ACHIEVEMENT THROUGH ELECTRONICS

TRAINING KIT MANUAL

9K

NATIONAL RADIO INSTITUTE • WASHINGTON, D. C.

INDEX OF SECTIONS

NATIONAL RADIO INSTITUTE, WASHINGTON, D.C. 20016

FP5C975 1975 EDITION Litho in U.S.A.

INSTRUCTIONS FOR PERFORMING EXPERIMENTS 81 THROUGH 90

At this point you have thoroughly studied most of the basic digital logic circuits that are used in practical digital systems. You have studied both sequential and combinational logic networks, particularly the most common ones used in computers and digital instruments. In this kit you will learn additional digital techniques and applications.

In the previous two kits you demonstrated digital logic fundamentals at the gate and functional circuit levels. It is easier to learn how different types of digital circuits operate when they are demonstrated individually. However, in this kit you will begin to combine these circuits to form larger circuits that perform some useful functions, involving arithmetic computations by digital techniques. These circuits will be more sophisticated than those you have worked with in the past. As a result you will be interconnecting more circuitry, using the experimental chassis and as many as five of the computer printed circuit boards at a time. There will be numerous interconnections to make and you should be extremely careful to make these properly. Complete and detailed instructions are given for each experiment in wiring these circuits. If you follow the directions carefully, you should have no trouble in getting the circuits to work.

Despite the seemingly complex nature of some of these experiments, they are typical of what you will encounter in industry. As a technician, you will be working with even more sophisticated equipment. In many cases the experimental circuits that you wire here are identical to typical breadboarding setups that you will make as a development technician.

The experiments in this kit are designed to teach you memory applications and arithmetic circuitry. In the first three experiments, you will build and use one type of digital memory. You will demonstrate read and write operations and use the memory to perform a wide variety of logic functions. The memory that you will build is known as the read-only-memory (ROM) that finds wide application in modern digital systems. The memory that you construct is a programmable read-only-memory that permits you to vary its function easily and demonstrate important concepts. After building this memory and demonstrating it in the experiments, you will test it thoroughly and use it as the main memory element in the digital computer that you will construct in the next kit.

Most of the remaining experiments cover arithmetic circuitry – logic circuits for performing addition, subtraction, and multiplication by digital methods. You will review basic binary arithmetic and construct circuits for performing these arithmetic operations. In addition, you will perform experiments covering BCD and other special arithmetic circuits.

The experiments in this kit will conclude your detailed study of digital circuitry. In the next kit you will construct the digital computer and use it to learn computer organization, operation, programming, and troubleshooting. In this kit as in the others, you will construct several of the printed circuit boards to be

used in the computer. You will use some of these boards in performing the experiments. In each case, construct the PC boards carefully. Because of the large amount of circuitry involved in the digital computer, problems can occur due to poor construction techniques. This means paying particular attention to your soldering.

As with your previous kits, check the contents against the photo in Fig. 1 and the related parts list. Be sure that you have all of the components so that you can proceed with the experiments without interruption. If you should be missing a part or some important piece of your kit, let the NRI Consultation Service know immediately.

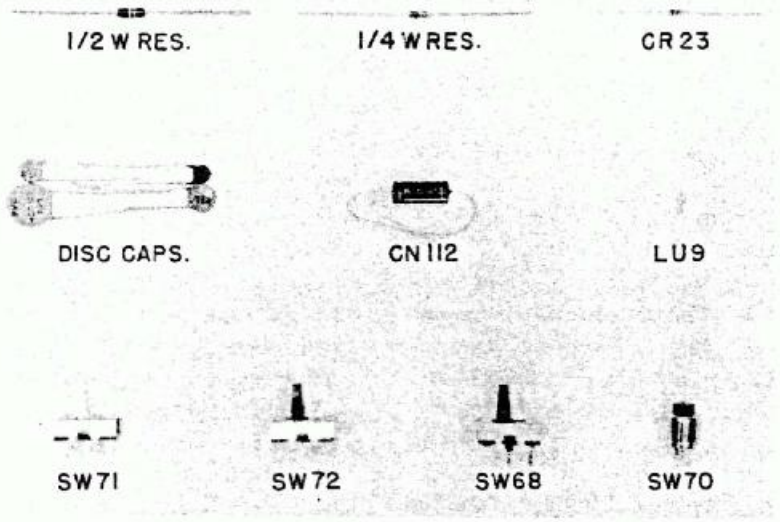


Fig. 1. Some of the experimental parts for this training kit are shown above.

Quan.	Part No.	Description	Price Each
31	CN102	.01 mfd, 50 volt disc cap.	.38
1	CN104	.1 mfd disc cap.	.36
6	CN112	100 mfd, 10 volt electrolytic cap.	.45
1	CN143	500 pf disc cap.	.15
3	CN218	.001 mfd disc cap.	.15
130	CR23	1N914 silicon diode	.40
1	EC32	C1 printed circuit board	4.00
1	EC34	A1 printed circuit board	4.00
1	EC35	A2 printed circuit board	4.00
1	EC36	M3 printed circuit board	15.50
1	EC37	C3 printed circuit board	5.00
2	HA80	10 ft. roll .032" solder	.75

Quan.	Part No.	Description	Price Each
9	IC5	7400 integrated circuit	1.00
3	IC6	7420 integrated circuit	1.00
6	IC7	15844 integrated circuit	1.00
1	IC9	7451 integrated circuit	1.00
2	IC10	7442 integrated circuit	3.50
5	IC12	7404 integrated circuit	1.00
2	IC13	7440 integrated circuit	1.00
1	IC14	74164 integrated circuit	3.75
1	IC17	7480 integrated circuit	1.75
2	IC18	7476 integrated circuit	2.00
1	IC19	7453 integrated circuit	1.00
1	IC20	7460 integrated circuit	1.00
1	IC21	7423 integrated circuit	1.50
1	IC25	74198 integrated circuit	7.80
225	LU9	Tubular terminals	12/.25
5	NU1	6-32 hex nut	12/.15
18	RE30	1k-ohm, 1/2 watt resistor	.15
4	RE33	22k-ohm, 1/2 watt resistor	.15
3	RE58	2.2k-ohm, 1/2 watt resistor	.15
1	RE166	4.7k-ohm, 1/4 watt resistor	.25
2	RE167	470-ohm, 1/4 watt resistor	.25
5	SC5	6-32 X 1" machine screw	12/.15
31	SO84	14-pin DIP socket	.75
5	SO86	16-pin DIP socket	.75
1	SO87	24-pin DIP socket	1.00
130	SW68	SPST slide switch, black	.25
1	SW70	Push button switch, black	.87
11	SW71	SPDT slide switch, white	.30
1	SW72	SPDT slide switch, red	.30
3	TS21	2N5134 transistor	.19
1	TS22	2N5138 transistor	.19
1	WR286	30 ft. #22 yellow wire	.12

Constructing the Memory Printed Circuit Board

Before you can proceed with the first three experiments in this kit, you will need to construct the memory printed circuit board. This is the large printed circuit board labeled EC36 that you received with this kit. Gather the following components in preparation for constructing this circuit board.

- 1 M3 circuit board (EC36)
- 4 14-pin integrated circuit sockets (SO84)
- 4 .01 mfd disc capacitors (CN102)
- 4 Type 15844 integrated circuits (IG7)
- 1 100 mfd, 10-volt electrolytic capacitor (CN112)
- 8 1k-ohm, 1/2 watt 10% resistors (RE30)
- 28 Miniature tubular terminals (LU9)
- 128 1N914 diodes (CR23)
- 128 Single-pole, single-throw PC mounting slide switches (SW68)
- 5 6-32 × 1″ machine screws
- 5 6-32 nuts
- Hookup wire
- Solder

The assembly of the circuit board is relatively simple. It is just a matter of installing the components in the proper locations and soldering them in place. The screening on the board designates the location of each of the components. By following the illustration, the board screening, and the instructions to follow, you should have no difficulty in constructing this board.

Install short wire jumpers in the J1 through J16 positions. A short piece of hookup wire is used to connect the copper pattern on the top of the board to another copper pattern on the bottom of the board at 16 different locations. This is done by passing a short piece of bare wire through the board and soldering it on both the top and the bottom sides. There are 16 individual positions. Simply strip the insulation from two pieces of hookup wire and push it through the hole from the top side of the board. Next, solder the wire to the conductor pad on the top side of the board, then solder the wire to the pad on the bottom side of the board. Finally, trim off the leads on both the top and bottom sides of the board.

Fig. 2 shows the proper procedure for doing this. Be careful not to use too much heat. Remember that the connection is so short that when you apply heat to solder the top connection, this will also melt the solder on the bottom connection.

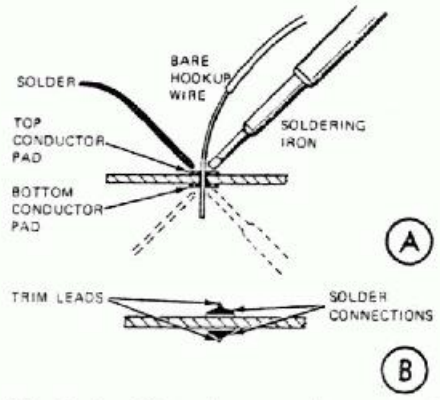


Fig. 2. Installing a jumper wire on the M3 circuit board. (A) soldering the jumper, and (B) the completed connection.

Install the 28 miniature tubular terminals in the peripheral holes provided on the left and bottom sides of the EC36 PC board. Insert the terminals from the top and crimp each tightly on the bottom to hold the terminal firmly in place. Solder all terminals on both the top and the bottom of the board.

Install a 100 mfd, 10-volt electrolytic capacitor between terminals G1 and +5B. Wrap the leads of the capacitor around these terminals neatly near the bottom, trimming off the excess. Solder both connections. Be sure to observe the polarity indicated by the screening on the top of the board.

Install four .01 mfd disc capacitors in the four locations designated to the right of the IC socket holes. Insert the capacitors from the top and mount them so that they are flush with the board. Solder both leads of each capacitor on the bottom of the board to hold them in position. Also in those locations provided, solder the capacitor leads on top of the board as well. As with the other circuit boards, these capacitor leads are used to carry power and ground connections from the top to the bottom of the board, making it very important for these capacitor leads to be soldered on both top and bottom. Double check your work here before you go on and clip off any excess lead lengths.

Install eight 1k-ohm, 1/2-watt resistors in the eight locations shown on the top of the EC36 circuit board. Bend the resistor leads at right angles as close to the body as possible. Insert each resistor from the top and push it down so that it is mounted flush with the PC board. Bend the leads slightly on the back and solder each of the leads. One lead of each resistor must also be soldered on the top of the board. Solder these connections carefully, then clip off any excess lead lengths.

Install four 14-pin DIP IC sockets in the locations designated by the screening on the circuit board. Insert the sockets from the top so that the notches are pointing to the left as indicated. Be sure that the sockets are flush with the top of the board, then using the thin solder supplied, solder the 14 pins on each socket.

You are now ready to install 128 1N914 diodes. The diode symbols are screened on top of the board with the correct polarity; however, you will mount the diodes on the bottom of the printed circuit board. As you mount each diode, be absolutely certain that you install it with the correct polarity. Remember that the black band around the end of each diode designates the cathode. In all cases, the cathode should be toward the top of the board.

Bend the leads of the diode at right angles very near to the body and insert the diode from the bottom of the board in the holes provided. These holes are marked with the diode symbol on top of the board. Bend the leads slightly on the top of the board to hold the diode flush with the board as you solder it. Solder each diode on both the bottom and the top of the board, but do not use too much heat because the diodes are very sensitive. Apply just enough heat to make a good solder connection, using the thin solder provided with this kit. Clip off the excess lead lengths. Install each diode this way. Once they are installed, scan all 128 diodes on the bottom of the board to be sure that all of the cathode ends are in the same direction, upward. Double check to be sure that all the leads are soldered on both the top and the bottom. Now be extremely careful when you lay the board

back down on your workbench. The diodes are made of glass and can be cracked. Therefore, from this point on the memory should be treated carefully so that no diodes on the bottom of the board will be damaged.

To prevent diode damage, install a one-inch long 6-32 screw and nut in each of the four corner holes of the PC board. Also install a one-inch screw and nut in the center hole. Insert the screw head from the top and tighten the nut on the bottom. This provides one-inch legs that will hold the memory board off the bench to prevent damage to the diodes.

Next install all of the 128 black slide switches on the top side of the PC board, beginning with the 16 switches across the top of the circuit board. Be sure to install them correctly so that the body of each switch fits within the areas screened on the PC board. Solder all of the switch terminals which protrude on the bottom side of the board. Since each switch has two terminals, a total of 256 solder connections must be made on the bottom side of the board. Be certain each switch rests squarely on the board before soldering.

Now install the four 15844 IC's at positions IC35, IC36, IC37 and IC38. Be sure that the IC and socket notches are properly aligned. Notice that all of the notches must point to the left.

The assembly of your M3 board is complete. You may now proceed with the first experiment in this kit to demonstrate the operation of this board.

Performing Experiments 81 Through 83

EXPERIMENT 81

Purpose: To demonstrate the operation of a programmable switch-diode matrix read-only-memory.

Introductory Discussion: Most digital computers today use magnetic core memories. These memories are extremely reliable, versatile, and can achieve the high speed necessary for fast computations. However, their complexity and high cost prevent us from demonstrating such a memory here. For that reason, we use a simpler, less expensive form of memory to help you in demonstrating memory concepts.

One of the simplest means of representing binary data is by using a switch. The switch has two positions, off and on, which we can assign to the binary 0 and binary 1 states. By providing a group of switches, complete binary words can be represented. This is exactly what we do in the memory board you have just constructed. This memory contains sixteen 8-bit words of binary data. A slide switch is used for each bit of the data word. There is a diode associated with each switch. The diode and switches are connected to form a large matrix. The diodes in series with the switches prevent the switches from interferring with one another. These switches are connected to the integrated circuits in groups, that are used to buffer the 8-bit parallel output word. The separate drawing (insert sheet) shows the detailed circuit of the M3 circuit board.

In this experiment you are going to demonstrate the switch memory you have just constructed. You will study its theory of operation so that you will thoroughly understand its use.

Experimental Procedure: The experimental circuit for this experiment is shown in Fig. 81-1. Before you begin wiring your circuit, you must partially disassemble the ADC circuit that is wired on your experimental chassis. First disconnect the resistive summing network from the M1 board, but do not unsolder the individual 100k-ohm and 200k-ohm resistors in the network. Also remove the wire going from this network to terminal IC1-4 on the experimental chassis PC board. Set the resistive network with its attached wire aside for later use. Leave the four yellow wires on terminals A, B, C and D of the M1 board connected to the lamp driver board. Disconnect the jumper between JA and KA, and remove the wire between KA and terminal IC1-10. Also, remove the yellow lamp driver lead from terminal IC1-10. Leave the other end connected to the lamp driver board.

Next remove the .001 mfd capacitor between terminals G1 and CLR on the M1 board. Then remove the 4.7k-ohm resistor between the +5A and CLR terminals; also remove the red push button switch with its attached wires simply by removing the yellow and black switch leads from the CLR and G1 terminals. Set aside the switch with its attached leads for later use.

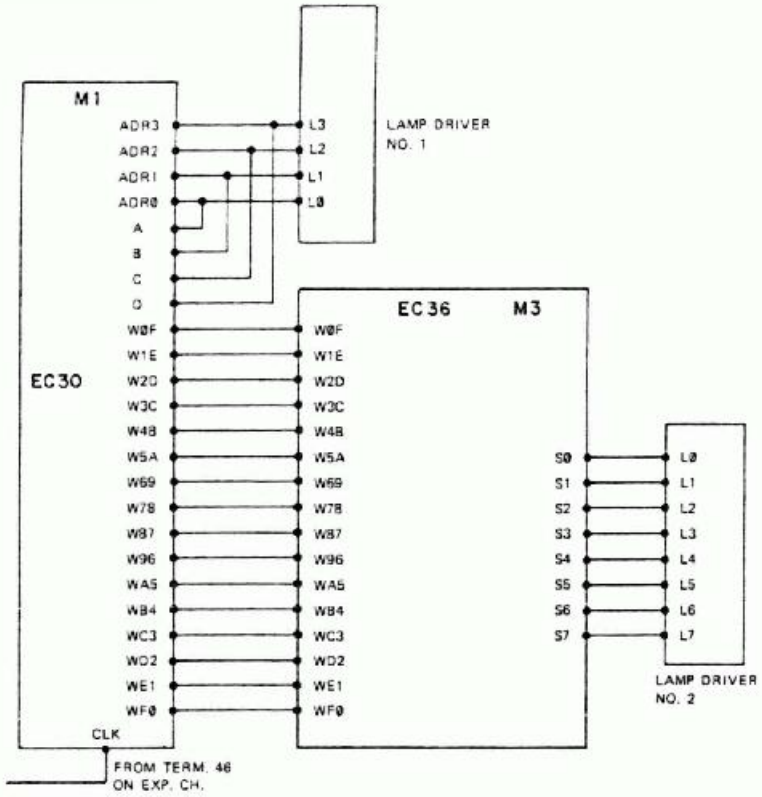


Fig. 81-1. Circuit for Experiment 81.

You will make one additional change on the experimental chassis PC board. Move the M1 clock wire from terminal 39 to terminal 46. This will apply the output of the buffered latch circuit to the CLK input on the M1 board.

Now you are ready to wire the experimental circuit shown in Fig. 81-1. In addition to the M1 board and lamp driver board which are now connected to the experimental chassis, you will need the M3 circuit board and the other lamp driver board. Wire them as shown in Fig. 81-1, using the yellow hookup wire.

You must provide power to the M3 board with an 18-inch piece of red hookup wire and an 18-inch piece of black hookup wire. Connect the red wire between terminal +5A on the M3 board and terminal 43 on the experimental chassis. Connect the black wire between terminal G2 on the M3 board and terminal 40 on the experimental chassis. Also connect the red and black power leads coming from the second lamp driver board to terminals 43 and 40 respectively on the experimental chassis.

Check all of your wiring to be sure that it is exactly as shown in Fig. 81-1. Also make certain that a 7400 IC is *not*

installed at the IC8 position on the M1 board. You are now ready to perform the experiment.

Step 1: To demonstrate basic memory operation.

The memory storage locations on the printed circuit board are organized into 8-bit word locations. These are numbered in hexadecimal, Ø through 9 and A through F down the center of the board, with switches Ø through 7 on the left and switches 8 through F on the right, reading from top to bottom. These designations are the addresses or locations of each memory word. Since there are a total of sixteen words, we can address these with a 4-bit address word. We feed a 4-bit address word to the memory, then that particular memory location is read out.

Turn on the experimental chassis. Depress the black push button and step the binary counter on the M1 circuit board until lamps I3 through IØ on the lamp driver board connected to lines A, B, C and D read 0000. Lamps I7 through I4 should be on at this time, but you can disregard them since they are not used here. You may disable these lamps if you wish by grounding L4 though L7 or by removing the bulbs. For the moment, you can also disregard the light pattern on the other lamp driver printed circuit board. Switch all eight switches in memory word Ø to the down position, noting the state of the lights on the light driver PC board that is connected to the SØ through S7 output terminals. Then switch the eight memory switches in word Ø to the up position one at a time, making note of the results on the indicator lamps as you do so.

Connect the ground clip of your tvom to one of the 1-inch screw legs on the memory PC board. This will permit you to pick up the common ground at a convenient location. Set the tvom for reading positive voltage on the +12 volt scale. Now touch the probe to the sixteen lines labeled WØF through WFØ one at a time. Mentally note the voltage at each of these sixteen pins.

Discussion of Step 1: Refer to the separate logic diagram of the M1 circuit board. In this experiment you are using the 4-bit binary counter on the M1 circuit board to provide an address for the switch memory. We take the 4-bit word from the binary counter and feed it directly to the memory address register. The strobing of the 4-bit address (by signal TØ) into the memory address register (7475) has been disabled because IC8 on the M1 circuit board has been removed. This means that the 4-bit address directly from the counter is applied through the memory address register flip-flops to the two 7442 decoders. These two decoders look at the 4-bit address, decode it, and generate a signal to enable one of the sixteen memory words.

In the first part of the experiment when you set the counter to the 0000 position, the decoder output selected word Ø in the memory. This is identified by the WØF line on the M3 circuit board. The decoder enables *only* this word.

With all eight switches of memory word Ø in the down position, the eight lamps on the light driver board should have been off, indicating binary 0's. With all of the switches in the up position all of the lamps should have been on, indicating binary 1's. The lamps should have turned on as you set the switches from the down (0) to the up (1) position one at a time. This indicates that on the

memory printed circuit board a switch in the down position is a binary 0 and a switch in the up position is a binary 1. The logical output signals from the memory correspond to the switch indication as you observed on the lamp driver lamps.

By switching the individual switches off and on you should have found that the lamp corresponding to that bit position in the word changed as you switched between the binary 0 and binary 1 positions.

Next you used your tvom to measure the voltage at the sixteen address lines coming from the M1 circuit board. You should have found all of the lines at the binary 1 voltage level (between +4 and +5 volts) except for one, the WØF line. The WØF line should be in the binary 0 state (near zero volts). The decoder on the M1 circuit board selects one of the sixteen words, the selected line being a binary 0 while all others are at the binary 1 indication. You can always determine which word on the memory printed circuit board has been enabled by looking at the voltage on the address input lines. The line that is in the binary 0 position is enabled.

