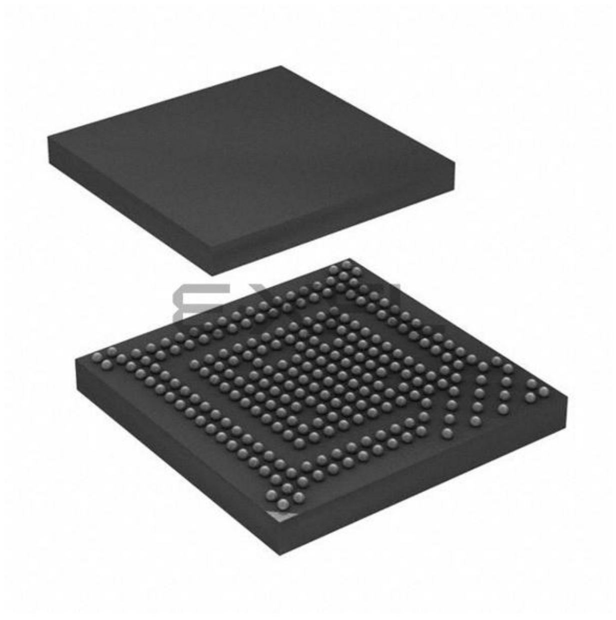
Microchip Technology - AT91SAM9G25-BFU-999 Datasheet



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-VFBGA
Supplier Device Package	247-VFBGA (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/at91sam9g25-bfu-999

Table 2-1. Signal Description List (Continued)

Signal Name	Function	Type	Active Level
External Bus Interface - EBI			
D0–D15	Data Bus	I/O	
D16–D31	Data Bus	I/O	
A0–A25	Address Bus	Output	
NWAIT	External Wait Signal	Input	Low
Static Memory Controller - SMC			
NCS0–NCS5	Chip Select Lines	Output	Low
NWR0–NWR3	Write Signal	Output	Low
NRD	Read Signal	Output	Low
NWE	Write Enable	Output	Low
NBS0–NBS3	Byte Mask Signal	Output	Low
NAND Flash Support			
NFD0–NFD16	NAND Flash I/O	I/O	
NANDCS	NAND Flash Chip Select	Output	Low
NANDOE	NAND Flash Output Enable	Output	Low
NANDWE	NAND Flash Write Enable	Output	Low
DDR2/SDRAM/LPDDR Controller			
SDCK,#SDCK	DDR2/SDRAM Differential Clock	Output	
SDCKE	DDR2/SDRAM Clock Enable	Output	High
SDCS	DDR2/SDRAM Controller Chip Select	Output	Low
BA[0..2]	Bank Select	Output	Low
SDWE	DDR2/SDRAM Write Enable	Output	Low
RAS-CAS	Row and Column Signal	Output	Low
SDA10	SDRAM Address 10 Line	Output	
DQS[0..1]	Data Strobe	I/O	
DQM[0..3]	Write Data Mask	Output	
High Speed MultiMedia Card Interface - HSMCI0–1			
MCI0_CK, MCI1_CK	Multimedia Card Clock	I/O	
MCI0_CDA, MCI1_CDA	Multimedia Card Slot Command	I/O	
MCI0_DA0–MCI0_DA3	Multimedia Card 0 Slot A Data	I/O	
MCI1_DA0–MCI1_DA3	Multimedia Card 1 Slot A Data	I/O	
Universal Synchronous Asynchronous Receiver Transmitter - USARTx			
SCKx	USARTx Serial Clock	I/O	
TXDx	USARTx Transmit Data	Output	
RXDx	USARTx Receive Data	Input	
RTSx	USARTx Request To Send	Output	
CTSx	USARTx Clear To Send	Input	

12.8.3.2 Interrupt Nesting

The priority controller utilizes interrupt nesting in order for the high priority interrupt to be handled during the service of lower priority interrupts. This requires the interrupt service routines of the lower interrupts to re-enable the interrupt at the processor level.

When an interrupt of a higher priority happens during an already occurring interrupt service routine, the nIRQ line is re-asserted. If the interrupt is enabled at the core level, the current execution is interrupted and the new interrupt service routine should read the AIC_IVR. At this time, the current interrupt number and its priority level are pushed into an embedded hardware stack, so that they are saved and restored when the higher priority interrupt servicing is finished and the AIC_EOICR is written.

The AIC is equipped with an 8-level wide hardware stack in order to support up to eight interrupt nestings pursuant to having eight priority levels.

12.8.3.3 Interrupt Vectoring

The interrupt handler addresses corresponding to each interrupt source can be stored in the registers AIC_SVR1 to AIC_SVR31 (Source Vector Register 1 to 31). When the processor reads AIC_IVR (Interrupt Vector Register), the value written into AIC_SVR corresponding to the current interrupt is returned.

This feature offers a way to branch in one single instruction to the handler corresponding to the current interrupt, as AIC_IVR is mapped at the absolute address 0xFFFFF100 and thus accessible from the ARM interrupt vector at address 0x00000018 through the following instruction:

```
LDR         PC,[PC,# -&F20]
```

When the processor executes this instruction, it loads the read value in AIC_IVR in its program counter, thus branching the execution on the correct interrupt handler.

This feature is often not used when the application is based on an operating system (either real time or not). Operating systems often have a single entry point for all the interrupts and the first task performed is to discern the source of the interrupt.

