Lab 9

Flip-Flops, Shift Registers, Counters, Decoders, 7-Segment Display

REFERENCE:	Horowitz and Hill	Sections 8.16 -8.26	
		Section 9.10	[LED display]
		Section 9.04	[De-bouncer]

INTRODUCTION

- In this lab we study two versions of the Flip-Flop (bistable multivibrator). Then we test some counters and learn about BCD (binary-coded decimal). Next, we take the BCD output produced by the decade counter and convert it with a decoder-driver to power a 7-segment LED display. After building a counter, we will use it to demonstrate how a switch de-bouncer works. Finally, we try out a BCD-to-decimal converter.
- You will look up a few items in data sheets, to learn how to find information for chips.

EQUIPMENT

Prototyping board
Digital Oscilloscope
Pulse Generator
Digital Multimeter
Digital ICs: [TTL (LS) or CMOS (HC)]

7400	Quad NAND
7442	BCD to Decimal Decoder
7447	BCD to 7-segment decoder driver (common anode)
7490	decade counter
74112	JK flip-flop
74175	D flip-flop
74161	4-Bit Synchronous Binary Counter
74195	4-Bit Shift Register

Capacitors	0.1 μF
Resistors	150 Ω, 10 k (2)
Seven-Segment LED (common anode)	LN513RA
Wires for Prototyping board (35)	[for 7-segment]
Switch SPDT (momentary contact) with wires attached	[for de-bounce]

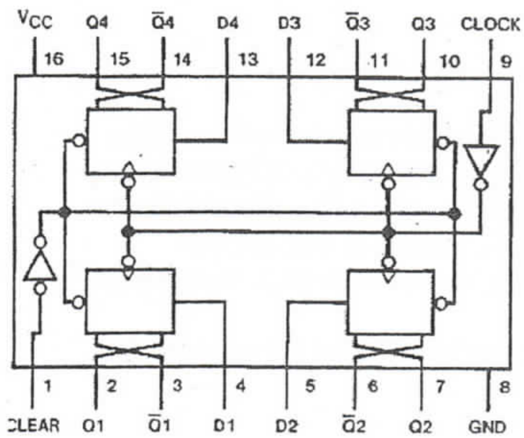
PROCEDURE

1. D Flip-Flop (74HC175 or 74LS175)

   " The D Flip-Flop is the simplest but most important Flip-Flop. It simply saves at its output, Q, what it saw at its input, D, just before the most recent clocking edge.

   The chip used below, the `175, has four flip flops, and they respond to a rising edge on the clock input.

Dual-In-Line Package



TL/F/6557-

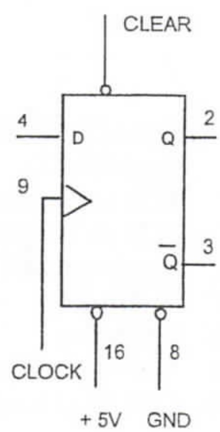
Order Number 54175DMQB, 54175FMQB, DM54175J, DM54175W or DM74175N

Function Table (Each Flip-Flop)

Inputs			Outputs	
Clear	**Clock**	**D**	**Q**	**$\overline{Q}$†**
L	X	X	L	H
H	↑	H	H	L
H	↑	L	L	H
H	L	X	Q_0	$\overline{Q}_0$

H = High Level (steady state)
L = Low Level (steady state)
X = Don't Care
↑ = Transition from low to high level
Q_0 = The level of Q before the indicated steady-state input conditions were established.
† = 175 only

'151 Flip-Flop



(a) Basic operation

☞ Use Vcc = +5 V to power this chip.

☞ Connect CLR (pin 1) to a breadboard switch, set to HI.

☞ Provide data to the D1 input (pin 4) from a prototyping board switch, by setting the D1 input to HIGH .

☞ Clock the flop (pin 9) with a momentary-contact switch (for example, "logic switch A on the prototyping board).

☞ Set up LED indicators of: the input D1 output Q1 (pin 2) and its complement $\overline{Q}_1$ (pin 3).

☞ ✍

(i) Verify that the D flop ignores information presented to its input before a rising edge on the clock, and that it then shifts that value to the output. Draw a timing diagram, showing CLK, D1, and Q1.

(ii) Verify the Truth Table in the specification sheet.

