EXPERIMENT 8
Introduction to Digital Circuits:
Timing, Gates, Flip-flops and Counters

Digital electronics carry information as a series of pulses on a single line (serial) or on several lines simultaneously (parallel). It is the number of pulses sent, the pattern of pulses or their frequency which carries the information. This is in contrast to analog systems where the information is carried as the amplitude or frequency of the voltage or current. The digital method of information transfer requires accurate timing and pulse generation.

Part I of this lab will be devoted to the basic aspects of timing and pulse generation circuits. Part II will focus on the use of digital circuits for logical operations. In Part III basic digital gates will be combined to assemble more complex logic arrays. The arrays (often combined as single integrated circuit (IC) chips) will be used to construct counters (some of which can drive numeric displays).

I. Timing and Pulse Generation

The pulse generation circuits we will focus on are a combination of analog and digital circuits. They may produce a continuous series of pulses (these are called astable multivibrators) or a pulse of a specific duration (these are called monostable multivibrators). Combining two or more of multivibrators provides generation of a desired pattern of pulses (including pulse width, time between pulses and frequency of pulses).

The multivibrators are useful in many simple circuits where high speeds and absolute synchronization are not necessary. (In contrast, computers use crystal-controlled oscillators and totally digital timing by counting clock pulses.) The 74121 is one of the most versatile and widely used monostable multivibrators. The 74121 is capable of generating pulses as short as 30 nanoseconds in width. (See http://www.national.com/pf/54/54121.html for specification sheets, pin configuration etc.)

The 74121 device has several input and output options. A1 and A2 inputs are negative edge triggered logic gate inputs that will trigger the one-shot when either or both go to logical zero (less than 0.8 V) when B is a logical one (greater than about 2 V). However, since the A inputs trigger at a specific voltage on the negative slope of the input pulse, there is no “noise immunity”. If the input signal slope is sufficiently slow and noisy, the one-shot may trigger repeatedly.

The B input is a Schmitt-trigger with a much higher immunity to noise. It triggers the one-shot to output a pulse with A1 or A2 is at logical zero and B goes to logical one (positive edge triggered).

Set up the one-shot as shown in Figure 1.

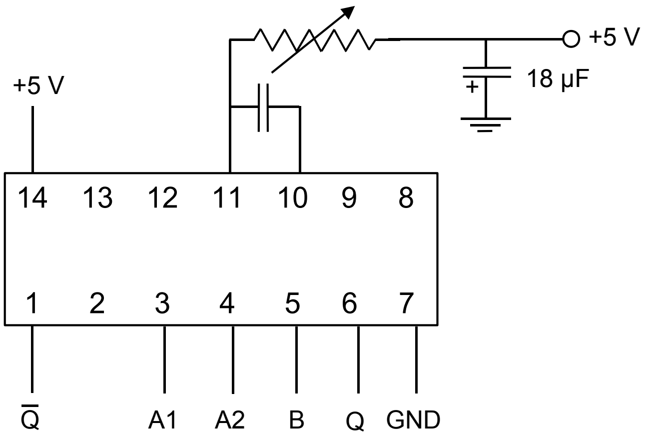


Figure 1

A. **Comparison of experimental and theoretical pulse width.** Connect A1 and A2 to ground, B to the input signal. Look at the effect of different values of C (0.001 and 0.033 μF) and different values of R on the outputs (Look at input and Q on channels 1 and 2 of the oscilloscope). Plot the expected pulse width (from the values of R and C and the specification sheets) vs. the observed pulse width.

B. **Triggering levels for A and B inputs.** With R=0 and C=0.033 μF, look at the difference in input characteristics of the A and B inputs using a 50 kHz square wave signal. Adjust the magnitude of the input signal (starting at 0 V_{p-p}) so that triggering is just initiated. Do this for both the A and B inputs. What voltages are required for triggering each?

