Welcome to E-XFL.COM

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

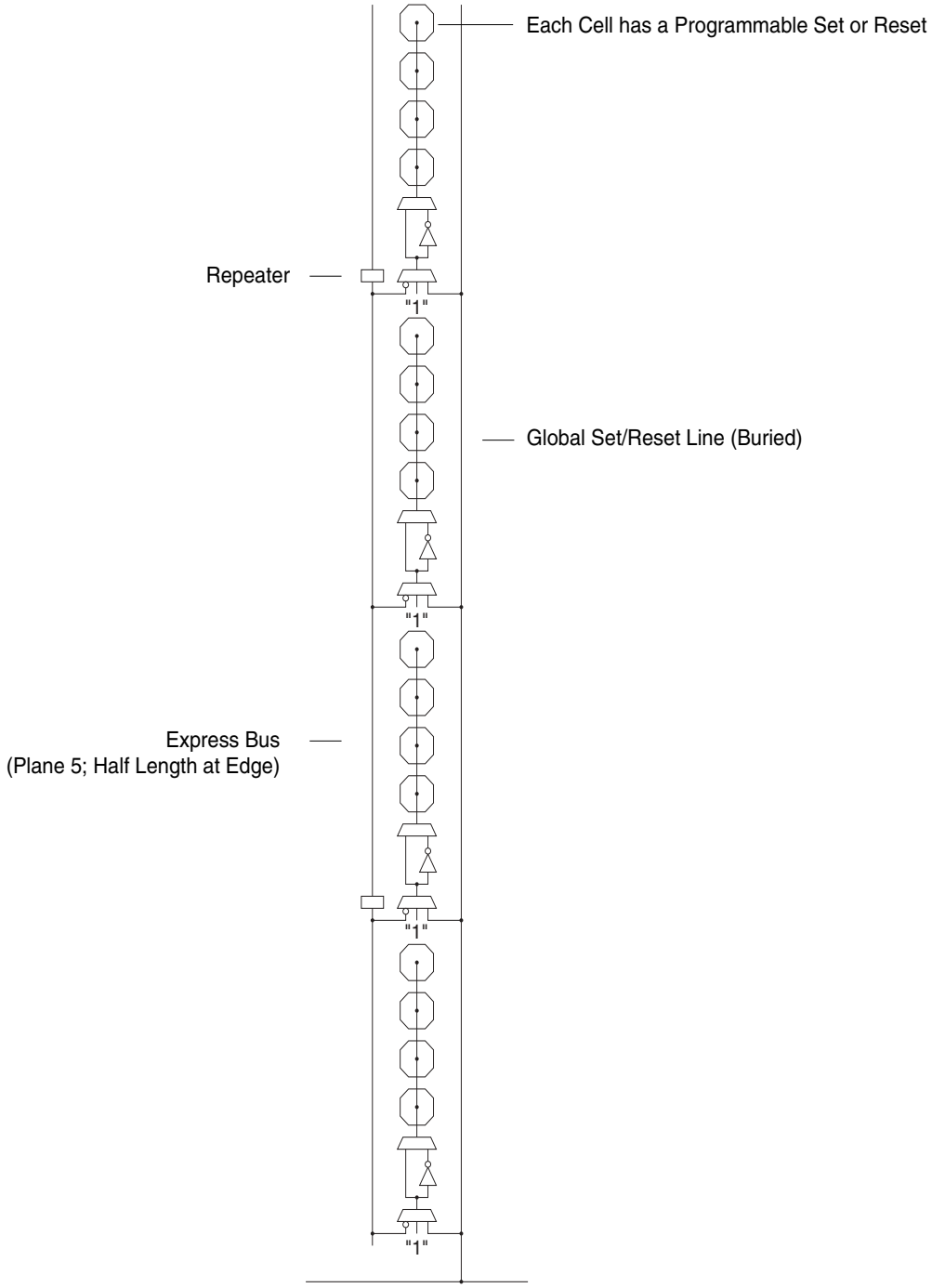
**Embedded - FPGAs (Field Programmable Gate Arrays) with Microcontrollers** represent a cutting-edge category of electronic components that combine the flexibility of FPGA technology with the processing power of integrated microcontrollers. This hybrid approach offers a versatile solution for designing 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 | -40°C ~ 85°C |
| Package / Case | 144-LQFP |
| Supplier Device Package | 144-LQFP (20x20) |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/at94k10al-25bqi |

**Figure 13.** Set/Reset (for One Column of Cells)




Some of the bus resources on the embedded FPGA core are used as dual-function resources. Table 4 shows which buses are used in a dual-function mode and which bus plane is used. The FPGA software tools are designed to automatically accommodate dual-function buses in an efficient manner.

