Microchip Technology - AT94K40AL-25BQI Datasheet

Welcome to E-XFL.COM

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

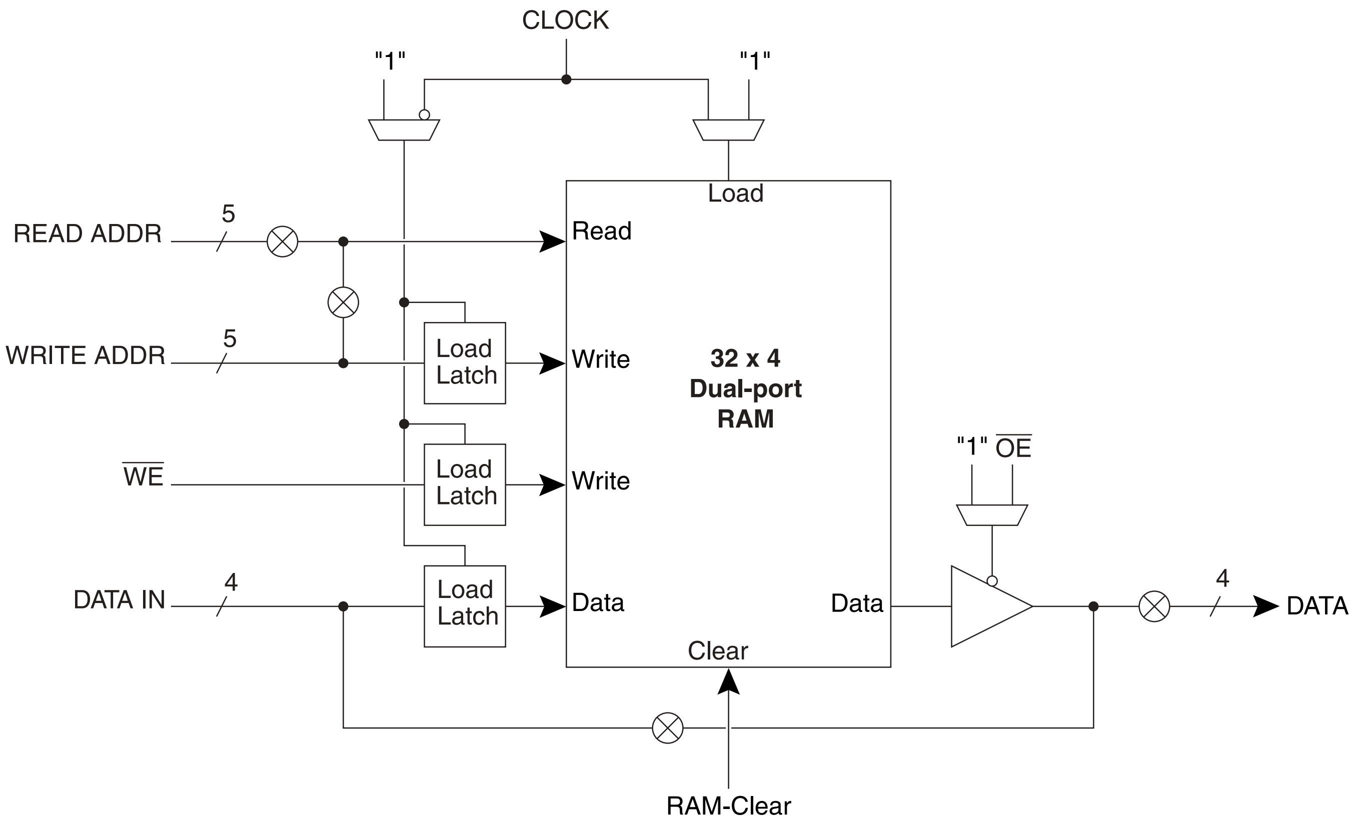
Embedded - FPGAs (Field Programmable Gate Arrays) with Microcontrollers represent a cutting-edge category of electronic components that combine the flexibility of FPGA technology with the processing power of integrated microcontrollers. This hybrid approach offers a versatile solution for designing and 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	18 MHz
Interface	I²C, UART
Program SRAM Bytes	20K-32K
FPGA SRAM	18kb
EEPROM Size	-
Data SRAM Bytes	4K ~ 16K
FPGA Core Cells	2304
FPGA Gates	40K
FPGA Registers	2862
Voltage - Supply	3V ~ 3.6V
Mounting Type	Surface Mount
Operating Temperature	-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/at94k40al-25bqi

Figure 9. FreeRAM Logic[1]



Note: 1. For dual port, the switches on READ ADDR and DATA OUT would be on. The other two would be off. The reverse is true for single port.