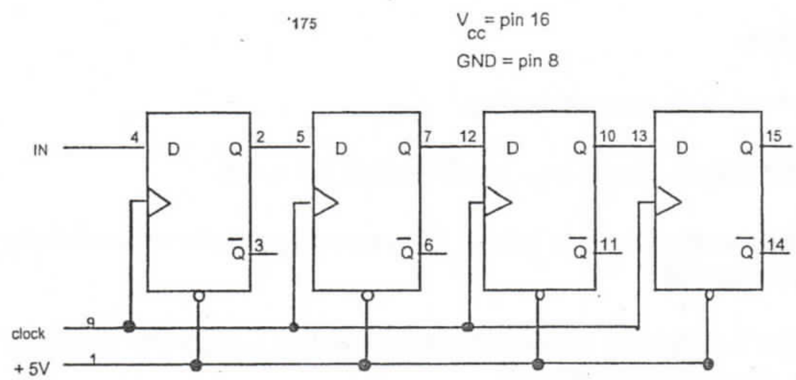
(b) Shift Register

☞ Leaving the first flip-flop set up as it is, apply its output Q1 to the D2 input of another flip flop, and so on, for four flip-flops total, as shown below.

☞ Connect the four outputs Q1, through Q4 to LED indicators.

☞ Set CLR to HI using a switch on the prototyping board.

☞ Use a clock frequency of about 1 to 10 Hz, or use a momentary switch (for example, "Logic Switch A" on the prototyping board) as a manually-operated clock.



shift register

☛ ✍ Watch all the states. Observe that the input is shifted to the right once each clock cycle. Draw a timing diagram showing the four D inputs, the final output, and the clock.

2. J-K Flip-Flop (74LS112)

❝ The J-K Flip-Flop is used in the insides of counter ICs, although you might seldom use a J-K Flip-Flop by itself. The '112 JK flip-flop changes state on the falling clock edge.

☛ Refer to the specification sheet for the pin configuration. Use four switches on your prototyping board to provide inputs to SD, CD, J, K. (The terminology in a data sheet is often like this: SD = "Preset", CD = "Clear"). Use either a momentary switch or the prototyping board's built-in clock to provide the clock signal CP . Use LEDs to observe the outputs Q and $\overline{Q}$.

(a) Truth table

☛ ✍ Verify the truth table shown in the specification sheet.

(b) Triggering

* - Using a switch (for example, a "Data Switch" on your prototyping board) as the clock input, verify that the flop changes state on the falling clock edge.

3. The '161 Synchronous Binary Counter

❝ This is a 4-bit binary counter with data inputs that can be preset, i.e., programmed, so that it begins counting from a desired number. Binary counters contain JK flip-flops.

☞ Connect V_{CC} to pin 16 and GND to pin 8. Connect the inputs of the 74LS 161 to switches on the prototyping board, and the outputs to LED's, as follows:

	INPUTS					OUTPUTS			
binary	2^3	2^2	2^1	2^0		2^3	2^2	2^1	2^0
state	IN_D	IN_C	IN_B	IN_A	LOAD	Q_D	Q_C	Q_B	Q_A
IC pin	6	5	4	3	9	11	12	13	14
board	SW1	SW2	SW3	SW4	SWA	LED1	LED2	LED3	LED4

☞ Connect pin 2 (CLK) to a 1 Hz clock on your prototyping board.

☞ Set the following pins on the 74LS161 to HI (you may use data input switch on your prototyping board to accomplish this).

```
Enable P   (pin 7)
Enable T   (pin 10)
Clear      (pin 1)
```

66 When LOAD' (pin 9) is LO, the counter can be loaded with a binary number on data inputs IN_D, IN_C, IN_B, IN_A. When it is HI, clock signals are counted.

☞ Before using the circuit, convert the decimal number "12" to binary.

☞ Program the counter so that it will count from 0. Do this as follows: With LOAD' LO, input the decimal number "12" on the input switches. Then set LOAD' to HI.

☞ Now clock the CLK input and watch the chip count in binary from 0 to 15. Note that it will start at the value 12 that you programmed, then count up to 15, then start over at 0.

✍ Draw a timing diagram for the counting, showing CLK and outputs Q_A, Q_B, Q_C, Q_D.

66 If you are going to do the switch de-bouncing experiment, you may do it now using this binary counter (in that case, leave this circuit connected). Or, you may do it later with the decade counter.

4. The `195 4-Bit Shift Register

❝ The `195 Shift Register is like the shift register using four D flip-flops, except that it has more features. It accepts both series and parallel inputs and produces both series and parallel outputs. Thus, the `195 can be used to convert series input to parallel output and *vice versa*.

