Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	AVR
Core Size	32-Bit Single-Core
Speed	66MHz
Connectivity	Ethernet, I²C, SPI, SSC, UART/USART, USB OTG
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	69
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	32K x 8
Voltage - Supply (Vcc/Vdd)	1.65V ~ 3.6V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	100-TQFP
Supplier Device Package	100-TQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/at32uc3a1128-aur

Table 5-1. Signal Description List

Signal Name	Function	Type	Active Level	Comments
Analog to Digital Converter - ADC				
AD0 - AD7	Analog input pins	Analog input		
ADVREF	Analog positive reference voltage input	Analog input		2.6 to 3.6V
Pulse Width Modulator - PWM				
PWM0 - PWM6	PWM Output Pins	Output		
Universal Serial Bus Device - USB				
DDM	USB Device Port Data -	Analog		
DDP	USB Device Port Data +	Analog		
VBUS	USB VBUS Monitor and OTG Negociation	Analog Input		
USBID	ID Pin of the USB Bus	Input		
USB_VBOF	USB VBUS On/off: bus power control port	output		
Audio Bitstream DAC (ABDAC)				
DATA0-DATA1	D/A Data out	Outpu		
DATAN0-DATAN1	D/A Data inverted out	Outpu		

The effect of writing or reading the bits listed above depends on which register is being accessed:

- **IER (Write-only)**
  - 0: No effect
  - 1: Enable Interrupt
- **IDR (Write-only)**
  - 0: No effect
  - 1: Disable Interrupt
- **IMR (Read-only)**
  - 0: Interrupt is disabled
  - 1: Interrupt is enabled
- **ISR (Read-only)**
  - 0: An interrupt event has not occurred or has been previously cleared
  - 1: An interrupt event has not occurred
- **ICR (Write-only)**
  - 0: No effect
  - 1: Clear corresponding event

through a dedicated Peripheral Bus address. Some of the general-purpose fuse bits are reserved for special purposes, and should not be used for other functions.:

Table 18-2. General-purpose fuses with special functions

General-Purpose fuse number	Name	Usage
15:0	LOCK	Region lock bits.
16	EPFL	External Privileged Fetch Lock. Used to prevent the CPU from fetching instructions from external memories when in privileged mode. This bit can only be changed when the security bit is cleared. The address range corresponding to external memories is device-specific, and not known to the flash controller. This fuse bit is simply routed out of the CPU or bus system, the flash controller does not treat this fuse in any special way, except that it can not be altered when the security bit is set. If the security bit is set, only an external JTAG Chip Erase can clear EPFL. No internal commands can alter EPFL if the security bit is set. When the fuse is erased (i.e. "1"), the CPU can execute instructions fetched from external memories. When the fuse is programmed (i.e. "0"), instructions can not be executed from external memories.
19:17	BOOTPROT	Used to select one of four different bootloader sizes. Pages included in the bootloader area can not be erased or programmed except by a JTAG chip erase. BOOTPROT can only be changed when the security bit is cleared. If the security bit is set, only an external JTAG Chip Erase can clear BOOTPROT, and thereby allow the pages protected by BOOTPROT to be programmed. No internal commands can alter BOOTPROT or the pages protected by BOOTPROT if the security bit is set.

The BOOTPROT fuses protects the following address space for the Boot Loader:

Table 18-3. Boot Loader area specified by BOOTPROT

BOOTPROT	Pages protected by BOOTPROT	Size of protected memory
7	None	0
6	0-1	1kByte
5	0-3	2kByte
4	0-7	4kByte
3	0-15	8kByte
2	0-31	16kByte
1	0-63	32kByte
0	0-127	64kByte

To erase or write a general-purpose fuse bit, the commands Write General-Purpose Fuse Bit (WGPB) and Erase General-Purpose Fuse Bit (EGPB) are provided. Writing one of these

