Microchip Technology - AT94K05AL-25AQI Datasheet

Welcome to E-XFL.COM

Embedded - FPGAs (Field Programmable Gate Array) with Microcontrollers: Enhancing Flexibility and Performance

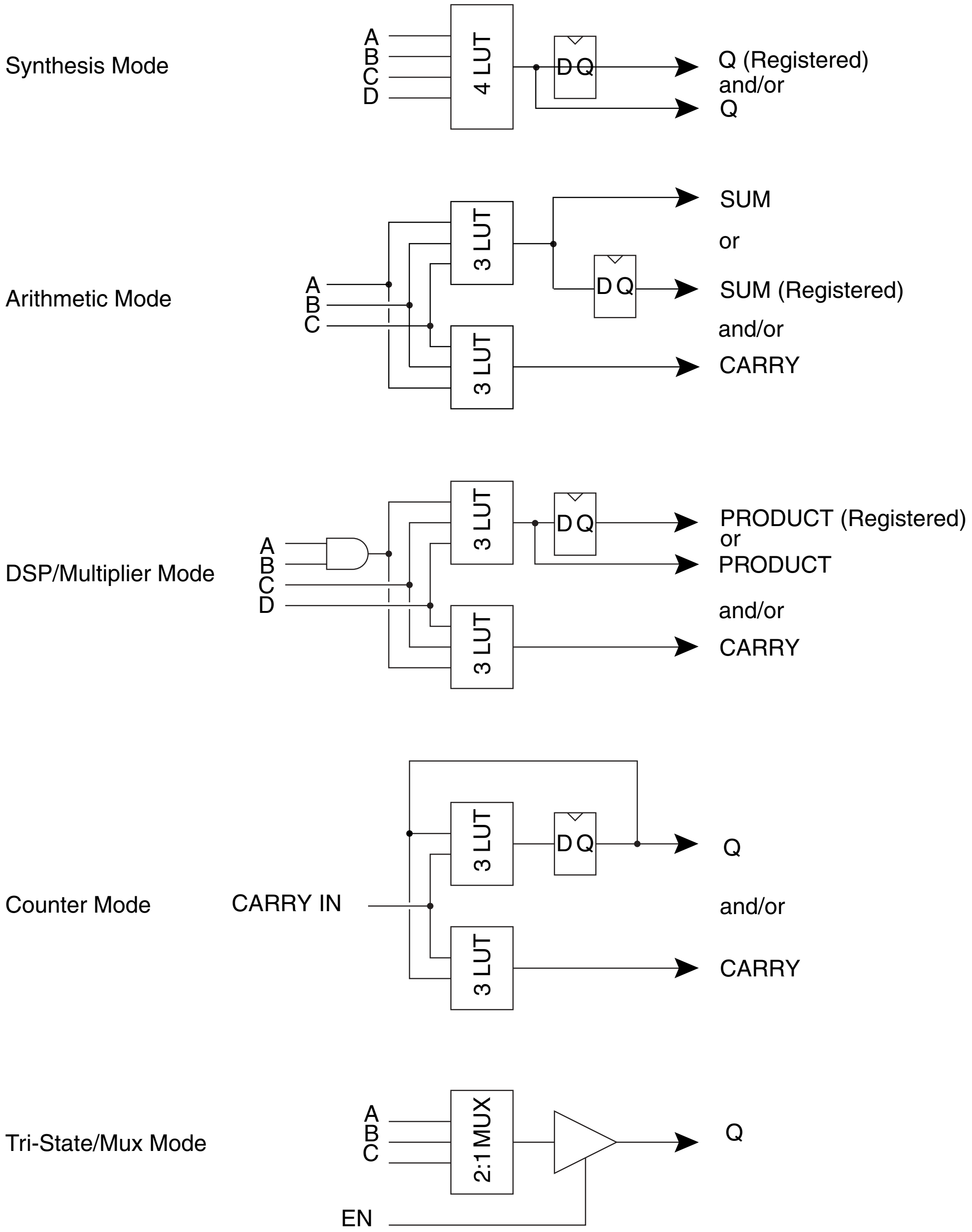
Embedded - FPGAs (Field Programmable Gate Arrays) with Microcontrollers represent a cutting-edge category of electronic components that combine the flexibility of FPGA technology with the processing power of integrated microcontrollers. This hybrid approach offers a versatile solution for designing and implementing complex digital systems that require both programmable logic and embedded processing capabilities.

What Are Embedded - FPGAs with Microcontrollers?

At their core, **FPGAs** are semiconductor devices that can

Details	
Product Status	Obsolete
Core Type	8-Bit AVR
Speed	25 MHz
Interface	I²C, UART
Program SRAM Bytes	4K-16K
FPGA SRAM	2kb
EEPROM Size	-
Data SRAM Bytes	4K ~ 16K
FPGA Core Cells	256
FPGA Gates	5K
FPGA Registers	436
Voltage - Supply	3V ~ 3.6V
Mounting Type	Surface Mount
Operating Temperature	-40°C ~ 85°C
Package / Case	100-TQFP
Supplier Device Package	100-TQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/at94k05al-25aqi

