**Welcome to E-XFL.COM**

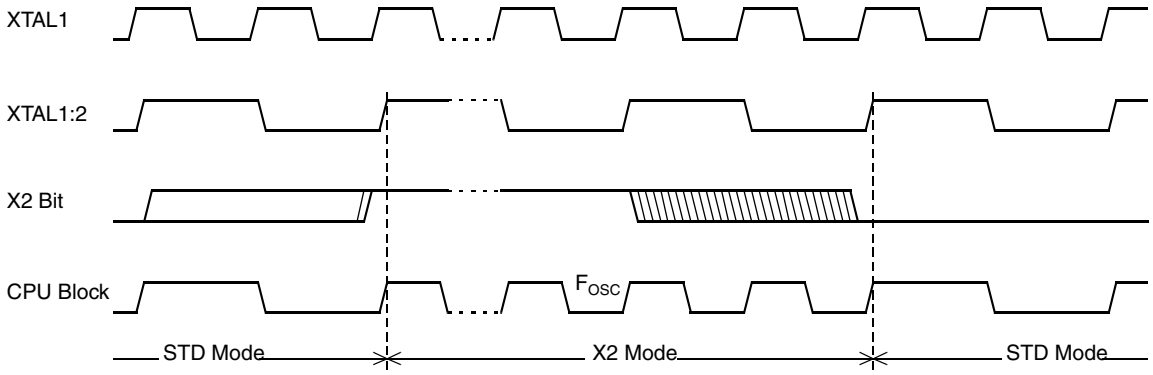
## What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

## Applications of "Embedded - Microcontrollers"

| Details | |
|---|---|
| Product Status | Obsolete |
| Core Processor | 80C51 |
| Core Size | 8-Bit |
| Speed | 40MHz |
| Connectivity | UART/USART |
| Peripherals | POR, PWM, WDT |
| Number of I/O | 32 |
| Program Memory Size | - |
| Program Memory Type | ROMless |
| EEPROM Size | - |
| RAM Size | 1K x 8 |
| Voltage - Supply (Vcc/Vdd) | 2.7V ~ 5.5V |
| Data Converters | - |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 44-LQFP |
| Supplier Device Package | 44-VQFP (10x10) |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/at80c51rd2-rltum |

**Figure 6-2.** Mode Switching Waveforms




The X2 bit in the CKCON0 register (see Table 6-1) allows to switch from 12 clock periods per instruction to 6 clock periods and vice versa. At reset, the speed is set according to X2 bit of Hardware Config Byte (HCB). By default, Standard mode is activated. Setting the X2 bit activates the X2 feature (X2 mode).

The T0X2, T1X2, T2X2, UARTX2, PCAX2 and WDX2 bits in the CKCON0 register (Table 6-1) allow to switch from standard peripheral speed (12 clock periods per peripheral clock cycle) to fast peripheral speed (6 clock periods per peripheral clock cycle). These bits are active only in X2 mode.

**Table 6-1.** CKCON0 Register

CKCON0 - Clock Control Register (8Fh)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| - | WDX2 | PCAX2 | SIX2 | T2X2 | T1X2 | T0X2 | X2 |

| Bit Number | Bit Mnemonic | Description |
|---|---|---|
| 7 | - | **Reserved**<br>Do not set this bit. |
| 6 | WDX2 | **Watchdog clock** (This control bit is validated when the CPU clock X2 is set; when X2 is low, this bit has no effect).<br>Cleared to select 6 clock periods per peripheral clock cycle.<br>Set to select 12 clock periods per peripheral clock cycle. |

# 7. Dual Data Pointer Register

The additional data pointer can be used to speed up code execution and reduce code size.

The dual DPTR structure is a way by which the chip will specify the address of an external data memory location. There are two 16-bit DPTR registers that address the external memory, and a single bit called DPS = AUXR1.0 (see Table 7-1) that allows the program code to switch between them (Refer to Figure 7-1).

**Figure 7-1.** Use of Dual Pointer

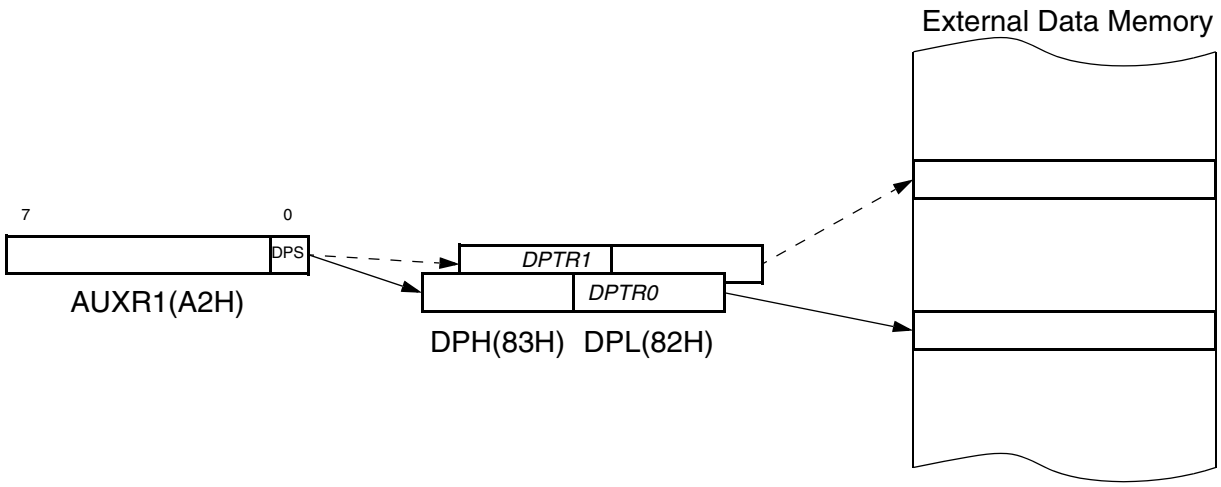



**Table 7-1.** AUXR1 Register

AUXR1- Auxiliary Register 1(0A2h)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| - | - | - | - | GF3 | 0 | - | DPS |

| Bit Number | Bit Mnemonic | Description |
|---|---|---|
| 7 | - | **Reserved**<br>The value read from this bit is indeterminate. Do not set this bit. |
| 6 | - | **Reserved**<br>The value read from this bit is indeterminate. Do not set this bit. |
| 5 | - | Reserved |
| 4 | - | **Reserved**<br>The value read from this bit is indeterminate. Do not set this bit. |
| 3 | GF3 | This bit is a general purpose user flag. |
| 2 | 0 | Always cleared(1). |
| 1 | - | **Reserved**<br>The value read from this bit is indeterminate. Do not set this bit. |
| 0 | DPS | **Data Pointer Selection**<br>Cleared to select DPTR0.<br>Set to select DPTR1. |

Reset Value: XXXX XXXX0b

Not bit addressable

Note: 1. Bit 2 stuck at 0; this allows to use INC AUXR1 to toggle DPS without changing GF3.