C. **Comparison of the noise immunity characteristics of the A and B inputs.** Trigger the monostable with a waveform which is the sum of a square wave of variable magnitude at a low frequency (approximately 10 Hz) and a higher frequency sine wave (approximately 60 Hz). We will treat the square wave as the "signal" and the 60 Hz sine wave as "noise". Use the "Arbitrary Waveform Generator" feature of your ELVIS station to output the waveform (see instructions below):

To generate the waveform: Open the "Arbitrary Waveform Generator" from the ELVIS Instrument Launcher panel. Select "Waveform Editor", which brings up a new window. Set the sample rate to 1000 Hz and the duration to 1 second. Press the "New Component" button twice. Select the first component, and change the type to square, the amplitude and offset to 0.5, and the frequency to 10 Hz. Select the second component and change its type to sine, it's amplitude to 0.5, its offset to 0 and its frequency to 60 Hz. This should give you a wave which looks like the wave shown in Figure 2. Then choose File -> Save As, select "Waveform File." Click next and set sample rate and number of samples to 1000. Click next again and save the waveform you created with a descriptive name in a location you will remember. Then close the waveform editor (don't save changes if prompted). Back in the Arbitrary Waveform Generator window, press the folder icon next to the DAC0 Waveform Name dialog. Find the file you just saved and open it. Then press the DAC0 play button to begin sending your waveform to the DAC0 pin of your ELVIS station.

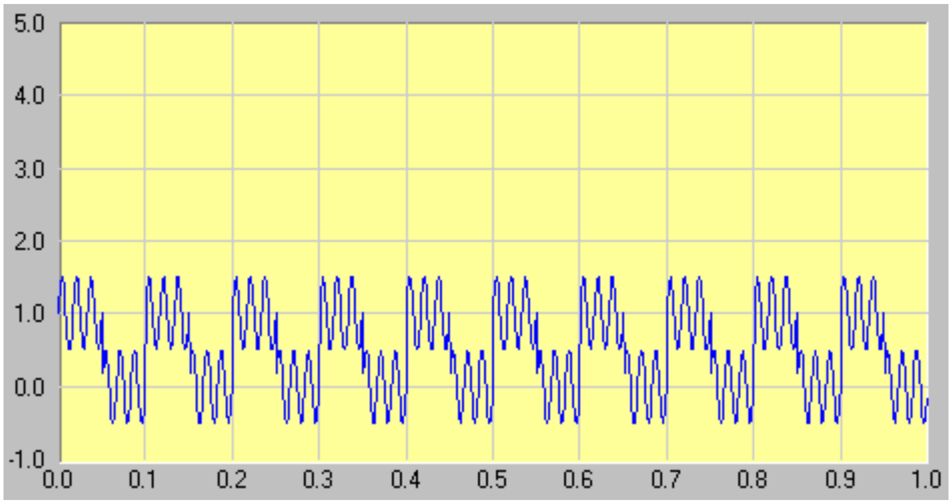


Figure 2. Waveforms from the A/D converter

First we will investigate the A inputs. Connect the B input to 5 V so that triggering of the monstable will be controlled by the A inputs. Connect the A input to the DAC0 pin, which should be set up to put out the waveform shown above. Set up the monostable with a pulse width of 5 ms. Beginning with the square wave "on" voltage of 1 V and a sine wave amplitude of 1 V_{p-p}. Increase the upper level of the square wave in 1 V increments up to a voltage of 5 V, observing the output of the monostable for each. Is the frequency of the monostable output 10 Hz or 60 Hz?

Now perform the same experiment using the B input to trigger the monostable. What is the frequency of the monostable output? Comment on the noise characteristics of the A versus B inputs. (You might want to try other values for the peak-to-peak voltage of the sine wave).

D. Generation of pulses of desired delay and width. Set up the circuit shown in Figure 3 to generate output pulses of a desired width and delay from the input signal. (Use variable resistors for each monostable. Select values for R and C that produce delays and widths that are conveniently viewed on the oscilloscope.) A timing diagram for the circuit is shown in Figure 4.

