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	18 MHz
Interface	I²C, UART
Program SRAM Bytes	20K-32K
FPGA SRAM	18kb
EEPROM Size	-
Data SRAM Bytes	4K ~ 16K
FPGA Core Cells	2304
FPGA Gates	40K
FPGA Registers	2862
Voltage - Supply	3V ~ 3.6V
Mounting Type	Surface Mount
Operating Temperature	0°C ~ 70°C
Package / Case	144-LQFP
Supplier Device Package	144-LQFP (20x20)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/at94k40al-25bqc

Figure 4. Busing Plane (One of Five)

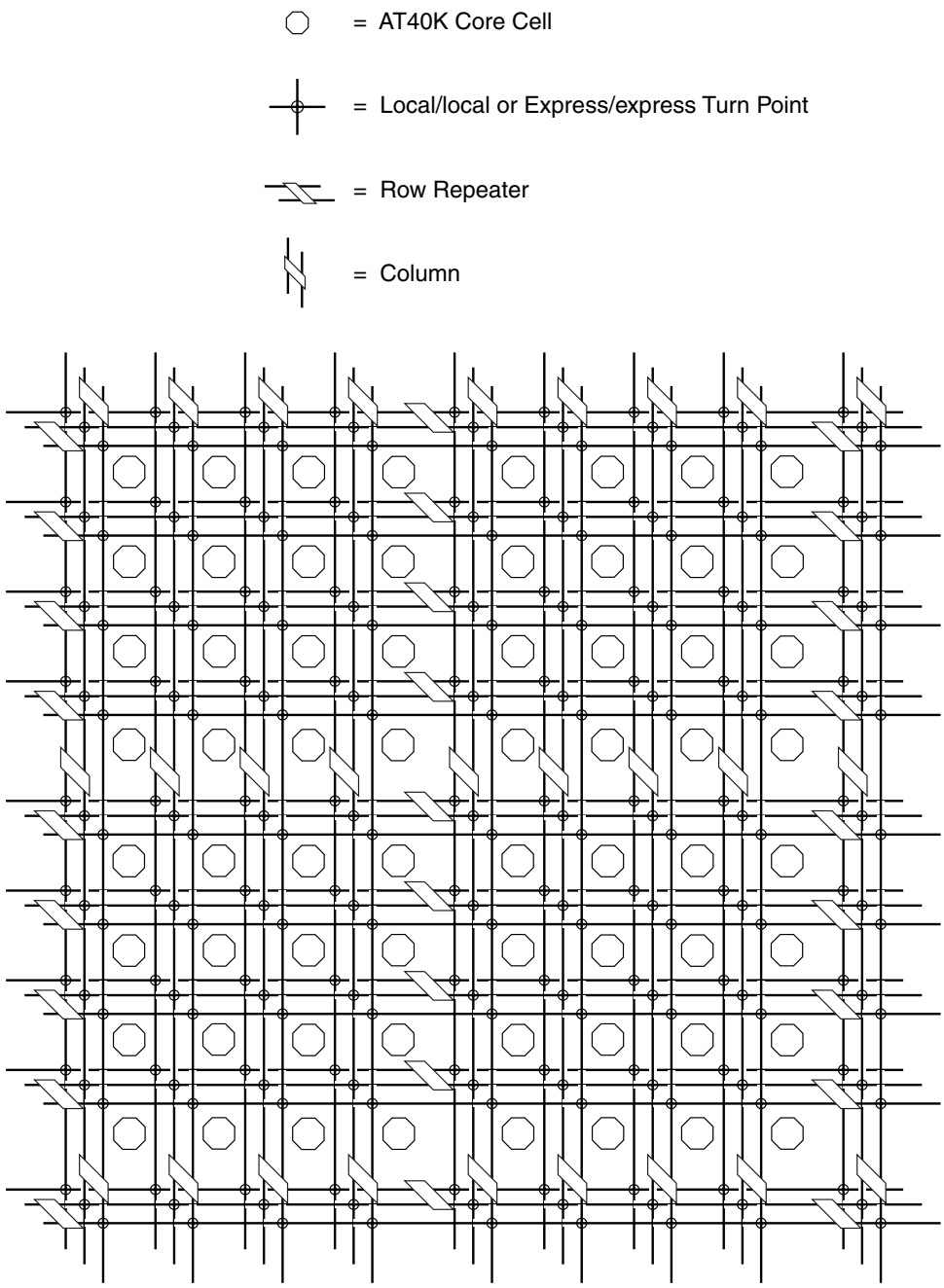
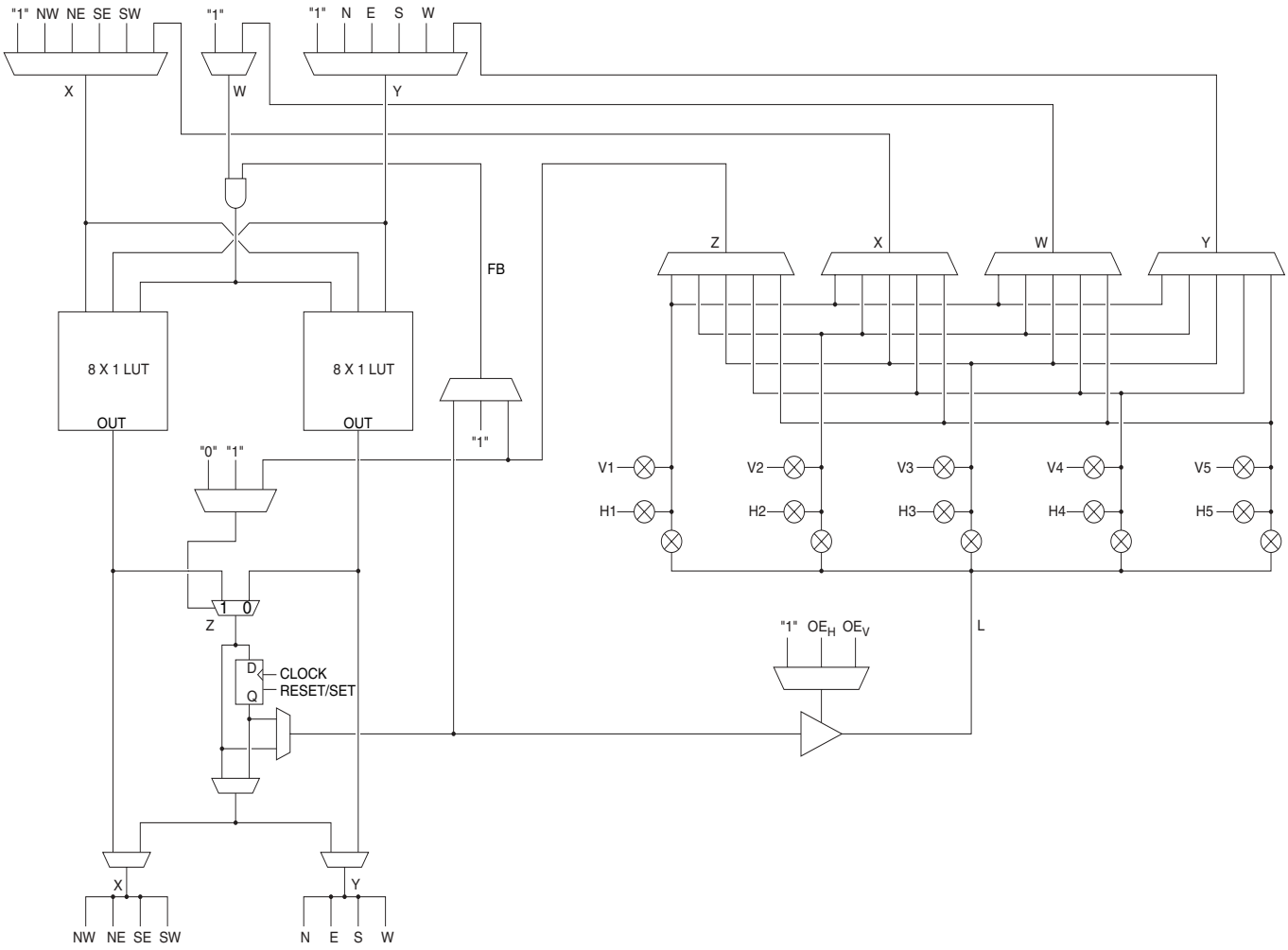