However, it is strongly recommended to port the operating system on AT91 products by supporting the interrupt vectoring. This can be performed by defining all the AIC_SVR of the interrupt source to be handled by the operating system at the address of its interrupt handler. When doing so, the interrupt vectoring permits a critical interrupt to transfer the execution on a specific very fast handler and not onto the operating system’s general interrupt handler. This facilitates the support of hard real-time tasks (input/outputs of voice/audio buffers and software peripheral handling) to be handled efficiently and independently of the application running under an operating system.

14.6.2 RTC Mode Register

Name: RTC_MR

Address: 0xFFFFFEB4

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**
–	–	–	–	–	–	–	HRMOD

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode Register (SYSC_WPMR).

- **HRMOD: 12-/24-hour Mode**

0: 24-hour mode is selected.

1: 12-hour mode is selected.

24.2.1 Matrix Masters

The Bus Matrix manages 12 masters, which means that each master can perform an access concurrently with others, depending on whether the slave it accesses is available.

Each master has its own decoder, which can be defined specifically for each master. In order to simplify the addressing, all the masters have the same decodings.

Table 24-1. List of Bus Matrix Masters

Master 0	ARM926 Instruction
Master 1	ARM926 Data
Master 2 & 3	DMA Controller 0
Master 4 & 5	DMA Controller 1
Master 6	UDP HS DMA
Master 7	UHP EHCI DMA
Master 8	UHP OHCI DMA
Master 9	ISI DMA
Master 10	EMAC DMA
Master 11	Reserved

24.2.2 Matrix Slaves

The Bus Matrix manages 10 slaves. Each slave has its own arbiter, thus allowing a different arbitration per slave to be programmed.

Table 24-2. List of Bus Matrix Slaves

Slave 0	Internal SRAM
Slave 1	Internal ROM
Slave 2	Soft Modem (SMD)
Slave 3	USB Device High Speed Dual Port RAM (DPR)
	USB Host EHCI registers
	USB Host OHCI registers
Slave 4	External Bus Interface
Slave 5	DDR2 port 1
Slave 6	DDR2 port 2
Slave 7	DDR2 port 3
Slave 8	Peripheral Bridge 0
Slave 9	Peripheral Bridge 1

27.5.7 Error Location Interrupt Disable Register

Name: PMERRLOC_ELIDR

Address: 0xFFFFE618

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**
–	–	–	–	–	–	–	DONE

- **DONE: Computation Terminated Interrupt Disable**

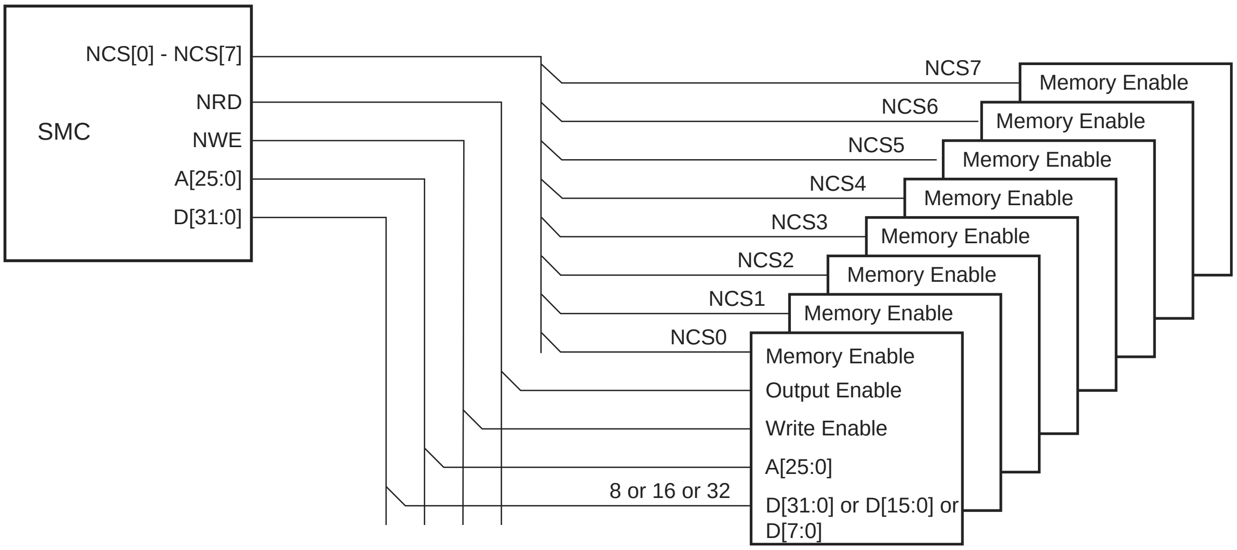
28.7 External Memory Mapping

The SMC provides up to 26 address lines, A[25:0]. This allows each chip select line to address up to 64 Mbytes of memory.

If the physical memory device connected on one chip select is smaller than 64 Mbytes, it wraps around and appears to be repeated within this space. The SMC correctly handles any valid access to the memory device within the page (see Figure 28-2).

A[25:0] is only significant for 8-bit memory, A[25:1] is used for 16-bit memory, A[25:2] is used for 32-bit memory.

Figure 28-2. Memory Connections for Eight External Devices



28.8 Connection to External Devices

28.8.1 Data Bus Width

A data bus width of 8, 16, or 32 bits can be selected for each chip select. This option is controlled by the field DBW in SMC_MODE (Mode Register) for the corresponding chip select.

