ADVANCED CERTIFICATE IN APPLIED INDUSTRIAL ELECTRONICS

YEAR 1 DIGITAL PRINCIPLES

THEORY LESSON 4

Written by Peter Phillips, 1990

BINARY NUMBERS - BINARY ADDITION

OBJECTIVES: At the end of this lesson you should be able to:

(a) *Convert a decimal number (0 to 127) to its binary equivalent.*
(b) *Convert an 8 bit (max) binary number to its decimal equivalent.*
(c) *Perform addition with four bit binary numbers.*
(d) *Draw the circuit of a binary half adder.*
(e) *Show how two half adders can be connected to form a full adder.*
(f) *Show how four full adders can be interconnected to add 2, four bit binary numbers.*

1. INTRODUCTION

The decimal number system is based on a total of 10 characters, 0 through to 9. Other numbering systems that use any number of characters are just as valid as the decimal numbering system, and find application in digital circuits. The three most common systems used are:

(a) The **BINARY** system. Contains 2 characters, 0 and 1. Used in logic circuits, and forms the basis for any form of mathematics when considering a digital logic circuit.

(b) The **OCTAL** system. Contains 8 characters, 0 to 7. Used by computers, and microprocessor based systems.

(c) The **HEXADECIMAL** system. Contains 16 characters, 0 to 9 and A to F. Used in the same applications as the octal system.

In this lesson the binary system is examined, along with techniques in adding binary numbers. As well, the basic logic circuits that perform binary addition are discussed. Following lessons will examine the Hexadecimal and Octal numbering systems as well as subtraction, division and multiplication of binary numbers.

2. THE BINARY SYSTEM

Because digital circuits operate at two levels, the binary system is the only possible number system that can be applied directly to a digital logic circuit. Conversion between binary and decimal numbers is therefore essential to be able to read numerical data within a logic circuit. Because there are only two characters in the binary system, each numerical value must be expressed using a series of 1's and 0's. When a number is written in the decimal system 10 characters are available, and

the characters for numbers greater than 9 are written in a particular order, in which the position of the character within the number identifies its "weighting". Consider the decimal number 4628. In effect, this number is the sum of:

4×10^3, 6×10^2, 2×10^1, and 8×10^0.

That is, 4000 + 600 + 20 + 8, which equals 4628

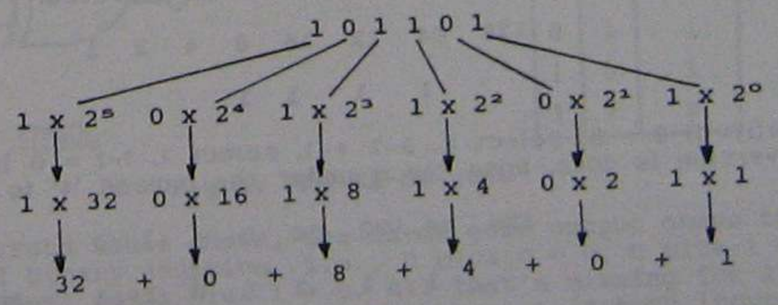
** (note that any number raised to the power of 0 = 1)

This example demonstrates the *weighting* of each digit in that its effect is determined by its *position* within the number. The left-hand digit is the most significant, and the right-hand digit is the least. In a decimal number, each digit is multiplied by a power of 10, because the system is based on the number 10. Thus, "10" is the base, or radix, of the decimal system. In the binary system, the radix is "2", and each digit has a value within the complete number equal to its numerical value (either 0 or 1) multiplied by its weighting. The important values to know are the various powers of two, the first eight of which are listed below.

$2^0 = 1$
$2^1 = 2$
$2^2 = 4$
$2^3 = 8$
$2^4 = 16$
$2^5 = 32$
$2^6 = 64$
$2^7 = 128$

EXAMPLE: BINARY TO DECIMAL
Convert the binary number 101101 to its decimal equivalent.

SOLUTION
(a) Write the binary number as shown, and for each '1' in the number, write the value it represents. Then sum these numbers for the answer.



which equals 45_{10}

Conversion from a decimal value to its binary equivalent can be done in various ways, including successive division/multiplication by 2, a calculator or with the following very simple method:

EXAMPLE: DECIMAL TO BINARY
Convert the decimal number 123 to its binary equivalent.

SOLUTION:
(a) Write the powers of 2 across the top of the page:

128 64 32 16 8 4 2 1

(b) Write a 1 beneath the highest value from this list that when subtracted from the decimal number gives a positive answer. In this example, choose 64 from the list as 128 is too high:

128 64 32 16 8 4 2 1

1

(c) Subtract this value (64) from the decimal number:

123 - 64 = 59

(d) Repeat step (b) using the result of the previous subtraction as the decimal number. Write a 0 beneath unused numbers.

128 64 32 16 8 4 2 1

1 1

(e) Subtract selected value from result of previous subtraction:

59 - 32 = 27

(f) Repeat steps (d) and (e) until the conversion is complete: Thus, next power of two selection is 16: (next less than 27).

128 64 32 16 8 4 2 1

1 1 1

Subtracting: 27 - 16 = 11, allowing 8 to be selected:

128 64 32 16 8 4 2 1

1 1 1 1

Finally: 11-8 = 3, select 2. 3-2 = 1, select 1. 1-1 = 0 indicating conversion is done. Note the 0 under the unused '4' in the list.

128 64 32 16 8 4 2 1

1 1 1 1 0 1 1 = **Answer**

This method is best done by entering the number to be converted on a calculator and keeping a running total as each subtraction is performed. Note that an odd decimal number will always have a 1 as the least significant bit (LSB) in the binary value, and an even decimal number will have 0 as the LSB in the binary value.

The highest decimal number that can be expressed by N binary digits (bits) equals $2^N - 1$. The range of numbers that can be expressed by N bits equals 2^N. *(Teacher note: give more examples of the above).*

3. BINARY ADDITION

Binary numbers are added in the same way as decimal numbers, using the following rules:

```
    0 + 0 = 0
    0 + 1 = 1
    1 + 0 = 1
    1 + 1 = 0 with a carry of 1
1 + 1 + 1 = 1 with a carry of 1
```

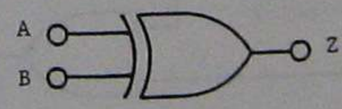
EXAMPLE:

```
  1 1 1 1
  1 0 1 1 +
  \ \ \ \
1 1 0 1 0
```

Analysis: The 1st column produced a carry and a sum of 0. The 2nd column has the sum of 1+1+ carry from column 1, giving a total of 1 and a carry of 1. (Note: 1+1+1 = 3 which in binary equals 11). The 3rd column has the sum of 1 (carry from column 2) +1+0 giving a sum of 0 and a carry of 1. The 4th column has a sum of 1 (carry from column 3) +1+1, giving a sum of 1 and a carry of 1.

4. ADDER CIRCUITS

Binary addition is relatively easy to accomplish. The basic gate used in binary addition is the EX-OR gate and the symbol and truth table are shown in Fig.1 below.



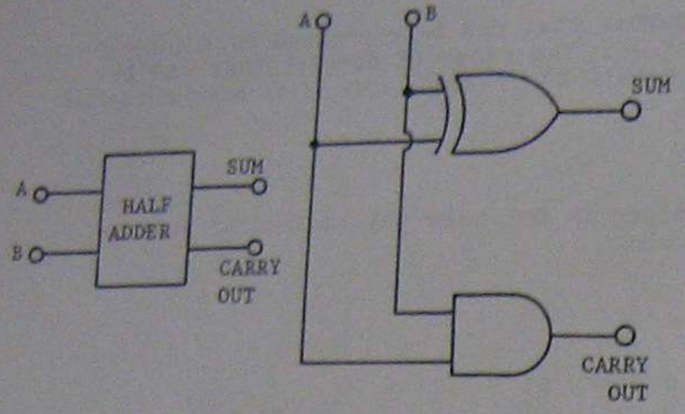
SYMBOL

TRUTH TABLE

A	B	A ⊕ B
0	0	0
0	1	1
1	0	1
1	1	0

Fig.1: *The EX-OR gate*

As the truth table shows, the EX-OR gate output obeys the rules of binary addition, i.e., 0 plus 0 = 0, 0 plus 1 = 1, 1 plus 0 = 1 and 1 plus 1 = 0. All that's missing for 1 plus 1 is the carry. The gate that outputs a 1 if both inputs are 1 is the AND gate. Thus, by connecting an AND gate in parallel with the EX-OR gate, a circuit capable of adding two binary bits can be realised. This circuit is called a half adder, shown along with its truth table in Fig.2.



TRUTH TABLE

A	B	SUM	CARRY
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Fig.2: The Half Adder

The half adder is only able to add two binary bits and doesn't allow for a carry in from a previous addition. By combining two half adders and ORing the Carry Out terminals from both, the *Full Adder* is constructed. The sum out of one half adder feeds one input of the other half adder, giving a total of three inputs as shown in Fig.3. The truth table for the Full Adder is shown in Fig.4.

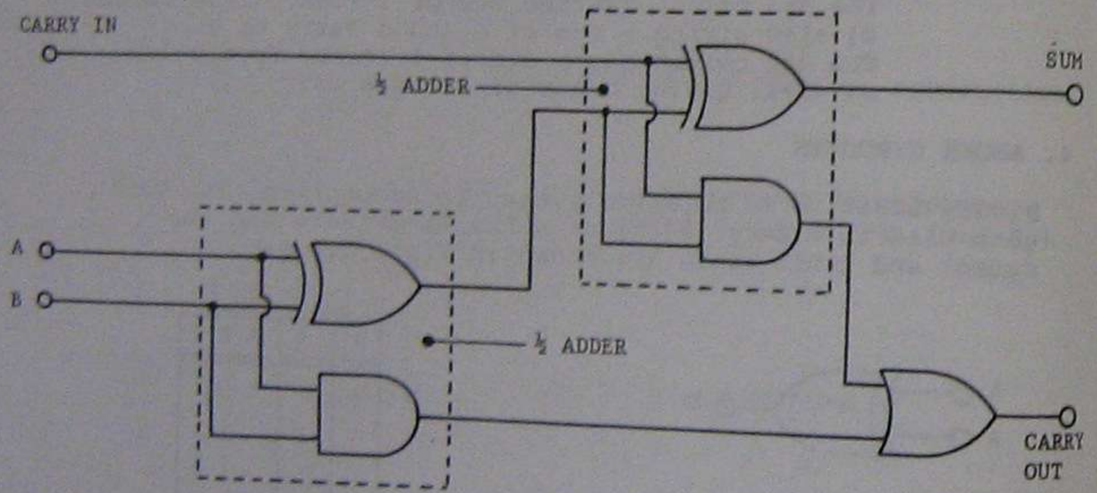


Fig.3(a): *The Full Adder*

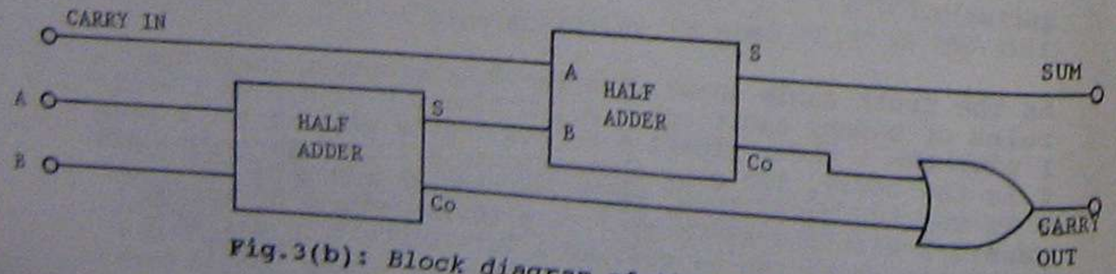


Fig.3(b): *Block diagram of the Full Adder*

INPUTS			OUTPUTS	
A	B	Cin	SUM	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Truth Table for the Full Adder

Combining full adders to allow the addition of two, 4 bit binary numbers requires four full adders to be cascaded, as shown in Fig.5. The first Carry In terminal is set to a logic 0 by grounding it, and each Carry Out terminal connects to the next full adder's Carry In terminal. The 74LS83 is a four bit full adder IC used in binary addition.

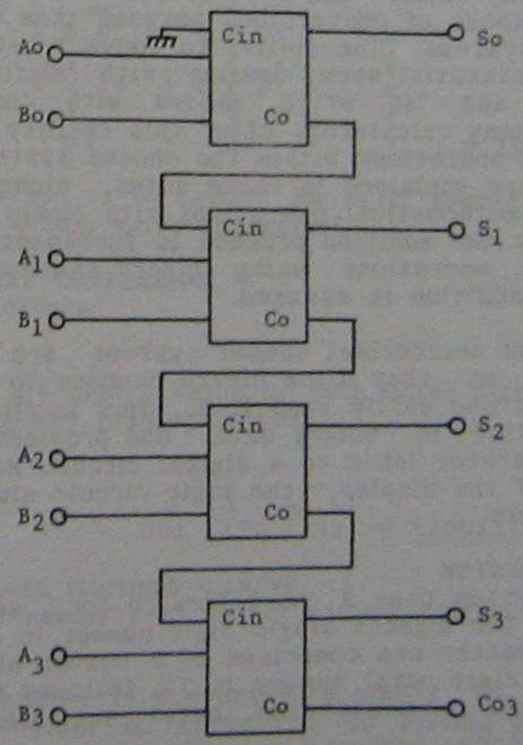


Fig.5: *Cascading 4 full adders to add two, 4 bit binary numbers*

output lights a segment in the

d measure the voltage drop
minals that are a logic 0 (which
s are passing the current for

nal 'b': _____ Volts

nal 'c': _____ Volts

segments of the LED display.
e previous step divided by the

_____ mA

_____ mA

' and 'c' of the LED display. To
pply to terminal 'b' and then to

display: _____ Volts

display: _____ Volts

voltage drop across a forward-

_____

istor for a LED display operating

_____

Section 8: Digital/Analog Conversion

SUGGESTED DURATION	PREAMBLE
4 hours	This section introduces students to Analog to Digital Conversion (ADC) and Digital to Analog Conversion (DAC) using commercial integrated circuits.
This section covers learning outcome 1 of the National Module Descriptor.	

Objectives

At the end of this section, successful students will be able to:

- ☐ Sketch a block diagram of ADC and DAC circuits.
- ☐ Give examples of where ADC and DAC circuits can be found.
- ☐ Calculate the resolution of ADC and DAC circuits.
- ☐ Calculate input and output signal levels of ADC and DAC circuits.
- ☐ Explain the relevance of ADC conversion time.

Review Questions

1. Briefly describe an application of a DAC and an ADC.

_____

_____

_____

_____

2. Calculate the percent resolution of a 12-bit DAC.

_____

3. Calculate the full-scale output voltage of a single-polarity, 10-bit DAC that has a step size of 5mV.

_____

4. A single-polarity, 8-bit DAC has a full scale output of 5V. Calculate the output voltage if the input is set to 1Eh.

_____

5. A single-polarity, 8-bit ADC produces its highest output when 5.1V is applied to the input. Determine its resolution and calculate the maximum change of input voltage to cause the output to change.

_____

6. Explain 'ADC conversion time'.

_____

_____

Skill practice 8: Digital/Analog

Aim

In this exercise you will use commercial A
Digital to Analog Converters (DAC) to co
socket.

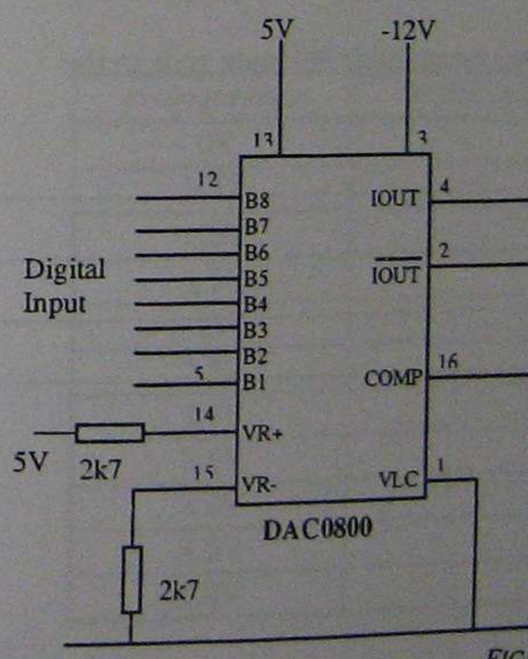
Equipment

- IC types: DAC0800, ADC0802
- resistors: 10k , 3 x 2k7
- 10k potentiometer
- capacitors: 150 pF, 10 nF
- digital bread-boarding system
- logic probe
- dual-trace oscilloscope
- voltmeter
- IC extractor.

Procedure

Part 1: Digital to analog conversion

- Connect the digital to analog conv



FIG

multi-digit display.

Section 8

1. An ADC would be found in a DVM. A DAC could be used in a CD player.

2. 0.0244 %

3. 5.12V

4. 0.588V

5. Resolution = $5.1V/(2^8-1) = 20$ mV
   A maximum of 20mV change is required to cause the output to change.

6. Conversion time is the time required by the ADC to convert the analog input signal to the digital output signal.

inimum change in input voltage that is required to

_____

erated for an analog input 2.25 V.

.25V = _____

for the ADC by connecting the 'start conversion'
and to one channel of the CRO; and the
TR) to the other channel of the CRO.

_____

9

Section Encoders and decoders

Purpose

To introduce the commonly used binary codes such as ASCII, BCD and Gray and to explain the operation of binary encoders and decoders.

Objectives

At the end of this section you should be able to:

- ☐ Convert values between decimal and BCD.
- ☐ Give an application of the BCD code.
- ☐ Describe the ASCII code and give an application.
- ☐ Find the ASCII code for an alphanumeric character using an ASCII table.
- ☐ Describe the format and at least one application of the Gray code.
- ☐ Sketch a general diagram to show the function of a decoder.
- ☐ Sketch the basic circuit of a 1 of 4 decoder (with AND gates and inverters) and explain its operation.
- ☐ Describe the operation and complete the truth table of a
  - 3-line to 8-line decoder
  - 4-line to 16-line decoder
  - BCD to decimal decoder
- ☐ Sketch a general diagram to show the function of an encoder.
- ☐ Describe the operation and complete the truth table of
  - an 8-line to 3-line encoder
  - a decimal to BCD encoder
- ☐ Explain the term *priority* as applied to encoders.

Review questions for Section 9

1. Convert 479_{10} to BCD.

2. Convert the BCD value 1001 0101 0111 0010 to decimal.

3. Convert the BCD value 1001 0101 to binary.

4. What's a typical use of the BCD code?

5. Where is the ASCII code mainly used?

6. What is it about the Gray code that makes it different from other codes?

7. What's a typical use for the Gray code?

8. What's the difference between an encoder and a decoder?

9. The 3-line to 8-line decoder shown in Figure 1 has active low outputs. Complete the truth table for the decoder.

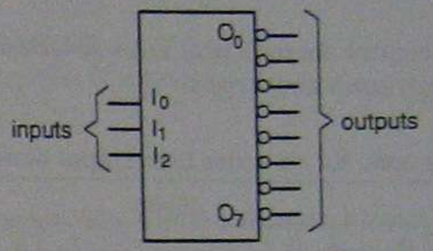


Figure 1

truth table for the decoder of Figure 1

inputs			outputs							
I_2	I_1	I_0	O_0	O_1	O_2	O_3	O_4	O_5	O_6	O_7
0	0	0								
0	0	1								
0	1	0								
0	1	1								
1	0	0								
1	0	1								
1	1	0								
1	1	1								

10. How many inputs does a 1-of-16 decoder have?

11. Sketch a logic symbol of a BCD to decimal decoder.

12. What's the difference between a priority encoder and an ordinary encoder?

_____

_____

_____

13. Complete the truth table below, which is for a decimal-to-BCD encoder with active high inputs and outputs.

truth table for an active high output decimal-to-BCD encoder

inputs										outputs			
I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	O_3	O_2	O_1	O_0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0				
0	1	0	0	0	0	0	0	0	0				
0	0	1	0	0	0	0	0	0	0				
0	0	0	1	0	0	0	0	0	0				
0	0	0	0	1	0	0	0	0	0				
0	0	0	0	0	1	0	0	0	0				
0	0	0	0	0	0	1	0	0	0				
0	0	0	0	0	0	0	1	0	0				
0	0	0	0	0	0	0	0	1	0				
0	0	0	0	0	0	0	0	0	1				

14. Sketch a general diagram of an octal to binary (8-line to 3-line) encoder.

15. What's the purpose of the ENABLE inputs on a decoder or an encoder IC?

_____

_____

Skill practice for Section 9

Title **Encoders and decoders**

Task In this exercise you build a number of encoder and decoder circuits and analyse their operation by completing truth tables for each one.

