Microchip Technology - AT94K05AL-25DQC Datasheet

Welcome to E-XFL.COM

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

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

What Are Embedded - FPGAs with Microcontrollers?

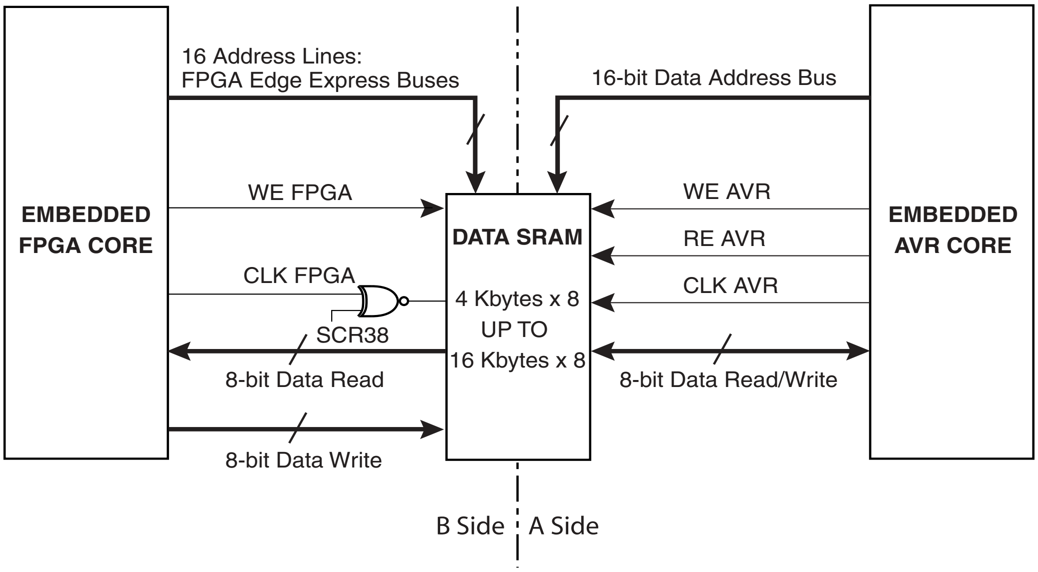
At their core, **FPGAs** are semiconductor devices that can

Details	
Product Status	Obsolete
Core Type	8-Bit AVR
Speed	25 MHz
Interface	I²C, UART
Program SRAM Bytes	4K-16K
FPGA SRAM	2kb
EEPROM Size	-
Data SRAM Bytes	4K ~ 16K
FPGA Core Cells	256
FPGA Gates	5K
FPGA Registers	436
Voltage - Supply	3V ~ 3.6V
Mounting Type	Surface Mount
Operating Temperature	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/at94k05al-25dqc

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

AVR Core and Peripherals

- AVR Core
- Watchdog Timer/On-chip Oscillator
- Oscillator-to-Internal Clock Circuit
- Oscillator-to-Timer/Counter for Real-time Clock
- 16-bit Timer/Counter and Two 8-bit Timer/Counters
- Interrupt Unit
- Multiplier
- UART (0)
- UART (1)
- I/O Port D (full 8 bits available on 144-pin or higher devices)
- I/O Port E

The embedded AVR core is a low-power CMOS 8-bit microcontroller based on the AVR RISC architecture. The embedded AVR core achieves throughputs approaching 1 MIPS per MHz by executing powerful instructions in a single-clock-cycle, and allows the system architect to optimize power consumption versus processing speed.

The AVR core is based on an enhanced RISC architecture that combines a rich instruction set with 32 x 8 general-purpose working registers. All the 32 x 8 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent register bytes to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The embedded AVR core provides the following features: 16 general-purpose I/O lines, 32 x 8 general-purpose working registers, Real-time Counter (RTC), 3 flexible timer/counters with compare modes and PWM, 2 UARTs, programmable Watchdog Timer with internal oscillator, 2-wire serial port, and three software-selectable Power-saving modes. The Idle mode stops the CPU while allowing the SRAM, timer/counters, two-wire serial port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the oscillator, disabling all other chip functions until the next interrupt or hardware reset. In Power-save mode, the timer oscillator continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping.

The embedded AVR core is supported with a full suite of program and system development tools, including C compilers, macro assemblers, program debugger/simulators and evaluation kits.

