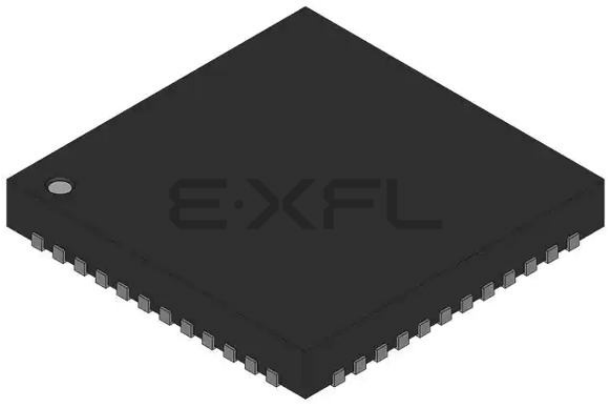


Welcome to E-XFL.COM

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	AVR
Core Size	32-Bit Single-Core
Speed	50MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	36
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	32K x 8
Voltage - Supply (Vcc/Vdd)	1.62V ~ 3.6V
Data Converters	A/D 8x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	48-UFLGA Exposed Pad
Supplier Device Package	48-TLLGA (5.5x5.5)
Purchase URL	https://www.e-xfl.com/product-detail/atmel/at32uc3l0256-d3ht

7.7.19 Performance Channel 0 Read Stall Cycles

Name: PRSTALL0

Access Type: Read-only

Offset: 0x808

Reset Value: 0x00000000

31	30	29	28	27	26	25	24
STALL[31:24]							

23	22	21	20	19	18	17	16
STALL[23:16]							

15	14	13	12	11	10	9	8
STALL[15:8]							

7	6	5	4	3	2	1	0
STALL[7:0]							

- **STALL: Stall Cycles Counted Since Last Reset**
  Clock cycles are counted using the CLK_PDCA_HSB clock

After programming, the page can be locked to prevent miscellaneous write or erase sequences. Locking is performed on a per-region basis, so locking a region locks all pages inside the region. Additional protection is provided for the lowermost address space of the flash. This address space is allocated for the Boot Loader, and is protected both by the lock bit(s) corresponding to this address space, and the BOOTPROT[2:0] fuses.

Data to be written is stored in an internal buffer called the page buffer. The page buffer contains *w* words. The page buffer wraps around within the internal memory area address space and appears to be repeated by the number of pages in it. Writing of 8-bit and 16-bit data to the page buffer is not allowed and may lead to unpredictable data corruption.

Data must be written to the page buffer before the programming command is written to the Flash Command Register (FCMD). The sequence is as follows:

- Reset the page buffer with the Clear Page Buffer command.
- Fill the page buffer with the desired contents as described in Section 8.4.8 on page 81.
- Programming starts as soon as the programming key and the programming command are written to the Flash Command Register. The PAGEN field in the Flash Command Register (FCMD) must contain the address of the page to write. PAGEN is automatically updated when writing to the page buffer, but can also be written to directly. The FRDY bit in the Flash Status Register (FSR) is automatically cleared when the page write operation starts.
- When programming is completed, the FRDY bit in the Flash Status Register (FSR) is set. If an interrupt was enabled by writing FCR.FRDY to one, an interrupt request is generated.

Two errors can be detected in the FSR register after a programming sequence:

- Programming Error: A bad keyword and/or an invalid command have been written in the FCMD register.
- Lock Error: Can have two different causes:
  - The page to be programmed belongs to a locked region. A command must be executed to unlock the corresponding region before programming can start.
  - A bus master without secure status attempted to program a page requiring secure privileges.

8.5.2 Erase All Operation

The entire memory is erased if the Erase All command (EA) is written to the Flash Command Register (FCMD). Erase All erases all bits in the flash array. The User page is not erased. All flash memory locations, the general-purpose fuse bits, and the security bit are erased (reset to 0xFF) after an Erase All.

The EA command also ensures that all volatile memories, such as register file and RAMs, are erased before the security bit is erased.

