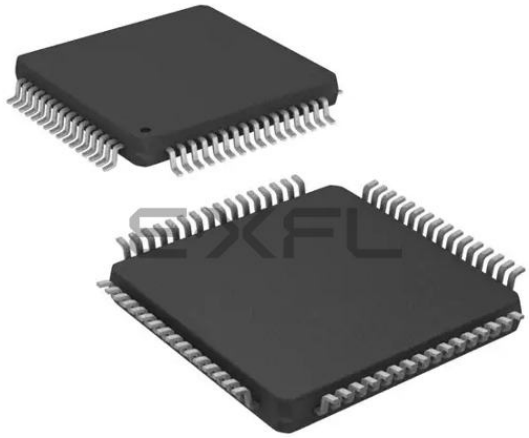


Welcome to E-XFL.COM

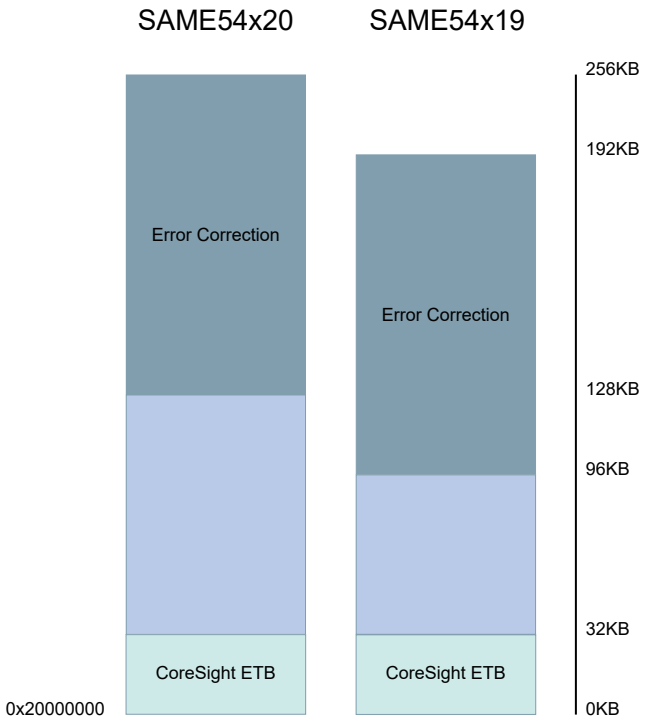
What is "Embedded - Microcontrollers"?

"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®-M4F
Core Size	32-Bit Single-Core
Speed	120MHz
Connectivity	EBI/EMI, I²C, IrDA, LINbus, MMC/SD, QSPI, SPI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, I²S, POR, PWM
Number of I/O	51
Program Memory Size	1MB (1M x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256K x 8
Voltage - Supply (Vcc/Vdd)	1.71V ~ 3.63V
Data Converters	A/D 24x12b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	64-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsamd51j20a-au

Figure 9-3. Memory with ECC and CoreSight ETB



9.4 NVM User Page Mapping

The NVM User Page can be read at address 0x00804000. The size of the NVM User Page is 512 Bytes.

The first eight 32-bit words (32 Bytes) of the Non Volatile Memory (NVM) User Page contain calibration data that are automatically read at device power-on. The remaining 480 Bytes can be used for storing custom parameters.

To write the NVM User Page refer to the documentation of the NVMCTRL (Non-Volatile Memory Controller).

When writing to the user pages, the new values do not get loaded by the other peripheral on the device until a device reset occurs.

Note: Before erasing the NVM User Page, ensure that the first 32 Bytes are read to a buffer and later written back to the same area unless a configuration change is intended.

Table 9-3. NVM User Page Mapping - Dedicated Entries

Bit Pos.	Name	Usage	Related Peripheral Register	Default Values
0	BOD33 Disable	BOD33 Disable at power-on.	SUPC.BOD33	0x1
8:1	BOD33 Level	BOD33 threshold level at power-on.	SUPC.BOD33	0x1C
10:9	BOD33 Action	BOD33 Action at power-on.	SUPC.BOD33	0x1

12.13.12 CoreSight ROM Table End

Name: END
Offset: 0x1008
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	END[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	END[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	END[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	END[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – END[31:0] End Marker
Indicates the end of the CoreSight ROM table entries.

