INTEL 8086

This chapter describes the internal architecture, addressing modes, instruction set, and I/O techniques associated with the 8086 microprocessor. Interfacing capabilities to typical memory and I/O chips such as the 2716, 6116, and 8255 are included.

A design technique is presented showing interconnection of the 8086 to 2716 EPROM, 6116 RAM, and 8255 I/O chips. The memory and I/O maps are then determined.

3.1 Introduction

The 8086 is Intel's first 16-bit microprocessor.

The 8086 is designed using the HMOS technology and contains approximately 29,000 transistors. The 80C86A is the low power version of the 8086 designed using HCMOS technology. The 8086 is packaged in a 40-pin CERDIP or plastic package and requires a single 5V power supply. The 8086 can be operated at three different clock speeds. The standard 8086 runs at 5 MHz internal clock frequency, whereas the 8086-1 and 8086-2 run at internal clock frequencies of 10 and 8 MHz, respectively. An external clock generator/driver chip such as the Intel 8284 is needed to generate the 8086 clock input signal. The 8284 divides the external crystal input internally by three. This means that for a 5-MHz 8086 internal clock, the 8284's X1 and X2 pins must be connected to a 15-MHz crystal. The 8284 will then generate a 5 MHz clock at its CLK pin which should be connected to the 8086 CLK input.

The 8086 has a 20-bit address and, hence, it can directly address up to one megabyte (2^{20}) of memory. The 8086 uses a segmented memory. An interesting feature of the 8086 is that it prefetches up to six instruction bytes from memory and queues them in order to speed up instruction execution.

There are some advantages of working with the segmented memory. First of all, after initializing the 16-bit segment registers, the 8086 has to deal with only 16-bit effective addresses. That is, the 8086 has to manipulate and store 16-bit address components. Secondly, because of memory segmentation, the 8086 can be effectively used in time-shared systems. For example, in a time-shared system, several users share a microprocessor. The microprocessor works with one user's program for say, 10 milliseconds. After spending 10 milliseconds with each of the other users, the microprocessor returns to execute the first user's program. Each time the microprocessor switches from one user's program to the next, it must execute a new section of code and new sections of data. Segmentation makes it easy to switch from one user program (and data) to another.

The memory of an 8086-based microcomputer is organized as bytes. Each byte can be uniquely addressed with 20-bit addresses of 00000_{16}, 00001_{16}, 00002_{16} ..., $FFFFF_{16}$. An 8086 16-

bit word consists of any two consecutive bytes; the low-addressed byte is the low byte of the word and the high-addressed byte contains the high byte as follows:

16bit 8bit → 8bit

Low byte of the word	High byte of the word
07_{16}	26_{16}
Address 00520_{16} even	Address 00521_{16} (20bit) odd

The 16-bit word stored at the even address 00520_{16} is 2607_{16}.

Next consider a word stored at an odd address as follows:

Low byte of the word	High byte of the word
05_{16}	$3F_{16}$
Address 01257_{16}	Address 01258_{16}

The 16-bit word stored at the odd address 01257_{16} is $3F05_{16}$. Note that for word addresses, the programmer uses the low-order address (odd or even) to specify the whole 16-bit word.

The 8086 always accesses a 16-bit word to or from memory. The 8086 can read a 16-bit word in one operation if the first byte of the word is at an even address. On the other hand, the 8086 must perform two memory accesses to two consecutive memory even addresses, if the first byte of the word is at an odd address. In this case, the 8086 discards the unwanted bytes of each. For example, consider MOV BX, [ADDR]. Note that the X or H (or L) following the 8086 register name in an instruction indicates whether the transfer is 16-bit (for X) or 8-bit (for H or L). The instruction, MOV BX, [ADDR] moves the contents of a memory location addressed by ADDR into the 8086 16-bit register BX. Now, if ADDR along with the data segment register provides a 20-bit even address such as 30024_{16}, then this MOV instruction loads the low (BL) and high (BH) bytes of the 8086 16-bit register BX with the contents of memory locations 30024_{16} and 30025_{16}, respectively, in a single access. Now, if ADDR is an odd address such as 40005_{16}, then the MOV BX, [ADDR] instruction loads BL and BH with the contents of memory locations 40005_{16} and 40006_{16}, respectively, in two accesses. Note that the 8086 accesses locations 40004_{16} and 40005_{16} in the first operation but discards the contents of 40004_{16}, and in the second operation accesses 40006_{16} and 40007_{16} but ignores the contents of 40007_{16}.

