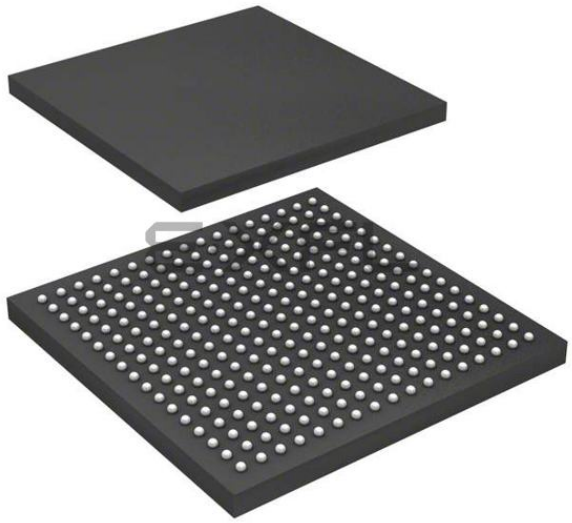


Welcome to E-XFL.COM

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	289-LFBGA
Supplier Device Package	289-LFBGA (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsama5d41b-cur

2. Signal Description

Table 2-1 gives details on signal names classified by peripheral.

Table 2-1: Signal Description List

Signal Name	Function	Type	Active Level
Clocks, Oscillators and PLLs			
XIN	Main Oscillator Input	Input	–
XOUT	Main Oscillator Output	Output	–
XIN32	Slow Clock Oscillator Input	Input	–
XOUT32	Slow Clock Oscillator Output	Output	–
VBG	Bias Voltage Reference for USB	Analog	–
PCK0–PCK2	Programmable Clock Output	Output	–
Shutdown, Wakeup Logic			
SHDN	Shutdown Control	Output	–
WKUP	Wakeup Input	Input	–
ICE and JTAG			
TCK/SWCLK	Test Clock/Serial Wire Clock	Input	–
TDI	Test Data In	Input	–
TDO	Test Data Out	Output	–
TMS/SWDIO	Test Mode Select/Serial Wire Input/Output	I/O	–
JTAGSEL	JTAG Selection	Input	–
Reset/Test			
NRST	Microprocessor Reset	Input	Low
TST	Test Mode Select	Input	–
NTRST	Test Reset Signal	Input	–
Debug Unit - DBGU			
DRXD	Debug Receive Data	Input	–
DTXD	Debug Transmit Data	Output	–
Advanced Interrupt Controller - AIC			
IRQ	External Interrupt Input	Input	–
Secured Advanced Interrupt Controller - SAIC			
FIQ	Fast Interrupt Input	Input	–
PIO Controller - PIOA - PIOB - PIOC - PIOD - PIOE			
PA0–PAxx	Parallel IO Controller A	I/O	–
PB0–PBxx	Parallel IO Controller B	I/O	–
PC0–PCxx	Parallel IO Controller C	I/O	–
PD8–PDxx	Parallel IO Controller D	I/O	–
PE0–PExx	Parallel IO Controller E	I/O	–

Table 2-1: Signal Description List (Continued)

Signal Name	Function	Type	Active Level
CTSx	USARTx Clear To Send	Input	–
Universal Asynchronous Receiver Transmitter - UARTx [1..0]			
UTXDx	UARTx Transmit Data	Output	–
URXDx	UARTx Receive Data	Input	–
Synchronous Serial Controller - SSCx [1..0]			
TDx	SSC Transmit Data	Output	–
RDx	SSC Receive Data	Input	–
TKx	SSC Transmit Clock	I/O	–
RKx	SSC Receive Clock	I/O	–
TFx	SSC Transmit Frame Sync	I/O	–
RFx	SSC Receive Frame Sync	I/O	–
Timer/Counter - TCx [8..0]			
TCLKx	TC Channel x External Clock Input	Input	–
TIOAx	TC Channel x I/O Line A	I/O	–
TIOBx	TC Channel x I/O Line B	I/O	–
Serial Peripheral Interface - SPIx [2..0]			
SPIx_MISO	Master In Slave Out	I/O	–
SPIx_MOSI	Master Out Slave In	I/O	–
SPIx_SPCK	SPI Serial Clock	I/O	–
SPIx_NPCS0	SPI Peripheral Chip Select 0	I/O	Low
SPIx_NPCS[3..1]	SPI Peripheral Chip Select	Output	Low
Two-wire Interface - TWIx [3..0]			
TWDx	Two-wire Serial Data	I/O	–
TWCKx	Two-wire Serial Clock	I/O	–
Pulse Width Modulation Controller - PWM			
PWMH0–3	PWM Waveform Output High	–	Output
PWML0–3	PWM Waveform Output Low	–	Output
PWMFI0–1	PWM Fault Inputs	–	Input
USB Host High Speed Port - UHPHS			
HHSDPA	USB Host Port A High Speed Data +	Analog	–
HHSDMA	USB Host Port A High Speed Data -	Analog	–
HHSDPB	USB Host Port B High Speed Data +	Analog	–
HHSDMB	USB Host Port B High Speed Data -	Analog	–
HHSDPC	USB Host Port C High Speed Data +	Analog	–
HHSDMC	USB Host Port C High Speed Data -	Analog	–
USB Device High Speed Port - UDPHS			

27.19.9 PMC Clock Generator Main Clock Frequency Register

Name:CKGR_MCFR

Address:0xF0018024

Access: Read/Write

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

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

15	14	13	12	11	10	9	8
MAINF							

7	6	5	4	3	2	1	0
MAINF							

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

MAINF: Main Clock Frequency

Gives the number of Main clock cycles within 16 Slow clock periods. To calculate the frequency of the measured clock:

$f_{MAINCK} = (MAINF \times f_{SLCK}) / 16$

where frequency is in MHz.

MAINFRDY: Main Clock Frequency Measure Ready

0: MAINF value is not valid or the measured oscillator is disabled or a measure has just been started by means of RCMEAS.

