Welcome to E-XFL.COM

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

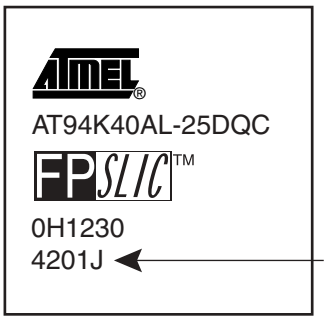
Embedded - FPGAs (Field Programmable Gate Arrays) with Microcontrollers represent a cutting-edge category of electronic components that combine the flexibility of FPGA technology with the processing power of integrated microcontrollers. This hybrid approach offers a versatile solution for designing and 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	20K-32K
FPGA SRAM	4kb
EEPROM Size	-
Data SRAM Bytes	4K ~ 16K
FPGA Core Cells	576
FPGA Gates	10K
FPGA Registers	846
Voltage - Supply	3V ~ 3.6V
Mounting Type	Surface Mount
Operating Temperature	0°C ~ 70°C
Package / Case	208-BFQFP
Supplier Device Package	208-PQFP (28x28)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/at94k10al-25dqc

Figure 1. FPSLIC Device Date Code with JTAG ICE Support



The AT94K series architecture is shown in Figure 2.

Figure 2. AT94K Series Architecture

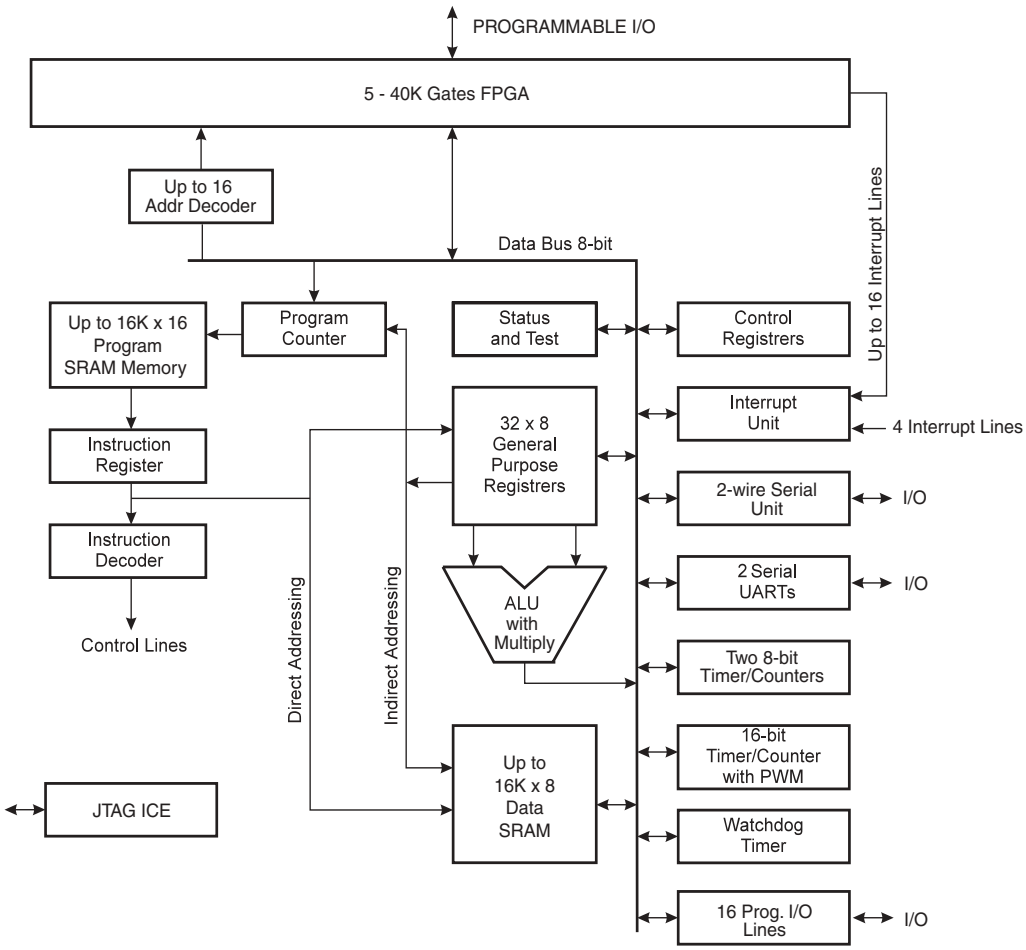
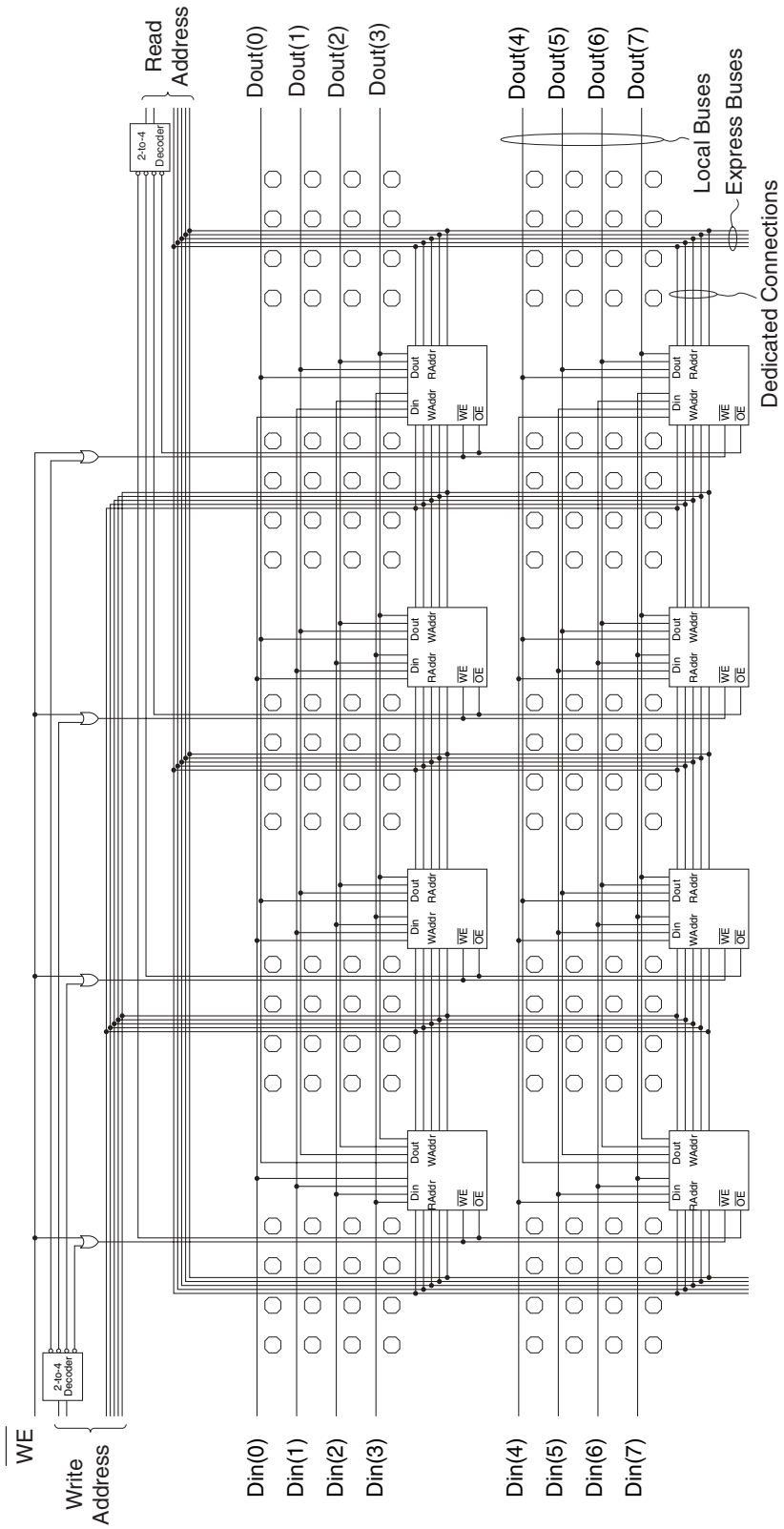


