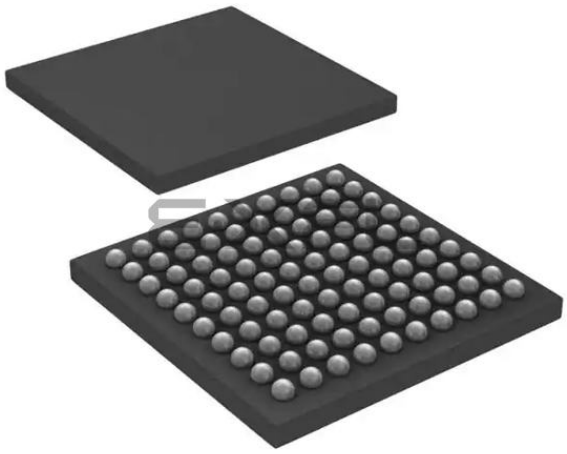


Welcome to E-XFL.COM

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	ARM® Cortex®-M4
Core Size	32-Bit Single-Core
Speed	120MHz
Connectivity	EBI/EMI, I²C, IrDA, Memory Card, SPI, SSC, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	79
Program Memory Size	1MB (1M x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	160K x 8
Voltage - Supply (Vcc/Vdd)	1.62V ~ 3.6V
Data Converters	A/D 16x12b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	100-VFBGA
Supplier Device Package	100-VFBGA (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsam4sa16cb-cfnr

continuous). If GPNVM2 is DISABLE, the Flash 0 is mapped at address 0x0040_0000 (Flash 0 and Flash 1 are continuous).

Table 8-3. General-purpose Non-volatile Memory Bits

Device Name	GPNVM0	GPNVM1	GPNVM2
SAM4SD32	Security Bit	Boot Mode Selection	Flash Selection (Flash 0 or Flash 1)
SAM4SD16			
SAM4SA16			
SAM4S16			Not available
SAM4S8			
SAM4S4			
SAM4S2			

8.1.4 Boot Strategies

The system always boots at address 0x0. To ensure maximum boot possibilities, the memory layout can be changed using GPNVM bits.

A general-purpose NVM (GPNVM) bit is used to boot either on the ROM (default) or from the Flash.

The GPNVM bit can be cleared or set respectively through the commands “Clear GPNVM Bit” and “Set GPNVM Bit” of the EEFC User Interface.

Setting GPNVM1 selects the boot from the Flash. Clearing it selects the boot from the ROM. Asserting ERASE clears the GPNVM1 and thus selects the boot from the ROM by default.

Setting the GPNVM2 selects Flash 1, clearing it selects the boot from Flash 0. Asserting ERASE clears GPNVM2 and thus selects the boot from Flash 0 by default. GPNVM2 is available only on SAM4SD32/SD16/SA16.

8.2 External Memories

The SAM4S features one External Bus Interface to provide an interface to a wide range of external memories and to any parallel peripheral.

12.11.2.6 MPU Region Base Address Register Alias 1

Name: MPU_RBAR_A1

Access: Read/Write

31	30	29	28	27	26	25	24
ADDR							

23	22	21	20	19	18	17	16
ADDR							

15	14	13	12	11	10	9	8
ADDR							

7	6	5	4	3	2	1	0
ADDR			VALID	REGION			

The MPU_RBAR defines the base address of the MPU region selected by the MPU_RNR, and can update the value of the MPU_RNR.

Write MPU_RBAR with the VALID bit set to 1 to change the current region number and update the MPU_RNR.

- **ADDR: Region Base Address**

Software must ensure that the value written to the ADDR field aligns with the size of the selected region.

The value of N depends on the region size. The ADDR field is bits[31:N] of the MPU_RBAR. The region size, as specified by the SIZE field in the MPU_RASR, defines the value of N:

N = Log2(Region size in bytes),

If the region size is configured to 4 GB, in the MPU_RASR, there is no valid ADDR field. In this case, the region occupies the complete memory map, and the base address is 0x00000000.

The base address is aligned to the size of the region. For example, a 64 KB region must be aligned on a multiple of 64 KB, for example, at 0x00010000 or 0x00020000.

- **VALID: MPU Region Number Valid**

Write:

0: MPU_RNR not changed, and the processor updates the base address for the region specified in the MPU_RNR, and ignores the value of the REGION field.

1: The processor updates the value of the MPU_RNR to the value of the REGION field, and updates the base address for the region specified in the REGION field.

Always reads as zero.

- **REGION: MPU Region**

For the behavior on writes, see the description of the VALID field.

On reads, returns the current region number, as specified by the MPU_RNR.

14. Reset Controller (RSTC)

14.1 Description

The Reset Controller (RSTC), based on power-on reset cells, handles all the resets of the system without any external components. It reports which reset occurred last.

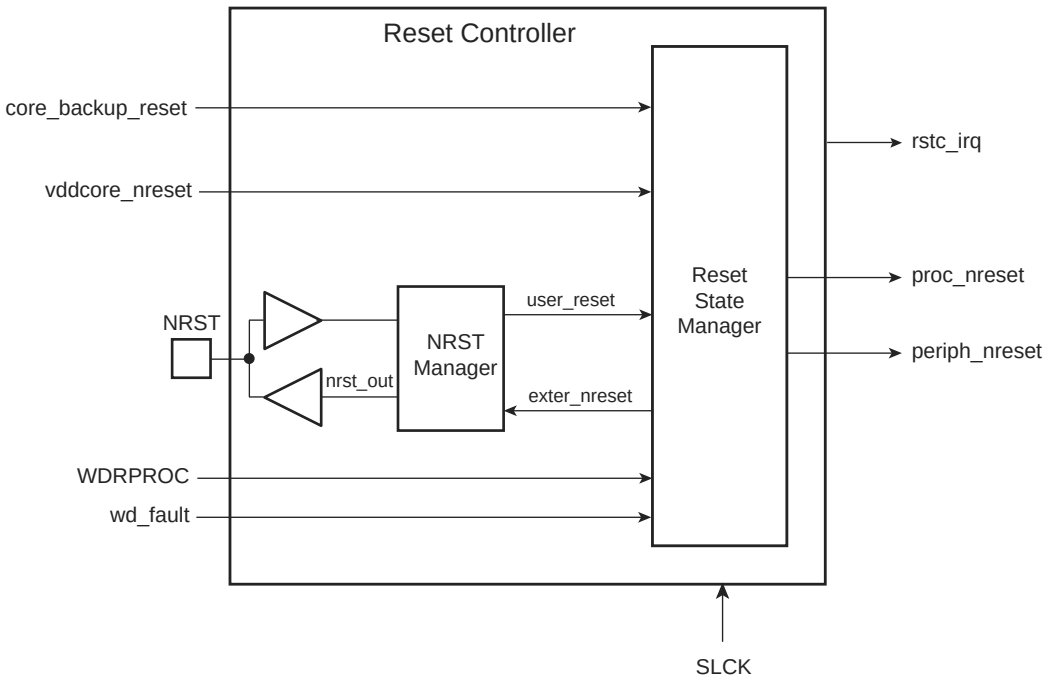
The Reset Controller also drives independently or simultaneously the external reset and the peripheral and processor resets.

