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"

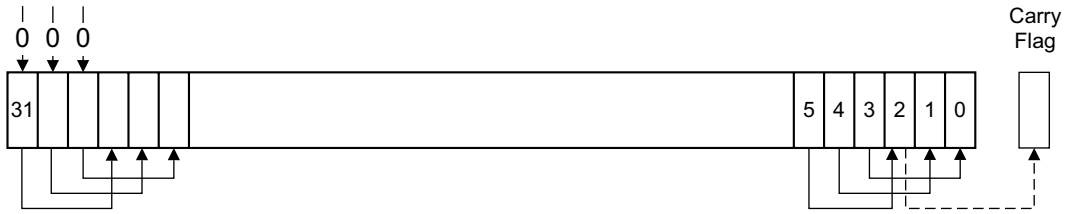
Details	
Product Status	Active
Core Processor	ARM® Cortex®-M4/M4F
Core Size	32-Bit Dual-Core
Speed	120MHz
Connectivity	EBI/EMI, I²C, IrDA, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, LCD, POR, PWM, WDT
Number of I/O	74
Program Memory Size	2MB (2M x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	304K x 8
Voltage - Supply (Vcc/Vdd)	1.62V ~ 3.6V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	100-LQFP
Supplier Device Package	100-LQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsam4c32ca-au

The LSR #n operation can be used to divide the value in the register Rm by 2^n, if the value is regarded as an unsigned integer.

When the instruction is LSRS or when LSR #n is used in *Operand2* with the instructions MOVS, MVNS, ANDS, ORRS, ORNS, EORS, BICS, TEQ or TST, the carry flag is updated to the last bit shifted out, bit[n-1], of the register Rm.

- If *n* is 32 or more, then all the bits in the result are cleared to 0.
- If *n* is 33 or more and the carry flag is updated, it is updated to 0.

Figure 12-9. LSR #3



LSL

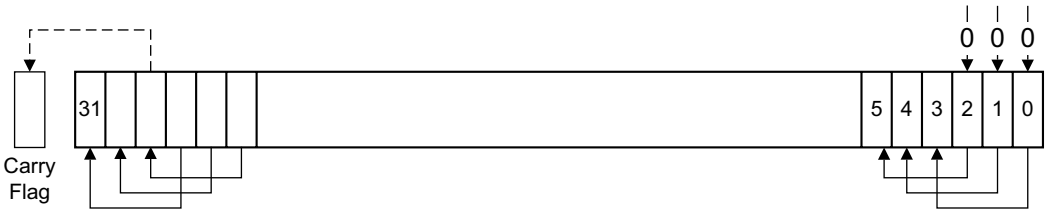
Logical shift left by *n* bits moves the right-hand 32-n bits of the register Rm, to the left by *n* places, into the left-hand 32-n bits of the result; and it sets the right-hand *n* bits of the result to 0. See Figure 12-10.

The LSL #n operation can be used to multiply the value in the register *Rm* by 2^n, if the value is regarded as an unsigned integer or a two's complement signed integer. Overflow can occur without warning.

When the instruction is LSLS or when LSL #n, with non-zero *n*, is used in *Operand2* with the instructions MOVS, MVNS, ANDS, ORRS, ORNS, EORS, BICS, TEQ or TST, the carry flag is updated to the last bit shifted out, bit[32-*n*], of the register *Rm*. These instructions do not affect the carry flag when used with LSL #0.

- If *n* is 32 or more, then all the bits in the result are cleared to 0.
- If *n* is 33 or more and the carry flag is updated, it is updated to 0.

Figure 12-10. LSL #3



ROR

Rotate right by *n* bits moves the left-hand 32-*n* bits of the register *Rm*, to the right by *n* places, into the right-hand 32-*n* bits of the result; and it moves the right-hand *n* bits of the register into the left-hand *n* bits of the result. See Figure 12-11.

When the instruction is RORS or when ROR #*n* is used in *Operand2* with the instructions MOVS, MVNS, ANDS, ORRS, ORNS, EORS, BICS, TEQ or TST, the carry flag is updated to the last bit rotation, bit[*n*-1], of the register *Rm*.

- If *n* is 32, then the value of the result is same as the value in *Rm*, and if the carry flag is updated, it is updated to bit[31] of *Rm*.
- ROR with shift length, *n*, more than 32 is the same as ROR with shift length *n*-32.

12.6.10.2 CBZ and CBNZ

Compare and Branch on Zero, Compare and Branch on Non-Zero.

Syntax

```
CBZ Rn, label
CBNZ Rn, label
```

where:

Rn is the register holding the operand.

label is the branch destination.

Operation

Use the CBZ or CBNZ instructions to avoid changing the condition code flags and to reduce the number of instructions.

CBZ Rn, label does not change condition flags but is otherwise equivalent to:

```
CMP     Rn, #0
BEQ     label
```

CBNZ Rn, label does not change condition flags but is otherwise equivalent to:

```
CMP     Rn, #0
BNE     label
```

Restrictions

The restrictions are:

- *Rn* must be in the range of R0 to R7
- The branch destination must be within 4 to 130 bytes after the instruction
- These instructions must not be used inside an IT block.

Condition Flags

These instructions do not change the flags.

Examples

```
CBZ    R5, target  ; Forward branch if R5 is zero
CBNZ   R0, target  ; Forward branch if R0 is not zero
```

12.9.1.8 System Handler Priority Registers

The SCB_SHPR1–SCB_SHPR3 registers set the priority level, 0 to 15 of the exception handlers that have configurable priority. They are byte-accessible.

The system fault handlers and the priority field and register for each handler are:

Table 12-34. System Fault Handler Priority Fields

Handler	Field	Register Description
Memory management fault (MemManage)	PRI_4	System Handler Priority Register 1
Bus fault (BusFault)	PRI_5	
Usage fault (UsageFault)	PRI_6	
SVCall	PRI_11	System Handler Priority Register 2
PendSV	PRI_14	System Handler Priority Register 3
SysTick	PRI_15	

Each PRI_N field is 8 bits wide, but the processor implements only bits [7:4] of each field, and bits [3:0] read as zero and ignore writes.