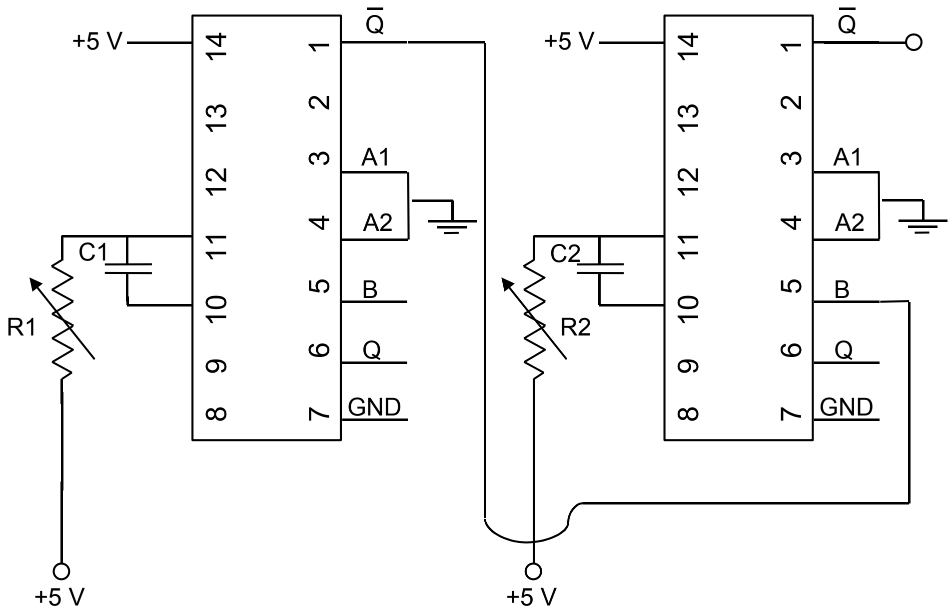


Figure 3. Two 74121s to produce a variable width, variable delay pulse.

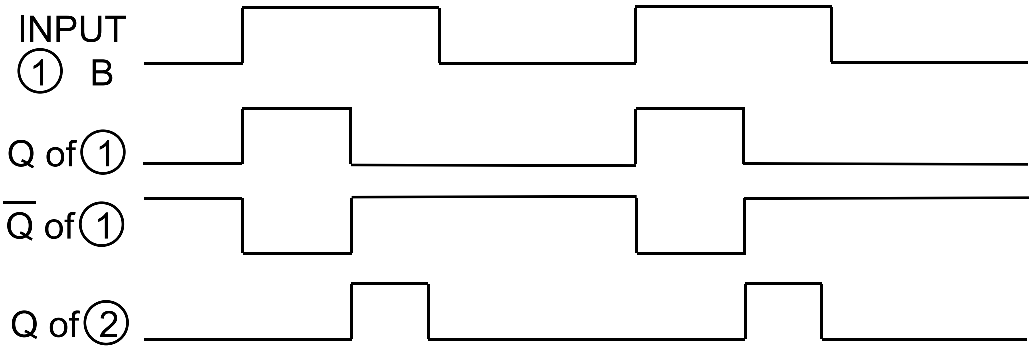


Figure 4. Timing diagram for dual monostable circuit.

Trigger the oscilloscope externally from the waveform generator. Study the variable delay (position of leading edge of the output pulse at Q of the second 74121 relative to the leading edge of the input signal) and variable width introduced by the first and second monostables, respectively.

II. Introduction to Digital Logic Gates - TTL Logic Gates

A. **TTL NAND gate.** We will use the 7400 Quad-two-inputs NAND gate IC (i.e. there are four NAND gates on a single chip and each NAND gate has two inputs). Connect the 7400 properly, using the specification sheet http://www.national.com/pf/54/5400.html. Note that unused gate inputs "float high", representing a HI or 1 level and will not affect the NAND operation of the connected inputs. However, it is possible for an unused input to pick up noise. Therefore, unused inputs should be connected to +5 V. In general, it is good practice to minimize the number of unused inputs. (Use a 2-input gate rather than a 4-input gate to build an inverter from a NAND gate, for example).

Verify the logic function of the NAND gate.