14.2 Embedded Characteristics

- Management of All System Resets, Including
  - External Devices through the NRST Pin
  - Processor Reset
  - Processor Peripheral Set Reset
- Based on Embedded Power-on Cell
- Reset Source Status
  - Status of the Last Reset
  - Either Software Reset, User Reset, Watchdog Reset
- External Reset Signal Shaping

14.3 Block Diagram

Figure 14-1. Reset Controller Block Diagram



14.4 Functional Description

14.4.1 Reset Controller Overview

The Reset Controller is made up of an NRST manager and a reset state manager. It runs at slow clock and generates the following reset signals:

- proc_nreset: processor reset line (also resets the Watchdog Timer)
- periph_nreset: affects the whole set of embedded peripherals
- nrst_out: drives the NRST pin

These reset signals are asserted by the Reset Controller, either on events generated by peripherals, events on NRST pin, or on software action. The reset state manager controls the generation of reset signals and provides a signal to the NRST manager when an assertion of the NRST pin is required.

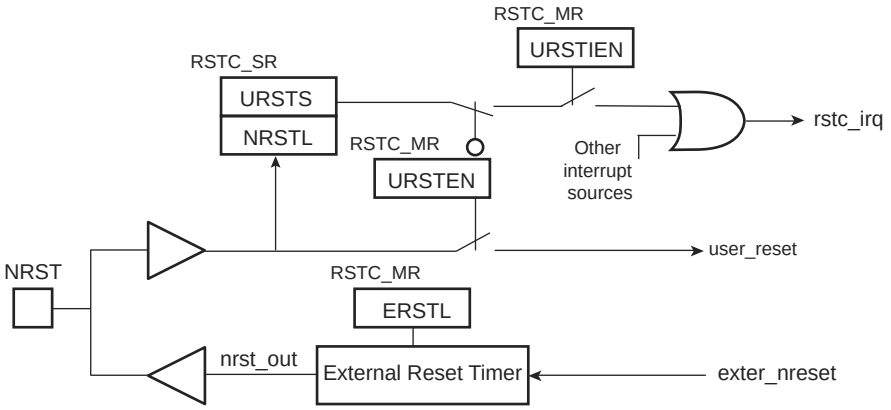
The NRST manager shapes the NRST assertion during a programmable time, thus controlling external device resets.

The Reset Controller Mode Register (RSTC_MR), used to configure the Reset Controller, is powered with VDDIO, so that its configuration is saved as long as VDDIO is on.

14.4.2 NRST Manager

The NRST manager samples the NRST input pin and drives this pin low when required by the reset state manager. Figure 14-2 shows the block diagram of the NRST manager.

Figure 14-2. NRST Manager



14.4.2.1 NRST Signal or Interrupt

The NRST manager samples the NRST pin at slow clock speed. When the line is detected low, a User Reset is reported to the reset state manager.

However, the NRST manager can be programmed to not trigger a reset when an assertion of NRST occurs. Writing a 0 to the URSTEN bit in the RSTC_MR disables the User Reset trigger.

The level of the pin NRST can be read at any time in the bit NRSTL (NRST level) in the Reset Controller Status Register (RSTC_SR). As soon as the NRST pin is asserted, bit URSTS in the RSTC_SR is set. This bit is cleared only when the RSTC_SR is read.

The Reset Controller can also be programmed to generate an interrupt instead of generating a reset. To do so, set the URSTIEN bit in the RSTC_MR.

20.4.2 Read Operations

An optimized controller manages embedded Flash reads, thus increasing performance when the processor is running in Thumb-2 mode by means of the 128- or 64-bit-wide memory interface.

The Flash memory is accessible through 8-, 16- and 32-bit reads.

As the Flash block size is smaller than the address space reserved for the internal memory area, the embedded Flash wraps around the address space and appears to be repeated within it.

The read operations can be performed with or without wait states. Wait states must be programmed in the field FWS in the Flash Mode register (EEFC_FMR). Defining FWS as 0 enables the single-cycle access of the embedded Flash. For mre details, refer to the section “Electrical Characteristics” of this datasheet.

20.4.2.1 128- or 64-bit Access Mode

By default, the read accesses of the Flash are performed through a 128-bit wide memory interface. It improves system performance especially when two or three wait states are needed.

For systems requiring only 1 wait state, or to focus on current consumption rather than performance, the user can select a 64-bit wide memory access via the bit EEFC_FMR.FAM.

For more details, refer to the section “Electrical Characteristics” of this datasheet.

20.4.2.2 Code Read Optimization

Code read optimization is enabled if the bit EEFC_FMR.SCOD is cleared.