Instruction Set Summary (Continued)

Mnemonics	Operands	Description	Operation	Flags	#Clock
CBR	Rd, K	Clear Bit(s) in Register	Rd ← Rd • ($FFh - K)	Z,N,V,S	1
INC	Rd	Increment	Rd ← Rd + 1	Z,N,V,S	1
DEC	Rd	Decrement	Rd ← Rd - 1	Z,N,V,S	1
TST	Rd	Test for Zero or Minus	Rd ← Rd • Rd	Z,N,V,S	1
CLR	Rd	Clear Register	Rd ← Rd ⊕ Rd	Z,N,V,S	1
SER	Rd	Set Register	Rd ← $FF	None	1
MUL	Rd, Rr	Multiply Unsigned	R1:R0 ← Rd × Rr (UU)	Z,C	2
MULS	Rd, Rr	Multiply Signed	R1:R0 ← Rd × Rr (SS)	Z,C	2
MULSU	Rd, Rr	Multiply Signed with Unsigned	R1:R0 ← Rd × Rr (SU)	Z,C	2
FMUL	Rd, Rr	Fractional Multiply Unsigned	R1:R0 ← (Rd × Rr)<<1 (UU)	Z,C	2
FMULS	Rd, Rr	Fractional Multiply Signed	R1:R0 ← (Rd × Rr)<<1 (SS)	Z,C	2
FMULSU	Rd, Rr	Fractional Multiply Signed with Unsigned	R1:R0 ← (Rd × Rr)<<1 (SU)	Z,C	2
		Branch Instructions			
RJMP	k	Relative Jump	PC ← PC + k + 1	None	2
IJMP		Indirect Jump to (Z)	PC(15:0) ← Z	None	2
JMP	k	Jump	PC ← k	None	3
RCALL	k	Relative Call Subroutine	PC ← PC + k + 1	None	3
ICALL		Indirect Call to (Z)	PC(15:0) ← Z	None	3
CALL	k	Call Subroutine	PC ← k	None	4
RET		Subroutine Return	PC ← STACK	None	4
RETI		Interrupt Return	PC ← STACK	I	4
CPSE	Rd, Rr	Compare, Skip if Equal	if (Rd = Rr) PC ← PC + 2 or 3	None	1 / 2 / 3
CP	Rd, Rr	Compare	Rd - Rr	Z,C,N,V,S,H	1
CPC	Rd, Rr	Compare with Carry	Rd - Rr - C	Z,C,N,V,S,H	1
CPI	Rd, K	Compare with Immediate	Rd - K	Z,C,N,V,S,H	1
SBRC	Rr, b	Skip if Bit in Register Cleared	if (Rr(b) = 0) PC ← PC + 2 or 3	None	1 / 2 / 3
SBRS	Rr, b	Skip if Bit in Register Set	if (Rr(b) = 1) PC ← PC + 2 or 3	None	1 / 2 / 3
SBIC	A, b	Skip if Bit in I/O Register Cleared	if(I/O(A,b) = 0) PC ← PC + 2 or 3	None	1 / 2 / 3
SBIS	A, b	Skip if Bit in I/O Register Set	If(I/O(A,b) = 1) PC ← PC + 2 or 3	None	1 / 2 / 3
BRBS	s, k	Branch if Status Flag Set	if (SREG(s) = 1) then PC ←PC+k+1	None	1 / 2
BRBC	s, k	Branch if Status Flag Cleared	if (SREG(s) = 0) then PC ←PC+k+1	None	1 / 2
BREQ	k	Branch if Equal	if (Z = 1) then PC ← PC + k + 1	None	1 / 2
BRNE	k	Branch if Not Equal	if (Z = 0) then PC ← PC + k + 1	None	1 / 2
BRCS	k	Branch if Carry Set	if (C = 1) then PC ← PC + k + 1	None	1 / 2
BRCC	k	Branch if Carry Cleared	if (C = 0) then PC ← PC + k + 1	None	1 / 2

X-register, Y-register and Z-register

Registers R26..R31 have some added functions to their general-purpose usage. These registers are address pointers for indirect addressing of the SRAM. The three indirect address registers X, Y and Z have functions as fixed displacement, automatic increment and decrement (see the descriptions for the different instructions).

ALU – Arithmetic Logic Unit

The high-performance AVR ALU operates in direct connection with all the 32 general-purpose working registers. Within a single clock cycle, ALU operations between registers in the register file are executed. The ALU operations are divided into three main categories – arithmetic, logical and bit-functions.

Multiplier Unit

The high-performance AVR Multiplier operates in direct connection with all the 32 general-purpose working registers. This unit performs 8 x 8 multipliers every two clock cycles. See multiplier details on page 106.

SRAM Data Memory

External data SRAM (or program) cannot be used with the FPSLIC AT94K family.

The five different addressing modes for the data memory cover: Direct, Indirect with Displacement, Indirect, Indirect with Pre-decrement and Indirect with Post-increment. In the register file, registers R26 to R31 feature the indirect addressing pointer registers.

The Indirect with Displacement mode features a 63 address locations reach from the base address given by the Y- or Z-register.

When using register indirect addressing modes with automatic Pre-decrement and Post-increment, the address registers X, Y and Z are decremented and incremented.

The entire data address space including the 32 general-purpose working registers and the 64 I/O registers are all accessible through all these addressing modes. See the next section for a detailed description of the different addressing modes.

Program and Data Addressing Modes

The embedded AVR core supports powerful and efficient addressing modes for access to the program memory (SRAM) and data memory (SRAM, Register File and I/O Memory). This section describes the different addressing modes supported by the AVR architecture.

