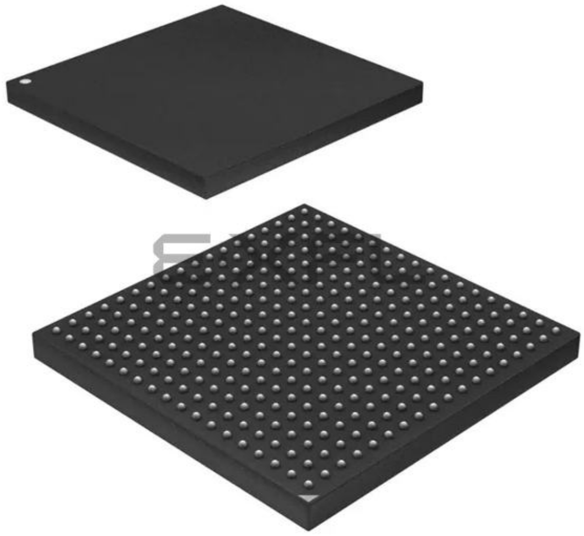
E·XFL

Microchip Technology - AT91SAM9M10C-CU 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.

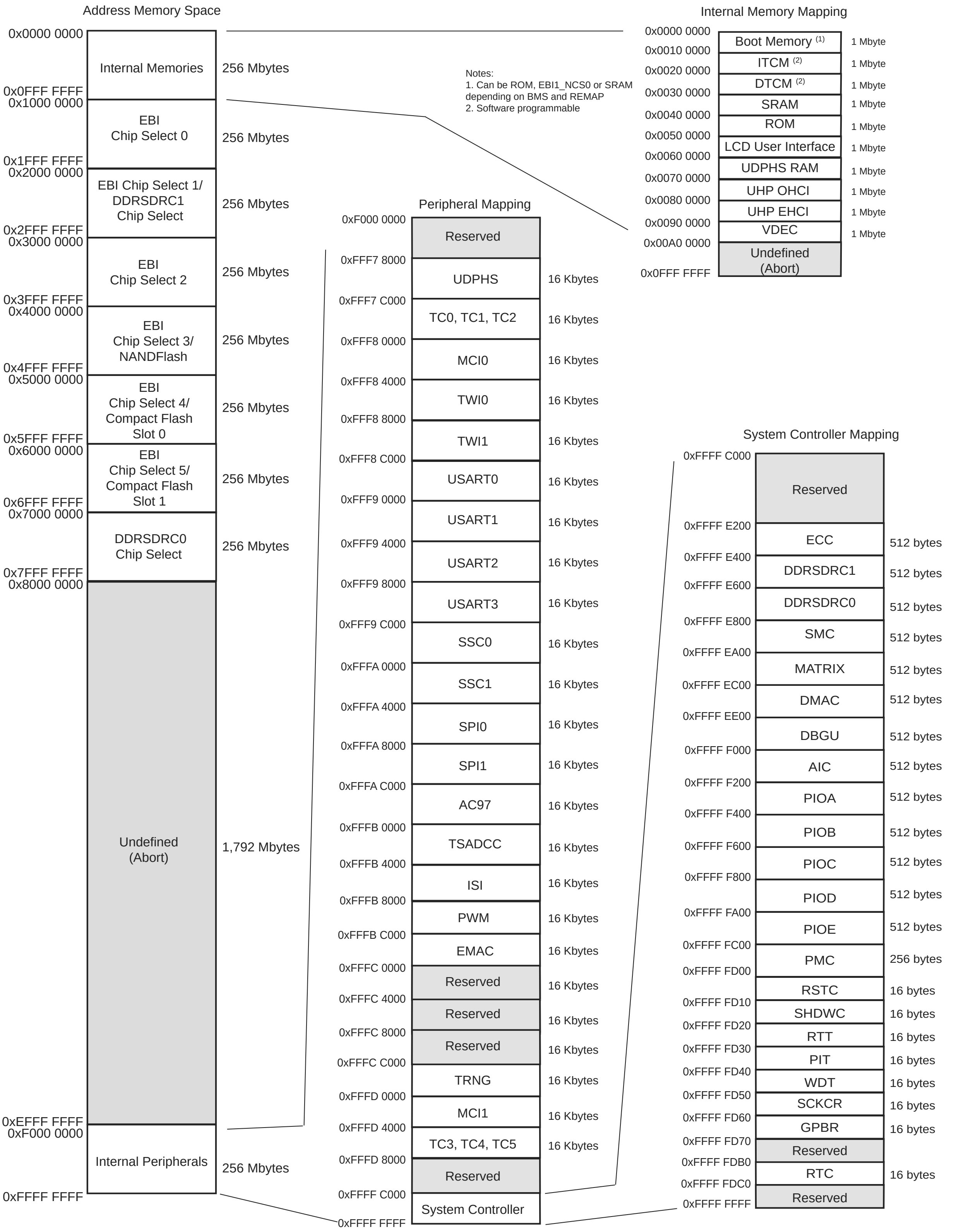
Applications of Embedded - Microprocessors

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	LPDDR, LPSDR, DDR2, SDR, SRAM
Graphics Acceleration	No
Display & Interface Controllers	LCD, Touchscreen, Video Decoder
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	324-TFBGA
Supplier Device Package	324-TFBGA (15x15)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/at91sam9m10c-cu

5. Memories

Figure 5-1. SAM9M10 Memory Mapping



29.6.22 PIO Pull-up Enable Register

Name: PIO_PUER

Address: 0xFFFFF264 (PIOA), 0xFFFFF464 (PIOB), 0xFFFFF664 (PIOC), 0xFFFFF864 (PIOD), 0xFFFFFA64 (PIOE)

