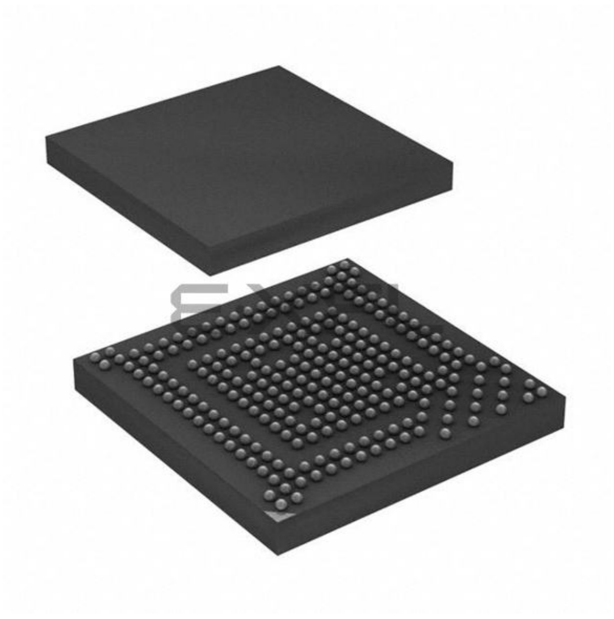
Microchip Technology - AT91SAM9G25-CFUR Datasheet



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	ARM926EJ-S
Number of Cores/Bus Width	1 Core, 32-Bit
Speed	400MHz
Co-Processors/DSP	-
RAM Controllers	DDR2, SDRAM, SRAM
Graphics Acceleration	No
Display & Interface Controllers	-
Ethernet	10/100Mbps
SATA	-
USB	USB 2.0 (3)
Voltage - I/O	1.8V, 3.3V
Operating Temperature	-40°C ~ 85°C (TA)
Security Features	-
Package / Case	247-LFBGA
Supplier Device Package	247-BGA (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/at91sam9g25-cfur

12.9.11 AIC Interrupt Clear Command Register

Name: AIC_ICCR

Address: 0xFFFFF128

Access: Write-only

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

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

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

7	6	5	4	3	2	1	0
PID7	PID6	PID5	PID4	PID3	PID2	SYS	FIQ

- **FIQ: Interrupt Clear**

0: No effect.

1: Clears corresponding interrupt.

- **SYS: Interrupt Clear**

0: No effect.

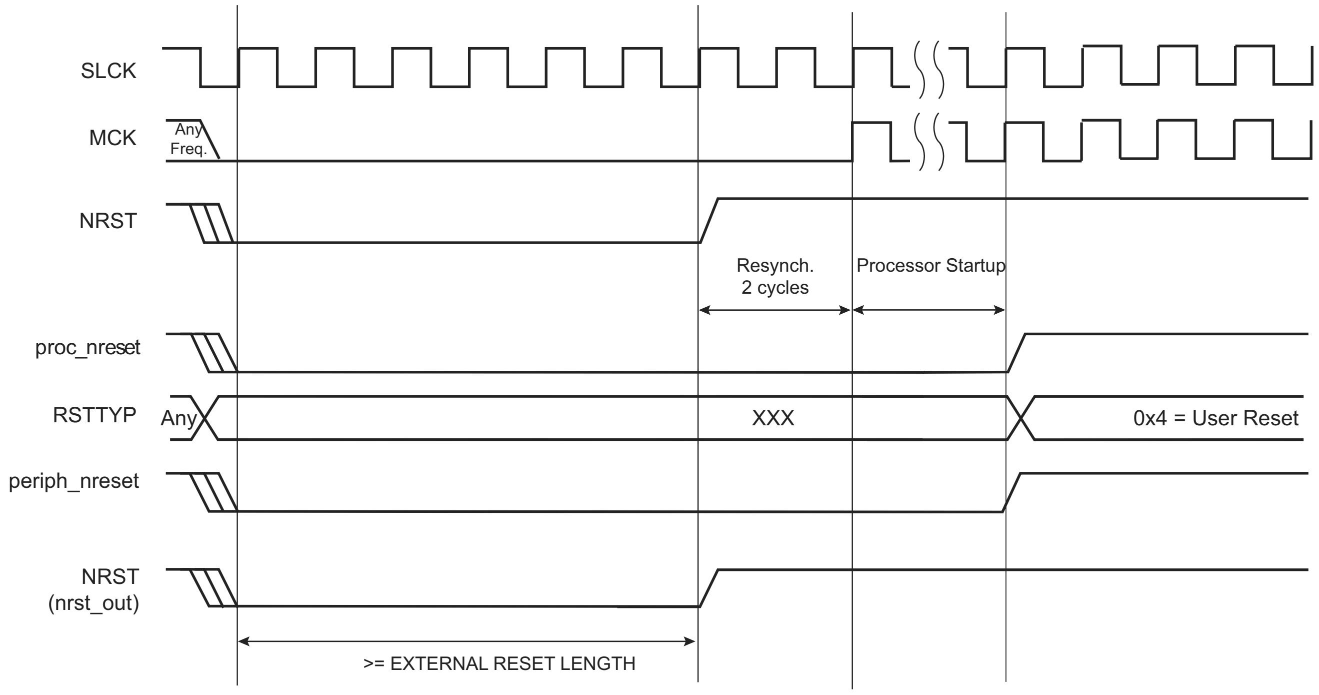
1: Clears corresponding interrupt.