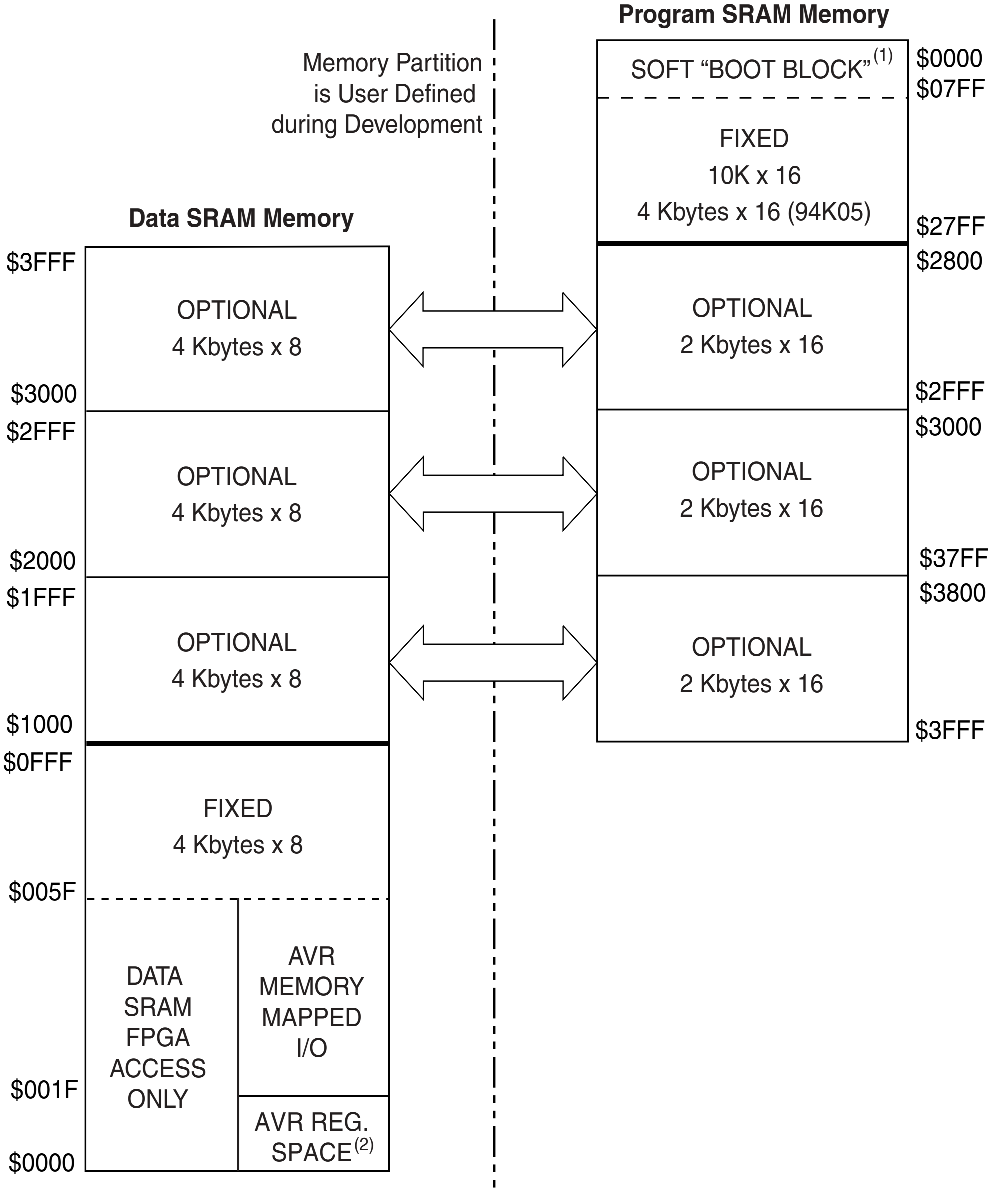
Figure 7. Some Single Cell Modes



RAM

There are two types of RAM in the FPSLIC device: the FreeRAM distributed through the FPGA Core and the SRAM shared by the AVR and FPGA. The SRAM is described in "FPGA/AVR Interface and System Control" on page 21. The 32 x 4 dual-ported FPGA FreeRAM blocks are dispersed throughout the array and are connected in each sector as shown in Figure 8. A four-bit Input Data bus connects to four horizontal local buses (Plane 1) distributed over four sector rows. A four-bit Output Data bus connects to four horizontal local buses (Plane 2) distributed over four sector rows. A five-bit Input-address bus connects to five vertical express buses in the same sector column (column 3). A five-bit Output-address bus connects to five vertical express buses in the same column. WAddr (Write Address) and RAddr (Read Address) alternate positions in horizontally aligned RAM blocks. For the left-

Figure 19. FPSLIC Configurable Allocation SRAM Memory[1][2]



Notes:
1. The Soft "BOOT BLOCK" is an area of memory that is first loaded when the part is powered up and configured. The remainder of the memory can be reprogrammed while the device is in operation for switching functions in and out of memory. The Soft "BOOT BLOCK" can only be programmed by a full device configuration on power-up.
2. The lower portion of the Data memory is not shared between the AVR and FPGA. The AVR uses addresses $0000 - $001F for the AVR CPU general working registers. $001F - $005F are the addresses used for Memory Mapped I/O and store the information in dedicated registers. Therefore, on the FPGA side $0000 - $005F are available for data that is only needed by the FPGA.

Memory-mapped I/O

The I/O space definition of the embedded AVR core is shown in the following table:

AT94K Register Summary

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reference Page
$3F ($5F)	SREG	I	T	H	S	V	N	Z	C	51
$3E ($5E)	SPH	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	57
$3D ($5D)	SPL	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	51
$3C ($5C)	Reserved									
$3B ($5B)	EIMF	INTF3	INTF2	INTF1	INTF0	INT3	INT2	INT1	INT0	62
$3A ($5A)	SFTCR					FMXOR	WDTS	DBG	SRST	51
$39 ($59)	TIMSK	TOIE1	OCIE1A	OCIE1B	TOIE2	TICIE1	OCIE2	TOIE0	OCIE0	62
$38 ($58)	TIFR	TOV1	OCF1A	OCF1B	TOV2	ICF1	OCF2	TOV0	OCF0	63
$37 ($57)	Reserved									
$36 ($56)	TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN		TWIE	110
$35 ($55)	MCUR	JTRF	JTD	SE	SM1	SM0	PORF	WDRF	EXTRF	51
$34 ($54)	Reserved									
$33 ($53)	TCCR0	FOC0	PWM0	COM01	COM00	CTC0	CS02	CS01	CS00	69
$32 ($52)	TCNT0	Timer/Counter0 (8-bit)								70
$31 ($51)	OCR0	Timer/Counter0 Output Compare Register								71
$30 ($50)	SFIOR							PSR2	PSR10	66
$2F ($4F)	TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	FOC1A	FOC1B	PWM11	PWM10	76
$2E ($4E)	TCCR1B	ICNC1	ICES1	ICPE		CTC1	CS12	CS11	CS10	77
$2D ($4D)	TCNT1H	Timer/Counter1 - Counter Register High Byte								78
$2C ($4C)	TCNT1L	Timer/Counter1 - Counter Register Low Byte								78
$2B ($4B)	OCR1AH	Timer/Counter1 - Output Compare Register A High Byte								79
$2A ($4A)	OCR1AL	Timer/Counter1 - Output Compare Register A Low Byte								79
$29 ($49)	OCR1BH	Timer/Counter1 - Output Compare Register B High Byte								79
$28 ($48)	OCR1BL	Timer/Counter1 - Output Compare Register B Low Byte								79
$27 ($47)	TCCR2	FOC2	PWM2	COM21	COM20	CTC2	CS22	CS21	CS20	69
$26 ($46)	ASSR					AS2	TCN20B	OCR2UB	TCR2UB	73
$25 ($45)	ICR1H	Timer/Counter1 - Input Capture Register High Byte								80
$24 ($44)	ICR1L	Timer/Counter1 - Input Capture Register Low Byte								80
$23 ($43)	TCNT2	Timer/Counter2 (8-bit)								70
$22 ($42)	OCR2	Timer/Counter 2 Output Compare Register								71
$21 ($41)	WDTCR				WDTOE	WDE	WDP2	WDP1	WDP0	83
$20 ($40)	UBRRHI	UART1 Baud Rate High Nibble [11..8]				UART0 Baud Rate Low Nibble [11..8]				105
$1F ($3F)	TWDR	2-wire Serial Data Register								111
$1E ($3E)	TWAR	2-wire Serial Address Register								112
$1D ($3D)	TWSR	2-wire Serial Status Register								112
$1C ($3C)	TWBR	2-wire Serial Bit Rate Register								109
$1B ($3B)	FPGAD	FPGA Cache Data Register (D7 - D0)								52
$1A ($3A)	FPGAZ	FPGA Cache Z Address Register (T3 - T0) (Z3 - Z0)								53
$19 ($39)	FPGAY	FPGA Cache Y Address Register (Y7 - Y0)								53
$18 ($38)	FPGAX	FPGA Cache X Address Register (X7 - X0)								53
$17 ($37)	FISUD	FPGA I/O Select, Interrupt Mask/Flag Register D (Reserved on AT94K05)								54, 56