Figure 6. The Cell



X = Diagonal Direct Connect or Bus
Y = Orthogonal Direct Connect or Bus
W = Bus Connection
Z = Bus Connection
FB = Internal Feedback

System Control

Configuration Modes

The AT94K family has four configuration modes controlled by mode pins M0 and M2, see Table 10.

Table 10. Configuration Modes

M2	M0	Name
0	0	Mode 0 - Master Serial
0	1	Mode 1 - Slave Serial Cascade
1	0	Mode 2 - Reserved
1	1	Mode 3 - Reserved

Modes 2 and 3 are reserved and are used for factory test.

Modes 0 and 1 are pin-compatible with the appropriate AT40K counterpart. AVR I/O will be taken over by the configuration logic for the CHECK pin during both modes.

Refer to the “AT94K Series Configuration” application note for details on downloading bitstreams.

System Control Register – FPGA/AVR

The configuration control register in the FPSLIC consists of 8 bytes of data, which are loaded with the FPGA/Prog. Code at power-up from external nonvolatile memory. FPSLIC System Control Register values, see Table 11, can be set in the System Designer software. Recommended defaults are included in the software.

Table 11. FPSLIC System Control Register

Bit	Description
SCR0 - SCR1	Reserved
SCR2	0 = Enable Cascading 1 = Disable Cascading SCR2 controls the operation of the dual-function I/O CSOUT. When SCR2 is set, the CSOUT pin is not used by the configuration during downloads, set this bit for configurations where two or more devices are cascaded together. This applies for configuration to another FPSLIC device or to an FPGA.
SCR3	0 = Check Function Enabled 1 = Check Function Disabled SCR3 controls the operation of the CHECK pin and enables the Check Function. When SCR3 is set, the dual use AVR I/O/CHECK pin is not used by the configuration during downloads, and can be used as AVR I/O.
SCR4	0 = Memory Lockout Disabled 1 = Memory Lockout Enabled SCR4 is the Security Flag and controls the writing and checking of configuration memory during any subsequent configuration download. When SCR4 is set, any subsequent configuration download initiated by the user, whether a normal download or a CHECK function download, causes the INIT pin to immediately activate. CON is released, and no further configuration activity takes place. The download sequence during which SCR4 is set is NOT affected. The Control Register write is also prohibited, so bit SCR4 may only be cleared by a power-on reset or manual reset.
SCR5	Reserved

Conditional Branch Summary

Test	Boolean	Mnemonic	Complementary	Boolean	Mnemonic	Comment
Rd > Rr	Z•(N ⊕ V) = 0	BRLT	Rd ≤ Rr	Z+(N ⊕ V) = 1	BRGE	Signed
Rd ≥ Rr	(N ⊕ V) = 0	BRGE	Rd < Rr	(N ⊕ V) = 1	BRLT	Signed
Rd = Rr	Z = 1	BREQ	Rd ≠ Rr	Z = 0	BRNE	Signed
Rd ≤ Rr	Z+(N ⊕ V) = 1	BRGE	Rd > Rr	Z•(N ⊕ V) = 0	BRLT	Signed
Rd < Rr	(N ⊕ V) = 1	BRLT	Rd ≥ Rr	(N ⊕ V) = 0	BRGE	Signed
Rd > Rr	C + Z = 0	BRLO	Rd ≤ Rr	C + Z = 1	BRSH	Unsigned
Rd ≥ Rr	C = 0	BRSH/BRCC	Rd < Rr	C = 1	BRLO/BRCS	Unsigned
Rd = Rr	Z = 1	BREQ	Rd ≠ Rr	Z = 0	BRNE	Unsigned
Rd ≤ Rr	C + Z = 1	BRSH	Rd > Rr	C + Z = 0	BRLO	Unsigned
Rd < Rr	C = 1	BRLO/BRCS	Rd ≥ Rr	C = 0	BRSH/BRCC	Unsigned
Carry	C = 1	BRCS	No Carry	C = 0	BRCC	Simple
Negative	N = 1	BRMI	Positive	N = 0	BRPL	Simple
Overflow	V = 1	BRVS	No Overflow	V = 0	BRVC	Simple
Zero	Z = 1	BREQ	Not Zero	Z = 0	BRNE	Simple

Complete Instruction Set Summary

Instruction Set Summary

Mnemonics	Operands	Description	Operation	Flags	#Clock
Arithmetic and Logic Instructions					
ADD	Rd, Rr	Add without Carry	Rd ← Rd + Rr	Z,C,N,V,S,H	1
ADC	Rd, Rr	Add with Carry	Rd ← Rd + Rr + C	Z,C,N,V,S,H	1
ADIW	Rd, K	Add Immediate to Word	Rd+1:Rd ← Rd+1:Rd + K	Z,C,N,V,S	2
SUB	Rd, Rr	Subtract without Carry	Rd ← Rd - Rr	Z,C,N,V,S,H	1
SUBI	Rd, K	Subtract Immediate	Rd ← Rd - K	Z,C,N,V,S,H	1
SBC	Rd, Rr	Subtract with Carry	Rd ← Rd - Rr - C	Z,C,N,V,S,H	1
SBCI	Rd, K	Subtract Immediate with Carry	Rd ← Rd - K - C	Z,C,N,V,S,H	1
SBIW	Rd, K	Subtract Immediate from Word	Rd+1:Rd ← Rd+1:Rd - K	Z,C,N,V,S	2
AND	Rd, Rr	Logical AND	Rd ← Rd • Rr	Z,N,V,S	1
ANDI	Rd, K	Logical AND with Immediate	Rd ← Rd • K	Z,N,V,S	1
OR	Rd, Rr	Logical OR	Rd ← Rd v Rr	Z,N,V,S	1
ORI	Rd, K	Logical OR with Immediate	Rd ← Rd v K	Z,N,V,S	1
EOR	Rd, Rr	Exclusive OR	Rd ← Rd ⊕ Rr	Z,N,V,S	1
COM	Rd	One's Complement	Rd ← $FF - Rd	Z,C,N,V,S	1
NEG	Rd	Two's Complement	Rd ← $00 - Rd	Z,C,N,V,S,H	1
SBR	Rd, K	Set Bit(s) in Register	Rd ← Rd v K	Z,N,V,S	1

