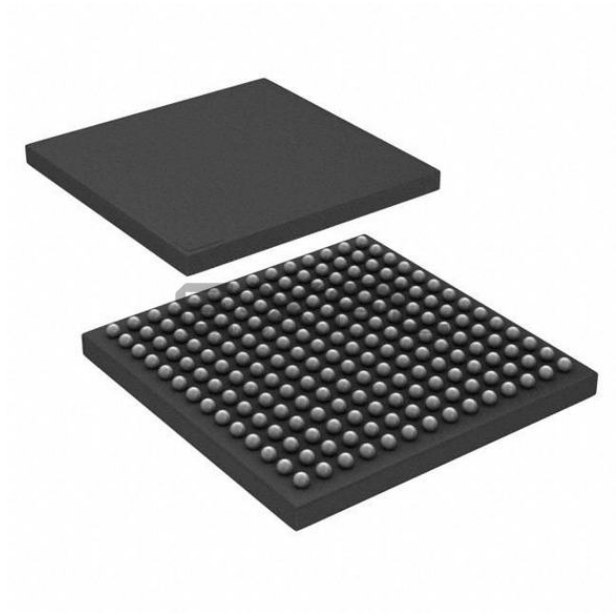


Welcome to E-XFL.COM

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	Obsolete
Core Processor	ARM® Cortex®-A5
Number of Cores/Bus Width	1 Core, 32-Bit
Speed	500MHz
Co-Processors/DSP	Multimedia; NEON™ MPE
RAM Controllers	LPDDR1, LPDDR2, LPDDR3, DDR2, DDR3, DDR3L, QSPI
Graphics Acceleration	Yes
Display & Interface Controllers	Keyboard, LCD, Touchscreen
Ethernet	10/100Mbps (1)
SATA	-
USB	USB 2.0 + HSIC
Voltage - I/O	3.3V
Operating Temperature	-40°C ~ 85°C (TA)
Security Features	ARM TZ, Boot Security, Cryptography, RTIC, Secure Fusebox, Secure JTAG, Secure Memory, Secure RTC
Package / Case	196-TFBGA, CSBGA
Supplier Device Package	196-TFBGA (11x11)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsama5d21b-cu

Table 6-2: Pin Description (Continued)

289-pin BGA	256-pin BGA	196-pin BGA	Power Rail	I/O Type	Primary		Alternate		PIO peripheral				Reset State (Signal, Dir, PU, PD, HiZ, ST)[1][2]
					Signal	Dir	Signal	Dir	Func	Signal	Dir	IO Set	
C4	G7	C2	VDDIOP0	GPIO_IO	PB17	I/O	–	–	A	LCDDAT6	O	1	PIO, I, PU, ST
									B	A6	O	1	
									C	RD1	I	2	
									D	I2SC1_DI0	I	1	
									E	QSPI1_IO1	I/O	3	
									F	GRXER	I	3	
A3	A2	D4	VDDIOP0	GPIO_IO	PB18	I/O	–	–	A	LCDDAT7	O	1	PIO, I, PU, ST
									B	A7	O	1	
									C	RK1	I/O	2	
									D	I2SC1_DO0	O	1	
									E	QSPI1_IO2	I/O	3	
									F	GRX0	I	3	
D4	H7	C4	VDDIOP0	GPIO_IO	PB19	I/O	–	–	A	LCDDAT8	O	1	PIO, I, PU, ST
									B	A8	O	1	
									C	RF1	I/O	2	
									D	TIOA3	I/O	2	
									E	QSPI1_IO3	I/O	3	
									F	GRX1	I	3	
B3	A1	C3	VDDIOP0	GPIO	PB20	I/O	–	–	A	LCDDAT9	O	1	PIO, I, PU, ST
									B	A9	O	1	
									C	TK0	I/O	1	
									D	TIOB3	I/O	2	
									E	PCK1	O	4	
									F	GTX0	O	3	
A2	D2	D1	VDDIOP0	GPIO	PB21	I/O	–	–	A	LCDDAT10	O	1	PIO, I, PU, ST
									B	A10	O	1	
									C	TF0	I/O	1	
									D	TCLK3	I	2	
									E	FLEXCOM3_IO2	I/O	3	
									F	GTX1	O	3	
C3	G5	D2	VDDIOP0	GPIO	PB22	I/O	–	–	A	LCDDAT11	O	1	PIO, I, PU, ST
									B	A11	O	1	
									C	TD0	O	1	
									D	TIOA2	I/O	2	
									E	FLEXCOM3_IO1	I/O	3	
									F	GMDC	O	3	

14.5.4 L2CC Auxiliary Control Register

Name: L2CC_ACR

Address: 0x00A00104

Access: Read/Write in Secure mode

Read-only in Non-secure mode

31	30	29	28	27	26	25	24
–	–	IPEN	DPEN	NSIAC	NSLEN	CRPOL	FWA

23	22	21	20	19	18	17	16
FWA	SAOEN	PEN	EMBEN	WAYSIZE			ASS

15	14	13	12	11	10	9	8
–	–	SAIE	EXCC	SBDLE	HPSO	–	–

7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	–

Note: The L2 Cache Controller (L2CC) must be disabled in the L2CC Control Register prior to any write access to this register.

HPSO: High Priority for SO and Dev Reads Enable

0: Strongly Ordered and Device reads have lower priority than cacheable accesses when arbitrated in the L2CC master ports. This is the default value.

1: Strongly Ordered and Device reads get the highest priority when arbitrated in the L2CC master ports.

SBDLE: Store Buffer Device Limitation Enable

0: Store buffer device limitation is disabled. Device writes can take all slots in the store buffer. This is the default value.

1: Store buffer device limitation is enabled.

EXCC: Exclusive Cache Configuration

0: Disabled. This is the default value.

1: Enabled.

SAIE: Shared Attribute Invalidate Enable

0: Shared invalidate behavior is disabled. This is the default value.

1: Shared invalidate behavior is enabled if the Shared Attribute Override Enable bit is not set.

Shared invalidate behavior is enabled if both:

- Shareable Attribute Invalidate Enable bit is set in the Auxiliary Control Register, bit[13]
- Shared Attribute Override Enable bit is not set in the Auxiliary Control Register, bit[22]

ASS: Associativity

0: 8-way.This is the default value.

1: Reserved.

