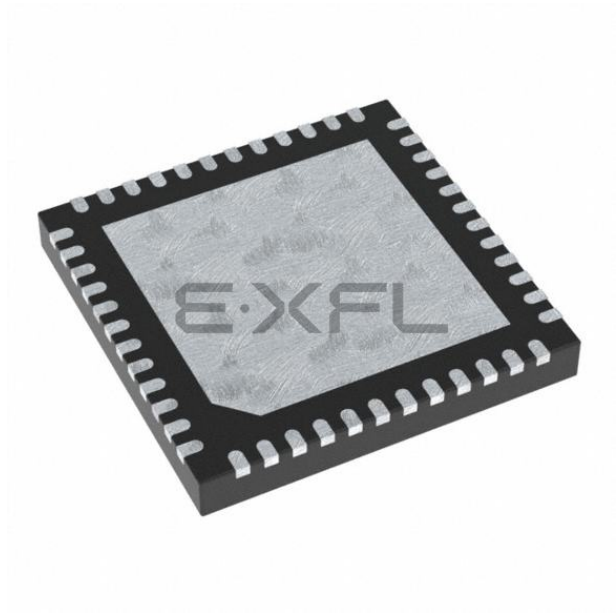


Welcome to E-XFL.COM

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	37
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 20x12b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	48-VFQFN Exposed Pad
Supplier Device Package	48-QFN (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsamd51g19a-mut

Bit 0 – BOD33RDY BOD33 Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the BOD33 Ready Interrupt Enable bit, which disables the BOD33 Ready interrupt.

Value	Description
0	The BOD33 Ready interrupt is disabled.
1	The BOD33 Ready interrupt is enabled, and an interrupt request will be generated when the BOD33 Ready Interrupt flag is set.

21.12.3 Event Control in Clock/Calendar mode (CTRLA.MODE=2)

Name: EVCTRL
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
								TAMPEVEI
Access								R/W
Reset								0

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

Bit	7	6	5	4	3	2	1	0
	PEREOn[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 16 – TAMPEVEI Tamper Event Input Enable

Value	Description
0	Tamper event input is disabled, and incoming events will be ignored.
1	Tamper event input is enabled, and all incoming events will capture the CLOCK value.

Bit 15 – OVFEO Overflow Event Output Enable

Value	Description
0	Overflow event is disabled and will not be generated.
1	Overflow event is enabled and will be generated for every overflow.

Bit 14 – TAMPEREO Tamper Event Output Enable

Value	Description
0	Tamper event output is disabled, and will not be generated
1	Tamper event output is enabled, and will be generated for every tamper input.

Bits 9:8 – ALARMEOn[1:0] Alarm n Event Output Enable [n = 1..0]

Value	Description
0	Alarm n event is disabled and will not be generated.
1	Alarm n event is enabled and will be generated for every compare match.

- The SRAM address of where the descriptor memory section is located must be written to the Description Base Address (BASEADDR) register
- The SRAM address of where the write-back section should be located must be written to the Write-Back Memory Base Address (WRBADDR) register
- Priority level x of the arbiter can be enabled by setting the Priority Level x Enable bit in the Control register (CTRL.LVLENx=1)

DMA Channel Initialization

Before a DMA channel is enabled, the DMA channel and the corresponding first transfer descriptor must be configured, as outlined by the following steps:

- DMA Channel Configuration
  - The channel number of the DMA channel to configure must be written to the Channel Control A register (CHCTRLA) register
  - Trigger action must be selected by writing the Trigger Action bit field in the Channel Control A register (CHCTRLA.TRIGACT)
  - Trigger source must be selected by writing the Trigger Source bit field in the Channel Control A register (CHCTRLA.TRIGSRC)
- Transfer Descriptor
  - The size of each access of the data transfer bus must be selected by writing the Beat Size bit group in the Block Transfer Control register (BTCTRL.BEATSIZE)
  - The transfer descriptor must be made valid by writing a one to the Valid bit in the Block Transfer Control register (BTCTRL.VALID)
  - Number of beats in the block transfer must be selected by writing the Block Transfer Count (BTCNT) register
  - Source address for the block transfer must be selected by writing the Block Transfer Source Address (SRCADDR) register
  - Destination address for the block transfer must be selected by writing the Block Transfer Destination Address (DSTADDR) register