**Figure 17.** Corner I/Os

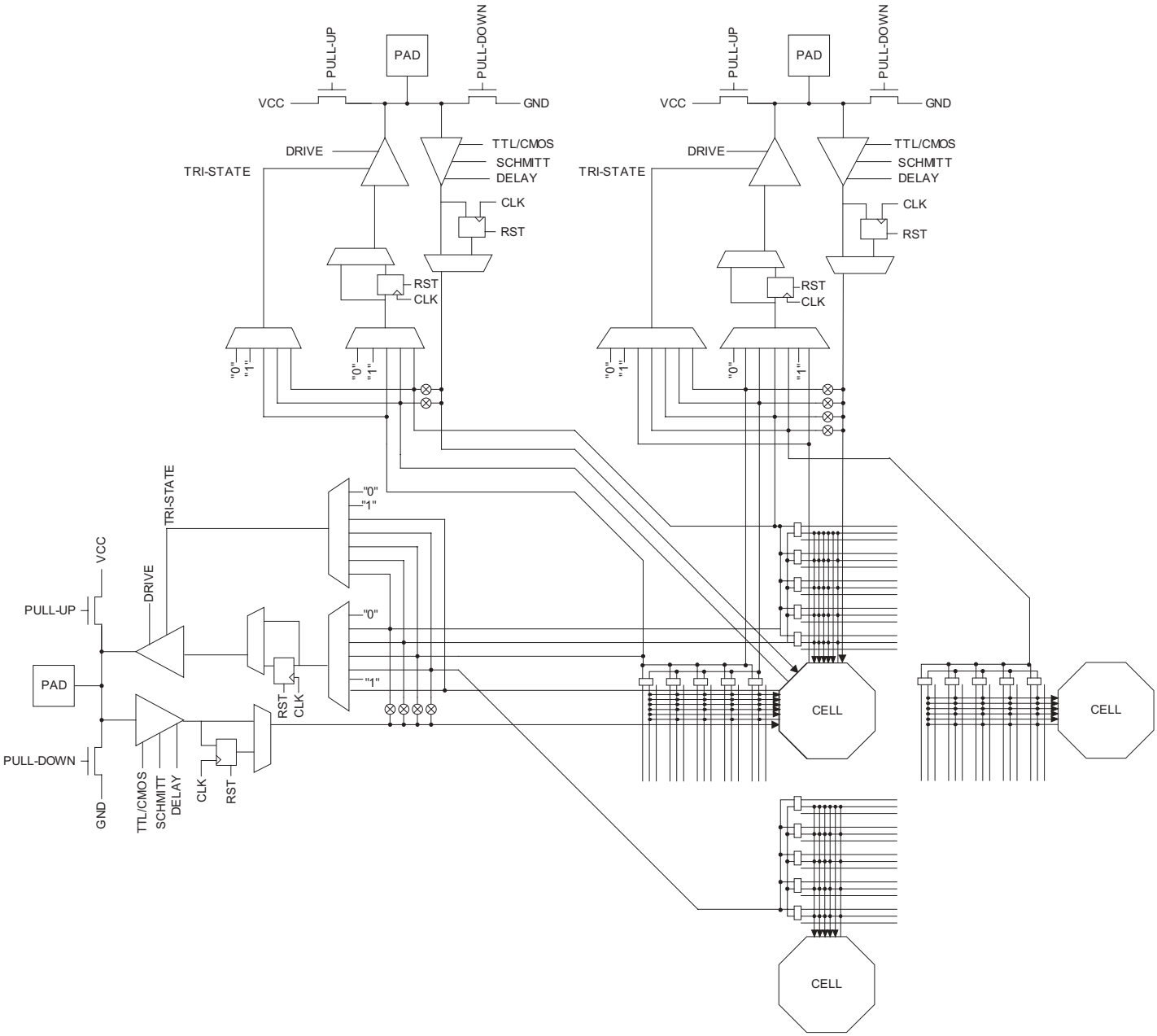


**Table 7.** Summary Table for AVR and FPGA SRAM Addressing (Continued)

| SRAM | FPGA and AVR DBG Address Range | AVR Data Address Range | AVR PC Address Range |
|---|---|---|---|
| 05[1] | $2800 - $2FFF | $2800 - $2FFF | $3000 - $37FF (MS Byte) |
| 06[1] | $3000 - $37FF | $3000 - $37FF | $2800 - $2FFF (LS Byte) |
| 07[1] | $3800 - $3FFF | $3800 - $3FFF | $2800 - $2FFF (MS Byte) |
| 08 | $4000 - $47FF | | $2000 - $27FF (LS Byte) |
| 09 | $4800 - $4FFF | | $2000 - $27FF (MS Byte) |
| 10 | $5000 - $57FF | | $1800 - $1FFF (LS Byte) |
| 11 | $5800 - $5FFF | | $1800 - $1FFF (MS Byte) |
| 12 | $6000 - $67FF | | $1000 - $17FF (LS Byte) |
| 13 | $6800 - $6FFF | | $1000 - $17FF (MS Byte) |
| 14 | $7000 - $77FF | | $0800 - $0FFF (LS Byte) |
| 15 | $7800 - $7FFF | | $0800 - $0FFF (MS Byte) |
| 16 | $8000 - $87FF | | $0000 - $07FF (LS Byte) |
| 17 = n | $8800 - $8FFF | | $0000 - $07FF (MS Byte) |

Note: 1. Whether these SRAMs are “Data” or “Program” depends on the SCR40 and SCR41 values.

Example: Frame (and AVR debug mode) write of instructions to associated AVR PC addresses, see Table 8 and Table 9.

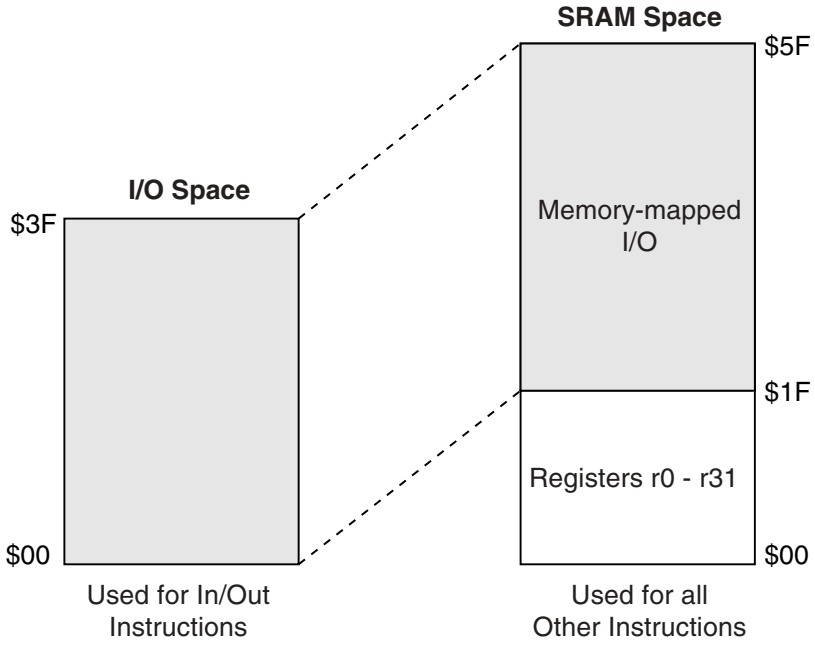
**Table 8.** AVR PC Addresses

| AVR PC | Instruction |
|---|---|
| 0FFE | 9B28 |
| 0FFF | CFFE |
| 1000 | B300 |
| 1001 | 9A39 |

**Table 9.** Frame Addresses

| Frame Address | Frame Data |
|---|---|
| 77FE | 28 |
| 77FF | FE |
| 6000 | 00 |
| 6001 | 39 |
| 7FFE | 9B |
| 7FFF | CF |
| 6800 | B3 |
| 6801 | 9A |

**Figure 32.** Memory-mapped I/O




For single-cycle access (In/Out Commands) to I/O, the instruction has to be less than 16 bits:

| opcode | register | address |
|---|---|---|
| | r0 - 31 ($1F) | r0 - 63 ($3F) |
| 5 bits | 5 bits | 6 bits |