Erase All operation is allowed only if no regions are locked, and the BOOTPROT fuses are configured with a BOOTPROT region size of 0. Thus, if at least one region is locked, the bit LOCKE in FSR is set and the command is cancelled. If the LOCKE bit in FCR is one, an interrupt request is set generated.

When the command is complete, the FRDY bit in the Flash Status Register (FSR) is set. If an interrupt has been enabled by writing FCR.FRDY to one, an interrupt request is generated. Two errors can be detected in the FSR register after issuing the command:

9.6.3 Channel Enable Register High

Name: CERH

Access Type: Read/Write

Offset: 0x08

Reset Value: 0x00000000

31	30	29	28	27	26	25	24
-	CERH[30:24]						

23	22	21	20	19	18	17	16
CERH[23:16]							

15	14	13	12	11	10	9	8
CERH[15:8]							

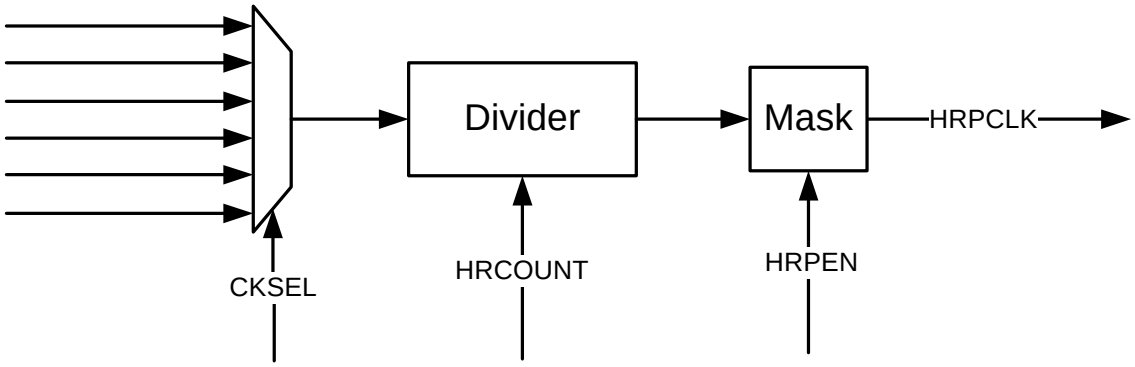
7	6	5	4	3	2	1	0
CERH[7:0]							

- **CERH[n]: Channel Enable Register High**
  0: Channel (n+32) is not enabled.
  1: Channel (n+32) is enabled.

Table 10-1. HMATRIX Register Memory Map (Continued)

Offset	Register	Name	Access	Reset Value
0x012C	Special Function Register 7	SFR7	Read/Write	–
0x0130	Special Function Register 8	SFR8	Read/Write	–
0x0134	Special Function Register 9	SFR9	Read/Write	–
0x0138	Special Function Register 10	SFR10	Read/Write	–
0x013C	Special Function Register 11	SFR11	Read/Write	–
0x0140	Special Function Register 12	SFR12	Read/Write	–
0x0144	Special Function Register 13	SFR13	Read/Write	–
0x0148	Special Function Register 14	SFR14	Read/Write	–
0x014C	Special Function Register 15	SFR15	Read/Write	–

Figure 13-7. High Resolution Prescaler Generation



The HRP is enabled by writing a one to the High Resolution Prescaler Enable (HRPEN) bit in the High Resolution Prescaler Control Register (HRPCR).

The user can select a clock source for the HRP by writing to the Clock Selection (CKSEL) field of the HRPCR register.