CRC Calculation

If CRC calculation is needed, the CRC engine must be configured before it is enabled, as outlined by the following steps:

- The CRC input source must selected by writing the CRC Input Source bit group in the CRC Control register (CRCCTRL.CRCSRC)
- The type of CRC calculation must be selected by writing the CRC Polynomial Type bit group in the CRC Control register (CRCCTRL.CRCPOLY)
- If I/O is selected as input source, the beat size must be selected by writing the CRC Beat Size bit group in the CRC Control register (CRCCTRL.CRCBEATSIZE)

Register Properties

The following DMAC registers are enable-protected, meaning that they can only be written when the DMAC is disabled (CTRL.DMAENABLE=0):

- Descriptor Base Memory Address register (BASEADDR)
- Write-Back Memory Base Address register (WRBADDR)

Value	Name	Description
0x2A	CH10	DMA channel 10
0x2B	CH11	DMA channel 11
0x2C	CH12	DMA channel 12
0x2D	CH13	DMA channel 13
0x2E	CH14	DMA channel 14
0x2F	CH15	DMA channel 15
0x30	CH16	DMA channel 16
0x31	CH17	DMA channel 17
0x32	CH18	DMA channel 18
0x33	CH19	DMA channel 19
0x34	CH20	DMA channel 20
0x35	CH21	DMA channel 21
0x36	CH22	DMA channel 22
0x37	CH23	DMA channel 23
0x38	CH24	DMA channel 24
0x39	CH25	DMA channel 25
0x3A	CH26	DMA channel 26
0x3B	CH27	DMA channel 27
0x3C	CH28	DMA channel 28
0x3D	CH29	DMA channel 29
0x3E	CH30	DMA channel 30
0x3F	CH31	DMA channel 31

Bits 3:2 – CRCPOLY[1:0] CRC Polynomial Type
These bits select the CRC polynomial type.

Value	Name	Description
0x0	CRC16	CRC-16 (CRC-CCITT)
0x1	CRC32	CRC32 (IEEE 802.3)
0x2-0x3		Reserved

Bits 1:0 – CRCBEATSIZE[1:0] CRC Beat Size
These bits define the size of the data transfer for each bus access when the CRC is used with I/O interface.

Value	Name	Description
0x0	BYTE	8-bit bus transfer
0x1	HWORD	16-bit bus transfer
0x2	WORD	32-bit bus transfer
0x3		Reserved

22.8.10 Interrupt Status

Name: INTSTATUS
Offset: 0x24
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	CHINTn[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
	CHINTn[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
	CHINTn[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
	CHINTn[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CHINTn[31:0] Channel n Pending Interrupt [n=31..0]
This bit is set when Channel n has a pending interrupt/the interrupt request is received.

This bit is cleared when the corresponding Channel n interrupts are disabled or the interrupts sources are cleared.

24.9.56 GMAC Deferred Transmission Frames Register

Name: DTF
Offset: 0x148
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
							DEFT[17:16]	
Access							R	R
Reset							0	0

Bit	15	14	13	12	11	10	9	8
	DEFT[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
	DEFT[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 17:0 – DEFT[17:0] Deferred Transmission
This register counts the number of frames experiencing deferral due to carrier sense being active on their first attempt at transmission. Frames involved in any collision are not counted nor are frames that experienced a transmit underrun.