1: The measured oscillator has been enabled previously and MAINF value is available.

Note: To ensure that a correct value is read on the MAINF field, the MAINFRDY flag must be read at 1 then another read access must be performed on the register to get a stable value on the MAINF field.

RCMEAS: RC Oscillator Frequency Measure (write-only)

0: No effect.

1: Restarts measuring of the frequency of the Main clock source. MAINF will carry the new frequency as soon as a low to high transition occurs on the MAINFRDY flag.

The measure is performed on the main frequency (i.e., not limited to RC oscillator only), but if the Main clock frequency source is the 12 MHz crystal oscillator, the restart of measuring is not needed because of the well known stability of crystal oscillators.

28.5.4 Output Control

When the I/O line is assigned to a peripheral function, i.e., the corresponding bit in PIO_PSR is at zero, the drive of the I/O line is controlled by the peripheral. Peripheral A or B or C or D depending on the value in PIO_ABCDSR1 and PIO_ABCDSR2 determines whether the pin is driven or not.

When the I/O line is controlled by the PIO Controller, the pin can be configured to be driven. This is done by writing the Output Enable Register (PIO_OER) and Output Disable Register (PIO_ODR). The results of these write operations are detected in the Output Status Register (PIO_OSR). When a bit in this register is at zero, the corresponding I/O line is used as an input only. When the bit is at one, the corresponding I/O line is driven by the PIO Controller.

The level driven on an I/O line can be determined by writing in the Set Output Data Register (PIO_SODR) and the Clear Output Data Register (PIO_CODR). These write operations, respectively, set and clear the Output Data Status Register (PIO_ODSR), which represents the data driven on the I/O lines. Writing in PIO_OER and PIO_ODR manages PIO_OSR whether the pin is configured to be controlled by the PIO Controller or assigned to a peripheral function. This enables configuration of the I/O line prior to setting it to be managed by the PIO Controller.

Similarly, writing in PIO_SODR and PIO_CODR affects PIO_ODSR. This is important as it defines the first level driven on the I/O line.

28.5.5 Synchronous Data Output

Clearing one or more PIO line(s) and setting another one or more PIO line(s) synchronously cannot be done by using PIO_SODR and PIO_CODR. It requires two successive write operations into two different registers. To overcome this, the PIO Controller offers a direct control of PIO outputs by single write access to PIO_ODSR. Only bits unmasked by the Output Write Status Register (PIO_OWSR) are written. The mask bits in PIO_OWSR are set by writing to the Output Write Enable Register (PIO_OWER) and cleared by writing to the Output Write Disable Register (PIO_OWDR).

After reset, the synchronous data output is disabled on all the I/O lines as PIO_OWSR resets at 0x0.

28.5.6 Multi-Drive Control (Open Drain)

Each I/O can be independently programmed in open drain by using the multi-drive feature. This feature permits several drivers to be connected on the I/O line which is driven low only by each device. An external pullup resistor (or enabling of the internal one) is generally required to guarantee a high level on the line.

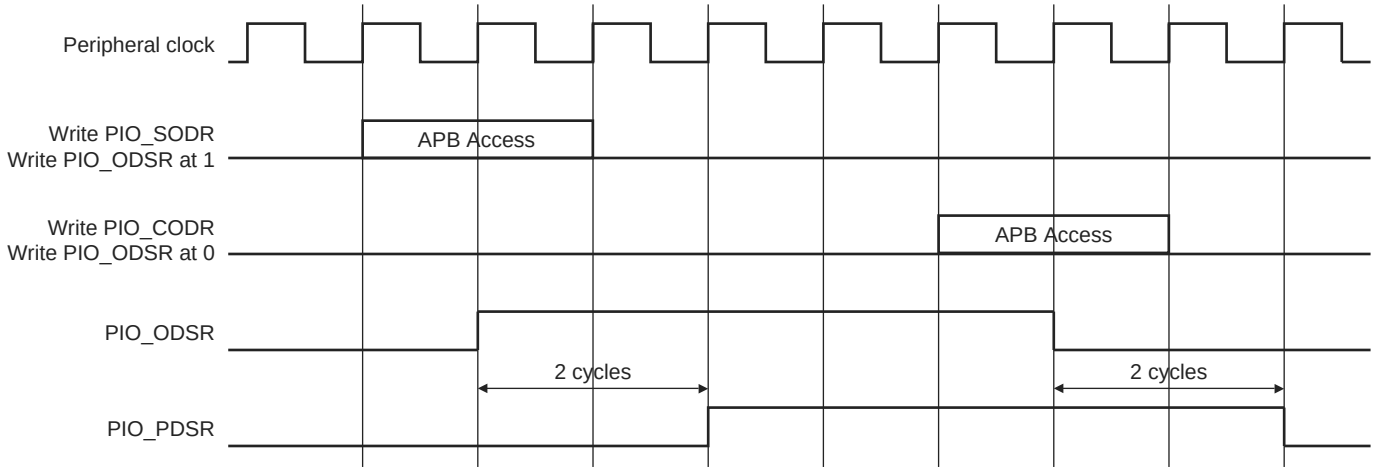
The multi-drive feature is controlled by the Multi-driver Enable Register (PIO_MDER) and the Multi-driver Disable Register (PIO_MDDR). The multi-drive can be selected whether the I/O line is controlled by the PIO Controller or assigned to a peripheral function. The Multi-driver Status Register (PIO_MDSR) indicates the pins that are configured to support external drivers.

After reset, the multi-drive feature is disabled on all pins, i.e., PIO_MDSR resets at value 0x0.

28.5.7 Output Line Timings

Figure 28-3 shows how the outputs are driven either by writing PIO_SODR or PIO_CODR, or by directly writing PIO_ODSR. This last case is valid only if the corresponding bit in PIO_OWSR is set. Figure 28-3 also shows when the feedback in the Pin Data Status Register (PIO_PDSR) is available.