Value	Name	Description
0xB	DIV1024	CLK_RTC_CNT = GCLK_RTC/1024
0xC-0xF	-	Reserved

Bit 7 – MATCHCLR Clear on Match
This bit is valid only in Mode 0 (COUNT32) and Mode 2 (CLOCK). This bit can be written only when the peripheral is disabled. This bit is not synchronized.

Value	Description
0	The counter is not cleared on a Compare/Alarm 0 match
1	The counter is cleared on a Compare/Alarm 0 match

Bit 6 – CLKREP Clock Representation
This bit is valid only in Mode 2 and determines how the hours are represented in the Clock Value (CLOCK) register. This bit can be written only when the peripheral is disabled. This bit is not synchronized.

Value	Description
0	24 Hour
1	12 Hour (AM/PM)

Bits 3:2 – MODE[1:0] Operating Mode
This field defines the operating mode of the RTC. This bit is not synchronized.

Value	Name	Description
0x0	COUNT32	Mode 0: 32-bit counter
0x1	COUNT16	Mode 1: 16-bit counter
0x2	CLOCK	Mode 2: Clock/calendar
0x3	-	Reserved

Bit 1 – ENABLE Enable
Due to synchronization there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the Enable bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

Value	Description
0	The peripheral is disabled
1	The peripheral is enabled

Bit 0 – SWRST Software Reset
Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets all registers in the RTC, except DBGCTRL, to their initial state, and the RTC will be disabled.

Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded.

Due to synchronization there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST will be cleared when the reset is complete.

initial checksum value (CHKINIT) stored in the Block Transfer Destination Address register (DSTADDR). The DMA read and calculate the checksum over the data from the source address.When the checksum calculation is completed, the CRC value is stored in the CRC Checksum register (CRCCHKSUM), the Transfer Complete interrupt flag is set (CHINTFLAGn.TCMPL) and optional interrupt is generated.

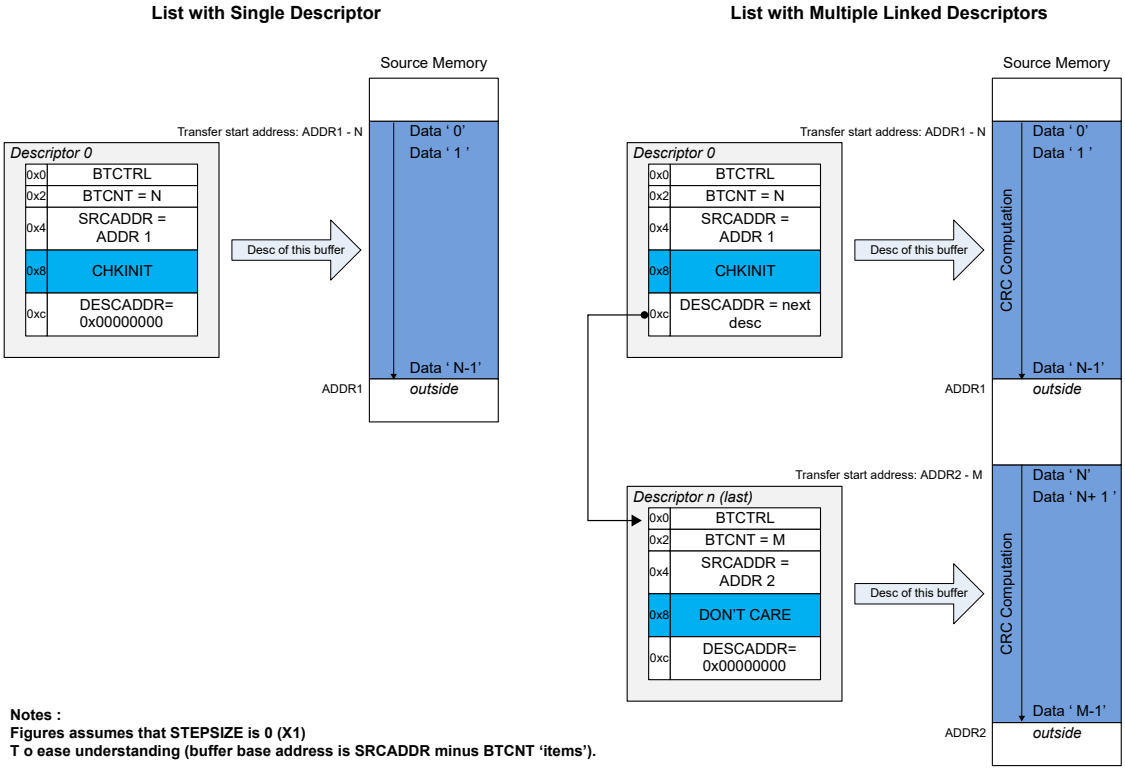
If linked descriptor is in the list (DESCADDR !=0), the DMA will fetch the next descriptor and CRC calculation continues as described above. When the last list descriptor is executed, the channel is automatically disabled.