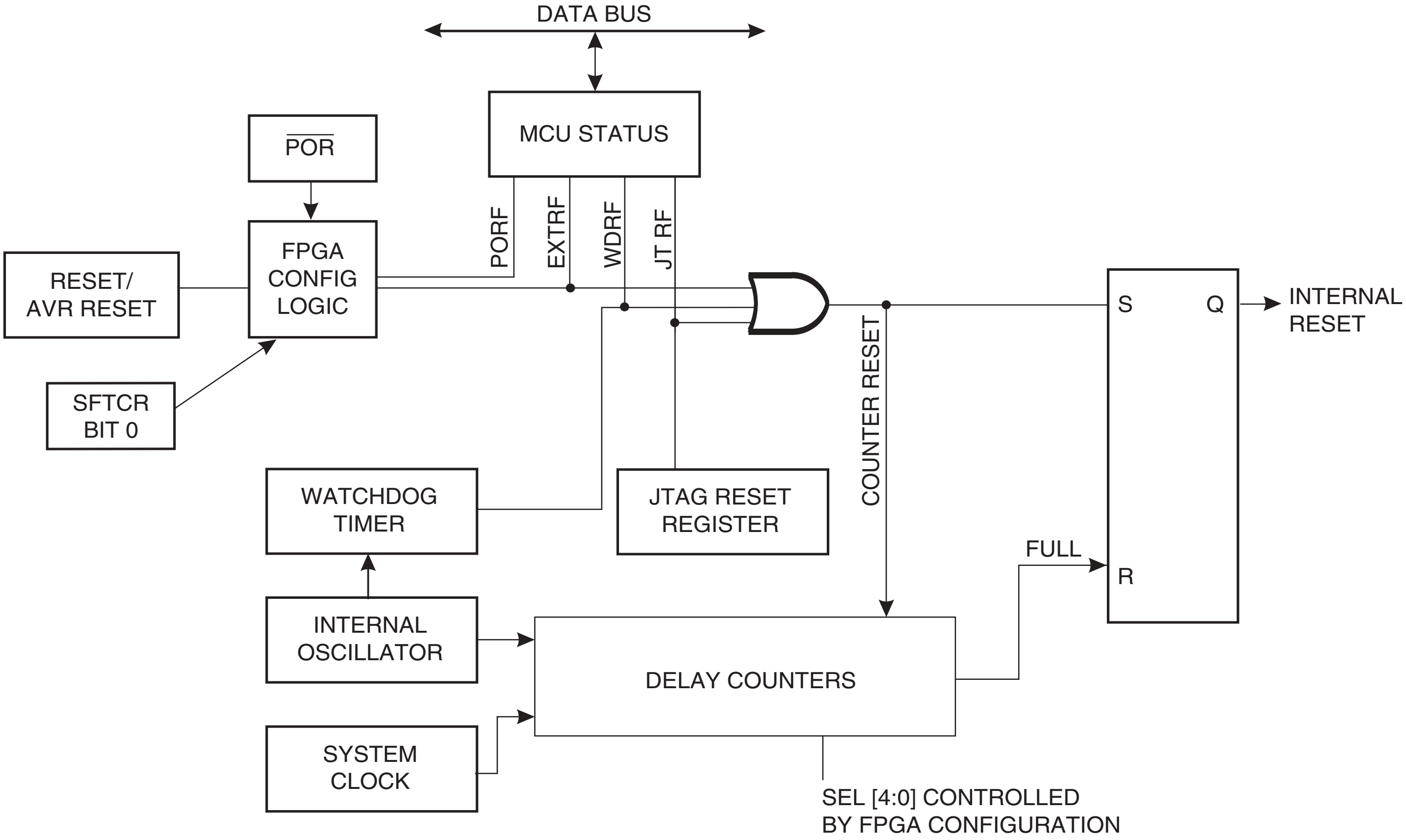
Reset Sources

The embedded AVR core has five sources of reset:

- External Reset. The MCU is reset immediately when a low-level is present on the RESET or AVR RESET pin.
- Power-on Reset. The MCU is reset upon chip power-up and remains in reset until the FPGA configuration has entered Idle mode.
- Watchdog Reset. The MCU is reset when the Watchdog Timer period expires and the watchdog is enabled.
- Software Reset. The MCU is reset when the SRST bit in the Software Control register is set (one).
- JTAG AVR Reset. The MCU is reset as long as there is a logic one in the Reset Register, one of the scan chains of the JTAG system. See “IEEE 1149.1 (JTAG) Boundary-scan” on page 73.

During reset, all I/O registers except the MCU Status register are then set to their Initial Values, and the program starts execution from address $0000. The instruction placed in address $0000 must be a JMP – absolute jump instruction to the reset handling routine. If the program never enables an interrupt source, the interrupt vectors are not used, and regular program code can be placed at these locations. The circuit diagram in Figure 35 shows the reset logic. Table 16 defines the timing and electrical parameters of the reset circuitry.

Figure 35. Reset Logic



Scanning an oscillator output gives unpredictable results as there is a frequency drift between the internal oscillator and the JTAG TCK clock.

The clock configuration is programmed in the SCR. As an SCR bit is not changed run-time, the clock configuration is considered fixed for a given application. The user is advised to scan the same clock option as to be used in the final system. The enable signals are supported in the scan chain because the system logic can disable clock options in sleep modes, thereby disconnecting the oscillator pins from the scan path if not provided.

The XTAL or TOSC “Clock In” Scan chain bit will always capture “1” if the oscillator is disabled (“Enable Clock” bit is active Low).

FPSLIC Boundary-scan Order

Table 20 shows the Scan order between TDI and TDO when the Boundary-Scan chain is selected as data path. Bit 0 is the LSB; the first bit scanned in, and the first bit scanned out. In Figure 43, “Data Out/In – PXn” corresponds to FF0, “Enable Output – PXn” corresponds to FF1, and “Pull-up – PXn” corresponds to FF2.

Table 20. AVR I/O Boundary Scan – JTAG Instructions $0/$2

I/O Ports	Description	Bit	
PORTE	Data Out/In - PE7	68	<- TDI
	Enable Output - PE7	67	
	Pull-up - PE7	66	
	Data Out/In - PE6	65	
	Enable Output - PE6	64	
	Pull-up - PE6	63	
	Data Out/In - PE5	62	
	Enable Output - PE5	61	
	Pull-up - PE5	60	
	Data Out/In - PE4	59	
	Enable Output - PE4	58	
	Pull-up - PE4	57	
	Data Out/In - PE3	56	
	Enable Output - PE3	55	
	Pull-up - PE3	54	
	Data Out/In - PE2	53	
	Enable Output - PE2	52	
	Pull-up - PE2	51	
	Data Out/In - PE1	50	
	Enable Output - PE1	49	
	Pull-up - PE1	48	
	Data Out/In - PE0	47	
	Enable Output - PE0	46	
	Pull-up - PE0	45	

- **Bits 5,4 - COM01, COM00/COM21, COM20: Compare Output Mode, Bits 1 and 0**

The COMn1 and COMn0 control bits determine any output pin action following a compare match in Timer/Counter0 or Timer/Counter2. Output pin actions affect pins PE1(OC0) or PE3(OC2). This is an alternative function to an I/O port, and the corresponding direction control bit must be set (one) to control an output pin. The control configuration is shown in Table 22.

Table 22. Compare Output Mode Select[1]

COMn1	COMn0	Description
0	0	Timer/Counter disconnected from output pin OCn[2]
0	1	Toggles the OCn[2] output line.
1	0	Clears the OCn[2] output line (to zero).
1	1	Sets the OCn[2] output line (to one).

Notes: 1. In PWM mode, these bits have a different function. Refer to Table 25 for a detailed description.
2. n = 0 or 2

- **Bit 3 - CTC0/CTC2: Clear Timer/Counter on Compare Match**

When the CTC0 or CTC2 control bit is set (one), Timer/Counter0 or Timer/Counter2 is reset to $00 in the CPU clock-cycle after a compare match. If the control bit is cleared, Timer/Counter continues counting and is unaffected by a compare match. When a prescaling of 1 is used, and the compare register is set to C, the timer will count as follows if CTC0/CTC2 is set:

... | C-1 | C | 0 | 1 | ...

When the prescaler is set to divide by 8, the timer will count like this:

... | C-1, C-1, C-1, C-1, C-1, C-1, C-1, C-1 | C, C, C, C, C, C, C, C | 0, 0, 0, 0, 0, 0, 0, 0 | 1, 1, 1, ...

In PWM mode, this bit has a different function. If the CTC0 or CTC2 bit is cleared in PWM mode, the Timer/Counter acts as an up/down counter. If the CTC0 or CTC2 bit is set (one), the Timer/Counter wraps when it reaches $FF. Refer to page 91 for a detailed description.

- **Bits 2,1,0 - CS02, CS01, CS00/ CS22, CS21, CS20: Clock Select Bits 2,1 and 0**

The Clock Select bits 2,1 and 0 define the prescaling source of Timer/Counter0 and Timer/Counter2, see Table 23 and Table 24.