Determine the input voltage levels corresponding to HI and LO. Connect a variable dc voltage source to the NAND gate input. CAUTION: always stay in the range 0 to +5 V. Monitor the output voltage as you slowly change the input voltage to determine the 0 and 1 level ranges for the inputs.

Measure the current flowing in or out of one of the inputs when it is held LO. What is the magnitude and direction of this current?

B. **TTL Inverter.** Use the 7404 IC hex-inverter (six inverters on one chip. Specification sheet is at http://www.national.com/pf/54/5404.html).

Connect +5 V to the inverter input. The inverter output will then be LO. Measure the amount of current the inverter output can sink before it rises above 1 V. (This can be done with a +5 V power supply connected to the inverter output through the decade resistance box while measuring the voltage drop across the resistor. Be sure to start with a high resistance and then decrease it.)

Is the inverter output a current sink or a current source? What determines the fan out of such a circuit? What is the fan out of the inverter? Show the circuit you used to make the measurement.

Recall that the output of a TTL gate is connected to ground when it is in the low state. Therefore, current flows into it from the device connected to the output. Also note that inputs must be pulled low to input a LO signal. If disconnected, the voltage at the input may be in the HI state.

C. **Binary addition.** Wire the half-adder circuit shown in Figure 5a using NAND gates. Wire a second half-adder and combine it with the first to make a full-adder as shown in Figure 5b. Verify the operation of each of these circuits.

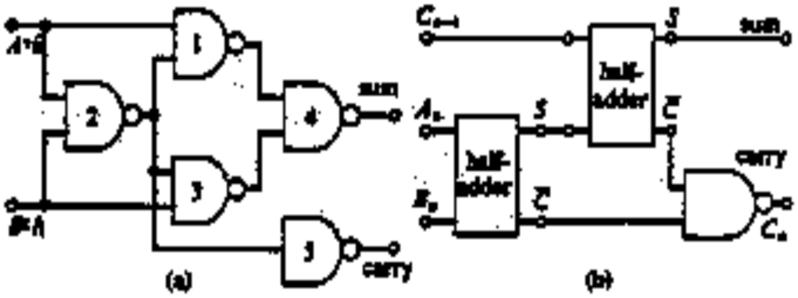


Figure 5. **a. Half-adder** **b. Full-adder**

III. Flip-flops: memory and counters

A. Logic Gate Memory

The simplest memory circuit, a flip-flop can be constructed from a pair of NAND gates as shown in Figure 6. The flip-flop has two stable states and can be readily set in either state.

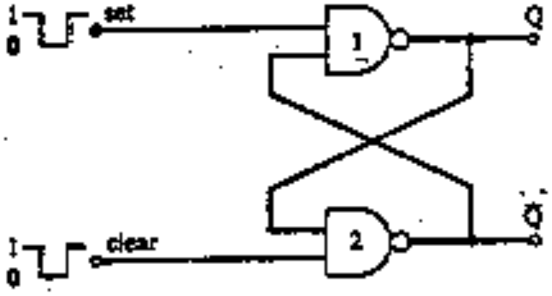


Figure 6. **A simple flip-flop.**

The outputs can be easily monitored by connecting them to the LAMP INDICATORS on the prototype board. The virtual LOGIC SWITCHES (remember that switches “bounce”) from ELVIS can be used to generate momentary low inputs for flip-flop circuits. One way to do this is to push the toggle button in the digital writer software; this will quickly invert the logic and when the toggle is pressed again the switch will be in its original state. You will need to use two virtual switches to make this work. By pushing the toggle button all of the digital outputs are inverted. (Note you may want them both in the same state to start off.)

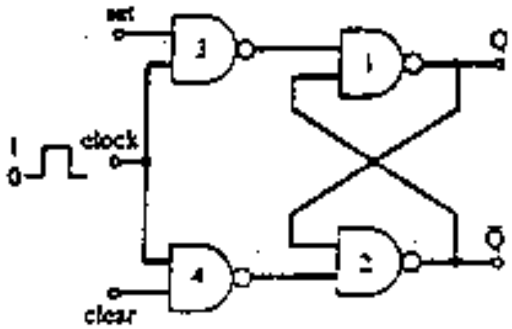
Verify that a momentary LO input to NAND gate 1 sets the Q output HI (and not Q LO) and that a momentary LO input to NAND gate 2 sets the Q output LO (and not Q HI).

