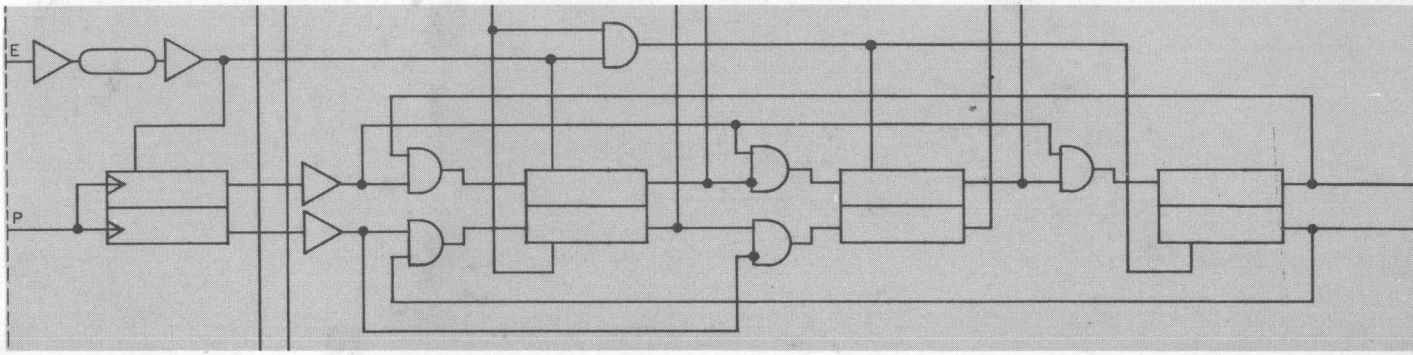
HEWLETT hp PACKARD

LOGIC SYMBOLOGY



application note 88

application note 88

LOGIC SYMBOLOGY

02415-1

PRINTED OCTOBER 1967

TABLE OF CONTENTS

LIST OF TABLES

introduction

SECTION I

1.1 LOGIC SYMBOLOGY.

1-2. Logic symbols and diagrams are a means of communication -- a language. A logic symbol or diagram presents, in simplified terms, the operation of complex logical circuits. As with any language, intelligent communication results only when the basic rules and fundamentals governing the language are understood.

1-3. At the present time there is no single system for representing logic circuits in universal use by all manufacturers of digital equipment, although a common basic approach has been adopted. A system in common use is that defined in MIL-STD-806B, and this system is largely followed in Hewlett-Packard instruments. An adaptation of this system, designed to provide greater clarity for complex digital equipment such as computers and related products, is described in this Application Note. A "Logic Dictionary" at the end of this Note summarizes the difference between the two systems.

1-4. BINARY LOGIC.

1-5. In the binary system, information is broken down into elementary bits. Each bit can exist in only one of two possible states. For logic, these states are designated "1" and "0". A "1" means yes, assertion, enable, or true, and hence "0" means no, negation, disable (inhibit), or false. Use of the words true and false does not imply that one state is more important than the other. The states are conditions, and both states are equally significant and used equally in a two-state system.

1-6. In logic notation, a single bit is always used to represent a function. For example, suppose the information to be conveyed is the presence (or absence) of a count. If the count has been received (or perhaps stored), the bit representing the count would be in the "1" state for presence or truth, otherwise the bit would be in the "0" state, denoting absence or falsity. The word "bit" is used to denote:

(a) an assertion or negation of a variable
(b) storage of a variable

1-7. Note that we have not defined or talked about what constitutes a bit being true ("1") or false ("0"). In electronic equipment, true and false are generally represented by voltage or current levels. "True" and "false" must be defined -- for example, true could be +12v, false -12v; true could be presence of current, false absence of current. In this Application Note, we will frequently refer to "signals", "inputs", or "outputs"; it should be understood that these are binary (true/false) signals.

1-8. Generally, voltage levels are used to define the true and false state. Additionally, a + or - sign may be used within any logic symbol to further define the true state for that element. A + sign within a logic symbol means the relatively positive level of the two logic voltages at which that circuit operates is said to be true.

1-9. Note that the term "relatively positive", for the true voltage level, does not have to be absolutely positive, i.e., above ground or 0v reference. The two voltage levels at which a logic circuit operates could be -24 vdc and -12 vdc; a + sign within the logic symbol representing that circuit would indicate that the -12 vdc level is defined as true and that -24 vdc level is therefore false; a - sign within the symbol would indicate that the -24 vdc level is true and that the -12 vdc level is therefore false.

1-10. For logic elements in which true and false levels are meaningful, the sign is included. Alternatively, a logic diagram may state in a note that all logic is positive true or all negative true.

1-11. BINARY REPRESENTATION OF NUMBERS AND FUNCTIONS.

1-12. By combining bits, numbers can be represented in binary form. The bits representing the number are each assigned a weight, thus the number of bits required depends upon the magnitude of the number to be represented. Because each bit can exist in only two states ("1" or "0"), the weights assigned to each successive bit can increase by a maximum factor of 2. So in a pure binary number, the weights assigned to successive bits are:

--	32	16	8	4	2	1	.	1/2	1/4	1/8	1/16	--
	2^5	2^4	2^3	2^2	2^1	2^0	.	2^{-1}	2^{-2}	2^{-3}	2^{-4}	

1-13. Each bit is either true or false and to obtain the number, the weights of all the true bits are added. The number 22.5 in pure binary is 10110.1000:

Binary Weight	2^4	2^3	2^2	2^1	2^0	.	2^{-1}	2^{-2}	2^{-3}	2^{-4}
Decimal Weight	16	8	4	2	1	.	1/2	1/4	1/8	1/16
Binary Number	1	0	1	1	0	.	1	0	0	0

1-14. The addition of the weights of all the true bits is 16 + 4 + 2 + 1/2 = 22.5. If all the possible bits are true (binary number 11111.1111), the decimal number represented is 16 + 8 + 4 + 2 + 1 + 1/2 + 1/4 + 1/8 + 1/16, or 31-15/16. For the number 32, the 2^5 bit would have to be added. Note that bits could be added to the right of the "decimal" point for resolution. (In the pure binary system, the decimal point is called the binary point.)

1-15. Also in use is the binary coded decimal (BCD) system, which combines the advantages of the binary system and the convenience of decimal representation. A number is expressed in normal decimal coding, however, each digit is expressed in binary. For example, the number 25 in pure binary BCD form would appear as:

	Tens Digit	Units Digit
BCD	0 0 1 0	0 1 0 1
(Pure Binary Weights)	(8 4 2 1)	(8 4 2 1)
Decimal	2	5

1-16. Four bits are needed for each digit. In the general case, four bits yields 2^4 or sixteen possible combinations (see Table 1). Because of the types of counting circuits sometimes used, instead of an 8-4-2-1 or pure binary code for each digit, a 4-2'-2-1 code can be used (again four bits are used for each digit). Other codings exist, e.g., XS-3 and 2-4-2-1; the sixteen possible combinations of four bits and several codes in use are shown in Table 1. These combinations contain all possible four bit fixed weight codes. In each case, however, six combinations are unused in specifying the familiar ten decimal numbers. The six unused combinations are frequently referred to as forbidden codes, in reference to a particular fixed-bit weight assignment.

1-17. Besides the binary and BCD forms, other systems for representation of numbers exist. One such is the ten-line code, where each digit is represented by ten bits, bit weighting being zero through nine. For a given digit, only one bit of the ten-line code can be true at one time. In more general terms, this scheme is known as multiple-line coding.

1-18. Conceivably, we might want to know when a count is not true, but for ease of analysis we would want the "count not true" to be in the "1" state when the count was in fact, not true. This leads to negation, or the "not" function. Let us examine the eight bit of the 8-4-2-1 code to see how negation works.

	8-4-2-1	$\overline{8}$-$\overline{4}$-$\overline{2}$-$\overline{1}$
0	0 0 0 0	1 1 1 1
1	0 0 0 1	1 1 1 0
2	0 0 1 0	1 1 0 1
3	0 0 1 1	1 1 0 0

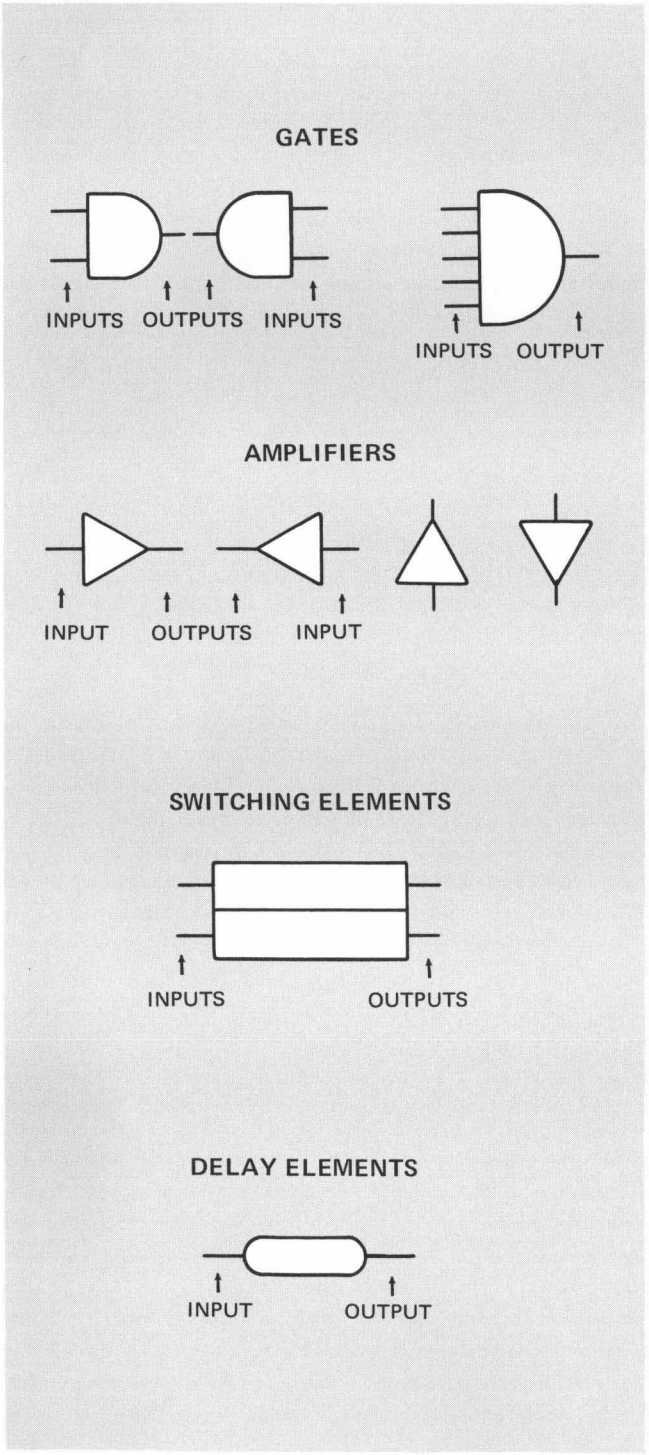
1-19. "Not" is indicated by a bar above the bit identification; e.g., not eight is $\overline{\text{eight}}$. For all the numbers where the eight bit is true, the "not eight" bit is, by definition, false. Note that the $\overline{\text{eight}}$ bit is a separate bit, but related by definition to the eight bit. Conversely, for all the numbers where the eight bit is false (not true), the $\overline{\text{eight}}$ bit is true.