To understand how the switch memory works, refer to the memory printed circuit board logic diagram on the separate insert sheet. Notice the large diode-switch matrix. Each bit position consists of a switch and a diode connected in series to one of the 128 points formed by the 8 by 16 matrix. Eight lines feed the input node

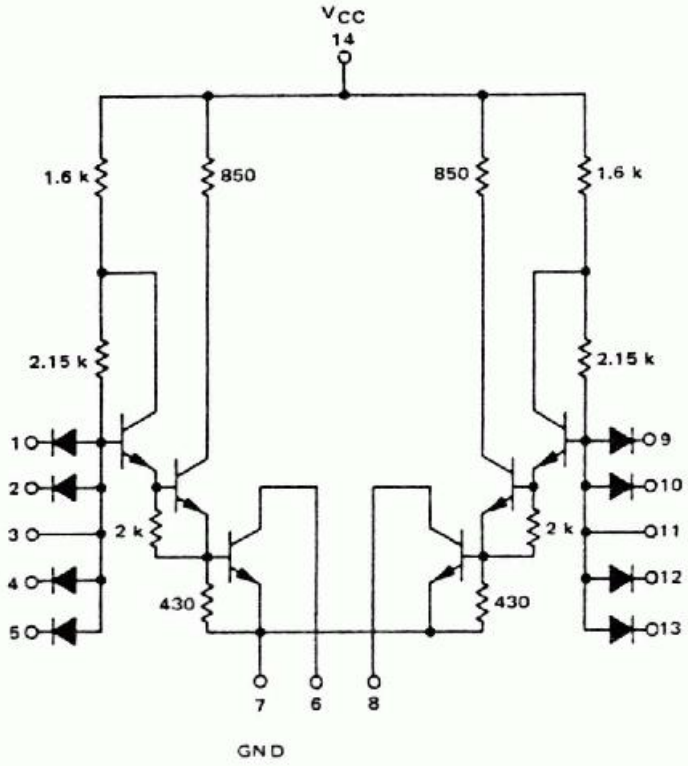


Fig. 81-2. Circuit of 15844 (or 944) IC used in the memory.

connections to the 15844 IC NAND gates used. The circuitry in each NAND gate is shown in Fig. 81-2. This is a DTL NAND gate. There are two 4-input gates with an open collector output in each IC. We are using external 1k-ohm load resistors for these NAND gates on the memory printed circuit board, but we are not using the input diodes on the NAND gate. These are connected together and enabled by a +5 volt line on the memory circuit board, as you can see in the illustration. However, we are using the node input, which is pins 3 and 11 respectively. As you can see in Fig. 81-2, the diodes in the matrix are connected to the gate node and are used to enable the gates, depending upon how the switches are set.

To fully understand how the memory works, recall the method of operation of the NAND gate. If any one of the diode inputs to the NAND gate is brought to ground (or made a binary 0), the output of the NAND gate goes high. If all of the diode inputs to the NAND gate are open or at the binary 1 level, the NAND gate output will be a binary 0.

Now refer to the memory board diagram. When the WØF address input line goes low, it will bring the cathode side of the diodes connected to it to ground, forcing the output of the corresponding NAND gate to be high if the switch in series with the diode is closed. Consider the NAND gate associated with output SØ in the diagram. Notice the diode connected to the switch attached to pin 11, the node of the NAND gate. The cathode of the diode is connected to the WØF input. If this switch is closed and WØF is grounded indicating that word Ø is addressed, the SØ output will be a binary 1. However, opening the switch will simply leave the node open and the NAND gate will have all of its normal diode inputs enabled. This will cause th SØ output to go low. As you can see, wit the switch closed the NAND gate outpu is forced high and a binary 1 is produced With the switch open, a binary 0 i produced. This same action occurs o each of the eight bit positions for th word Ø in the memory.

As you refer to the memory logi diagram, keep in mind that all othe address input lines at this time are at th binary 1 level. For that reason none o the other switches in the memory affec the state of the output. If the switches should be open they will have no effect on the circuit. If the switch happens to be closed, it will also have no effect since all of the other address lines are at a binary 1 level at this time. These conditions clearly enable the NAND gates in such a way as to permit only the diodes and switches in the addressed word to control the state of the output.

Step 2: To check the condition of all bits in every memory word.

In this step you are going to verify the operation of every memory word. You will address each memory word in sequence and test all bit switches to be sure that they are functioning properly. You have already tested memory location Ø by first setting the address to the 0000 state. You stepped the counter by depressing the black push button until the desired address was obtained. You placed all eight switches in the down position, then in the up position, and observed the outputs on the indicator lights. You will now test the remaining fifteen words in memory this same way.

Depress the black push button once to step the address counter to the 0001 position. You have addressed the memory

location number 1. Place all of the memory word 1 switches in the down position and observe the indicator lights. Now put all of them in the binary 1 position one at a time, noting that they should go on as they are switched to the up position.

Using the black push button, step the address to the 0010 position, repeating this test on memory location 2. First put all of the switches in the down position and observe the lights. Then place them in the up position and continue to watch the lights. Repeat this for all of the remaining memory word locations through address 1111 or hex F.

Discussion of Step 2: You sequenced through each of the memory locations and repeatedly wrote all 0's and all 1's into all eight bit positions of the memory location. You should have found that with all the switches down, that all of the eight indicator lamps are off. With the eight switches in the up position, the indicator lamps should be on. All memory locations should have given you exactly the same result. If you should find a lamp that did not respond properly to the memory switch in question, it is possible that you have a defective switch or diode.

If you should have trouble, follow this procedure. First check your solder connections for that particular switch and diode, by simply touching the hot soldering iron to the diode and switching terminals one at a time to let the solder remelt again and flow over the connections. Then retest the switch. Sometimes the switches themselves will have dirty contacts. This can often be cleared up by turning the switch off and on rapidly several times to clean off the dirt. If you should find a defective or intermittent switch, replace it with one of the extras we supplied you in this kit. With the experimental chassis power off, you can use your tvom in the ohmmeter mode to test each diode with a forward and reverse resistance test. Replace any defective diodes or reverse any that may have been installed backwards on the board.

Step 3: To become familiar with the memory switch positions and the method for writing data into the memory locations.

Step the counter to the 1011 position with the black push button. Then go to the appropriate memory location and set the switches from left to right according to the following pattern, 11001100. Observe the indicator lights. Treating this as a binary number, convert it into its decimal equivalent. Write your result in the margin of the text.

While you are observing the indicator lamps, try switching any of the other memory switches in locations other than 1011 and notice the effect on the indicator lamps.

Discussion of Step 3: The purpose of this particular step is to help you become familiar with the locations of the various memory word locations on the printed circuit board and how they correspond to the 4-bit input address. By setting the counter to the 1011 position, you addressed memory location B. This is the fourth group of eight switches down from the top on the right-hand side of the memory printed circuit board. You then set the bit pattern 11001100 from left to right at this memory location. The lamps should display the same bit pattern as the addressed switches. By interpreting this bit pattern as a binary number with the right-hand switch position and lamp I0 representing the LSB position, you should have been able to convert this binary number into its decimal equivalent of 204. In using the memory in the future, you will use this arrangement. The right-hand bit position in a word is the least significant bit while the left-most bit is the most significant bit. In some of the experiments that follow, the left-most bit will be the sign bit of a signed binary number while bit 6 will be the most significant bit of the number.

Instructions for Statement 81: For this statement you are going to demonstrate what happens when two of the memory words are addressed simultaneously. Solder a short piece of black hookup wire between terminals W78 and G2 on the M3 board. Then step the black push button so that you address memory location 0000. By connecting the piece of black hookup wire to ground, you have also addressed word 7 in the memory.

In word location Ø set the following bit pattern into the switches: 0011 0011. Set the pattern 0101 0101 from left to right in memory location 7. Observe the eight outputs on the lamp driver board and record your results in Fig. 81-3. Carefully study the words stored in memory locations Ø and 7 and the result obtained at the output, then answer the statement question.

MEM LOC Ø	0 0 1 1 0 0 1 1
MEM LOC 7	0 1 0 1 0 1 0 1
MEM OUT	

Fig. 81-3. Record your results for Statement 81 in this table.

After you answer the statement question, turn off the experimental chassis and remove the wire between terminals W78 and G2 on the memory board. However, leave the remaining circuits connected as they are used for your next experiment.

Statement No. 81: The binary number that I observed on the output lamps was:

(1) 1110 1110
(2) 0111 0111
(3) 1100 1100

which indicates that both memory words are being read out simultaneously since they have been effectively:

(1) ANDed
(2) added
(3) subtracted
(4) ORed

to produce the result.

EXPERIMENT 82

Purpose: To demonstrate a read-only-memory and to show several useful applications.

Introductory Discussion: A read-only-memory (ROM) is a special type of memory for storing fixed binary data. In other words, data is permanently written into each storage location. Once this is done, only read-out operations can be performed. Each word stored in the memory can be addressed either randomly or sequentially. As it is addressed, the data is read out of the memory.

In this experiment you are going to demonstrate several useful applications of

the ROM. The memory printed circuit board that you constructed earlier in this kit is a form of ROM. It is not a true ROM in the sense that data is permanently stored in it; it can only perform the read function. Data cannot be written into the memory under logic control, but the contents of the memory can be programmed by setting the switches to the desired bit pattern. For that reason the memory is best called a *programmable* read-only-memory. By being able to program the contents of the memory, we can illustrate many ROM applications.

Experimental Procedure: The test setup that you used in the previous experiment will be used in this experiment also. However, you will need to make a few modifications to the circuit.

First, remove the wire coming from the clock input on the M1 circuit board from terminal 46 on the experimental chassis. Solder it to terminal 39. This will permit the clock oscillator on the experimental chassis to step the binary counter.

Now you must remove some of the wiring on your experimental chassis. First remove the 1N914 diode and the 2N5134 transistor that are connected between terminals IC1-4 and IC1-10 on the experimental chassis PC board. Also remove the 220k-ohm resistor between terminal IC1-4 and terminal 16 on the 1k-ohm potentiometer, the wire between terminal 15 on the 1k-ohm potentiometer and terminal 9, and the wire between terminal 17 on the 1k-ohm potentiomenter and terminal 14.

Connect a 100k-ohm resistor between terminals IC1-4 and IC1-10. Then connect the DAC resistive summing network to terminals SØ, S1, S2 and S3 on the M3 circuit board, referring to Fig. 82-1 as you make these connections. Connect the wire coming from the resistive network to terminal IC1-4.

Check all of your wiring against Fig. 82-1. Notice that you have converted the ADC circuit that was wired on the experimental chassis into a DAC circuit, identical to the one you used in an earlier experiment. The 709 IC at position IC1 is used in conjunction with the 100k-ohm feedback resistor and the summing resistors to form the DAC circuit. Although not shown in Fig. 82-1, a wire should be connected between terminals 9 and IC1-6 and between terminals 13 and IC1-11. These two connections provide power to the IC. The rest of the experimental circuit is identical to the circuit used in your last experiment.

Step 1: To demonstrate a hybrid function generator.

Fig. 82-2 is a table showing the contents of each memory location for this step. Using the table as a guide, set the switches in all sixteen memory words to those specified in the table. Place the switches up for binary 1's and down for binary 0's. In order to obtain the proper results from this experiment, the memory words should be exactly as shown in the table. Therefore, it is a good idea to go back once you have programmed the memory to double check it. Notice that the four left-hand or most significant bits of each memory word are 0's for all portions of the step.

Turn on the experimental chassis. Connect the vertical input of your oscilloscope to terminal IC1-10, and connect the ground clip of the oscilloscope to the experimental chassis (ground). Adjust the vertical and horizontal controls to display at least one or preferably two cycles of the DAC output waveform on the screen.

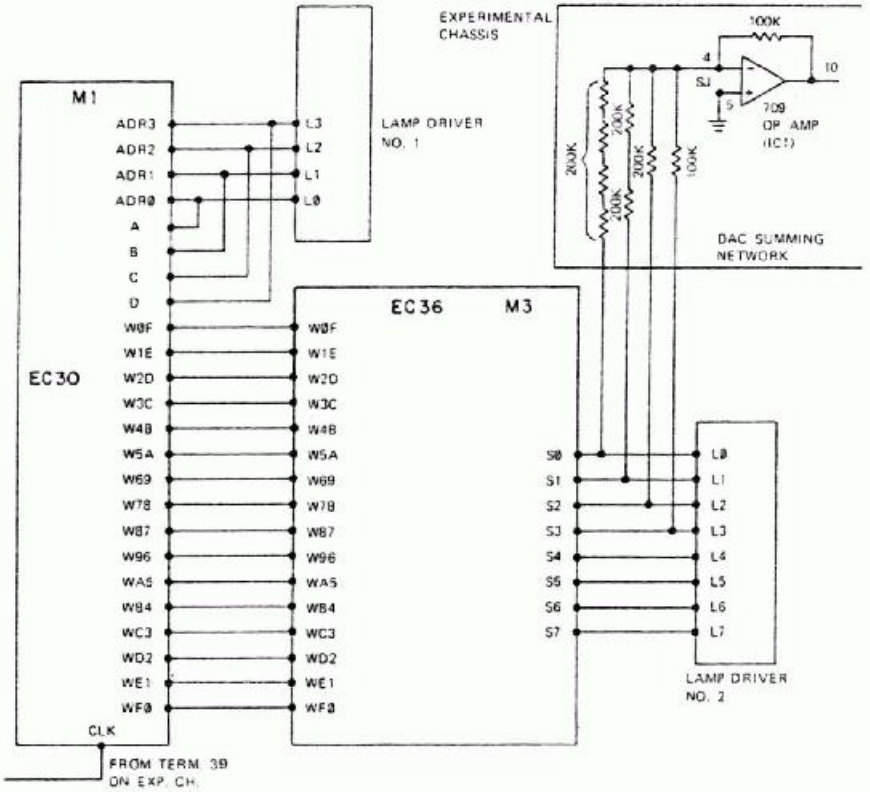


Fig. 82-1. Circuit for Step 1, Experiment 82.

MEM. LOC.	CONTENTS	MEM. LOC.	CONTENTS
0	00000111	8	00000111
1	00000100	9	00001010
2	00000010	A	00001100
3	00000001	B	00001101
4	00000000	C	00001110
5	00000001	D	00001101
6	00000010	E	00001100
7	00000100	F	00001010

Fig. 82-2. Memory contents for Step 1.

Adjust the controls to stabilize the waveform and then sketch the waveform on a separate sheet of paper.

Next, solder a 100k-ohm resistor between terminal IC1-10 and terminal 42 on the 4-lug terminal strip on the experimental chassis PC board. Solder a .1 mfd capacitor between terminals 40 and 42. Your circuit should appear as shown in Fig.82-3. Leave the ground clip of your oscilloscope on the experimental chassis, but move the vertical input lead to terminal 42. There is no need to turn off the experimental chassis while adding these components.

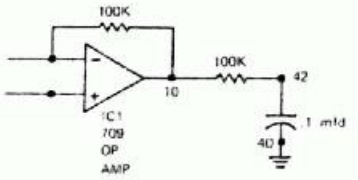


Fig. 82-3. Experimental circuit for addition in Step 1.

Observe the waveform at terminal 42. Readjust the vertical gain control and other controls as necessary to stabilize the waveform. Again note the waveform shape and copy it on a separate sheet of paper beneath the waveform you just copied. Turn off the experimental chassis.

Discussion of Step 1: In this step you are using the four least significant output bits of the programmable read-only-memory to drive the resistor summing network of the digital-to-analog converter that you constructed in previous kits. The DAC will accept the words stored in memory and generate an output voltage proportional to the binary word value. We are using the binary counter stepped by the high speed clock circuit to sequentially address all sixteen memory locations. As the counter steps through its sixteen states, 0000 through 1111, the contents of each of the memory locations in sequence will be fed to the DAC. As the clock pulses step the counter, the counter will continually recycle from 1111 to 0000, repetitively transferring the memory contents in sequence to the DAC.

The output waveform produced by the DAC will be a function of the contents of each memory location. Since the memory is fully programmable, virtually any waveform can be generated. Because the DAC has only four input bits, we can use a maximum of four bit memory words and specify the magnitude in any sequence we desire.

In this experiment you stored a sequence of binary words in the memory like that shown in Fig. 82-2. This sequence of words closely approximates a sine curve. Therefore, the output waveform that you should have observed at the DAC output at terminal IC1-10 should closely approximate a sine wave. Naturally it is not smooth and perfect because of the stepped or discrete level nature of the DAC. However, for many practical applications such a simulated sine wave would produce a result just as good as that of a pure sine wave.

Fig. 82-4 shows the waveform that you should have observed. As you know, the DAC output voltage swings from a minimum of 0 volts to a maximum of −7.5 volts. The sine wave is set up to swing between 0 volts and 7 volts. Therefore, this is a peak-to-peak waveform of 7 volts. It is equivalent to a sine wave riding or superimposed upon a −3.5 volt dc level. The sine wave swings above and below this value.

If you will notice the waveform of Fig.

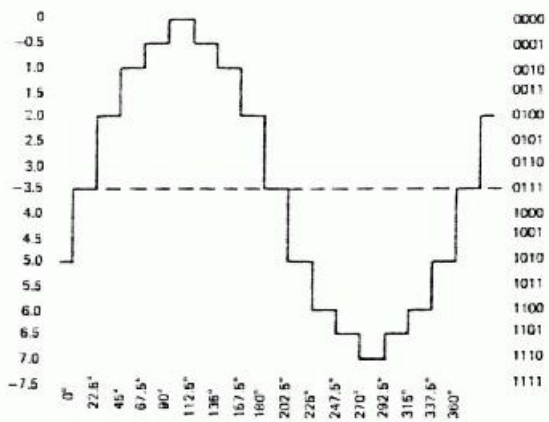


Fig. 82-4. Results for Step 1.

82-4, one cycle of the sine wave is split into sixteen time increments. Since there is a total of 360° for each sine wave cycle, each increment represents 360/16 = 22.5°. To generate this waveform, we program the memory with sixteen sequential words of data that are representative of the value of the sine of the angles 0° through 360° in 22.5° steps. Using these values of the sine, we scale the value to match the capabilities of the DAC circuit. As you can see, the results closely approximate the sine wave.

At the output of the DAC op amp you next connected a resistor/capacitor combination and observed the output voltage across the capacitor. This R–C network forms a low pass filter which will reject all of the high frequency components in the step waveform appearing at the op amp output. For that reason the output

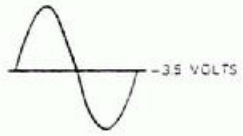


Fig. 82-5. Output of low pass filter on the DAC.

of the low pass filter at terminal 42 should be very nearly equal to a smooth sine wave. For those applications which require a more nearly perfect sine wave, filters such as this can be easily added. The waveform that you should have observed at terminal 42 is shown in Fig. 82-5.

Step 2: To show how an ROM can be used to perform binary multiplication by using table look-up techniques.

Remove the wires connecting terminals A, B, C and D to terminals ADRØ, ADR1, ADR2, and ADR3 on the M1 circuit board. Next, connect the four least significant bit positions of your switch register to the ADRØ through ADR3 lines. Connect the right-hand or LSB switch to the ADRØ terminal, etc. This will permit you to use these switches as the input to the read-only-memory rather than the binary counter on the M1 circuit board. Don't forget to connect the black and red wires from the switch register to ground and +5 volts on the experimental chassis before proceeding. Connect the

LOC	HEX CONTENTS
0	00
1	00
2	00
3	00
4	00
5	01
6	02
7	03
8	00
9	02
A	04
B	06
C	00
D	03
E	06
F	09

Fig. 82-6. ROM contents for Step 2 in hexadecimal form.

red lead to terminal 33 and the black lead to terminal 30.

In this step you are going to demonstrate the multiplication of binary numbers. You will use two 2-bit numbers, both of which will be represented by the four switches in the switch register that you just connected. The two right-most switches will represent the multiplicand and the two switches to the left will represent the multiplier. With 2-bit numbers there will be a total of four decimal numbers that can be represented, 0, 1, 2 and 3. For this step you will multiply these numbers by themselves in all possible combinations. To do this you will need to set the switch register switches to the various combinations of input numbers. You will then record the output on the lamp driver.

To begin the experiment, set the memory switches as shown in the table of Fig. 82-6, which shows the memory location or address in hexadecimal notation. Notice also that the contents of the memory are expressed in hexadecimal format. All of the programming of the memory that you will do from now on will be presented to you in hexadecimal form to simplify the programming. Since each word in the memory contains eight bits, it can hold two hexadecimal digits. As you recall, hexadecimal digits are those sixteen states represented by all possible combinations of a 4-bit number. For each hex digit in the table of Fig. 82-6, you will store in memory the binary equivalent in each 4-bit section of the word. For example, for memory location A the contents in hex is 04; therefore, you will store the following binary word in memory: 0000 0100.

MULTIPLIER	MULTIPLICAND	PRODUCT
00	00	
00	01	
00	10	
00	11	
01	00	
01	01	
01	10	
01	11	
10	00	
10	01	
10	10	
10	11	
11	00	
11	01	
11	10	
11	11	

Fig. 82-7. Table for Step 2.

Once you have programmed the memory according to Fig. 82-6, turn on the experimental chassis. Refer to the table in Fig. 82-7. This table shows you all possible combinations of the multiplier and multiplicand inputs that can occur with two 2-bit binary numbers. Set the switch register switches to the positions indicated for the multiplier and multiplicand. Record the 4-bit product in the space provided. You will monitor the product on the four right-hand lamps on the lamp driver No. 2 board. Once you have filled in the table, convert all of your binary numbers into their decimal equivalent. Then turn off the experimental chassis.

Discussion of Step 2: Figs. 82-8 and 82-9 show the results you should have obtained in this step. If you entered the

MULTIPLIER	MULTIPLICAND	PRODUCT
00	00	0000
00	01	0000
00	10	0000
00	11	0000
01	00	0000
01	01	0001
01	10	0010
01	11	0011
10	00	0000
10	01	0010
10	10	0100
10	11	0110
11	00	0000
11	01	0011
11	10	0110
11	11	1001

Fig. 82-8. Correct results for Step 2.

MULTIPLIER	MULTIPLICAND	PRODUCT
0	0	0
0	1	0
0	2	0
0	3	0
1	0	0
1	1	1
1	2	2
1	3	3
2	0	0
2	1	2
2	2	4
2	3	6
3	0	0
3	1	3
3	2	6
3	3	9

Fig. 82-9. Correct results for Step 2.

multiplier and multiplicand properly in the switch register and programmed the memory correctly, then for each of the two input numbers you should have obtained a product output displayed on the lamp driver.

The key to the operation of this multiplier is in how the memory has been programmed. If you will look carefully at Fig. 82-8 you will see that the 2-bit multiplier and multiplicand numbers, when combined, form a 4-bit binary number. These binary numbers are used to address a particular word in memory. The addressed memory location contains the product of the two numbers that constitute the address.