WAYSIZE: Way Size

Value	Name	Description
0x0	RESERVED	Reserved
0x1	16KB_WAY	16-Kbyte way set associative
0x2	RESERVED	Reserved
0x3	RESERVED	Reserved

A set of Bus Matrix security registers allows to specify, for each AHB slave, slave security region or slave security area, the security mode required to access this slave, slave security region or slave security area.

Additional Bus Matrix security registers allow to specify, for each APB slave, the security mode required to access this slave (see Section 18.13.15 “Security Peripheral Select x Registers”).

See Section 18.13.12 “Security Slave Registers”.

The Bus Matrix registers can only be accessed in Secure mode.

The Bus Matrix propagates the AHB security bit down to the AHB slaves to let them perform additional security checks, and the Bus Matrix itself allows, or not, the access to the slaves by means of its TrustZone embedded controller.

Access violations may be reported either by an AHB slave through the bus error response (example from the AHB/APB Bridge), or by the Bus Matrix embedded TrustZone controller. In both cases, a bus error response is sent to the offending master and the error is flagged in the Master Error Status Register. An interrupt can be sent to the Secure world, if it has been enabled for that master by writing into the Master Error Interrupt Enable Register. Thus, the offending master is identified. The offending address is registered in the Master Error Address Registers, so that the slave and the targeted security region are also known.

Depending on the hardware parameters and software configuration, the address space of each AHB slave security region may or may not be split into two parts, one belonging to the Secure world and the other one to the Normal world.

Five different security types of AHB slaves are supported. The number of security regions is set by design for each slave, independently, from 1 to 8, totalling from 1 up to 16 security areas for security configurable slaves.

18.12.1 Security Types of AHB Slaves

18.12.1.1 Principles

The Bus Matrix supports five different security types of AHB slaves: two fixed types and three configurable types. The security type of an AHB slave is set at hardware design among the following:

- Always Non-secured
- Always Secured
- Internal Securable
- External Securable
- Scalable Securable

The security type is set at hardware design on a per-master and a per-slave basis. **Always Non-secured** and **Always Secured** security types are not software configurable.

The different security types have the following characteristics:

- **Always Non-secured** slaves have no security mode access restriction. Their address space is precisely set by design. Any out-of-address range access is denied and reported.
- **Always Secured** slaves can only be accessed by a secure master request. Their address space is precisely set by design. Any non-secure or out-of-address range access is denied and reported.
- **Internal Securable** is intended for internal memories such as RAM, ROM or embedded Flash. The Internal Securable slave has one slave region which has a hardware fixed base address and Security Region Top. This slave region may be split through software configuration into one Non-secured area plus one Secured area. Inside the slave security region, the split boundary is programmable in powers of 2 from 4 Kbytes up to the full slave security region address space. The security area located below the split boundary may be configured as the Non-secured or the Secured one. The Securable area may be independently configured as Read Secured and/or Write Secured. Any access with security or address range violation is denied and reported.
- **External Securable** is intended for external memories on the EBI, such as DDR, SDRAM, external ROM or NAND Flash. The External Securable slave has identical features as the Internal Securable slave, plus the ability to configure each of its slave security region address space sizes according to the external memory parts used. This avoids mirroring Secured areas into Non-secured areas, and further restricts the overall accessible address range. Any access with security or configured address range violation is denied and reported.
- **Scalable Securable** is intended for external memories with a dedicated slave, such as DDR. The Scalable Securable slave is divided into a fixed number of scalable, equally sized, and contiguous security regions. Each of them can be split in the same way as for Internal or External Securable slaves. The security region size must be configured by software, so that the equally-sized regions fill the actual available memory. This avoids mirroring Secured areas into Non-secured areas, and further restricts the overall accessible address range. Any access with security or configured address range violation is denied and reported.

As the security type is set at hardware design on a per-master and per-slave basis, it is possible to set some slave access security as configurable from one or some particular masters, and to set the access as Always Secured from all the other masters.

As the security type is set by design at the slave region level, different security region types can be mixed inside a single slave.

37.20.27 PMECC Error Location Interrupt Disable Register

Name: HSMC_ELIDR

Address: 0xF8014518

Access: Write-only

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

DONE: Computation Terminated Interrupt Disable

0: No effect

1: Interrupt disable.

Table 39-55: Register Mapping (Continued)

Offset	Register	Name	Access	Reset
0x0000039C	High-End Overlay Configuration Register 4	LCDC_HEOCFG4	Read/ Write	0x00000000
0x000003A0	High-End Overlay Configuration Register 5	LCDC_HEOCFG5	Read/ Write	0x00000000
0x000003A4	High-End Overlay Configuration Register 6	LCDC_HEOCFG6	Read/ Write	0x00000000
0x000003A8	High-End Overlay Configuration Register 7	LCDC_HEOCFG7	Read/ Write	0x00000000
0x000003AC	High-End Overlay Configuration Register 8	LCDC_HEOCFG8	Read/ Write	0x00000000
0x000003B0	High-End Overlay Configuration Register 9	LCDC_HEOCFG9	Read/ Write	0x00000000
0x000003B4	High-End Overlay Configuration Register 10	LCDC_HEOCFG10	Read/ Write	0x00000000
0x000003B8	High-End Overlay Configuration Register 11	LCDC_HEOCFG11	Read/ Write	0x00000000
0x000003BC	High-End Overlay Configuration Register 12	LCDC_HEOCFG12	Read/ Write	0x00000000
0x000003C0	High-End Overlay Configuration Register 13	LCDC_HEOCFG13	Read/ Write	0x00000000
0x000003C4	High-End Overlay Configuration Register 14	LCDC_HEOCFG14	Read/ Write	0x00000000
0x000003C8	High-End Overlay Configuration Register 15	LCDC_HEOCFG15	Read/ Write	0x00000000
0x000003CC	High-End Overlay Configuration Register 16	LCDC_HEOCFG16	Read/ Write	0x00000000
0x000003D0	High-End Overlay Configuration Register 17	LCDC_HEOCFG17	Read/ Write	0x00000000
0x000003D4	High-End Overlay Configuration Register 18	LCDC_HEOCFG18	Read/ Write	0x00000000
0x000003D8	High-End Overlay Configuration Register 19	LCDC_HEOCFG19	Read/ Write	0x00000000
0x000003DC	High-End Overlay Configuration Register 20	LCDC_HEOCFG20	Read/ Write	0x00000000
0x000003E0	High-End Overlay Configuration Register 21	LCDC_HEOCFG21	Read/ Write	0x00000000
0x000003E4	High-End Overlay Configuration Register 22	LCDC_HEOCFG22	Read/ Write	0x00000000
0x000003E8	High-End Overlay Configuration Register 23	LCDC_HEOCFG23	Read/ Write	0x00000000
0x000003EC	High-End Overlay Configuration Register 24	LCDC_HEOCFG24	Read/ Write	0x00000000
0x000003F0	High-End Overlay Configuration Register 25	LCDC_HEOCFG25	Read/ Write	0x00000000

