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	CANbus, Ethernet, IrDA, MMC/SD, SPI, 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	128K 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 ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	100-TFBGA
Supplier Device Package	100-TFBGA (9x9)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsam4e16ca-cu

4.2 144-ball LFBGA Package and Pinout

4.2.1 144-ball LFBGA Package Outline

The 144-ball LFBGA package has a 0.8 mm ball pitch and respects Green Standards. Refer to Section 47.2 “144-ball LFBGA Package Drawing” for details.

4.2.2 144-ball LFBGA Pinout

Table 4-2. SAM4E 144-ball LFBGA Pinout

A1	PE1	D1	ADVREF	G1	PC15	K1	PE4
A2	PB9	D2	GND	G2	PC13	K2	PA21/PGMD9
A3	PB8	D3	PD31	G3	PB1	K3	PA22/PGMD10
A4	PB11	D4	PD0	G4	GND	K4	PC2
A5	PD2	D5	GNDPLL	G5	GND	K5	PA16/PGMD4
A6	PA29	D6	PD4	G6	GND	K6	PA14/PGMD2
A7	PC21	D7	PD5	G7	GND	K7	PC6
A8	PD6	D8	PC19	G8	VDDIO	K8	PA25/PGMD13
A9	PC20	D9	PD9	G9	PD13	K9	PD20
A10	PA30	D10	PD29	G10	PD12	K10	PD28
A11	PD15	D11	PC16	G11	PC9	K11	PD16
A12	PB4	D12	PA1/PGMEN1	G12	PB12	K12	PA4/PGMNCMD
B1	PE2	E1	PC31	H1	PA19/PGMD7	L1	PE5
B2	PB13	E2	PC27	H2	PA18/PGMD6	L2	PA7/PGMNVALID
B3	VDDPLL	E3	PE3	H3	PA20/PGMD8	L3	PC3
B4	PB10	E4	PC0	H4	PB0	L4	PA23/PGMD11
B5	PD1	E5	GND	H5	VDDCORE	L5	PA15/PGMD3
B6	PC24	E6	GND	H6	VDDIO	L6	PD26
B7	PD3	E7	VDDIO	H7	VDDIO	L7	PA24/PGMD12
B8	PD7	E8	VDDCORE	H8	VDDCORE	L8	PC5
B9	PA6/PGMNOE	E9	PD8	H9	PD21	L9	PA10/PGMM2
B10	PC18	E10	PC14	H10	PD14	L10	PA12/PGMD0
B11	JTAGSEL	E11	PD11	H11	TEST	L11	PD17
B12	PC17	E12	PA2	H12	NRST	L12	PC28
C1	VDDIN	F1	PC30	J1	PA17/PGMD5	M1	PD30
C2	PE0	F2	PC26	J2	PB2	M2	PA8/PGMM0
C3	VDDOUT	F3	PC29	J3	PB3	M3	PA13/PGMD1
C4	PB14	F4	PC12	J4	PC1	M4	PC7
C5	PC25	F5	GND	J5	PC4	M5	PD25
C6	PC23	F6	GND	J6	PD27	M6	PD24
C7	PC22	F7	GND	J7	VDDCORE	M7	PD23
C8	PA31	F8	VDDIO	J8	PA26/PGMD14	M8	PD22
C9	PA28	F9	PB7	J9	PA11/PGMM3	M9	PD19
C10	PB5	F10	PC10	J10	PA27/PGMD15	M10	PD18
C11	PA0/PGMEN0	F11	PC11	J11	PB6	M11	PA5/PGMRDY
C12	PD10	F12	PA3	J12	PC8	M12	PA9/PGMM1

11.4.1.4 General-purpose Registers

R0–R12 are 32-bit general-purpose registers for data operations.

11.4.1.5 Stack Pointer

The *Stack Pointer* (SP) is register R13. In Thread mode, bit[1] of the Control Register indicates the stack pointer to use:

- 0 = *Main Stack Pointer* (MSP). This is the reset value.
- 1 = *Process Stack Pointer* (PSP).

On reset, the processor loads the MSP with the value from address 0x00000000.

11.4.1.6 Link Register

The *Link Register* (LR) is register R14. It stores the return information for subroutines, function calls, and exceptions. On reset, the processor loads the LR value 0xFFFFFFFF.

11.4.1.7 Program Counter

The *Program Counter* (PC) is register R15. It contains the current program address. On reset, the processor loads the PC with the value of the reset vector, which is at address 0x00000004. Bit[0] of the value is loaded into the EPSR T-bit at reset and must be 1.

11.4.3.5 Exception Priorities

As Table 11-9 shows, all exceptions have an associated priority, with:

- A lower priority value indicating a higher priority
- Configurable priorities for all exceptions except Reset, Hard fault and NMI.

If the software does not configure any priorities, then all exceptions with a configurable priority have a priority of 0. For information about configuring exception priorities see “System Handler Priority Registers” , and “Interrupt Priority Registers” .

Note: Configurable priority values are in the range 0–15. This means that the Reset, Hard fault, and NMI exceptions, with fixed negative priority values, always have higher priority than any other exception.

For example, assigning a higher priority value to IRQ[0] and a lower priority value to IRQ[1] means that IRQ[1] has higher priority than IRQ[0]. If both IRQ[1] and IRQ[0] are asserted, IRQ[1] is processed before IRQ[0].