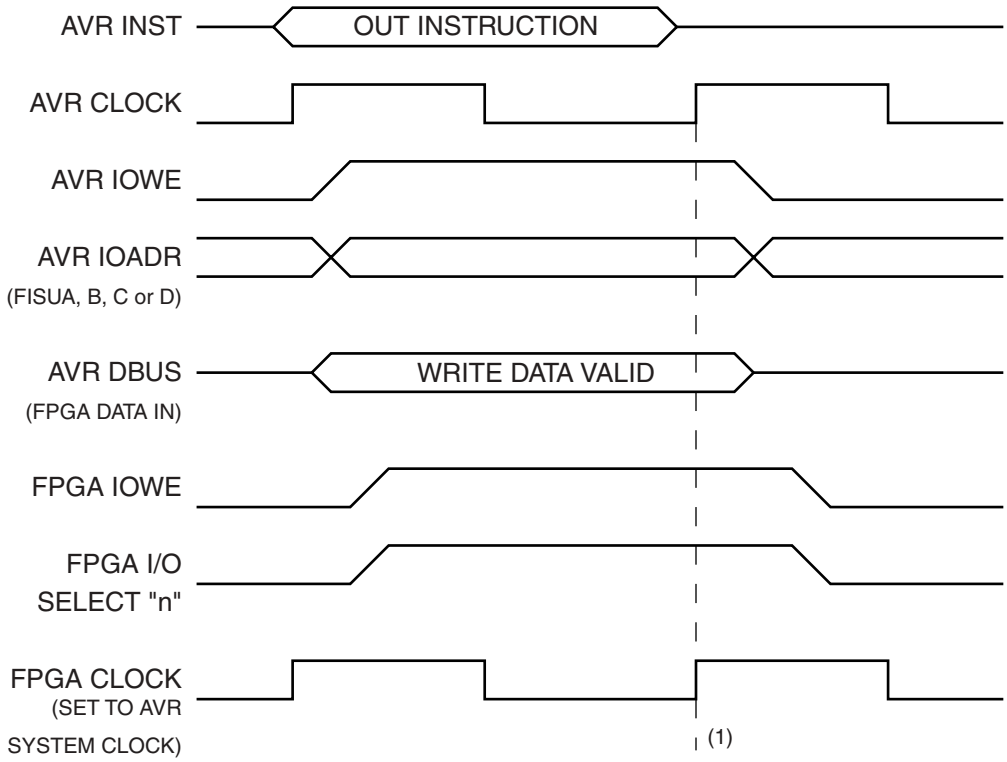
In the data SRAM, the registers are located at memory addresses $00 - $1F and the I/O space is located at memory addresses $20 - $5F.

As there are only 6 bits available to refer to the I/O space, the address is shifted down 2 bits. This means the In/Out commands access $00 to $3F which goes directly to the I/O and maps to $20 to $5F in SRAM. All other instructions access the I/O space through the $20 - $5F addressing.

For compatibility with future devices, reserved bits should be written zero if accessed. Reserved I/O memory addresses should never be written.

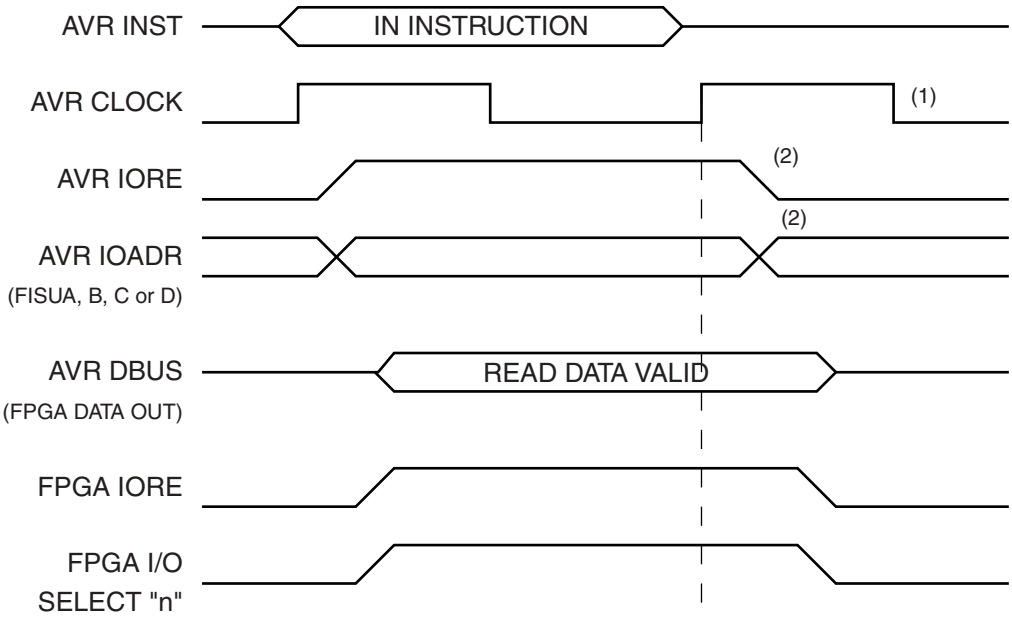
The status flags are cleared by writing a logic 1 to them. Note that the CBI and SBI instructions will operate on all bits in the I/O register, writing a one back into any flag read as set, thus clearing the flag. The CBI and SBI instructions work with registers $00 to $1F only.

**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.

## 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 |

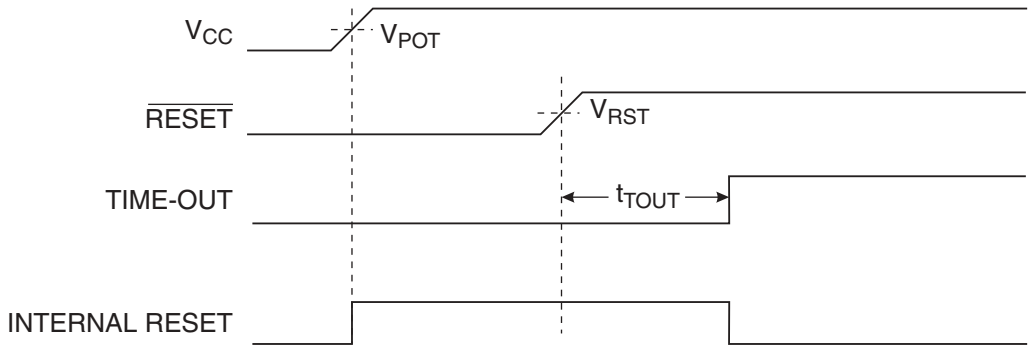
**Table 15.** Reset and Interrupt Vectors (Continued)

| Vector No. (hex) | Program Address | Source | Interrupt Definition |
|---|---|---|---|
| 17 | $002C | UART0_DRE | UART0 Data Register Empty Interrupt Handle |
| 18 | $002E | UART0_TXC | UART0 Transmit Complete Interrupt Handle |
| 19 | $0030 | FPGA_INT8 | FPGA Interrupt8 Handle (not available on AT94K05) |
| 1A | $0032 | FPGA_INT9 | FPGA Interrupt9 Handle (not available on AT94K05) |
| 1B | $0034 | FPGA_INT10 | FPGA Interrupt10 Handle (not available on AT94K05) |
| 1C | $0036 | FPGA_INT11 | FPGA Interrupt11 Handle (not available on AT94K05) |
| 1D | $0038 | UART1_RXC | UART1 Receive Complete Interrupt Handle |
| 1E | $003A | UART1_DRE | UART1 Data Register Empty Interrupt Handle |
| 1F | $003C | UART1_TXC | UART1 Transmit Complete Interrupt Handle |
| 20 | $003E | FPGA_INT12 | FPGA Interrupt12 Handle (not available on AT94K05) |
| 21 | $0040 | FPGA_INT13 | FPGA Interrupt13 Handle (not available on AT94K05) |
| 22 | $0042 | FPGA_INT14 | FPGA Interrupt14 Handle (Not Available on AT94K05) |
| 23 | $0044 | FPGA_INT15 | FPGA Interrupt15 Handle (not available on AT94K05) |
| 24 | $0046 | TWS_INT | 2-wire Serial Interrupt |

The MCU after five CPU clock-cycles, and can be used when an external clock signal is applied to the XTAL1 pin. This setting does not use the WDT oscillator, and enables very fast start-up from the Sleep, Power-down or Power-save modes if the clock signal is present during sleep.