24.9.80 GMAC IP Header Checksum Errors Register

Name: IHCE
Offset: 0x1A8
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								

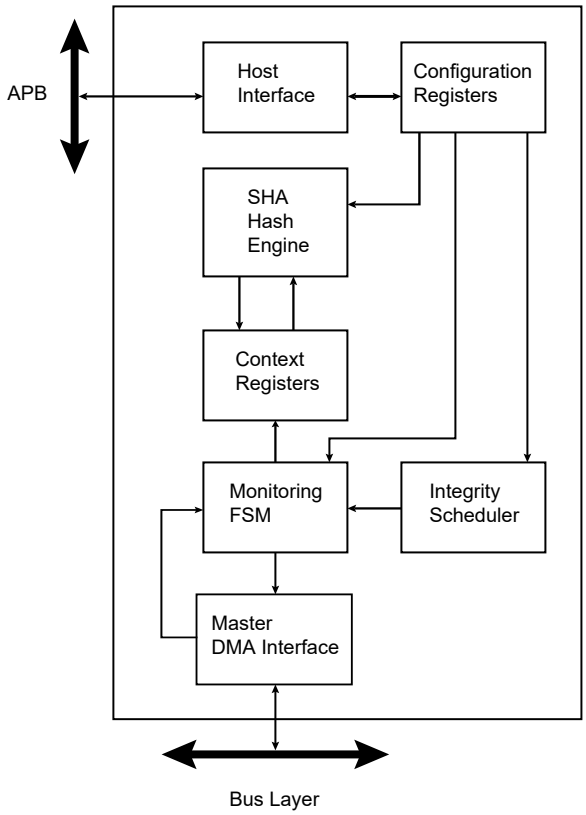
Bit	15	14	13	12	11	10	9	8
Access								
Reset								

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

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

26.3 Block Diagram

Figure 26-1. Integrity Check Monitor Block Diagram



26.4 Signal Description

Not applicable.

26.5 Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

26.5.1 Power Management

The ICM will run only when the source clocks are running, i.e. when the CPU is in Active mode.

26.5.2 Clocks

The ICM bus clocks (CLK_ICM_AHB and CLK_ICM_APB) can be enabled and disabled in the Main Clock module (MCLK) by writing the respective bit in the mask registers (MCLK.AHBMASK.ICM and MCLK.APBCMASK.ICM).

The default states of CLK_ICM_AHB and CLK_ICM_APB are given by the reset values of the respective mask registers.

Related Links

15.7 Register Summary

26.6.3.1 Region Descriptor Structure Overview

Offset	Name	Bit Pos.								
0x00	RADDR0	7:0	RADDR[7:0]							
		15:8	RADDR[15:8]							
		23:16	RADDR[23:16]							
		31:24	RADDR[31:24]							
0x04	RCFG0	7:0	WCIEN	BEIEN	DMIEN	RHIEN		EOM	WRAP	CDWBN
		15:8		ALGO[2:0]				PROCDLY	SUIEN	ECIEN
		23:16								
		31:24								
0x08	RCTRL0	7:0	TRSIZE[7:0]							
		15:8	TRSIZE[15:8]							
		23:16								
		31:24								
0x0C	RADDR1	7:0	RADDR[7:0]							
		15:8	RADDR[15:8]							
		23:16	RADDR[23:16]							
		31:24	RADDR[31:24]							
0x0C	RNEXT0	7:0								
		15:8								
		23:16								
		31:24								
0x10	RCFG1	7:0	WCIEN	BEIEN	DMIEN	RHIEN		EOM	WRAP	CDWBN
		15:8		ALGO[2:0]				PROCDLY	SUIEN	ECIEN
		23:16								
		31:24								
0x14	RCTRL1	7:0	TRSIZE[7:0]							
		15:8	TRSIZE[15:8]							
		23:16								
		31:24								
0x18	RADDR2	7:0	RADDR[7:0]							
		15:8	RADDR[15:8]							
		23:16	RADDR[23:16]							
		31:24	RADDR[31:24]							
0x18	RNEXT1	7:0								
		15:8								
		23:16								
		31:24								
0x1C	RCFG2	7:0	WCIEN	BEIEN	DMIEN	RHIEN		EOM	WRAP	CDWBN
		15:8		ALGO[2:0]				PROCDLY	SUIEN	ECIEN
		23:16								
		31:24								
0x20	RCTRL2	7:0	TRSIZE[7:0]							
		15:8	TRSIZE[15:8]							
		23:16								
		31:24								

28. OSCCTRL – Oscillators Controller

28.1 Overview

The Oscillators Controller (OSCCTRL) provides a user interface to the XOSCn, DFLL48M, and two FDPLL200M.