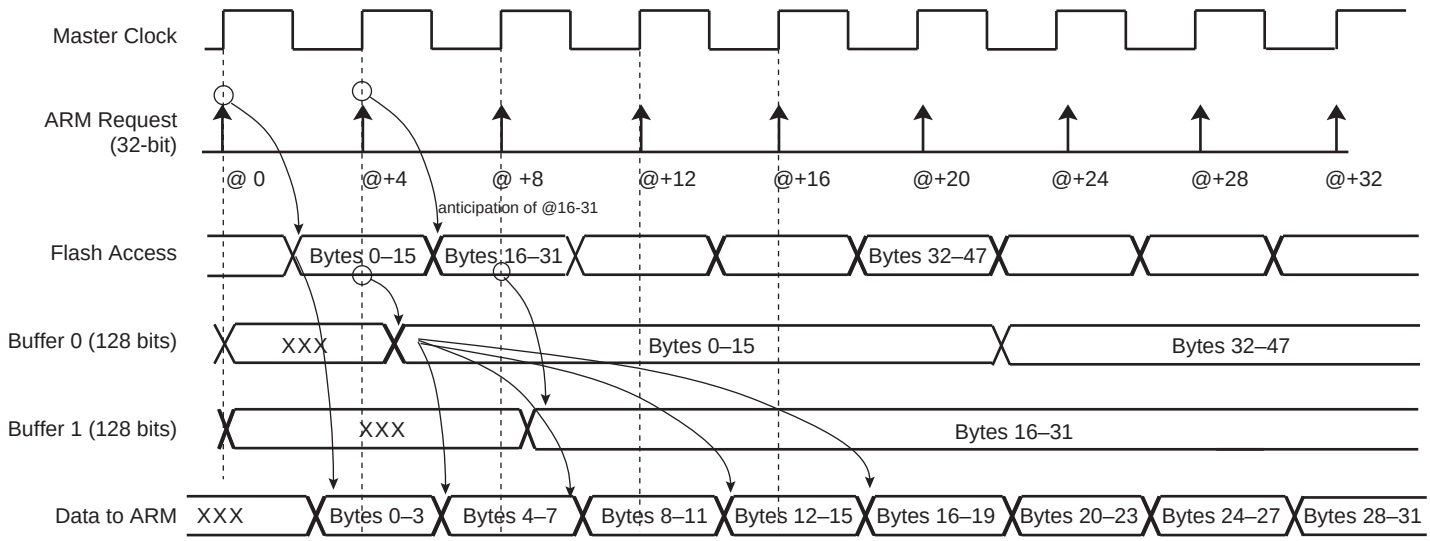
A system of 2 x 128-bit or 2 x 64-bit buffers is added in order to optimize sequential code fetch.

Note: Immediate consecutive code read accesses are not mandatory to benefit from this optimization.

The sequential code read optimization is enabled by default. If the bit EEFC_FMR.SCOD is set to 1, these buffers are disabled and the sequential code read is no longer optimized.

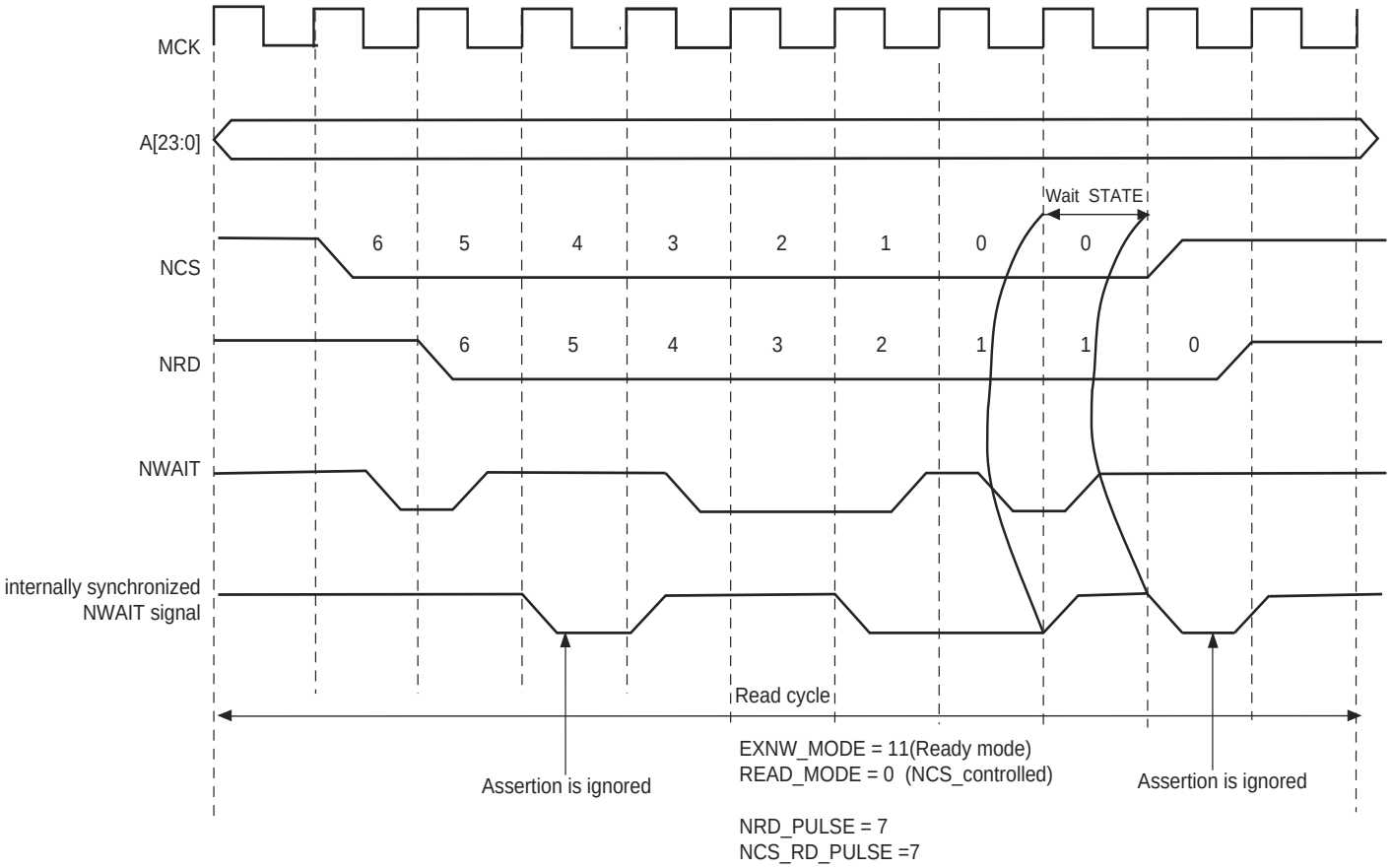
Another system of 2 x 128-bit or 2 x 64-bit buffers is added in order to optimize loop code fetch. Refer to Section 20.4.2.3 ”Code Loop Optimization” for more details.