RESET can be connected to $V_{CC}$ directly or via an external pull-up resistor. By holding the pin Low for a period after $V_{CC}$ has been applied, the Power-on Reset period can be extended. Refer to Figure 38 for a timing example on this.

**Figure 38.** MCU Start-up, $\overline{\text{RESET}}$ Controlled Externally




**External Reset**

An external reset is generated by a low-level on the AVRRESET pin. When the applied signal reaches the Reset Threshold Voltage – $V_{RST}$ – on its positive edge, the delay timer starts the MCU after the Time-out period $t_{TOUT}$ has expired.

**Watchdog Reset**

When the Watchdog times out, it will generate a short reset pulse of 1 XTAL cycle duration. On the falling edge of this pulse, the delay timer starts counting the Time-out period $t_{TOUT}$. Time-out period $t_{TOUT}$ is approximately 3 µs – at $V_{CC}$ = 3.3V. the period of the time out is voltage dependent.

**Software Reset**

See “Software Control of System Configuration” on page 51.

**Interrupt Handling**

The embedded AVR core has one dedicated 8-bit Interrupt Mask control register: TIMSK – Timer/Counter Interrupt Mask Register. In addition, other enable and mask bits can be found in the peripheral control registers.

When an interrupt occurs, the Global Interrupt Enable I-bit is cleared (zero) and all interrupts are disabled. The user software can set (one) the I-bit to enable nested interrupts. The I-bit is set (one) when a Return from Interrupt instruction (RETI) is executed.

When the Program Counter is vectored to the actual interrupt vector in order to execute the interrupt handling routine, the hardware clears the corresponding flag that generated the interrupt. Some of the interrupt flags can also be cleared by writing a logic 1 to the flag bit position(s) to be cleared.

If an interrupt condition occurs when the corresponding interrupt enable bit is cleared (zero), the interrupt flag will be set and remembered until the interrupt is enabled, or the flag is cleared by software.

If one or more interrupt conditions occur when the global interrupt enable bit is cleared (zero), the corresponding interrupt flag(s) will be set and remembered until the global interrupt enable bit is set (one), and will be executed by order of priority.

The status register is not automatically stored when entering an interrupt routine and restored when returning from an interrupt routine. This must be handled by software.

- **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

### Part Number

The part number is a 16 bit code identifying the component. The JTAG Part Number for AVR devices is listed in Table 19.

**Table 19.** JTAG Part Number

| Device | Part Number (Hex) |
|---|---|
| AT94K05 | 0xdd77 |
| AT94K10 | 0xdd73 |
| AT94K40 | 0xdd76 |

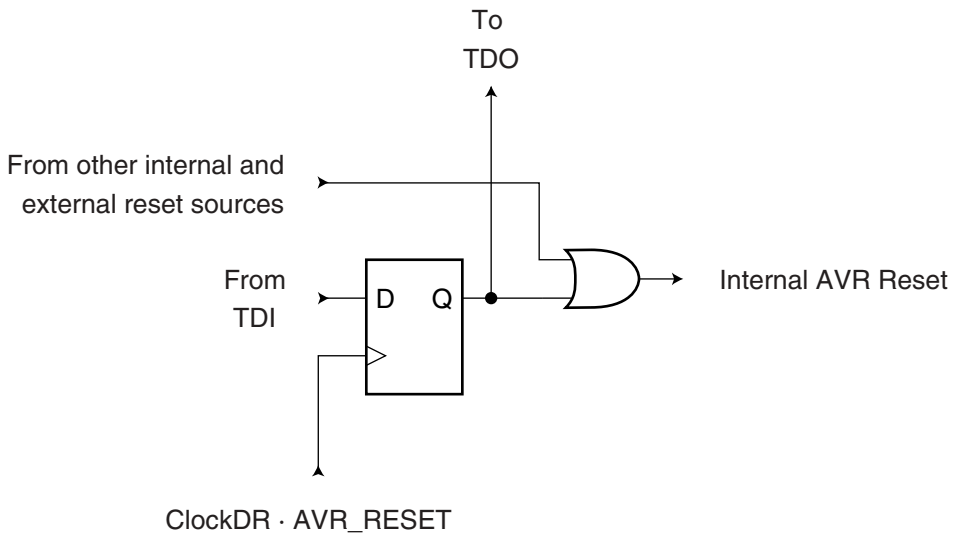
### Manufacturer ID

The manufacturer ID for ATMEL is 0x01F (11 bits).

## AVR Reset Register

The AVR Reset Register is a Test Data Register used to reset the AVR. A high value in the Reset Register corresponds to pulling the external AVRResetn Low. The AVR is reset as long as there is a high value present in the AVR Reset Register. Depending on the Bit settings for the clock options, the CPU will remain reset for a Reset Time-Out Period after releasing the AVR Reset Register. The output from this Data Register is not latched, so the reset will take place immediately, see Figure 42.

**Figure 42.** Reset Register




*Boundary-scan Chain*

The Boundary-scan Chain has the capability of driving and observing the logic levels on the AVR's digital I/O pins.

See "Boundary-scan Chain" on page 76 for a complete description.

### Boundary-scan Specific JTAG Instructions

The instruction register is 4-bit wide, supporting up to 16 instructions. Listed below are the JTAG instructions useful for Boundary-Scan operation. Note that the optional HIGHZ instruction is not implemented.

As a definition in this data sheet, the LSB is shifted in and out first for all shift registers.

The OPCODE for each instruction is shown behind the instruction name in hex format. The text describes which data register is selected as path between TDI and TDO for each instruction.

*EXTEST; $0*

Mandatory JTAG instruction for selecting the Boundary-Scan Chain as Data Register for testing circuitry external to the AVR package. For port-pins, Pull-up Disable, Output Control, Output Data, and Input Data are all accessible in the scan chain. For Analog circuits having off-chip connections, the interface between the analog and the digital logic is in the scan chain. The contents of the latched outputs of the Boundary-Scan chain are driven out as soon as the JTAG IR-register is loaded by the EXTEST instruction.

The active states are:

- Capture-DR: Data on the external pins are sampled into the Boundary-Scan Chain.
- Shift-DR: The Internal Scan Chain is shifted by the TCK input.
- Update-DR: Data from the scan chain is applied to output pins.

*IDCODE; $1*

Optional JTAG instruction selecting the 32-bit ID register as Data Register. The ID register consists of a version number, a device number and the manufacturer code chosen by JEDEC. This is the default instruction after power-up.

The active states are:

- Capture-DR: Data in the IDCODE register is sampled into the Boundary-Scan Chain.
- Shift-DR: The IDCODE scan chain is shifted by the TCK input.

*SAMPLE_PRELOAD; $2*

Mandatory JTAG instruction for pre-loading the output latches and taking a snap-shot of the input/output pins without affecting the system operation. However, the output latches are not connected to the pins. The Boundary-Scan Chain is selected as Data Register.