19.7 User Interface

Table 19-1. Register Mapping

Offset	Register	Name	Access	Reset Value
0x0000	Master Configuration Register 0	MCFG0	Read/Write	0x00000002
0x0004	Master Configuration Register 1	MCFG1	Read/Write	0x00000002
0x0008	Master Configuration Register 2	MCFG2	Read/Write	0x00000002
0x000C	Master Configuration Register 3	MCFG3	Read/Write	0x00000002
0x0010	Master Configuration Register 4	MCFG4	Read/Write	0x00000002
0x0014	Master Configuration Register 5	MCFG5	Read/Write	0x00000002
0x0018	Master Configuration Register 6	MCFG6	Read/Write	0x00000002
0x001C	Master Configuration Register 7	MCFG7	Read/Write	0x00000002
0x0020	Master Configuration Register 8	MCFG8	Read/Write	0x00000002
0x0024	Master Configuration Register 9	MCFG9	Read/Write	0x00000002
0x0028	Master Configuration Register 10	MCFG10	Read/Write	0x00000002
0x002C	Master Configuration Register 11	MCFG11	Read/Write	0x00000002
0x0030	Master Configuration Register 12	MCFG12	Read/Write	0x00000002
0x0034	Master Configuration Register 13	MCFG13	Read/Write	0x00000002
0x0038	Master Configuration Register 14	MCFG14	Read/Write	0x00000002
0x003C	Master Configuration Register 15	MCFG15	Read/Write	0x00000002
0x0040	Slave Configuration Register 0	SCFG0	Read/Write	0x00000010
0x0044	Slave Configuration Register 1	SCFG1	Read/Write	0x00000010
0x0048	Slave Configuration Register 2	SCFG2	Read/Write	0x00000010
0x004C	Slave Configuration Register 3	SCFG3	Read/Write	0x00000010
0x0050	Slave Configuration Register 4	SCFG4	Read/Write	0x00000010
0x0054	Slave Configuration Register 5	SCFG5	Read/Write	0x00000010
0x0058	Slave Configuration Register 6	SCFG6	Read/Write	0x00000010
0x005C	Slave Configuration Register 7	SCFG7	Read/Write	0x00000010
0x0060	Slave Configuration Register 8	SCFG8	Read/Write	0x00000010
0x0064	Slave Configuration Register 9	SCFG9	Read/Write	0x00000010
0x0068	Slave Configuration Register 10	SCFG10	Read/Write	0x00000010
0x006C	Slave Configuration Register 11	SCFG11	Read/Write	0x00000010
0x0070	Slave Configuration Register 12	SCFG12	Read/Write	0x00000010
0x0074	Slave Configuration Register 13	SCFG13	Read/Write	0x00000010
0x0078	Slave Configuration Register 14	SCFG14	Read/Write	0x00000010
0x007C	Slave Configuration Register 15	SCFG15	Read/Write	0x00000010
0x0080	Priority Register A for Slave 0	PRAS0	Read/Write	0x00000000
0x0084	Priority Register B for Slave 0	PRBS0	Read/Write	0x00000000
0x0088	Priority Register A for Slave 1	PRAS1	Read/Write	0x00000000

21.5.11 PDCA Interrupt Enable Register

Name: IER

Access Type: 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
-	-	-	-	-	TERR	TRC	RCZ

- **TERR: Transfer Error**

0 = No effect.

1 = Enable Transfer Error interrupt.

- **TRC: Transfer Complete**

0 = No effect.

1 = Enable Transfer Complete interrupt.

- **RCZ: Reload Counter Zero**

0 = No effect.

1 = Enable Reload Counter Zero interrupt.

23.7.3.8 Mode Fault Detection

A mode fault is detected when the SPI is programmed in Master Mode and a low level is driven by an external master on the NPCS0/NSS signal. NPCS0, MOSI, MISO and SPCK must be configured in open-drain through the PIO controller, so that external pull up resistors are needed to guarantee high level.

When a mode fault is detected, the MODF bit in the SR is set until the SR is read and the SPI is automatically disabled until re-enabled by writing the SPIEN bit in the CR (Control Register) at 1.

By default, the Mode Fault detection circuitry is enabled. The user can disable Mode Fault detection by setting the MODFDIS bit in the SPI Mode Register (MR).

23.7.4 SPI Slave Mode

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

The SPI waits for NSS to go active before receiving the serial clock from an external master. When NSS falls, the clock is validated on the serializer, which processes the number of bits defined by the BITS field of the Chip Select Register 0 (CSR0). These bits are processed following a phase and a polarity defined respectively by the NCPHA and CPOL bits of the CSR0. Note that BITS, CPOL and NCPHA of the other Chip Select Registers have no effect when the SPI is programmed in Slave Mode.

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

When all the bits are processed, the received data is transferred in the Receive Data Register and the RDRF bit rises. If RDRF is already high when the data is transferred, the Overrun bit rises and the data transfer to RDR is aborted.

When a transfer starts, the data shifted out is the data present in the Shift Register. If no data has been written in the Transmit Data Register (TDR), the last data received is transferred. If no data has been received since the last reset, all bits are transmitted low, as the Shift Register resets at 0.