In order to enable the memory CRC generation, the following actions must be performed:

1. The CRC module must be set to be used with a DMA channel (CRCCTRL.CRCSRC)
2. Reserve memory space addresses to configure a descriptor or a list of descriptors
3. Configure each descriptor:
   – Set the next descriptor address (DESCADDR)
   – Set the destination address with the initial checksum value (DSTADDR = CHKINIT) in the first descriptior in a list
   – Set the transfer source address (SRCADDR)
   – Set the block transfer count (BTCNT)
   – Set the memory CRC generation operation mode (CRCCTRL.CRCMODE = CRCGEN)
   – Enable optional interrupts
4. Enable the corresponding DMA channel (CHCTRLAn.ENABLE)

The figure below shows the CRC computation slots and descriptor configuration when single or linked-descriptors transfers are enabled.

Figure 22-20. CRC Computation with Single Linked Transfers



22.10.3 Block Transfer Source Address

Name: SRCADDR
Offset: 0x04
Property: -

The SRCADDR register offset is relative to (BASEADDR or WRBADDR) + Channel Number * 0x10

Bit	31	30	29	28	27	26	25	24
	SRCADDR[31:24]							
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	SRCADDR[23:16]							
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	SRCADDR[15:8]							
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	SRCADDR[7:0]							
Access								
Reset								

Bits 31:0 – SRCADDR[31:0] Transfer Source Address
This bit group holds the source address corresponding to the last beat transfer address in the block transfer.

24.9.22 GMAC Specific Address n Bottom Register

Name: SAB
Offset: 0x88 + n*0x08 [n=0..3]
Reset: 0x00000000
Property: -

The addresses stored in the Specific Address Registers are deactivated at reset or when their corresponding Specific Address Register Bottom is written. They are activated when Specific Address Register Top is written.

Bit	31	30	29	28	27	26	25	24
	ADDR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ADDR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – ADDR[31:0] Specific Address n
Least significant 32 bits of the destination address, that is, bits 31:0. Bit zero indicates whether the address is multicast or unicast and corresponds to the least significant bit of the first byte received.

24.9.32 GMAC 1588 Timer Second Comparison Low Register

Name: SCL
Offset: 0x0E0
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	SEC[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	SEC[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	SEC[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SEC[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – SEC[31:0] 1588 Timer Second Comparison Value
Value is compared to seconds value bits [31:0] of the TSU timer count value.

The lower blocks in the NVM main address space can be allocated as a boot loader section by using the BOOTPROT fuses, and the upper rows can be allocated to EEPROM.

The NVM memory is separated into six parts:

1. CB space
   Contains factory calibration and system configuration information.
   - Address; 0x00800000
   - Size: 1 page
   - Property: Read-Only
2. FS space
   Contains the factory signature information.
   - Address; 0x00806000
   - Size: 4 pages
   - Property: Read-Only.
3. USER space
   Contains user defined startup configuration. The first word is reserved, and used during the NVMCTRL start-up to automatically configure the device.
   - Address: 0x00804000
   - Size: 1 page
   - Property: Read-Write
4. Main address space
   The main address space is divided into 32 equally sized regions. Each region can be protected against write or erase operation. The 32-bit RUNLOCK register reflects the protection of each region. This register is automatically updated after power-up with the region lock user fuse data; To lock or unlock a region, the LR or UR commmands can be issued.
   - Address: 0x00000000
   - Size: PARAM.NVMP pages.
   - Property: Read-Write
5. Bootloader space
   The bootloader section starts at the beginning of the main address space; Its size is defined by the BOOTPROT[3:0] fuse. It is protected against write or erase operations, except if STATUS.BPDIS is set. Issuing a write or erase command at an address inside the BOOTPROT section sets STATUS.PROGE and STATUS.LOCKE. STATUS.BPDIS can be set by issuing the Set BOOTPROT Disable command (SBPDIS). It is cleared by issuing the Clear BOOTPROT Disable command (CBPDIS). This allows to program an new bootloader without changing the user page and issuing a new NVMCTRL startup sequence to reload the user configuration. The BOOTPROT section is not erased during a Chip-Erase operation even if STATUS.BPDIS is high.
   - Address: 0x00000000
   - Size: (15 - STATUS.BOOTPROT) × 8192
   - Property: Read-Only.
6. SmartEEPROM raw data space
   The SmartEEPROM algorithm emulates an EEPROM with a portion of the NVM main. Smart-EEPROM raw data is mapped at the end of the main address space. SmartEEPROM allocated space in the main address space is not accessible from AHB0/1. Any AHB access throws a

25.8.10 Page Buffer Load Data x

Name: PBLDATAn
Offset: 0x1C + n*0x04 [n=0..1]
Reset: 0xFFFFFFFF
Property: -

Bit	31	30	29	28	27	26	25	24
	DATA[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DATA[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DATA[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	1

Bits 31:0 – DATA[31:0] Page Buffer Data

26.8.10 Hash Area Start Address Register

Name: HASH
Offset: 0x34
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	HASA[24:17]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	HASA[16:9]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	HASA[8:1]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	HASA[0:0]							
Access	R/W							
Reset	0							

Bits 31:7 – HASA[24:0] Hash Area Start Address
This field points at the Hash memory location. The address must be a multiple of 128 bytes.

This bit is not synchronized.

Mode	CPOL	CPHA	Leading Edge	Trailing Edge
0x0	0	0	Rising, sample	Falling, change
0x1	0	1	Rising, change	Falling, sample
0x2	1	0	Falling, sample	Rising, change
0x3	1	1	Falling, change	Rising, sample

Value	Description
0	The data is sampled on a leading SCK edge and changed on a trailing SCK edge.
1	The data is sampled on a trailing SCK edge and changed on a leading SCK edge.

Bits 27:24 – FORM[3:0] Frame Format
This bit field selects the various frame formats supported by the SPI in slave mode. When the 'SPI frame with address' format is selected, the first byte received is checked against the ADDR register.

FORM[3:0]	Name	Description
0x0	SPI	SPI frame
0x1	-	Reserved
0x2	SPI_ADDR	SPI frame with address
0x3-0xF	-	Reserved

Bits 21:20 – DIPO[1:0] Data In Pinout
These bits define the data in (DI) pad configurations.

In master operation, DI is MISO.

In slave operation, DI is MOSI.

These bits are not synchronized.

DIPO[1:0]	Name	Description
0x0	PAD[0]	SERCOM PAD[0] is used as data input
0x1	PAD[1]	SERCOM PAD[1] is used as data input
0x2	PAD[2]	SERCOM PAD[2] is used as data input
0x3	PAD[3]	SERCOM PAD[3] is used as data input

Bits 17:16 – DOPO[1:0] Data Out Pinout
This bit defines the available pad configurations for data out (DO) and the serial clock (SCK). In slave operation, the slave select line ($\overline{SS}$) is controlled by DOPO, while in master operation the $\overline{SS}$ line is controlled by the port configuration.

In master operation, DO is MOSI.

In slave operation, DO is MISO.

These bits are not synchronized.

If CTRLA.RUNSTDBY=0, the GLK_SERCOMx_CORE will be disabled after any ongoing transaction is finished. Any interrupt can wake up the device.

I²C Slave Operation

Writing CTRLA.RUNSTDBY=1 will allow the Address Match interrupt to wake up the device.

When CTRLA.RUNSTDBY=0, all receptions will be dropped.

