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®-M7
Core Size	32-Bit Single-Core
Speed	300MHz
Connectivity	CANbus, Ethernet, I²C, IrDA, LINbus, MMC/SD/SDIO, QSPI, SPI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, I²S, POR, PWM, WDT
Number of I/O	75
Program Memory Size	2MB (2M x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	384K x 8
Voltage - Supply (Vcc/Vdd)	1.62V ~ 3.6V
Data Converters	A/D 10x12b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	100-LQFP
Supplier Device Package	100-LQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsame70n21a-an

21.3.1 Chip ID Register

Name: CHIPID_CIDR
Offset: 0x0
Reset: -
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	EXT	NVPTYP[2:0]			ARCH[7:4]			
Access	R	R	R	R	R	R	R	R
Reset								

Bit	23	22	21	20	19	18	17	16
	ARCH[3:0]				SRAMSIZ[3:0]			
Access	R	R	R	R	R	R	R	R
Reset								

Bit	15	14	13	12	11	10	9	8
	NVPSIZ2[3:0]				NVPSIZ[3:0]			
Access	R	R	R	R	R	R	R	R
Reset								

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

Bit 31 – EXT Extension Flag

Value	Description
0	Chip ID has a single register definition without extension.
1	An extended Chip ID exists.

Bits 30:28 – NVPTYP[2:0] Nonvolatile Program Memory Type

Value	Name	Description
0	ROM	ROM
1	ROMLESS	ROMless or on-chip Flash
2	FLASH	Embedded Flash Memory
3	ROM_FLASH	ROM and Embedded Flash Memory – NVPSIZ is ROM size – NVPSIZ2 is Flash size
4	SRAM	SRAM emulating ROM

Bits 27:20 – ARCH[7:0] Architecture Identifier

Clock Name	Peripheral
PCK5	MCANx
PCK6	TCx
PCK7	TC0

Note: USB, GMAC and MLB do not require PCKx to operate independently of core and bus peripherals.

31.9 Peripheral and Generic Clock Controller

The PMC controls the clocks of the embedded peripherals by means of the Peripheral Control register (PMC_PCR). With this register, the user can enable and disable the different clocks used by the peripherals:

- Peripheral clocks (periph_clk[PID]), routed to every peripheral and derived from the master clock (MCK), and
- Generic clocks (GCLK[PID]), routed to I2SC0 and I2SC1. These clocks are independent of the core and bus clocks (HCLK, MCK and periph_clk[PID]). They are generated by selection and division of the following sources: SLCK, MAINCK, UPLLCKDIV, PLLACK and MCK. Refer to the description of each peripheral for the limitation to be applied to GCLK[PID] compared to periph_clk[PID].

To configure a peripheral's clocks, PMC_PCR.CMD must be written to '1' and PMC_PCR.PID must be written with the index of the corresponding peripheral. All other configuration fields must be correctly set.

To read the current clock configuration of a peripheral, PMC_PCR.CMD must be written to '0' and PMC_PCR.PID must be written with the index of the corresponding peripheral regardless of the values of other fields. This write does not modify the configuration of the peripheral. The PMC_PCR can then be read to know the configuration status of the corresponding PID.

The user can also enable and disable these clocks by configuring the Peripheral Clock Enable (PMC_PCERx) and Peripheral Clock Disable (PMC_PCDRx) registers. The status of the peripheral clock activity can be read in the Peripheral Clock Status registers (PMC_PCSRx).

When a peripheral or a generic clock is disabled, it is immediately stopped. These clocks are disabled after a reset.

To stop a peripheral clock, it is recommended that the system software wait until the peripheral has executed its last programmed operation before disabling the clock. This is to avoid data corruption or erroneous behavior of the system.

The bit number in PMC_PCERx, PMC_PCDRx, and PMC_PCSRx is the Peripheral Identifier defined at the product level. The bit number corresponds to the interrupt source number assigned to the peripheral.

31.10 Asynchronous Partial Wakeup

31.10.1 Description

The asynchronous partial wakeup wakes up a peripheral in a fully asynchronous way when activity is detected on the communication line. The asynchronous partial wakeup function automatically manages the peripheral clock. It reduces overall power consumption of the system by clocking peripherals only when needed.

Asynchronous partial wakeup can be enabled in Wait mode (SleepWalking), or in Active mode.