The user must configure the High Resolution Prescaler Clock (HRPCLK) frequency by writing to the High Resolution Count (HRCOUNT) field of the High Resolution Counter (HRPCR) register. This results in the output frequency:

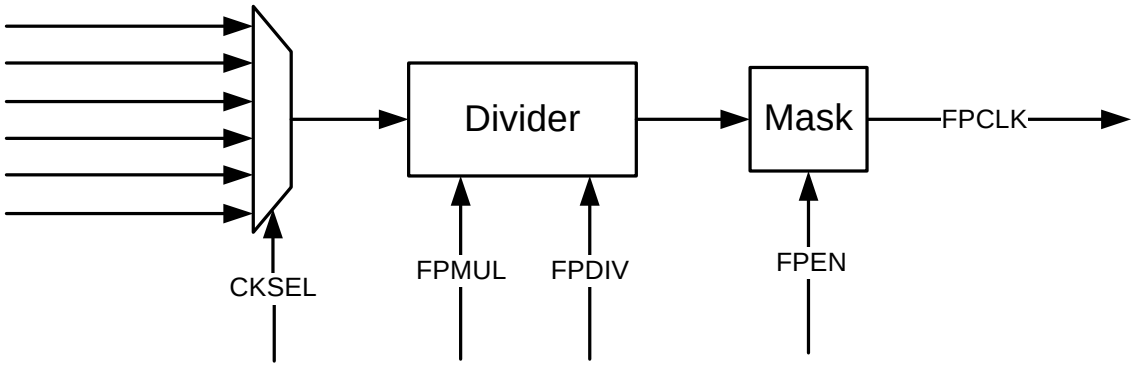
$$f_{HRPCLK} = f_{SRC} / (2*(HRCOUNT+1))$$

The CKSEL field can not be changed dynamically but the HRCOUNT field can be changed on-the-fly.

13.5.14.2 Fractional prescaler

The FP generates a clock whose average frequency is more precise than the HRP. However, this clock frequency is subject to jitter around the target clock frequency. This jitter influence can be decreased by dividing this clock with the GCLK divider. Moreover the duty cycle of this clock is not precisely 50%.

Figure 13-8. Fractional Prescaler Generation



The FP is enabled by writing a one to the FPEN bit in the Fractional Prescaler Control Register (FPCR).

The user can select a clock source for the FP by writing to the CKSEL field of the FPCR register.

13.7 Module Configuration

The specific configuration for each SCIF instance is listed in the following tables.The module bus clocks listed here are connected to the system bus clocks. Please refer to the Power Manager chapter for details.

Table 13-7. MODULE Clock Name

Module Name	Clock Name	Description
SCIF	CLK_SCIF	Clock for the SCIF bus interface

Table 13-8. Oscillator Startup Times

STARTUP	Number of System RC oscillator clock cycle	Approximative Equivalent time (RCSYS = 115 kHz)
0	0	0
1	64	557 us
2	128	1.1 ms
3	2048	18 ms
4	4096	36 ms
5	8192	71 ms
6	16384	143 ms
7	32768	285 ms
8	4	35 us
9	8	70 us
10	16	139 us
11	32	278 us
12	256	2.2 ms
13	512	4.5 ms
14	1024	8.9 ms
15	32768	285 ms

Table 13-9. Oscillator Gain Settings

GAIN[1:0]	Function
0	Oscillator is used with gain G0 (XIN from 0.45 MHz to 12.0 MHz)
1	Oscillator is used with gain G1 (XIN from 12.0 MHz to 16.0 MHz)
2	Oscillator is used with gain G2 (XIN equals 16.0 MHz. Used for e.g. increasing S/N ratio, better drive strength for high ESR crystals)
3	Oscillator is used with gain G3 (XIN equals 16.0 MHz. Used for e.g. increasing S/N ratio, better drive strength for high ESR crystals)

16.7.8 Level Register

Name: LEVEL

Access Type: Read/Write

Offset: 0x01C

Reset Value: 0x00000000

31	30	29	28	27	26	25	24
-	INT30	INT29	INT28	INT27	INT26	INT25	INT24
23	22	21	20	19	18	17	16
INT23	INT22	INT21	INT20	INT19	INT18	INT17	INT16
15	14	13	12	11	10	9	8
INT15	INT14	INT13	INT12	INT11	INT10	INT9	INT8
7	6	5	4	3	2	1	0
INT7	INT6	INT5	INT4	INT3	INT2	INT1	NMI

- **INTn: External Interrupt n**
  0: The external interrupt triggers on low level.
  1: The external interrupt triggers on high level.
  Please refer to the Module Configuration section for the number of external interrupts.
- **NMI: Non-Maskable Interrupt**
  0: The Non-Maskable Interrupt triggers on low level.
  1: The Non-Maskable Interrupt triggers on high level.

