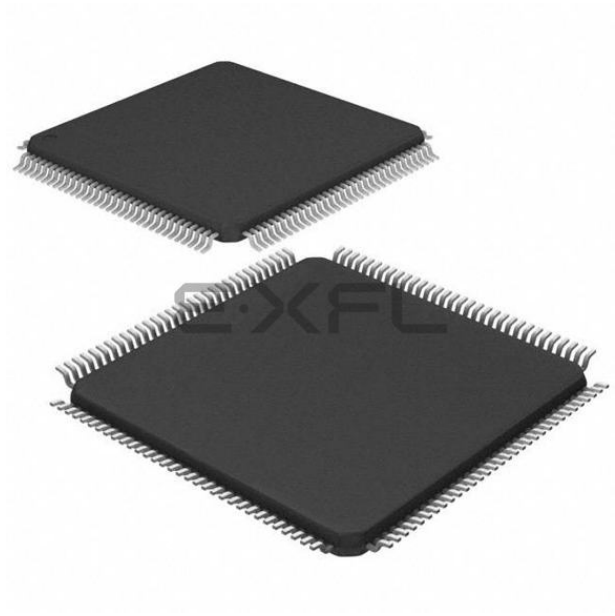


**Welcome to E-XFL.COM**

## 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 | 99 |
| Program Memory Size | 512KB (512K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 192K x 8 |
| Voltage - Supply (Vcc/Vdd) | 1.71V ~ 3.63V |
| Data Converters | A/D 32x12b; D/A 2x12b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 128-TQFP |
| Supplier Device Package | 128-TQFP (14x14) |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/atsamd51p19a-aut |

44. TRNG – True Random Number Generator....1613
44.1. Overview....1613
44.2. Features....1613
44.3. Block Diagram....1613
44.4. Signal Description....1613
44.5. Product Dependencies....1613
44.6. Functional Description....1615
44.7. Register Summary....1617
44.8. Register Description....1617

45. ADC – Analog-to-Digital Converter....1624
45.1. Overview....1624
45.2. Features....1624
45.3. Block Diagram....1625
45.4. Signal Description....1625
45.5. Product Dependencies....1626
45.6. Functional Description....1627
45.7. Register Summary....1645
45.8. Register Description....1646

46. AC – Analog Comparators....1679
46.1. Overview....1679
46.2. Features....1679
46.3. Block Diagram....1680
46.4. Signal Description....1680
46.5. Product Dependencies....1680
46.6. Functional Description....1682
46.7. Register Summary....1692
46.8. Register Description....1692

47. DAC – Digital-to-Analog Converter....1711
47.1. Overview....1711
47.2. Features....1711
47.3. Block Diagram....1711
47.4. Signal Description....1712
47.5. Product Dependencies....1712
47.6. Functional Description....1714
47.7. Register Summary....1724
47.8. Register Description....1724

48. TC – Timer/Counter....1750
48.1. Overview....1750
48.2. Features....1750
48.3. Block Diagram....1751
48.4. Signal Description....1751
48.5. Product Dependencies....1752
48.6. Functional Description....1753

### 19.8.5 3.3V Brown-Out Detector (BOD33) Control

**Name:** BOD33
**Offset:** 0x10
**Reset:** Determined from NVM User Row
**Property:** Write-Synchronized, PAC Write-Protection

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | VBATLEVEL[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 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | LEVEL[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | x | x | x | x | x | x |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | PSEL[2:0] | | | HYST[3:0] | | | |
| Access | | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | 0 | 0 | 0 | x | x | x | x |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RUNBKUP | RUNHIB | RUNSTDBY | STDBYCFG | ACTION[1:0] | | ENABLE | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| Reset | 0 | 0 | 0 | 0 | y | y | z | |

**Bits 31:24 – VBATLEVEL[7:0]** BOD33 Threshold Level on VBAT
This field sets the triggering voltage threshold for the BOD33 when the BOD33 monitors VBAT in battery backup sleep mode.

This field is not synchronized.

**Bits 23:16 – LEVEL[7:0]** BOD33 Threshold Level on VDD
This field sets the triggering voltage threshold for the BOD33 when the BOD33 monitors VDD. If an hysteresis value is programmed (BOD33.HYST), this field corresponds to the lower threshold ($V_{BOD-}$).

These bits are loaded from NVM User Row at start-up.

This field is not synchronized.

The VBOD- input voltage can be calculated as follows: VBOD- = 1.5 + LEVEL[7:0) x Level_Step

And the upper threshold (VBOD+) is then: VBOD+ = VBOD- + N x HYST_STEP, With N=0 to 15 according to HYST[3:0] value and HYST_STEP = Level_Step, (refer to Bits 11:8 – HYST[3:0]: BOD33 Hysteresis voltage value on VDD).

At the upper side of Level[7:0] values depending on the Hysteresis value chosen with HYST[3:0], the VBOD+ level reaches an overflow, e.g., for HYST[3:0] = 0d2 the hysteresis is 2 x Level_Step = 12 mV up to position 253 and position 254 to 255 above must not be used.