31.20.18 PMC Fast Startup Mode Register

Name: PMC_FSMR
Offset: 0x0070
Reset: 0x00000000
Property: Read/Write

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

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	FFLPM	FLPM[1:0]		LPM		USBAL	RTCAL	RTTAL
Access								
Reset	0	0	0	0		0	0	0

Bit	15	14	13	12	11	10	9	8
	FSTT15	FSTT14	FSTT13	FSTT12	FSTT11	FSTT10	FSTT9	FSTT8
Access								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	FSTT7	FSTT6	FSTT5	FSTT4	FSTT3	FSTT2	FSTT1	FSTT0
Access								
Reset	0	0	0	0	0	0	0	0

Bit 23 – FFLPM Force Flash Low-power Mode

Value	Description
0	The Flash Low-power mode, defined in the FLPM field, is automatically applied when in Wait mode and released when going back to Active mode.
1	The Flash Low-power mode is user defined by the FLPM field and immediately applied.

Bits 22:21 – FLPM[1:0] Flash Low-power Mode

Value	Name	Description
0	FLASH_STANDBY	Flash is in Standby Mode when system enters Wait Mode
1	FLASH_DEEP_POWERDOWN	Flash is in Deep-powerdown mode when system enters Wait Mode
2	FLASH_IDLE	Idle mode

Bit 20 – LPM Low-power Mode

Value	Description
0	The WaitForInterrupt (WFI) or the WaitForEvent (WFE) instruction of the processor makes the processor enter Sleep mode.
1	The WaitForEvent (WFE) instruction of the processor makes the system enter Wait mode.

32.6.1.52 PIO Parallel Capture Interrupt Disable Register

Name: PIO_PCIDR
Offset: 0x0158
Property: Write-only

The following configuration values are valid for all listed bit names of this register:

0: No effect

1: Disables the corresponding interrupt

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
Access					RXBUFF	ENDRX	OVRE	DRDY
Reset								

Bit 3 – RXBUFF Reception Buffer Full Interrupt Disable

Bit 2 – ENDRX End of Reception Transfer Interrupt Disable

Bit 1 – OVRE Parallel Capture Mode Overrun Error Interrupt Disable

Bit 0 – DRDY Parallel Capture Mode Data Ready Interrupt Disable

36.9.24 XDMAC Channel x Next Descriptor Address Register [x = 0..23]

Name: XDMAC_CNDA
Offset: 0x68 + n*0x40 [n=0..23]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	NDA[29:22]							
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
	NDA[21:14]							
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
	NDA[13:6]							
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
	NDA[5:0]							NDAIF
Access	R/W	R/W	R/W	R/W	R/W	R/W		R/W
Reset	0	0	0	0	0	0		0

Bits 31:2 – NDA[29:0] Channel x Next Descriptor Address
The 30-bit width of the NDA field represents the next descriptor address range 31:2. The descriptor is word-aligned and the two least significant register bits 1:0 are ignored.

Bit 0 – NDAIF Channel x Next Descriptor Interface

Value	Description
0	The channel descriptor is retrieved through system interface 0.
1	The channel descriptor is retrieved through system interface 1.

38.8.35 GMAC PTP Event Frame Received Seconds High Register

Name: GMAC_EFRSH
Offset: 0x0EC
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
	RUD[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
	RUD[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – RUD[15:0] Register Update
The register is updated with the value that the IEEE 1588 timer seconds register held when the SFD of a PTP transmit primary event crosses the MII interface. An interrupt is issued when the register is updated.

41. Serial Peripheral Interface (SPI)

41.1 Description

The Serial Peripheral Interface (SPI) circuit is a synchronous serial data link that provides communication with external devices in Master or Slave mode. It also enables communication between processors if an external processor is connected to the system.

The Serial Peripheral Interface is essentially a shift register that serially transmits data bits to other SPIs. During a data transfer, one SPI system acts as the “master”' which controls the data flow, while the other devices act as “slaves" which have data shifted into and out by the master. Different CPUs can take turn being masters (multiple master protocol, contrary to single master protocol where one CPU is always the master while all of the others are always slaves). One master can simultaneously shift data into multiple slaves. However, only one slave can drive its output to write data back to the master at any given time.

A slave device is selected when the master asserts its NSS signal. If multiple slave devices exist, the master generates a separate slave select signal for each slave (NPCS).