18.7.12 Interrupt Enable Register

Name: IER

Access: Read/Write, Set, Clear, Toggle

Offset: 0x090, 0x094, 0x098, 0x09C

Reset Value: -

31	30	29	28	27	26	25	24
P31	P30	P29	P28	P27	P26	P25	P24
23	22	21	20	19	18	17	16
P23	P22	P21	P20	P19	P18	P17	P16
15	14	13	12	11	10	9	8
P15	P14	P13	P12	P11	P10	P9	P8
7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0

- **P0-31: Interrupt Enable**
  0: Interrupt is disabled for the corresponding pin.
  1; Interrupt is enabled for the corresponding pin.

18.7.19 Unlock Register

Name: UNLOCK

Access: Write-only

Offset: 0x1E0

Reset Value: -

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

- **OFFSET: Register Offset**
  This field must be written with the offset value of the LOCK, LOCKC or LOCKT register to unlock. This offset must also include the port offset for the register to unlock. LOCKS can not be locked so no unlock is required before writing to this register.
- **KEY: Unlocking Key**
  This bitfield must be written to 0xAA for a write to this register to have an effect.

  This register always reads as zero.

18.8 Module Configuration

The specific configuration for each GPIO instance is listed in the following tables. The module bus clocks listed here are connected to the system bus clocks. Refer to the Power Manager chapter for details.

Table 18-3. GPIO Configuration

Feature	GPIO
Number of GPIO ports	2
Number of peripheral functions	8

Table 18-4. Implemented Pin Functions

Pin Function	Implemented	Notes
Pull-up	On all pins	Controlled by PUER or peripheral

Table 18-5. GPIO Clocks

Module Name	Clock Name	Description
GPIO	CLK_GPIO	Clock for the GPIO bus interface

The reset values for all GPIO registers are zero, with the following exceptions:

Table 18-6. Register Reset Values

Port	Register	Reset Value
0	GPER	0x004DFF5F
0	PMR0	0x00320020
0	PMR1	0x00020080
0	PMR2	0x00100800
0	PUER	0x00000001
0	GFER	0x007FFFFF
0	PARAMETER	0x007FFFFF
0	VERSION	0x00000213
1	GPER	0x0FFFFFCF
1	PMR0	0x00000030
1	PMR1	0x00000030
1	GFER	0x0FFFFFFF
1	PARAMETER	0x0FFFFFFF
1	VERSION	0x00000213

slave to pull it down in order to generate the acknowledge. The master polls the data line during this clock pulse.

The SR.RXRDY bit indicates that a data byte is available in the RHR. The RXRDY bit is also used as Receive Ready for the Peripheral DMA Controller receive channel.