Figure 28-3: Output Line Timings



28.6.21 PIO Multi-driver Status Register

Name:PIO_MDSR

Address:0xFC06A058 (PIOA), 0xFC06B058 (PIOB), 0xFC06C058 (PIOC), 0xFC068058 (PIOD), 0xFC06D058 (PIOE)

Access: Read-only

31	30	29	28	27	26	25	24
P31	P30	P29	P28	P27	P26	P25	P24
23	22	21	20	19	18	17	16
P23	P22	P21	P20	P19	P18	P17	P16
15	14	13	12	11	10	9	8
P15	P14	P13	P12	P11	P10	P9	P8
7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0

P0–P31: Multi-drive Status

0: The multi-drive is disabled on the I/O line. The pin is driven at high- and low-level.

1: The multi-drive is enabled on the I/O line. The pin is driven at low-level only.

28.6.39 PIO Additional Interrupt Modes Mask Register

Name:PIO_AIMMR

Address:0xFC06A0B8 (PIOA), 0xFC06B0B8 (PIOB), 0xFC06C0B8 (PIOC), 0xFC0680B8 (PIOD), 0xFC06D0B8 (PIOE)

Access: Read-only

31	30	29	28	27	26	25	24
P31	P30	P29	P28	P27	P26	P25	P24
23	22	21	20	19	18	17	16
P23	P22	P21	P20	P19	P18	P17	P16
15	14	13	12	11	10	9	8
P15	P14	P13	P12	P11	P10	P9	P8
7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0

P0–P31: IO Line Index

Selects the IO event type triggering an interrupt.

0: The interrupt source is a both-edge detection event.

1: The interrupt source is described by the registers PIO_ELSR and PIO_FRLHSR.

The application must write a 2 to the MODE field in the MPDDRC_MR. Read the MPDDRC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to any DDR2-SDRAM address to acknowledge this command.

7. An Extended Mode Register Set (EMRS2) cycle is issued to choose between commercial or high temperature operations. The application must write a 5 to the MODE field in the MPDDRC_MR. Read the MPDDRC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to the DDR2-SDRAM to acknowledge this command. The write address must be chosen so that signal BA[1] is set to 1 and signal BA[0] is set to 0. For example: with a 16-bit, 128-Mbit, DDR2-SDRAM (12 rows, 9 columns, 4 banks), the DDR2-SDRAM write access should be done at the address: BASE_ADDRESS_DDR + 0x00800000; with a 32-bit, 1-Gbit, DDR2-SDRAM (14 rows, 10 columns, 8 banks), the SDRAM write access should be done at the address: BASE_ADDRESS_DDR + 0x08000000.

Note: This address is given as an example only. The real address depends on implementation in the product.

8. An Extended Mode Register Set (EMRS3) cycle is issued to set the Extended Mode Register to 0. The application must write a 5 to the MODE field in the MPDDRC_MR. Read the MPDDRC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to the DDR2-SDRAM to acknowledge this command. The write address must be chosen so that signal BA[1] is set to 1 and signal BA[0] is set to 1. For example: with a 16-bit, 128-Mbit, DDR2-SDRAM (12 rows, 9 columns, 4 banks), the DDR2-SDRAM write access should be done at the address: BASE_ADDRESS_DDR + 0x00C00000; with a 32-bit, 1-Gbit, DDR2-SDRAM (14 rows, 10 columns, 8 banks), the SDRAM write access should be done at the address: BASE_ADDRESS_DDR + 0x0C000000.
9. An Extended Mode Register Set (EMRS1) cycle is issued to enable DLL and to program D.I.C. (Output Driver Impedance Control). The application must write a 5 to the MODE field in the MPDDRC_MR. Read the MPDDRC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to the DDR2-SDRAM to acknowledge this command. The write address must be chosen so that signal BA[1] is set to 0 and signal BA[0] is set to 1. For example: with a 16-bit, 128-Mbit, DDR2-SDRAM (12 rows, 9 columns, 4 banks), the DDR2-SDRAM write access should be done at the address: BASE_ADDRESS_DDR + 0x00400000; with a 32-bit, 1-Gbit, DDR2-SDRAM (14 rows, 10 columns, 8 banks), the SDRAM write access should be done at the address: BASE_ADDRESS_DDR + 0x04000000.
10. An additional 200 cycles of clock are required for locking DLL
11. Write a one to the DLL bit (enable DLL reset) in the MPDDRC Configuration Register (MPDDRC_CR).
12. A Mode Register Set (MRS) cycle is issued to reset DLL. The application must write a 3 to the MODE field in the MPDDRC_MR. Read the MPDDRC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to the DDR2-SDRAM to acknowledge this command. The write address must be chosen so that signals BA[1:0] are set to 0. For example, the SDRAM write access should be done at the address: BASE_ADDRESS_DDR.
13. An All Banks Precharge command is issued to the DDR2-SDRAM. Program the All Banks Precharge command in the MPDDRC_MR. The application must write a 2 to the MODE field in the MPDDRC_MR. Read the MPDDRC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to any DDR2-SDRAM address to acknowledge this command.
14. Two auto-refresh (CBR) cycles are provided. Program the Auto Refresh command (CBR) in the MPDDRC_MR. The application must write a 4 to the MODE field in the MPDDRC_MR. Read the MPDDRC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to any DDR2-SDRAM location twice to acknowledge these commands.
15. Write a zero to the DLL bit (disable DLL reset) in the MPDDRC_CR.
16. A Mode Register Set (MRS) cycle is issued to program parameters of the DDR2-SDRAM device, in particular CAS latency and to disable DLL reset. The application must write a 3 to the MODE field in the MPDDRC_MR. Read the MPDDRC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to the DDR2-SDRAM to acknowledge this command. The write address must be chosen so that signals BA[1:0] are set to 0. For example: with a 16-bit, 128-Mbit, DDR2-SDRAM (12 rows, 9 columns, 4 banks) bank address, the SDRAM write access should be done at the address: BASE_ADDRESS_DDR; with a 32-bit, 1-Gbit, DDR2-SDRAM (14 rows, 10 columns, 8 banks), the SDRAM write access should be done at the address: BASE_ADDRESS_DDR.
17. Write a seven to the OCD field (default OCD calibration) in the MPDDRC_CR.
18. An Extended Mode Register Set (EMRS1) cycle is issued to the default OCD value. The application must write a 5 to the MODE field in the MPDDRC_MR. Read the MPDDRC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to the DDR2-SDRAM to acknowledge this command. The write address must be chosen so that signal BA[1] is set to 0 and signal BA[0] is set to 1. For example: with a 16-bit, 128-Mbit, DDR2-SDRAM (12 rows, 9 columns, 4 banks), the DDR2-SDRAM write access should be done at the address: BASE_ADDRESS_DDR + 0x00400000; with a 32-bit, 1-Gbit, DDR2-SDRAM (14 rows, 10 columns, 8 banks), the SDRAM write access should be done at the address: BASE_ADDRESS_DDR + 0x04000000.
19. Write a zero to the OCD field (exit OCD calibration mode) in the MPDDRC_CR.
20. An Extended Mode Register Set (EMRS1) cycle is issued to enable OCD exit. The application must write a 5 to the MODE field in the MPDDRC_MR. Read the MPDDRC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to the DDR2-SDRAM to acknowledge this command. The write address must be chosen so that signal BA[1] is set to 0 and