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

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

Instruction Set Summary (Continued)

Mnemonics	Operands	Description	Operation	Flags	#Clock
ST	-Y, Rr	Store Indirect and Pre-Decrement	Y ← Y - 1, (Y) ← Rr	None	2
STD	Y+q, Rr	Store Indirect with Displacement	(Y + q) ← Rr	None	2
ST	Z, Rr	Store Indirect	(Z) ← Rr	None	2
ST	Z+, Rr	Store Indirect and Post-Increment	(Z) ← Rr, Z ← Z + 1	None	2
ST	-Z, Rr	Store Indirect and Pre-Decrement	Z ← Z - 1, (Z) ← Rr	None	2
STD	Z+q, Rr	Store Indirect with Displacement	(Z + q) ← Rr	None	2
LPM		Load Program Memory	R0 ← (Z)	None	3
LPM	Rd, Z	Load Program Memory	Rd ← (Z)	None	3
LPM	Rd, Z+	Load Program Memory and Post-Increment	Rd ← (Z), Z ← Z + 1	None	3
IN	Rd, A	In From I/O Location	Rd ← I/O(A)	None	1
OUT	A, Rr	Out To I/O Location	I/O(A) ← Rr	None	1
PUSH	Rr	Push Register on Stack	STACK ← Rr	None	2
POP	Rd	Pop Register from Stack	Rd ← STACK	None	2
Bit and Bit-test Instructions					
LSL	Rd	Logical Shift Left	Rd(n+1)←Rd(n),Rd(0)←0,C←Rd(7)	Z,C,N,V,H	1
LSR	Rd	Logical Shift Right	Rd(n)←Rd(n+1),Rd(7)←0,C←Rd(0)	Z,C,N,V	1
ROL	Rd	Rotate Left Through Carry	Rd(0)←C,Rd(n+1)←Rd(n),C←Rd(7)	Z,C,N,V,H	1
ROR	Rd	Rotate Right Through Carry	Rd(7)←C,Rd(n)←Rd(n+1),C←Rd(0)	Z,C,N,V	1
ASR	Rd	Arithmetic Shift Right	Rd(n) ← Rd(n+1), n=0..6	Z,C,N,V	1
SWAP	Rd	Swap Nibbles	Rd(3..0) ↔ Rd(7..4)	None	1
BSET	s	Flag Set	SREG(s) ← 1	SREG(s)	1
BCLR	s	Flag Clear	SREG(s) ← 0	SREG(s)	1
SBI	A, b	Set Bit in I/O Register	I/O(A, b) ← 1	None	2
CBI	A, b	Clear Bit in I/O Register	I/O(A, b) ← 0	None	2
BST	Rr, b	Bit Store from Register to T	T ← Rr(b)	T	1
BLD	Rd, b	Bit load from T to Register	Rd(b) ← T	None	1
SEC		Set Carry	C ← 1	C	1
CLC		Clear Carry	C ← 0	C	1
SEN		Set Negative Flag	N ← 1	N	1
CLN		Clear Negative Flag	N ← 0	N	1
SEZ		Set Zero Flag	Z ← 1	Z	1
CLZ		Clear Zero Flag	Z ← 0	Z	1
SEI		Global Interrupt Enable	I ← 1	I	1
CLI		Global Interrupt Disable	I ← 0	I	1
SES		Set Signed Test Flag	S ← 1	S	1

AT94K Register Summary (Continued)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reference Page
$16 ($36)	FISUC	FPGA I/O Select, Interrupt Mask/Flag Register C (Reserved on AT94K05)								54, 56
$15 ($35)	FISUB	FPGA I/O Select, Interrupt Mask/Flag Register B								54, 56
$14 ($34)	FISUA	FPGA I/O Select, Interrupt Mask/Flag Register A								54, 56
$13 ($33)	FISCR	FIADR						XFIS1	XFIS0	53
$12 ($32)	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	124
$11 ($31)	DDRD	DDRD7	DDRD6	DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0	124
$10 ($30)	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	124
$0F ($2F)	Reserved									
$0E ($2E)	Reserved									
$0D ($2D)	Reserved									
$0C ($2C)	UDR0	UART0 I/O Data Register								101
$0B ($2B)	UCSR0A	RXC0	TXC0	UDRE0	FE0	OR0		U2X0	MPCM0	101
$0A ($2A)	UCSR0B	RXCIE0	TXCIE0	UDRIE0	RXEN0	TXEN0	CHR90	RXB80	TXB80	103
$09 ($29)	UBRR0	UART0 Baud-rate Register								105
$08 ($28)	OCDR (Reserved)	IDRD								Reserved[1]
$07 ($27)	PORTE	PORTE7	PORTE6	PORTE5	PORTE4	PORTE3	PORTE2	PORTE1	PORTE0	126
$06 ($26)	DDRE	DDRE7	DDRE6	DDRE5	DDRE4	DDRE3	DDRE2	DDRE1	DDRE0	126
$05 ($25)	PINE	PINE7	PINE6	PINE5	PINE4	PINE3	PINE2	PINE1	PINE0	126
$04 ($24)	Reserved									
$03 ($23)	UDR1	UART1 I/O Data Register								101
$02 ($22)	UCSR1A	RXC1	TXC1	UDRE1	FE1	OR1		U2X1	MPCM1	101
$01 ($21)	UCSR1B	RXCIE1	TXCIE1	UDRIE1	RXEN1	TXEN1	CHR91	RXB81	TXB81	103
$00 ($20)	UBRR1	UART1 Baud-rate Register								105