Figure 22-10. Slave Receiver with One Data Byte

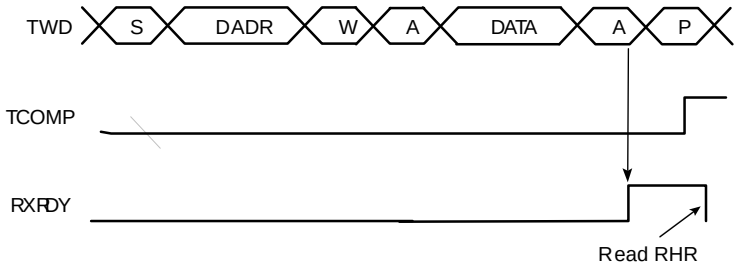
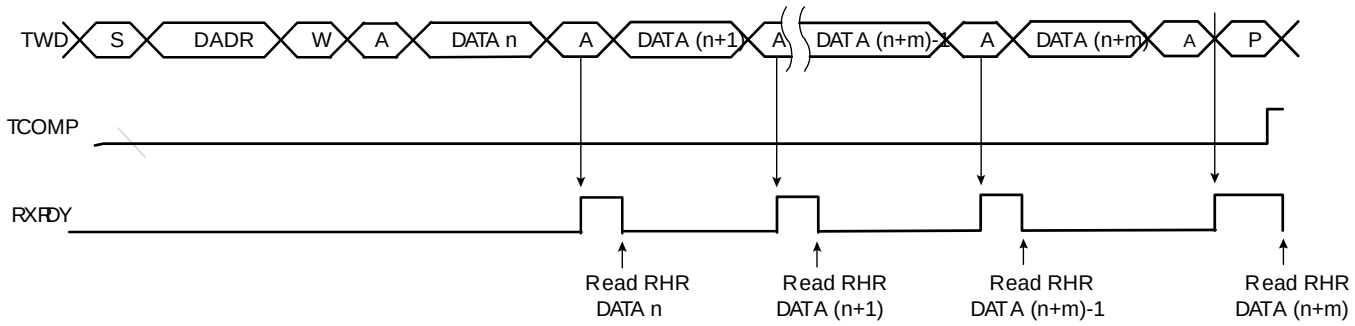


Figure 22-11. Slave Receiver with Multiple Data Bytes



22.8.5 Interactive ACKing Received Data Bytes

When implementing a register interface over TWI, it may sometimes be necessary or just useful to report reads and writes to invalid register addresses by sending a NAK to the host. To be able to do this, one must first receive the register address from the TWI bus, and then tell the TWIS whether to ACK or NAK it. In normal operation of the TWIS, this is not possible because the controller will automatically ACK the byte at about the same time as the RXRDY bit changes from zero to one. Writing a one to the Stretch on Data Byte Received bit (CR.SODR) will stretch the clock allowing the user to update CR.ACK bit before returning the desired value. After the last bit in the data byte is received, the TWI bus clock is stretched, the received data byte is transferred to the RHR register, and SR.BTF is set. At this time, the user can examine the received byte and write the desired ACK or NACK value to CR.ACK. When the user clears SR.BTF, the desired ACK value is transferred on the TWI bus. This makes it possible to look at the byte received, determine if it is valid, and then decide to ACK or NAK it.

22.8.6 Using the Peripheral DMA Controller

The use of the Peripheral DMA Controller significantly reduces the CPU load. The user can set up ring buffers for the Peripheral DMA Controller, containing data to transmit or free buffer space to place received data. By initializing NBYTES to zero before a transfer, and writing a one to CR.CUP, NBYTES is incremented by one each time a data has been transmitted or received. This allows the user to detect how much data was actually transferred by the DMA system.

22.9.10 Interrupt Mask Register

Name: IMR

Access Type: Read-only

Offset: 0x24

Reset Value: 0x00000000

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

23	22	21	20	19	18	17	16
BTF	REP	STO	SMBDAM	SMBHHM	SMBALERTM	GCM	SAM

15	14	13	12	11	10	9	8
-	BUSERR	SMBPECERR	SMBTOUT	-	-	-	NAK

7	6	5	4	3	2	1	0
ORUN	URUN	-	-	TCOMP	-	TXRDY	RXRDY

0: The corresponding interrupt is disabled.
1: The corresponding interrupt is enabled.
This bit is cleared when the corresponding bit in IDR is written to one.
This bit is set when the corresponding bit in IER is written to one.

23.5.1 I/O Lines

The pins used for interfacing the PWMA may be multiplexed with I/O Controller lines. The programmer must first program the I/O Controller to assign the desired PWMA pins to their peripheral function.

It is only required to enable the PWMA outputs actually in use.

23.5.2 Power Management

If the CPU enters a sleep mode that disables clocks used by the PWMA, the PWMA will stop functioning and resume operation after the system wakes up from sleep mode.

