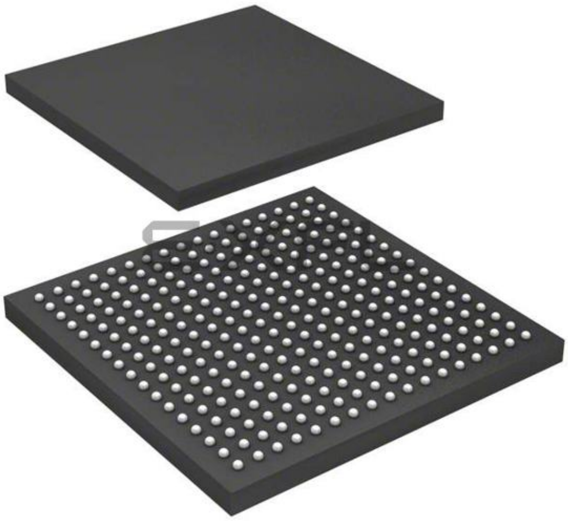
Microchip Technology - ATSAMA5D27B-CN Datasheet



Welcome to E-XFL.COM

Understanding Embedded - Microprocessors

Embedded microprocessors are specialized computing chips designed to perform specific tasks within an embedded system. Unlike general-purpose microprocessors found in personal computers, embedded microprocessors are tailored for dedicated functions within larger systems, offering optimized performance, efficiency, and reliability. These microprocessors are integral to the operation of countless electronic devices, providing the computational power necessary for controlling processes, handling data, and managing communications.

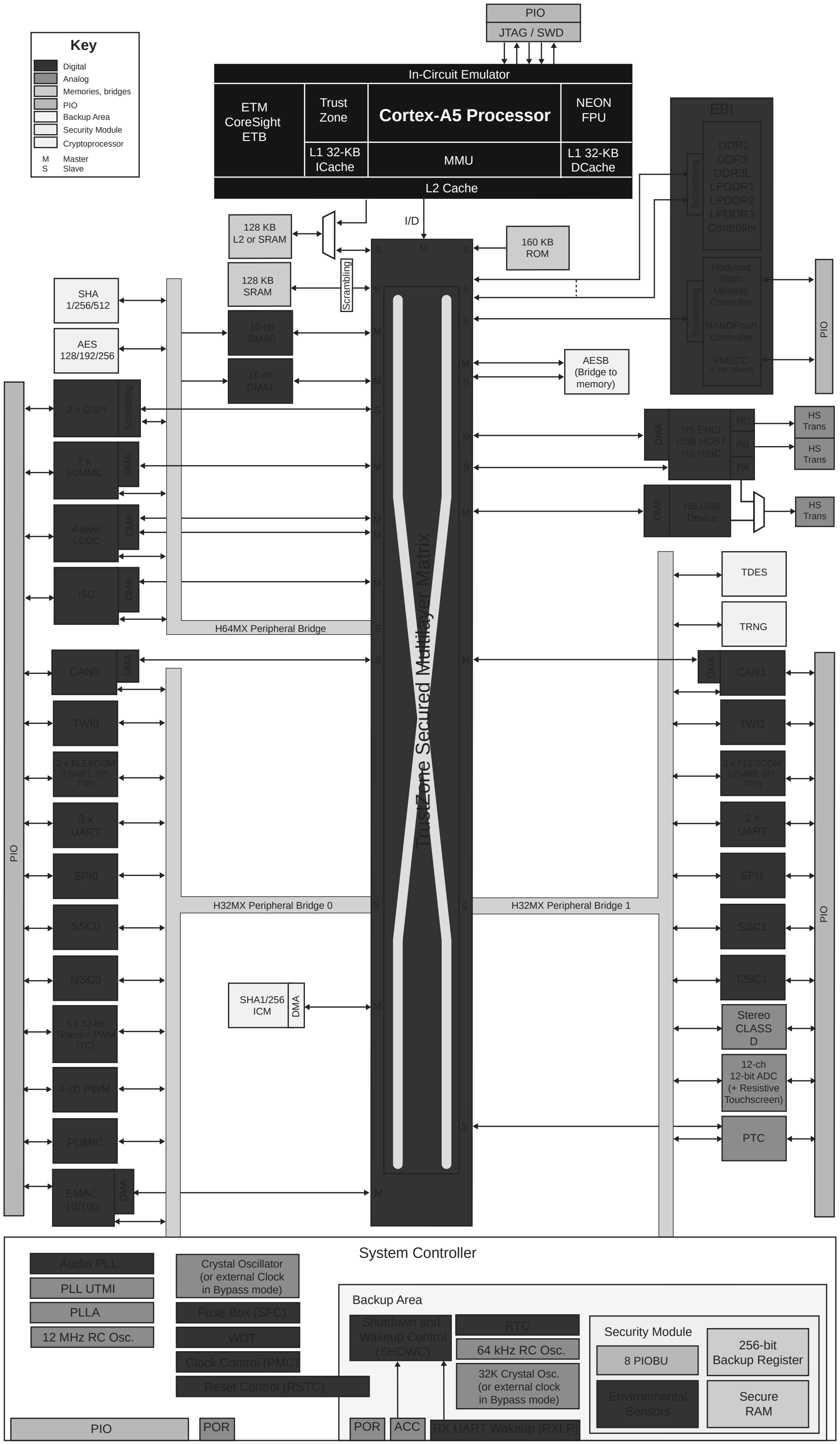
Applications of Embedded - Microprocessors