- **PID2–PID31: Interrupt Clear**

0: No effect.

1: Clears corresponding interrupt.

Figure 13-6. User Reset State



13.4.4.4 Software Reset

The Reset Controller offers several commands used to assert the different reset signals. These commands are performed by writing the Control Register (RSTC_CR) with the following bits at 1:

- PROCRST: Writing a 1 to PROCRST resets the processor and the watchdog timer.
- PERRST: Writing a 1 to PERRST resets all the embedded peripherals, including the memory system, and, in particular, the Remap Command. The Peripheral Reset is generally used for debug purposes.
  PERRST must always be used in conjunction with PROCRST (PERRST and PROCRST bot set to 1 simultaneously.)
- EXTRST: Writing a 1 to EXTRST asserts low the NRST pin during a time defined by the field ERSTL in the Mode Register (RSTC_MR).

The software reset is entered if at least one of these bits is set by the software. All these commands can be performed independently or simultaneously. The software reset lasts 3 Slow Clock cycles.

The internal reset signals are asserted as soon as the register write is performed. This is detected on the Master Clock (MCK). They are released when the software reset is left, i.e., synchronously to SLCK.

If EXTRST is set, the nrst_out signal is asserted depending on the programming of the field ERSTL. However, the resulting falling edge on NRST does not lead to a User Reset.

If and only if the PROCRST bit is set, the reset controller reports the software status in the field RSTTYP of the RSTC_SR. Other software resets are not reported in RSTTYP.

As soon as a software operation is detected, the bit SRCMP (Software Reset Command in Progress) is set in the RSTC_SR. It is cleared as soon as the software reset is left. No other software reset can be performed while the SRCMP bit is set, and writing any value in the RSTC_CR has no effect.

14.6.3 RTC Time Register

Name: RTC_TIMR

Address: 0xFFFFFEB8

Access: Read/Write

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

23	22	21	20	19	18	17	16
–	AMPM	HOUR					

15	14	13	12	11	10	9	8
–	MIN						

7	6	5	4	3	2	1	0
–	SEC						

- **SEC: Current Second**

The range that can be set is 0–59 (BCD).

The lowest four bits encode the units. The higher bits encode the tens.

- **MIN: Current Minute**

The range that can be set is 0–59 (BCD).

The lowest four bits encode the units. The higher bits encode the tens.

- **HOUR: Current Hour**

The range that can be set is 1–12 (BCD) in 12-hour mode or 0–23 (BCD) in 24-hour mode.

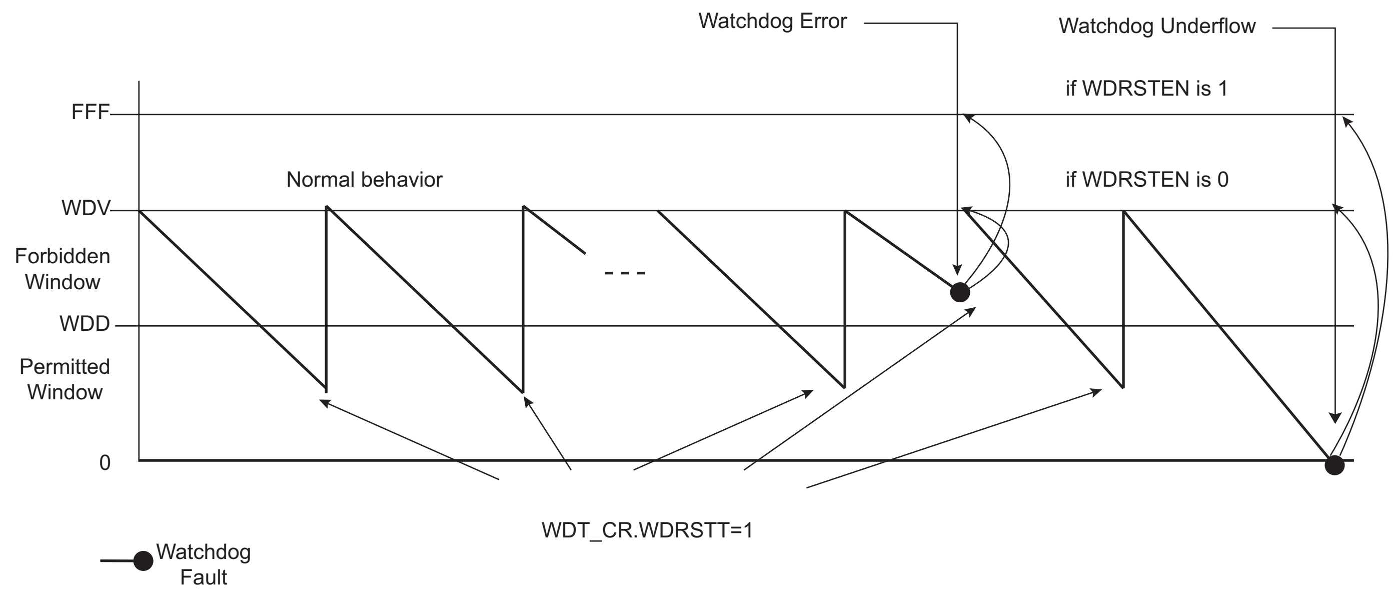
- **AMPM: Ante Meridiem Post Meridiem Indicator**

This bit is the AM/PM indicator in 12-hour mode.