❝ When LOAD is LO, the outputs are set to the inputs, on a rising edge of CLK. When LOAD is HI, the register shifts.

'195 Truth table

INPUTS at t_N		OUTPUTS at t_{N+1}			
J	K'	Q_A	Q_B	Q_C	Q_D
L	L	L	Q_{AN}	Q_{BN}	Q_{CN}
H	H	H	Q_{AN}	Q_{BN}	Q_{CN}
L	H	Q_{AN}	Q_{AN}	Q_{BN}	Q_{CN}
H	L	Q_{AN}'	Q_{AN}	Q_{BN}	Q_{CN}

☛ Connect the 74LS195 to switches (for input) and LEDs (to indicate output):

INPUTS						OUTPUTS			
binary	2^3	2^2	2^1	2^0		2^3	2^2	2^1	2^0
state	IN_D	IN_C	IN_B	IN_A	CLK	Q_D	Q_C	Q_B	Q_A
IC pin	7	6	5	4	10	12	13	14	15
board	SW1	SW2	SW3	SW4	SWA	LED1	LED2	LED3	LED4

☛ Set CLR (pin 1) to HI.

(a) Parallel data loading and serial shift

☛ Set LOAD to LO to allow parallel loading. Input the number "1010". Clock the chip by pressing SWA. Confirm the number shows up on the LED indicators.

(i) serial shifting
☛ Now set LOAD to HI to allow serial shifting. Clock the register by pressing SWA to advance the data (you will need to do this to move the data forward).

☛ ✍ Examine Q_B, Q_C, and Q_D on the LED indicators to verify that the register shifts data serially in the direction A to D, as indicated in the truth table.

☛ ✍ Fill in the timing diagram. Repeat for the four combinations of J and K'.

☛ ✍ Examine the entries in the timing diagram and verify that they obey the truth table.

(ii) truth table

☛ ✍ Verify the part of the truth table that indicates that, upon a shift, state A is filled with either a LO, a HI, the previous value, or the complement of the previous value, depending on the values of the J and K' input

(b) Serial to Parallel conversion S → P

☛ ✍ Observe Q_A to verify that when you present a datum to both the J and K' inputs (first two lines of truth table) that the number is sent to the output Q_A on the clock.

❝ Note that this allows 4 serial bits (entered on four clocks) to be readied for parallel output.

(c) Parallel to Serial conversion P → S

☛ ✍ Examining the data in your diagram, verify that the initial input data is converted into serial data where the output QD shows the input data as the clock is advanced.

Diagram for '195 shift register

J=LO	K'=LO					
CLK						
Load						
Outputs at T1		Outputs at T2		Outputs at T3		Outputs at T4
QA						
QB						
QC						
QD						
J=HI	K'=HI					
CLK						
Load						
Outputs at T1						
QA						
QB						
QC						
QD						
J=LO	K'=LO					
CLK						
Load						
Outputs at T1						
QA						
QB						
QC						
QD						
J=HI	K'=LO					
CLK						
Load						
Outputs at T1						
QA						
QB						
QC						
QD						

5. Decade Counter (7490)

" Referring to its data sheet we see that the `90 has four internal Flip-Flops. The first one (A) is not internally connected to the others while B, C, and D are cascaded to form a divide-by-5 synchronous counter.

☛ Connect the `90 as follows:

1.	÷5, ÷ 10 input	8.	Q_C
2.	GND	9.	Q_B
3.	GND	10.	GND
4.	NC	11.	Q_D, ÷ 5 output
5.	+5 V	12.	Q_A, ÷ 2 output
6.	GND	13.	NC
7.	GND	14.	÷ 2 input

☛ For an input in parts (a) to (c), use the pulse generator to produce square pulses, between 0 and 5 V, at 5 kHz.

☛ Set up a scope for dual-trace operation to show the pulse input from the pulse generator and an output (either pin 11 or 12 depending on what you are measuring) of the `90. Trigger the scope internally, using as the trigger source the scope channel CH1 or CH2 that you have connected to the counter's output.

☛ ✍ Use the `90 to perform the following operations.

(a) Divide by 2

☛ Connect the pulse input to the ÷2 input.

☛ ✍ Print the waveforms. Explain how the waveform is "divided by two."

(b) Divide by 5

☛ Connect the pulse input to the ÷5 input.

☛ ✍ Print the waveforms. Explain how the waveform is "divided by five." Note the asymmetry of the output waveform.

(c) Divide by 10

☛ Connect the ÷5 output to the ÷2 input. Use the scope to observe the ÷2 output.