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

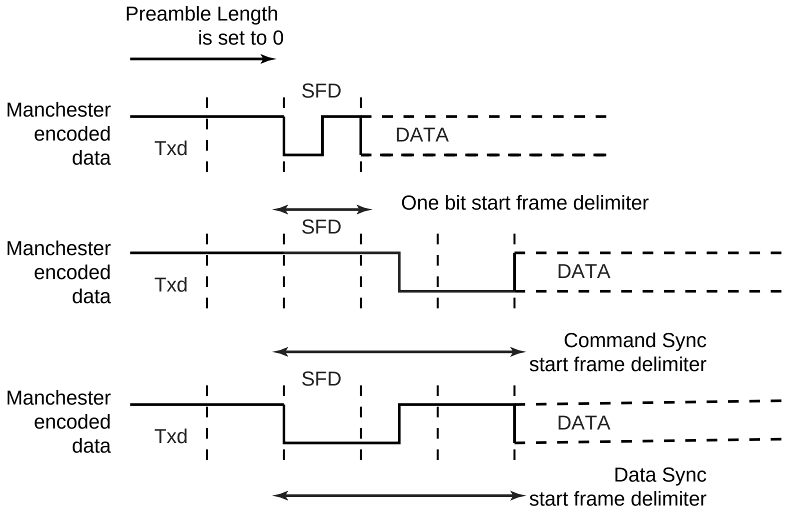
7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0

- **P0–P31: Pull-up Enable**

0: No effect.

1: Enables the pull-up resistor on the I/O line.

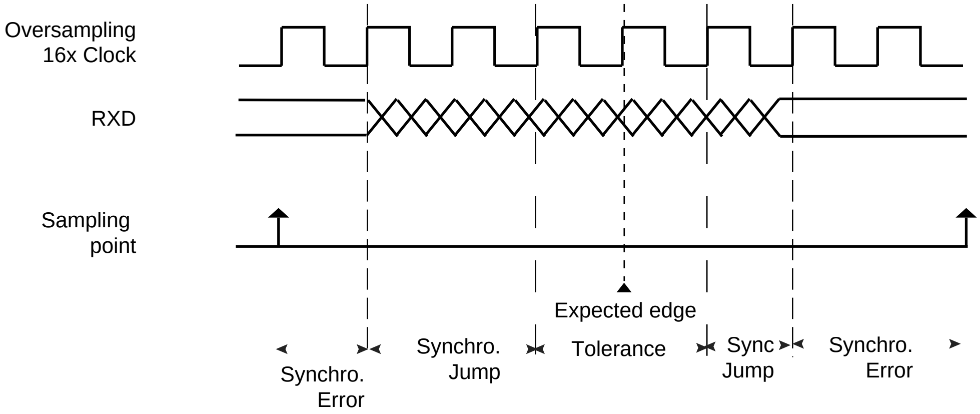
Figure 30-10. Start Frame Delimiter



30.7.3.3 Drift Compensation

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

Figure 30-11. Bit Resynchronization



30.7.3.4 Asynchronous Receiver

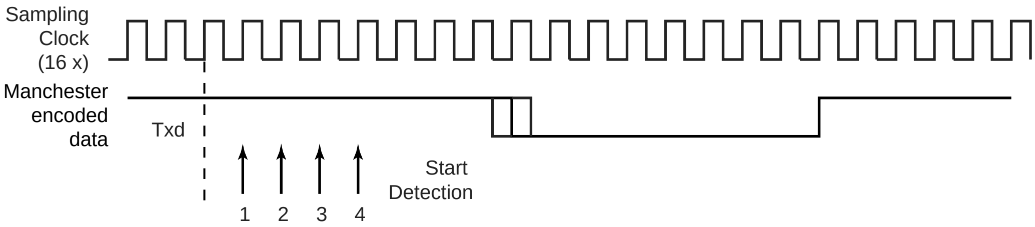
If the USART is programmed in asynchronous operating mode (SYNC = 0), the receiver oversamples the RXD input line. The oversampling is either 16 or 8 times the Baud Rate clock, depending on the OVER bit in the Mode Register (US_MR).

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.

Unlike preamble, the start frame delimiter is shared between Manchester Encoder and Decoder. So, if ONEBIT field is set to 1, only a zero encoded Manchester can be detected as a valid start frame delimiter. If ONEBIT is set to 0, only a sync pattern is detected as a valid start frame delimiter. Decoder operates by detecting transition on incoming stream. If RXD is sampled during one quarter of a bit time at zero, a start bit is detected. See Figure 30-14. The sample pulse rejection mechanism applies.

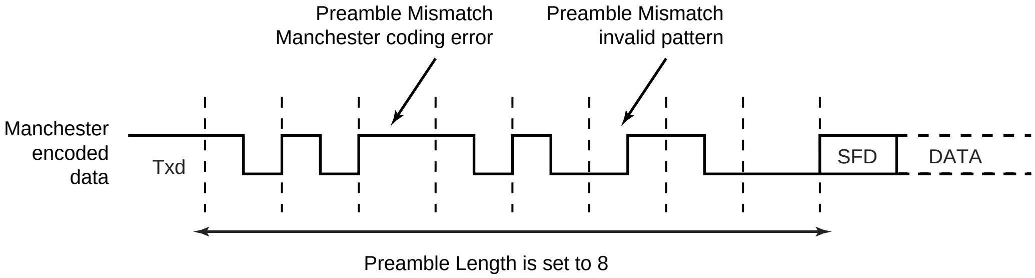
Figure 30-14. Asynchronous Start Bit Detection



The receiver is activated and starts Preamble and Frame Delimiter detection, sampling the data at one quarter and then three quarters. If a valid preamble pattern or start frame delimiter is detected, the receiver continues decoding with the same synchronization. If the stream does not match a valid pattern or a valid start frame delimiter, the receiver re-synchronizes on the next valid edge.The minimum time threshold to estimate the bit value is three quarters of a bit time.