Embedded microprocessors are utilized across a broad spectrum of applications, making them indispensable in

Details	
Product Status	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 ~ 105°C (TA)
Security Features	ARM TZ, Boot Security, Cryptography, RTIC, Secure Fusebox, Secure JTAG, Secure Memory, Secure RTC
Package / Case	289-LFBGA
Supplier Device Package	289-LFBGA (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsama5d27b-cn

3. Block Diagram

Figure 3-1: SAMA5D2 Series Block Diagram



Note: Refer to Section 38. “DMA Controller (XDMAC)” for peripheral connections to DMA.

Figure 16-5: NVM Boot Diagram

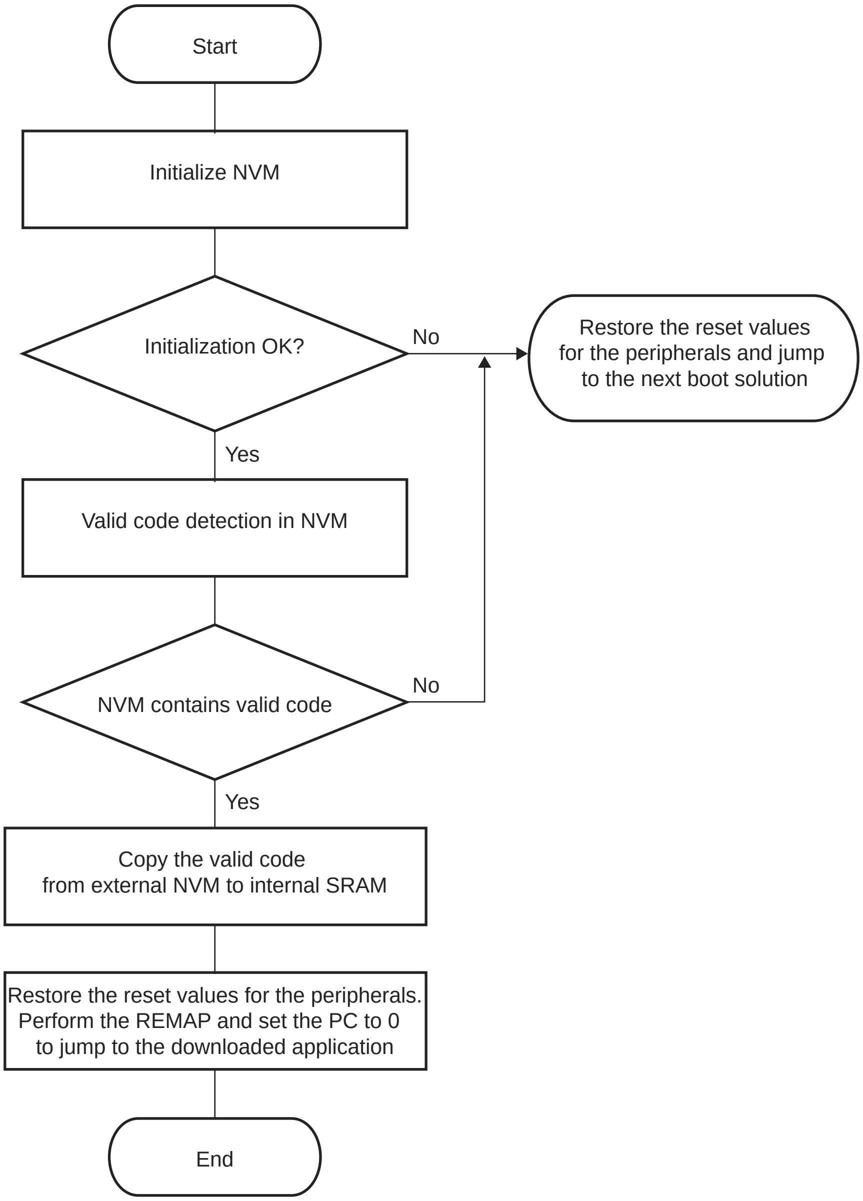

Start
Initialize NVM
Initialization OK?
No
Restore the reset values for the peripherals and jump to the next boot solution
Yes
Valid code detection in NVM
NVM contains valid code
No
Yes
Copy the valid code from external NVM to internal SRAM
Restore the reset values for the peripherals. Perform the REMAP and set the PC to 0 to jump to the downloaded application
End

Figure 18-3 shows an External Securable slave example. This example is constructed with the following hypothesis:

- The slave is an interface with the external bus (EBI) containing two regions. The slave size is 2 × 256 Mbytes. Each slave region Max Size is 256 Mbytes.
- The slave region 0 base address equals 0x10000000. It is connected to a 32 Mbyte memory, for example an external DDR. The slave region 0 Top Size must be set to 32 Mbytes.
- The slave region 1 base address equals 0x20000000. It is connected to a 2 Mbyte memory, for example an external NAND Flash. The slave region 1 Top Size must be set to 2 Mbytes.
- The slave software configuration is:
  - TOP0 is set to 32 Mbytes
  - TOP1 is set to 2 Mbytes
  - SPLIT0 is set to 4 Mbytes
  - SPLIT1 is set to 1 Mbyte
  - LANSECH0 is set to 1, the low area of region 0 is the non-securable one
  - RDNSECH0 is set to 0, region 0 Securable area is secured for reads
  - WRNSECH0 is set to 0, region 0 Securable area is secured for writes
  - LANSECH1 is set to 0, the low area of region 1 is the Securable one
  - RDNSECH1 is set to 1, region 1 Securable area is non-secured for reads
  - WRNSECH1 is set to 0, region 1 Securable area is secured for writes

29.7.3 PTC Enable Register

Name: PTC_IED

Access: Write-only

7	6	5	4	3	2	1	0
IER3	IER2	IER1	IER0	–	–	–	–

IERx: Interrupt Enable

Enables interrupt for device-to-host interrupt.

Writing a zero to this field has no effect.

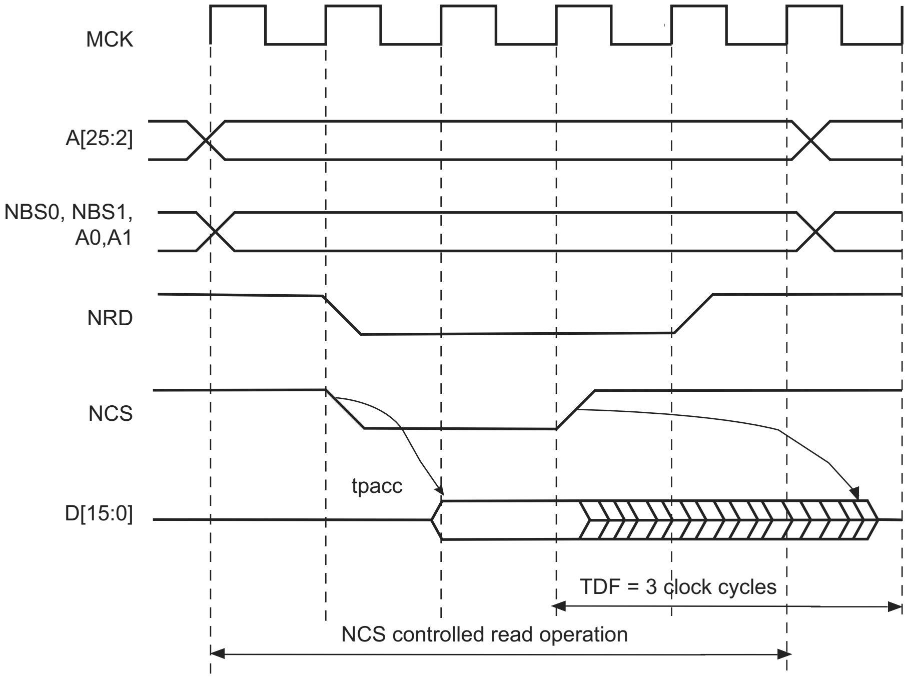
PLLADIV2: PLLA Divisor by 2

Bit PLLADIV2 must always be set to 1 when MDIV is set to 3.

H32MXDIV: AHB 32-bit Matrix Divisor

Value	Name	Description
0	H32MXDIV1	The AHB 32-bit Matrix frequency is equal to the AHB 64-bit Matrix frequency. It is possible only if the AHB 64-bit Matrix frequency does not exceed 83 MHz.
1	H32MXDIV2	The AHB 32-bit Matrix frequency is equal to the AHB 64-bit Matrix frequency divided by 2.

Figure 37-18: TDF Period in NCS Controlled Read Operation (TDF = 3)



37.13.2 TDF Optimization Enabled (TDF_MODE = 1)

When the TDF_MODE of the HSMC_MODE register is set to 1 (TDF optimization is enabled), the SMC takes advantage of the setup period of the next access to optimize the number of wait states cycle to insert.

Figure 37-19 shows a read access controlled by NRD, followed by a write access controlled by NWE, on Chip Select 0. Chip Select 0 has been programmed with:

NRD_HOLD = 4; READ_MODE = 1 (NRD controlled)

NWE_SETUP = 3; WRITE_MODE = 1 (NWE controlled)

TDF_CYCLES = 6; TDF_MODE = 1 (optimization enabled).

39.7.8 LCD Controller Enable Register

Name: LCDC_LCDEN

Address: 0xF0000020

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
–	–	–	–	PWMEN	DISPEN	SYNCEN	CLKEN

CLKEN: LCD Controller Pixel Clock Enable

0: No effect

1: Pixel clock logical unit is activated.

SYNCEN: LCD Controller Horizontal and Vertical Synchronization Enable

0: No effect

1: Both horizontal and vertical synchronization (LCDVSYNC and LCDHSYNC) signals are generated.

DISPEN: LCD Controller DISP Signal Enable

0: No effect

1: LCDDISP signal is generated.