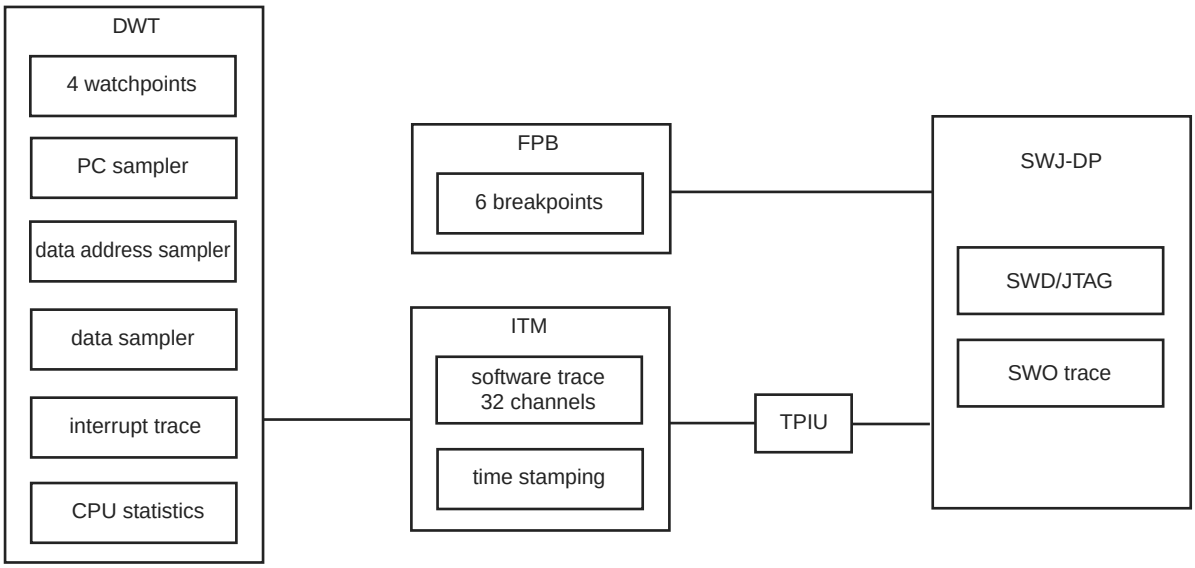
13.7.2 Debug Architecture

Figure 13-5 illustrates the debug architecture. The Cortex-M4 embeds four functional units for debug:

- SWJ-DP (Serial Wire/JTAG Debug Port)
- FPB (Flash Patch Breakpoint)
- DWT (Data Watchpoint and Trace)
- ITM (Instrumentation Trace Macrocell)
- TPIU (Trace Port Interface Unit)

The information that follows is mainly dedicated to developers of SWJ-DP Emulators/Probes and debugging tool vendors for Cortex-M4 based microcontrollers. For further details on SWJ-DP, refer to the Cortex-M4 technical reference manual.

Figure 13-5. Debug Architecture



13.7.3 Serial Wire/JTAG Debug Port (SWJ-DP)

The Cortex-M4 embeds a SWJ-DP Debug port which is the standard CoreSight debug port. It combines the Serial Wire Debug Port (SW-DP), from 2 to 3 pins, and the JTAG Debug Port (JTAG-DP), 5 pins.

By default, the JTAG-DP is active. If the host debugger needs to switch to the SW-DP, it must provide a dedicated JTAG sequence on TMS/SWDIO and TCK/SWCLK. This disables JTAG-DP and enables SW-DP.

When SW-DP is active, TDO/TRACESWO can be used for trace. The asynchronous TRACE output (TRACESWO) is multiplexed with TDO and thus the asynchronous trace can only be used with SW-DP.

Table 13-2. SWJ-DP Pin List

Pin Name	JTAG Port	Serial Wire Debug Port
TMS/SWDIO	TMS	SWDIO
TCK/SWCLK	TCK	SWCLK
TDI	TDI	-
TDO/TRACESWO	TDO	TRACESWO (optional: trace)

SW-DP or JTAG-DP mode is selected when JTAGSEL is low. It is not possible to switch directly between SWJ-DP and JTAG boundary scan operations. A chip reset must be performed after JTAGSEL is changed.

16. Real-time Timer (RTT)

16.1 Description

The Real-time Timer (RTT) is built around a 32-bit counter used to count roll-over events of the programmable 16-bit prescaler driven from the 32-kHz slow clock source. It generates a periodic interrupt and/or triggers an alarm on a programmed value.

The RTT can also be configured to be driven by the 1Hz RTC signal, thus taking advantage of a calibrated 1Hz clock.