Table 39-55: Register Mapping (Continued)

Offset	Register	Name	Access	Reset
0x000003F4	High-End Overlay Configuration Register 26	LCDC_HEOCFG26	Read/ Write	0x00000000
0x000003F8	High-End Overlay Configuration Register 27	LCDC_HEOCFG27	Read/ Write	0x00000000
0x000003FC	High-End Overlay Configuration Register 28	LCDC_HEOCFG28	Read/ Write	0x00000000
0x00000400	High-End Overlay Configuration Register 29	LCDC_HEOCFG29	Read/ Write	0x00000000
0x00000404	High-End Overlay Configuration Register 30	LCDC_HEOCFG30	Read/ Write	0x00000000
0x00000408	High-End Overlay Configuration Register 31	LCDC_HEOCFG31	Read/ Write	0x00000000
0x0000040C	High-End Overlay Configuration Register 32	LCDC_HEOCFG32	Read/ Write	0x00000000
0x00000410	High-End Overlay Configuration Register 33	LCDC_HEOCFG33	Read/ Write	0x00000000
0x00000414	High-End Overlay Configuration Register 34	LCDC_HEOCFG34	Read/ Write	0x00000000
0x00000418	High-End Overlay Configuration Register 35	LCDC_HEOCFG35	Read/ Write	0x00000000
0x0000041C	High-End Overlay Configuration Register 36	LCDC_HEOCFG36	Read/ Write	0x00000000
0x00000420	High-End Overlay Configuration Register 37	LCDC_HEOCFG37	Read/ Write	0x00000000
0x00000424	High-End Overlay Configuration Register 38	LCDC_HEOCFG38	Read/ Write	0x00000000
0x00000428	High-End Overlay Configuration Register 39	LCDC_HEOCFG39	Read/ Write	0x00000000
0x0000042C	High-End Overlay Configuration Register 40	LCDC_HEOCFG40	Read/ Write	0x00000000
0x00000430	High-End Overlay Configuration Register 41	LCDC_HEOCFG41	Read/ Write	0x00000000
0x00000434–0x0000053C	Reserved	–	–	–
0x00000540	Post Processing Channel Enable Register	LCDC_PPCHER	Write-only	–
0x00000544	Post Processing Channel Disable Register	LCDC_PPCHDR	Write-only	–
0x00000548	Post Processing Channel Status Register	LCDC_PPCHSR	Read-only	0x00000000
0x0000054C	Post Processing Interrupt Enable Register	LCDC_PPIER	Write-only	–
0x00000550	Post Processing Interrupt Disable Register	LCDC_PPIDR	Write-only	–
0x00000554	Post Processing Interrupt Mask Register	LCDC_PPIMR	Read-only	0x00000000
0x00000558	Post Processing Interrupt Status Register	LCDC_PPISR	Read-only	0x00000000
0x0000055C	Post Processing Head Register	LCDC_PPHEAD	Read/ Write	0x00000000

The recommended means to disable DMA are as follows:

```
// Reset IP UDPHS
    AT91C_BASE_UDPHS->UDPHS_CTRL &= ~AT91C_UDPHS_EN_UDPHS;
    AT91C_BASE_UDPHS->UDPHS_CTRL |= AT91C_UDPHS_EN_UDPHS;
// With OR without DMA !!!
    for( i=1; i<=((AT91C_BASE_UDPHS->UDPHS_IPFEATURES &
AT91C_UDPHS_DMA_CHANNEL_NBR)>>4); i++ ) {
// RESET endpoint canal DMA:
        // DMA stop channel command
        AT91C_BASE_UDPHS->UDPHS_DMA[i].UDPHS_DMACONTROL = 0;  // STOP
command
// Disable endpoint
        AT91C_BASE_UDPHS->UDPHS_EPT[i].UDPHS_EPTCTLDIS |= 0XFFFFFFFF;
// Reset endpoint config
        AT91C_BASE_UDPHS->UDPHS_EPT[i].UDPHS_EPTCTLCFG = 0;
// Reset DMA channel (Buff count and Control field)
        AT91C_BASE_UDPHS->UDPHS_DMA[i].UDPHS_DMACONTROL = 0x02;  // NON
STOP command
// Reset DMA channel 0 (STOP)
        AT91C_BASE_UDPHS->UDPHS_DMA[i].UDPHS_DMACONTROL = 0;  // STOP
command
// Clear DMA channel status (read the register for clear it)
        AT91C_BASE_UDPHS->UDPHS_DMA[i].UDPHS_DMASTATUS =
AT91C_BASE_UDPHS->UDPHS_DMA[i].UDPHS_DMASTATUS;
}
```

41.6.10 Handling Transactions with USB V2.0 Device Peripheral

41.6.10.1 Setup Transaction

The setup packet is valid in the DPR while RX_SETUP is set. Once RX_SETUP is cleared by the application, the UDPHS accepts the next packets sent over the device endpoint.

When a valid setup packet is accepted by the UDPHS:

- The UDPHS device automatically acknowledges the setup packet (sends an ACK response)
- Payload data is written in the endpoint
- Sets the RX_SETUP interrupt
- The BYTE_COUNT field in the UDPHS_EPTSTAx register is updated

An endpoint interrupt is generated while RX_SETUP in the UDPHS_EPTSTAx register is not cleared. This interrupt is carried out to the microcontroller if interrupts are enabled for this endpoint.

Thus, firmware must detect RX_SETUP polling UDPHS_EPTSTAx or catching an interrupt, read the setup packet in the FIFO, then clear the RX_SETUP bit in the UDPHS_EPTCLRSTA register to acknowledge the setup stage.

If STALL_SNT was set to 1, then this bit is automatically reset when a setup token is detected by the device. Then, the device still accepts the setup stage. (Refer to Section 41.6.10.5 “STALL”).