If multiple pending exceptions have the same priority, the pending exception with the lowest exception number takes precedence. For example, if both IRQ[0] and IRQ[1] are pending and have the same priority, then IRQ[0] is processed before IRQ[1].

When the processor is executing an exception handler, the exception handler is preempted if a higher priority exception occurs. If an exception occurs with the same priority as the exception being handled, the handler is not preempted, irrespective of the exception number. However, the status of the new interrupt changes to pending.

11.4.3.6 Interrupt Priority Grouping

To increase priority control in systems with interrupts, the NVIC supports priority grouping. This divides each interrupt priority register entry into two fields:

- An upper field that defines the *group priority*
- A lower field that defines a *subpriority* within the group.

Only the group priority determines preemption of interrupt exceptions. When the processor is executing an interrupt exception handler, another interrupt with the same group priority as the interrupt being handled does not preempt the handler.

If multiple pending interrupts have the same group priority, the subpriority field determines the order in which they are processed. If multiple pending interrupts have the same group priority and subpriority, the interrupt with the lowest IRQ number is processed first.

For information about splitting the interrupt priority fields into group priority and subpriority, see “Application Interrupt and Reset Control Register” .

11.4.3.7 Exception Entry and Return

Descriptions of exception handling use the following terms:

Preemption

When the processor is executing an exception handler, an exception can preempt the exception handler if its priority is higher than the priority of the exception being handled. See “Interrupt Priority Grouping” for more information about preemption by an interrupt.

When one exception preempts another, the exceptions are called nested exceptions. See “Exception Entry” more information.

Return

This occurs when the exception handler is completed, and:

- There is no pending exception with sufficient priority to be serviced
- The completed exception handler was not handling a late-arriving exception.

11.6.6.3 SMLA and SMLAW

Signed Multiply Accumulate (halfwords).

Syntax

```
op{XY}{cond} Rd, Rn, Rm
op{Y}{cond} Rd, Rn, Rm, Ra
```

where:

op is one of:

SMLA Signed Multiply Accumulate Long (halfwords).

X and *Y* specifies which half of the source registers *Rn* and *Rm* are used as the first and second multiply operand.

If *X* is *B*, then the bottom halfword, bits [15:0], of *Rn* is used.
If *X* is *T*, then the top halfword, bits [31:16], of *Rn* is used.

If Y is *B*, then the bottom halfword, bits [15:0], of *Rm* is used.
If *Y* is *T*, then the top halfword, bits [31:16], of *Rm* is used

SMLAW Signed Multiply Accumulate (word by halfword).

Y specifies which half of the source register *Rm* is used as the second multiply operand.

If *Y* is T, then the top halfword, bits [31:16] of *Rm* is used.

If *Y* is B, then the bottom halfword, bits [15:0] of *Rm* is used.

cond is an optional condition code, see “Conditional Execution” .

Rd is the destination register. If *Rd* is omitted, the destination register is *Rn*.

Rn, Rm are registers holding the values to be multiplied.

Ra is a register holding the value to be added or subtracted from.

Operation

The SMALBB, SMLABT, SMLATB, SMLATT instructions:

- Multiplies the specified signed halfword, top or bottom, values from *Rn* and *Rm*.
- Adds the value in *Ra* to the resulting 32-bit product.
- Writes the result of the multiplication and addition in *Rd*.

The non-specified halfwords of the source registers are ignored.

The SMLAWB and SMLAWT instructions:

- Multiply the 32-bit signed values in *Rn* with:
  - The top signed halfword of *Rm*, *T* instruction suffix.
  - The bottom signed halfword of *Rm*, *B* instruction suffix.
- Add the 32-bit signed value in *Ra* to the top 32 bits of the 48-bit product
- Writes the result of the multiplication and addition in *Rd*.

The bottom 16 bits of the 48-bit product are ignored.

If overflow occurs during the addition of the accumulate value, the instruction sets the Q flag in the APSR. No overflow can occur during the multiplication.

Restrictions

In these instructions, do not use SP and do not use PC.

Condition Flags

If an overflow is detected, the `Q` flag is set.

Table 11-27. Floating-point Instructions (Continued)

Mnemonic	Description
VPUSH	Push extension registers
VSQRT	Floating-point square root
VSTM	Store Multiple extension registers
VSTR	Stores an extension register to memory
VSUB	Floating-point Subtract

11.9.1.16 MemManage Fault Address Register

Name: SCB_MMFAR

Access: Read/Write

31	30	29	28	27	26	25	24
ADDRESS							

23	22	21	20	19	18	17	16
ADDRESS							

15	14	13	12	11	10	9	8
ADDRESS							

7	6	5	4	3	2	1	0
ADDRESS							

The SCB_MMFAR contains the address of the location that generated a memory management fault.

- **ADDRESS: Memory Management Fault Generation Location Address**

When the MMARVALID bit of the MMFSR subregister is set to 1, this field holds the address of the location that generated the memory management fault.

Notes:
1. When an unaligned access faults, the address is the actual address that faulted. Because a single read or write instruction can be split into multiple aligned accesses, the fault address can be any address in the range of the requested access size.
2. Flags in the MMFSR subregister indicate the cause of the fault, and whether the value in the SCB_MMFAR is valid. See “MMFSR: Memory Management Fault Status Subregister” .

