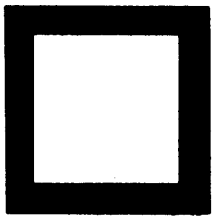


10.56 Explain the difference in an ADD accumulator (ABA) and an ADD (ADDA or ADDB) instruction for the 6800 microprocessor.

10.57 Explain a ROTATE RIGHT instruction on the 6800 microprocessor.

10.58 What are the differences between ROTATE and SHIFT instructions on the 6800 microprocessor? Why are both useful?

10.59 Explain the branch-to-subroutine and the return-from-subroutine instructions for the 6800 microprocessor. How would they be used to enter and then exit from a subroutine?

10.60 Contrast the JMS instruction in the 6100 with the JSR instruction in the 6800.

10.61 Explain the register indirect and immediate addressing modes for the 8080 microprocessor.

10.62 Explain how two of the arithmetic-type instructions for the 8080 affect the setting of the condition flags.

10.63 How do conditional JUMP instructions work in the 8080 microprocessor?

10.64 Explain the PUSH and POP instructions in the 8080 microprocessor.

10.65 Explain the CALL and RET instructions for the 8080 microprocessor. Compare these with the JSR and RTS instructions for the 6800 microprocessor.

10.66 Explain the STA and LDA instructions for the 8080 microprocessor.

10.67 Explain the ADI instruction for the 8080 microprocessor, and contrast it to the ADM instruction.

10.68 Compare the ADM instruction for the 8080 with the ADDA and ADDB instructions for the 6800.

10.69 Contrast the PUSH and POP instructions for the 8080 with the DES and INS instructions for the 6800.

10.70 Compare the BRANCH instructions for the 6800 with those for the 8080.

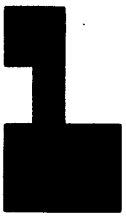
10.71 Write a program to service the printer interface in Chap. 8 for the 8080 microprocessor.

10.72 Show how the BPL instruction in Table 10.23 operates by calculating the value of the offset for the branch and adding it to the program counter to see whether the branch goes to the right place.

10.73 The plus before the indirect address in Table 10.23 for the MOVB instruction indexes through the table where index characters are to be stored. Explain how this is accomplished.

10.74 How did the assembler calculate the value for the TSTB instruction in Table 10.23?

10.75 Explain how the assembler calculated the binary translation of the BNE LOOP instruction in Table 10.18.



10.76 Modify Tables 10.19 and 10.20 so that the parameter being passed from calling program to subroutine is a pointer to the beginning of the table to be searched instead of the end of the table.

10.77 Explain how the BNE LOOP instruction operates in Table 10.18, calculating the value of the jump in the computer word and seeing whether it points to the right location in the memory.

10.78 Modify the program in Table 10.10 so that the number of characters in the table is passed by using register *D* instead of register *B*.

10.79 Modify the programs so that Table 10.11 passes the lowest table address in memory instead of the end of the table. This will require modification to both Tables 10.10 and 10.11.

10.80 Explain how the JZ instruction operates in Table 10.10.

10.81 Convert the program in Table 10.10 into binary as an assembler would.

10.82 For the 8080 or 6800 microprocessor, explain how you would pass parameters if two tables should both be searched for an input character and the start or end points of each table must be given to the subroutine as well as the character to be searched for and the number of characters in each table.

10.83 Explain how the bracket notation is used for the LDA instruction in the instruction repertoire table for the 8080 (Table 10.7).

10.84 Write a program like that in Table 10.9 with the aim to find the smallest number instead of the largest.

10.85 Modify the program in Table 10.2 so that it checks for all 1s in the character transferred as well as for periods, jumping to the same section of program if all 1s are found.

10.86 What is the character code for a period in the 6100?

10.87 Change the JMP I LISN instruction in Table 10.2 so that it causes a jump to location 254 in the memory and not to 252. Change memory contents also.

10.88 Explain how the index instruction in the 6800 can be used to sequence through two tables, each of which is located at a different position in the memory, but has the same number of elements.

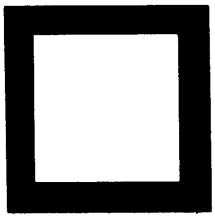
10.89 The MOVE instruction in the 8080 uses register pair *H, L* as a pointer to the memory. Explain how register pair *H, L* can be loaded.

10.90 Find two examples of relative addressing in the sample programs given for the computers in this chapter.

10.91 Compare paging to relative addressing as an address strategy.

10.92 Compare the pin-outs for the 8086 and 68000 microprocessors with regard to economy and interfacing considerations.

10.93 Explain how programs and data can be distributed through a memory by using the 8086 segment registers.



10.94 What are the advantages of having four segment registers versus one segment register in the 8086?

10.95 For the 68000 write in binary an AND I instruction which will AND the 01011110 with the word at location FF in memory. The value FF is in general register 3.