### 21.10.14 Tamper Control

**Name:** TAMPCTRL
**Offset:** 0x60
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-Protected

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | DEBNC4 | DEBNC3 | DEBNC2 | DEBNC1 | DEBNC0 |
| Access | | | | | | | | |
| Reset | | | | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | TAMLVL4 | TAMLVL3 | TAMLVL2 | TAMLVL1 | TAMLVL0 |
| Access | | | | | | | | |
| Reset | | | | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | IN4ACT[1:0] | |
| Access | | | | | | | | |
| Reset | | | | | | | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | IN3ACT[1:0] | | IN2ACT[1:0] | | IN1ACT[1:0] | | IN0ACT[1:0] | |
| Access | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 24, 25, 26, 27, 28 – DEBNC** Debounce Enable of Tamper Input INn

| Value | Description |
|---|---|
| 0 | Debouncing is disabled for Tamper input INn |
| 1 | Debouncing is enabled for Tamper input INn |

**Bits 16, 17, 18, 19, 20 – TAMLVL** Tamper Level Select of Tamper Input INn

| Value | Description |
|---|---|
| 0 | A falling edge condition will be detected on Tamper input INn. |
| 1 | A rising edge condition will be detected on Tamper input INn. |

**Bits 0:1, 2:3, 4:5, 6:7, 8:9 – INACT** Tamper Channel n Action
These bits determine the action taken by Tamper Channel n.

| Value | Name | Description |
|---|---|---|
| 0x0 | OFF | Off (Disabled) |
| 0x1 | WAKE | Wake and set Tamper flag |
| 0x2 | CAPTURE | Capture timestamp and set Tamper flag |
| 0x3 | ACTL | Compare RTC signal routed between INn and OUT pins . When a mismatch occurs, capture timestamp and set Tamper flag |

Incrementation for the source address of a block transfer is enabled by writing the Source Address Incrementation Enable bit in the Block Transfer Control register (BTCTRL.SRCINC=1). The step size of the incrementation is configurable and can be chosen by writing the Step Selection bit in the Block Transfer Control register (BTCTRL.STEPSEL=1) and writing the desired step size in the Address Increment Step Size bit group in the Block Transfer Control register (BTCTRL.STEPSIZE). If BTCTRL.STEPSEL=0, the step size for the source incrementation will be the size of one beat.

When source address incrementation is configured (BTCTRL.SRCINC=1), SRCADDR is calculated as follows:

If BTCTRL.STEPSEL=1:

$$\mathrm{SRCADDR} = \mathrm{SRCADDR}_{START} + BTCNT \cdot \left(BEATSIZE + 1\right) \cdot 2^{\mathrm{STEPSIZE}}$$

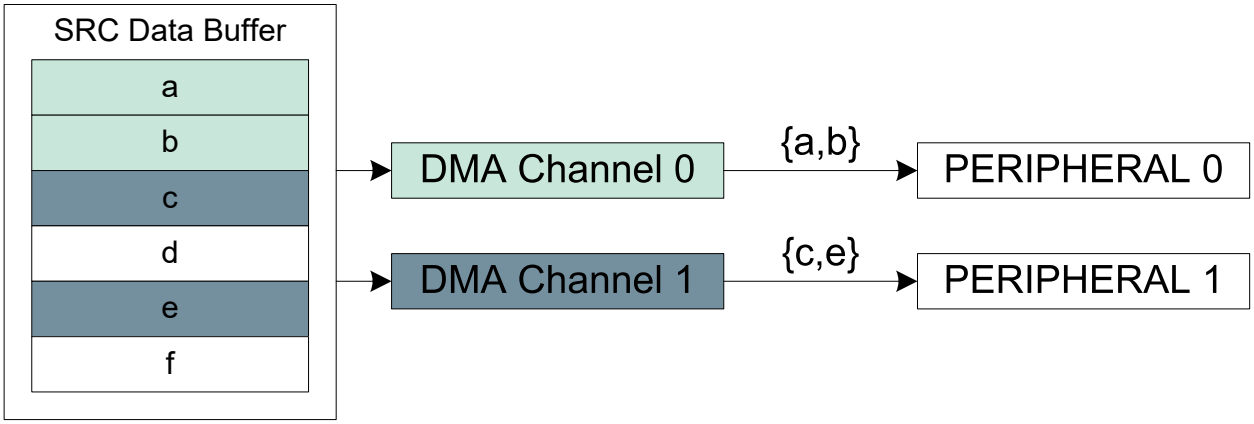
If BTCTRL.STEPSEL=0:

$$\mathrm{SRCADDR} = \mathrm{SRCADDR}_{START} + BTCNT \cdot \left(BEATSIZE + 1\right)$$

- $\mathrm{SRCADDR}_{\mathrm{START}}$ is the source address of the first beat transfer in the block transfer
- BTCNT is the initial number of beats remaining in the block transfer
- BEATSIZE is the configured number of bytes in a beat
- STEPSIZE is the configured number of beats for each incrementation

The following figure shows an example where DMA channel 0 is configured to increment the source address by one beat after each beat transfer (BTCTRL.SRCINC=1), and DMA channel 1 is configured to increment the source address by two beats (BTCTRL.SRCINC=1, BTCTRL.STEPSEL=1, and BTCTRL.STEPSIZE=0x1). As the destination address for both channels are peripherals, destination incrementation is disabled (BTCTRL.DSTINC=0).