Table 1. Common BCD Codes

Code Bits	Decimal Equivalents			
	8-4-2-1	4-2'-2-1	XS-3	2-4-2-1
0 0 0 0	0	0		0
0 0 0 1	1	1		1
0 0 1 0	2	2		2
0 0 1 1	3	3	0	3
0 1 0 0	4		1	4
0 1 0 1	5		2	5
0 1 1 0	6	4	3	6
0 1 1 1	7	5	4	7
1 0 0 0	8		5	
1 0 0 1	9		6	
1 0 1 0			7	
1 0 1 1			8	
1 1 0 0		6	9	
1 1 0 1		7		
1 1 1 0		8		8
1 1 1 1		9		9

basic logic symbols

SECTION II



2-1. GENERAL CLASSIFICATIONS.

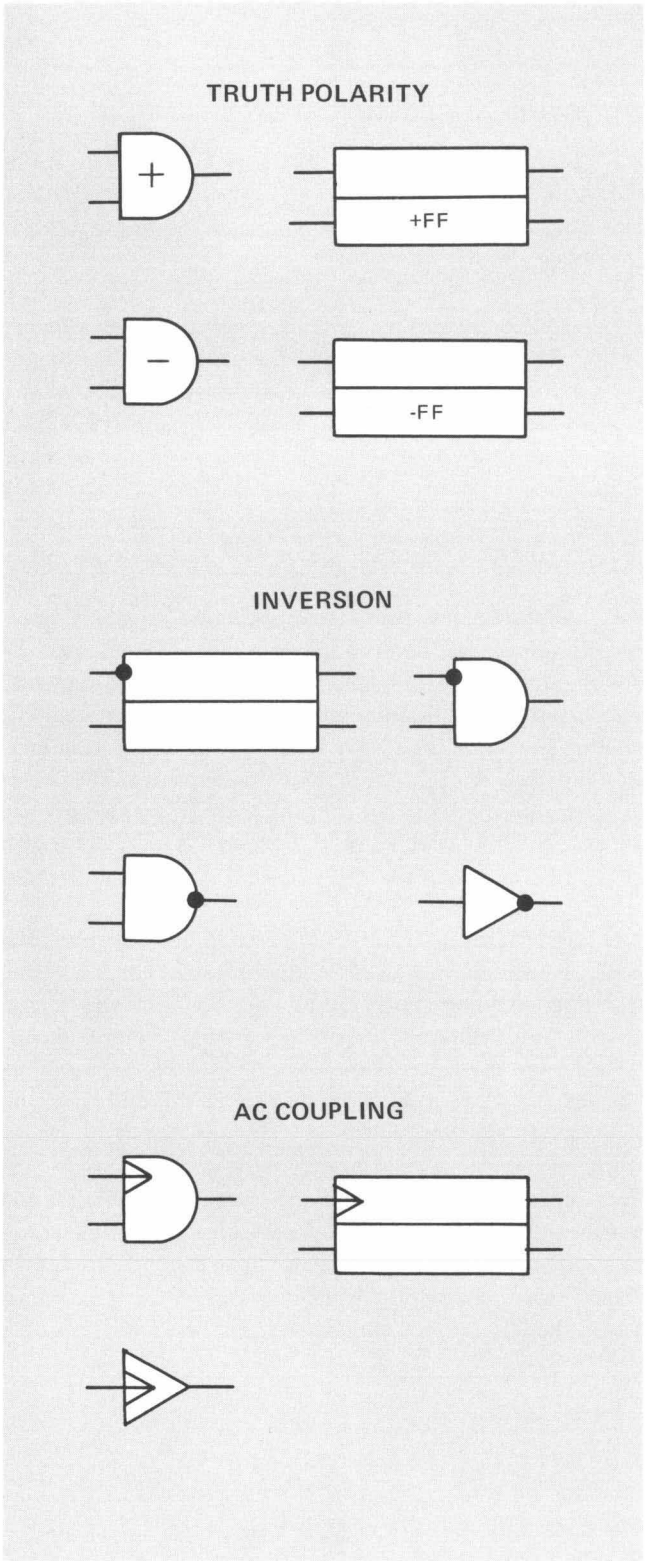
2-2. Four basic envelope shapes distinguish the major classes of logic circuits. These classes are: gates, amplifiers, switching elements, and delay elements. The four envelopes are shown at left.

2-3. GATES. A gate is a circuit which produces a binary output on condition of certain rules governing binary input combinations. (Specific rules for each type of gate will be defined later under Paragraph 2-12.) The gate symbol has input lines connecting to the flat side of the symbol, and output lines connecting to the curved side. Since inputs and outputs are thus easily identifiable, the symbol can be shown left-facing, right-facing, or even facing up or down.

2-4. AMPLIFIERS. Amplifiers, of themselves, are not necessarily binary in nature; however, in logic circuits the driving signals will normally be binary, and consequently the output of the amplifier will be an amplified or otherwise modified form of the binary input. The amplifier symbol is an equilateral triangle, with the input applied to the center of one side, and the output connected to the opposite point of the triangle. Like gates, the amplifier may be shown in any of four positions.

2-5. SWITCHING ELEMENTS. Switching elements comprise the various forms of multivibrator circuits: bistable (flip-flop, Schmitt trigger), monostable (one-shot), and astable (multivibrator). According to the type of circuit, inputs cause the state of the circuit to switch, reversing the outputs (i.e., an output formerly true switches to false, and vice versa). The symbol for all switching circuits is a horizontal rectangle, divided horizontally, with the upper portion representing the "set side" and the lower portion representing the "reset side". A switching element is said to be "set" when the output from the set side is true. It is "reset" when the output from the reset side is true. Inputs are on the left; outputs are on the right. To avoid confusion, switching elements are always drawn facing the same way.

2-6. DELAY ELEMENTS. A delay element provides a time delay between input and output signals. The symbol accepts the input on the left, and provides output on the right. Like switching elements, delay elements are always drawn facing the same way.



2-7. SYMBOL MODIFIERS.

2-8. Additional markings on the above basic symbols provide interpretive details, making possible the determination of actual circuit action. Logic theory is then easily translatable to electronic voltages, currents, etc.

2-9. TRUTH POLARITY. As previously mentioned (Paragraphs 1-8 through 1-10), positive or negative indicators (+/-) may be placed inside a symbol to designate whether the true state for that circuit is positive or negative, relative to the false state. This is frequently done with gates and switching elements, as shown at left. (The FF designation indicates that the particular switching element is a flip-flop.) Where all symbols on a particular diagram have the same polarity, a note to the effect that all logic is positive-true or all negative-true may be used instead of having individual polarity signs in each symbol. Polarity signs also used in amplifier symbols do not have a logic significance, but rather are a troubleshooting aid, indicating the polarity required to turn the amplifier "on". In the example at left. the positive-true gate and flip-flop operate with true levels being positive with respect to the false levels. Similarly, the negative-true gate and flip-flop operate with true levels being negative with respect to the false levels.

2-10. INVERSION. Logic inversion is indicated by an inversion dot at inputs or outputs of symbols. When this dot appears on an input (generally only on gates and switching elements), the input will be effective when the input signal is of opposite polarity to that normally required. (e.g., If the switching element at left is positive-true, a negative input at the inversion dot will set the circuit.) When the dot appears at an output (generally only on gates and amplifiers), the output will be of opposite polarity to that normally delivered. (e.g., The amplifier at left will provide an output of opposite polarity to that of the input.)

2-11. AC COUPLING. Capacitor inputs to logic elements are indicated by an arrow, as shown at left. In the case of gates and switching elements, the element responds only to a change of the ac-coupled input in the "true-going" direction. An inversion dot used in conjunction with the ac-coupling arrow indicates that the element responds to a change in the "false-going" direction. In the case of amplifiers, a pulse edge of the same polarity as given in the symbol turns the amplifier on briefly, then off as the capacitor discharges. The output is then a pulse of the same width as the amplifier on-time. With an inversion dot at the output, the output pulse is inverted.

2-12. SYMBOLS AND DEFINITIONS

2-13. "And" Gate.

2-14. The symbol for the "and" gate is shown at right. By definition, for output C to be true, inputs A and B must be true, hence the term "and" gate. To help illustrate logic notations, "truth tables" are used. The truth table for a two-input "and" gate is shown adjacent to the symbol. In this table, "1" represents true, and "0" represents false. Note that C is true ("1") only when both A and B are true, and that the truth table does not define true and false.

2-15. If a + sign is given in the gate symbol, the gate is defined as a positive-true "and" gate. A typical schematic for such a gate is shown adjacent to the symbol. A negative-true "and" gate would have a - sign in the gate symbol. The symbol and a typical schematic are also shown at right. If the respective true/false levels for the two gates are +5v/0v and -5v/0v, as shown, we can substitute these values in the truth table. See Table 2. In both cases the true voltage level appears at output C when the A and B input diodes are both reverse-biased by true inputs. At all other times, the false level (approximately 0v) exists at output C.

Table 2. Voltage Truth Tables for "And" Gate

Positive Gate		
A	B	C
0v	0v	0v
0v	+5v	0v
+5v	0v	0v
+5v	+5v	+5v

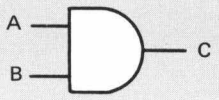
Negative Gate		
A	B	C
0v	0v	0v
0v	-5v	0v
-5v	0v	0v
-5v	-5v	-5v

2-16. "And" gates are not restricted to two inputs, but may have any number of inputs, including the single-input case. The rule is that all inputs must be true for the output to be true. For example, see Table 3, the truth table for the three-input gate shown at right.