The SAMPLE, EXTEST and BYPASS functions are implemented. In SWD/JTAG debug mode, the ARM processor responds with a non-JTAG chip ID that identifies the processor. This is not IEEE 1149.1 JTAG-compliant.

It is not possible to switch directly between JTAG Boundary Scan and SWJ Debug Port operations. A chip reset must be performed after JTAGSEL is changed.

A Boundary-scan Descriptor Language (BSDL) file to set up the test is provided on www.atmel.com.

12.6.9.1 JTAG Boundary-scan Register

The Boundary-scan Register (BSR) contains a number of bits which correspond to active pins and associated control signals.

Each SAM4 input/output pin corresponds to a 3-bit register in the BSR. The OUTPUT bit contains data that can be forced on the pad. The INPUT bit facilitates the observability of data applied to the pad. The CONTROL bit selects the direction of the pad.

For more information, please refer to BDSL files available for the SAM4 Series.

15.5.7 RTC Accurate Clock Calibration

The crystal oscillator that drives the RTC may not be as accurate as expected mainly due to temperature variation. The RTC is equipped with circuitry able to correct slow clock crystal drift.

To compensate for possible temperature variations over time, this accurate clock calibration circuitry can be programmed on-the-fly and also programmed during application manufacturing, in order to correct the crystal frequency accuracy at room temperature (20–25°C). The typical clock drift range at room temperature is ±20 ppm.

In the device operating temperature range, the 32.768 kHz crystal oscillator clock inaccuracy can be up to -200 ppm.

The RTC clock calibration circuitry allows positive or negative correction in a range of 1.5 ppm to 1950 ppm.

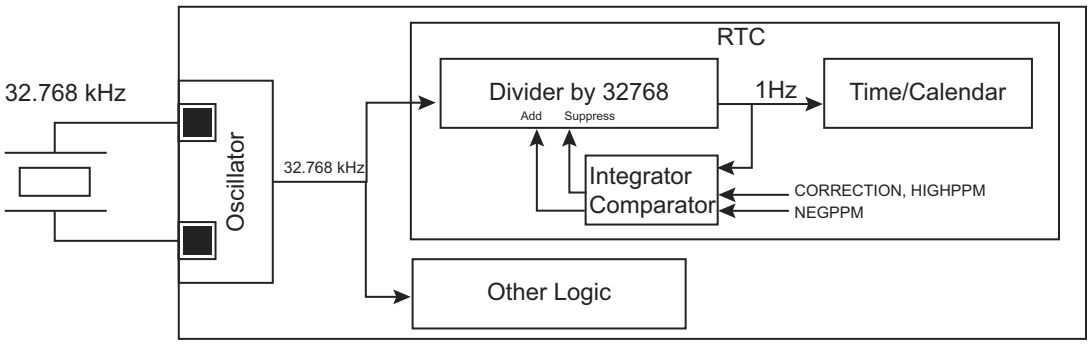
The calibration circuitry is fully digital. Thus, the configured correction is independent of temperature, voltage, process, etc., and no additional measurement is required to check that the correction is effective.

If the correction value configured in the calibration circuitry results from an accurate crystal frequency measure, the remaining accuracy is bounded by the values listed below:

- Below 1 ppm, for an initial crystal drift between 1.5 ppm up to 20 ppm, and from 30 ppm to 90 ppm
- Below 2 ppm, for an initial crystal drift between 20 ppm up to 30 ppm, and from 90 ppm to 130 ppm
- Below 5 ppm, for an initial crystal drift between 130 ppm up to 200 ppm

The calibration circuitry does not modify the 32.768 kHz crystal oscillator clock frequency but it acts by slightly modifying the 1 Hz clock period from time to time. The correction event occurs every 1 + [(20 - (19 x HIGHPPM)) x CORRECTION] seconds. When the period is modified, depending on the sign of the correction, the 1 Hz clock period increases or reduces by around 4 ms. Depending on the CORRECTION, NEGPPM and HIGHPPM values configured in RTC_MR, the period interval between two correction events differs.

Figure 15-4. Calibration Circuitry



- **LCKDOWN: Lockdown Supported**

0: Lockdown is not supported.

1: Lockdown is supported.

- **CSIZE: Data Cache Size**

Value	Name	Description
0	CSIZE_1KB	Data cache size is 1 Kbyte
1	CSIZE_2KB	Data cache size is 2 Kbytes
2	CSIZE_4KB	Data cache size is 4 Kbytes
3	CSIZE_8KB	Data cache size is 8 Kbytes

- **CLSIZE: Cache LIne Size**

Value	Name	Description
0	CLSIZE_1KB	Cache line size is 4 bytes
1	CLSIZE_2KB	Cache line size is 8 bytes
2	CLSIZE_4KB	Cache line size is 16 bytes
3	CLSIZE_8KB	Cache line size is 32 bytes

29.18.26 PLL Maximum Multiplier Value Register

Name: PMC_PMMR

Address: 0x400E0530

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
–	–	–	–	–	PLLA_MMAX		
7	6	5	4	3	2	1	0
PLLA_MMAX							

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

- **PLLA_MMAX: PLLA Maximum Allowed Multiplier Value**

Defines the maximum value of multiplication factor that can be sent to PLLA. Any value of the MULA field (see “PMC Clock Generator PLLA Register”) above PLLA_MMAX is saturated to PLLA_MMAX. PLLA_MMAX write operation is cancelled in the following cases:

- The value of MULA is currently saturated by PLLA_MMAX
- The user is trying to write a value of PLLA_MMAX that is smaller than the current value of MULA

- **TXBUFE: TX Buffer Empty (cleared by writing SPI_TCR or SPI_TNCR)**

0: SPI_TCR(1) or SPI_TNCR(1) has a value other than 0.