29.7.25 MPDDRC DLL Offset Selection Register

Name:MPDDRC_DLL_OS

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
–	–	–	–	–	–	–	SELOFF

SELOFF: Offset Selection

0: The hardcoded offsets are selected.

1: The programmable offsets are selected.

30.20.8 NFC Bank Register

Name: HSMC_BANK

Address:0xFC05C01C

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
–	–	–	–	–	–	–	BANK

BANK: Bank Identifier

0: Bank 0 is used.

1: Bank 1 is used.

31.9.19 XDMAC Channel x [x = 0..15] Interrupt Disable Register

Name:XDMAC_CIDx [x = 0..15]

Address:0xF0004054 (1)[0], 0xF0004094 (1)[1], 0xF00040D4 (1)[2], 0xF0004114 (1)[3], 0xF0004154 (1)[4], 0xF0004194 (1)[5], 0xF00041D4 (1)[6], 0xF0004214 (1)[7], 0xF0004254 (1)[8], 0xF0004294 (1)[9], 0xF00042D4 (1)[10], 0xF0004314 (1)[11], 0xF0004354 (1)[12], 0xF0004394 (1)[13], 0xF00043D4 (1)[14], 0xF0004414 (1)[15], 0xF0014054 (0)[0], 0xF0014094 (0)[1], 0xF00140D4 (0)[2], 0xF0014114 (0)[3], 0xF0014154 (0)[4], 0xF0014194 (0)[5], 0xF00141D4 (0)[6], 0xF0014214 (0)[7], 0xF0014254 (0)[8], 0xF0014294 (0)[9], 0xF00142D4 (0)[10], 0xF0014314 (0)[11], 0xF0014354 (0)[12], 0xF0014394 (0)[13], 0xF00143D4 (0)[14], 0xF0014414 (0)[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
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	ROID	WBEID	RBEID	FID	DID	LID	BID

BID: End of Block Interrupt Disable Bit

0: No effect.

1: Disables end of block interrupt.

LID: End of Linked List Interrupt Disable Bit

0: No effect.

1: Disables end of linked list interrupt.

DID: End of Disable Interrupt Disable Bit

0: No effect.

1: Disables end of disable interrupt.

FID: End of Flush Interrupt Disable Bit

0: No effect.

1: Disables end of flush interrupt.

RBEID: Read Bus Error Interrupt Disable Bit

0: No effect.

1: Disables bus error interrupt.

WBEID: Write Bus Error Interrupt Disable Bit

0: No effect.

1: Disables bus error interrupt.

ROID: Request Overflow Error Interrupt Disable Bit

0: No effect.

1: Disables request overflow error interrupt.

32.7.11 LCD Controller Interrupt Enable Register

Name: LCDC_LCDIER

Address:0xF000002C

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
–	–	–	–	HEOIE	OVR2IE	OVR1IE	BASEIE
7	6	5	4	3	2	1	0
–	–	–	FIFOERRIE	–	DISPIE	DISIE	SOFIE

SOFIE: Start of Frame Interrupt Enable

0: No effect.

1: Enables the interrupt.

DISIE: LCD Disable Interrupt Enable

0: No effect.

1: Enables the interrupt.

DISPIE: Powerup/Powerdown Sequence Terminated Interrupt Enable

0: No effect.

1: Enables the interrupt.

FIFOERRIE: Output FIFO Error Interrupt Enable

0: No effect.

1: Enables the interrupt.

BASEIE: Base Layer Interrupt Enable

0: No effect.

1: Enables the interrupt.

OVR1IE: Overlay 1 Interrupt Enable

0: No effect.

1: Enables the interrupt.

OVR2IE: Overlay 2 Interrupt Enable

0: No effect.

1: Enables the interrupt.

HEOIE: High End Overlay Interrupt Enable

0: No effect.

1: Enables the interrupt.