Register Direct, Single-register Rd

The operand is contained in register d (Rd).

Register Direct, Two Registers Rd and Rr

Operands are contained in register r (Rr) and d (Rd). The result is stored in register d (Rd).

I/O Direct

Operand address is contained in 6 bits of the instruction word. *n* is the destination or source register address.

Data Direct

A 16-bit data address is contained in the 16 LSBs of a two-word instruction. Rd/Rr specify the destination or source register.

Data Indirect with Displacement

Operand address is the result of the Y- or Z-register contents added to the address contained in 6 bits of the instruction word.

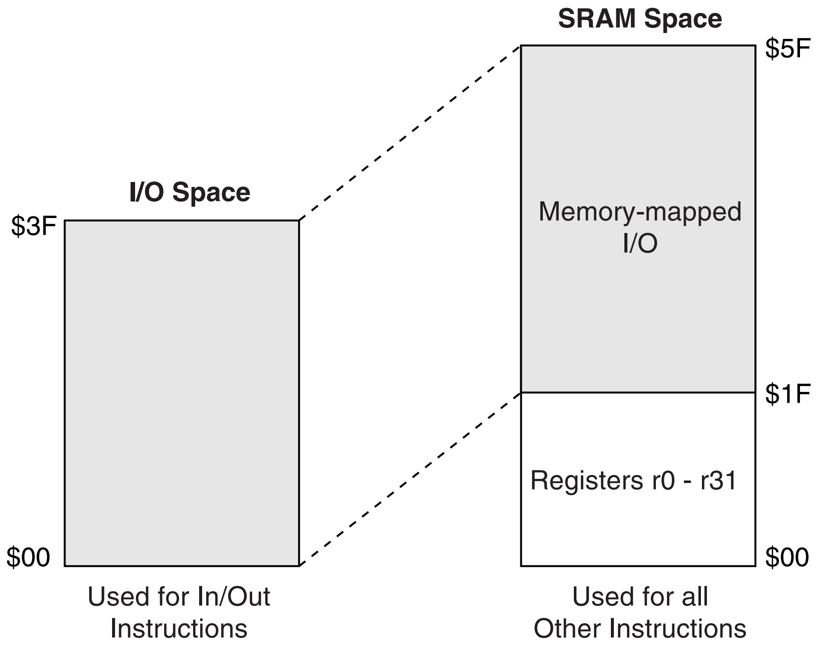
Memory-mapped I/O

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

AT94K Register Summary

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

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.

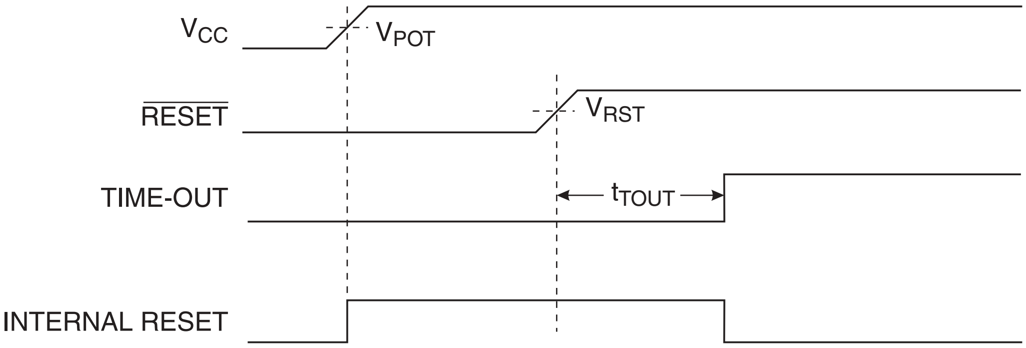
For compatibility with future devices, reserved bits should be written zero if accessed. Reserved I/O memory addresses should never be written.

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.

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.

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.

The mechanisms for reading TCNT2, OCR2 and TCCR2 are different. When reading TCNT2, the actual timer value is read. When reading OCR2 or TCCR2, the value in the temporary storage register is read.

Asynchronous Operation of Timer/Counter2

When Timer/Counter2 operates asynchronously, some considerations must be taken:

- When switching between asynchronous and synchronous clocking of Timer/Counter2, the timer registers TCNT2, OCR2 and TCCR2 might get corrupted. A safe procedure for switching the clock source is:
  1. Disable the Timer/Counter2 interrupts by clearing OCIE2 and TOIE2.
  2. Select clock source by setting AS2 as appropriate.
  3. Write new values to TCNT2, OCR2 and TCCR2.
  4. To switch to asynchronous operation: Wait for TCN2UB, OCR2UB, and TCR2UB.
  5. Enable interrupts, if needed.
- The oscillator is optimized for use with a 32.768 kHz watch crystal. An external clock signal applied to this pin goes through the same amplifier having a bandwidth of 256 kHz. The external clock signal should therefore be in the interval 0 Hz – 1 MHz. The frequency of the clock signal applied to the TOSC1 pin must be lower than one fourth of the CPU main clock frequency.
- When writing to one of the registers TCNT2, OCR2, or TCCR2, the value is transferred to a temporary register, and latched after two positive edges on TOSC1. The user should not write a new value before the contents of the temporary register have been transferred to its destination. Each of the three mentioned registers have their individual temporary register, which means that, e.g., writing to TCNT2 does not disturb an OCR2 write in progress. To detect that a transfer to the destination register has taken place, an Asynchronous Status Register – ASSR has been implemented.
- When entering Power-save mode after having written to TCNT2, OCR2, or TCCR2, the user must wait until the written register has been updated if Timer/Counter2 is used to wake-up the device. Otherwise, the MCU will go to sleep before the changes have had any effect. This is extremely important if the Output Compare2 interrupt is used to wake-up the device; Output compare is disabled during write to OCR2 or TCNT2. If the write cycle is not finished (i.e., the MCU enters Sleep mode before the OCR2UB bit returns to zero), the device will never get a compare match and the MCU will not wake-up.
- If Timer/Counter2 is used to wake-up the device from Power-save mode, precautions must be taken if the user wants to re-enter Power-save mode: The interrupt logic needs one TOSC1 cycle to be reset. If the time between wake-up and reentering Power-save mode is less than one TOSC1 cycle, the interrupt will not occur and the device will fail to wake up. If the user is in doubt whether the time before re-entering power-save is sufficient, the following algorithm can be used to ensure that one TOSC1 cycle has elapsed:
  1. Write a value to TCCR2, TCNT2, or OCR2.
  2. Wait until the corresponding Update Busy flag in ASSR returns to zero.
  3. Enter Power-save mode.
- When asynchronous operation is selected, the 32.768 kHz oscillator for Timer/Counter2 is always running, except in Power-down mode. After a power-up reset or wake-up from power-down, the user should be aware of the fact that this oscillator might take as long as one second to stabilize. Therefore, the contents of all Timer2 registers must be considered lost after a wake-up from power-down, due to the unstable clock signal. The user is advised to wait for at least one second before using Timer/Counter2 after power-up or wake-up from power-down.
- Description of wake-up from Power-save mode when the timer is clocked asynchronously. When the interrupt condition is met, the wake-up process is started on the following cycle

Example 3 – Multiply-accumulate Operation

The final example of 8-bit multiplication shows a multiply-accumulate operation. The general formula can be written as:

$$c(n) = a(n) \times b + c(n-1)$$

```
                ; r17:r16 = r18 * r19 + r17:r16

in    r18,PINB  ; Get the current pin value on port B
ldi   r19,b     ; Load constant b into r19
muls  r19,r18   ; r1:r0 = variable A * variable B
add   r16,r0    ; r17:r16 += r1:r0
adc   r17,r1
```

Typical applications for the multiply-accumulate operation are FIR (Finite Impulse Response) and IIR (Infinite Impulse Response) filters, PID regulators and FFT (Fast Fourier Transform). For these applications the FMULS instruction is particularly useful. The main advantage of using the FMULS instruction instead of the MULS instruction is that the 16-bit result of the FMULS operation always may be approximated to a (well-defined) 8-bit format, see “Using Fractional Numbers” on page 111.

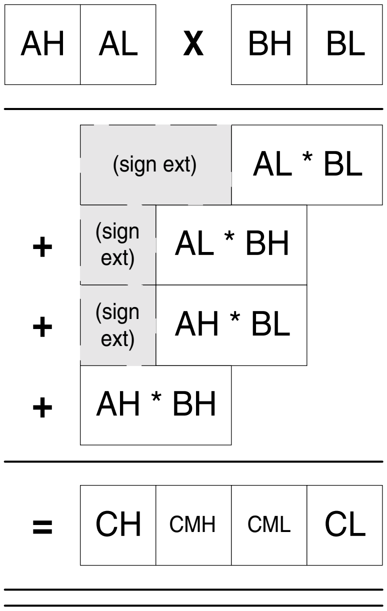
16-bit Multiplication

The new multiply instructions are specifically designed to improve 16-bit multiplication. This section presents solutions for using the hardware multiplier to do multiplication with 16-bit operands.

Figure 60 schematically illustrates the general algorithm for multiplying two 16-bit numbers with a 32-bit result (C = A • B). AH denotes the high byte and AL the low byte of the A operand. CMH denotes the middle high byte and CML the middle low byte of the result C. Equal notations are used for the remaining bytes.

The algorithm is basic for all multiplication. All of the partial 16-bit results are shifted and added together. The sign extension is necessary for signed numbers only, but note that the carry propagation must still be done for unsigned numbers.

