
Chapter 6.1 Addressing Modes

Objectives of this Chapter

Having studied this chapter you will be able to:

- Describe the operation of the 80286 Addressing Modes:
 - Implied
 - Immediate
 - Register
 - Direct
 - Register Indirect Modes
- Write programs which use these addressing modes.
- Describe the operation of 80286 Multiply and Divide Instructions
- Write programs which use these instructions.

Introduction

Addressing is concerned with the way in which operands are specified. You have already used a number of addressing modes.

This chapter will formalize what you have already learned about addressing modes and will also introduce some new addressing modes. The 80286 has a number of basic modes of addressing. These can be combined to extend the types of addressing possible.

80286 Instruction Format

You will recall that the order in which the registers/data/addresses are written specifies the direction of data transfer thus:

MOV destination, source

The **source** is the location in memory or register from which the data is to be fetched, while the **destination** is the location or register to which the data will be sent. So the instruction "MOV CX, AX" means "move the 16 bit word from the AX Register to the CX Register".

The D2000 Cross Assembler accepts binary, octal, decimal and hexadecimal numbers. The type of number is specified by a suffix character:

Suffix	Type
B	Binary
Q	Octal
None	Decimal
H	Hexadecimal

For example:

MOV AL,10100110B	;Moves the binary value ;101000110 ₂ into the AL ;Register
MOV BH,27Q	;Moves the octal value 27 ₈ into ;the BH Register
MOV CX,1234	;Moves the decimal value 1234 ₁₀ ;into location the CX Register
MOV DX,5678H	;Moves the hexadecimal value ;5678 _H into the DX Register

Implied Addressing

In this addressing mode the operand is **implied** by the operator and so need not be explicitly stated. The "Halt" instruction (HLT), which halts the processor, is an example of this mode.

Immediate Addressing

In immediate addressing, the operand is explicitly stated as a value. This can be a byte or a word. In this mode the data is **immediately** specified, without the need to examine a register/memory location.

Examples:

MOV	AL,21H	;Moves the value 21 _H into the ;AL Register
ADD	AL,43H	;Adds the value 43 _H to the AL ;Register
SUB	AL,32H	;Subtracts the value 32 _H from ;the AL Register

Register Addressing

In register addressing the operand is within one of the 80286 Registers.

Examples:

MOV	AL,BL	;Moves the contents of the BL ;Register into the AL Register
ADD	DX,CX	;Adds the contents of the CX ;Register to the DX Register
SUB	AL,DL	;Subtracts the contents of the ;DL Register from the AL ;Register

Direct Addressing

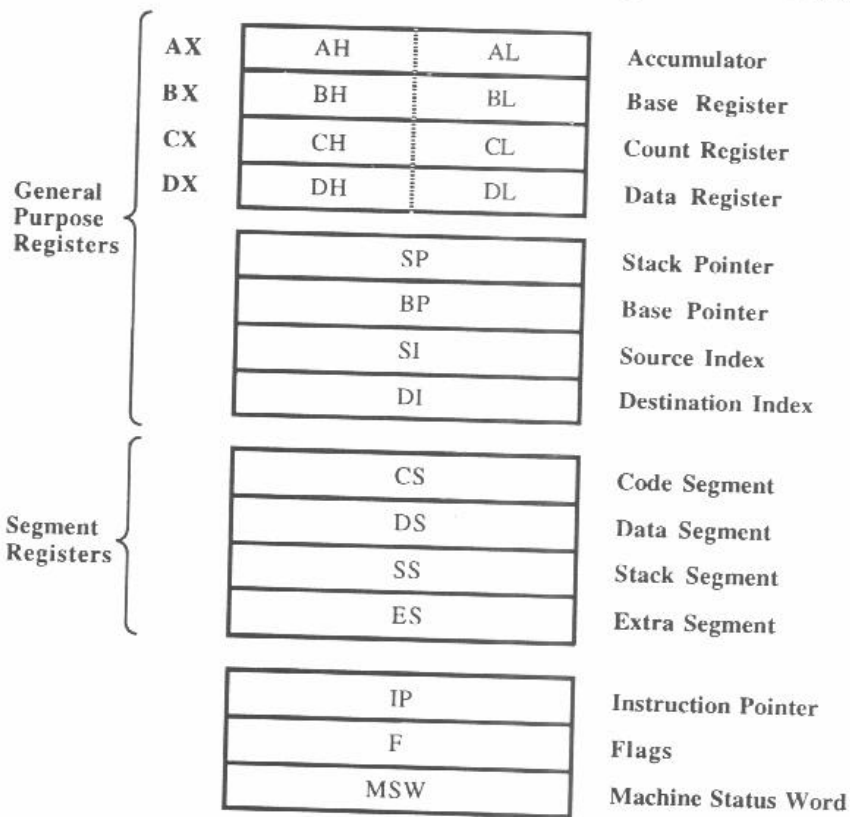
This mode of addressing specifies a memory location **directly**. This location holds the data to be acted upon.