0: AM.

1: PM.

Figure 16-2. Watchdog Behavior



21.16.4 PMC Peripheral Clock Enable Register

Name: PMC_PCER

Address: 0xFFFFFC10

Access: Write-only

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

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

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

7	6	5	4	3	2	1	0
PID7	PID6	PID5	PID4	PID3	PID2	–	–

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

- **PIDx: Peripheral Clock x Enable**

0: No effect.

1: Enables the corresponding peripheral clock.

Note: PID2 to PID31 refer to identifiers as defined in the section “Peripheral Identifiers”.

Note: Programming the control bits of the Peripheral ID that are not implemented has no effect on the behavior of the PMC.

22.6.14 PIO Interrupt Enable Register

Name: PIO_IER

Address: 0xFFFFF440 (PIOA), 0xFFFFF640 (PIOB), 0xFFFFF840 (PIOC), 0xFFFFFA40 (PIOD)

Access: Write-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: Input Change Interrupt Enable**

0: No effect.

1: Enables the input change interrupt on the I/O line.

– A21 <=> PD[2] <=> DELAY7[23:20]
– A22 <=> PD[3] <=> DELAY7[27:24]

25.5.3.4 Power supplies

The product embeds a dual power supply for EBI: VDDNF for NAND Flash signals and VDDIOM for others. This makes it possible to use a 1.8V or 3.3V NAND Flash independently of the SDRAM power supply.

The switch NFD0_ON_D16 is used to select the NAND Flash path on D0–D15 or D16–D31 depending on memory power supplies. This switch is located in the CCFG_EBICSA register in the Bus Matrix.

Figure 25-2 illustrates an example of the NAND Flash and the external RAM (DDR2 or LP-DDR or 16-bit LP-SDR) in the same power supply range (NFD0_ON_D16 = default).

Figure 25-2. NAND Flash and External RAM in Same Power Supply Range (NFD0_ON_D16 = default)