36.6.6 Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following bits are synchronized when written:

- Software Reset bit in the CTRLA register (CTRLA.SWRST)
- Enable bit in the CTRLA register (CTRLA.ENABLE)
- Command bits in CTRLB register (CTRLB.CMD)
- Write to Bus State bits in the Status register (STATUS.BUSSTATE)
- Address bits in the Address register (ADDR.ADDR) when in master operation.

The following registers are synchronized when written:

- Data (DATA) when in master operation
- Length (LENGTH) when in slave operation

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

37.8.11 Instruction Code

Name: INSTRCTRL
Offset: 0x34
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	OPTCODE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

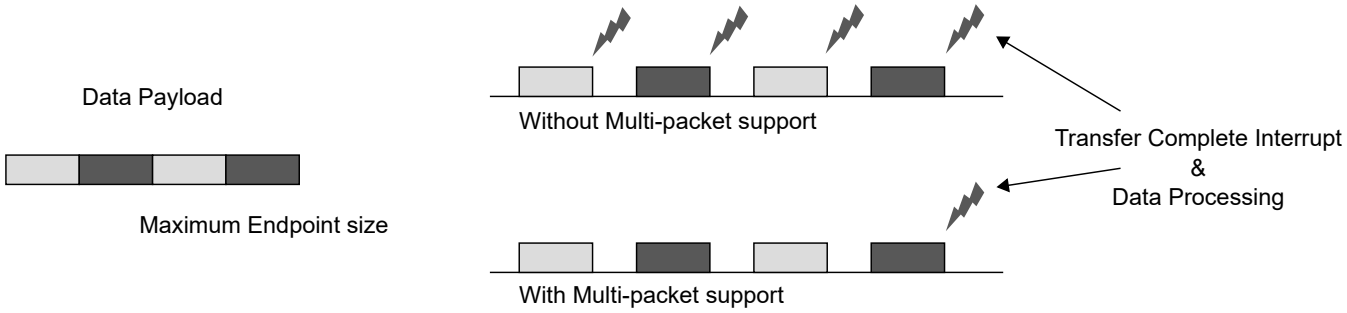
Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	INSTR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – OPTCODE[7:0] Option Code
These bits define the option code to send to the serial flash memory.

Bits 7:0 – INSTR[7:0] Instruction Code
Instruction code to send to the serial flash memory.

Figure 38-3. Multi-Packet Feature - Reduction of CPU Overhead



38.6.2.4 USB Reset

The USB bus reset is initiated by a connected host and managed by hardware.

During USB reset the following registers are cleared:

- Device Endpoint Configuration (EPCFG) register - except for Endpoint 0
- Device Frame Number (FNUM) register
- Device Address (DADD) register
- Device Endpoint Interrupt Enable Clear/Set (EPINTENCLR/SET) register
- Device Endpoint Interrupt Flag (EPINTFLAG) register
- Transmit Stall 0 bit in the Endpoint Status register (EPSTATUS.STALLRQ0)
- Transmit Stall 1 bit in the Endpoint Status register (EPSTATUS.STALLRQ1)
- Endpoint Interrupt Summary (EPINTSMRY) register
- Upstream resume bit in the Control B register (CTRLB.UPRSM)

At the end of the reset process, the End of Reset bit is set in the Interrupt Flag register (INTFLAG.EORST).

38.6.2.5 Start-of-Frame

When a Start-of-Frame (SOF) token is detected, the frame number from the token is stored in the Frame Number field in the Device Frame Number register (FNUM.FNUM), and the Start-of-Frame interrupt bit in the Device Interrupt Flag register (INTFLAG.SOF) is set. If there is a CRC or bit-stuff error, the Frame Number Error status flag (FNUM.FNCERR) in the FNUM register is set.

38.6.2.6 Management of SETUP Transactions

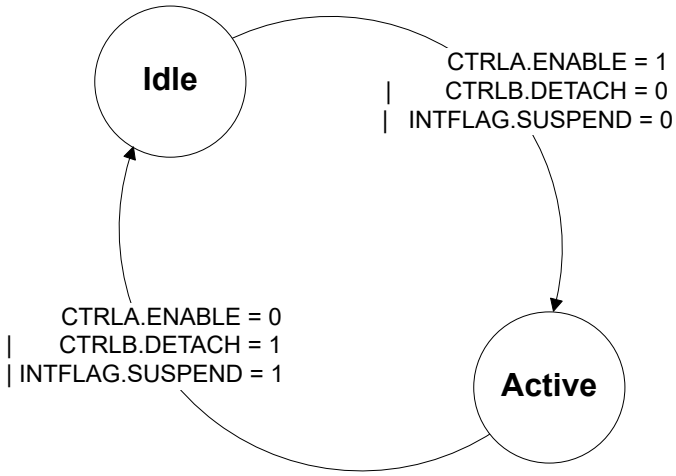
When a SETUP token is detected and the device address of the token packet does not match DADD.DADD, the packet is discarded and the USB module returns to idle and waits for the next token packet.