Figure 60. 16-bit Multiplication, General Algorithm



```
   adc   r17, r1
   adc   r18, r2
   adc   r19, r2

   fmulsu   r23, r20          ; ( (signed)ah * bl ) << 1
   sbc   r19, r2
   add   r17, r0
   adc   r18, r1
   adc   r19, r2

   fmulsu   r21, r22          ; ( (signed)bh * al ) << 1
   sbc   r19, r2
   add   r17, r0
   adc   r18, r1
   adc   r19, r2

   ret

fmac16x16_32_method_B         ; uses two temporary registers (r4,r5), speed/Size
                              optimized
                              ; but reduces cycles/words by 2
   clr   r2

   fmuls r23, r21             ; ( (signed)ah * (signed)bh ) << 1
   movw  r5:r4,r1:r0
   fmul  r22, r20             ; ( al * bl ) << 1
   adc   r4, r2

add   r16, r0
   adc   r17, r1
   adc   r18, r4
   adc   r19, r5
fmulsu   r23, r20             ; ( (signed)ah * bl ) << 1
   sbc   r19, r2
   add   r17, r0
   adc   r18, r1
   adc   r19, r2
   fmulsu   r21, r22          ; ( (signed)bh * al ) << 1
   sbc   r19, r2
   add   r17, r0
   adc   r18, r1
   adc   r19, r2

   ret
```

Comment on Implementations

All 16-bit x 16-bit = 32-bit functions implemented here start by clearing the R2 register, which is just used as a “dummy” register with the “add with carry” (ADC) and “subtract with carry” (SBC) operations. These operations do not alter the contents of the R2 register. If the R2 register is not used elsewhere in the code, it is not necessary to clear the R2 register each time these functions are called, but only once prior to the first call to one of the functions.