10.96 Write an ADD instruction in binary for the 68000 which will add two operands of your choice, explaining how the instruction works.

10.97 Explain the meaning of $(A_4)+$ in the 68000 register transfer language.

10.98 Discuss how the supervisor and user modes for the 68000 might be useful in a timeshared system.

10.99 Does a computer with I/O instructions and a bus provide all the facility of a memory-mapped system? Explain your answer.

APPENDIX A
CIRCUIT PRINCIPLES

The intent of this appendix is first to explain some of the circuit principles which relate to all digital circuits and then to examine, one at a time, the major circuit technologies. Basic facts about the general properties of each circuit technology are given, and some of the advantages and disadvantages of each type of circuit are pointed out.

Before we analyze logic circuits, the general characteristics of transistors and diodes are discussed. No attempt will be made to explain the physics of junction semiconductor devices. Semiconductor physics and a detailed analysis of the operating characteristics of junction devices are so interwoven with the manufacturing techniques, the geometry of the junctions, and many other considerations that the subject has grown into a highly specialized (and very fascinating) field, which is treated in detail in other books. For our purposes, and for the purposes of most users of circuit devices, the operational characteristics of these devices are important, and we limit our discussion to these operational characteristics.

The characteristics of diodes are first briefly noted. A semiconductor diode is made of two pieces of semiconductor material[1] of different types joined together. One type of semiconductor material is called *p-type* material, and the other *n-type* material [refer to Fig. A.1(*a*)].[2] When two differing types of semiconductor ma-

---

[1] A semiconductor has a conductivity roughly halfway between those of metals, which are good conductors, and insulators, which conduct poorly.

[2] *n*-type material has an excess of negative current carriers (electrons), while *p*-type material has an excess of positive current carriers (holes). Thus current flows readily when the *p*-type material is positive and the *n*-type material negative. Alternatively, very little current flows when the *p*-type material is negative and the *n*-type material is positive.

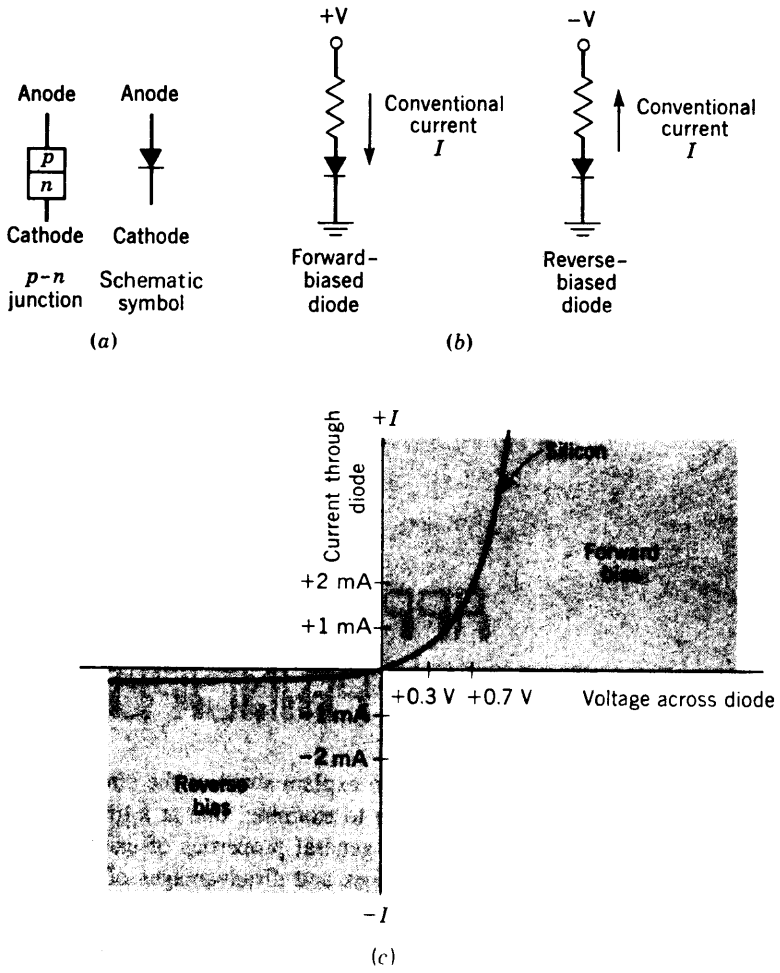


FIGURE A.1

Diode symbol and characteristics.

terial are joined, a *semiconductor junction* is formed, and a single junction is also called a *diode*. In Fig. A.1(*a*), the *p*-type material is referred to as the *anode* of the diode, and the *n*-type material is called the *cathode* of the diode. Figure A.1(*a*) also shows the schematic symbol for the diode.