Note: 1. The On-chip Debug Register (OCDR) is detailed on the “FPSLIC On-chip Debug System” distributed within Atmel and select third-party vendors only under Non-Disclosure Agreement (NDA). Contact fpslic@atmel.com for a copy of this document.

The embedded AVR core I/Os and peripherals, and all the virtual FPGA peripherals are placed in the I/O space. The different I/O locations are directly accessed by the IN and OUT instructions transferring data between the 32 x 8 general-purpose working registers and the I/O space. I/O registers within the address range $00 – $1F are directly bit-accessible using the SBI and CBI instructions. In these registers, the value of single bits can be checked by using the SBIS and SBIC instructions. When using the I/O specific instructions IN, OUT, the I/O register address $00 – $3F are used, see Figure 32. When addressing I/O registers as SRAM, $20 must be added to this address. All I/O register addresses throughout this document are shown with the SRAM address in parentheses.

(FPGAIOWE ← IOWE). FPGA macros utilizing one or more FPGA I/O select lines must use the FPGA I/O read/write enables, FPGAIORE or FPGAIOWE, to qualify each select line. The FIADR bit will be cleared (zero) during AVR reset.

- **Bits 6..2 - Res: Reserved Bits**

These bits are reserved and always read as zero.

- **Bits 1, 0 - XFIS1, 0: Extended FPGA I/O Select Bits 1, 0**

XFIS[1:0] determines which one of the four FPGA I/O select lines will be set (one) within the accessed group. An I/O read or write to one of the four dual-purpose I/O addresses, FISUA..D, will access one of four groups. Table 13 details the FPGA I/O selection scheme.

Table 13. FPGA I/O Select Line Scheme

Read or Write I/O Address	**FISCR Register**		**FPGA I/O Select Lines**			
	XFIS1	**XFIS0**	**15..12**	**11..8**	**7..4**	**3..0**
FISUA $14 ($34)	0	0	0000	0000	0000	0001
	0	1	0000	0000	0000	0010
	1	0	0000	0000	0000	0100
	1	1	0000	0000	0000	1000
FISUB $15 ($35)	0	0	0000	0000	0001	0000
	0	1	0000	0000	0010	0000
	1	0	0000	0000	0100	0000
	1	1	0000	0000	1000	0000
FISUC $16 ($36)(1)	0	0	0000	0001	0000	0000
	0	1	0000	0010	0000	0000
	1	0	0000	0100	0000	0000
	1	1	0000	1000	0000	0000
FISUD $17 ($37)(1)	0	0	0001	0000	0000	0000
	0	1	0010	0000	0000	0000
	1	0	0100	0000	0000	0000
	1	1	1000	0000	0000	0000

Note: 1. Not available on AT94K05.

In summary, 16 select signals are sent to the FPGA for I/O addressing. These signals are decoded from four base I/O Register addresses (FISUA..D) and extended to 16 with two bits from the FPGA I/O Select Control Register, XFIS1 and XFIS0. The FPGA I/O read and write signals, FPGAIORE and FPGAIOWE, are qualified versions of the AVR IORE and IOWE signals. Each will only be active if one of the four base I/O addresses is accessed.

Reset: all select lines become active and an FPGAIOWE strobe is enabled. This is to allow the FPGA design to load zeros (8'h00) from the D-bus into appropriate registers.