For example:

MOV	DS:1000H,AX	;Moves the value in the AX ;Register into location 1000H ;within the current data ;segment
ADD	AX,DS:1100H	;Adds the value in location ;1100H, within the current data ;segment, to the AX Register
SUB	BYTE PTR DS:1200H,20H	;Subtracts the value 20 _H from ;the contents of location 1200H ;within current data segment

80286 Registers

We have already used the General Purpose Registers (AX, BX, CX and DX). Each of these is also referred to by the name: **Accumulator** (AX), **Base** (BX), **Count** (CX) and **Data** (DX). These names indicate the special functions of each register. There are several other 80286 registers associated with more complex addressing modes and operations. The 80286 Registers are shown below:



Notice that the General Purpose Registers also include two Pointer and two Index Registers. The function of these will be explained later in this text. We have already used the Data Segment to define our 'Current Segment'. Three other segment registers are also available. These can be used to define other segments of memory.

Register Indirect Addressing

In Register Indirect Addressing, the location of data is "pointed to" by a register. The

For example:

```
MOV DI,2000H           ;Moves the value 2000H into the
                       ;Destination Index Register
MOV [DI],AX           ;Moves the value in the
                       ;Accumulator into location
                       ;2000H within the current
```

In 80286 assembly language, brackets ([]) indicate **indirect** addressing.

The DI, SI, BX or BP Registers can all be used in this way.

It is also possible to add to the Base or Index Register within the instruction.

For example:

```
MOV SI,3000H          ;Moves the value 3000H into the
                       ;Source Index Register
MOV AX,[SI]           ;Moves the value in location
                       ;2000H + 12H = 2012H within the
                       ;current segment, into the
                       ;Accumulator
```

Notes:

.....

.....

.....

.....

.....

.....

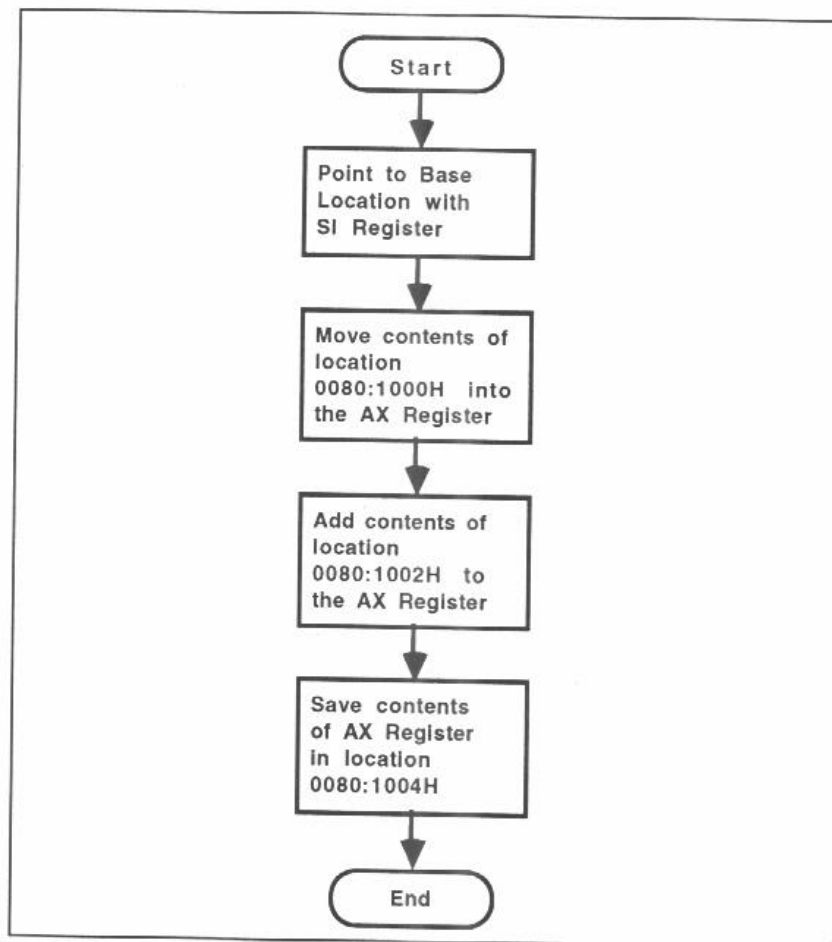
.....