A study of semiconductor, or solid-state, devices would explain the physical internal workings of the diode. However, our sole interest here is to view the diode as a component in electronic switching circuits; so we examine the diode from the viewpoint of its electrical characteristics only. When we apply an electric voltage, possibly through a resistor as in Fig. A.1(*b*), so that the anode of the diode is positive with respect to the cathode, the diode is said to be *forward-biased*. A diode which is forward-biased will conduct current rather freely. (*Conventional current* is used in our discussion; conventional current flows from positive to negative.)

Figure A.1(*c*) shows a typical characteristic curve for a diode. Notice that,

when forward-biased, the diode drops on the order of 0.7 V.[3] The forward-biased region on the graph lies to the right of the ordinate of the graph.

When the cathode of a diode is positive with respect to the anode, the diode is said to be *reverse-biased*, and it will present a very high resistance to current flow [refer to Fig. A.1(*b*) and (*c*)]. This ranges from tens to hundreds of megohms.[4]

A diode may, therefore, be thought of as a kind of electronic switch which is closed (freely passes current) when forward-biased and open (passes almost no current) when reverse-biased.

There are two general types of transistors used in computer circuits: bipolar and field-effect transistors (FETs). FETs are covered later. A *bipolar transistor* consists of either a piece of *n* material between two pieces of *p* material or a piece of *p* material between two pieces of *n* material. The first is called a *pnp* transistor, and the second an *npn* transistor. Figure A.2(*a*) and (*b*) shows this. The pieces of *n* and *p* material are named; for the *pnp* transistor the "middle" piece is *n* type and is called the *base*, while the two *p*-type pieces are called the *collector* and *emitter* and are further identified with the schematic symbols shown in Fig. A.2(*a*). For the *npn* transistor, the *p*-type material is the base, and the two pieces of *n*-type material are the emitter and collector.

Figure A.3 shows an *npn* transistor in a circuit with the emitter grounded. The currents and voltages in the circuit are identified as follows: The current *into* the base of the transistor is called I_b; the current *into* the collector is called I_c; the current *into* the emitter is called I_e. The voltage of the collector is called V_{ce} (this is the voltage between the collector and the emitter); the voltage at the base is called V_{be} (the voltage between base and emitter).

In this configuration the transistor can be operated in three modes: (1) *active region*, (2) *saturated*, and (3) *cutoff*. First let us define these terms, and then we examine the transistor's characteristics in each region of operation.

[3]Provided enough current flows.

[4]The part of the curve for silicon diodes which is below the 0 current level is somewhat distorted since a diode actually passes less than a few hundred nanoamperes when reverse-biased (1 nA = 10^{-9} A).

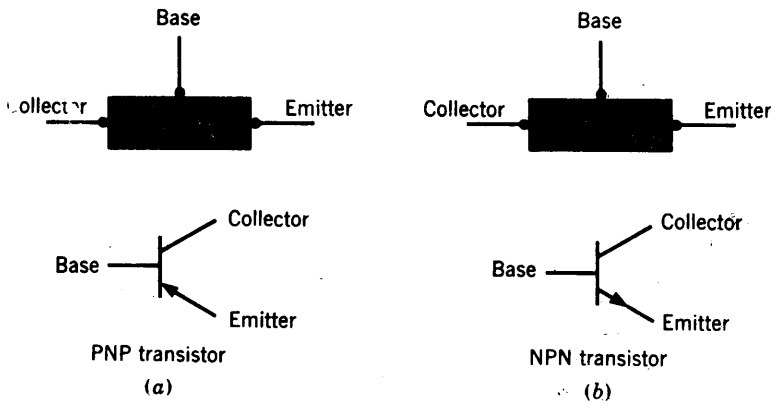


FIGURE A.2
Transistor symbols and designations.