41.6.10.2 NYET

NYET is a High Speed only handshake. It is returned by a High Speed endpoint as part of the PING protocol.

High Speed devices must support an improved NAK mechanism for Bulk OUT and control endpoints (except setup stage). This mechanism allows the device to tell the host whether it has sufficient endpoint space for the next OUT transfer (refer to USB 2.0 spec 8.5.1 NAK Limiting via Ping Flow Control).

The NYET/ACK response to a High Speed Bulk OUT transfer and the PING response are automatically handled by hardware in the UDPHS_EPTCTLx register (except when the user wants to force a NAK response by using the NYET_DIS bit).

If the endpoint responds instead to the OUT/DATA transaction with an NYET handshake, this means that the endpoint accepted the data but does not have room for another data payload. The host controller must return to using a PING token until the endpoint indicates it has space available.

Figure 41-17: Stall Handshake Data OUT Transfer

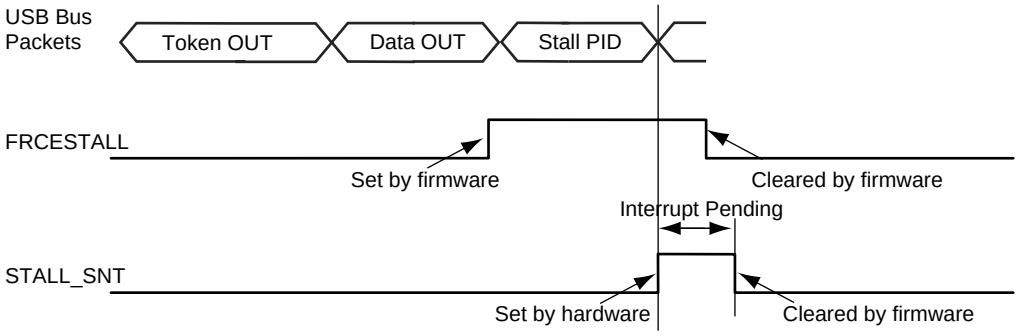
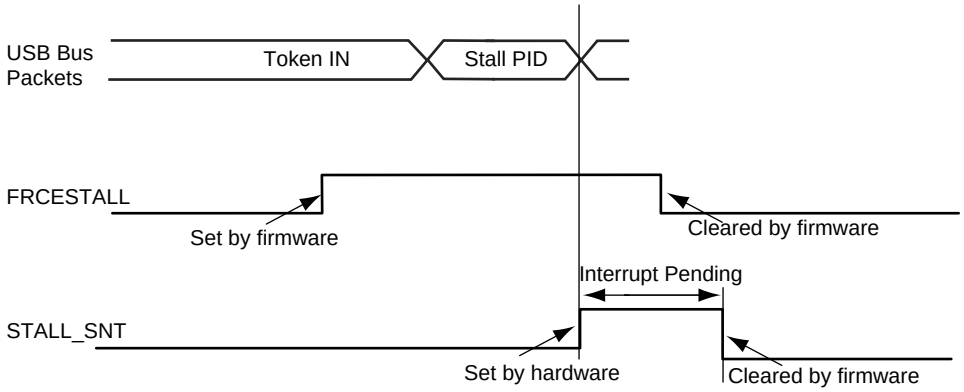


Figure 41-18: Stall Handshake Data IN Transfer



41.6.11 Speed Identification

The high speed reset is managed by hardware.

At the connection, the host makes a reset which could be a classic reset (full speed) or a high speed reset.

At the end of the reset process (full or high), the ENDRESET interrupt is generated.

Then the CPU should read the SPEED bit in UDPHS_INTSTAx to ascertain the speed mode of the device.

41.6.12 USB V2.0 High Speed Global Interrupt

Interrupts are defined in Section 41.7.3 “UDPHS Interrupt Enable Register” (UDPHS_IEN) and in Section 41.7.4 “UDPHS Interrupt Status Register” (UDPHS_INTSTA).

41.6.13 Endpoint Interrupts

Interrupts are enabled in UDPHS_IEN (refer to Section 41.7.3 “UDPHS Interrupt Enable Register”) and individually masked in UDPHS_EPTCTLENBx (refer to Section 41.7.9 “UDPHS Endpoint Control Enable Register (Control, Bulk, Interrupt Endpoints)”).

Table 41-5: Endpoint Interrupt Source Masks

SHRT_PCKT	Short Packet Interrupt
BUSY_BANK	Busy Bank Interrupt
NAK_OUT	NAKOUT Interrupt
NAK_IN/ERR_FLUSH	NAKIN/Error Flush Interrupt
STALL_SNT/ERR_CRC_NTR	Stall Sent/CRC error/Number of Transaction Error Interrupt
RX_SETUP/ERR_FL_ISO	Received SETUP/Error Flow Interrupt
TXRDY_TRER	TX Packet Read/Transaction Error Interrupt

45.8.1.4 Serial Clock Ratio Considerations

The transmitter and the receiver can be programmed to operate with the clock signals provided on either the TK or RK pins. This allows the SSC to support many Slave mode data transfers. In this case, the maximum clock speed allowed on the RK pin is:

- Peripheral clock divided by 2 if Receive Frame Synchronization is input
- Peripheral clock divided by 3 if Receive Frame Synchronization is output

In addition, the maximum clock speed allowed on the TK pin is:

- Peripheral clock divided by 6 if Transmit Frame Synchronization is input
- Peripheral clock divided by 2 if Transmit Frame Synchronization is output

45.8.2 Transmit Operations

A transmit frame is triggered by a start event and can be followed by synchronization data before data transmission.