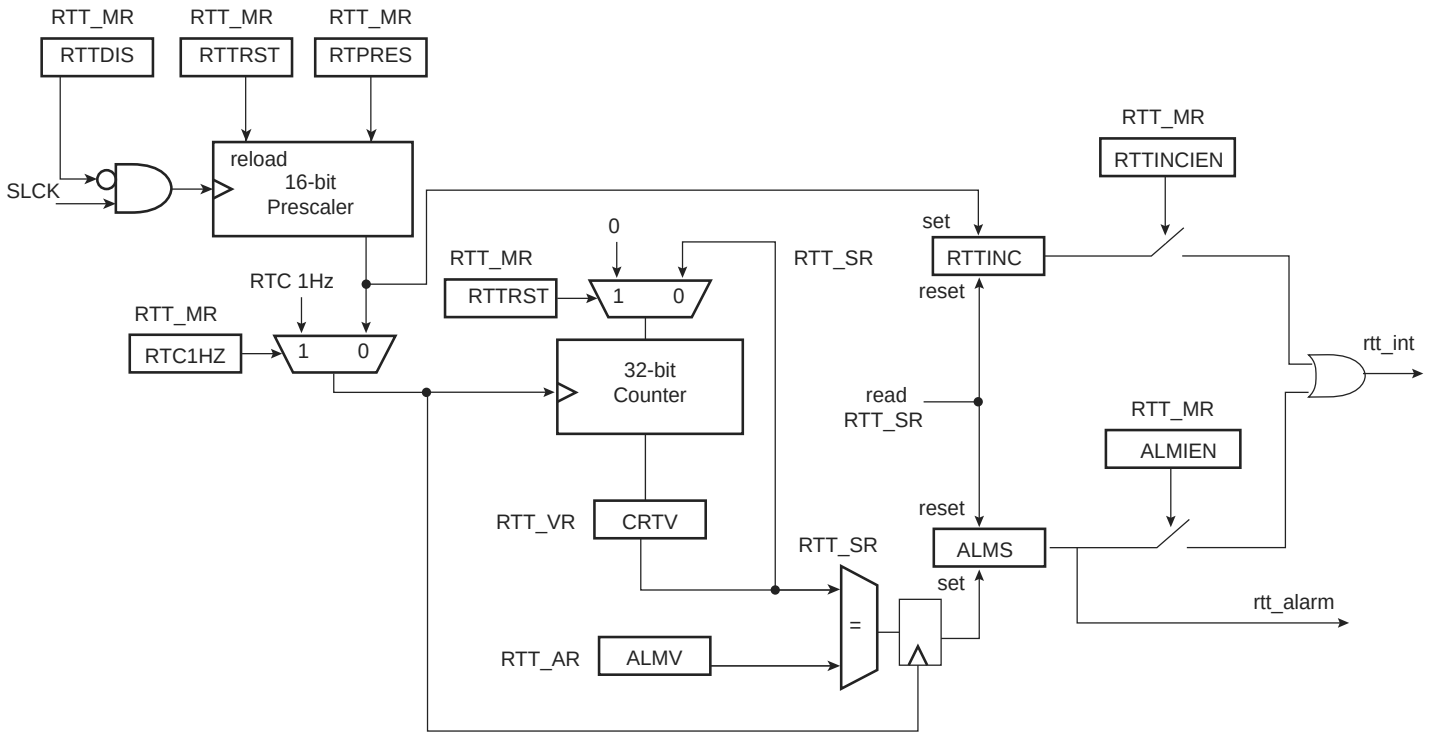
The slow clock source can be fully disabled to reduce power consumption when only an elapsed seconds count is required.

16.2 Embedded Characteristics

- 32-bit Free-running Counter on prescaled slow clock or RTC calibrated 1Hz clock
- 16-bit Configurable Prescaler
- Interrupt on Alarm or Counter Increment

16.3 Block Diagram

Figure 16-1. Real-time Timer



18.5.2 Watchdog Timer Mode Register

Name: WDT_MR

Address: 0x400E1454

Access: Read/Write Once

31	30	29	28	27	26	25	24
–	–	WDIDLEHLT	WDDBGHLT	WDD			
23	22	21	20	19	18	17	16
WDD							
15	14	13	12	11	10	9	8
WDDIS	WDRPROC	WDRSTEN	WDFIEN	WDV			
7	6	5	4	3	2	1	0
WDV							

Note: The first write access prevents any further modification of the value of this register. Read accesses remain possible.

Note: The WDD and WDV values must not be modified within three slow clock periods following a restart of the watchdog performed by a write access in WDT_CR. Any modification will cause the watchdog to trigger an end of period earlier than expected.

- **WDV: Watchdog Counter Value**

Defines the value loaded in the 12-bit watchdog counter.

- **WDFIEN: Watchdog Fault Interrupt Enable**

0: A watchdog fault (underflow or error) has no effect on interrupt.

1: A watchdog fault (underflow or error) asserts interrupt.

- **WDRSTEN: Watchdog Reset Enable**

0: A watchdog fault (underflow or error) has no effect on the resets.

1: A watchdog fault (underflow or error) triggers a watchdog reset.

- **WDRPROC: Watchdog Reset Processor**

0: If WDRSTEN is 1, a watchdog fault (underflow or error) activates all resets.

1: If WDRSTEN is 1, a watchdog fault (underflow or error) activates the processor reset.

- **WDDIS: Watchdog Disable**

0: Enables the Watchdog Timer.

1: Disables the Watchdog Timer.

- **WDD: Watchdog Delta Value**

Defines the permitted range for reloading the Watchdog Timer.

If the Watchdog Timer value is less than or equal to WDD, setting bit WDT_CR.WDRSTT restarts the timer.

If the Watchdog Timer value is greater than WDD, setting bit WDT_CR.WDRSTT causes a watchdog error.

26.2.1.2 Matrix 0 Slaves

The Bus Matrix manages the slaves listed in Table 26-2. Each slave has its own arbiter providing a dedicated arbitration per slave.

Table 26-2. List of Bus Matrix Slaves

Slave 0	Internal SRAM0
Slave 1	Internal ROM
Slave 2	Internal Flash
Slave 3	External Bus Interface
Slave 4	Peripheral Bridge 0
Slave 5	CPKCC RAM and ROM
Slave 6	Matrix1
Slave 7	CMCC0
Slave 8	USB DPRAM

26.2.1.3 Master to Slave Access (Matrix 0)

Table 26-3 gives valid paths for master to slave access on Matrix 0. The paths shown as "-" are forbidden or not wired, e.g. access from the Cortex-M4 S Bus to the Internal ROM.

Table 26-3. Matrix 0 Master to Slave Access

Slaves	Masters	0 Cortex-M4 I/D Bus	1 Cortex-M4 S Bus	2 PDC0	3 ICM	4 Matrix1	5 EBI Matrix 1	6 CMCC0	7 USB DMA
0	Internal SRAM0	-	X	X	X	X	-	-	-X
1	Internal ROM	X	-	X	X	-	-	-	-
2	Internal Flash	X	-	-	X	X	-	X	-
3	External Bus Interface	X	X	X	X	-	X	X	-X
4	Peripheral Bridge 0	-	X	X	-	X	-	-	-
5	CPKCC SRAM, ROM	-	X	-	X	-	-	-	-
6	Matrix1	-	X	-	X	-	-	-	-
7	CMCC0	X	-	-	-	-	-	-	-
8	USB DPRAM	-	X	X	-	-	-	-	-