FPGA I/O Interrupt Control by AVR

This is an alternate memory space for the FPGA I/O Select addresses. If the FIADR bit in the FISCR register is set to logic 1, the four I/O addresses, FISUA - FISUD, are mapped to physical registers and provide memory space for FPGA interrupt masking and interrupt flag status. If the FIADR bit in the FISCR register is cleared to a logic 0, the I/O register addresses will be decoded into FPGA select lines.

All FPGA interrupt lines into the AVR are negative edge triggered. See page 58 for interrupt priority.

Interrupt Control Registers – FISUA..D

Bit	7	6	5	4	3	2	1	0	
$14 ($34)	**FIF3**	**FIF2**	**FIF1**	**FIF0**	**FINT3**	**FINT2**	**FINT1**	**FINT0**	**FISUA**
$15 ($35)	**FIF7**	**FIF6**	**FIF5**	**FIF4**	**FINT7**	**FINT6**	**FINT5**	**FINT4**	**FSUB**
$16 ($36)	**FIF11**	**FIF10**	**FIF9**	**FIF8**	**FINT11**	**FINT10**	**FINT9**	**FINT8**	**FISUC**
$17 ($37)	**FIF15**	**FIF14**	**FIF13**	**FIF12**	**FINT15**	**FINT14**	**FINT13**	**FINT12**	**FISUD**
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	

- **Bits 7..4 - FIF3 - 0: FPGA Interrupt Flags 3 - 0**

The 16 FPGA interrupt flag bits all work the same. Each is set (one) by a valid negative edge transition on its associated interrupt line from the FPGA. Valid transitions are defined as any change in state preceded by at least two cycles of the old state and succeeded by at least two cycles of the new state. Therefore, it is required that interrupt lines transition from 1 to 0 at least two cycles after the line is stable High; the line must then remain stable Low for at least two cycles following the transition. Each bit is cleared by the hardware when executing the corresponding interrupt handling vector. Alternatively, each bit will be cleared by writing a logic 1 to it. When the I-bit in the Status Register, the corresponding FPGA interrupt mask bit and the given FPGA interrupt flag bit are set (one), the associated interrupt is executed.

- **Bits 7..4 - FIF7 - 4: FPGA Interrupt Flags 7 - 4**

See *Bits 7..4 - FIF3 - 0: FPGA Interrupt Flags 3 - 0*.

- **Bits 7..4 - FIF11 - 8: FPGA Interrupt Flags 11 - 8**

See *Bits 7..4 - FIF3 - 0: FPGA Interrupt Flags 3 - 0*. Not available on the AT94K05.

- **Bits 7..4 - FIF15 - 12: FPGA Interrupt Flags 15 - 12**

See *Bits 7..4 - FIF3 - 0: FPGA Interrupt Flags 3 - 0*. Not available on the AT94K05.

- **Bits 3..0 - FINT3 - 0: FPGA Interrupt Masks 3 - 0**[1]

The 16 FPGA interrupt mask bits all work the same. When a mask bit is set (one) and the I-bit in the Status Register is set (one), the given FPGA interrupt is enabled. The corresponding interrupt handling vector is executed when the given FPGA interrupt flag bit is set (one) by a negative edge transition on the associated interrupt line from the FPGA.

Note: 1. FPGA interrupts 3 - 0 will cause a wake-up from the AVR Sleep modes. These interrupts are treated as low-level triggered in the Power-down and Power-save modes, see “Sleep Modes” on page 66.

- **Bits 3..0 - FINT7 - 4: FPGA Interrupt Masks 7 - 4**

See *Bits 3..0 - FINT3 - 0: FPGA Interrupt Masks 3 - 0.*

- **Bits 3..0 - FINT11 - 8: FPGA Interrupt Masks 11 - 8**

See *Bits 3..0 - FINT3 - 0: FPGA Interrupt Masks 3 - 0.* Not available on the AT94K05.

- **Bits 3..0 - FINT15 - 12: FPGA Interrupt Masks 15 -12**

See *Bits 3..0 - FINT3 - 0: FPGA Interrupt Masks 3 - 0.* Not available on the AT94K05.

Reset and Interrupt Handling

The embedded AVR and FPGA core provide 35 different interrupt sources. These interrupts and the separate reset vector each have a separate program vector in the program memory space. All interrupts are assigned individual enable bits (masks) which must be set (one) together with the I-bit in the status register in order to enable the interrupt.

The lowest addresses in the program memory space must be defined as the Reset and Interrupt vectors. The complete list of vectors is shown in Table 15. The list also determines the priority levels of the different interrupts. The lower the address the higher the priority level. RESET has the highest priority, and next is FPGA_INT0 – the FPGA Interrupt Request 0 etc.

Table 15. Reset and Interrupt Vectors

Vector No. (hex)	Program Address	Source	Interrupt Definition
01	$0000	RESET	Reset Handle: Program Execution Starts Here
02	$0002	FPGA_INT0	FPGA Interrupt0 Handle
03	$0004	EXT_INT0	External Interrupt0 Handle
04	$0006	FPGA_INT1	FPGA Interrupt1 Handle
05	$0008	EXT_INT1	External Interrupt1 Handle
06	$000A	FPGA_INT2	FPGA Interrupt2 Handle
07	$000C	EXT_INT2	External Interrupt2 Handle
08	$000E	FPGA_INT3	FPGA Interrupt3 Handle
09	$0010	EXT_INT3	External Interrupt3 Handle
0A	$0012	TIM2_COMP	Timer/Counter2 Compare Match Interrupt Handle
0B	$0014	TIM2_OVF	Timer/Counter2 Overflow Interrupt Handle
0C	$0016	TIM1_CAPT	Timer/Counter1 Capture Event Interrupt Handle
0D	$0018	TIM1_COMPA	Timer/Counter1 Compare Match A Interrupt Handle
0E	$001A	TIM1_COMPB	Timer/Counter1 Compare Match B Interrupt Handle
0F	$001C	TIM1_OVF	Timer/Counter1 Overflow Interrupt Handle
10	$001E	TIM0_COMP	Timer/Counter0 Compare Match Interrupt Handle
11	$0020	TIM0_OVF	Timer/Counter0 Overflow Interrupt Handle
12	$0022	FPGA_INT4	FPGA Interrupt4 Handle
13	$0024	FPGA_INT5	FPGA Interrupt5 Handle
14	$0026	FPGA_INT6	FPGA Interrupt6 Handle
15	$0028	FPGA_INT7	FPGA Interrupt7 Handle
16	$002A	UART0_RXC	UART0 Receive Complete Interrupt Handle

- **Bit 2 - OCIE2: Timer/Counter2 Output Compare Interrupt Enable**

When the OCIE2 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter2 Compare Match interrupt is enabled. The corresponding interrupt is executed if a Compare match in Timer/Counter2 occurs, i.e., when the OCF2 bit is set in the Timer/Counter interrupt flag register – TIFR.