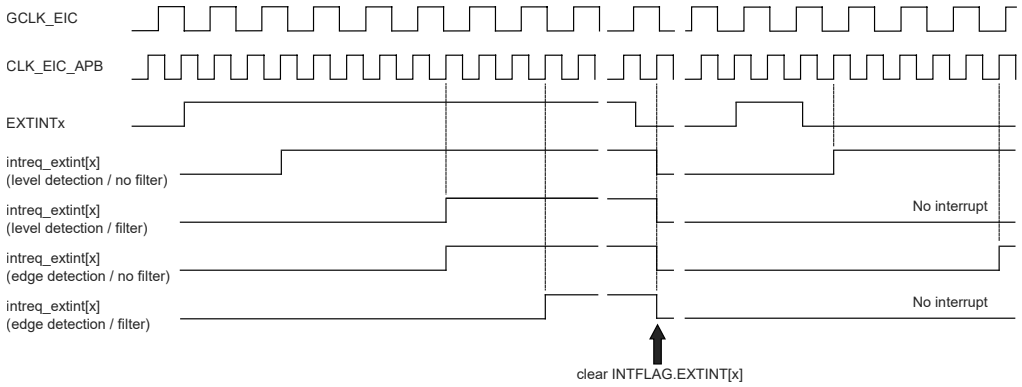
**Figure 22-8. Source Address Increment**




Incrementation for the destination address of a block transfer is enabled by setting the Destination Address Incrementation Enable bit in the Block Transfer Control register (BTCTRL.DSTINC=1). The step size of the incrementation is configurable by clearing BTCTRL.STEPSEL=0 and writing BTCTRL.STEPSIZE to the desired step size. If BTCTRL.STEPSEL=1, the step size for the destination incrementation will be the size of one beat.

When the destination address incrementation is configured (BTCTRL.DSTINC=1), DSTADDR must be set and calculated as follows:

**Figure 23-2. Interrupt Detection Latency by modes (Rising Edge)**




The detection latency depends on the detection mode.

**Table 23-2. Detection Latency**

| Detection mode | Latency (worst case) |
|---|---|
| Level without filter | Five CLK_EIC_APB periods |
| Level with filter | Four GCLK_EIC/CLK_ULP32K periods + five CLK_EIC_APB periods |
| Edge without filter | Four GCLK_EIC/CLK_ULP32K periods + five CLK_EIC_APB periods |
| Edge with filter | Six GCLK_EIC/CLK_ULP32K periods + five CLK_EIC_APB periods |

**Related Links**

14. GCLK - Generic Clock Controller

23.8.10 CONFIG

### 23.6.4 Additional Features

#### 23.6.4.1 Non-Maskable Interrupt (NMI)

The non-maskable interrupt pin can also generate an interrupt on edge or level detection, but it is configured with the dedicated NMI Control register (NMICTRL). To select the sense for NMI, write to the NMISENSE bit group in the NMI Control register (NMICTRL.NMISENSE). NMI filtering is enabled by writing a '1' to the NMI Filter Enable bit (NMICTRL.NMIFILTEN).

If edge detection or filtering is required, enable GCLK_EIC or CLK_ULP32K.

NMI detection is enabled only by the NMICTRL.NMISENSE value, and the EIC is not required to be enabled.

When an NMI is detected, the non-maskable interrupt flag in the NMI Flag Status and Clear register is set (NMIFLAG.NMI). NMI interrupt generation is always enabled, and NMIFLAG.NMI generates an interrupt request when set.

#### 23.6.4.2 Asynchronous Edge Detection Mode (No Debouncing)

The EXTINT edge detection can be operated synchronously or asynchronously, selected by the Asynchronous Control Mode bit for external pin x in the External Interrupt Asynchronous Mode register (ASYNCH.ASYNCH[x]). The EIC edge detection is operated synchronously when the Asynchronous Control Mode bit (ASYNCH.ASYNCH[x]) is '0' (default value). It is operated asynchronously when ASYNCH.ASYNCH[x] is written to '1'.

| Frame Segment | Value |
|---|---|
| UDP (Octet 20) | 11 |
| IP stuff (Octets 21–37) | — |
| IP DA (Octets 38–53) | FF0200000000006B |
| Source IP port (Octets 54–55) | — |
| Dest IP port (Octets 56–57) | 013F |
| Other stuff (Octets 58–61) | — |
| Message type (Octet 62) | 03 |
| Other stuff (Octets 63–93) | — |
| Version PTP (Octet 94) | 02 |

For the multicast address 011B19000000 sync and delay request frames are recognized depending on the message type field, 00 for sync and 01 for delay request.