PWMEN: LCD Controller Pulse Width Modulation Enable

0: No effect

1: PWM is enabled.

39.7.147 Post Processing Next Register

Name: LCDC_PPNEXT

Address: 0xF0000568

Access: Read/Write

31	30	29	28	27	26	25	24
NEXT							

23	22	21	20	19	18	17	16
NEXT							

15	14	13	12	11	10	9	8
NEXT							

7	6	5	4	3	2	1	0
NEXT							

NEXT: DMA Descriptor Next Address

The transfer descriptor address must be aligned on a 64-bit boundary.

39.7.150 Post Processing Configuration Register 2

Name: LCDC_PPCFG2

Address: 0xF0000574

Access: Read/Write

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">XSTRIDE</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">XSTRIDE</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">XSTRIDE</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">XSTRIDE</td></tr>
</table>

XSTRIDE: Horizontal Stride

XSTRIDE represents the memory offset, in bytes, between two rows of the image memory.

39.7.156 Overlay 2 CLUT Register x

Name: LCDC_OVR2CLUTx [x=0..255]

Address: 0xF0000E00

Access: Read/Write

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

BCLUT: Blue Color Entry

This field indicates the 8-bit width Blue color of the color lookup table.

GCLUT: Green Color Entry

This field indicates the 8-bit width Green color of the color lookup table.

RCLUT: Red Color Entry

This field indicates the 8-bit width Red color of the color lookup table.

ACLUT: Alpha Color Entry

This field indicates the 8-bit width Alpha channel of the color lookup table.

- If bit 12 is written with a one, the pause quantum will be zero.

After transmission, a pause frame transmitted interrupt will be generated (bit 14 of the Interrupt Status register) and the only the statistics register Pause Frames Transmitted is incremented.

Pause frames can also be transmitted by the MAC using normal frame transmission methods.

40.6.17 MAC PFC Priority-based Pause Frame Support

Note: Refer to the 802.1Qbb standard for a full description of priority-based pause operation.

The following table shows the start of a Priority-based Flow Control (PFC) pause frame.

Table 40-17: Start of a PFC Pause Frame

Address		Type (Mac Control Frame)	Pause Opcode	Priority Enable Vector	Pause Time
Destination	Source				
0x0180C2000001	6 bytes	0x8808	0x1001	2 bytes	8 × 2 bytes

The GMAC supports PFC priority-based pause transmission and reception. Before PFC pause frames can be received, bit 16 of the Network Control register must be set.

40.6.17.1 PFC Pause Frame Reception

The ability to receive and decode priority-based pause frames is enabled by setting bit 16 of the Network Control register. When this bit is set, the GMAC will match either classic 802.3 pause frames or PFC priority-based pause frames. Once a priority-based pause frame has been received and matched, then from that moment on the GMAC will only match on priority-based pause frames (this is an 802.1Qbb requirement, known as PFC negotiation). Once priority-based pause has been negotiated, any received 802.3x format pause frames will not be acted upon.

If a valid priority-based pause frame is received then the GMAC will decode the frame and determine which, if any, of the eight priorities require to be paused. Up to eight Pause Time registers are then updated with the eight pause times extracted from the frame regardless of whether a previous pause operation is active or not. An interrupt (either bit 12 or bit 13 of the Interrupt Status register) is triggered when a pause frame is received, but only if the interrupt has been enabled (bit 12 and bit 13 of the Interrupt Mask register). Pause frames received with non zero quantum are indicated through the interrupt bit 12 of the Interrupt Status register. Pause frames received with zero quantum are indicated on bit 13 of the Interrupt Status register. The loading of a new pause time only occurs when the GMAC is configured for full duplex operation. If the GMAC is configured for half duplex, the pause time counters will not be loaded, but the pause frame received interrupt will still be triggered. A valid pause frame is defined as having a destination address that matches either the address stored in Specific Address 1 register or if it matches the reserved address of 0x0180C2000001. It must also have the MAC control frame type ID of 0x8808 and have the pause opcode of 0x0101.

Pause frames that have frame check sequence (FCS) or other errors will be treated as invalid and will be discarded. Valid pause frames received will increment the Pause Frames Received Statistic register.

The Pause Time registers decrement every 512 bit times immediately following the PFC frame reception. For test purposes, the retry test bit can be set (bit 12 in the Network Configuration register) which causes the Pause Time register to decrement every GRXCK cycle once transmission has stopped.

The interrupt (bit 13 in the Interrupt Status register) is asserted whenever the Pause Time register decrements to zero (assuming it has been enabled by bit 13 in the Interrupt Mask register). This interrupt is also set when a zero quantum pause frame is received.

40.6.17.2 PFC Pause Frame Transmission

Automatic transmission of pause frames is supported through the transmit priority-based pause frame bit of the Network Control register. If bit 17 of the Network Control register is written with logic 1, a PFC pause frame will be transmitted providing full duplex is selected in the Network Configuration register and the transmit block is enabled in the Network Control register. When bit 17 of the Network Control register is set, the fields of the priority-based pause frame will be built using the values stored in the Transmit PFC Pause register.