Figure 20-3. Code Read Optimization for FWS = 0



Note: When FWS is equal to 0, all the accesses are performed in a single-cycle access.

Figure 26-26. NWAIT Assertion in Read Access: Ready Mode (EXNW_MODE = 11)



28. Clock Generator

28.1 Description

The Clock Generator user interface is embedded within the Power Management Controller and is described in Section 29.17 ”Power Management Controller (PMC) User Interface”. However, the Clock Generator registers are named CKGR_.

28.2 Embedded Characteristics

The Clock Generator is made up of:

- A low-power 32768 Hz slow clock oscillator with Bypass mode
- A low-power RC oscillator
- A 3 to 20 MHz crystal or ceramic resonator-based oscillator, which can be bypassed.
- A factory-programmed fast RC oscillator. Three output frequencies can be selected: 4/8/12 MHz. By default 4 MHz is selected.
- Two 80 to 240 MHz programmable PLL (input from 3 to 32 MHz), capable of providing the clock MCK to the processor and to the peripherals.

It provides the following clocks:

- SLCK, the slow clock, which is the only permanent clock within the system.
- MAINCK is the output of the main clock oscillator selection: either the crystal or ceramic resonator-based oscillator or 4/8/12 MHz fast RC oscillator.
- PLLACK is the output of the divider and 80 to 240 MHz programmable PLL (PLLA).
- PLLBCK is the output of the divider and 80 to 240 MHz programmable PLL (PLLB).

29.17.7 PMC Clock Generator Main Oscillator Register

Name: CKGR_MOR

Address: 0x400E0420

Access: Read/Write

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

23	22	21	20	19	18	17	16
KEY							

15	14	13	12	11	10	9	8
MOSCXTST							

7	6	5	4	3	2	1	0
–	MOSCRCF			MOSCRCEN	WAITMODE	MOSCXTBY	MOSCXTEN

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

- **MOSCXTEN: Main Crystal Oscillator Enable**

A crystal must be connected between XIN and XOUT.

0: The main crystal oscillator is disabled.

1: The main crystal oscillator is enabled. MOSCXTBY must be cleared.

When MOSCXTEN is set, the MOSCXTS flag is set once the main crystal oscillator start-up time is achieved.

- **MOSCXTBY: Main Crystal Oscillator Bypass**

0: No effect.

1: The main crystal oscillator is bypassed. MOSCXTEN must be cleared. An external clock must be connected on XIN.

When MOSCXTBY is set, the MOSCXTS flag in PMC_SR is automatically set.

Clearing MOSCXTEN and MOSCXTBY bits resets the MOSCXTS flag.

Note: When the main crystal oscillator bypass is disabled (MOSCXTBY = 0), the MOSCXTS flag must be read at 0 in PMC_SR before enabling the main crystal oscillator (MOSCXTEN = 1).

- **WAITMODE: Wait Mode Command (Write-only)**

0: No effect.

1: Puts the device in Wait mode.

- **MOSCRCEN: Main On-Chip RC Oscillator Enable**

0: The main on-chip RC oscillator is disabled.

1: The main on-chip RC oscillator is enabled.

When MOSCRCEN is set, the MOSCRCS flag is set once the main on-chip RC oscillator start-up time is achieved.

Value	Name	Description
12	128K	128 Kbytes
13	256K	256 Kbytes
14	96K	96 Kbytes
15	512K	512 Kbytes

- **ARCH: Architecture Identifier**

Value	Name	Description
0x88	SAM4SxA	SAM4SxA (48-pin version)
0x89	SAM4SxB	SAM4SxB (64-pin version)
0x8A	SAM4SxC	SAM4SxC (100-pin version)

- **NVPTYP: Nonvolatile Program Memory Type**

Value	Name	Description
0	ROM	ROM
1	ROMLESS	ROMless or on-chip Flash
2	FLASH	Embedded Flash Memory
3	ROM_FLASH	ROM and Embedded Flash Memory ● NVPSIZ is ROM size ● NVPSIZ2 is Flash size
4	SRAM	SRAM emulating ROM

- **EXT: Extension Flag**

0: Chip ID has a single register definition without extension.

1: An extended Chip ID exists.

registers results in setting or clearing the corresponding bit in the Pull-down Status Register (PIO_PPDSR). Reading a one in PIO_PPDSR means the pull-up is disabled and reading a zero means the pull-down is enabled.

Enabling the pull-down resistor while the pull-up resistor is still enabled is not possible. In this case, the write of PIO_PPDER for the relevant I/O line is discarded. Likewise, enabling the pull-up resistor while the pull-down resistor is still enabled is not possible. In this case, the write of PIO_PUER for the relevant I/O line is discarded.

Control of the pull-up resistor is possible regardless of the configuration of the I/O line.

After reset, depending on the I/O, pull-up or pull-down can be set.

31.5.2 I/O Line or Peripheral Function Selection

When a pin is multiplexed with one or two peripheral functions, the selection is controlled with the Enable Register (PIO_PER) and the Disable Register (PIO_PDR). The Status Register (PIO_PSR) is the result of the set and clear registers and indicates whether the pin is controlled by the corresponding peripheral or by the PIO Controller. A value of zero indicates that the pin is controlled by the corresponding on-chip peripheral selected in the ABCD Select registers (PIO_ABCDSR1 and PIO_ABCDSR2). A value of one indicates the pin is controlled by the PIO Controller.

If a pin is used as a general-purpose I/O line (not multiplexed with an on-chip peripheral), PIO_PER and PIO_PDR have no effect and PIO_PSR returns a one for the corresponding bit.