Through the interface registers, it is possible to enable, disable, calibrate, and monitor the oscillators.

The status of all oscillators are collected in the Status register (STATUS). They can additionally trigger interrupts upon status changes via the INTENSET, INTENCLR, and INTFLAG registers.

28.2 Features

- Digital Frequency-Locked Loop (DFLL48M)
  - Internal oscillator with no external components
  - 48 MHz output frequency
  - Operates stand-alone as a high-frequency programmable oscillator in Open Loop mode
  - Operates as an accurate frequency multiplier against a known frequency in Closed Loop mode
- Two 8-48 MHz Crystal Oscillators (XOSCn)
  - Tunable gain control
  - Programmable start-up time
  - Crystal or external input clock on XIN I/O
  - Clock failure detection with safe clock switch
  - Clock failure event output
- Two Digital Phase-Locked Loop (DPLLn)
  - 96 MHz to 200 MHz output frequency from a 32 kHz to 3.2 MHz reference clock
  - Two DPLLs, each with four selectable reference clocks
  - Adjustable digital filter for jitter optimization
  - Adjustable DCO filter for a 4-stages differential ring oscillator
  - Fractional part used to achieve 1/32th of reference clock step
  - Embedded test mode controller

31.6 Register Summary

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0								SWRST
0x01 ... 0x03	Reserved									
0x04	SWEVT	7:0	CHANNEL[7:0]							
		15:8	CHANNEL[15:8]							
		23:16	CHANNEL[23:16]							
		31:24	CHANNEL[31:24]							
0x08	PRICTRL	7:0	RREN			PRI[4:0]				
0x09 ... 0x0F	Reserved									
0x10	INTPEND	7:0				ID[4:0]				
		15:8	BUSY	READY					EVD	OVR
0x12 ... 0x13	Reserved									
0x14	INTSTATUS	7:0	CHINT7	CHINT6	CHINT5	CHINT4	CHINT3	CHINT2	CHINT1	CHINT0
		15:8					CHINT11	CHINT10	CHINT9	CHINT8
		23:16								
		31:24								
0x18	BUSYCH	7:0	BUSYCHx7	BUSYCHx6	BUSYCHx5	BUSYCHx4	BUSYCHx3	BUSYCHx2	BUSYCHx1	BUSYCHx0
		15:8					BUSYCHx11	BUSYCHx10	BUSYCHx9	BUSYCHx8
		23:16								
		31:24								
0x1C	READYUSR	7:0	READYUSR7	READYUSR6	READYUSR5	READYUSR4	READYUSR3	READYUSR2	READYUSR1	READYUSR0
		15:8					READYUSR11	READYUSR10	READYUSR9	READYUSR8
		23:16								
		31:24								
0x20	CHANNEL0	7:0	EVGEN[7:0]							
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0x24	CHINTENCLR0	7:0							EVD	OVR
0x25	CHINTENSET0	7:0							EVD	OVR
0x26	CHINTFLAG0	7:0							EVD	OVR
0x27	CHSTATUS0	7:0							BUSYCH	RDYUSR
0x28	CHANNEL1	7:0	EVGEN[7:0]							
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0x2C	CHINTENCLR1	7:0							EVD	OVR

Value	Description
0	The corresponding I/O pin in the PORT group will keep its configuration.
1	The corresponding I/O pin output is driven low, or the input is connected to an internal pull-down.

1. Abort the current transfer.
2. Flush the transmit buffer.
3. Disable transmitter (CTRLB.TXEN=0)
   - This is done after a synchronization delay. The CTRLB Synchronization Busy bit (SYNCBUSY.CTRLB) will be set until this is complete.
   - After disabling, the TxD pin will be tri-stated.
4. Set the Collision Detected bit (STATUS.COLL) along with the Error interrupt flag (INTFLAG.ERROR).
5. Set the Transmit Complete interrupt flag (INTFLAG.TXC), since the transmit buffer no longer contains data.

After a collision, software must manually enable the transmitter again before continuing, after assuring that the CTRLB Synchronization Busy bit (SYNCBUSY.CTRLB) is not set.