Next, consider a byte move such as MOV BH, [ADDR]. If ADDR is an even address such as 50002_{16}, then this MOV instruction accesses both 50002_{16} and 50003_{16}, but loads BH with the contents of 50002_{16} and ignores the contents of 50003_{16}. However, if ADDR is an odd address such as 50003_{16}, then this MOV loads BH with the contents of 50003_{16} and ignores the contents of 50002_{16}.

The 8086 family consists of two types of 16-bit microprocessors — the 8086 and 8088. The 8088 has an 8-bit external data path to memory and I/O, while the 8086 has a 16-bit external data path. This means that the 8088 will have to do two read operations to read a 16-bit word into memory. In most other respects, the processors are identical. Note that the 8088 accesses memory in bytes. No alterations are needed to run software written for one microprocessor on the other. Because of similarities, only the 8086 will be considered here. The 8088 was used in designing the original IBM Personal computer.

An 8086 can be configured as a small uniprocessor system (minimum mode if the MN/MX pin is tied to HIGH) or as a multiprocessor system (maximum mode when MN/MX pin is tied to LOW). In a given system, the MN/MX pin is permanently tied to either HIGH or LOW. Some of the 8086 pins have dual functions depending on the selection of the MN/MX pin level. In the minimum mode (MN/MX pin high), these pins transfer control signals directly to memory and input/output devices. In the maximum mode (MN/$\overline{MX}$ pin low), these same pins have different functions which facilitate multiprocessor systems. In the maximum mode, the control functions normally present in minimum mode are performed by a support chip, the 8288 bus controller.

Due to technological advances, Intel introduced the high performance 80186 and 80188 which are enhanced versions of the 8086 and 8088, respectively. The 8-MHz 80186/80188 provides two times greater throughput than the standard 5-MHz 8086/8088. Both have integrated several new peripheral functional units such as a DMA controller, a 16-bit timer unit, and an interrupt controller unit into a single chip. Just like the 8086 and 8088, the 80186 has a 16-bit data bus and the 80188 has an 8-bit data bus; otherwise, the architecture and instruction set of the 80186 and 80188 are identical. The 80186/80188 has an on-chip clock generator so that only an external crystal is required to generate the clock. The 80186/80188 can operate at 6 MHz, 8 MHz, and other frequencies. Like the 8085, the crystal frequency is divided by 2 internally. In other words, external crystals of 12 or 16 MHz must be connected to generate the 6- or 8-MHz internal clock frequency. The 80186/80188 is fabricated in a 68-pin package. Both processors have on-chip priority interrupt controller circuits to provide five interrupt pins. Like the 8086/8088, the 80186/80188 can directly address one megabyte of memory. The 80186/80188 is provided with 10 new instructions beyond the 8086/8088 instruction set. Examples of these instructions include INS and OUTS for inputting and outputting string byte or string word. The 80286, on the other hand, has added memory protection and management capabilities to the basic 8086 architecture. An 8-MHz 80286 provides up to six times greater throughput than the 5-MHz 8086. The 80286 is fabricated in a 68-pin package. The 80286 can be operated at 6, 8, 10, 12.5, 16.67, or 20 MHz clock frequency. The 80286 is typically used in a multiuser or multitasking system. The 80286 was used as the CPU of the IBM PC/AT Personal computer. Intel's 32-bit microprocessor family includes 80386, 80486 and Pentium microprocessors which will be covered later in this book.

3.2 8086 Architecture

Figure 3.1 shows a block diagram of the 8086 internal architecture. As shown in the figure, the 8086 microprocessor is internally divided into two separate functional units. These are the Bus Interface Unit (BIU) and the Execution Unit (EU). The BIU fetches instructions, reads data from memory and ports, and writes data to memory and I/O ports. The EU executes instructions that have already been fetched by the BIU. The BIU and EU function independently. The BIU interfaces the 8086 to the outside world. The BIU contains segment registers, instruction pointer, instruction queue, and address generation/bus control circuitry to provide functions such as fetching and queuing of instructions, and bus control.

The BIU's instruction queue is a First-In-First-Out (FIFO) group of registers in which up to six bytes of instruction code are prefetched from memory ahead of time. This is done in order to speed up program execution by overlapping instruction fetch with execution. This mechanism is known as pipelining.