Figure 35-10: Data IN Transfer for Endpoint with Two Banks

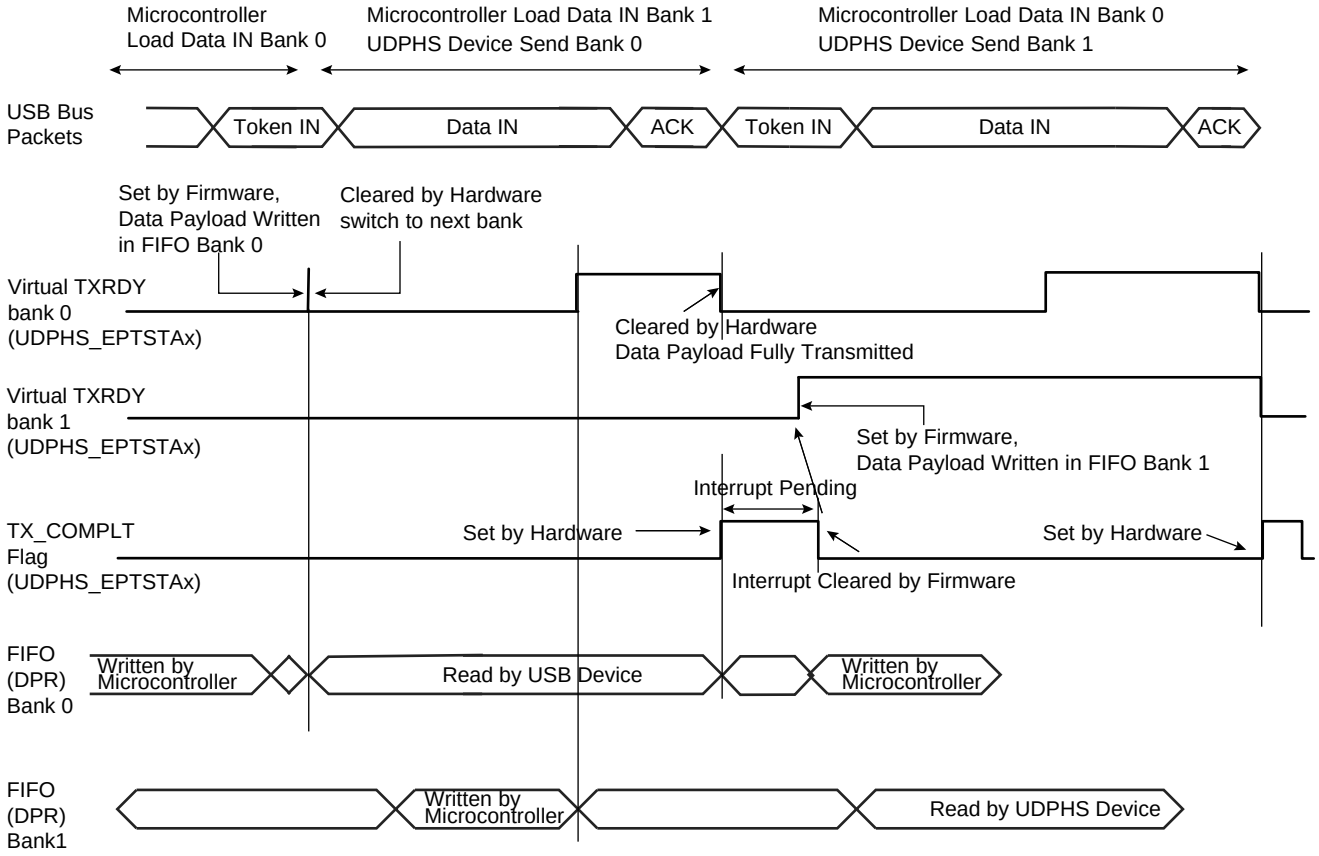
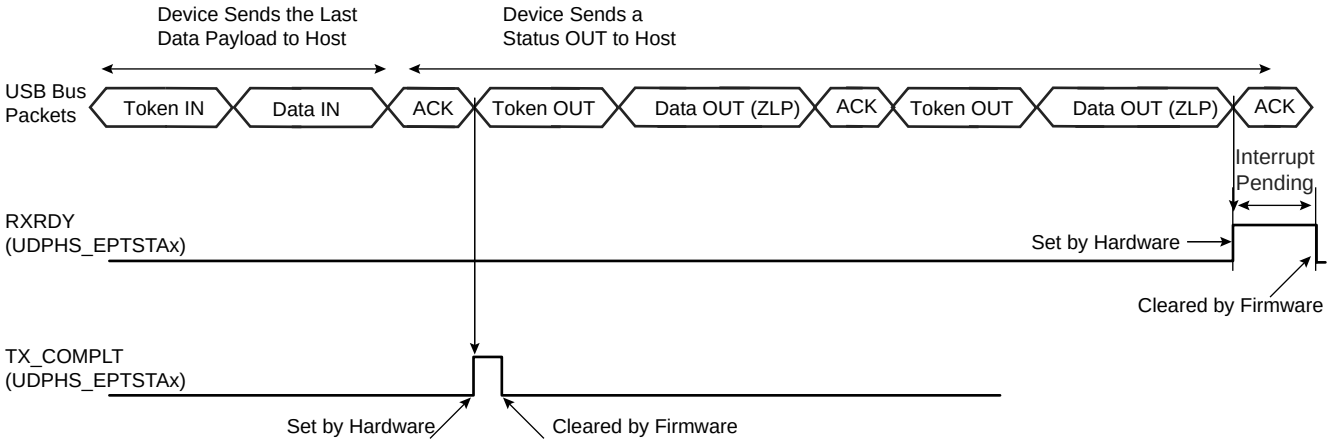


Figure 35-11: Data IN Followed By Status OUT Transfer at the End of a Control Transfer



Note: A NAK handshake is always generated at the first status stage token.

Table 37-14: Example of Pdelay_Req Frame in 1588 Version 2 (Ethernet Multicast) Format

Frame Segment	Value
Preamble/SFD	55555555555555D5
DA (Octets 0–5)	0180C200000E
SA (Octets 6–11)	—
Type (Octets 12–13)	88F7
Message type (Octet 14)	00
Version PTP (Octet 15)	02

37.6.15 Time Stamp Unit

The TSU consists of a timer and registers to capture the time at which PTP event frames cross the message timestamp point. An interrupt is issued when a capture register is updated.