Equipment IC types: 74LS00 x 2, 74LS04, '08 and '32
digital breadboarding system, logic probe, IC extractor

Part 1 4-input encoder and 1-of-4 decoder

The encoder circuit in this part accepts a decimal input (from 0 to 3) and encodes it to a 2-bit binary code (from 00 to 11). Input 0 has no effect as the code will be 00 when all other inputs are 0. It is shown on the circuit to illustrate the fourth condition. IC encoders overcome this anomaly with an output called *group select* (or GS output). This output becomes active if any input is active. Therefore, you can tell if an output code of 00 is the result of input 0 being active, rather than the result of *no* inputs being active.

1. Connect the circuit of Figure 1. Notice that pin 14 connects to the +5V supply and ground connects to pin 7 of the IC. Connect logic level switches to the inputs and LED indicators to the outputs. Then complete truth table 1 by applying each of the four input combinations.

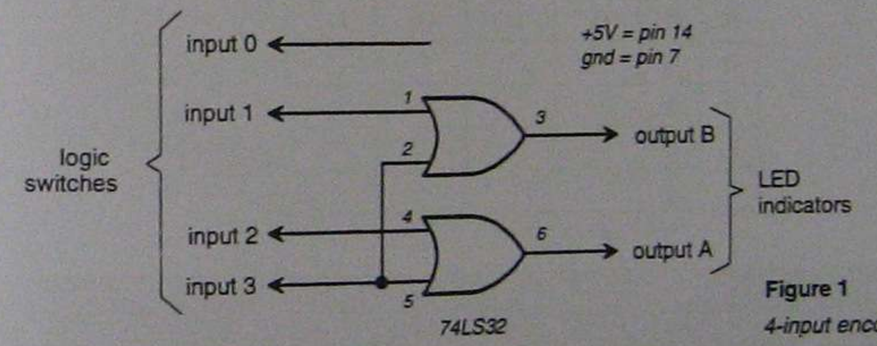
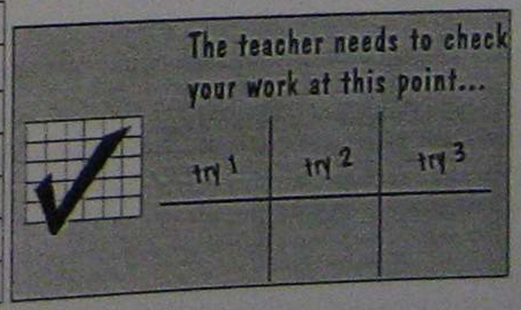


Figure 1
4-input encoder

truth table for 4-input encoder

inputs				outputs	
I_0	I_1	I_2	I_3	A	B
0	0	0	0		
1	0	0	0		
0	1	0	0		
0	0	1	0		
0	0	0	1		



Effect

CD compared to a LED display are
onsumption
n bright conditions
nds itself to having more complex characters

ED display compared to an LCD include
sier to read in dark light or from a distance
se
multiplexing

g in terminal allows leading or trailing zero suppression in a

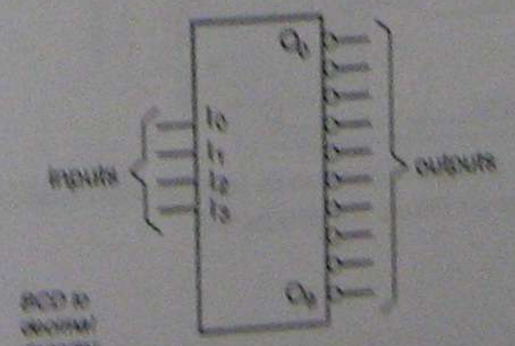
e found in a DVM. A DAC could be used in a CD player.

$V(2^8-1) = 20$ mV
20mV change is required to cause the output to change.

e is the time required by the ADC to convert the analog input
ital output signal.

1. 0100 0111 1001
2. 9572
3. $= 95_{10}$ which is 1011111 in binary
4. BCD is used in a digital circuit when decimal information is either applied as inputs or displayed as outputs.
5. The ASCII code is used to represent in binary form the alphanumeric and control characters of an electronic keyboard.
6. Only one bit changes between adjacent codes.
7. As a digital code in a control system where the position of something (a shaft, water level) is given as a digital signal.
8. An encoder is a device that has a number of inputs and outputs, but produces a specific code at the output when only one input at a time is asserted. A decoder is a device that has a number of inputs and outputs, but which activates only one output for a given input code.
9. **truth table for a 3-line to 8-line decoder**

inputs			outputs							
I_2	I_1	I_0	O_0	O_1	O_2	O_3	O_4	O_5	O_6	O_7
0	0	0	0	1	1	1	1	1	1	1
0	0	1	1	0	1	1	1	1	1	1
0	1	0	1	1	0	1	1	1	1	1
0	1	1	1	1	1	0	1	1	1	1
1	0	0	1	1	1	1	0	1	1	1
1	0	1	1	1	1	1	1	0	1	1
1	1	0	1	1	1	1	1	1	0	1
1	1	1	1	1	1	1	1	1	1	0

10. four
11. 

Effect

CD compared to a LED display are
onsumption
n bright conditions
nds itself to having more complex characters

ED display compared to an LCD include
ier to read in dark light or from a distance
se
multiplexing

in terminal allows leading or trailing zero suppression in a

found in a DVM. A DAC could be used in a CD player.

$(2^8-1) = 20$ mV
mV change is required to cause the output to change.

the time required by the ADC to convert the analog input
output signal.

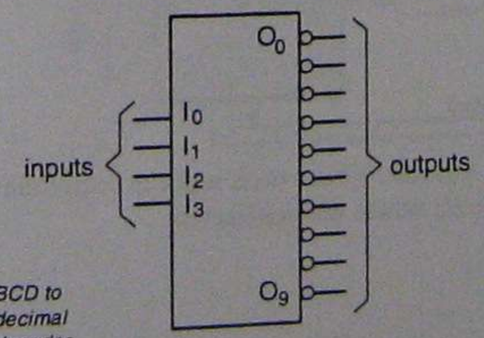
Section 9

1. 0100 0111 1001
2. 9572
3. $= 95_{10}$ which is 1011111 in binary
4. BCD is used in a digital circuit when decimal information is either applied as inputs or displayed as outputs.
5. The ASCII code is used to represent in binary form the alphanumeric and control characters of an electronic keyboard.
6. Only one bit changes between adjacent codes.
7. As a digital code in a control system where the position of something (a shaft, water level) is given as a digital signal.
8. An encoder is a device that has a number of inputs and outputs, but produces a specific code at the output when only one input at a time is asserted. A decoder is a device that has a number of inputs and outputs, but which activates only one output for a given input code.
9.

truth table for a 3-line to 8-line decoder

inputs			outputs							
I_2	I_1	I_0	O_0	O_1	O_2	O_3	O_4	O_5	O_6	O_7
0	0	0	*0*	*1*	*1*	*1*	*1*	*1*	*1*	*1*
0	0	1	*1*	*0*	*1*	*1*	*1*	*1*	*1*	*1*
0	1	0	*1*	*1*	*0*	*1*	*1*	*1*	*1*	*1*
0	1	1	*1*	*1*	*1*	*0*	*1*	*1*	*1*	*1*
1	0	0	*1*	*1*	*1*	*1*	*0*	*1*	*1*	*1*
1	0	1	*1*	*1*	*1*	*1*	*1*	*0*	*1*	*1*
1	1	0	*1*	*1*	*1*	*1*	*1*	*1*	*0*	*1*
1	1	1	*1*	*1*	*1*	*1*	*1*	*1*	*1*	*0*

10. four
11.



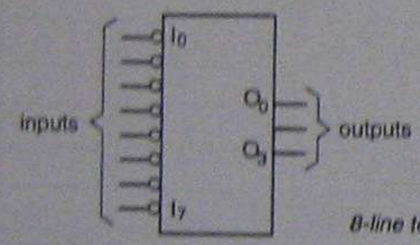
12. A priority encoder responds to the highest order input if two or more inputs are asserted together.

13.

truth table for an active high output decimal-to-BCD encoder

inputs										outputs			
I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	O_3	O_2	O_1	O_0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	*0*	*0*	*0*	*0*
0	1	0	0	0	0	0	0	0	0	*0*	*0*	*0*	*1*
0	0	1	0	0	0	0	0	0	0	*0*	*0*	*1*	*0*
0	0	0	1	0	0	0	0	0	0	*0*	*0*	*1*	*1*
0	0	0	0	1	0	0	0	0	0	*0*	*1*	*0*	*0*
0	0	0	0	0	1	0	0	0	0	*0*	*1*	*0*	*1*
0	0	0	0	0	0	1	0	0	0	*0*	*1*	*1*	*0*
0	0	0	0	0	0	0	1	0	0	*0*	*1*	*1*	*1*
0	0	0	0	0	0	0	0	1	0	*1*	*0*	*0*	*0*
0	0	0	0	0	0	0	0	0	1	*1*	*0*	*0*	*1*

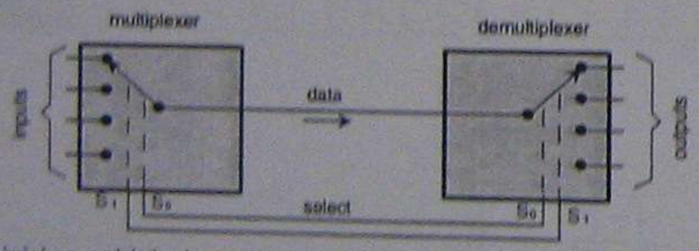
14. When asserted, the ENABLE input(s) let the output(s) respond to the data inputs.



8-line to 3-line encoder

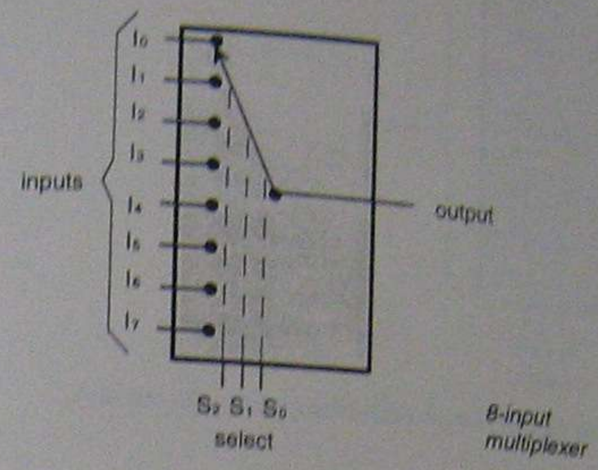
Section 10

1.



2. Time division multiplexing refers to a system where each input is individually selected in turn for a fixed time while all others are deselected.

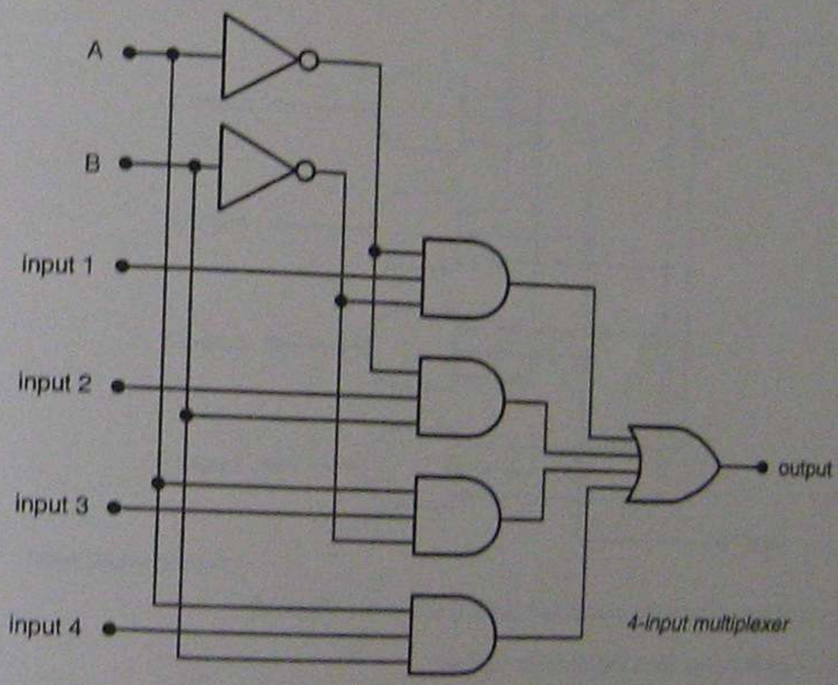
3.



8-input multiplexer

4. four

5.



4-input multiplexer

hat's the difference between a priority encoder and an ordinary encoder?

mplete the truth table below, which is for a decimal-to-BCD encoder with active
h inputs and outputs.

table for an active high output decimal-to-BCD encoder

inputs									outputs			
I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	O_3	O_2	O_1	O_0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0				
1	0	0	0	0	0	0	0	0				
0	1	0	0	0	0	0	0	0				
0	0	1	0	0	0	0	0	0				
0	0	0	1	0	0	0	0	0				
0	0	0	0	1	0	0	0	0				
0	0	0	0	0	1	0	0	0				
0	0	0	0	0	0	1	0	0				
0	0	0	0	0	0	0	1	0				
0	0	0	0	0	0	0	0	1				

etch a general diagram of an octal to binary (8-line to 3-line) encoder.

hat's the purpose of the ENABLE inputs on a decoder or an encoder IC?

Skill practice for Section 9

Title **Encoders and decoders**

Task In this exercise you build a number of encoder and decoder circuits and analyse their operation by completing truth tables for each one.

Equipment IC types: 74LS00 x 2, 74LS04, '08 and '32
digital breadboarding system, logic probe, IC extractor

Part 1 4-input encoder and 1-of-4 decoder

The encoder circuit in this part accepts a decimal input (from 0 to 3) and encodes it to a 2-bit binary code (from 00 to 11). Input 0 has no effect as the code will be 00 when all other inputs are 0. It is shown on the circuit to illustrate the fourth condition. IC encoders overcome this anomaly with an output called *group select* (or GS output). This output becomes active if any input is active. Therefore, you can tell if an output code of 00 is the result of input 0 being active, rather than the result of *no* inputs being active.

1. Connect the circuit of Figure 1. Notice that pin 14 connects to the +5V supply and ground connects to pin 7 of the IC. Connect logic level switches to the inputs and LED indicators to the outputs. Then complete truth table 1 by applying each of the four input combinations.

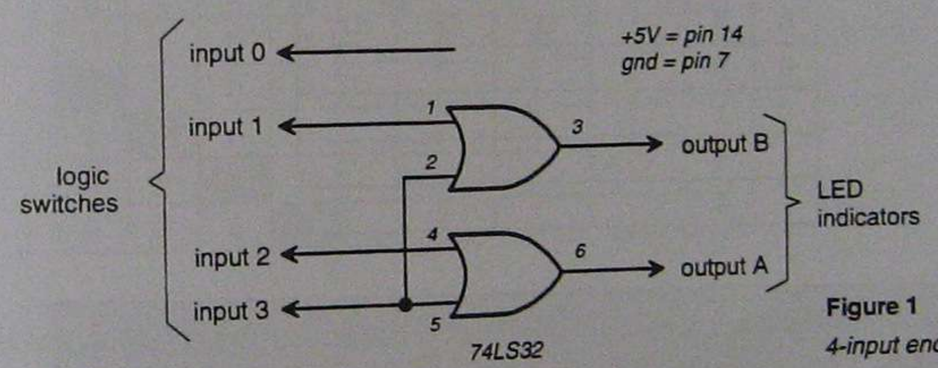
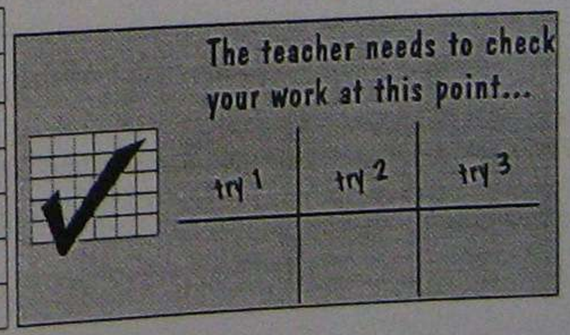


Figure 1
4-input encoder

truth table for 4-input encoder

inputs				outputs	
I_0	I_1	I_2	I_3	A	B
0	0	0	0		
1	0	0	0		
0	1	0	0		
0	0	1	0		
0	0	0	1		



2. Leaving the circuit of Figure 1 intact, construct the 1-of-4 decoder circuit shown in Figure 2. This circuit decodes a 2-bit binary input to decimal.

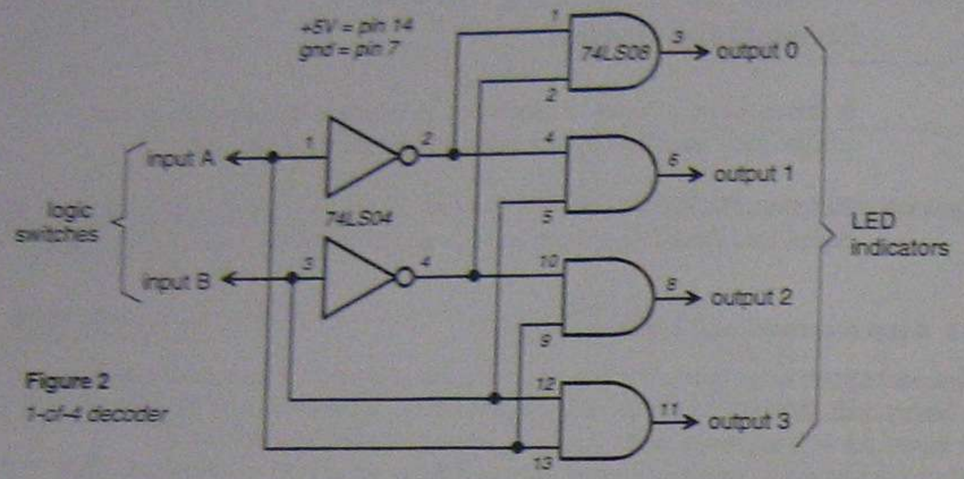


Figure 2
1-of-4 decoder

3. Complete the truth table by operating the input switches and recording the state of the outputs for each input combination.

truth table for 1-of-4 decoder

inputs		outputs			
A	B	O_0	O_1	O_2	O_3
0	0				
0	1				
1	0				
1	1				

The teacher needs to check your work at this point...	try 1	try 2	try 3

Part 2 Combined 4-input encoder and 1-of-4 decoder

In this part, you connect the encoder of Figure 1 to the decoder of Figure 2. In effect, the four decimal inputs are encoded onto two lines, then decoded back to four separate outputs. In a practical situation, this means four digital inputs are being sent over two lines, therefore saving wiring.