Also verify the truth table for the following conditions:

CLEAR	SET	Q	not Q
0	0		
1 (Last 0)	1		
1	1 (Last 0)		

B. Gated Memory

By adding two additional NAND gates, memory can be constructed which responds to input levels only during a specified time interval. In other words, the inputs can be enabled or disabled. Construct the NAND gated memory shown in Figure 7. Use the LOGIC SWITCHES for the SET and CLEAR (reset) signals and a PULSER (or a square wave from the ELVIS function generator set at a very low frequency) for the clock. Follow the two outputs by using the LAMP INDICATORS. Determine the truth table and compare it to what you expect. Why does the SET = 1, clear = 1 input condition result in indeterminate output?



Inputs		Outputs	
Clear	Set	Q	Q̄
0	1		
1	0		
0	0		
1 (Last 0)	1		
1	1 (Last 0)		

Figure 8. a. NAND gated memory. b. Truth table.

C. JK Master-Slave Flip-Flop

Figure 7

C. JK Master-Slave Flip-flop

The JK flip-flop is an important circuit element used for digital signal processing. A basic JK flip-flop can be made from NAND gates as shown in Figure 8. However, the JK flip-flop is available as an integrated circuit, so we will use the 7476 edge-triggered JK flip-flop IC chip.

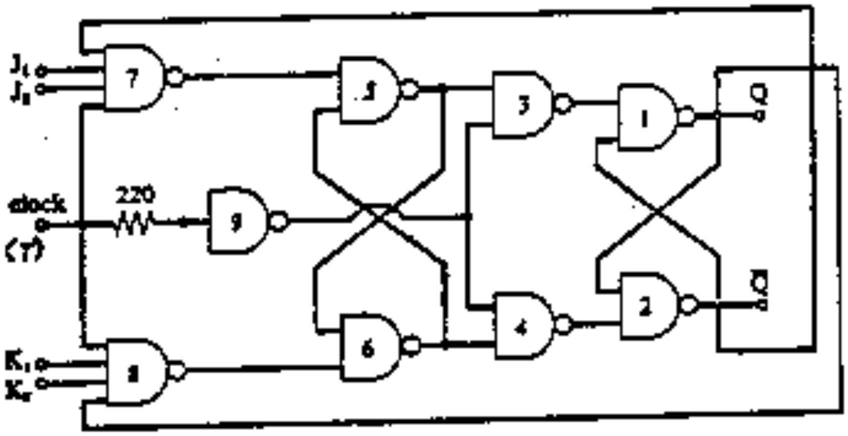


Figure 8. JK **flip-flop made from NAND gates.**

Verify the truth table for the JK flip-flop. Pin configurations for the 7476 are given with its specifications at http://www.national.com/pf/54/5476.html.

Set up the circuit shown in Figure 9 with a 74121 monostable as a variable pulse width generator. Choose R and C to produce a low duty cycle (small ON or HI time, long OFF or LO time) train of pulses at 1 Hz, similar to that shown in Figure 9.

Note that the output from the flip-flop is a square wave which is essentially perfectly symmetrical (duty cycle = ½) even though the input is asymmetrical (duty cycle is much less than ½). Why? What is the exact point in time relative to the clock input (to the JK flip-flop) at which the JK flip-flop toggles?