The timer is implemented as a 94-bit register with the upper 48 bits counting seconds, the next 30 bits counting nanoseconds and the lowest 16 bits counting sub-nanoseconds. The lower 46 bits rolls over when they have counted to one second. An interrupt is generated when the seconds increment. The timer value can be read, written and adjusted through the APB interface. The timer is clocked by MCK.

The amount by which the timer increments each clock cycle is controlled by the timer increment registers (GMAC_TI). Bits 7:0 are the default increment value in nanoseconds and an additional 16 bits of sub-nanosecond resolution are available using the Timer Increment Sub-nanoseconds register (GMAC_TISUBN). If the rest of the register is written with zero, the timer increments by the value in [7:0], plus the value of GMAC_TISUBN, each clock cycle.

The GMAC_TISUBN register allows a resolution of approximately 15 femtoseconds.

Bits 15:8 of the increment register are the alternative increment value in nanoseconds and bits 23:16 are the number of increments after which the alternative increment value is used. If 23:16 are zero then the alternative increment value will never be used.

Taking the example of 10.2 MHz, there are 102 cycles every ten microseconds or 51 every five microseconds. So a timer with a 10.2 MHz clock source is constructed by incrementing by 98 ns for fifty cycles and then incrementing by 100 ns ($98 \times 50 + 100 = 5000$). This is programmed by setting the 1588 Timer Increment register to 0x00326462.

For a 49.8 MHz clock source it would be 20 ns for 248 cycles followed by an increment of 40 ns ($20 \times 248 + 40 = 5000$) programmed as 0x00F82814.

Having eight bits for the “number of increments” field allows frequencies up to 50 MHz to be supported with 200 kHz resolution.

Without the alternative increment field the period of the clock would be limited to an integer number of nanoseconds, resulting in supported clock frequencies of 8, 10, 20, 25, 40, 50, 100, 125, 200 and 250 MHz.

There are eight additional 80-bit registers that capture the time at which PTP event frames are transmitted and received. An interrupt is issued when these registers are updated. The TSU timer count value can be compared to a programmable comparison value. For the comparison, the 48 bits of the seconds value and the upper 22 bits of the nanoseconds value are used. An interrupt can also be generated (if enabled) when the TSU timer count value and comparison value are equal, mapped to bit 29 of the Interrupt Status register.

37.6.16 MAC 802.3 Pause Frame Support

Note: Refer to Clause 31, and Annex 31A and 31B of the IEEE standard 802.3 for a full description of MAC 802.3 pause operation.

The following table shows the start of a MAC 802.3 pause frame.

Table 37-15: Start of an 802.3 Pause Frame

Address		Type	Pause	
Destination	Source	(MAC Control Frame)	Opcode	Time
0x0180C2000001	6 bytes	0x8808	0x0001	2 bytes

The GMAC supports both hardware controlled pause of the transmitter, upon reception of a pause frame, and hardware generated pause frame transmission.

37.8.85 GMAC Receive Resource Errors Register

Name:GMAC_RRE

Address:0xF80201A0 (0), 0xFC0281A0 (1)

Access: Read-only

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

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

15	14	13	12	11	10	9	8
RXRER							

7	6	5	4	3	2	1	0
RXRER							

RXRER: Receive Resource Errors

This register 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 bit 8 is set in Network Configuration Register). This register 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. Refer to Section 37.8.2 “GMAC Network Configuration Register”.

38.14.1 HSMCI Control Register

Name: HSMCI_CR

Address:0xF8000000 (0), 0xFC000000 (1)

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
SWRST	–	–	–	PWSDIS	PWSEN	MCIDIS	MCIEN

MCIEN: Multi-Media Interface Enable

0: No effect.

1: Enables the Multi-Media Interface if MCDIS is 0.

MCIDIS: Multi-Media Interface Disable

0: No effect.

1: Disables the Multi-Media Interface.

PWSEN: Power Save Mode Enable

0: No effect.

1: Enables the Power Saving Mode if PWSDIS is 0.

WARNING: Before enabling this mode, the user must set a value different from 0 in the PWSDIV field of the HSMCI_MR.

PWSDIS: Power Save Mode Disable

0: No effect.