The SPI system consists of two data lines and two control lines:

- Master Out Slave In (MOSI)—This data line supplies the output data from the master shifted into the input(s) of the slave(s).
- Master In Slave Out (MISO)—This data line supplies the output data from a slave to the input of the master. There may be no more than one slave transmitting data during any particular transfer.
- Serial Clock (SPCK)—This control line is driven by the master and regulates the flow of the data bits. The master can transmit data at a variety of baud rates; there is one SPCK pulse for each bit that is transmitted.
- Slave Select (NSS)—This control line allows slaves to be turned on and off by hardware.

41.2 Embedded Characteristics

- Master or Slave Serial Peripheral Bus Interface
  - 8-bit to 16-bit programmable data length per chip select
  - Programmable phase and polarity per chip select
  - Programmable transfer delay between consecutive transfers and delay before SPI clock per chip select
  - Programmable delay between chip selects
  - Selectable mode fault detection
- Master Mode can Drive SPCK up to Peripheral Clock
- Master Mode Bit Rate can be Independent of the Processor/Peripheral Clock
- Slave Mode Operates on SPCK, Asynchronously with Core and Bus Clock
- Four Chip Selects with External Decoder Support Allow Communication with up to 15 Peripherals
- Communication with Serial External Devices Supported
  - Serial memories, such as DataFlash and 3-wire EEPROMs
  - Serial peripherals, such as ADCs, DACs, LCD controllers, CAN controllers and sensors
  - External coprocessors
- Connection to DMA Channel Capabilities, Optimizing Data Transfers

PCS = 1111 forbidden (no peripheral is selected)

(x = don't care)

If SPI_MR.PCSDEC = 1:

NPCS[3:0] output signals = PCS.

Bit 7 – LLB Local Loopback Enable
LLB controls the local loopback on the data shift register for testing in Master mode only (MISO is internally connected on MOSI).

Value	Description
0	Local loopback path disabled.
1	Local loopback path enabled.

Bit 5 – WDRBT Wait Data Read Before Transfer

Value	Description
0	No Effect. In Master mode, a transfer can be initiated regardless of SPI_RDR state.
1	In Master mode, a transfer can start only if SPI_RDR is empty, i.e., does not contain any unread data. This mode prevents overrun error in reception.

Bit 4 – MODFDIS Mode Fault Detection

Value	Description
0	Mode fault detection enabled
1	Mode fault detection disabled

Bit 2 – PCSDEC Chip Select Decode
When PCSDEC = 1, up to 15 chip select signals can be generated with the four NPCS lines using an external 4-bit to 16-bit decoder. The chip select registers define the characteristics of the 15 chip selects, with the following rules:

SPI_CSR0 defines peripheral chip select signals 0 to 3.

SPI_CSR1 defines peripheral chip select signals 4 to 7.

SPI_CSR2 defines peripheral chip select signals 8 to 11.

SPI_CSR3 defines peripheral chip select signals 12 to 14.

Value	Description
0	The chip select lines are directly connected to a peripheral device.
1	The four NPCS chip select lines are connected to a 4-bit to 16-bit decoder.

Bit 1 – PS Peripheral Select

Value	Description
0	Fixed Peripheral Select
1	Variable Peripheral Select

Bit 0 – MSTR Master/Slave Mode

43.7.16 TWIHS Write Protection Status Register

Name: TWIHS_WPSR
Offset: 0xE8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	WPVSRC[23:16]							
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
	WPVSRC[15:8]							
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
	WPVSRC[7:0]							
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
								WPVS
Access								R
Reset								0

Bits 31:8 – WPVSRC[23:0] Write Protection Violation Source
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

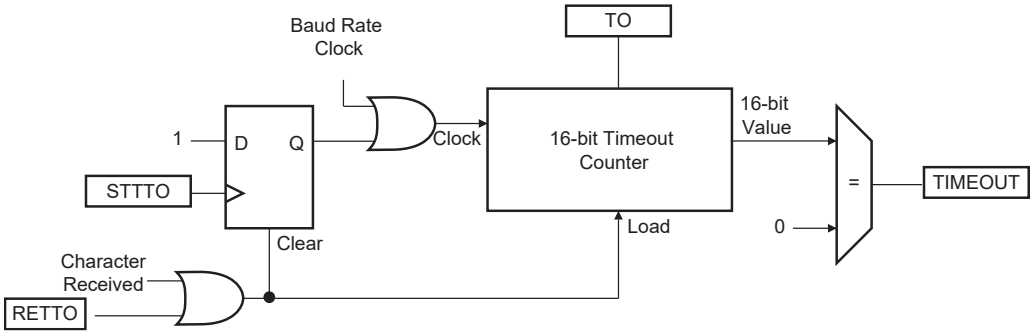
Bit 0 – WPVS Write Protection Violation Status