**Table 24-11. Example of Sync Frame in 1588 Version 2 (Ethernet Multicast) Format**

| Frame Segment | Value |
|---|---|
| Preamble/SFD | 55555555555555D5 |
| DA (Octets 0–5) | 011B19000000 |
| SA (Octets 6–11) | — |
| Type (Octets 12–13) | 88F7 |
| Message type (Octet 14) | 00 |
| Version PTP (Octet 15) | 02 |

Pdelay request frames need a special multicast address so they can pass through ports blocked by the spanning tree protocol. For the multicast address 0180C200000E sync, Pdelay_Req and Pdelay_Resp frames are recognized depending on the message type field, 00 for sync, 02 for pdelay request and 03 for pdelay response.

**Table 24-12. Example of Pdelay_Req Frame in 1588 Version 2 (Ethernet Multicast) Format**

| Frame Segment | Value |
|---|---|
| Preamble/SFD | 55555555555555D5 |
| DA (Octets 0–5) | 0180C200000E |
| SA (Octets 6–11) | — |
| Type (Octets 12–13) | 88F7 |
| Message type (Octet 14) | 00 |
| Version PTP (Octet 15) | 02 |

### 24.9.10 GMAC Interrupt Status Register

**Name:** ISR
**Offset:** 0x024
**Reset:** 0x00000000
**Property:** -

This register indicates the source of the interrupt. An interrupt source must be enabled in the mask register first so the corresponding bits of this register will be set and the GMAC interrupt signal will be asserted in the system.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | WOL | | SRI | PDRSFT | PDRQFT |
| Access | | | | R | | R | R | R |
| Reset | | | | 0 | | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | PDRSFR | PDRQFR | SFT | DRQFT | SFR | DRQFR | | |
| Access | R | R | R | R | R | R | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | PFTR | PTZ | PFNZ | HRESP | ROVR | | |
| Access | | R | R | R | R | R | | |
| Reset | | 0 | 0 | 0 | 0 | 0 | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TCOMP | TFC | RLEX | TUR | TXUBR | RXUBR | RCOMP | MFS |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 28 – WOL** Wake On LAN
WOL interrupt. Indicates a WOL message has been received.

**Bit 26 – SRI** TSU Seconds Register Increment
Indicates the register has incremented.

Cleared on read.

**Bit 25 – PDRSFT** PDelay Response Frame Transmitted
Indicates a PTP pdelay_resp frame has been transmitted.

Cleared on read.

**Bit 24 – PDRQFT** PDelay Request Frame Transmitted
Indicates a PTP pdelay_req frame has been transmitted.

Cleared on read.

**Bit 23 – PDRSFR** PDelay Response Frame Received
Indicates a PTP pdelay_resp frame has been received.

### 24.9.81 GMAC TCP Checksum Errors Register

**Name:** TCE
**Offset:** 0x1AC
**Reset:** 0x00000000
**Property:** Read-Only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TCKER[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7:0 – TCKER[7:0]** TCP Checksum Errors
This register counts the number of frames discarded due to an incorrect TCP checksum, but are between 64 and 1518 Bytes (1536 Bytes if NCFGR.MAXFS=1) and do not have a CRC error, an alignment error, nor a symbol error.

### 25.6.7 Safe Flash Update Using Dual Banks

This feature enables a firmware to execute from the NVM and at the same time program the Flash with a new version of itself.

The new firmware has to be programmed in BANKB if STATUS.AFIRST=1, or BANKA otherwise.

After programming is completed one can issue the BKSWRST command to swap the banks and to reset the device. The information of which BANK is mapped to the NVM main address space base address is self contained in the NVM using a special fuse that can be programmed or erased individually. This fuse is managed by the BKSWRST command. STATUS.AFIRST reflects the status of this fuse after Reset. The BKSWRST command is atomic meaning that no fetch in the NVM can occur while executing this command. This command executes with the following steps:

1. Stall AHB interfaces.
2. If PARAM.SEE is ‘1’ and 0<SEESTAT.SBLK<11, the NVMCTRL starts to reallocate the SmartEEPROM data to the first bank. Active SEES remains the same at the end of the reallocation.
3. Is STATUS.AFIRST=1: program the AFIRST fuse (new value=0) otherwise erase it (new value=1)
4. Resets the device, After reset, RSTC RCAUSE indicates that the reset was triggered by the NVMCTRL.

After Reset the new firmware is executed from the last programmed bank.

If the SmartEEPROM is configured, the size of the the reserved space in flash must not exceed the bank size. In other words 2*SEESTAT.SBLK.8192 must be lower than half the NVM size in Bytes. In situations where both the banks contain separate applications (or an application in one bank and a bootloader in the other bank), both the banks must have Flash area reserved for SmartEEPROM. This means that the usable area for code in each bank is "Size of the Bank", that is, the size of the Flash configured for the SmartEEPROM using SBLK Fuse.

### 25.6.8 SmartEEPROM

#### 25.6.8.1 Principle of Operation

The SmartEEPROM feature is provided through the AHB2 interface and makes a portion of the NVM appear like a RAM. 8-bit, 16-bit, 32-bit access is supported.

The SmartEEPROM concept relies on the following NVM physical property: It is always possible to write a '0' in a NVM word, even if this word has been previously programmed - but it is not possible to write a '1' to a bit already programmed (holding a '0').