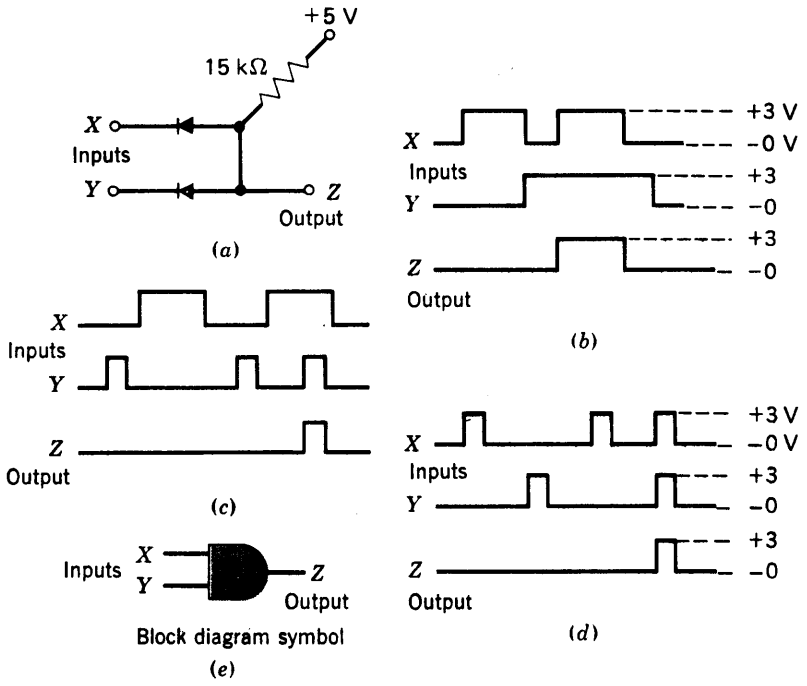


FIGURE B.1

Diode AND gate.

just as the two diodes are connected in the figure. If four diodes are connected as shown in Fig. B.1, the output will rise to the +3-V signal level only when the input signals to all four diodes are positive. There is a practical limit to the number of diodes which can be connected in this manner, however, because the diodes do not actually have an infinite back resistance or zero forward resistance. With a large number of inputs, the finite forward and back resistances of the diodes will cause varying output levels, depending on the state of the inputs.

The diode OR gate

The OR gate has the property that a signal representing a 1 will appear at the output if any one of the inputs represents a 1. Figure B.2 illustrates a diode OR gate circuit. There are two inputs to the circuit (X and Y) and one output. The input signals to the circuit consist of 0-V signals representing 0s and +3-V signals representing 1s. If both inputs to the circuit are at 0 V dc, both diodes will be forward-biased and the output of the circuit will be at 0 V dc, representing a 0. If either input to the circuit rises to +3 V dc, the diode at this input will be forward-biased and the output will rise to +3 V, representing a 1. The diode at the input remaining at 0 V will then be reverse-biased by the +3-V signal at the output. This circuit has the property that the output level will be at the level of the most positive input.

If both inputs to the circuit rise to +3 V, the output will again be at +3 V. (This circuit is sometimes referred to as an *inclusive* OR circuit, because the output is a 1 when both inputs are 1s.)

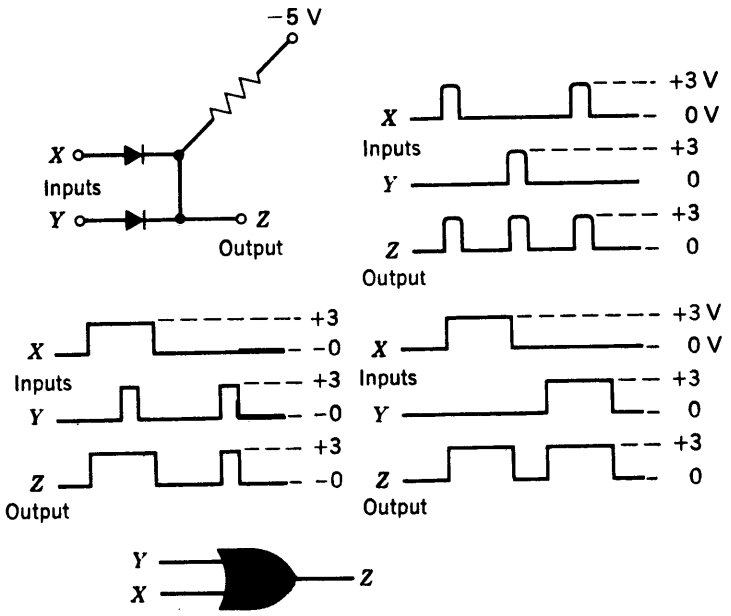


FIGURE B.2

Diode OR gate.

More inputs may be added to the circuit illustrated in Fig. B.2. A diode is then required for each input. If any one or any combination of the inputs rises to the +3-V level, the output will be at +3 V.

As in the case of the diode AND gate, it is not practical to have too many inputs to the circuit because the forward and back resistances of the diodes are finite, and different combinations of input signals will cause different signal levels at the outputs.