- **Bit 1 - TOIE0: Timer/Counter0 Overflow Interrupt Enable**

When the TOIE0 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter0 Overflow interrupt is enabled. The corresponding interrupt is executed if an overflow in Timer/Counter0 occurs, i.e., when the TOV0 bit is set in the Timer/Counter Interrupt Flag Register – TIFR.

- **Bit 0 - OCIE0: Timer/Counter0 Output Compare Interrupt Enable**

When the OCIE0 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter0 Compare Match interrupt is enabled. The corresponding interrupt is executed if a Compare match in Timer/Counter0 occurs, i.e., when the OCF0 bit is set in the Timer/Counter Interrupt Flag Register – TIFR.

Timer/Counter Interrupt Flag Register – TIFR

Bit	7	6	5	4	3	2	1	0	
$38 ($58)	**TOV1**	**OCF1A**	**OCF1B**	**TOV2**	**ICF1**	**OCF2**	**TOV0**	**OCF0**	**TIFR**
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	

- **Bit 7 - TOV1: Timer/Counter1 Overflow Flag**

The TOV1 is set (one) when an overflow occurs in Timer/Counter1. TOV1 is cleared by the hardware when executing the corresponding interrupt handling vector. Alternatively, TOV1 is cleared by writing a logic 1 to the flag. When the I-bit in SREG, and TOIE1 (Timer/Counter1 Overflow Interrupt Enable), and TOV1 are set (one), the Timer/Counter1 Overflow Interrupt is executed. In PWM mode, this bit is set when Timer/Counter1 advances from $0000.

- **Bit 6 - OCF1A: Output Compare Flag 1A**

The OCF1A bit is set (one) when compare match occurs between the Timer/Counter1 and the data in OCR1A – Output Compare Register 1A. OCF1A is cleared by the hardware when executing the corresponding interrupt handling vector. Alternatively, OCF1A is cleared by writing a logic 1 to the flag. When the I-bit in SREG, and OCIE1A (Timer/Counter1 Compare Interrupt Enable), and the OCF1A are set (one), the Timer/Counter1 Compare A match Interrupt is executed.

- **Bit 5 - OCF1B: Output Compare Flag 1B**

The OCF1B bit is set (one) when compare match occurs between the Timer/Counter1 and the data in OCR1B – Output Compare Register 1B. OCF1B is cleared by the hardware when executing the corresponding interrupt handling vector. Alternatively, OCF1B is cleared by writing a logic 1 to the flag. When the I-bit in SREG, and OCIE1B (Timer/Counter1 Compare match Interrupt Enable), and the OCF1B are set (one), the Timer/Counter1 Compare B match Interrupt is executed.

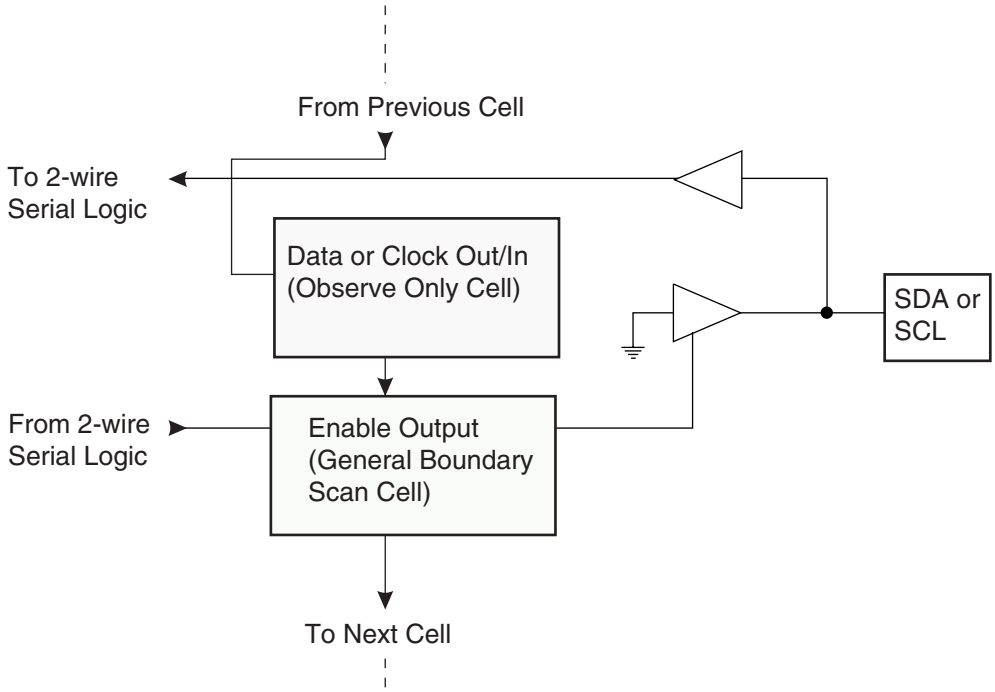
- **Bit 4 - TOV2: Timer/Counter2 Overflow Flag**

The TOV2 bit is set (one) when an overflow occurs in Timer/Counter2. TOV2 is cleared by the hardware when executing the corresponding interrupt handling vector. Alternatively, TOV2 is cleared by writing a logic 1 to the flag. When the I-bit in SREG, and TOIE2 (Timer/Counter1 Overflow Interrupt Enable), and TOV2 are set (one), the Timer/Counter2 Overflow Interrupt is executed. In PWM mode, this bit is set when Timer/Counter2 advances from $00.

Scanning 2-wire Serial

The SCL and SDA pins are open drain, bi-directional and enabled separately. The “Enable Output” bits (active High) in the scan chain are supported by general boundary-scan cells. Enabling the output will drive the pin Low from a tri-state. External pull-ups on the 2-wire bus are required to pull the pins High if the output is disabled. The “Data Out/In” and “Clock Out/In” bits in the scan chain are observe-only cells. Figure 46 shows how each pin is connected in the scan chain.

Figure 46. Boundary-scan Cells for 2-wire Serial



Scanning the Clock Pins

Figure 47 shows how each oscillator with external connection is supported in the scan chain. The Enable signal is supported with a general boundary-scan cell, while the oscillator/clock output is attached to an observe-only cell. In addition to the main clock, the timer oscillator is scanned in the same way. The output from the internal RC-Oscillator is not scanned, as this oscillator does not have external connections.