The algorithm consists of virtually mapping physical portions of the NVM to logical addresses with an indirection mechanism. A physical page is assigned to a virtual page address and is kept as long as no bit has to be flipped from '0' to '1', as this operation requires a full block erase. In case such a transition is required, a new physical page is assigned to the modified virtual page (placed in the Flash area reserved for the SmartEEPROM). Writing the virtual page affects the cycling endurance of the SmartEEPROM.

A region can overlap the SmartEEPROM region (depending on the allocated space for the SmartEEPROM), but SmartEEPROM is independent of the Region Lock Bits.

If NVMCTRL.STATUS.AFIRST contains '1', BANKA is mapped to the NVM main address space base address (0x0000). In this case, SmartEEPROM will be in BANKB. Conversely, when BANKB is mapped to the NVM main address space base address, SmartEEPROM will be in BANKA. Thus, the CPU is not halted when accessing the SmartEEPROM.

#### 25.6.8.2 Address Spaces

The SmartEEPROM address space is divided in two distinct areas:

**Bit 2 – XTALEN** Crystal Oscillator Enable
This bit controls the connections between the I/O pads and the external clock or crystal oscillator XOSCn:

0: External clock connected on XIN. XOUT can be used as general-purpose I/O.

1: Crystal connected to XIN/XOUT.

**Bit 1 – ENABLE** Oscillator Enable
0: The oscillator XOSCn is disabled.

1: The oscillator XOSCn is enabled.

**Protocol T=0**

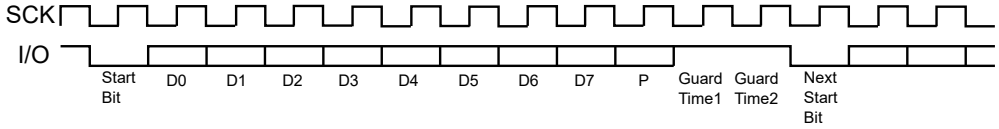
In T=0 protocol, a character is made up of:

- one start bit,
- eight data bits,
- one parity bit
- and one guard time, which lasts two bit times.

The transfer is synchronous (CTRLA.CMODE=1). The transmitter shifts out the bits and does not drive the I/O line during the guard time. Additional guard time can be added by programming the Guard Time (CTRLC.GTIME).

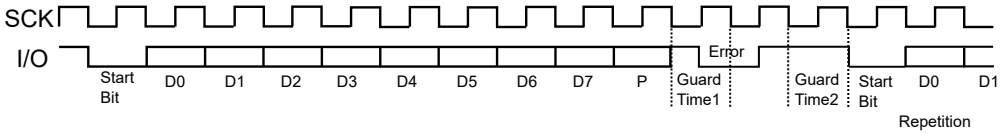
If no parity error is detected, the I/O line remains during the guard time and the transmitter can continue with the transmission of the next character, as shown in the figure below.

**Figure 34-18. T=0 Protocol without Parity Error**




If a parity error is detected by the receiver, it drives the I/O line to 0 during the guard time, as shown in the next figure. This error bit is also named NACK, for Non Acknowledge. In this case, the character lasts 1 bit time more, as the guard time length is the same and is added to the error bit time, which lasts 1 bit time.

**Figure 34-19. T=0 Protocol with Parity Error**




When the USART is the receiver and it detects a parity error, the parity error bit in the Status Register (STATUS.PERR) is set and the character is not written to the receive FIFO.

**Receive Error Counter**

The receiver also records the total number of errors (receiver parity errors and NACKs from the remote transmitter) up to a maximum of 255. This can be read in the Receive Error Count (RXERRCNT) register. RXERRCNT is automatically cleared on read.

**Receive NACK Inhibit**

The receiver can also be configured to inhibit error generation. This can be achieved by setting the Inhibit Not Acknowledge (CTRLC.INACK) bit. If CTRLC.INACK is 1, no error signal is driven on the I/O line even if a parity error is detected. Moreover, if CTRLC.INACK is set, the erroneous received character is stored in the receive FIFO, and the STATUS.PERR bit is set. Inhibit not acknowledge (CTRLC.INACK) takes priority over disable successive receive NACK (CTRLC.DSNACK).

**Transmit Character Repetition**

When the USART is transmitting a character and gets a NACK, it can automatically repeat the character before moving on to the next character. Repetition is enabled by writing the Maximum Iterations register (CTRLC.MAXITER) to a non-zero value. The USART repeats the character the number of times specified in CTRLC.MAXITER.

### 39.8.40 Tx Buffer Cancellation Request

**Name:** TXBCR
**Offset:** 0xD4
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | CRn[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 |
|---|---|---|---|---|---|---|---|---|
| | CRn[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 |
|---|---|---|---|---|---|---|---|---|
| | CRn[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 |
|---|---|---|---|---|---|---|---|---|
| | CRn[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 – CRn[31:0]** Cancellation Request
Each Tx Buffer has its own Cancellation Request bit.