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

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

Timer/Counter 0 and 2 in PWM Mode

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

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

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

PWM Modes (Up/Down and Overflow)

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

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

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

Table 25. Compare Mode Select in PWM Mode

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

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

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

- **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 30. Timer TOP Values and PWM Frequency

CTC1	PWM11	PWM10	PWM Resolution	Timer TOP Value	Frequency
0	0	1	8-bit	$00FF (255)	f_{TCK1}/510
0	1	0	9-bit	$01FF (511)	f_{TCK1}/1022
0	1	1	10-bit	$03FF(1023)	f_{TCK1}/2046
1	0	1	8-bit	$00FF (255)	f_{TCK1}/256
1	1	0	9-bit	$01FF (511)	f_{TCK1}/512
1	1	1	10-bit	$03FF(1023)	f_{TCK1}/1024

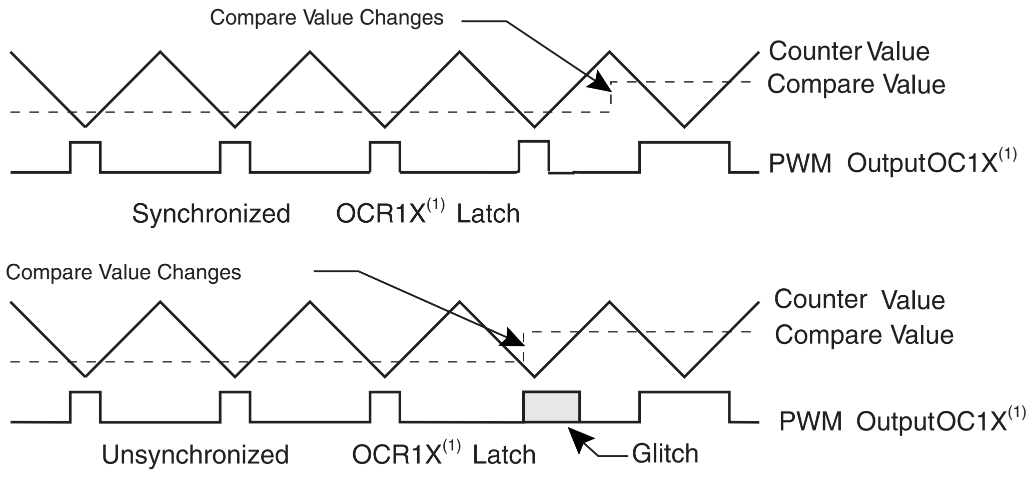
Table 31. Compare1 Mode Select in PWM Mode

CTC1(1)	COM1X1(1)	COM1X0(1)	Effect on OCX1
x(2)	0	x(2)	Not connected
0	1	0	Cleared on compare match, up-counting. Set on compare match, down-counting (non-inverted PWM)
0	1	1	Cleared on compare match, down-counting. Set on compare match, up-counting (inverted PWM)
1	1	0	Cleared on compare match, set on overflow
1	1	1	Set on compare match, set on overflow

Notes: 1. X = A or B
2. x = Don't care

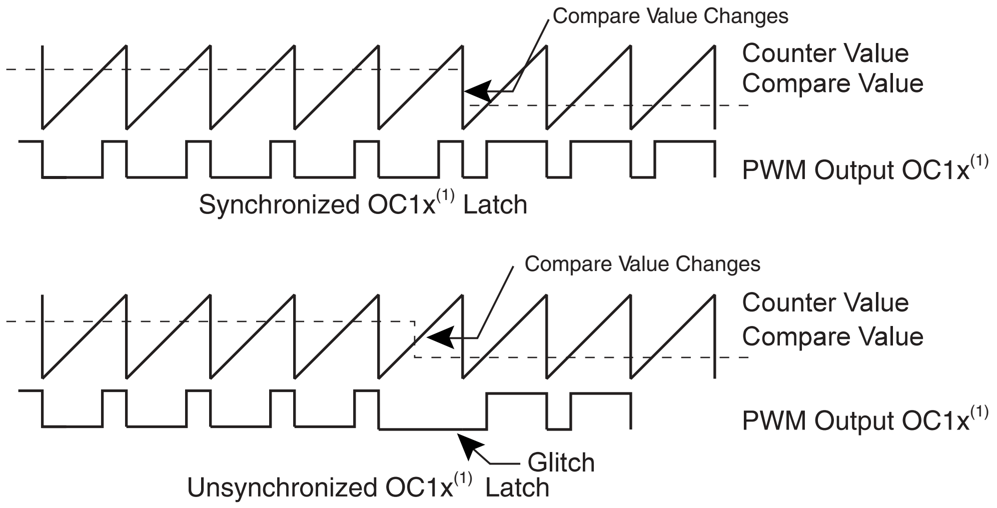
In the PWM mode, the 8, 9 or 10 least significant OCR1A/OCR1B bits (depends of resolution), when written, are transferred to a temporary location. They are latched when Timer/Counter1 reaches the value TOP. This prevents the occurrence of odd-length PWM pulses (glitches) in the event of an unsynchronized OCR1A/OCR1B write. See Figure 56 and Figure 57 for an example in each mode.