Figure 10. FreeRAM Example: 128 x 8 Dual-ported RAM (Asynchronous)[1]

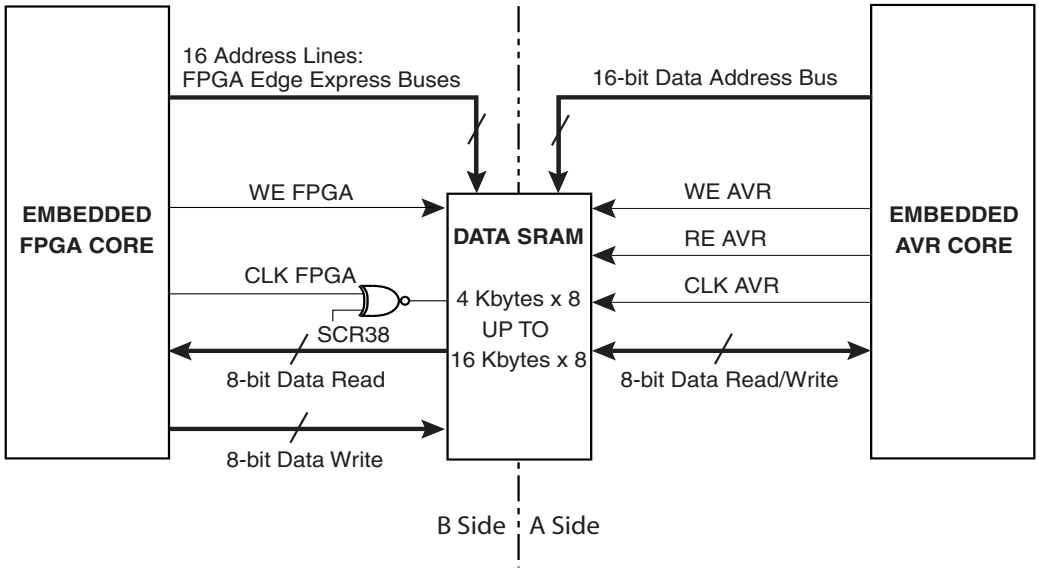


Note: 1. These layouts can be generated automatically using the Macro Generators.

Data SRAM Access by FPGA – FPGAFrame Mode

The FPGA user logic has access to the data SRAM directly through the FPGA side of the dual-port memory, see Figure 20. A single bit in the configuration control register (SCR63 – see "System Control Register – FPGA/AVR" on page 30) enables this interface. The interface is disabled during configuration downloads. Express buses on the East edge of the array are used to interface the memory. Full read and write access is available. To allow easy implementation, the interface itself is dedicated in routing resources, and is controlled in the System Designer software suite using the AVR FPGA interface dialog.

Figure 20. Internal SRAM Access – Normal Use



Once the SCR63 bit is set there is no additional read enable from the FPGA side. This means that the read is always enabled. You can also perform a read or write from the AVR at the same time as an FPGA read or write. If there is a possibility of a write address being accessed by both devices at the same time, the designer should add arbitration to the FPGA Logic to control who has priority. In most cases the AVR would be used to restrict access by the FPGA using the FMXOR bit, see "Software Control Register – SFTCR" on page 51. You can read from the same location from both sides simultaneously.

SCR bit 38 controls the polarity of the clock to the SRAM from the AT40K FPGA.

SRAM Access by FPGA/AVR

This option is used to allow for code (Program Memory) changes.

Accessing and Modifying the Program Memory from the AVR

The FPSLIC SRAM is up to 36 x 8 Kbytes of dual port, see Figure 19):

- The A side (port) is accessed by the AVR.
- The B side (port) is accessed by the FPGA/Configuration Logic.
- The B side (port) can be accessed by the AVR with ST and LD instructions in DBG mode for code self-modify.

Structurally, the [(n • 2) Kbytes 8] memory is built from (n)2 Kbytes 8 blocks, numbered SRAM0 through SRAM(n).