Instructions for Statement 82: For this statement you will need to make a few changes in your experimental setup. First, completely disconnect the switch register from the circuit and set it aside for later use. Reconnect the four wires between lines A, B, C and D and ADRØ through ADR3 on the M1 board as shown in Fig. 82-1. The lamp driver No. 1 lines should still be connected. In addition, move the wire from the CLK input on the M1 circuit board from terminal 39 back to terminal 46. This will permit you to step the counter with the black push button switch as you have done previously.

For this statement you are going to apply the decimal numbers 0 through 15 in binary form to the read-only-memory. These numbers, of course, will come from the binary counter on the M1 circuit board. Turn on the experimental chassis and step the counter with the black push button until the lamp driver reads 0000.

Now program the ROM according to the table in Fig. 82-10. Again the memory location or address and the

LOC	CONTENTS (HEX)
0	00
1	40
2	5A
3	6E
4	80
5	8F
6	9C
7	A9
8	B5
9	CO
A	CA
B	D4
C	DD
D	E6
E	EE
F	F7

Fig. 82-10. ROM program for Statement 82.

contents are given in hexadecimal notation. Be very careful in programming the memory to be sure that the switches are positively set to the up or down position as required by the program.

Next step the binary counter, using the black push button switch for each of the numbers 0 through 15, and record the output displayed by lamp driver No. 2 in the second column of Fig. 82-11.

To interpret the output data that you obtain from the ROM, you will assume that the 8-bit number stored in these memory locations consists of two whole number bits and six fractional bits. In other words, the binary point for the number will be placed between the second and third bit positions from the left. By referring to Fig. 82-11, you can see that the output you should record for a zero input is a string of zero's. However, note the position of the binary point. The

INPUT	OUTPUT (BINARY)	OUTPUT (DECIMAL)
0	0 0 . 0 0 0 0 0 0	0
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		

Fig. 82-11. Table for Statement 82.

numbers that you will record in here will have both integer and fractional parts.

After you have recorded all sixteen 8-bit output states, use you knowledge of the conversion of binary numbers to their decimal form and fill in the column provided in Fig. 82-11 with the decimal equivalent. When you have done this, study the decimal output equivalents and compare them to the decimal input number. By observing both the input and output numbers, you should be able to determine the relationship between them.

After you answer the statement question, turn off the experimental chassis. However, leave you wiring as it is since the same circuit will be used in your next experiment.

Statement No. 82: By carefully examining the output numbers obtained from the ROM in this statement, I find that the output number has the following mathematical relationship to the input number.

(1) The number obtained by dividing the input number by 3.

(2) Square of the input number.

(3) Square root of the input number.

(4) Common logarithm of the input number.

EXPERIMENT 83

Purpose: To demonstrate several additional applications of read-only-memories.

Introductory Discussion: In the previous experiment you saw how the ROM could be used for function generation and arithmetic processing. The ROM is capable of many additional operations. In this experiment you are going to demonstrate the application of an ROM performing basic logic operations, such as code conversion.

Experimental Procedure: At this point you do not need to make any further changes to your experimental circuit. However, you will be required to modify the program stored in the memory. Complete instructions are given to you as to how to do this for each step.

Step 1: To show how a read-only-memory can be used to perform basic logic functions and to implement Boolean techniques.

Program your memory according to the chart in Fig. 83-1. As before, the chart designates each memory location and the contents of that location in hexadecimal notation.

LOC	HEX CONTENTS
0	01
1	02
2	04
3	08
4	10
5	20
6	40
7	80
8	00
9	00
A	00
B	00
C	00
D	00
E	00
F	00

Fig. 83-1. Program for Step 1.

After you have programmed the memory according to the table, turn on the experimental chassis. Step the black push button until the counter is in the 0000 state, then step the counter through each of the sixteen states, 0000 through 1111, and record the memory output for each condition in the table of Fig. 83-2. Examine your results and try to determine the function that the ROM is performing. It is a common digital logic function that you have studied before, so make every effort to recognize it and write the result in the margin of the text. Refer to your earlier lessons or kit manuals if you should need to do so.

INPUT	OUTPUT
0000	
0001	
0010	
0011	
0100	
0101	
0110	
0111	
1000	
1001	
1010	
1011	
1100	
1101	
1110	
1111	

Fig. 83-2. Table for Step 1.

Discussion of Step 1: The correct results for Step 1 are shown in Fig. 83-3. The ROM is performing the function of an octal or one-of-eight decoder. An output is generated for only the first

INPUT	OUTPUT
0000	00000001
0001	00000010
0010	00000100
0011	00001000
0100	00010000
0101	00100000
0110	01000000
0111	10000000
1000	00000000
1001	00000000
1010	00000000
1011	00000000
1100	00000000
1101	00000000
1110	00000000
1111	00000000

Fig. 83-3. Results for Step 1.

eight states of the input. Since the left-most bit of this 4-bit input is a binary 0, we can consider the input word to be an octal number. The ROM looks at the input number, performing the function of a decoder by enabling one of the eight outputs. For example, with a 000 input the I0 lamp is on, indicating that the zero state has been decoded. With an input of 001, the I1 lamp is on, indicating that the 1 state has been decoded. When the I7 lamp is on, the 111 state is decoded. The other eight states (1000 through 1111) are not octal numbers; therefore, the memory has been programmed not to respond to them. The output for these eight states should be all zeros.

This decoder application of the ROM is a simple but frequently used one. It also clearly illustrates that the ROM can be programmed to implement any

Boolean algebra function. When a number of logic signals that are a function of a certain number of logical input signals are to be generated, an ROM can be used to implement all of the signals that are functions of these inputs. In other words, virtually any combinational logic function can be implemented with the ROM.

Step 2: To show how the ROM can be used to perform code conversion.

Program the memory according to the table in Fig. 83-4. By now you should be familiar with the hexadecimal programming scheme we are using, and the programming process should be simpler and quicker.

When you have programmed the memory, step the counter with the black push button through its sixteen states,

LOC	HEX CONTENTS
0	00
1	01
2	02
3	03
4	04
5	05
6	06
7	07
8	08
9	09
A	10
B	11
C	12
D	13
E	14
F	15

Fig. 83-4. Program for first part of Step 2.

INPUT	OUTPUT
0000	
0001	
0010	
0011	
0100	
0101	
0110	
0111	
1000	
1001	
1010	
1011	
1100	
1101	
1110	
1111	

Fig. 83-5. Table for first part of Step 2.

0000 through 1111, and record the 8-bit output word for each step in Fig. 83-5.

Next, reprogram the memory according to the chart in Fig. 83-6. Step the counter through its sixteen states and record the 8-bit output of the memory in the table provided in Fig. 83-7. Turn off the experimental chassis.

Now examine the results recorded in Figs. 83-5 and 83-7 and identify the output by comparing it with the binary input. Note your decisions in the margin of the text.

Discussion of Step 2: This step shows how the ROM can be used to perform code conversion. By carefully studying the results that you recorded in Fig. 83-5, you should have found that the 8-bit output of the ROM should be interpreted as two 4-bit BCD digits. In other words, the binary input applied to the ROM

LOC	HEX CONTENTS
0	03
1	04
2	05
3	06
4	07
5	08
6	09
7	0A
8	0B
9	0C
A	0D
B	0E
C	0F
D	10
E	11
F	12

Fig. 83-6. Program for second part of Step 2.

INPUT	OUTPUT
0000	
0001	
0010	
0011	
0100	
0101	
0110	
0111	
1000	
1001	
1010	
1011	
1100	
1101	
1110	
1111	

Fig. 83-7. Table for second part of Step 2.

from the counter on the M1 circuit boar causes the equivalent BCD number to b reproduced at the 8-bit memory outpu

In the next program you should hav found that the ROM is converting a 4-bi binary number into its equivalent XS code. Remember, to find the XS3 valu of a number, take the decimal equivalen of the binary number, add three to it then convert it back to binary.

As you can see, it is very easy t convert one code into another with a ROM. The input code is simply applied t the addressing lines of the ROM. The input code then addresses some particula memory location where the desired output code for that input is stored. With such an arrangement, virtually any form of code conversion can be handled with an ROM, which further proves that virtually any combinational logic network can be implemented with an ROM.

Instructions for Statement 83: In this statement you are going to show how the ROM can be used as a special decoder for driving a seven-segment decimal display. Fig. 83-8 shows a typical seven-segment decimal display. There are seven lighted segments labeled A through G. When we wish to display one of the 10 decimal numbers 0 through 9, we simply turn on

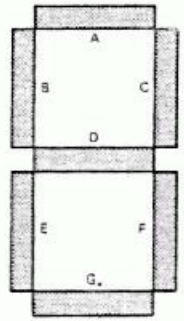


Fig. 83-8. Seven-segment decimal display.

DECIMAL	BCD INPUT	SEGMENTS A B C D E F G
0	0000	
1	0001	
2	0010	
3	0011	
4	0100	
5	0101	
6	0110	
7	0111	
8	1000	
9	1001	

Fig. 83-9. Table for Statement 83.

the appropriate segments of the display. For example, to display the number 1 we would turn on segments C and F. To display the number 6, we would turn on segments A, B, D, E, F and G. There would be a light driver and lamp associated with each of the segments. Our purpose here is to convert the 4-bit 8421 BCD input code into seven signals that can be used to operate the light drivers for each of the segments in the display.

To do this, simply consider the seven segment display in Fig. 83-8 and determine which segments should be on for each of the decimal numbers 0 through 9. Record your results in the table in Fig. 83-9. Place a binary 1 under the lettered column corresponding to the lamp segment that should be on for the specified decimal digit. All other segments will be off, so record a binary 0 in those columns. Once you have established the sequence in Fig. 83-9, you will readily see that you can use the ROM as a decoder to operate such a display.

For this statement question fill in the table in Fig. 83-10 giving *in hexadecimal* notation the contents of each of the desired memory locations for this application. Assume that the left or most significant bit of your memory output always remains at 0, and that only ten of the sixteen memory locations will be used.

After you complete the statement question you may proceed to the next experiment. Do not change any of your wiring until you are instructed to do so in your next experiment.

LOC	HEX CONTENTS
0	
1	
2	
3	
4	
5	
6	
7	
8	
9	

Fig. 83-10. Record your answer for Statement 83 here.

Statement No. 83: The table representing the BCD to seven-segment conversion program is shown in Fig.

(1) 83-11.
(2) 83-12.
(3) 83-13.
(4) 83-14.

LOC	HEX CONTENTS	
0	7	7
1	2	1
2	5	D
3	6	B
4	4	A
5	F	6
6	2	5
7	F	7
8	F	A
9	1	2

Fig. 83-11.

LOC	HEX CONTENTS	
0	7	7
1	1	2
2	5	D
3	5	B
4	3	A
5	6	B
6	6	F
7	5	2
8	7	F
9	7	B

Fig. 83-12.

LOC	HEX CONTENTS	
0	7	7
1	1	2
2	D	5
3	B	6
4	A	4
5	6	F
6	5	2
7	7	F
8	A	F
9	2	1

Fig. 83-13.

LOC	HEX CONTENTS	
0	7	7
1	2	1
2	6	B
3	5	D
4	4	A
5	2	5
6	F	6
7	F	7
8	F	A
9	2	1

Fig. 83-14.

Assembling the A1 and A2 Circuit Boards

Before you can proceed with the remaining experiments in this kit, you will need to assemble the A1 and A2 printed circuit boards. These boards contain the arithmetic circuitry for the digital computer that you will build in the final kit, but you will use the logic circuitry and registers on these boards in the experiments prior to this.

ASSEMBLING THE A1 BOARD

To assemble this board you will need the following parts:

- 1 Printed circuit board (EC34)
- 1 100 mfd, 10-volt capacitor (CN112)
- 44 Terminal lugs (LU9)
- 7 14-pin IC sockets (SO84)
- 1 16-pin IC socket (SO86)
- 7 .01 mfd disc capacitors (CN102)
- 1 500 pf disc capacitor
- 1 470-ohm, 1/4 watt resistor (RE167)
- 1 7420 IC (IG6)
- 1 7400 IC (IG5)
- 1 7404 IC (IG12)
- 1 7451 IC (IG9)
- 1 7476 IC (IG18)
- 2 7495 IC's (IG16)

ASSEMBLY PROCEDURE

Insert the 44 terminal lugs in the labeled peripheral holes from the top of the EC34 board. Press the terminals into the board so that they are seated properly, and then use your longnose pliers to crimp the terminal on the bottom of the board to hold it in place. Solder each of the terminals on the bottom of the board. Also solder the terminals on the top of the board where a copper pad appears.

Install a short jumper wire, designated J1, on the top of the board directly to the left of IC41. Bend a short piece of bare hookup wire to fit neatly into the holes, flush with the board. Solder both connections and clip off the excess lead length.

Install seven .01 mfd disc capacitors in the locations designated on the circuit board. These are installed adjacent to IC39, IC40, IC41, IC42, IC44, IC45, and IC46. Push the capacitor into the board from the top so that about 1/16" of the lead is exposed above the board, then solder the connections. Be sure to solder the leads on both the top and the bottom of the board since these leads carry the power and ground connection from one side of the board to the other. Clip off the excess lead length.

Install the 470-ohm resistor and 500 pf disc capacitor in the locations designated below IC41. Bend the leads of the resistor close to the body and mount the resistor so that it is flush with the board. Solder both the resistor and capacitor connections, then clip off the excess lead length.

Install a 100 mfd capacitor between terminals G1 and +5D. The positive terminal of the capacitor should go to the +5D terminal. Simply wrap the capacitor lead around the appropriate terminal and clip off the excess length after you solder it.

Install seven 14-pin IC sockets in the locations designated IC39, IC40, IC41, IC42, IC44, IC45, and IC46. Be sure that the notch in the end of the socket is aligned properly with the notch on the screening on the top of the board. Check to see that the socket is pressed into the holes firmly and that it is set flush with the board. Solder the IC socket terminals using the thin solder supplied with this kit.

Install a 16-pin IC socket in location IC43. Again double check to be sure that the notch in the end of the socket is pointing in the correct direction. With the socket mounted flush against the board, solder all 16 terminals with the thin solder.

Install the integrated circuits (except the 7480, IC46) as designated on the top of the board in the sockets. The two 7495 IC's are left over from Kit 7K. Be sure that each IC is aligned properly with respect to the socket and that it is pressed firmly into place to seat all pins.

As a final step it is a good idea to look over the board carefully once you complete it. Make sure that all of the components are in place, the sockets are aligned properly, the IC's are set in their sockets correctly and all connections have been soldered. It is particularly important to make a check of the solder connections. Check to see that the sockets are soldered and that there are no solder bridges between pins. Inspect the terminal lugs to be sure they are soldered on top of the board in those places where copper pads are provided.

ASSEMBLING THE A2 BOARD

To construct this board, you will need the following components:

- 1 Circuit Board (EC35)
- 1 100 mfd, 10 volt capacitor (CN112)
- 50 Terminal lugs (LU9)
- 7 .01 mfd disc capacitors (CN102)
- 1 470-ohm, 1/4 watt resistor (RE167)
- 1 .1 mfd disc capacitor (CN104)
- 1 4.7k-ohm, 1/4 watt resistor (RE166)
- 8 14-pin IC sockets (SO84)
- 1 16-pin IC socket (SO86)
- 1 24-pin IC socket (SO87)
- 2 7400 IC's (IG5)
- 1 7404 IC (IG12)
- 1 7420 IC (IG6)
- 1 7423 IC (IG21)
- 1 7453 IC (IG19)
- 2 15844 IC's (IG7)
- 1 7460 IC (IG20)
- 1 74198 IC (IG25)

ASSEMBLY PROCEDURE

Install the 50 terminal lugs in the marked holes on the top of the circuit board. Press the terminals into place to be sure that they are flush with the top of the board. Crimp each terminal with your longnose pliers on the bottom of the board to hold it firmly in place. Solder all terminals on both the top and the bottom of the board.

Install a short wire in the J1, J2 and J3 positions as designated on the top of the board in the block labeled accumulator. Cut short pieces of bare hookup wire the proper length and install them in the J1 and J3 positions. Use a short piece of insulated hookup wire at the J2 position.

This will prevent the jumper from accidentally shorting the +5 volt line to ground. Bend the leads neatly so that the jumper drops into the holes provided and rests flush with the board. Solder all connections and clip off excess lead length.

Install the 470-ohm resistor and .1 mfd disc capacitor at the locations designated in the block labeled accumulator. This is directly to the left of IC56. Bend the resistor leads close to the body of the resistor so that the resistor drops into the holes provided and fits flush with the board. Install the capacitor so that about 1/16" of lead is exposed above the board. Solder the resistor and capacitor leads and clip off any excess lead length. Be sure to solder the capacitor lead on top of the board where a foil pad is provided.

Install a 4.7k-ohm resistor in the holes provided directly to the right of position IC51. Again bend the resistor leads close to the body so that the resistor will drop into the holes and lay directly on the board. Solder the resistor leads and clip off the excess lead length. Solder the resistor at the pad provided on top of the board.

Install the seven .01 mfd disc capacitors in the locations designated. These are the positions directly to the right of IC47, IC48, IC49, IC50, IC52, IC53, and IC54. Drop the capacitor into the holes provided so that about 1/16" of the capacitor lead is exposed on top of the board. Solder the capacitor leads on both the top and the bottom of the board, then clip off any excess lead length.

Install the 100 mfd electrolytic capacitor between terminals +5C and G1. Wrap the leads around the terminals and solder them into place, then clip off the excess lead length. The positive terminal of the capacitor should connect to the +5C.

Install the eight 14-pin IC sockets in positions IC47 through IC53 and IC56. The notches on the sockets should be aligned with the notches on the screening on top of the board. As before, be sure that the socket is mounted firmly in place, flush with the board before you solder the pins on the bottom. Use the thin solder and the small tip, low wattage soldering iron to make the solder connections. Be extremely careful not to cause solder bridges between adjacent terminals, but use enough heat and solder to thoroughly cover the connection.

Install a 16-pin IC socket in location IC55. Be sure that the notch is aligned properly with the screened designation. Be sure that the socket is flush with the top of the board, then solder all 16 pins with the thin solder.

Install a 24-pin IC socket in location IC54. This socket cannot be used exactly as it is supplied. The two sections containing the pins must be broken apart from one another. You can use your side cutters to cut through the soft plastic holding the two strips together. Trim the excess plastic from each strip. Fig. 3 shows the before and after versions. Next, install the two 12-pin strips in the IC54

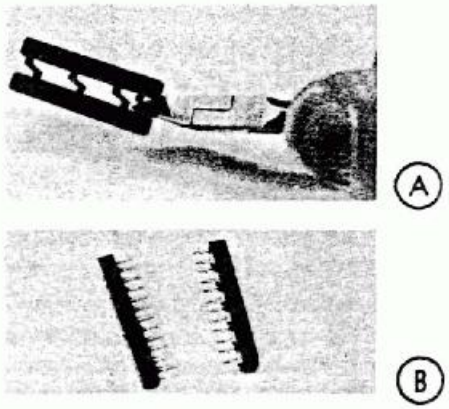


Fig. 3. Preparing the 24-pin IC socket.

position. Solder all 24 pins with the thin solder.

Install the designated IC's in the appropriate sockets. Note the alignment of the notch in the socket and the IC so that they correspond. Press each IC firmly into its socket so that it makes good connection. When installing the 24-pin IC, align pin 1 of the IC with the number 1 screened on the board.

Double check all of the connections on the circuit board. Be sure that you have soldered all terminal lugs on both the top and bottom where designated. Also check all of the solder connections on the IC sockets to be sure that there are no solder bridges between adjacent pins. When you have thoroughly examined this board, you are ready to continue the experiments in this kit.

Performing Experiments 84 Through 88

EXPERIMENT 84

Purpose: To demonstrate the principles of binary addition and examine several methods of accomplishing this function.

Introductory Discussion: Binary addition is the basic function performed within the arithmetic section of any digital computer. Other arithmetic functions such as subtraction, multiplication, and division are often performed by adding complements, using successive additions, or by using successive additions and subtractions. Since binary addition is so fundamental to all digital computations, it is extremely important for you to understand exactly how it is performed.

In this experiment you will demonstrate single-bit half adder and full adder circuits. You will actually wire a half adder circuit using NAND gates and demonstrate its operation. However, the full adder circuit that you will demonstrate will be contained within a single IC and will not require extensive wiring. You will also demonstrate how two binary words can be added serially.

Experimental Procedure: Before you can begin this experiment, you must partially disassemble the circuit that was used in your last experiment. First disconnect the resistive summing network from terminals SØ through S3 on the M3 circuit board. Also remove the wire coming from this network from terminal IC1-4 of the experimental chassis. Next, disconnect all of the wires on the M3 board including the red and black power and ground leads, which will also be disconnected from the experimental chassis. Set the M3 board aside for use in the next kit.

Now remove all of the wires on the M1 circuit board, except the red and black power and ground leads, which should be disconnected at the experimental chassis. Set the M1 board with its attached power and ground leads aside for later use. Next, disconnect lamp driver No. 2. Simply disconnect the power and ground leads associated with the board from the experimental chassis. Set the lamp driver board with its attached power and ground leads aside for later use. The power and ground leads associated with the lamp driver No. 1 should remain connected to the experimental chassis. The yellow wires coming from this board should hang free at this time.

Remove the loose wire from terminal 46 of the experimental chassis. Then unsolder the 100k-ohm resistor between terminals IC1-4 and IC1-10, the 1.5k-ohm resistor and .005 mfd capacitor between IC1-3 and IC1-12, the 200 pf capacitor between IC1-9 and IC1-10, the 100k-ohm resistor between terminals 42 and IC1-10, and the .1 mfd capacitor between terminals 42 and 40. Remove the wire from terminals 9 and IC1-6. Next remove the red wire from terminals 13 and IC1-11, immediately reconnecting this same wire between terminals 33 and IC1-14. Also, move the ground wire from terminal IC1-5 to terminal IC1-7.