# 8. Expanded RAM (XRAM)

The AT80C51RD2 devices provide additional Bytes of Random Access Memory (RAM) space for increased data parameter handling and high level language usage.

The devices have expanded RAM in external data space; maximum size and location are described in Table 8-1.

**Table 8-1.** Expanded RAM

| | XRAM size | Address | |
|---|---|---|---|
| | | Start | End |
| T83C51RB2/RC2<br>T80C51RD2 | 1024 | 00h | 3FFh |

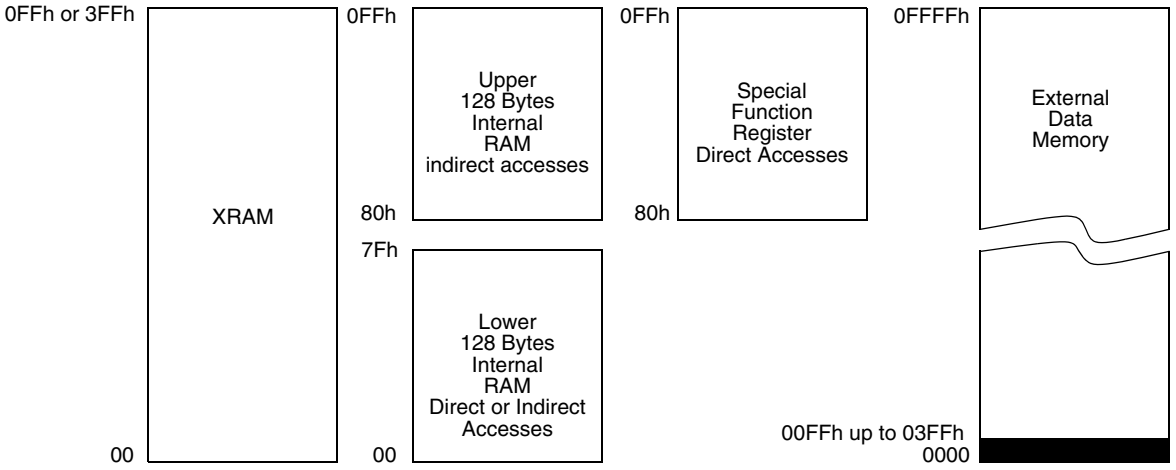
The AT80C51RD2 has internal data memory that is mapped into four separate segments.

The four segments are:

1. The Lower 128 bytes of RAM (addresses 00h to 7Fh) are directly and indirectly addressable.
2. The Upper 128 bytes of RAM (addresses 80h to FFh) are indirectly addressable only.
3. The Special Function Registers (SFRs) (addresses 80h to FFh) are directly addressable only.
4. The expanded RAM bytes are indirectly accessed by MOVX instructions, and with the EXTRAM bit cleared in the AUXR register (see Table 8-1).

The lower 128 bytes can be accessed by either direct or indirect addressing. The Upper 128 bytes can be accessed by indirect addressing only. The Upper 128 bytes occupy the same address space as the SFR. That means they have the same address, but are physically separate from SFR space.

**Figure 8-1.** Internal and External Data Memory Address




When an instruction accesses an internal location above address 7Fh, the CPU knows whether the access is to the upper 128 bytes of data RAM or to SFR space by the addressing mode used in the instruction.

- Instructions that use direct addressing access SFR space. For example: MOV 0A0H, # data, accesses the SFR at location 0A0h (which is P2).

- Instructions that use indirect addressing access the Upper 128 bytes of data RAM. For example: MOV @R0, # data where R0 contains 0A0h, accesses the data byte at address 0A0h, rather than P2 (whose address is 0A0h).
- The XRAM bytes can be accessed by indirect addressing, with EXTRAM bit cleared and MOVX instructions. This part of memory which is physically located on-chip, logically occupies the first bytes of external data memory. The bits XRS0 and XRS1 are used to hide a part of the available XRAM as explained in Table 8-1. This can be useful if external peripherals are mapped at addresses already used by the internal XRAM.
- With EXTRAM = 0, the XRAM is indirectly addressed, using the MOVX instruction in combination with any of the registers R0, R1 of the selected bank or DPTR. An access to XRAM will not affect ports P0, P2, P3.6 (WR) and P3.7 (RD). For example, with EXTRAM = 0, MOVX @R0, # data where R0 contains 0A0H, accesses the XRAM at address 0A0H rather than external memory. An access to external data memory locations higher than the accessible size of the XRAM will be performed with the MOVX DPTR instructions in the same way as in the standard 80C51, with P0 and P2 as data/address busses, and P3.6 and P3.7 as write and read timing signals. Accesses to XRAM above 0FFH can only be done by the use of DPTR.
- With EXTRAM = 1, MOVX @Ri and MOVX @DPTR will be similar to the standard 80C51. MOVX @ Ri will provide an eight-bit address multiplexed with data on Port 0 and any output port pins can be used to output higher order address bits. This is to provide the external paging capability. MOVX @DPTR will generate a sixteen-bit address. Port2 outputs the high-order eight address bits (the contents of DPH) while Port0 multiplexes the low-order eight address bits (DPL) with data. MOVX @ Ri and MOVX @DPTR will generate either read or write signals on P3.6 ($\overline{WR}$) and P3.7 ($\overline{RD}$).

The stack pointer (SP) may be located anywhere in the 256 bytes RAM (lower and upper RAM) internal data memory. The stack may not be located in the XRAM.