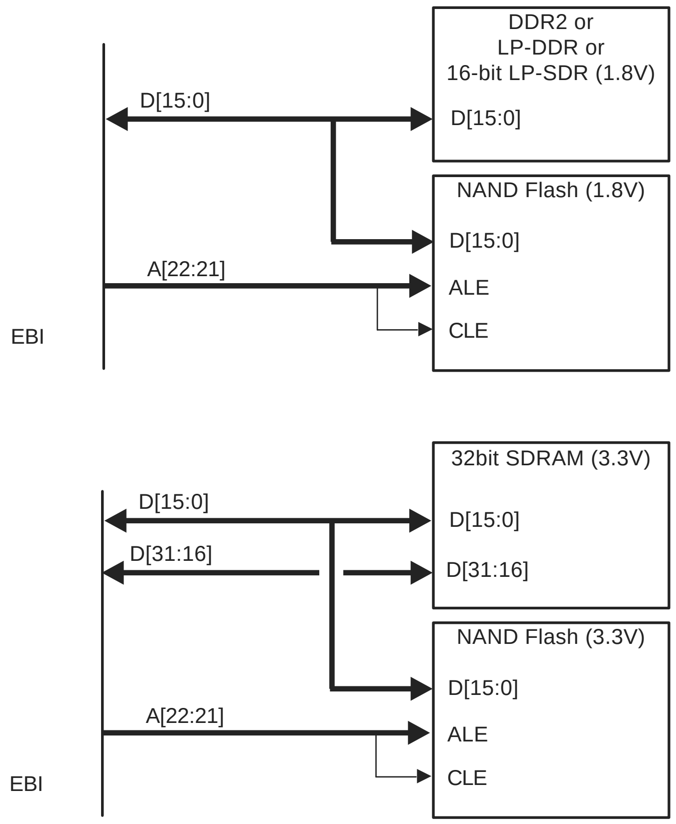
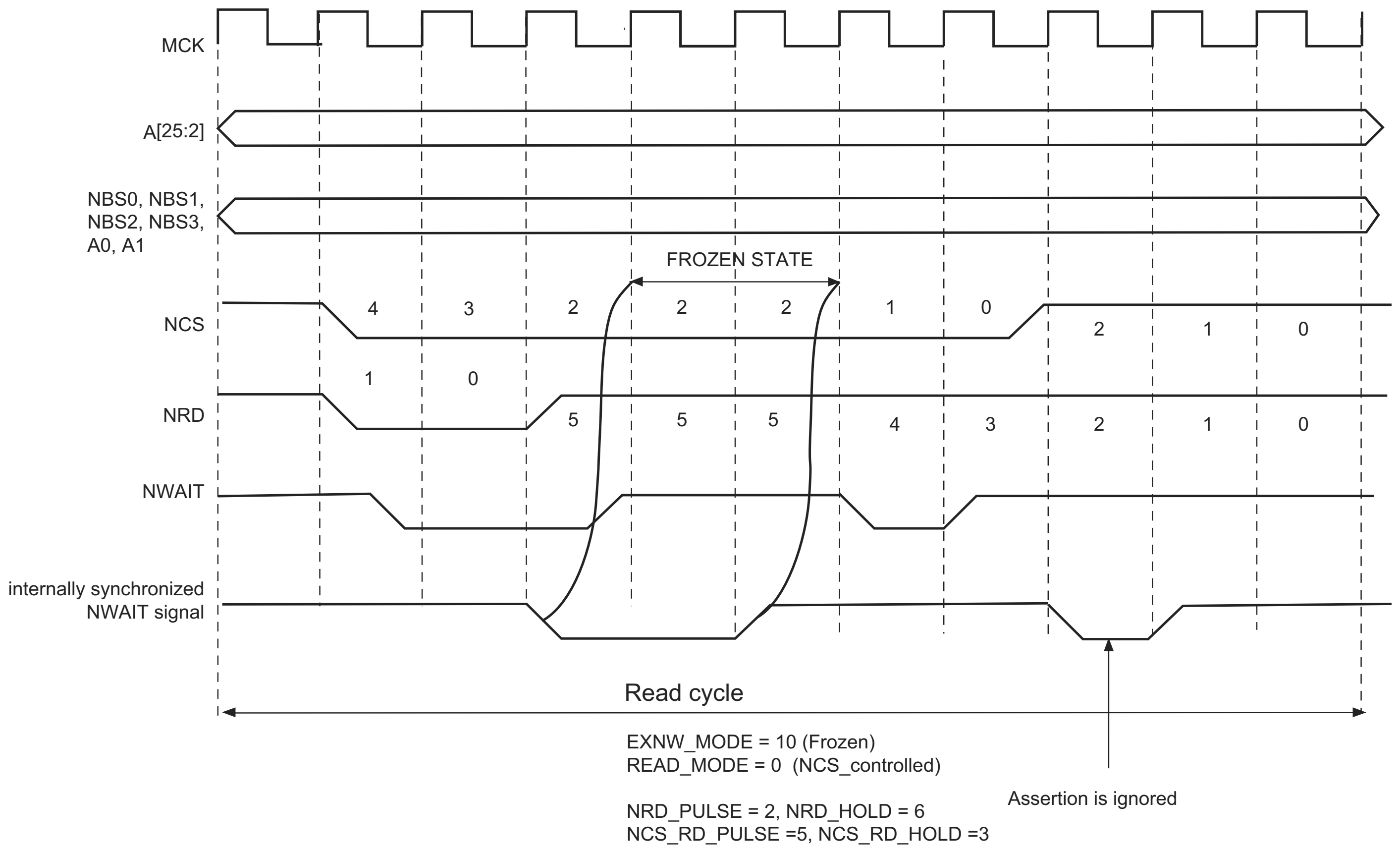


Figure 25-3 illustrates an example of the NAND Flash and the external RAM (DDR2 or LP-DDR or 16-bit LP-SDR) **not** in the same power supply range (NFD0_ON_D16 = 1).

This can be used if the SMC connects to the NAND Flash only. Using this function with another device on the SMC will lead to an unpredictable behavior of that device. In that case, the default value must be selected.

Figure 28-27. Read Access with NWAIT Assertion in Frozen Mode (EXNW_MODE = 10)


MCK
A[25:2]
NBS0, NBS1, NBS2, NBS3, A0, A1
FROZEN STATE
NCS
4
3
2
2
2
1
0
2
1
0
NRD
1
0
5
5
5
4
3
2
1
0
NWAIT
internally synchronized NWAIT signal
Read cycle
EXNW_MODE = 10 (Frozen)
READ_MODE = 0 (NCS_controlled)
NRD_PULSE = 2, NRD_HOLD = 6
NCS_RD_PULSE =5, NCS_RD_HOLD =3
Assertion is ignored