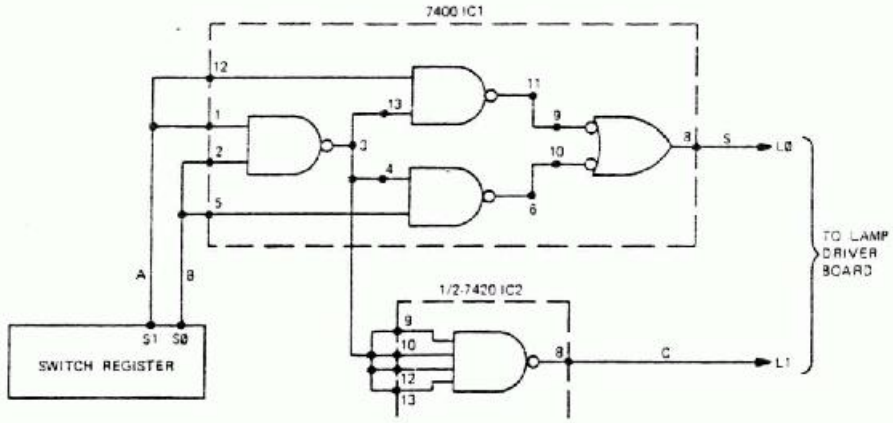


Fig. 84-1. Circuit for Step 1.

You can also remove the 709 IC from the IC1 socket and set it aside for later use. However, the 7420 IC should remain in the IC2 position.

Wire the circuit shown in Fig. 84-1. Using short pieces of black hookup wire, connect the terminals of IC1 exactly as shown. Connect terminals 1 and 12, 2 and 5, 3 and 4, 3 and 13, 9 and 11, and also 6 and 10 of the IC. Then connect terminals 9, 10, 12 and 13 of IC2 using short pieces of black hookup wire. Connect terminals IC1-3 and IC2-10, then connect your switch register as shown. Connect the yellow lead, coming from the extreme right switch (designated SØ) to terminal IC1-5. Connect the switch next to the right-most switch (designated S1) to terminal IC1-12. Also be sure to connect the power and ground leads coming from the switch register to terminals 53 and 50 on the experimental chassis. Now complete your wiring by connecting the LØ input lead on the lamp driver board to terminal IC1-8 and the L1 input lead to terminal IC2-8. The remaining input leads on the lamp driver board and the remaining leads on the switch register will not be used. As a last step, insert a 7400 IC into the IC1 socket. Be sure that the IC and socket notches are properly aligned.

The experimental circuit is now properly wired and you are ready to proceed.

Step 1: To demonstrate the operation of a half adder.

Turn on the experimental chassis and set switches S1 and SØ to the down or binary 0 position. This will apply a binary 0 to the A and B inputs of the experimental circuit. Observe the indication of lamps IØ and I1. Lamp IØ is being used to monitor the S output of the circuit and lamp I1 is used to monitor the C output. Keeping this fact in mind, record the binary output states of the S and C outputs in the spaces provided in Fig. 84-2.

Use the switch register to apply the remaining input combinations as shown in Fig. 84-2 and record the resulting S and C output levels as indicated by the lamps. When you have completed this chart, turn off the experimental chassis.

INPUTS		OUTPUTS	
A	B	S	C
0	0		
1	0		
0	1		
1	1		

Fig. 84-2. Chart for Step 1.

Discussion of Step 1: The circuit that you wired is referred to as a half adder, capable of adding two binary bits (A and B) to produce a sum, S, and also a carry, C, when applicable. Basically the circuit is nothing more than an exclusive OR gate and a separate inverter. All four of the gates in the 7400 IC are used to perform the exclusive OR function and one of the gates in the 7420 IC is used as the inverter. The exclusive OR produces an output that is equal to the sum, S, of the two binary inputs, A and B. The output of the exclusive OR may be expressed as $S = \bar{A}B + A\bar{B}$. The inverter provides the carry output, C. The carry output may be represented by the expression $C = AB$.

The truth table that you completed for this step should confirm the fact that the circuit does indeed add two binary bits and produce a carry when necessary. Notice that the sum output, S, will be a binary 0 when both inputs are at a binary 0. When one input is a binary 0 and the other a binary 1, the sum output will be a binary 1. However, when both inputs are at a binary 1, the sum output will be a binary 0 and a binary 1 will appear at the carry output, C. Notice that a binary 1 appears at the carry output only when both inputs are at a binary 1. These conditions satisfy all of the requirements for binary addition.

Although the circuit shown in Fig. 84-1 is capable of providing binary addition, its application is limited to situations where only two 2-bit binary numbers must be added. In other words only two inputs, A and B, are provided. This means that the circuit cannot be used to add multiple-bit binary numbers because there is no way of handling the carries which may occur from each pair of bits that are added. Since the circuit has this limited capability, it is usually referred to as a half adder.

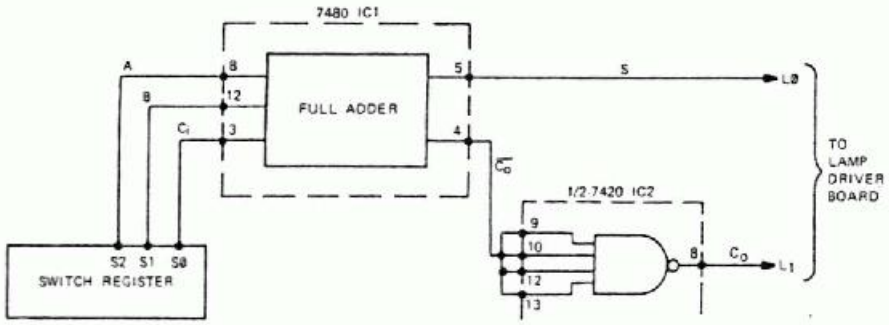


Fig. 84-3. Circuit for Step 2.

Step 2: To demonstrate the operation of a full adder.

Construct the experimental circuit shown in Fig. 84-3. First remove all of the wires on the terminals of the IC1 socket except the +5 volt and ground leads on terminals IC1-14 and IC1-7. Connect the yellow wires coming from switches SØ, S1 and S2, as shown. Remember that switch SØ is the right-most switch in the register. You will again be using one of the NAND gates in the 7420 IC. This gate is already wired in the IC2 position; however, you must connect the wire coming from the terminal IC2-10 to terminal IC1-4 as shown. Also you must reconnect the LØ input lead from the lamp driver board to terminal IC1-5. This will allow you to use lamps IØ and I1 to monitor the outputs from the experimental circuit. Finally, you must remove the 7400 IC from position IC1 and in its place install a 7480 IC.

A B C_i	S	C_o
0 0 0		
0 0 1		
0 1 0		
0 1 1		
1 0 0		
1 0 1		
1 1 0		
1 1 1		

Fig. 84-4. Table for Step 2.

Notice that your experimental circuit has three inputs labeled A, B and C_i, and two outputs labeled S and C_o. For this step you will apply the various input binary combinations shown in Fig. 84-4 to the experimental circuit. You will use switches S2, S1, and SØ in the switch register to apply inputs to the A, B and C_i inputs of the experimental circuit. You will monitor the S and C_o outputs on lamps IØ and I1 on the lamp driver board.

Now turn on the experimental chassis and apply the various input combinations shown in Fig. 84-4. Notice the indications of lamps IØ and I1 with each input combination and record your results in Fig. 84-4. Remember than an on lamp indicates a binary 1 and an off lamp indicates a binary 0. After you complete the chart shown in Fig. 84-4, turn off the experimental chassis.

Discussion of Step 2: In this step you demonstrated the operation of a full adder, capable of adding two 1-bit numbers, A and B, plus a carry input, C_i. The circuit produces a sum output, S, and also a carry output, C_o.

The 7480 IC that you used in this step contains the full adder circuit; however, the carry output from this integrated circuit is inverted and is, therefore, represented by the symbol $\overline{C_o}$ in Fig. 84-3. The NAND gate in the 7420 IC is wired as an inverter, used to invert the carry output to its normal condition which is represented by the symbol C_o. The sum, S, and the normal carry output, C_o, were monitored by lamps IØ and I1 respectively, while switches SØ, S1 and S2 were used to apply the various input combinations to the A, B, and C_i inputs. The symbols C_i and C_o should not be confused. They are used to distinguish between the carry input applied to the full adder and the carry output produced by the circuit.

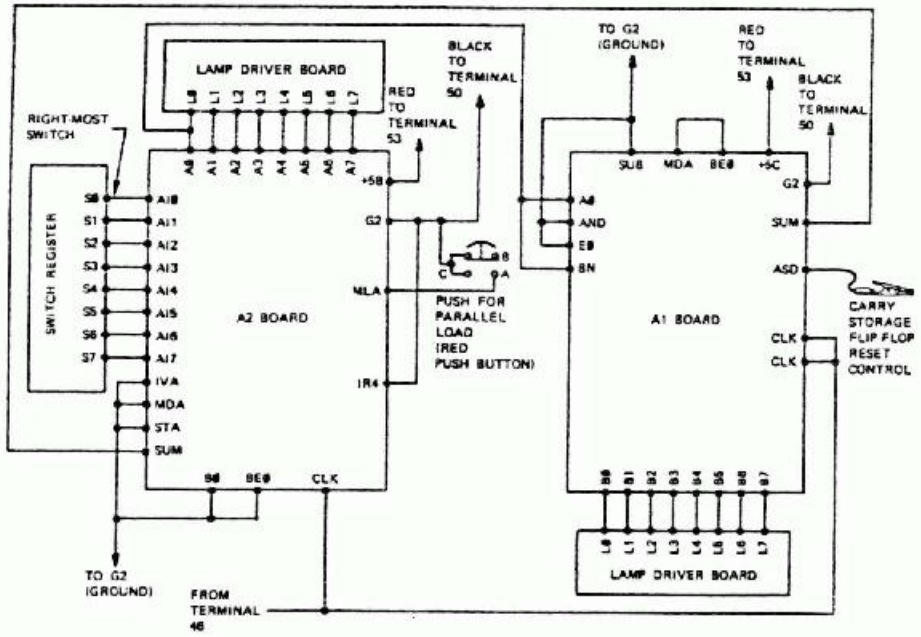


Fig. 84-5. Circuit for Step 3.

Step 3: To show how serial addition of multiple-bit numbers is performed.

To perform this step you must construct the circuit shown in Fig. 84-5. However, before you can assemble this circuit, you must partially disassemble the experimental circuit that you used in the previous step. First remove the switch register input leads that are connected to the terminals on the experimental chassis PC board. This will allow all of the switch register input leads to hang free. Remove the LØ and L1 input leads from terminals IC1-5 and IC2-8. All of the input leads to the lamp driver board should hang free at this time. The wiring associated with positions IC1 and IC2 should not be changed. The IC1 and IC2 positions will not be used again until you start the next experiment, at which time you will make the necessary changes. Remove the 7480 IC from position IC1 and install it in socket IC46 on the A1 circuit board.

Wire the circuit shown in Fig. 84-5. Study this figure carefully and wire the circuit exactly as shown. Notice that you will need both the A1 and A2 circuit boards. You will also need the switch register, both of your lamp driver boards, a red push button switch, an alligator clip, and some yellow and black hookup wire. You will have to connect red and black leads for the +5 volt and ground leads for both the circuit boards. The +5 volts is available at terminals 53, 43, and 33 while terminals 50, 40, and 30 are connected to ground.

Notice that some of the terminals on the A1 board must be connected together and then grounded to the G2 (ground) terminal on that board. The same is also true on the A2 board. When you make these connections, use short pieces of black hookup wire. When you connect the red push button switch, use a 12-inch piece of yellow hookup wire to connect to terminal A of the switch. Connect terminal C of the switch with a 12-inch piece of black hookup wire. The alligator clip may be connected with a 12-inch piece of black hookup wire. Also, you should use a short piece of the black hookup wire to connect the MDA and BEØ terminals on the A1 circuit board. Black hookup wire is also used to connect the three CLK terminals together and to connect these terminals to terminal 46 on the experimental chassis PC board. Terminal 46 is the output of the buffered latch circuit. You will use the buffered latch and its associated push button switch to provide clock pulses to the experimental circuit. Be sure to connect the two CLK terminals together between terminals ASD and RØA on the A1 board. Check your circuit again to be sure that it is properly connected. Check each wire in the circuit against Fig. 84-5.

You may wish to refer to the two logic diagrams (insert sheet) of the A1 and A2 circuit boards to verify your wiring and to see exactly what the circuit consists of. Refer to these diagrams as you run the experiment and read the related discussion.

Now, turn on the experimental chassis. Set all of the switches in the switch register to the down or binary 0 position. Next, depress the red parallel load push button (connects to terminal MLA) momentarily. This operation will allow you to parallel load binary 0's into the accumulator on the A2 board. All of the lamps that are connected to terminals AØ through A7 on the A2 board should be off at this time, indicating that the accumulator has been cleared.

Insert the first of two numbers to be added (the augend) into the switch register. Set the switches as shown for the augend in Fig. 84-6. Notice that the augend has a decimal value of 25. Push and release the parallel load push button, which will load the augend into the accumulator. The augend should now appear on the lamp driver board that is connected to the A2 board. Check to be sure that the number indicated by the lamps is identical to the number you set into your switch register. Disregard the number displayed by the lamps connected to the A1 board.

Next momemtarily depress and release the black clock push button eight times. This operation will shift the contents of

	BINARY CODE MSB LSB	DEC. VALUE
AUGEND	0 0 0 1 1 0 0 1	25
ADDEND	0 0 0 0 1 1 0 0	12
SUM		

Fig. 84-6. Chart for Step 3.

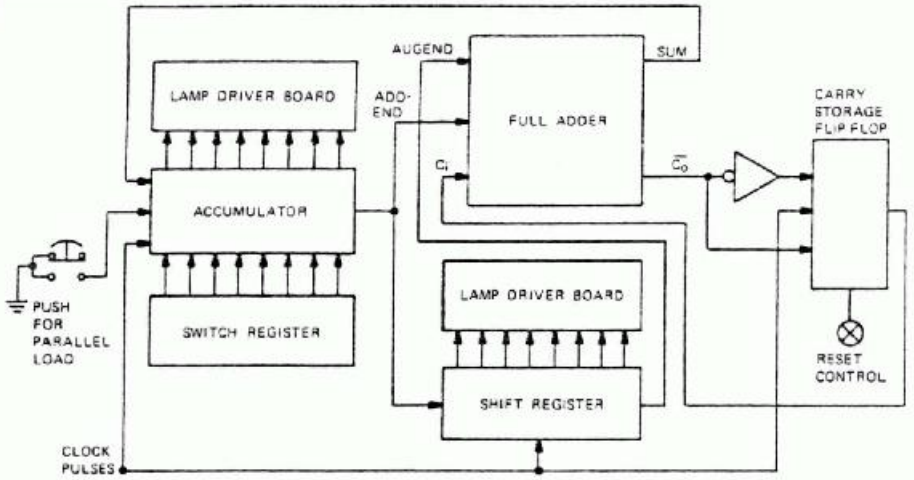


Fig. 84-7. Block diagram of circuit used in Step 3.

the accumulator (the augend) into the B shift register on the A1 board. At the same time the random number already in the B register will be added to the number you are shifting out of the accumulator. This sum will be recirculated into the accumulator, so disregard the display of the lamps connected to the A2 board at this time. The augend should now appear on the lamp driver board that is connected to the A1 board. Check to be sure that the number indicated by the lamps on this board is correct.

Set up the second number (the addend) with the switch register. Set the switches as shown for the addend in Fig. 84-6. Notice that the addend has a decimal value of 12. Push and release the parallel load push button, loading the addend into the accumulator. The addend should now appear on the lamp driver board that is connected to the A2 board. Check to be sure that the number indicated is correct.

Momentarily touch the carry storage flip-flop reset control alligator clip (on terminal ASD) to terminal G2 of the A1 circuit board. By momentarily grounding this alligator clip, you will clear the carry storage flip-flop.

Now apply eight clock pulses to the circuit by depressing and releasing the black push button eight times. This operation will cause both the augend and the addend to be applied to the full adder and the resulting sum will be returned to the accumulator. Observe the binary number that is displayed on the lamp driver board connected to the A2 board. This number represents the sum of the augend and addend. Record this binary number in the space provided in Fig. 84-6. Be sure to keep track of the MSB and LSB positions in this number. Determine the decimal value of the number, recording this value in the appropriate column of Fig. 84-6. Turn off the experimental chassis.

Discussion of Step 3: A block diagram of the circuit that you constructed in this step is shown in Fig. 84-7. Except for the buffered latch circuit which supplies the clock pulses, the entire experimental circuit is contained on the A1 and A2 circuit boards. Notice that the switch register is connected to the accumulator so that the input data may be loaded in parallel form. The lamp driver board connected to the accumulator monitors each binary bit that is stored. The B shift register is also used in the circuit and each bit that is stored in this register is monitored by the other lamp driver board. The serial outputs from the accumulator and the B register are connected to the inputs of a full adder. The sum output of the full adder is connected to the serial input of the accumulator. A carry storage flip-flop is connected between the $\overline{C_o}$ (inverted carry output) and the C_i (carry input) of the full adder. When a carry occurs, this flip-flop will store the carry temporarily and add it to the next pair of bits to be added.

Notice in Fig. 84-7 that the sum is returned to the accumulator input. In other words, the output (sum) of the full adder is shifted back into the accumulator in serial form one bit at a time. As the addend is shifted out of the accumulator, the sum is shifted in. After the last clock pulse was applied, the accumulator should have contained the sum of the augend and addend, as indicated by the lamps on the lamp driver board that was connected to the accumulator outputs.

Check to be sure that you obtained the correct sum. Since the augend had a decimal value of 25 and the addend had a decimal value of 12, the sum must be equal to 37.

Instructions for Statement 84: For this statement you will add two binary numbers using the circuit that you wired in the last step. You will follow the same procedure that you used before.

First turn on the experimental chassis, then add the augend and addend that are shown in Fig. 84-8, following the same procedure that was outlined in your last step. Record the resulting sum in the space provided in Fig. 84-8. Determine the decimal value of the augend, the addend, and the sum and record them in Fig. 84-8. Now turn off the experimental chassis and answer the statement question.

After you answer the statement question, proceed to your next experiment. Do not change any of the wiring associated with your experimental circuit until you are instructed to do so in the next experiment.

	BINARY CODE MSB LSB	DEC. VALUE
AUGEND	0 1 0 0 1 0 1 1	
ADDEND	0 0 1 1 1 1 0 0	
SUM		

Fig. 84-8. Chart for Statement 84.

Statement No. 84: When I added the two numbers specified, I found that the augend had a decimal value of

(1) 65
(2) 75
(3) 58

and the addend had a decimal value of

(1) 78.
(2) 70.
(3) 60.

The resulting sum had a binary value of

(1) 10001000
(2) 10000111
(3) 10000000
(4) 1000111

and a decimal value of

(1) 143.
(2) 136.
(3) 135.
(4) 128.

EXPERIMENT 85

Purpose: To demonstrate the principles of binary subtraction and examine several methods of accomplishing this function.

Introductory Discussion: In the previous experiment you saw how binary addition was performed. You will now construct several experimental circuits which may be used to perform binary subtraction. First, you will build a half subtractor circuit using NAND gates and demonstrate its operation. Finally, you will use the A1 and A2 boards to demonstrate the principles of serial subtraction.

You may wish to refer to your regular NRI lessons as you perform these experiments, particularly lesson K412. Take your time to perform each step according to the instructions given. This subtraction experiment and the previous experiment on addition are extremely important ones. You must understand the techniques used to add and subtract binary numbers before you can consider the more complex operations of multiplication and division.

Experimental Procedure: Before you begin this experiment, you must partially disassemble the circuit that is now wired on your experimental chassis. First remove the wire connected between IC1-4 and IC2-10. Remove the wire between IC2-12 and 13. Leave the +5 volts and ground connections on both IC sockets, and the lead connected between IC2-9 and 10. Remove the 7420 IC from the experimental chassis and install a 7400 IC at each position. Now wire the circuit shown in Fig. 85-1. You will also find it necessary to disconnect the two switch register leads and two of the lamp driver leads from the A2 circuit board and reconnect them as shown in Fig. 85-1. Do not change any of the remaining wiring associated with the A1 and A2 boards, since they will be used later in this experiment and the necessary wiring changes will be made at that time.

Step 1: To demonstrate the operation of a half subtractor.

Turn on the experimental chassis. Set switches SØ and S1 to the down or binary 0 position. This will apply a binary 0 to the A and C inputs of the experimental circuit. Observe the indication of lamps IØ and I1. Lamp I1 is being used to monitor the D output of the experimental circuit and lamp IØ is being used to monitor the B output. Keeping this fact in mind, record the binary output states of the D and B outputs in the spaces provided in Fig. 85-2. Use the switch register to apply the remaining input combinations as shown in Fig. 85-2, and record the resulting D and B output states as indicated by lamps I1 and IØ. After you complete the table, turn off the experimental chassis.

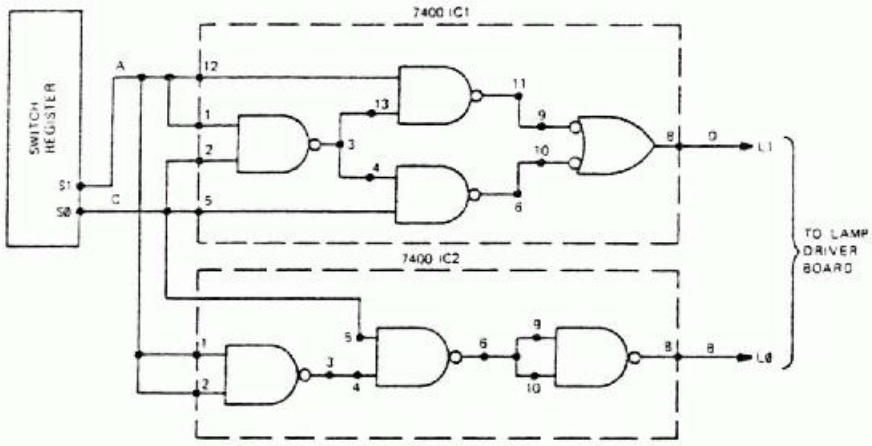


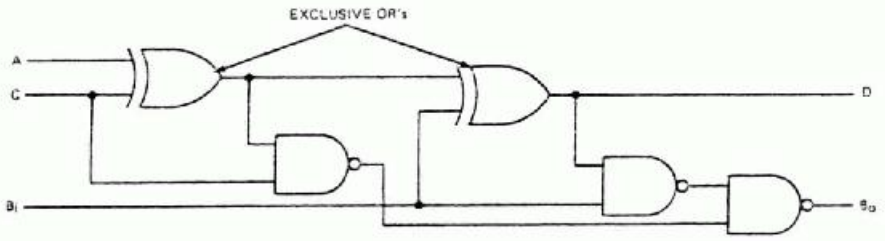
Fig. 85-1. Circuit used for Step 1.