1. Connect the circuits of Figure 1 and Figure 2 to give the circuit shown in Figure 3.

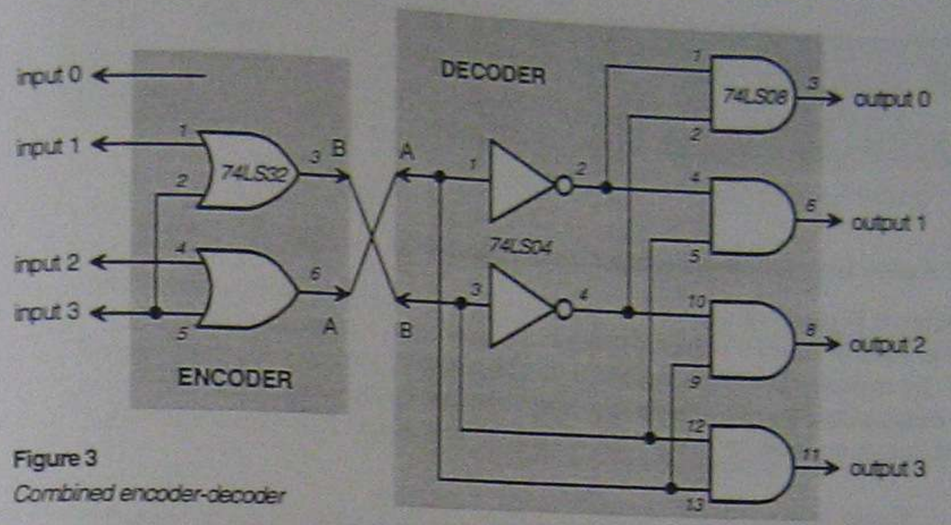


Figure 3
Combined encoder-decoder

2. Complete the truth table for the circuit by operating the input switches and recording the state of the outputs.

truth table for combined encoder-decoder

inputs				outputs			
I_0	I_1	I_2	I_3	O_0	O_1	O_2	O_3
0	0	0	0				
1	0	0	0				
0	1	0	0				
0	0	1	0				
0	0	0	1				

If two or more inputs are active at the same time (at a logic 1), the output will be invalid. A priority encoder overcomes this problem by responding only to the highest order input. For example, if input 0 and input 2 were both active, a priority encoder would respond by recognising input 2 only.

3. Determine the output of Figure 3 when more than one input is at a logic 1.

   a) output when inputs 1 and 2 are both a logic 1 _____

   b) output when inputs 2 and 3 are both a logic 1 _____

   c) output when inputs 1 and 3 are both a logic 1 _____

   d) output when inputs 1, 2 and 3 are a logic 1 _____

The teacher needs to check your work at this point...	try 1	try 2	try 3

Part 3 Active low combined 4-input encoder and 1-of-4 decoder

The circuit of Figure 3 operates with *active high* inputs and *active high* outputs. This means an input is asserted when it's a logic 1 and an output responds by becoming a logic 1. By replacing the AND and OR gates with NAND gates, the circuit of Figure 3 can be changed to *active low* inputs and outputs, where a logic 0 asserts an input and a selected output responds by becoming a logic 0. Many encoder ICs operate this way.

1. Replace the 74LS32 OR gate and the 74LS08 AND gate with a 74LS00 NAND gate, giving the circuit of Figure 4.

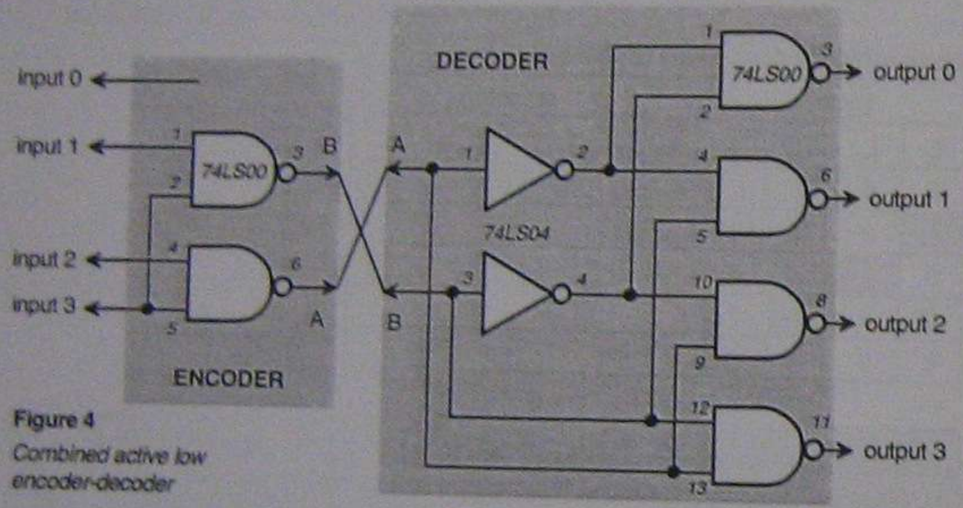


Figure 4
Combined active low encoder-decoder

2. Complete the truth table for the circuit.

truth table for active low combined encoder-decoder

inputs				outputs			
I_0	I_1	I_2	I_3	O_0	O_1	O_2	O_3
1	1	1	1				
0	1	1	1				
1	0	1	1				
1	1	0	1				
1	1	1	0				

Section 10 Multiplexers and demultiplexers

Purpose

To introduce basic multiplexers and demultiplexer ICs and their principle of operation. This section covers learning outcome 8.

Objectives

At the end of this section you should be able to:

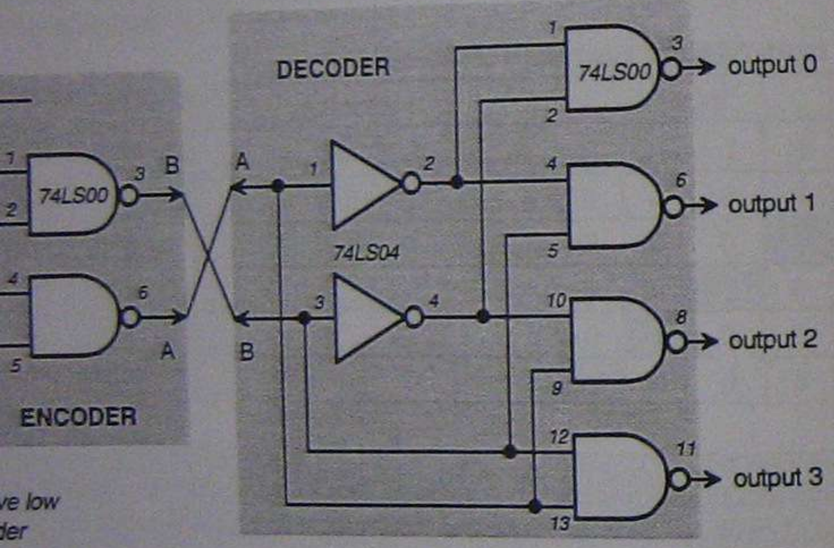
- ☐ Sketch the block diagram of a basic 4-way multiplexing system.
- ☐ Explain the principle of time-division multiplexing.
- ☐ Explain the operation of a 1 of 4 multiplexer circuit.
- ☐ Explain the operation of a 1 of 8 multiplexer circuit.
- ☐ Describe the relationship between the number of select inputs and the number of data inputs of a multiplexer IC.
- ☐ Identify and explain the function of the pin designations of a typical multiplexer IC.
- ☐ List typical applications for multiplexer circuits.
- ☐ Explain the operation of a 1 of 4 demultiplexer circuit.
- ☐ Explain the operation of a 1 of 8 demultiplexer circuit.
- ☐ Describe the relationship between the number of select inputs and the number of data outputs of a demultiplexer IC.
- ☐ Identify and explain the function of the pin designations of a typical demultiplexer IC.
- ☐ List typical applications for demultiplexer circuits.
- ☐ Connect and verify the operation of circuits using multiplexer and demultiplexer ICs.
- ☐

The teacher needs to check your work at this point...	try 1	try 2	try 3

low combined 4-input encoder and 1-of-4 decoder

Figure 3 operates with *active high* inputs and *active high* outputs. This is asserted when it's a logic 1 and an output responds by becoming a lacing the AND and OR gates with NAND gates, the circuit of Figure 3 l to *active low* inputs and outputs, where a logic 0 asserts an input and a responds by becoming a logic 0. Many encoder ICs operate this way.

e 74LS32 OR gate and the 74LS08 AND gate with a 74LS00 NAND g the circuit of Figure 4.



ve low
der

e the truth table for the circuit.

truth table for active low combined encoder-decoder

inputs				outputs			
I_0	I_1	I_2	I_3	O_0	O_1	O_2	O_3
1	1	1	1				
0	1	1	1				
1	0	1	1				
1	1	0	1				
1	1	1	0				

Section 10 Multiplexers and demultiplexers

Purpose

To introduce basic multiplexers and demultiplexer ICs and their principle of operation. This section covers learning outcome 8.

Objectives

At the end of this section you should be able to:

- ☐ Sketch the block diagram of a basic 4-way multiplexing system.
- ☐ Explain the principle of time-division multiplexing.
- ☐ Explain the operation of a 1 of 4 multiplexer circuit.
- ☐ Explain the operation of a 1 of 8 multiplexer circuit.
- ☐ Describe the relationship between the number of select inputs and the number of data inputs of a multiplexer IC.
- ☐ Identify and explain the function of the pin designations of a typical multiplexer IC.
- ☐ List typical applications for multiplexer circuits.
- ☐ Explain the operation of a 1 of 4 demultiplexer circuit.
- ☐ Explain the operation of a 1 of 8 demultiplexer circuit.
- ☐ Describe the relationship between the number of select inputs and the number of data outputs of a demultiplexer IC.
- ☐ Identify and explain the function of the pin designations of a typical demultiplexer IC.
- ☐ List typical applications for demultiplexer circuits.
- ☐ Connect and verify the operation of circuits using multiplexer and demultiplexer ICs.
- ☐

Review questions for Section 10

1. Show with a diagram how a basic 4-way multiplexed data distribution system can be made with two 4-way single-pole switches.

2. What's meant by the term *time division multiplexing*?

3. Sketch a logic symbol that shows all the inputs and outputs for an 8-input multiplexer.

4. How many address lines does a multiplexer IC with 16 inputs need?

5. Complete the logic circuit of the 4-input multiplexer shown in Figure 1.

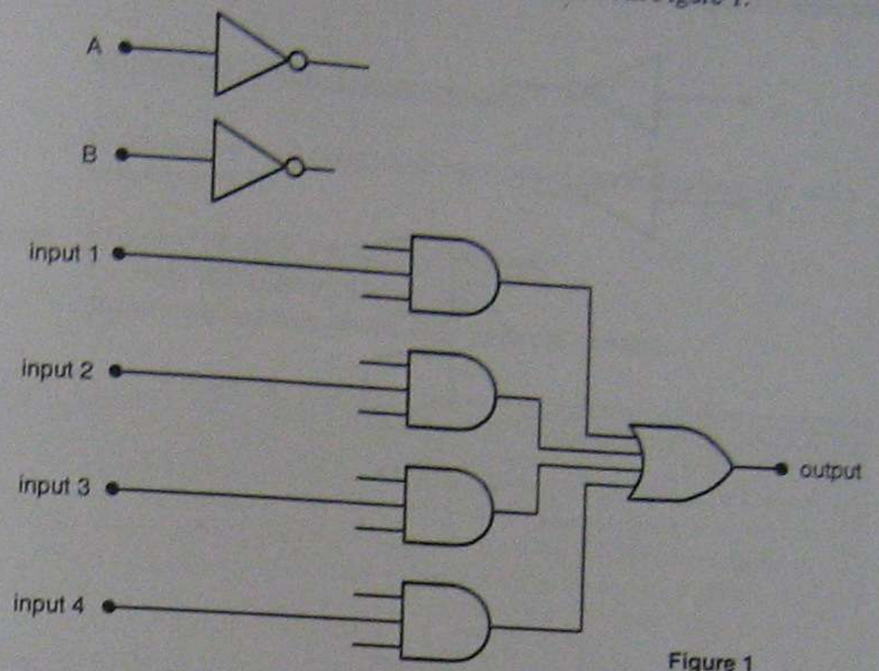


Figure 1

6. Sketch a logic symbol that shows all the inputs and outputs for an 1-of-8 input demultiplexer.

7. What is the function of the *enable* (or EN) input on a multiplexer or demultiplexer IC?

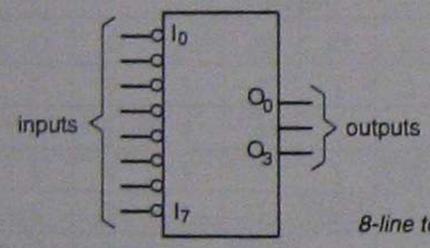
12. A priority encoder responds to the highest order input if two or more inputs are asserted together.

13.

truth table for an active high output decimal-to-BCD encoder

inputs										outputs			
I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	O_3	O_2	O_1	O_0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	*0*	*0*	*0*	*0*
0	1	0	0	0	0	0	0	0	0	*0*	*0*	*0*	*1*
0	0	1	0	0	0	0	0	0	0	*0*	*0*	*1*	*0*
0	0	0	1	0	0	0	0	0	0	*0*	*0*	*1*	*1*
0	0	0	0	1	0	0	0	0	0	*0*	*1*	*0*	*0*
0	0	0	0	0	1	0	0	0	0	*0*	*1*	*0*	*1*
0	0	0	0	0	0	1	0	0	0	*0*	*1*	*1*	*0*
0	0	0	0	0	0	0	1	0	0	*0*	*1*	*1*	*1*
0	0	0	0	0	0	0	0	1	0	*1*	*0*	*0*	*0*
0	0	0	0	0	0	0	0	0	1	*1*	*0*	*0*	*1*

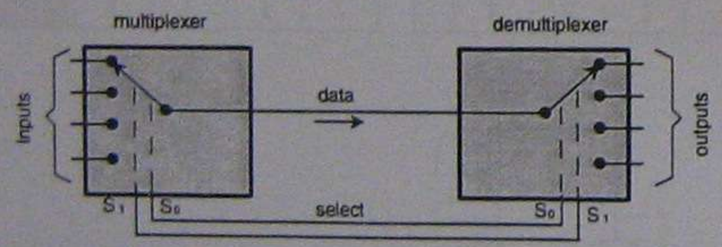
14. When asserted, the ENABLE input(s) let the output(s) respond to the data inputs.



8-line to 3-line encoder

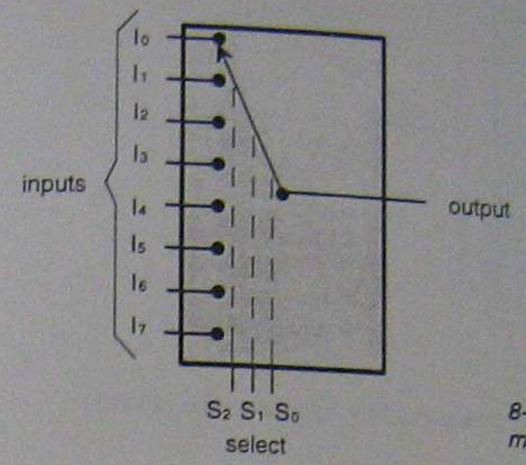
Section 10

1.



2. Time division multiplexing refers to a system where each input is individually selected in turn for a fixed time while all others are deselected.

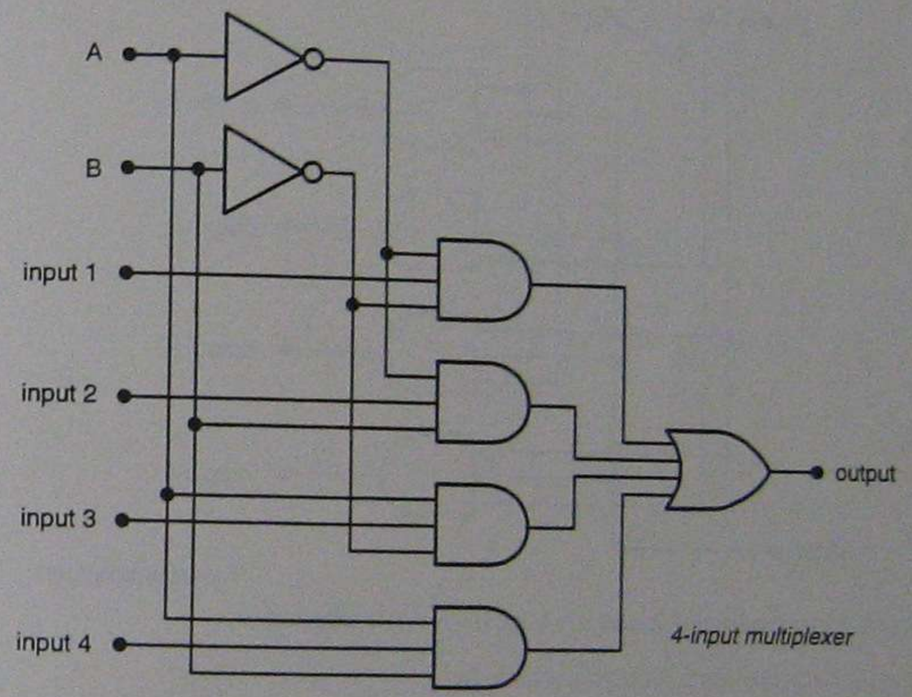
3.



8-input multiplexer

4. four

5.



4-input multiplexer

Input 2

D_1

Fig

Note that Q_2 transistor. inputs of TTL ringing that or is coupled prevent the forward bias

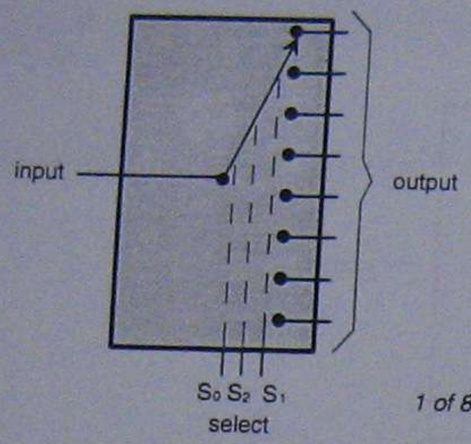
2.1 OPERATION

(1) If either forward to ground low) and *total* I_{IL} not excee and the i ground.

(2) As a I junction zero. T voltage

(3) The col there i is thu correct

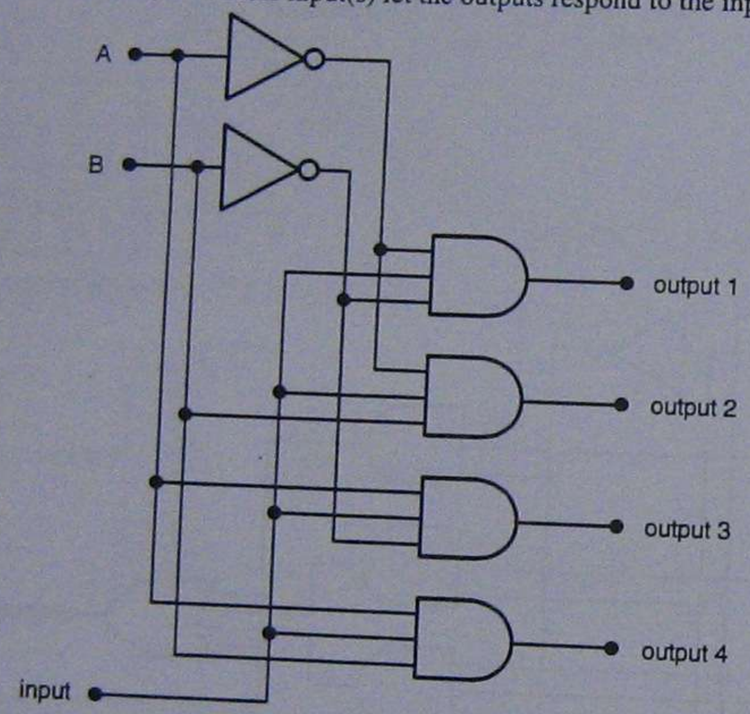
6.