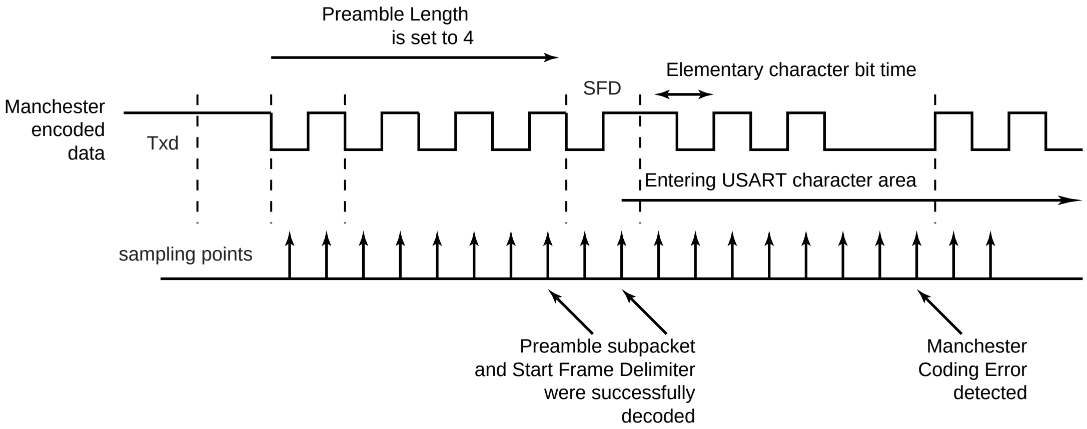
If a valid preamble (if used) followed with a valid start frame delimiter is detected, the incoming stream is decoded into NRZ data and passed to USART for processing. Figure 30-15 illustrates Manchester pattern mismatch. When incoming data stream is passed to the USART, the receiver is also able to detect Manchester code violation. A code violation is a lack of transition in the middle of a bit cell. In this case, MANE flag in US_CSR is raised. It is cleared by writing the Control Register (US_CR) with the RSTSTA bit at 1. See Figure 30-16 for an example of Manchester error detection during data phase.

Figure 30-15. Preamble Pattern Mismatch



Preamble Mismatch Manchester coding error
Preamble Mismatch invalid pattern
Manchester encoded data
Txd
SFD
DATA
Preamble Length is set to 8

Figure 30-16. Manchester Error Flag



Preamble Length is set to 4
SFD
Elementary character bit time
Manchester encoded data
Txd
Entering USART character area
sampling points
Preamble subpacket and Start Frame Delimiter were successfully decoded
Manchester Coding Error detected

- **PARE: Parity Error**

0: No parity error has been detected since the last RSTSTA.

1: At least one parity error has been detected since the last RSTSTA.

- **TIMEOUT: Receiver Time-out**

0: There has not been a time-out since the last Start Time-out command (STTTO in US_CR) or the Time-out Register is 0.

1: There has been a time-out since the last Start Time-out command (STTTO in US_CR).

- **TXEMPTY: Transmitter Empty**

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.

- **ITER/UNRE: Max number of Repetitions Reached or SPI Underrun Error**

– If USART does not operate in SPI Slave Mode (USART_MODE ≠ 0xF):

ITER = 0: Maximum number of repetitions has not been reached since the last RSTSTA.

ITER = 1: Maximum number of repetitions has been reached since the last RSTSTA.

– If USART operates in SPI Slave Mode (USART_MODE = 0xF):

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

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

- **TXBUFE: Transmission Buffer Empty**

0: The signal Buffer Empty from the Transmit PDC channel is inactive.

1: The signal Buffer Empty from the Transmit PDC channel is active.

- **RXBUFF: Reception Buffer Full**

0: The signal Buffer Full from the Receive PDC channel is inactive.

1: The signal Buffer Full from the Receive PDC channel is active.

- **NACK/LINBK Non Acknowledge or LIN Break Sent or LIN Break Received**

– If USART does not operate in LIN Mode (USART_MODE ≠ 0xA AND ≠ 0xB):

0: No Non Acknowledge has not been detected since the last RSTNACK.

1: At least one Non Acknowledge has been detected since the last RSTNACK.

– If USART operates in LIN Master Mode (USART_MODE = 0xA):

0: No LIN Break has been sent since the last RSTSTA.

