SECTION 15: CPU TIMING

OSCILLATOR

The Secure Microcontroller provides an on–chip oscillator circuit which may be driven either by using an external crystal as a time base or from a TTL–compatible clock signal. The oscillator circuitry provides the internal clocking signals to the on–chip CPU and I/O circuitry.

The schematic shown in Figure 15–1 illustrates the required connections when using a crystal. Typically, the values of C1 and C2 should both be 33 pF. If a resonator is used, C1 and C2 should be 47 pF.

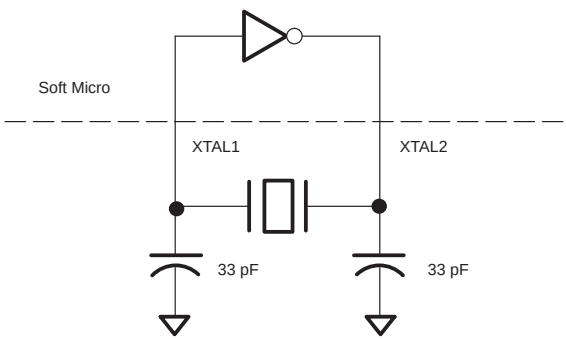
XTAL1
Input to the inverting oscillator amplifier and input to the internal clock generating circuits.

XTAL2
Output from the inverting oscillator amplifier. This pin is also used to distribute the clock to other devices.

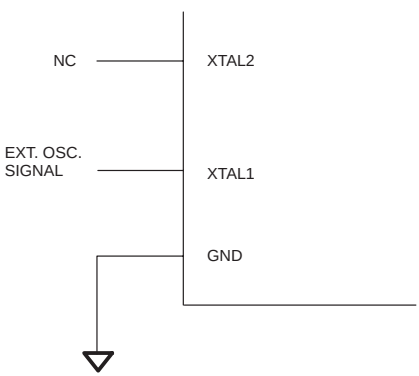
Oscillator Characteristics
XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on–chip oscillator as shown in Figure 15–1. The crystal should be parallel resonant, AT cut type.

To drive the device from an external clock source, XTAL1 should be driven, while XTAL2 is left unconnected as shown in Figure 15–2. There are no requirements on the duty cycle of the external clock signal since the input to the internal clocking circuitry is through a divide–by–two flip–flop. However, minimum and maximum high and low times specified in the electrical specifications must be met to insure proper operation.

CRYSTAL CONNECTION Figure 15–1



CLOCK SOURCE INPUT Figure 15–2



INSTRUCTION TIMING

The internal clocking signals are divided to produce the necessary clock phases, state times, and machine cycles which define the sequential execution of instructions. Two clock oscillator periods define one state time. The first clock oscillator pulse period of a state time is called the Phase 1 clock. while the second is called the Phase 2 clock. In general, arithmetic and logical operations take place during Phase 1 and internal register–to–register transfers take place during Phase 2.

A machine–cycle is composed of a total of twelve oscillator periods or six state times. The state times within the machine cycle are numbered S1 through S6. Each clock oscillator period within the machine cycle is designated according to the state number and the phase it represents within the state. Thus, the oscillator periods are numbered S1P1 (State 1, Phase 1) through S6P2 (State 6, Phase 2).

All of the instruction sequences executed by the CPU are preceded by a single byte (8–bit) opcode and consist of a total of either one, two, or three bytes. Most of the instructions execute in one machine cycle. The rest of the instructions execute in two machine cycles, except for multiply (MUL) and divide (DIV) which execute in four cycles each.

Figure 15–3 is a timing diagram illustrating the memory access and execution timing for typical instructions when they are executed from Byte–wide RAM. The timing shown is referenced to the internally–generated machine cycles composed of state times and clock oscillator phases. The relationship between the internal instruction execution timing and the external signals XTAL2 and ALE is illustrated in the diagram. Except for the MOVX instructions, two code bytes from Program Memory are always read during each machine cycle of instruction execution. These read operations take place at state times S1 and S4.