Figure 28-3 shows how to connect a 512K x 8-bit memory on NCS2. Figure 28-4 shows how to connect a 512K x 16-bit memory on NCS2. Figure 28-5 shows two 16-bit memories connected as a single 32-bit memory

28.8.2 Byte Write or Byte Select Access

Each chip select with a 16-bit or 32-bit data bus can operate with one of two different types of write access: byte write or byte select access. This is controlled by the BAT field of the SMC_MODE register for the corresponding chip select.

Figure 29-4. Single Write Access, Row Closed, SDR-SDRAM Device

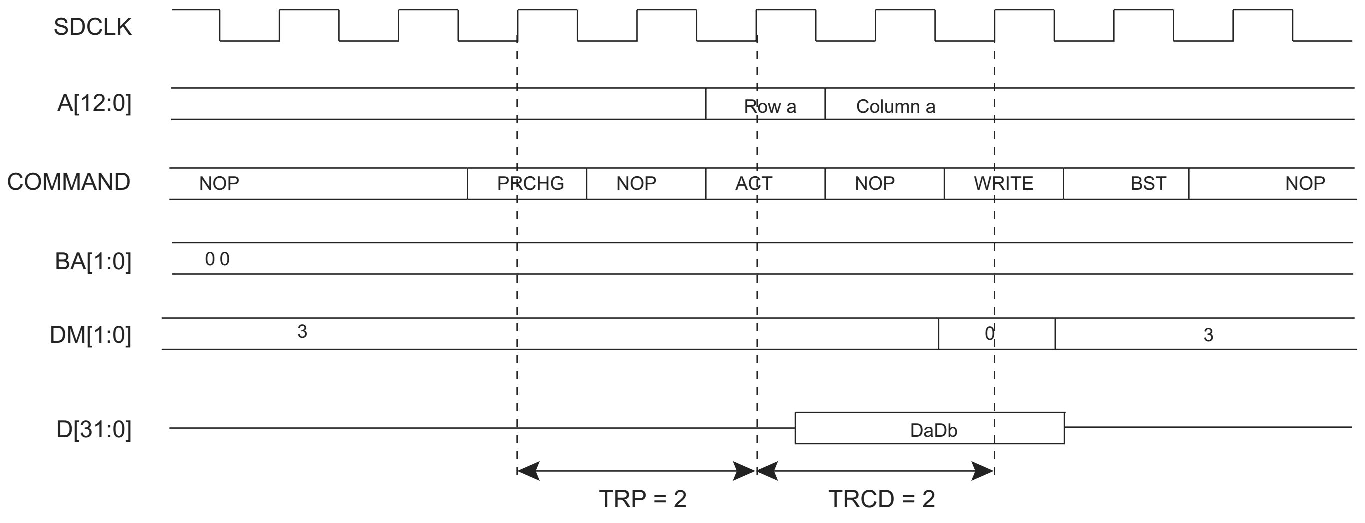
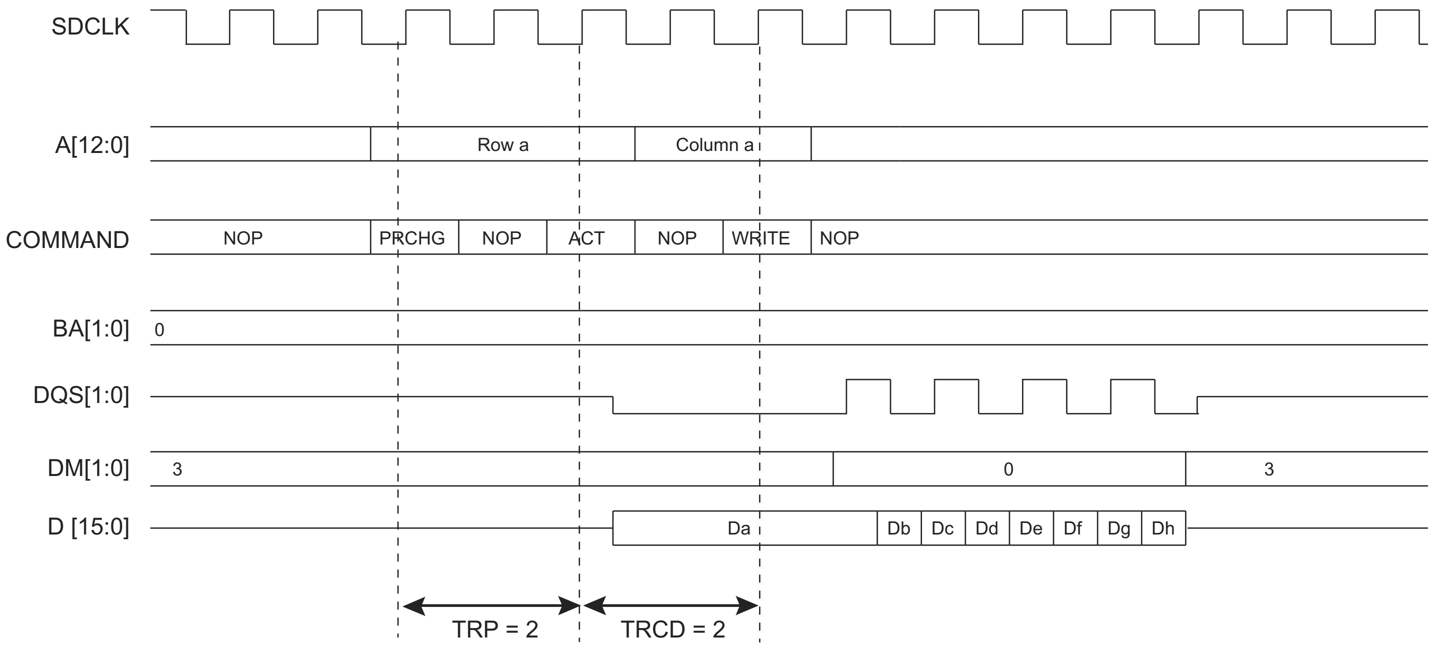