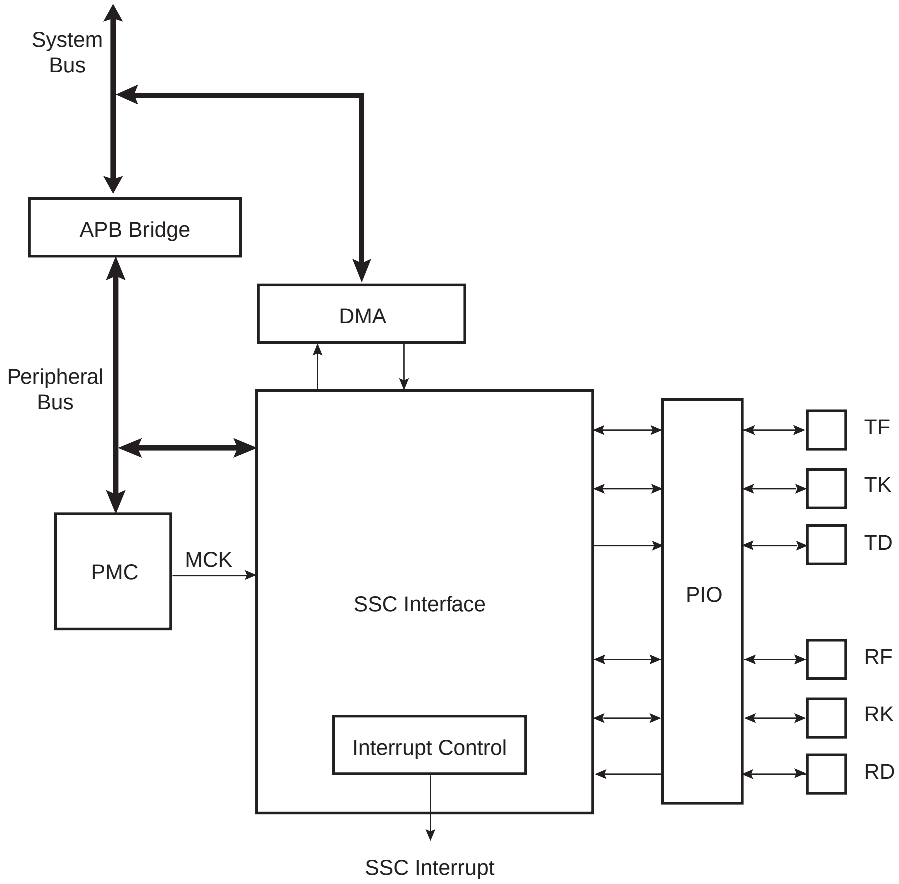
1:At least one LIN Break has been sent since the last RSTSTA.

– If USART operates in LIN Slave Mode (USART_MODE = 0xB):

0: No LIN Break has received sent since the last RSTSTA.

1:At least one LIN Break has been received since the last RSTSTA.

Figure 32-2. Block Diagram (with DMA)

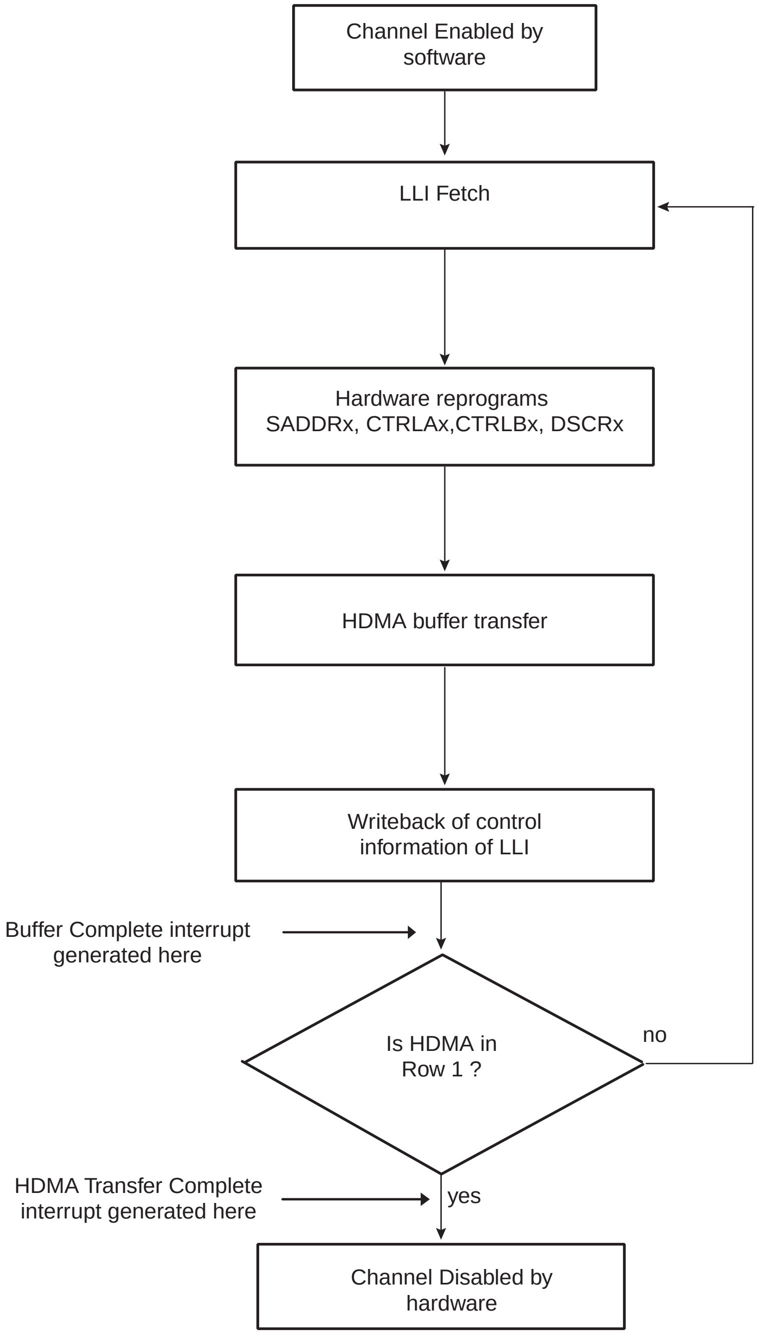


32.4 Application Block Diagram

Figure 32-3. Application Block Diagram

<table>
<tr><td colspan="2">OS or RTOS Driver</td><td>Power Management</td><td>Interrupt Management</td><td>Test Management</td></tr>
<tr><td colspan="5">SSC</td></tr>
<tr><td>Serial AUDIO</td><td>Codec</td><td>Time Slot Management</td><td>Frame Management</td><td>Line Interface</td></tr>
</table>

Figure 34-16. DMAC Transfer Flow for Linked List Source Address and Contiguous Destination Address


Channel Enabled by software
LLI Fetch
Hardware reprograms SADDRx, CTRLAx,CTRLBx, DSCRx
HDMA buffer transfer
Writeback of control information of LLI
Buffer Complete interrupt generated here
Is HDMA in Row 1 ?
no
HDMA Transfer Complete interrupt generated here
yes
Channel Disabled by hardware

36.8.1 Command - Response Operation

After reset, the HSMCI is disabled and becomes valid after setting the MCIEN bit in the HSMCI Control Register (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 MultiMediaCard 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 36-6 and Table 36-7.