23.5.3 Clocks

The clock for the PWMA bus interface (CLK_PWMA) is controlled by the Power Manager. This clock is enabled at reset, and can be disabled in the Power Manager. It is recommended to disable the PWMA before disabling the clock, to avoid freezing the PWMA in an undefined state.

Additionally, the PWMA depends on a dedicated Generic Clock (GCLK). The GCLK can be set to a wide range of frequencies and clock sources and must be enabled in the System Control Interface (SCIF) before the PWMA can be used.

23.5.4 Interrupts

The PWMA interrupt request lines are connected to the interrupt controller. Using the PWMA interrupts requires the interrupt controller to be programmed first.

23.5.5 Peripheral Events

The PWMA peripheral events are connected via the Peripheral Event System. Refer to the Peripheral Event System chapter for details.

23.5.6 Debug Operation

When an external debugger forces the CPU into debug mode, the PWMA continues normal operation. If the PWMA is configured in a way that requires it to be periodically serviced by the CPU through interrupts, improper operation or data loss may result during debugging.

23.6 Functional Description

The PWMA embeds a number of PWM channel submodules, each providing an output PWM waveform. Each PWM channel contains a duty cycle register and a comparator. A common timebase counter for all channels determines the frequency and the period for all the PWM waveforms.

23.6.1 Enabling the PWMA

Once the GCLK has been enabled, the PWMA is enabled by writing a one to the EN bit in the Control Register (CR).

23.6.2 Timebase Counter

The top value of the timebase counter defines the period of the PWMA output waveform. The timebase counter starts at zero when the PWMA is enabled and counts upwards until it reaches its effective top value (ETV). The effective top value is defined by specifying the desired number of GCLK clock cycles in the TOP field of Top Value Register (TVR.TOP) in normal operation (the

23.7.13 Interlinked Single Value Channel Set

Name: ISCHSETm

Access Type: Write-only

Offset: 0x30+m*0x10

Reset Value: 0x00000000

31	30	29	28	27	26	25	24
SET							

23	22	21	20	19	18	17	16
SET							

15	14	13	12	11	10	9	8
SET							

7	6	5	4	3	2	1	0
SET							

- **SET: Single Value Channel Set**
  If the bit *n* in SET is one, the duty cycle of PWMA channel *n* will be updated with the value written to ISDUTY.
  If more than one ISCHSET register is present, ISCHSET0 controls channels 31 to 0 and ISCHSET1 controls channels 63 to 32.

Note: The duty registers will be updated with the value stored in the ISDUTY register when any ISCHSETm register is written. Synchronization takes place immediately when an ISCHSET register is written. The duty cycle registers will, however, not be updated until the synchronization is completed and the timebase counter reaches its top value in order to avoid glitches.

23.8 Module Configuration

The specific configuration for each PWMA instance is listed in the following tables. The module bus clocks listed here are connected to the system bus clocks. Please refer to the Power Manager chapter for details.

Table 23-4. PWMA Configuration

Feature	PWMA
Number of PWM channels	36
Channels supporting incoming peripheral events	0, 6, 8, 9, 11, 14, 19, and 20
PWMA channels with Open Drain mode	21, 27, and 28

Table 23-5. PWMA Clocks

Clock Name	Descripton
CLK_PWMA	Clock for the PWMA bus interface
GCLK	The generic clock used for the PWMA is GCLK3

Table 23-6. Register Reset Values

Register	Reset Value
VERSION	0x00000201
PARAMETER	0x00000024

24.7.5 Channel Register A

Name: RA

Access Type: Read-only if CMRn.WAVE = 0, Read/Write if CMRn.WAVE = 1

Offset: 0x14 + n * 0X40

Reset Value: 0x00000000

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

23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-

15	14	13	12	11	10	9	8
RA[15:8]							

7	6	5	4	3	2	1	0
RA[7:0]							

- **RA: Register A**
  RA contains the Register A value in real time.