Figure 29-5. Burst Write Access, Row Closed, Low-power DDR1-SDRAM Device



29.7.4 DDRSDRC Timing Parameter 0 Register

Name: DDRSDRC_TPR0

Address: 0xFFFFE80C

Access: Read/Write

31	30	29	28	27	26	25	24
TMRD				REDUCE_WRRD	TWTR		

23	22	21	20	19	18	17	16
TRRD				TRP			

15	14	13	12	11	10	9	8
TRC				TWR			

7	6	5	4	3	2	1	0
TRCD				TRAS			

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

- **TRAS: Active to Precharge Delay**

Reset value is 5 cycles.

This field defines the delay between an Activate command and a Precharge command in number of cycles. Number of cycles is between 0 and 15.

- **TRCD: Row to Column Delay**

Reset value is 2 cycles.

This field defines the delay between an Activate command and a Read/Write command in number of cycles. Number of cycles is between 0 and 15.

- **TWR: Write Recovery Delay**

Reset value is 2 cycles.

This field defines the Write Recovery Time in number of cycles. Number of cycles is between 1 and 15.

- **TRC: Row Cycle Delay**

Reset value is 7 cycles.

This field defines the delay between an Activate command and Refresh command in number of cycles. Number of cycles is between 0 and 15

- **TRP: Row Precharge Delay**

Reset value is 2 cycles.

This field defines the delay between a Precharge command and another command in number of cycles. Number of cycles is between 0 and 15.

- **TRRD: Active BankA to Active BankB**

Reset value is 2 cycles.

This field defines the delay between an Active command in BankA and an active command in BankB in number of cycles. Number of cycles is between 1 and 15.

30.6.2 Memory Peripherals

Figure 30-3 on page 481 shows the DMAC transfer hierarchy of the DMAC for a memory peripheral. There is no handshaking interface with the DMAC, and therefore the memory peripheral can never be a flow controller. Once the channel is enabled, the transfer proceeds immediately without waiting for a transaction request. The alternative to not having a transaction-level handshaking interface is to allow the DMAC to attempt AMBA transfers to the peripheral once the channel is enabled. If the peripheral slave cannot accept these AMBA transfers, it inserts wait states onto the bus until it is ready; it is not recommended that more than 16 wait states be inserted onto the bus. By using the handshaking interface, the peripheral can signal to the DMAC that it is ready to transmit/receive data, and then the DMAC can access the peripheral without the peripheral inserting wait states onto the bus.

30.6.3 Handshaking Interface

Handshaking interfaces are used at the transaction level to control the flow of single or chunk transfers. The operation of the handshaking interface is different and depends on whether the peripheral or the DMAC is the flow controller.

The peripheral uses the handshaking interface to indicate to the DMAC that it is ready to transfer/accept data over the AMBA bus. A non-memory peripheral can request a DMAC transfer through the DMAC using one of two handshaking interfaces:

- Hardware handshaking
- Software handshaking

Software selects between the hardware or software handshaking interface on a per-channel basis. Software handshaking is accomplished through memory-mapped registers, while hardware handshaking is accomplished using a dedicated handshaking interface.

30.6.3.1 Software Handshaking

When the slave peripheral requires the DMAC to perform a DMAC transaction, it communicates this request by sending an interrupt to the CPU or interrupt controller.

The interrupt service routine then uses the software registers to initiate and control a DMAC transaction. These software registers are used to implement the software handshaking interface.

The SRC_H2SEL/DST_H2SEL bit in the Channel Configuration Register (DMAC_CFGx) must be cleared to enable software handshaking.

When the peripheral is not the flow controller, then the Software Last Transfer Flag Register (DMAC_LAST) is not used, and the values in these registers are ignored.

Chunk Transactions

Writing a ‘1’ to the Software Chunk Transfer Request Register (DMAC_CREQ[2x]) starts a source chunk transaction request, where x is the channel number. Writing a ‘1’ to the DMAC_CREQ[2x+1] register starts a destination chunk transfer request, where x is the channel number.

Upon completion of the chunk transaction, the hardware clears the DMAC_CREQ[2*x*] or DMAC_CREQ[2x+1].