Table 36-6. ALL_SEND_CID Command Description

CMD Index	Type	Argument	Resp	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.

Table 36-7. Fields and Values for HSMCI_CMDR

Field	Value
CMDNB (command number)	2 (CMD2)
RSPTYP (response type)	2 (R2: 136 bits response)
SPCMD (special command)	0 (not a special command)
OPCMD (open drain command)	1
MAXLAT (max latency for command to response)	0 (NID cycles ==> 5 cycles)
TRCMD (transfer command)	0 (No transfer)
TRDIR (transfer direction)	X (available only in transfer command)
TRTYP (transfer type)	X (available only in transfer command)
IOSPCMD (SDIO special command)	0 (not a special command)

The HSMCI Argument Register (HSMCI_ARGR) contains the argument field of the command.

To send a command, the user must perform the following steps:

- Fill the HSMCI_ARGR with the command argument.
- Set the HSMCI_CMDR (see Table 36-7).

The command is sent immediately after writing the HSMCI_CMDR.

While the card maintains a busy indication (at the end of a STOP_TRANSMISSION command CMD12, for example), a new command shall not be sent. The NOTBUSY flag in the HSMCI Status Register (HSMCI_SR) is asserted when the card releases the busy indication.

37.6.27.9 Late Collisions Register

Name: EMAC_LCOL

Address: 0xFFFBC05C

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
LCOL							

- **LCOL: Late Collisions**

An 8-bit register counting the number of frames that experience a collision after the slot time (512 bits) has expired. A late collision is counted twice; i.e., both as a collision and a late collision.

39.5 Functional Description

39.5.1 USB V2.0 High Speed Device Port Introduction

The USB V2.0 High Speed Device Port provides communication services between host and attached USB devices. Each device is offered with a collection of communication flows (pipes) associated with each endpoint. Software on the host communicates with a USB Device through a set of communication flows.

39.5.2 USB V2.0 High Speed Transfer Types

A communication flow is carried over one of four transfer types defined by the USB device.

A device provides several logical communication pipes with the host. To each logical pipe is associated an endpoint. Transfer through a pipe belongs to one of the four transfer types:

- Control Transfers: Used to configure a device at attach time and can be used for other device-specific purposes, including control of other pipes on the device.
- Bulk Data Transfers: Generated or consumed in relatively large burst quantities and have wide dynamic latitude in transmission constraints.
- Interrupt Data Transfers: Used for timely but reliable delivery of data, for example, characters or coordinates with human-perceptible echo or feedback response characteristics.
- Isochronous Data Transfers: Occupy a prenegotiated amount of USB bandwidth with a prenegotiated delivery latency. (Also called streaming real time transfers.)

As indicated below, transfers are sequential events carried out on the USB bus.

Endpoints must be configured according to the transfer type they handle.

Table 39-2. USB Communication Flow

Transfer	Direction	Bandwidth	Endpoint Size	Error Detection	Retrying
Control	Bidirectional	Not guaranteed	8,16,32,64	Yes	Automatic
Isochronous	Unidirectional	Guaranteed	8–1024	Yes	No
Interrupt	Unidirectional	Not guaranteed	8–1024	Yes	Yes
Bulk	Unidirectional	Not guaranteed	8–512	Yes	Yes

A response should be made to the first token IN recognized inside a microframe under the following conditions:

- If at least one bank has been validated, the correct DATAx corresponding to the programmed Number Of Transactions per Microframe (NB_TRANS) should be answered. In case of a subsequent missed or corrupted token IN inside the microframe, the USB 2.0 Core available data bank(s) that should normally have been transmitted during that microframe shall be flushed at its end. If this flush occurs, an error condition is flagged (ERR_FLUSH is set in UDPHS_EPTSTAx).
- If no bank is validated yet, the default DATA0 ZLP is answered and underflow is flagged (ERR_FL_ISO is set in UDPHS_EPTSTAx). Then, no data bank is flushed at microframe end.
- If no data bank has been validated at the time when a response should be made for the second transaction of NB_TRANS = 3 transactions microframe, a DATA1 ZLP is answered and underflow is flagged (ERR_FL_ISO is set in UDPHS_EPTSTAx). If and only if remaining untransmitted banks for that microframe are available at its end, they are flushed and an error condition is flagged (ERR_FLUSH is set in UDPHS_EPTSTAx).
- If no data bank has been validated at the time when a response should be made for the last programmed transaction of a microframe, a DATA0 ZLP is answered and underflow is flagged (ERR_FL_ISO is set in UDPHS_EPTSTAx). If and only if the remaining untransmitted data bank for that microframe is available at its end, it is flushed and an error condition is flagged (ERR_FLUSH is set in UDPHS_EPTSTAx).
- If at the end of a microframe no valid token IN has been recognized, no data bank is flushed and no error condition is reported.

At the end of a microframe in which at least one data bank has been transmitted, if less than NB_TRANS banks have been validated for that microframe, an error condition is flagged (ERR_TRANS is set in UDPHS_EPTSTAx).