When the address matches, the USB module checks if the endpoint is enabled in EPCFG. If the addressed endpoint is disabled, the packet is discarded and the USB module returns to idle and waits for the next token packet.

When the endpoint is enabled, the USB module then checks on the EPCFG of the addressed endpoint. If the EPCFG.EPTYPE0 is not set to control, the USB module returns to idle and waits for the next token packet.

When the EPCFG.EPTYPE0 matches, the USB module then fetches the Data Buffer Address (ADDR) from the addressed endpoint's descriptor and waits for a DATA0 packet. If a PID error or any other PID than DATA0 is detected, the USB module returns to idle and waits for the next token packet.

Figure 38-7. Pad Behavior



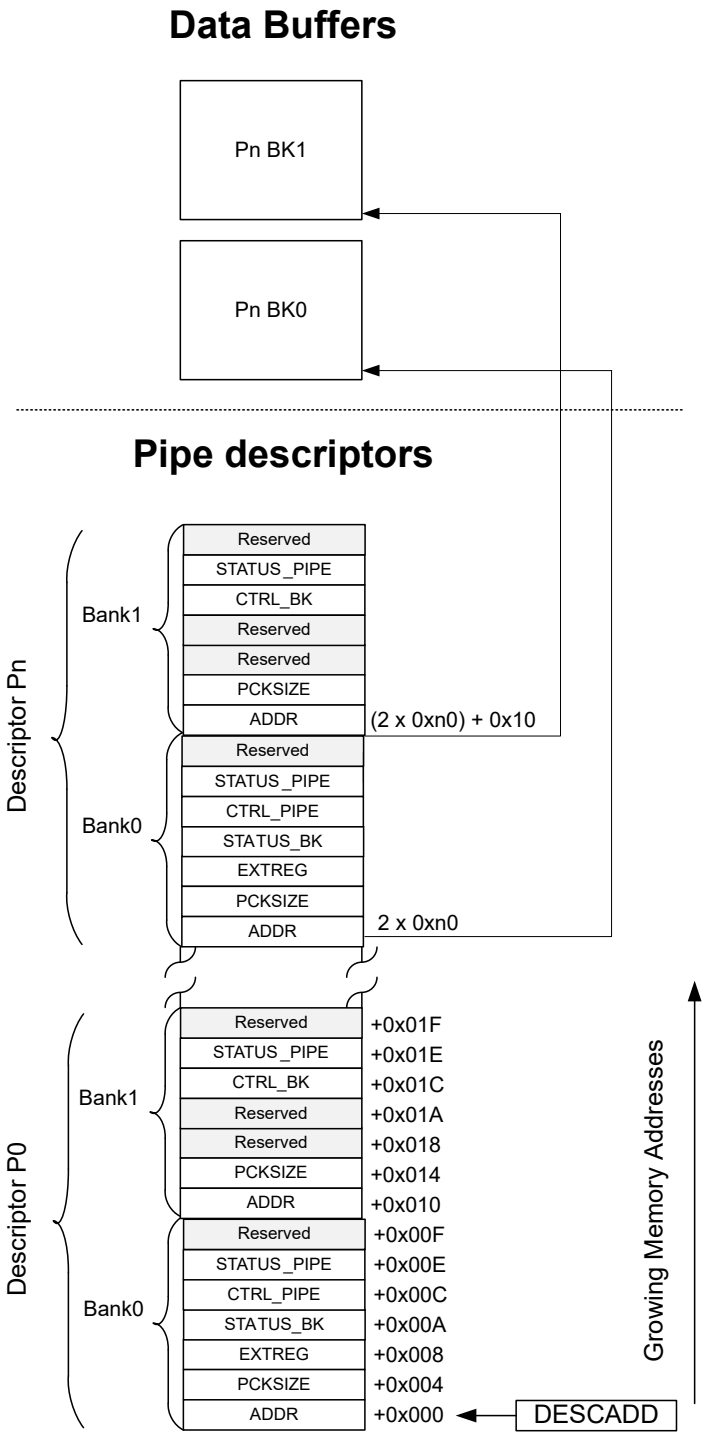
In Idle state, the pad is in Low Power Consumption mode.