The active states are:

- Capture-DR: Data on the external pins are sampled into the Boundary-Scan Chain.
- Shift-DR: The Boundary-Scan Chain is shifted by the TCK input.
- Update-DR: Data from the Boundary-Scan chain is applied to the output latches. However, the output latches are not connected to the pins.

*AVR_RESET; $C*

The AVR specific public JTAG instruction for forcing the AVR device into the Reset Mode or releasing the JTAG reset source. The TAP controller is not reset by this instruction. The one bit Reset Register is selected as Data Register. Note that the reset will be active as long as there is a logic “1” in the Reset Chain. The output from this chain is not latched.

The active state is:

- Shift-DR: The Reset Register is shifted by the TCK input.

*BYPASS; $F*

Mandatory JTAG instruction selecting the Bypass Register for Data Register.

The active states are:

- Capture-DR: Loads a logic “0” into the Bypass Register.
- Shift-DR: The Bypass Register cell between TDI and TDO is shifted.

**Boundary-scan Chain**

The Boundary-Scan chain has the capability of driving and observing the logic levels on the AVR’s digital I/O pins.

*Scanning the Digital Port Pins*

Figure 43 shows the boundary-scan cell for bi-directional port pins with pull-up function. The cell consists of a standard boundary-scan cell for the pull-up function, and a bi-directional pin cell that combines the three signals Output Control (OC), Output Data (OD), and Input Data (ID), into only a two-stage shift register.

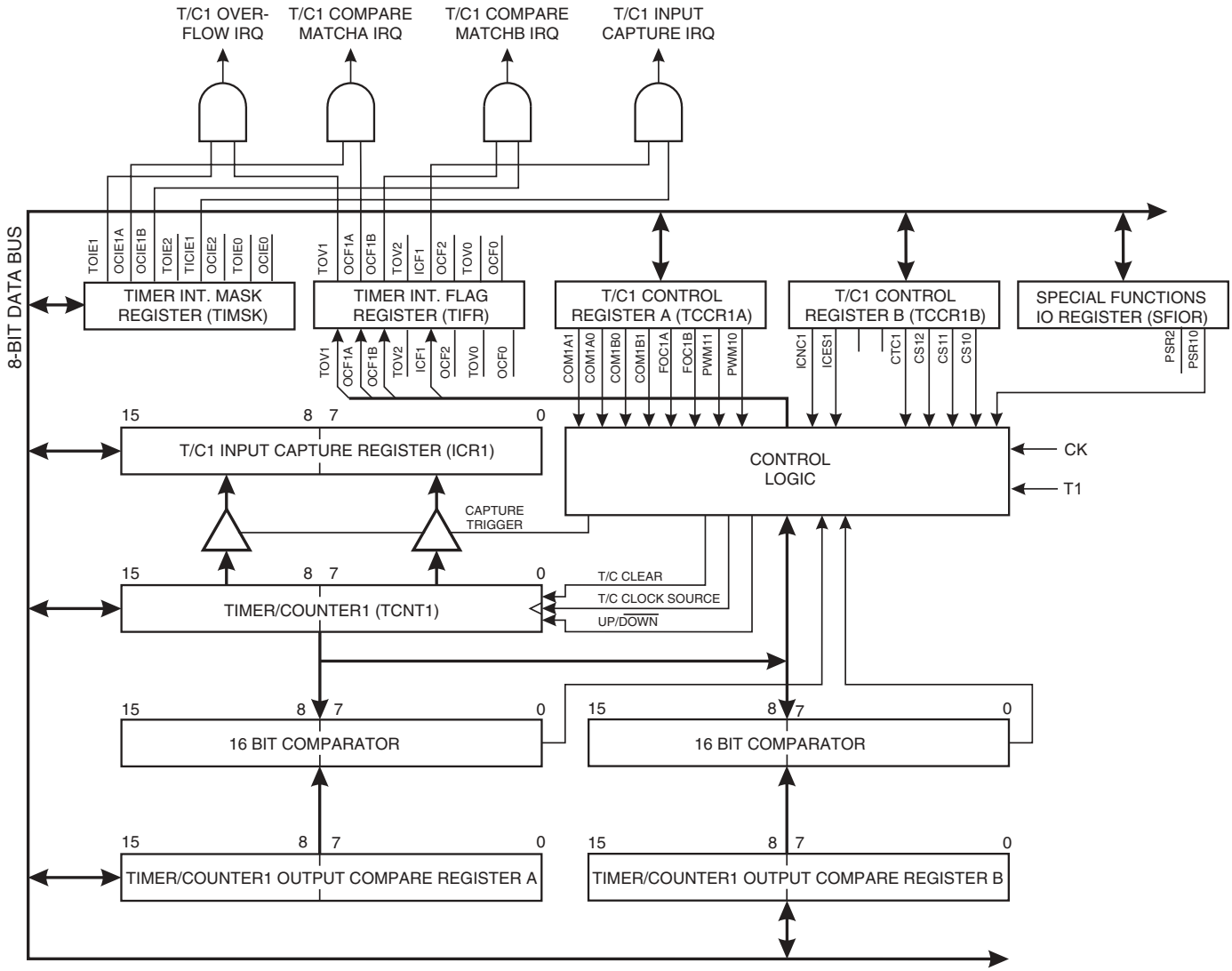
of the timer clock, that is, the timer is always advanced by at least one before the processor can read the counter value. The interrupt flags are updated three processor cycles after the processor clock has started. During these cycles, the processor executes instructions, but the interrupt condition is not readable, and the interrupt routine has not started yet.

- During asynchronous operation, the synchronization of the interrupt flags for the asynchronous timer takes three processor cycles plus one timer cycle. The timer is therefore advanced by at least one before the processor can read the timer value causing the setting of the interrupt flag. The output compare pin is changed on the timer clock and is not synchronized to the processor clock.

## Timer/Counter1

Figure 54 shows the block diagram for Timer/Counter1.

**Figure 54.** Timer/Counter1 Block Diagram



The 16-bit Timer/Counter1 can select the clock source from CK, prescaled CK, or an external pin. In addition it can be stopped as described in section “Timer/Counter1 Control Register B – TCCR1B” on page 98. The different status flags (overflow, compare match and capture event) are found in the Timer/Counter Interrupt Flag Register – TIFR. Control signals are found in the Timer/Counter1 Control Registers – TCCR1A and TCCR1B. The interrupt enable/disable settings for Timer/Counter1 are found in the Timer/Counter Interrupt Mask Register – TIMSK.

When Timer/Counter1 is externally clocked, the external signal is synchronized with the oscillator frequency of the CPU. To assure proper sampling of the external clock, the minimum time between two external clock transitions must be at least one internal CPU clock period. The external clock signal is sampled on the rising edge of the internal CPU clock.