26.9.1 Control Register

Register Name: CR

Access Type: Write-only

Offset: 0x00

Reset Value: 0x00000000

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

- **DIS: ADCDIFB Disable**
  Writing a zero to this bit has no effect.
  Writing a one to this bit disables the ADCIFB.
  Note: Disabling the ADCIFB effectively stops all clocks in the module so the user must make sure the ADCIFB is idle before disabling the ADCIFB.
- **EN: ADCIFB Enable**
  Writing a zero to this bit has no effect.
  Writing a one to this bit enables the ADCIFB.
  Note: The ADCIFB must be enabled before use.
- **START: Start Conversion**
  Writing a zero to this bit has no effect.
  Writing a one to this bit starts an Analog-to-Digital conversion.
- **SWRST: Software Reset**
  Writing a zero to this bit has no effect.
  Writing a one to this bit resets the ADCIFB, simulating a hardware reset.

26.9.10 Interrupt Disable Register

Name: IDR

Access Type: Write-only

Offset: 0x24

Reset Value: 0x00000000

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

23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-

15	14	13	12	11	10	9	8
-	CELSE	CGT	CLT	-	-	BUSY	READY

7	6	5	4	3	2	1	0
-	-	NOCNT	PENCNT	-	-	OVRE	DRDY

Writing a zero to a bit in this register has no effect.
Writing a one to a bit in this register will clear the corresponding bit in IMR.

Table 32-7. High-drive I/O Pin Characteristics[1]

Symbol	Parameter	Condition	Min	Typ	Max	Units
V_{IL}	Input low-level voltage	V_{VDD} = 3.0V	-0.3		0.3*V_{VDD}	V
		V_{VDD} = 1.62V	-0.3		0.3*V_{VDD}	
V_{IH}	Input high-level voltage	V_{VDD} = 3.6V	0.7*V_{VDD}		V_{VDD} + 0.3	V
		V_{VDD} = 1.98V	0.7*V_{VDD}		V_{VDD} + 0.3	
V_{OL}	Output low-level voltage	V_{VDD} = 3.0V, I_{OL} = 6mA			0.4	V
		V_{VDD} = 1.62V, I_{OL} = 4mA			0.4	
V_{OH}	Output high-level voltage	V_{VDD} = 3.0V, I_{OH} = 6mA	V_{VDD}-0.4			V
		V_{VDD} = 1.62V, I_{OH} = 4mA	V_{VDD}-0.4			
f_{MAX}	Output frequency, all High-drive I/O pins, except PA08 and PA09[2]	V_{VDD} = 3.0V, load = 10pF			45	MHz
		V_{VDD} = 3.0V, load = 30pF			23	
t_{RISE}	Rise time, all High-drive I/O pins, except PA08 and PA09[2]	V_{VDD} = 3.0V, load = 10pF			4.7	ns
		V_{VDD} = 3.0V, load = 30pF			11.5	
t_{FALL}	Fall time, all High-drive I/O pins, except PA08 and PA09[2]	V_{VDD} = 3.0V, load = 10pF			4.8	
		V_{VDD} = 3.0V, load = 30pF			12	
f_{MAX}	Output frequency, PA08 and PA09[2]	V_{VDD} = 3.0V, load = 10pF			54	MHz
		V_{VDD} = 3.0V, load = 30pF			40	
t_{RISE}	Rise time, PA08 and PA09[2]	V_{VDD} = 3.0V, load = 10pF			2.8	ns
		V_{VDD} = 3.0V, load = 30pF			4.9	
t_{FALL}	Fall time, PA08 and PA09[2]	V_{VDD} = 3.0V, load = 10pF			2.4	
		V_{VDD} = 3.0V, load = 30pF			4.6	
I_{LEAK}	Input leakage current	Pull-up resistors disabled			1	µA
C_{IN}	Input capacitance, all High-drive I/O pins, except PA08 and PA09	TQFP48 package		2.2		pF
		QFN48 package		2.0		
		TLLGA48 package		2.0		
C_{IN}	Input capacitance, PA08 and PA09	TQFP48 package		7.0		
		QFN48 package		6.7		
		TLLGA48 package		6.7		

Notes:
1. V_{VDD} corresponds to either V_{VDDIN} or V_{VDDIO}, depending on the supply for the pin. Refer to Section 3.2.1 on page 8 for details.
2. These values are based on simulation and characterization of other AVR microcontrollers manufactured in the same process technology. These values are not covered by test limits in production.