In Active state, the pad is active.

The following figure, Pad Events, illustrates the pad events leading to a PAD state change.

38.8.7 Host Registers - Pipe RAM

38.8.7.1 Pipe Descriptor Structure



Parameter	Type	Direction	Location	Data Length	Before Executing the Service	After Executing the Service
u1MillerRabin-Iterations	u1	I	–	–	Miller Rabin's T parameter	Miller Rabin's T parameter
u2MaxIncrement	u2	I	–	–	Maximum Increment (see **Note 4**)	Maximum Increment

Note:

1. This zone contains the number to be either tested or used as a seed for generation. It has to be provided with one zero word on the MSB side. This area has supplementary constraints (see the following Important note).

1. This parameter does not have to be provided and is used as an internal value for computing the reduction's constant.
2. The area {nu1ExpBase, u2NLength + 4} must be entirely in the Crypto RAM.
3. The generation starts from the number in {nu1NBase,u2NLength + 4} and increments it until a number is found as probable prime. However, the generation may stop for two reasons: The number has been incremented in a way it is bigger than <u2NLength> bytes, or the original number has been incremented by more than <u2MaxIncrement>.

In case of probable prime generation, ensure that the addition of NSeed and Maximum Increment is not a number with more bytes than u2NLength, as this would produce an overflow.

Important:
One additional word is used on the LSB side of the NBase parameter; this word is restored at the end of the calculus. As a consequence, the parameter nu1NBase must never be at the beginning of the Crypto RAM, but at least at one word from the beginning.

One additional word is used on the MSB side of the NBase parameter; this word is not corrupted. As a consequence the Area {nu1NBase, u2NLength} must not be at the end of the Crypto RAM but at least at one word from the end.

Prime numbers of a size lower than 96 bits (three 32-bit words) cannot be generated or tested by this service.

43.3.5.3.6 Options

Some of the Prime Generation options configure the Modular Exponentiation steps and so are very similar to the Modular Exponentiation options.

The options are set by the u2Options input parameter, which is composed of:

- the mandatory Operation Option described in Table 43-57
- the mandatory Calculus Mode Option described in Table 43-58
- the mandatory Window Size Option described in Table 43-59

The u2Options number is calculated by an "Inclusive OR" of the options. Some Examples in C language are:

- Operation: Probable Prime Testing with Fast Modular Exponentiation and the window size equal to 1

```
PUKCL(u2Options) = PUKCL_PRIMEGEN_TEST | PUKCL_EXPMOD_FASTRSA |
PUKCL_EXPMOD_WINDOWSIZE_1;
```

43.3.7.3.2 How to Use the Service

43.3.7.3.3 Description

The operation performed is:

$Pt_C = 2 \times Pt_A$

In this computation, the following parameters need to be provided:

- A the input point is filled in projective coordinates (X,Y,Z) (pointed by {nu1PointABase, 3*u2ModLength + 12}). This point can be the Infinite Point.
- Cns the Fast Modular Constant filled (pointed by {nu1CnsBase,u2ModLength +8})
- P the modulus filled (pointed by {nu1ModBase,u2ModLength +4})
- The workspace not initialized (pointed by {nu1WorkSpace, 4*u2ModLength +28}
- The a and b Parameters relative to the Elliptic Curve Equation (pointed by {nu1ABBase, 2*u2ModLength+ 8})
- The resulting C point is represented in projective coordinates (X,Y,Z) and is stored at the very same place than the input point A. This point can be the Infinite Point.

The service name for this operation is `GF2NEccDblFast`. This service uses Fast mode and Fast Modular Reduction for computation.

Important: Before using this service, ensure that the constant Cns has been calculated with the setup of the Fast Modular Reductions service.