Writing a ‘1’ will set the corresponding Cancellation Request bit; writing a ‘0’ has no impact. This enables the Host to set cancellation requests for multiple Tx Buffers with one write to TXBCR. TXBCR bits are set only for those Tx Buffers configured via TXBC. The bits remain set until the corresponding bit of TXBRP is reset.

| Value | Description |
|---|---|
| 0 | No cancellation pending. |
| 1 | Cancellation pending. |

### 40.8.27 Capabilities 1 Register

**Name:** CA1R
**Offset:** 0x44
**Reset:** 0x00000070
**Property:** -

**Note:** The Capabilities 1 Register is not supposed to be written by the user.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | CLKMULT[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 |
|---|---|---|---|---|---|---|---|---|
| | | DRVDSUP | DRVCSUP | DRVASUP | | DDR50SUP | SDR104SUP | SDR50SUP |
| Access | | R/W | R/W | R/W | | R/W | R/W | R/W |
| Reset | | 0 | 0 | 0 | | 0 | 0 | 0 |

**Bits 23:16 – CLKMULT[7:0]** Clock Multiplier
This field indicates the multiplier factor between the Base Clock (BASECLK) used for the Divided Clock Mode and the Multiplied Clock (MULTCLK) used for the Programmable Clock mode (refer to CCR).

Reading this field to 0 means that the Programmable Clock mode is not supported.

$F_{\mathrm{MULTCLK}} = F_{\mathrm{BASECLK}} \times (\mathrm{CLKMULT}+1)$

**Bit 6 – DRVDSUP** Driver Type D Support

| Value | Description |
|---|---|
| 0 | Driver type D is not supported. |

**Bit 5 – DRVCSUP** Driver Type C Support

| Value | Description |
|---|---|
| 0 | Driver type C is not supported. |

**Bit 4 – DRVASUP** Driver Type A Support

| Value | Description |
|---|---|
| 0 | Driver type A is not supported. |

| Offset | Name | Bit Pos. | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | 23:16 | GHASH[23:16] | | | | | | | |
| | | 31:24 | GHASH[31:24] | | | | | | | |
| 0x70 | GHASH1 | 7:0 | GHASH[7:0] | | | | | | | |
| | | 15:8 | GHASH[15:8] | | | | | | | |
| | | 23:16 | GHASH[23:16] | | | | | | | |
| | | 31:24 | GHASH[31:24] | | | | | | | |
| 0x74 | GHASH2 | 7:0 | GHASH[7:0] | | | | | | | |
| | | 15:8 | GHASH[15:8] | | | | | | | |
| | | 23:16 | GHASH[23:16] | | | | | | | |
| | | 31:24 | GHASH[31:24] | | | | | | | |
| 0x78 | GHASH3 | 7:0 | GHASH[7:0] | | | | | | | |
| | | 15:8 | GHASH[15:8] | | | | | | | |
| | | 23:16 | GHASH[23:16] | | | | | | | |
| | | 31:24 | GHASH[31:24] | | | | | | | |
| 0x7C ... 0x7F | Reserved | | | | | | | | | |
| 80 | CIPLEN | 7:0 | CIPLEN[7:0] | | | | | | | |
| | | 15:8 | CIPLEN[15:8] | | | | | | | |
| | | 23:16 | CIPLEN[23:16] | | | | | | | |
| | | 31:24 | CIPLEN[31:24] | | | | | | | |
| 0x84 | RANDSEED | 7:0 | RANDSEED[7:0] | | | | | | | |
| | | 15:8 | RANDSEED[15:8] | | | | | | | |
| | | 23:16 | RANDSEED[23:16] | | | | | | | |
| | | 31:24 | RANDSEED[31:24] | | | | | | | |

## 42.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to 42.5.8 Register Access Protection.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

| Parameter | Type | Direction | Location | Data Length | Before Executing the Service | After Executing the Service |
|---|---|---|---|---|---|---|
| nu1HashBase (see **Note 1**) | nu1 | I | Crypto RAM | u2ScalarLength + 4 | Base of the hash value resulting from the previous SHA | Corrupted |
| u2ScalarLength | u2 | I | – | – | Length of scalar | Length of scalar |
| nu1PointABase | nu1 | I/O | Crypto RAM | 3*u2ModLength + 12 | Generator point | Corrupted |
| nu1PointPublicKeyGen | nu1 | I/O | Crypto RAM | 3*u2ModLength + 12 | Public point | Corrupted |
| nu1ABase | nu1 | I | Crypto RAM | 2*u2ModLength + 8 | Parameter a and b of the elliptic curve | Unchanged |
| nu1Workspace | nu1 | I | Crypto RAM | 8*u2ModLength + 44 | – | Corrupted workspace |

**Note:**

1. Whatever the chosen SHA, the resulting hash value may have a length inferior or equal to the modulo length and be padded with zeros until its total length is u2ModLength + 4.