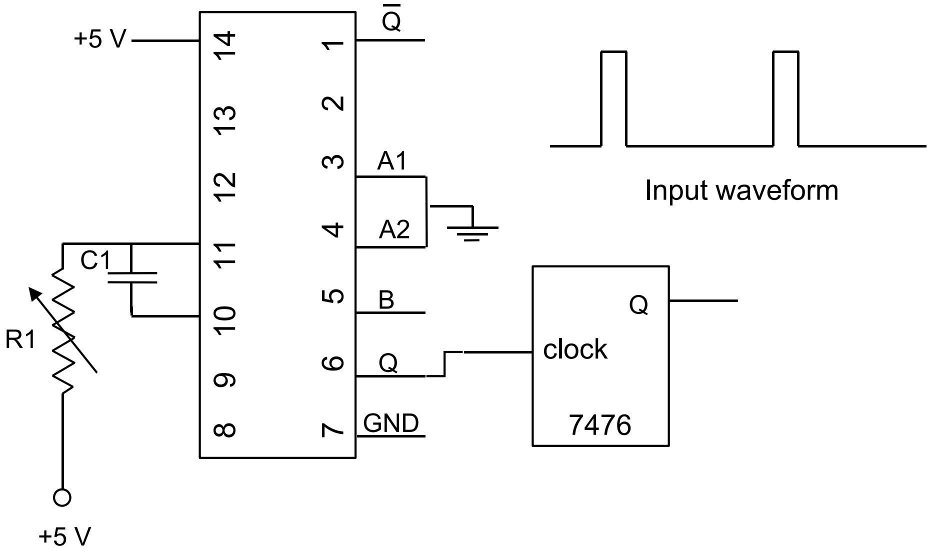


Figure 9. Variable pulse width input to JK flip-flop.

Now drive the JK flip-flop input directly from the ELVIS function generator at 1000 Hz. What is the output frequency? Suggest a way to divide the input frequency by 8.

D. Binary Counters

The binary counter is the simplest and most efficient form of the counting register. The output of each flip-flop outputs to the INDICATOR LIGHTS to display the stored binary number. RESET the counter with a momentary low pulse to the CLEAR (or reset) input. Use a PULSER to generate input pulses.

Note that you MAY need to use the debouncer circuit shown in Figure 11 to eliminate spurious input pulses. Try it without the debouncer circuit and see what happens. If the behavior is not as expected, use the debouncer circuit. Why is the debouncer not needed for the CLEAR input?

Verify that the output is a count of the number of pulses input since the counter was reset. What happens after 16 pulses are input?

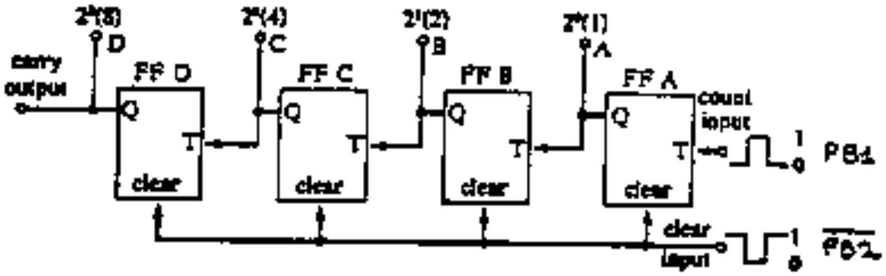


Figure 10. 4-digit binary counter.

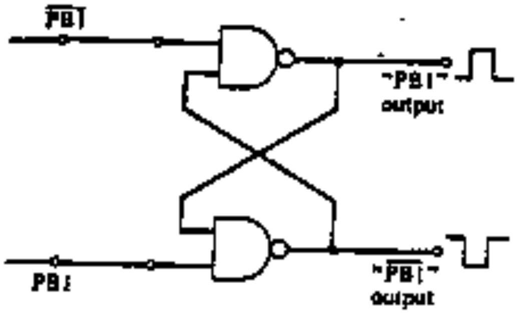


Figure 11. **Debouncer circuit.**

Connect a 1000 Hz square wave from the ELVIS function generator to the input replacing the PULSER. Measure the output frequencies at each flip flop. Make a timing diagram showing the input and the 4 outputs all on the same horizontal time scale.

Signal	Output vs. Time	Frequency
INPUT		
FLIP FLOP A OUTPUT		
FLIP FLOP B OUTPUT		
FLIP FLOP C OUTPUT		
FLIP FLOP D OUTPUT		

How many flip-flops would be required to count 700 pulses? How could the binary counter used above be converted to count down instead of up?