The bus control logic of the BIU generates all the bus control signals such as read and write signals for memory and I/O. The 8086 contains the on-chip logical address to physical address mapping hardware. The programmer works with the logical address which includes the 16-bit contents of a segment register and a 16-bit displacement or offset value. The 8086 on-chip mapping hardware translates this logical address to 20-bit physical address which it then generates on its twenty addressing pins.

The BIU has four 16-bit segment registers. These are the Code Segment (CS), the Data Segment (DS), the Stack Segment (SS), and the Extra Segment (ES). The 8086's one megabyte memory is divided into segments of up to 64K bytes each. The 8086 can directly address four segments (256K byte within the 1 Mbyte memory) at a particular time. Programs obtain access to code and data in the segments by changing the segment register contents to point to the

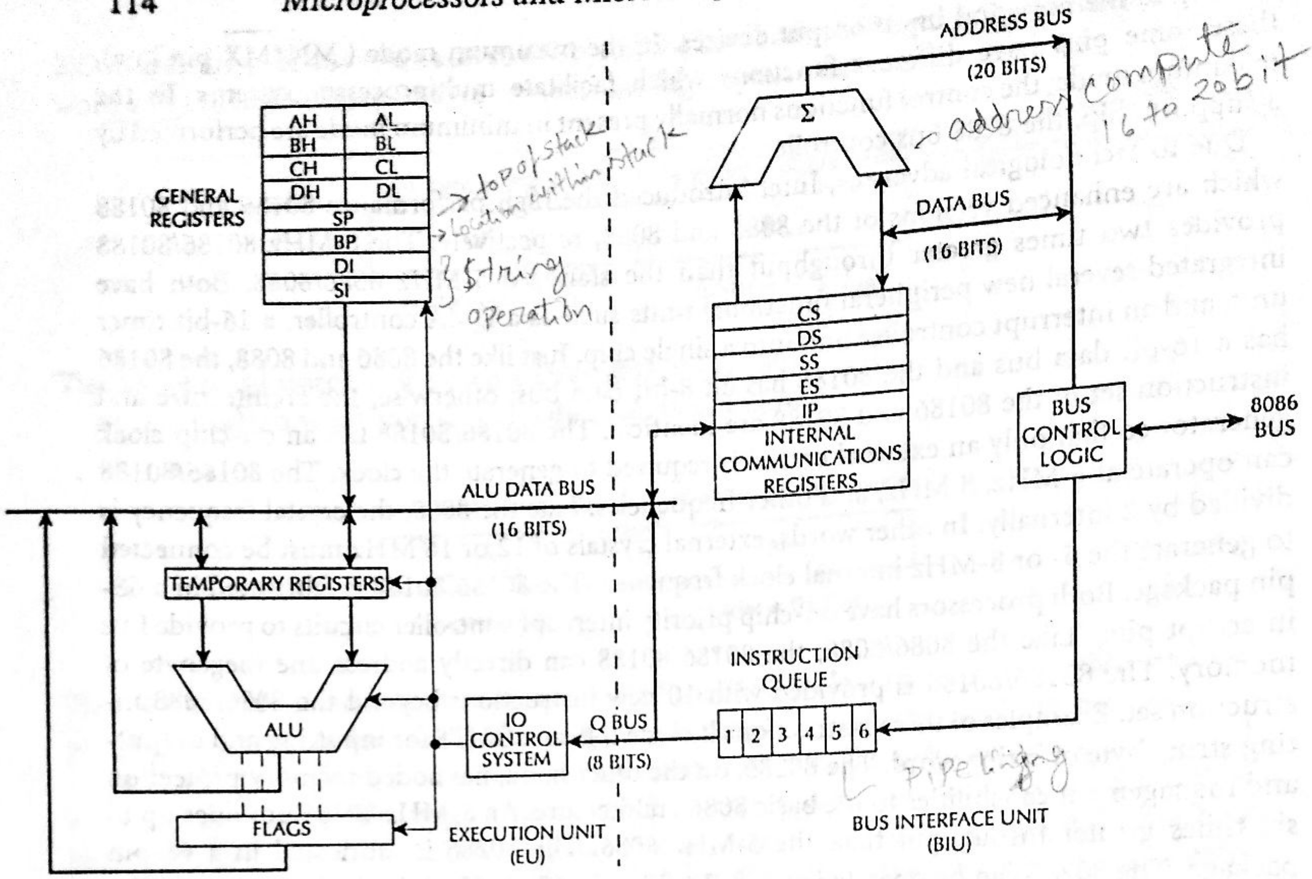


FIGURE 3.1 Internal architecture of the 8086.