34.6.3.9 Loop-Back Mode

For loop-back mode, configure the Receive Data Pinout (CTRLA.RXPO) and Transmit Data Pinout (CTRLA.TXPO) to use the same data pins for transmit and receive. The loop-back is through the pad, so the signal is also available externally.

34.6.3.10 Start-of-Frame Detection

The USART start-of-frame detector can wake up the CPU when it detects a start bit. In standby sleep mode, the internal fast startup oscillator must be selected as the GCLK_SERCOMx_CORE source.

When a 1-to-0 transition is detected on RxD, the 8MHz Internal Oscillator is powered up and the USART clock is enabled. After startup, the rest of the data frame can be received, provided that the baud rate is slow enough in relation to the fast startup internal oscillator start-up time. Refer to *Electrical Characteristics* for details. The start-up time of this oscillator varies with supply voltage and temperature.

The USART start-of-frame detection works both in asynchronous and synchronous modes. It is enabled by writing ‘1’ to the Start of Frame Detection Enable bit in the Control B register (CTRLB.SFDE).

If the Receive Start Interrupt Enable bit in the Interrupt Enable Set register (INTENSET.RXS) is set, the Receive Start interrupt is generated immediately when a start is detected.

When using start-of-frame detection without the Receive Start interrupt, start detection will force the 8MHz Internal Oscillator and USART clock active while the frame is being received. In this case, the CPU will not wake up until the Receive Complete interrupt is generated.

34.6.3.11 Sample Adjustment

In asynchronous mode (CTRLA.CMODE=0), three samples in the middle are used to determine the value based on majority voting. The three samples used for voting can be selected using the Sample Adjustment bit field in Control A register (CTRLA.SAMPA). When CTRLA.SAMPA=0, samples 7-8-9 are used for 16x oversampling, and samples 3-4-5 are used for 8x oversampling.

34.6.3.12 32-bit Extension

For better system bus utilization, 32-bit data receive and transmit can be enabled separately by writing to the Data 32-bit bit field in the Control C register (CTRLC.DATA32B). When enabled, writes and/or reads to the DATA register are 32 bit in size.

If frames are not multiples of 4 Bytes, the length counter (LENGTH.LEN) and length enable (LENGTH.LENEN) must be configured before data transfer begins, LENGTH.LEN must be enabled only when CTRLC.DATA32B is enabled.

The figure below shows the order of transmit and receive when using 32-bit extension. Bytes are transmitted or received, and stored in order from 0 to 3. Only 8-bit and smaller character sizes are

The CCL can take the following actions on an input event:

- INSELx: The event is used as input for the TRUTH table. For further details refer to 41.5.6 Events.

Writing a '1' to the LUT Control Event Input Enable bit (LUTCTRL.LUTEI) enables the corresponding action on input event. Writing a '0' to this bit disables the corresponding action on input event.

Related Links

31. EVSYS – Event System

41.6.4 Sleep Mode Operation

When using the GCLK_CCL internal clocking, writing the Run In Standby bit in the Control register (CTRL.RUNSTDBY) to '1' will allow GCLK_CCL to be enabled in Standby Sleep mode.

If CTRL.RUNSTDBY=0, the GCLK_CCL will be disabled in Standby Sleep mode. If the Filter, Edge Detector or Sequential logic are enabled, the LUT output will be forced to zero in STANDBY mode. In all other cases, the TRUTH table decoder will continue operation and the LUT output will be refreshed accordingly.

Related Links

18. PM – Power Manager

42.5.7 Debug Operation

When the CPU is halted in debug mode, the AES module continues normal operation. If the AES module is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging. The AES module can be forced to halt operation during debugging.

42.5.8 Register Access Protection

All registers with write-access are optionally write-protected by the peripheral access controller (PAC), except the

following register:

- Interrupt Flag Register (INTFLAG)

Write-protection is denoted by the Write-Protected property in the register description.

Write-protection does not apply to accesses through an external debugger. Refer to *PAC - Peripheral Access Controller* chapter for details.

Related Links

27. PAC - Peripheral Access Controller

42.5.9 Analog Connections

Not applicable.