Table 32-8. High-drive I/O, 5V Tolerant, Pin Characteristics[1]

Symbol	Parameter	Condition	Min	Typ	Max	Units
R_{PULLUP}	Pull-up resistance		30	50	110	kOhm
V_{IL}	Input low-level voltage	V_{VDD} = 3.0V	-0.3		0.3*V_{VDD}	V
		V_{VDD} = 1.62V	-0.3		0.3*V_{VDD}	

Table 32-8. High-drive I/O, 5V Tolerant, Pin Characteristics[1]

Symbol	Parameter	Condition	Min	Typ	Max	Units
V_{IH}	Input high-level voltage	V_{VDD} = 3.6V	0.7*V_{VDD}		5.5	V
		V_{VDD} = 1.98V	0.7*V_{VDD}		5.5	
V_{OL}	Output low-level voltage	V_{VDD} = 3.0V, I_{OL} = 6mA			0.4	V
		V_{VDD} = 1.62V, I_{OL} = 4mA			0.4	
V_{OH}	Output high-level voltage	V_{VDD} = 3.0V, I_{OH} = 6mA	V_{VDD}-0.4			V
		V_{VDD} = 1.62V, I_{OH} = 4mA	V_{VDD}-0.4			
f_{MAX}	Output frequency[2]	V_{VDD} = 3.0V, load = 10pF			87	MHz
		V_{VDD} = 3.0V, load = 30pF			58	
t_{RISE}	Rise time[2]	V_{VDD} = 3.0V, load = 10pF			2.3	ns
		V_{VDD} = 3.0V, load = 30pF			4.3	
t_{FALL}	Fall time[2]	V_{VDD} = 3.0V, load = 10pF			1.9	
		V_{VDD} = 3.0V, load = 30pF			3.7	
I_{LEAK}	Input leakage current	5.5V, pull-up resistors disabled			10	µA
C_{IN}	Input capacitance	TQFP48 package		4.5		pF
		QFN48 package		4.2		
		TLLGA48 package		4.2		

Notes: 1. V_{VDD} corresponds to either V_{VDDIN} or V_{VDDIO}, depending on the supply for the pin. Refer to Section 3.2.1 on page 8 for details.
2. These values are based on simulation and characterization of other AVR microcontrollers manufactured in the same process technology. These values are not covered by test limits in production.

Table 32-9. TWI Pin Characteristics[1]

Symbol	Parameter	Condition	Min	Typ	Max	Units
R_{PULLUP}	Pull-up resistance		25	35	60	kOhm
V_{IL}	Input low-level voltage	V_{VDD} = 3.0V	-0.3		0.3*V_{VDD}	V
		V_{VDD} = 1.62V	-0.3		0.3*V_{VDD}	
V_{IH}	Input high-level voltage	V_{VDD} = 3.6V	0.7*V_{VDD}		V_{VDD} + 0.3	V
		V_{VDD} = 1.98V	0.7*V_{VDD}		V_{VDD} + 0.3	
	Input high-level voltage, 5V tolerant SMBUS compliant pins	V_{VDD} = 3.6V	0.7*V_{VDD}		5.5	V
		V_{VDD} = 1.98V	0.7*V_{VDD}		5.5	
V_{OL}	Output low-level voltage	I_{OL} = 3mA			0.4	V
I_{LEAK}	Input leakage current	Pull-up resistors disabled			1	µA
I_{IL}	Input low leakage				1	
I_{IH}	Input high leakage				1	
C_{IN}	Input capacitance	TQFP48 package		3.8		pF
		QFN48 package		3.5		
		TLLGA48 package		3.5		