After reset, the I/O lines are controlled by the PIO Controller, i.e., PIO_PSR resets at one. However, in some events, it is important that PIO lines are controlled by the peripheral (as in the case of memory chip select lines that must be driven inactive after reset, or for address lines that must be driven low for booting out of an external memory). Thus, the reset value of PIO_PSR is defined at the product level and depends on the multiplexing of the device.

31.5.3 Peripheral A or B or C or D Selection

The PIO Controller provides multiplexing of up to four peripheral functions on a single pin. The selection is performed by writing PIO_ABCDSR1 and PIO_ABCDSR2.

For each pin:

- The corresponding bit at level zero in PIO_ABCDSR1 and the corresponding bit at level zero in PIO_ABCDSR2 means peripheral A is selected.
- The corresponding bit at level one in PIO_ABCDSR1 and the corresponding bit at level zero in PIO_ABCDSR2 means peripheral B is selected.
- The corresponding bit at level zero in PIO_ABCDSR1 and the corresponding bit at level one in PIO_ABCDSR2 means peripheral C is selected.
- The corresponding bit at level one in PIO_ABCDSR1 and the corresponding bit at level one in PIO_ABCDSR2 means peripheral D is selected.

Note that multiplexing of peripheral lines A, B, C and D only affects the output line. The peripheral input lines are always connected to the pin input (see Figure 31-2).

Writing in PIO_ABCDSR1 and PIO_ABCDSR2 manages the multiplexing regardless of the configuration of the pin. However, assignment of a pin to a peripheral function requires a write in PIO_ABCDSR1 and PIO_ABCDSR2 in addition to a write in PIO_PDR.

After reset, PIO_ABCDSR1 and PIO_ABCDSR2 are zero, thus indicating that all the PIO lines are configured on peripheral A. However, peripheral A generally does not drive the pin as the PIO Controller resets in I/O line mode.

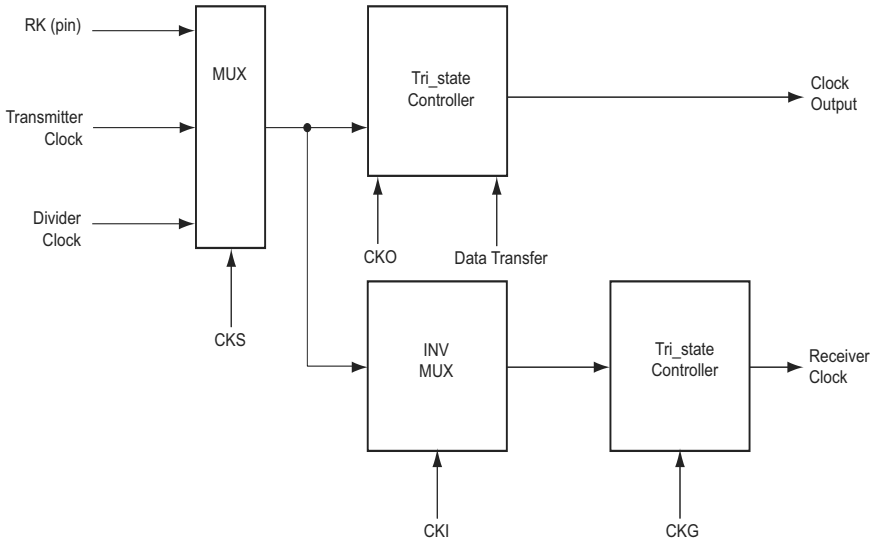
If the software selects a peripheral A, B, C or D which does not exist for a pin, no alternate functions are enabled for this pin and the selection is taken into account. The PIO Controller does not carry out checks to prevent selection of a peripheral which does not exist.

32.8.1.3 Receiver Clock Management

The receiver clock is generated from the transmitter clock or the divider clock or an external clock scanned on the RK I/O pad. The Receive Clock is selected by the CKS field in SSC_RCMR (Receive Clock Mode Register). Receive Clocks can be inverted independently by the CKI bits in SSC_RCMR.

The receiver can also drive the RK I/O pad continuously or be limited to the actual data transfer. The clock output is configured by the SSC_RCMR. The Receive Clock Inversion (CKI) bits have no effect on the clock outputs. Programming the SSC_RCMR to select RK pin (CKS field) and at the same time Continuous Receive Clock (CKO field) can lead to unpredictable results.

Figure 32-10. Receiver Clock Management



32.8.1.4 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 Receiver Frame Synchro is input
- Peripheral clock divided by 3 if Receiver Frame Synchro is output

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

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

32.8.2 Transmitter Operations

A transmitted 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. See Section 32.8.4 “Start” on page 652.

The frame synchronization is configured setting the Transmit Frame Mode Register (SSC_TFMR). See Section 32.8.5 “Frame Sync” on page 654.

To transmit data, the transmitter uses a shift register clocked by the transmitter 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.