26.2.1.4 Accesses through Matrix 0

- CM4P0 I/D Bus access to:
  - Flash, ROM
  - EBI (0x03000000 to 0x06FFFFFF)
  - Flash and EBI through Cache Controller CMCC0 (respectively through 0x11000000 to 0x11FFFFFF and 0x13000000 to 0x16FFFFFF)
- CMP4P0 S Bus access to:
  - SRAM0, SRAM1 through Matrix1, SRAM2 through Matrix1
  - PKCC
  - EBI (through 0x60000000 to 0x63FFFFFF, 0xA0000000 to A3FFFFFF)

27.16.1 SMC Setup Register

Name: SMC_SETUP[0..3]

Address: 0x400E0000 (0)[0], 0x400E0010 (0)[1], 0x400E0020 (0)[2], 0x400E0030 (0)[3], 0x4801C000 (1)[0], 0x4801C010 (1)[1], 0x4801C020 (1)[2], 0x4801C030 (1)[3]

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	NCS_RD_SETUP					

23	22	21	20	19	18	17	16
–	–	NRD_SETUP					

15	14	13	12	11	10	9	8
–	–	NCS_WR_SETUP					

7	6	5	4	3	2	1	0
–	–	NWE_SETUP					

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

- **NWE_SETUP: NWE Setup Length**

The NWE signal setup length is defined as:

NWE setup length = (128* NWE_SETUP[5] + NWE_SETUP[4:0]) clock cycles

- **NCS_WR_SETUP: NCS Setup Length in WRITE Access**

In write access, the NCS signal setup length is defined as:

NCS setup length = (128* NCS_WR_SETUP[5] + NCS_WR_SETUP[4:0]) clock cycles

- **NRD_SETUP: NRD Setup Length**

The NRD signal setup length is defined in clock cycles as:

NRD setup length = (128* NRD_SETUP[5] + NRD_SETUP[4:0]) clock cycles

- **NCS_RD_SETUP: NCS Setup Length in READ Access**

In read access, the NCS signal setup length is defined as:

NCS setup length = (128* NCS_RD_SETUP[5] + NCS_RD_SETUP[4:0]) clock cycles

Access: Write-only

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

23	22	21	20	19	18	17	16
CPKEY				–	–	CPBMCK	CPCK

15	14	13	12	11	10	9	8
–	–	–	–	–	PCK2	PCK1	PCK0

7	6	5	4	3	2	1	0
UHDP	–	–	–	–	–	–	–

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

- **UHDP: USB Host/Device Port Clock Disable**

0: No effect.

1: Disables the 48 MHz clock (UHDPCK) of the USB Host/Device Port.

- **PCKx: Programmable Clock x Output Disable**

0: No effect.

1: Disables the corresponding Programmable Clock output.

- **CPCK: Coprocessor Clocks Disable**

0: No effect.

1: Enables the corresponding Coprocessor Clocks (CPHCLK, CPFCLK, CPSYSTICK) if CPKEY = 0xA.

- **CPBMCK: Coprocessor Bus Master Clocks Disable**

0: No effect.

1: Disables the corresponding Coprocessor Bus Master Clock (CPBMCK, CPFCLK) if CPKEY = 0xA.

Note: Disabling CPBMCK must not be performed if CPCK is 1 in PMC_SCSR.

- **CPKEY: Coprocessor Clocks Disable Key**

Value	Name	Description
0xA	PASSWD	This field must be written to 0xA in order to validate CPCK field.

32.6.22 PIO Pull-Up Enable Register

Name: PIO_PUER

Address: 0x400E0E64 (PIOA), 0x400E1064 (PIOB), 0x4800C064 (PIOC), 0x400E1264 (PIOD)

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

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

- **P0–P31: Pull-Up Enable**

0: No effect.

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

33. Serial Peripheral Interface (SPI)

33.1 Description

The Serial Peripheral Interface (SPI) circuit is a synchronous serial data link that provides communication with external devices in Master or Slave mode. It also enables communication between processors if an external processor is connected to the system.

The Serial Peripheral Interface is essentially a Shift register that serially transmits data bits to other SPIs. During a data transfer, one SPI system acts as the “master”' which controls the data flow, while the other devices act as “slaves'' which have data shifted into and out by the master. Different CPUs can take turn being masters (multiple master protocol, contrary to single master protocol where one CPU is always the master while all of the others are always slaves). One master can simultaneously shift data into multiple slaves. However, only one slave can drive its output to write data back to the master at any given time.

A slave device is selected when the master asserts its NSS signal. If multiple slave devices exist, the master generates a separate slave select signal for each slave (NPCS).

The SPI system consists of two data lines and two control lines:

- Master Out Slave In (MOSI)—This data line supplies the output data from the master shifted into the input(s) of the slave(s).
- Master In Slave Out (MISO)—This data line supplies the output data from a slave to the input of the master. There may be no more than one slave transmitting data during any particular transfer.
- Serial Clock (SPCK)—This control line is driven by the master and regulates the flow of the data bits. The master can transmit data at a variety of baud rates; there is one SPCK pulse for each bit that is transmitted.
- Slave Select (NSS)—This control line allows slaves to be turned on and off by hardware.

33.2 Embedded Characteristics

- Master or Slave Serial Peripheral Bus Interface
  - 8-bit to 16-bit programmable data length per chip select
  - Programmable phase and polarity per chip select
  - Programmable transfer delay between consecutive transfers and delay before SPI clock per chip select
  - Programmable delay between chip selects
  - Selectable mode fault detection
- Master Mode can drive SPCK up to Peripheral Clock
- Master Mode Bit Rate can be Independent of the Processor/Peripheral Clock
- Slave mode operates on SPCK, asynchronously with core and bus clock
- Four chip selects with external decoder support allow communication with up to 15 peripherals
- Communication with Serial External Devices Supported
  - Serial memories, such as DataFlash and 3-wire EEPROMs
  - Serial peripherals, such as ADCs, DACs, LCD controllers, CAN controllers and sensors
  - External coprocessors
- Connection to PDC Channel Capabilities, Optimizing Data Transfers
  - One channel for the receiver
  - One channel for the transmitter
- Register Write Protection

- **BITS: Bits Per Transfer**

(See note below the register bitmap.)