Table 37. UBR Settings at Various Crystal Frequencies in Double UART Speed Mode

Clock	UBRRHI		UBR		Actual	Desired	%	Clock	UBRRHI		UBR		Actual	Desired	%
MHz	7:4 or 3:0	UBRRn	HEX	UBR	Freq	Freq.	Error	MHz	7:4 or 3:0	UBRRn	HEX	UBR	Freq	Freq.	Error
1	**0000**	**00110011**	**033**	**51**	**2404**	**2400**	**0.2**	1.843	**0000**	**01011111**	**05F**	**95**	**2400**	**2400**	**0.0**
	0000	**00011001**	**019**	**25**	**4808**	**4800**	**0.2**		**0000**	**00101111**	**02F**	**47**	**4800**	**4800**	**0.0**
	0000	**00001100**	**00C**	**12**	**9615**	**9600**	**0.2**		**0000**	**00010111**	**017**	**23**	**9600**	**9600**	**0.0**
	0000	00001000	008	8	13889	14400	3.7		**0000**	**00001111**	**00F**	**15**	**14400**	**14400**	**0.0**
	0000	00000110	006	6	17857	19200	7.5		**0000**	**00001011**	**00B**	**11**	**19200**	**19200**	**0.0**
	0000	00000011	003	3	31250	28880	7.6		**0000**	**00000111**	**007**	**7**	**28800**	**28880**	**0.3**
	0000	00000010	002	2	41667	38400	7.8		**0000**	**00000101**	**005**	**5**	**38400**	**38400**	**0.0**
	0000	00000001	001	1	62500	57600	7.8		**0000**	**00000011**	**003**	**3**	**57600**	**57600**	**0.0**
	0000	00000001	001	1	62500	76800	22.9		**0000**	**00000010**	**002**	**2**	**76800**	**76800**	**0.0**
	0000	00000000	000	0	125000	115200	7.8		**0000**	**00000001**	**001**	**1**	**115200**	**115200**	**0.0**
Clock	UBRRHI		UBR		Actual	Desired	%	Clock	UBRRHI		UBR		Actual	Desired	%
MHz	7:4 or 3:0	UBRRn	HEX	UBR	Freq	Freq.	Error	MHz	7:4 or 3:0	UBRRn	HEX	UBR	Freq	Freq.	Error
9.216	**0001**	**11011111**	**1DF**	**479**	**2400**	**2400**	**0.0**	18.43	**0011**	**10111111**	**3BF**	**959**	**2400**	**2400**	**0.0**
	0000	**11101111**	**0EF**	**239**	**4800**	**4800**	**0.0**		**0001**	**11011111**	**1DF**	**479**	**4800**	**4800**	**0.0**
	0000	**01110111**	**077**	**119**	**9600**	**9600**	**0.0**		**0000**	**11101111**	**0EF**	**239**	**9600**	**9600**	**0.0**
	0000	**01001111**	**04F**	**79**	**14400**	**14400**	**0.0**		**0000**	**10011111**	**09F**	**159**	**14400**	**14400**	**0.0**
	0000	**00111011**	**03B**	**59**	**19200**	**19200**	**0.0**		**0000**	**01110111**	**077**	**119**	**19200**	**19200**	**0.0**
	0000	**00100111**	**027**	**39**	**28800**	**28880**	**0.3**		**0000**	**01001111**	**04F**	**79**	**28800**	**28880**	**0.3**
	0000	**00011101**	**01D**	**29**	**38400**	**38400**	**0.0**		**0000**	**00111011**	**03B**	**59**	**38400**	**38400**	**0.0**
	0000	**00010011**	**013**	**19**	**57600**	**57600**	**0.0**		**0000**	**00100111**	**027**	**39**	**57600**	**57600**	**0.0**
	0000	**00001110**	**00E**	**14**	**76800**	**76800**	**0.0**		**0000**	**00011101**	**01D**	**29**	**76800**	**76800**	**0.0**
	0000	**00001001**	**009**	**9**	**115200**	**115200**	**0.0**		**0000**	**00010011**	**013**	**19**	**115200**	**115200**	**0.0**
	0000	**00000100**	**004**	**4**	**230400**	**230400**	**0.0**		**0000**	**00001001**	**009**	**9**	**230400**	**230400**	**0.0**
	0000	00000010	002	2	384000	460800	20.0		**0000**	**00000100**	**004**	**4**	**460800**	**460800**	**0.0**
	0000	00000000	000	0	1152000	912600	20.8		0000	00000010	002	2	**768000**	912600	18.8
Clock	UBRRHI		UBR		Actual	Desired	%	Clock	UBRRHI		UBR		Actual	Desired	%
MHz	7:4 or 3:0	UBRRn	HEX	UBR	Freq	Freq.	Error	MHz	7:4 or 3:0	UBRRn	HEX	UBR	Freq	Freq.	Error
25.576	**0101**	**00110011**	**533**	**1331**	**2400**	**2400**	**0.0**	40	**1000**	**00100010**	**822**	**2082**	**2400**	**2400**	**0.0**
	0010	**10011001**	**299**	**665**	**4800**	**4800**	**0.0**		**0100**	**00010001**	**411**	**1041**	**4798**	**4800**	**0.0**
	0001	**01001110**	**14E**	**334**	**9543**	**9600**	**0.6**		**0010**	**00001000**	**208**	**520**	**9597**	**9600**	**0.0**
	0000	**11011101**	**0DD**	**221**	**14401**	**14400**	**0.0**		**0001**	**01011010**	**15A**	**346**	**14409**	**14400**	**0.1**
	0000	**10100110**	**0A6**	**166**	**19144**	**19200**	**0.3**		**0001**	**00000011**	**103**	**259**	**19231**	**19200**	**0.2**
	0000	**01101110**	**06E**	**110**	**28802**	**28880**	**0.3**		**0000**	**10101100**	**0AC**	**172**	**28902**	**28880**	**0.1**
	0000	**01010010**	**052**	**82**	**38518**	**38400**	**0.3**		**0000**	**10000001**	**081**	**129**	**38462**	**38400**	**0.2**
	0000	00110111	037	55	57089	57600	0.9		**0000**	**01010110**	**056**	**86**	**57471**	**57600**	**0.2**
	0000	00101001	029	41	76119	76800	0.9		**0000**	**01000000**	**040**	**64**	**76923**	**76800**	**0.2**
	0000	00011011	01B	27	114179	115200	0.9		0000	00101010	02A	42	116279	115200	0.9
	0000	00001101	00D	13	228357	230400	0.9		0000	00010101	015	21	227273	230400	1.4
	0000	00000110	006	6	456714	460800	0.9		0000	00001010	00A	10	454545	460800	1.4
	0000	00000011	003	3	799250	912600	14.2		0000	00000100	004	4	1000000	912600	8.7