Value	Description
0	No write protection violation has occurred since the last read of the TWIHS_WPSR.
1	A write protection violation has occurred since the last read of the TWIHS_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

- Obtain an interrupt while no character is received. This is performed by writing a ‘1’ to the RETTO (Reload and Start Timeout) bit in the US_CR. In this case, the counter starts counting down immediately from the value TO. This generates a periodic interrupt so that a user timeout can be handled, for example when no key is pressed on a keyboard.

The following figure shows the block diagram of the Receiver Timeout feature.

Figure 46-23. Receiver Timeout Block Diagram



The following table provides the maximum timeout period for some standard baud rates.

Table 46-8. Maximum Timeout Period

Baud Rate (bit/s)	Bit Time (µs)	Timeout (ms)
600	1,667	109,225
1,200	833	54,613
2,400	417	27,306
4,800	208	13,653
9,600	104	6,827
14,400	69	4,551
19,200	52	3,413
28,800	35	2,276
38,400	26	1,704
56,000	18	1,170
57,600	17	1,138
200,000	5	328

46.6.3.12 Framing Error

The receiver is capable of detecting framing errors. A framing error happens when the stop bit of a received character is detected at level 0. This can occur if the receiver and the transmitter are fully desynchronized.

A framing error is reported in US_CSR.FRAME. FRAME is asserted in the middle of the stop bit as soon as the framing error is detected. It is cleared by writing a ‘1’ to US_CR.RSTSTA.

Bit 21 – DSR Image of DSR Input

Value	Description
0	DSR input is driven low.
1	DSR input is driven high.

Bit 20 – RI Image of RI Input

Value	Description
0	RI input is driven low.
1	RI input is driven high.

Bit 19 – CTSIC Clear to Send Input Change Flag (cleared on read)

Value	Description
0	No input change has been detected on the CTS pin since the last read of US_CSR.
1	At least one input change has been detected on the CTS pin since the last read of US_CSR.

Bit 18 – DCDIC Data Carrier Detect Input Change Flag (cleared on read)

Value	Description
0	No input change has been detected on the DCD pin since the last read of US_CSR.
1	At least one input change has been detected on the DCD pin since the last read of US_CSR.

Bit 17 – DSRIC Data Set Ready Input Change Flag (cleared on read)

Value	Description
0	No input change has been detected on the DSR pin since the last read of US_CSR.
1	At least one input change has been detected on the DSR pin since the last read of US_CSR.

Bit 16 – RIIC Ring Indicator Input Change Flag (cleared on read)

Value	Description
0	No input change has been detected on the RI pin since the last read of US_CSR.
1	At least one input change has been detected on the RI pin since the last read of US_CSR.

Bit 13 – NACK Non Acknowledge Interrupt (cleared by writing a one to bit US_CR.RSTNACK)

Value	Description
0	Non acknowledge has not been detected since the last RSTNACK.
1	At least one non acknowledge has been detected since the last RSTNACK.

Bit 10 – ITER Max Number of Repetitions Reached (cleared by writing a one to bit US_CR.RSTIT)

Value	Description
0	Maximum number of repetitions has not been reached since the last RSTIT.
1	Maximum number of repetitions has been reached since the last RSTIT.

Bit 9 – TXEMPTY Transmitter Empty (cleared by writing US_THR)

Device is frame-locked, it should check every frame continuing to validate that it remains frame-locked. While frame-locked, the Device can access MediaLB according the rules of the MediaLB protocol.