Table 3. Truth Table for Three-Input "And" Gate

A	B	C	D
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

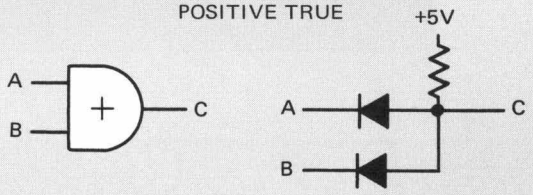
"AND" GATE



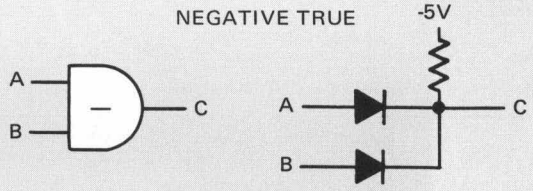
TRUTH TABLE

A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

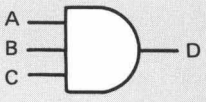
POSITIVE TRUE

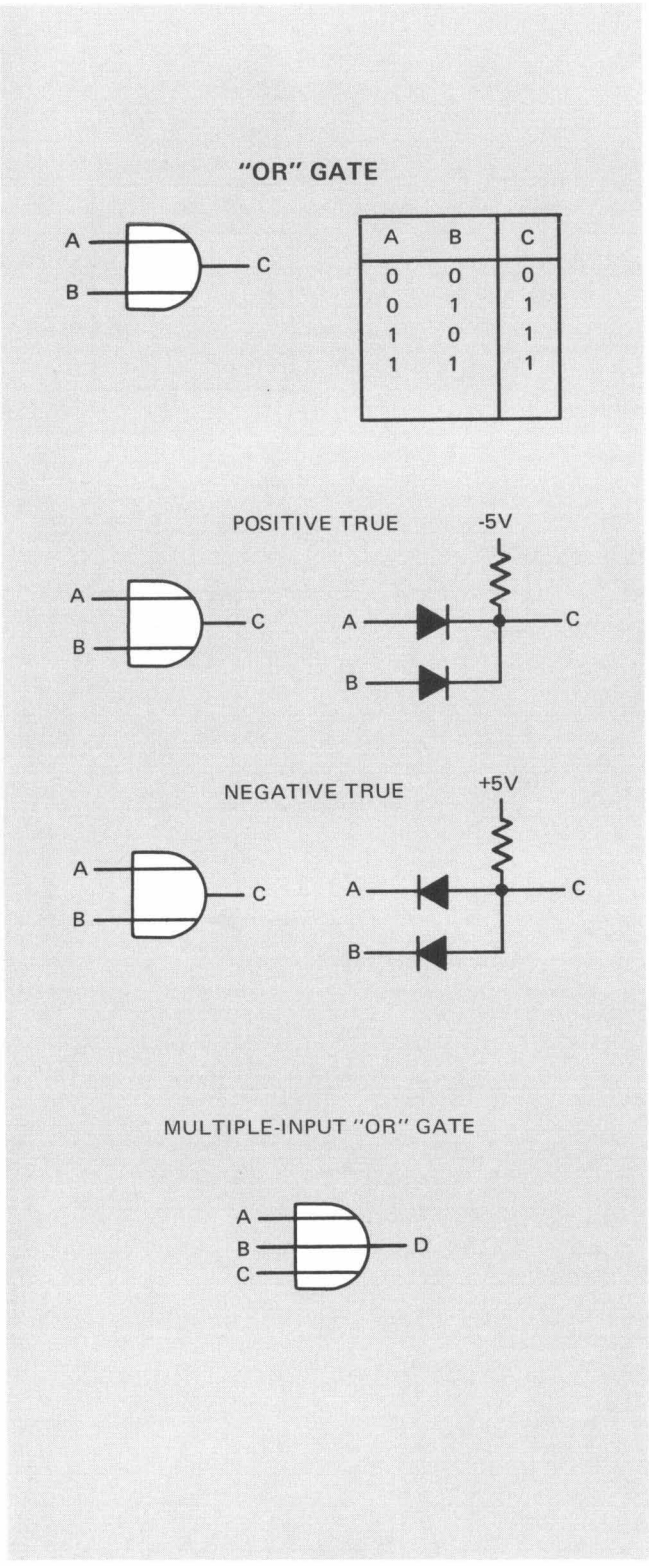


NEGATIVE TRUE



MULTIPLE-INPUT "AND" GATE





2-17. "Or" Gate.

2-18. The symbol for the "or" gate is shown at left. By definition, for C to be true, either input A or input B must be true, hence the name "or" gate. The truth table for a two-input "or" gate is shown adjacent to the symbol. Note that C is true ("1") whenever any of the inputs is true, and that the truth table does not define true and false.

2-19. If a + sign or a - sign is given in the symbol, the gate is defined as a positive-true "or" gate or a negative-true "or" gate respectively. The symbols and typical circuits for both gates are shown at left. Note that in the circuit sense the positive-true "or" gate is the same as a negative-true "and" gate. The choice of logic symbol used depends upon the logical operation in a particular application. If the same voltage levels are used as in the "and" gate examples, the 0v level is now defined as true, rather than the +5v or -5v levels. Substitution of voltage levels in the truth tables yields the result shown in Table 4 (compare with Table 2).

Table 4. Voltage Truth Tables for "Or" Gate

Positive Gate			Negative Gate		
A	B	C	A	B	C
-5v	-5v	-5v	+5v	+5v	+5v
-5v	0v	0v	+5v	0v	0v
0v	-5v	0v	0v	+5v	0v
0v	0v	0v	0v	0v	0v

2-20. As with "and" gates, "or" gates may have more than two inputs. Any true input will produce a true output. Table 5 is the truth table for the three-input "or" gate shown at left.

Table 5. Truth Table for Three-Input "Or" Gate

A	B	C	D
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

2-21. "Nand" Gate.

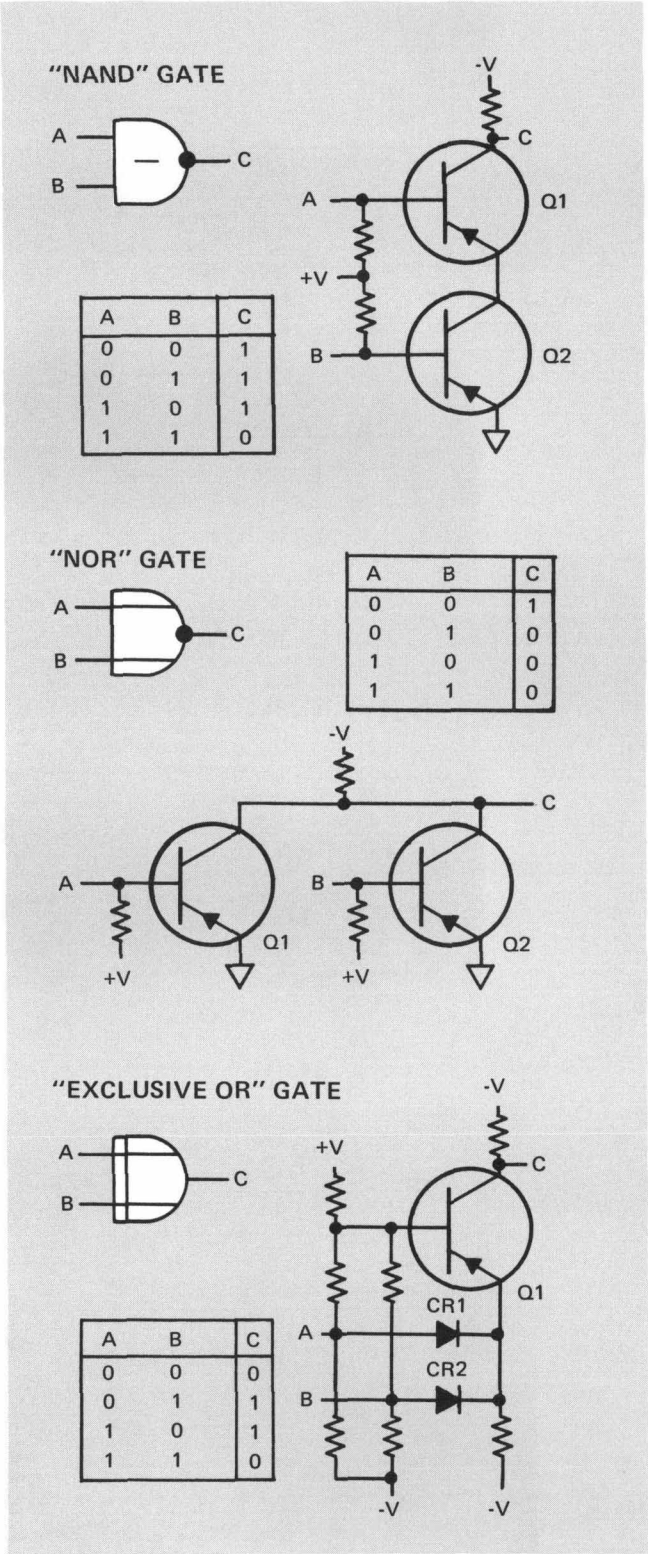
2-22. The "nand" gate is a variation of the conventional "and" gate, delivering an inverted (false) output when all inputs are true. The symbol, its truth table, and a typical schematic are shown at right. The term "and" is a contraction of "not and". Note that the output is true when either or both of the inputs are false. Essentially a "nand" gate is the result of using an active inverting element in the gate circuitry. As with a conventional "and" gate, the "nand" gate may have any number of inputs. Operation of the circuit is as follows: for output C to be positive (at ground), both Q1 and Q2 must be conducting, which requires both A and B inputs to be negative. If either A or B is positive, then either Q1 or Q2 is biased off, and the output is negative.

2-23. "Nor" Gate.

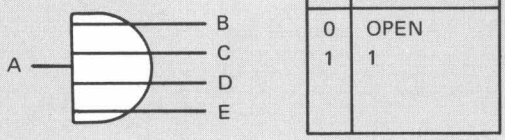
2-24. Similar to the "nand" gate, the "nor" gate is a variation of the conventional "or" gate, delivering an inverted (false) output when any or all of its inputs are true. The symbol, its truth table, and a typical schematic are shown at right. The term "nor" is a contraction of "not or". Note that the output is true when all inputs are false. Like the "nand" gate, the 'nor" gate uses an active inverting element in the gate circuitry. The "nor" gate also may have any number of inputs. Operation of the circuit is as follows: for output C to be positive (at ground), either or both Q1 and Q2 must be conducting. This requires that either or both inputs A and B are negative. If both A and B are positive, Q1 and Q2 are biased off, and the output is negative.

2-25. "Exclusive Or" Gate.

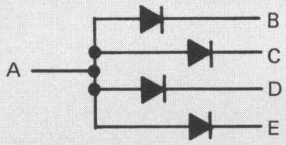
2-26. An "exclusive or" gate is a special type of "or" gate. It has two inputs, and the output will be true if one, but not both, of the inputs is true. The converse statement is equally accurate: the output will be false if the inputs are both true or both false. The "exclusive or" gate is therefore independent of polarity, and is not generally spoken of as being either positive-true or negative-true. The symbol, its truth table, and a typical schematic are shown at right. Action of the circuit is as follows. If A or B is positive, then Q1 is biased on via CR1 or CR2, and the C output is positive. If A and B are both negative, Q1 is biased on, but the voltage at the Q1 emitter is negative, and the C output is negative. With A and B both positive, Q1 is biased off, and the C output remains negative. (These results require proper selection of resistor values.)



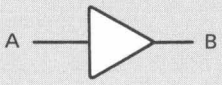
"ENCODE" GATE



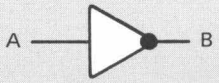
A	B- C- D- E-
0	OPEN
1	1



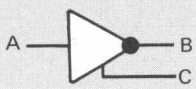
AMPLIFIER



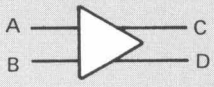
INVERTER



PHASE SPLITTER



DIFFERENTIAL AMPLIFIER



2-27. "Encode" Gate.

