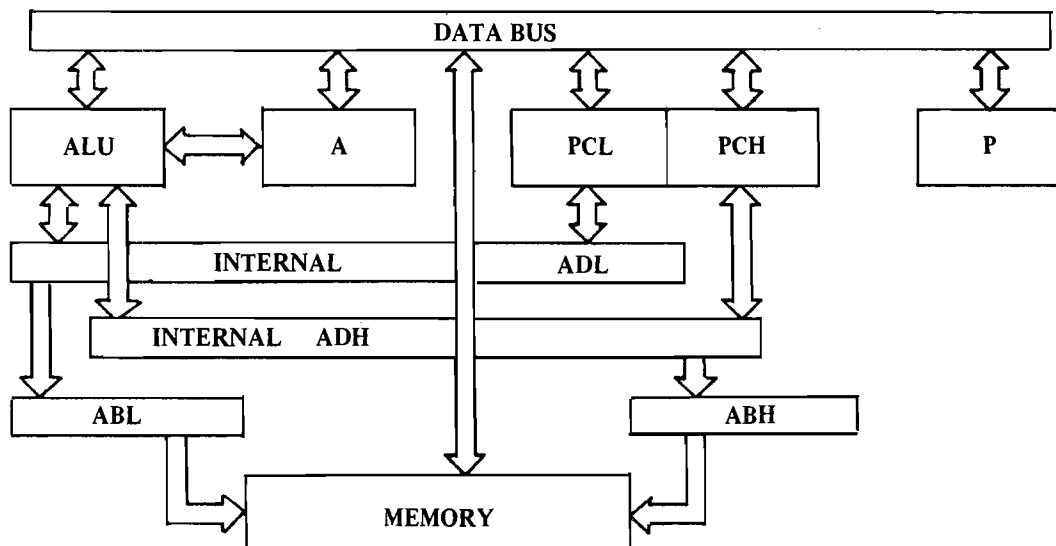


CHAPTER 4

TEST, BRANCH AND JUMP INSTRUCTIONS

4.0 CONCEPTS OF PROGRAM SEQUENCE

In all the discussions up until now, there has been little discussion about how the microprocessor understands the instructions used to perform various arithmetic and accumulator manipulations. However, it is appropriate that the concept of a program and how the microprocessor determines each instruction be developed. More registers are required in the machine as shown in the figure below.



*Partial Block Diagram of MCS650X Including Program
Counter and Internal Address Bus*

FIGURE 4.1

Although two 8 bit registers have been added, they are the only registers in the machine that act as though they are one 16 bit register. They implement a concept known as program count or program sequence and subsequently their value will be referred to as PC or program count. In certain operations it may be convenient to talk about how one affects the program count low (PCL) which will be the lower 8 bit register or the program count high (PCH) which will be the higher 8 bit register. The reason for this register being 16 bits in length is that if it had only 8 bits it would only be able to reference 256 locations. Since it is through the address bus that one accesses memory, the program counter which defines the addressable location, should be as wide a word as possible.

The accessing of a memory location is called "addressing". It is the selection of a particular eight-bit data word (byte) out of the 65,536 possibilities for memory data locations. This selection is transmitted to the memory through the 16 address lines (ADH, ADL) of the microprocessor.

For a more detailed discussion of how an individual memory byte is selected by the address lines, the reader is referred to Chapter 1 of the Hardware Manual.

If the program counter was only 1 byte and if the bit pattern which allows the microprocessor to choose which instruction it wants to act on next, such as "LDA" as opposed to an "AND", was contained in one byte of data we could only have 256 program steps. Although the machine of this length might make an interesting toy, it would have no real practical value. Therefore, almost all of the competitive 8 bit microprocessors have chosen to go to a double length program counter. Even though some of the microprocessors of the MCS650X family do not have all of the output address lines necessary to allow the user to address 65K bytes of program (due to package pinout constraints), in all cases the program counter is capable of addressing a full 65K by virtue of it's 16 bit length.

4.0.1 Use of Program Counter to Fetch an Instruction

The microprocessor contains an internal timing and state control counter. This counter, along with a decode matrix, governs the operation of the microprocessor on each clock cycle. When the state of the microprocessor indicates that a new instruction is needed, the program counter (program address pointer) is used to choose (address) the next memory location and the value which the memory sends back is decoded in order to determine what operation the MCS650X is going to perform next.

To use the program counter to perform this operation correctly, it must always be addressing the operation the user wants to perform next. This operation may be an instruction or may be data on which the instruction will operate.

In the MCS650X family, the program counter is set with the value of the address of an instruction. The microprocessor then puts the value of the program counter onto the address bus, transferring the 8 bits of data at that memory address into the instruction decode. The program counter then automatically increments by one and the microprocessor fetches further data for address operation necessary to complete the instruction. In the simple example below,

Example 4.1: Accessing Instructions with the P Counter Value

<u>P Counter*</u>	<u>Location</u>	<u>Contents</u>
0100**	LDA	*Program Counter
0101	ADC	**Hexadecimal
0102	STA	Notation

one can see how the program counter is used to access the instruction sequence load A, add with carry, and store the result. In this example, the program counter would start out containing 0100. The microprocessor would read location 0100 by using the program counter to access memory and would then interpret and implement the LDA instruction as previously described. The program counter will automatically increment by one on each instruction fetch, stepping to 0101. After performing the LDA, the microprocessor would fetch the

next instruction addressing memory with the program counter. This would pick up the ADC instruction, the add would then be performed, the program counter which has been incremented to 0102 would be used to address the next instruction, STA. The P counter incrementing once with each instruction is an oversimplified view of what actually transpires within the microprocessor.

The MCS650X processors usually require more than one byte to correctly interpret an instruction. The first byte of an instruction is called the OP CODE and is coded to contain the basic operation such as LDA (load accumulator with memory) and also the data necessary to allow the microprocessor to interpret the address of the data on which the operation will occur. In most cases, this address will appear in memory right after the OP CODE byte. This allows the microprocessor to use the program counter to access the address as well as the OP CODE.

The following example shows how the program counter picks up the instruction and the address of data located at address 5155.

Example 4.2: Accessing Data Address With P Counter Value

<u>P Counter</u>	<u>Location Contents</u>
0100	LDA
0101	55
0102	51
0103	Next Instruction

The OP CODE appears in Location Address 0100. The code for the 55 would appear next in Location Address 0101 and the 51 would appear in Location Address 0102, and the OP CODE for the next instruction appears in Location Address 0103. In this example, we see that the program counter is used not only to pick up the operation code, LDA, but is also used to pick up the address of the memory location from which the LDA is going to obtain its data. In this case, the program counter automatically is incremented three times to pick up the full instruction with the microprocessor interpreting each of the individual fetches as the appropriate data. In other words, the first