Figure 32-13. Transmit Start Mode

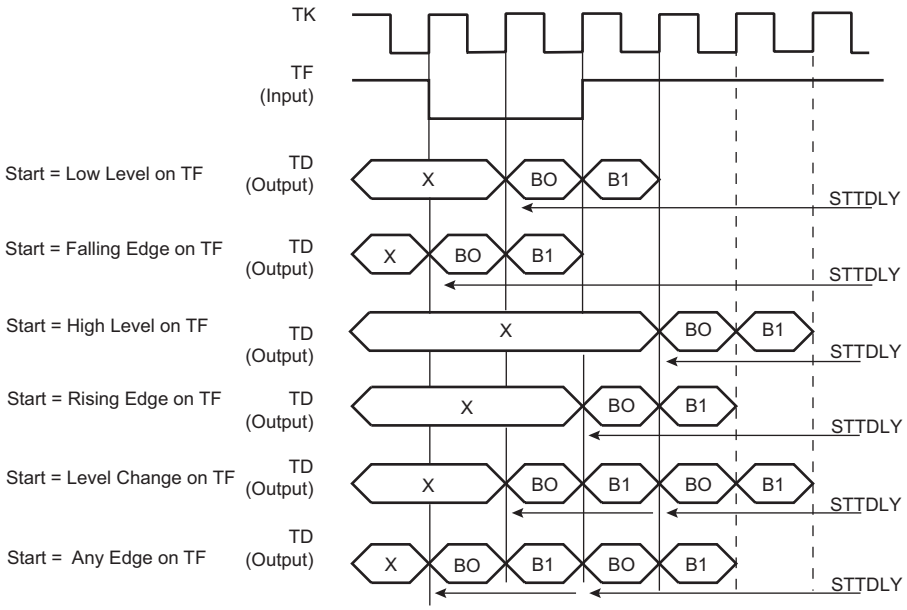
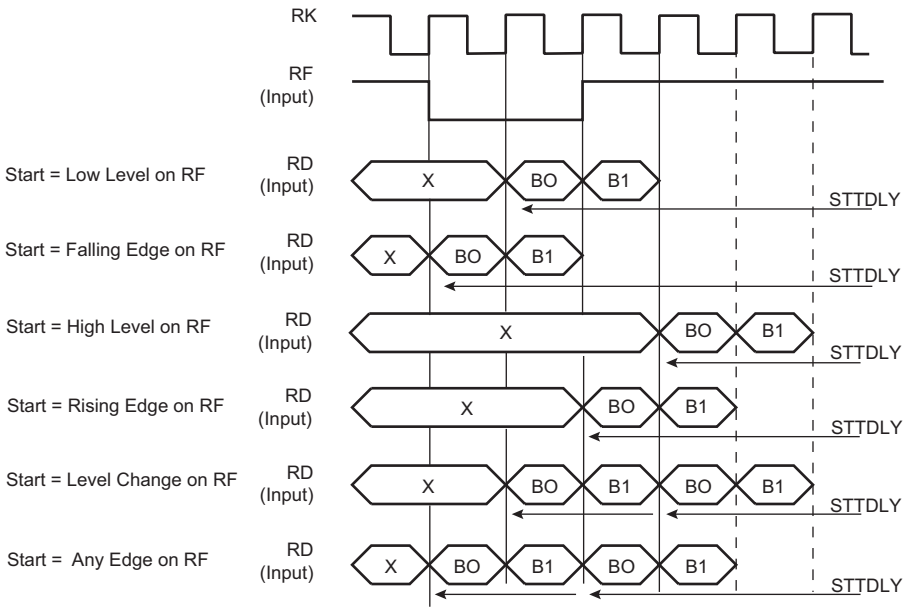


Figure 32-14. Receive Pulse/Edge Start Modes



- **LLB: Local Loopback Enable**

0: Local loopback path disabled.

1: Local loopback path enabled.

LLB controls the local loopback on the data shift register for testing in Master mode only (MISO is internally connected on MOSI).

- **PCS: Peripheral Chip Select**

This field is only used if fixed peripheral select is active (PS = 0).

If SPI_MR.PCSDEC = 0:

PCS = xxx0 NPCS[3:0] = 1110

PCS = xx01 NPCS[3:0] = 1101

PCS = x011 NPCS[3:0] = 1011

PCS = 0111 NPCS[3:0] = 0111

PCS = 1111 forbidden (no peripheral is selected)

(x = don’t care)

If SPI_MR.PCSDEC = 1:

NPCS[3:0] output signals = PCS.

- **DLYBCS: Delay Between Chip Selects**

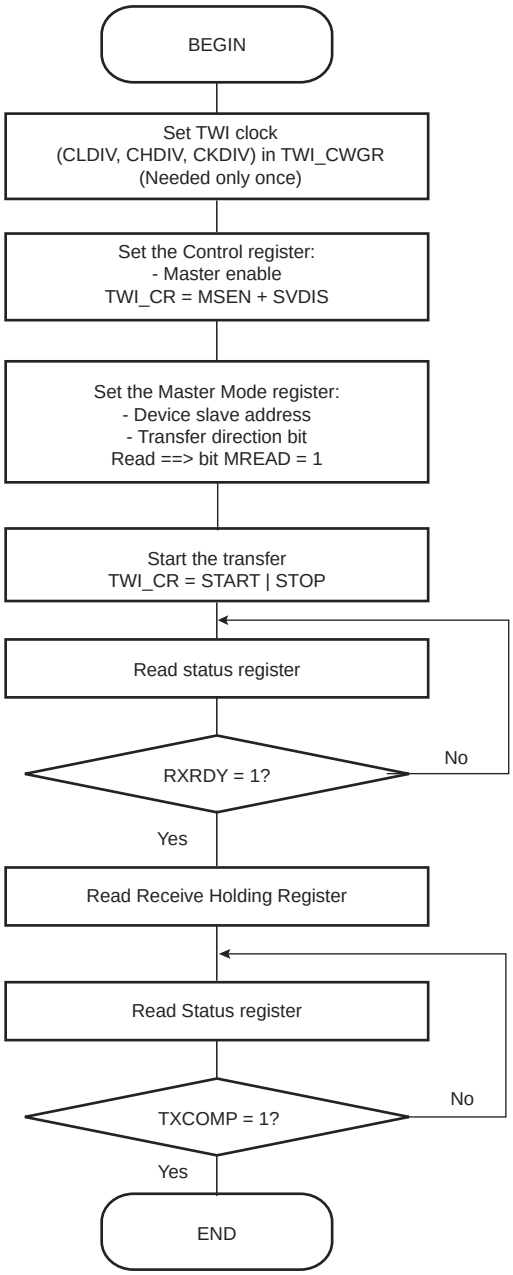
This field defines the delay between the inactivation and the activation of NPCS. The DLYBCS time guarantees non-overlapping chip selects and solves bus contentions in case of peripherals having long data float times.

If DLYBCS is lower than 6, six peripheral clock periods are inserted by default.

Otherwise, the following equation determines the delay:

$$\text{Delay Between Chip Selects} = \frac{\text{DLYBCS}}{f_{\text{peripheral clock}}}$$

Figure 34-17. TWI Read Operation with Single Data Byte without Internal Address



34.8.4 TWI Internal Address Register

Name: TWI_IADR

Address: 0x4001800C (0), 0x4001C00C (1)

Access: Read/Write

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

23	22	21	20	19	18	17	16
IADR							

15	14	13	12	11	10	9	8
IADR							

7	6	5	4	3	2	1	0
IADR							

- **IADR: Internal Address**

0, 1, 2 or 3 bytes depending on IADRSZ.