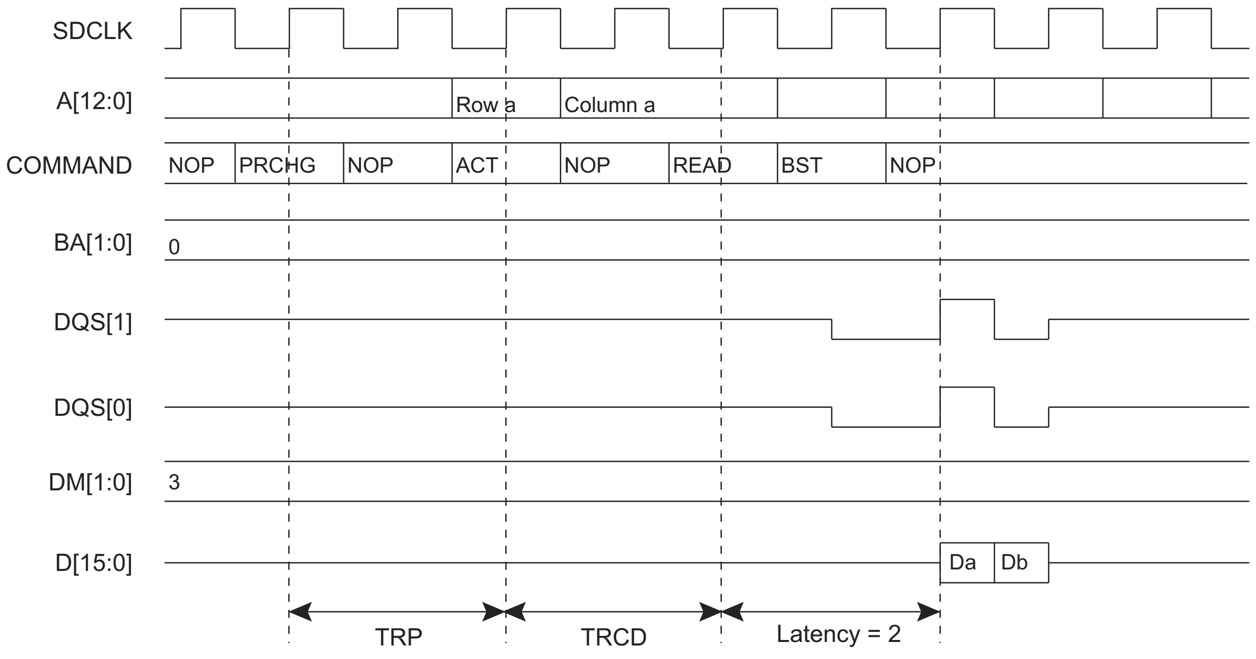
For a definition of timing parameters, refer to Section 29.7.3 “DDRSDRC Configuration Register” on page 462.

Read accesses to the SDRAM are burst oriented and the burst length is programmed to 8. It determines the maximum number of column locations that can be accessed for a given read command. When the read command is issued, eight columns are selected. All accesses for that burst take place within these eight columns, meaning that the burst wraps within these eight columns if the boundary is reached. These eight columns are selected by addr[13:3]; addr[2:0] is used to select the starting location within the block.

In the case of incrementing burst (INCR/INCR4/INCR8/INCR16), the addresses can cross the 16-byte boundary of the SDRAM device. For example, when a transfer (INCR4) starts at address 0x0C, the next access is 0x10, but since the burst length is programmed to 8, the next access is 0x00. Since the boundary is reached, the burst wraps. The DDRSDRC takes into account this feature of the SDRAM device. In the case of DDR-SDRAM devices, transfers start at address 0x04/0x08/0x0C. In the case of SDR-SDRAM devices, transfers start at address 0x14/0x18/0x1C. Two read commands are issued to avoid wrapping when the boundary is reached. The last read command may generate additional reading (1 read cmd = 4 DDR words or 1 read cmd = 8 SDR words).

To avoid additional reading, it is possible to use the burst stop command to truncate the read burst and to decrease power consumption.

Figure 29-11. Single Read Access, Row Closed, Latency = 2, Low-power DDR1-SDRAM Device



31.6.14.2 Not Powered State

Self powered devices can detect 5V VBUS using a PIO. When the device is not connected to a host, device power consumption can be reduced by the DETACH bit in UDPHS_CTRL. Disabling the transceiver is automatically done. HSDM, HSDP, FSDP and FSDP lines are tied to GND pull-downs integrated in the hub downstream ports.

31.6.14.3 Entering Attached State

When no device is connected, the USB FSDP and FSDM signals are tied to GND by 15 KΩ pull-downs integrated in the hub downstream ports. When a device is attached to an hub downstream port, the device connects a 1.5 KΩ pull-up on FSDP. The USB bus line goes into IDLE state, FSDP is pulled-up by the device 1.5 KΩ resistor to 3.3V and FSDM is pulled-down by the 15 KΩ resistor to GND of the host.

After pull-up connection, the device enters the powered state. The transceiver remains disabled until bus activity is detected.

In case of low power consumption need, the device can be stopped. When the device detects the VBUS, the software must enable the USB transceiver by enabling the EN_UDPHS bit in UDPHS_CTRL register.

The software can detach the pull-up by setting DETACH bit in UDPHS_CTRL register.

31.6.14.4 From Powered State to Default State (Reset)

After its connection to a USB host, the USB device waits for an end-of-bus reset. The unmasked flag ENDRESET is set in the UDPHS_IEN register and an interrupt is triggered.

Once the ENDRESET interrupt has been triggered, the device enters Default State. In this state, the UDPHS software must:

- Enable the default endpoint, setting the EPT_ENABL flag in the UDPHS_EPTCTLENB[0] register and, optionally, enabling the interrupt for endpoint 0 by writing 1 in EPT_0 of the UDPHS_IEN register. The enumeration then begins by a control transfer.
- Configure the Interrupt Mask Register which has been reset by the USB reset detection
- Enable the transceiver.

In this state, the EN_UDPHS bit in UDPHS_CTRL register must be enabled.

31.6.14.5 From Default State to Address State (Address Assigned)

After a Set Address standard device request, the USB host peripheral enters the address state.

Warning: before the device enters address state, it must achieve the Status IN transaction of the control transfer, i.e., the UDPHS device sets its new address once the TX_COMPLT flag in the UDPHS_EPTCTL[0] register has been received and cleared.