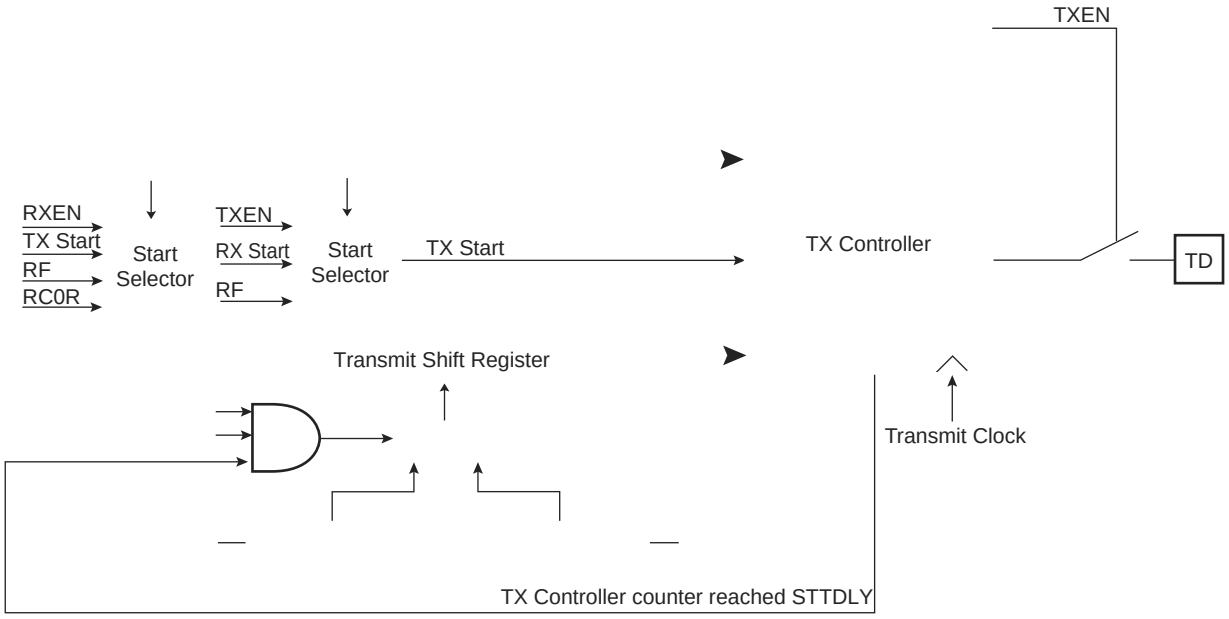
The start event is configured by setting the SSC_TCMR. Refer to Section 45.8.4 “Start”.

The frame synchronization is configured setting the Transmit Frame Mode Register (SSC_TFMR). Refer to Section 45.8.5 “Frame Synchronization”.

To transmit data, the transmitter uses a shift register clocked by the transmit clock signal and the start mode selected in the SSC_TCMR. Data is written by the application to the SSC_THR then transferred to the shift register according to the data format selected.

When both the SSC_THR and the transmit shift register are empty, the status flag TXEMPTY is set in the SSC_SR. When the Transmit Holding register is transferred in the transmit shift register, the status flag TXRDY is set in the SSC_SR and additional data can be loaded in the holding register.

Figure 45-11: Transmit Block Diagram



45.8.3 Receive Operations

A receive frame is triggered by a start event and can be followed by synchronization data before data transmission.

The start event is configured setting the Receive Clock Mode Register (SSC_RCMR). Refer to Section 45.8.4 “Start”.

The frame synchronization is configured by setting the Receive Frame Mode Register (SSC_RFMR). Refer to Section 45.8.5 “Frame Synchronization”.

The receiver uses a shift register clocked by the receive clock signal and the start mode selected in the SSC_RCMR. The data is transferred from the shift register depending on the data format selected.

45.8.7 Data Format

The data framing format of both the transmitter and the receiver are programmable through the Transmitter Frame Mode Register (SSC_TFMR) and the Receive Frame Mode Register (SSC_RFMR). In either case, the user can independently select the following parameters:

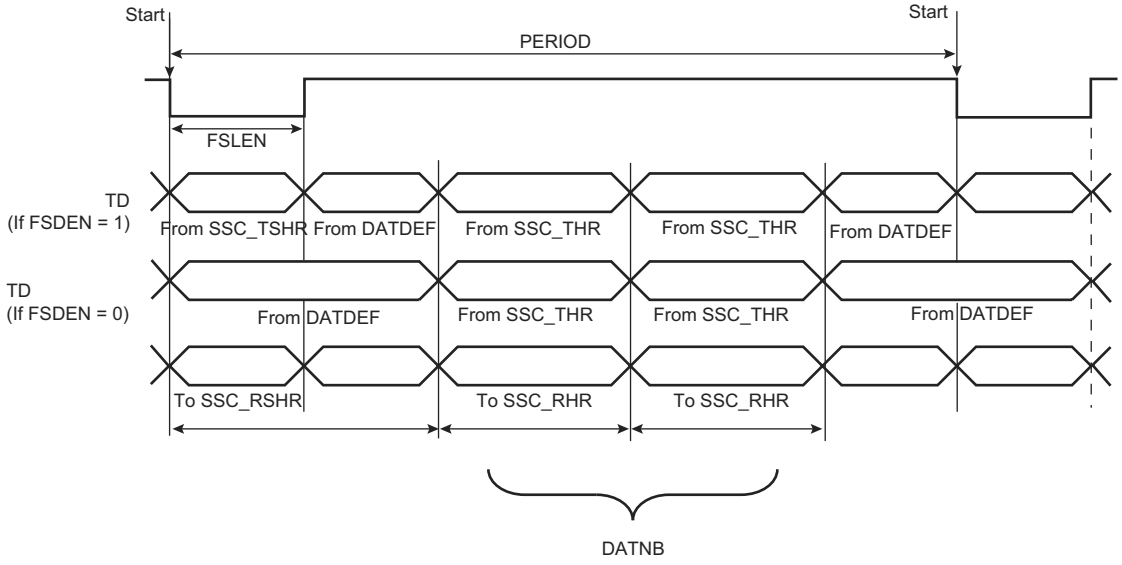
- Event that starts the data transfer (START)
- Delay in number of bit periods between the start event and the first data bit (STTDLY)
- Length of the data (DATLEN)
- Number of data to be transferred for each start event (DATNB)
- Length of synchronization transferred for each start event (FSLEN)
- Bit sense: most or least significant bit first (MSBF)

Additionally, the transmitter can be used to transfer synchronization and select the level driven on the TD pin while not in data transfer operation. This is done respectively by the Frame Sync Data Enable (FSDEN) and by the Data Default Value (DATDEF) bits in SSC_TFMR.