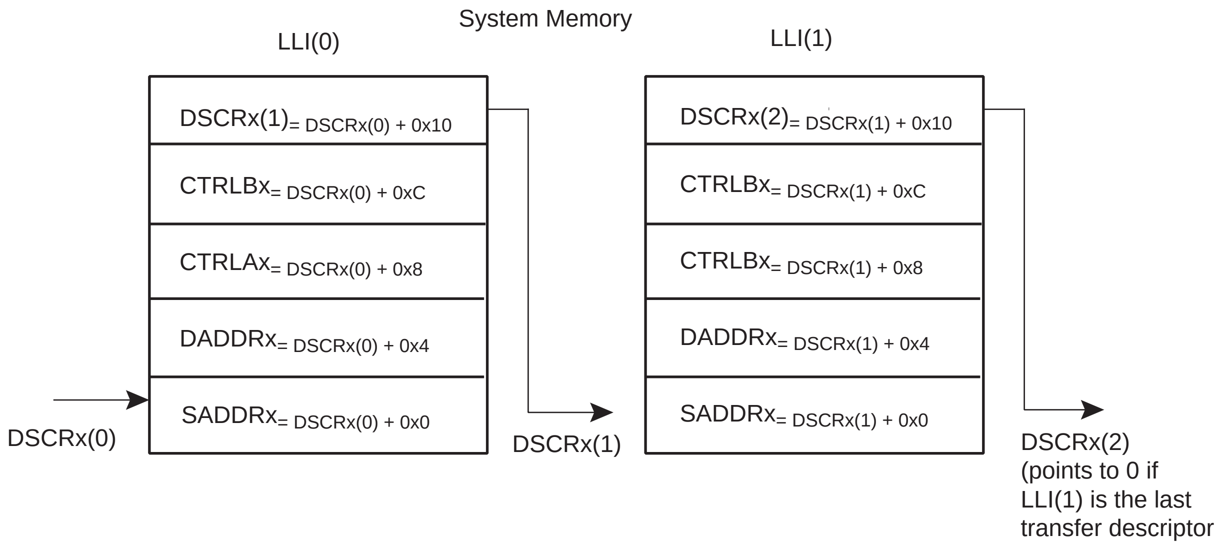
Single Transactions

Writing a ‘1’ to the Software Single Request Register (DMAC_SREQ[2x]) starts a source single transaction request, where x is the channel number. Writing a ‘1’ to the DMAC_SREQ[2x+1] register starts a destination single transfer request, where x is the channel number.

Upon completion of the chunk transaction, the hardware clears the DMAC_SREQ[x] or DMAC_SREQ[2x+1].

The software can poll the relevant channel bit in the DMAC_CREQ[2x]/DMAC_CREQ[2x+1] and DMAC_SREQ[x]/DMAC_SREQ[2x+1] registers. When both are 0, then either the requested chunk or single transaction has completed.

Figure 30-5. Multi-Buffer Transfer Using Linked List



30.8.13 DMAC Channel x [x = 0..7] Source Address Register

Name: DMAC_SADDRx [x = 0..7]

Address: 0xFFFFEC3C (0)[0], 0xFFFFEC64 (0)[1], 0xFFFFEC8C (0)[2], 0xFFFFECB4 (0)[3], 0xFFFFECDC (0)[4], 0xFFFFED04 (0)[5], 0xFFFFED2C (0)[6], 0xFFFFED54 (0)[7], 0xFFFFEE3C (1)[0], 0xFFFFEE64 (1)[1], 0xFFFFEE8C (1)[2], 0xFFFFEEB4 (1)[3], 0xFFFFEEDC (1)[4], 0xFFFFEF04 (1)[5], 0xFFFFEF2C (1)[6], 0xFFFFEF54 (1)[7]

Access: Read/Write

31	30	29	28	27	26	25	24
SADDR							

23	22	21	20	19	18	17	16
SADDR							

15	14	13	12	11	10	9	8
SADDR							

7	6	5	4	3	2	1	0
SADDR							

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

- **SADDR: Channel x Source Address**

This register must be aligned with the source transfer width.

31.7.2 UDPHS Frame Number Register

Name:	UDPHS_FNUM
Address:	0xF803C004
Access:	Read-only

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

- **MICRO_FRAME_NUM: Microframe Number (cleared upon USB reset)**

Number of the received microframe (0 to 7) in one frame.This field is reset at the beginning of each new frame (1 ms).

One microframe is received each 125 microseconds (1 ms/8).

- **FRAME_NUMBER: Frame Number as defined in the Packet Field Formats (cleared upon USB reset)**

This field is provided in the last received SOF packet (see INT_SOF in the UDPHS Interrupt Status Register).

- **FNUM_ERR: Frame Number CRC Error (cleared upon USB reset)**

This bit is set by hardware when a corrupted Frame Number in Start of Frame packet (or Micro SOF) is received.

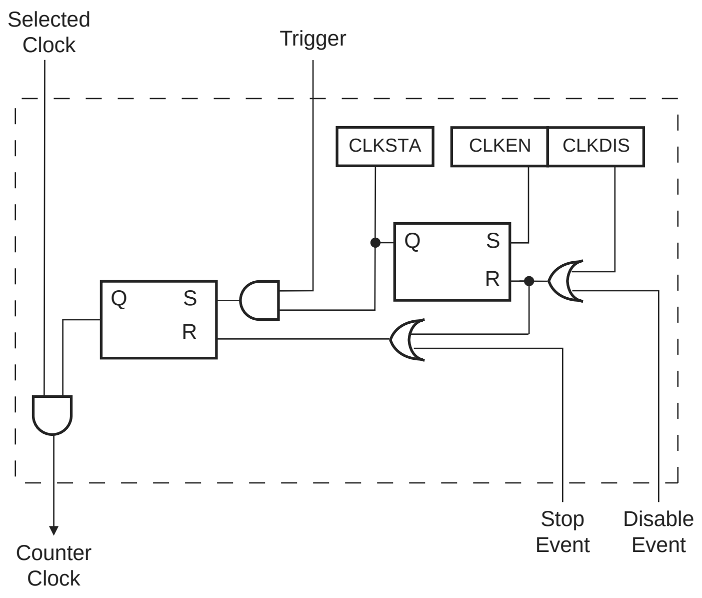
This bit and the INT_SOF (or MICRO_SOF) interrupt are updated at the same time.