The BITS field determines the number of data bits transferred. Reserved values should not be used.

Value	Name	Description
0	8_BIT	8 bits for transfer
1	9_BIT	9 bits for transfer
2	10_BIT	10 bits for transfer
3	11_BIT	11 bits for transfer
4	12_BIT	12 bits for transfer
5	13_BIT	13 bits for transfer
6	14_BIT	14 bits for transfer
7	15_BIT	15 bits for transfer
8	16_BIT	16 bits for transfer
9	–	Reserved
10	–	Reserved
11	–	Reserved
12	–	Reserved
13	–	Reserved
14	–	Reserved
15	–	Reserved

- **SCBR: Serial Clock Bit Rate**

In Master mode, the SPI Interface uses a modulus counter to derive the SPCK bit rate from the peripheral clock. The bit rate is selected by writing a value from1 to 255 in the SCBR field. The following equation determines the SPCK bit rate:

SCBR = $f_{peripheral\ clock}$ / SPCK Bit Rate

Programming the SCBR field to 0 is forbidden. Triggering a transfer while SCBR is at 0 can lead to unpredictable results.

If BRSRCCLK = 1 in SPI_MR, SCBR must be programmed with a value greater than 1.

At reset, SCBR is 0 and the user has to program it at a valid value before performing the first transfer.

Note: If one of the SCBR fields in SPI_CSRx is set to 1, the other SCBR fields in SPI_CSRx must be set to 1 as well, if they are used to process transfers. If they are not used to transfer data, they can be set at any value.

- **DLYBS: Delay Before SPCK**

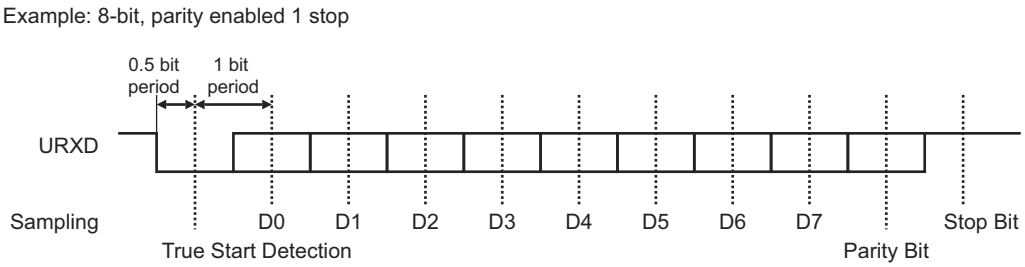
This field defines the delay from NPCS falling edge (activation) to the first valid SPCK transition.

When DLYBS = 0, the delay is half the SPCK clock period.

Otherwise, the following equation determines the delay:

DLYBS = Delay Before SPCK × $f_{peripheral\ clock}$

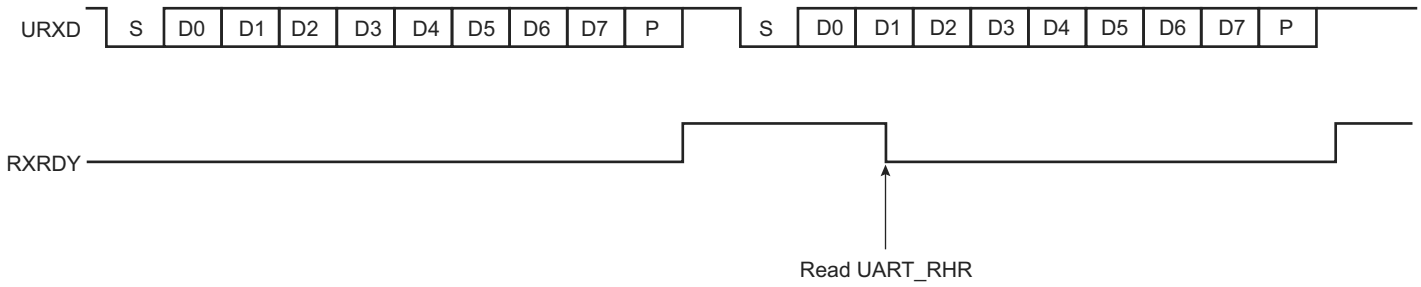
Figure 35-4. Character Reception



35.5.2.3 Receiver Ready

When a complete character is received, it is transferred to the Receive Holding Register (UART_RHR) and the RXRDY status bit in the Status Register (UART_SR) is set. The bit RXRDY is automatically cleared when UART_RHR is read.

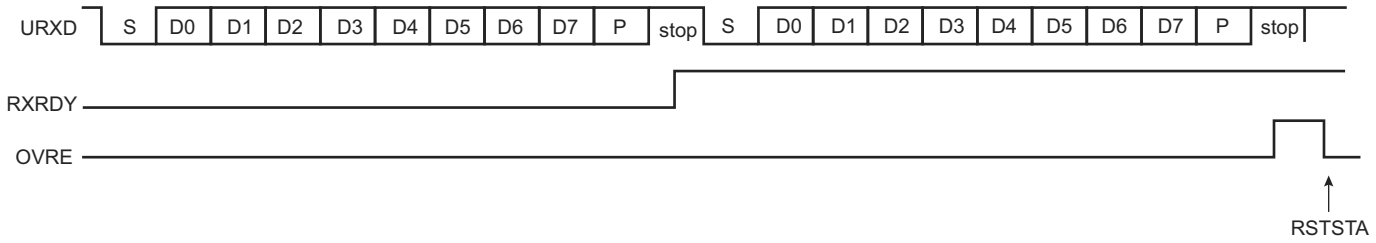
Figure 35-5. Receiver Ready



35.5.2.4 Receiver Overrun

The OVRE status bit in UART_SR is set if UART_RHR has not been read by the software (or the PDC) since the last transfer, the RXRDY bit is still set and a new character is received. OVRE is cleared when the software writes a 1 to the bit RSTSTA (Reset Status) in UART_CR.