Pause frame transmission will happen immediately if transmit is inactive or if transmit is active between the current frame and the next frame due to be transmitted.

Transmitted pause frames comprise the following:

- A destination address of 01-80-C2-00-00-01
- A source address taken from Specific Address 1 register
- A type ID of 88-08 (MAC control frame)
- A pause opcode of 01-01

40.8.59 GMAC Greater Than 1518 Byte Frames Transmitted Register

Name: GMAC_GTBFT1518

Address: 0xF8008130

Access: Read-only

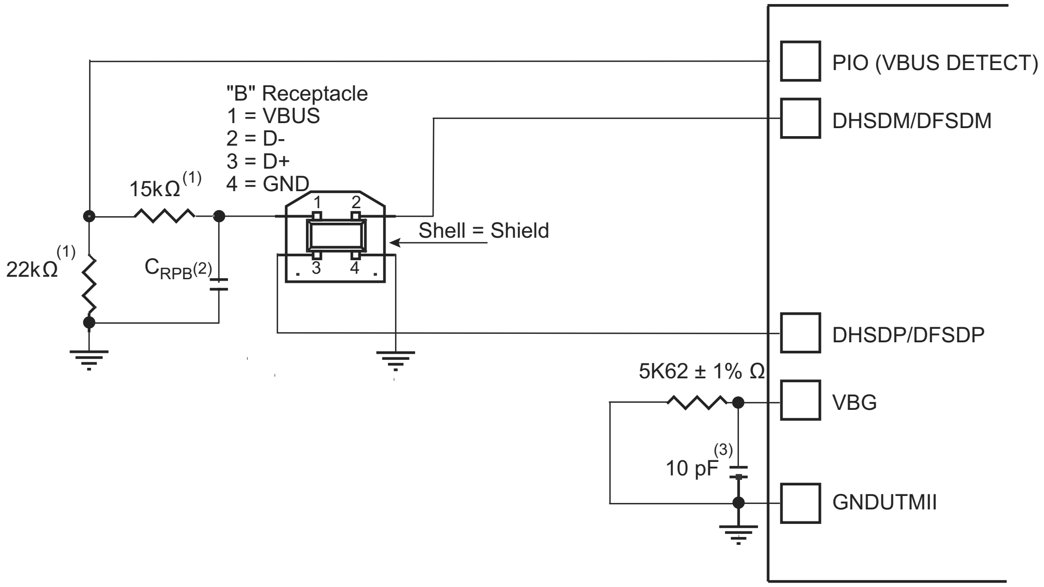
31	30	29	28	27	26	25	24
NFTX							
23	22	21	20	19	18	17	16
NFTX							
15	14	13	12	11	10	9	8
NFTX							
7	6	5	4	3	2	1	0
NFTX							

NFTX: Greater than 1518 Byte Frames Transmitted without Error

This register counts the number of 1518 or above byte frames successfully transmitted without error i.e., no underrun and not too many retries.

41.4 Typical Connection

Figure 41-2: Board Schematic



Note 1: The values shown on the 22 kΩ and 15 kΩ resistors are only valid with 3V3-supplied PIOs.

2: CRPB: Upstream Facing Port Bypass Capacitance of 1 µF to 10 µF (refer to “DC Electrical Characteristics” in Universal Serial Bus Specification Rev. 2)

3: 10 pF capacitor on VBG is a provision and may not be populated.

41.5 Product Dependencies

41.5.1 Power Management

The UDPHS is not continuously clocked.

For using the UDPHS, the programmer must first enable the UDPHS Clock in the Power Management Controller Peripheral Clock Enable Register (PMC_PCER). Then enable the PLL in the PMC UTMI Clock Configuration Register (CKGR_UCKR). Finally, enable BIAS in CKGR_UCKR.

However, if the application does not require UDPHS operations, the UDPHS clock can be stopped when not needed and restarted later.

41.5.2 Interrupt Sources

The UDPHS interrupt line is connected on one of the internal sources of the interrupt controller. Using the UDPHS interrupt requires the interrupt controller to be programmed first.

Table 41-1: Peripheral IDs

Instance	ID
UDPHS	42

This bit is set by hardware after a new packet has been stored in the endpoint FIFO.

This bit is cleared by the device firmware after reading the OUT data from the endpoint.

For multi-bank endpoints, this bit may remain active even when cleared by the device firmware, this if an other packet has been received meanwhile.

Hardware assertion of this bit may generate an interrupt if enabled by the UDPHS_EPTCTLx register RXRDY_TXKL bit.

This bit is reset by UDPHS_EPTRST register EPT_x (reset endpoint) and by UDPHS_EPTCTLDISx (disable endpoint).

– **KILL Bank** (for IN endpoint):

– The bank is really cleared or the bank is sent, BUSY_BANK_STA is decremented.

– The bank is not cleared but sent on the IN transfer, TX_COMPLT

– The bank is not cleared because it was empty. The user should wait that this bit is cleared before trying to clear another packet.

Note: “Kill a packet” may be refused if at the same time, an IN token is coming and the current packet is sent on the UDPHS line. In this case, the TX_COMPLT bit is set. Take notice however, that if at least two banks are ready to be sent, there is no problem to kill a packet even if an IN token is coming. In fact, in that case, the current bank is sent (IN transfer) and the last bank is killed.