1 of 8 demultiplexer

7. When asserted, the enable input(s) let the outputs respond to the input(s).

8.



1-of-4demultiplexer

9. Active low means the output is a logic 0 when it's selected.

10. (a) Applications for a multiplexer include:
    - selecting data from a number of points and switching it to a common data line
    - implementing a combinational logic truth table

    (b) Applications for a demultiplexer include
    - distributing data from a common point to a number of points
    - a part of a decoder circuit

Review Questions

1. What is the main difference between an analog signal and a digital signal?

2. If a logic 1 represents a closed switch, how is an open switch represented?

3. If a voltage of +5V is a logic 1 (or a HIGH), what is a voltage of 0V?

4. Under what input conditions will the output of a 4-input NOR gate be a logic 1?

5. What is the output logic level of a 4-input NAND gate with one input held at logic 0?

6. Complete Table 1 by writing the logic levels for the inputs and outputs of each gate.

TABLE 1

Gate type	*Input A*	*Input B*	*Input C*	*Output*
NAND	0	0	1	
NOR	0	0		1
AND	1	1		0
OR	0	0	1	
AND	0	1	1	
NAND	1			1
NOT	1			
OR		0		1
EXOR	1	0		
NOR	0			1

7. How many different binary numbers can be applied to the digital circuit shown in Figure 1?

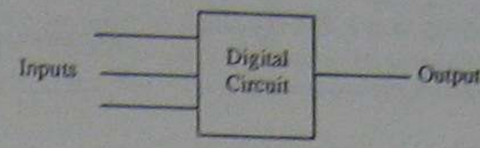


Figure 1

8. How many different binary numbers can the output of the digital circuit in figure 1 produce?

9. Complete these truth tables:

A	*B*	*Z*
0	0	
0	1	
1	0	
1	1	

AND gate truth table

A	*B*	*Z*
0	0	
0	1	
1	0	
1	1	

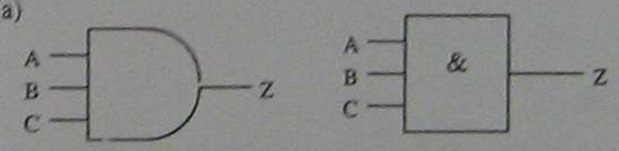
NOR gate truth table

A	*B*	*Z*
0	0	
0	1	
1	0	
1	1	

EXOR gate truth table

10. Identify each gate shown here and write its Boolean expression.

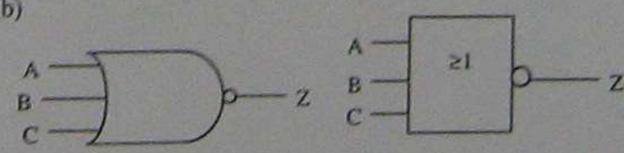
(a)



Type of gate: AND

Boolean expression: Z = Z = A·B·C

(b)



Type of gate: NOR

Boolean expression: Z =

$\overline{A+B+C}$

ANSWERS TO REVIEW QUESTIONS

Section 1

1. An analog signal varies over a continuous range of values. A digital signal has two values only.

2. An open switch is represented by a logic 0.

3. A voltage of 0V is a logic 0 or a LOW.

4. If all the inputs are a logic 0.

5. A logic 1.

6.

TABLE 1

Gate type	*Input A*	*Input B*	*Input C*	*Output*
NAND	0	0	1	1
NOR	0	0	0	1
AND	1	1	0	0
OR	0	0	1	1
AND	0	1	1	0
NAND	1	0		1
NOT	1			0
OR	1	0		1
EXOR	1	0		1
NOR	0	0		1

7. Eight

8. Two

9.

A	*B*	*Z*
0	0	0
0	1	0
1	0	0
1	1	1

AND gate truth table

A	*B*	*Z*
0	0	1
0	1	0
1	0	0
1	1	0

NOR gate truth table

A	*B*	*Z*
0	0	0
0	1	1
1	0	1
1	1	0

EXOR gate truth table

The EX-OR gate is more specialised than the previous five gates, and has the truth table, symbol and definitive statement as shown in Fig.6.

6.7 THE EXCLUSIVE NOR (EX-NOR) and the BUFFER GATES

The EX-NOR gate is similar to the EX-OR device, except it has an inverter at its output. Its definitive statement is *'If the inputs to an EX-NOR gate differ, the output is a logic 0.'* The EX-NOR gate is the least common of all the gates.

The non-inverting BUFFER gate is used where the load is too great for a particular output to drive. Some gates incorporate buffers inside the IC, and the buffer does not alter the logic levels.

7. GATE TYPE NUMBERS

All of the above gates are represented in the various families of digital ICs. The following list shows commonly used type numbers available in the TTL and CMOS families.

Gate Type	*No. of gates per package*	*No. of pins per package*	*Package type*	*CMOS type number*	*TTL type number*
AND 2 INPUT	4	14	DIL	4081	7408
OR 2 INPUT	4	14	DIL	4071	7432
INVERTER	6	14	DIL	4069	7404
NAND 2 INPUT	4	14	DIL	4011	7400
NAND 3 INPUT	3	14	DIL	4023	7410
NAND 4 INPUT	2	14	DIL	4012	7420
NAND 8 INPUT	1	14	DIL	4068	7430
NOR 2 INPUT	4	14	DIL	4001	7402
Exclusive OR	4	14	DIL	4070	7486
Exclusive NOR	4	14	DIL	4077	9386
Buffer/driver	6	14	DIL	4050	7417

The above list is by no means comprehensive, but lists the most common types. Generally, the TTL device would be purchased as a TTL LS type. Prices range from 20c per package to about $1.00.

****

ADVANCED CERTIFICATE IN APPLIED INDUSTRIAL ELECTRONICS

YEAR 1 DIGITAL PRINCIPLES

THEORY LESSON 2

Written by Peter Phillips, 1990

BOOLEAN ALGEBRA

OBJECTIVES: At the end of this lesson you should be able to:

(a) Write the Boolean expression for a basic combinational logic circuit.
(b) Derive the Boolean expression from a basic combinational circuit.
(c) Apply basic simplification to a Boolean expression.

INTRODUCTION

To facilitate the analysis and design of digital circuitry, use is made of Boolean algebra, so named after its originator, George Boole, who devised algebraic rules for use in deductive logic in 1847. Claude Shannon, a Bell Laboratories engineer, adapted the algebra for use in telephone exchange design in 1938. It is this form of Boolean algebra that is used in digital logic. A specific application of the algebra was formulated by DeMorgan, whose two theorems allow, in effect, any given logic circuit to be implemented using either NAND or NOR gates. These two gates are referred to as *universal gates* for this reason.

1. SUMMARY OF THE SYMBOLS USED IN BOOLEAN ALGEBRA

(.) means AND ..... (1)

Thus, A.B is read as A *and* B. Often the (.) is understood, and is deleted. The output (Z) for the AND gate of Fig.1 is written as A.B.C = Z or ABC = Z

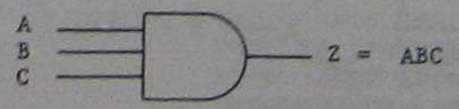


Fig.1

(+) means OR ..... (2)

Thus A + B is read as A *or* B. The output (Z) for the OR gate of Fig.2 is written as A + B + C = Z

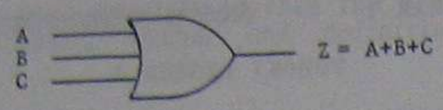


Fig.2

($\bar{}$) means INVERT ..... (3)

Thus $\bar{A}$ is read as *NOT* A. The output of an inverter is often written as shown in Fig.3.

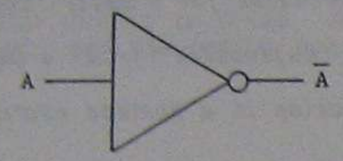


Fig.3

Combining the signs of (1) and (3) enables a NAND gate output to be expressed in Boolean form. The output (Z) for the NAND gate of figure 4 is written as $\overline{A.B.C} = Z$ or $\overline{ABC} = Z$

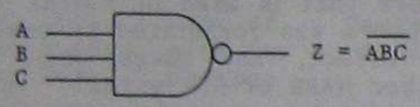


Fig.4.

Combining signs (2) and (3) enables a NOR gate output to be defined in Boolean form. The output (Z) for the NOR gate of Fig.5 is written as $\overline{A+B+C} = Z$

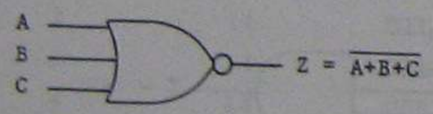


Fig.5 .

Using these *logic operators* it is now possible to;

(a) Express a given combinational logic problem as a Boolean expression.
(b) Implement the logic circuitry required.
(c) Write a Boolean expression for a given combinational logic circuit.

(Teacher note: Provide exercises and examples of (a), (b) and (c).

2. BOOLEAN POSTULATES

Boolean algebra finds considerable use in the following areas:

(a) Simplifying a given logic expression. In this instance, a certain complex control situation is written as a Boolean expression, and then simplified with the removal of redundant terms.

(b) Converting a given Boolean expression from one form to another, thus enabling the circuit to be implemented in NAND or NOR (universal) gates.

The following Postulates, (a statement which can be accepted without further proof) enable Boolean expressions to be manipulated, and are presented in a list form, with a brief explanation where required.

1.	0.0 = 0	These 6 postulates will be recognised as the results obtained from truth tables for an AND and an OR gate.
2.	1 + 1 = 1	
3.	0 + 0 = 0	
4.	1.1 = 1	
5.	1.0 = 0.1 = 0	
6.	1 + 0 = 0 + 1 = 1	
7.	A + 0 = A	Postulates 7 - 17 may seem obvious, but need to be expressed. Their significance can be more readily seen if the gate function is drawn. It will be seen for example, that postulates 11 & 12 mean that the circuit is best implemented with a piece of copper wire, rather than the specified gate.
8.	A.0 = 0	
9.	A + 1 = 1	
10.	A.1 = A	
11.	A + A = A	
12.	A.A = A	
13.	A + $\bar{A}$ = 1	
14.	A.$\bar{A}$ = 0	
15.	If A = 0, then $\bar{A}$ = 1	
16.	If A = 1, then $\bar{A}$ = 0	
17.	$\bar{\bar{A}}$ = A	
18.	A + B = B + A A.B = B.A	Showing that the position within a given term is irrelevant. **(Commutative Laws.)**
19.	A + (B + C) = (A + B) + C	Means that two input gates can be used to provide, collectively, a multi-input gate of that function. **(Associative Laws).** See Figs.6 & 7
20.	(A.B).C = A.(B.C)	

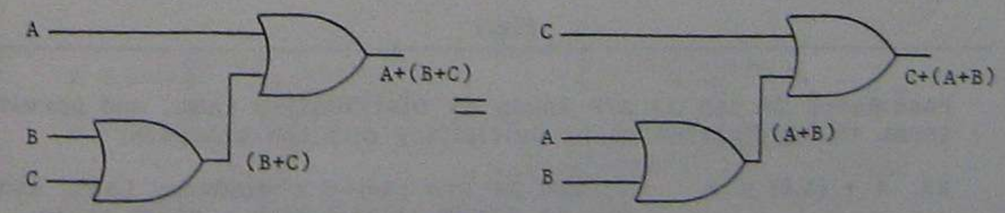


Fig.6

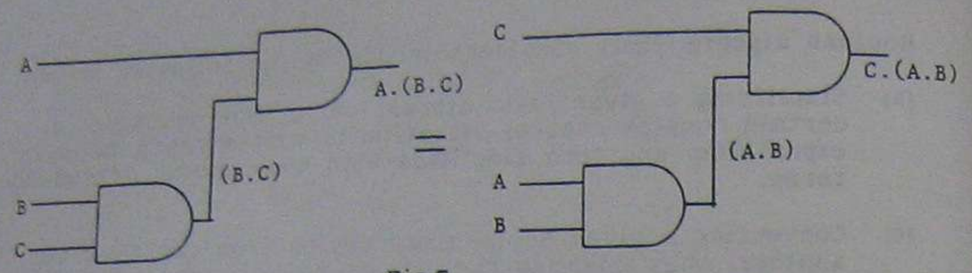


Fig.7 .

21. A.(B + C) = A.B + A.C See Fig.8.

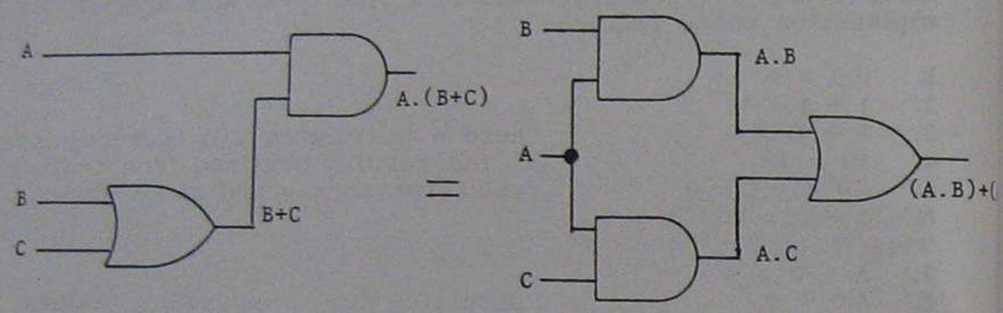


Fig.8

22. A + (B.C) = (A + B).(A + C) See Fig.9.

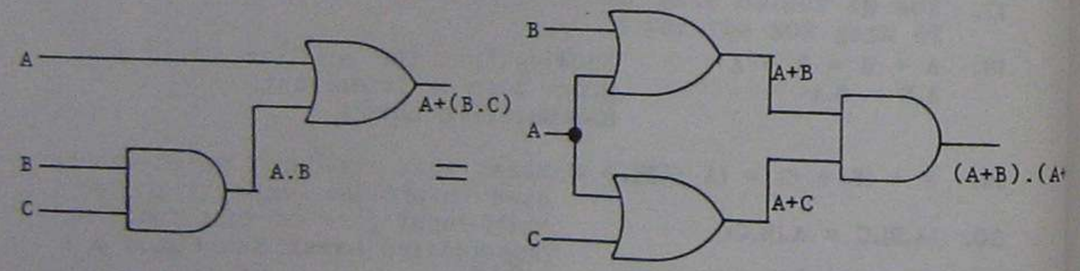


Fig.9

Postulates 21 and 22 are known as **Distributive Laws**, and permit terms to be distributed by multiplying out the brackets.

23. A + (A.B) = A
24. A.(A + B) = A

23 and 24 are called **Redundancy Laws**, and can be easily proved with a truth table. Useful in simplification procedures.

25. A + ($\overline{A}$.B) = A + B See Fig.10 for circuit equivalent.

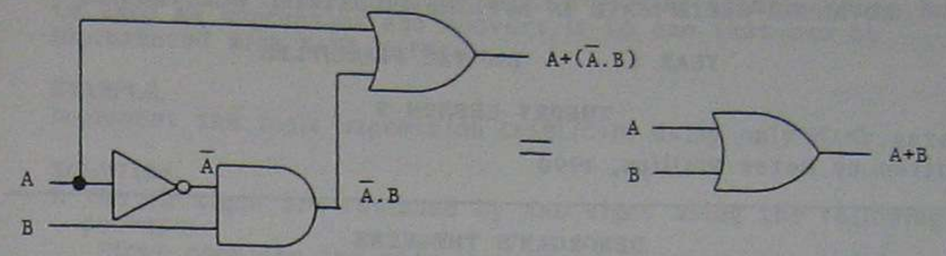


Fig.10

26. A.($\overline{A}$ + B) = A.B See Fig.11 for circuit equivalent.

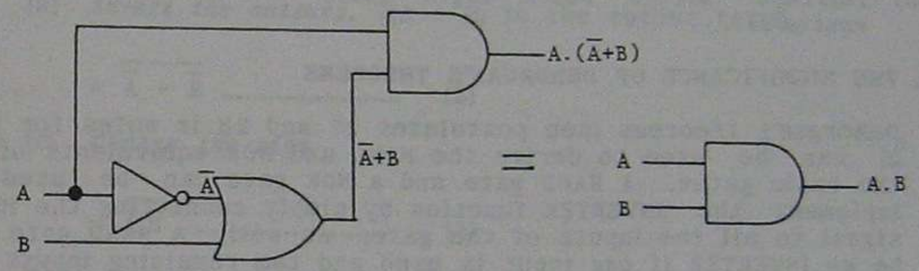


Fig.11

27. $\overline{A.B}$ = $\overline{A}$ + $\overline{B}$ See Fig.12 for circuit equivalent.

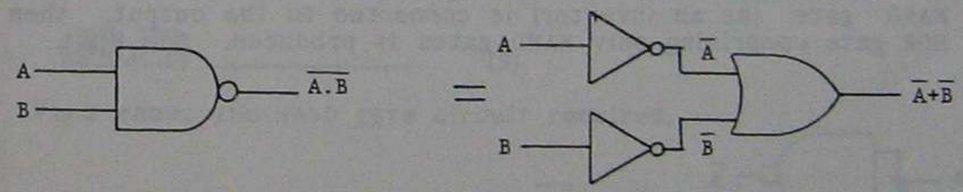


Fig.12

28. $\overline{A + B}$ = $\overline{A}$.$\overline{B}$ See figure 13 for circuit equivalent.

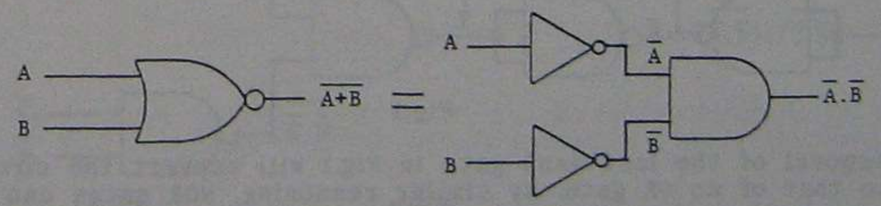


Fig.13

Postulates 27 & 28 are refered to as **DeMorgan's Theorems**, the subject of the next lesson.