Cases of Error (in UDPHS_EPTSTAx)

- ERR_FL_ISO: There was no data to transmit inside a microframe, so a ZLP is answered by default.
- ERR_FLUSH: At least one packet has been sent inside the microframe, but the number of token IN received is lesser than the number of transactions actually validated (TX_BK_RDY) and likewise with the NB_TRANS programmed.
- ERR_TRANS: At least one packet has been sent inside the microframe, but the number of token IN received is lesser than the number of programmed NB_TRANS transactions and the packets not requested were not validated.
- ERR_FL_ISO + ERR_FLUSH: At least one packet has been sent inside the microframe, but the data has not been validated in time to answer one of the following token IN.
- ERR_FL_ISO + ERR_TRANS: At least one packet has been sent inside the microframe, but the data has not been validated in time to answer one of the following token IN and the data can be discarded at the microframe end.
- ERR_FLUSH + ERR_TRANS: The first token IN has been answered and it was the only one received, a second bank has been validated but not the third, whereas NB_TRANS was waiting for three transactions.
- ERR_FL_ISO + ERR_FLUSH + ERR_TRANS: The first token IN has been treated, the data for the second Token IN was not available in time, but the second bank has been validated before the end of the microframe. The third bank has not been validated, but three transactions have been set in NB_TRANS.

39.5.8.4 Data OUT

Bulk OUT or Interrupt OUT

Like data IN, data OUT packets are sent by the host during the data or the status stage of control transfer or during an interrupt/bulk/isochronous OUT transfer. Data buffers are sent packet by packet under the control of the application or under the control of the DMA channel.

- **END_TR_EN: End of Transfer Enable (Control)**

Used for OUT transfers only.

0: USB end of transfer is ignored.

1: UDPHS device can put an end to the current buffer transfer.

When set, a BULK or INTERRUPT short packet or the last packet of an ISOCHRONOUS (micro) frame (DATAX) will close the current buffer and the UDPHS_DMASTATUSx register END_TR_ST flag will be raised.

This is intended for UDPHS non-prenegotiated end of transfer (BULK or INTERRUPT) or ISOCHRONOUS microframe data buffer closure.

- **END_B_EN: End of Buffer Enable (Control)**

0: DMA Buffer End has no impact on USB packet transfer.

1: Endpoint can validate the packet (according to the values programmed in the UDPHS_EPTCTLx register AUTO_VALID and SHRT_PCKT fields) at DMA Buffer End, i.e., when the UDPHS_DMASTATUS register BUFF_COUNT reaches 0.

This is mainly for short packet IN validation initiated by the DMA reaching end of buffer, but could be used for OUT packet truncation (discarding of unwanted packet data) at the end of DMA buffer.

- **END_TR_IT: End of Transfer Interrupt Enable**

0: UDPHS device initiated buffer transfer completion will not trigger any interrupt at UDPHS_STATUSx/END_TR_ST rising.

1: An interrupt is sent after the buffer transfer is complete, if the UDPHS device has ended the buffer transfer.

Use when the receive size is unknown.

- **END_BUFFIT: End of Buffer Interrupt Enable**

0: UDPHS_DMA_STATUSx/END_BF_ST rising will not trigger any interrupt.

1: An interrupt is generated when the UDPHS_DMASTATUSx register BUFF_COUNT reaches zero.

- **DESC_LD_IT: Descriptor Loaded Interrupt Enable**

0: UDPHS_DMASTATUSx/DESC_LDST rising will not trigger any interrupt.

1: An interrupt is generated when a descriptor has been loaded from the bus.

- **BURST_LCK: Burst Lock Enable**

0: The DMA never locks bus access.

1: USB packets AHB data bursts are locked for maximum optimization of the bus bandwidth usage and maximization of fly-by AHB burst duration.

- **BUFF_LENGTH: Buffer Byte Length (Write-only)**

This field determines the number of bytes to be transferred until end of buffer. The maximum channel transfer size (64 Kbytes) is reached when this field is 0 (default value). If the transfer size is unknown, this field should be set to 0, but the transfer end may occur earlier under UDPHS device control.

When this field is written, The UDPHS_DMASTATUSx register BUFF_COUNT field is updated with the write value.

Note: Bits [31:2] are only writable when issuing a channel Control Command other than “Stop Now”.

Note: For reliability it is highly recommended to wait for both UDPHS_DMASTATUSx register CHAN_ACT and CHAN_ENB flags are at 0, thus ensuring the channel has been stopped before issuing a command other than “Stop Now”.

41.5 Product Dependencies

41.5.1 Power Management

The TSADC controller is not continuously clocked. The programmer must first enable the TSADC controller Clock in the Power Management Controller (PMC) before using the TSADC controller. However, if the application does not require TSADC controller operations, the TSADC controller clock can be stopped when not needed and be restarted later.

Configuring the TSADC controller does not require the TSADC controller clock to be enabled.

41.5.2 Interrupt Sources

The TSADCC interrupt line is connected on one of the internal sources of the Advanced Interrupt Controller. Using the TSADCC interrupt requires the AIC to be programmed first.

Table 41-2. Peripheral IDs

Instance	ID
TSADCC	20

41.5.3 Analog Inputs

The analog input pins can be multiplexed with PIO lines. In this case, the assignment of the TSADCC input is automatically done as soon as the corresponding channel is enabled by writing the register TSADCC_CHER. By default, after reset, the PIO lines are configured as input with its pull-up enabled and the TSADCC inputs are connected to the GND.

41.5.4 I/O Lines

The pin TSADTRG may be shared with other peripheral functions through the PIO Controller. In this case, the PIO Controller should be set accordingly to assign the pin TSADTRG to the TSADCC function.