Table 45-4: Data Frame Registers

Transmitter	Receiver	Field	Length	Comment
SSC_TFMR	SSC_RFMR	DATLEN	Up to 32	Size of word
SSC_TFMR	SSC_RFMR	DATNB	Up to 16	Number of words transmitted in frame
SSC_TFMR	SSC_RFMR	MSBF	–	Most significant bit first
SSC_TFMR	SSC_RFMR	FSLEN	Up to 256	Size of Synchro data register
SSC_TFMR	–	DATDEF	0 or 1	Data default value ended
SSC_TFMR	–	FSDEN	–	Enable send SSC_TSHR
SSC_TCMR	SSC_RCMR	PERIOD	Up to 512	Frame size
SSC_TCMR	SSC_RCMR	STTDLY	Up to 255	Size of transmit start delay

Figure 45-16: Transmit and Receive Frame Format in Edge/Pulse Start Modes



Note: 1. Example of input on falling edge of TF/RF.

In the example illustrated in Figure 45-17, the SSC_THR is loaded twice. The FSDEN value has no effect on the transmission. SyncData cannot be output in Continuous mode.

46.7.15 TWIHS Receive Holding Register

Name: TWIHS_RHR

Address: 0xF8028030 (0), 0xFC028030 (1)

Access: Read-only

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

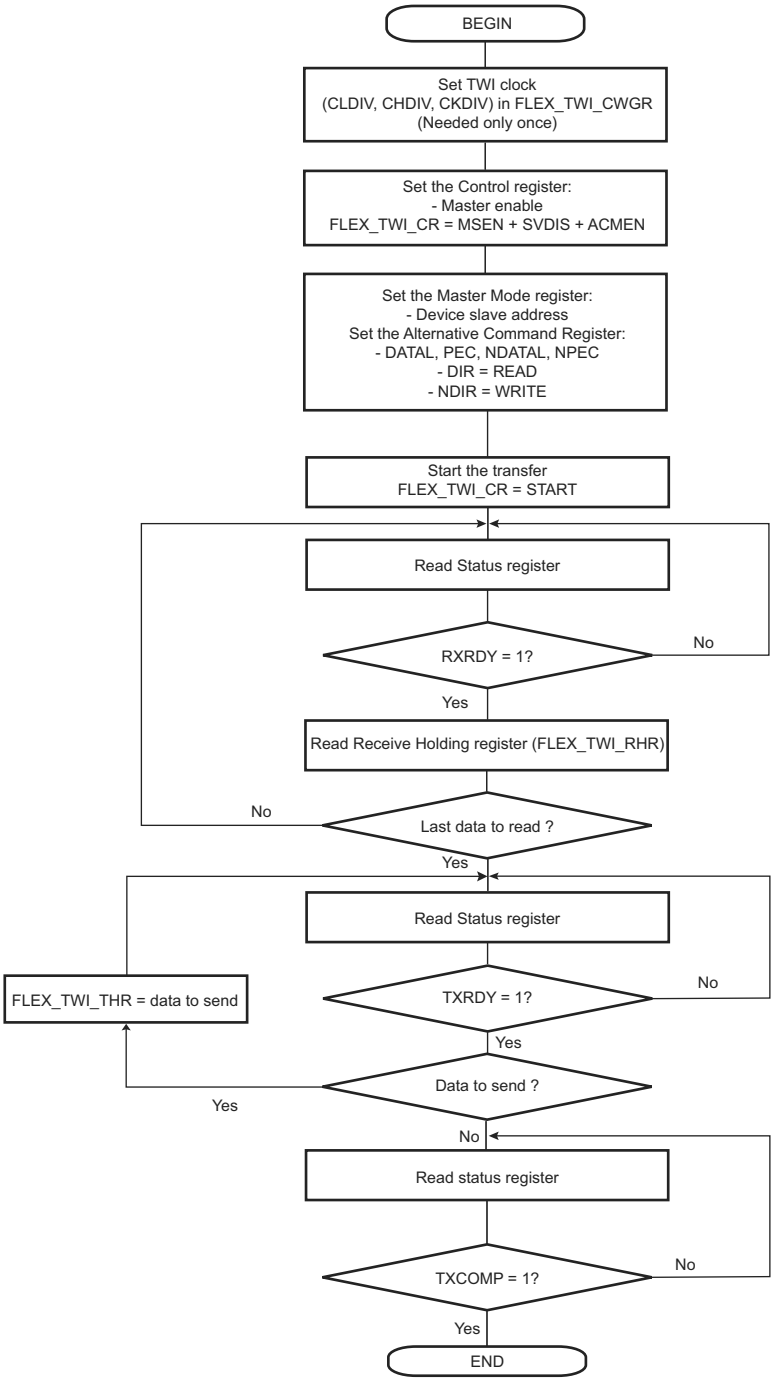
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–

15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–

7	6	5	4	3	2	1	0
RXDATA							

RXDATA: Master or Slave Receive Holding Data

Figure 47-109: TWI Read Operation with Multiple Data Bytes + Write with Alternative Command Mode with PEC



47.10.30 USART Manchester Configuration Register

Name: FLEX_US_MAN

Address: 0xF8034250 (0), 0xF8038250 (1), 0xFC010250 (2), 0xFC014250 (3), 0xFC018250 (4)

Access: Read/Write

31	30	29	28	27	26	25	24
RXIDLEV	DRIFT	ONE	RX_MPOL	–	–	RX_PP	

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

15	14	13	12	11	10	9	8
–	–	–	TX_MPOL	–	–	TX_PP	

7	6	5	4	3	2	1	0
–	–	–	–	TX_PL			

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

TX_PL: Transmitter Preamble Length

0: The transmitter preamble pattern generation is disabled

1–15: The preamble length is TX_PL × Bit Period

TX_PP: Transmitter Preamble Pattern

The following values assume that TX_MPOL field is not set:

Value	Name	Description
0	ALL_ONE	The preamble is composed of ‘1’s
1	ALL_ZERO	The preamble is composed of ‘0’s
2	ZERO_ONE	The preamble is composed of ‘01’s
3	ONE_ZERO	The preamble is composed of ‘10’s

TX_MPOL: Transmitter Manchester Polarity

0: Logic zero is coded as a zero-to-one transition, Logic one is coded as a one-to-zero transition.

1: Logic zero is coded as a one-to-zero transition, Logic one is coded as a zero-to-one transition.