TX_COMPLT: Transmitted IN Data Complete (cleared upon USB reset)

This bit is set by hardware after an IN packet has been sent.

This bit is reset by UDPHS_EPTRST register EPT_x (reset endpoint), and by UDPHS_EPTCTLDISx (disable endpoint).

TXRDY_TRER: TX Packet Ready/Transaction Error (cleared upon USB reset)

– **TX Packet Ready**:

This bit is cleared by hardware, as soon as the packet has been sent.

For Multi-bank endpoints, this bit may remain clear even after software is set if another bank is available to transmit.

Hardware clear of this bit may generate an interrupt if enabled by the UDPHS_EPTCTLx register TXRDY_TRER bit.

This bit is reset by UDPHS_EPTRST register EPT_x (reset endpoint), and by UDPHS_EPTCTLDISx (disable endpoint).

– **Transaction Error** (for high bandwidth isochronous OUT endpoints) (Read-Only):

This bit is set by hardware when a transaction error occurs inside one microframe.

If one toggle sequencing problem occurs among the n-transactions (n = 1, 2 or 3) inside a microframe, then this bit is still set as long as the current bank contains one “bad” n-transaction (refer to “CURBK: Current Bank (cleared upon USB reset)”). As soon as the current bank is relative to a new “good” n-transactions, then this bit is reset.

Note 1: A transaction error occurs when the toggle sequencing does not comply with the *Universal Serial Bus Specification, Rev 2.0* (5.9.2 High Bandwidth Isochronous endpoints) (Bad PID, missing data, etc.)

2: When a transaction error occurs, the user may empty all the “bad” transactions by clearing the Received OUT Data flag (RXRDY_TXKL).

If this bit is reset, then the user should consider that a new n-transaction is coming.

This bit is reset by UDPHS_EPTRST register EPT_x (reset endpoint), and by UDPHS_EPTCTLDISx (disable endpoint).

ERR_FL_ISO: Error Flow (cleared upon USB reset)

This bit is set by hardware when a transaction error occurs.

– Isochronous IN transaction is missed, the micro has no time to fill the endpoint (underflow).

– Isochronous OUT data is dropped because the bank is busy (overflow).

This bit is reset by UDPHS_EPTRST register EPT_x (reset endpoint) and by UDPHS_EPTCTLDISx (disable endpoint).

ERR_CRC_NTR: CRC ISO Error/Number of Transaction Error (cleared upon USB reset)

– **CRC ISO Error** (for Isochronous OUT endpoints) (Read-only):

This bit is set by hardware if the last received data is corrupted (CRC error on data).

This bit is updated by hardware when new data is received (Received OUT Data bit).

– **Number of Transaction Error** (for High Bandwidth Isochronous IN endpoints):

43.7.1 CLASSD Control Register

Name: CLASSD_CR

Address: 0xFC048000

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
–	–	–	–	–	–	–	SWRST

SWRST: Software Reset

0: No effect.

1: Reset the CLASSD simulating a hardware reset.

56.7.14 PWM Interrupt Enable Register 2

Name: PWM_IER2

Address: 0xF802C034

Access: Write-only

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

23	22	21	20	19	18	17	16
CMPU7	CMPU6	CMPU5	CMPU4	CMPU3	CMPU2	CMPU1	CMPU0

15	14	13	12	11	10	9	8
CMPM7	CMPM6	CMPM5	CMPM4	CMPM3	CMPM2	CMPM1	CMPM0

7	6	5	4	3	2	1	0
–	–	–	–	UNRE	–	–	WRDY

WRDY: Write Ready for Synchronous Channels Update Interrupt Enable

UNRE: Synchronous Channels Update Underrun Error Interrupt Enable

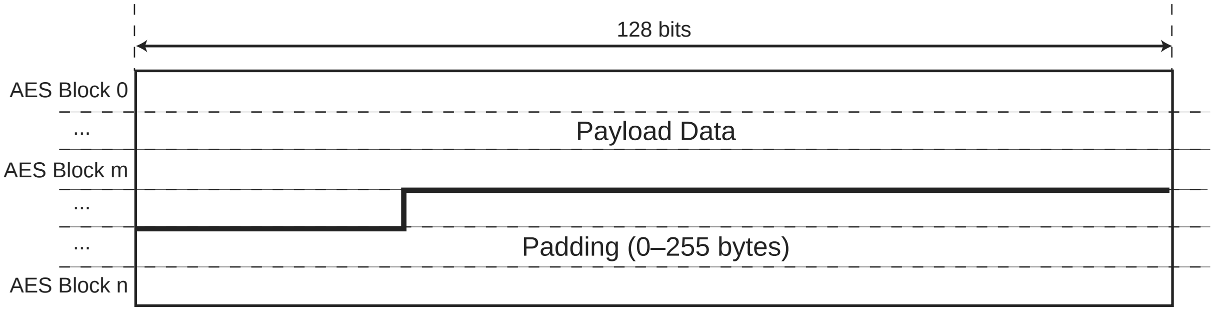
CMPMx: Comparison x Match Interrupt Enable

CMPUx: Comparison x Update Interrupt Enable

60.4.8.2 SSL Padding

Auto Padding is enabled by writing a ‘1’ to AES_EMR.APEN and SSL padding mode is selected by writing a ‘1’ to AES_EMR.APM.