Figure 56. Effects on Unsynchronized OCR1 Latching



Note: 1. X = A or B

Figure 57. Effects of Unsynchronized OCR1 Latching in Overflow Mode



Note: 1. X = A or B

During the time between the write and the latch operation, a read from OCR1A or OCR1B will read the contents of the temporary location. This means that the most recently written value always will read out of OCR1A/B.

When the OCR1X contains $0000 or TOP, and the up/down PWM mode is selected, the output OC1A/OC1B is updated to Low or High on the next compare match according to the settings of COM1A1/COM1A0 or COM1B1/COM1B0. This is shown in Table 32. In overflow PWM mode, the output OC1A/OC1B is held Low or High only when the Output Compare Register contains TOP.

```
   mulsu r23, r20            ; (signed)ah * bl
   sbc   r19, r2
   add   r17, r0
   adc   r18, r1
   adc   r19, r2

   mulsu r21, r22            ; (signed)bh * al
   sbc   r19, r2             ; Sign extend
   add   r17, r0
   adc   r18, r1
   adc   r19, r2

   ret

mac16x16_32_method_B:        ; uses two temporary registers (r4,r5), Speed / Size
                             Optimized
                             ; but reduces cycles/words by 1
   clr   r2

   muls  r23, r21            ; (signed)ah * (signed)bh
   movw  r5:r4,r1:r0

   mul   r22, r20            ; al * bl

   add   r16, r0
   adc   r17, r1
   adc   r18, r4
   adc   r19, r5

mulsu r23, r20               ; (signed)ah * bl
   sbc   r19, r2             ; Sign extend
   add   r17, r0
   adc   r18, r1
   adc   r19, r2

   mulsu r21, r22            ; (signed)bh * al
   sbc   r19, r2             ; Sign extend
   add   r17, r0
   adc   r18, r1
   adc   r19, r2

   ret
   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.

UART0 Control and Status Registers – UCSR0A

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

UART1 Control and Status Registers – UCSR1A

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

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

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

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

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

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

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

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

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

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

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

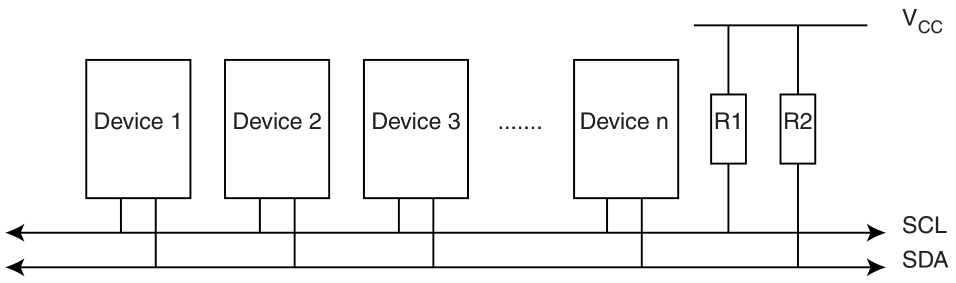
This bit is set if a Framing Error condition is detected, i.e., when the stop bit of an incoming character is zero.

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