42.6 Functional Description

42.6.1 Principle of Operation

The following is a high level description of the algorithm. These are the steps:

- KeyExpansion: Round keys are derived from the cipher key using Rijndael's key schedule.
- InitialRound:
  - AddRoundKey: Each byte of the state is combined with the round key using bitwise XOR.
- Rounds:
  - SubBytes: A non-linear substitution step where each byte is replaced with another according to a lookup table.
  - ShiftRows: A transposition step where each row of the state is shifted cyclically a certain number of steps.
  - MixColumns: A mixing operation which operates on the columns of the state, combining the four bytes in each column.
  - AddRoundKey
- Final Round (no MixColumns):
  - SubBytes
  - ShiftRows
  - AddRoundKey

The relationship between the module's clock frequency and throughput (in bytes per second) is given by:

Clock Frequency = (Throughput/2) x (Nr+1) for 2 byte parallel processing

Clock Frequency = (Throughput/4) x (Nr+1) for 4 byte parallel processing

	system RAM with no part in PUKCC RAM this can be signaled by using this option. In all other cases this option must not be used.

43.3.6.13.6 Code Example

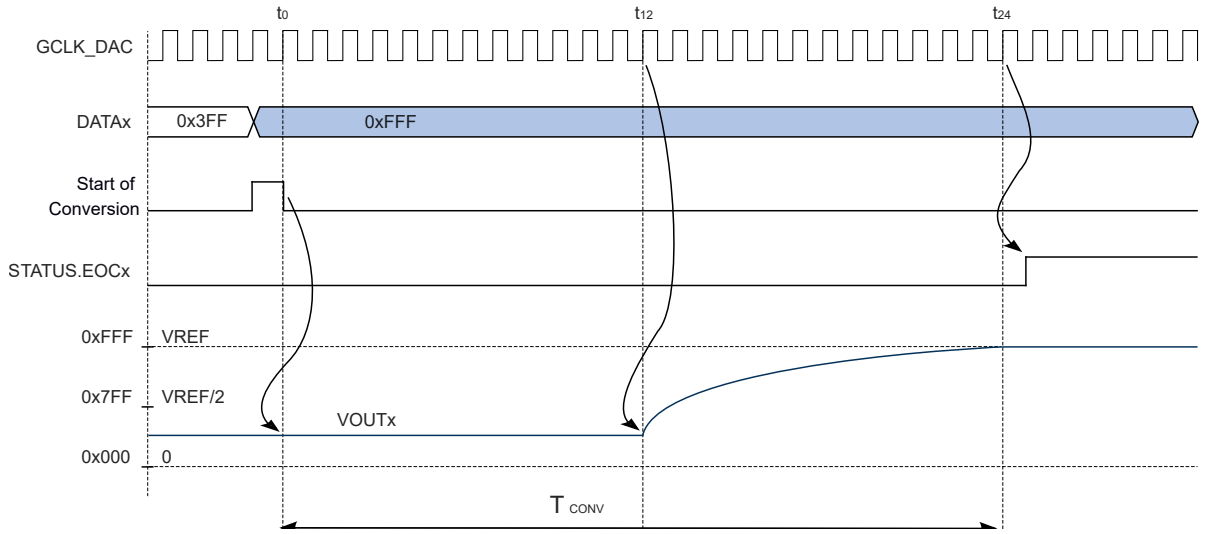
```
PUKCL_PARAM PUKCLParam;
PPUKCL_PARAM pvPUKCLParam = &PUKCLParam;
PUKCL(u2Option) = <Point Signature location and windows sizes>;
PUKCL_ZpEcDsaQuickVerify(pu1ModCnsBase) = <Base of the ram location of P and Cns>;
PUKCL_ZpEcDsaQuickVerify(u2ModLength) = <Byte length of P>;
PUKCL_ZpEcDsaQuickVerify(pu1PointABase) = <Base of the ram location of the A point>;
PUKCL_ZpEcDsaQuickVerify(pu1PointPublicKeyGen) = <Base of the Public Key>;
PUKCL_ZpEcDsaQuickVerify(pu1PointSignature) = <Base of the Signature (r, s)>;
PUKCL_ZpEcDsaQuickVerify(pu1OrderPointBase) = <Base of the order of the A point>;
PUKCL_ZpEcDsaQuickVerify(pu1AWorkBase) = <Base of the ram location of the parameter A of the
elliptic curve and workspace>;
PUKCL_ZpEcDsaQuickVerify(pu1HashBase) = <Base of the SHA resulting hash>;
PUKCL_ZpEcDsaQuickVerify(u2ScalarLength) = <Byte length of R and S in Point Signature>;
. . .
// vPUKCL_Process() is a macro command, which populates the service name
// and then calls the library...
vPUKCL_Process(ZpEcDsaQuickVerify, pvPUKCLParam);
if (PUKCL(u2Status) == PUKCL_OK)
                {
                ...
                }
else
                if ( PUKCL(u2Status) = PUKCL_WRONG_SIGNATURE )
                                {
                                ...
                                }
                else // Manage the error
```