Table 23. Clock 0 Prescale Select

CS02	CS01	CS00	Description
0	0	0	Stop, the Timer/Counter0 is stopped
0	0	1	CK
0	1	0	CK/8
0	1	1	CK/64
1	0	0	CK/256
1	0	1	CK/1024
1	1	0	External pin PE0(T0), falling edge
1	1	1	External pin PE0(T0), rising edge

Table 24. Clock 2 Prescale Select

CS22	CS21	CS20	Description
0	0	0	Stop, the Timer/Counter2 is stopped
0	0	1	PCK2
0	1	0	PCK2/8
0	1	1	PCK2/32
1	0	0	PCK2/64
1	0	1	PCK2/128
1	1	0	PCK2/256
1	1	1	PCK2/1024

The Stop condition provides a Timer Enable/Disable function. The prescaled modes are scaled directly from the CK oscillator clock for Timer/Counter0 and PCK2 for Timer/Counter2. If the external pin modes are used for Timer/Counter0, transitions on PE0/(T0) will clock the counter even if the pin is configured as an output. This feature can give the user SW control of the counting.

Timer Counter0 – TCNT0

Bit	7	6	5	4	3	2	1	0	
$32 ($52)	**MSB**							**LSB**	**TCNT0**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Timer/Counter2 – TCNT2

Bit	7	6	5	4	3	2	1	0	
$23 ($43)	**MSB**							**LSB**	**TCNT2**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

These 8-bit registers contain the value of the Timer/Counters.

Both Timer/Counters are realized as up or up/down (in PWM mode) counters with read and write access. If the Timer/Counter is written to and a clock source is selected, it continues counting in the timer clock cycle following the write operation.

Timer/Counter0 Output Compare Register – OCR0

Bit	7	6	5	4	3	2	1	0	
$31 ($51)	**MSB**							**LSB**	**OCR0**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Timer/Counter2 Output Compare Register – OCR2

Bit	7	6	5	4	3	2	1	0	
$22 ($42)	**MSB**							**LSB**	**OCR2**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The output compare registers are 8-bit read/write registers. The Timer/Counter Output Compare Registers contains the data to be continuously compared with the Timer/Counter. Actions on compare matches are specified in TCCR0 and TCCR2. A compare match does only occur if the Timer/Counter counts to the OCR value. A software write that sets Timer/Counter and Output Compare Register to the same value does not generate a compare match.

A compare match will set the compare interrupt flag in the CPU clock-cycle following the compare event.

Timer/Counter 0 and 2 in PWM Mode

When PWM mode is selected, the Timer/Counter either wraps (overflows) when it reaches $FF or it acts as an up/down counter.

If the up/down mode is selected, the Timer/Counter and the Output Compare Registers – OCR0 or OCR2 form an 8-bit, free-running, glitch-free and phase correct PWM with outputs on the PE1(OC0/PWM0) or PE3(OC2/PWM2) pin.

If the overflow mode is selected, the Timer/Counter and the Output Compare Registers – OCR0 or OCR2 form an 8-bit, free-running and glitch-free PWM, operating with twice the speed of the up/down counting mode.

PWM Modes (Up/Down and Overflow)

The two different PWM modes are selected by the CTC0 or CTC2 bit in the Timer/Counter Control Registers – TCCR0 or TCCR2 respectively.

If CTC0/CTC2 is cleared and PWM mode is selected, the Timer/Counter acts as an up/down counter, counting up from $00 to $FF, where it turns and counts down again to zero before the cycle is repeated. When the counter value matches the contents of the Output Compare Register, the PE1(OC0/PWM0) or PE3(OC2/PWM2) pin is set or cleared according to the settings of the COMn1/COMn0 bits in the Timer/Counter Control Registers TCCR0 or TCCR2.

If CTC0/CTC2 is set and PWM mode is selected, the Timer/Counters will wrap and start counting from $00 after reaching $FF. The PE1(OC0/PWM0) or PE3(OC2/PWM2) pin will be set or cleared according to the settings of COMn1/COMn0 on a Timer/Counter overflow or when the counter value matches the contents of the Output Compare Register. Refer to Table 25 for details.

Table 25. Compare Mode Select in PWM Mode

CTCn[1]	COMn1[1]	COMn0[1]	Effect on Compare Pin	Frequency
x[2]	0	x[2]	Not connected	–
0	1	1	Cleared on compare match, up-counting. Set on compare match, down-counting (non-inverted PWM)	$f_{TCK0/2}$/510
0	1	1	Cleared on compare match, down-counting. Set on compare match, up-counting (inverted PWM)	$f_{TCK0/2}$/510
1	1	0	Cleared on compare match, set on overflow	$f_{TCK0/2}$/256
1	1	1	Set on compare match, set on overflow	$f_{TCK0/2}$/256

Notes: 1. n = 0 or 2
2. x = don' t care

In PWM mode, the value to be written to the Output Compare Register is first transferred to a temporary location, and then latched into the OCR when the Timer/Counter reaches $FF. This prevents the occurrence of odd-length PWM pulses (glitches) in the event of an unsynchronized OCR0 or OCR2 write. See Figure 52 and Figure 53 for examples.

fmuls16x16_32

Description

Signed fractional multiply of two 16-bit numbers with a 32-bit result.

Usage

R19:R18:R17:R16 = (R23:R22 • R21:R20) << 1

Statistics

Cycles: 20 + ret

Words: 16 + ret

Register usage: R0 to R2 and R16 to R23 (11 registers)[1]

Note: [1] The routine is non-destructive to the operands.

```
fmuls16x16_32:
   clr   r2
   fmuls r23, r21             ; ( (signed)ah * (signed)bh ) << 1
   movw  r19:r18, r1:r0
   fmul  r22, r20             ; ( al * bl ) << 1
   adc   r18, r2
   movw  r17:r16, r1:r0
   fmulsu   r23, r20          ; ( (signed)ah * bl ) << 1
   sbc   r19, r2              ; Sign extend
   add   r17, r0
   adc   r18, r1
   adc   r19, r2
   fmulsu   r21, r22          ; ( (signed)bh * al ) << 1
   sbc   r19, r2              ; Sign extend
   add   r17, r0
   adc   r18, r1
   adc   r19, r2
   ret
```

fmac16x16_32

Description

Signed fractional multiply-accumulate of two 16-bit numbers with a 32-bit result.

Usage

R19:R18:R17:R16 += (R23:R22 • R21:R20) << 1

Statistics

Cycles: 25 + ret

Words: 21 + ret

Register usage: R0 to R2 and R16 to R23 (11 registers)

```
fmac16x16_32:                 ; Register usage optimized
   clr   r2

   fmuls r23, r21             ; ( (signed)ah * (signed)bh ) << 1
   add   r18, r0
   adc   r19, r1

   fmul  r22, r20             ; ( al * bl ) << 1
   adc   r18, r2
   adc   r19, r2
   add   r16, r0
```

Multi-processor Communication Mode

The Multi-processor Communication Mode enables several Slave MCUs to receive data from a Master MCU. This is done by first decoding an address byte to find out which MCU has been addressed. If a particular Slave MCU has been addressed, it will receive the following data bytes as normal, while the other Slave MCUs will ignore the data bytes until another address byte is received.

For an MCU to act as a Master MCU, it should enter 9-bit transmission mode (CHR9n in UCSRnB set). The 9-bit must be one to indicate that an address byte is being transmitted, and zero to indicate that a data byte is being transmitted.

For the Slave MCUs, the mechanism appears slightly different for 8-bit and 9-bit Reception mode. In 8-bit Reception mode (CHR9n in UCSRnB cleared), the stop bit is one for an address byte and zero for a data byte. In 9-bit Reception mode (CHR9n in UCSRnB set), the 9-bit is one for an address byte and zero for a data byte, whereas the stop bit is always High.