#### 43.3.7.10.5 Code Example

```
PUKCL_PARAM PUKCLParam;
PPUKCL_PARAM pvPUKCLParam = &PUKCLParam;

// ! The Random Number Generator must be initialized and started
// ! following the directives given for the RNG on the chip

PUKCL (u2Option) = 0;

// Depending on the option specified, not all fields should be filled PUKCL
_GF2NEcDsaVerify(nu1ModBase) = <Base of the ram location of P>;
PUKCL _GF2NEcDsaVerify(u2ModLength) = <Byte length of P>;
PUKCL _GF2NEcDsaVerify(nu1CnsBase) = <Base of the ram location of Cns>;
PUKCL _GF2NEcDsaVerify(nu1PointABase) = <Base of the A point>;
PUKCL _GF2NEcDsaVerify(nu1PrivateKey) = <Base of the Private Key>;
PUKCL _GF2NEcDsaVerify(nu1ScalarNumber) = <Base of the ScalarNumber>;
PUKCL _GF2NEcDsaVerify(nu1OrderPointBase) = <Base of the order of A point>;
PUKCL _GF2NEcDsaVerify(nu1ABase) = <Base of the a parameter of the curve>; PUKCL
_GF2NEcDsaVerify(nu1Workspace) = <Base of the workspace>;
PUKCL _GF2NEcDsaVerify(nu1HashBase) = <Base of the SHA resulting hash>;
...

// vPUKCL_Process() is a macro command, which populates the service name
// and then calls the library...
vPUKCL_Process(GF2NEcDsaVerifyFast, &PUKCLParam);
if (PUKCL (u2Status) == PUKCL_OK)
            {
            ...
            }
else
            if(PUKCL(u2Status) == PUKCL_WRONG_SIGNATURE)
            {
            ...
```

#### 48.7.3.14 Channel x Compare/Capture Value, 32-bit Mode

**Name:** CCx
**Offset:** 0x1C + x*0x04 [x=0..1]
**Reset:** 0x00000000
**Property:** Write-Synchronized

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | CC[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 |
|---|---|---|---|---|---|---|---|---|
| | CC[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 |
|---|---|---|---|---|---|---|---|---|
| | CC[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 |
|---|---|---|---|---|---|---|---|---|
| | CC[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 – CC[31:0]** Channel x Compare/Capture Value
These bits contain the compare/capture value in 32-bit TC mode. In Match frequency (MFRQ) or Match PWM (MPWM) waveform operation (WAVE.WAVEGEN), the CC0 register is used as a period register.

Writing a '1' to this bit will disable the one-shot operation.

| Value | Description |
|---|---|
| 0 | The TCC will update the counter value on overflow/underflow condition and continue operation. |
| 1 | The TCC will stop counting on the next underflow/overflow condition. |

**Bit 1 – LUPD** Lock Update
This bit controls the update operation of the TCC buffered registers.

When CTRLB.LUPD is cleared, the hardware UPDATE registers with value from their buffered registers is enabled.

This bit has no effect when input capture operation is enabled.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will enable the registers updates on hardware UPDATE condition.

| Value | Description |
|---|---|
| 0 | The CCBx, PERB, PGVB, PGOB, and SWAPBx buffer registers values *are* copied into the corresponding CCx, PER, PGV, PGO and SWAPx registers on hardware update condition. |
| 1 | The CCBx, PERB, PGVB, PGOB, and SWAPBx buffer registers values are *not* copied into the corresponding CCx, PER, PGV, PGO and SWAPx registers on hardware update condition. |

**Bit 0 – DIR** Counter Direction
This bit is used to change the direction of the counter.

Writing a '0' to this bit has no effect

Writing a '1' to this bit will clear the bit and make the counter count up.

| Value | Description |
|---|---|
| 0 | The timer/counter is counting up (incrementing). |
| 1 | The timer/counter is counting down (decrementing). |

| Value | Description |
|---|---|
| 0 | PDEC/HALL decoding is running. |
| 1 | PDEC/HALL decoding is stopped. |

**Bit 5 – HERR** Hall Error Flag
This flag is set when an invalid HALL code is detected.

The flag is cleared by writing a '1' to this bit location.

Outside of HALL mode, this bits is always read '0'.

**Bit 4 – WINERR** Window Error Flag
This flag is set when the counter is outside the window monitor.

The flag is cleared by writing a '1' to this bit location.

Outside of HALL mode, this bits is always read '0'.

**Bit 2 – MPERR** Missing Pulse Error flag
This flag is set when a missing pulse error condition is detected.

The flag is cleared by writing a '1' to this bit location.

Outside of QDEC mode, this bits is always read '0'.

**Bit 1 – IDXERR** Index Error Flag
This flag is set when an index error condition is detected.

The flag is cleared by writing a '1' to this bit location.

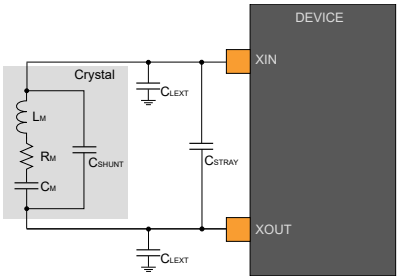
Outside of QDEC mode, this bits is always read '0'.

**Bit 0 – QERR** Quadrature Error Flag
This flag is set when an invalid QDEC transition is detected.

The flag is cleared by writing a '1' to this bit location.

Outside of QDEC mode, this bits is always read '0'.