☛ ✍ Print the waveform. Explain how the waveform is "divided by five."

(d) <u>BCD Counting</u>

❝ Binary coded decimal (BCD) is the same as binary for numbers 0 to 9. For larger numbers it is different: BCD is composed of groups of four binary digits, where each group represents one digit in base 10.

Example

314 =	decimal	`3    1    4`
	BCD	`0011 0001 0100`
	binary	`100111010`

For this chip, outputs Q_A, Q_B, Q_C, and Q_D are the four-bit BCD outputs.

☞ Disconnect the pulse generator.

☞ Use a switch on your prototyping board (SWA) to apply a pulse input on pin 14.

☞ Connect Q_A (pin 12) to the ÷ 5 input (pin 1) for BCD counting. (It may already be connected this way from part c above.)

☞ By means of LED indicator lights (you may use those built into the prototyping board), test the levels of the D, C, B, and A outputs (pins 11, 8, 9, 12 respectively) after each pulse.

(i) counting

☞ ✍ Write a copy of Table 9-1, and check off each state 0 through 9 to confirm that the outputs D C B and A of the counter accumulate in a BCD counting cycle. (Note: the output may at first indicate an invalid number, such as 1011,when you first turn on the power. If this happens, just apply several pulse inputs until a new cycle begins.)

(ii) resetting

☞ ✍ Verify that setting the reset inputs R_0 (pin 6) and R_1 (pin 7) to HI will zero the outputs at D, C, B, and A of the decade counter.

☞ Leave this circuit wired up for use with the 7-segment display in the next part.

Table 9-1

decimal	BCD
	D C B A
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1
0	0 0 0 0
	etc.

6. Decoder-Driver and 7-Segment LED Display (7447)

[Forewarning: this circuit is not hard, but it requires hooking up more wires than most; thus it might take more time.]

❝ Displays are popular output devices. LED's are brighter, while LCD's consume far less power. Both types of displays require a decoder-driver to convert (decode) from BCD input and to drive the display.

❝ LED displays have 8 leads for the 7 LED's. The displays and their corresponding driver IC's come in two varieties, common anode and common cathode.

❝ Ordinarily, when using LED's, one uses a series resistor of about 150 Ω to limit the current.

☛ Leave the BCD counter hooked up as it was at the end of part 4, above, with a switch for the input and the 4 BCD outputs indicated by 4 LED's.

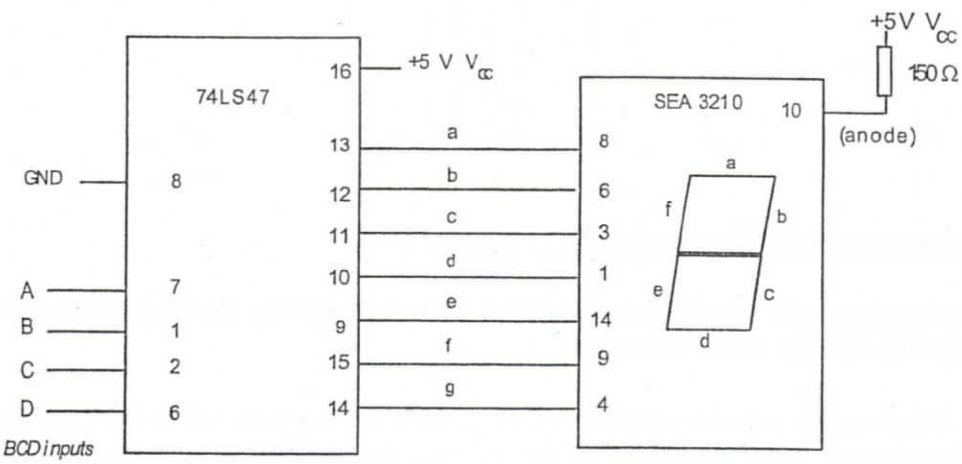
☛ Connect the BCD outputs of the BCD counter to the BCD inputs of the 7447. Connect the 7447 to the 7-segment LED display as shown. [Note: pin configurations for LED displays is not standardized. Pin configurations are shown here for the SENIOR SEA3210 display. (Your display might have a different model number but the same pins.) This display has an additional LED for a decimal, which we will not use.]