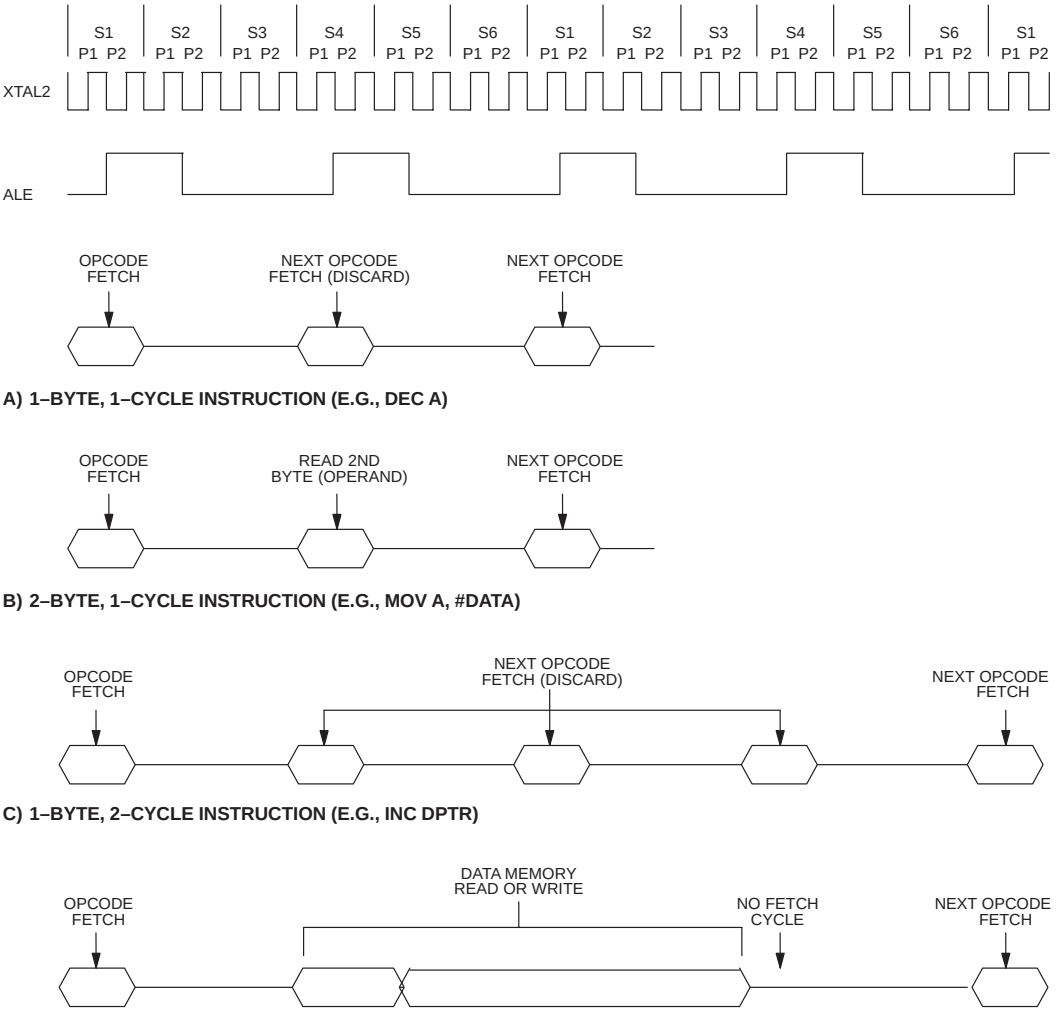
Execution of a 1–byte, 1–cycle instruction is illustrated in Figure 15–3A. It begins with the opcode byte fetch which occurs during S1 and the opcode byte is latched into the Instruction register at S1P2. The code byte which is read during S4, in this case, is actually the opcode byte of the next instruction. This byte is effectively discarded and the Program Counter is not incremented. Execution of the instruction is completed S6P2, the end of the machine cycle.

In the 2–byte 1–cycle instruction shown in Figure 15–3B, the opcode is read during S1 while the second byte of the instruction, or the operand, is read during S4. Again, execution of the instruction is complete at the end of S6P2.

A 1–byte, 2–cycle instruction is shown in Figure 15–3C. In this case the opcode byte is read at S1 of the first machine cycle. The next opcode is then read three times during the S1 and S4 of the second machine cycle. The information is discarded each of these times until it is finally read when the next instruction is actually executed.

Finally, Figure 15–3D illustrates the execution of one of the MOVX instruction which is also a 1–byte, 2–cycle instruction. However, the execution timing of this unique in that a Data Memory location is accessed during the execution of the instruction. This access takes place during the time period from S4 of the first cycle through S3 of the second cycle. If the access is made from Data Memory mapped on the Expanded Bus, then ports P0, P2, and pins P3.6 and P3.7 will automatically be enabled and the read or write operation will take place on external memory. If the access is made from Data Memory space which is mapped within the Byte–wide RAM, then the read or write operation will take place on the Byte–wide RAM bus and the external port pins will not be affected.