To move to address state, the driver software sets the DEV_ADDR field and the FADDR_EN flag in the UDPHS_CTRL register.

31.6.14.6 From Address State to Configured State (Device Configured)

Once a valid Set Configuration standard request has been received and acknowledged, the device enables endpoints corresponding to the current configuration. This is done by setting the BK_NUMBER, EPT_TYPE, EPT_DIR and EPT_SIZE fields in the UDPHS_EPTCFGx registers and enabling them by setting the EPT_ENABL flag in the UDPHS_EPTCTLENBx registers, and, optionally, enabling corresponding interrupts in the UDPHS_IEN register.

31.6.14.7 Entering Suspend State (Bus Activity)

When a Suspend (no bus activity on the USB bus) is detected, the DET_SUSPD signal in the UDPHS_STA register is set. This triggers an interrupt if the corresponding bit is set in the UDPHS_IEN register. This flag is cleared by writing to the UDPHS_CLRINT register. Then the device enters Suspend mode.

The structure of commands, responses and data blocks is described in the High Speed MultiMedia Card System Specification. See also Table 33-6 on page 620.

High Speed MultiMedia Card bus data transfers are composed of these tokens.

There are different types of operations. Addressed operations always contain a command and a response token. In addition, some operations have a data token; the others transfer their information directly within the command or response structure. In this case, no data token is present in an operation. The bits on the DAT and the CMD lines are transferred synchronous to the clock HSMCI clock.

Two types of data transfer commands are defined:

- Sequential commands—These commands initiate a continuous data stream. They are terminated only when a stop command follows on the CMD line. This mode reduces the command overhead to an absolute minimum.
- Block-oriented commands—These commands send a data block succeeded by CRC bits.

Both read and write operations allow either single or multiple block transmission. A multiple block transmission is terminated when a stop command follows on the CMD line similarly to the sequential read or when a multiple block transmission has a predefined block count (see Section 33.8.2 “Data Transfer Operation”).

The HSMCI provides a set of registers to perform the entire range of High Speed MultiMedia Card operations.

33.8.1 Command - Response Operation

After reset, the HSMCI is disabled and becomes valid after setting the MCIEN bit in the HSMCI_CR.

The PWSEN bit saves power by dividing the HSMCI clock by 2^{PWSDIV} + 1 when the bus is inactive.

The two bits, RDPROOF and WRPROOF in the HSMCI Mode Register (HSMCI_MR) allow stopping the HSMCI clock during read or write access if the internal FIFO is full. This will guarantee data integrity, not bandwidth.

All the timings for High Speed MultiMedia Card are defined in the High Speed MultiMedia Card System Specification.

The two bus modes (open drain and push/pull) needed to process all the operations are defined in the HSMCI Command Register (HSMCI_CMDR). The HSMCI_CMDR allows a command to be carried out.

For example, to perform an ALL_SEND_CID command:

	Host Command					N_{ID} Cycles			Response			High Impedance State		
CMD	S	T	Content	CRC	E	Z	******	Z	S	T	CID Content	Z	Z	Z

The command ALL_SEND_CID and the fields and values for the HSMCI_CMDR are described in Table 33-6 and Table 33-7.

Table 33-6. ALL_SEND_CID Command Description

CMD Index	Type	Argument	Response	Abbreviation	Command Description
CMD2	bcr(1)	[31:0] stuff bits	R2	ALL_SEND_CID	Asks all cards to send their CID numbers on the CMD line

Note: 1. bcr means broadcast command with response.

33.14 High Speed MultiMedia Card Interface (HSMCI) User Interface

Table 33-8. Register Mapping

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

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

35.7.9 TC Status Register

Name: TC_SRx [x=0..2]

Address: 0xF8008020 (0)[0], 0xF8008060 (0)[1], 0xF80080A0 (0)[2], 0xF800C020 (1)[0], 0xF800C060 (1)[1], 0xF800C0A0 (1)[2]

Access: Read-only

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

23	22	21	20	19	18	17	16
–	–	–	–	–	MTIOB	MTIOA	CLKSTA

15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–

7	6	5	4	3	2	1	0
ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS

- **COVFS: Counter Overflow Status (cleared on read)**

0: No counter overflow has occurred since the last read of the Status Register.

1: A counter overflow has occurred since the last read of the Status Register.

- **LOVRS: Load Overrun Status (cleared on read)**

0: Load overrun has not occurred since the last read of the Status Register or TC_CMRx.WAVE = 1.

1: RA or RB have been loaded at least twice without any read of the corresponding register since the last read of the Status Register, if TC_CMRx.WAVE = 0.

- **CPAS: RA Compare Status (cleared on read)**

0: RA Compare has not occurred since the last read of the Status Register or TC_CMRx.WAVE = 0.

1: RA Compare has occurred since the last read of the Status Register, if TC_CMRx.WAVE = 1.

- **CPBS: RB Compare Status (cleared on read)**

0: RB Compare has not occurred since the last read of the Status Register or TC_CMRx.WAVE = 0.

1: RB Compare has occurred since the last read of the Status Register, if TC_CMRx.WAVE = 1.