(a) Counting

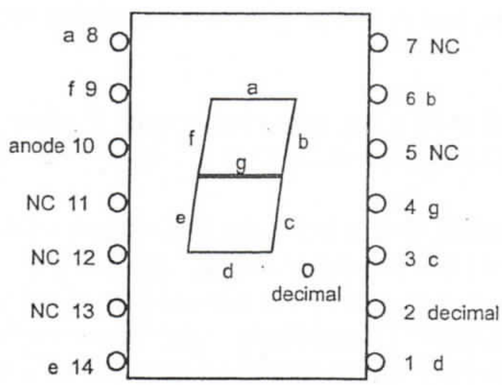
☛ ✍ Write a copy of Table 9-1 and check off each state 0 through 9 to confirm that the LED outputs D C B and A of the counter are properly indicated by the LED display.

(b) Current consumption

☞ ✍ Use your multimeter in the dc-current-mode to measure the current from the common anode to the + 5 V V_{CC}. How much current is consumed by the LED display when the display reads "1" and when it reads"8"? [Note that LED displays are power-hungry devices.]



LN513RA
7-segment common-anode LED
(top-view)

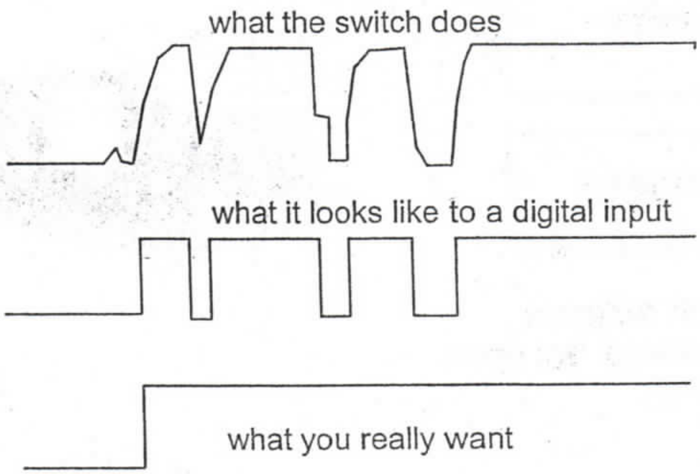


note: this LED display does not actually have 14 leads, but the package is shaped like a 14-pin DIP, with some leads missing. Missing leads are indicated here as NC

7. Latch Used for Switch De-Bouncing

❝ Note to instructor: this circuit must be used with one of the counter circuits above. It is possible to use either the binary counter or the decade counter.

❝ An ordinary mechanical switch "bounces" several times when it opens or closes. This means that it opens and closes several times, on a time scale of about 100 μ s, before reaching its final state. When using a switch to input data to a logic circuit, bounce is unfortunate because it looks like real data that is changing several times.



(a) Bouncy switch

☛ Use a momentary-contact switch (not the switch built into your prototyping board – those are already de-bounced). Use the multimeter to check continuity, to see which two terminals to use on the switch so that it is normally open, and closed only when you push.

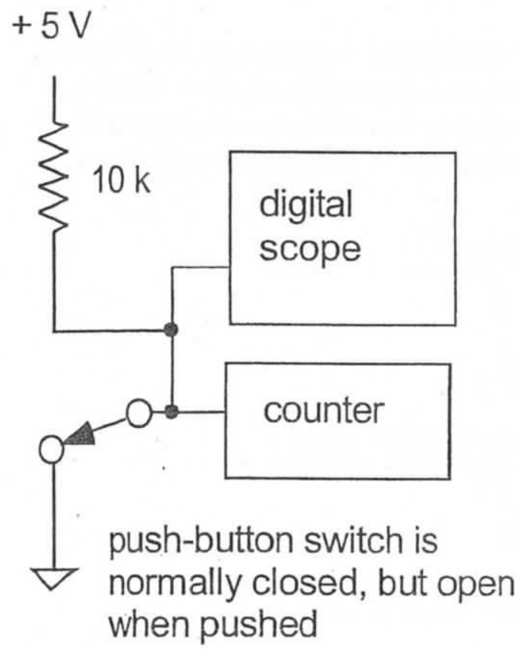
Connect the simple input data switch shown below, so that the switch is normally connected to LO.

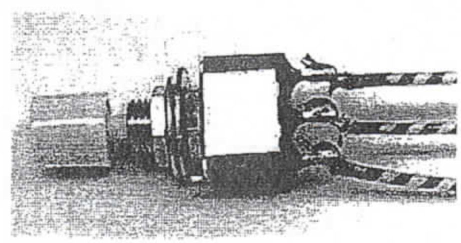
☛ Adjust a digital oscilloscope with DC coupling and approximately 100 usec / div. Observe the waveform of the output of the switch, adjusting the trigger and timebase so that you can see any bounce, as in the photo below.