RX_PL: Receiver Preamble Length

0: The receiver preamble pattern detection is disabled

1–15: The detected preamble length is RX_PL × Bit Period

RX_PP: Receiver Preamble Pattern detected

The following values assume that RX_MPOL field is not set:

Value	Name	Description
0	ALL_ONE	The preamble is composed of ‘1’s
1	ALL_ZERO	The preamble is composed of ‘0’s
2	ZERO_ONE	The preamble is composed of ‘01’s
3	ONE_ZERO	The preamble is composed of ‘10’s

RX_MPOL: Receiver Manchester Polarity

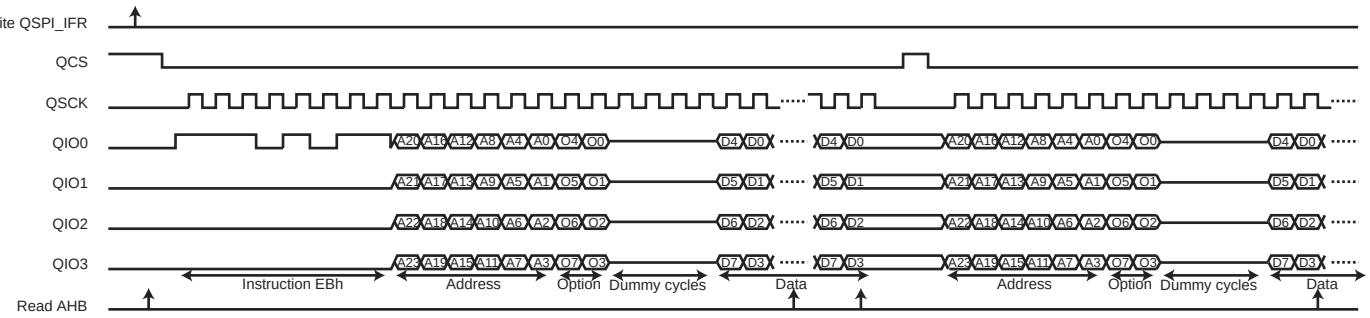
Example 7:

Instruction in Single-bit SPI, with address and option in Quad SPI, with data read in Quad SPI, with four dummy cycles, with fetch and continuous read.

Command: FAST READ QUAD I/O (EBh) - 8-BIT OPTION (0x30h)

- Write 0x0030_00EB in QSPI_ICR.
- Write 0x0004_33F4 in QSPI_IFR.
- Read QSPI_IFR (dummy read) to synchronize system bus accesses.
- Read data in the QSPI system bus memory space (0x9000_00000-0x9800_00000/0XD000_0000--0XD800_0000). Fetch is enabled, the address of the system bus read accesses is always used.
- Write a ‘1’ to QSPI_CR.LASTXFR.
- Wait for QSPI_SR.INSTRE to rise.

Figure 50-17: Instruction Transmission Waveform 7



51.13.36 SDMMC Maximum Current Capabilities Register

Name: SDMMC_MCCAR

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
MAXCUR18V							
15	14	13	12	11	10	9	8
MAXCUR30V							
7	6	5	4	3	2	1	0
MAXCUR33V							

MAXCUR33V: Maximum Current for 3.3V

This field indicates the maximum current capability for 3.3V voltage. This value is meaningful only if V33VSUP is set to 1 in SDMMC_CA0R. Reading MAXCUR33V at 0 means that the user must get information via another method.

$Imax_{mA} = 4 \times MAXCURR33V$

MAXCUR30V: Maximum Current for 3.0V

This field indicates the maximum current capability for 3.0V voltage. This value is meaningful only if V30VSUP is set to 1 in SDMMC_CA0R. Reading MAXCUR30V at 0 means that the user must get information via another method.

$Imax_{mA} = 4 \times MAXCURR30V$

MAXCUR18V: Maximum Current for 1.8V

This field indicates the maximum current capability for 1.8V voltage. This value is meaningful only if V18VSUP is set to 1 in SDMMC_CA0R. Reading MAXCUR18V at 0 means that the user must get information via another method.

$Imax_{mA} = 4 \times MAXCURR18V$

BCPC: RC Compare Effect on TIOBx

Value	Name	Description
0	NONE	None
1	SET	Set
2	CLEAR	Clear
3	TOGGLE	Toggle

BEEVT: External Event Effect on TIOBx

Value	Name	Description
0	NONE	None
1	SET	Set
2	CLEAR	Clear
3	TOGGLE	Toggle

BSWTRG: Software Trigger Effect on TIOBx

Value	Name	Description
0	NONE	None
1	SET	Set
2	CLEAR	Clear
3	TOGGLE	Toggle

Table 64-6: Register Mapping (Continued)

Offset	Register	Name	Access	Reset
0x0088	Normal Interrupt Disable Protection Register	SECUMOD_NIDPR	Write-only	–
0x008C	Normal Interrupt Mask Protection Register	SECUMOD_NIMPR	Read-only	0x0(4)
0x0090	Wakeup Protection Register	SECUMOD_WKPR	Read/Write	0x0

Note 1: When fuse DEFDBG is not programmed.

2: When fuse DEFDBG is programmed.

3: PIO backup protections are off after backup reset whatever the reset value of this register. See SECUMOD_PIOBUx register descriptions to enable these protections.

4: After Peripheral Reset (other reset values are defined after Backup Reset).

64.6.2 SECUMOD System Status Register

Name: SECUMOD_SYSR

Address: 0xFC040004

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
SCRAMB	AUTOBKP	–	–	SWKUP	BACKUP	ERASE_ON	ERASE_DONE

ERASE_DONE: Erasable Memories State (RW)

0: Secure memories content has not been erased since the last clear.

1: Secure memories content has been erased since the last clear. The user must write 1 into this bit to clear this flag. Note that not clearing this flag does not prevent the next erase processes. This flag also activates the SECURAM interrupt line as long as it is not cleared.

ERASE_ON: Erase Process Ongoing (RO)

0: Erase automaton is not running.

1: Erase automaton is currently running, memories are not accessible.

When ERASE_ON returns to 0, ERASE_DONE is set after half a period of ICLK.

ERASE_ON	ERASE_DONE	Status	Action
0	0	No Erase ongoing or since the last Erase.	Nothing.
1	0	An Erase process is running.	Wait until the ERASE_ON flag is reset. ERASE_DONE will rise, see line below.
0	1	An Erase occurred and is finished.	Clear the ERASE_DONE flag.
1	1	An Erase process is running. The ERASE_DONE flag refers to a previous Erase process, but was not cleared.	Wait until the ERASE_ON flag is reset, then clear the ERASE_DONE flag.