- **CPCS: RC Compare Status (cleared on read)**

0: RC Compare has not occurred since the last read of the Status Register.

1: RC Compare has occurred since the last read of the Status Register.

- **LDRAS: RA Loading Status (cleared on read)**

0: RA Load has not occurred since the last read of the Status Register or TC_CMRx.WAVE = 1.

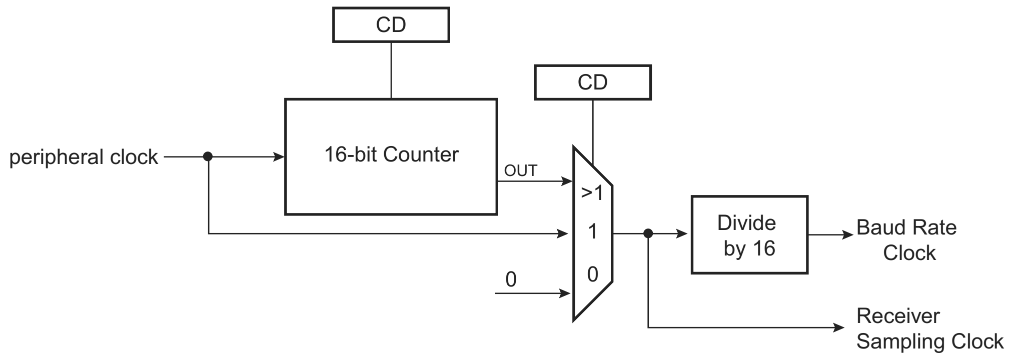
1: RA Load has occurred since the last read of the Status Register, if TC_CMRx.WAVE = 0.

- **LDRBS: RB Loading Status (cleared on read)**

0: RB Load has not occurred since the last read of the Status Register or TC_CMRx.WAVE = 1.

1: RB Load has occurred since the last read of the Status Register, if TC_CMRx.WAVE = 0.

Figure 39-2. Baud Rate Generator



39.5.2 Receiver

39.5.2.1 Receiver Reset, Enable and Disable

After device reset, the UART receiver is disabled and must be enabled before being used. The receiver can be enabled by writing the Control Register (UART_CR) with the bit RXEN at 1. At this command, the receiver starts looking for a start bit.

The programmer can disable the receiver by writing UART_CR with the bit RXDIS at 1. If the receiver is waiting for a start bit, it is immediately stopped. However, if the receiver has already detected a start bit and is receiving the data, it waits for the stop bit before actually stopping its operation.

The receiver can be put in reset state by writing UART_CR with the bit RSTRX at 1. In this case, the receiver immediately stops its current operations and is disabled, whatever its current state. If RSTRX is applied when data is being processed, this data is lost.

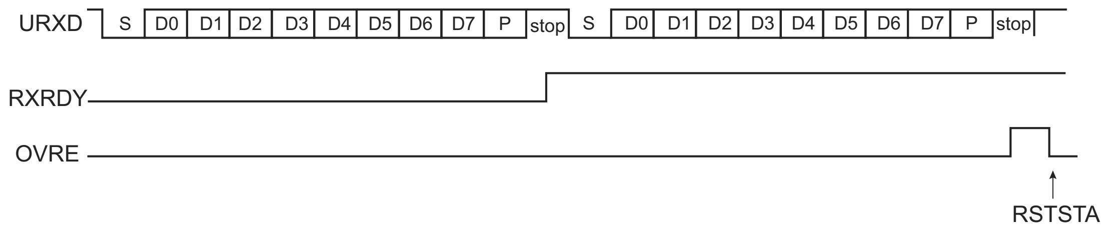
39.5.2.2 Start Detection and Data Sampling

The UART only supports asynchronous operations, and this affects only its receiver. The UART receiver detects the start of a received character by sampling the URXD signal until it detects a valid start bit. A low level (space) on URXD is interpreted as a valid start bit if it is detected for more than seven cycles of the sampling clock, which is 16 times the baud rate. Hence, a space that is longer than 7/16 of the bit period is detected as a valid start bit. A space which is 7/16 of a bit period or shorter is ignored and the receiver continues to wait for a valid start bit.

When a valid start bit has been detected, the receiver samples the URXD at the theoretical midpoint of each bit. It is assumed that each bit lasts 16 cycles of the sampling clock (1-bit period) so the bit sampling point is eight cycles (0.5-bit period) after the start of the bit. The first sampling point is therefore 24 cycles (1.5-bit periods) after detecting the falling edge of the start bit.

Each subsequent bit is sampled 16 cycles (1-bit period) after the previous one.

Figure 39-3. Start Bit Detection



40.7.12 ADC Overrun Status Register

Name: ADC_OVER

Address: 0xF804C03C

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**
–	–	–	–	OVRE11	OVRE10	OVRE9	OVRE8
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
OVRE7	OVRE6	OVRE5	OVRE4	OVRE3	OVRE2	OVRE1	OVRE0

- **OVREx: Overrun Error x**

0: No overrun error on the corresponding channel since the last read of ADC_OVER.

1: An overrun error has occurred on the corresponding channel since the last read of ADC_OVER.