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

Instruction Set Summary (Continued)

Mnemonics	Operands	Description	Operation	Flags	#Clock
BRSH	k	Branch if Same or Higher	if (C = 0) then PC ← PC + k + 1	None	1 / 2
BRLO	k	Branch if Lower	if (C = 1) then PC ← PC + k + 1	None	1 / 2
BRMI	k	Branch if Minus	if (N = 1) then PC ← PC + k + 1	None	1 / 2
BRPL	k	Branch if Plus	if (N = 0) then PC ← PC + k + 1	None	1 / 2
BRGE	k	Branch if Greater or Equal, Signed	if (N ⊕ V= 0) then PC ← PC + k + 1	None	1 / 2
BRLT	k	Branch if Less Than, Signed	if (N ⊕ V= 1) then PC ← PC + k + 1	None	1 / 2
BRHS	k	Branch if Half-carry Flag Set	if (H = 1) then PC ← PC + k + 1	None	1 / 2
BRHC	k	Branch if Half-carry Flag Cleared	if (H = 0) then PC ← PC + k + 1	None	1 / 2
BRTS	k	Branch if T Flag Set	if (T = 1) then PC ← PC + k + 1	None	1 / 2
BRTC	k	Branch if T Flag Cleared	if (T = 0) then PC ← PC + k + 1	None	1 / 2
BRVS	k	Branch if Overflow Flag is Set	if (V = 1) then PC ← PC + k + 1	None	1 / 2
BRVC	k	Branch if Overflow Flag is Cleared	if (V = 0) then PC ← PC + k + 1	None	1 / 2
BRIE	k	Branch if Interrupt Enabled	if (I = 1) then PC ← PC + k + 1	None	1 / 2
BRID	k	Branch if Interrupt Disabled	if (I = 0) then PC ← PC + k + 1	None	1 / 2
Data Transfer Instructions					
MOV	Rd, Rr	Copy Register	Rd ← Rr	None	1
MOVW	Rd, Rr	Copy Register Pair	Rd+1:Rd ← Rr+1:Rr	None	1
LDI	Rd, K	Load Immediate	Rd ← K	None	1
LDS	Rd, k	Load Direct from Data Space	Rd ← (k)	None	2
LD	Rd, X	Load Indirect	Rd ← (X)	None	2
LD	Rd, X+	Load Indirect and Post-Increment	Rd ← (X), X ← X + 1	None	2
LD	Rd, -X	Load Indirect and Pre-Decrement	X ← X - 1, Rd ← (X)	None	2
LD	Rd, Y	Load Indirect	Rd ← (Y)	None	2
LD	Rd, Y+	Load Indirect and Post-Increment	Rd ← (Y), Y ← Y + 1	None	2
LD	Rd, -Y	Load Indirect and Pre-Decrement	Y ← Y - 1, Rd ← (Y)	None	2
LDD	Rd, Y+q	Load Indirect with Displacement	Rd ← (Y + q)	None	2
LD	Rd, Z	Load Indirect	Rd ← (Z)	None	2
LD	Rd, Z+	Load Indirect and Post-Increment	Rd ← (Z), Z ← Z+1	None	2
LD	Rd, -Z	Load Indirect and Pre-Decrement	Z ← Z - 1, Rd ← (Z)	None	2
LDD	Rd, Z+q	Load Indirect with Displacement	Rd ← (Z + q)	None	2
STS	k, Rr	Store Direct to Data Space	Rd ← (k)	None	2
ST	X, Rr	Store Indirect	(X) ← Rr	None	2
ST	X+, Rr	Store Indirect and Post-Increment	(X) ← Rr, X ← X + 1	None	2
ST	-X, Rr	Store Indirect and Pre-Decrement	X ← X - 1, (X) ← Rr	None	2
ST	Y, Rr	Store Indirect	(Y) ← Rr	None	2
ST	Y+, Rr	Store Indirect and Post-Increment	(Y) ← Rr, Y ← Y + 1	None	2

The Stack Pointer points to the data SRAM stack area where the Subroutine and Interrupt Stacks are located. This Stack space in the data SRAM must be defined by the program before any subroutine calls are executed or interrupts are enabled. The stack pointer must be set to point above $60. The Stack Pointer is decremented by one when data is pushed onto the Stack with the PUSH instruction, and it is decremented by two when an address is pushed onto the Stack with subroutine calls and interrupts. The Stack Pointer is incremented by one when data is popped from the Stack with the POP instruction, and it is incremented by two when an address is popped from the Stack with return from subroutine RET or return from interrupt RETI.

Software Control of System Configuration

The software control register will allow the software to manage select system level configuration bits.

Software Control Register – SFTCR

Bit	7	6	5	4	3	2	1	0	
$3A ($5A)	-	-	-	-	FMXOR	WDTS	DBG	SRST	SFTCR
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7..4 - Res: Reserved Bits**

These bits are reserved in the AT94K and always read as zero.

- **Bit 3 - FMXOR: Frame Mode XOR (Enable/Disable)**

This bit is XORed with the System Control Register's Enable Frame Interface bit. The behavior when this bit is set to 1 is dependent on how the SCR was initialized. If the Enable Frame Interface bit in the SCR is 0, the FMXOR bit enables the Frame Interface when set to 1. If the Enable Frame Interface bit in the SCR is 1, the FMXOR bit disables the Frame Interface when set to 1. During AVR reset, the FMXOR bit is cleared by the hardware.

- **Bit 2 - WDTS: Software Watchdog Test Clock Select**

When this bit is set to 1, the test clock signal is selected to replace the AVR internal oscillator into the associated watchdog timer logic. During AVR reset, the WDTS bit is cleared by the hardware.

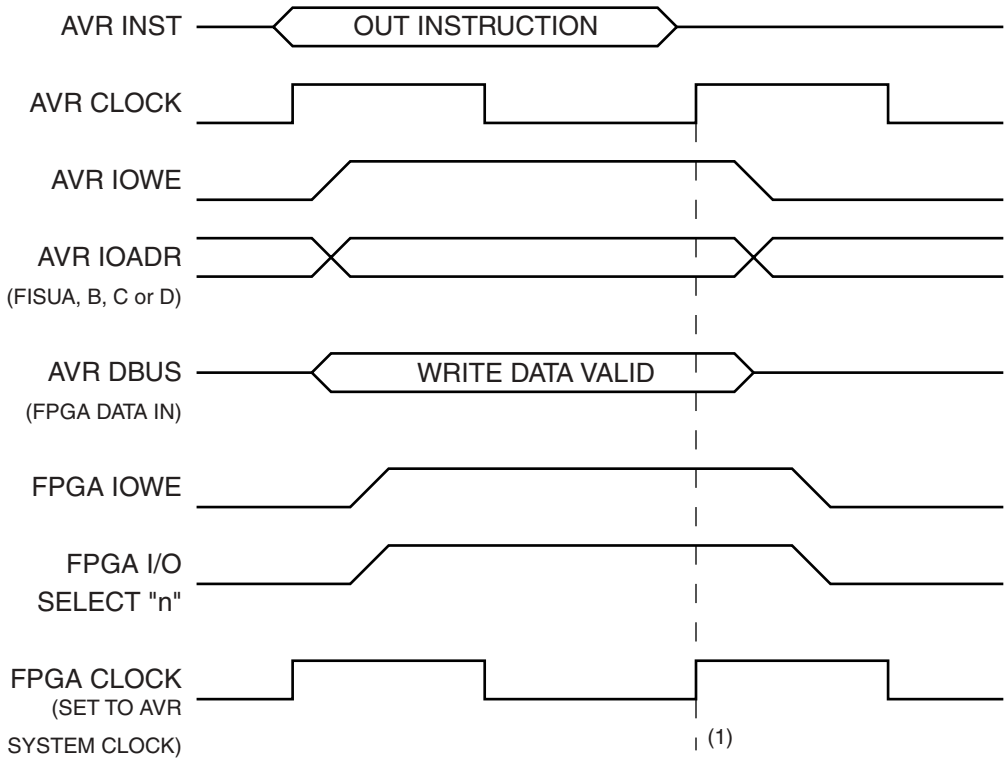
- **Bit 1 - DBG: Debug Mode**

When this bit is set to 1, the AVR can write its own program SRAM. During AVR reset, the DBG bit is cleared by the hardware.