The M0 bit allows to stretch the XRAM timings; if M0 is set, the read and write pulses are extended from 6 to 30 clock periods. This is useful to access external slow peripherals.

**Table 8-2.** AUXR Register

AUXR - Auxiliary Register (8Eh)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| - | - | M0 | - | XRS1 | XRS0 | EXTRAM | AO |

| Bit Number | Bit Mnemonic | Description |
|---|---|---|
| 7 | - | **Reserved**<br>The value read from this bit is indeterminate. Do not set this bit |
| 6 | - | **Reserved**<br>The value read from this bit is indeterminate. Do not set this bit |
| 5 | M0 | **Pulse length**<br>Cleared to stretch MOVX control: the $\overline{RD}$ and the $\overline{WR}$ pulse length is 6 clock periods (default).<br>Set to stretch MOVX control: the $\overline{RD}$ and the $\overline{WR}$ pulse length is 30 clock periods. |
| 4 | - | **Reserved**<br>The value read from this bit is indeterminate. Do not set this bit |

# 9. Timer 2

The Timer 2 in the AT80C51RD2 is the standard C52 Timer 2.
It is a 16-bit timer/counter: the count is maintained by two eight-bit timer registers, TH2 and TL2 are cascaded. It is controlled by T2CON (Table 9-1) and T2MOD (Table 9-2) registers. Timer 2 operation is similar to Timer 0 and Timer 1. C/$\overline{T2}$ selects $F_{OSC}/12$ (timer operation) or external pin T2 (counter operation) as the timer clock input. Setting TR2 allows TL2 to be incremented by the selected input.

Timer 2 has 3 operating modes: capture, auto-reload and Baud Rate Generator. These modes are selected by the combination of RCLK, TCLK and CP/$\overline{RL2}$ (T2CON).

Refer to the Atmel 8-bit Microcontroller Hardware description for Capture and Baud Rate Generator Modes.

Timer 2 includes the following enhancements:

- Auto-reload mode with up or down counter
- Programmable clock-output

## 9.1 Auto-reload Mode

The auto-reload mode configures Timer 2 as a 16-bit timer or event counter with automatic reload. If DCEN bit in T2MOD is cleared, Timer 2 behaves as in 80C52 (refer to the Atmel 8-bit Microcontroller Hardware description). If DCEN bit is set, Timer 2 acts as an Up/down timer/counter as shown in Figure 9-1. In this mode the T2EX pin controls the direction of count.

When T2EX is high, Timer 2 counts up. Timer overflow occurs at FFFFh which sets the TF2 flag and generates an interrupt request. The overflow also causes the 16-bit value in RCAP2H and RCAP2L registers to be loaded into the timer registers TH2 and TL2.

When T2EX is low, Timer 2 counts down. Timer underflow occurs when the count in the timer registers TH2 and TL2 equals the value stored in RCAP2H and RCAP2L registers. The underflow sets TF2 flag and reloads FFFFh into the timer registers.

The EXF2 bit toggles when Timer 2 overflows or underflows according to the direction of the count. EXF2 does not generate any interrupt. This bit can be used to provide 17-bit resolution.

**Table 9-2.** T2MOD Register

T2MOD - Timer 2 Mode Control Register (C9h)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| - | - | - | - | - | - | T2OE | DCEN |

| Bit Number | Bit Mnemonic | Description |
|---|---|---|
| 7 | - | **Reserved**<br>The value read from this bit is indeterminate. Do not set this bit. |
| 6 | - | **Reserved**<br>The value read from this bit is indeterminate. Do not set this bit. |
| 5 | - | **Reserved**<br>The value read from this bit is indeterminate. Do not set this bit. |
| 4 | - | **Reserved**<br>The value read from this bit is indeterminate. Do not set this bit. |
| 3 | - | **Reserved**<br>The value read from this bit is indeterminate. Do not set this bit. |
| 2 | - | **Reserved**<br>The value read from this bit is indeterminate. Do not set this bit. |
| 1 | T2OE | **Timer 2 Output Enable bit**<br>Cleared to program P1.0/T2 as clock input or I/O port.<br>Set to program P1.0/T2 as clock output. |
| 0 | DCEN | **Down Counter Enable bit**<br>Cleared to disable Timer 2 as up/down counter.<br>Set to enable Timer 2 as up/down counter. |

Reset Value = XXXX XX00b

Not bit addressable

Table 10-3 shows the CCAPMn settings for the various PCA functions.

**Table 10-3.** CCAPMn Registers (n = 0-4)

CCAPM0 - PCA Module 0 Compare/Capture Control Register (0DAh)

CCAPM1 - PCA Module 1 Compare/Capture Control Register (0DBh)

CCAPM2 - PCA Module 2 Compare/Capture Control Register (0DCh)

CCAPM3 - PCA Module 3 Compare/Capture Control Register (0DDh)

CCAPM4 - PCA Module 4 Compare/Capture Control Register (0DEh)