A MediaLB Device must perform the following operations:

- Rules for synchronization to MediaLB:
  - When locked, as long as FRAMESYNC is detected at the expected time, the Device must not synchronize to unexpected FRAMESYNC patterns.
  - When locked and FRAMESYNC is not detected at the expected time for two consecutive frames, declare unlock, and the Device stops driving MLBS and MLBD.
  - When unlocked and FRAMESYNC is detected at the same time for three consecutive frames, declare lock, and the Device can resume driving MLBS and MLBD when appropriate.
- When the Tx Device for a physical channel, it drives Command onto MLBS at the beginning of the physical channel and then sets MLBS to a high impedance state. In addition, the Tx Device drives the data quadlet onto MLBD line for the duration of the physical channel, and then sets the MLBD line to a high impedance state. The NoData command is the default for the MLBS line and does not need to be driven by the Tx Device.
- When the Rx Device for a physical channel, it drives RxStatus onto MLBS in the second byte of the physical channel and then sets MLBS to a high impedance state for asynchronous, control and isochronous (non-broadcast) transmissions. When no RxStatus is driven, the MLBS line defaults to ReceiverReady; however, it is recommended that the Rx Device drive the ReceiverReady response for non-broadcast transmissions.
- When the Rx Device for a physical channel, it must not drive any RxStatus (defaulting to ReceiverReady) for synchronous and isochronous (broadcast) transmissions.

48.6.1.8 Data Transport Methods

MediaLB supports four data transport methods: synchronous stream data, asynchronous packet data, control message data and isochronous data. Synchronous stream data is transmitted in a broadcast fashion, where the only response allowed by an Rx Device is ReceiverReady (MLBS default). Control and asynchronous transport methods are packet based and support only one Rx Device at a time. Control and asynchronous transmissions require start and end commands to delineate the packets. Isochronous data can be broadcast if all Rx Devices do not use the status response of ReceiverBusy. Otherwise, isochronous transmissions must be to a single Rx Device.

Control and Asynchronous

Both the control and asynchronous commands define the boundaries of a packet message and work similarly. The following discussion on control packets also applies to asynchronous packets with the commands changed to the asynchronous versions.

For control packets, the ControlStart command is sent by the Tx Device at the start of a message. After the first quadlet of the message, middle quadlets will use the ControlContinue command. For the last quadlet of the packet, the Tx Device uses the ControlEnd command. If the command sequence is received out of order, the Rx Device sends the status response of ReceiverProtocolError in the next quadlet of the logical channel.

If the Tx Device must abort the packet while it's being transmitted, the ControlBreak command is sent. Assuming a message is to be retransmitted after the ControlBreak command is sent, the message must be restarted from the beginning (cannot resume with the ControlContinue command).

The protocol flow for a Tx Device is illustrated in Figure 48-6 through Figure 48-8. Although these diagrams illustrate control packet transmission, they also apply to asynchronous packets where the commands that start with Control are replaced by Async. The data transfer blocks (slanted rectangle

Offset	Name	Bit Pos.								
		15:8	TIE15	TIE14	TIE13	TIE12	TIE11	TIE10	TIE9	TIE8
		23:16	TIE23	TIE22	TIE21	TIE20	TIE19	TIE18	TIE17	TIE16
		31:24	TIE31	TIE30	TIE29	TIE28	TIE27	TIE26	TIE25	TIE24
0xE4	MCAN_TXBCIE	7:0	CFIE7	CFIE6	CFIE5	CFIE4	CFIE3	CFIE2	CFIE1	CFIE0
		15:8	CFIE15	CFIE14	CFIE13	CFIE12	CFIE11	CFIE10	CFIE9	CFIE8
		23:16	CFIE23	CFIE22	CFIE21	CFIE20	CFIE19	CFIE18	CFIE17	CFIE16
		31:24	CFIE31	CFIE30	CFIE29	CFIE28	CFIE27	CFIE26	CFIE25	CFIE24
0xE8 ... 0xEF	Reserved									
0xF0	MCAN_TXEFC	7:0	EFSA[5:0]							
		15:8	EFSA[13:6]							
		23:16			EFS[5:0]					
		31:24			EFWM[5:0]					
0xF4	MCAN_TXEFS	7:0			EFFL[5:0]					
		15:8				EFGI[4:0]				
		23:16				EFPI[4:0]				
		31:24							TEFL	EFF
0xF8	MCAN_TXEFA	7:0				EFAI[4:0]				
		15:8								
		23:16								
		31:24								

49.6.35 MCAN Tx Buffer Configuration

Name: MCAN_TXBC
Offset: 0xC0
Reset: 0x00000000
Property: Read/Write

This register can only be written if the bits CCE and INIT are set in MCAN CC Control Register.