34.8.9 TWI Interrupt Mask Register

Name: TWI_IMR

Address: 0x4001802C (0), 0x4001C02C (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
TXBUFE	RXBUFF	ENDTX	ENDRX	EOSACC	SCL_WS	ARBLST	NACK

7	6	5	4	3	2	1	0
–	OVRE	GACC	SVACC	–	TXRDY	RXRDY	TXCOMP

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

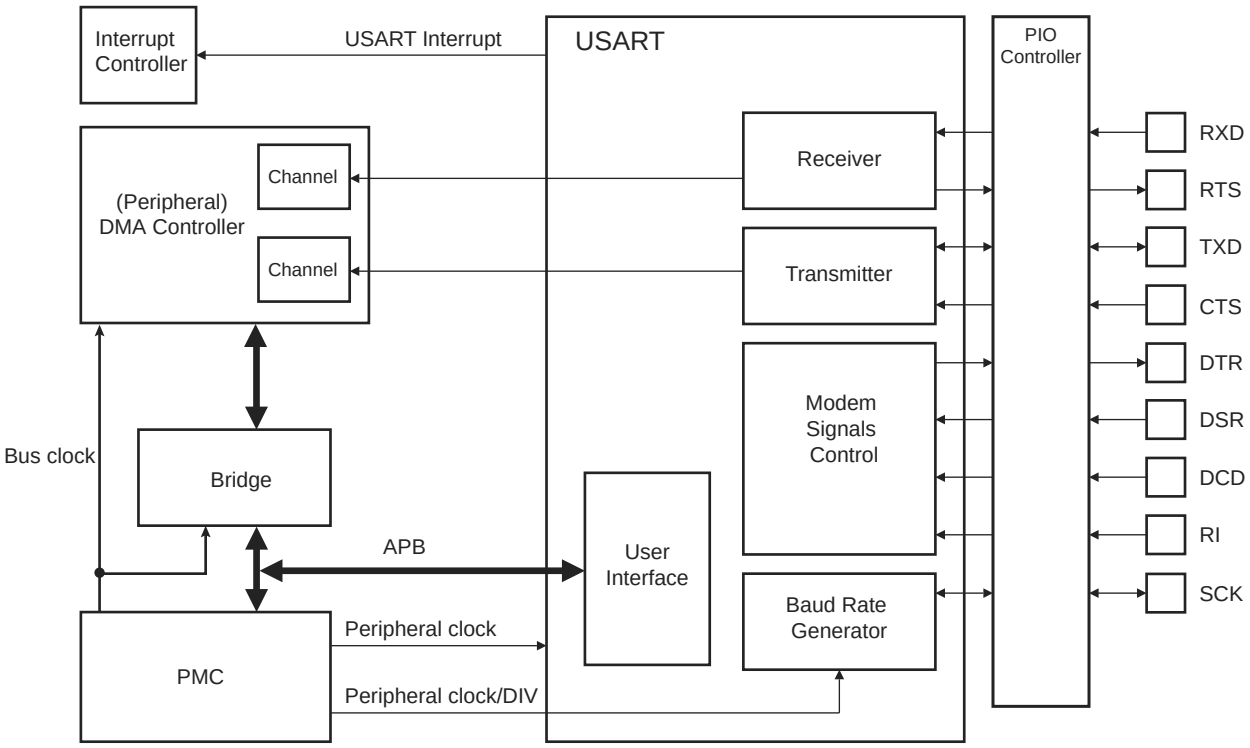
0: The corresponding interrupt is disabled.

1: The corresponding interrupt is enabled.

- **TXCOMP: Transmission Completed Interrupt Mask**
- **RXRDY: Receive Holding Register Ready Interrupt Mask**
- **TXRDY: Transmit Holding Register Ready Interrupt Mask**
- **SVACC: Slave Access Interrupt Mask**
- **GACC: General Call Access Interrupt Mask**
- **OVRE: Overrun Error Interrupt Mask**
- **NACK: Not Acknowledge Interrupt Mask**
- **ARBLST: Arbitration Lost Interrupt Mask**
- **SCL_WS: Clock Wait State Interrupt Mask**
- **EOSACC: End Of Slave Access Interrupt Mask**
- **ENDRX: End of Receive Buffer Interrupt Mask**
- **ENDTX: End of Transmit Buffer Interrupt Mask**
- **RXBUFF: Receive Buffer Full Interrupt Mask**
- **TXBUFE: Transmit Buffer Empty Interrupt Mask**

36.3 Block Diagram

Figure 36-1. USART Block Diagram

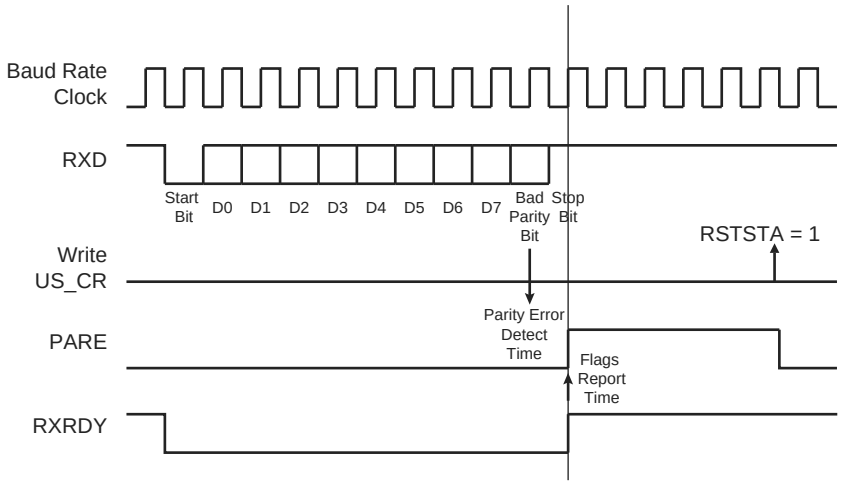


36.4 I/O Lines Description

Table 36-1. I/O Line Description

Name	Description	Type	Active Level
SCK	Serial Clock	I/O	—
TXD	Transmit Serial Data or Master Out Slave In (MOSI) in SPI master mode or Master In Slave Out (MISO) in SPI slave mode	I/O	—
RXD	Receive Serial Data or Master In Slave Out (MISO) in SPI master mode or Master Out Slave In (MOSI) in SPI slave mode	Input	—
RI	Ring Indicator	Input	Low
DSR	Data Set Ready	Input	Low
DCD	Data Carrier Detect	Input	Low
DTR	Data Terminal Ready	Output	Low
CTS	Clear to Send or Slave Select (NSS) in SPI slave mode	Input	Low
RTS	Request to Send or Slave Select (NSS) in SPI master mode	Output	Low