2-28. The "encode" gate has one input and multiple outputs. When the input is true, all outputs are true. When the input is false, the outputs may be true or false, depending on the state of the logic element to which they are connected. The symbol, its truth table, and a typical schematic are shown at left. A + or - sign within the symbol indicates the true state. In the circuit shown, with A positive, all diodes conduct and all outputs are clamped positive. With A negative, all diodes are open.

2-29. Amplifier and Inverter.

2-30. As logic elements are combined in series to perform logic functions, voltage levels tend to degrade. Thus, amplifiers are sometimes needed to restore voltages (or currents) to proper levels. Assuming no inversion is encountered, a true input will produce a true output, and a false input will produce a false output. When inversion occurs, an inversion dot is placed at the output, and the symbol is usually termed an inverter. The symbols for both cases are shown at left. If a + or - sign is used in the symbol, this indicates the input polarity required to turn the amplifier "on". One amplifier or inverter symbol may represent any number of amplification stages or, optionally, separate symbols may be shown for each stage. Logic symbols, by themselves, do not necessarily imply a specific number of components, but rather relate to overall logic effect.

2-31. Phase Splitter and Differential Amplifier.

2-32. Variations of amplifiers, in the form of dual output (phase splitter) and dual input/output (differential amplifier), are shown at left. The rule for use of inversion dots is that the dot indicates inversion of an output with respect to the corresponding input (not with respect to the opposite side of the amplifier).

2-33. Flip-Flops.

2-34. A flip-flop is a bistable switching element, meaning that it takes an external signal to set the element, and another signal to reset it. It will remain in its current state until switched to the opposite state by the appropriate external signal. Various forms of flip-flops exist, of which six are described in this Application Note: the reset-set, reset-set with clock, J-K, toggle, latching, and delay flip-flops. Before individually dealing with these specific forms, the general rules governing the logical representation of flip-flops will be presented. By application of these rules, other

forms of flip-flops can also be represented. General rules for flip-flops (additional to those for all switching elements, Paragraph 2-5) are as follows:

a. The letters FF will appear in either the upper or the lower portion of the symbol (see Logic Symbol Identification, Section III), thus identifying the element as a flip-flop.

b. A flip-flop is assumed to be the simple R-S (reset-set) type if no other identification is made. When a clock input is added, identifying letters are placed inside the symbol (as will be shown in the examples following) to tell what kind of flip-flop the circuit is.

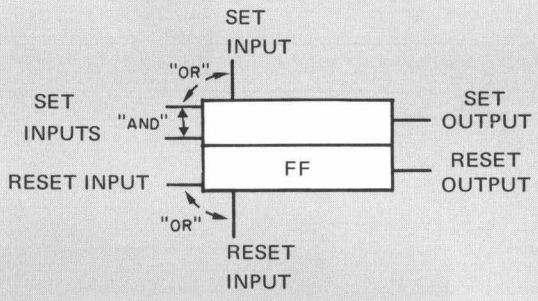
c. Multiple inputs on the same side of the symbol require the logical "and" condition to be effective; multiple inputs diagonally on the corner of the symbol are effective as "or" inputs. (Separate gate symbols may be used if large numbers of inputs need to be shown.) This is summarized in the general symbol at right.

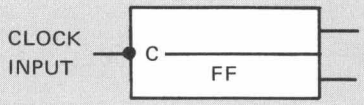
d. An input shown connected to the center of the input side of the symbol is a "clock" input, parallel-connected to both the set and the reset side. This input is transient operated; i.e., it is effective on leading or trailing edges of pulses, in the same manner as ac-coupled inputs. No inversion dot indicates that the input is effective on the true-going edge of the clock pulse; when present, as shown at right, the dot indicates effectiveness on the false-going edge.

2-35. Definitions and descriptions of the six major flip-flop types follow.

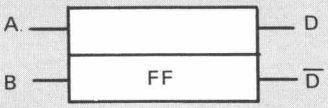
2-36. R-S (RESET-SET) FLIP-FLOP. The R-S flip-flop has a minimum of two inputs, set and reset (A and B), and usually two outputs, set output and reset output (D and $\overline{D}$). The $\overline{D}$ letter indicates that the reset output, whether a 1 or a 0, is always the complement of the set output. (See Paragraphs 1-18 and 1-19.) When D is true and $\overline{D}$ is false, the flip-flop is defined as being in the set state. With D false and $\overline{D}$ true, the flip-flop is in the reset state. The flip-flop is set by a true input to A (assuming no inversion dot on the symbol), and is reset by a true input to B. False inputs have no effect. Simultaneous true inputs to A and B is forbidden combination, since an indeterminate output state would result. A truth table for the three allowable input combinations is shown below the symbol. If ac-coupled inputs are used (see Paragraph 2-11), the flip-flop would be set or reset by true-going transitions at A and B respectively. If, in addition, input inversion dots are also used, false-going transitions at A or B would set or reset the flip-flop.

FLIP-FLOP (GENERAL)



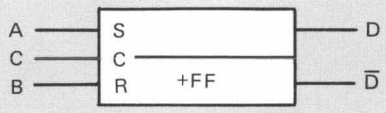


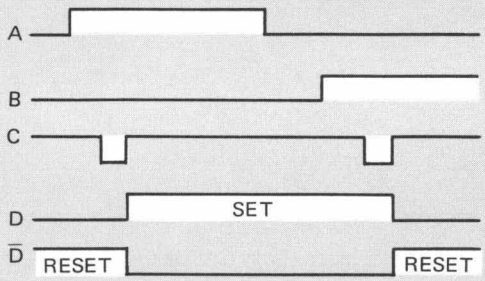
R-S FLIP-FLOP



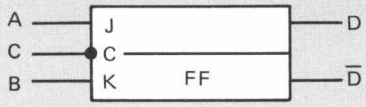
A	B	D	$\overline{D}$
1	0	1	0
0	1	0	1
0	0	NO CHANGE	

R-S WITH CLOCK





J-K FLIP-FLOP



J-K FLIP-FLOP (AC-COUPLED)

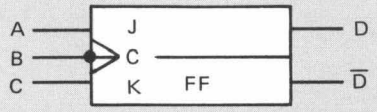


Table 6. Truth Table for J-K Flip-Flop

		Initial State		Resulting State	
A	B	D	$\overline{D}$	D	$\overline{D}$
1	0			1	0
0	1			0	1
1	1	0	1	1	0
1	1	1	0	0	1
0	0			No change	

2-37. R-S FLIP-FLOP WITH CLOCK. This flip-flop is the same as the R-S type described in the preceding paragraph, except for the addition of a clock input. Observing rule c in Paragraph 2-34, regarding multiple inputs, it follows that a true input to both A and C (logical "and") is required to set the flip-flop, and a true input to B and C is required to reset. Since rule d states that the clock input operates on a pulse edge, the setting or resetting signal must be present at A or B before the clock pulse transition occurs. An example of positive-true timing waveforms is shown below the symbol.

2-38. J-K FLIP-FLOP. As stated in Paragraph 2-36, a feature of the R-S flip-flop is that simultaneous true inputs for set and reset cannot be allowed, since the behavior of the circuit under these conditions cannot be predicted or defined. The J-K flip-flop, however, does define this condition: simultaneous true inputs for both set and reset will reverse the existing state of the flip-flop. This requires some method of storing two information conditions until the clock pulse time: the existing output state and the new input state. Two common methods of doing this involve the use of ac-coupling or a "dual rank" flip-flop, both shown at left. (Labeling of the J (set) and K (reset) inputs identifies the J-K flip-flop.) The ac-coupling method uses the RC time constant of the capacitive input for short-term "storage" of the input information. The dual-rank method avoids the slow-reacting capacitor arrangement by actually combining two flip-flops (input storage and output storage) and several gates as a single logic element. For simplicity of representation, the internal dual-rank arrangement of the flip-flop is not usually shown. The overall operation of the J-K flip-flop (ac or dual-rank) is as follows, summarized also in Table 6.

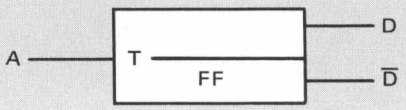
a. True input at A only. Leading edge of clock pulse at C acknowledges (stores) the input information at A; trailing edge of clock pulse sets the flip-flop.

b. True input at B only. Leading edge of clock pulse at C acknowledges (stores) the input information at B; trailing edge of clock pulse resets the flip-flop.

c. True inputs at A and B. Leading edge of clock pulse at C acknowledges the input information at A and B; trailing edge of clock pulse switches the existing state of the flip-flop.

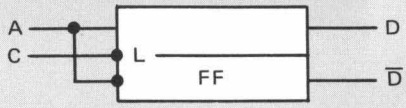
2-39. TOGGLE FLIP-FLOP. The toggle flip-flop is distinguished by having a single input. Each time input A goes true, outputs D and $\overline{D}$ switch states. Since two input pulses or cycles are required to produce one complete cycle of the output, the toggle flip-flop acts as a divide-by-two element, and is commonly used in counting circuits. The letter T inside the symbol identifies the toggle flip-flop.

TOGGLE FLIP-FLOP



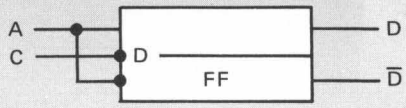
2-40. LATCHING FLIP-FLOP. The latching flip-flop has a single signal input and a clock input. The symbol is identified by the letter L inside the symbol as shown. Note that the set input is responsive to true signals at A, and the reset input is responsive to false signals at A. The flip-flop will "latch" into the state existing at A when the clock pulse at C transits from true to false (i.e., usually at the trailing edge of the clock pulse). One unusual feature (not always desirable) is that during the clock pulse duration, the flip-flop is "unlatched", so that if A switches true and false several times during this period, outputs D and $\overline{D}$ will freely switch accordingly. Compare with the delay flip-flop, next described.

LATCHING FLIP-FLOP



2-41. DELAY FLIP-FLOP. The delay flip-flop is similar to the latching flip-flop, and its symbol is identified by the letter D inside the symbol as shown. Like the latching flip-flop, it will latch into the state existing at A when C transits from true to false. However, unlike the latching flip-flop, the delay flip-flop "delays" any switching of outputs until the trailing-edge clock transition occurs. It does not have an "unlatched" condition. The delay flip-flop requires more circuitry than the latching flip-flop and is therefore more costly to construct.