1: Both SPI_TCR(1) and SPI_TNCR(1) have a value of 0.

- **NSSR: NSS Rising (cleared on read)**

0: No rising edge detected on NSS pin since the last read of SPI_SR.

1: A rising edge occurred on NSS pin since the last read of SPI_SR.

- **TXEMPTY: Transmission Registers Empty (cleared by writing SPI_TDR)**

0: As soon as data is written in SPI_TDR.

1: SPI_TDR and internal shift register are empty. If a transfer delay has been defined, TXEMPTY is set after the end of this delay.

- **UNDES: Underrun Error Status (Slave mode only) (cleared on read)**

0: No underrun has been detected since the last read of SPI_SR.

1: A transfer starts whereas no data has been loaded in SPI_TDR.

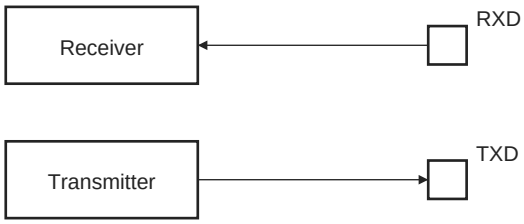
- **SPIENS: SPI Enable Status**

0: SPI is disabled.

1: SPI is enabled.

Note: 1. SPI_RCR, SPI_RNCR, SPI_TCR, SPI_TNCR are PDC registers.

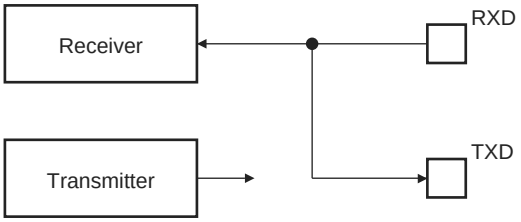
Figure 37-39. Normal Mode Configuration



37.6.9.2 Automatic Echo Mode

Automatic Echo mode allows bit-by-bit retransmission. When a bit is received on the RXD pin, it is sent to the TXD pin, as shown in Figure 37-40. Programming the transmitter has no effect on the TXD pin. The RXD pin is still connected to the receiver input, thus the receiver remains active.

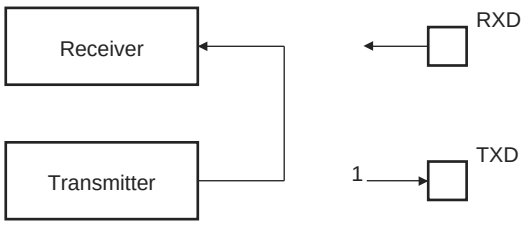
Figure 37-40. Automatic Echo Mode Configuration



37.6.9.3 Local Loopback Mode

Local Loopback mode connects the output of the transmitter directly to the input of the receiver, as shown in Figure 37-41. The TXD and RXD pins are not used. The RXD pin has no effect on the receiver and the TXD pin is continuously driven high, as in idle state.

Figure 37-41. Local Loopback Mode Configuration



37.6.9.4 Remote Loopback Mode

Remote Loopback mode directly connects the RXD pin to the TXD pin, as shown in Figure 37-42. The transmitter and the receiver are disabled and have no effect. This mode allows bit-by-bit retransmission.