INPUTS		OUTPUTS	
A	C	D	B
0	0		
0	1		
1	0		
1	1		

Fig. 85-2. Chart for Step 1.

Discussion of Step 1: The circuit that you wired in this step is referred to as a half subtractor. This circuit is capable of subtracting a binary bit, C, from another binary bit, A, and producing a difference, D, as well as a borrow output, B, when applicable. Basically, the circuit consists of an exclusive OR gate with an additional inverter and an AND gate. The exclusive OR gate is formed by the four NAND gates in IC1. One of the NAND gates in IC2 is used as an inverter while two of the NAND gates in this IC are connected as an AND gate. The remaining NAND gate in IC2 is not used at all.

The output of the exclusive OR gate may be expressed in Boolean terms as $D = A\overline{C} + \overline{A}C$. Since an exclusive OR gate is also used in a half adder circuit to produce a sum (S) output, the sum (S) output of the half adder circuit will be identical to the D output in the half subtractor circuit. The letter D, however, represents the difference between the two binary input bits rather than the sum. In other words, for various input combinations the D or difference output of the half subtractor used in this step will be identical to the S or sum output of the half adder circuit that you wired in the previous experiment.



A	C	Bi	D	Bo
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Fig. 85-3. Full Subtractor.

The B or borrow output of the half subtractor is taken from an AND gate which is formed by two of the NAND gates in IC2. The C input is applied directly to this AND gate; however, the A input is inverted by another NAND gate before being applied to the AND gate. The B or borrow output may be expressed in Boolean terms as $B = \overline{A}C$.

In this step you applied various input combinations to the inputs of the experimental half subtractor circuit using the switch register and observed the outputs on the lamp driver board.

A full subtractor circuit and its associated truth table is shown in Fig. 85-3. Notice that this circuit uses a borrow input (B_i) along with the minuend (A) and subtrahend (C) to generate the difference (D) and borrow (B_o) outputs. Unlike the half subtractor you just constructed, a number of these circuits can be connected together to form a multiple-bit subtractor.

You will not construct and demonstrate the operation of this circuit since it is not often used in a computer. Subtraction is usually performed by complementing the subtrahend and adding.

Step 2: To demonstrate subtraction using 2's complement addition.

You will be using the A1 and A2 circuit boards in this experiment. Disconnect the SØ and S1 switch leads from the experimental chassis and reconnect them to the AIØ and AI1 terminals on the A2 circuit board. Next, remove the LØ and L1 lamp leads from the experimental chassis and reconnect them to the AØ and A1 terminals on the A2 circuit board. It will not be necessary to disconnect the remaining wires from the IC1 and IC2 terminals at this time. Wire your A1 and A2 circuit boards as shown in Fig. 85-4.

Only a few wiring changes will be required since the A1 and A2 boards are

GROUP OF 8 CLK PULSES	A7	A6	A5	A4	A3	A2	A1	AØ	B7	B6	B5	B4	B3	B2	B1	BØ
START	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
2	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	1
3	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0
4	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0
5	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0
6	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0
7	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0
8	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0

Fig. 86-3. Results for Step 1. Contents of the A and B registers after each group of eight clock pulses.

GROUP OF 8 CLK PULSES	A7	A6	A5	A4	A3	A2	A1	AØ	B7	B6	B5	B4	B3	B2	B1	BØ
START	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
1																
2																
3																
4																
5																
6																
7																
8																

Fig. 86-2. Table for Step 1. Contents of the A and B registers after each group of eight clock pulses.

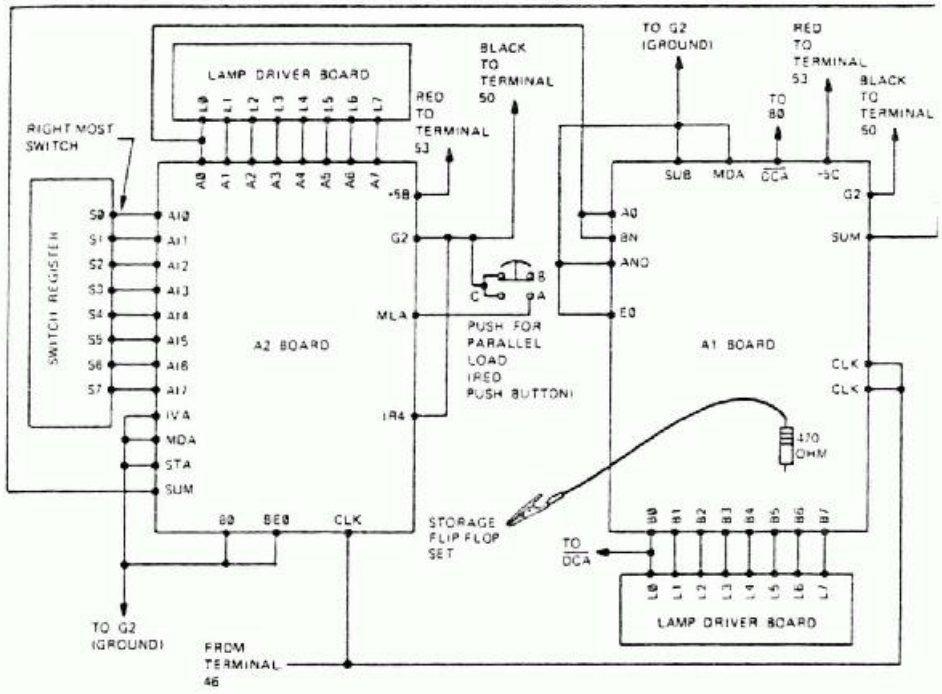


Fig. 85-4. Circuit for Step 2.

still connected as they were for Experiment 84. Changes will be made on the A1 circuit board only. Disconnect the wire between terminals BEØ and MDA at terminal BEØ. Connect the end of this lead to terminal SUB. Disconnect the alligator clip lead from terminal ASD. Tack-solder the end of this lead to the end of the 470-ohm resistor nearest IC41. Use another piece of hookup wire to connect terminal BØ to terminal $\overline{\text{DCA}}$. Check your connections against Fig. 85-4 to be sure they are correct.

Turn on the experimental chassis. Enter the subtrahend into the switch register as shown in Fig. 85-5. Push the parallel load push button momentarily to transfer the contents of the switch register to the accumulator. Check the display on the accumulator lamp driver

	BINARY CODE MSB LSB	DEC. VALUE
MINUEND	0 0 1 1 0 1 1 0	54
SUBTRAHEND	0 0 0 1 0 1 1 0	-22
DIFFERENCE		32

Fig. 85-5. Chart for Step 2.

board to be sure that the number has been entered correctly. Now press the clock push button eight times. Notice that the number in the accumulator has been transferred to the B register. Check the B register display to be sure that the number is still correct.

Enter the minuend into the switch register. Push the parallel load push button momentarily. Check the accumulator display to be sure the minuend has been correctly entered. Notice that the minuend is stored in the accumulator and the subtrahend is stored in the shift register.

Momentarily touch the alligator clip lead (attached to one end of the 470-ohm resistor) to +5C. This will set the carry flip-flop. Again press the clock push button eight times. Examine the number stored in the accumulator. Write this number in the space provided in Fig. 85-5. After you have entered the result, turn off the experimental chassis.

Discussion of Step 2: The circuit you wired for this experiment performs subtraction by adding the 2's complement of the subtrahend to the minuend. Let's see how this is accomplished.

A block diagram of the circuit is shown in Fig. 85-6. Refer to this diagram as you study the circuit operation. In performing this experiment the subtrahend is entered into the switch register, then transferred to the accumulator. By appling eight clock pulses to the accumulator and shift register while holding the shift right push button down, the number is transferred to the B register. Next, the minuend is entered into the switch register and loaded into the accumulator. The minuend and subtrahend are stored in the accumulator and B register respectively.

The next application of eight clock pulses will cause these numbers to be applied serially to the full adder circuit. The difference will be stored in the accumulator.

Notice how the 2's complement of the subtrahend is generated. The output of the B register is applied through an inverter to the input of the full adder.

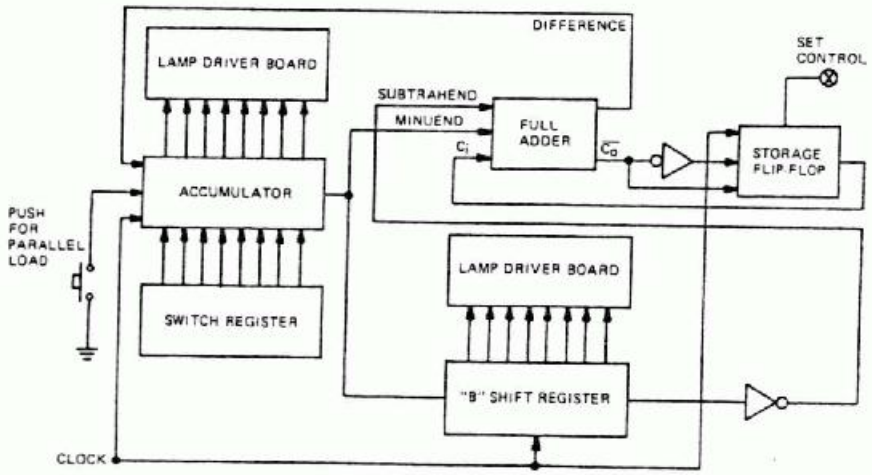


Fig. 85-6. Block diagram of circuit used in Step 2.

	BINARY CODE MSB LSB	DEC. VALUE
MINUEND	0 1 0 0 0 1 0 0	
SUBTRAHEND	1 0 0 1 0 0 1 1	
DIFFERENCE		

Fig. 85-7. Chart for Statement 85.

This will automatically form the 1's complement of the number. By initially setting the carry storage flip-flop, a binary one is applied to the carry input of the full adder. This must be done at the beginning of the subtract operation. Adding a binary one to the LSB of 1's complement of a number will produce 2's complement of the number. Thus, the final shifting operation is actually causing the minuend and 2's complement of the subtrahend to be applied to the full adder, resulting in the correct difference being stored in the accumulator.

Instructions for Statement 85: For this statement you will be subtracting a large number from a smaller number. Be sure to carefully consider the result before answering.

No wiring changes will be required. Enter the minuend and subtrahend as shown in Fig. 85-7 then perform the subtraction using the procedures just outlined. Record the result in the space provided. Turn the experimental chassis off and answer the statement questions.

Statement No. 85: The binary number stored in the accumulator after the subtraction was:

(1) 01001111.
(2) 10110001.
(3) 01001110.
(4) 10110000.

The decimal equivalent of the difference is:

(1) 177.
(2) –79.
(3) 215.
(4) –177.

EXPERIMENT 86

Purpose: To build and demonstrate a binary multiplier.

Introductory Discussion: Binary multiplication can be accomplished in two basic ways. By using the standard adder circuit, multiplication can be accomplished by repeated additions. The multiplicand is simply added to itself a number of times indicated by the multiplier. The result is the correct product. Another way of producing multiplication of binary numbers is to use an add and shift technique. In this scheme, we essentially duplicate the pencil and paper method of multiplying binary numbers that you studied in an earlier lesson. Partial products of each of the bits in the multiplier are obtained and shifted with respect to one another, then added. The result is the correct product.

While adder and subtractor circuits are relatively simple, the circuits for binary multiplication are of necessity much more complex. You will find this to be true in this experiment as you construct a binary multiplier using the add and shift technique. With the circuitry that you will assemble, you will be able to multiply two 8-bit numbers and produce a 16-bit product. You will use much of the circuitry that you have already used before. Since there will be quite a bit of wiring in this experiment, you should take care in wiring the circuit. It takes only one wiring mistake to cause the circuit to malfunction. Be extremely careful in your wiring; follow the directions explicitly. To avoid confusion and to minimize errors, keep your circuitry neat. Place the printed circuit boards as close to one another as possible and keep the wiring neat and straight. This will not only aid in the wiring process but will also permit you to easily locate all of the pertinent points in the circuit.

Experimental Procedure: The circuit that you will use in this experiment is shown in Fig. 86-1. You will use the M2, A1, and A2 boards. You will also need the experimental chassis and the two lamp driver boards.

Make the following changes on the A1 board: Remove the ground wire from terminal MDA. Move the lead from terminal $\overline{DCA}$ to terminal MDA, and remove the clip lead from the 470-ohm resistor.

Make these changes on the A2 board: Connect a length of hookup wire from terminal ASD to terminal T7. Connect a short length of hookup wire from terminal $\overline{HLT}$ to terminal G2.

Take the M2 board and connect the red and black power and ground leads to terminals 53 and 50 respectively of the experimental chassis.

Solder a length of hookup wire from terminal CLK of M2 to terminal $\overline{PAUS}$ on the A2 board.

Solder a length of hookup wire from terminal EØ of M2 to terminal ADD on the A1 board.

Solder a length of hookup wire from terminal AØN of M2 to terminal AØ on the A2 board.

Solder a length of hookup wire from terminal T7 of A2 to terminal ASD on the A1 board.

Take the other red push button switch and solder a length of hookup wire equal in length to the two leads already in place to terminal B of the switch. Solder the A, B, and C leads of this switch to terminals G2, +5A, and SMC respectively on the M2 board.

Remove all the leads connected to IC1 on the experimental chassis with the exception of ground and +5 volts at terminals IC1-7 and IC1-14 respectively. Using short lengths of hookup wire, connect the following terminals:

IC1-1 to IC1-6
IC1-3 to IC1-4.

Take the black push button you received with this kit and connect 10" lengths of hookup wire to terminals A, B, and C of the switch. Solder the A lead to terminal IC1-2. Solder the B lead to terminal IC1-5 and the C lead to terminal IC1-7.

Connect a length of hookup wire from terminal IC1-6 to terminal ASD on the A1 board. Connect a length of hookup wire from terminal IC1-3 to terminal RØA on the A1 board. Finally, make sure a 7400 integrated circuit is installed in the IC1 socket.

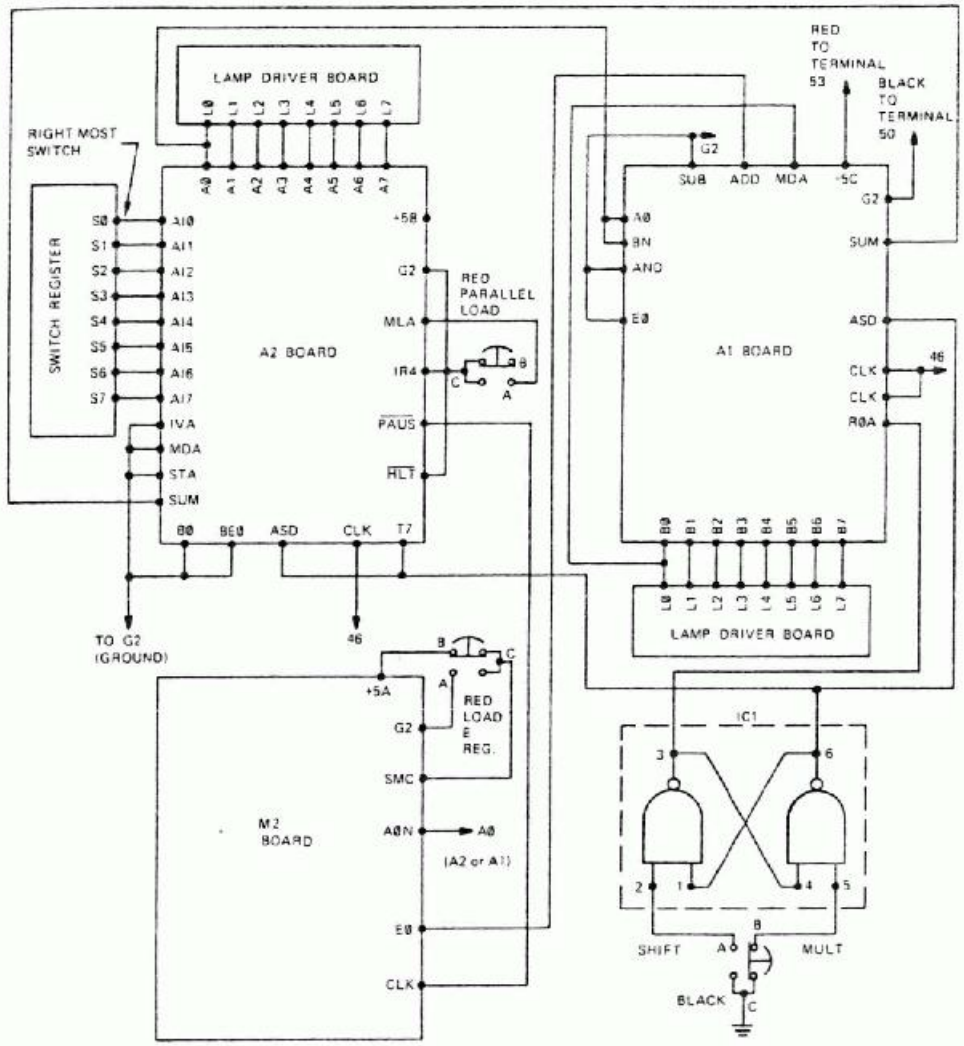


Fig. 86-1. Binary Multiplier.

Carefully check over all the wiring before you begin Step 1.

Step 1: To demonstrate a serial binary multiplier.

The circuit that you have just wired is a serial binary multiplier that uses the add-shift technique. The circuit consists of three 8-bit shift registers and the serial adder on the A1 circuit board. The three registers involved are the accumulator register on the A2 board, the B register on the A1 board and the E register on the M2 board. The E register on the M2 board will be loaded with the multi-

plicand. The multiplier will be initially loaded into the B register on the A1 board. At the beginning of the computations, the accumulator register will be clear. Once the numbers have been loaded, you will apply clock pulses to the circuit and generate the product.

The product, which can be twice as long as either the multiplier or the multiplicand, will be stored in the B register and A register. The least signficant part of the product will be stored in the B register, while the most significant part will be stored in the accumulator. The multiplier, formerly in the B register, is replaced with the least significant part of the product.

The first step is to load the registers with the appropriate numbers. To load the various registers, you will use the switch register and the accumulator as the basic number source. You will parallel load the accumulator with the switch register and transfer the number in the accumulator to either the E or B register.

Turn on the experimental chassis and set your switch register with the binary number 7 or 0000 0111. Depress the red push button connected to the MLA terminal of the A2 circuit board. This will transfer the contents of the switch register into the accumulator. The lamp driver connected to the accumulator should now display the number 7. To shift this number into the E register, first press and hold in the red push button connected to the M2 board. Now use the black push button connected to the discrete component latch on the experimental chassis to generate the clock pulses. Depress the black push button eight times, then release the red push button. This will cause the number stored in the accumulator to be transferred a bit at a time into the E register. This is the multiplicand to be used in the problem.

Since there is no light driver circuitry connected to the E register, you have no convenient means of monitoring the number contained there. However, you can use your tvom to check the dc voltage levels at terminals E0 through E7 on the M2 board.

You will next load the multiplier into the B register on the A1 board. Set the switch register to the multiplier value, which in this case will be the decimal number 6. Set the switches from left to right to 0000 0110. Momentarily depress the red push button connected to the MLA terminal on the A2 circuit board. This will load the accumulator register with the switch register contents. Observe the lamp driver to see that the number 6 has been loaded. Next, depress and hold the black push button connected to IC1 on the experimental chassis and again operate the clock push button eight times. This will shift the number in the accumulator into the B register on the A1 circuit board. Release the IC1 (shift) push button. At this time the lamp driver should read the 8-bit number representing the decimal number 6.

You are now ready to perform the experiment in which you multiply the numbers 7 and 6. Before you begin the computation, check to be sure that all of the lamps in the accumulator register are out, indicating that the register is clear. If any of the lamps are on, it will be necessary to set 00000000 into the switch register and press the parallel load push button once more.

To perform the computation you will need to depress the clock push button 72 times. Since each number is a maximum of eight bits in length and we need to examine each bit in the multiplier one at a time, there will be a total of eight clock pulses occurring for each of the eight multiplier bits, plus an additional clock pulse for the shift. This makes a total of 72 clock pulses to complete the multiplication. This will ensure that all bits of both the multiplicand and the multiplier will be used in the multiplication process. Of course, here you are using only three bits for each number, but in order to complete the multiplication and have the product stored in the proper register locations you will need to perform the entire operation.

Depress the black clock push button eight times. After you have applied the eight clock pulses, press and hold the black shift push button connected to IC1 and apply one more clock pulse. Release the shift push button. Stop and observe the contents of both the B and accumulator registers. Record the results in Fig. 86-2 (centerfold).

Again depress the clock push button eight times plus one more with the shift push button held in. After this second group of nine clock pulses, again examine the contents of the A and B registers and record the results in Fig. 86-2. Repeat these 8-bit groups of clock pulses and one shift clock pulse for a total of eight times. After you have applied the 72nd clock pulse, the multiplication is complete and you should record the contents of the A and B registers in the final position noted. This is your product.

Discussion of Step 1: In this step you multiplied the two numbers 7 and 6 and as a result should have obtained the product of 42. In binary form this is 101010. This number should be stored in the B register after the multiplication.

Fig. 86-3 (centerfold) shows the results that you should have obtained for this step. At the starting position the accumulator is reset, while the B registe contains the multiplier, 6. The E register of course, contains the multiplicand, 7, which we are not monitoring with lamps at this time. The contents of both the A and B registers after the first eight clock pulses and one shift pulse are illustrated in row 1 of Fig. 86-3. The contents of each of these registers are shown for each group of eight clock pulses and one shift pulse. After the final group of eight clock pulses and one shift pulse, the product as shown will be contained in the B register. Check the results you recorded in Fig. 86-2 with the correct results in Fig. 86-3. If necessary, go back and reload the registers and repeat this step to be sure that you obtain the correct result.