Figure 60-10: SSL Padding



The padding length is configured in AES_EMR.PADLEN.

AES_BCNT.BCNT defines the length, in bytes, of the message to process. It must be configured before writing the first data in AES_IDATARx and the remaining bytes to process can be read at anytime (BCNT value is decremented after each AES_IDATARx access).

AES_BCNT.BCNT and AES_EMR.PADLEN must be configured so that the length of the message plus the length of the padding section is a multiple of the AES block size (128 bits).

To process a complete SSL message, the sequence is as follows:

1. Set AES_MR.OPMOD to either CBC or CTR mode.
2. Set AES_EMR.APEN to ‘1’, AES_EMR.APM to ‘1’, AES_EMR.PADLEN to the desired padding length in bytes.
3. Set AES_BCNT.BCNT with the whole message length, without padding, in bytes.
4. Set the AES Key Register.
5. Set AES_IVRx.IV if needed.
6. Fill AES_IDATARx.IDATA with the message to process according to the SMOD configuration used. On the last data block write only what is necessary (e.g., write only AES_IDATAR0 if last block size is ≤ 32 bits).
7. Wait for the DATRDY flag to be raised, meaning auto-padding completion and last block processing.

60.4.8.3 Flags

AES_ISR.EOPAD rises as soon as the automatic padding phase is over, meaning that all the extra padding blocks have been processed. Reading AES_ISR clears this flag.

AES_ISR.PLENERR indicates an error in the frame configuration, meaning that the whole message length including padding does not respect the standard selected. AES_ISR.PLENERR rises at the end of the frame in case of wrong message length and is cleared reading AES_ ISR.

In IPSEC/SSL standard message length including padding must be a multiple of the AES block size when CBC mode is used and multiple of 32-bit if CTR mode is used.

60.4.9 Secure Protocol Layers Improved Performances

Secure protocol layers such as IPSec require encryption and authentication. For IPSec, the authentication is based on HMAC, thus SHA is required. To optimize performance, the AES embeds a mode of operation that enables the SHA module to process the input or output data of the AES module. If this mode is enabled, write access is required only into AES_IDATARx registers, since SHA input data registers are automatically written by AES without software intervention. When the DMA is configured to transfer a buffer of data (input frame), only one transfer descriptor is required for both authentication and encryption/decryption processes and only one buffer is transferred through the system bus (reducing the load of the system bus).

Improved performance for secure protocol layers requires AES_EMR.PLIPEN to be set.

61. Secure Hash Algorithm (SHA)

61.1 Description

The Secure Hash Algorithm (SHA) is compliant with the American *FIPS (Federal Information Processing Standard) Publication 180-2* specification.

The 512/1024-bit block of message is respectively stored in 16/32 x 32-bit registers, (SHA_IDATARx/SHA_IODATARx) which are write-only.

As soon as the input data is written, the hash processing may be started. The registers comprising the block of a padded message must be entered consecutively. Then the message digest is ready to be read out on the 5 up to 8/16 x 32-bit output data registers (SHA_IODATARx) or through the DMA channels.

61.2 Embedded Characteristics

- Supports Secure Hash Algorithm (SHA1, SHA224, SHA256, SHA384, SHA512)
- Supports Hash-based Message Authentication Code (HMAC) Algorithm (HMAC-SHA1, HMAC-SHA224, HMAC-SHA256, HMAC-SHA384, HMAC-SHA512)
- Compliant with *FIPS Publication 180-2*
- Supports Automatic Padding of Messages
- Supports Up to 2 Sets of Initial Hash Values Registers (HMAC Acceleration or other)
- Supports Automatic Check of the Hash (HMAC Acceleration or other)
- Tightly Coupled to AES for Protocol Layers Improved Performances
- Configurable Processing Period:
  - 85 Clock Cycles to obtain a fast SHA1 runtime, 88 clock cycles for SHA384, SHA512 or 209 Clock Cycles for Maximizing Bandwidth of Other Applications
  - 72 Clock Cycles to obtain a fast SHA224, SHA256 runtime or 194 Clock Cycles for Maximizing Bandwidth of Other Applications
- Connection to DMA Channel Capabilities Optimizes Data Transfers
- Double Input Buffer Optimizes Runtime

61.3 Product Dependencies

61.3.1 Power Management

The SHA may be clocked through the Power Management Controller (PMC), so the programmer must first configure the PMC to enable the SHA clock.

61.3.2 Interrupt Sources

The SHA interface has an interrupt line connected to the Interrupt Controller.