The sum of TFQS and NDTB may not exceed 32. There is no check for erroneous configurations. The Tx Buffers section in the Message RAM starts with the dedicated Tx Buffers.

Bit	31	30	29	28	27	26	25	24
		TFQM	TFQS[5:0]					
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
			NDTB[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	TBSA[13:6]							
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
	TBSA[5:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0		

Bit 30 – TFQM Tx FIFO/Queue Mode

Value	Description
0	Tx FIFO operation.
1	Tx Queue operation.

Bits 29:24 – TFQS[5:0] Transmit FIFO/Queue Size

Value	Description
0	No Tx FIFO/Queue.
1-32	Number of Tx Buffers used for Tx FIFO/Queue.
>32	Values greater than 32 are interpreted as 32.

Bits 21:16 – NDTB[5:0] Number of Dedicated Transmit Buffers

Value	Description
0	No dedicated Tx Buffers.
1-32	Number of dedicated Tx Buffers.
>32	Values greater than 32 are interpreted as 32.

50.7.4 TC Stepper Motor Mode Register

Name: TC_SMMRx
Offset: 0x08 + x*0x40 [x=0..2]
Reset: 0x00000000
Property: R/W

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

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
							DOWN	GCEN
Access							R/W	R/W
Reset							0	0

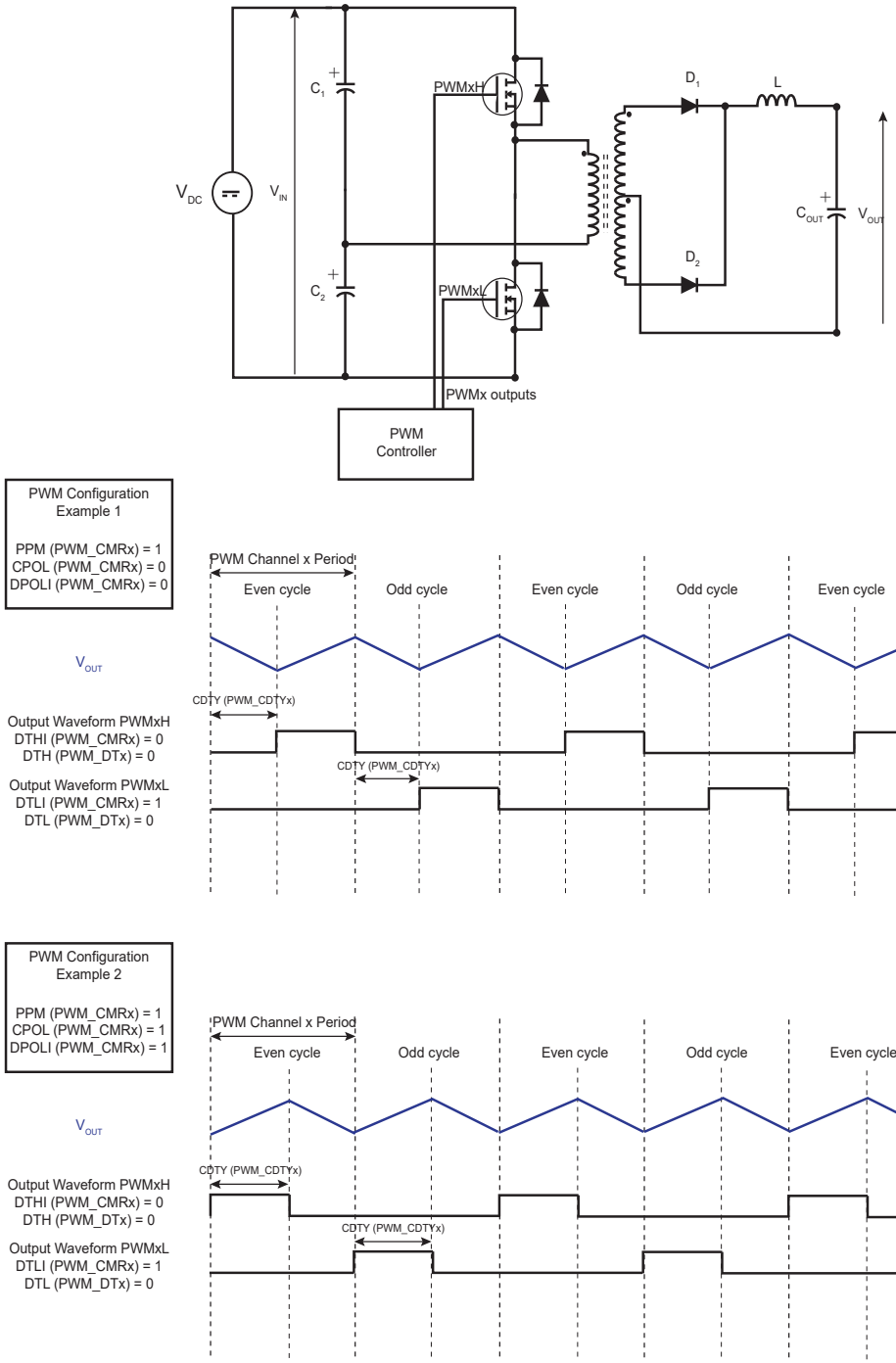
Bit 1 – DOWN Down Count

Value	Description
0	Up counter.
1	Down counter.

Bit 0 – GCEN Gray Count Enable

Value	Description
0	TIOAx [x=0..2] and TIOBx [x=0..2] are driven by internal counter of channel x.
1	TIOAx [x=0..2] and TIOBx [x=0..2] are driven by a 2-bit Gray counter.

Figure 51-13. Half-Bridge Converter Application: No Feedback Regulation



51.7.46 PWM Channel Dead Time Register

Name: PWM_DTx
Offset: 0x0218 + x*0x20 [x=0..3]
Reset: 0x00000000
Property: Read/Write

This register can only be written if bits WPSWS4 and WPHWS4 are cleared in the PWM Write Protection Status Register.

Only the first 12 bits (dead-time counter size) of fields DTH and DTL are significant.

Bit	31	30	29	28	27	26	25	24
	DTL[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	23	22	21	20	19	18	17	16