BACKUP: Backup Mode (RO)

0: Normal mode active.

1: Backup mode active.

SWKUP: SWKUP State (RO)

0: No SWKUP signal sent since the last clear.

1: SWKUP signal has been sent since the last clear.

66.14.3.2 SMC IOSET2 Write Timings

Table 66-50: SMC IOSET2 Write Signals - NWE Controlled (WRITE_MODE = 1)

Symbol	Parameter	Min		Unit
	Power supply	1.8V	3.3V	
HOLD or NO HOLD SETTINGS (nwe hold ≠ 0, nwe hold = 0)				
SMC_{15}	Data Out Valid before NWE High	nwe pulse × t_{CPMCK}	nwe pulse × t_{CPMCK}	ns
SMC_{16}	NWE Pulse Width	nwe pulse × t_{CPMCK}	nwe pulse × t_{CPMCK}	ns
SMC_{17}	NBS0/A0 NBS1, NBS2/A1, NBS3, A2–A25 valid before NWE low	nwe setup × t_{CPMCK}	nwe pulse × t_{CPMCK}	ns
SMC_{18}	NCS low before NWE high	(nwe setup - ncs rd setup + nwe pulse) × t_{CPMCK}	(nwe setup - ncs rd setup + nwe pulse) × t_{CPMCK}	ns
HOLD SETTINGS (nwe hold ≠ 0)				
SMC_{19}	NWE High to Data OUT, NBS0/A0 NBS1, NBS2/A1, NBS3, A2–A25 change	nwe hold × t_{CPMCK}	nwe hold × t_{CPMCK}	ns
SMC_{20}	NWE High to NCS Inactive(1)	(nwe hold - ncs wr hold) × t_{CPMCK}	(nwe hold - ncs wr hold) × t_{CPMCK}	ns
NO HOLD SETTINGS (nwe hold = 0)				
SMC_{21}	NWE High to Data OUT, NBS0/A0 NBS1, NBS2/A1, NBS3, A2–A25, NCS change((1)	1.2	0.6	ns

Note 1: hold length = total cycle duration - setup duration - pulse duration. “hold length” is for “ncs wr hold length” or “NWE hold length”.

Table 66-51: SMC IOSET2 Write NCS Controlled (WRITE_MODE = 0)

Symbol	Parameter	Min		Unit
	Power supply	1.8V	3.3V	
SMC_{22}	Data Out Valid before NCS High	ncs wr pulse × t_{CPMCK}	ncs wr pulse × t_{CPMCK}	ns
SMC_{23}	NCS Pulse Width	SMC_{14}	SMC_{14}	ns
SMC_{24}	NBS0/A0 NBS1, NBS2/A1, NBS3, A2–A25 valid before NCS low	ncs wr setup × t_{CPMCK}	ncs wr setup × t_{CPMCK}	ns
SMC_{25}	NWE low before NCS high	(ncs wr setup - nwe setup + ncs pulse) × t_{CPMCK}	(ncs wr setup - nwe setup + ncs pulse) × t_{CPMCK}	ns
SMC_{26}	NCS High to Data Out, NBS0/A0, NBS1, NBS2/A1, NBS3, A2–A25, change	ncs wr hold × t_{CPMCK}	ncs wr hold × t_{CPMCK}	ns
SMC_{27}	NCS High to NWE Inactive	(ncs wr hold - nwe hold) × t_{CPMCK}	(ncs wr hold - nwe hold) × t_{CPMCK}	ns

crosstalk. Based on EMI testing experience, the minimum suggested spacing to clock signals is 50 mil.
- Use a minimum of 20 mil spacing between the differential signal pairs and other signal traces for optimal signal quality. This helps to prevent crosstalk.
- Route all traces over continuous planes (VCC or GND) with no interruptions.
- Avoid crossing over anti-etch if at all possible. Crossing over anti-etch (split planes) increases inductance and radiation levels by forcing a greater loop area.

68.12.3 DDR Layout and Design Considerations

Refer to the document “SAMA5D2 Layout Recommendations”, document no. 44041.

68.12.4 e.MMC routing

Refer to the Micron Technical Note TN-FC-35: e•MMC PCB Design Guide. This document is intended as guide for PCB designers using Micron e•MMC devices and discusses the primary issues affecting design and layout.

68.12.5 USB Trace Routing Guidelines

- Maintain parallelism between USB differential signals with the trace spacing needed to achieve 90-ohm differential impedance. Deviations will normally occur due to package breakout and routing to connector pins. Just ensure the amount and lengths of the deviations are kept to the minimum possible.
- Use an impedance calculator to determine the trace width and spacing required for the specific board stack-up being used. Example: Layer Stacking, 7.5-mil traces with 7.5-mil spacing results in approximately 90-ohm differential trace impedance.
- Minimize the length of high-speed clock and periodic signal traces that run parallel to high-speed USB signal lines, to minimize crosstalk. Based on EMI testing experience, the minimum suggested spacing to clock signals is 50 mils.
- Based on simulation data, use 20-mil minimum spacing between high-speed USB signal pairs and other signal traces for optimal signal quality. This helps to prevent crosstalk.

Table 68-14: USB Trace Routing Guidelines

Parameter	Trace Routing
Signal length allowance for the SAMA5D2	14.0 inches Valid for a damping value of the PCB trace of 0.11 dB/ inch @ 0.4 GHz (common value for FR-4 based material)
Differential impedance	90 ohms +/-15%
Single-ended impedance	45 ohms +/-10%
Trace width (W)	5 mils (microstrip routing)
Spacing between differential pairs (intra-pair)	6 mils (microstrip routing)
Spacing between pairs (inter-pair)	Min. 20 mils
Spacing between differential pairs and high-speed periodic signals	Min. 50 mils
Spacing between differential pairs and low-speed non-periodic signals	Min. 20 mils
Length matching between differential pairs (intra-pair)	150 mils
Reference plane	GND referenced preferred
Spacing from edge of plane	Min. 40 mils
Vias usage	Try to minimize the number of vias

68.12.6 QSPI Pull-up Resistors

The ROM code **removes** the internal pull-up resistors when it configures PIO controller to mux the QSPI controller I/O lines. Therefore the probing step may fail if the Quad I/O mode of the memory has not been enabled yet and if this memory does not embed internal pull-up resistor on #HOLD or #RESET pin.