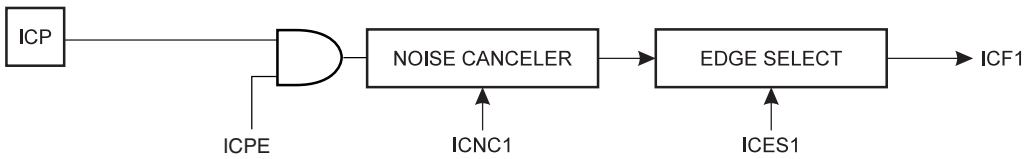
The 16-bit Timer/Counter1 features both a high-resolution and a high-accuracy usage with the lower prescaling opportunities. Similarly, the high-prescaling opportunities makes the Timer/Counter1 useful for lower speed functions or exact-timing functions with infrequent actions.

The Timer/Counter1 supports two Output Compare functions using the Output Compare Register 1 A and B – OCR1A and OCR1B as the data sources to be compared to the Timer/Counter1 contents. The Output Compare functions include optional clearing of the counter on compareA match, and actions on the Output Compare pins on both compare matches.

Timer/Counter1 can also be used as a 8-, 9- or 10-bit Pulse Width Modulator. In this mode, the counter and the OCR1A/OCR1B registers serve as a dual-glitch-free stand-alone PWM with centered pulses. Alternatively, the Timer/Counter1 can be configured to operate at twice the speed in PWM mode, but without centered pulses. Refer to page 101 for a detailed description on this function.

The Input Capture function of Timer/Counter1 provides a capture of the Timer/Counter1 contents to the Input Capture Register – ICR1, triggered by an external event on the Input Capture Pin – PE7(ICP). The actual capture event settings are defined by the Timer/Counter1 Control Register – TCCR1B.

**Figure 55.** ICP Pin Schematic Diagram




ICPE: Input Capture Pin Enable

If the noise canceler function is enabled, the actual trigger condition for the capture event is monitored over four samples, and all four must be equal to activate the capture flag.

- **Bits 2,1,0 - CS12, CS11, CS10: Clock Select1, Bits 2, 1 and 0**

The Clock Select1 bits 2,1 and 0 define the prescaling source of Timer/Counter1.

**Table 29.** Clock 1 Prescale Select

| CS12 | CS11 | CS10 | Description |
|---|---|---|---|
| 0 | 0 | 0 | Stop, the Timer/Counter1 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 PE4 (T1), falling edge |
| 1 | 1 | 1 | External pin PE4 (T1), rising edge |

The Stop condition provides a Timer Enable/Disable function. The CK down-divided modes are scaled directly from the CK oscillator clock. If the external pin modes are used for Timer/Counter1, transitions on PE4/(T1) 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/Counter1 Register – TCNT1H AND TCNT1L***

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | |
|---|---|---|---|---|---|---|---|---|---|
| $2D ($4D) | MSB | | | | | | | | TCNT1H |
| $2C ($4C) | | | | | | | | LSB | TCNT1L |
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| | 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 | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

This 16-bit register contains the prescaled value of the 16-bit Timer/Counter1. To ensure that both the High and low bytes are read and written simultaneously when the CPU accesses these registers, the access is performed using an 8-bit temporary register (TEMP). This temporary register is also used when accessing OCR1A, OCR1B and ICR1. If the main program and also interrupt routines perform access to registers using TEMP, interrupts must be disabled during access from the main program and interrupt routines.

***TCNT1 Timer/Counter1 Write***

When the CPU writes to the high byte TCNT1H, the written data is placed in the TEMP register. Next, when the CPU writes the low byte TCNT1L, this byte of data is combined with the byte data in the TEMP register, and all 16 bits are written to the TCNT1 Timer/Counter1 register simultaneously. Consequently, the high byte TCNT1H must be accessed first for a full 16-bit register write operation.

**Table 36.** UBR Settings at Various Crystal Frequencies

| Clock MHz | UBRRHI 7:4 or 3:0 | UBRRn | UBR HEX | UBR | Actual Freq | Desired Freq. | % Error |
|---|---|---|---|---|---|---|---|
| 1 | **0000** | **00011001** | **019** | **25** | **2404** | **2400** | **0.2** |
| | **0000** | **00001100** | **00C** | **12** | **4808** | **4800** | **0.2** |
| | 0000 | 00000110 | 006 | 6 | 8929 | 9600 | 7.5 |
| | 0000 | 00000011 | 003 | 3 | 15625 | 14400 | 7.8 |
| | 0000 | 00000010 | 002 | 2 | 20833 | 19200 | 7.8 |
| | 0000 | 00000001 | 001 | 1 | 31250 | 28880 | 7.6 |
| | 0000 | 00000001 | 001 | 1 | 31250 | 38400 | 22.9 |
| | 0000 | 00000000 | 000 | 0 | 62500 | 57600 | 7.8 |
| | 0000 | 00000000 | 000 | 0 | 62500 | 76800 | 22.9 |
| | 0000 | 00000000 | 000 | 0 | 62500 | 115200 | 84.3 |

| Clock MHz | UBRRHI 7:4 or 3:0 | UBRRn | UBR HEX | UBR | Actual Freq | Desired Freq. | % Error |
|---|---|---|---|---|---|---|---|
| 1.8432 | **0000** | **00101111** | **02F** | **47** | **2400** | **2400** | **0.0** |
| | **0000** | **00010111** | **017** | **23** | **4800** | **4800** | **0.0** |
| | **0000** | **00001011** | **00B** | **11** | **9600** | **9600** | **0.0** |
| | **0000** | **00000111** | **007** | **7** | **14400** | **14400** | **0.0** |
| | **0000** | **00000101** | **005** | **5** | **19200** | **19200** | **0.0** |
| | **0000** | **00000011** | **003** | **3** | **28800** | **28880** | **0.3** |
| | **0000** | **00000010** | **002** | **2** | **38400** | **38400** | **0.0** |
| | **0000** | **00000001** | **001** | **1** | **57600** | **57600** | **0.0** |
| | 0000 | 00000001 | 001 | 1 | 57600 | 76800 | 33.3 |
| | **0000** | **00000000** | **000** | **0** | **115200** | **115200** | **0.0** |

| Clock MHz | UBRRHI 7:4 or 3:0 | UBRRn | UBR HEX | UBR | Actual Freq | Desired Freq. | % Error |
|---|---|---|---|---|---|---|---|
| 9.216 | **0000** | **11101111** | **0EF** | **239** | **2400** | **2400** | **0.0** |
| | **0000** | **01110111** | **077** | **119** | **4800** | **4800** | **0.0** |
| | **0000** | **00111011** | **03B** | **59** | **9600** | **9600** | **0.0** |
| | **0000** | **00100111** | **027** | **39** | **14400** | **14400** | **0.0** |
| | **0000** | **00011101** | **01D** | **29** | **19200** | **19200** | **0.0** |
| | **0000** | **00010011** | **013** | **19** | **28800** | **28880** | **0.3** |
| | **0000** | **00001110** | **00E** | **14** | **38400** | **38400** | **0.0** |
| | **0000** | **00001001** | **009** | **9** | **57600** | **57600** | **0.0** |
| | 0000 | 00000111 | 007 | 7 | 72000 | 76800 | 6.7 |
| | **0000** | **00000100** | **004** | **4** | **115200** | **115200** | **0.0** |
| | 0000 | 00000001 | 001 | 1 | 288000 | 230400 | 20.0 |
| | 0000 | 00000000 | 000 | 0 | 576000 | 460800 | 20.0 |
| | 0000 | 00000000 | 000 | 0 | 576000 | 912600 | 58.4 |