2-wire Serial Interface (Byte Oriented)

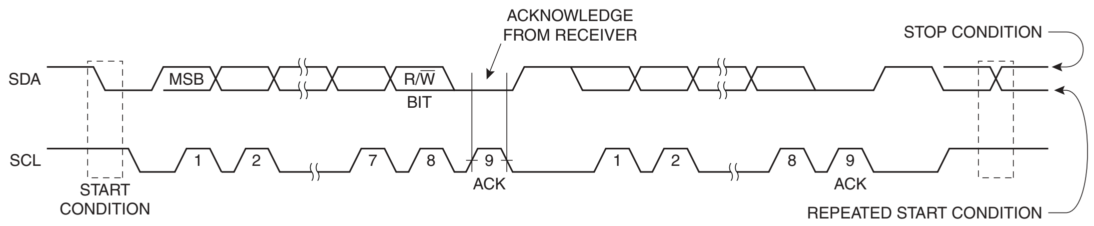
The 2-wire Serial Bus is a bi-directional two-wire serial communication standard. It is designed primarily for simple but efficient integrated circuit (IC) control. The system is comprised of two lines, SCL (Serial Clock) and SDA (Serial Data) that carry information between the ICs connected to them. 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.

- **Bit 4 - TWSTO: 2-wire Serial Bus STOP Condition Flag**

TWSTO is a stop condition flag. In Master mode, setting the TWSTO bit in the control register will generate a STOP condition on the 2-wire Serial Bus. When the STOP condition is executed on the bus, the TWSTO bit is cleared automatically. In Slave mode, setting the TWSTO bit can be used to recover from an error condition. No stop condition is generated on the bus then, but the 2-wire Serial Interface returns to a well-defined unaddressed Slave mode.

- **Bit 3 - TWWC: 2-wire Serial Write Collision Flag**

Set when attempting to write to the 2-wire Serial Data Register – TWDR when TWINT is Low. This flag is updated at each attempt to write the TWDR register.

- **Bit 2 - TWEN: 2-wire Serial Interface Enable Flag**

The TWEN bit enables 2-wire serial operation. If this flag is cleared (zero), the bus outputs SDA and SCL are set to high impedance state and the input signals are ignored. The interface is activated by setting this flag (one).

- **Bit 1 - Res: Reserved Bit**

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

- **Bit 0 - TWIE: 2-wire Serial Interrupt Enable**

When this bit is enabled and the I-bit in SREG is set, the 2-wire Serial Interrupt will be activated for as long as the TWINT flag is High.

The TWCR is used to control the operation of the 2-wire Serial Interface. It is used to enable the 2-wire Serial Interface, to initiate a Master access, to generate a receiver acknowledge, to generate a stop condition, and control halting of the bus while the data to be written to the bus are written to the TWDR. It also indicates a write collision if data is attempted written to TWDR while the register is inaccessible.

The 2-wire Serial Status Register – TWSR

Bit	7	6	5	4	3	2	1	0	
$1D ($3D)	**TWS7**	**TWS6**	**TWS5**	**TWS4**	**TWS3**	**-**	**-**	**-**	**TWSR**
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	1	1	1	1	1	0	0	0	

- **Bits 7..3 - TWS: 2-wire Serial Status**

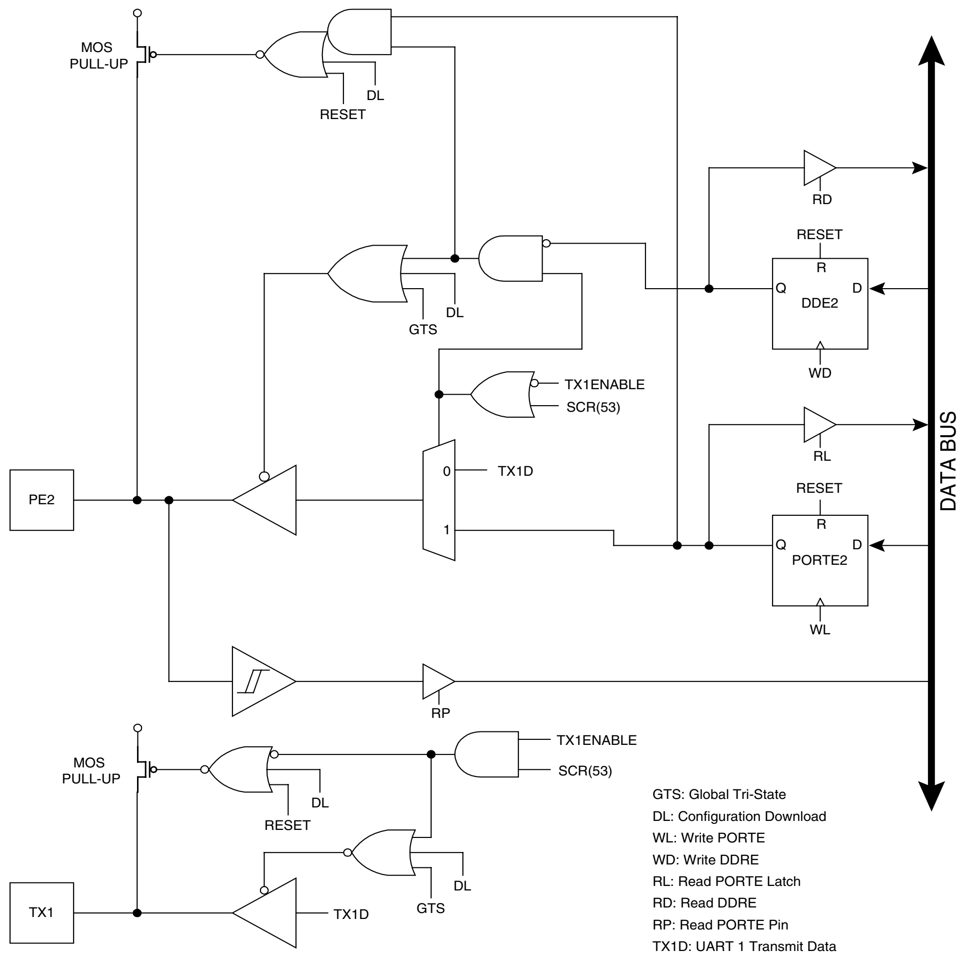
These 5 bits reflect the status of the 2-wire Serial Logic and the 2-wire Serial Bus.

- **Bits 2..0 - Res: Reserved Bits**

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