Exercise 7

Write a program, using Register Indirect and other addressing modes which will add the words at locations 0080:1000H and 0080:1002H, saving the result in long word location 0080:1004H.

Solution:

Flowchart:



The assembly language source program will be:

```

ORG 0100H                ;Defines the start address for
                        ;object code as 0080:0100H
MOV SI,1000H            ;Point to base location
                        ;0080:1000H with SI Register
MOV AX,[SI]             ;Moves the contents of the
                        ;first location into the
                        ;Accumulator
ADD AX,[SI+2]           ;Adds the contents of the
                        ;second value to the
                        ;Accumulator
MOV [SI+4],AX           ;Saves the result in the third
                        ;memory location
MOV BX,0000H            ;Returns to PAT system
MOV AH,04H
INT 28H

```

Create a source program, using the Merlin Text Editor and assemble the object program. Transfer the object program to the PAT and load locations 0080:1000_H and 0080:1002_H with known values. Run the program and verify correct operation by examination of locations 0080:1000_H, 0080:1002_H and 0080:1004_H before and after program execution.

Register Indirect with Displacement

In this mode, a constant value (**displacement**) is added to the Base or Index Register to form the actual address of the data.

For example:

```

COUNT EQU 04C2H        ;Defines the label "COUNT" as
                        ;the value 04C2H
MOV [DI],5000H          ;Moves the value 5000H into the
                        ;DI Register
MOV AX,COUNT [DI]       ;Moves the value in location
                        ;5000H + 04C2H = 54C2H within
                        ;current segment, into the
                        ;Accumulator

```

Register Indirect with Base Register and Index Register

This mode of addressing uses the sum of two Registers to produce the actual address of the data.

For example:

MOV DI,2000H	;Moves the value 2000 _H into the ;DI Register
MOV BX,3000H	;Moves the value 3000 _H into the ;BX Register
MOV AX,9876H	;Moves the value 9876 _H into ;the Accumulator
MOV [BX+DI],AX	;Moves the value in the ;Accumulator into location ;2000 _H + 3000 _H = 5000 _H within ;current segment

Register Indirect with Base Register, Index Register and Constant

This is an extension of the previous mode of addressing. A **constant** value is added to the sum of two Registers to produce the final address of the data.

For example:

COUNT EQU 0A32H	;Defines the label "COUNT" as ;the value 0A32 _H
MOV DI,1000H	;Moves the value 1000 _H into the ;DI Register
MOV BX,2000H	;Moves the value 2000 _H into the ;BX Register
MOV AX,5678H	;Moves the value 5678 _H into ;the Accumulator
MOV COUNT[BX+DI],AX	;Moves the value in the ;Accumulator into location ;1000 _H + 2000 _H + 0A32 _H = 3A32 _H ;within current segment