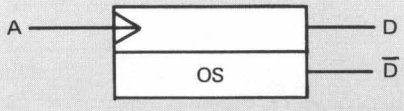
DELAY FLIP-FLOP

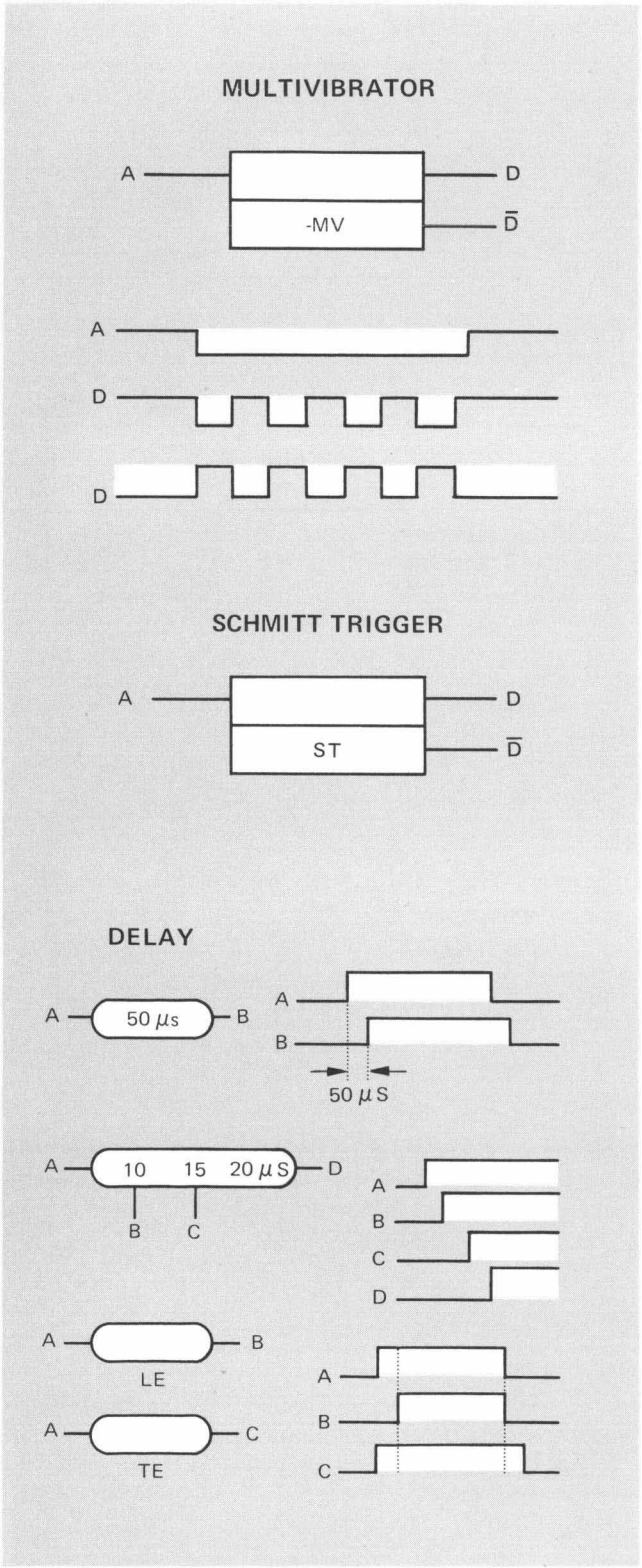


2-42. One-Shot.

2-43. The one-shot is a monostable switching element, commonly used as an active delay device (vs. the normally passive delay elements described later in Paragraph 2-48). The one-shot is triggered into its unstable state by an external signal. It returns automatically to the stable state after an interval determined by circuit constants, thus providing a known, fixed delay time. One-shot inputs are frequently ac-coupled, and triggering is accomplished when input A goes through a false-to-true transition. The abbreviation OS within the symbol identifies a one-shot, and a + or - sign may be used to indicate the true state of the D output during the "on time".

ONE-SHOT





2-44. Multivibrator.

2-45. Although all logic switching elements are actually forms of multivibrators, the logic symbol for "multivibrator" assumes the astable type. The astable multivibrator will start free-running operation when input A goes true, and will continue to generate complementary pulse trains at outputs D and $\overline{D}$ until A goes false. Typical negative-true timing waveforms are shown with the symbol. Note that the - sign indicates both the relative level required to start operation and the direction of the first output pulse at D. (Waveforms do not necessarily have to be symmetrical as shown.)

2-46. Schmitt Trigger.

2-47. The Schmitt trigger is a two-state element used for level sensing and signal squaring. When the input voltage is below a reference level, the element is in one state. When the input level goes above the reference level, the element switches to the other state. Switching between states takes place rapidly, and Schmitt triggers are therefore useful for squaring signals with poor rise and fall times (e.g., sine waves) and for voltage level restoration. With input A below the reference level (false), D is false and $\overline{D}$ is true. When the input is above the reference level, D switches to true and $\overline{D}$ switches to false. Circuit constants establish the reference level.

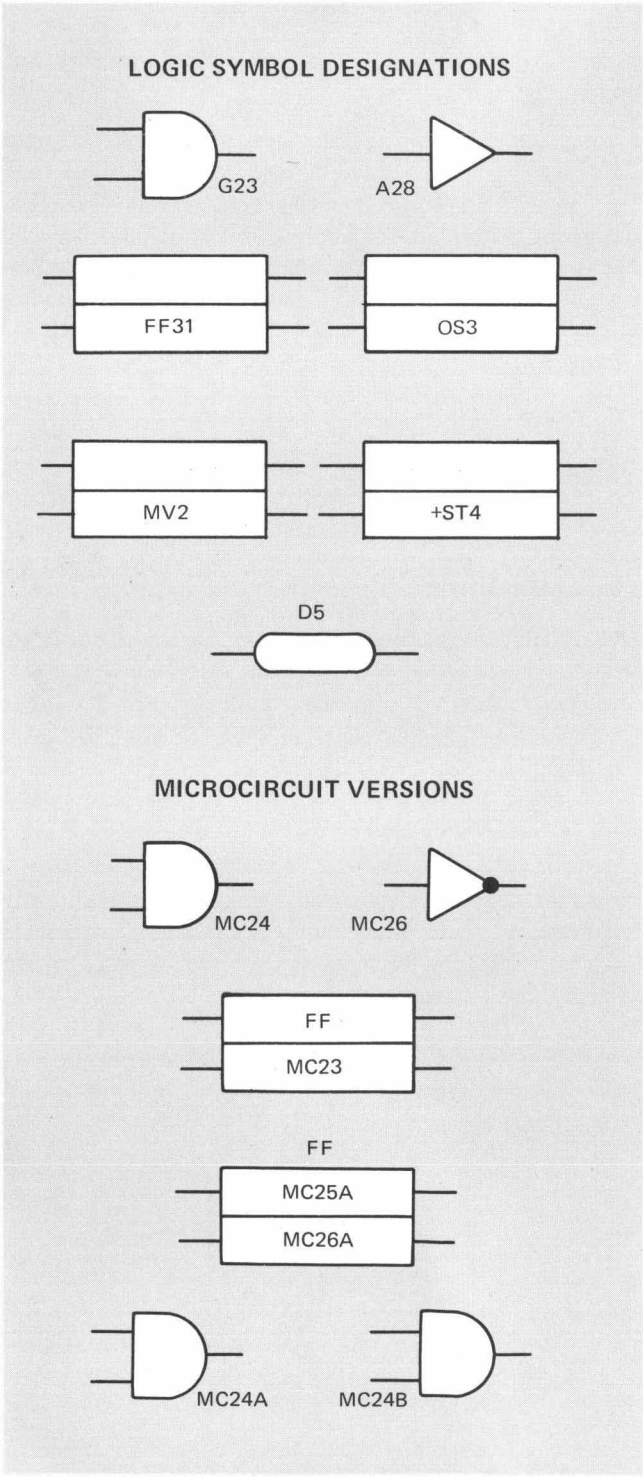
2-48. Delay Element.

2-49. A delay element provides a finite time delay between the input and output signals. The symbol is shown at left, with an example of actual delay time shown within the symbol. Typical theoretical waveforms for this element are shown adjacent to it.

2-50. Other configurations of delay elements exist, such as the tapped delay and delays which are effective only on leading or trailing pulse edges. The tapped delay symbol shows each separate output with delay times indicated inside the symbol next to each output. Delays introduced on leading or trailing edges are signified by LE (leading edge) or TE (trailing edge) immediately below the symbol as shown.

logic symbol identification

SECTION III



3-1. REFERENCE DESIGNATIONS.

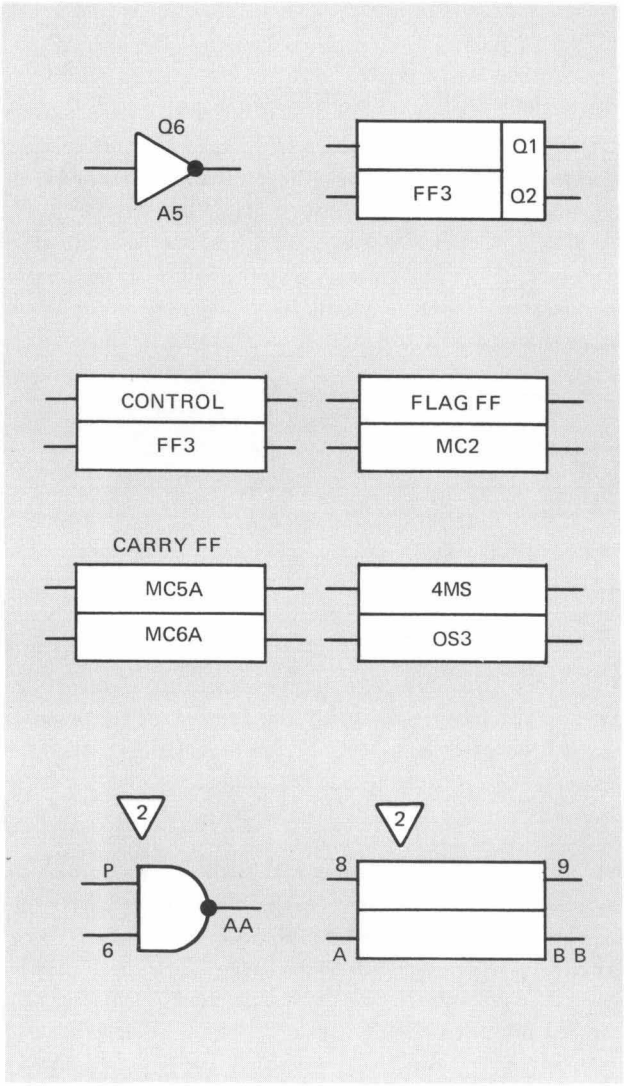
3-2. Since logic elements are frequently made up of many electronic components, logic diagrams often cannot use designators which relate directly to components as would be listed in a parts list. However, designations are required for reference in description texts, and the following designators are used in Hewlett-Packard logic diagrams (examples of usage shown at left):

Gates	G
Amplifiers and inverters	A
Flip-Flops	FF
One-Shot	OS
Multivibrator	MV
Schmitt trigger	ST
Delays	D

3-3. Note that designations for switching elements are given in the lower portion of the symbol. (The true-state sign can be used as a prefix to the designation.) All other designations are placed beside the symbol.

3-4. The reverse case (compared with the above case of having several components per logic element) is to have one or more logic elements in one physical component. Microcircuit ("integrated-circuit") packages illustrate this case. Here, it is possible to use the "parts list" designator, MC, reserved for microcircuit packages. The envelope shape of gates, amplifiers, and delay symbols identifies the function of these elements. Flip-flops and other switching elements, however, are required to be identified as such, and the appropriate abbreviation appears in the "name" (upper) portion of the symbol (see Paragraph 3-7). Examples of microcircuit designations are given at left.