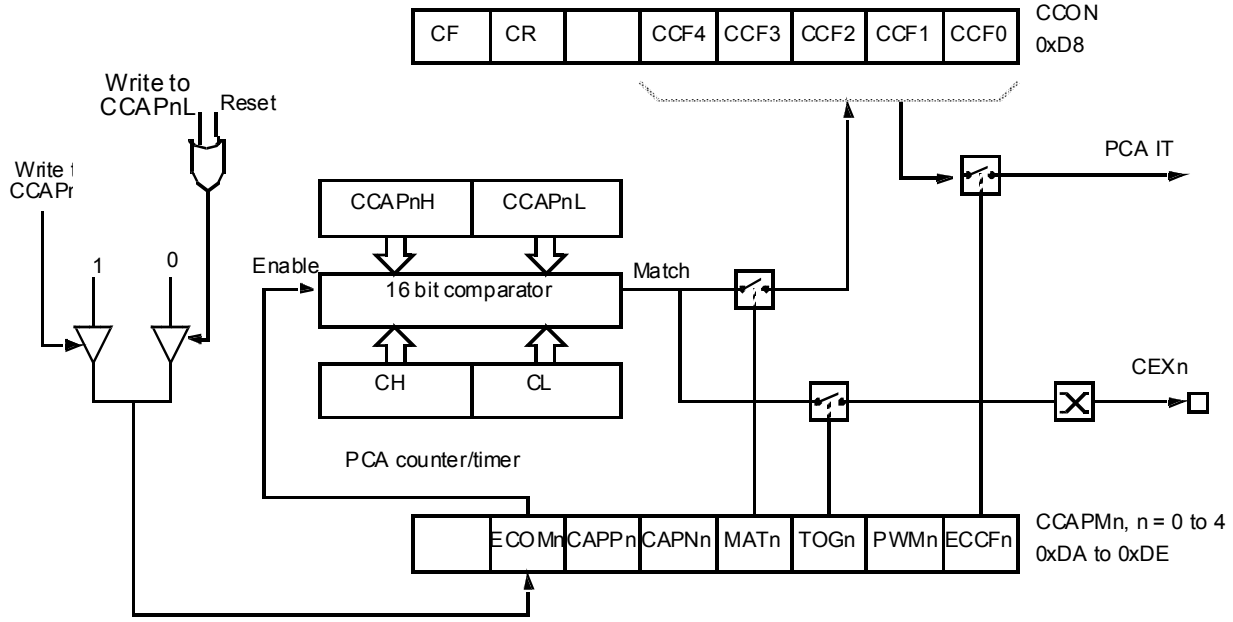
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| - | ECOMn | CAPPn | CAPNn | MATn | TOGn | PWMn | ECCFn |

| Bit Number | Bit Mnemonic | Description |
|---|---|---|
| 7 | - | **Reserved**<br>The value read from this bit is indeterminate. Do not set this bit. |
| 6 | ECOMn | **Enable Comparator**<br>Cleared to disable the comparator function.<br>Set to enable the comparator function. |
| 5 | CAPPn | **Capture Positive**<br>Cleared to disable positive edge capture.<br>Set to enable positive edge capture. |
| 4 | CAPNn | **Capture Negative**<br>Cleared to disable negative edge capture.<br>Set to enable negative edge capture. |
| 3 | MATn | **Match**<br>When MATn = 1, a match of the PCA counter with this module's compare/capture register causes the CCFn bit in CCON to be set, flagging an interrupt. |
| 2 | TOGn | **Toggle**<br>When TOGn = 1, a match of the PCA counter with this module's compare/capture register causes the CEXn pin to toggle. |
| 1 | PWMn | **Pulse Width Modulation Mode**<br>Cleared to disable the CEXn pin to be used as a pulse width modulated output.<br>Set to enable the CEXn pin to be used as a pulse width modulated output. |
| 0 | CCF0 | **Enable CCF interrupt**<br>Cleared to disable compare/capture flag CCFn in the CCON register to generate an interrupt.<br>Set to enable compare/capture flag CCFn in the CCON register to generate an interrupt. |

Reset Value = X000 0000b

Not bit addressable

**Figure 10-5.** PCA High-speed Output Mode




Before enabling ECOM bit, CCAPnL and CCAPnH should be set with a non zero value, otherwise an unwanted match could occur.

Once ECOM is set, writing CCAPnL will clear ECOM so that an unwanted match doesn't occur while modifying the compare value. Writing to CCAPnH will set ECOM. For this reason, user software should write CCAPnL first, and then CCAPnH. Of course, the ECOM bit can still be controlled by accessing the CCAPMn register.

## 10.4 Pulse Width Modulator Mode

All of the PCA modules can be used as PWM outputs. Figure 10-6 shows the PWM function. The frequency of the output depends on the source for the PCA timer. All of the modules will have the same frequency of output because they all share the PCA timer. The duty cycle of each module is independently variable using the module's capture register CCAPLn. When the value of the PCA CL SFR is less than the value in the module's CCAPLn SFR the output will be low, when it is equal to or greater than the output will be high. When CL overflows from FF to 00, CCAPLn is reloaded with the value in CCAPHn. This allows updating the PWM without glitches. The PWM and ECOM bits in the module's CCAPMn register must be set to enable the PWM mode.

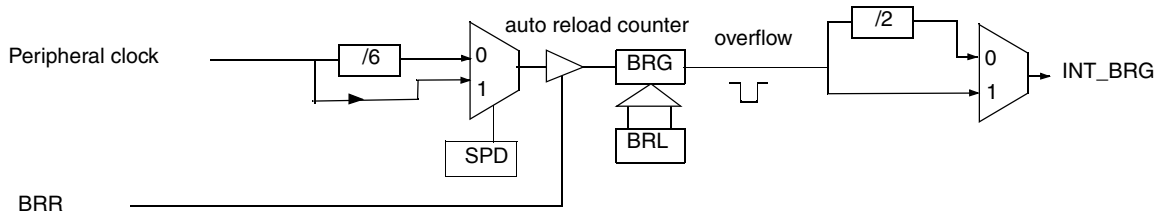
**Table 11-3.** Baud Rate Selection Table UART

| TCLK (T2CON) | RCLK (T2CON) | TBCK (BDRCON) | RBCK (BDRCON) | Clock Source UART Tx | Clock Source UART Rx |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | Timer 1 | Timer 1 |
| 1 | 0 | 0 | 0 | Timer 2 | Timer 1 |
| 0 | 1 | 0 | 0 | Timer 1 | Timer 2 |
| 1 | 1 | 0 | 0 | Timer 2 | Timer 2 |
| X | 0 | 1 | 0 | INT_BRG | Timer 1 |
| X | 1 | 1 | 0 | INT_BRG | Timer 2 |
| 0 | X | 0 | 1 | Timer 1 | INT_BRG |
| 1 | X | 0 | 1 | Timer 2 | INT_BRG |
| X | X | 1 | 1 | INT_BRG | INT_BRG |

### 11.3.1 Internal Baud Rate Generator (BRG)

When the internal Baud Rate Generator is used, the Baud Rates are determined by the BRG overflow depending on the BRL reload value, the value of SPD bit (Speed Mode) in BDRCON register and the value of the SMOD1 bit in PCON register.