35.6.4 Clock Control

The clock of each counter can be controlled in two different ways: it can be enabled/disabled and started/stopped. See Figure 35-4.

- The clock can be enabled or disabled by the user with the CLKEN and the CLKDIS commands in the TC Channel Control Register (TC_CCR). In Capture mode it can be disabled by an RB load event if LDBDIS is set to 1 in the TC_CMR. In Waveform mode, it can be disabled by an RC Compare event if CPCDIS is set to 1 in TC_CMR. When disabled, the start or the stop actions have no effect: only a CLKEN command in the TC_CCR can re-enable the clock. When the clock is enabled, the CLKSTA bit is set in the TC_SR.
- The clock can also be started or stopped: a trigger (software, synchro, external or compare) always starts the clock. The clock can be stopped by an RB load event in Capture mode (LDBSTOP = 1 in TC_CMR) or an RC compare event in Waveform mode (CPCSTOP = 1 in TC_CMR). The start and the stop commands are effective only if the clock is enabled.

Figure 35-4. Clock Control



35.6.5 Operating Modes

Each channel can operate independently in two different modes:

- Capture mode provides measurement on signals.
- Waveform mode provides wave generation.

The TC operating mode is programmed with the WAVE bit in the TC_CMR.

In Capture mode, TIOA and TIOB are configured as inputs.

In Waveform mode, TIOA is always configured to be an output and TIOB is an output if it is not selected to be the external trigger.

35.6.6 Trigger

A trigger resets the counter and starts the counter clock. Three types of triggers are common to both modes, and a fourth external trigger is available to each mode.

Regardless of the trigger used, it will be taken into account at the following active edge of the selected clock. This means that the counter value can be read differently from zero just after a trigger, especially when a low frequency signal is selected as the clock.

35.6.11.3 WAVSEL = 01

When WAVSEL = 01, the value of TC_CV is incremented from 0 to 2^{32}-1 . Once 2^{32}-1 is reached, the value of TC_CV is decremented to 0, then re-incremented to 2^{32}-1 and so on. See Figure 35-11.

A trigger such as an external event or a software trigger can modify TC_CV at any time. If a trigger occurs while TC_CV is incrementing, TC_CV then decrements. If a trigger is received while TC_CV is decrementing, TC_CV then increments. See Figure 35-12.

RC Compare cannot be programmed to generate a trigger in this configuration.

At the same time, RC Compare can stop the counter clock (CPCSTOP = 1) and/or disable the counter clock (CPCDIS = 1).