desired segments. All program instructions must be located in main memory pointed to by the 16-bit CS register with a 16-bit offset in the segment contained in the 16-bit instruction pointer (IP). The BIU computes the 20-bit physical address internally using the programmer-provided logical address (16-bit contents of CS and IP) by logically shifting the contents of CS four bits to left and then adding the 16-bit contents of IP. For example, if $[CS] = 456A_{16}$ and $[IP] = 1620_{16}$, then the 20-bit physical address is generated by the BIU as follows:

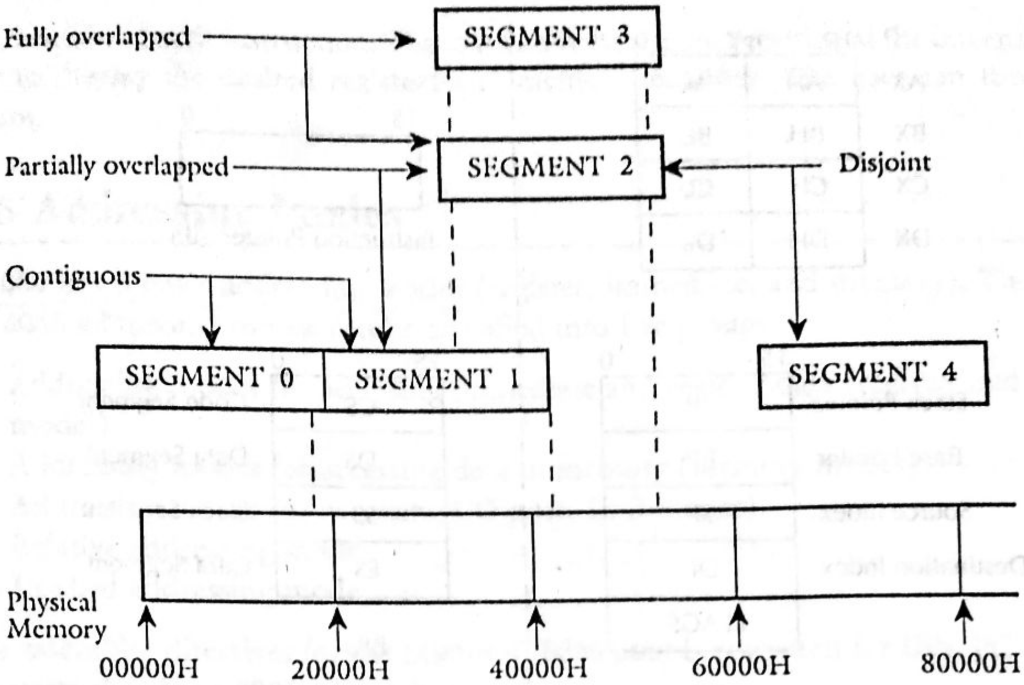
$$\begin{aligned} &\text{Four times log ically shifted [CS] to left} && = 456A0_{16} \\ &+[\text{IP}] \text{ as offset} && = 1620_{16} \\ &\text{20 - bit physical address} && = 46CC0_{16} \end{aligned}$$

The SS register points to the current stack. The 20-bit physical stack address is calculated from SS and SP for stack instructions such as PUSH and POP. The programmer can use the BP register instead of SP for accessing the stack using the based addressing mode. In this case, the 20-bit physical stack address is calculated from BP and SS.

The DS register points to the current data segment; operands for most instructions are fetched from this segment. A 16-bit offset (Effective Address, EA) along with the 16-bit contents of DS are used for computing the 20-bit physical address.

The ES register points to the extra segment in which data (in excess of 64K pointed to by DS) is stored. String instructions use ES and DI to determine the 20-bit physical address for the destination, and DS and SI for the source address.

The segments can be contiguous, partially overlapped, fully overlapped, or disjoint. An example of how five segments (segment 0 through segment 4) may be stored in physical memory are shown below:



In the above, SEGMENTs 0 and 1 are contiguous (adjacent), SEGMENTs 1 and 2 are partially overlapped, SEGMENTs 2 and 3 are fully overlapped, and SEGMENTs 2 and 4 are disjoint. Every segment must start on 16-byte memory boundaries.