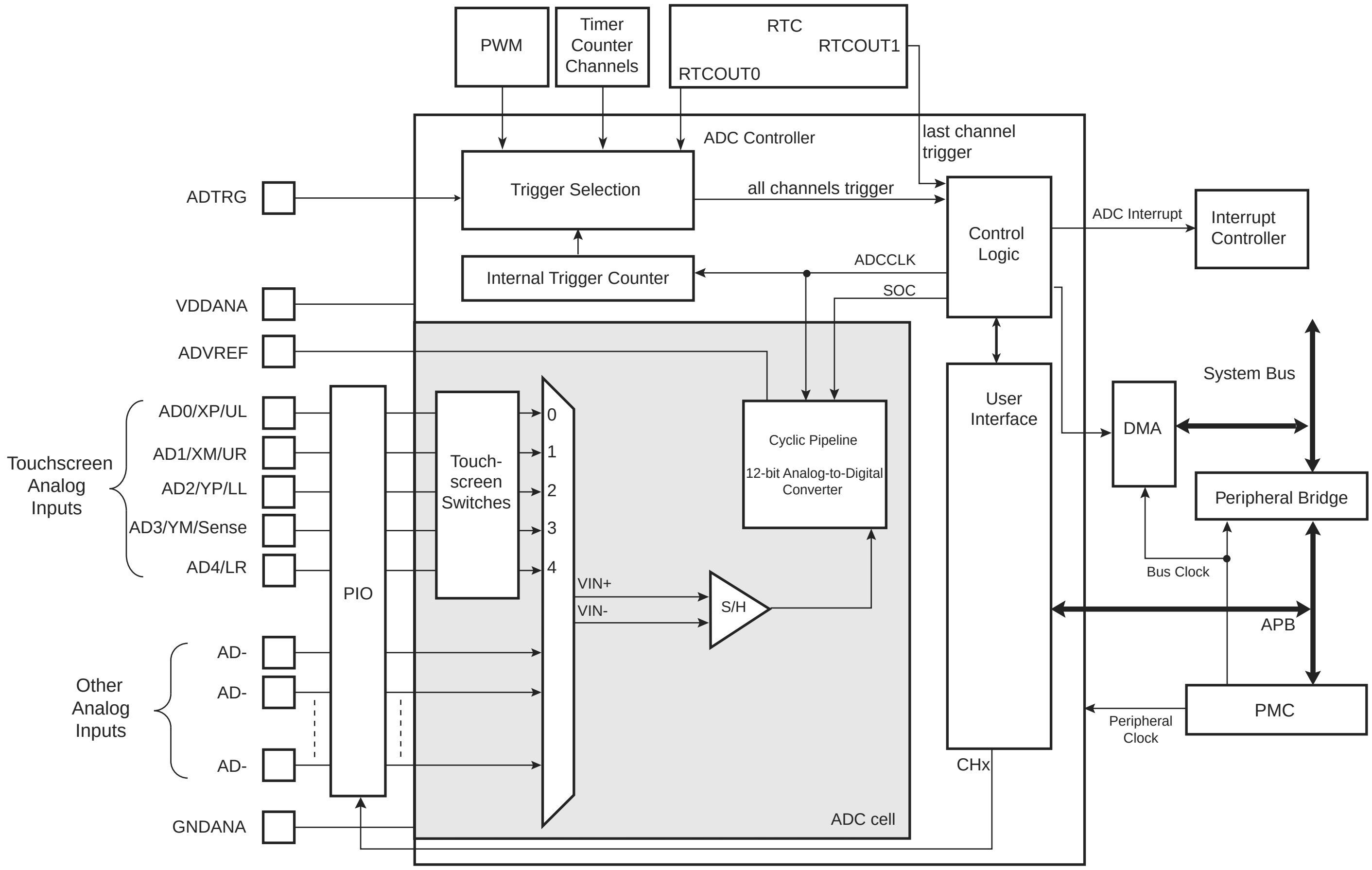
Handling the SHA interrupt requires programming the Interrupt Controller before configuring the SHA.

Table 61-1: Peripheral IDs

Instance	ID
SHA	12

65.3 Block Diagram

Figure 65-1: Analog-to-Digital Converter Block Diagram



65.4 Signal Description

Table 65-1: ADC Pin Description

Pin Name	Description
VDDANA	Analog power supply
ADVREF	Reference voltage
AD0–AD11	Analog input channels
ADTRG	External trigger

65.5 Product Dependencies

65.5.1 Power Management

The ADC Controller is not continuously clocked. The programmer must first enable the ADC Controller peripheral clock in the Power Management Controller (PMC) before using the ADC Controller. However, if the application does not require ADC operations, the ADC Controller clock can be stopped when not needed and restarted when necessary. Configuring the ADC Controller does not require the ADC Controller clock to be enabled.

65.7.20 ADC Analog Control Register

Name: ADC_ACR

Address: 0xFC030094

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
–	–	–	–	–	–	IBCTL	
7	6	5	4	3	2	1	0
–	–	–	–	–	–	PENDETSENS	

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

Note 1: By default, bits 12 and 13 are set to 1 and 0, respectively, and must not be modified.

PENDETSENS: Pen Detection Sensitivity

Modifies the pen detection input pull-up resistor value. Refer to section “Electrical Characteristics” for further details.

IBCTL: ADC Bias Current Control

Adapts performance versus power consumption. Refer to “Electrical Characteristics” for further details.