- **Bit 0 - SRST: Software Reset**

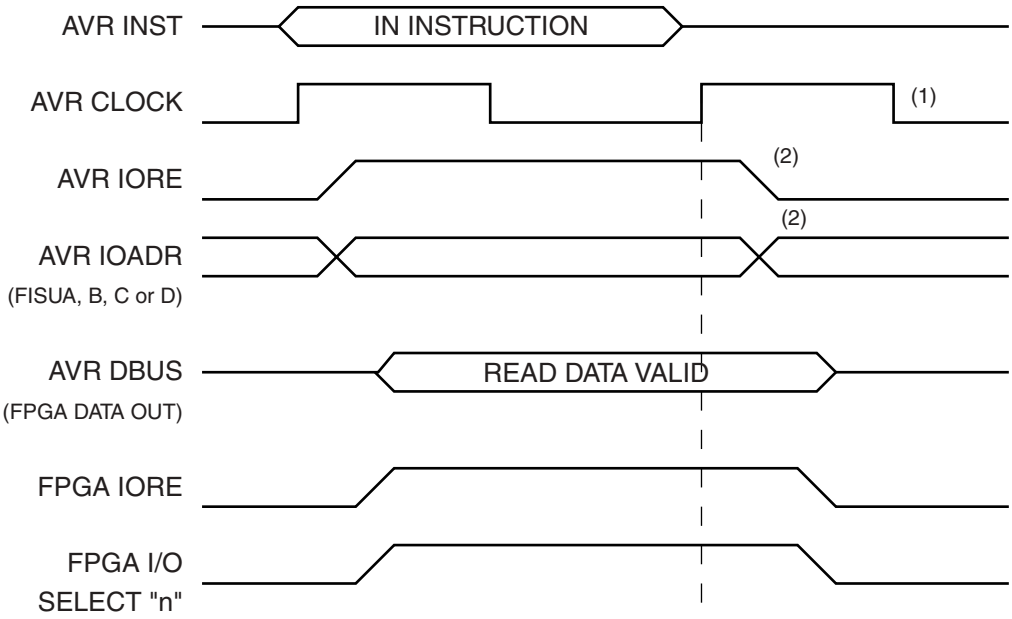
When this bit is set (one), a reset request is sent to the system configuration external to the AVR. Appropriate reset signals are generated back into the AVR and configuration download is initiated. A software reset will cause the EXTRF bit in the MCUR register to be set (one), which remains set throughout the AVR reset and may be read by the restarted program upon reset complete. The external reset flag is set (one) since the requested reset is issued from the system configuration external to the AVR core. During AVR reset, the SRST bit is cleared by the hardware.

Figure 33. Out Instruction – AVR Writing to the FPGA



Note: 1. AVR expects Write to be captured by the FPGA upon posedge of the AVR clock.

Figure 34. In Instruction – AVR Reading FPGA



Notes: 1. AVR captures read data upon posedge of the AVR clock.
2. At the end of an FPGA read cycle, there is a chance for the AVR data bus contention between the FPGA and another peripheral to start to drive (active IORE at new address versus FPGAIORE + Select “n”), but since the AVR clock would have already captured the data from AVR DBUS (= FPGA Data Out), this is a “don’t care” situation.

The most typical program setup for the Reset and Interrupt Vector Addresses are:

```
Address     Labels     Code              Comments
$0000       jmp        RESET             Reset Handle: Program Execution Starts Here
$0002       jmp        FPGA_INT0         ; FPGA Interrupt0 Handle
$0004       jmp        EXT_INT0          ; External Interrupt0 Handle
$0006       jmp        FPGA_INT1         ; FPGA Interrupt1 Handle
$0008       jmp        EXT_INT1          ; External Interrupt1 Handle
$000A       jmp        FPGA_INT2         ; FPGA Interrupt2 Handle
$000C       jmp        EXT_INT2          ; External Interrupt2 Handle
$000E       jmp        FPGA_INT3         ; FPGA Interrupt3 Handle
$0010       jmp        EXT_INT3          ; External Interrupt3 Handle
$0012       jmp        TIM2_COMP         ; Timer/Counter2 Compare Match Interrupt Handle
$0014       jmp        TIM2_OVF          ; Timer/Counter2 Overflow Interrupt Handle
$0016       jmp        TIM1_CAPT         ; Timer/Counter1 Capture Event Interrupt Handle
$0018       jmp        TIM1_COMPA        ; Timer/Counter1 Compare Match A Interrupt Handle
$001A       jmp        TIM1_COMPB        ; Timer/Counter1 Compare Match B Interrupt Handle
$001C       jmp        TIM1_OVF          ; Timer/Counter1 Overflow Interrupt Handle
$001E       jmp        TIM0_COMP         ; Timer/Counter0 Compare Match Interrupt Handle
$0020       jmp        TIM0_OVF          ; Timer/Counter0 Overflow Interrupt Handle
$0022       jmp        FPGA_INT4         ; FPGA Interrupt4 Handle
$0024       jmp        FPGA_INT5         ; FPGA Interrupt5 Handle
$0026       jmp        FPGA_INT6         ; FPGA Interrupt6 Handle
$0028       jmp        FPGA_INT7         ; FPGA Interrupt7 Handle
$002A       jmp        UART0_RXC         ; UART0 Receive Complete Interrupt Handle
$002C       jmp        UART0_DRE         ; UART0 Data Register Empty Interrupt Handle
$002E       jmp        UART0_TXC         ; UART0 Transmit Complete Interrupt Handle
$0030       jmp        FPGA_INT8         ; FPGA Interrupt8 Handle(1)
$0032       jmp        FPGA_INT9         ; FPGA Interrupt9 Handle(1)
$0034       jmp        FPGA_INT10        ; FPGA Interrupt10 Handle(1)
$0036       jmp        FPGA_INT11        ; FPGA Interrupt11 Handle(1)
$0038       jmp        UART1_RXC         ; UART1 Receive Complete Interrupt Handle
$003A       jmp        UART1_DRE         ; UART1 Data Register Empty Interrupt Handle
$003C       jmp        UART1_TXC         ; UART1 Transmit Complete Interrupt Handle
$003E       jmp        FPGA_INT12        ; FPGA Interrupt12 Handle(1)
$0040       jmp        FPGA_INT13        ; FPGA Interrupt13 Handle(1)
$0042       jmp        FPGA_INT14        ; FPGA Interrupt14 Handle(1)
$0044       jmp        FPGA_INT15        ; FPGA Interrupt15 Handle(1)
$0046       jmp        TWS_INT           ; 2-wire Serial Interrupt
;
RESET:
$0048       ldi        r16,high(RAMEND)  ; Main program start
$0049       out        SPH,r16

$004A       ldi        r16,low(RAMEND)
$004B       out        SPL,r16
$004C       <instr>    xxx
  ...         ...        ...
```