**Figure 11-5.** Internal Baud Rate




- The baud rate for UART is token by formula:

$$BaudRate = \frac{2_{SMOD} \times F_{CLKPERIPH}}{2 \times 2 \times 6\langle 1 - SPD\rangle \times 16 \times [256 - (BRL)]}$$

$$(BRL) = 256 - \frac{2_{SMOD1} \times F_{CLKPERIPH}}{2 \times 2 \times 6_{(1-SPD)} \times 16 \times BaudRate}$$

**Table 11-4.** SCON Register

SCON - Serial Control Register (98h)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| **FE/SM0** | **SM1** | **SM2** | **REN** | **TB8** | **RB8** | **TI** | **RI** |

| Bit Number | Bit Mnemonic | Description |
|---|---|---|
| 7 | FE | **Framing Error bit (SMOD0 = 1)**<br>Clear to reset the error state, not cleared by a valid stop bit.<br>Set by hardware when an invalid stop bit is detected.<br>SMOD0 must be set to enable access to the FE bit |
| | SM0 | **Serial port Mode bit 0**<br>Refer to SM1 for serial port mode selection.<br>SMOD0 must be cleared to enable access to the SM0 bit |
| 6 | SM1 | **Serial port Mode bit 1**<br>SM1ModeDescriptionBaud Rate<br>0 0Shift Register$f_{CPU\ PERIPH/6}$<br>1 18-bit UARTVariable<br>0 29-bit UART$f_{CPU\ PERIPH\ /32\ or\ /16}$<br>1 39-bit UARTVariable |
| 5 | SM2 | **Serial port Mode 2 bit/Multiprocessor Communication Enable bit**<br>Clear to disable multiprocessor communication feature.<br>Set to enable multiprocessor communication feature in mode 2 and 3, and eventually mode 1. This bit should be cleared in mode 0. |
| 4 | REN | **Reception Enable bit**<br>Clear to disable serial reception.<br>Set to enable serial reception. |
| 3 | TB8 | **Transmitter Bit 8/Ninth bit to transmit in modes 2 and 3**<br>o transmit a logic 0 in the 9th bit.<br>Set to transmit a logic 1 in the 9th bit. |
| 2 | RB8 | **Receiver Bit 8/Ninth bit received in modes 2 and 3**<br>Cleared by hardware if 9th bit received is a logic 0.<br>Set by hardware if 9th bit received is a logic 1.<br>In mode 1, if SM2=0, RB8 is the received stop bit. In mode 0 RB8 is not used. |
| 1 | TI | **Transmit Interrupt flag**<br>Clear to acknowledge interrupt.<br>Set by hardware at the end of the 8th bit time in mode 0 or at the beginning of the stop bit in the other modes. |
| 0 | RI | **Receive Interrupt flag**<br>Clear to acknowledge interrupt.<br>Set by hardware at the end of the 8th bit time in mode 0, see Figure 11-2. and Figure 11-3. in the other modes. |

Reset Value = 0000 0000b

Bit addressable

**Table 11-9.** SBUF Register

SBUF - Serial Buffer Register for UART (99h)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| | | | | | | | |

Reset Value = XXXX XXXXb

**Table 11-10.** BRL Register

BRL - Baud Rate Reload Register for the internal baud rate generator, UART (9Ah)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| | | | | | | | |

Reset Value = 0000 0000b

**Table 11-11.** T2CON Register

T2CON - Timer 2 Control Register (C8h)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| TF2 | EXF2 | RCLK | TCLK | EXEN2 | TR2 | C/T2# | CP/RL2# |

| Bit Number | Bit Mnemonic | Description |
|---|---|---|
| 7 | TF2 | **Timer 2 overflow Flag**<br>Must be cleared by software.<br>Set by hardware on Timer 2 overflow, if RCLK=0 and TCLK=0. |
| 6 | EXF2 | **Timer 2 External Flag**<br>Set when a capture or a reload is caused by a negative transition on T2EX pin if EXEN2 = 1.<br>When set, causes the CPU to vector to Timer 2 interrupt routine when Timer 2 interrupt is enabled.<br>Must be cleared by software. EXF2 doesn't cause an interrupt in Up/down counter mode (DCEN=1) |
| 5 | RCLK | **Receive Clock bit for UART**<br>Cleared to use timer 1 overflow as receive clock for serial port in mode 1 or 3.<br>Set to use Timer 2 overflow as receive clock for serial port in mode 1 or 3. |
| 4 | TCLK | **Transmit Clock bit for UART**<br>Cleared to use timer 1 overflow as transmit clock for serial port in mode 1 or 3.<br>Set to use Timer 2 overflow as transmit clock for serial port in mode 1 or 3. |
| 3 | EXEN2 | **Timer 2 External Enable bit**<br>Cleared to ignore events on T2EX pin for Timer 2 operation.<br>Set to cause a capture or reload when a negative transition on T2EX pin is detected, if Timer 2 is not used to clock the serial port. |
| 2 | TR2 | **Timer 2 Run control bit**<br>Cleared to turn off Timer 2.<br>Set to turn on Timer 2. |
| 1 | C/T2# | **Timer/Counter 2 select bit**<br>Cleared for timer operation (input from internal clock system: $F_{CLK\ PERIPH}$).<br>Set for counter operation (input from T2 input pin, falling edge trigger). Must be 0 for clock out mode. |
| 0 | CP/RL2# | **Timer 2 Capture/Reload bit**<br>If RCLK = 1 or TCLK = 1, CP/RL2# is ignored and timer is forced to auto-reload on Timer 2 overflow.<br>Cleared to auto-reload on Timer 2 overflows or negative transitions on T2EX pin if EXEN2 = 1.<br>Set to capture on negative transitions on T2EX pin if EXEN2 = 1. |