Table 41-3. I/O Lines

Instance	Signal	I/O Line	Peripheral
TSADCC	TSADTRG	PD28	A

41.5.5 Conversion Performances

For performance and electrical characteristics of the TSADCC, see the section “Electrical Characteristics” of the datasheet.

41.6 Analog-to-digital Converter Functional Description

The TSADCC embeds a Successive Approximation Register (SAR) Analog-to-Digital Converter (ADC). The ADC supports 8-bit or 10-bit resolutions.

The conversion is performed on a full range between 0V and the reference voltage pin TSADVREF. Analog inputs between these voltages convert to values based on a linear conversion.

41.6.1 ADC Resolution

The ADC supports 8-bit or 10-bit resolutions. The 8-bit selection is performed by setting the bit LOWRES in the TSADCC Mode Register. See Section 41.11.2 “TSADCC Mode Register” on page 1029.

By default, after a reset, the resolution is the highest and the DATA field in the “TSADCC Channel Data Register x (x = 0..7)” are fully used.

44.12.11 LCD Control Register 2

Name: LCDCON2

Address: 0x00500804

Access: Read/Write

31	30	29	28	27	26	25	24
MEMOR		–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
CLKMOD	–	–	INVDVAL	INVCLK	INVLINE	INVFRAME	INVVD
7	6	5	4	3	2	1	0
PIXELSIZE			IFWIDTH		SCANMOD	DISTYPE	

- **DISTYPE: Display Type**

DISTYPE		
0	0	STN Monochrome
0	1	STN Color
1	0	TFT
1	1	Reserved

- **SCANMOD: Scan Mode**

0: Single Scan

1: Dual Scan

- **IFWIDTH: Interface width (STN)**

IFWIDTH		
0	0	4-bit (Only valid in single scan STN mono or color)
0	1	8-bit (Only valid in STN mono or Color)
1	0	16-bit (Only valid in dual scan STN mono or color)
1	1	Reserved

- **PIXELSIZE: Bits per pixel**

PIXELSIZE			
0	0	0	1 bit per pixel
0	0	1	2 bits per pixel
0	1	0	4 bits per pixel
0	1	1	8 bits per pixel
1	0	0	16 bits per pixel
1	0	1	24 bits per pixel, packed (Only valid in TFT mode)
1	1	0	24 bits per pixel, unpacked (Only valid in TFT mode)
1	1	1	Reserved

- **REFFIELD: Indicates Which Field Should Be Used As Reference**

Used only if REF_FRAMES =’0’ in Decoder control register 1 (picture parameters).

0: Bottom field

1: Top field

- **FORWMODE: Coding Mode of Forward Reference Picture**

0: Progressive

1: Interlaced

Note: For backward reference picture the coding mode is always the same as for current picture.

- **PICFIELD: Picture Field**

If field structure is enabled this bit informs which one of the fields is being decoded:

0: Bottom field

1: Top field

- **PICTYPE: Picture Type**

0: Intra type (I)

1: Inter type (P)

- **PICBEN: B Picture Enable**

B picture enable for current picture:

0: Picture type is I or P depending on PICTYPE.

1: Picture type is BI (vc1)/D (mpeg1) or B depending on PICTYPE.

- **PICSTRUCT: Structure of the Current Picture**

Structure of the current picture (residual structure).

0: Frame structure, For H.264 this means MBAFF structured picture for interlaced sequence.

1: Field structure

- **PICMODE: Coding mode of Current Picture**

0: Progressive

1: Interlaced

- **RLCEN: RLC Mode Enable**

0: Decode from input stream (VLC mode).

1: Decode from RLC input data.

45.6.9 Decoder Control Register 5 (H.264 Control)

Name: VDEC_CTLR5

Address: 0x00900020

Access: Read/Write

31	30	29	28	27	26	25	24
CONS_INTRA	FILT_CTRL	RD_PIC	T8X8FE	REF_PIC_LEN			

23	22	21	20	19	18	17	16
REF_PIC_LEN							IDREN

15	14	13	12	11	10	9	8
IDR_PIC_ID							

7	6	5	4	3	2	1	0
IDR_PIC_ID							

- **IDR_PIC_ID: IDR Picture**

Identifies IDR (instantaneous decoding refresh) picture.

- **IDREN: IDR Picture Enable**

IDR (instantaneous decoding refresh) picture flag.

- **REF_PIC_LEN: Reference Picture Length**

Length of decoded reference picture marking bits.

- **T8X8FE: 8x8 Transform Flag Enable**

8x8 transform flag enable

- **RD_PIC: Redundant Picture Present**

Specifies whether redundant picture count syntax elements are present in the slice header.

- **FILT_CTRL: Extra Variables Controlling Characteristics of The Deblocking Filter**

Indicates whether extra variables controlling characteristics of the deblocking filter are present in the slice header.

- **CONS_INTRA: Intra in Prediction**

1: Specifies that intra prediction uses only neighboring intra macroblocks in prediction.

0: Also neighboring inter macroblocks are used in intra prediction process.

45.7.19 Reference Picture Index 6 Base Address (MPEG-4/H.263 TRB/TRD Delta -1)

Name: VDEC_PIDXBA6

Address: 0x00900050

Access: Read/Write

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

- **TRB_PER_TRD: TRB per TRD**

Defines MPEG-4 reference distance syntax TRB/TRD when delta value -1 is used.