Figure 37-42. Remote Loopback Mode Configuration

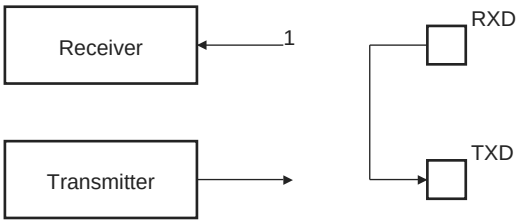
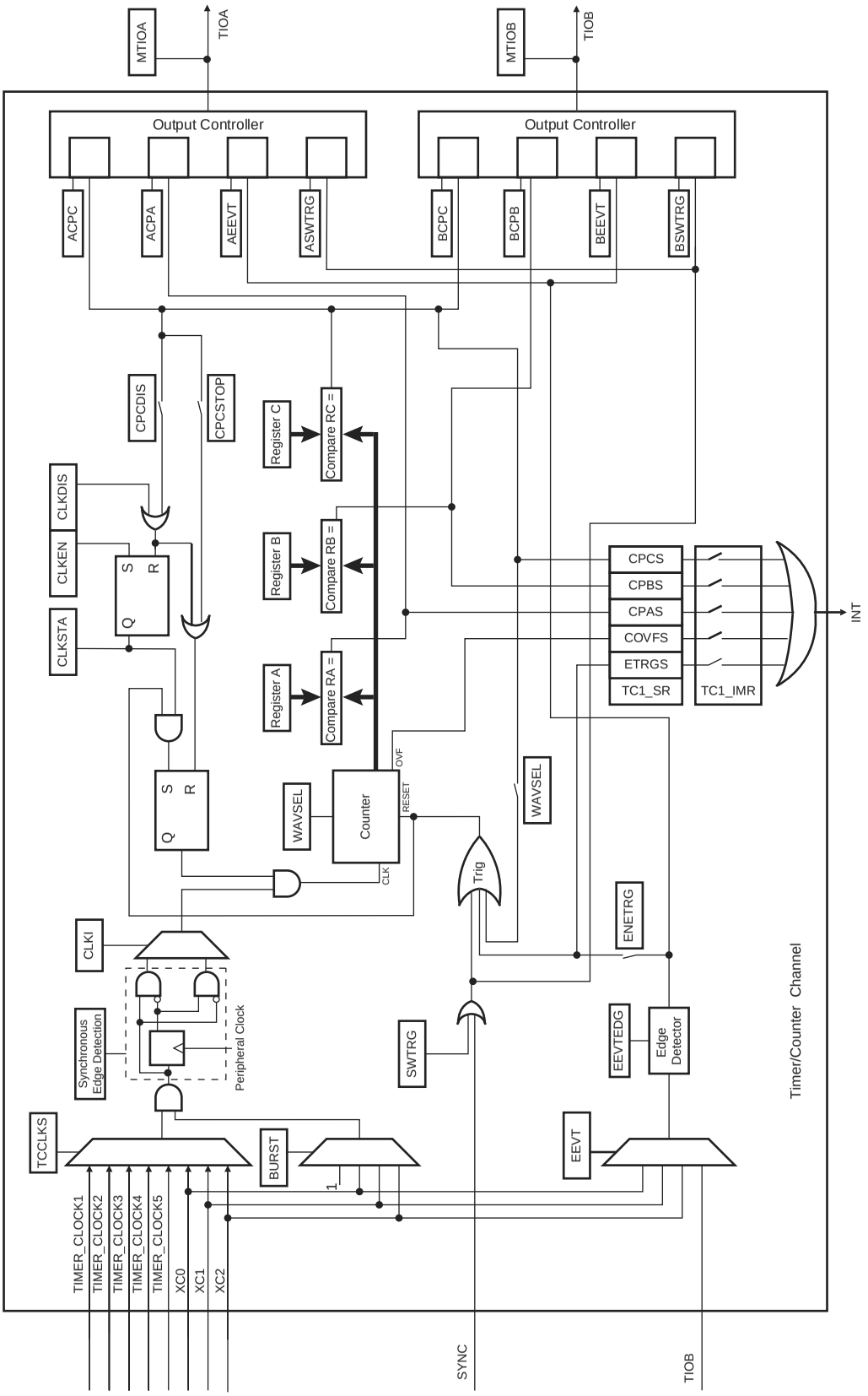


Figure 38-7. Waveform Mode



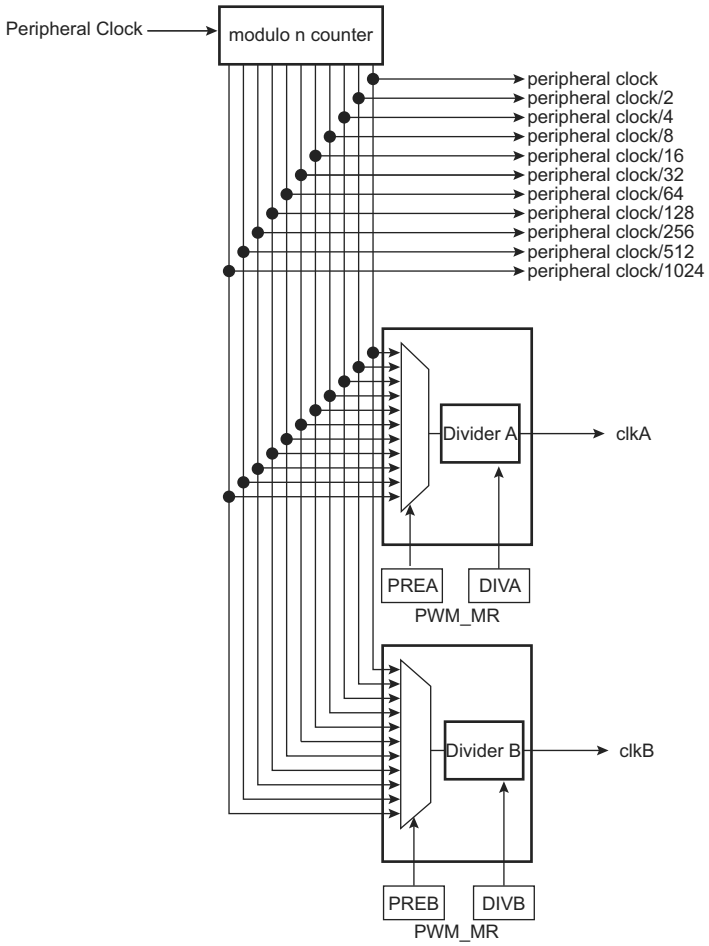
39.6 Functional Description

The PWM controller is primarily composed of a clock generator module and 4 channels.

- Clocked by the peripheral clock, the clock generator module provides 13 clocks.
- Each channel can independently choose one of the clock generator outputs.
- Each channel generates an output waveform with attributes that can be defined independently for each channel through the user interface registers.

39.6.1 PWM Clock Generator

Figure 39-2. Functional View of the Clock Generator Block Diagram



The PWM peripheral clock is divided in the clock generator module to provide different clocks available for all channels. Each channel can independently select one of the divided clocks.