43.3.6.13.7 Constraints

No overlapping between either input and output are allowed. The following conditions must be avoided to ensure that the service works correctly:

- pu1ModCnsBase, pu1PointABase, pu1PointPublicKeyGen, pu1PointSignature,pu1OrderPointBase, pu1AWorkBase or pu1HashBase are not aligned on 32-bit boundaries
- {pu1ModCnsBase, u2ModLength + 4 + u2MaxLength + 12}, {pu1PointABase, (3 * u2ModLength + 12)* ($2^{(WA-2)}$)}, {pu1PointPublicKeyGen, (3 * u2ModLength + 12) * ($2^{(WPub-2)}$)}, {pu1OrderPointBase, u2ScalarLength + 4}, {nu1ABase, u2ModLength + 4}, {pu1AWorkBase, (u2ModLength + 4) + (8 * u2MaxLength + 44)} or {nu1HashBase, u2ScalarLength + 4} are not in Crypto RAM
- u2ModLength is either: < 12, > 0xffc or not a 32-bit length
- All overlapping between {pu1ModCnsBase, u2ModLength + 4 + u2MaxLength + 12}, {pu1PointABase, (3 * u2ModLength + 12) * ($2^{(WA-2)}$)}, {pu1PointPublicKeyGen, (3 * u2ModLength + 12) *($2^{(WPub-2)}$)}, {pu1OrderPointBase, u2ScalarLength + 4}, {pu1PointSignature, 2 * u2ScalarLength + 8}, {nu1ABase, u2ModLength + 4}, {pu1AWorkBase, (u2ModLength + 4) + (8 * u2MaxLength + 44)} and {nu1HashBase, u2ScalarLength + 4}