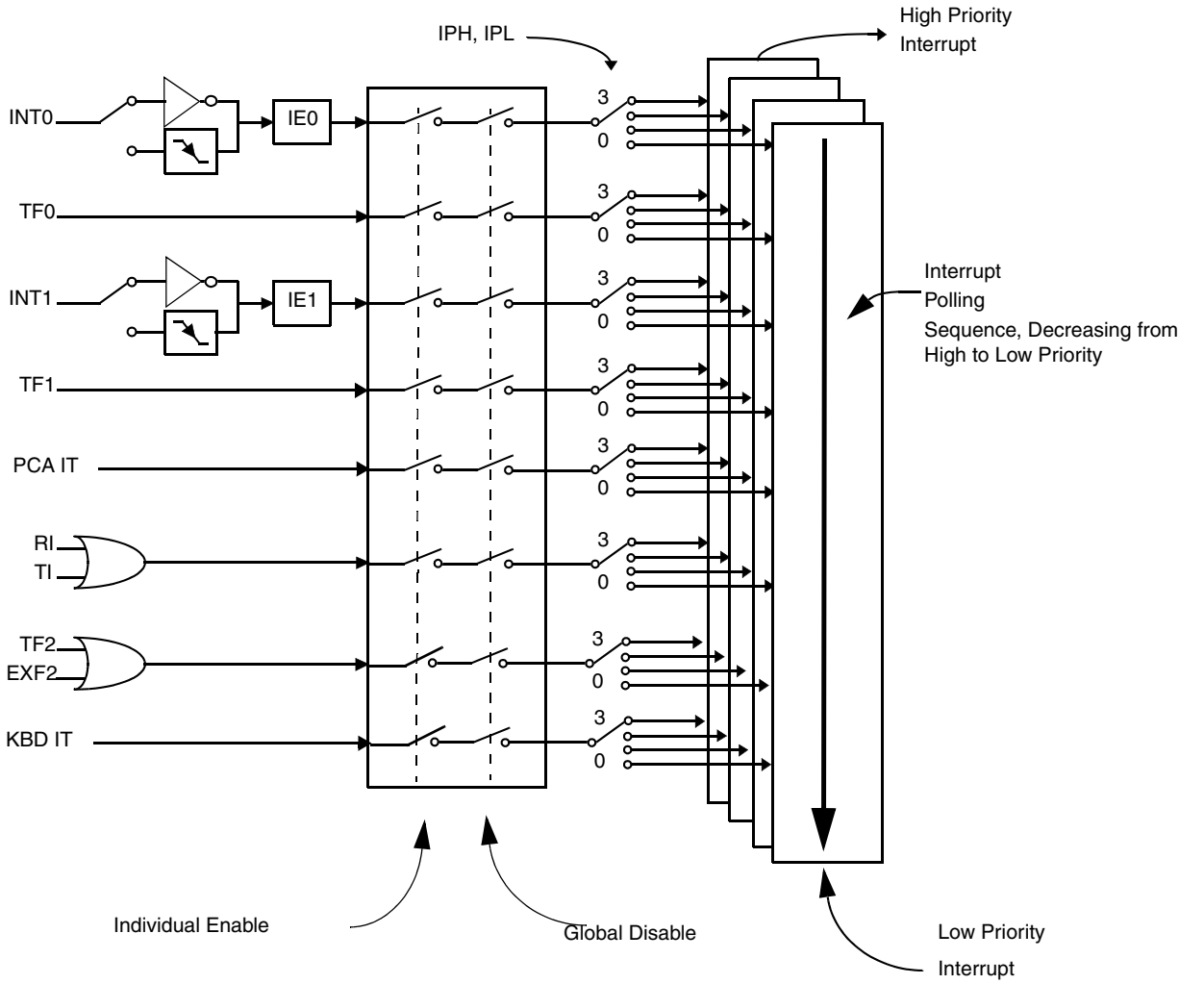
Reset Value = 0000 0000b

Bit addressable

# 12. Interrupt System

The AT80C51RD2 have a total of 8 interrupt vectors: two external interrupts ($\overline{INT0}$ and $\overline{INT1}$), three timer interrupts (timers 0, 1 and 2), the serial port interrupt, Keyboard interrupt and the PCA global interrupt. These interrupts are shown in Figure 12-1.

**Figure 12-1.** Interrupt Control System




Each of the interrupt sources can be individually enabled or disabled by setting or clearing a bit in the Interrupt Enable register (Table 12-5 and Table 12-3). This register also contains a global disable bit, which must be cleared to disable all interrupts at once.

Each interrupt source also can be individually programmed to one out of four priority levels by setting or clearing a bit in the Interrupt Priority register (Table 12-6) and in the Interrupt Priority High register (Table 12-4 and Table 12-5) shows the bit values and priority levels associated with each combination.

## 12.1 Registers

The PCA interrupt vector is located at address 0033H, the Keyboard interrupt vector is located at address 004BH. All other vectors addresses are the same as standard C52 devices.

| Bit Number | Bit Mnemonic | Description |
|---|---|---|
| 7 | - | **Reserved**<br>The value read from this bit is indeterminate. Do not set this bit. |
| 6 | PPCH | **PCA interrupt priority high bit.**<br>PPCHPPCLPriority Level<br>0 0Lowest<br>0 1<br>1 0<br>1 1Highest |
| 5 | PT2H | **Timer 2 overflow interrupt priority high bit**<br>PT2HPT2LPriority Level<br>0 0Lowest<br>0 1<br>1 0<br>1 1Highest |
| 4 | PSH | **Serial port priority high bit**<br>PSH PSLPriority Level<br>0 0Lowest<br>0 1<br>1 0<br>1 1Highest |
| 3 | PT1H | **Timer 1 overflow interrupt priority high bit**<br>PT1HPT1LPriority Level<br>0 0 Lowest<br>0 1<br>1 0<br>1 1Highest |
| 2 | PX1H | **External interrupt 1 priority high bit**<br>PX1HPX1LPriority Level<br>0 0Lowest<br>0 1<br>1 0<br>1 1Highest |
| 1 | PT0H | **Timer 0 overflow interrupt priority high bit**<br>PT0HPT0LPriority Level<br>0 0Lowest<br>0 1<br>1 0<br>1 1Highest |
| 0 | PX0H | **External interrupt 0 priority high bit**<br>PX0H PX0LPriority Level<br>0 0Lowest<br>0 1<br>1 0<br>1 1Highest |

Reset Value = X000 0000b
Not bit addressable

**Table 12-5.** IE1 Register

IE1 - Interrupt Enable Register (B1h)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| - | - | - | - | - | - | - | KBD |