The clock generator is divided into different blocks:

- a modulo n counter which provides 11 clocks: $f_{peripheral\ clock}$, $f_{peripheral\ clock}/2$, $f_{peripheral\ clock}/4$, $f_{peripheral\ clock}/8$, $f_{peripheral\ clock}/16$, $f_{peripheral\ clock}/32$, $f_{peripheral\ clock}/64$, $f_{peripheral\ clock}/128$, $f_{peripheral\ clock}/256$, $f_{peripheral\ clock}/512$, $f_{peripheral\ clock}/1024$
- two linear dividers (1, 1/2, 1/3, ... 1/255) that provide two separate clocks: clkA and clkB

Each linear divider can independently divide one of the clocks of the modulo n counter. The selection of the clock to be divided is made according to the PREA (PREB) field of the PWM Clock register (PWM_CLK). The resulting clock clkA (clkB) is the clock selected divided by DIVA (DIVB) field value.

39.7.23 PWM Output Selection Clear Update Register

Name: PWM_OSCUPD

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	OSCUPL3	OSCUPL2	OSCUPL1	OSCUPL0
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	–	OSCUPH3	OSCUPH2	OSCUPH1	OSCUPH0

- **OSCUPHx: Output Selection Clear for PWMH output of the channel x**

0: No effect.

1: Dead-time generator output DTOHx selected as PWMH output of channel x at the beginning of the next channel x PWM period.

- **OSCUPLx: Output Selection Clear for PWML output of the channel x**

0: No effect.

1: Dead-time generator output DTOLx selected as PWML output of channel x at the beginning of the next channel x PWM period.

42.8.27 GMAC Type ID Match 1 Register

Name: GMAC_TIDM1

Address: 0x400340A8

Access: Read/Write

31	30	29	28	27	26	25	24
ENID1	–	–	–	–	–	–	–

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

15	14	13	12	11	10	9	8
TID							

7	6	5	4	3	2	1	0
TID							

- **TID: Type ID Match 1**

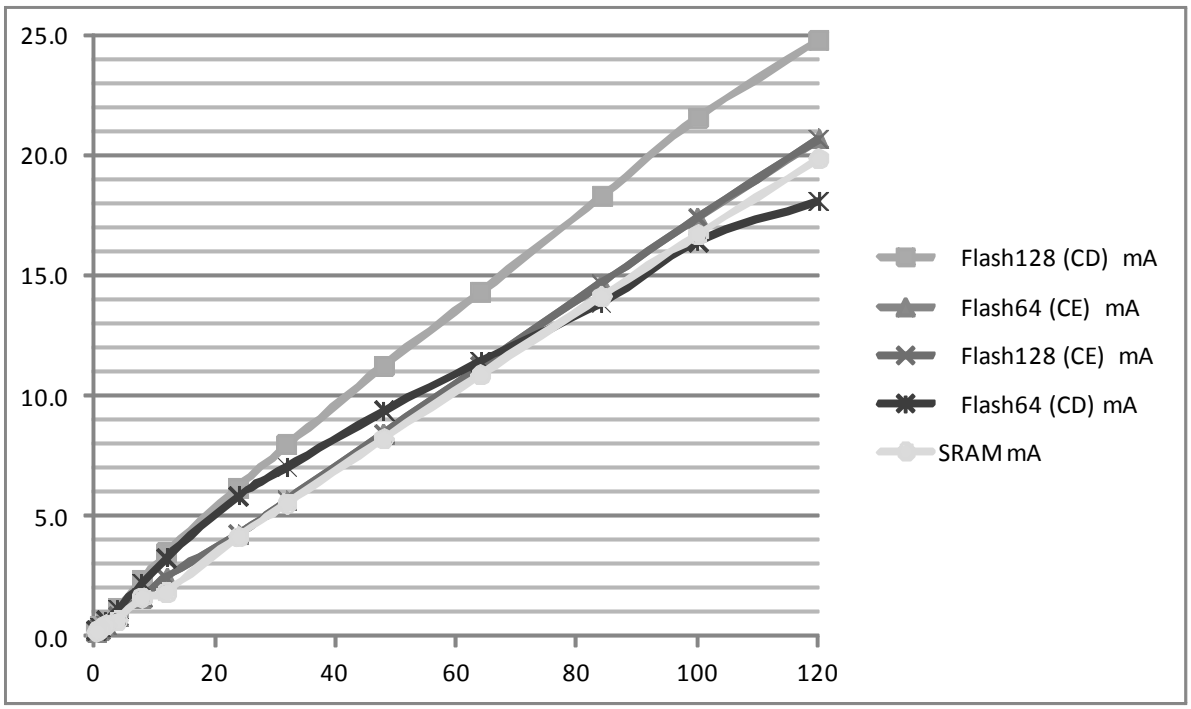
For use in comparisons with received frames type ID/length frames.

- **ENID1: Enable Copying of TID Matched Frames**

0: TID is not part of the comparison match.

1: TID is processed for the comparison match.