Note: 1. Not Available on AT94K05. However, the vector jump table positions must be maintained for appropriate UART and 2-wire serial interrupt jumps.

- **Bit 3 - ICF1: Input Capture Flag 1**

The ICF1 bit is set (one) to flag an input capture event, indicating that the Timer/Counter1 value has been transferred to the input capture register – ICR1. ICF1 is cleared by the hardware when executing the corresponding interrupt handling vector. Alternatively, ICF1 is cleared by writing a logic 1 to the flag. When the SREG I-bit, and TICIE1 (Timer/Counter1 Input Capture Interrupt Enable), and ICF1 are set (one), the Timer/Counter1 Capture Interrupt is executed.

- **Bit 2 - OCF2: Output Compare Flag 2**

The OCF2 bit is set (one) when compare match occurs between Timer/Counter2 and the data in OCR2 – Output Compare Register 2. OCF2 is cleared by the hardware when executing the corresponding interrupt handling vector. Alternatively, OCF2 is cleared by writing a logic 1 to the flag. When the I-bit in SREG, and OCIE2 (Timer/Counter2 Compare Interrupt Enable), and the OCF2 are set (one), the Timer/Counter2 Output Compare Interrupt is executed.

- **Bit 1 - TOV0: Timer/Counter0 Overflow Flag**

The TOV0 bit is set (one) when an overflow occurs in Timer/Counter0. TOV0 is cleared by the hardware when executing the corresponding interrupt handling vector. Alternatively, TOV0 is cleared by writing a logic 1 to the flag. When the SREG I-bit, and TOIE0 (Timer/Counter0 Overflow Interrupt Enable), and TOV0 are set (one), the Timer/Counter0 Overflow interrupt is executed. In PWM mode, this bit is set when Timer/Counter0 advances from $00.

- **Bit 0 - OCF0: Output Compare Flag 0**

The OCF0 bit is set (one) when compare match occurs between Timer/Counter0 and the data in OCR0 – Output Compare Register 0. OCF0 is cleared by the hardware when executing the corresponding interrupt handling vector. Alternatively, OCF0 is cleared by writing a logic 1 to the flag. When the I-bit in SREG, and OCIE0 (Timer/Counter2 Compare Interrupt Enable), and the OCF0 are set (one), the Timer/Counter0 Output Compare Interrupt is executed.

Interrupt Response Time

The interrupt execution response for all the enabled AVR interrupts is four clock cycles minimum. Four clock cycles after the interrupt flag has been set, the program vector address for the actual interrupt handling routine is executed. During this four clock-cycle period, the Program Counter (2 bytes) is pushed onto the Stack, and the Stack Pointer is decremented by 2. The vector is normally a jump to the interrupt routine, and this jump takes three clock cycles. If an interrupt occurs during execution of a multi-cycle instruction, this instruction is completed before the interrupt is serviced.

A return from an interrupt handling routine (same as for a subroutine call routine) takes four clock cycles. During these four clock cycles, the Program Counter (2 bytes) is popped back from the Stack, and the Stack Pointer is incremented by 2. When the AVR exits from an interrupt, it will always return to the main program and execute one more instruction before any pending interrupt is serviced.

Sleep Modes

To enter any of the three Sleep modes, the SE bit in MCUR must be set (one) and a SLEEP instruction must be executed. The SM1 and SM0 bits in the MCUR register select which Sleep mode (Idle, Power-down, or Power-save) will be activated by the SLEEP instruction, see Table 12 on page 52.

In Power-down and Power-save modes, the four external interrupts, EXT_INT0...3, and FPGA interrupts, FPGA INT0...3, are triggered as low level-triggered interrupts. If an enabled interrupt occurs while the MCU is in a Sleep mode, the MCU awakes, executes the interrupt routine, and resumes execution from the instruction following SLEEP. The contents of the register file, SRAM, and I/O memory are unaltered. If a reset occurs during Sleep mode, the MCU wakes up and executes from the Reset vector

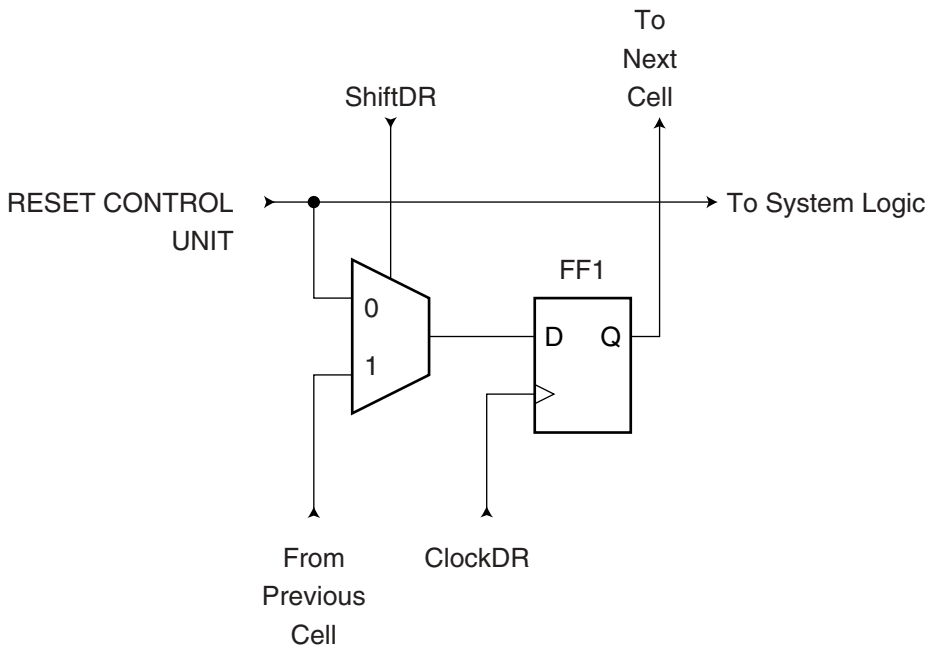
When no alternate port function is present, the Input Data - ID corresponds to the PINn register value, Output Data corresponds to the PORTn register, Output Control corresponds to the Data Direction (DDn) register, and the PuLL-up Disable (PLD) corresponds to logic expression (DDn OR NOT(PORTBn)).