The following procedure should be used to exchange data in Multi-processor Communication mode:

1. All Slave MCUs are in Multi-processor Communication Mode (MPCMn in UCSRnA is set).
2. The Master MCU sends an address byte, and all Slaves receive and read this byte. In the Slave MCUs, the RXCn flag in UCSRnA will be set as normal.
3. Each Slave MCU reads the UDRn register and determines if it has been selected. If so, it clears the MPCMn bit in UCSRnA, otherwise it waits for the next address byte.
4. For each received data byte, the receiving MCU will set the receive complete flag (RXCn in UCSRnA. In 8-bit mode, the receiving MCU will also generate a framing error (FEn in UCSRnA set), since the stop bit is zero. The other Slave MCUs, which still have the MPCMn bit set, will ignore the data byte. In this case, the UDRn register and the RXCn, FEn, or flags will not be affected.
5. After the last byte has been transferred, the process repeats from step 2.

UART Control

UART0 I/O Data Register – UDR0

Bit	7	6	5	4	3	2	1	0	
$0C ($2C)	MSB							LSB	UDR0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

UART1 I/O Data Register – UDR1

Bit	7	6	5	4	3	2	1	0	
$03 ($23)	MSB							LSB	UDR1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The UDRn register is actually two physically separate registers sharing the same I/O address. When writing to the register, the UART Transmit Data register is written. When reading from UDRn, the UART Receive Data register is read.

UART0 Control and Status Registers – UCSR0A

Bit	7	6	5	4	3	2	1	0	
$0B ($2B)	RXC0	TXC0	UDRE0	FE0	OR0	-	U2X0	MPCM0	UCSR0A
Read/Write	R	R/W	R	R	R	R	R/W	R/W	
Initial Value	0	0	1	0	0	0	0	0	

UART1 Control and Status Registers – UCSR1A

Bit	7	6	5	4	3	2	1	0	
$02 ($22)	RXC1	TXC1	UDRE1	FE1	OR1	-	U2X1	MPCM1	UCSR1A
Read/Write	R	R/W	R	R	R	R	R/W	R/W	
Initial Value	0	0	1	0	0	0	0	0	

- **Bit 7 - RXC0/RXC1: UART Receive Complete**

This bit is set (one) when a received character is transferred from the Receiver Shift register to UDRn. The bit is set regardless of any detected framing errors. When the RXCIEn bit in UCSRnB is set, the UART Receive Complete interrupt will be executed when RXCn is set (one). RXCn is cleared by reading UDRn. When interrupt-driven data reception is used, the UART Receive Complete Interrupt routine must read UDRn in order to clear RXCn, otherwise a new interrupt will occur once the interrupt routine terminates.

- **Bit 6 - TXC0/TXC1: UART Transmit Complete**

This bit is set (one) when the entire character (including the stop bit) in the Transmit Shift register has been shifted out and no new data has been written to UDRn. This flag is especially useful in half-duplex communications interfaces, where a transmitting application must enter receive mode and free the communications bus immediately after completing the transmission.

When the TXCIEn bit in UCSRnB is set, setting of TXCn causes the UART Transmit Complete interrupt to be executed. TXCn is cleared by the hardware when executing the corresponding interrupt handling vector. Alternatively, the TXCn bit is cleared (zero) by writing a logic 1 to the bit.

- **Bit 5 - UDRE0/UDRE1: UART Data Register Empty**

This bit is set (one) when a character written to UDRn is transferred to the Transmit shift register. Setting of this bit indicates that the transmitter is ready to receive a new character for transmission.

When the UDRIEn bit in UCSRnB is set, the UART Transmit Complete interrupt will be executed as long as UDREn is set and the global interrupt enable bit in SREG is set. UDREn is cleared by writing UDRn. When interrupt-driven data transmittal is used, the UART Data Register Empty Interrupt routine must write UDRn in order to clear UDREn, otherwise a new interrupt will occur once the interrupt routine terminates.

UDREn is set (one) during reset to indicate that the transmitter is ready.

- **Bit 4 - FE0/FE1: Framing Error**

This bit is set if a Framing Error condition is detected, i.e., when the stop bit of an incoming character is zero.

The FEn bit is cleared when the stop bit of received data is one.

- **Bits 7..0 - 2-wire Serial Bit-rate Register**

TWBR selects the division factor for the bit-rate generator. The bit-rate generator is a frequency divider which generates the SCL clock frequency in the Master modes according to the following equation:

$$\text{Bit-rate} = \frac{f_{CK}}{16 + 2(\text{TWBR})}$$

- Bit-rate = SCL frequency
- f_{CK} = CPU Clock frequency
- TWBR = Contents of the 2-wire Serial Bit Rate Register

Both the receiver and the transmitter can stretch the Low period of the SCL line when waiting for user response, thereby reducing the average bit rate.

The 2-wire Serial Control Register – TWCR

Bit	7	6	5	4	3	2	1	0	
$36 ($56)	**TWINT**	**TWEA**	**TWSTA**	**TWSTO**	**TWWC**	**TWEN**	**-**	**TWIE**	**TWCR**
Read/Write	R/W	R/W	R/W	R/W	R	R/W	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 - TWINT: 2-wire Serial Interrupt Flag**

This bit is set by the hardware when the 2-wire Serial Interface has finished its current job and expects application software response. If the I-bit in the SREG and TWIE in the TWCR register are set (one), the MCU will jump to the interrupt vector at address $0046. While the TWINT flag is set, the bus SCL clock line Low period is stretched. The TWINT flag must be cleared by software by writing a logic 1 to it. Note that this flag is not automatically cleared by the hardware when executing the interrupt routine. Also note that clearing this flag starts the operation of the 2-wire Serial Interface, so all accesses to the 2-wire Serial Address Register – TWAR, 2-wire Serial Status Register – TWSR, and 2-wire Serial Data Register – TWDR must be complete before clearing this flag.

- **Bit 6 - TWEA: 2-wire Serial Enable Acknowledge Flag**

TWEA flag controls the generation of the acknowledge pulse. If the TWEA bit is set, the ACK pulse is generated on the 2-wire Serial Bus if the following conditions are met:

- The device’s own Slave address has been detected
- A general call has been received, while the TWGCE bit in the TWAR is set
- A data byte has been received in Master Receiver or Slave Receiver mode

By setting the TWEA bit Low the device can be virtually disconnected from the 2-wire Serial Bus temporarily. Address recognition can then be resumed by setting the TWEA bit again.

- **Bit 5 - TWSTA: 2-wire Serial Bus START Condition Flag**

The TWSTA flag is set by the CPU when it desires to become a Master on the 2-wire Serial Bus. The 2-wire serial hardware checks if the bus is available, and generates a Start condition on the bus if the bus is free. However, if the bus is not free, the 2-wire Serial Interface waits until a STOP condition is detected, and then generates a new Start condition to claim the bus Master status.