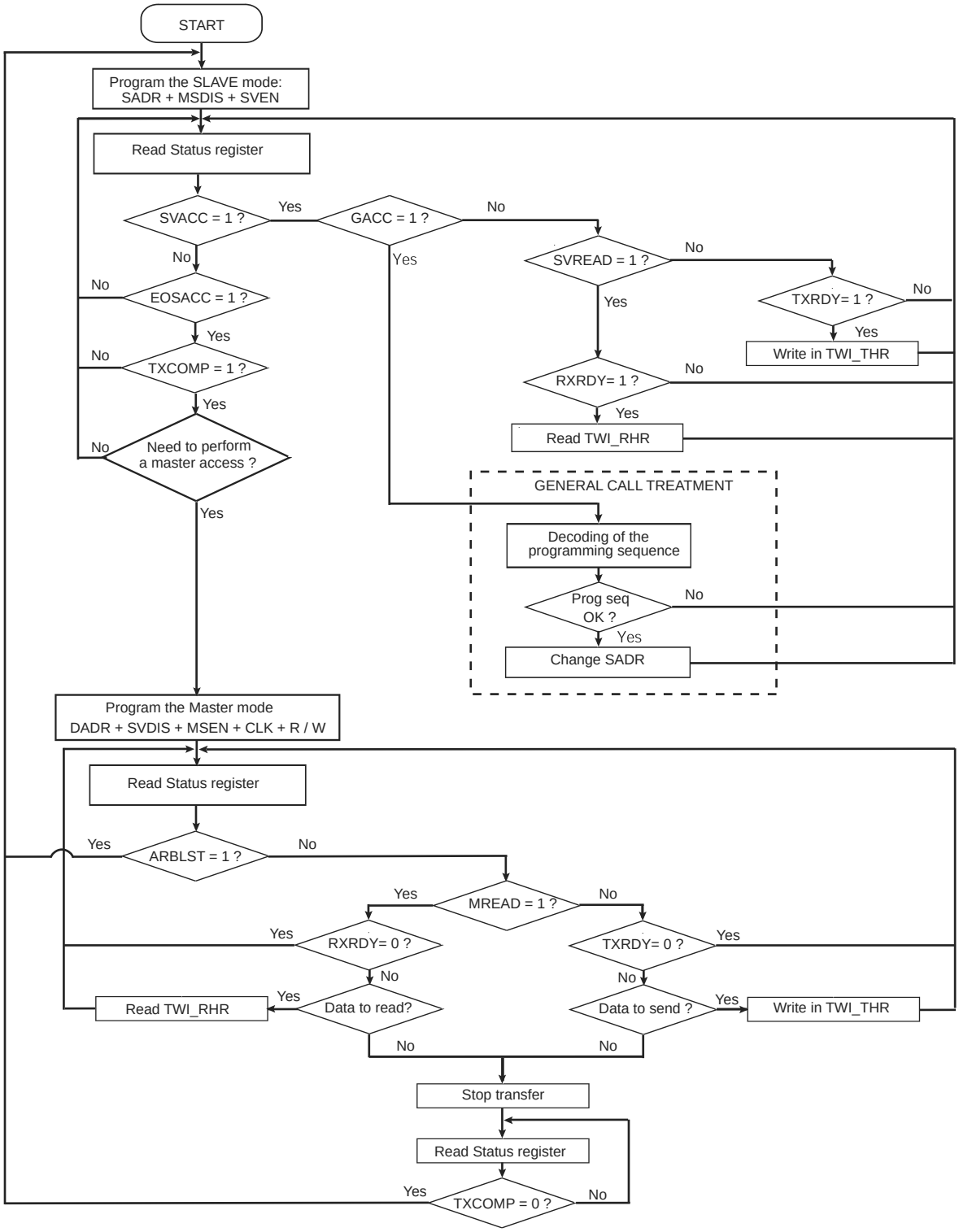
1: Disables the Power Saving Mode.

SWRST: Software Reset

0: No effect.

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

Figure 40-21: Multi-master Flowchart



- Data Receive with the DMA

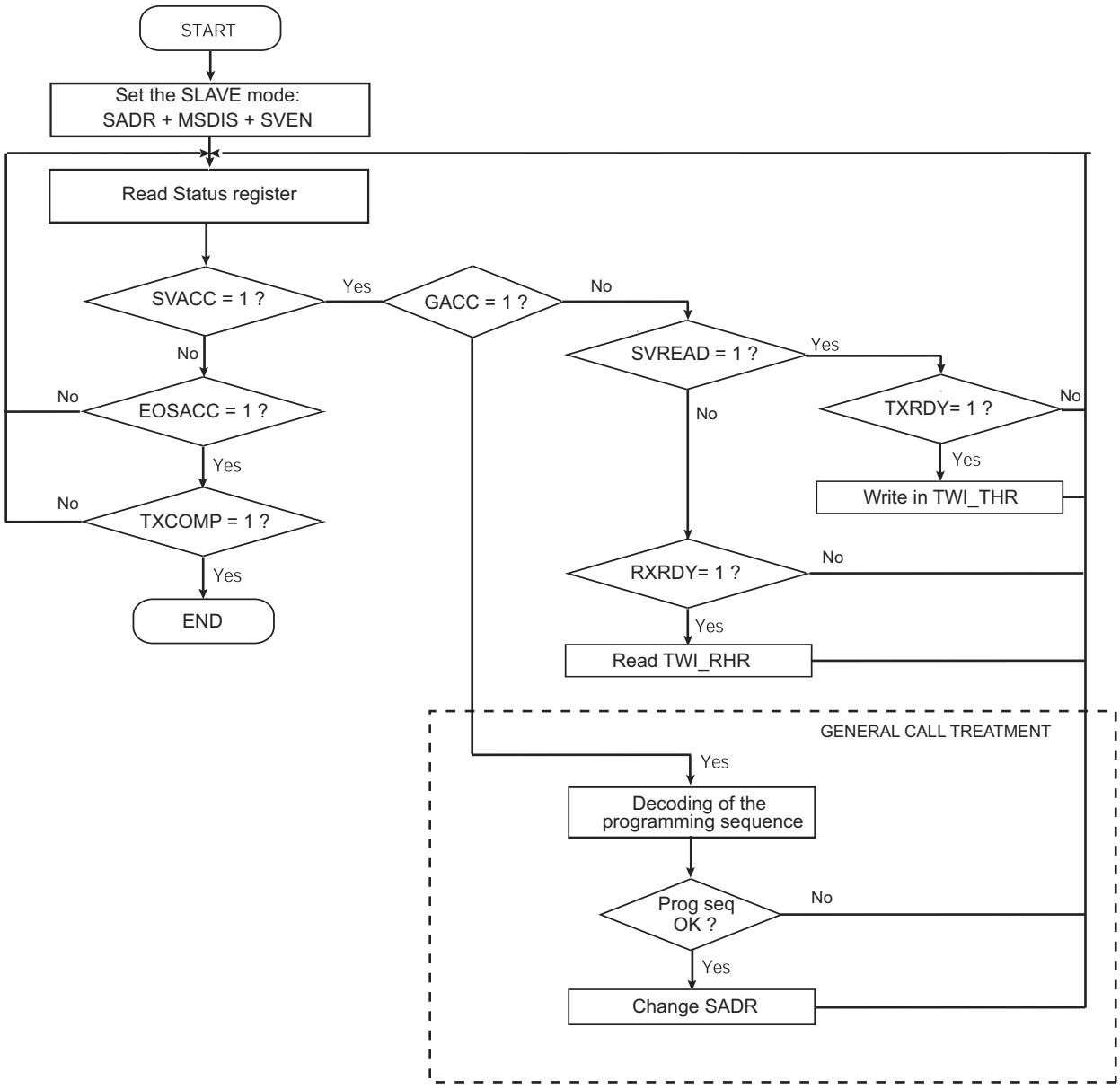
The DMA transfer size must be defined with the buffer size. In Slave mode, the number of characters to be received must be known in order to configure the DMA.