E. Binary Coded Decimal (BCD) Counters

A synchronous BCD decade counting unit is shown in Figure 12. (**Don't Build it.**) The counter is made from four, multiple-input JK flip-flops. The multiple J and K inputs are used to make the counter conform to the specific pattern for binary counting from 0 to 9. The output signal is exactly one-tenth the frequency of the input signal. Precise decade frequency scaling (dividing) is a very common application of this circuit.

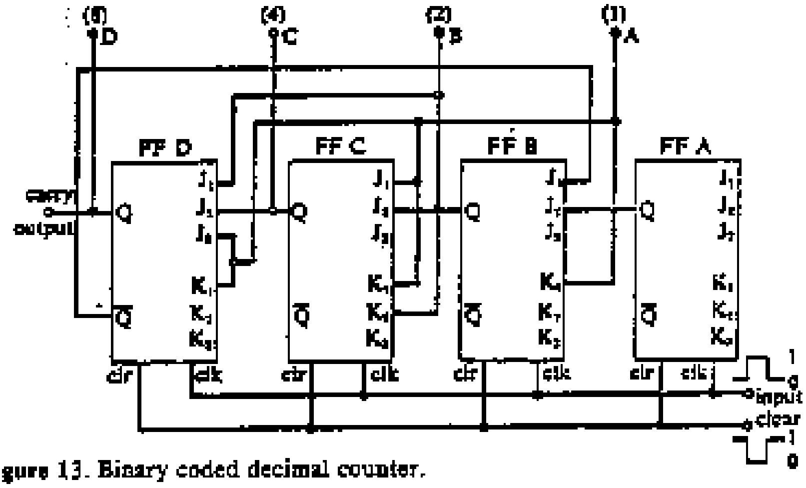


Figure 13. Binary coded decimal counter.

Figure 12

Be sure you understand how the BCD counter clears to all zeros on the tenth pulse (considering there are normally sixteen discrete states for four flip-flops). Also, understand synchronous counting and its advantages.

F. 7490 BCD counter IC

The 7490 BCD counter IC functions very much as the discrete component BCD counter shown in Figure 12. This monolithic counter contains four master-slave flip-flops and additional gates to provide a divide-by-two counter and a three-stage binary counter for which the count cycle is divide-by-three.

1. Set up a symmetrical divide-by-ten (BCD) counter by connecting the Q_o output to the (not C_{P1}) input and applying the input count to the (not Cp_o) input. This configuration provides a divide-by-ten square wave at the output Q_3. Construct it by referring to http://www.national.com/pf/DM/DM5490A.html for the pin configuration and specifications of the 7490. Connect the Q outputs to the ELVIS indicator lights and verify the truth table (count sequence). Use a PULSER and contact bounce eliminator to cycle the counter. Record the frequency at each of the 4 outputs for a 1 kHz square wave input.

2. 7448 BCD-to-Seven Segment Decoder/Driver and HDSP-7803 Seven Segment Light Emitting Diode (LED) Display.

By now you may be tiring of looking at a set of lights and using a binary number representation. It would be nice to have a base 10 numerical display of the BCD value. This is easily done with a 7-segment display and proper decoding as the BCD signal.

In a BCD signal, four bits are encoded to represent each decimal digit. The four-bit BCD group can be decoded into the normal decimal digits by use of a 7448 decoder/driver chip. This chip can also provide enough current to “drive” the light emitting diode (LED) numeric display.

See Appendix 1 for the pin-out diagram, truth table and specifications of the 7448 chip and the data sheet at the TA lab bench for a description of the HDSP-7803 7-segment LED display.

The seven-segment LED display is an array of eight light emitting diodes which are optically magnified to form seven individual elements and a decimal point. Each LED is powered (lit) separately according to the individual output states of the 7448 decoder/driver in order to produce the decimal equivalent to the binary count.

Connect the outputs of the 7448 to the seven-segment display inputs. Apply a "0" logic level to the lamp test pin of the 7448 to insure that all segments are working (it should look like an 8). Referring to the logic diagram for the 7448, explain why a zero logic level applied to the lamp test input causes all seven segments to light. Consulting the 7448 truth table, what is the function of the RBI and BI operations?