Digital alternate port functions are connected outside the dashed box in Figure 44 to make the scan chain read the actual pin value.

Scanning AVR RESET

Multiple sources contribute to the internal AVR reset; therefore, the AVR reset pin is not observed. Instead, the internal AVR reset signal output from the Reset Control Unit is observed, see Figure 45. The scanned signal is active High if AVRResetn is Low and enabled or the device is in general reset (Resetn or power-on) or configuration download.

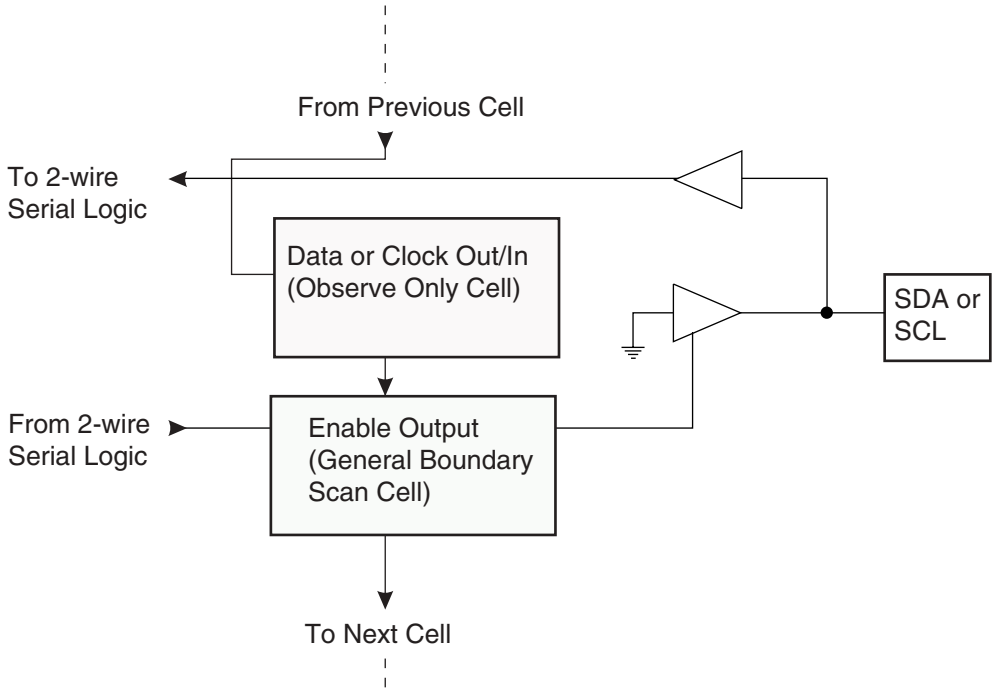
Figure 45. Observe-only Cell



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.

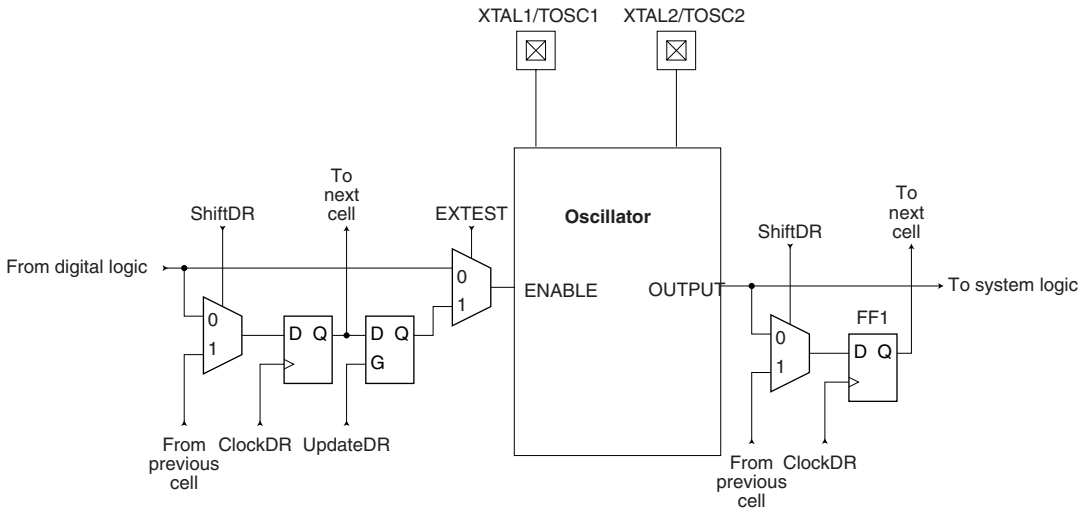
Figure 46. Boundary-scan Cells for 2-wire Serial



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.

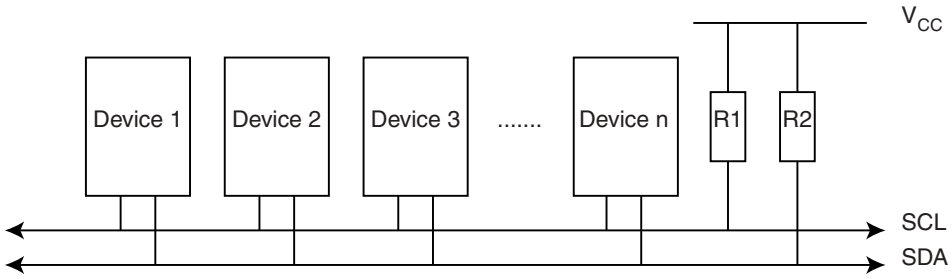
Figure 47. Boundary-scan Cells for Oscillators and Clock Options



2-wire Serial Interface (Byte Oriented)