Fig. 86-4 is a simplified block diagram of the serial multiplier circuit that you constructed. By referring back to Fig. 86-1, you can compare the two circuits and see that they are alike. However, by using the simplified diagram in Fig. 86-4, we can more easily explain the operation of the circuit.

The multiplicand is initially stored in the E register and the multiplier is initially stored in the B register. Before the multiplication begins, the accumulator is cleared. When the clock pulses are applied, the adder will produce the sum of the two numbers. One of the numbers is stored in the accumulator while the other is obtained from either the E register or from a source of binary zeros. AND gate 1 looks at the least significant bit position of the B register. If this bit is a binary 0, then the AND gate will be inhibited. As clock pulses are applied, the multiplicand stored in the E register will be shifted out one bit at a time and applied to gate 1. It will also be recirculated and restored in the E register. At the same time, the contents of the accu-

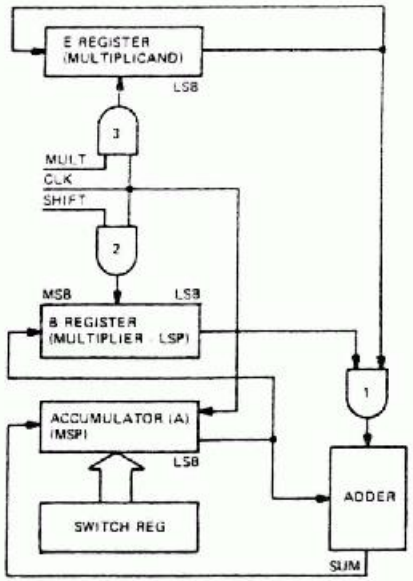


Fig. 86-4. Simplified block diagram of the Serial Multiplier.

mulator will be shifted into the adder. Since AND gate 1 is inhibited, naturally the contents of the E register will not appear at the upper input to the adder. As a result, binary 0's will be added to the contents of the accumulator.

AND gates 2 and 3 direct the clock pulses to the B register or the E register, depending on whether we are multiplying or shifting. During multiplication, gate 3 is enabled and gate 2 is disabled, preventing the B register from being shifted. During the shift operation, gate 3 is disabled and gate 2 is enabled, allowing the B register to be shifted and not the E register. The accumulator receives clock pulses at all times.

In the first step of our experiment the accumulator is initially zero, so zeros are loaded back into the accumulator from the adder. We see this condition existing in row A of Fig. 86-3. As soon as the first eight clock pulses are completed, the application of the shift pulse causes the contents of the accumulator and the B registers to be shifted one position to the right. The least significant bit in the accumulator is shifted into the most significant bit position of the B register. At the same time, the least significant bit of the B register is shifted to the right and lost.

A binary 1 now appears at the lower input to AND gate 1 in Fig. 86-4, meaning that the gate will be enabled. Therefore, for the next eight clock pulses that occur, the contents of the accumulator will be added to the contents of the E register with the sum stored back in the accumulator. Since the number 7 is stored in the E register, it will be added to the 0 contents of the accumulator. The accumulator will contain the number 7 at the end of the eight clock pulses. However, the shift pulse shifts the contents of the accumulator and the B registers one bit to the right. This results in the number shown in row 2 of Fig. 86-3.

Again you can see the least significant bit (BØ) of the B register is a binary 1. This permits the contents of the E register to be added to the accumulator. The result appears back in the accumulator, with the resulting sum again shifted to the right one bit position by the shift pulse. The resulting number appears in row 3 of Fig. 86-3. All of the remaining bits of the multiplier are binary 0's from here on. For that reason, AND gate 1 will be inhibited and no further sums will be created in the accumulator. Instead, after each eight clock pulses the contents of the accumulator and B registers will simply be shifted to the right. After a total of 72 clock pulses has occurred, the product appears in the correct position in the B register and the accumulator. The least significant part will be in the B register, while the most significant part will be in the accumulator. For example, if the product is 520 (1000001000) then 00001000 will be in the B register while 00000010 will be in the accumulator.

Instructions for Statement 86: You do not need to make any further changes in your experimental circuit to perform this statement. At this time the experimental chassis should still be on and the product properly displayed on the B register lamps.

Using the procedure used in the experiment, multiply the number in the B register by the number in the E register. Simply generate eight clock pulses, press the shift push button and generate another clock pulse. Repeat this process seven more times and record the contents of both the accumulator and B registers in the margin of the text. Convert the resulting binary number into its decimal equivalent. Now you can answer the statement question.

Now remove all the circuit wiring made for this experiment to the A1, A2 and M2 circuit boards. Leave the power and ground leads connected to the three circuit boards and also leave the power and ground leads and the yellow leads connected to the two lamp driver boards and the switch register. Remove all wiring from the IC1 and IC2 sockets on the experimental chassis, except the power and ground connections to IC1-14, IC2-14, IC1-7 and IC2-7.

Statement No. 86: When I performed the multiplication indicated by this statement, I found the product to be:

(1) 42
(2) 49
(3) 252
(4) 294

which is the number that results when multiplied the numbers:

(1) 6 and 7.
(2) 7 and 7.
(3) 6 and 42.
(4) 7 and 42.

EXPERIMENT 87

Purpose: To demonstrate the operatio of a binary rate multiplier.

Introductory Discussion: A binary rat multiplier is a circuit that multiplies tw digital input numbers and generates a output equal to their product. One of th digital input numbers is a unitary puls train while the other is a parallel binar input word. The output is a unitary puls train. Fig. 87-1 is a simplified bloc diagram of a binary rate multiplier.

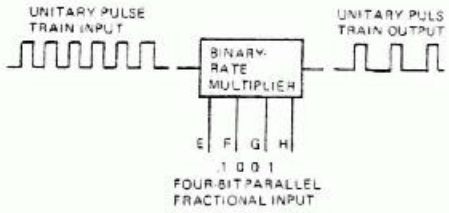


Fig. 87-1. A Binary Rate Multiplier.

A unitary pulse train is simply a series of off-on pulses that represents some decimal quantity. For example, to represent the decimal number 15, we would generate a sequence of 15 pulses. To represent the decimal number 735, a unitary pulse train would be a sequence of 735 individual pulses.

The other input to the rate multiplier is a binary number whose magnitude is always between zero and one. That is, it always has a fractional value. For this reason the number of output pulses in the output train will always be less than the number of input pulses. The number of pulses in the output will be equal to the number of pulses in the input multiplied by the parallel binary fractional input.

For example, if we apply 80 input pulses to the binary rate multiplier and the parallel binary input word is $.1001_2$ ($.5625_{10}$), the output will contain 45 pulses (80 X .5625).

Keep in mind two important facts about the rate multiplier. First, multiplying a number by a fraction is equivalent to dividing that number by a whole number. Therefore, the binary rate multiplier can also be considered as a divider. Secondly, since the output pulse train has fewer pulses than the input, the binary rate multiplier could also be considered a frequency divider. The parallel binary input word permits the binary rate multiplier to perform as a programmable frequency divider in many applications.

In this experiment you are going to demonstrate a basic 4-bit binary rate multiplier circuit. You will demonstrate its basic characteristics and several modes of operation. By demonstrating the basic operation of the circuit, you will become familiar with this important computational unit.

Experimental Procedure: Fig. 87-2 shows the circuit for this experiment. You will need the M1 and C2 circuit boards and a lamp driver circuit board as well as the switch register and the experimental chassis. Connect the +5 volt and ground leads of the M1, C2 and lamp driver boards to the appropriate +5 volt and ground terminals on the experimental chassis. Also connect the ground and +5 volt connections of the switch register.

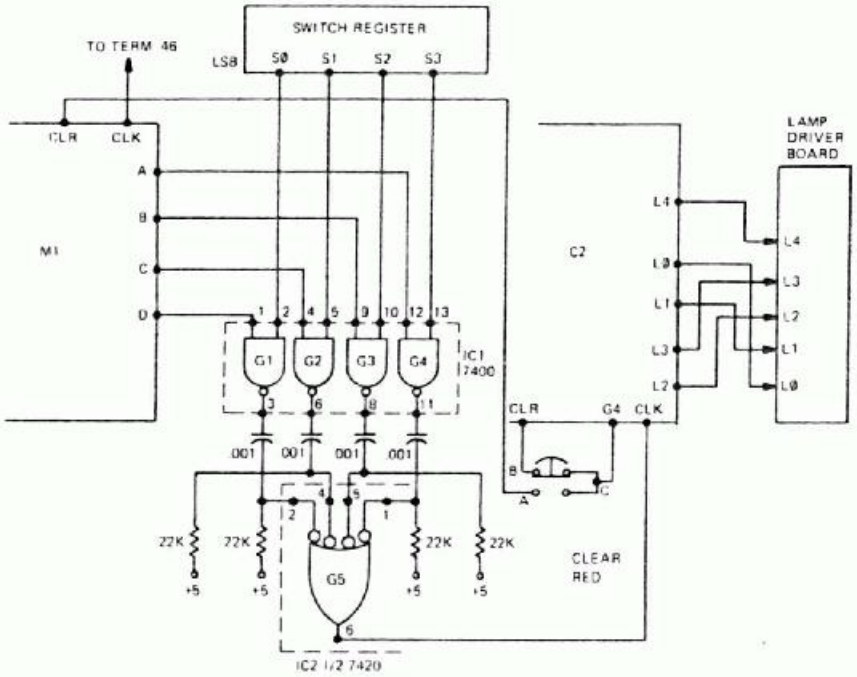


Fig. 87-2. Binary Rate Multiplier used in Step 1.

Next, wire the four right-most or least significant bit outputs of the switch register to terminals IC1-2, IC1-5, IC1-10 and IC1-13. The right-most or LSB of the switch register is connected to terminal IC1-2 as you can see in Fig. 87-2.

Connect four lengths of hookup wire from the A, B, C and D outputs on the M1 circuit board to terminals IC1-12, IC1-10, IC1-4 and IC1-1 as shown in Fig. 87-2. The CLK input to the binary counter on the M1 circuit board will, as usual, be wired to terminal 46 on the experimental chassis. This will permit you to set the counter from the black push button buffered by the discrete component latch.

Solder .001 mfd capacitors between the following terminals:

IC1-3 to IC2-2
IC1-6 to IC2-4
IC1-8 to IC2-5
IC1-11 to IC2-1

Solder 22k-ohm resistors between the following terminals:

IC2-1 to IC1-14
IC2-2 to IC1-14
IC2-4 to IC2-14
IC2-5 to IC2-14

Connect a length of hookup wire from terminal IC2-6 to the CLK terminal on the C2 circuit board. Connect the LØ through L4 leads from the lamp driver board to terminals LØ through L4 on the C2 circuit board. You may remove lamps L5, L6, and L7 if you wish, or you ca disable them by grounding the L5, L6 and L7 leads.

Connect the C lead of one of the re push buttons to terminal G4 on the C circuit board. Connect the B lead t terminal CLR on C2 and connect the A lead to terminal CLR on the M1 circui board. Finally, connect a piece of hooku wire from terminal 38 to terminal 40 o the experimental chassis. Carefully check your wiring against Fig. 87-2, then in stall a 7400 IC in the IC1 position and a 7420 IC in the IC2 position. Remove the 7400 IC from socket IC25 on the C2 circuit board.

The binary rate multiplier that you have just constructed consists of circuitry on the M1 circuit board and that wired on the experimental chassis. In other words, the binary rate multiplier consists of a 4-bit binary counter connected to a group of NAND gates whose outputs are differentiated and then ORed together in the 7420 gate. The switch register provides a 4-bit fractional binary number that is used to enable the various gates on the experimental chassis. This will determine which of the counter flip-flop outputs will be passed through the gates, differentiated, then ORed together. The unitary pulse train is applied to the binary rate multiplier by way of the black push button into the CLK input of the binary counter. The rate multiplier output is the output of the NOR gate at terminal IC2-6.

In order to help you conveniently keep track of the number of output pulses that occur without counting them individually, you have wired the program counter or 5-bit binary counter on the C2 circuit board to accumulate these pulses and display the count on the lamp driver PC board. Each time an output pulse

occurs from the binary rate multiplier, the counter on the C2 circuit board will be incremented. The contents of the counter are displayed on the lamp drivers. The red push button is used to clear or reset the counter when necessary.

Step 1: To show how the binary rate multiplier is used to multiply a unitary pulse train by a fractional binary input.

The four right-most switches in your switch register represent a 4-bit fractional binary number when the right-most switch is the LSB position. Set the switches so that they read 1100 from left to right (SØ and S1 down, S2 and S3 up).

Turn on the experimental chassis. Some of the lamps on the lamp driver board may light due to the state of the program counter on the C2 circuit board. Reset this counter by pressing the red push button.

As you depress the clock push button 16 times, the binary rate multiplier circuitry will go into action and generate a unitary pulse train output. The number of output pulses will be counted by the program counter on the C2 circuit board. The total number of pulses occurring at the output will appear as a binary number on the lamp driver, where LØ is the LSB position. The number of pulses is determined by the product of the number of input pulses, in this case 16, and the binary number set in on the switch register. Determine the fractional binary number on the switch register, multiply by 16, the total number of input pulses, and compare it to the binary number displayed on the lamp driver board.

Now set your switch register switches to the number 0110. Reset the program counter on the C2 circuit board and depress the clock push button 40 times. Determine the decimal equivalent of the binary number in the switch register and multiply it by 40 to determine if it equals the number in the program counter.

Discussion of Step 1: This step should clearly show that the output of the binary rate multiplier is a number of pulses equal to the product of the number of input pulses multiplied by the fractional binary number. With your switch register set to the number .0110, the binary input fraction becomes .375.

When you applied 40 input pulses, the rate multiplier produced the number of output pulses equal to 40 times .375 or 15 output pulses. The program counter on the C2 circuit board should have counted these 15 pulses and the lamp driver display should have read 01111.

Setting the switch register to the number .1100 produces an input fraction of .75. Multiplying this by a total of 16 input pulses should have produced an output of 12 (1100) on the counter display lights.

Step 2: To show how the rate multiplier can be used as a programmable frequency divider.

For this step you will drive the rate multiplier from the clock oscillator on the experimental chassis. To do this you will move the wire coming from the CLK terminal on the M1 circuit board from terminal 46 on the experimental chassis and solder it to terminal 39. This will permit the clock to step the rate multiplier. Place all of the four right-most switch register switches in the down position. To enable the clock oscillator, remove the ground wire you connected to terminal 38.

Turn on the experimental chassis. Connect the vertical input lead of your oscilloscope to terminal 39 on the experimental chassis and the ground lead to the chassis. This will let you monitor the clock output. Adjust the vertical gain control to display the clock output, then adjust the horizontal sweep controls to display eight positive pulses. Use the sync and horizontal sweep controls to stabilize the waveform on the screen.

Next, move the vertical lead of your oscilloscope to terminal IC2-6. Switch SØ to the up position with S1, S2, and S3 down. With this switch in the up position, count the number of output pulses that occur. Do not change the horizontal sweep frequency control but use the sync control to stabilize the waveform if necessary. Record the number of pulses in the margin of the text.

Set SØ to zero and S1 to the up position, again monitoring the output. Count the number of pulses that occur but do not change the horizontal sweep frequency setting.

Turn off the S1 and switch S2 to the up position and monitor the number of pulses that occur. Repeat this for S3. Record all your numbers in the margin of the text.

Set the switches in your switch register to the binary number .1100 that you used in the previous step. While monitoring the output of the binary rate multiplier, study the output waveform carefully. Make note of the special periodic nature of the output pulses and study particularly the spacing between adjacent pulses. Adjust the horizontal sweep frequency control to display twelve pulses on the scope screen. Stabilize the waveform with the sync control if necessary. Copy the waveform that you see.

Discussion of Step 2: In this step yo demonstrated how the rate multiplier ca be used as a programmable frequenc divider. When you turned on the mos significant bit switch (SØ) in your 4-bi switch register, you allowed pulses fron the least significant bit flip-flop of th counter on the M1 circuit board to pas through gate G4 into the differentiato and, therefore, appear at the output o the NOR gate. The frequency of thes pulses is one-half that of the clock pulse rate. Therefore, with eight clock pulses initially displayed on the scope screen you should have seen half this number of pulses at the output.

With the next most significant bit switch on and all others off, you are permitting output pulses from the B flip-flop in the binary counter to pass through the output. These pulses are occurring at a frequency one-fourth that of the clock freuqency; therefore, you should have counted only two pulses on the output scope screen. In other words, for eight input pulses applied only two occur, indicating a frequency division of 4.

For the other two switch positions you should have noted frequency decreasing by a factor of 2 in each case. Here you show how these switches can be set to provide division ratios of 2, 4, 8 and 16. Of course, any normal binary counter can provide such binary weighted frequency division output without the need for the additional circuitry that we used in the rate multiplier. However, by setting the switches to other combinations, the frequency division ratio over a wide range can be obtained. For example, when you set the switch register to .1100 you actually multiplied the input frequency by .75 as you demonstrated in Step 1. This is equivalent to dividing the input

frequency by 1/.75 or 1.333. Recall that we said multiplying an input pulse train by a fractional binary number is the same as dividing it by a whole number equal to the reciprocal of that fraction. This means that if the input clock frequency is 2000 Hz, the output of the binary rate multiplier when the input is set to .1100 will be 2000/1.333 = 1500 Hz. As you can see, the frequency division ratio is programmable over a fairly wide range by simply setting the switch register to the appropriate value.

If you looked at the waveform carefully you should see that the pulses appear to be occurring in periodic groups of three. The spacing between each three pulses is equal, but it is less than the spacing between the three pulse group. While we can legitimately call this a periodic waveform, it is an irregular periodic waveform. For many applications this irregular output pulse sequence is of no consequence, particularly if you are going to simply count the pulses of the binary counter and display the result as a parallel binary word. However, in frequency division applications where the output frequency must be perfectly regular this could be a disadvantage. In any case it is an important characteristic of the binary rate multiplier that you should understand.

The input and output of the rate multiplier when used as a programmable frequency divider is indicated by the simple formula $F_2 = n F_1$.

Instructions for Statement 87: Turn off the experimental chassis and remove the wire connecting the CLK input to the M1 circuit board from terminal 39 and reconnect it to terminal 46. Reconnect the ground lead to terminal 38 to disable the clock oscillator. This will again permit you to step the counter with the black push button. Disconnect your oscilloscope.

Next, set your switch register so that the binary number is .1010 (S1 and S3 on, SØ and S4 off). Turn on the experimental chassis and press the red push button momentarily to reset the binary counter on the C2 circuit board.

Depress the clock push button a total of 30 times. This is quite a few pulses so be sure that you count accurately. Once you have applied the 30th pulse, stop and note the binary number displayed on the lamp drivers. Record the decimal value of this binary number in the margin of the text and convert the fractional binary number in the switch register, .1010, into its decimal equivalent. Using this information, consider the computational accuracy of the rate multiplier. With this data you can answer the statement question.

Statement No. 87: In this problem I am multiplying the input number 30 by a fractional value of

(1) .375
(2) .625
(3) .75

which gives an answer of

(1) 22.5.
(2) 10.25.
(3) 18.75.

The number stored in the binary counter at the end of my computation was

(1) 10
(2) 22
(3) 18
(4) 19

which indicates that the rate multiplier

(1) did
(2) did not

produce perfectly accurate computation for this problem.

EXPERIMENT 88

Purpose: To build and demonstrate a BCD adder circuit.

Introductory Discussion: Most of the arithmetic operations that you will encounter in digital equipment will be performed with pure binary numbers. The pure binary number system is more efficient and arithmetic circuitry to handle it is much easier to implement. However, you will find numerous applications where special arithmetic circuitry is used, such as those requiring the processing of BCD numbers. Some digital computers process BCD numbers as well as binary numbers. This is particularly true of business data processing computers.

One of the most popular, widely used applications for BCD arithmetic circuitry is in electronic desk calculators. Millions of these calculators have been sold and are is use today. All of them use BCD calculating circuitry.

The most popular form of BCD arithmetic circuitry is one that processes the decimal numbers a digit at a time. In other words, the individual decimal digits in a number are processed serially from the least significant digit to the most significant. The BCD numbers themselves, however, are processed in parallel. I other words, two BCD numbers to b added together are added in parallel. 4-bit adder is required for this operatio We call such an arrangement bit-parall digit-serial BCD arithmetic. In desk calcu lators, however, the arithmetic process often completely serial. The reason fo this is that shift register memories ar used to store the number in the calcu lator. Each BCD digit is stored as a serie of four bits in the memory in seria fashion. To process these digits as they are shifted out of their serial memories, completely serial arithmetic circuit i required. In this experiment you are goin to demonstrate one type of serial BCI adder.

There are numerous means of accom plishing BCD addition in serial form Some of the techniques are quite com plex, but the one we have chosen here is relatively easy to implement with the circuitry available on your computer cir cuit boards. We will discuss the addition of BCD numbers in both the 8421 and excess 3 word format.

Experimental Procedure: Fig. 88-1 shows the wiring for this experiment. You will be using the experimental chassis, the switch register, a lamp driver printed circuit board, and computer boards M1, M2, and A1.

On the experimental chassis you will wire two 4-bit shift registers using 7495 integrated circuits. These registers will hold two BCD numbers that will be used in the arithmetic process. You will load these registers in parallel from the switch register. The outputs of these registers will be monitored with the lamps on the lamp driver board.