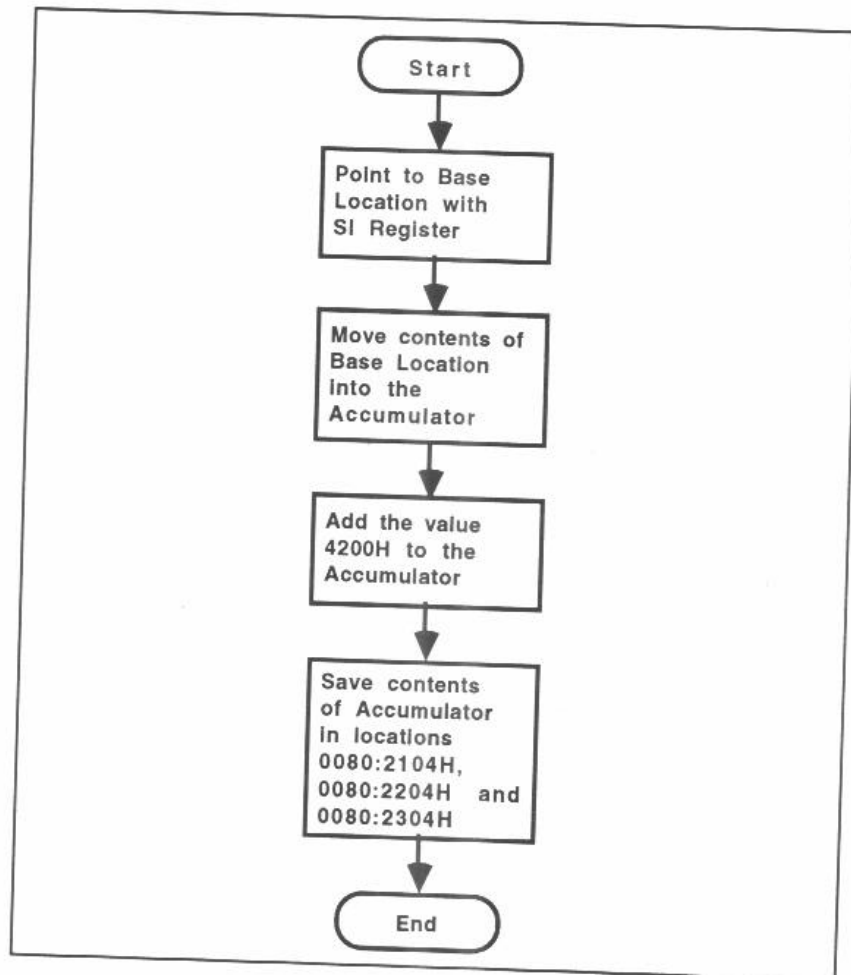
Exercise 8

Write a program which will add 4200_H to the word at locations 0080:2000_H and save the result in locations 0080:2100_H, 0080:2200_H and 0080:2300_H.

Solution:

Here we have an opportunity to use both Registers and a constant in addressing the location of data.

Flowchart:



The assembly language source program will be:

```

ORG 0200H                ;Defines the start address for
                        ;object code as 0080:0200H
MOV SI,2000H            ;Point to Base Location with SI
                        ;Register
MOV AX,[SI]             ;Moves contents of Base
                        ;Location into the Accumulator
ADD AX,4200H           ;Adds the value 4200H to the
                        ;Accumulator
MOV 0100H[SI],AX       ;Saves the contents of the
                        ;Accumulator into the first
                        ;location
MOV 0200H[SI],AX       ;Saves the contents of the
                        ;Accumulator into the second
                        ;location
MOV 0300H[SI],AX       ;Saves the contents of the
                        ;Accumulator into the third
                        ;location

MOV BX,0000H           ;Returns to PAT system
MOV AH,04H
INT 28H

```

Produce this source program and then assemble. Transfer the object program to the PAT and load locations 0080:2000H, 0080:2100H, 0080:2200H and 0080:2300H with known values.

Run the program and verify correct operation as before by examination of relevant memory locations before and after program execution.

Practical Assignments

- 3 Write a program which will copy the data word in each Register shown into the memory location indicated:

Register	Location
AX	0080:3000H
BX	0080:3002H
CX	0080:3004H
DX	0080:3006H

- 4 The contents of byte locations 0080:4000H, 0080:4001H, 0080:4002H and 0080:4003H are to be added and the result saved in location 0080:4004H. Write a program which will perform this task.

Multiplication Using the 80286

The Multiply instruction (MUL) multiplies the contents of a Register or memory location and the AX or AL Registers.

For example:

```
MUL  BX                ;Multiplies the contents of the  
                        ;AX and BX registers, placing  
                        ;the result in the AX register
```

Division Using the 80286

The Divide instruction (DIV) divides the contents of the Accumulator by a source operand.

For example:

```
DIV  BL                ;Divides the byte in the  
                        ;Accumulator by the BL  
                        ;register, placing the quotient  
                        ;in the AL register and the  
                        ;remainder in the AH register  
  
DIV  CX                ;Divides the word in the  
                        ;Accumulator by the CX  
                        ;register, placing the quotient  
                        ;in the AX register and the  
                        ;remainder in the DX register
```

Practical Assignment

- 5 Write a program which will calculate:

$E7_{16} \times C3_{16}$
 $B9_{16}$

The quotient should be saved in location 0080:3000H and any remainder in location 0080:3002H.

Student Assessment 6

1. The 80286 addressing mode in which the data to be acted upon is explicitly stated as a value is called
 - a implied addressing
 - b immediate addressing
 - c register addressing
 - d direct addressing
2. The instruction "MOV AX, [SI]" will place in the Accumulator the data
 - a within the AX Register
 - b within the SI Register
 - c pointed to by the AX Register
 - d pointed to by the SI Register
3. The addressing mode used in the instruction "MOV 0215[DI], BX" is
 - a Register Indirect
 - b Register Indirect with Displacement
 - c Register Indirect with Base and Index Registers
 - d Register Indirect with Base and Index Registers and Constant
4. If the SI Register contains 1200_H and the BX Register is 3020_H, then the instruction "MOV DX, 2400_H[BX+SI]" will load the DX Register from location
 - a 2400_H
 - b 3020_H
 - c 4220_H
 - d 6620_H
5. When dividing byte data, the 80286 will place the quotient in the
 - a AX Register
 - b AH Register
 - c AL Register
 - d DX Register