fault y stays active until the Fault condition is removed at the peripheral(1) level AND until it is cleared in the “PWM Fault Clear Register” .

Note: 1. The Peripheral generating the fault.

- **FFIL: Fault Filtering**

For each field bit y (fault input number):

0: The fault input y is not filtered.

1: The fault input y is filtered.

CAUTION: To prevent an unexpected activation of the status flag FSy in the “PWM Fault Status Register” , the bit FMODy can be set to ‘1’ only if the FPOLy bit has been previously configured to its final value.

Table 49-4. Input Pins and Channel Number in Single Ended Mode

Input Pins	Channel Number
AD0	CH0
AD1	CH1
AD2	CH2
AD3	CH3
AD4	CH4
AD5	CH5
AD6	CH6
AD7	CH7
AD8	CH8
AD9	CH9
AD10	CH10
AD11	CH11
AD12	CH12
AD13	CH13
AD14	CH14
AD15	CH15

Table 49-5. Input Pins and Channel Number In Differential Mode

Input Pins	Channel Number
AD0-AD1	CH0
AD2-AD3	CH2
AD4-AD5	CH4
AD6-AD7	CH6
AD8-AD9	CH8
AD10-AD11	CH10
AD12-AD13	CH12
AD14-AD15	CH14

49.6.9 Input Gain and Offset

The ADC has a built in Programmable Gain Amplifier (PGA) and Programmable Offset.

The Programmable Gain Amplifier can be set to gains of 1/2, 1, 2 and 4. The Programmable Gain Amplifier can be used either for single ended applications or for fully differential applications.

If ANACH is set in ADC_MR the ADC can apply different gain and offset on each channel. Otherwise the parameters of CH0 are applied to all channels.

49.8.19 ADC Analog Control Register

Name: ADC_ACR

Address: 0xF8018094

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
–	–	–	–	–	–	PENDETSENS	

This register can only be written if the WPEN bit is cleared in “ADC Write Protect Mode Register” .

- **PENDETSENS: Pen Detection Sensitivity**

Allows to modify the pen detection input pull-up resistor value. (See the product electrical characteristics for further details).