Typical examples of values of segments should then be selected based on physical addresses starting at 00000_{16}, 00010_{16}, 00020_{16}, 00030_{16} ..., $FFFF0_{16}$. A physical memory location may be mapped into (contained in) one or more logical segments. Many applications can be written to simply initialize the segment registers and then forget them. One can then work with a 64K memory as with the 8085.

The EU decodes and executes instructions. A decoder in the EU control system translates instructions. The EU has a 16-bit ALU for performing arithmetic and logic operations.

The EU has eight 16-bit general registers. These are AX, BX, CX, DX, SP, BP, SI, and DI. The 16-bit registers AX, BX, CX, and DX can each be used as two 8-bit registers (AH, AL, BH, BL, CH, CL, DH, DL). For example, the 16-bit register DX can be considered as two 8-bit registers DH (high byte of DX) and DL (low byte of DX). The general-purpose registers AX, BX, CX, and DX are named after special functions carried out by each one of them. For example, the AX is called the 16-bit accumulator while the AL is the 8-bit accumulator. The use of accumulator registers is assumed by some instructions. The Input/Output (IN or OUT) instructions always use AX or AL for inputting/outputting 16- or 8-bit data to or from an I/O port.

Multiplication and division instructions also use AX or AL. The AL register is the same as the 8085 A register.

The BX register is called the base register. This is the only general-purpose register, the contents of which can be used for addressing 8086 memory. All memory references utilizing these register contents for addressing use DS as the default segment register. The BX register is similar to 8085 HL register. In other words, 8086 BH and BL are equivalent to 8085 H and L registers, respectively.

The CX register is known as the counter register. This is because some instructions such as shift, rotate, and loop instructions use the contents of CX as a counter. For example, the instruction LOOP START will automatically decrement CX by 1 without affecting flags and will check if [CX] = 0. If it is zero, the 8086 executes the next instruction; otherwise the 8086 branches to the label START.

The data register DX is used to hold high 16-bit result (data) in 16 × 16 multiplication or high 16-bit dividend (data) before a 32 ÷ 16 division and the 16-bit remainder after the division.

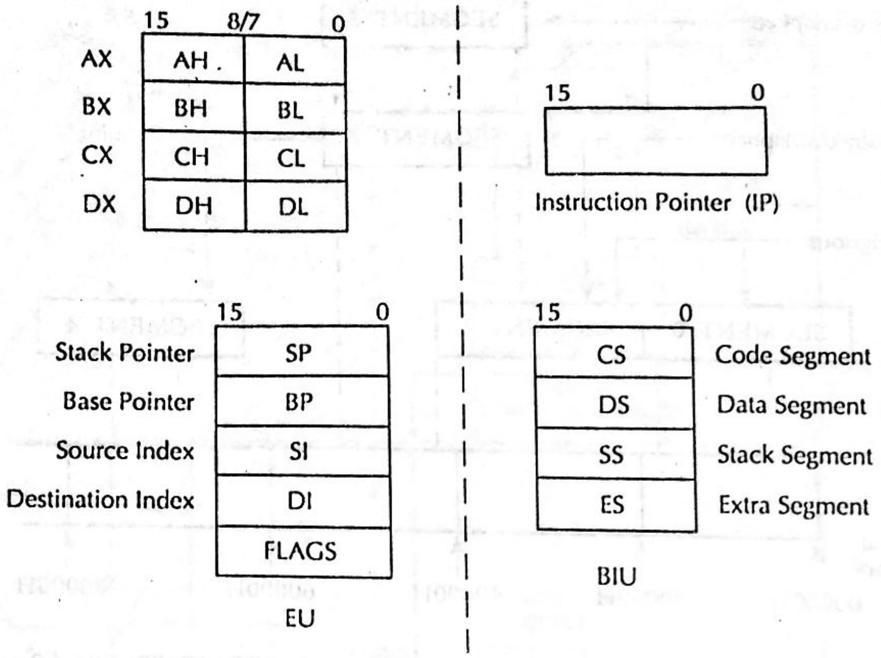


FIGURE 3.2 8086 registers.

The two pointer registers, SP (stack pointer) and BP (base pointer), are used to access data in stack segment. The SP is used as an offset from the current SS during execution of instructions that involve stack segment in external memory. The SP contents are automatically updated (incremented or decremented) due to execution of POP or PUSH instruction.