40.7.16 ADC Trigger Register

Name: ADC_TRGR

Address: 0xF804C0C0

Access: Read/Write

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

- **TRGMOD: Trigger Mode**

Value	Name	Description
0	NO_TRIGGER	No trigger, only software trigger can start conversions
1	EXT_TRIG_RISE	External trigger rising edge
2	EXT_TRIG_FALL	External trigger falling edge
3	EXT_TRIG_ANY	External trigger any edge
4	–	Reserved
5	PERIOD_TRIG	ADC internal periodic trigger (see field TRGPER)
6	CONTINUOUS	Continuous Mode

- **TRGPER: Trigger Period**

Effective only if TRGMOD defines a periodic trigger.

Defines the periodic trigger period, with the following equation:

Trigger Period = (TRGPER + 1) / ADCCLK

The minimum time between two consecutive trigger events must be strictly greater than the duration time of the longest conversion sequence depending on the configuration of registers ADC_MR, ADC_CHSR, ADC_SEQRx .

- **CRC_ERR: CRC Synchronization Error**

0: The CRC Synchronization Error interrupt is disabled.

1: The CRC Synchronization Error interrupt is enabled.

- **FR_OVR: Frame Rate Overrun**

0: The Frame Rate Overrun interrupt is disabled.

1: The Frame Rate Overrun is enabled.

44.6.21 Specific Address 3 Top Register

Name:	EMAC_SA3T
Address:	0xF802C0AC
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
ADDR							

7	6	5	4	3	2	1	0
ADDR							

- **ADDR: Specific Address 3 Top**

The most significant bits of the destination address, that is bits 47 to 32.

44.6.26.14 Receive Overrun Errors Register

Name: EMAC_ROV

Address: 0xF802C070

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
ROVR							

- **ROVR: Receive Overrun**

An 8-bit register counting the number of frames that are address recognized but were not copied to memory due to a receive DMA overrun.

Doc. Rev. 11052G	Comments
31-Aug-15	Section 37. “Two-wire Interface (TWI)” Instances of “shift register” replaced by “internal shifter” Updated Section 37.1 “Description” Table 37-1 “Atmel TWI Compatibility with I2C Standard”: added clock synchronization as a supported feature Updated Section 37.2 “Embedded Characteristics” Updated Figure 37-1 “Block Diagram” Removed section “Application Block Diagram” Updated Table 37-3 “I/O Lines Description” Section 37.6.2 “Power Management”: deleted bullet “Enable the peripheral clock” Section 37.7.3 “Master Mode” moved to Section 37.7 “Functional Description”; removed section “Application Block Diagram” Updated Section 37.7.3.2 “Programming Master Mode” and Section 37.7.3.3 “Master Transmitter Mode” Section 37.7.3.4 “Master Receiver Mode”: removed “after the STOP condition” from end of second sentence in second paragraph; removed reference to clock stretching in the “Warning” (clock stretching is a slave-only mechanism) Figure “Master Read Clock Stretching with Multiple Data Bytes” replaced by Figure 37-9 “Master Read Wait State with Multiple Data Bytes” Section 37.7.3.5 “Internal Address”: - changed ‘N’ to ‘NA’ as abbreviation for “Not Acknowledge” under “7-bit Slave Addressing” - at end of text under “10-bit Slave Addressing”, “byte write to an Atmel AT24LC512 EEPROM” replaced by “byte write to a memory device” Updated Section 37.7.3.6 “Using the DMA Controller” Added missing “Yes” and “No” in Figure 37-15 “TWI Write Operation with Multiple Data Bytes with or without Internal Address”, in Figure 37-17 “TWI Read Operation with Single Data Byte and Internal Address”, and in Figure 37-18 “TWI Read Operation with Multiple Data Bytes with or without Internal Address” Section 37.7.4 “Multi-master Mode” moved to Section 37.7 “Functional Description” Section 37.7.5 “Slave Mode” moved to Section 37.7 “Functional Description” and removed section “Application Block Diagram” Section 37.7.5.3 “Receiving Data”: - under “Read Sequence”, added sentence describing how to clear TXRDY flag - under “Clock Synchronization Sequence”, removed reference to TWI_THR - added “Clock Stretching Sequence” Section 37.7.5.4 “Data Transfer”: - changed heading “Clock Synchronization” to “Clock Synchronization/Stretching” and updated content - at end of last sentence under “Clock Synchronization in Write Mode”, changed “in Read mode” to “in Write mode” - corrected EOSVACC to EOSACC in Figure 37-22 “Read Access Ordered by a Master” and Figure 37-23 “Write Access Ordered by a Master” - corrected GCACC to GACC in Figure 37-24 “Master Performs a General Call” - replaced “SCL is stretched” with “TWCK is stretched” in Figure 37-26 “Clock Synchronization in Write Mode” Added Section 37.7.5.5 “Using the DMA Controller” Figure 37-29 “Read Write Flowchart in Slave Mode”: “SVREAD = 0” corrected to “SVREAD = 1”; “RXRDY = 0 ?” corrected to “RXRDY = 1 ?” Section 37.7.6 “Register Write Protection”: updated title (was “Write Protection System”) and revised content Table 37-7 “Register Mapping”: removed TWI_THR reset value; defined offset range 0x38–0xE0 as reserved