45.9.20 Reference Picture Index 5 Base Address

Name: VDEC_PIDXBA5

Address: 0x0090004C

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	ICOMP3

23	22	21	20	19	18	17	16
IScale3							

15	14	13	12	11	10	9	8
ISHIFT3							

7	6	5	4	3	2	1	0
ISHIFT3							

- **ISHIFT3: Ishift Value**

IShift value used for VC-1 intensity compensation.

- **IScale3: IScale Value**

IScale value used for VC-1 intensity compensation.

- **ICOMP3: Intensity Compensation Value**

Intensity compensation 3 enable.

45.11.8 Post Processor Output C Picture Base Address

Name: VDEC_POCBA

Access: Read/Write

31	30	29	28	27	26	25	24
BA							

23	22	21	20	19	18	17	16
BA							

15	14	13	12	11	10	9	8
BA							

7	6	5	4	3	2	1	0
BA						–	–

- **BA: Base Address**

Base address for post-processing output chrominance picture (interleaved chrominance).

46.15.1.2 Timing Conditions

Timings are given assuming a capacitance load on MISO, SPCK and MOSI as defined in Table 46-33.

Table 46-33. Capacitance Load for MISO, SPCK and MOSI (product dependent)

Supply	Corner	
	Max	Min
3.3V	40 pF	5 pF
1.8V	20 pF	5 pF

46.15.1.3 Timing Extraction

In Figure 46-7 “SPI Master Mode 1 and 2” and Figure 46-8 “SPI Master Mode 0 and 3”, the MOSI line shifting edge is represented with a hold time = 0. However, it is important to note that for this device, the MISO line is sampled prior to the MOSI line shifting edge. As shown in Figure 46-6 “MISO Capture in Master Mode”, the device sampling point extends the propagation delay (tp) for slave and routing delays to more than half the SPI clock period, whereas the common sampling point allows only less than half the SPI clock period.

As an example, an SPI Slave working in Mode 0 is safely driven if the SPI Master is configured in Mode 0.

Figure 46-6. MISO Capture in Master Mode

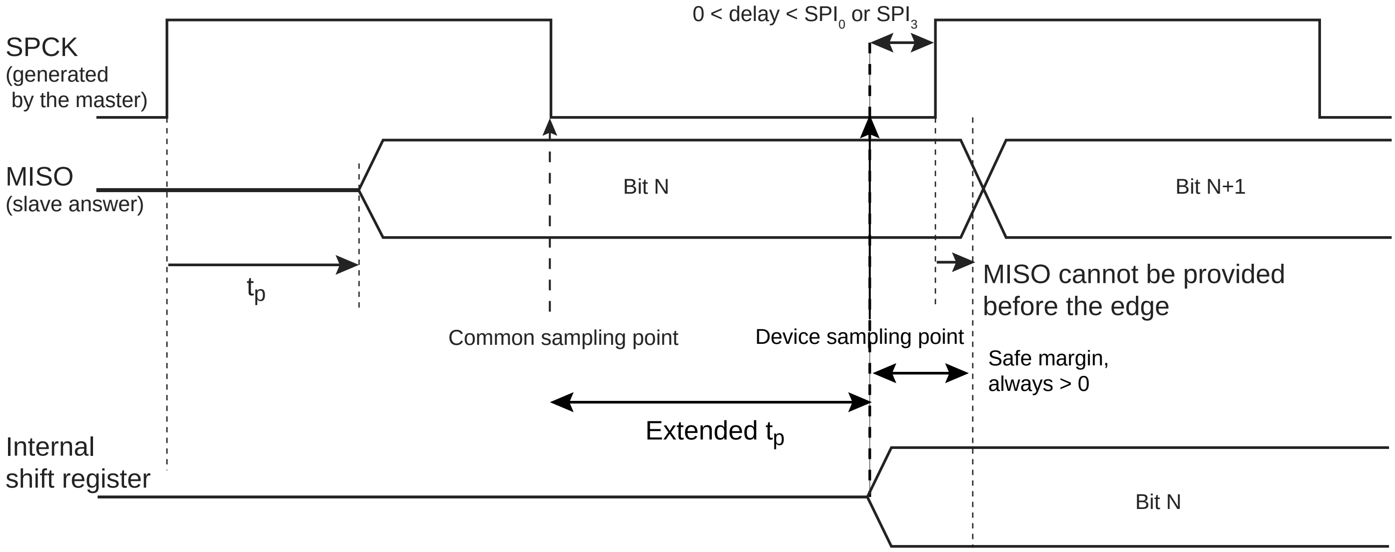
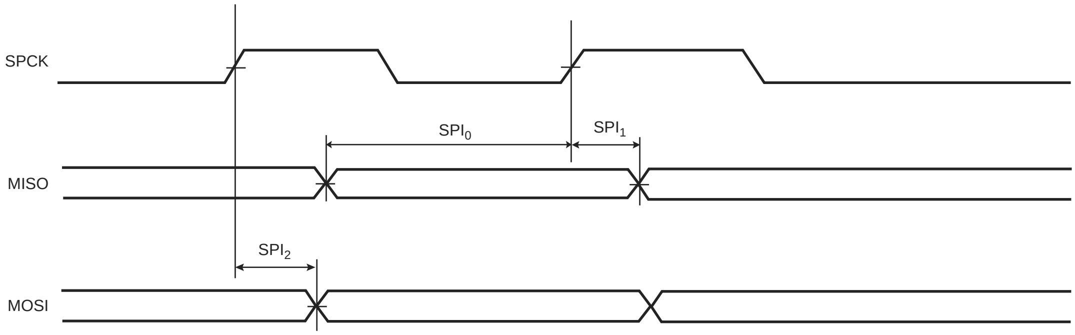


Figure 46-7. SPI Master Mode 1 and 2


SPCK
SPI_0
SPI_1
MISO
SPI_2
MOSI