Table 41. Status Codes for Master Transmitter Mode

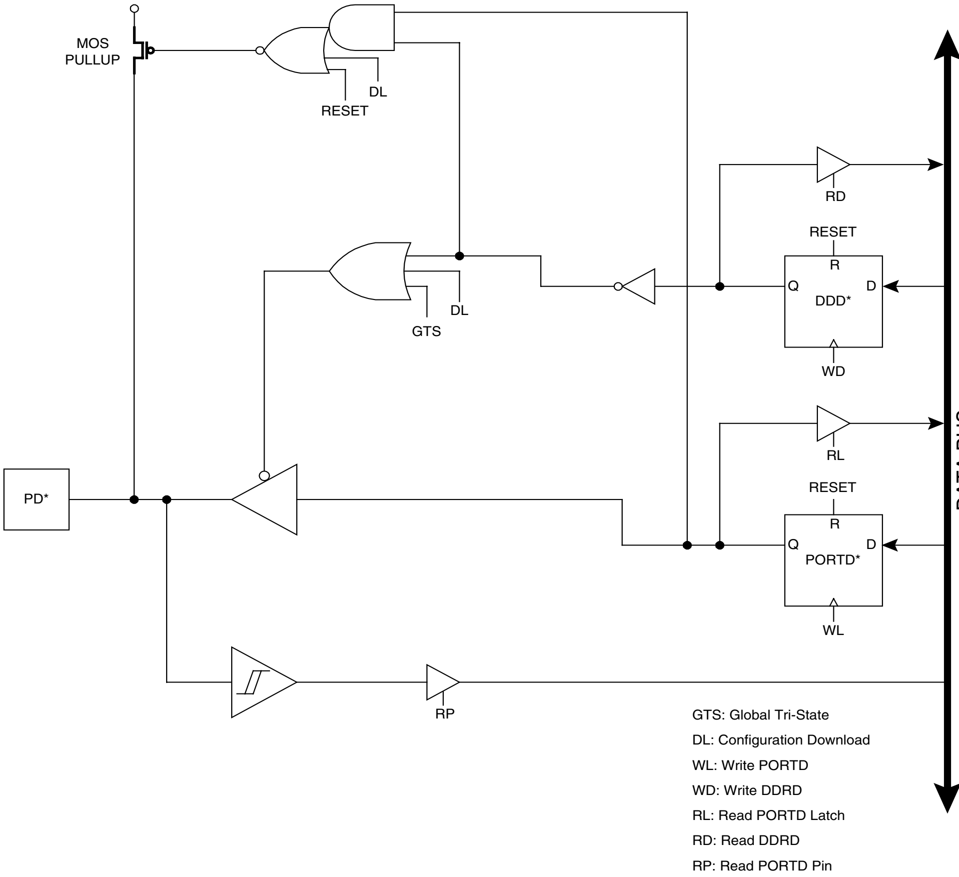
Status Code (TWSR)	Status of the 2-wire Serial Bus and 2-wire Serial Hardware	Application Software Response					Next Action Taken by 2-wire Serial Hardware
			To TWCR				
		To/From TWDR	STA	STO	TWINT	TWEA	
$08	A START condition has been transmitted	Load SLA+W	X	0	1	X	SLA+W will be transmitted; ACK or NOT ACK will be received
$10	A repeated START condition has been transmitted	Load SLA+W or	X	0	1	X	SLA+W will be transmitted; ACK or NOT ACK will be received
		Load SLA+R	X	0	1	X	SLA+R will be transmitted; Logic will switch to Master Receiver mode
$18	SLA+W has been transmitted; ACK has been received	Load data byte or	0	0	1	X	Data byte will be transmitted and ACK or NOT ACK will be 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
$20	SLA+W has been transmitted; NOT ACK has been received	Load data byte or	0	0	1	X	Data byte will be transmitted and ACK or NOT ACK will be 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
$28	Data byte has been transmitted; ACK has been received	Load data byte or	0	0	1	X	Data byte will be transmitted and ACK or NOT ACK will be 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
$30	Data byte has been transmitted; NOT ACK has been received	Load data byte or	0	0	1	X	Data byte will be transmitted and ACK or NOT ACK will be 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
$38	Arbitration lost in SLA+W or data bytes	No TWDR action or	0	0	1	X	2-wire serial bus will be released and not addressed Slave mode entered
		No TWDR action	1	0	1	X	A START condition will be transmitted when the bus becomes free

Table 46. DDDn[1] Bits on PortD Pins

DDDn[1]	PORTDn[1]	I/O	Pull-up	Comment
0	0	Input	No	Tri-state (High-Z)
0	1	Input	Yes	PDn will source current if external pulled low (default)
1	0	Output	No	Push-pull zero output
1	1	Output	No	Push-pull one output

Note: 1. n: 7,6...0, pin number

Figure 75. PortD Schematic Diagram



PortE

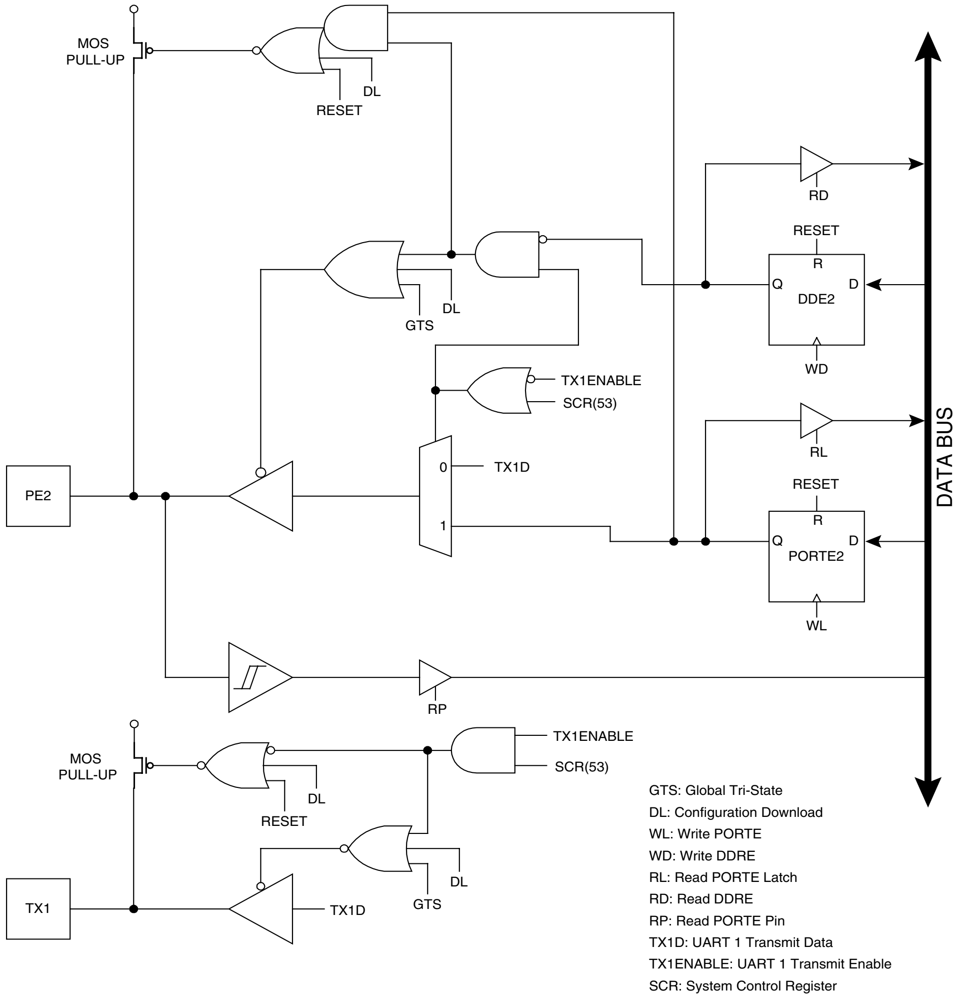
PortE is an 8-bit bi-directional I/O port with internal pull-up resistors.