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

Then, a new data is loaded in the Shift Register from the Transmit Data Register. In case no character is ready to be transmitted, i.e. no character has been written in TDR since the last load from TDR to the Shift Register, the Shift Register is not modified and the last received character is retransmitted.

Figure 23-9 shows a block diagram of the SPI when operating in Slave Mode.

24.13.6 Clock Synchronization

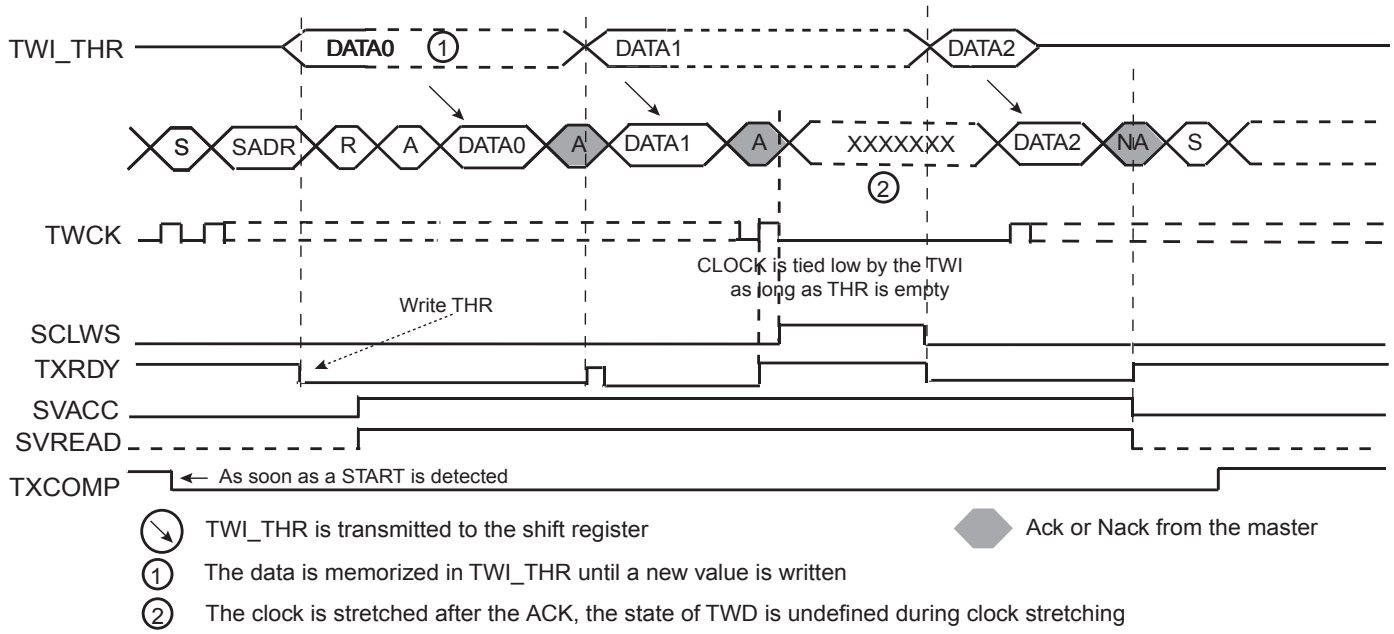
In both read and write modes, it may happen that THR/RHR buffer is not filled /emptied before the emission/reception of a new character. In this case, to avoid sending/receiving undesired data, a clock stretching mechanism is implemented.

24.13.6.1 Clock Synchronization in Read Mode

The clock is tied low if the shift register is empty and if a STOP or REPEATED START condition was not detected. It is tied low until the shift register is loaded.

Figure 24-26 on page 242 describes the clock synchronization in Read mode.

Figure 24-26. Clock Synchronization in Read Mode



Notes:
1. TXRDY is reset when data has been written in the TH to the shift register and set when this data has been acknowledged or non acknowledged.
2. At the end of the read sequence, TXCOMP is set after a STOP or after a REPEATED_START + an address different from SADR.
3. SCLWS is automatically set when the clock synchronization mechanism is started.

The receiver samples the RXD line. If the line is sampled during one half of a bit time at 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 at 0), a start is detected at the eighth sample at 0. Then, data bits, parity bit and stop bit are sampled on each 16 sampling clock cycle. If the oversampling is 8 (OVER at 1), a start bit is detected at the fourth sample at 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 effect on the receiver as it considers only one stop bit, regardless of the field NBSTOP, so that resynchronization between the receiver and the transmitter can occur. Moreover, as soon as the stop bit is sampled, the receiver starts looking for a new start bit so that resynchronization can also be accomplished when the transmitter is operating with one stop bit.