Figure 35-11. WAVSEL = 01 without Trigger

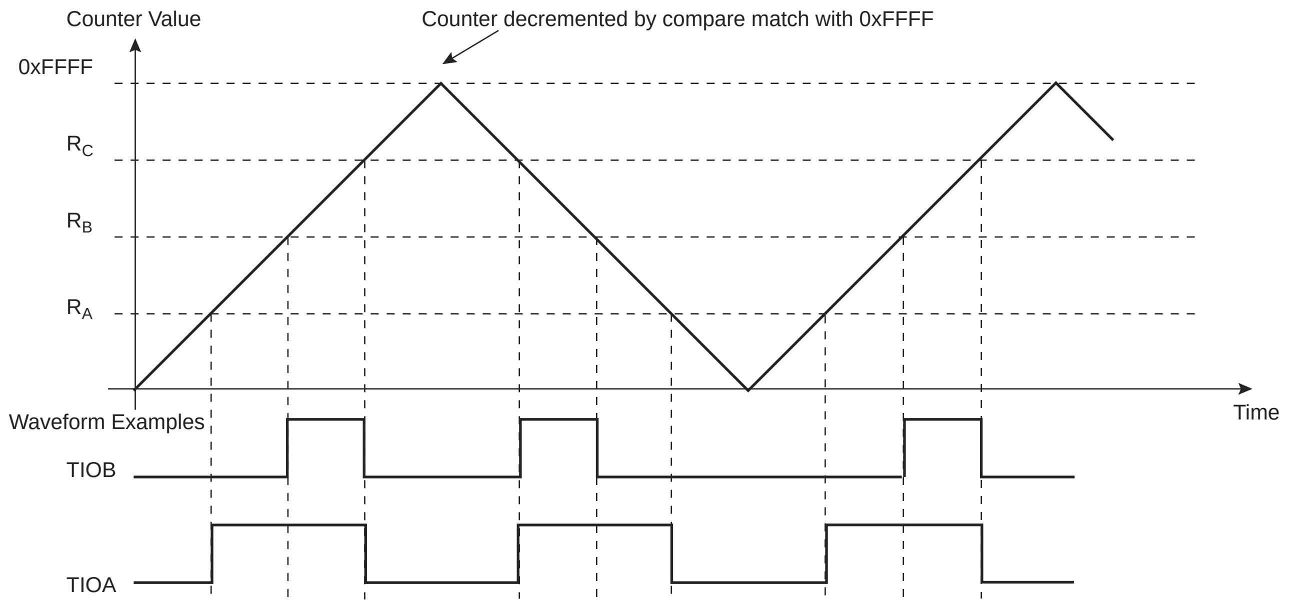
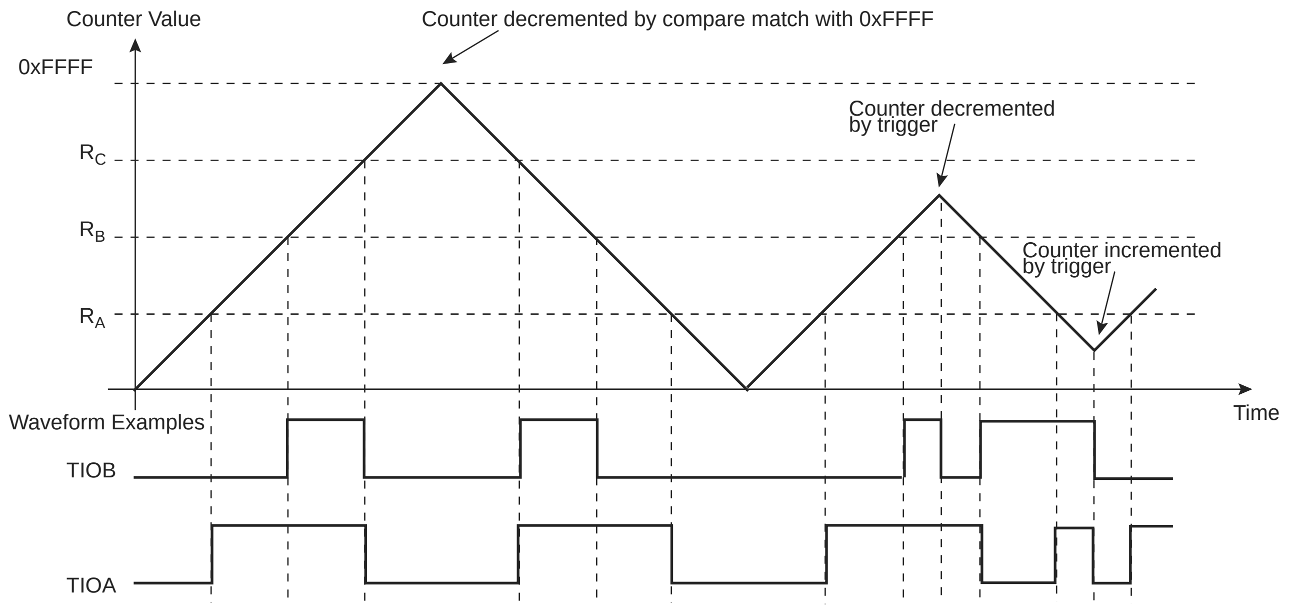


Figure 35-12. WAVSEL = 01 with Trigger


Counter Value
Counter decremented by compare match with 0xFFFF
0xFFFF
Counter decremented by trigger
R_C
R_B
Counter incremented by trigger
R_A
Time
Waveform Examples
TIOB
TIOA

37.8.3 TWI Slave Mode Register

Name: TWI_SMR

Address: 0xF8010008 (0), 0xF8014008 (1), 0xF8018008 (2)

Access: Read/Write

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

23	22	21	20	19	18	17	16
–	SADR						

15	14	13	12	11	10	9	8
–	–	–	–	–	–		

7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	–

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

- **SADR: Slave Address**

The slave device address is used in Slave mode in order to be accessed by master devices in Read or Write mode.

SADR must be programmed before enabling the Slave mode or after a general call. Writes at other times have no effect.

Figure 38-22. Timeguard Operations

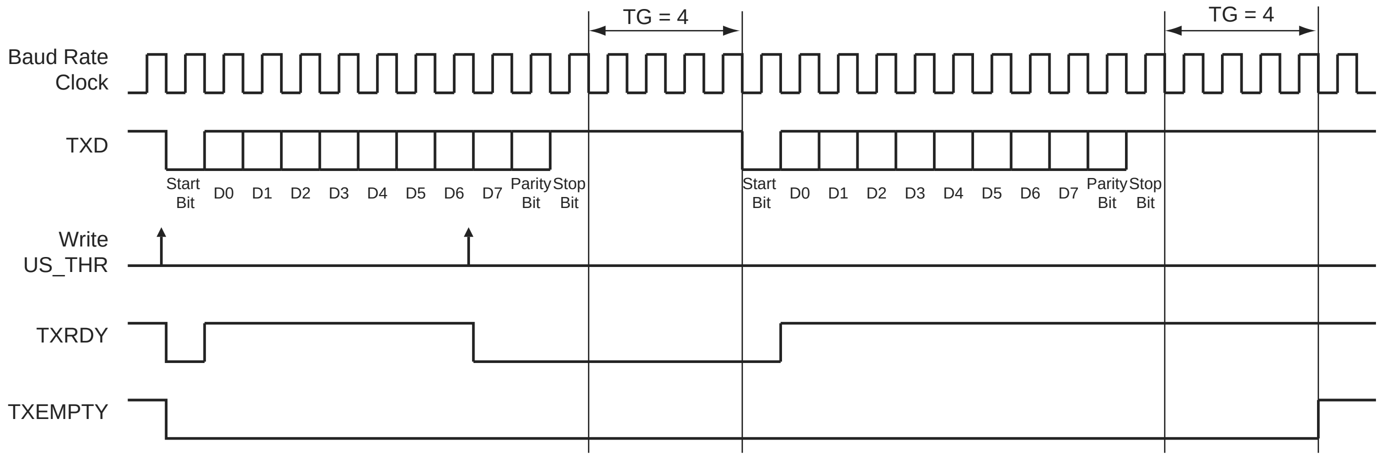


Table 38-9 indicates the maximum length of a timeguard period that the transmitter can handle in relation to the function of the baud rate.