	DTL[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
	DTH[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
	DTH[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 31:16 – DTL[15:0] Dead-Time Value for PWMLx Output
Defines the dead-time value for PWMLx output. This value must be defined between 0 and CDTY (PWM_CDTYx).

Bits 15:0 – DTH[15:0] Dead-Time Value for PWMHx Output
Defines the dead-time value for PWMHx output. This value must be defined between 0 and the value (CPRD – CDTY) (PWM_CPRDx and PWM_CDTYx).

51.7.47 PWM Channel Dead Time Update Register

Name: PWM_DTUPDx
Offset: 0x021C + x*0x20 [x=0..3]
Reset: 0x00000000
Property: Write-only

This register can only be written if bits WPSWS4 and WPHWS4 are cleared in the PWM Write Protection Status Register.

This register acts as a double buffer for the DTH and DTL values. This prevents an unexpected waveform when modifying the dead-time values.

Only the first 12 bits (dead-time counter size) of fields DTHUPD and DTLUPD are significant.

Bit	31	30	29	28	27	26	25	24
	DTLUPD[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DTLUPD[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	–

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

Bit	7	6	5	4	3	2	1	0
	DTHUPD[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	–

Bits 31:16 – DTLUPD[15:0] Dead-Time Value Update for PWMLx Output
Defines the dead-time value for PWMLx output. This value must be defined between 0 and CDTY (PWM_CDTYx). This value is applied only at the beginning of the next channel x PWM period.

Bits 15:0 – DTHUPD[15:0] Dead-Time Value Update for PWMHx Output
Defines the dead-time value for PWMHx output. This value must be defined between 0 and the value (CPRD – CDTY) (PWM_CPRDx and PWM_CDTYx). This value is applied only at the beginning of the next channel x PWM period.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
C_{SHUNT}	Shunt capacitance	–	–	–	7	pF
$C_{CRYSTAL}$	Allowed Crystal Capacitance Load	From crystal specification	12.5	–	17.5	pF
P_{ON}	Drive Level	3 MHz	–	–	15	µW
		8 MHz	–	–	30	
		12 MHz, 20 MHz	–	–	50	