Figure 26-12 on page 314 and Figure 26-13 on page 314 illustrate start detection and character reception when USART operates in asynchronous mode.

Figure 26-12. Asynchronous Start Detection

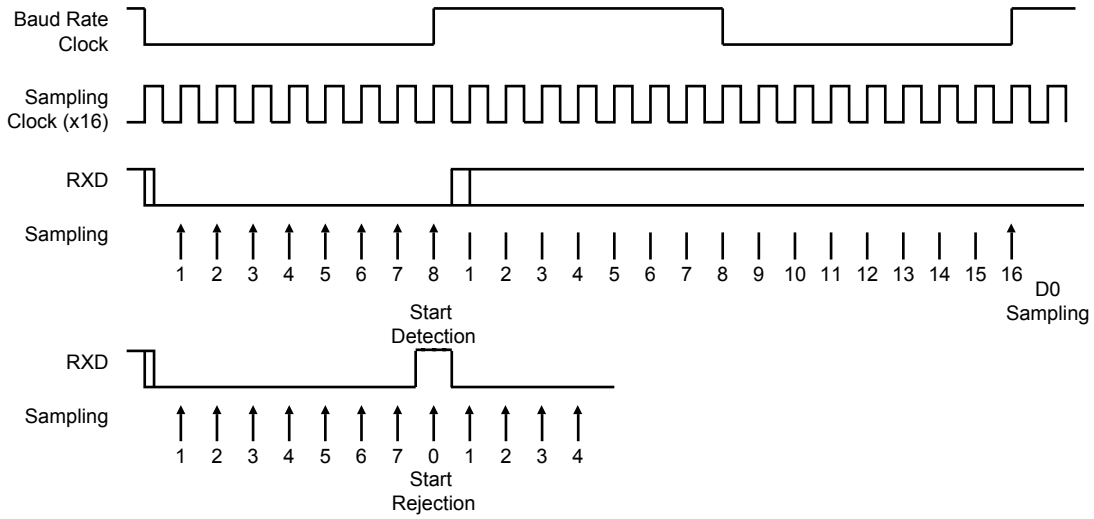
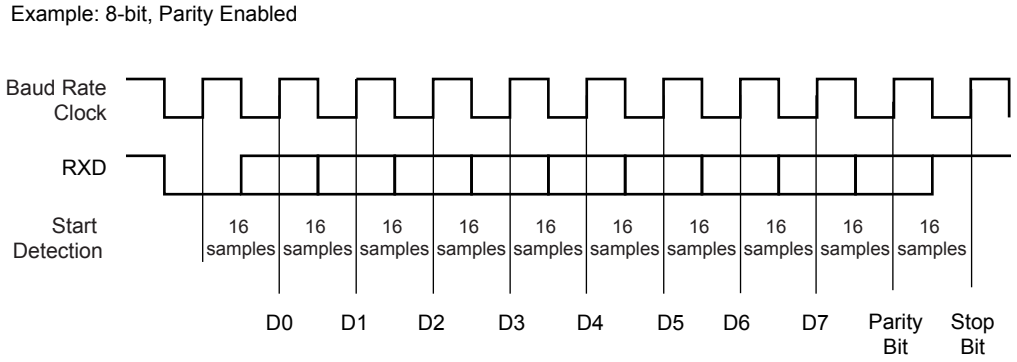


Figure 26-13. Asynchronous Character Reception



26.8.2 USART Control Register

Name: CR

Access Type: Write-only

Offset: 0x0

Reset Value: -

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	RTSDIS/RCS	RTSEN/FCS	–	–
15	14	13	12	11	10	9	8
RETTO	RSTNACK	RSTIT	SENDA	STTTO	STPBRK	STTBRK	RSTSTA
7	6	5	4	3	2	1	0
TXDIS	TXEN	RXDIS	RXEN	RSTTX	RSTRX	–	–

- **RTSDIS/RCS: Request to Send Disable/Release SPI Chip Select**
  - If USART does not operate in SPI Master Mode (MODE … 0xE):

0: No effect.

1: Drives the pin RTS to 1.

  - If USART operates in SPI Master Mode (MODE = 0xE):

RCS = 0: No effect.

RCS = 1: Releases the Slave Select Line NSS (RTS pin).

- **RTSEN/FCS: Request to Send Enable/Force SPI Chip Select**
  - If USART does not operate in SPI Master Mode (MODE … 0xE):

0: No effect.

1: Drives the pin RTS to 0.

  - If USART operates in SPI Master Mode (MODE = 0xE):

FCS = 0: No effect.