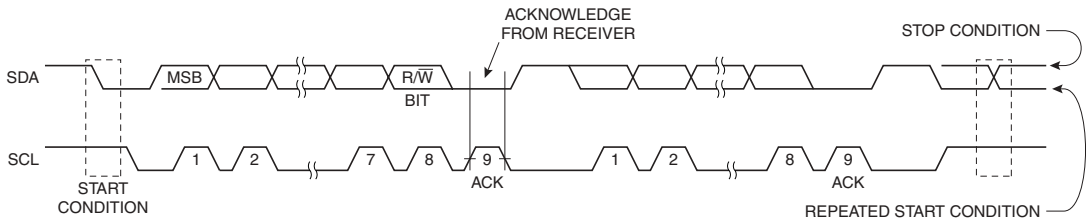
The 2-wire Serial Bus is a bi-directional two-wire serial communication standard. It is designed primarily for simple but efficient integrated circuit (IC) control. The system is comprised of two lines, SCL (Serial Clock) and SDA (Serial Data) that carry information between the ICs connected to them. Various communication configurations can be designed using this bus.
Figure 68 shows a typical 2-wire Serial Bus configuration. Any device connected to the bus can be Master or Slave.

Figure 68. 2-wire Serial Bus Configuration



The 2-wire Serial Interface provides a serial interface that meets the 2-wire Serial Bus specification and supports Master/Slave and Transmitter/Receiver operation at up to 400 kHz bus clock rate. The 2-wire Serial Interface has hardware support for the 7-bit addressing, but is easily extended to 10-bit addressing format in software. When operating in 2-wire Serial mode, i.e., when TWEN is set, a glitch filter is enabled for the input signals from the pins SCL and SDA, and the output from these pins are slew-rate controlled. The 2-wire Serial Interface is byte oriented. The operation of the serial 2-wire Serial Bus is shown as a pulse diagram in Figure 69, including the START and STOP conditions and generation of ACK signal by the bus receiver.

Figure 69. 2-wire Serial Bus Timing Diagram



The block diagram of the 2-wire Serial Bus interface is shown in Figure 70.

Figure 74. Formats and States in the Slave Transmitter Mode

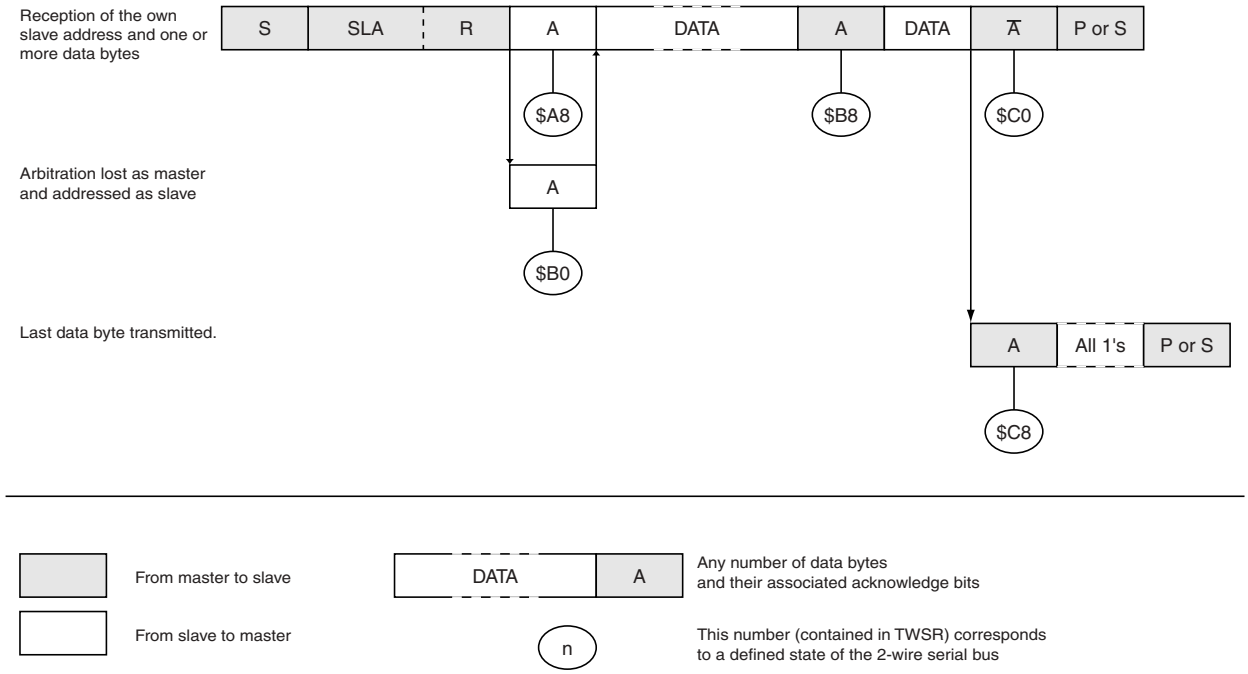
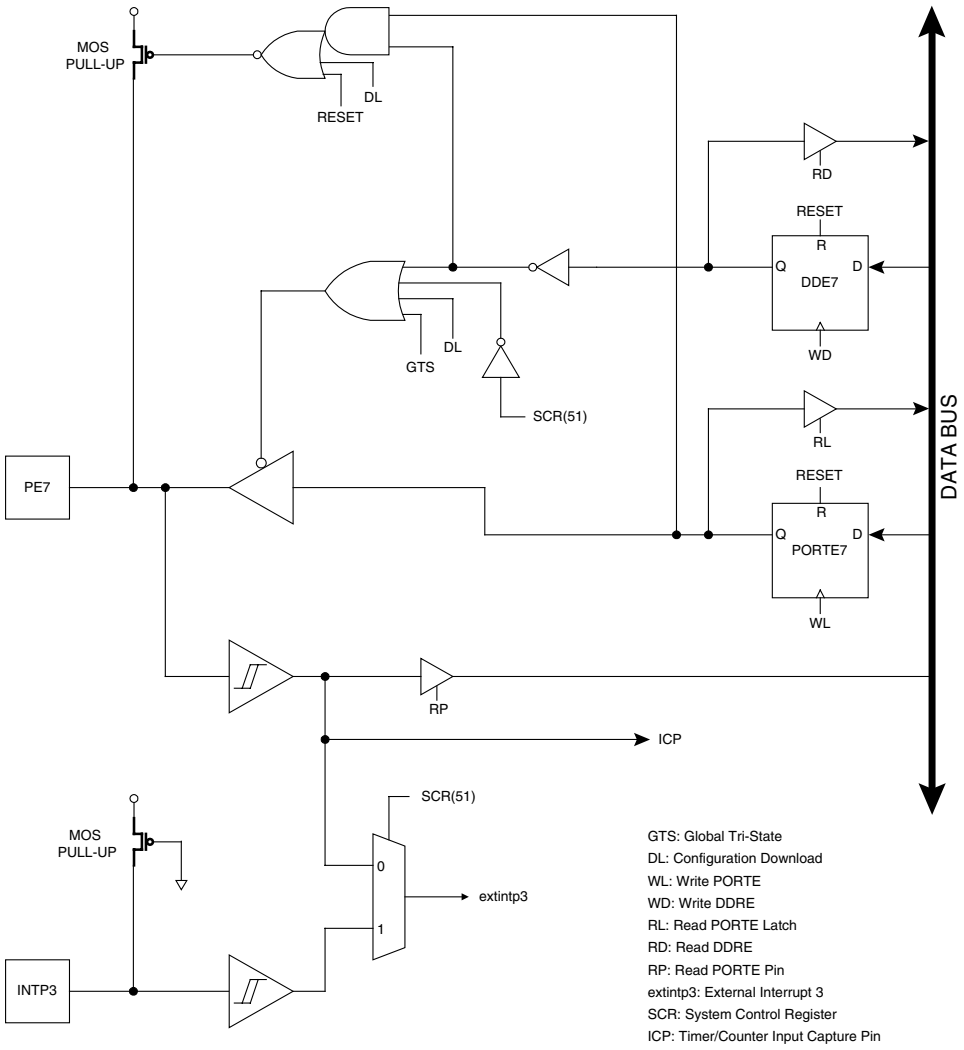