59.4.8 3 to 20 MHz XIN Clock Input Characteristics in Bypass Mode

Table 59-25. 3 to 20 MHz XIN Clock Input Characteristics in Bypass Mode

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$1/(t_{CPXIN})$	XIN Clock Frequency	(see **Note 1**)	–	–	20	MHz
t_{CHXIN}	XIN Clock High Half-period	(see **Note 1**)	25	–	–	ns
t_{CLXIN}	XIN Clock Low Half-period	(see **Note 1**)	25	–	–	ns
V_{XIN_IL}	V_{XIN} Input Low-level Voltage	(see **Note 1**)	Min of V_{IL} for CLOCK pad	–	Max of V_{IL} for CLOCK pad	V
V_{XIN_IH}	V_{XIN} Input High-level Voltage	(see **Note 1**)	Min of V_{IH} for CLOCK pad	–	Max of V_{IH} for CLOCK pad	V

Note:

1. These characteristics apply only when the 3–20 MHz crystal oscillator is in Bypass mode.

59.4.9 Crystal Oscillator Design Considerations

59.4.9.1 Choosing a Crystal

When choosing a crystal for the 32768 Hz Slow Clock Oscillator or for the 3–20 MHz oscillator, several parameters must be taken into account. Important parameters between crystal and product specifications are as follows:

- Crystal Load Capacitance
  - The total capacitance loading the crystal, including the oscillator's internal parasitics and the PCB parasitics, must match the load capacitance for which the crystal's frequency is specified. Any mismatch in the load capacitance with respect to the crystal's specification will lead to inaccurate oscillation frequency
- Drive Level
  - Crystal drive level ≥ Oscillator Drive Level. Having a crystal drive level number lower than the oscillator specification may damage the crystal.
- Equivalent Series Resistor (ESR)
  - Crystal ESR ≤ Oscillator ESR Max. Having a crystal with ESR value higher than the oscillator may cause the oscillator to not start.
- Shunt Capacitance
  - Max. crystal shunt capacitance ≤ Oscillator Shunt Capacitance (C_{SHUNT}). Having a crystal with C_{SHUNT} value higher than the oscillator may cause the oscillator to not start.

Signal Name	Recommended Pin Connection	Description
		Pulled-up inputs (100 kOhm) to VDDIO at reset.
TIOBx	Application dependent.	TC Channel x I/O Line B Pulled-up inputs (100 kOhm) to VDDIO at reset.
Pulse Width Modulation Controller		
PWMC0_PWMHx PWMC1_PWMHx	Application dependent.	Waveform Output High for Channel x Pulled-up inputs (100 kOhm) to VDDIO at reset.
PWMC0_PWMLx PWMC1_PWMLx	Application dependent.	Waveform Output Low for Channel x Pulled-up inputs (100 kOhm) to VDDIO at reset.
PWMC0_PWMFI0– PWMC0_PWMFI2 PWMC1_PWMFI0– PWMC1_PWMFI2	Application dependent.	Fault Inputs Pulled-up inputs (100 kOhm) to VDDIO at reset.
PWMC0_PWMEXTRG0 PWMC0_PWMEXTRG1 PWMC1_PWMEXTRG0 PWMC1_PWMEXTRG1	Application dependent.	External Trigger Inputs Pulled-up inputs (100 kOhm) to VDDIO at reset.
Serial Peripheral Interface		
SPIx_MISO	Application dependent.	Master In Slave Out Pulled-up inputs (100 kOhm) to VDDIO at reset.
SPIx_MOSI	Application dependent.	Master Out Slave In Pulled-up inputs (100 kOhm) to VDDIO at reset.
SPIx_SPCK	Application dependent.	SPI Serial Clock Pulled-up inputs (100 kOhm) to VDDIO at reset.
SPIx_NPCS0	Application dependent. (Pullup at VDDIO)	SPI Peripheral Chip Select 0 Pulled-up inputs (100 kOhm) to VDDIO at reset.
SPIx_NPCS1– SPIx_NPCS3	Application dependent. (Pullup at VDDIO)	SPI Peripheral Chip Select Pulled-up inputs (100 kOhm) to VDDIO at reset.
Two-Wire Interface		
TWDx	Application dependent. (4.7kOhm Pulled-up on VDDIO)	TWIx Two-wire Serial Data Pulled-up inputs (100 kOhm) to VDDIO at reset.
TWCKx	Application dependent. (4.7kOhm Pulled-up on VDDIO)	TWIx Two-wire Serial Clock Pulled-up inputs (100 kOhm) to VDDIO at reset.