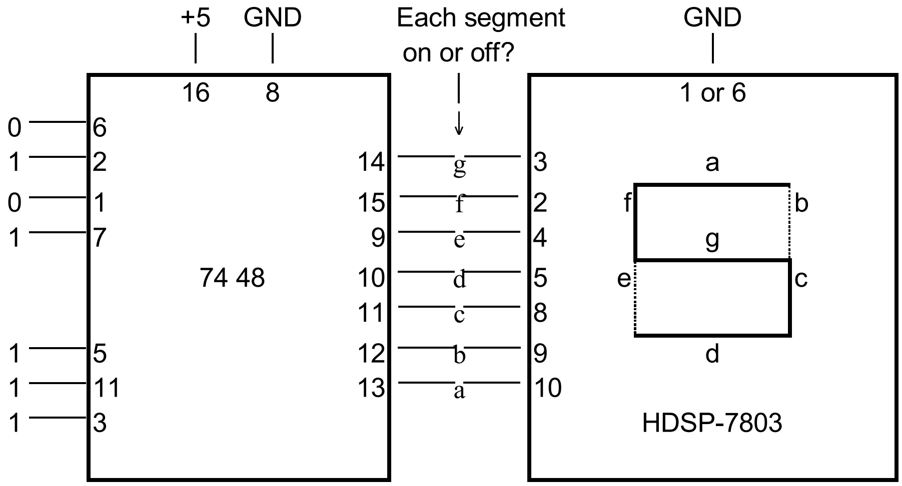
Connect the outputs of the 7490 BCD counter to the inputs of the 7448 decoder/driver. Cycle the counter using a PULSER and contact bounce eliminator. Verify that the display is accurately providing the count.

Appendix 1. 7448 BCD-to-Seven Segment Decoder/Driver

The purpose of the 7448 is to convert BCD (binary coded decimal) information into a form suitable for driving a seven segment LED (light emitting diode) display. BCD information comes in on pins 1,2,6 and 7 and is output to the seven segment display chip on pins 9 through 15.

Features of the 7448.

1. Internal pull-up resistors (eliminates the need for external resisitors).
2. Active high outputs.
3. Maximum current sink rated at 6.4 mA.
4. Generally used for driving a lamp buffer, but in this lab it will be used directly to drive LED segments (with or without pull-up resistors).



Wiring diagram for 7448 connected to HP HDSP-7803 seven segment display. (NOTE: The actual pin location is not as shown on the figure above, see the spec sheet for pin location)

To get brighter LED output, a 1K pull-up resistor can be added to each of the 7 lines (a total of 7 resistors).

Although we are using the 7448 as a driver, it is not generally intended for the purpose. It was built to interact with other TTL divices.

The inputs at A,B,C,D (Pins 7,1,2 and 6, respectively) determine which of the outputs (a,b,c,d,e,f,g) are high or low. Thus a binary number is input to generate the appropriate base ten number on the seven segment display.

There are some pins on the 7448 whose function may not be clear. Many of these will not be used in this lab, but are included here.

Pin 4, BI, Blanking Input. When this input is low all segment outputs (pins 9-15) go low regardless of the state of any other input. Thus all segments are turned off (blanked). (Note that when an output is high on the 7448, it causes the corresponding segment to which it is attached to go on).

This pin is also used as an output (this is an unusual case). When the RBI (ripple blanking input) is at logical 0 and the A,B,C and D input are LO, then pin 5 responds by going low. When this pin is used in this way it is called the RBO, ripple blanking output.

In this experiment simply let pin 4 float high (no connection).

Pin 3, Lamp Test. This is used to check all the segments on the LED. When this pin is held low, all outputs go high so that all of the LED segments go on.

In this experiment let pin 3 float high.

Pin 5, RBI, Ripple Blanking Input. This input is used to blank or turn of zeroes (leading or trailing).

In this experiment let pin 5 float high.