Figure 71. Formats and States in the Master Transmitter Mode

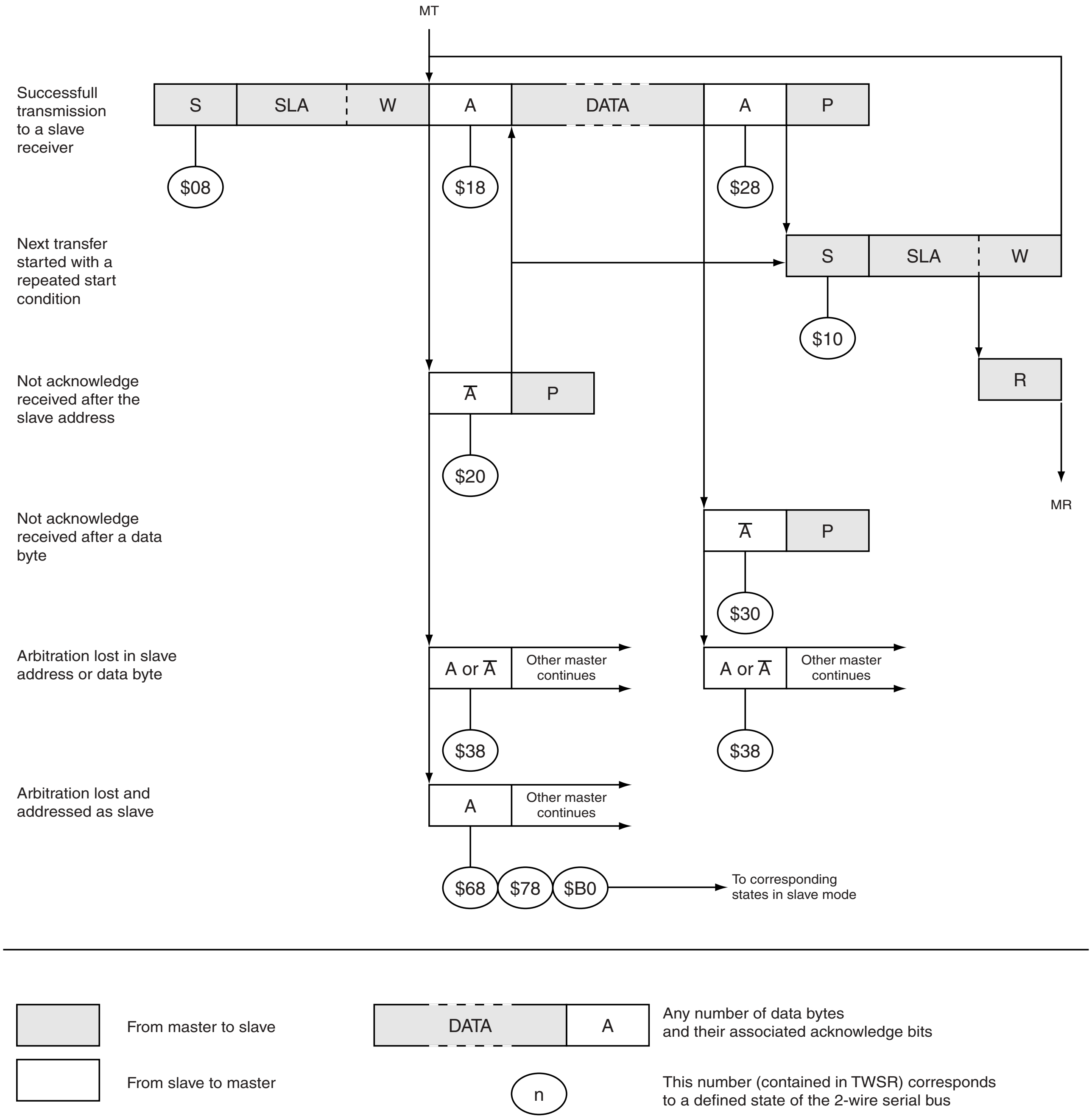
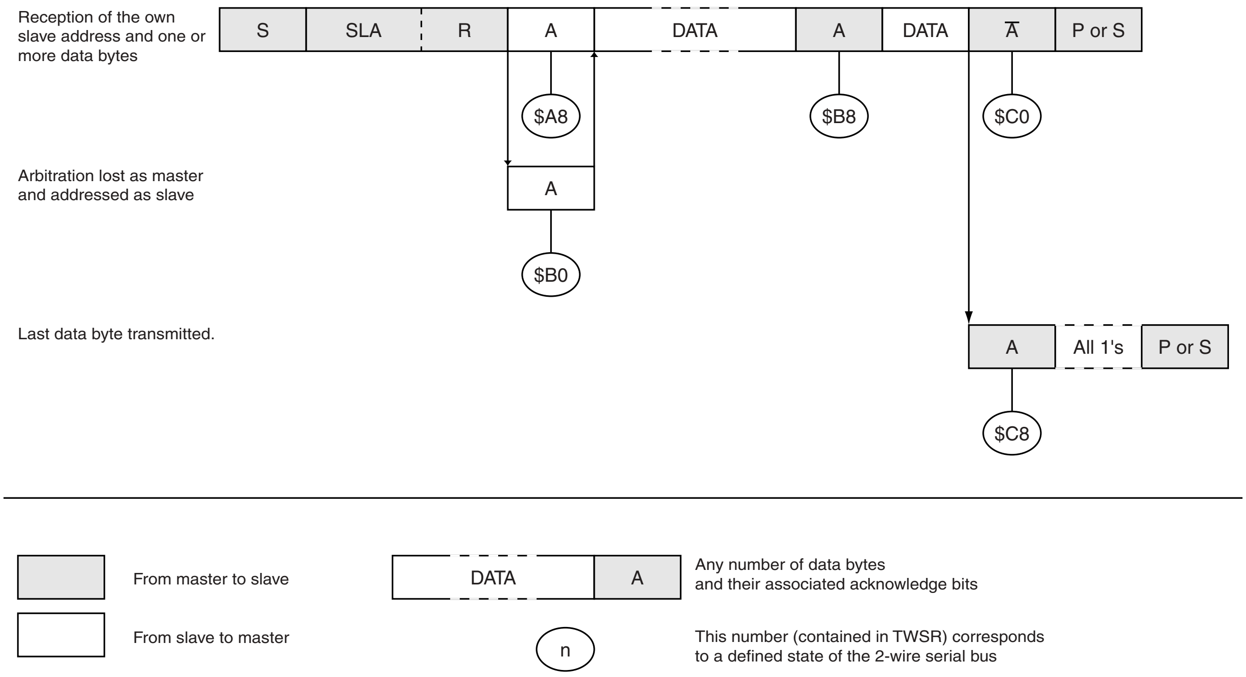

MT
Successfull transmission to a slave receiver
S
SLA
W
A
DATA
A
P
$08
$18
$28
Next transfer started with a repeated start condition
S
SLA
W
$10
R
MR
Not acknowledge received after the slave address
$\overline{A}$
P
$20
Not acknowledge received after a data byte
$\overline{A}$
P
$30
Arbitration lost in slave address or data byte
A or $\overline{A}$
Other master continues
A or $\overline{A}$
Other master continues
$38
$38
Arbitration lost and addressed as slave
A
Other master continues
$68
$78
$B0
To corresponding states in slave mode
From master to slave
From slave to master
DATA
A
Any number of data bytes and their associated acknowledge bits
n
This number (contained in TWSR) corresponds to a defined state of the 2-wire serial bus

Table 43. Status Codes for Slave Receiver Mode

Status Code (TWSR)	Status of the 2-wire Serial Bus and 2-wire Serial Hardware	Application Software Response					Next Action Taken by 2-wire Serial Hardware
			To TWCR				
		To/From TWDR	STA	STO	TWINT	TWEA	
$60	Own SLA+W has been received; ACK has been returned	No TWDR action or	X	0	1	0	Data byte will be received and NOT ACK will be returned
		No TWDR action	X	0	1	1	Data byte will be received and ACK will be returned
$68	Arbitration lost in SLA+R/W as Master; own SLA+W has been received; ACK has been returned	No TWDR action or	X	0	1	0	Data byte will be received and NOT ACK will be returned
		No TWDR action	X	0	1	1	Data byte will be received and ACK will be returned
$70	General call address has been received; ACK has been returned	No TWDR action or	X	0	1	0	Data byte will be received and NOT ACK will be returned
		No TWDR action	X	0	1	1	Data byte will be received and ACK will be returned
$78	Arbitration lost in SLA+R/W as Master; General call address has been received; ACK has been returned	No TWDR action or	X	0	1	0	Data byte will be received and NOT ACK will be returned
		No TWDR action	X	0	1	1	Data byte will be received and ACK will be returned
$80	Previously addressed with own SLA+W; data has been received; ACK has been returned	No TWDR action or	X	0	1	0	Data byte will be received and NOT ACK will be returned
		No TWDR action	X	0	1	1	Data byte will be received and ACK will be returned
$88	Previously addressed with own SLA+W; data has been received; NOT ACK has been returned	Read data byte or	0	0	1	0	Switched to the not addressed Slave mode; no recognition of own SLA or GCA
		Read data byte or	0	0	1	1	Switched to the not addressed Slave mode; own SLA will be recognized; GCA will be recognized if GC = "1"
		Read data byte or	1	0	1	0	Switched to the not addressed Slave mode; no recognition of own SLA or GCA; a START condition will be transmitted when the bus becomes free
		Read data byte	1	0	1	1	Switched to the not addressed Slave mode; own SLA will be recognized; GCA will be recognized if GC = "1"; a START condition will be transmitted when the bus becomes free