Table 38-9. Maximum Timeguard Length Depending on Baud Rate

Baud Rate (bit/s)	Bit Time (µs)	Timeguard (ms)
1,200	833	212.50
9,600	104	26.56
14,400	69.4	17.71
19,200	52.1	13.28
28,800	34.7	8.85
38,400	26	6.63
56,000	17.9	4.55
57,600	17.4	4.43
115,200	8.7	2.21

38.6.3.11 Receiver Time-out

The Receiver Time-out provides support in handling variable-length frames. This feature detects an idle condition on the RXD line. When a time-out is detected, the bit TIMEOUT in the US_CSR rises and can generate an interrupt, thus indicating to the driver an end of frame.

The time-out delay period (during which the receiver waits for a new character) is programmed in the TO field of the Receiver Time-out register (US_RTOR). If the TO field is written to 0, the Receiver Time-out is disabled and no time-out is detected. The TIMEOUT bit in the US_CSR remains at 0. Otherwise, the receiver loads a 16-bit counter with the value programmed in TO. This counter is decremented at each bit period and reloaded each time a new character is received. If the counter reaches 0, the TIMEOUT bit in US_CSR rises. Then, the user can either:

- Stop the counter clock until a new character is received. This is performed by writing a 1 to the STTTO (Start Time-out) bit in the US_CR. In this case, the idle state on RXD before a new character is received will not provide a time-out. This prevents having to handle an interrupt before a character is received and allows waiting for the next idle state on RXD after a frame is received.
- Obtain an interrupt while no character is received. This is performed by writing a 1 to the RETTO (Reload and Start Time-out) bit in the US_CR. If RETTO is performed, the counter starts counting down immediately from the value TO. This enables generation of a periodic interrupt so that a user time-out can be handled, for example when no key is pressed on a keyboard.

38.7.21 USART Transmitter Timeguard Register

Name: US_TTGR

Address: 0xF801C028 (0), 0xF8020028 (1), 0xF8024028 (2), 0xF8028028 (3)

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**
TG							

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

- **TG: Timeguard Value**

0: The transmitter timeguard is disabled.

1–255: The transmitter timeguard is enabled and TG is Timeguard Delay / Bit Period.

38.7.22 USART FI DI RATIO Register

Name: US_FIDI

Address: 0xF801C040 (0), 0xF8020040 (1), 0xF8024040 (2), 0xF8028040 (3)

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
–	–	–	–	–	FI_DI_RATIO		

7	6	5	4	3	2	1	0
FI_DI_RATIO							

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

- **FI_DI_RATIO: FI Over DI Ratio Value**

0: If ISO7816 mode is selected, the baud rate generator generates no signal.

1–2: Do not use.

3–2047: If ISO7816 mode is selected, the baud rate is the clock provided on SCK divided by FI_DI_RATIO.

39.6.1 UART Control Register

Name: UART_CR

Address: 0xF8040000 (0), 0xF8044000 (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**
–	–	–	–	–	–	–	RSTSTA
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TXDIS	TXEN	RXDIS	RXEN	RSTTX	RSTRX	–	–

- **RSTRX: Reset Receiver**

0: No effect.

1: The receiver logic is reset and disabled. If a character is being received, the reception is aborted.

- **RSTTX: Reset Transmitter**

0: No effect.

1: The transmitter logic is reset and disabled. If a character is being transmitted, the transmission is aborted.

- **RXEN: Receiver Enable**

0: No effect.

1: The receiver is enabled if RXDIS is 0.

- **RXDIS: Receiver Disable**

0: No effect.

1: The receiver is disabled. If a character is being processed and RSTRX is not set, the character is completed before the receiver is stopped.

- **TXEN: Transmitter Enable**

0: No effect.

1: The transmitter is enabled if TXDIS is 0.

- **TXDIS: Transmitter Disable**

0: No effect.

1: The transmitter is disabled. If a character is being processed and a character has been written in the UART_THR and RSTTX is not set, both characters are completed before the transmitter is stopped.

- **RSTSTA: Reset Status**

0: No effect.

1: Resets the status bits PARE, FRAME and OVRE in the UART_SR.

39.6.5 UART Interrupt Mask Register

Name: UART_IMR

Address: 0xF8040010 (0), 0xF8044010 (1)

Access: Read-only

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

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is disabled.

1: The corresponding interrupt is enabled.

- **RXRDY: Mask RXRDY Interrupt**

- **TXRDY: Disable TXRDY Interrupt**

- **OVRE: Mask Overrun Error Interrupt**

- **FRAME: Mask Framing Error Interrupt**

- **PARE: Mask Parity Error Interrupt**

- **TXEMPTY: Mask TXEMPTY Interrupt**