$\overline{A}.\overline{B}$ ..... (3)

By similar reasoning
the entire exp

Review Questions

1. What is the main difference between an analog signal and a digital signal?

_____

_____

2. If a logic 1 represents a closed switch, how is an open switch represented?

_____

3. If a voltage of +5V is a logic 1 (or a HIGH), what is a voltage of 0V?

_____

4. Under what input conditions will the output of a 4-input NOR gate be a logic 1?

_____

5. What is the output logic level of a 4-input NAND gate with one input held at logic 0?

_____

6. Complete Table 1 by writing the logic levels for the inputs and outputs of each gate.

TABLE 1

Gate type	*Input A*	*Input B*	*Input C*	*Output*
NAND	0	0	1	
NOR	0	0		1
AND	1	1		0
OR	0	0	1	
AND	0	1	1	
NAND	1			1
NOT	1			
OR		0		1
EXOR	1	0		
NOR	0			1

7. How many different binary numbers can be applied to the digital circuit shown in Figure 1?

combinations of IP's

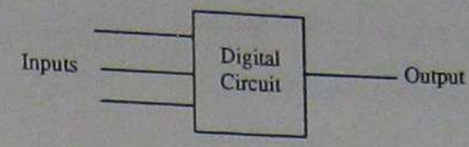


Figure 1

_____

8. How many different binary numbers can the output of the digital circuit in figure 1 produce?

_____

9. Complete these truth tables:

A	B	Z
0	0	
0	1	
1	0	
1	1	

AND gate truth table

A	B	Z
0	0	
0	1	
1	0	
1	1	

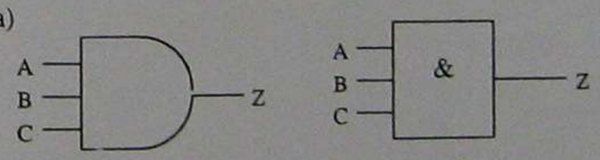
NOR gate truth table

A	B	Z
0	0	
0	1	
1	0	
1	1	

EXOR gate truth table

10. Identify each gate shown here and write its Boolean expression.

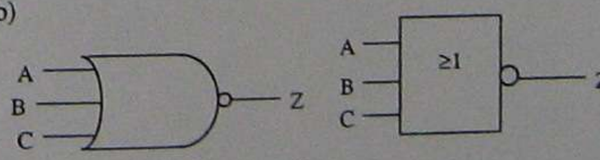
(a)



Type of gate: AND

Boolean expression: Z = $Z = A \cdot B \cdot C$

(b)



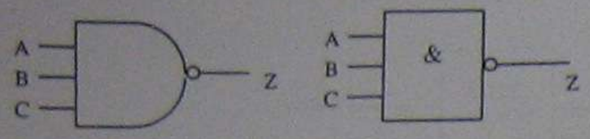
Type of gate: NOR

Boolean expression: Z = $\overline{A+B+C}$

$A + B = \overline{\overline{A}.\overline{B}}$ ..... (3)

By similar reasoni-
the entire

(c)

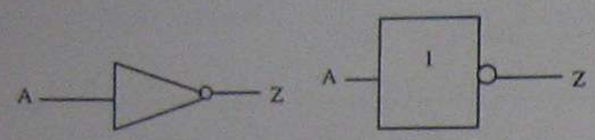


Type of gate: NAND

Boolean expression: Z =

$\overline{A \cdot B \cdot C}$

(d)



Type of gate: INV

Boolean expression: Z =

$\overline{A}$

Skill Practice 1: Logic gates and the logic probe

Aim

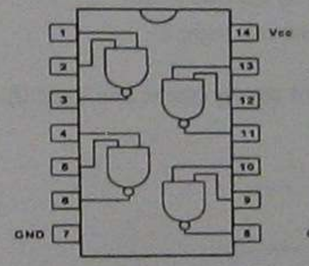
In this exercise, you will use manufacturer's data to determine the IC pin connections and the function of each IC used in this exercise, construct basic digital circuits on a bread-boarding socket, and use a logic probe to determine logic levels.

Equipment

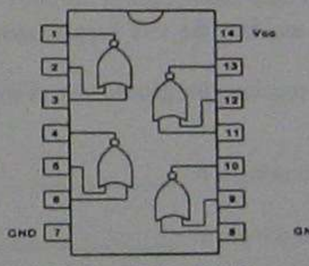
- IC types: 74LS00, '02, '04, '08, '32 and '86
- digital bread-boarding system
- logic probe
- dual trace oscilloscope
- voltmeter
- IC extractor
- 0.1µF capacitor

Procedure

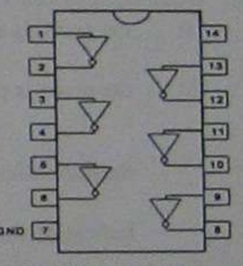
1. Using the data below, complete table 1 (on next page) by determining the pin numbers for one output and its associated input(s).



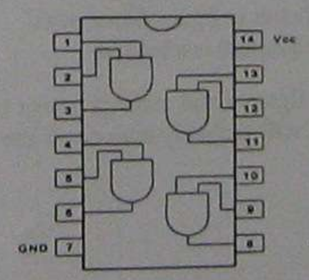
74LS00 quad 2-input NAND



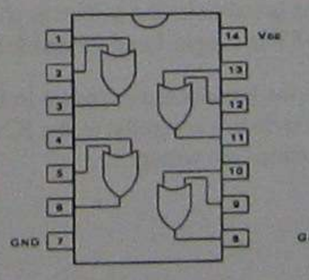
74LS02 quad 2-input NOR



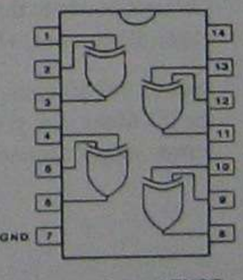
74LS04 hex inverter



74LS08 quad 2-input AND



74LS32 quad 2-input OR



74LS86 quad 2-input EXOR

DATA SHEET FOR THE ICs USED IN THIS EXERCISE

ADVANCED CERTIFICATE IN APPLIED INDUSTRIAL ELECTRONICS

YEAR 1 DIGITAL PRINCIPLES

THEORY LESSON 3

Written by Peter Phillips, 1990

DEMORGAN'S THEOREMS

OBJECTIVES: At the end of this lesson you should be able to:

(a) Apply DeMorgan's theorems to change a Boolean expression from one form to another.
(b) Implement all 5 basic logic gates in their NAND or NOR equivalent.

1. THE SIGNIFICANCE OF DEMORGAN'S THEOREMS

DeMorgan's theorems (see postulates 27 and 28 in notes for Week 2) can be used to derive the NAND and NOR equivalents of all the basic gates. A NAND gate and a NOR gate can be used to implement the INVERTER function by simply connecting the input signal to all the inputs of the gate. As well, a NAND gate can be an INVERTER if *one* input is used and the remaining inputs are connected to a logic 1. Similarly a NOR gate can be an INVERTER by using *one* input, and connecting the others to a logic 0. *(Teacher note: explain with truth tables).* If for example, the circuit shown in Fig.13 in the notes for week 2 has the inverters replaced by the NAND gate equivalent, and another NAND gate (as an inverter) is connected to the output, then a NOR gate comprising only NAND gates is produced. See Fig.1.

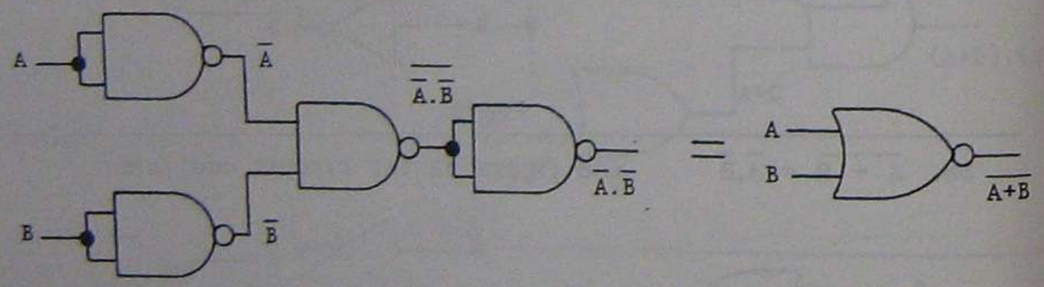


Fig.1

Removal of the last NAND gate in Fig.1 will convert the circuit to that of an OR gate. By similar reasoning, NOR gates can be used to implement the NAND and AND functions. Using DeMorgan's theorems also allows a given Boolean expression to be altered by changing the (.) and (+) signs to permit its implementation

directly in NAND or NOR gates. The following example shows how to take an expression and convert it to one that can be implemented with NAND gates.

EXAMPLE.
Implement the logic expression (A+B).(C+D) using only NAND gates.

SOLUTION.
1. The OR signs are replaced by AND signs using the following process:
First consider the term (A + B).

(b) ***Invert the inputs.*** [A and B are the individual inputs]

$$= \overline{A} + \overline{B} \quad \text{.....} \quad (1)$$

(a) ***Invert the outputs.*** [(A + B) is the output term].

$$= \overline{\overline{A} + \overline{B}} \quad \text{.....} \quad (2)$$

(c) ***Change the sign***

$$= \overline{\overline{A}.\overline{B}} \quad \text{.....} \quad (3)$$

Thus $A + B = \overline{\overline{A}.\overline{B}}$ **.....** **(4)**

By similar reasoning, the term (C + D) can be DeMorganised, and the entire expression now becomes:

$$\overline{\overline{A}.\overline{B}}.\overline{\overline{C}.\overline{D}} \quad \text{.....} \quad (5)$$

Fig.2 shows the NAND gate circuit required.

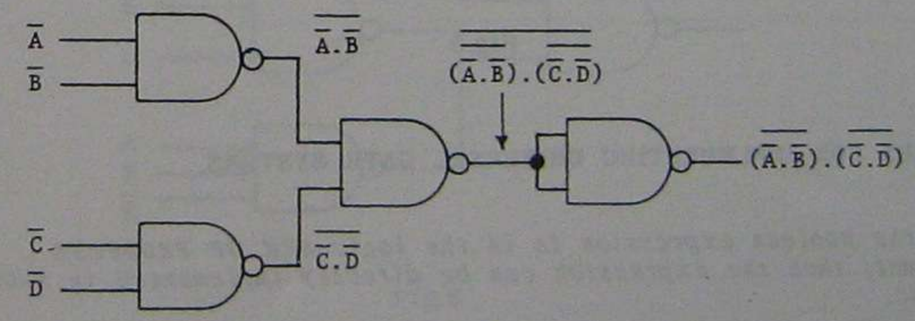


Fig.2

Similarly, DeMorgan's Theorems can be used to change an expression to allow its implementation using NOR gates.

EXAMPLE
Implement the logic expression (A.B).(C+D) using NOR gates.

SOLUTION

1. The term (A.B) has its sign changed as in the previous example, i.e, by *inverting the inputs*, (A & B), *inverting the outputs* (A.B), and *changing the sign* to give:

$$\overline{\bar{A} + \bar{B}} \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad (1)$$

2. The term (C+D) is already in OR form, but the (.) sign between the bracketed terms requires Demorganising to change it.
The inputs are the terms $(\overline{\bar{A}+\bar{B}})$ and (C+D), and the output is the whole expression. Inverting the inputs; output; and sign changing gives:

$$\overline{\overline{(\bar{A}+\bar{B})} + \overline{(C\ D)}} \quad \ldots\ldots\ldots\ldots\ldots \quad (2)$$

The double NOT signs over the term (A+B) cancel.

Thus: $(A.B).(C+D) = \overline{(\bar{A}+\bar{B}) + \overline{(C+D)}}$

The circuit to implement this expression is shown in Fig.3.

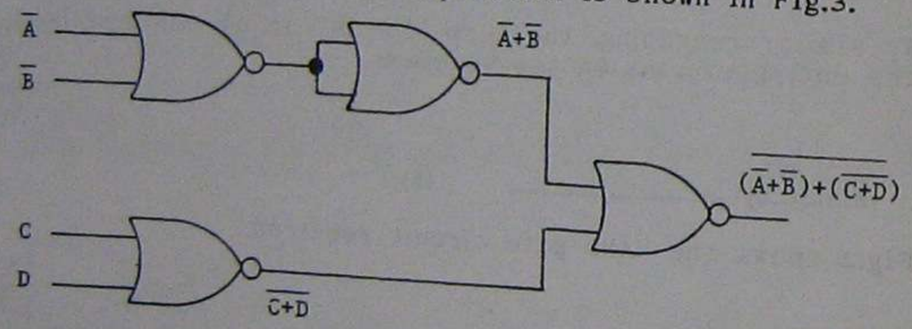


Fig.3

2. RULES FOR IMPLEMENTING UNIVERSAL GATE SYSTEMS

If the Boolean expression is in the logic ***SUM OF PRODUCTS*** *format, then the expression can be directly implemented in NAND logic.*

For example, the expression (A.B)+(C.D) is in the Sum Of Products (SOP) form, and can be drawn using NAND gates instead of conventional gates as shown in Fig.4.

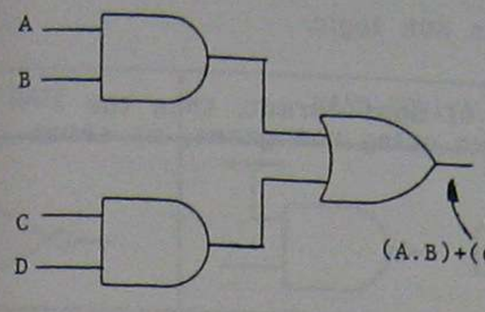


Fig.4(a).

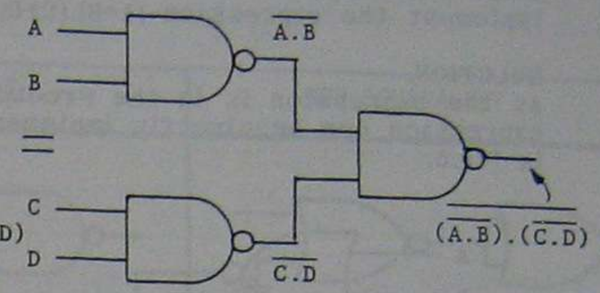


Fig.4(b).

Proof can be obtained by DeMorganising the original expression, giving $\overline{(\bar{A}.\bar{B}).(\bar{C}.\bar{D})}$ which is the output term of Fig.4(b).

EXAMPLE.
Implement the expression $A + (\bar{B}.C) + (AB\bar{C})$ using only NAND gates;

SOLUTION.
From postulate 10, (Week 2) A = A.1 giving the expression:

$(A.1) + (\bar{B}C) + (AB\bar{C})$.

Because it is now in SOP form, its direct implementation in NAND gates is possible, as shown in Fig.5.

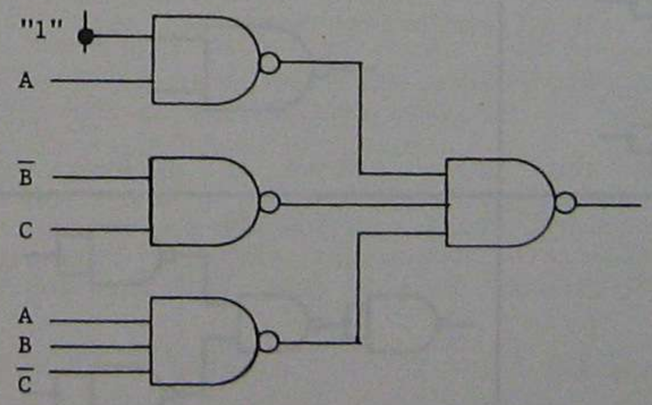


Fig.5

By similar reasoning, it can be shown that if the expression is in the **LOGIC PRODUCT OF SUMS** format, the expression can be implemented directly using NOR logic.

EXAMPLE.
Implement the expression (A+B).(C+D) in NOR logic.

SOLUTION.
As the expression is in the Product of Sums format, then the expression can be directly implemented using NOR gates, as shown in Fig.6.

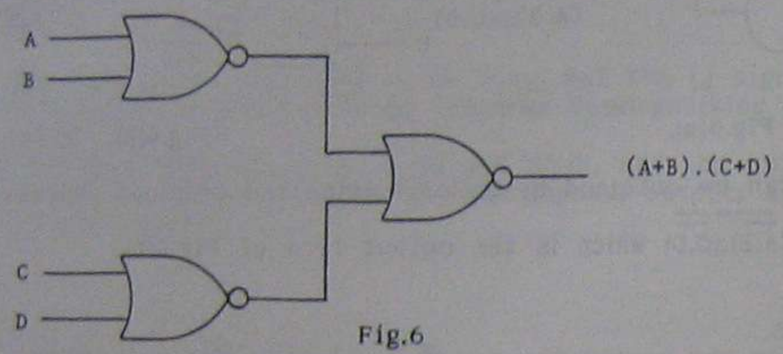
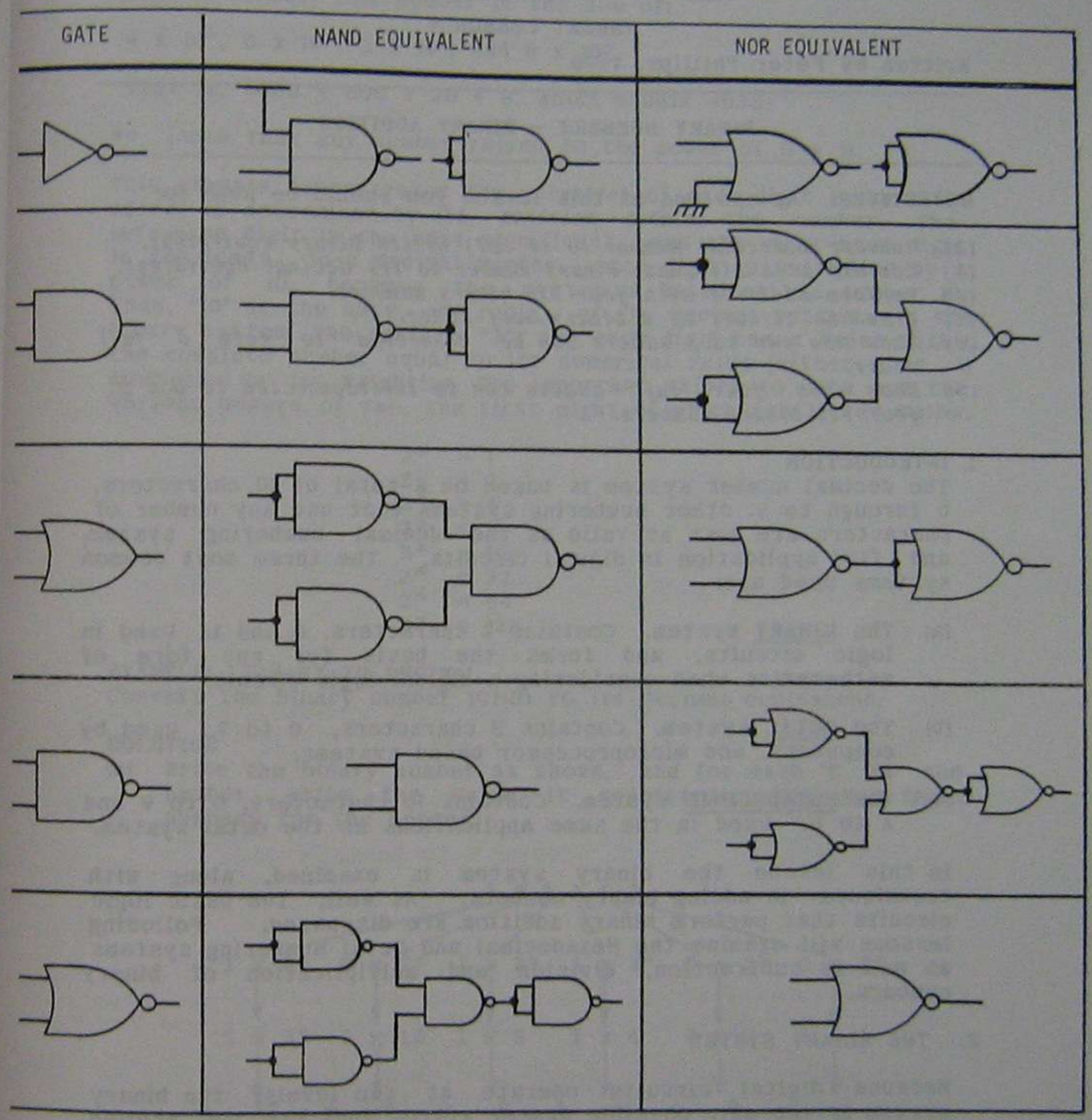


Fig.6

From the foregoing, DeMorgan's theorems find their greatest use in changing a Boolean expression into either SOP or POS forms to allow direct implementation of the expression with universal gates. An alternative method is to draw the universal equivalent of each gate in a logic circuit and to cancel inverter pairs. The NAND and NOR equivalents for all five basic logic gates are shown in Fig.7. Note that the inverter function can be achieved in four ways, as already described.