IPH1 - Interrupt Priority High Register (B3h)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| - | - | - | - | - | - | - | KBDH |

| Bit Number | Bit Mnemonic | Description |
|---|---|---|
| 7 | - | **Reserved**<br>The value read from this bit is indeterminate. Do not set this bit. |
| 6 | - | **Reserved**<br>The value read from this bit is indeterminate. Do not set this bit. |
| 5 | - | **Reserved**<br>The value read from this bit is indeterminate. Do not set this bit. |
| 4 | - | **Reserved**<br>The value read from this bit is indeterminate. Do not set this bit. |
| 3 | - | **Reserved**<br>The value read from this bit is indeterminate. Do not set this bit. |
| 2 | - | **Reserved**<br>The value read from this bit is indeterminate. Do not set this bit. |
| 1 | - | **Reserved**<br>The value read from this bit is indeterminate. Do not set this bit. |
| 0 | KBDH | **Keyboard interrupt Priority High bit**<br>KB DHKBDLPriority Level<br>0 0 Lowest<br>0 1<br>1 0<br>1 1Highest |

Reset Value = XXXX XXX0b
Not bit addressable

## 12.2 Interrupt Sources and Vector Addresses

**Table 12-8.** Interrupt Sources and Vector Addresses

| Number | Polling Priority | Interrupt Source | Interrupt Request | Vector Address |
|---|---|---|---|---|
| 0 | 0 | Reset | | 0000h |
| 1 | 1 | INT0 | IE0 | 0003h |
| 2 | 2 | Timer 0 | TF0 | 000Bh |
| 3 | 3 | INT1 | IE1 | 0013h |
| 4 | 4 | Timer 1 | IF1 | 001Bh |
| 5 | 6 | UART | RI+TI | 0023h |
| 6 | 7 | Timer 2 | TF2+EXF2 | 002Bh |
| 7 | 5 | PCA | CF + CCFn (n = 0-4) | 0033h |
| 8 | 8 | Keyboard | KBDIT | 003Bh |

# 14. Power Management

## 14.1 Idle Mode

An instruction that sets PCON.0 indicates that it is the last instruction to be executed before going into Idle mode. In Idle mode, the internal clock signal is gated off to the CPU, but not to the interrupt, Timer, and Serial Port functions. The CPU status is preserved in its entirety: the Stack Pointer, Program Counter, Program Status Word, Accumulator and all other registers maintain their data during idle. The port pins hold the logical states they had at the time Idle was activated. ALE and PSEN hold at logic high level.

There are two ways to terminate the Idle mode. Activation of any enabled interrupt will cause PCON.0 to be cleared by hardware, terminating the Idle mode. The interrupt will be serviced, and following RETI the next instruction to be executed will be the one following the instruction that put the device into idle.

The flag bits GF0 and GF1 can be used to give an indication if an interrupt occurred during normal operation or during idle. For example, an instruction that activates idle can also set one or both flag bits. When idle is terminated by an interrupt, the interrupt service routine can examine the flag bits.

The other way of terminating the Idle mode is with a hardware reset. Since the clock oscillator is still running, the hardware reset needs to be held active for only two machine cycles (24 oscillator periods) to complete the reset.

## 14.2 Power-down Mode

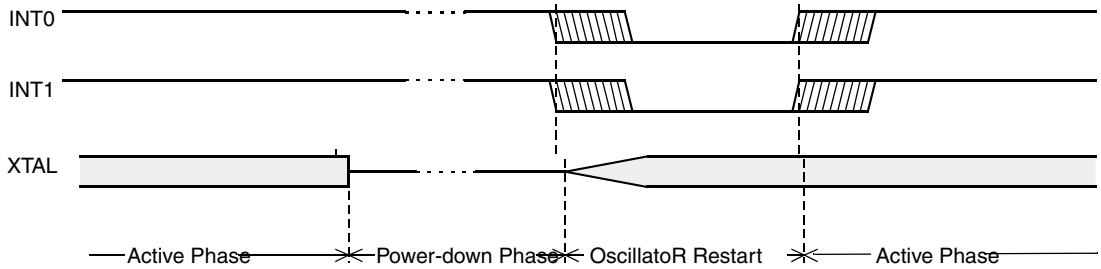
To save maximum power, a power-down mode can be invoked by software (refer to Table 11-12, PCON register).

In power-down mode, the oscillator is stopped and the instruction that invoked power-down mode is the last instruction executed. The internal RAM and SFRs retain their value until the power-down mode is terminated. $V_{CC}$ can be lowered to save further power. Either a hardware reset or an external interrupt can cause an exit from power-down. To properly terminate power-down, the reset or external interrupt should not be executed before $V_{CC}$ is restored to its normal operating level and must be held active long enough for the oscillator to restart and stabilize.

Only external interrupts $\overline{INT0}$, $\overline{INT1}$ and Keyboard Interrupts are useful to exit from power-down. Thus, the interrupt must be enabled and configured as level - or edge - sensitive interrupt input. When Keyboard Interrupt occurs after a power-down mode, 1024 clocks are necessary to exit to power-down mode and enter in operating mode.

Holding the pin low restarts the oscillator but bringing the pin high completes the exit as detailed in Figure 14-1. When both interrupts are enabled, the oscillator restarts as soon as one of the two inputs is held low and power-down exit will be completed when the first input is released. In this case, the higher priority interrupt service routine is executed. Once the interrupt is serviced, the next instruction to be executed after RETI will be the one following the instruction that put AT80C51RD2 into power-down mode.

**Figure 14-1.** Power-down Exit Waveform




Exit from power-down by reset redefines all the SFRs, exit from power-down by external interrupt does no affect the SFRs.

Exit from power-down by either reset or external interrupt does not affect the internal RAM content.

Note: If idle mode is activated with power-down mode (IDL and PD bits set), the exit sequence is unchanged, when execution is vectored to interrupt, PD and IDL bits are cleared and idle mode is not entered.