The base pointer contains an offset address in the current SS. This offset is used by the instructions utilizing the based addressing mode.

The FLAG register in the EU holds the status flags after an ALU operation. Figure 3.2 shows the 8086 registers.

The 8086 has six one-bit flags. Figure 3.3 shows the flag register. The AF (Auxiliary carry Flag) is used by BCD arithmetic instructions. AF = 1 if there is a carry from the low nibble (4-bit) into the high nibble or a borrow from the high nibble into the low nibble of the low-order 8-bit of a 16-bit number. The CF (Carry Flag) is set if there is a carry from addition or borrow from subtraction. The OF (Overflow Flag) is set if there is an arithmetic overflow, that is, if the size of the result exceeds the capacity of the destination location. An interrupt on overflow instruction is available which will generate an interrupt in this situation. The SF (Sign Flag) is set if the most significant bit of the result is one (negative) and is cleared to zero for non-negative result. The PF (Parity Flag) is set if the result has even parity; PF is zero for odd parity of the result. The ZF (Zero Flag) is set if result is zero; ZF is zero for nonzero result.

The 8086 has three control bits in the flag register which can be set or reset by the programmer: Setting DF (Direction Flag) to one causes string instructions to autodecrement the appropriate index register(s), and clearing DF to zero causes string instructions to autoincrement. Setting IF (Interrupt Flag) to one causes the 8086 to recognize external maskable interrupts; clearing IF to zero disables these interrupts. Setting TF (Trace Flag) to one places the 8086 in the single-step mode. In this mode, the 8086 generates an internal interrupt

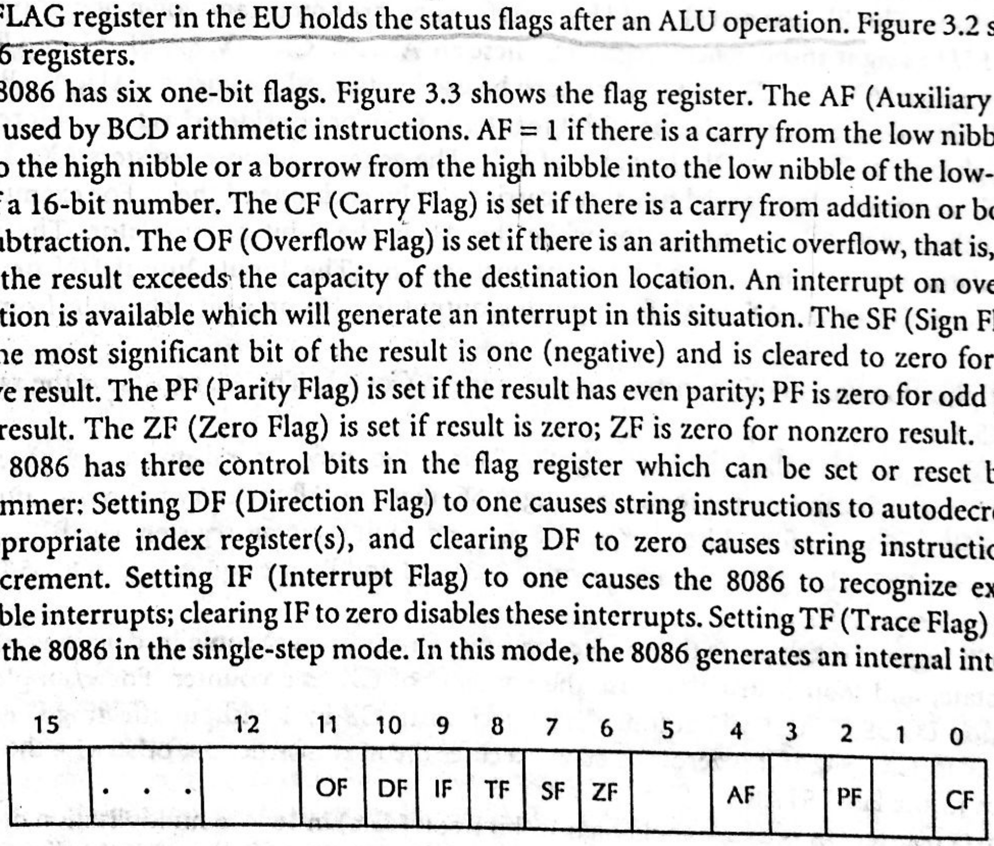


FIGURE 3.3 8086 flag register.