3-5. Where more than one logic element is included in the microcircuit package, each element is identified with an A, B, C, etc., suffix. Occasionally, also, a switching element may be composed of portions of different microcircuit packages. This case is illustrated at left; note that the FF identification appears outside the symbol. Also shown is an example of two gates which are part of one microcircuit package.



3-6. On logic diagrams which correlate to a schematic diagram of the same circuit, the active components of the element may also be designated, as shown at left. This is primarily an aid in troubleshooting. The symbol shown for the switching element is a variation of the standard symbol for this special prupose.

3-7. REFERENCE NAMES.

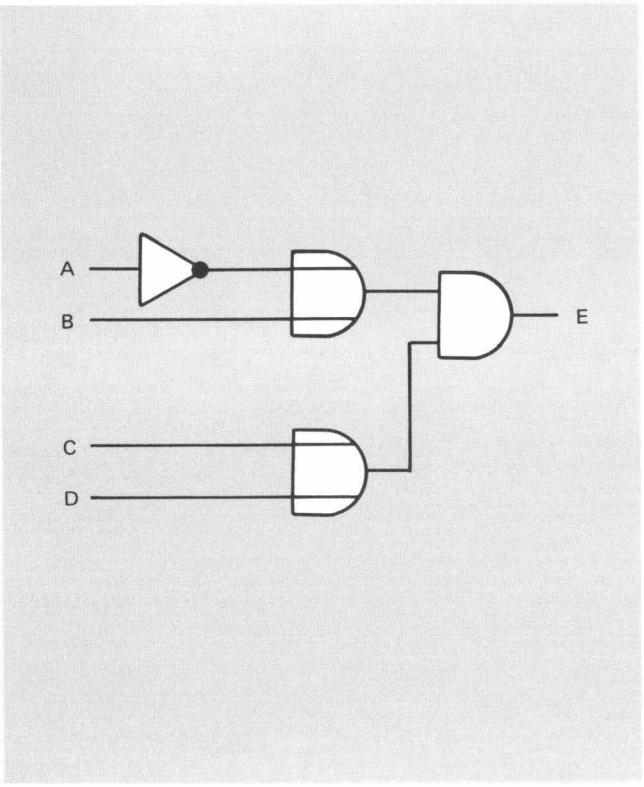
3-8. Identification of logic elements by functional name in a diagram is normally done only with switching elements. The upper portion of the symbol is reserved for this purpose. If the type of switching element (FF, OS, etc.) is not part of the designation (lower portion of the symbol), the reference name is obliged to include the appropriate abbreviation. If designations appear in both upper and lower portions, the name will be given above the symbol. For one-shots, it is often convenient to give the time duration as the reference name. Examples of the use of reference names are shown at left.

3-9. LOCATION INFORMATION.

3-10. Logic diagrams are intended to show the combination of logic elements that together form an instrument, a part of an instrument, or a system of instruments. To aid in correlating the diagram with physical locations in instruments, additional information is given with logic symbols as shown at left. The number in the small triangle is sometimes used to indicate the circuit board on which the element appears (e.g., circuit board A2). If all elements of a diagram are on the same board, individual identification in this way is unnecessary. Letters and numbers adjacent to inputs and outputs indicate pin numbers of the board (or microcircuit package) where inputs/outputs appear.

logic equations

SECTION IV



4-1. To aid in design or explanation of logic circuits, logic equations are sometimes used. In these equations, the familiar algebraic symbols have the following meanings:

+ means	"or"
· means	"and"
= means	"equals" (or "is the result of")
$\bar{}$ means	"not" (or "complement of")

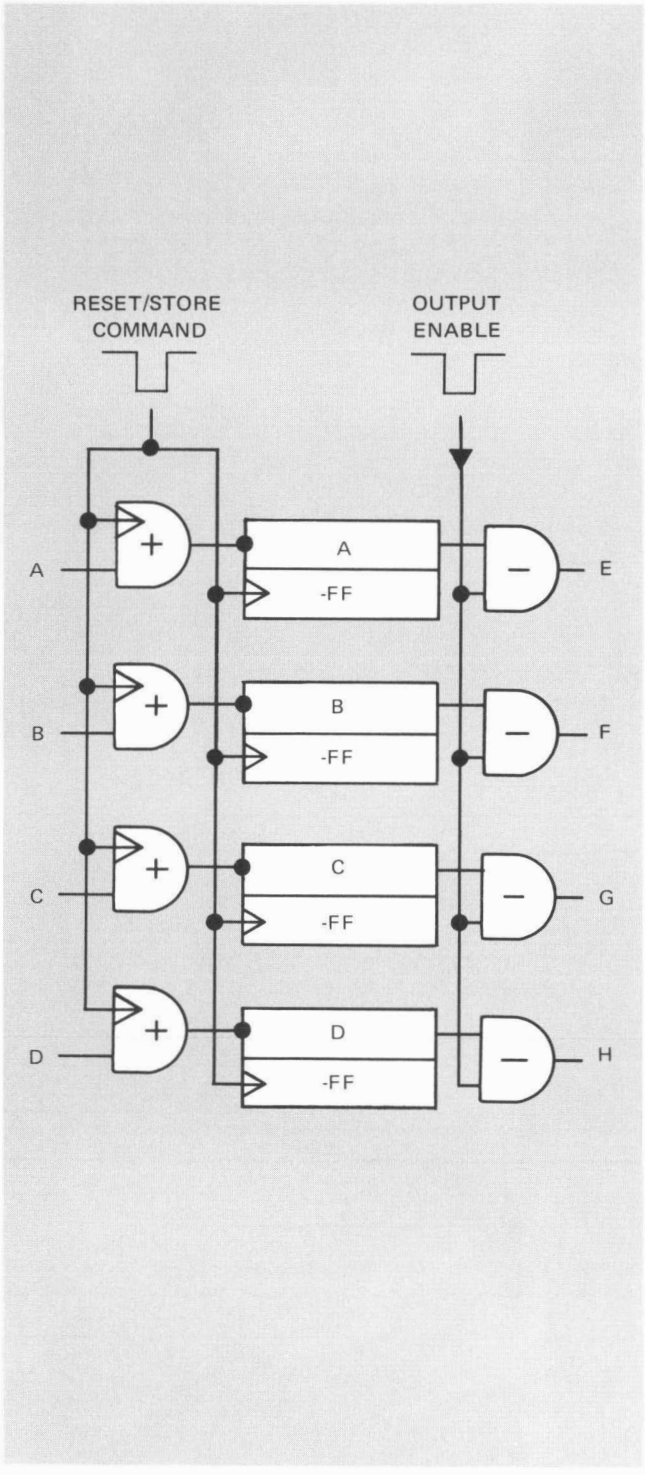
4-2. For example, the logic equation for a simple two-input "and" gate is:

$$C = B \cdot A$$

4-3. The logic equation for the combination of elements shown at left is:

$$E = (\bar{A} + B) \cdot (C + D)$$

4-4. Logic equations are especially useful in designing logic instruments where Boolean algebra and other mathematical tools can be used to simplify design and eliminate logical redundancy (see Montgomery Phister Jr., "Logical Design of Digital Computers", John Wiley and Sons, Inc., New York (1958)). Logic equations are used extensively in Hewlett-Packard computer manuals.



5-1. STORAGE REGISTER.

5-2. The storage register is a combination of gates and flip-flops used to store binary information. A typical storage register is illustrated at left. This register can store four bits of binary information, applied to inputs A, B, C, and D. The inputs must be positive true. Assume, for illustration. that A and C are at positive levels (true), and that B and D are negative (false). Operation is as follows:

a. Before information is stored, all four flip-flops are reset by the leading edge of the Reset/Store Command pulse. This clears the register of any previously stored information.

b. The trailing (positive-going) edge of the Reset/Store Command pulse is coupled into the four positive-true input "and" gates. Inputs that are true (A and C assumed) will enable a true output from the corresponding gates, which in turn will set flip-flops A and C (positive input required). The other two flip-flops (B and D) remain in the reset state as determined in step a. The information at A, B, C, D is now stored in the four flip-flops.

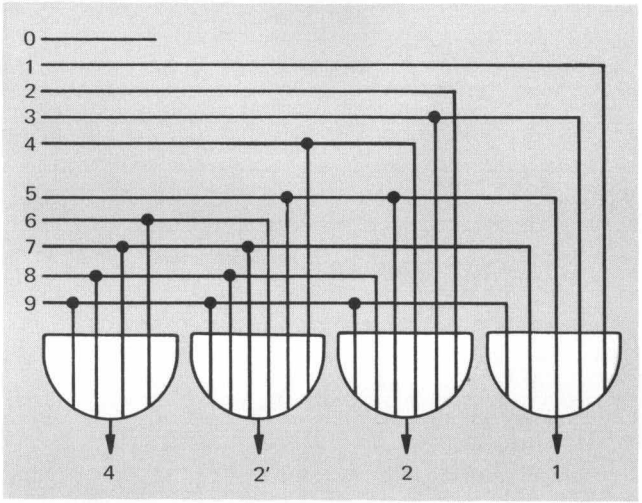
c. When the stored information is required, the Output Enable line is made negative, and output lines E and G from the set flip-flops are true (negative true, now). Outputs F and H will be false (positive). If positive-true outputs were required from these flip-flops, note that the reset outputs of the flip-flops could be used.

5-3. The above storage register is called a 4 X 1 register. The 4 denotes the number of bits per character; the 1 indicates the number of characters. Storage capacity is expanded by adding additional 4 X 1 registers, one for each added character. The expanded register is then called a 4 X 2, 4 X 3, etc. register. If the characters require more than four bits, for example by an added flag bit, the register is increased to 5 X 1, 5 X 2, etc. by the addition of storage flip-flops and associated gates.

5-4. By using one store pulse, parallel data is stored. Then by successively enabling the register for each character, parallel-to-serial conversion is obtained (corresponding bits of each register are "or" gated together). The process of successively enabling character storage register is called "commutation".

5-5. ENCODER.

5-6. The encoder (or matrix) converts multiple-line information to binary coded form. The encoder shown at right converts ten-line decimal information to 4-2'-2-1 BCD code. If, for example, input line 5 is true, output lines 2', 2, and 1 (coded form of the number 5) will be made true.