Figure 47. Boundary-scan Cells for Oscillators and Clock Options

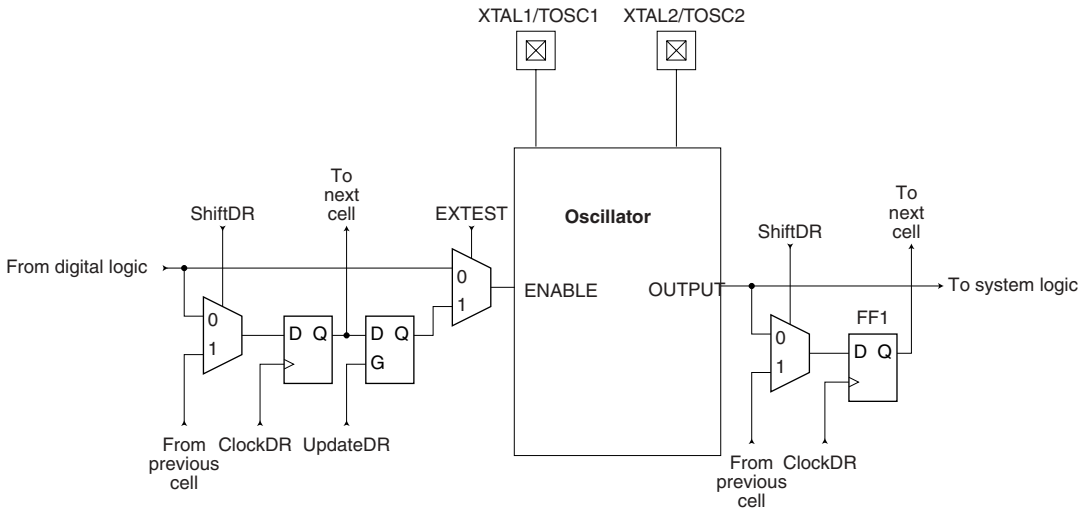


Table 20. AVR I/O Boundary Scan – JTAG Instructions $0/$2

I/O Ports	Description	Bit	
XTAL	Clock In - XTAL1	8[1]	
	Enable Clock - XTAL 1	7	
TOSC	Clock In - TOSC 1	6[1]	
	Enable Clock - TOSC 1	5	
2-wire Serial	Data Out/In - SDA	4[1]	
	Enable Output - SDA	3	
	Clock Out/In - SCL	2[1]	
	Enable Output - SCL	1	
[2]	AVR Reset	0[1]	-> TDO

Notes: 1. Observe-only scan cell.
2. AVR Reset is High (one) if AVRResetn activated (Low) and enabled or the device is in general reset (Resetn or power-on) or configuration download.

Table 21. Bit EXTEST and SAMPLE_PRELOAD

Bit Type	EXTEST	SAMPLE_PRELOAD
Data Out/In - PXn	**Defines value driven if enabled**. Capture-DR grabs signal on pad.	Capture-DR grabs signal from pad if output disabled, or from the AVR if the output drive is enabled.
Enable Output - PXn	**1 = output drive enabled**. Capture-DR grabs output enable scan latch.	Capture-DR grabs output enable from the AVR.
Pull-up - PXn	**1 = pull-up disabled**. Capture-DR grabs pull-up control from the AVR.	Capture-DR grabs pull-up control from the AVR.
Input with Pull-up - INTPn	**Observe only**. Capture-DR grabs signal from pad.	Capture-DR grabs signal from pad.
Data Out - TXn	**Defines value driven if enabled**. Capture-DR grabs signal on pad.	Capture-DR always grabs “0” since Tx input is NC and tied to ground internally.
Enable Output - TXn	**1 = output drive enabled**. Capture-DR grabs output enable scan latch.	Capture-DR grabs output enable from the AVR.
Pull-up - TXn	**1 = pull-up disabled**. Capture-DR grabs pull-up control from the AVR.	Capture-DR grabs pull-up control from the AVR.
Input with Pull-up - RXn	**Observe only**. Capture-DR grabs signal from pad.	Capture-DR grabs signal from pad.
Clock In - XTAL1	**Observe only**. Capture-DR grabs signal from pad.	Capture-DR grabs signal from pad if clock is enabled, “1” if disabled.
Enable Clock - XTAL 1	**1 = clock disabled**. Capture-DR grabs clock enable from the AVR.	Capture-DR grabs enable from the AVR.

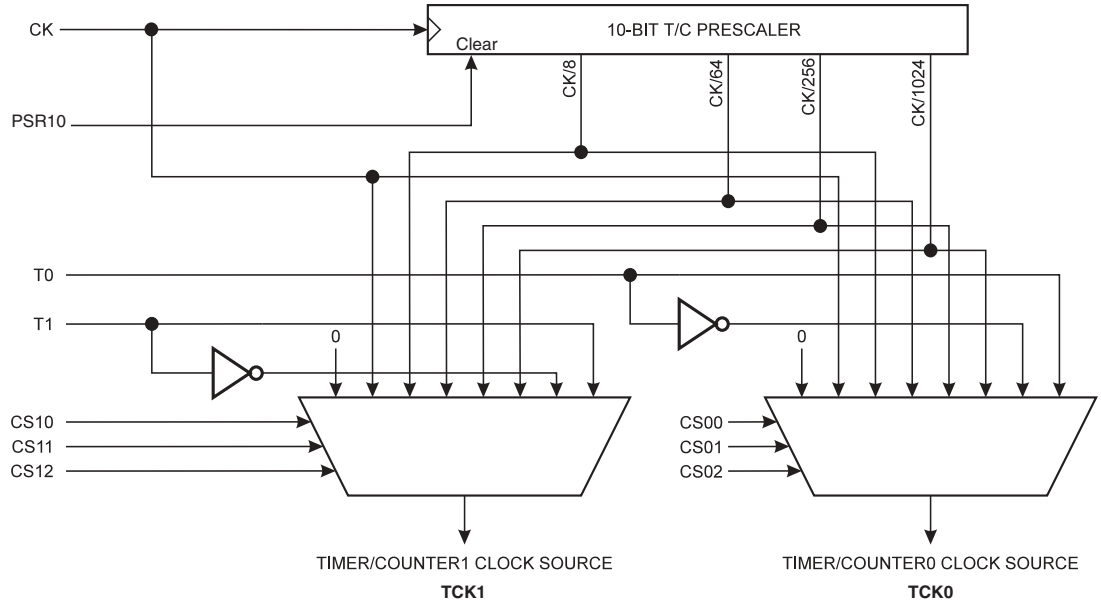
Timer/Counters

The FPSLIC provides three general-purpose Timer/Counters: two 8-bit T/Cs and one 16-bit T/C. Timer/Counter2 can optionally be asynchronously clocked from an external oscillator. This oscillator is optimized for use with a 32.768 kHz watch crystal, enabling use of Timer/Counter2 as a Real-time Clock (RTC). Timer/Counters 0 and 1 have individual prescaling selection from the same 10-bit prescaling timer. Timer/Counter2 has its own prescaler. Both these prescalers can be reset by setting the corresponding control bits in the Special Functions I/O Register (SFIOR). See “Special Function I/O Register – SFIOR” on page 86 for a detailed description. These Timer/Counters can either be used as a timer with an internal clock time-base or as a counter with an external pin connection which triggers the counting.