Figure 74. Formats and States in the Slave Transmitter Mode



Reception of the own slave address and one or more data bytes
S
SLA
R
A
DATA
A
DATA
A̅
P or S
$A8
$B8
$C0
Arbitration lost as master and addressed as slave
A
$B0
Last data byte transmitted.
A
All 1's
P or S
$C8
From master to slave
From slave to master
DATA
A
Any number of data bytes and their associated acknowledge bits
n
This number (contained in TWSR) corresponds to a defined state of the 2-wire serial bus

Table 45. Status Codes for Miscellaneous States

Status Code (TWSR)	Status of the 2-wire Serial Bus and 2-wire Serial Hardware	Application Software Response					Next Action Taken by 2-wire Serial Hardware
			To TWCR				
		To/From TWDR	STA	STO	TWINT	TWEA	
$F8	No relevant state information available; TWINT = “0”	No TWDR action	No TWCR action				Wait or proceed current transfer
$00	Bus error due to an illegal START or STOP condition	No TWDR action	0	1	1	X	Only the internal hardware is affected; no STOP condition is sent on the bus. In all cases, the bus is released and TWSTO is cleared.

Alternate I/O Functions of PortE

PortE may also be used for various Timer/Counter functions, such as External Input Clocks (TC0 and TC1), Input Capture (TC1), Pulse Width Modulation (TC0, TC1 and TC2), and toggling upon an Output Compare (TC0, TC1 and TC2). For a detailed pinout description, consult Table 47 on page 149. For more information on the function of each pin, See “Timer/Counters” on page 85.

PortE Schematics

Note that all port pins are synchronized. The synchronization latches are, however, not shown in the figures.

Figure 76. PortE Schematic Diagram (Pin PE0)

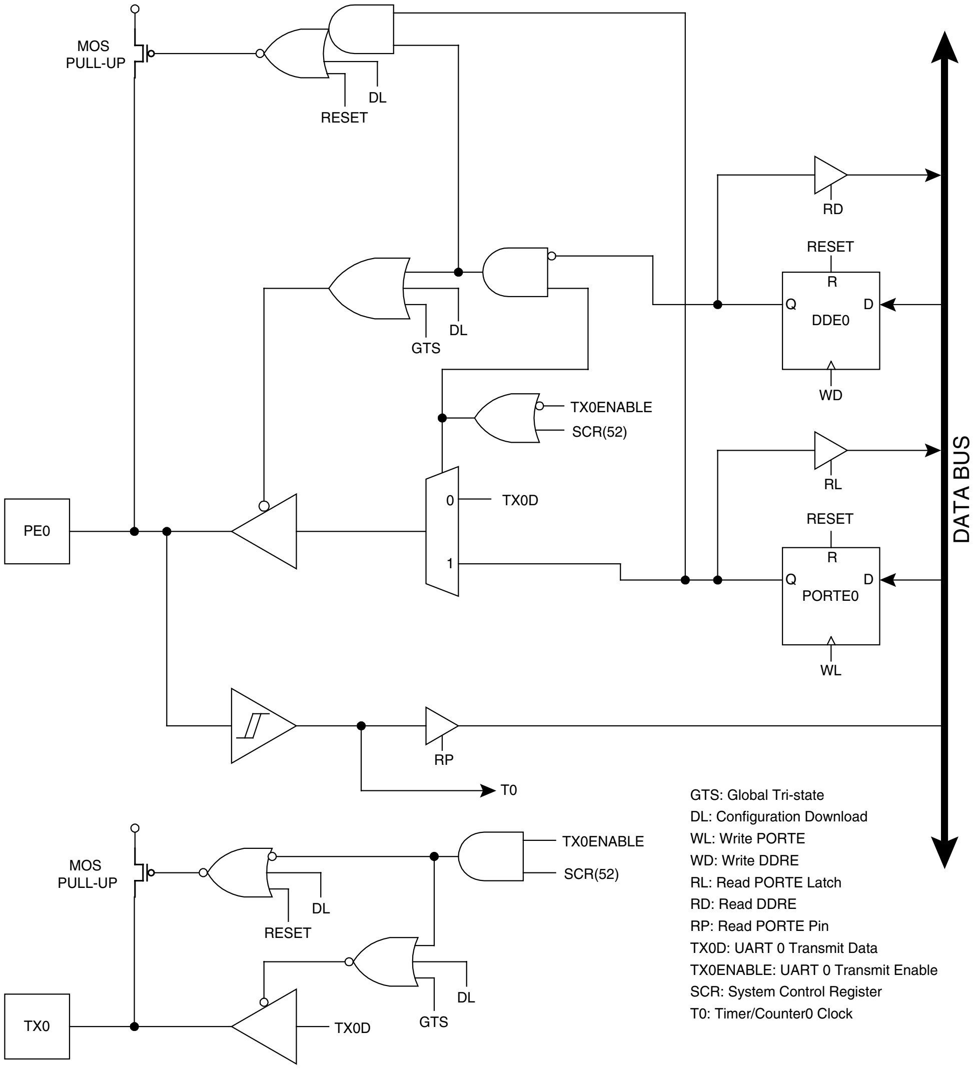
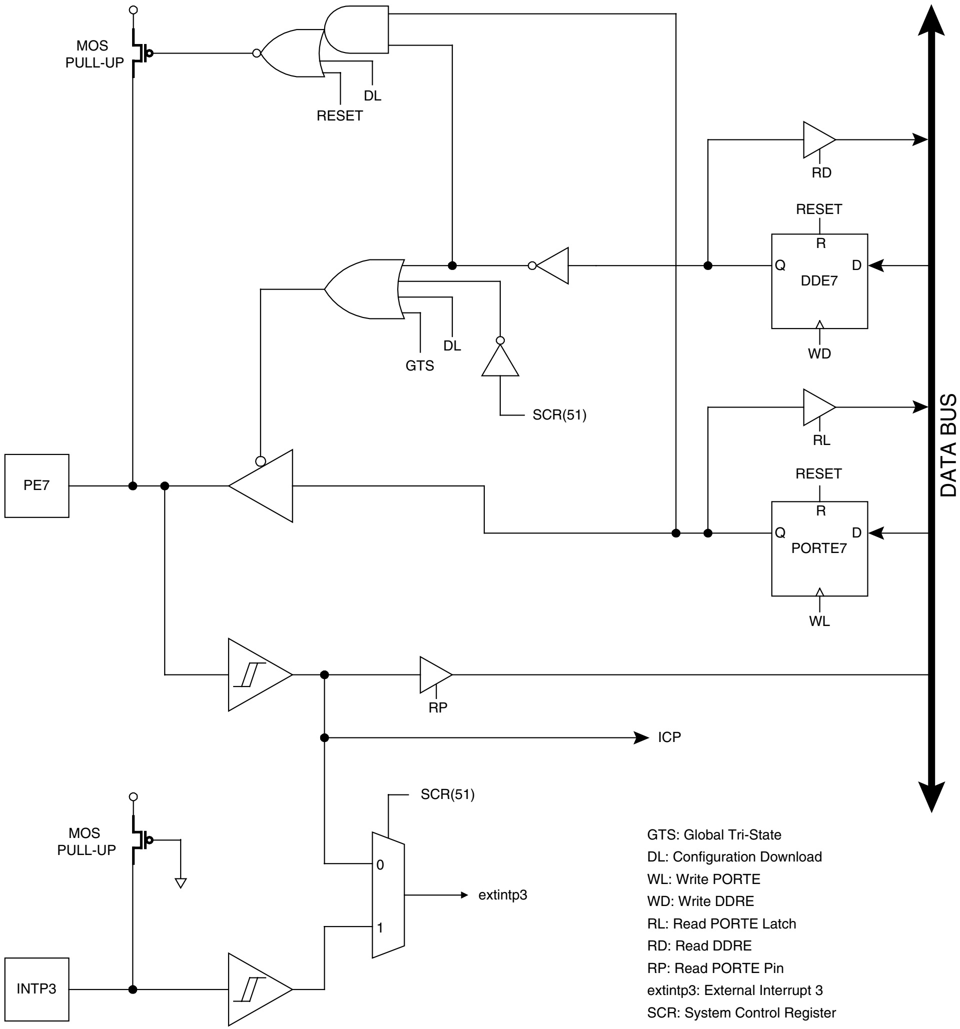