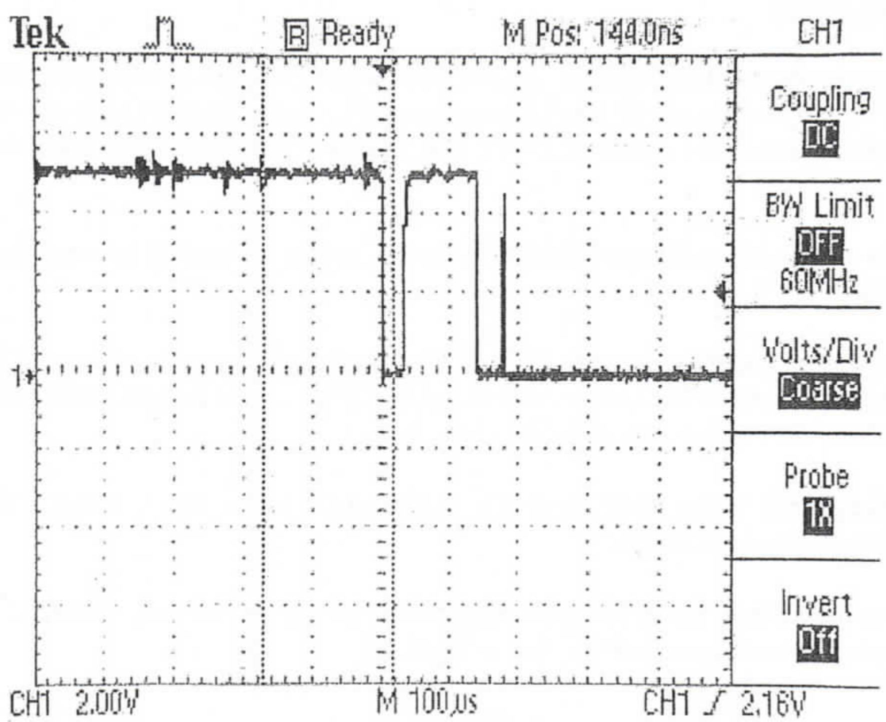
☛ Apply the output to the clock input of a counter circuit (you may use any of the counter circuits you built above.

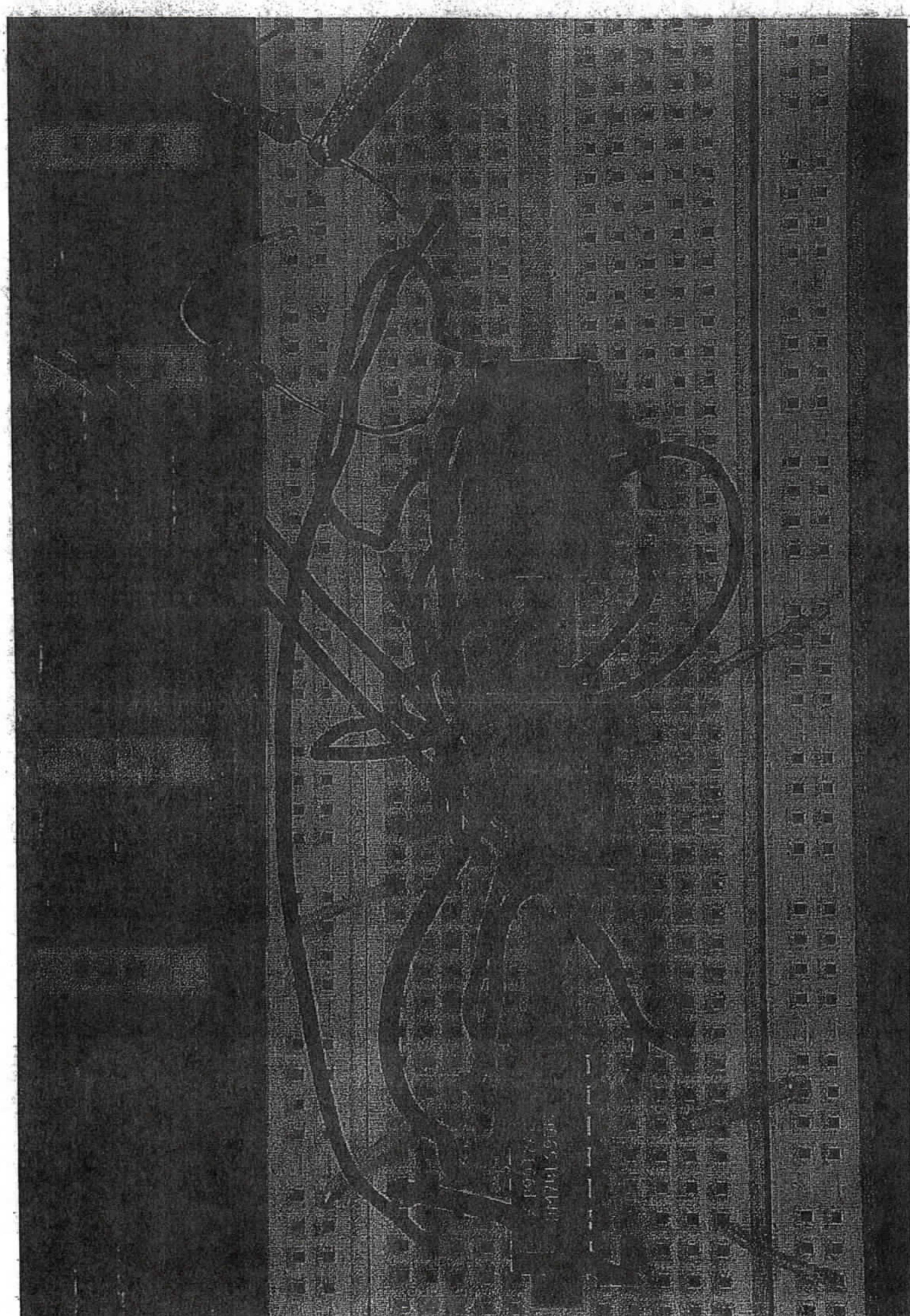
☛ ✍ Press the switch, watch an LED indicator of the counter output. Does the counter count more than one clock, due to bounce?