FCS = 1: Forces the Slave Select Line NSS (RTS pin) to 0, even if USART is no transmitting, in order to address SPI slave devices supporting the CSAAT Mode (Chip Select Active After Transfer).

- **RETTO: Rearm Time-out**

0: No effect

1: Restart Time-out

- **RSTNACK: Reset Non Acknowledge**

0: No effect

1: Resets NACK in CSR.

Figure 27-8. Connection of 2x16-bit Data Bus on a 32-bit Data Bus (Byte Select Option)

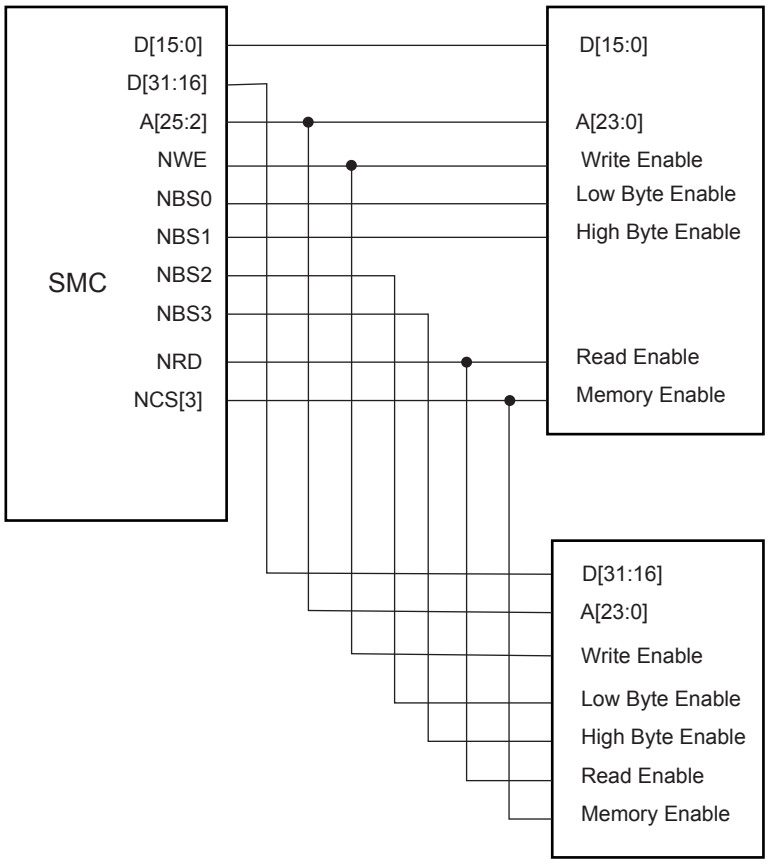


Table 27-2. SMC Multiplexed Signal Translation

Signal Name	32-bit Bus			16-bit Bus		8-bit Bus
Device Type	**1x32-bit**	**2x16-bit**	**4 x 8-bit**	**1x16-bit**	**2 x 8-bit**	**1 x 8-bit**
Byte Access Type (BAT)	Byte Select	Byte Select	Byte Write	Byte Select	Byte Write	
NBS0_A0	NBS0	NBS0		NBS0		A0
NWE_NWR0	NWE	NWE	NWR0	NWE	NWR0	NWE
NBS1_NWR1	NBS1	NBS1	NWR1	NBS1	NWR1	
NBS2_NWR2_A1	NBS2	NBS2	NWR2	A1	A1	A1
NBS3_NWR3	NBS3	NBS3	NWR3			

27.6.4 Standard Read and Write Protocols

In the following sections, the byte access type is not considered. Byte select lines (NBS0 to NBS3) always have the same timing as the A address bus. NWE represents either the NWE signal in byte select access type or one of the byte write lines (NWR0 to NWR3) in byte write access type. NWR0 to NWR3 have the same timings and protocol as NWE. In the same way, NCS represents one of the NCS[0..3] chip select lines.

27.6.9.3 Page Mode Restriction

The page mode is not compatible with the use of the NWAIT signal. Using the page mode and the NWAIT signal may lead to unpredictable behavior.