Three I/O memory address locations are allocated for the PortE, one each for the Data Register – PORTE, $07($27), Data Direction Register – DDRE, $06($26) and the PortE Input Pins – PINE, $05($25). The PortE Input Pins address is read only, while the Data Register and the Data Direction Register are read/write.

The PortE output buffers can sink 20 mA. As inputs, PortE pins that are externally pulled Low will source current if the pull-up resistors are activated.

All PortE pins have alternate functions as shown in Table 47.

Figure 78. PortE Schematic Diagram (Pin PE2)



GTS: Global Tri-State
DL: Configuration Download
WL: Write PORTE
WD: Write DDRE
RL: Read PORTE Latch
RD: Read DDRE
RP: Read PORTE Pin
TX1D: UART 1 Transmit Data
TX1ENABLE: UART 1 Transmit Enable
SCR: System Control Register

AC & DC Timing Characteristics

Absolute Maximum Ratings*(1)

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage(2) on Any Pin with Respect to Ground	-0.5V to +5.0V
Supply Voltage (V_{CC})	-0.5V to +5.0V
Maximum Soldering Temp. (10 sec. @ 1/16 in.)	250°C
ESD (R_{ZAP} = 1.5K, C_{ZAP} = 100 pF)	2000V

*NOTICE: Stresses beyond those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those listed under operating conditions is not implied. Exposure to Absolute Maximum Rating conditions for extended periods of time may affect device reliability.

Notes: 1. For AL parts only

2. Minimum voltage of -0.5V DC which may undershoot to -2.0V for pulses of less than 20 ns.

DC and AC Operating Range – 3.3V Operation

		AT94K Commercial	AT94K Industrial
Operating Temperature (Case)		0°C - 70°C	-40°C - 85°C
V_{CC} Power Supply		3.3V ± 0.3V	3.3V ± 0.3V
Input Voltage Level (CMOS)	High (V_{IHC})	70% - 100% V_{CC}	70% - 100% V_{CC}
	Low (V_{ILC})	0 - 30% V_{CC}	0 - 30% V_{CC}

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	

Table 56. AT94K Pin List (Continued)

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

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

Table 56. AT94K Pin List (Continued)

AT94K05 96 FPGA I/O	AT94K10 192 FPGA I/O	AT94K40 384 FPGA I/O	Packages			
			PC84	TQ100	PQ144	PQ208
		I/O113				
		I/O114				
		GND				
		I/O115				
		I/O116				
	I/O59	I/O117				
	I/O60	I/O118				
		I/O119				
		I/O120				
GND	GND	GND			45	67
I/O41	I/O61	I/O121			46	68
I/O42	I/O62	I/O122			47	69
I/O43/TMS	I/O63/TMS	I/O123/TMS	38	31	48	70
I/O44/TCK	I/O64/TCK	I/O124/TCK	39	32	49	71
	VCC[1]	VCC[1]				
	I/O65	I/O125				72
	I/O66	I/O126				73
		GND				
		I/O127				
		I/O128				
		I/O129				
		I/O130				
		I/O131				
		I/O132				
		GND				
		VCC[1]				
		I/O133				
		I/O134				
	I/O67	I/O135				
	I/O68	I/O136				
I/O45	I/O69	I/O137		33	50	74
I/O46	I/O70	I/O138		34	51	75
		GND				
		I/O139				

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.