Figure 35-6. Receiver Overrun



35.5.2.5 Parity Error

Each time a character is received, the receiver calculates the parity of the received data bits, in accordance with the field PAR in the Mode Register (UART_MR). It then compares the result with the received parity bit. If different, the parity error bit PARE in UART_SR is set at the same time RXRDY is set. The parity bit is cleared when UART_CR is written with the bit RSTSTA (Reset Status) at 1. If a new character is received before the reset status command is written, the PARE bit remains at 1.

35.6.9 UART Baud Rate Generator Register

Name: UART_BRGR

Address: 0x400E0620 (0), 0x48004020 (1)

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
CD							
7	6	5	4	3	2	1	0
CD							

- **CD: Clock Divisor**

0: Baud rate clock is disabled

1 to 65,535:

$$CD = \frac{f_{peripheral\ clock}}{16 \times Baud\ Rate}$$

36.7.16 USART Receiver Time-out Register

Name: US_RTOR

Address: 0x40024024 (0), 0x40028024 (1), 0x4002C024 (2), 0x40030024 (3), 0x40034024 (4)

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
TO							
7	6	5	4	3	2	1	0
TO							

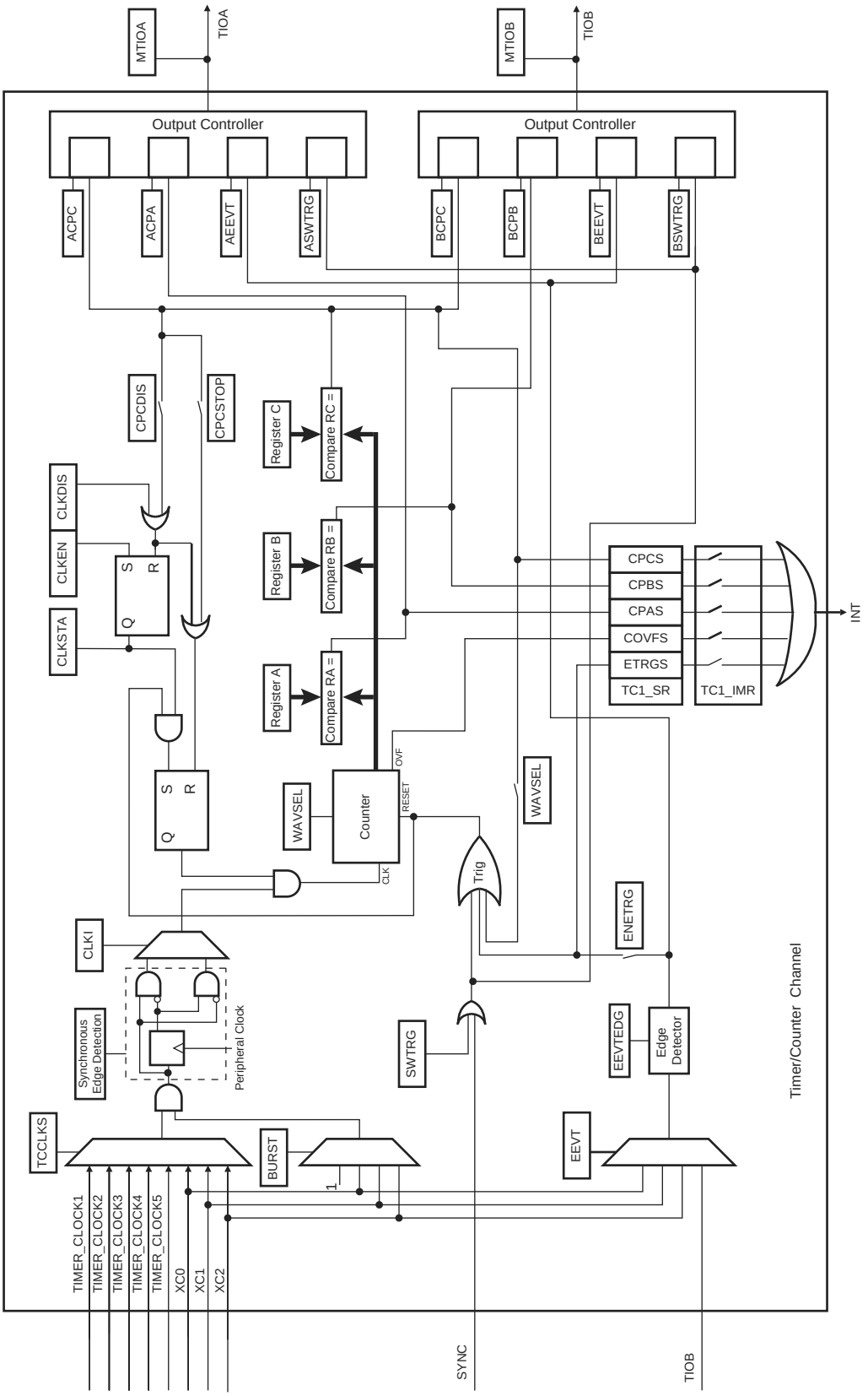
This register can only be written if the WPEN bit is cleared in the USART Write Protection Mode Register.

- **TO: Time-out Value**

0: The receiver time-out is disabled.

1–65535: The receiver time-out is enabled and TO is Time-out Delay / Bit Period.

Figure 37-6. Waveform Mode



37.6.11.4 WAVSEL = 11

When WAVSEL = 11, the value of TC_CV is incremented from 0 to RC. Once RC is reached, the value of TC_CV is decremented to 0, then re-incremented to RC and so on. See Figure 37-13.

A trigger such as an external event or a software trigger can modify TC_CV at any time. If a trigger occurs while TC_CV is incrementing, TC_CV then decrements. If a trigger is received while TC_CV is decrementing, TC_CV then increments. See Figure 37-14.