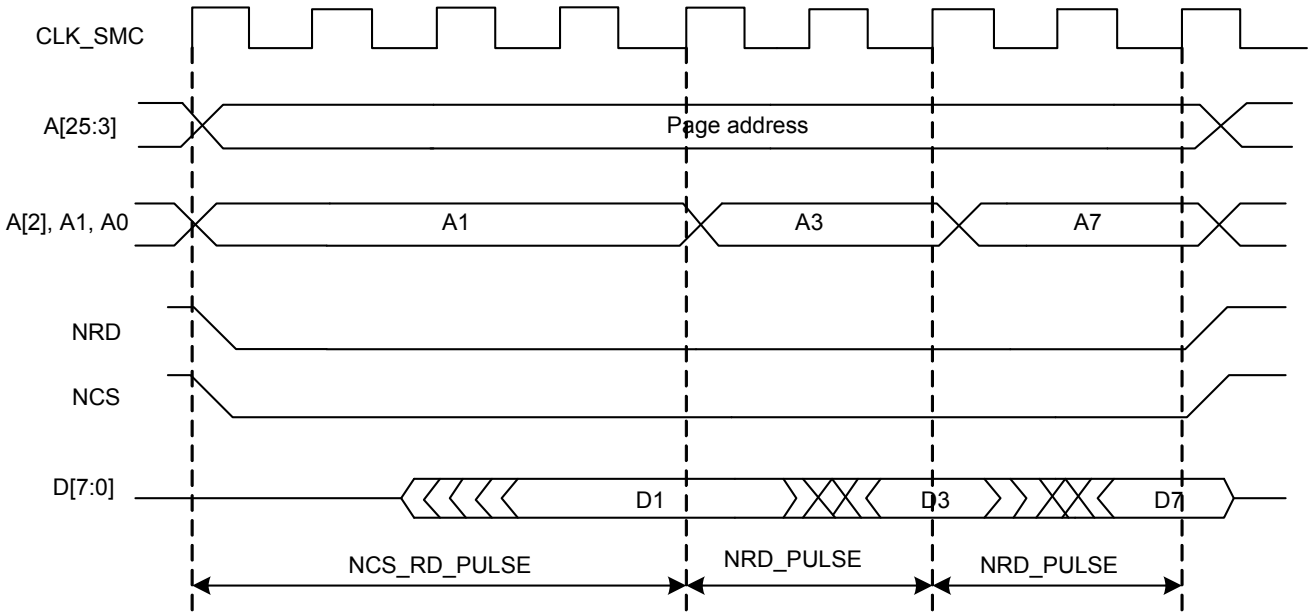
27.6.9.4 Sequential and Non-sequential Accesses

If the chip select and the MSB of addresses as defined in Table 27-5 are identical, then the current access lies in the same page as the previous one, and no page break occurs.

Using this information, all data within the same page, sequential or not sequential, are accessed with a minimum access time (t_{sa}). Figure 27-36 illustrates access to an 8-bit memory device in page mode, with 8-byte pages. Access to D1 causes a page access with a long access time (t_{pa}). Accesses to D3 and D7, though they are not sequential accesses, only require a short access time (t_{sa}).

If the MSB of addresses are different, the SMC performs the access of a new page. In the same way, if the chip select is different from the previous access, a page break occurs. If two sequential accesses are made to the page mode memory, but separated by an other internal or external peripheral access, a page break occurs on the second access because the chip select of the device was deasserted between both accesses.

Figure 27-36. Access to Non-sequential Data within the Same Page



29.7.26 User Input/Output Register

Register Name: USRIO

Access Type: 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
–	–	–	–	TX_PAUSE_ ZERO	TX_PAUSE	EAM	RMII

- **RMII**

When set, this bit enables the MII operation mode. When reset, it selects the RMII mode.

- **EAM**

When set, this bit causes a frame to be copied to memory, if this feature is enabled by the EAE bit in NCFGR. Otherwise, no frame is copied.

- **TX_PAUSE**

Toggling this bit causes a PAUSE frame to be transmitted.

- **TX_PAUSE_ZERO**

Selects either zero or the transmit quantum register as the transmitted pause frame quantum.

30.8.1.3 USB General Status Clear Register (USBSTACLR)

Offset: 0x0808

Register Name: USBSTACLR

Access Type: Write-Only

Read Value: 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
–	–	–	–	–	–	VBUSRQC	–
						w	
						0	
7	6	5	4	3	2	1	0
STOIC	HNPERRIC	ROLEEXIC	BCERRIC	VBERRIC	SRPIC	VBUSTIC	IDTIC
w	w	w	w	w	w	w	w
0	0	0	0	0	0	0	0

- **IDTIC: ID Transition Interrupt Flag Clear**

Set to clear IDTI.

Clearing has no effect.

Always read as 0.

- **VBUSTIC: VBus Transition Interrupt Flag Clear**

Set to clear VBUSTI.

Clearing has no effect.

Always read as 0.

- **SRPIC: SRP Interrupt Flag Clear**

Set to clear SRPI.