Timer/Counter Prescalers

For Timer/Counters 0 and 1, see Figure 48, the four prescaled selections are: CK/8, CK/64, CK/256 and CK/1024, where CK is the oscillator clock. For the two Timer/Counters 0 and 1, CK, external source, and stop, can also be selected as clock sources. Setting the PSR10 bit in SFIOR resets the prescaler. This allows the user to operate with a predictable prescaler. Note that Timer/Counter1 and Timer/Counter0 share the same prescaler and a prescaler reset will affect both Timer/Counters.

Figure 48. Prescaler for Timer/Counter0 and 1



The clock source for Timer/Counter2 prescaler, see Figure 49, is named PCK2. PCK2 is by default connected to the main system clock CK. By setting the AS2 bit in ASSR, Timer/Counter2 is asynchronously clocked from the TOSC1 pin. This enables use of Timer/Counter2 as a Real-time Clock (RTC). When AS2 is set, pins TOSC1 and TOSC2 are disconnected from Port D. A crystal can then be connected between the TOSC1 and TOSC2 pins to serve as an independent clock source for Timer/Counter2. The oscillator is optimized for use with a 32.768 kHz crystal. Alternatively, an external clock signal can be applied to TOSC1. The frequency of this clock must be lower than one fourth of the CPU clock and not higher than 1 MHz. Setting the PSR2 bit in SFIOR resets the prescaler. This allows the user to operate with a predictable prescaler.

muls16x16_32

Description

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

Usage

R19:R18:R17:R16 = R23:R22 • R21:R20

Statistics

Cycles: 19 + ret

Words: 15 + ret

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

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

```
muls16x16_32:
   clr   r2
   muls  r23, r21             ; (signed)ah * (signed)bh
   movw  r19:r18, r1:r0
   mul   r22, r20             ; al * bl
   movw  r17:r16, r1:r0
   mulsu r23, r20             ; (signed)ah * bl
   sbc   r19, r2              ; Sign extend
   add   r17, r0
   adc   r18, r1
   adc   r19, r2
   mulsu r21, r22             ; (signed)bh * al
   sbc   r19, r2              ; Sign Extend
   add   r17, r0
   adc   r18, r1
   adc   r19, r2
   ret
```

mac16x16_32

Description

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

Usage

R19:R18:R17:R16 += R23:R22 • R21:R20

Statistics

Cycles: 23 + ret

Words: 19 + ret

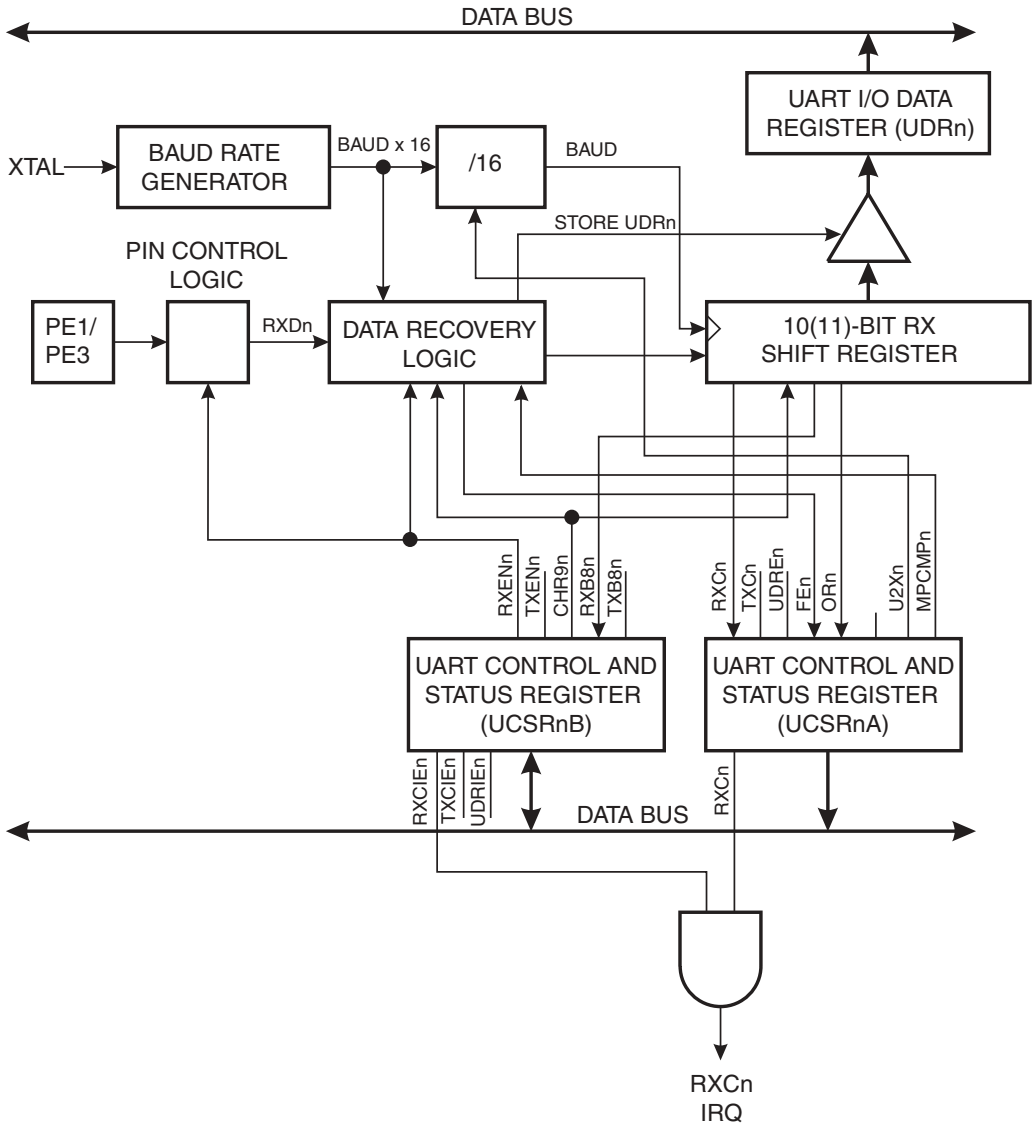
Register usage: R0 to R2 and R16 to R23 (11 registers)

```
mac16x16_32:                  ; Register Usage Optimized
   clr   r2

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

   mul   r22, r20             ; al * bl
   add   r16, r0
   adc   r17, r1
   adc   r18, r2
   adc   r19, r2
```

Data Reception

Figure 65 shows a block diagram of the UART Receiver.