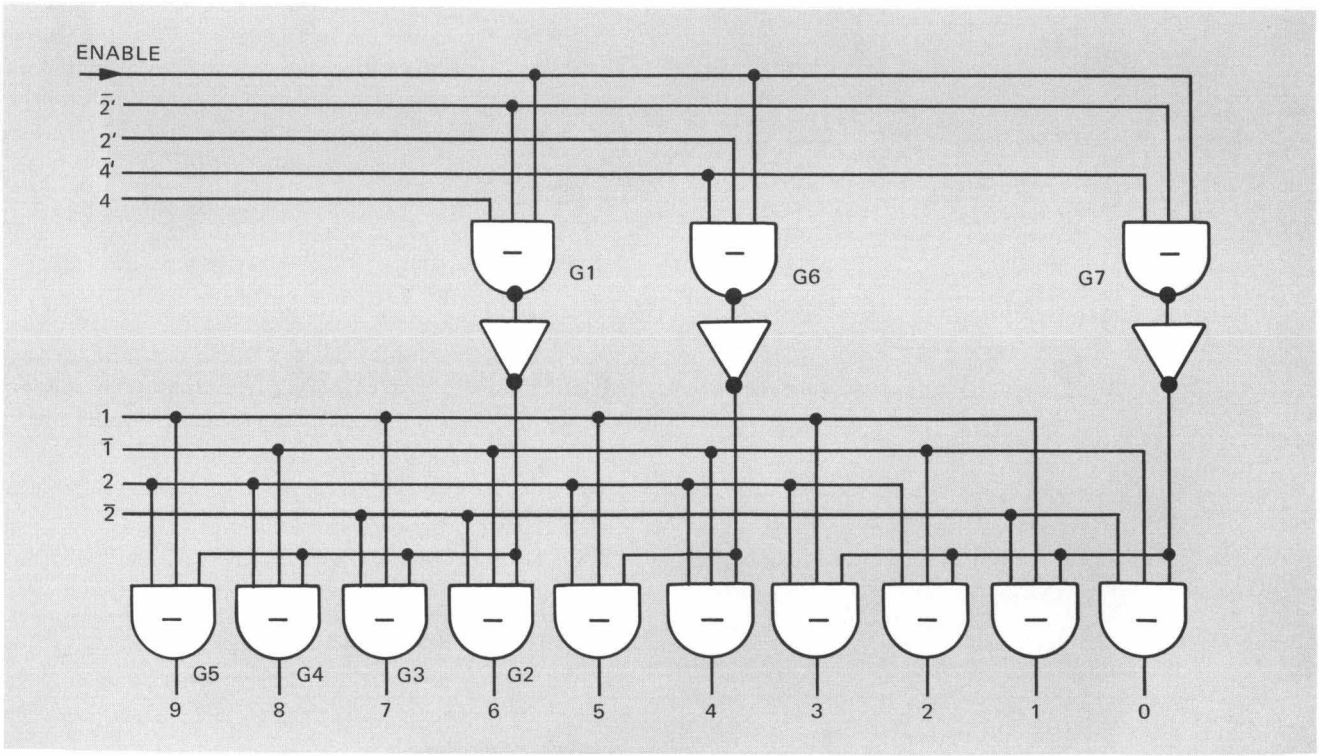
5-7. An encoder may have any number of inputs (only one of which is true at one time) and any number of binary bits on the output (various combinations of which may be made true). For example, an encoder is used to convert single-line numeral/function information to IBM eight-level code. Encoders are often shown simplified on logic diagrams as a rectangular box with the type of matrix (Ten-line to 4-2'-2-1, etc.) indicated within the symbol.

5-8. DECODER.

5-9. The decoder is the opposite of an encoder; i.e., the decoder converts binary information to multiple-line form. A typical decoder is shown below. Input to the decoder is weighted 4-2'-2-1 negative true. Note that for this type of decoder, $\overline{4}$-$\overline{2}$'-$\overline{2}$-$\overline{1}$ inputs are also required. Output is positive true.

5-10. Decoder operation can be seen by analyzing the operation for a BCD input, say seven (1-1-0-1 for 4-2'-2-1). If the Enable line is true, then G1 is enabled. Gates G6 and G7 are disabled by the 4 line, thus only G2 through G5 can be enabled. With the 2 and 1 bits true, G3 is the only gate of this group that has all three inputs true. Thus output line 7 is the only true output.

5-11. Different logic arrangements may be used for decoding other input codes (8-4-2-1, etc.), and for enabling special outputs, such as instrument functions.



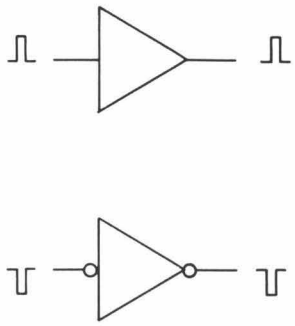
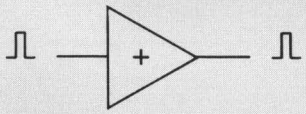
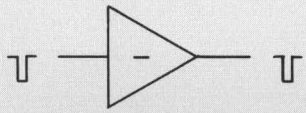
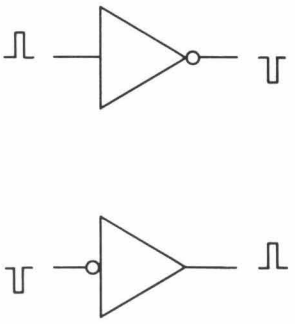
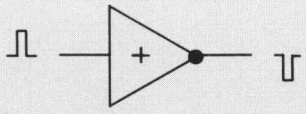
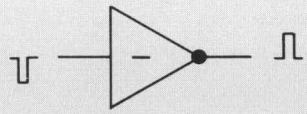
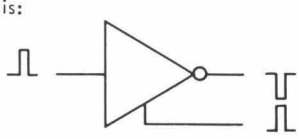
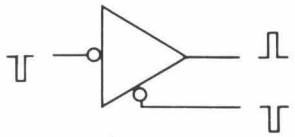
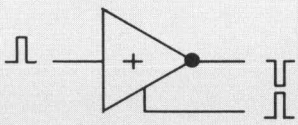
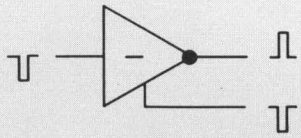
02415-1

H-P LOGIC DICTIONARY

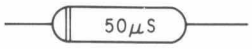
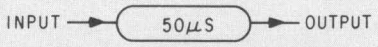
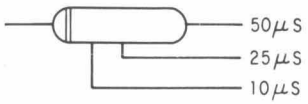
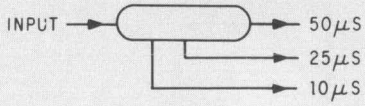
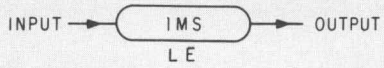
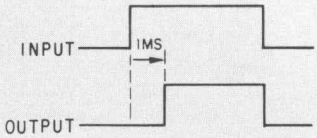
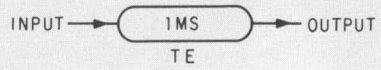
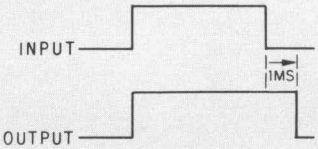
introduction

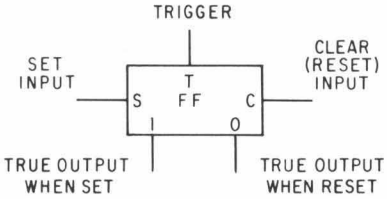
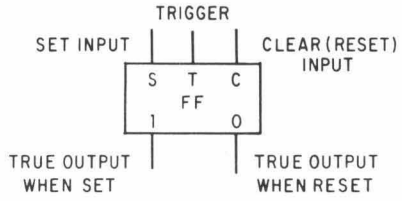
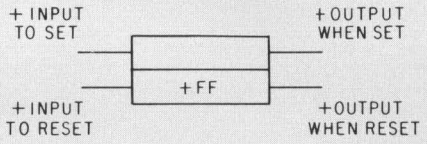
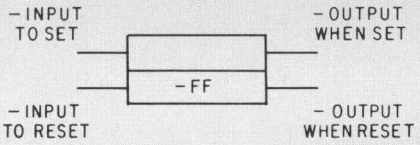
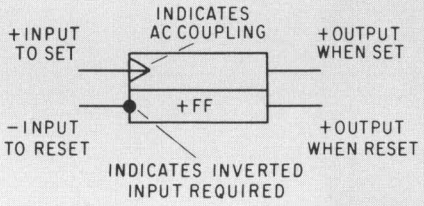
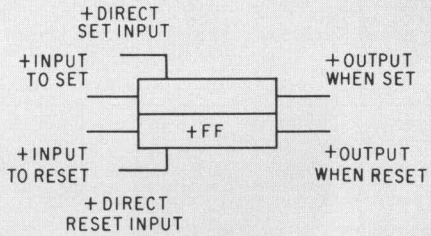
This dictionary is a direct comparison of the logic symbols of MIL-STD-806B with the logic symbology used for digital computers and related devices. This adaptation is designed to achieve the following objectives:

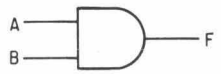
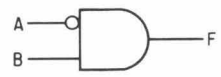
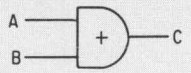
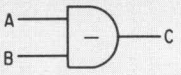
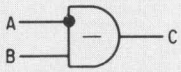
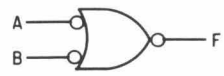
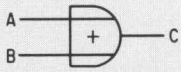
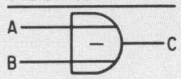
1	**VERSATILITY.** The H-P system includes specific symbol notations for identification of ac coupled input, positive true and negative true logic, polarity inversion (at input or output), and identification of triggers.
2	**CLARITY.** Complementary output devices such as flip-flops, one-shots, and Schmitt triggers are symbolized by two stacked rectangles, which immediately identifies their complementary character. Each device is identified specifically by the abbreviation FF, OS, or ST.
3	**COVERAGE.** The H-P system includes an encode gate (inverted OR gate used to distribute a single function to several outputs) that is not included in the MIL-STD-806B logic.
4	**SIMPLICITY.** Only four symbol envelopes are required in the H-P system versus six symbol envelopes required by the MIL-STD-806B system. Positive true logic is shown by a simple plus sign (+), negative true logic by a simple minus sign (-). Inversion is shown by a single dot.