BYTE–WIDE RAM INSTRUCTION EXECUTION TIMING Figure 15–3



EXPANDED PROGRAM MEMORY TIMING

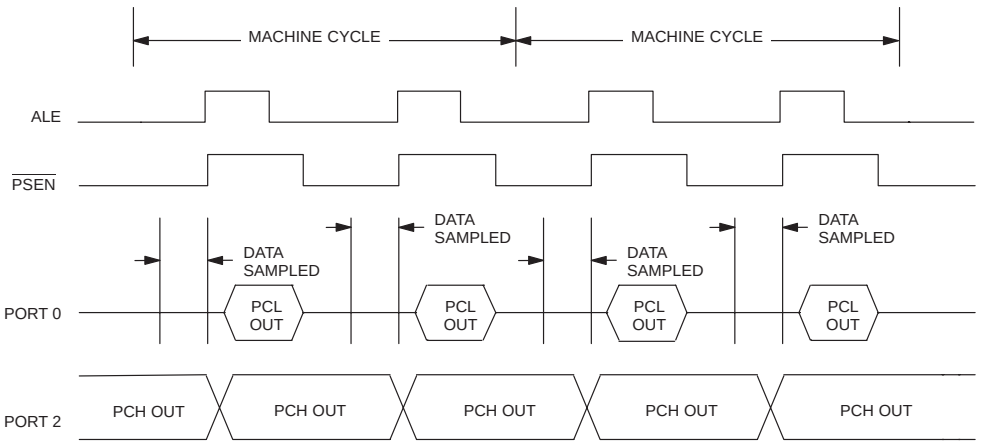
A Program Memory access will occur on the Expanded Bus any time that instructions are executed from Program Memory space which is mapped outside of the Byte–wide RAM. Mapping of Program Memory on the Expanded Bus is dependent on the programming of the Partition, Range, the state of the external $\overline{EA}$ pin, and the internal Security Lock. Refer to Section 4 for a detailed discussion on Program Memory mapping.

The external timing for the Expanded Program Memory fetch cycle is illustrated in Figure 15–4. A full 16–bit address is always output on the multiplexed Expanded Bus (P2, P0) pins whenever such an access is performed. The high–order eight bits will be output on the P2 pins while the low–order eight bits will be output on the P0 pins. Strong pull–ups are enabled onto Ports 0 and 2 for the duration of time that 1's are output on the port for address bits. As long as Program Memory is being executed from the Expanded Bus, P0 and P2 pins are unavailable for use as general–purpose I/O.

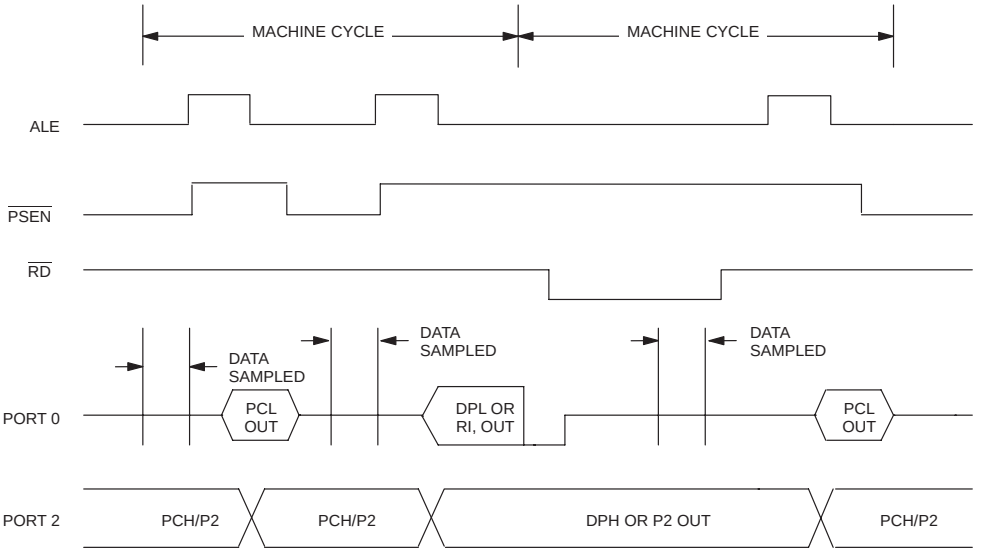
Multiplexed address and data information appear on the Port 0 pins as Program Memory fetches are performed on the Expanded Bus. The falling edge of ALE can be used to signal when the lowest eight bits of valid address information are being output on Port 0 when such a fetch occurs. In addition, ALE is activated twice every machine cycle during access to Program Memory, regardless of whether the fetch takes place to RAM or to the Expanded Bus. Whenever a Program Memory fetch takes place on the Expanded Bus, the SFR latch for Port 0 is written with all 1's (0FFH) so that the original information contained in this register is lost. Port 0 pins are driven with internal buffers when 1's are output during Expanded Program Memory cycles.

The $\overline{\text{PSEN}}$ signal is provided as the read strobe pulse for Expanded Program Memory fetches. When the Secure Microcontroller is accessing Program Memory from Byte–wide RAM, $\overline{\text{PSEN}}$ will remain inactive. During Program Memory fetches on the Expanded Bus, it is activated twice every machine cycle, except when a MOVX instruction is being executed. As discussed in the previous section, not all bytes fetched from Expanded Program Memory are actually used by the CPU during instruction execution. A complete memory cycle, including the enabled and disabling of both ALE and $\overline{\text{PSEN}}$, takes six clock oscillator periods. This is one–half of a machine cycle.