Figure 46-9. Active Power Consumption with VDDCORE @ 1.2V



- VDDCORE at 1.2V
- T_A = 25°C

46.3.3.2 SAM4E Active Total Power Consumption

Table 46-14. Active Total Power Consumption with VDDCORE @ 1.2V running from Embedded Memory (IDDIO + IDDIN - AMP2)

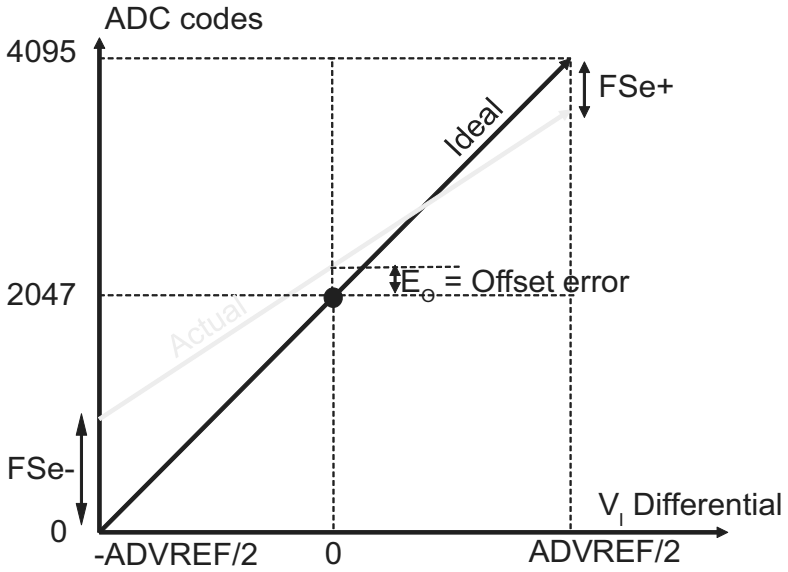
Core Clock (MHz)	CoreMark					Unit
	Cache Enable (CE)		Cache Disable (CD)		SRAM	
	128-bit Flash Access(1)	64-bit Flash Access(1)	128-bit Flash Access(1)	64-bit Flash Access(1)		
120	22.6	22.6	29.2	22.3	19.5	mA
100	19.5	19.5	25.7	20.2	16.4	
84	17.7	17.8	24.0	19.9	15.1	
64	13.6	13.6	19.7	16.7	11.5	
48	10.3	10.3	14.7	14.4	8.7	
32	7.9	7.9	12.1	11.9	6.8	
24	5.8	5.8	9.3	9.2	5.2	
12	3.8	3.6	6.3	6.2	3.4	
8	3.5	3.4	5.5	5.6	3.3	
4	2.8	2.7	4.1	4.4	2.7	
2	2.5	2.4	3.6	3.7	2.4	
1	1.5	1.5	1.9	2.1	1.5	
0.5	1.4	1.4	1.6	1.7	1.4	

Note: 1. Flash Wait State (FWS) in EEFC_FMR adjusted depending on Core Frequency

Differential Mode

In differential mode, the offset is defined when the differential input voltage is zero.

Figure 46-16. Gain and Offset Errors in Differential Mode



where:

- FSe = (FSe+) - (FSe-) is for full-scale error, unit is LSB code
- Offset error E_O is the offset error measured for V_I = 0V
- Gain error $E_G = 100 \times FSe / 4096$, unit in %

The error values in Table 46-35 and Table 46-36 include the sample and hold error as well as the PGA gain error.

Table 46-35. Differential Gain Error E_G

Gain Mode	0.5		1		2	
Auto Correction	**No**	**Yes**	**No**	**Yes**	**No**	**Yes**
Average Gain Error (%)	-0.107	0.005	0.444	0.112	0.713	0.005
Standard Deviation (%)	0.410	0.210	0.405	0.229	0.400	0.317
Gain Min Value (%)	-1.338	-0.625	-0.771	-0.576	-0.488	-0.947
Gain Max Value (%)	1.123	0.635	1.660	0.801	1.914	0.957

Table 46-36. Differential Output Offset Error E_O

Gain	0.5	1	2
Average Offset Error (LSB)	-1.2	-1.2	-0.6
Standard Deviation (LSB)	0.3	0.4	0.4
Offset Min value (LSB)	-2.1	-2.4	-1.8
Offset Max value (LSB)	-0.3	0	0.6

46.11.5 SMC Timings

Timings are given in the following domains:

- 1.8V domain: V_{DDIO} from 1.65V to 1.95V, maximum external capacitor = 30 pF
- 3.3V domain: V_{DDIO} from 2.85V to 3.6V, maximum external capacitor = 50 pF

Timings are given assuming a capacitance load on data, control and address pads.

In the tables that follow, t_{CPMCK} is MCK period.

46.11.5.1 Read Timings

Table 46-58. SMC Read Signals - NRD Controlled (READ_MODE = 1)

	VDDIO Supply	1.8V Domain	3.3V Domain	1.8V Domain	3.3V Domain	
Symbol	**Parameter**	**Min**		**Max**		**Unit**
NO HOLD Settings (NRD_HOLD = 0)						
SMC_1	Data Setup before NRD High	20.6	18.6	—	—	ns
SMC_2	Data Hold after NRD High	0	0	—	—	ns
HOLD Settings (NRD_HOLD ≠ 0)						
SMC_3	Data Setup before NRD High	15.9	14.1	—	—	ns
SMC_4	Data Hold after NRD High	0	0	—	—	ns
HOLD or NO HOLD Settings (NRD_HOLD ≠ 0, NRD_HOLD = 0)						
SMC_5	A0–A22 Valid before NRD High	(NRD_SETUP + NRD_PULSE) × t_{CPMCK} - 6.8	(NRD_SETUP + NRD_PULSE) × t_{CPMCK} - 6.5	—	—	ns
SMC_6	NCS low before NRD High	(NRD_SETUP + NRD_PULSE - NCS_RD_SETUP) × t_{CPMCK} - 4.6	(NRD_SETUP + NRD_PULSE - NCS_RD_SETUP) × t_{CPMCK} - 5.1	—	—	ns
SMC_7	NRD Pulse Width	NRD_PULSE × t_{CPMCK} - 7.5	NRD_PULSE × t_{CPMCK} - 6.6	—	—	ns