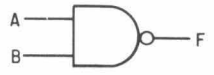
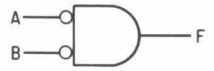
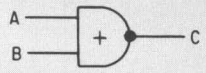
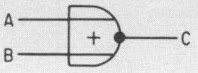
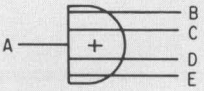
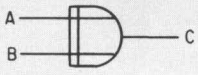
LOGIC ELEMENT	MIL-STD-806B SYMBOLS	HEWLETT-PACKARD SYMBOLS
AMPLIFIER (non-inverting)		AMPLIFIER POLARITY SIGNS ARE OPTIONAL  Use "+" if amplifier is "on" only with positive-true input.  Use "-" if amplifier is "on" only with negative-true input.
AMPLIFIER (inverting)		 
AMPLIFIER (phase splitter)	No Specific Symbol is provided by 806B, but this:  or this:  could be used	 

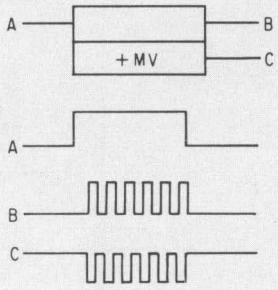
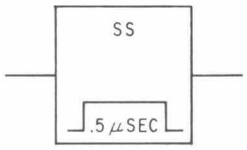
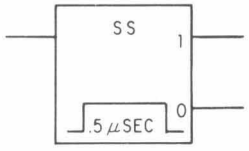
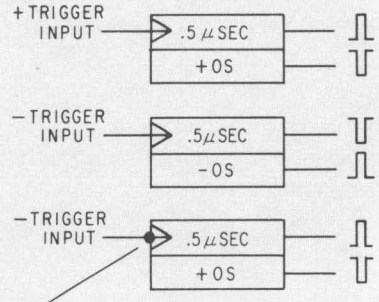
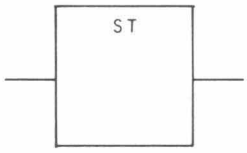
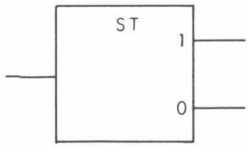
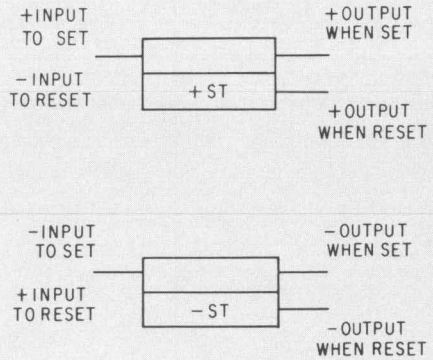
NOTE: Waveforms are not part of the symbols; they are shown to illustrate the logic

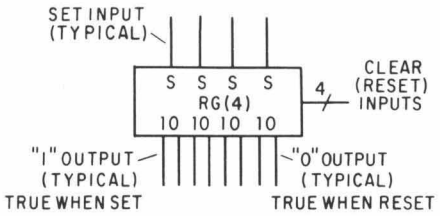
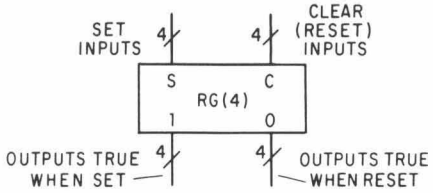
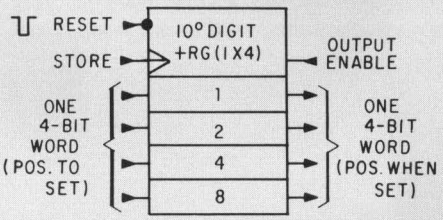
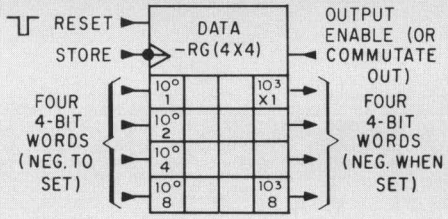
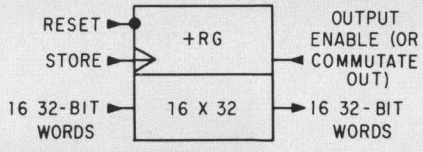
LOGIC ELEMENT	MIL-STD-806B SYMBOLS	HEWLETT-PACKARD SYMBOLS
DELAY		
Basic		
Tapped		
Leading Edge	No provision in 806B for distinguishing LE (Leading Edge) delay (delay introduced only when input signal is applied)	
	or	
Trailing Edge	TE (Trailing Edge) delay (delay introduced only when input signal is removed)	
		

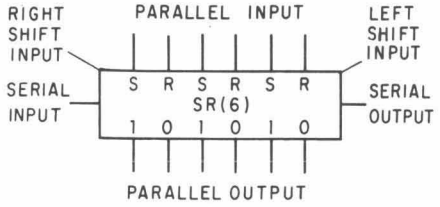
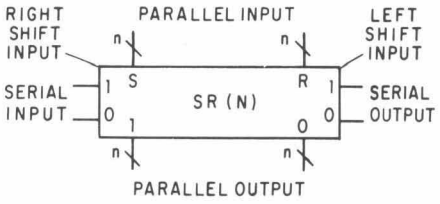
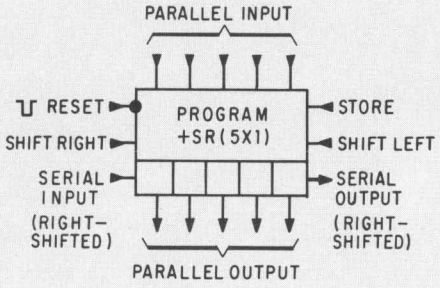
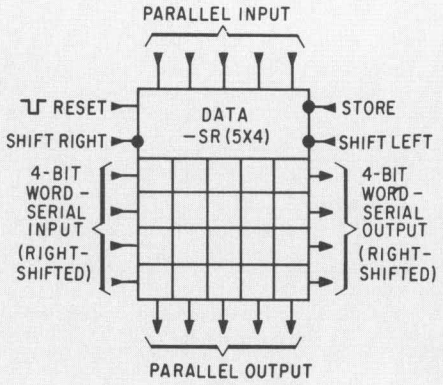
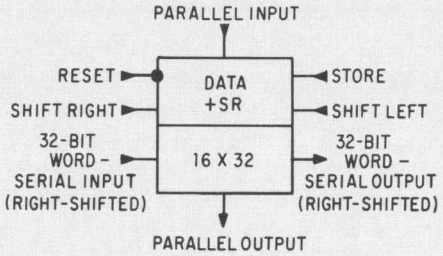
LOGIC ELEMENT	MIL-STD-806B SYMBOLS	HEWLETT-PACKARD SYMBOLS
FLIP-FLOP	 OR 	POSITIVE SET  NEGATIVE SET  MIXED INPUT (POSITIVE SET SHOWN)  MULTIPLE INPUT (POSITIVE SET SHOWN) 

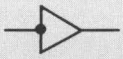
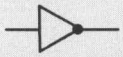
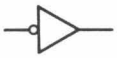
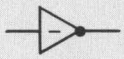
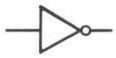
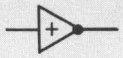
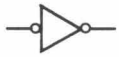
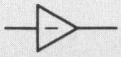
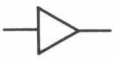
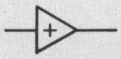
LOGIC ELEMENT	MIL-STD-806B SYMBOLS	HEWLETT-PACKARD SYMBOLS
GATES "AND"	BASIC  A AND B HIGH = F HIGH MIL-STD-806B DOES NOT DISTINGUISH POSITIVE TRUE FROM NEGATIVE TRUE MIXED  A LOW AND B HIGH = F HIGH	POSITIVE TRUE  ALL INPUTS (A AND B) TRUE (+) MAKES OUTPUT (C) TRUE (+) NEGATIVE TRUE  ALL INPUTS (A AND B) TRUE (-) MAKES OUTPUT (C) TRUE (-) MIXED  A FALSE (+) AND B TRUE (-) MAKES OUTPUT (C) TRUE (-)
GATES (Cont'd) "OR"	 A OR B HIGH = F HIGH  A OR B LOW = F LOW	POSITIVE TRUE  INPUT (A OR B) TRUE (+) MAKES OUTPUT (C) TRUE (+) ALL INPUTS FALSE (-) MAKES OUTPUT (C) FALSE (-) NEGATIVE TRUE  INPUT (A OR B) TRUE (-) MAKES OUTPUT (C) TRUE (-) ALL INPUTS FALSE (+) MAKES OUTPUT (C) FALSE (+)

LOGIC ELEMENT	MIL-STD-806B SYMBOLS	HEWLETT-PACKARD SYMBOLS
GATES (Cont'd) "NAND"	 A AND B HIGH = F LOW  A AND B LOW = F HIGH	(POSITIVE TRUE SHOWN)  ALL INPUTS (A AND B) TRUE (+) MAKES OUTPUT (C) FALSE (-) ANY INPUT (A OR B) FALSE (-) MAKES OUTPUT (C) TRUE (+)
GATES (Cont'd) "NOR"	 A OR B HIGH = F LOW  A OR B LOW = F HIGH	(POSITIVE TRUE SHOWN)  ANY INPUT (A OR B) TRUE (+) MAKES OUTPUT (C) FALSE (-) ALL INPUTS FALSE (-) MAKES OUTPUT (C) TRUE (+)
GATES (Cont'd) "Encode"	NO SYMBOL IS PROVIDED BY MIL-STD-806B	(POSITIVE TRUE SHOWN)  INPUT (A) TRUE (+) MAKES ALL OUTPUTS (B, C, D, E) TRUE (+)
GATES (Cont'd) "Exclusive OR"	 F IS HIGH IF AND ONLY IF ANY ONE OUTPUT IS HIGH AND ALL OTHER OUTPUTS ARE LOW	 ONE INPUT (A OR B) TRUE MAKES OUTPUT (C) TRUE BOTH INPUTS (A AND B) TRUE OR FALSE MAKES OUTPUT (C) FALSE

LOGIC ELEMENT	MIL-STD-806B SYMBOLS	HEWLETT-PACKARD SYMBOLS
MULTIVIBRATOR Astable	No specific symbol provided by MIL-STD-806B	(POSITIVE TRUE SHOWN) 
MULTIVIBRATOR One-Shot	ONE OUTPUT  OR TWO OUTPUT 	
SCHMITT TRIGGER	ONE OUTPUT  OR TWO OUTPUT 	

LOGIC ELEMENT	MIL-STD-806B SYMBOLS	HEWLETT-PACKARD SYMBOLS
REGISTER, BINARY	 	POSITIVE SET - 1 4-BIT WORD  NEGATIVE SET - 4 4-BIT WORDS  NOTE: The dimensions of the storage array with the ±RG heading are optional. Where detailed presentation is impractical, a generalized symbol may be used, as follows: GENERALIZED SYMBOL (POSITIVE SET - 16 32-BIT WORDS) 

LOGIC ELEMENT	MIL-STD-806B SYMBOLS	HEWLETT-PACKARD SYMBOLS
SHIFT REGISTER	 OR 	POSITIVE SET - 5 1-BIT WORDS  NEGATIVE SET - 5 4-BIT WORDS  NOTE: The dimensions of the storage array with the ±SR heading are optional. Where detailed presentation is impractical, a generalized symbol may be used, as follows: GENERALIZED SYMBOL (POSITIVE SET - 16 32-BIT WORDS)  NOTE: Either shift input may be omitted if not used.

LOGIC ELEMENT	MIL-STD-806B SYMBOLS	HEWLETT-PACKARD SYMBOLS
NOTATIONS		
Positive True	None Specified	+
Negative True	None Specified	−
Information Flow		
Polarity Inversion	None Specified	● (at output)
Inverted input required	None Specified	
AC coupling at input	None Specified	
Inverted AC input	None Specified	
Inverted output	None Specified	
Low input, high output		
Low output, high input		
Low input, low output		
High input, high output		
Trigger Identification Leading Edge	None Specified	⌐LE - this symbol at input of logic device indicates triggering by positive-going leading edge of pulse ¬LE - this symbol indicates triggering by negative-going leading edge of pulse
Trailing Edge	None Specified	⌐TE - indicates triggering by positive-going trailing edge of pulse ¬TE - indicates triggering by negative-going trailing edge of pulse

395 Page Mill Road, Palo Alto, California 94306 Area Code 415 326-1755

02415-1

Printed in U.S.A.