Clearing has no effect.

Always read as 0.

- **VBERRIC: VBus Error Interrupt Flag Clear**

Set to clear VBERRI.

Clearing has no effect.

Always read as 0.

- **BCERRIC: B-Connection Error Interrupt Flag Clear**

Set to clear BCERRI.

Shall be cleared by software (by setting the RXOUTIC bit) to acknowledge the interrupt and to free the bank.

For isochronous, bulk and interrupt OUT endpoints:

Set by hardware at the same time as FIFOCON when the current bank is full. This triggers an EPXINT interrupt if RXOUTE = 1.

Shall be cleared by software (by setting the RXOUTIC bit) to acknowledge the interrupt, what has no effect on the endpoint FIFO.

The software then reads from the FIFO and clears the FIFOCON bit to free the bank. If the OUT endpoint is composed of multiple banks, this also switches to the next bank. The RXOUTI and FIFOCON bits are updated by hardware in accordance with the status of the next bank.

RXOUTI shall always be cleared before clearing FIFOCON.

This bit is inactive (cleared) for isochronous, bulk and interrupt IN endpoints.

- **RXSTPI: Received SETUP Interrupt Flag**

For control endpoints, set by hardware to signal that the current bank contains a new valid SETUP packet. This triggers an EPXINT interrupt if RXSTPE = 1.

Shall be cleared by software (by setting the RXSTPIC bit) to acknowledge the interrupt and to free the bank.

This bit is inactive (cleared) for bulk and interrupt IN/OUT endpoints and it means UNDERFI for isochronous IN/OUT endpoints.

- **UNDERFI: Underflow Interrupt Flag**

For isochronous IN/OUT endpoints, set by hardware when an underflow error occurs. This triggers an EPXINT interrupt if UNDERFE = 1.

An underflow can occur during IN stage if the host attempts to read from an empty bank. A zero-length packet is then automatically sent by the USB controller.

An underflow can also occur during OUT stage if the host sends a packet while the bank is already full. Typically, the CPU is not fast enough. The packet is lost.

Shall be cleared by software (by setting the UNDERFIC bit) to acknowledge the interrupt.

This bit is inactive (cleared) for bulk and interrupt IN/OUT endpoints and it means RXSTPI for control endpoints.

- **NAKOUTI: NAKed OUT Interrupt Flag**

Set by hardware when a NAK handshake has been sent in response to an OUT request from the host. This triggers an EPXINT interrupt if NAKOUTE = 1.

Shall be cleared by software (by setting the NAKOUTIC bit) to acknowledge the interrupt.

- **NAKINI: NAKed IN Interrupt Flag**

Set by hardware when a NAK handshake has been sent in response to an IN request from the host. This triggers an EPXINT interrupt if NAKINE = 1.

Shall be cleared by software (by setting the NAKINIC bit) to acknowledge the interrupt.

- **OVERFI: Overflow Interrupt Flag**

Set by hardware when an overflow error occurs. This triggers an EPXINT interrupt if OVERFE = 1.

For all endpoint types, an overflow can occur during OUT stage if the host attempts to write into a bank that is too small for the packet. The packet is acknowledged and the Received OUT Data interrupt (RXOUTI) is raised as if no overflow had occurred. The bank is filled with all the first bytes of the packet that fit in.

30.8.3.22 USB Host DMA Channel X HSB Address Register (UHDMAX_ADDR)

Offset: 0x0714 + (X - 1) . 0x10

Register Name: *UHDMAX_ADDR*, X in [1..6]

Access Type: Read/Write

Reset Value: 0x00000000

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

Same as ”USB Device DMA Channel X HSB Address Register (UDDMAX_ADDR)” on page 592.

32.6.3 PWM Controller Operations

32.6.3.1 Initialization

Before enabling the output channel, this channel must have been configured by the software application:

- Configuration of the clock generator if DIVA and DIVB are required
- Selection of the clock for each channel (CPRE field in the CMRx register)
- Configuration of the waveform alignment for each channel (CALG field in the CMRx register)
- Configuration of the period for each channel (CPRD in the CPRDx register). Writing in CPRDx Register is possible while the channel is disabled. After validation of the channel, the user must use CUPDx Register to update CPRDx as explained below.
- Configuration of the duty cycle for each channel (CDTY in the CDTYx register). Writing in CDTYx Register is possible while the channel is disabled. After validation of the channel, the user must use CUPDx Register to update CDTYx as explained below.
- Configuration of the output waveform polarity for each channel (CPOL in the CMRx register)
- Enable Interrupts (Writing CHIDx in the IER register)
- Enable the PWM channel (Writing CHIDx in the ENA register)

It is possible to synchronize different channels by enabling them at the same time by means of writing simultaneously several CHIDx bits in the ENA register.