Figure 65. UART Receiver[1]



Note: 1. n = 0, 1

The receiver front-end logic samples the signal on the RXDn pin at a frequency 16 times the baud-rate. While the line is idle, one single sample of logic 0 will be interpreted as the falling edge of a start bit, and the start bit detection sequence is initiated. Let sample 1 denote the first zero-sample. Following the 1-to-0 transition, the receiver samples the RXDn pin at samples 8, 9 and 10. If two or more of these three samples are found to be logic 1s, the start bit is rejected as a noise spike and the receiver starts looking for the next 1-to-0 transition.

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.

- **Bit 3 - OR0/OR1: OverRun**

This bit is set if an Overrun condition is detected, i.e., when a character already present in the UDRn register is not read before the next character has been shifted into the Receiver Shift register. The ORn bit is buffered, which means that it will be set once the valid data still in UDRn is read.

The ORn bit is cleared (zero) when data is received and transferred to UDRn.

- **Bit 2 - Res: Reserved Bit**

This bit is reserved in the AT94K and will always read as zero.

- **Bits 1 - U2X0/U2X1: Double the UART Transmission Speed**

When this bit is set (one) the UART speed will be doubled. This means that a bit will be transmitted/received in eight CPU clock periods instead of 16 CPU clock periods. For a detailed description, see “Double Speed Transmission” on page 128”.

- **Bit 0 - MPCM0/MPCM1: Multi-processor Communication Mode**

This bit is used to enter Multi-processor Communication Mode. The bit is set when the Slave MCU waits for an address byte to be received. When the MCU has been addressed, the MCU switches off the MPCMn bit, and starts data reception.

For a detailed description, see “Multi-processor Communication Mode” on page 123.

UART0 Control and Status Registers – UCSR0B

Bit	7	6	5	4	3	2	1	0	
$0A ($2A)	**RXCIE0**	**TXCIE0**	**UDRIE0**	**RXEN0**	**TXEN0**	**CHR90**	**RXB80**	**TXB80**	**UCSR0B**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
Initial Value	0	0	0	0	0	0	1	0	

UART1 Control and Status Registers – UCSR1B

Bit	7	6	5	4	3	2	1	0	
$01 ($21)	**RXCIE1**	**TXCIE1**	**UDRIE1**	**RXEN1**	**TXEN1**	**CHR91**	**RXB81**	**TXB81**	**UCSR1B**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
Initial Value	0	0	0	0	0	0	1	0	

- **Bit 7 - RXCIE0/RXCIE1: RX Complete Interrupt Enable**

When this bit is set (one), a setting of the RXCn bit in UCSRnA will cause the Receive Complete interrupt routine to be executed provided that global interrupts are enabled.

- **Bit 6 - TXCIE0/TXCIE1: TX Complete Interrupt Enable**

When this bit is set (one), a setting of the TXCn bit in UCSRnA will cause the Transmit Complete interrupt routine to be executed provided that global interrupts are enabled.

- **Bit 5 - UDRIE0/UDREI1: UART Data Register Empty Interrupt Enable**

When this bit is set (one), a setting of the UDREn bit in UCSRnA will cause the UART Data Register Empty interrupt routine to be executed provided that global interrupts are enabled.

- **Bit 4 - RXEN0/RXEN1: Receiver Enable**

This bit enables the UART receiver when set (one). When the receiver is disabled, the TXCn, ORn and FEn status flags cannot become set. If these flags are set, turning off RXENn does not cause them to be cleared.

2-wire Serial Modes

The 2-wire Serial Interface can operate in four different modes:

- Master Transmitter
- Master Receiver
- Slave Receiver
- Slave Transmitter

Data transfer in each mode of operation is shown in Figure 71 to Figure 74. These figures contain the following abbreviations:

S: START condition

R: Read bit (High level at SDA)

W: Write bit (Low level at SDA)

A: Acknowledge bit (Low level at SDA)

$\overline{A}$: Not acknowledge bit (High level at SDA)

Data: 8-bit data byte

P: STOP condition

In Figure 71 to Figure 74, circles are used to indicate that the 2-wire Serial Interrupt flag is set. The numbers in the circles show the status code held in TWSR. At these points, an interrupt routine must be executed to continue or complete the 2-wire Serial Transfer. The 2-wire Serial Transfer is suspended until the 2-wire Serial Interrupt flag is cleared by software.

The 2-wire Serial Interrupt flag is not automatically cleared by the hardware when executing the interrupt routine. Also note that the 2-wire Serial Interface starts execution as soon as this bit is cleared, so that all access to TWAR, TWDR and TWSR must have been completed before clearing this flag.

When the 2-wire Serial Interrupt flag is set, the status code in TWSR is used to determine the appropriate software action. For each status code, the required software action and details of the following serial transfer are given in Table 41 to Table 45.

Master Transmitter Mode

In the Master Transmitter mode, a number of data bytes are transmitter to a Slave Receiver, see Figure 71. Before the Master Transmitter mode can be entered, the TWCR must be initialized as shown in Table 38.

Table 38. TWCR: Master Transmitter Mode Initialization

TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
value	0	X	0	0	0	1	0	X

TWEN must be set to enable the 2-wire Serial Interface, TWSTA and TWSTO must be cleared.

The Master Transmitter mode may now be entered by setting the TWSTA bit. The 2-wire Serial Logic will now test the 2-wire Serial Bus and generate a START condition as soon as the bus becomes free. When a START condition is transmitted, the 2-wire Serial Interrupt flag (TWINT) is set by the hardware, and the status code in TWSR will be $08. TWDR must then be loaded with the Slave address and the data direction bit (SLA+W). The TWINT flag must then be cleared by software before the 2-wire Serial Transfer can continue. The TWINT flag is cleared by writing a logic 1 to the flag.