43.3.7.3.4 Parameters Definition

Table 43-96. GF2NEccDblFast Service Parameters

Parameter	Type	Direction	Location	Data Length	Before Executing the Service	After Executing the Service
nu1ModBase	nu1	I	Crypto RAM	u2ModLength + 4	Base of modulus P	Base of modulus P
nu1CnsBase	nu1	I	Crypto RAM	u2ModLength + 12	Base of Cns	Base of Cns
u2ModLength	u2	I	–	–	Length of modulus P	Length of modulus P
nu1ABBase	u2	I	Crypto RAM	2*u2ModLength + 8	Parameters a and b of the elliptic curve	Parameter a and b of the elliptic curve
nu1PointABase	nu1	I/O	Crypto RAM	3*u2ModLength + 12	Input point A (projective coordinates)	Resulting point C (projective coordinates)
nu1Workspace	nu1	I	Crypto RAM	4*u2ModLength + 28	–	Corrupted workspace

45.8.20 DSEQCTRL

Name: DSEQCTRL
Offset: 0x38
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	AUTOSTART							
Access	R/W							
Reset	0							

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
								OFFSETCORR
Access								R/W
Reset								0

Bit	7	6	5	4	3	2	1	0
	GAINCORR	WINUT	WINLT	SAMPCTRL	AVGCTRL	REFCTRL	CTRLB	INPUTCTRL
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 31 – AUTOSTART ADC Auto-Start Conversion

Value	Description
0	ADC conversion starts when a DMA sequence is complete and a start software or event trigger is received.
1	ADC conversion automatically starts when a DMA sequence is complete. This setting is ignored if the convertion start by event is enabled (EVCTRL.STARTEI=1).

Bit 8 – OFFSETCORR Offset Correction

Value	Description
0	DMA update of the Offset Correction register is disabled.
1	DMA update of the Offset Correction register is enabled.

Bit 7 – GAINCORR Gain Correction

Value	Description
0	DMA update of the Gain Correction register is disabled.
1	DMA update of the Gain Correction register is enabled.

Bit 6 – WINUT Window Monitor Upper Threshold

To configure PDM2 mode, set SLOTSIZE = 0x01 (16-bits), NBSLOTS = 0x00 (1 slots) and RXCTRL.DATASIZE = 0x00 (32-bit).

51.6.7 Data Formatting Unit

To allow more flexibility, data words received by the Receive Serializer will be formatted by the Receive Formatting Unit before being stored into the Data Holding register (DATAm). The data words written into DATAm register will be formatted by the Transmit Formatting Unit before transmission by the Transmit Serializer .

The formatting options are defined in RXCTRL and TXCTRL:

- SLOTADJ for left or right justification in the slot
- BITREV for bit reversal
- WORDADJ for left or right justification in the data word
- EXTEND for extension to the word size

51.6.8 DMA, Interrupts and Events

Table 51-2. Module Request for I^2S

Condition	DMA request	DMA request is cleared	Interrupt request	Event input/ output
Receive Ready	YES	When data is read	YES	
Transmit Ready (Buffer empty)	YES	When data is written	YES	
Receive Overrun			YES	
Transmit Underrun			YES	

51.6.8.1 DMA Operation

Each Serializer can be connected either to one single DMAC channel or to one DMAC channel per data slot in stereo mode. This is selected by writing the RXCTRL/TXCTRL.DMA bit:

Table 51-3. I^2C DMA Request Generation

SERCTRLm.DMA	Mode	Slot Parity	DMA Request Trigger
0	Receiver	all	I2S_DMAC_ID_RX_m
	Transmitter	all	I2S_DMAC_ID_TX_m
1	Receiver	even	I2S_DMAC_ID_RX_0
		odd	I2S_DMAC_ID_RX_1
	Transmitter	even	I2S_DMAC_ID_TX_0
		odd	I2S_DMAC_ID_TX_1

The DMAC reads from the RXDATA register and writes to the TXDATA register for all data slots, successively.

The DMAC transfers may use 32-, 16- or or 8-bit transactions according to the value of the TXCTRL/ RXCTRL.DATASIZE field. 8-bit compact stereo uses 16-bit and 16-bit compact stereo uses 32-bit transactions.