Table 45. Status Codes for Miscellaneous States

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/From TWDR	To TWCR				
			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.

Figure 82. PortE Schematic Diagram (Pin PE7)



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} = 3.00V, temperature = 70°C
Minimum times based on best case: V_{CC} = 3.60V, temperature = 0°C
Maximum delays are the average of t_{PDLH} and t_{PDHL}.

Cell Function	Parameter	Path	-25	Units	Notes
Core					
2 Input Gate	t_{PD} (Maximum)	x/y -> x/y	2.9	ns	1 Unit Load
3 Input Gate	t_{PD} (Maximum)	x/y/z -> x/y	2.8	ns	1 Unit Load
3 Input Gate	t_{PD} (Maximum)	x/y/w -> x/y	3.4	ns	1 Unit Load
4 Input Gate	t_{PD} (Maximum)	x/y/w/z -> x/y	3.4	ns	1 Unit Load
Fast Carry	t_{PD} (Maximum)	y -> y	2.3	ns	1 Unit Load
Fast Carry	t_{PD} (Maximum)	x -> y	2.9	ns	1 Unit Load
Fast Carry	t_{PD} (Maximum)	y -> x	3.0	ns	1 Unit Load
Fast Carry	t_{PD} (Maximum)	x -> x	2.3	ns	1 Unit Load
Fast Carry	t_{PD} (Maximum)	w -> y	3.4	ns	1 Unit Load
Fast Carry	t_{PD} (Maximum)	w -> x	3.4	ns	1 Unit Load
Fast Carry	t_{PD} (Maximum)	z -> y	3.4	ns	1 Unit Load
Fast Carry	t_{PD} (Maximum)	z -> x	2.4	ns	1 Unit Load
DFF	t_{PD} (Maximum)	q -> x/y	2.8	ns	1 Unit Load
DFF	t_{setup} (Minimum)	x/y -> clk	–	–	–
DFF	t_{hold} (Minimum)	x/y -> clk	–	–	–
DFF	t_{PD} (Maximum)	R -> x/y	3.2	ns	1 Unit Load
DFF	t_{PD} (Maximum)	S -> x/y	3.0	ns	1 Unit Load
DFF	t_{PD} (Maximum)	q -> w	2.7	ns	–
incremental -> L	t_{PD} (Maximum)	x/y -> L	2.4	ns	–
Local Output Enable	t_{PZX} (Maximum)	oe -> L	2.8	ns	1 Unit Load
Local Output Enable	t_{PXZ} (Maximum)	oe -> L	2.4	ns	

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).

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.

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/O100	I/O148	I/O292			114	164
		I/O293				
		I/O294				
		GND				
		I/O295				
		I/O296				
I/O101 ($\overline{CS1}$, A2)	I/O149 ($\overline{CS1}$, A2)	I/O297 ($\overline{CS1}$, A2)	79	80	115	165
I/O102 (A3)	I/O150 (A3)	I/O298 (A3)	80	81	116	166
		I/O299				
		I/O300				
		VCC[1]				
		GND				
I/O104	I/O151	I/O301	Shorted to Testclock	Shorted to Testclock	Shorted to Testclock	Shorted to Testclock
	I/O152	I/O302				
I/O103	I/O153	I/O303			117	167
	I/O154	I/O304				168
		I/O305				
		I/O306				
		GND				
		I/O307				
		I/O308				
	I/O155	I/O309				169
	I/O156	I/O310				170
		I/O311				
		I/O312				
GND	GND	GND			118	171
I/O105	I/O157	I/O313			119	172
I/O106	I/O158	I/O314			120	173
	I/O159	I/O315				
	I/O160	I/O316				
	VCC[1]	VCC[1]				
		I/O317				
		I/O318				

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
		GND				
		I/O319				
		I/O320				
		I/O321				
		I/O322				
		I/O323				
		I/O324				
		GND				
		VCC(1)				
I/O107 (A4)	I/O161 (A4)	I/O325 (A4)	81	82	121	174
I/O108 (A5)	I/O162 (A5)	I/O326 (A5)	82	83	122	175
		GND				
	I/O163	I/O327				176
	I/O164	I/O328				177
I/O109	I/O165	I/O329		84	123	178
I/O110	I/O166	I/O330		85	124	179
		GND				
		I/O331				
		I/O332				
		I/O333				
		I/O334				
I/O111 (A6)	I/O167 (A6)	I/O335 (A6)	83	86	125	180
I/O112 (A7)	I/O168 (A7)	I/O336 (A7)	84	87	126	181
GND	GND	GND	1	88	127	182
VDD(2)	VDD(2)	VDD(2)	2	89	128	183
I/O113 (A8)	I/O169 (A8)	I/O337 (A8)	3	90	129	184
I/O114 (A9)	I/O170 (A9)	I/O338 (A9)	4	91	130	185
		I/O339				
		I/O340				
		I/O341				
		I/O342				
		GND				
I/O115	I/O171	I/O343		92	131	186
I/O116	I/O172	I/O344		93	132	187

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.