In such a situation, all channels may have the same clock selector configuration and the same period specified.

32.6.3.2 Source Clock Selection Criteria

The large number of source clocks can make selection difficult. The relationship between the value in the Period Register (CPRDx) and the Duty Cycle Register (CDTYx) can help the user in choosing. The event number written in the Period Register gives the PWM accuracy. The Duty Cycle quantum cannot be lower than *1/CPRDx* value. The higher the value of CPRDx, the greater the PWM accuracy.

For example, if the user sets 15 (in decimal) in CPRDx, the user is able to set a value between 1 up to 14 in CDTYx Register. The resulting duty cycle quantum cannot be lower than 1/15 of the PWM period.

32.6.3.3 Changing the Duty Cycle or the Period

It is possible to modulate the output waveform duty cycle or period.

To prevent unexpected output waveform, the user must use the update register (PWM_CUPDx) to change waveform parameters while the channel is still enabled. The user can write a new period value or duty cycle value in the update register (CUPDx). This register holds the new value until the end of the current cycle and updates the value for the next cycle. Depending on the CPD field in the CMRx register, CUPDx either updates CPRDx or CDTYx. Note that even if the update register is used, the period must not be smaller than the duty cycle.

33.7.1 Control Register

Name: CR

Access Type: Write-only

Offset: 0x00

Reset Value: –

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
–	–	–	–	–	–	START	SWRST

- **START: Start Conversion**

0 = No effect.

1 = Begins analog-to-digital conversion.

- **SWRST: Software Reset**

0 = No effect.

1 = Resets the ADC simulating a hardware reset.

TRGSEL			Selected TRGSEL
1	0	1	Internal Trigger 5, depending of chip integration
1	1	0	External trigger
1	1	1	Reserved

- **TRGEN: Trigger Enable**

TRGEN	Selected TRGEN
0	Hardware triggers are disabled. Starting a conversion is only possible by software.
1	Hardware trigger selected by TRGSEL field is enabled.

34.6.2 Interpolation filter

The interpolation filter interpolates from f_s to $128f_s$. This filter is a 4th order Cascaded Integrator-Comb filter, and the basic building blocks of this filter is a comb part and an integrator part.

34.6.3 Sigma Delta Modulator

This part is a 3rd order Sigma Delta Modulator consisting of three differentiators (delta blocks), three integrators (sigma blocks) and a one bit quantizer. The purpose of the integrators is to shape the noise, so that the noise is reduces in the band of interest and increased at the higher frequencies, where it can be filtered.

34.6.4 Data Format

Input data is on two’s complement format.

2. **USART RXBREAK problem when no timeguard**
   In asynchronous mode the RXBREAK flag is not correctly handled when the timeguard is 0 and the break character is located just after the stop bit.

   Fix/Workaround
   If the NBSTOP is 1, timeguard should be different from 0.

3. **USART Handshaking: 2 characters sent / CTS rises when TX**
   If CTS switches from 0 to 1 during the TX of a character, if the Holding register is not empty, the TXHOLDING is also transmitted.

   Fix/Workaround
   None.

4. **USART PDC and TIMEGUARD not supported in MANCHESTER**
   Manchester encoding/decoding is not working.

   Fix/Workaround
   Do not use manchester encoding.

5. **USART SPI mode is non functional on this revision.**
   Fix/Workaround
   Do not use the USART SPI mode.

6. **DCD is active High instead of Low.**
   In modem mode the DCD signal is assumed to be active high by the USART, butshould have been active low.
   Fix/Workaround
   Add an external inverter to the DCD line.

7. **ISO7816 info register US_NER cannot be read**
   The NER register always returns zero.
   Fix/Workaround
   None.

41.5.8 Power Manager

1. **Voltage regulator input and output is connected to VDDIO and VDDCORE inside the device**
   The voltage regulator input and output is connected to VDDIO and VDDCORE respectively inside the device.

   Fix/Workaround
   Do not supply VDDCORE externally, as this supply will work in paralell with the regulator.

2. **Wrong reset causes when BOD is activated**
   Setting the BOD enable fuse will cause the Reset Cause Register to list BOD reset as the reset source even though the part was reset by another source.

   Fix/Workaround
   Do not set the BOD enable fuse, but activate the BOD as soon as your program starts.

3. **PLL0/1 Lock control does not work**
   Lock Control does not work for PLL0 and PLL1.