1. Initialize the DMA (channels, memory pointers, size, etc.).
2. Configure the Slave mode.
3. Enable the DMA.
4. Wait for the DMA buffer transfer complete flag.
5. Disable the DMA.
6. (Only if peripheral clock must be disabled) Wait for the TXCOMP flag to be raised in TWI_SR.

40.7.5.6 Read Write Flowcharts

The flowchart shown in Figure 40-29 gives an example of read and write operations in Slave mode. A polling or interrupt method can be used to check the status bits. The interrupt method requires that the Interrupt Enable Register (TWI_IER) be configured first.

Figure 40-29: Read Write Flowchart in Slave Mode



40.8.1 TWI Control Register

Name: TWI_CR

Address: 0xF8014000 (0), 0xF8018000 (1), 0xF8024000 (2), 0xFC038000 (3)

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
SWRST	–	SVDIS	SVEN	MSDIS	MSEN	STOP	START

START: Send a START Condition

0: No effect.

1: A frame beginning with a START bit is transmitted according to the features defined in the TWI Master Mode Register (TWI_MMR).

This action is necessary for the TWI to read data from a slave. When configured in Master mode with a write operation, a frame is sent as soon as the user writes a character in the Transmit Holding Register (TWI_THR).

STOP: Send a STOP Condition

0: No effect.

1: STOP condition is sent just after completing the current byte transmission in Master read mode.

- In single data byte master read, the START and STOP must both be set.
- In multiple data bytes master read, the STOP must be set after the last data received but one.
- In Master read mode, if a NACK bit is received, the STOP is automatically performed.
- In multiple data write operation, when both THR and internal shifter are empty, a STOP condition is sent automatically.

MSEN: TWI Master Mode Enabled

0: No effect.

1: Enables the Master mode (MSDIS must be written to 0).

Note: Switching from Slave to Master mode is only permitted when TXCOMP = 1.

MSDIS: TWI Master Mode Disabled

0: No effect.

1: The Master mode is disabled, all pending data is transmitted. The shifter and holding characters (if it contains data) are transmitted in case of write operation. In read operation, the character being transferred must be completely received before disabling.

SVEN: TWI Slave Mode Enabled

0: No effect.

1: Enables the Slave mode (SVDIS must be written to 0)

Note: Switching from master to Slave mode is only permitted when TXCOMP = 1.

44.7.12 USART Channel Status Register (SPI_MODE)

Name:US_CSR (SPI_MODE)

Address:0xF802C014 (0), 0xF8030014 (1), 0xFC008014 (2), 0xFC00C014 (3), 0xFC010014 (4)

Access: Read-only

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

This configuration is relevant only if USART_MODE = 0xE or 0xF in the USART Mode Register.

RXRDY: Receiver Ready (cleared by reading US_RHR)

0: No complete character has been received since the last read of US_RHR or the receiver is disabled. If characters were being received when the receiver was disabled, RXRDY changes to 1 when the receiver is enabled.

1: At least one complete character has been received and US_RHR has not yet been read.

TXRDY: Transmitter Ready (cleared by writing US_THR)

0: A character is in the US_THR waiting to be transferred to the Transmit Shift Register or the transmitter is disabled. As soon as the transmitter is enabled, TXRDY becomes 1.

1: There is no character in the US_THR.

OVRE: Overrun Error (cleared by writing a one to bit US_CR.RSTSTA)

0: No overrun error has occurred since the last RSTSTA.

1: At least one overrun error has occurred since the last RSTSTA.

TXEMPTY: Transmitter Empty (cleared by writing US_THR)

0: There are characters in either US_THR or the Transmit Shift Register, or the transmitter is disabled.

1: There are no characters in US_THR, nor in the Transmit Shift Register.

UNRE: Underrun Error (cleared by writing a one to bit US_CR.RSTSTA)

0: No SPI underrun error has occurred since the last RSTSTA.

1: At least one SPI underrun error has occurred since the last RSTSTA.

NSSE: NSS Line (Driving CTS Pin) Rising or Falling Edge Event (cleared on read)

0: No NSS line event has been detected since the last read of US_CSR.

1: A rising or falling edge event has been detected on NSS line since the last read of US_CSR.

NSS: Image of NSS Line

0: NSS line is driven low (if NSSE = 1, falling edge occurred on NSS line).

1: NSS line is driven high (if NSSE = 1, rising edge occurred on NSS line).

44.7.15 USART Baud Rate Generator Register

Name:US_BRGR

Address:0xF802C020 (0), 0xF8030020 (1), 0xFC008020 (2), 0xFC00C020 (3), 0xFC010020 (4)

Access: Read/Write

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

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

15	14	13	12	11	10	9	8
CD							

7	6	5	4	3	2	1	0
CD							

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

CD: Clock Divider

	USART_MODE ≠ ISO7816			
	SYNC = 0		**SYNC = 1 or USART_MODE = SPI (Master or Slave)**	**USART_MODE = ISO7816**
CD	**OVER = 0**	**OVER = 1**		
0	Baud Rate Clock Disabled			
1 to 65535	CD = Selected Clock / (16 × Baud Rate)	CD = Selected Clock / (8 × Baud Rate)	CD = Selected Clock / Baud Rate	CD = Selected Clock / (FI_DI_RATIO × Baud Rate)

FP: Fractional Part

0: Fractional divider is disabled.

1–7: Baud rate resolution, defined by FP × 1/8.

Warning: When the value of field FP is greater than 0, the SCK (oversampling clock) generates nonconstant duty cycles. The SCK high duration is increased by “selected clock” period from time to time. The duty cycle depends on the value of the CD field.