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

Figure 37-13. WAVSEL = 11 without Trigger

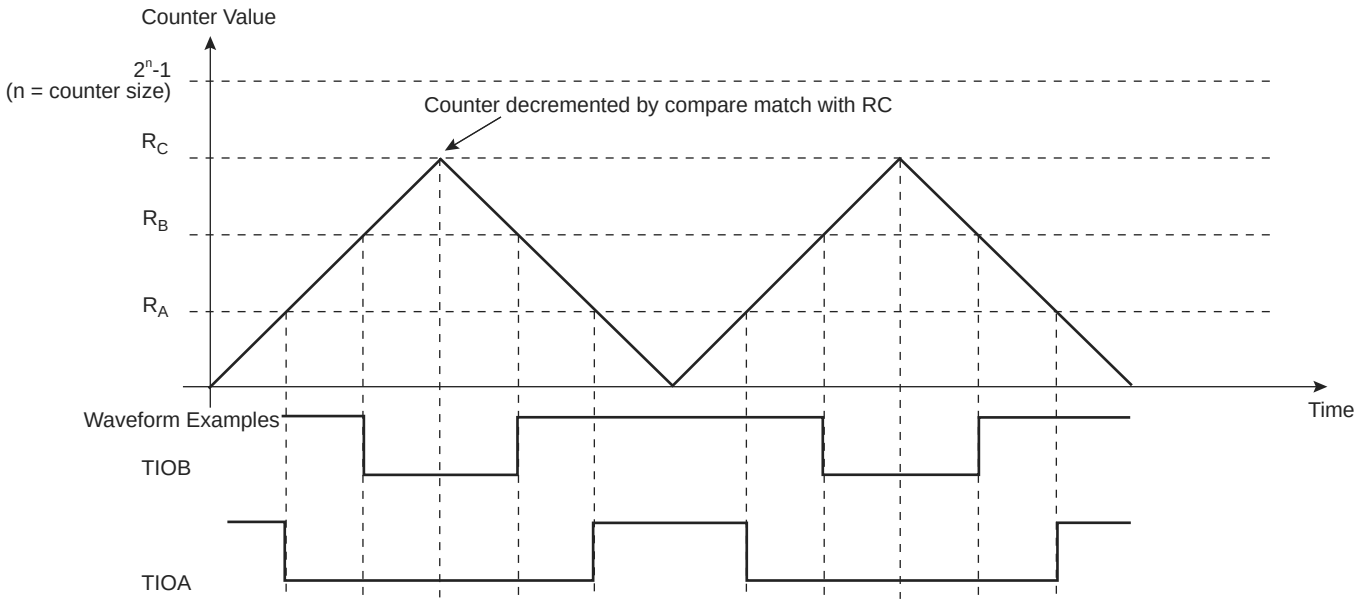
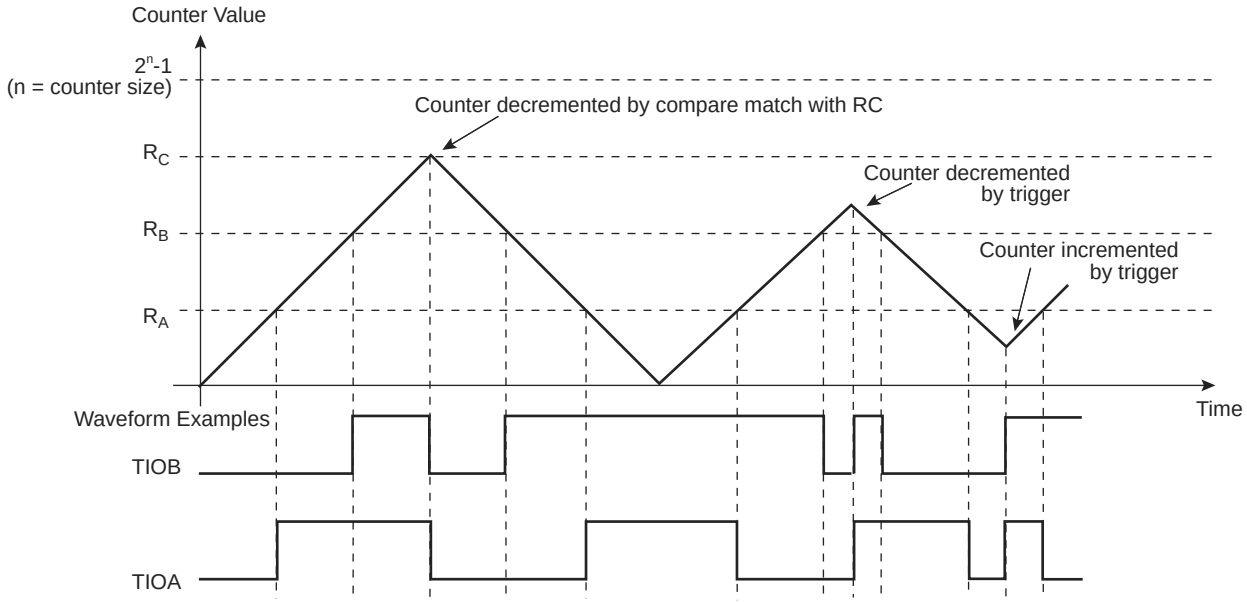


Figure 37-14. WAVSEL = 11 with Trigger



42.5.2.2 ICM Region Configuration Structure Member

Name: ICM_RCFG

Address: ICM_DSCR+0x004+RID*(0x10)

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	MRPROT					
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	ALGO			–	PROCDLY	SUIEN	ECIEN
7	6	5	4	3	2	1	0
WCIEN	BEIEN	DMIEN	RHIEN	–	EOM	WRAP	CDWBN

- **CDWBN: Compare Digest or Write Back Digest**

0: The digest is written to the Hash area.

1: The digest value is compared to the digest stored in the Hash area.

- **WRAP: Wrap Command**

0: The next region descriptor address loaded is the current region identifier descriptor address incremented by 0x10.

1: The next region descriptor address loaded is ICM_DSCR.

- **EOM: End Of Monitoring**

0: The current descriptor does not terminate the monitoring.

1: The current descriptor terminates the Main List. WRAP bit value has no effect.

- **RHIEN: Region Hash Completed Interrupt Disable (Default Enabled)**

0: The ICM_ISR RHC[*i*] flag is set when the field NEXT = 0 in a descriptor of the main or second list.

1: The ICM_ISR RHC[*i*] flag remains cleared even if the setting condition is met.

- **DMIEN: Digest Mismatch Interrupt Disable (Default Enabled)**

0: The ICM_ISR RBE[*i*] flag is set when the hash value just calculated from the processed region differs from expected hash value.