EXPANDED PROGRAM MEMORY FETCH Figure 15–4

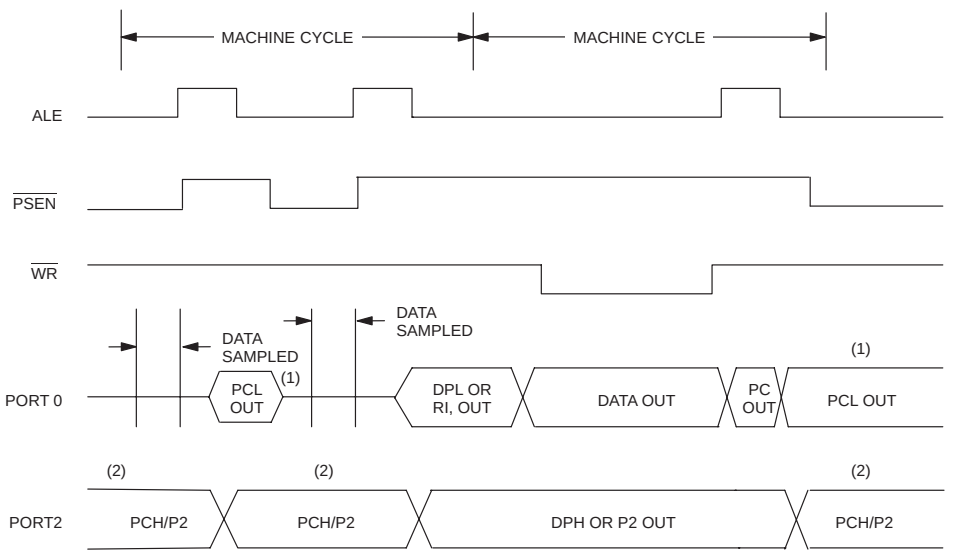


EXPANDED DATA MEMORY READ Figure 15–5



* PCL OUT if program memory also on Expanded Bus – float if not.
**PCH OUT if program memory also on Expanded Bus.

EXPANDED DATA MEMORY WRITE Figure 15–6



(1) PCL OUT if program memory also on Expanded Bus – float if not.
(2) PCH OUT if program memory also on Expanded Bus.

EXPANDED DATA MEMORY TIMING

The timing for the Expanded Data Memory access cycle is illustrated in Figures 15–5 and 6. Accesses to Data Memory on the Expanded Bus will occur any time that a MOVX instruction is executed that references a Data Memory location that is mapped outside the area which has been assigned to the Expanded Bus via the Partition and Range.

When a MOVX instruction is used with the Data Pointer register (e.g., MOVX @DPTR) to access a Data Memory location on the Expanded Bus, a full 16–bit address will be generated to the external memory. The 16–bit address is generated on P2 and P0 which are the same pins as for a Program Memory fetch from Expanded Memory. The contents of the SFR latch for Port 2 will not be modified, however, during the execution of a Data Memory fetch cycle on the Expanded Bus. If the MOVX instruction is not followed by another instruction requiring a cycle on the Expanded Bus, then the original contents of the Port 2 SFR latch will appear once again during the next machine cycle.

Multiplexed address/data information is output on Port 0 during the execution of a Data Memory cycle on the Expanded Bus. The falling edge of ALE can be used to latch the lower eight bits of address information into an external transparent latch (e.g., 74LS373 or equivalent). During the second cycle of a MOVX instruction, the first ALE pulse will not be generated so that valid address information will remain in the latch and be presented to the external memory device for the duration of the cycle. Port 0 is written with all 1's (0FFH) so that the original information contained in this register is lost. Also, Port 0 pins are driven with internal buffers when 1's are output during Expanded Data Memory cycles.

When a MOVX instruction is used with an indirect register address (e.g., MOVX @R0) for the same purpose, only an 8–bit address will be generated for the current instruction. This 8–bit address will appear on Port 0, while the contents of the SFR latch for Port 2 will remain on Port 2.

When data is to be read from Data Memory on the Expanded Bus, the external $\overline{RD}$ pin will be activated during the second machine cycle of the MOVX instruction. A complete $\overline{RD}$ cycle, including activation of ALE and $\overline{RD}$, takes twelve clock oscillator periods. $\overline{PSEN}$ is inactive during this machine cycle. This cycle is illustrated in Figure 15–5. When the MOVX instruction specifies a write operation to the external memory device, the $\overline{WR}$ signal will be activated as shown in Figure 15–6. Data is output on Port 0 just before $\overline{WR}$ is activated and remains valid until it goes back to its inactive level at the conclusion of the cycle.