TWSR is read only. It contains a status code which reflects the status of the 2-wire Serial Logic and the 2-wire Serial Bus. There are 26 possible status codes. When TWSR contains $F8, no relevant state information is available and no 2-wire Serial Interrupt is requested. A valid status code is available in TWSR one CPU clock cycle after the 2-wire Serial Interrupt flag (TWINT) is set by the hardware and is valid until one CPU clock cycle after TWINT is cleared by software. Table 41 to Table 45 give the status information for the various modes.

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

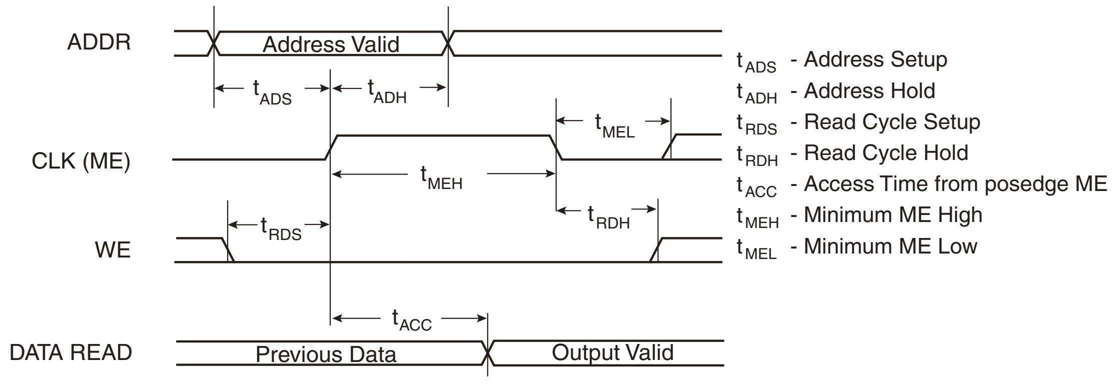
FPSLIC Dual-port SRAM Characteristics

The Dual-port SRAM operates in single-edge clock controlled mode during read operations, and a double-edge controlled mode during write operations. Addresses are clocked internally on the rising edge of the clock signal (ME). Any change of address without a rising edge of ME is not considered.

In read mode, the rising clock edge triggers data read without any significant constraint on the length of the clock pulse. The WE signal must be changed and held Low before the rising edge of ME to signify a read cycle. The WE signal should then remain Low until the falling edge of the clock.

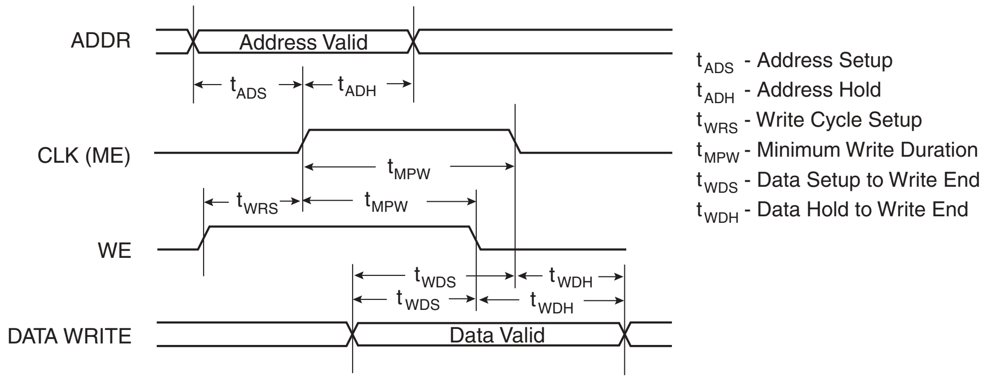
In write mode, data applied to the inputs is latched on either the falling edge of WE or the falling edge of the clock, whichever comes earlier, and written to memory. Also, WE must be High before the rising edge of ME to signify a write cycle. If data inputs change during a write cycle, only the value present at the write end is considered and written to the address clocked at the ME rise. A write cycle ending on WE fall does not turn into a read cycle – the next cycle will be a read cycle if WE remains Low during rising edge of ME.

Figure 83. SRAM Read Cycle Timing Diagram



ADDR
Address Valid
t_{ADS}
t_{ADH}
CLK (ME)
t_{MEL}
t_{MEH}
WE
t_{RDS}
t_{RDH}
t_{ACC}
DATA READ
Previous Data
Output Valid
t_{ADS} - Address Setup
t_{ADH} - Address Hold
t_{RDS} - Read Cycle Setup
t_{RDH} - Read Cycle Hold
t_{ACC} - Access Time from posedge ME
t_{MEH} - Minimum ME High
t_{MEL} - Minimum ME Low

Figure 84. SRAM Write Cycle Timing Diagram



ADDR
Address Valid
t_{ADS}
t_{ADH}
CLK (ME)
t_{MPW}
t_{WRS}
t_{MPW}
WE
t_{WDS}
t_{WDH}
t_{WDS}
t_{WDH}
DATA WRITE
Data Valid
t_{ADS} - Address Setup
t_{ADH} - Address Hold
t_{WRS} - Write Cycle Setup
t_{MPW} - Minimum Write Duration
t_{WDS} - Data Setup to Write End
t_{WDH} - Data Hold to Write End

Frame Interface