| Clock MHz | UBRRHI 7:4 or 3:0 | UBRRn | UBR HEX | UBR | Actual Freq | Desired Freq. | % Error |
|---|---|---|---|---|---|---|---|
| 18.432 | **0001** | **11011111** | **1DF** | **479** | **2400** | **2400** | **0.0** |
| | **0000** | **11101111** | **0EF** | **239** | **4800** | **4800** | **0.0** |
| | **0000** | **01110111** | **077** | **119** | **9600** | **9600** | **0.0** |
| | **0000** | **01001111** | **04F** | **79** | **14400** | **14400** | **0.0** |
| | **0000** | **00111011** | **03B** | **59** | **19200** | **19200** | **0.0** |
| | **0000** | **00100111** | **027** | **39** | **28800** | **28880** | **0.3** |
| | **0000** | **00011101** | **01D** | **29** | **38400** | **38400** | **0.0** |
| | **0000** | **00010011** | **013** | **19** | **57600** | **57600** | **0.0** |
| | **0000** | **00001110** | **00E** | **14** | **76800** | **76800** | **0.0** |
| | **0000** | **00001001** | **009** | **9** | **115200** | **115200** | **0.0** |
| | **0000** | **00000100** | **004** | **4** | **230400** | **230400** | **0.0** |
| | 0000 | 00000001 | 001 | 1 | 576000 | 460800 | 20.0 |
| | 0000 | 00000000 | 000 | 0 | 1152000 | 912600 | 20.8 |

| Clock MHz | UBRRHI 7:4 or 3:0 | UBRRn | UBR HEX | UBR | Actual Freq | Desired Freq. | % Error |
|---|---|---|---|---|---|---|---|
| 25.576 | **0010** | **10011001** | **299** | **665** | **2400** | **2400** | **0.0** |
| | **0001** | **01001100** | **14C** | **332** | **4800** | **4800** | **0.0** |
| | **0000** | **10100110** | **0A6** | **166** | **9572** | **9600** | **0.3** |
| | **0000** | **01101110** | **06E** | **110** | **14401** | **14400** | **0.0** |
| | **0000** | **01010010** | **052** | **82** | **19259** | **19200** | **0.3** |
| | **0000** | **00110110** | **036** | **54** | **29064** | **28880** | **0.6** |
| | **0000** | **00101001** | **029** | **41** | **38060** | **38400** | **0.9** |
| | **0000** | **00011011** | **01B** | **27** | **57089** | **57600** | **0.9** |
| | **0000** | **00010100** | **014** | **20** | **76119** | **76800** | **0.9** |
| | **0000** | **00001101** | **00D** | **13** | **114179** | **115200** | **0.9** |
| | **0000** | **00000110** | **006** | **6** | **228357** | **230400** | **0.9** |
| | 0000 | 00000011 | 003 | 3 | 399625 | 460800 | 15.3 |
| | 0000 | 00000001 | 001 | 1 | 799250 | 912600 | 14.2 |

| Clock MHz | UBRRHI 7:4 or 3:0 | UBRRn | UBR HEX | UBR | Actual Freq | Desired Freq. | % Error |
|---|---|---|---|---|---|---|---|
| 40 | **0100** | **00010001** | **411** | **1041** | **2399** | **2400** | **0.0** |
| | **0010** | **00001000** | **208** | **520** | **4798** | **4800** | **0.0** |
| | **0001** | **00000011** | **103** | **259** | **9615** | **9600** | **0.2** |
| | **0000** | **10101100** | **0AC** | **172** | **14451** | **14400** | **0.4** |
| | **0000** | **10000001** | **081** | **129** | **19231** | **19200** | **0.2** |
| | **0000** | **01010110** | **056** | **86** | **28736** | **28880** | **0.5** |
| | **0000** | **01000000** | **040** | **64** | **38462** | **38400** | **0.2** |
| | **0000** | **00101010** | **02A** | **42** | **58140** | **57600** | **0.9** |
| | 0000 | 00100000 | 020 | 32 | 75758 | 76800 | 1.4 |
| | 0000 | 00010101 | 015 | 21 | 113636 | 115200 | 1.4 |
| | 0000 | 00001010 | 00A | 10 | 227273 | 230400 | 1.4 |
| | 0000 | 00000100 | 004 | 4 | 500000 | 460800 | 7.8 |
| | 0000 | 00000010 | 002 | 2 | 833333 | 912600 | 9.5 |

### *UART0 and UART1 High Byte Baud-rate Register UBRRHI*

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| $20 ($40) | **MSB1** | | | **LSB1** | **MSB0** | | | **LSB0** | **UBRRHI** |
| 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 UART baud register is a 12-bit register. The 4 most significant bits are located in a separate register, UBRRHI. Note that both UART0 and UART1 share this register. Bit 7 to bit 4 of UBRRHI contain the 4 most significant bits of the UART1 baud register. Bit 3 to bit 0 contain the 4 most significant bits of the UART0 baud register.

**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 42.** Status Codes for Master 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 | |
| $08 | A START condition has been transmitted | Load SLA+R | X | 0 | 1 | X | SLA+R will be transmitted<br>ACK or NOT ACK will be received |
| $10 | A repeated START condition has been transmitted | Load SLA+R or | X | 0 | 1 | X | SLA+R will be transmitted<br>ACK or NOT ACK will be received |
| | | Load SLA+W | X | 0 | 1 | X | SLA+W will be transmitted<br>Logic will switch to Master Transmitter mode |
| $38 | Arbitration lost in SLA+R or NOT ACK bit | No TWDR action or | 0 | 0 | 1 | X | 2-wire serial bus will be released and not addressed Slave mode will be entered |
| | | No TWDR action | 1 | 0 | 1 | X | A START condition will be transmitted when the bus becomes free |
| $40 | SLA+R has been transmitted;<br>ACK has been received | No TWDR action or | 0 | 0 | 1 | 0 | Data byte will be received and NOT ACK will be returned |
| | | No TWDR action | 0 | 0 | 1 | 1 | Data byte will be received and ACK will be returned |
| $48 | SLA+R has been transmitted;<br>NOT ACK has been received | No TWDR action or | 1 | 0 | 1 | X | Repeated START will be transmitted |
| | | No TWDR action or | 0 | 1 | 1 | X | STOP condition will be transmitted and TWSTO flag will be reset |
| | | No TWDR action | 1 | 1 | 1 | X | STOP condition followed by a START condition will be transmitted and TWSTO flag will be reset |
| $50 | Data byte has been received;<br>ACK has been returned | Read data byte or | 0 | 0 | 1 | 0 | Data byte will be received and NOT ACK will be returned |
| | | Read data byte | 0 | 0 | 1 | 1 | Data byte will be received and ACK will be returned |
| $58 | Data byte has been received;<br>NOT ACK has been returned | Read data byte or | 1 | 0 | 1 | X | Repeated START will be transmitted |
| | | Read data byte or | 0 | 1 | 1 | X | STOP condition will be transmitted and TWSTO flag will be reset |
| | | Read data byte | 1 | 1 | 1 | X | STOP condition followed by a START condition will be transmitted and TWSTO flag will be reset |