☛ ✍ Observe and print the waveform produced by the switch. Use a digital storage oscillocope.

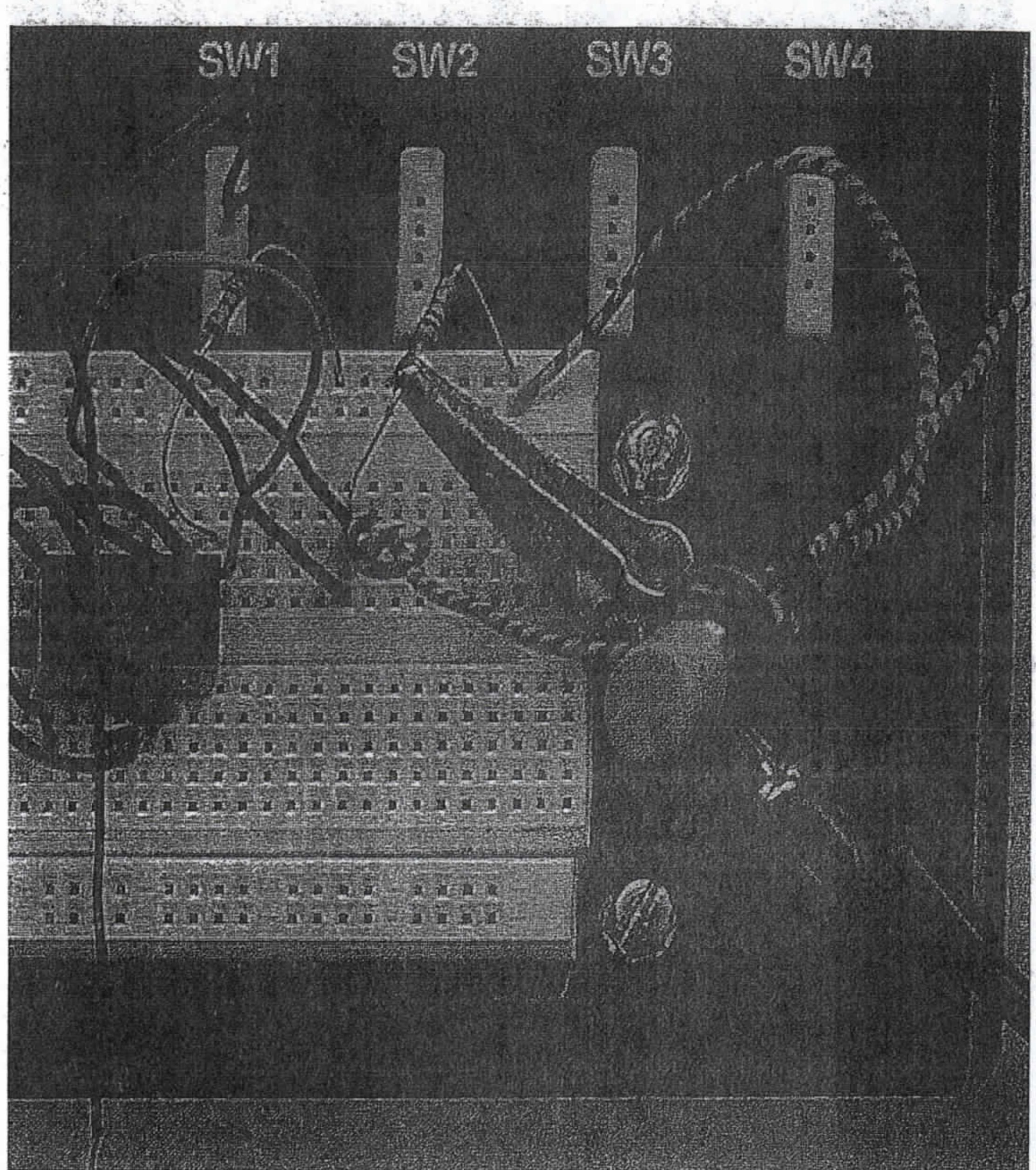








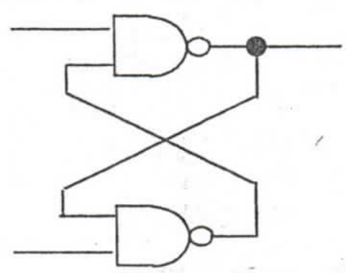
Hookup for debouncing test. – photo 1



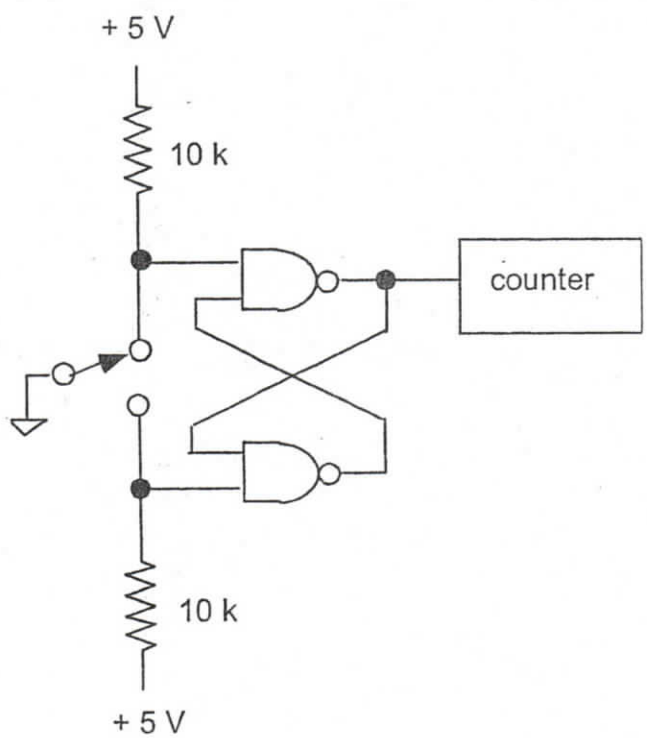
Hookup for debouncing test. – photo 2

(b) <u>De-bounced switch</u>

☛ The circuit shown below is called an "RS flip flop" or a "cross-coupled NAND latch". Wire it up using a 7400 quad NAND gate chip.



Now add a switch with pull-up resistors to make a data input, as shown below. Press the switch, and observe that the counter counts only one clock, because the switch is now de-bounced.



8. The '042 BCD to Decimal Decoder Chip

☛ Before wiring the circuit, convert the following decimal numbers to BCD:
`0, 1, 8.`

☛ Wire the decoder chip with four inputs for BCD input. Use the switches on your prototyping board to provide input, as shown below:

BCD INPUTS

binary	2^3	2^2	2^1	2^0
state	D	C	B	A
IC pin	12	13	14	15
board	SW1	SW2	SW3	SW4

☛ ✍ Apply the BCD input corresponding to "0" . Touch a wire connected to an LED indicator to test the states of the 10 output pins. Verify that the output pin corresponding to "0" is LO and the other outputs are HI. Repeat with data inputs "1" and "8".