When the Slave address and the direction bit have been transmitted and an acknowledgment bit has been received, TWINT is set again and a number of status codes in TWSR are possible. Status codes $18, $20, or $38 apply to Master mode, and status codes $68, $78, or $B0 apply to Slave mode. The appropriate action to be taken for each of these status codes is

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/From TWDR	To TWCR				
			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

External Clock Drive Waveforms

Figure 85. External Clock Drive Waveforms

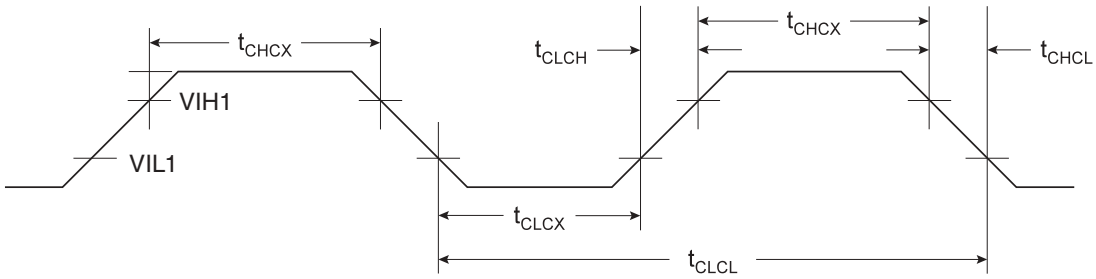


Table 53. External Clock Drive, V_{CC} = 3.0V to 3.6V

Symbol	Parameter	Minimum	Maximum	Units
$1/t_{CLCL}$	Oscillator Frequency	0	25	MHz
t_{CLCL}	Clock Period	40	–	ns
t_{CHCX}	High Time	15	–	ns
t_{CLCX}	Low Time	15	–	ns
t_{CLCH}	Rise Time	–	1.6	µs
t_{CHCL}	Fall Time	–	1.6	µs

AC Timing Characteristics – 3.3V Operation

Delays are based on fixed loads and are described in the notes.
Maximum times based on worst case: V_{CC} = 3.0V, temperature = 70°C
Minimum times based on best case: V_{CC} = 3.6V, temperature = 0°C

Cell Function	Parameter	Path	-25	Units	Notes
Async RAM					
Write	t_{WECYC} (Minimum)	cycle time	12.0	ns	–
Write	t_{WEL} (Minimum)	we	5.0	ns	Pulse Width Low
Write	t_{WEH} (Minimum)	we	5.0	ns	Pulse Width High
Write	t_{setup} (Minimum)	wr addr setup-> we	5.3	ns	
Write	t_{hold} (Minimum)	wr addr hold -> we	0.0	ns	–
Write	t_{setup} (Minimum)	din setup -> we	5.0	ns	
Write	t_{hold} (Minimum)	din hold -> we	0.0	ns	–
Write	t_{hold} (Minimum)	oe hold -> we	0.0	ns	
Write/Read	t_{PD} (Maximum)	din -> dout	8.7	ns	rd addr = wr addr
Read	t_{PD} (Maximum)	rd addr -> dout	6.3	ns	
Read	t_{PZX} (Maximum)	oe -> dout	2.9	ns	–
Read	t_{PXZ} (Maximum)	oe -> dout	3.5	ns	
Sync RAM					
Write	t_{CYC} (Minimum)	cycle time	12.0	ns	
Write	t_{CLKL} (Minimum)	clk	5.0	ns	–
Write	t_{CLKH} (Minimum)	clk	5.0	ns	Pulse Width High
Write	t_{setup} (Minimum)	we setup-> clk	3.2	ns	
Write	t_{hold} (Minimum)	we hold -> clk	0.0	ns	–
Write	t_{setup} (Minimum)	wr addr setup-> clk	5.0	ns	
Write	t_{hold} (Minimum)	wr addr hold -> clk	0.0	ns	–
Write	t_{setup} (Minimum)	wr data setup-> clk	3.9	ns	
Write	t_{hold} (Minimum)	wr data hold -> clk	0.0	ns	–
Write/Read	t_{PD} (Maximum)	din -> dout	8.7	ns	rd addr = wr addr
Write/Read	t_{PD} (Maximum)	clk -> dout	5.8	ns	rd addr = wr addr
Read	t_{PD} (Maximum)	rd addr -> dout	6.3	ns	
Read	t_{PZX} (Maximum)	oe -> dout	2.9	ns	–
Read	t_{PXZ} (Maximum)	oe -> dout	3.5	ns	

CMOS buffer delays are measured from a V_{IH} of 1/2 V_{CC} at the pad to the internal V_{IH} at A. The input buffer load is constant. Buffer delay is to a pad voltage of 1.5V with one output switching. Parameter based on characterization and simulation; not tested in production. An FPGA power calculation is available in Atmel's System Designer software (see also page 160).

Packaging and Pin List Information

FPSLIC devices should be laid out to support a split power supply for both AL and AX families. Please refer to the “Designing in Split Power Supply Support for AT94KAL and AT94SAL Devices” application note, available on the Atmel web site.

Table 54. Part and Package Combinations Available

Part #	Package	AT94K05	AT94K10	AT94K40
PLCC 84	AJ	46	46	
TQ 100	AQ	58	58	
LQ144	BQ	82	84	84
PQ 208	DQ	96	116	120

Table 55. AT94K JTAG ICE Pin List

Pin	AT94K05 96 FPGA I/O	AT94K10 192 FPGA I/O	AT94K40 384 FPGA I/O
TDI	IO34	IO50	IO98
TDO	IO38	IO54	IO102
TMS	IO43	IO63	IO123
TCK	IO44	IO64	IO124

Table 56. AT94K Pin List

AT94K05 96 FPGA I/O	AT94K10 192 FPGA I/O	AT94K40 384 FPGA I/O	Packages			
			PC84	TQ100	PQ144	PQ208
West Side						
GND	GND	GND	12	1	1	2
I/O1, GCK1 (A16)	I/O1, GCK1 (A16)	I/O1, GCK1 (A16)	13	2	2	4
I/O2 (A17)	I/O2 (A17)	I/O2 (A17)	14	3	3	5
I/O3	I/O3	I/O3			4	6
I/O4	I/O4	I/O4			5	7
I/O5 (A18)	I/O5 (A18)	I/O5 (A18)	15	4	6	8
I/O6 (A19)	I/O6 (A19)	I/O6 (A19)	16	5	7	9
		GND				
		I/O7				
		I/O8				
		I/O9				

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/O140				
		I/O141				
		I/O142				
I/O47 (TD7)	I/O71 (TD7)	I/O143 (TD7)	40	35	52	76
I/O48 (InitErr)	I/O72 (InitErr)	I/O144 (InitErr)	41	36	53	77
VDD(2)	VDD(2)	VDD(2)	42	37	54	78
GND	GND	GND	43	38	55	79
I/O49 (TD6)	I/O73 (TD6)	I/O145 (TD6)	44	39	56	80
I/O50 (TD5)	I/O74 (TD5)	I/O146 (TD5)	45	40	57	81
		I/O147				
		I/O148				
		I/O149				
		I/O150				
		GND				
I/O51	I/O75	I/O151		41	58	82
I/O52	I/O76	I/O152		42	59	83
	I/O77	I/O153				84
	I/O78	I/O154				85
		I/O155				
		I/O156				
		VCC(1)				
		GND				
		I/O157				
		I/O158				
		I/O159				
		I/O160				
		I/O161				
		I/O162				
		GND				
	I/O79	I/O163				
	I/O80	I/O164				
	VCC(1)	VCC(1)				
I/O53 (TD4)	I/O81 (TD4)	I/O165 (TD4)	46	43	60	86
I/O54 (TD3)	I/O82 (TD3)	I/O166 (TD3)	47	44	61	87

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.