## 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
Maximum delays are the average of $t_{PDLH}$ and $t_{PDHL}$.

Clocks and Reset Input buffers are measured from a $V_{IH}$ of 1.5V at the input pad to the internal $V_{IH}$ of 50% of $V_{CC}$.
Maximum times for clock input buffers and internal drivers are measured for rising edge delays only.

| Cell Function | Parameter | Path | Device | -25 | Units | Notes |
|---|---|---|---|---|---|---|
| **Global Clocks and Set/Reset** | | | | | | |
| GCK Input Buffer | $t_{PD}$ (Maximum) | pad -> clock<br>pad -> clock | AT94K05<br>AT94K10<br>AT94K40 | 1.2<br>1.5<br>1.9 | ns<br>ns | Rising Edge Clock |
| FCK Input Buffer | $t_{PD}$ (Maximum) | pad -> clock<br>pad -> clock | AT94K05<br>AT94K10<br>AT94K40 | 0.7<br>0.8<br>0.9 | ns<br>ns | Rising Edge Clock |
| Clock Column Driver | $t_{PD}$ (Maximum) | clock -> colclk<br>clock -> colclk | AT94K05<br>AT94K10<br>AT94K40 | 1.3<br>1.8<br>2.5 | ns<br>ns | Rising Edge Clock |
| Clock Sector Driver | $t_{PD}$ (Maximum) | colclk -> secclk<br>colclk -> secclk | AT94K05<br>AT94K10<br>AT94K40 | 1.0<br>1.0<br>1.0 | ns<br>ns | Rising Edge Clock |
| GSRN Input Buffer | $t_{PD}$ (Maximum) | colclk -> secclk<br>colclk -> secclk | AT94K05<br>AT94K10<br>AT94K40 | 5.4<br>8.2 | ns<br>ns | – |
| Global Clock to Output | $t_{PD}$ (Maximum) | clock pad -> out<br>clock pad -> out | AT94K05<br>AT94K10<br>AT94K40 | 12.6<br>13.4<br>14.5 | ns<br>ns | Rising Edge Clock<br>Fully Loaded Clock Tree<br>Rising Edge DFF<br>20 mA Output Buffer<br>50 pf Pin Load |
| Fast Clock to Output | $t_{PD}$ (Maximum) | clock pad -> out<br>clock pad -> out | AT94K05<br>AT94K10<br>AT94K40 | 12.1<br>12.7<br>13.5 | ns<br>ns | Rising Edge Clock<br>Fully Loaded Clock Tree<br>Rising Edge DFF<br>20 mA Output Buffer<br>50 pf Pin Load |

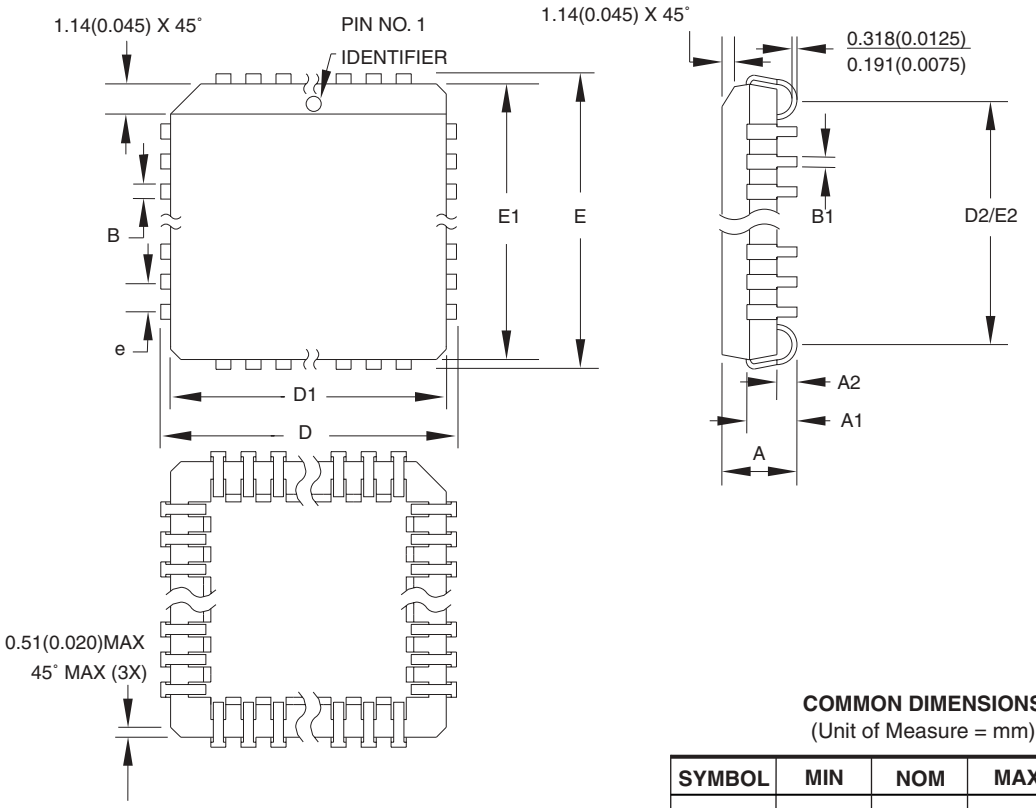
**Table 56.** AT94K Pin List (Continued)

| 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.

## Packaging Information

### 84J – PLCC




**COMMON DIMENSIONS**
(Unit of Measure = mm)

| SYMBOL | MIN | NOM | MAX | NOTE |
|---|---|---|---|---|
| A | 4.191 | – | 4.572 | |
| A1 | 2.286 | – | 3.048 | |
| A2 | 0.508 | – | – | |
| D | 30.099 | – | 30.353 | |
| D1 | 29.210 | – | 29.413 | Note 2 |
| E | 30.099 | – | 30.353 | |
| E1 | 29.210 | – | 29.413 | Note 2 |
| D2/E2 | 27.686 | – | 28.702 | |
| B | 0.660 | – | 0.813 | |
| B1 | 0.330 | – | 0.533 | |
| e | 1.270 TYP | | | |

Notes:
1. This package conforms to JEDEC reference MS-018, Variation AF.
2. Dimensions D1 and E1 do not include mold protrusion. Allowable protrusion is .010"(0.254 mm) per side. Dimension D1 and E1 include mold mismatch and are measured at the extreme material condition at the upper or lower parting line.
3. Lead coplanarity is 0.004" (0.102 mm) maximum.

10/04/01

| | TITLE | DRAWING NO. | REV. |
|---|---|---|---|
| 2325 Orchard Parkway<br>San Jose, CA 95131 | **84J**, 84-lead, Plastic J-leaded Chip Carrier (PLCC) | 84J | B |