Figure 36-21. Parity Error



36.6.3.9 Multidrop Mode

If the value 0x6 or 0x07 is written to the PAR field in the US_MR, the USART runs in Multidrop mode. This mode differentiates the data characters and the address characters. Data is transmitted with the parity bit at 0 and addresses are transmitted with the parity bit at 1.

If the USART is configured in Multidrop mode, the receiver sets the PARE parity error bit when the parity bit is high and the transmitter is able to send a character with the parity bit high when a 1 is written to the SENTA bit in the US_CR.

To handle parity error, the PARE bit is cleared when a 1 is written to the RSTSTA bit in the US_CR.

The transmitter sends an address byte (parity bit set) when SENDA is written to in the US_CR. In this case, the next byte written to the US_THR is transmitted as an address. Any character written in the US_THR without having written the command SENDA is transmitted normally with the parity at 0.

36.6.3.10 Transmitter Timeguard

The timeguard feature enables the USART interface with slow remote devices.

The timeguard function enables the transmitter to insert an idle state on the TXD line between two characters. This idle state actually acts as a long stop bit.

The duration of the idle state is programmed in the TG field of the Transmitter Timeguard register (US_TTGR). When this field is written to zero no timeguard is generated. Otherwise, the transmitter holds a high level on TXD after each transmitted byte during the number of bit periods programmed in TG in addition to the number of stop bits.

As illustrated in Figure 36-22, the behavior of TXRDY and TXEMPTY status bits is modified by the programming of a timeguard. TXRDY rises only when the start bit of the next character is sent, and thus remains to 0 during the timeguard transmission if a character has been written in US_THR. TXEMPTY remains low until the timeguard transmission is completed as the timeguard is part of the current character being transmitted.

38.13 Register Write Protection

To prevent any single software error from corrupting HSMCI behavior, certain registers in the address space can be write-protected by setting the WPEN bit in the HSMCI Write Protection Mode Register (HSMCI_WPMR).

If a write access to a write-protected register is detected, the WPVS bit in the HSMCI Write Protection Status Register (HSMCI_WPSR) is set and the field WPVSRC indicates the register in which the write access has been attempted.

The WPVS bit is automatically cleared after reading the HSMCI_WPSR.

The following registers can be protected:

- HSMCI Mode Register
- HSMCI Data Timeout Register
- HSMCI SDCard/SDIO Register
- HSMCI Completion Signal Timeout Register
- HSMCI Configuration Register

Table 44-66. SSC Timings

Symbol	Parameter	Conditions		Min	Max	Unit
Transmitter						
SSC_0	TK Edge to TF/TD (TK Output, TF Output)	1.8V domain 3.3V domain		-3 -2.6	5.4 5.0	ns
SSC_1	TK Edge to TF/TD (TK Input, TF Output)	1.8V domain 3.3V domain		4.5 3.8	19.6 13.3	ns
SSC_2	TF Setup Time before TK Edge (TK Output)	1.8V domain 3.3V domain		18.9 12.0	–	ns
SSC_3	TF Hold Time after TK Edge (TK Output)	1.8V domain 3.3V domain		0	–	ns
SSC_4	TK Edge to TF/TD (TK Output, TF Input)	1.8V domain	–	2.6	5.4	ns
			STTDLY = 0 START = 4, 5 or 7	$2.6 + (2 \times t_{CPMCK})$ (1)	$5.4 + (2 \times t_{CPMCK})$ (1)	
		3.3V domain	–	2.3	5.0	
			STTDLY = 0 START = 4, 5 or 7	$2.3 + (2 \times t_{CPMCK})$ (1)	$5.0 + (2 \times t_{CPMCK})$ (1)	
SSC_5	TF Setup Time before TK Edge (TK Input)	1.8V domain 3.3V domain		0	–	ns
SSC_6	TF Hold Time after TK edge (TK Input)	1.8V domain 3.3V domain		t_{CPMCK}	–	ns
SSC_7	TK Edge to TF/TD (TK Input, TF Input)	1.8V domain	–	4.5	16.3	ns
			STTDLY = 0 START = 4, 5 or 7	$4.5 + (3 \times t_{CPMCK})$ (1)	$16.3 + (3 \times t_{CPMCK})$ (1)	
		3.3V domain	–	3.8	13.3	
			STTDLY = 0 START = 4, 5 or 7	$3.8 + (3 \times t_{CPMCK})$ (1)	$13.3 + (3 \times t_{CPMCK})$ (1)	
Receiver						
SSC_8	RF/RD Setup Time before RK Edge (RK Input)	1.8V domain 3.3V domain		0	–	ns
SSC_9	RF/RD Hold Time after RK Edge (RK Input)	1.8V domain 3.3V domain		t_{CPMCK}	–	ns
SSC_{10}	RK Edge to RF (RK Input)	1.8V domain 3.3V domain		4.7 4	16.1 12.8	ns
SSC_{11}	RF/RD Setup Time before RK Edge (RK Output)	1.8V domain 3.3V domain		$15.8 - t_{CPMCK}$ $12.5 - t_{CPMCK}$	–	ns
SSC_{12}	RF/RD Hold Time after RK Edge (RK Output)	1.8V domain 3.3V domain		$t_{CPMCK} - 4.3$ $t_{CPMCK} - 3.6$	–	ns
SSC_{13}	RK Edge to RF (RK Output)	1.8V domain 3.3V domain		-3 -2.6	4.3 3.8	ns