Figure 47-2. Single DAC Conversion

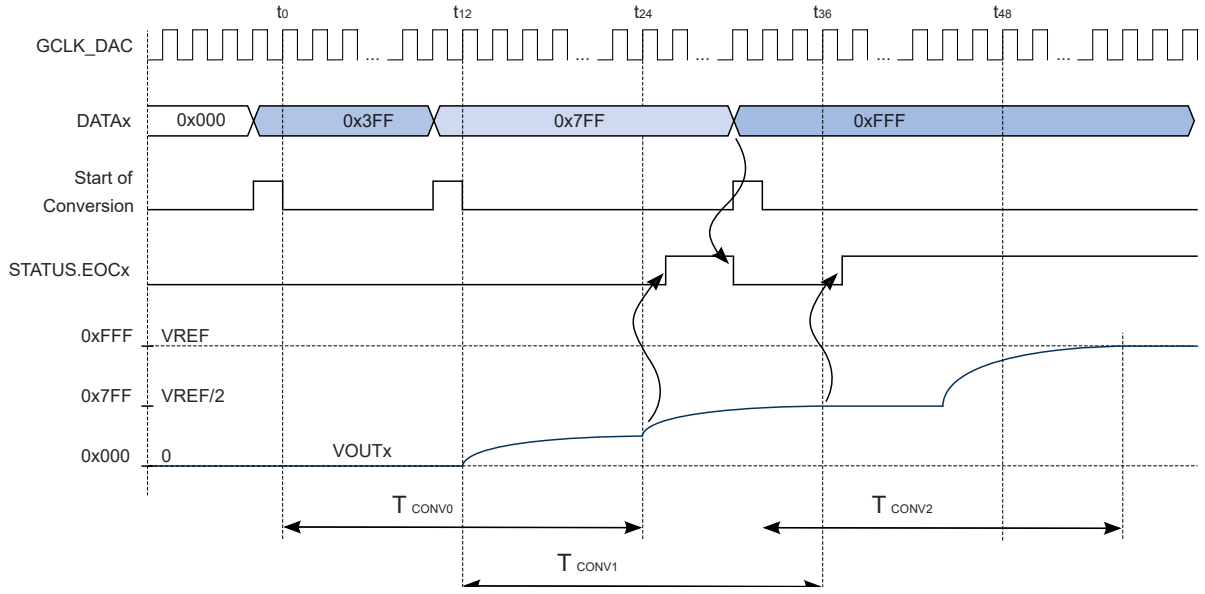


Since the DAC conversion is implemented as pipelined procedure, a new conversion can be started after only 12 GCLK_DAC periods. Therefore if DATAx is written while a conversion is ongoing, start of conversion is postponed until DACx is ready to start next conversion.

The maximum conversion rate (samples per second) is therefore:

$$\mathrm{CR}_{\max} = \frac{2}{T_{\mathrm{conv}}}$$

Figure 47-3. Multiple DAC Conversions



Related Links

19. SUPC – Supply Controller

47.6.3 Operating Conditions

- The DAC voltage reference must be below VDDANA.
- The maximum conversion rate of 1MSPS can be achieved only if VDDANA is above 2.4V.

Value	Description
0	Non-recoverable fault tri-state the output.
1	Non-recoverable faults set the output to NRVx level.

52. PCC - Parallel Capture Controller

52.1 Overview

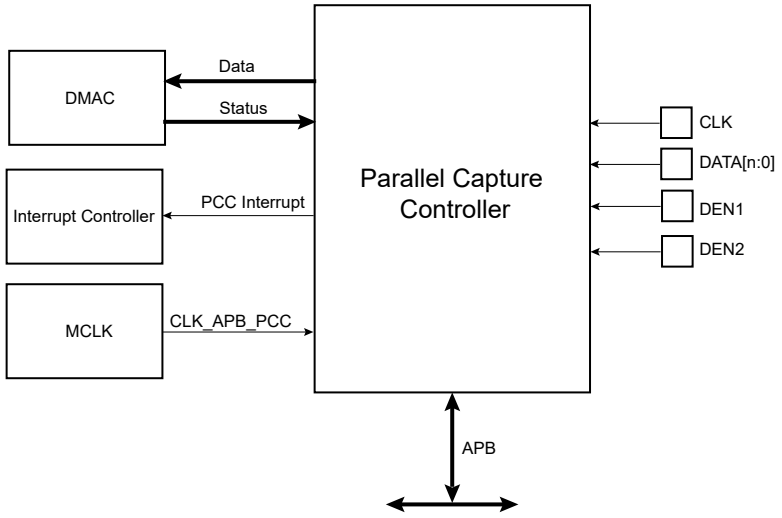
The Parallel Capture Controller can be used to interface an external system, such as a CMOS digital image sensor, ADC, or DSP, and capture its parallel data.

52.2 Features

- One clock, up to 14-bit parallel data and two Data Enable on I/O lines
- Data can be sampled every other time (e.g. for chrominance sampling)
- Supports connection of the DMAC which offers buffer reception without processor intervention
- Auto-scale feature available when 10, 12 or 14 bits data size is selected.
- Can be used to interface a CMOS Digital Image Sensor, an ADC, etc.

52.3 Block Diagram

Figure 52-1. Block Diagram



52.4 Signal Description

Signal	Description	Type
CLK	Digital input	PCC Clock
DATA[n:0]	Digital input	Data [n:0]
DEN1	Digital input	Data Enable 1
DEN2	Digital input	Data Enable 2