Table 46-59. SMC Read Signals - NCS Controlled (READ_MODE = 0)

	VDDIO Supply	1.8V Domain	3.3V Domain	1.8V Domain	3.3V Domain	
Symbol	**Parameter**	**Min**		**Max**		**Unit**
NO HOLD Settings (NCS_RD_HOLD = 0)						
SMC_8	Data Setup before NCS High	21.8	19.4	—	—	ns
SMC_9	Data Hold after NCS High	0	0	—	—	ns
HOLD Settings (NCS_RD_HOLD ≠ 0)						
SMC_{10}	Data Setup before NCS High	17.1	14.9	—	—	ns
SMC_{11}	Data Hold after NCS High	0	0	—	—	ns
HOLD or NO HOLD Settings (NCS_RD_HOLD ≠ 0, NCS_RD_HOLD = 0)						
SMC_{12}	A0–A22 valid before NCS High	(NCS_RD_SETUP + NCS_RD_PULSE) × t_{CPMCK} - 6.9	(NCS_RD_SETUP + NCS_RD_PULSE) × t_{CPMCK} - 6.6	—	—	ns
SMC_{13}	NRD low before NCS High	(NCS_RD_SETUP + NCS_RD_PULSE - NRD_SETUP) × t_{CPMCK} - 5.8	(NCS_RD_SETUP + NCS_RD_PULSE - NRD_SETUP) × t_{CPMCK} - 5.5	—	—	ns

Table 51-5. SAM4E Datasheet Rev. 11157D 12-Jun-14 Revision history

Doc. Date	Changes
12-Jun-2014	Modified Title of the document (SAM4E Series)
	Changed structure of the document (order of sections: GMAC, DAC...)
	Ethernet MAC (EMAC) replaced with Ethernet MAC (GMAC) and EMAC signals replaced with GMAC signals throughout the document (example: GTXCK instead of ETXCK, etc.).
	Section 1. “Features” Added tamper detection Modified note 1 (removed “or using internal voltage regulator”) Table 1-1 “Configuration Summary”: Modified information on Timer channels
	Section 2-1 “SAM4E 144-pin Block Diagram” Updated Figure 2-1 “SAM4E 144-pin Block Diagram” and Figure 2-2 “SAM4E 144-pin Block Diagram” (Tamper detection added ; AFE block ; WKUP pins) Timer Counter B and C added in Figure 2-1 “SAM4E 144-pin Block Diagram”
	Section 3. “Signal Description” Modified Table 3-1 “Signal Description List” (“DATRG” instead of “DACTRG”, WKUP[15:0] added, FFPI signals modified)
	Section 4. “Package and Pinout” Modified Table 4-1 “SAM4E 100-ball TFBGA Pinout”, Table 4-2 “SAM4E 144-ball LFBGA Pinout”, Table 4-3 “SAM4E 100-lead LQFP Pinout” and Table 4-4 “SAM4E 144-lead LQFP Pinout”
	Section 5. “Power Considerations” Modified notes after Figure 5-2 “Single Supply” and Figure 5-3 “Core Externally Supplied”: 2.0V replaced with 2.4V (VDDIN minimum value for FAE)
	Section 7.2 “Embedded Memories” Modified Section 7.2.3.1 “Flash Overview” (paragraph below Section 7-4 “Flash Size”) Updated information on the ERASE pin in Section 7.2.3.5 “Security Bit Feature”
	Section 10. “Peripherals” Removed note 1 in Table 10-2 “Multiplexing on PIO Controller A (PIOA)”, Table 10-3 “Multiplexing on PIO Controller B (PIOB)” and Table 10-4 “Multiplexing on PIO Controller C (PIOC)” Removed reset values for Write-only registers
	Section 11. “ARM Cortex-M4 Processor” Minor formatting and editorial changes throughout Updated 2nd instruction line, in Section 11.5.3 “Power Management Programming Hints” Section 11.9.1.2 “CPUID Base Register”: updated ‘Constant’ field description Section 11.9.1.5 “Application Interrupt and Reset Control Register”: updated ‘VECTCLRACTIVE’ and ‘VECTRESET’ field descriptions Section 11.9.1.7 “Configuration and Control Register”: updated ‘USERSETMPEND’ field description Section 11.9.1.16 “MemManage Fault Address Register”: updated ‘ADDRESS’ field description Section 11.9.1.16 “MemManage Fault Address Register”: updated ‘ADDRESS’ field description Section 11.10.1.1 “SysTick Control and Status”: updated ‘TICKINT’ and ‘ENABLE’ field descriptions Section 11.10.1.2 “SysTick Reload Value Registers””: updated ‘RELOAD’ field description Section 11.10.1.3 “SysTick Current Value Register”: updated ‘CURRENT' field description