Basic Gates implemented in NAND & NOR

****

Review Questions

1. Determine the logic state at points D, E and Z for the circuit of figure 1 for all logic conditions of inputs A, B and C.

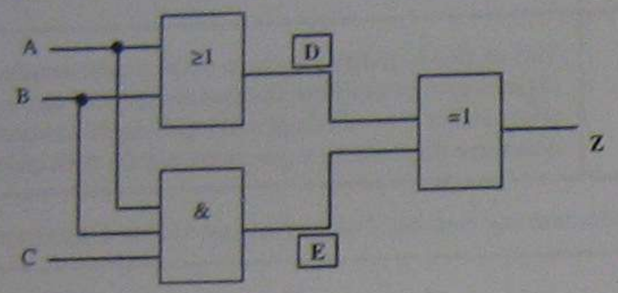


Figure 1

A	B	C	D	E	Z
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

2. Determine the Boolean expression for the output (Z) for the circuit shown in figure 2.

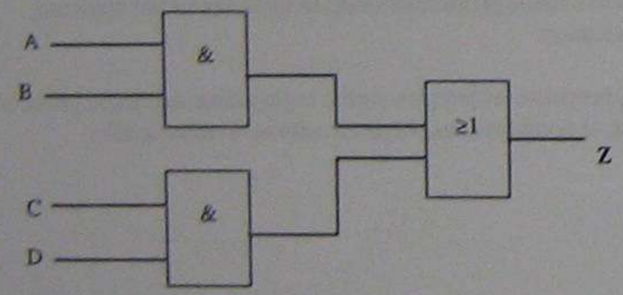


Figure 2

Boolean expression for Z = _____

3. Determine the Boolean expression for the output (Z) for the circuit shown in figure 3.

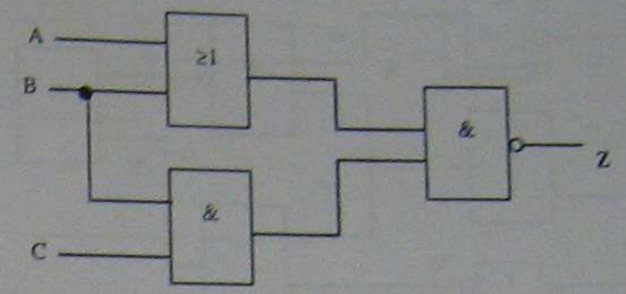


Figure 3

Boolean expression for Z = _____

4. Which gate is faulty is figure 4?

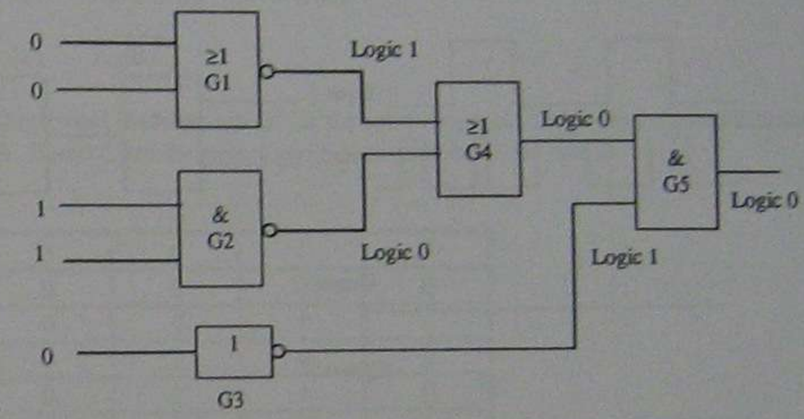


Figure 4

The faulty gate is _____

5. Determine the logic levels at points U to Z in figure 5 (next page).

Logic level at U = _____ Logic level at V = _____

Logic level at W = _____ Logic level at X = _____

Logic level at Y = _____ Logic level at Z = _____

listed below.

$2^0 = 1$
$2^1 = 2$
$2^2 = 4$
$2^3 = 8$

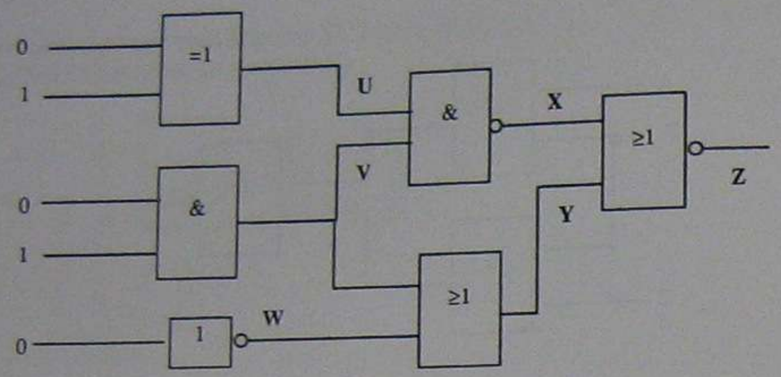


Figure 5

6. Sketch the output waveform of the circuit in figure 6.

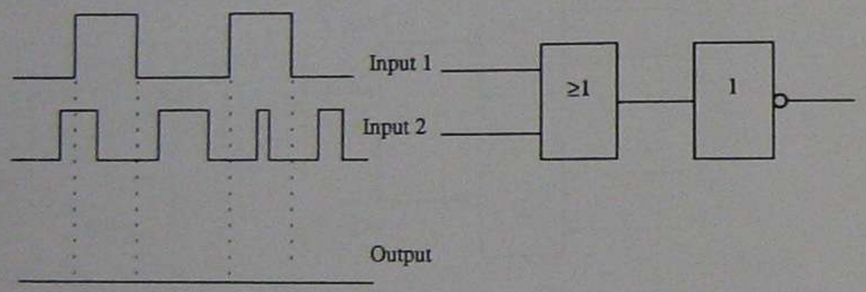
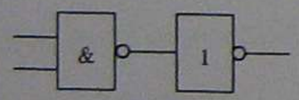


Figure 6

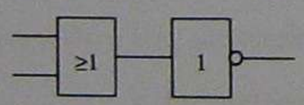
7. Name the function being implemented by each of the following circuits.

(a)



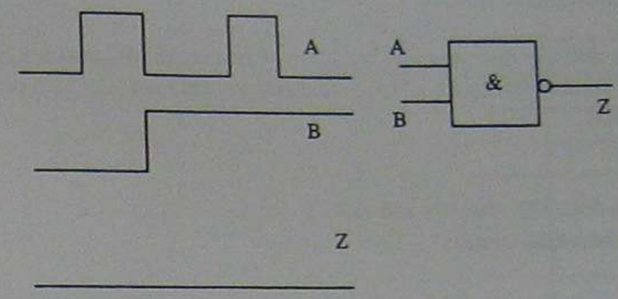
Function: _____

(b)



Function: _____

7. Sketch the output waveform for the logic circuit below.



8. Implement the following Boolean expressions without simplification.

(a) $(A + B) . C$

(b) $\overline{(A.B)} + C$

(c) $ABC + \overline{(A + C)} + (AB)$

(d) $A.\overline{(B.C)}$

9. Implement the logic circuit for the following truth table without simplification. A, B and C are the inputs and Z is the output of the circuit.

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

10. Implement a 2-input AND function using a 3-input AND gate.

Notes

Section 3: Universal Logic Gates

SUGGESTED DURATION	PREAMBLE
4 hours	This section introduces students to De Morgan's theorems and the universal nature of NAND and NOR gates.
This section covers part of learning outcome 2 of the National Module Descriptor.	

Objectives

At the end of this section, successful students will be able to:

- ☐ Explain the function of De Morgan's theorems.
- ☐ Use De Morgan's theorems to develop NAND gate equivalents for the AND, OR, NOT and NOR gates.
- ☐ Sketch two methods of connecting a NAND gate to perform the NOT function.
- ☐ Use De Morgan's theorems to develop NOR gate equivalents for the AND, OR, NOT and NAND gates.
- ☐ Sketch two methods of connecting a NOR gate to perform the NOT function.
- ☐ Convert a basic combinational logic circuit comprising AND, OR and NOT gates to either NAND or NOR gates.

Review Questions

1. Complete the following Boolean expressions (which are also known as De Morgan's theorems).

   a) $\overline{A.B}$ = $\overline{A} + \overline{B}$

   b) $\overline{A + B}$ = $\overline{A} . \overline{B}$

2. What is the importance of De Morgan's theorems?

   _____

   _____

3. What is the Boolean expression for the output (Z) of the circuit in Figure 1?

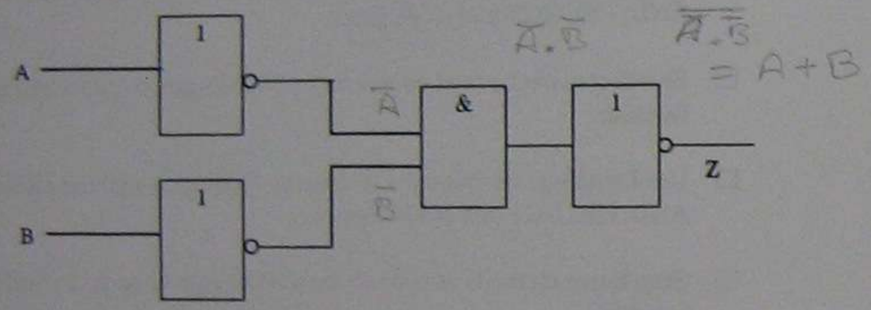


Figure 1

Z = OR

4. What type of gate is equivalent to the circuit in figure 1?

   _____

5. Show two ways of connecting a 2-input NAND gate to perform the inverter function.

6. Show two ways of connecting a 2-input NOR gate to perform the inverter function.

7. Show how a 2-input AND function can be achieved with 2-input NAND gates.

8. Show how a 2-input OR function can be achieved with 2-input NAND gates.

9. What type of gate is equivalent to the circuit of figure 2?

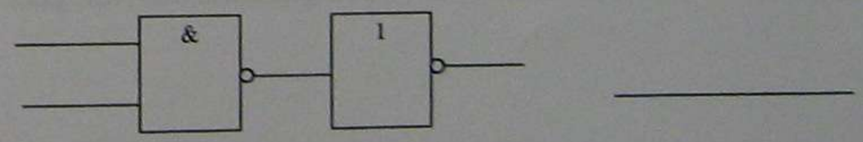


Figure 2

10. What is the equivalent of an **odd** number of inverters connected one after the other?

    _____

11. What is the equivalent of an even number of inverters connected one after the other?

_____

12. Show how the circuit of figure 3 can be implemented with NAND gates.

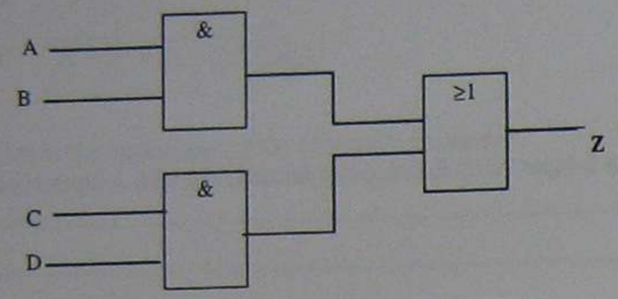


Figure 3

13. Show how a 2-input AND function can be implemented with NOR gates.

14. Show how a 2-input OR function can be implemented with NOR gates.

15. Show how the circuit of figure 4 can be implemented with NOR gates.

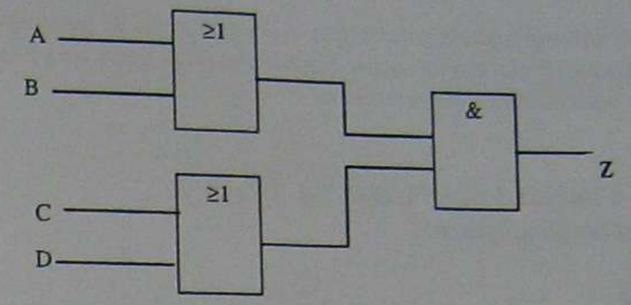


Figure 4

Week 4 (7)

(1)

Karnaugh map

Simplified sum of Products Boolean Expressions.

4 or less variable

Each square → min term (or) max term.

n variables ⟶ 2 square.

Product terms are denoted by 1's

Sum term ⟶ 0

0 are often left blank.

A K map consist of a grid of Squares, representing one canonical minterm combination of ~~the~~ variable or their inverse.

The map is arranged so that squares representing minterms which differ by only one variable are adjacent ~~vertically~~ both Vertically and horizontally.

$X Y' Z'$ would be adjacent to $X' Y' Z'$ & would be adjacent to $X Y' Z$ and $X Y Z'$

minimization Technique

$$X + X' = 1$$

expression → Sum of product form.

The function is mapped on to K map by marking a 1 in those squares corresponding to the terms in the expression to be simplified. The other squares may be filled with 0's

- Pairs of 1's on the map which are adjacent are combined by using $Y(X + X') = Y$
- minimization procedure consist of recognizing those pairs & multiple pairs.

$A + A\bar{B} + \bar{A}B$

$B = 0, 1$ $A = 0, 1$	$0 + 0 \times 0 + \bar{0} \cdot 0$ $0 + 0 + 1 = 1$
$B = 1$ $A = 1$	$1 + 1 \times 1 + \bar{1} \cdot 1$ $1 + 1 + 0 = 1$

$B = 0$, $A = 0$ $\longrightarrow 1$

$B = 1$, $A = 1$ $\longrightarrow 1$

There are circled indicating reduced terms.

Group which can be circled are those which have 2 (2^1) 1's four (2^2) 1's eight (2^3) 1's and so on.

note that because squares on one edge of the map are considered adjacent to those on the opposite edge groups can be formed with those squares.

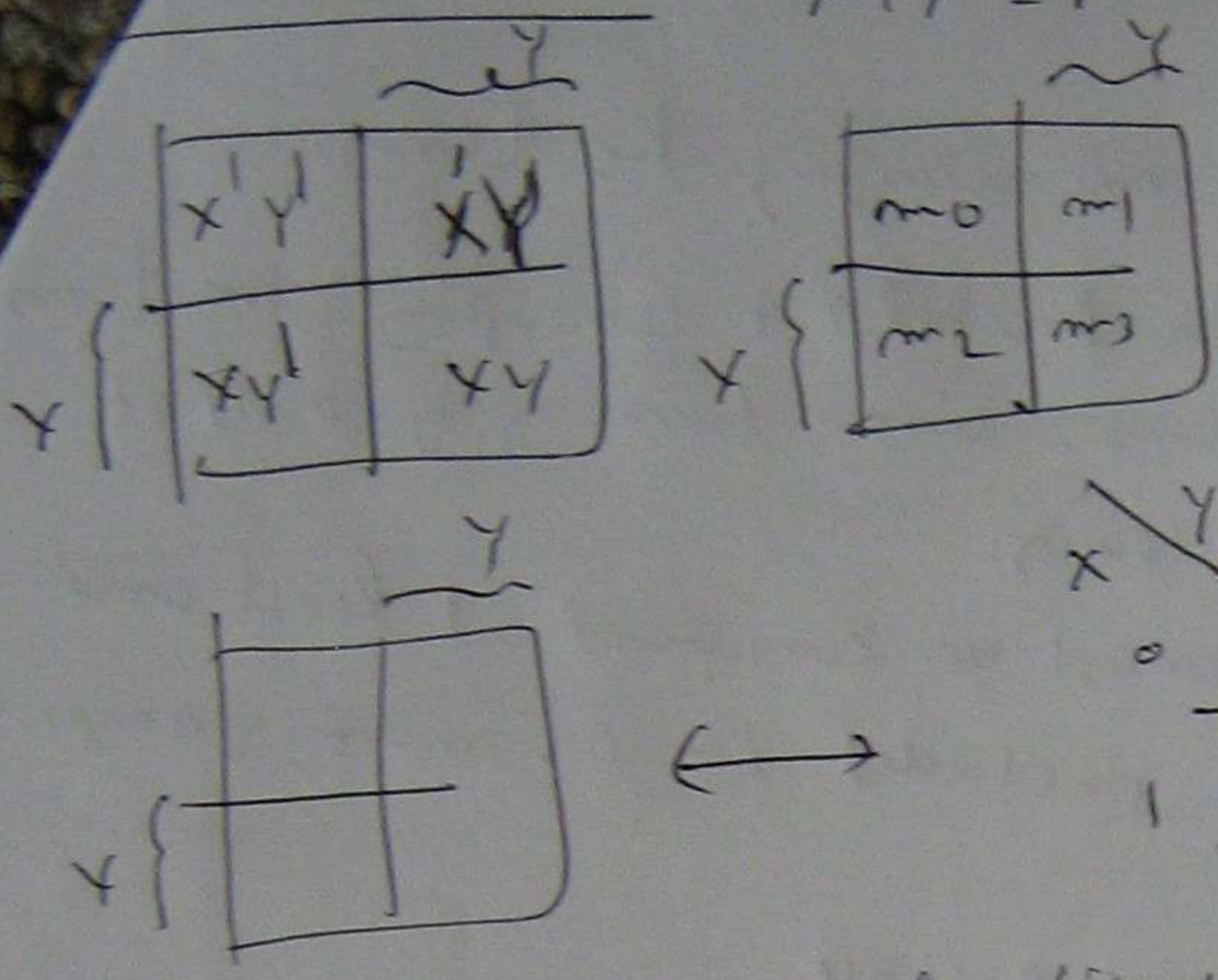
- Groups are allowed to overlap.

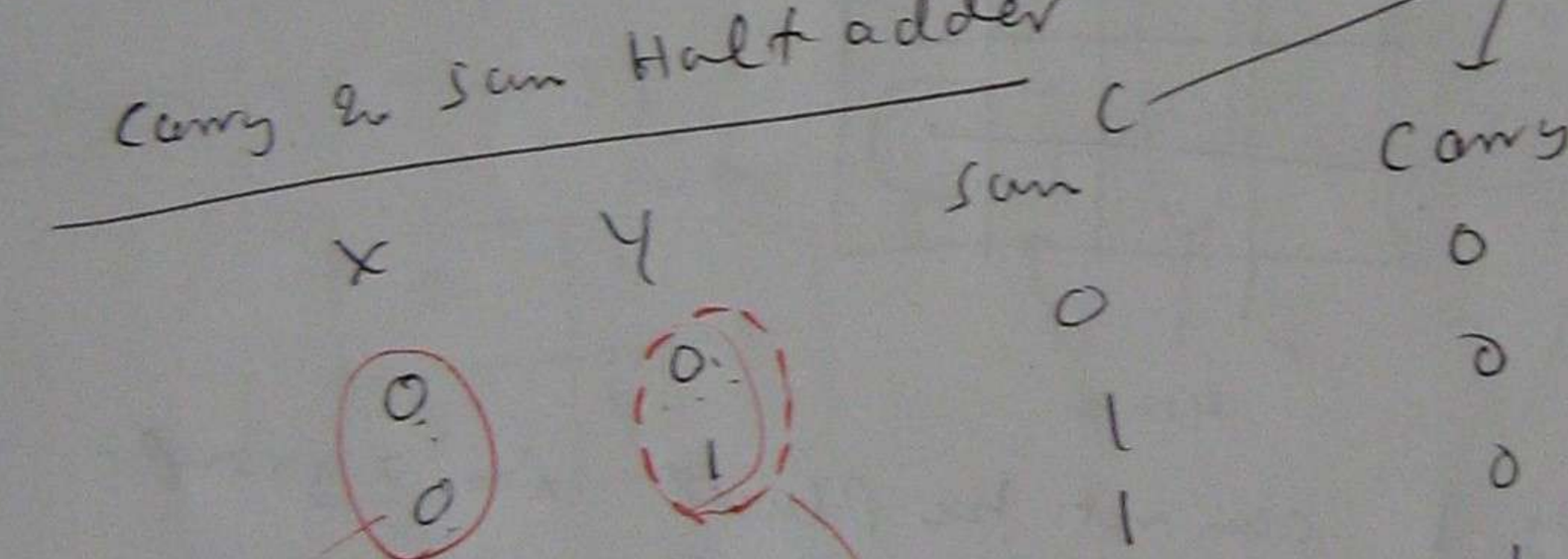
The objective is to cover all the 1's on the map in the fewest number of groups and to create the largest groups to do this.