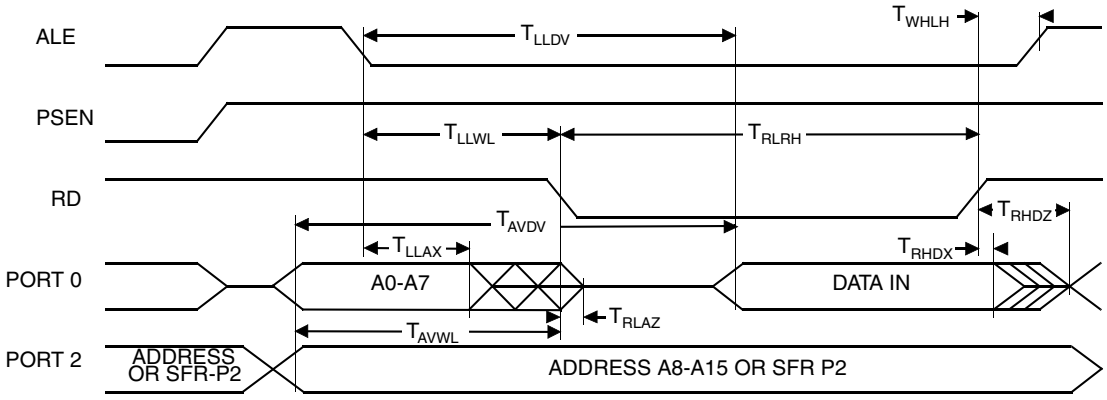
Table 14-1 shows the state of ports during idle and power-down modes.

**Table 14-1.** State of Ports

| Mode | Program Memory | ALE | PSEN | PORT0 | PORT1 | PORT2 | PORT3 |
|---|---|---|---|---|---|---|---|
| Idle | Internal | 1 | 1 | Port Data(1) | Port Data | Port Data | Port Data |
| Idle | External | 1 | 1 | Floating | Port Data | Address | Port Data |
| Power-down | Internal | 0 | 0 | Port Dat(1) | Port Data | Port Data | Port Data |
| Power-down | External | 0 | 0 | Floating | Port Data | Port Data | Port Data |

Note: 1. Port 0 can force a 0 level. A "one" will leave port floating.

### 17.3.6 External Data Memory Read Cycle




### 17.3.7 Serial Port Timing - Shift Register Mode

**Table 17-7.** Symbol Description

| Symbol | Parameter |
|---|---|
| $T_{XLXL}$ | Serial port clock cycle time |
| $T_{QVHX}$ | Output data set-up to clock rising edge |
| $T_{XHQX}$ | Output data hold after clock rising edge |
| $T_{XHDX}$ | Input data hold after clock rising edge |
| $T_{XHDV}$ | Clock rising edge to input data valid |

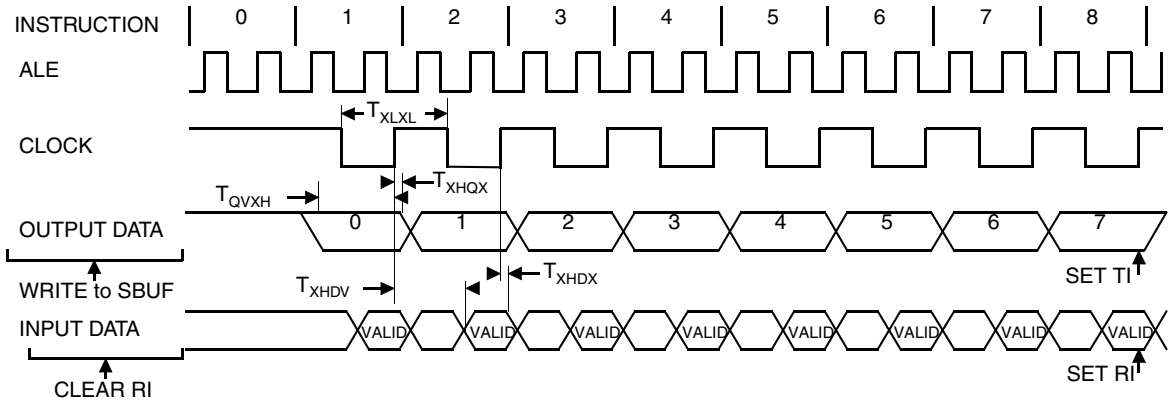
**Table 17-8.** AC Parameters for a Fix Clock

| | -M | | |
|---|---|---|---|
| Symbol | Min | Max | Units |
| $T_{XLXL}$ | 300 | | ns |
| $T_{QVHX}$ | 200 | | ns |
| $T_{XHQX}$ | 30 | | ns |
| $T_{XHDX}$ | 0 | | ns |
| $T_{XHDV}$ | | 117 | ns |

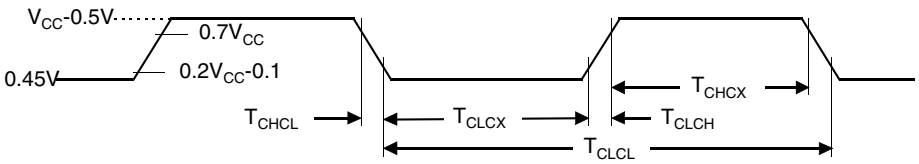
**Table 17-9.** AC Parameters for a Variable Clock

| Symbol | Type | Standard Clock | X2 Clock | X Parameter for -M Range | Units |
|---|---|---|---|---|---|
| $T_{XLXL}$ | Min | 12 T | 6 T | | ns |
| $T_{QVHX}$ | Min | 10 T - x | 5 T - x | 50 | ns |
| $T_{XHQX}$ | Min | 2 T - x | T - x | 20 | ns |
| $T_{XHDX}$ | Min | x | x | 0 | ns |
| $T_{XHDV}$ | Max | 10 T - x | 5 T- x | 133 | ns |

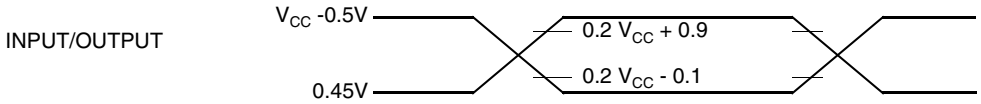
### 17.3.8 Shift Register Timing Waveforms




### 17.3.9 External Clock Drive Waveforms




### 17.3.10 AC Testing Input/Output Waveforms




AC inputs during testing are driven at $V_{CC}$ - 0.5 for a logic “1” and 0.45V for a logic “0”. Timing measurement are made at $V_{IH}$ min for a logic “1” and $V_{IL}$ max for a logic “0”.