**Figure 54-6. Oscillator Connection**




The user must choose a crystal oscillator where the crystal load capacitance $C_L$ is within the range given in the Table. The exact value of $C_L$ can be found in the crystal datasheet. The capacitance of the external capacitors ($C_{LEXT}$) can then be computed as follows:

$C_{LEXT} = 2(C_L - C_{STRAY} - SHUNT)$ ,

where $C_{SHUNT}$ is the shunt capacity of the crystal, and $C_{STRAY}$ is the capacitance of the pins and the PCB:

$C_{STRAY} = C_{StrayDevice} + C_{StrayPCB}$, and $1/C_{StrayDevice} = 1/C_{XIN} + 1/C_{XOUT}$.

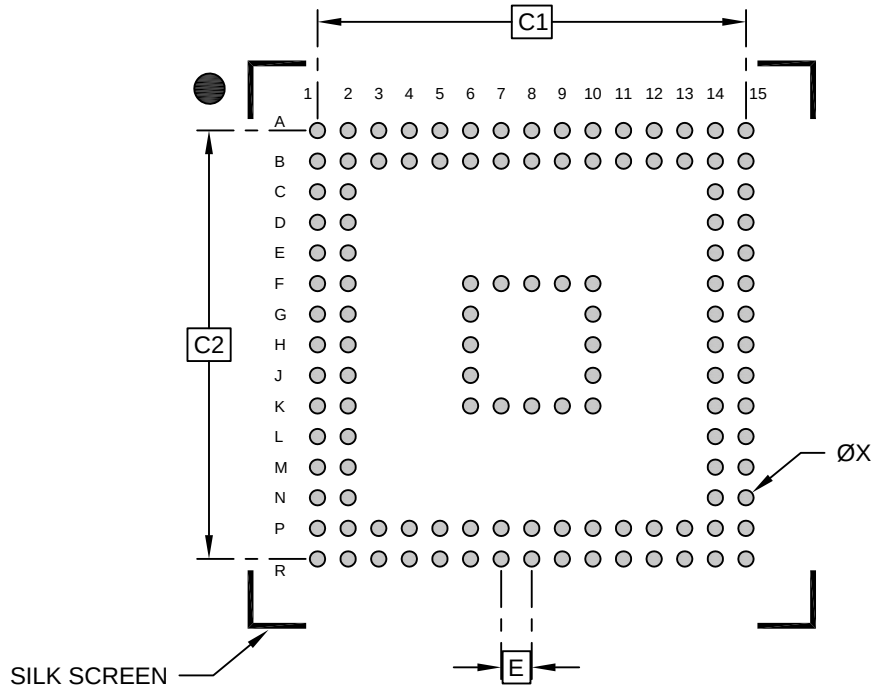
**Table 54-40. Multi-Crystal Oscillator Electrical Characteristics**

| Symbol | Parameter | Conditions | Min. | Typ. | Max. | Units |
|---|---|---|---|---|---|---|
| $F_{OUT}$ | Crystal oscillator frequency | | 8 | - | 48 | MHz |
| $C_L$ | Crystal Load | F = 8 MHz | - | - | 20 | pF |
| | | F = 16 MHz | - | - | 20 | |
| | | F = 32 MHz | - | - | 13 | |
| | | F = 48 MHz | - | - | 13 | |
| ESR | Crystal Equivalent Series Resistance - SF=3 | F = 8 MHz, $C_L$=20 pF - IMULT = 0x3 | - | - | 181 | Ω |
| | | F = 16 MHz, $C_L$ = 20 pF - IMULT = 0x4 | - | - | 180 | |
| | | F= 24 MHz, $C_L$ = 20 pF - IMULT = 0x5 | - | - | 70 | |
| | | F = 48 MHz, $C_L$ = 13 pF - IMULT = 0x6 | - | - | 70 | |
| $C_{XIN}$ | Parasitic load capacitor | - | - | 6.3 | - | pF |
| $C_{XOUT}$ | | - | - | 5.9 | - | |
| $D_L$ | Drive Level (see **Note 1**) | ENALC = ON | - | - | 100 | µW |
| $T_{START}$ | Startup time | F = 8 MHz, $C_L$ = 20 pF, $C_{SHUNT}$ = 2 pF - IMULT = 0x3 | - | 39700 | 72200 | Cycles |
| | | F = 16 MHz, $C_L$ = 20 pF, $C_{SHUNT}$ = 1.5 pF - IMULT = 0x4 | - | 37550 | 62000 | |
| | | F = 24 MHz, $C_L$ = 20 pF, $C_{SHUNT}$ = 2.5 pF - IMULT = 0x5 | - | 32700 | 68500 | |
| | | F = 48 MHz, $C_L$ = 13 pF, $C_{SHUNT}$ = 5 pF - IMULT = 0x6 | - | 18400 | 38500 | |

**Note:** To ensure that the crystal is not overdriven, the automatic loop control is recommended to be turned ON (ENALC = 1).

## 120-Ball Thin Fine Pitch Ball Grid Array Package (DGB) - 4x4 mm Body [TFBGA]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging




RECOMMENDED LAND PATTERN

| Units | | MILLIMETERS | | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Contact Pitch | E | | 0.50 BSC | |
| Contact Pad Spacing | C1 | | 7.00 BSC | |
| Contact Pad Spacing | C2 | | 7.00 BSC | |
| Contact Pad Width (X20) | X | | 0.25 | |

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-23465 Rev A