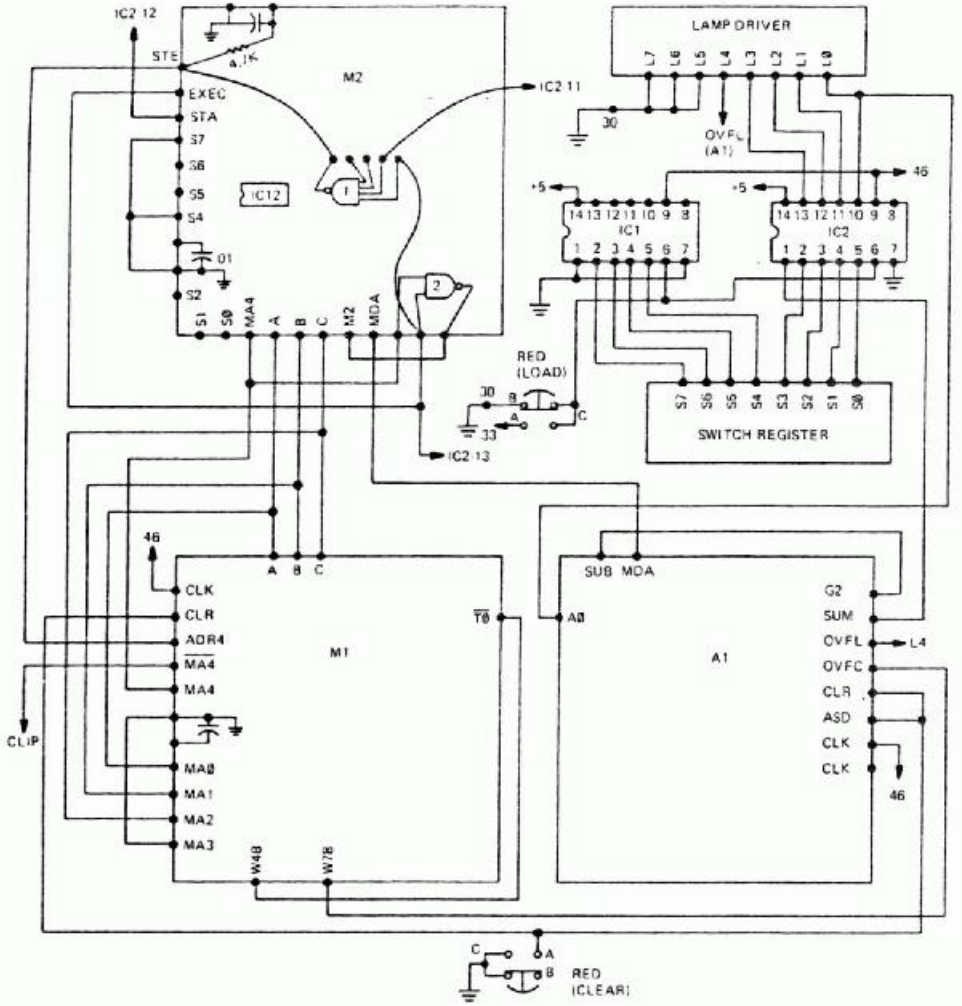


Fig. 88-1. Serial BCD Adder.

The adder circuit is already wired for you on the A1 circuit board. You will feed the outputs of the two shift registers into the adder to produce the sum which will appear back in one of the registers. The adder and its carry flip-flop circuit are all on the A1 board.

The M1 board will be used for timing and control purposes. The binary counter is connected so that it drives the two decoders on this board; the output of one of the decoders is used for timing signals. The MA4 latch flip-flop is used for control purposes.

The binary counter on the M1 circuit board also drives an 8-bit multiplexer on the M2 circuit board. This multiplexer is connected as a parallel-to-serial converter, which is used as a pattern generator. The two free gates available on this board are interconnected to form a logic network to detect when the contents of one of the registers on the experimental chassis is greater than 9.

Begin the assembly of the serial BCD adder by removing the switch register leads from terminals IC1-2, IC1-5, IC1-10 and IC1-13. Leave the power and grounds leads connected to the switch register. Next remove the four .001 mfd capacitors and the four 22k-ohm resistors from the IC1 and IC2 terminals. Disconnect the A, B, C, and D leads from terminals IC1-12, IC1-9, IC1-4, and IC1-1.

Remove the lead from terminal D of the M1 circuit board, but leave the leads connected to terminals A, B, C, CLK and CLR as well as the power and ground leads. Remove all connections from the C2 circuit board. Leave the eight yellow wires connected to the lamp driver board as well as the power and ground leads.

Fig. 88-1 shows the wiring needed for this experiment. The three circuit boards, lamp driver, switch register and experimental chassis are shown in roughly the proper physical relationship to one another. To simplify the wiring, you will first wire up the experimental chassis, lamp driver, and switch register. Then you will make interconnections on the M1 circuit board and the M2 circuit board. You will then interconnect the three circuit boards and the experimental chassis. Now make the following connections:

EXPERIMENTAL CHASSIS, LAMP DRIVER, AND SWITCH REGISTER

FROM	TO	FROM	TO	FROM	TO
				S7	IC1-2
IC1-1	IC1-7	Switch SØ	IC2-5	Lamp LØ	IC2-10
IC2-6	IC1-6	S1	IC2-4	L1	IC2-11
IC2-9	IC1-9	S2	IC2-3	L2	IC2-12
IC2-9	46	S3	IC2-2	L3	IC2-13
red push button (C)	IC2-6	S4	IC1-5	L5	30
red push button (B)	30	S5	IC1-4	L6	30
red push button (A)	33	S6	IC1-3	L7	30

M1 CIRCUIT BOARD

FROM	TO
A	MAØ
B	MA1
C	MA2
MA3	GROUND
TØ	W4B

M2 CIRCUIT BOARD

The two free gates are labeled 1 and 2 in Fig. 88-1. You will make connections to these gates in the following steps. First, however, connect a 4.7k-ohm resistor from terminal STE to the right-hand terminal of the .01 mfd capacitor as shown. This is +5 volts. Now make the following connections:

FROM	TO
Output of Gate 1	STE
Input of Gate 1	Input of Gate 2
EXEC	Same Input of Gate 2
Other Input of Gate 2	MA4
Output of Gate 2	M2
S7	S4
S4	GROUND

Interconnections. You will complete the wiring by making interconnections between the various circuit boards and the experimental chassis. First connect the power and ground leads of the M2 and A1 circuit boards to power and ground on the experimental chassis. Complete the wiring of the red (clear) push button by connecting the C lead to a convenient ground point. Solder a length of hookup wire with an alligator clip on one end to terminal $\overline{MA4}$ of the M1 circuit board. Now make the following connections:

FROM	TO
Lamp driver L4	OVFL (A1)
SUB (A1)	GROUND
AØ (A1)	IC2-10
MDA (A1)	MDA (M2)
SUM (A1)	IC2-1
OVFC (A1)	W78 (M1)
CLR (A1)	ASD (A1)
CLR (A1)	CLR (M1)
CLK (A1)	46
A (M1)	A (M2)
B (M1)	B (M2)
C (M1)	C (M2)
ADR4 (M1)	STE (M2)
MA4 (M1)	MA4 (M2)
STA (M2)	IC2-12
EXEC (M2)	IC2-13
Input of Gate 1	IC2-11

This completes the wiring for this experiment. Carefully check over your work to make sure all connections are securely made and that you have made the power and ground connections to all circuit boards and the switch register.

Now remove the IC's from the IC1 and IC2 sockets. In their places insert two 7495 IC's (which you must obtain from the A1 circuit board). Temporarily remove the 7475 (IC4) and the 7442 (IC7) from the M1 circuit board. Also remove the 7420 (IC12) from the M2 circuit board. In its place install a 15844 IC. You can use a 15844 from either the C1 or C2 circuit board.

Step 1: To demonstrate the addition of the 8421 BCD numbers.

The two numbers that are to be added are loaded into the two 4-bit registers by the switch register. The outputs of these two registers are then fed to the adder, which adds each bit of the two BCD numbers as they are shifted into the adder. The sum is fed back and shifted into the 4-bit register in the IC2 position. The number initially stored there is lost. The number stored in the register in the IC1 position is replaced with zeros.

The switch register switches should be numbered SØ through S7 from right to left. Switches SØ, S1, S2, and S3 will be used to enter BCD numbers that will be loaded into the register in the IC2 position. Switches S4, S5, S6, and S7 are used to load the other BCD number into the shift register in the IC1 position. The red push button (LOAD) will transfer the contents of the switch register into the two BCD registers. You will be using the clock push button and its associated buffer latch as a clock signal to shift the numbers out of the registers.

First let's demonstrate how the BCD adder can add the decimal numbers 4 and 5. Set the BCD equivalent of the decimal number 4 into switches SØ, S1, S2 and S3 of the switch register. Switch SØ is the LSB position. Set the number 5 in BCD 8421 code in switches S4 through S7 where bit 4 is the LSB position. Turn on the experimental chassis. Press the red LOAD push button. You should immediately notice that the number 4 is loaded into one register as indicated by lights IØ through I3.

Press the red CLEAR push button and also momentarily touch the alligator clip (MA4) to ground to clear all circuits. Then press the clock push button eight times and observe the BCD numbers displayed by the lamps. IØ through I3 should show a BCD 9 while I4 through I7 should show a BCD zero.

Leave switches S4 through S7 set to the BCD number 5, and change switches SØ through S3 to represent the BCD number 7. Press the red LOAD push button and then the CLEAR push button. Again depress the clock push button eight times, then record the states of lamps IØ through I7 and note the two BCD numbers stored in the two registers.

Discussion of Step 1: The operation of the BCD adder circuit that you just demonstrated is best explained by referring to a simplified block diagram of the circuit (Fig. 88-2). The switch register is used to load two BCD numbers into 4-bit registers. Both of these registers contain the numbers that will be added in the adder circuit. The 4-bit accumulator register feeds the adder directly. The other 4-bit storage register feeds the adder through the select logic when the select logic permits it. The sum appearing at the output of the adder is shifted back into the 4-bit accumulator, replacing the number that was there initially. The number originally stored in the other 4-bit register is shifted out.

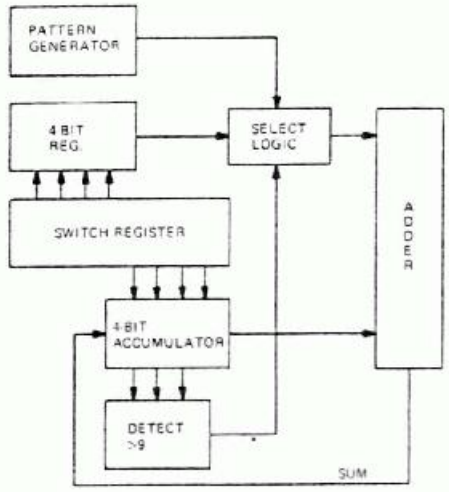


Fig. 88-2. Simplified block diagram of an 8421 BCD Serial Adder.

As you can recall from your study of BCD arithmetic, when two BCD numbers are added and their sum is greater than 9, then a carry is called for to the next digit position. In order to avoid the ambiguity produced by the six invalid binary numbers, 1010 through 1111, it is necessary to add a correction factor of 6 to the sum obtained.

In the first part of this experiment you demonstrated the addition of two numbers, 4 and 5. When these numbers are added one bit at a time, their sum 9 (1001) should appear in the accumulator. It replaces the number 4 stored there initially.

In the next step you added the numbers 5 and 7. The binary adder will, of course, produce the sum of 12 (1100) in the accumulator. This number is an invalid number for 8421 BCD code. The "detect-greater-than-9" circuit monitoring the 4-bit accumulator register will indicate that the sum in the accumulator is greater than 9. This conditions the select logic circuit to disable the contents of the 4-bit register containing the number 5 and enable a pattern generator circuit. The pattern generator generates the serial binary number 6 (0110), which is added to the contents of the accumulator as a correction factor to produce the correct BCD sum in the accumulator.

In the second part of this experiment when you added 5 and 7, the detect-greater-than-9 logic recognized the 12 and enabled the select logic, causing the number 6 to be added to the contents of the accumulator. This leaves the number 2 (0010) in the accumulator. At the same time this causes a carry flip-flop to be set, indicating a binary 1 carry to the next digit position which was displayed by lamp I4, the least signficant bit of the tens BCD display. This carry, in effect, represents a binary 1 in the tens position while the 2 in the accumulator represents the units position, thereby producing the appropriate BCD sum 12.

Fig. 88-3 shows how the correction circuits operate. The latch is set for the first four clock pulses. This condition disables the pattern generator and allows the number stored in the IC1 register to pass through the multiplexer to the adder. If the resulting sum shifted into the accumulator is greater than nine, the two wire ORed NAND gates (detect-over-9) output will go low. This condition will be stored in the latch, which will enable the pattern generator and change the input of the 2-bit multiplexer from the IC1 register to the pattern generator.

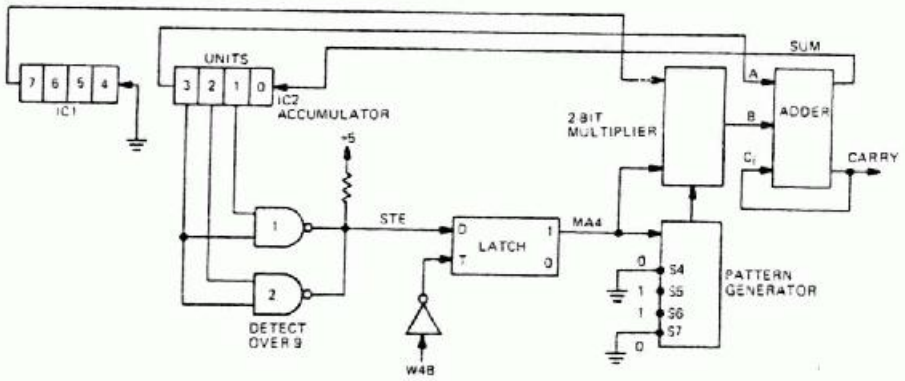


Fig. 88-3. Select Logic Operation.

During the next four clock pulses, the binary number appearing at the S4-S7 inputs of the pattern generator will be added to the number in the accumulator. The four inputs (S4, S5, S6 and S7) are set to 0110, so a decimal six will be added as a correction during the second group of four clock pulses.

If, after the first four clock pulses the number in the accumulator is nine or less, the output of the wire ORed NAND gates will be high, keeping the latch set. With the latch in the set state, there will be no change in either the 2-bit multiplexer or the pattern generator. That is, the multiplexer will continue to send the numbers from register IC1 to the adder. The pattern generator will be disabled. Notice, however, that during the first four clock pulses as the number was being shifted out of IC1 into the adder, the grounded input caused binary 0's to be shifted into IC1. After the fourth clock pulse, then, IC1 contains all zeros. During the next four clock pulses the zeros in IC1 will be added to the number in the accumulator, IC2. What this means is simply that the number in the accumulator (the correct answer) will be shifted out of the register, into the adder, and back into the register since no correction is needed.

Preparing for Step 2. To demonstrate the addition of excess 3 BCD numbers you will need to make several minor changes in your experimental circuit. First, remove the wire connecting terminal IC2-13 to the EXEC line on the M2 circuit board. Also remove the wire connecting IC2-12 to the STA terminal on the M2 circuit board. Remove the wire from terminal IC2-11 to one of the inputs on the four-input free gate. Also remove the wire between the other input of the four-input free gate and EXEC. Move the output of the four-input free gate from terminal STE to terminal EXEC. Remove the 4.7k-ohm resistor between terminal STE and +5 volts on the M2 circuit board. Removing these wires disassembles the detect-greater-than-9 state used in the 8421 BCD adder. You will not need to detect this condition in the excess 3 adder. Remove the 15844 IC from the IC12 position on the M2 circuit board and reinstall the 7420 IC there as it was previously.

Once you have removed the wiring indicated, you can proceed with the additions to make your circuit perform excess 3 BCD addition. Connect OVFL on the A1 circuit board to ADR4 on the M1 board.

Remove the wiring connecting MA4 on the M1 circuit board to MA4 on the M2 circuit board, and connect a piece of hookup wire between MA4 on M2 and STE. Then connect the W4B, W5A, W69, and W78 lines from the M1 circuit board to the four inputs of the free gate on the M2 circuit board.

Remove the jumper wires from terminals S4, S5, S6 and S7 on the M2 circuit board. Connect terminals S4, S6, and S7 together. Connect the S5 line to the MA4 line on the M1 circuit board, then solder a short length of hookup wire between terminal S4 and the $\overline{\text{MA4}}$ line on the M1 circuit board. Remove the lead from W78 (M1) to OVFC (A1) and connect OVFC to W3C on M1. Connect terminal W3C (M1) to terminal $\overline{\text{T}\emptyset}$ (M1). Finally, move the L6 lamp driver lead

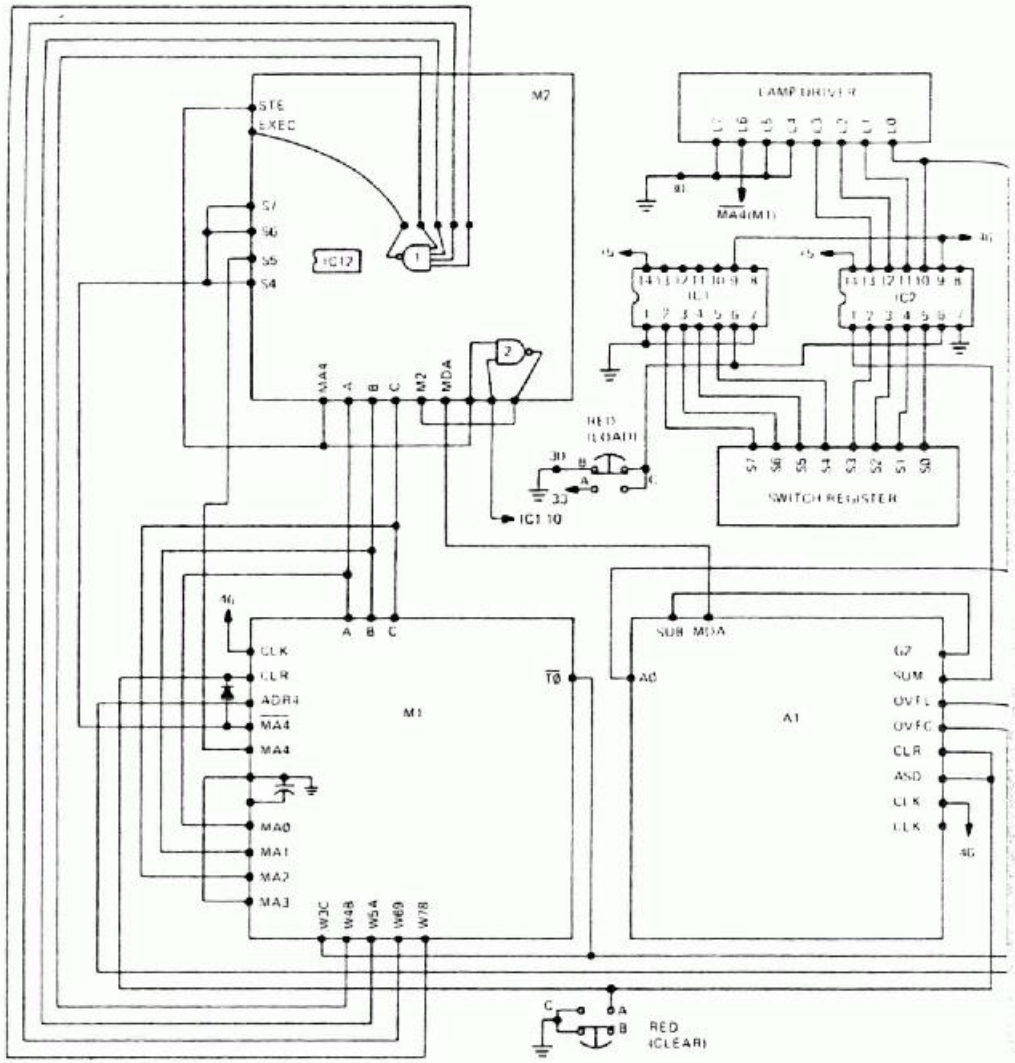


Fig. 88-4. Excess 3 BCD Serial Adder.

from ground to terminal $\overline{MA4}$ on the M1 circuit board and ground lamp driver lead L4 after removing it from terminal OVFL of the A1 circuit board. Double check all of your wiring connections according to Fig. 88-4 to be sure that they are correct.

Step 2, Part I: To demonstrate the addition of two excess 3 BCD numbers.

Set your switch register to enter the decimal numbers 2 and 5 in excess 3 code. Set switches SØ, S1, S2, and S3 to the binary number equivalent to the excess 3 code for 5 (1000). Set switches S4, S5, S6, and S7 to the excess 3 code for the decimal number 2 (0101).

Turn on the experimental chassis. Press the red (clear) push button to clear the counter and then load the two numbers into the shift registers by pressing the red (load) push button.

Next, depress the clock push button four times, making note of the contents of the register in the IC2 position on the experimental chassis by monitoring lamps I3 through IØ. Make note of the number stored there in the margin of the text. Depress the black push button an additional four times to complete the addition, recording the contents of the accumulator register by monitoring lamps I3 through IØ and the number stored in the tens BCD position by monitoring lamps I4 through I7.

Step 2, Part 2: To change the setting of the switch register so that you can add the decimal numbers 8 and 9 in excess 3 code.

Set switches SØ, S1, S2, and S3 to the excess 3 number for the decimal number 9 (1100). Set switches S4, S5, S6, and S7 to the excess 3 code for the decimal number 8 (1011). Press the red (LOAD) push button to load the two numbers into the registers. Depress the other red push button momentarily to reset the counter. Then press the clock push button four times. Record the contents of the accumulator register by monitoring lamps I3 through IØ. Press the clock push button an additional four times, again noting the contents of the accumulator register on the IØ through I3 lamps. Also record the contents of the tens BCD position by monitoring I7 through I4 after four clock pulses and eight clock pulses.

Discussion of Step 2: In this step you modified the basic BCD adder circuit to handle excess 3 numbers. The modifications that you made to the circuit were basically those involving the removal of the greater-than-9 detection circuit you used in the 8421 BCD adder, and the changing of the numbers produced by the pattern generator circuit. The remainder of the circuit is essentially the same.

In the first part of this step you added the numbers 2 and 5. You entered them in excess 3 form into the two registers of the adder by first setting the switch register. By depressing the clock push button eight times you caused the adder to produce the correct excess 3 sum. The following sequence illustrates the operation carried out by the adder.

2	0101
+ 5	1000
	1101
+ *correction factor*	1101
7	1010

Notice the decimal numbers 2 and 5 are expressed in their excess 3 form. The first four clock pulses cause the two excess 3 binary numbers to be added with their sum appearing in the accumulator. You should have recorded a 1101 number after the first four clock pulses.