- Once all possible groups have been formed, the corresponding terms are identified.
- A group of two 1's eliminate one variable from the original min term.
- A group of 4 1's eliminates two variables from original min term
- A Group of 8 1's eliminates three variables from the original min term and so on.
- The variables eliminated are those which are different in original min terms of the group

Karnaugh map

$x + x' = 1$

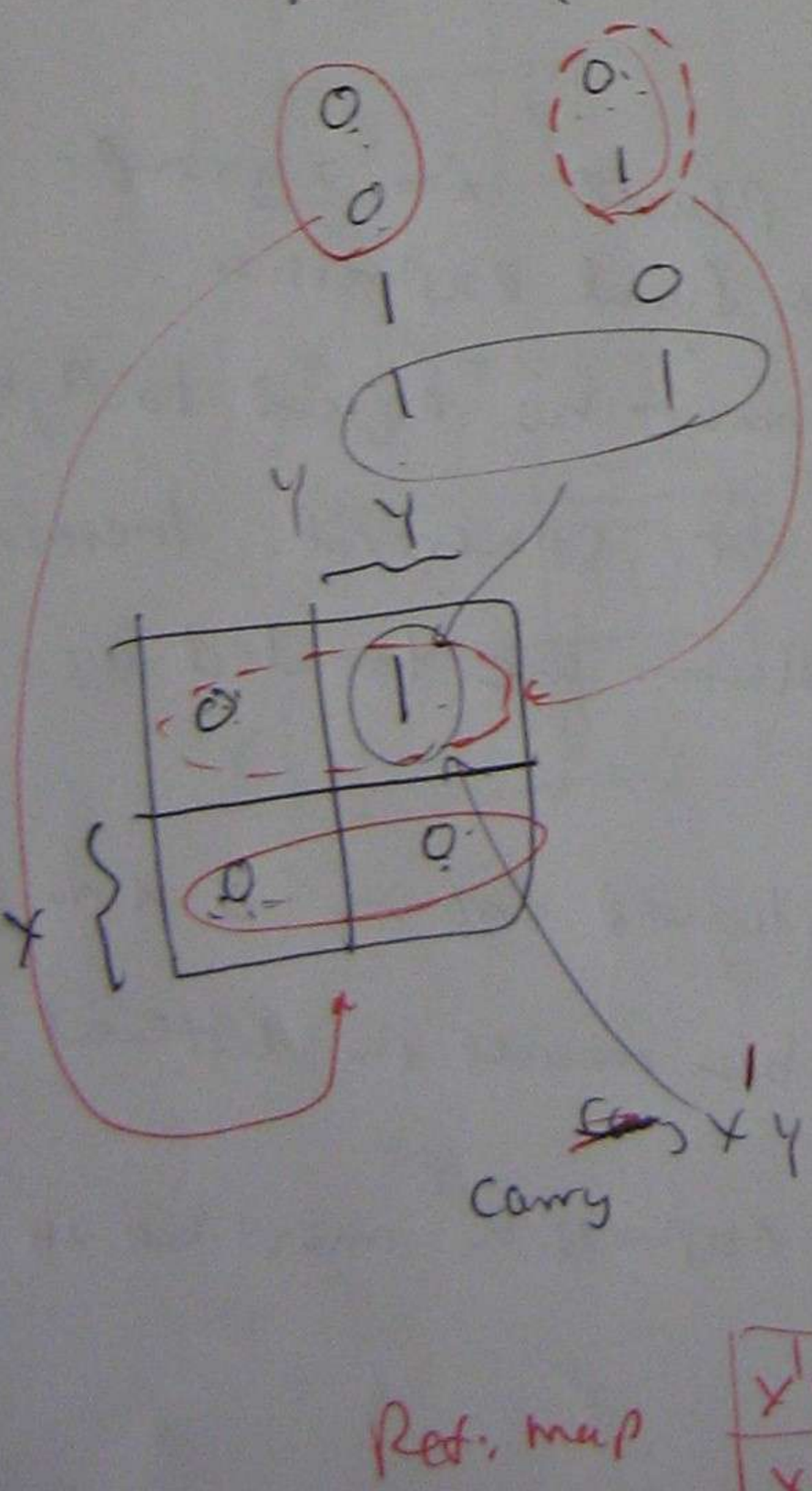




The figure is with changed position of m0, m1, m2, m3

The K map for the function is specified by putting a 1 in the square corresponding to a min term, or '0' otherwise.

Carry & Sum Half adder



2 outputs functions

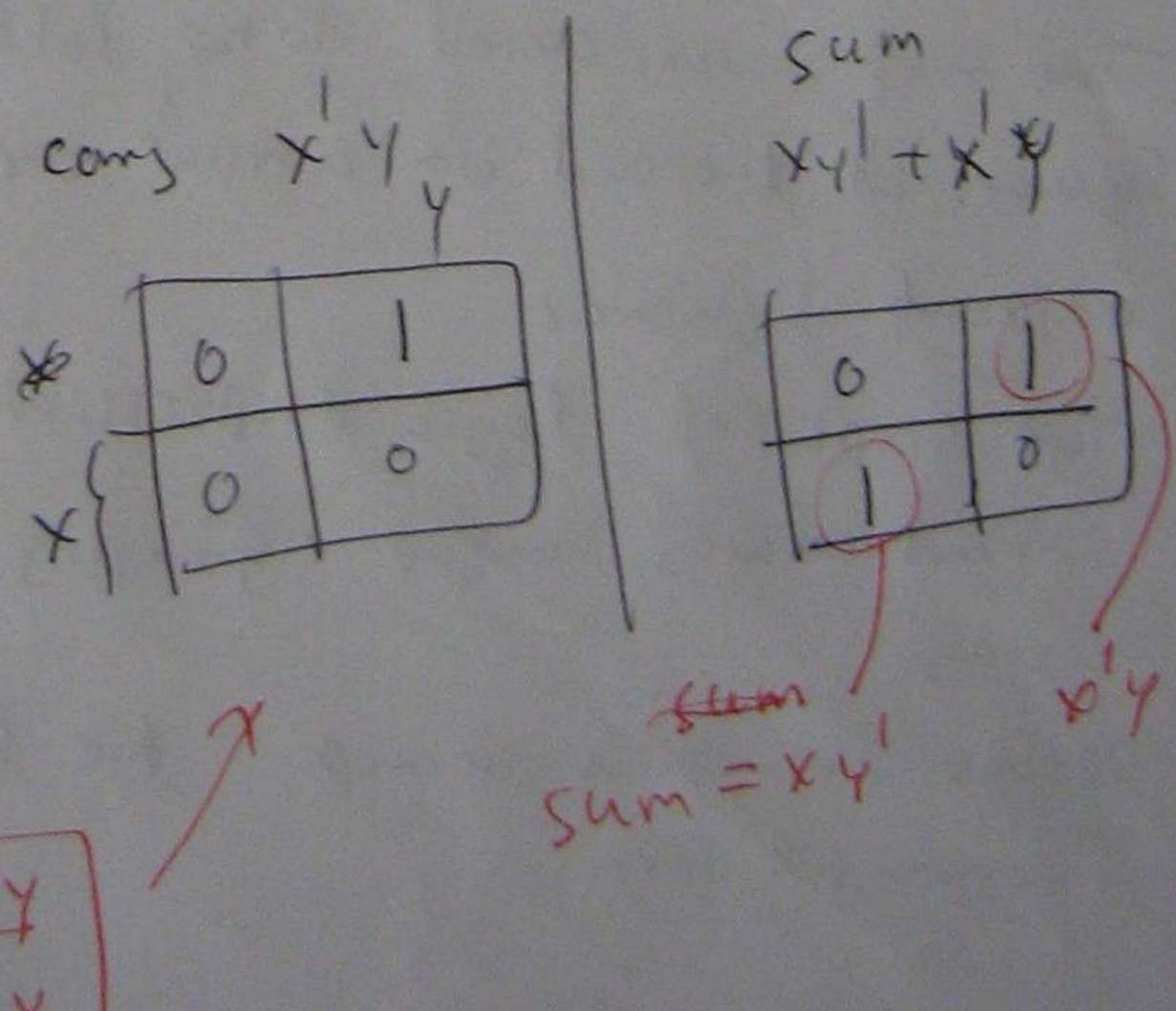
2 K maps

K map for Carry

Sum

$Carry = x' \cdot y$

$Sum = xy' + x'y$



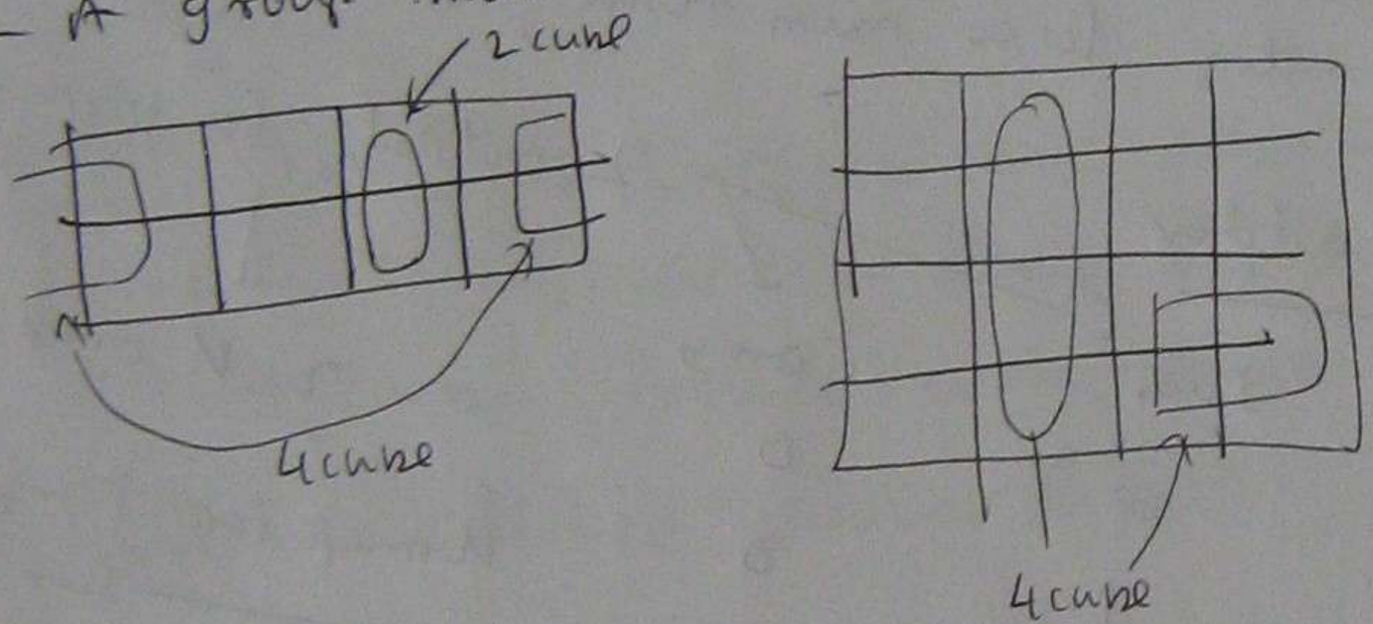
Red. map

Grouping cells in K maps

The power of K maps is minimizing the terms.

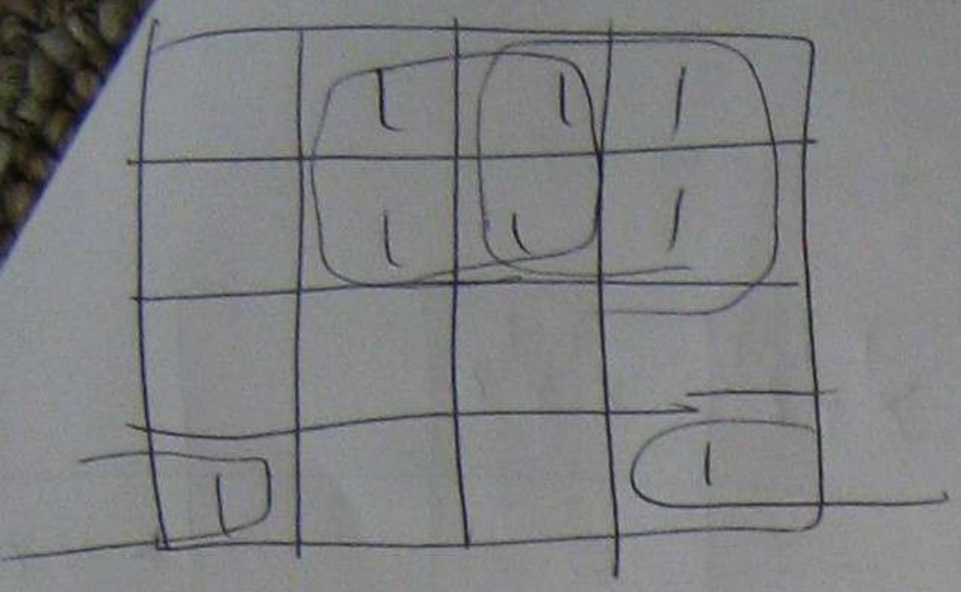
K maps can be minimized with the help of grouping the terms to form ~~sigma sigma~~ single terms. When forming groups of squares, observe / consider the following.

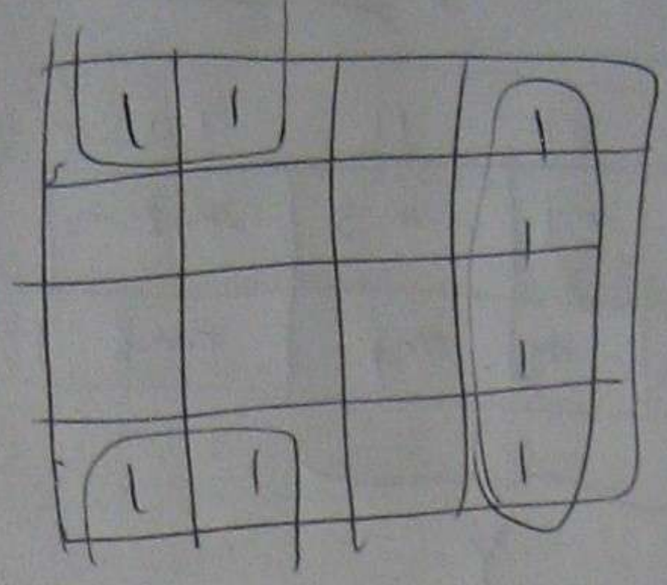
- Every square containing 1 must be considered at least once.
- A square containing 1 can be included in as many groups as desired
- A group must be as large as possible

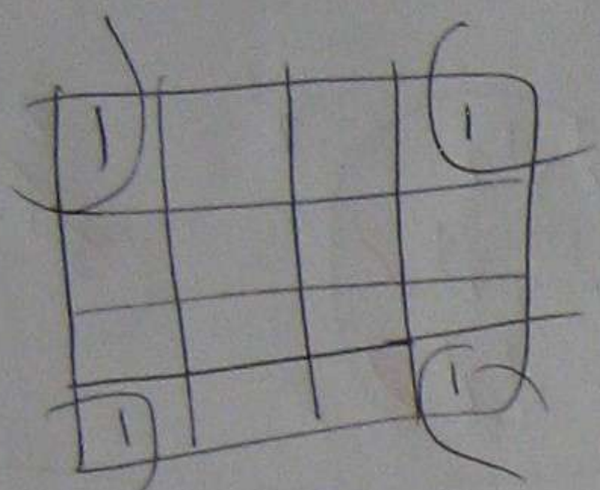


- If a square containing 1 cannot be placed in a group then leave it out to include in final expression.
- The number of squares in a group must be equal to 2, 4, 8
- The map is considered to be folded or spherical. Therefore squares at the end of a row or column are treated as adjacent squares.
- The simplified logic expression obtained from a K map is not always unique groups can be made in different ways.
- Before drawing a K map the logic expression must be in canonical form

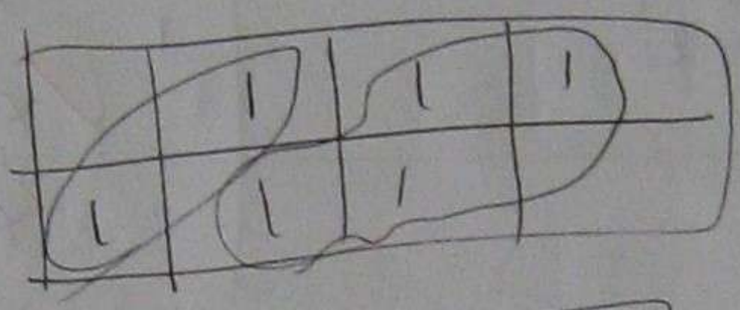
5



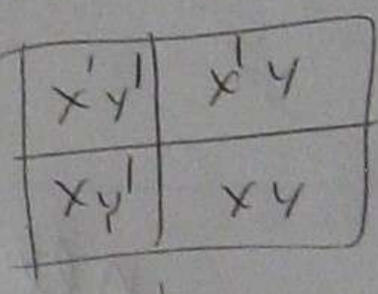


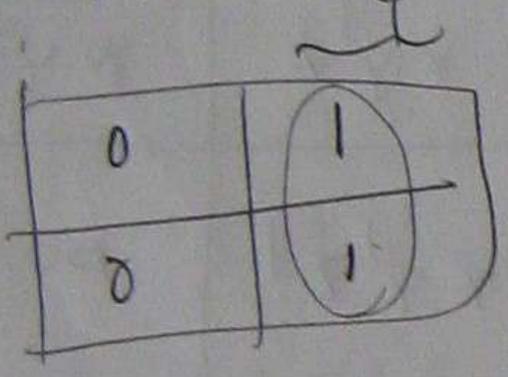


Invalid group



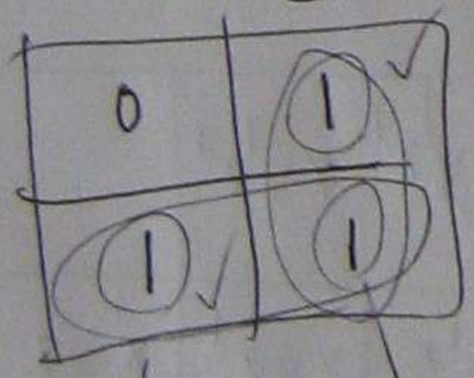
Exaple $F = x'y + xy$





$F = y$

Exaple $x'y + xy + xy'$



$x'y$

$F = x + y$

eliminate

xy'

Select $xy' + x'y$

eliminate

$x + y$

3 variables K map

z \ xy	00	01	11	10
0	m0	m1	m3	m2
1	m4	m5	m7	m6

Y

$x'y'z'$	$xy'z'$
$x'y'z$	xyz'
$x'yz$	
xyz	

$x'yz'$	NO 1	2	3
1	xyz ✓	$x'y'z$ ✓	$x'y'z'$ ✓
1	$x'yz$ ✓	$xy'z'$ ✓	
	$xy'z$ ✓		
	xyz' ✓		

8 combination.