1: The ICM_ISR RBE[*i*] flag remains cleared even if the setting condition is met.

- **BEIEN: Bus Error Interrupt Disable (Default Enabled)**

0: The flag is set when an error is reported on the system bus by the bus MATRIX.

1: The flag remains cleared even if the setting condition is met.

- **WCIEN: Wrap Condition Interrupt Disable (Default Enabled)**

0: The ICM_ISR RWC[*i*] flag is set when the WRAP bit is set in a descriptor of the main list.

1: The ICM_ISR RWC[*i*] flag remains cleared even if the setting condition is met.

46.7.4.2 Test Setup 2: CoreMark

- CoreMark on Core 1 (CM4P1) running out of SRAM1 (Code) / SRAM2 (Data)
- Core 0 (CM4P0) in Sleep mode.

Table 46-59. SAM4C4/8/16 Test Setup 2 Current Consumption

Clock (MHz)	SRAM1, SRAM2			Unit
	IDD_IN (AMP1)	IDD_I0 (AMP2)	IDD_CORE (AMP3)	
120	22.3	0.22	19.0	mA
100	18.7	0.22	16.0	
84	15.8	0.22	13.6	
64	12.1	0.22	10.5	
48	9.2	0.22	7.9	
32	7.1	0.22	5.5	
24	5.4	0.22	4.2	
12	2.1	0.01	2.1	
8	1.4	0.01	1.4	
4	0.78	0.01	0.77	
2	0.46	0.01	0.45	
1	0.29	0.01	0.28	
0.5	0.21	0.01	0.2	
0.25	0.13	0.01	0.12	

Table 46-60. SAM4C32 Test Setup 2 Current Consumption

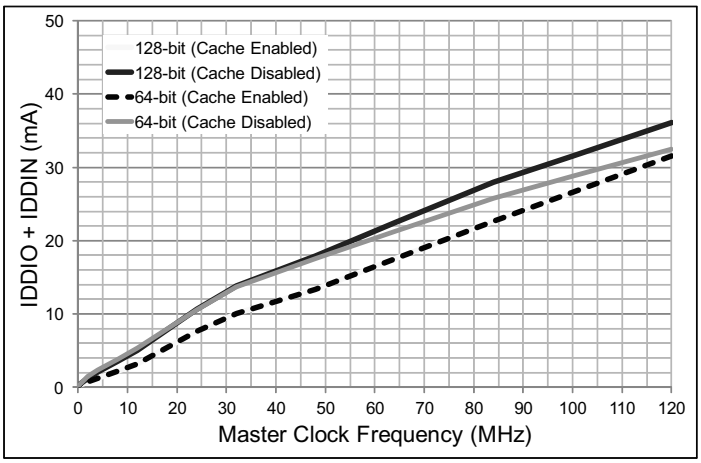
Clock (MHz)	SRAM1, SRAM2			Unit
	IDD_IN (AMP1)	IDD_I0 (AMP2)	IDD_CORE (AMP3)	
120	23.8	0.3	20.6	mA
100	20.0	0.3	17.3	
84	16.9	0.3	14.7	
64	13.0	0.3	11.3	
48	9.8	0.3	8.6	
32	7.6	0.3	5.9	
24	5.7	0.3	4.5	
12	2.3	0.09	2.3	
8	1.6	0.09	1.5	
4	0.86	0.09	0.84	
2	0.5	0.09	0.49	
1	0.32	0.09	0.31	
0.5	0.24	0.09	0.23	
0.25	0.15	0.09	0.14	

Table 46-62. SAM4C32 Test Setup 3 Current Consumption

Clock (MHz)	128-bit Flash Access						64-bit Flash Access						Unit
	Cache Enabled			Cache Disabled			Cache Enabled			Cache Disabled			
	IDD_IN (AMP1)	IDD_I0 (AMP2)	IDD_CORE (AMP3)	IDD_IN (AMP1)	IDD_I0 (AMP2)	IDD_CORE (AMP3)	IDD_IN (AMP1)	IDD_I0 (AMP2)	IDD_CORE (AMP3)	IDD_IN (AMP1)	IDD_I0 (AMP2)	IDD_CORE (AMP3)	
120	35.0	0.23	31.7	38.4	2.1	35.1	34.9	0.23	31.6	33.8	1.8	30.5	mA
100	29.5	0.23	26.8	33.8	2.0	31.0	29.4	0.23	26.7	29.5	1.7	27.0	
84	25.1	0.23	22.8	29.4	1.8	27.1	24.9	0.23	22.7	26.6	1.7	24.3	
64	19.3	0.23	17.7	23.2	1.5	21.5	19.2	0.23	17.6	21.8	1.5	20.1	
48	14.7	0.23	13.4	18.0	1.3	16.8	14.6	0.23	13.4	17.7	1.5	16.5	
32	10.9	0.23	9.2	13.3	1.1	11.7	10.8	0.23	9.2	13.5	1.3	11.8	
24	8.2	0.23	7.0	10.5	1.0	9.3	8.2	0.22	7.0	10.3	1.2	9.0	
12	3.5	0.02	3.5	4.8	0.86	4.7	3.5	0.02	3.5	4.7	1.1	4.6	
8	2.4	0.02	2.4	3.2	0.74	3.2	2.4	0.02	2.4	3.1	1.0	3.1	
4	1.3	0.02	1.3	1.7	0.42	1.7	1.3	0.02	1.3	1.7	0.87	1.7	
2	0.72	0.02	0.71	0.92	0.40	0.89	0.71	0.02	0.81	0.94	0.56	0.94	
1	0.43	0.02	0.42	0.52	0.18	0.52	0.43	0.02	0.42	0.55	0.36	0.54	
0.5	0.29	0.02	0.28	0.36	0.09	0.36	0.29	0.02	0.28	0.35	0.18	0.34	
0.25	0.16	0.02	0.15	0.18	0.02	0.16	0.16	0.02	0.15	0.17	0.06	0.16	

Figure 46-29. Typical Current Consumption in Active Mode (Test Setup 3)

SAM4Cx4/8/16



SAM4Cx32