The FPGA Frame Clock phase is selectable (see "System Control Register – FPGA/AVR" on page 30). This document refers to the clock at the FPGA/Dual-port SRAM interface as ME (the relation of ME to data, address and write enable does not change). By default, FrameClock is inverted (ME = ~FrameClock). Selecting the non-inverted phase assigns ME = FrameClock. Recall, the Dual-port SRAM operates in single-edge clock controlled mode during read operations, and double-edge clock controlled mode during writes. Addresses are clocked internally on the rising edge of the clock signal (ME). Any change of address without a rising edge of ME is not considered.

Table 50. SRAM Read Cycle Timing Numbers
Commercial 3.3V ± 10%/Industrial 3.3V ± 10%

		Commercial			Industrial			
Symbol	Parameter	Minimum	Typical	Maximum	Minimum	Typical	Maximum	Units
t_{ADS}	Address Setup	0.6	0.8	1.1	0.5	0.8	1.2	ns
t_{ADH}	Address Hold	0.7	0.9	1.3	0.6	0.9	1.5	ns
t_{RDS}	Read Cycle Setup	0	0	0	0	0	0	ns
t_{RDH}	Read Cycle Hold	0	0	0	0	0	0	ns
t_{ACC}	Access Time from Posedge ME	3.4	4.2	5.9	2.9	4.2	6.9	ns
t_{MEH}	Minimum ME High	0.7	0.9	1.3	0.6	0.9	1.5	ns
t_{MEl}	Minimum ME Low	0.6	0.8	1.1	0.6	0.8	1.3	ns

Table 51. SRAM Write Cycle Timing Numbers
Commercial 3.3V ± 10%/Industrial 3.3V ± 10%

		Commercial			Industrial			
Symbol	Parameter	Minimum	Typical	Maximum	Minimum	Typical	Maximum	Units
t_{ADS}	Address Setup	0.6	0.8	1.1	0.5	0.8	1.2	ns
t_{ADH}	Address Hold	0.7	0.9	1.3	0.6	0.9	1.5	ns
t_{WRS}	Write Cycle Setup	0	0	0	0	0	0	ns
t_{MPW}	Minimum Write Duration	1.4	1.8	2.5	1.2	1.8	3.0	ns
t_{WDS}	Data Setup to Write End	4.6	5.7	8.0	3.9	5.7	9.4	ns
t_{WDH}	Data Hold to Write End	0.6	0.8	1.1	0.5	0.8	1.3	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
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 pad -> clock	AT94K05 AT94K10 AT94K40	1.2 1.5 1.9	ns ns	Rising Edge Clock
FCK Input Buffer	t_{PD} (Maximum)	pad -> clock pad -> clock	AT94K05 AT94K10 AT94K40	0.7 0.8 0.9	ns ns	Rising Edge Clock
Clock Column Driver	t_{PD} (Maximum)	clock -> colclk clock -> colclk	AT94K05 AT94K10 AT94K40	1.3 1.8 2.5	ns ns	Rising Edge Clock
Clock Sector Driver	t_{PD} (Maximum)	colclk -> secclk colclk -> secclk	AT94K05 AT94K10 AT94K40	1.0 1.0 1.0	ns ns	Rising Edge Clock
GSRN Input Buffer	t_{PD} (Maximum)	colclk -> secclk colclk -> secclk	AT94K05 AT94K10 AT94K40	5.4 8.2	ns ns	–
Global Clock to Output	t_{PD} (Maximum)	clock pad -> out clock pad -> out	AT94K05 AT94K10 AT94K40	12.6 13.4 14.5	ns ns	Rising Edge Clock Fully Loaded Clock Tree Rising Edge DFF 20 mA Output Buffer 50 pf Pin Load
Fast Clock to Output	t_{PD} (Maximum)	clock pad -> out clock pad -> out	AT94K05 AT94K10 AT94K40	12.1 12.7 13.5	ns ns	Rising Edge Clock Fully Loaded Clock Tree Rising Edge DFF 20 mA Output Buffer 50 pf Pin Load

Table of Contents