Figure 82. PortE Schematic Diagram (Pin PE7)



GTS: Global Tri-State
DL: Configuration Download
WL: Write PORTE
WD: Write DDRE
RL: Read PORTE Latch
RD: Read DDRE
RP: Read PORTE Pin
extintp3: External Interrupt 3
SCR: System Control Register
ICP: Timer/Counter Input Capture Pin

Table 56. AT94K Pin List (Continued)

AT94K05 96 FPGA I/O	AT94K10 192 FPGA I/O	AT94K40 384 FPGA I/O	Packages			
			PC84	TQ100	PQ144	PQ208
		I/O91				
		I/O92				
I/O29	I/O45	I/O93			30	44
I/O30	I/O46	I/O94			31	45
I/O31 ($\overline{\text{OTS}}$)	I/O47 ($\overline{\text{OTS}}$)	I/O95 ($\overline{\text{OTS}}$)	28	20	32	46
I/O32, GCK2 (A29)	I/O48, GCK2 (A29)	I/O96, GCK2 (A29)	29	21	33	47
$\overline{\text{AVRRESET}}$	$\overline{\text{AVRRESET}}$	$\overline{\text{AVRRESET}}$	30	22	34	48
GND	GND	GND	31	23	35	49
M0	M0	M0	32	24	36	50
South Side						
VCC[1]	VCC[1]	VCC[1]	33	25	37	55
M2	M2	M2	34	26	38	56
I/O33, GCK3	I/O49, GCK3	I/O97, GCK3	35	27	39	57
I/O34 (HDC/TDI)	I/O50 (HDC/TDI)	I/O98 (HDC/TDI)	36	28	40	58
I/O35	I/O51	I/O99			41	59
I/O36	I/O52	I/O100			42	60
I/O37 Not a User I/O	I/O53 Not a User I/O	I/O101		29	43	61
I/O38 (LDC/TDO)	I/O54 (LDC/TDO)	I/O102 (LDC/TDO)	37	30	44	62
		GND				
		I/O103				
		I/O104				
		I/O105				
		I/O106				
		I/O107				
		I/O108				
		VCC[1]				
		GND				
I/O39	I/O55	I/O109				63
I/O40	I/O56	I/O110				64
	I/O57	I/O111				65
	I/O58	I/O112				66

Notes:
1. VCC is I/O high voltage. Please refer to the “Designing in Split Power Supply Support for AT94KAL and AT94SAL Devices” application note.
2. VDD is core high voltage. Please refer to the “Designing in Split Power Supply Support for AT94KAL and AT94SAL Devices” application note.
3. Unbonded pins are No Connects.

Table 56. AT94K Pin List (Continued)

AT94K05 96 FPGA I/O	AT94K10 192 FPGA I/O	AT94K40 384 FPGA I/O	Packages			
			PC84	TQ100	PQ144	PQ208
I/O55	I/O83	I/O167			62	88
I/O56	I/O84	I/O168			63	89
GND	GND	GND			64	90
		I/O169				
		I/O170				
	I/O85	I/O171				
	I/O86	I/O172				
		I/O173				
		I/O174				
		GND				
		I/O175				
		I/O176				
	I/O87	I/O177				91
	I/O88	I/O178				92
I/O57	I/O89	I/O179				93
I/O58	I/O90	I/O180				94
		GND				
		VCC[1]				
		I/O181				
		I/O182				
I/O59 (TD2)	I/O91 (TD2)	I/O183 (TD2)	48	45	65	95
I/O60 (TD1)	I/O92 (TD1)	I/O184 (TD1)	49	46	66	96
		I/O185				
		I/O186				
		GND				
		I/O187				
		I/O188				
I/O61	I/O93	I/O189			67	97
I/O62	I/O94	I/O190			68	98
I/O63 (TD0)	I/O95 (TD0)	I/O191 (TD0)	50	47	69	99
I/O64, GCK4	I/O96, GCK4	I/O192, GCK4	51	48	70	100
GND	GND	GND	52	49	71	101
$\overline{\text{CON}}$	$\overline{\text{CON}}$	$\overline{\text{CON}}$	53	50	72	103
East Side						

Notes: 1. VCC is I/O high voltage. Please refer to the “Designing in Split Power Supply Support for AT94KAL and AT94SAL Devices” application note.
2. VDD is core high voltage. Please refer to the “Designing in Split Power Supply Support for AT94KAL and AT94SAL Devices” application note.
3. Unbonded pins are No Connects.

Table 56. AT94K Pin List (Continued)

AT94K05 96 FPGA I/O	AT94K10 192 FPGA I/O	AT94K40 384 FPGA I/O	Packages			
			PC84	**TQ100**	**PQ144**	**PQ208**
VCC[1]	VCC[1]	VCC[1]	54	51	73	106
$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	55	52	74	108
PE0	PE0	PE0	56	53	75	109
PE1	PE1	PE1	57	54	76	110
PD0	PD0	PD0			77	111
PD1	PD1	PD1			78	112
		GND				
		VCC[1]				
		GND				
PE2	PE2	PE2	58	55	79	113
PD2	PD2	PD2		56	80	114
		GND				
No Connect	No Connect	No Connect			81	119
PD3	PD3	PD3			82	120
PD4	PD4	PD4			83	121
	VCC[1]	VCC[1]				
PE3	PE3	PE3	59	57	84	122
$\overline{\text{CS0}}$, Cs0n	$\overline{\text{CS0}}$, Cs0n	$\overline{\text{CS0}}$, Cs0n	60	58	85	123
		GND				
		GND				
		VCC[1]				
SDA	SDA	SDA				124
SCL	SCL	SCL				125
		GND				
PD5	PD5	PD5		59	86	126
PD6	PD6	PD6		60	87	127
PE4	PE4	PE4	61	61	88	128
PE5	PE5	PE5	62	62	89	129
VDD[2]	VDD[2]	VDD[2]	63	63	90	130
GND	GND	GND	64	64	91	131
PE6	PE6	PE6	65	65	92	132
PE7 ($\overline{\text{CHECK}}$)	PE7 ($\overline{\text{CHECK}}$)	PE7 ($\overline{\text{CHECK}}$)	66	66	93	133
PD7	PD7	PD7		67	94	134

Notes:
1. VCC is I/O high voltage. Please refer to the “Designing in Split Power Supply Support for AT94KAL and AT94SAL Devices” application note.
2. VDD is core high voltage. Please refer to the “Designing in Split Power Supply Support for AT94KAL and AT94SAL Devices” application note.
3. Unbonded pins are No Connects.