The number stored in the register (1101) is not the correct excess 3 value for the number 7. It should be 1010. In order to correct this condition we must add the correction factor 13 (1101) to the number in the accumulator register after the first four clock pulses. This is the purpose of the pattern generator.

Since there is no overflow output from the adder after the fourth or most significant bit of the two numbers is added, then the D flip-flop (MA4) on the M1 circuit board is not set. This causes the MA4

output to be high and supplies a high signal to the S4, S6 and S7 inputs of the eight-input multiplexer that is used as our pattern generator. The MA4 line output is low at this time, keeping the S5 input to the multiplexer low. By interpreting the levels of S4 through S7 inputs as a binary number, we see that they are high, low, high and high. Using positive logic this represents a binary number of 1101 or the binary equivalent of the decimal number 13 in serial form, LSB first. During the last four clock pulses, this parallel number will be passed on to the multiplexer a bit at a time and will appear as a serial number at the output that will be added to the contents of the accumulator. Notice that this unit does not act like a shift register. The clock pulses cause the level at S4 to appear at the output first. Then the level at S5 appears at the output on the next pulse. S6 and S7 follow in similar fashion. Thus the number is read out as 1101. After the 8th clock pulse, then, you should have found the accumulator contents to be 1010. The 1010 number is the appropriate excess 3 number equivalent of the sum 7.

In the next part of the experiment you added the numbers 8 and 9 in excess 3 form. The proper addition is shown as:

8		1011
9		1100
		10111

	+correction factor	0011
17	0100	1010

As you can see, whenever you add these two numbers the sum is greater than 16. This indicates that a different correction factor must be added to the number to produce the correct excess 3 sum. During the first four clock pulses, the two numbers are added as shown and the number in the accumulator should be 0111. The overflow flip-flop should be set at this time, causing the MA4 flip-flop on the M1 circuit board to become set. When this happens it causes the pattern generator to produce the number 0010, the binary equivalent of the decimal number 2. The correction factor needed is a decimal 3 however. The added correction (1) is taken care of by the carry flip-flop in the adder, which is also set at this time. S5 is high because MA4 is set. $\overline{MA4}$ is low at this time, holding S4, S6 and S7 low. This produces a low, low, high, low combination that becomes 0010 when interpreted with positive logic. The number 3 (0010 plus the carry) is added to the contents of the accumulator on the last four clock pulses, producing the correct excess 3 sum in the output. The correct sum, of course, is 17 and the excess 3 representation of this is 0100 1010. The 1010 is the least significant digit (7) in excess 3 form. The most significant 1 is the carry out produced by the MA4 flip-flop and appears as 0100 in the tens digit position.

Refer to Fig. 88-4. The W4B, W5A, W69 and W78 outputs of the decoder on the M1 circuit board are used to form timing and control circuitry for the pattern generator. During the last four clock pulses these four decoder outputs will go low sequentially. Gate 1 in Fig. 88-4 is part of the 7420 (IC12) on the M2 circuit board. This gate is used as an OR, which means that during the last four clock pulses one of the four decoder inputs will hold the output of gate 1 high. The other half of the 7420 is simply used as an inverter. It will hold the STE output and MA4 line on the M2 circuit board low for the last four clock pulses to enable the pattern generator. At this time it also inhibits the gate used to feed the number from the register in the IC1 position on the experimental chassis to the adder. This means that during the first four clock pulses, the number in the register is added to the number in the accumulator to obtain the initial sum of the two excess 3 numbers. During the second four clock pulses, that number is inhibited and the correction factor is added into the number stored in the accumulator to produce the correct excess 3 sum. The state of flip-flop MA4 on the M1 circuit board determines which correction factor number is to be added.

Instructions for Statement 88: For this statement you do not need to make any further hardware changes to your experimental circuit. You will simply answer the following question based upon your knowledge of the operation of the excess 3 adder circuit that you just demonstrated.

After you have answered the statement, disconnect all of the wiring from the A1, M1 and M2 boards. Remove all of the wiring from the experimental chassis at the IC1 and IC2 sockets, excepting +5 and ground at terminals 14 and 7.

Disconnect the switch register outputs from the experimental chassis, but leave the +5 volts and ground leads connected. Disconnect the lamp driver inputs from the experimental chassis, but do not remove the +5 volts and ground leads.

Reinstall the 7442 IC at IC7 and the 7475 IC at IC4 on the M1 board. Also reinstall the two 7495 IC's at IC39 and IC40 on the A1 circuit board.

Statement No. 88: To produce the appropriate excess 3 BCD sum, a correction factor number is added to the initial sum of the BCD numbers added. If the initial sum is less than 16, a correction factor of 13 is added. If the initial sum is greater than 16, a correction factor of 3 is added to produce the correct excess 3 sum. In the experimental circuit, which of the following circuits determines which correction factor is to be added?

(1) decoder outputs W4B, W5A, W69 and W78.

(2) the overflow flip-flop.

(3) the pattern generator.

(4) the carry flip-flop.

Constructing the C1 Circuit Board and Performing Experiment 89

Before you can begin the next experiment, you will need to construct the C1 (EC32) circuit board. Use the following step-by-step instructions for doing this. You will need the following parts:

1	Printed circuit board (EC32)
54	Tubular terminals (LU9)
6	14-pin IC sockets (SO84)
2	16-pin IC sockets (SO86)
1	100 mfd electrolytic capacitor (CN112)
9	.01 mfd disc capacitor (CN102)
1	7400 IC (IG5)
3	7404 IC (IG12)
1	7420 IC (IG6)
2	7442 IC (IG10)
1	74164 IC (IG14)

Install miniature tubular terminals in all of the peripheral holes on the C1 circuit board that are designated by lettered callouts. Press the terminals into the holes from the top of the board firmly. Use your longnose pliers to crimp each terminal flat on the bottom of the board, ensuring a firm mechanical mount. Solder all 54 terminal-to-pad connections on the bottom side of the board. Then solder the 15 terminals to the connections on the top side of the board.

Install .01 mfd, 50 volt disc ceramic capacitors in the nine locations designated on the circuit board. On those capacitors mounted near integrated circuit socket locations, be sure to solder the wire leads on both the top and the bottom of the circuit board. When you install those capacitors designated on the periphery of the board, cut the leads short and install them in the holes of the tubular terminals. Solder all connections.

Install eight integrated circuit sockets on the C1 circuit board. Two of these, IC21 and IC23, are 16-pin sockets. The remaining six are 14-pin sockets. Install the sockets so that the notch in the end edge aligns with the notch as shown on the screening on top of the C1 circuit board. Be careful to position the 14-pin socket for IC18 exactly as shown by the legend on the board. Press each socket firmly into place so that it seats completely on the top of the board. Then, using a low wattage, fine point soldering iron and the extra thin solder supplied with this kit, solder each of the terminals carefully, being sure that adjacent terminals do not short and that excess solder does not appear on each connection.

Install the 100 mfd, 10 volt capacitor between the +5C and G1 terminals of the circuit board. Be sure to observe the polarity of the markings indicated.

Install the IC's in the sockets according to the screened designations by each socket. Examine all of your soldering to be sure all terminals, pins, and components are soldered and that there are no solder bridges to adjacent points. Make sure all of the integrated circuits are in the correct sockets.

EXPERIMENT 89

Purpose: To demonstrate the circuitry used for decoding instructions in a digital computer and

To show a method of testing this circuitry to verify its proper operation.

Introductory Discussion: The instruction word in a digital computer defines the operation to be performed by the computer. The instructions making up a program are stored in sequential memory locations. The computer fetches these instructions one at a time, interprets them, and then generates the necessary signals to carry out the operation. The circuitry for interpreting the instruction is in the control section of the computer. This circuitry is basically a set of decoders for recognizing the various bit patterns in the instruction word and generates the necessary control signals to cause the specified operation to be carried out. In this experiment you are going to learn the instruction format of the NRI digital computer and you will demonstrate the logic circuitry used to monitor and interpret these instructions.

While learning the instruction decoding circuitry, you will also be demonstrating the method for testing logic circuitry t determine if it is operating properly. Th can be done by first knowing how the ci cuit operates. Its operation can be simu lated by applying appropriate inputs determining if the outputs produced co respond to those inputs.

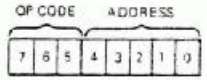


Fig. 89-1. Instruction word format memor reference instructions.

Fig. 89-1 shows the basic instructio word format for the NRI Model 832 Di gital Computer. This is the instructio word format for memory referenc instructions. Notice that bits 5, 6 and ' are used to specify the op code. This 3-bi code defines which operation the com puter will execute. Since there are thre bits in the op code, then there are 8 pos sible instructions that can be performed as specified in Table I. Remember that al of these instructions (except the last one) are memory reference instructions, whic means that a word stored in some mem ory location is used when this instructio is executed. The address portion of the instruction consists of bits Ø through 4

MNEMONIC	INSTRUCTION OPERATION
LDA	Load accumulator (A)
STA	Store accumulator
ADD	Addition
SUB	Subtraction
JMP	Unconditional jump
JOM	Jump if accumulator minus
JOZ	Jump if accumulator zero
OPR	Operate functions

Table I. Memory reference instructions.

MNEMONIC	FUNCTION OPERATION
RAB	Rotate A and B registers
CMA	Complement accumulator
AND	AND contents of A and B registers
RAE	Rotate A and E registers
SHA	Shift accumulator
SKC	Skip on condition
DCA	Decrement accumulator
HLT	Halt

Table II. Operate instruction functions.

and designates one of 32 possible memory locations where the data to be operated on is located.

In Table I notice that the last instruction word is the operate instruction (OPR). When the op code defining this instruction appears in bits 5 through 7, the bits Ø through 4 of the memory word are not used to specify an address. Since there is no word in memory referenced by the OPR instruction, bits Ø through 4 can be used for other purposes.

Fig. 89-2 shows the word format when the OPR instruction is indicated. Bits 5 through 7, of course, define the operation instruction. Bits Ø, 1 and 2 define a function code which in reality specifies eight additional functions that the computer can perform. These are instructions that do not need to reference a memory location. Almost all of the instructions involve operations on the accumulator and other registers in the computer.

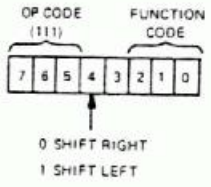


Fig. 89-2. Operate (OPR) instruction word format.

The operate instructions are defined in Table II. The SHA instruction is a shift accumulator operation that causes the contents of the accumulator to be shifted one bit at a time, to either the right or left. The direction of the shift is designated by bit 4 in the operate instruction format.

In this experiment, you are going to demonstrate how the C1 circuit board you just constructed is used to decode and recognize the various instruction codes indicated here. You will apply instruction words to the inputs and monitor the outputs to determine the functioning of the circuit.

Experimental Procedure: With this kit you receive an insert diagram showing the logic circuitry on the C1 circuit board. Find this drawing and keep it handy, as

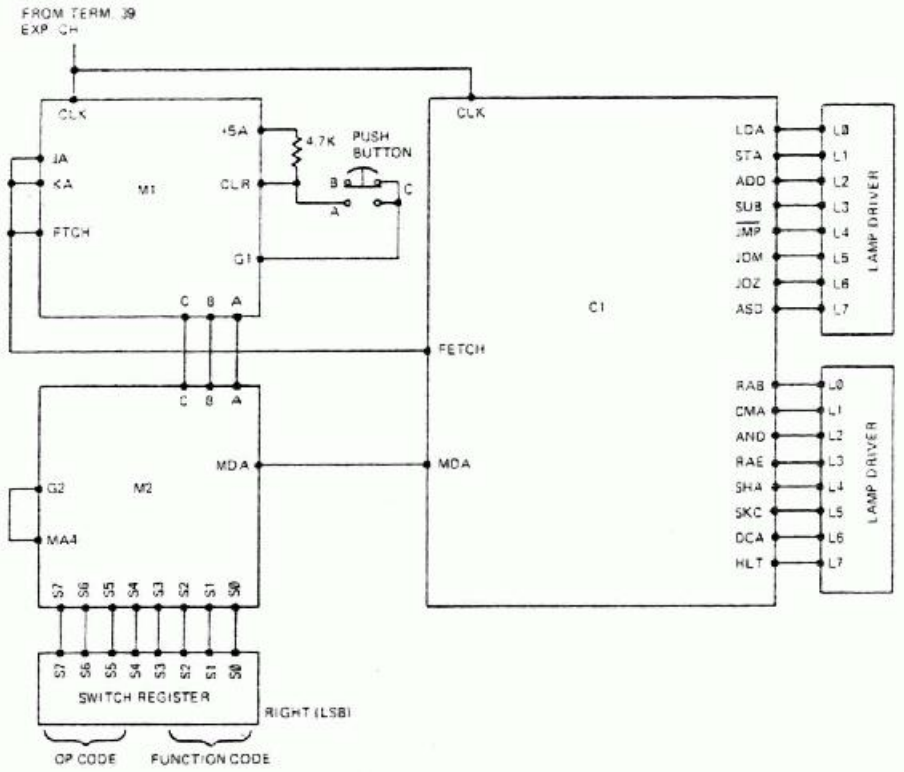


Fig. 89-3. Experimental circuit for Steps 1 and 2.

you will be using it in performing this experiment. The C1 circuit board is designated by its part number EC32.

Looking at the logic diagram of the C1 circuit board, you can see that it consists of an 8-bit instruction register designated IC24. The 74164 IC used here contains an 8-bit shift-right shift register that is loaded serially from the MDA input. This register is designated the instruction register (IR), as it stores the 8-bit instruction word. The outputs of this register are designated IRØ through IR7. These outputs feed two instruction decoders designated IC21 and IC23. IC21 is the decoder used to interpret the memory reference instructions, while IC23 interprets the operate instruction codes. The remainder of the circuitry consists of inverters used to obtain the proper logic levels at the outputs. Several other logic gates are used for generating special signals that are required in the computer.

Fig. 89-3 is the wiring diagram for the circuit that you will use in this experiment. You will need your switch register, the M1 and M2 circuit boards, the C1 circuit board, and both lamp drivers. Be sure that all of these units are properly connected to the 5-volt supply on the experimental chassis, then wire the circuit as shown in Fig. 89-3. Keep the lead lengths short where possible and

arrange your boards neatly to provide ready access to any part of the circuit.

For this experiment you are going to load various 8-bit words into the instruction register and then monitor the results of the decoder outputs on the 16-lamp driver lamps. To do this you will set your switch register to the appropriate word you wish to load into the instruction register. By depressing the red push button, you will automatically start the circuit which will cause that number to be loaded into the instruction register. This is done by the circuitry on the M1 and M2 circuit boards. The 8-bit memory multiplexer on the M2 circuit board is used to take the 8-bit parallel data from the switch register and convert it into a serial pulse train. The MDA output of the M2 board is applied to the serial MDA input on the C1 board. The counter on the M1 board is used to step the multiplexer on the M2 board, and also is used along with the FTCH signal to cause exactly eight clock pulses to shift the 8-bit shift register word into the instruction register and stop automatically.

Step 1: To verify the operation of the memory reference instruction decoding circuitry.

In this step you will apply the eight possible op codes to the circuit and note the condition of the lamps at the output. For this step you will determine which instruction is being specified by each of the op codes that you apply to the circuit. You will do this by observing which lamp turns on for each step. By noting which lamp is on you can determine which of the C1 output lines is a binary 1, as shown in Fig. 89-3.

Turn on the experimental chassis and set the switch register so that the three left-hand bits are set to 000. The other bits in the switch register can also be set to zero for this particular step. The five right-hand bits in the instruction word specify the address for this step. We will not be concerned with the address. Once the switch register is set, depress the red push button, which should load the instruction register automatically. At this time you should observe all of the lamps to determine which one is a binary 1. Knowing which lamp is on helps you to determine which C1 output line is a binary 1 by tracing the circuitry back from the input to the lamp driver that is on the back of the terminal on the C1 circuit board. Record the mnemonic designation on the board for that terminal in the space adjacent to the op code in Fig. 89-4.

OP CODE	INSTRUCTION
000	
001	
010	
011	
100	
101	
110	
111	

Fig. 89-4. Table for Step 1.

Set the switch register to the next op code in the table of Fig. 89-4. Depress the red push button to load the op code and again observe the lamp that is on and trace the circuitry to determine which instruction is being decoded. Record the mnemonic designation in Fig. 89-4. Do this for all eight op code conditions shown in Fig. 89-4.

There are two special cases that you will note in performing this step. Because the $\overline{JMP}$ output on the C1 circuit board is a complementary output signal, this will normally be on at all times except during the time which the JMP instruction is decoded. For that reason, simply disregard the $\overline{JMP}$ output line and the condition of lamp I4, except to note when that lamp goes out rather than when it goes on.

Second, the I7 lamp on one of the lamp driver boards will be monitoring the ASD output. As you can tell from Tables I and II, this is not an instruction. Instead it is a special control signal generated for a special purpose in the computer. For each op code input you are to note the status of this ASD line and record in the margin of the text the mnemonic of the instruction that is enabled at the same time that the ASD lamp is on. In other words, if the particular op code just loaded causes the I7 lamp to light, then you will know that the add instruction has been designated. If this add instruction op code also causes the ASD line to be a binary 1, and the I7 lamp to light, make note of this fact in the margin of the text. Do this for all eight op code conditions.

Discussion of Step 1: Fig. 89-5 shows the correct results that you should have obtained in this step. The thing that you should be most aware of in this experiment is that only one of the eight possible instruction output lines should be a binary 1 at any given time. The reason for this is simply that the decoder is designed to recognize only one instruction and cause that single instruction to be carried out. All of the others should be disabled at this time.

OP CODE	INSTRUCTION
000	LDA
001	STA
010	ADD
011	SUB
100	JMP
101	JOM
110	JOZ
111	$RAB_{(OPR)}$

Fig. 89-5. Correct results for Step 1.

There are two special cases in Fig. 89-5 that you should understand. First of all, by referring to the separate logic diagram (insert sheet) of the C1 circuit board, you can see that the output for the JMP instruction is a NOT signal. This means that when that instruction is decoded or selected, the output line will be a binary 0 rather than a binary 1. The decoders themselves, 7442 IC's, have an active low output. This means that the selected output line will be low while all others will be high. Since we generally require a binary 1 output on the selected line, inverter packages using the 7404 IC are used. During your experiment you should have found that the $\overline{JMP}$ line, which is connected to the lamp driver and displayed on lamp I4, should be on, indicating a binary 1 for all input conditions except for the op code 100.

Notice in Fig. 89-5 that the OPR instruction is not indicated. The OPR output of the instruction decoder, IC21 on the C1 board, is used to control the OPR function decoder which is IC23. When the OPR line goes low, it enables the function decoder IC23 and permits one of the eight possible operate instructions to be enabled. The exact operate

instruction is selected by setting the three most significant bits of the instruction register, IRØ, IR1, and IR2. If there are all zeros in this portion of the instruction word that you entered, the RAB instruction is selected; therefore, the light associated with that instruction, IØ, will be on when the operate instruction code is in the instruction register.

Step 2: To determine the function codes for the operate instructions.

For this step you will load the op code portion of the instruction register with the code 111, which designates the operate instruction. You will then apply the eight possible function codes and determine which of the operate instructions are designated for each. You will set the three right-most bits of your switch register to the eight codes, 000 through 111, one at a time. Load each one individually and enter it into the instruction register by depressing the red push button. After each operation, note again which lights are on. By noting the indicator lamps that are on, you can trace their input back to the C1 circuit board, and determine which operate instruction is being specified by noting the screening on the circuit board. Do this for each of the eight codes and record your results in Fig. 89-6. In addition, continue to monitor the ASD output on L7 as you did before. Note which of the operate instructions, if any, also cause the ASD output to become a binary 1, as indicated by the I7 lamp being on. Make note of these in the margin of the text along with your notations regarding the ASD signal, which was determined in the previous step.

FUNCTION CODE	INSTRUCTION
000	
001	
010	
011	
100	
101	
110	
111	

Fig. 89-6. Table for Step 2.

Discussion of Step 2: Fig. 89-7 shows the correct results for Step 2. Here you can see which of the 3-bit function codes in bits 0, 1 and 2 of the instruction register correspond to the operate instruction. Remember that only one of the sixteen possible instructions available should be selected at any given time, as indicated by the lights. Keep in mind that all during this time the $\overline{JMP}$ output will be high since this instruction is not selected. This will keep lamp I4 turned on. However, you should have remembered that this is a complement output signal and will cause this result.

FUNCTION CODE	INSTRUCTION
000	RAB
001	SHA
010	AND
011	DCA
100	CMA
101	SKC
110	RAE
111	HLT

Fig. 89-7. Correct results for Step 2.

Instructions for Statement 89: You will not need to make any further tests to answer the statement question for this experiment. During Steps 1 and 2 you have accumulated the necessary information to answer the statement question, which is based upon the state of the ASD output signal. During Steps 1 and 2 you were told to monitor ASD output line on L7 and to indicate which instruction is decoded when this lamp came on. Using this information, you will answer the statement question. You may wish to refer to the separate logic diagram (insert sheet) of the C1 circuit board to further verify the operation of the circuit concerned with the ASD signal.

After you have answered the statement disassemble your experimental circuit. Remove the M1, M2, and C1 circuit boards and all of the various interconnections, including those to the switch register. Remove the jumpers between JA, KA, and FTCH on the M1 board as well as the 4.7k-ohm resistor and switch on +5A, CLR and G1. Remove the jumper between MA4 and G2 on the M2 board. Unsolder the power and ground connections of M1, M2, and C1 at the experimental chassis. Leave the power and ground connections for both lamp driver boards and unsolder the lamp driver inputs from the C1 board at the C1 terminal.

Statement No. 89: Using the information that I gathered in this experiment and by observing the circuitry involved with the ASD signal on the C1 circuit board, I can see that it is possible to express the ASD output with a Boolean expression. The proper Boolean expression for the ASD signal is:

(1) $ASD = SUB + ADD + DCA$

(2) $ASD = \overline{SUB} + \overline{ADD} + \overline{DCA}$

(3) $ASD = (SUB) \cdot (ADD) \cdot (DCA)$.