	1			
nll'	$x'yz$	$xy'z$	xyz'	
$x'y'z$	$x'yz'$	$xy'z'$	NO 1	

2

	xyz'	$x'yz$
	$xy'z$	
$xy'z$	~~xyz'~~	xyz

$x'y'z'$	$xy'z'$		
$xy'z$	xyz'	$x'yz$	xyz

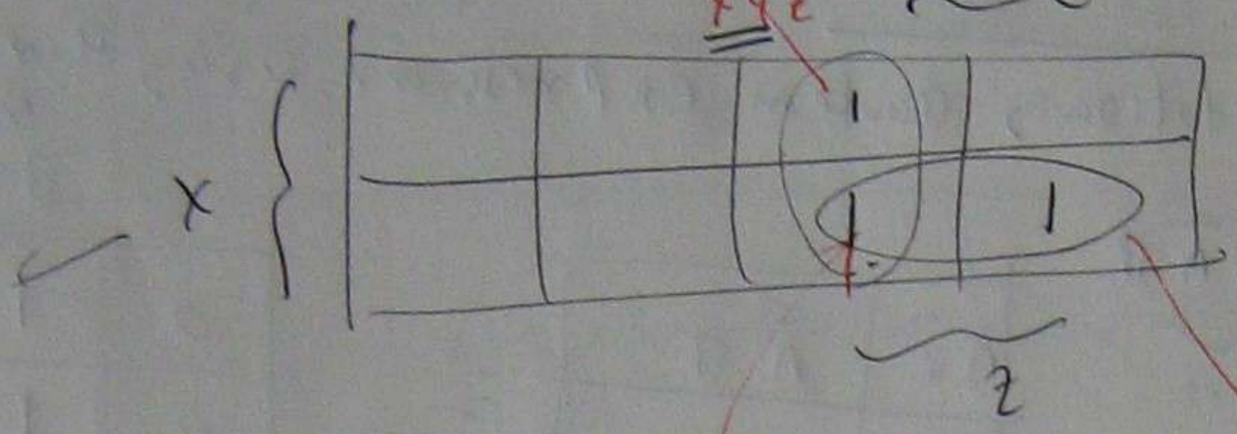
$x'y'z'$	$x'y'z$	xyz'	
$x'yz'$	$xy'z'$	xyz	
$x'yz$	$xy'z$		

$x'y'z'$	$x'y'z$	xyz'
$x'yz'$	$xy'z'$	xyz
$x'yz$	$xy'z$	

✓ $x'y'z'$	$x'y'z$	$xy'z'$ ✓	$xy'z$ ✓
$x'yz'$	$x'yz$	$x'yz$ ✓	xyz ✓

⑦

$F = xyz' + xyz + x'yz$



Select xyz'

$+$

$x'yz$

$= xy + yz$

eliminate

eliminate $x \rightarrow x'yz$

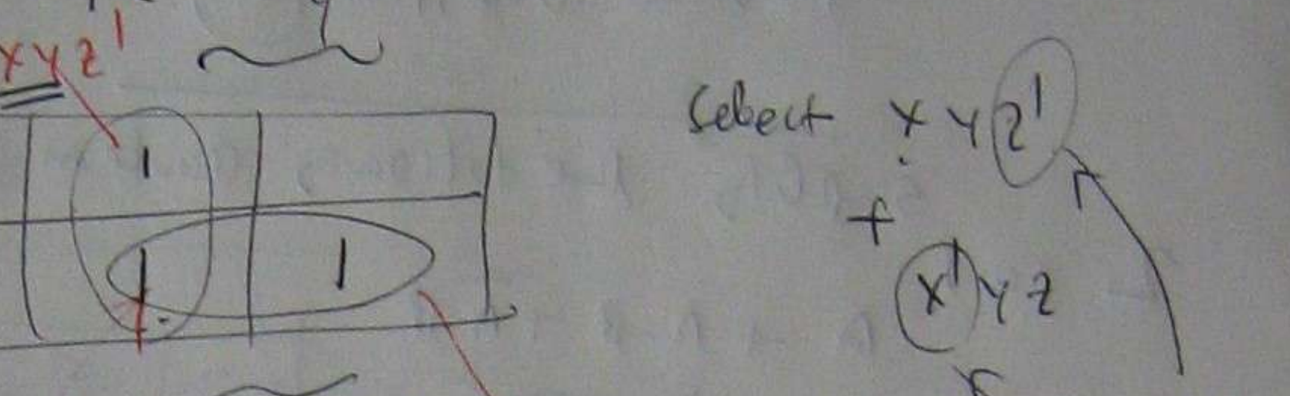
$\therefore F = xy + yz$ //

Planar

$F(x,y,z) = \Sigma(1, 3, 4, 5, 6, 7)$

$2 \mid 11$

0

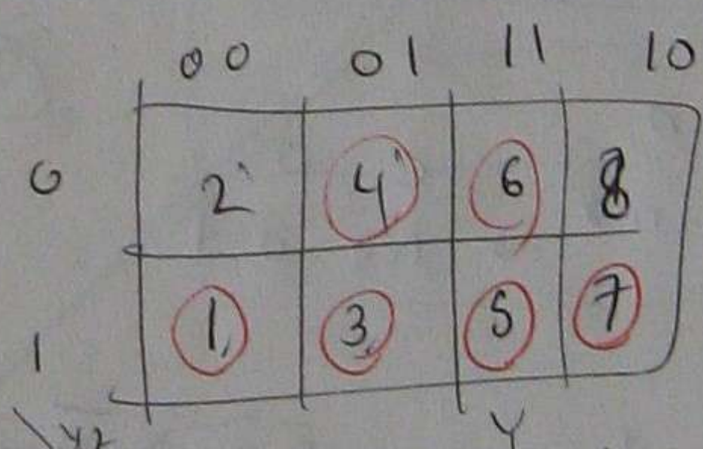


$00 = 2^0 + 2^0 = 2$

$01 = 2^0 + 2^1 = 3$

$11 = 2^1 + 2^1 = 4$

$10 = 2^1 + 2^0 =$



$F: xy + yz$

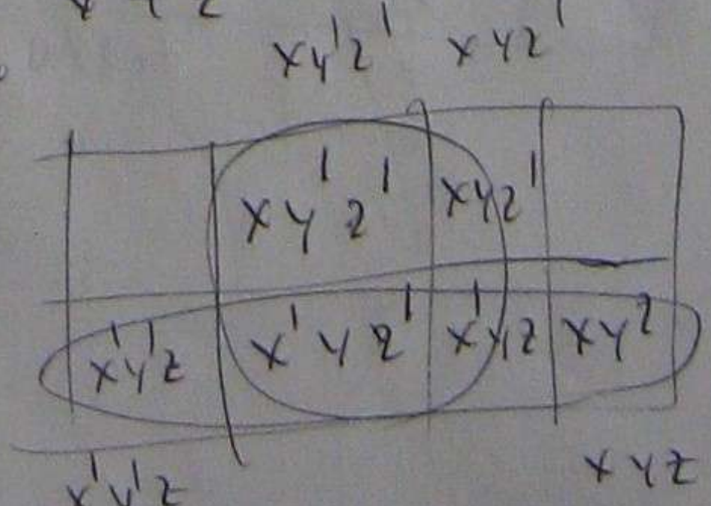
$= y + z$

Select $x'y'z + xyz + xy'z' + xyz'$

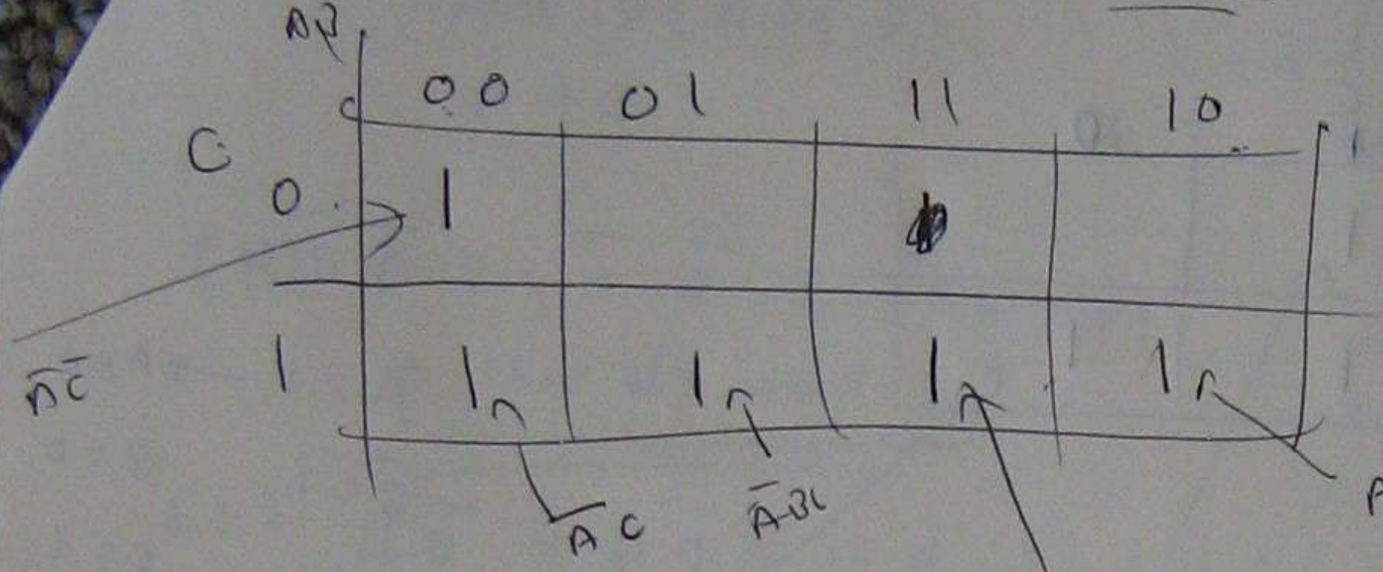
common factor

$z + xyz + xy$ $\times$ Select

Select $z + x$



2(d) $\bar{A}\bar{C} + A.B.C + A.C + \bar{A}.B.C + \bar{A}.C$



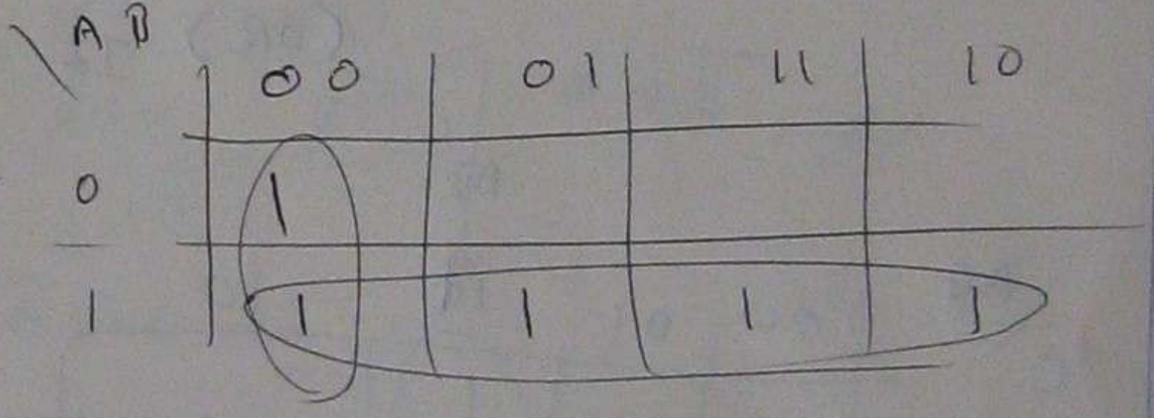
$\bar{A}\bar{C} \Rightarrow \bar{A} = 1$
$\therefore A = 0$
$\bar{C} = 1$
$C = 0$
$B = 0$

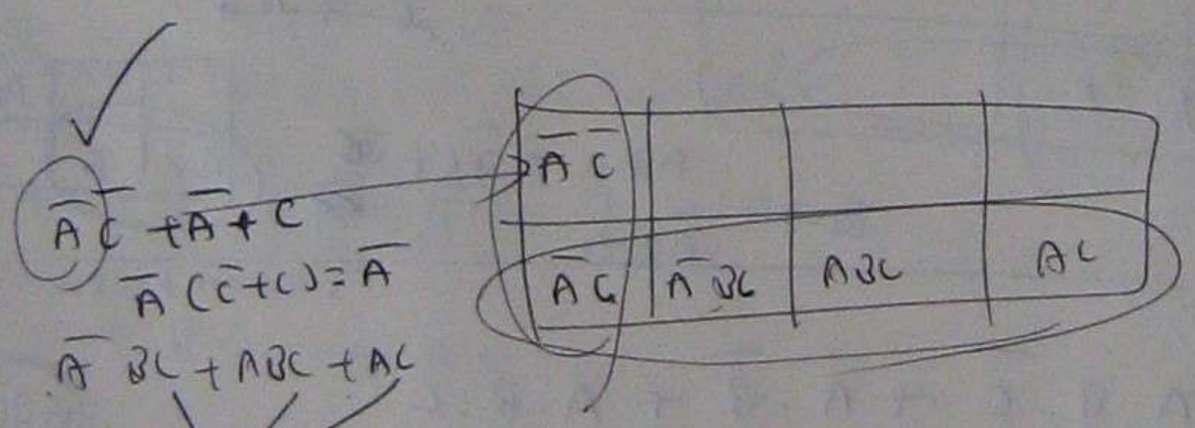
$ABC = A = 1$
$B = 1$
$C = 1$

$A.C \Rightarrow A = 1$
$C = 1$
$B = 0$

$\bar{A}C \rightarrow \bar{A} = 1 \therefore$
$\therefore A = 0$
$C = 1$
$B = 0$

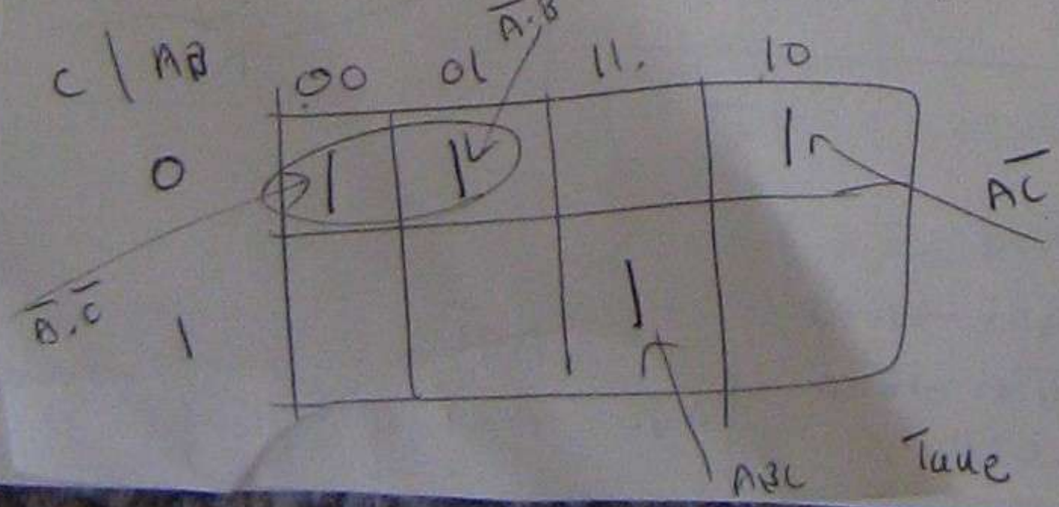
$\bar{A}BC = \bar{A} = 1$
$A = 0$
$B = 1$
$C = 1$





$C(\bar{A} + \bar{A}B + AB + A) = C(\bar{A} + A + \bar{A}B + AB) = C(\bar{A}B + AB) = C(B(\bar{A} + A)) = BC$

2(e) $\bar{A}.B + A.B.C + \bar{A}\bar{C} + A\bar{C}$



$\bar{A}.B = \bar{A} = 1$
$A = 0$
$\therefore B = 1$
$C = 0$

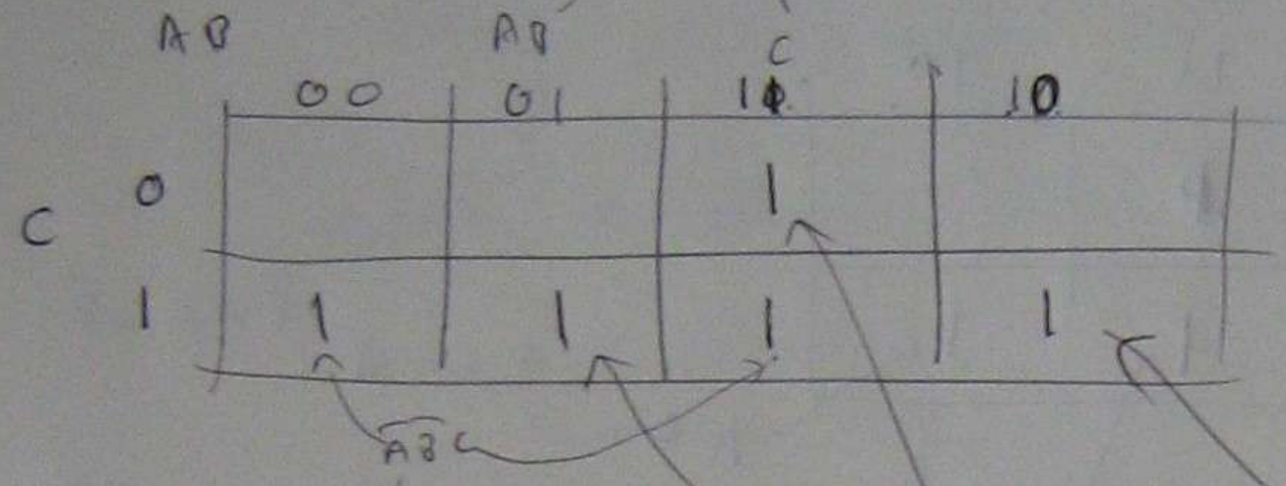
$A.B.C \rightarrow A = 1$
$B = 1$
$C = 1$

$\bar{A}.\bar{C} \rightarrow \bar{A} = 1$
$\therefore A = 0$
$\bar{C} = 1$
$\therefore C = 0$
$B = 0$

$A\bar{C} \rightarrow A = 1, \bar{C} = 1$
$C = 0$
$\therefore B = 0$

Take $\bar{A}.\bar{C} + \bar{A}.B = \bar{A}(B + \bar{C})$ $\bar{A} = 1 \therefore B + \bar{C}$

2(b) $A.B + \overline{A}\,\overline{B}\,C + AC + BC$



$A.B = A=1,\ B=1,\ C=0$

$AC = A=1,\ B=0,\ C=1$

$BC = A=0,\ B=1,\ C=1$

$\overline{A}\,\overline{B}\,C \rightarrow A=1 \therefore \overline{A}=0,\ B=1 \therefore \overline{B}=0,\ C=1$

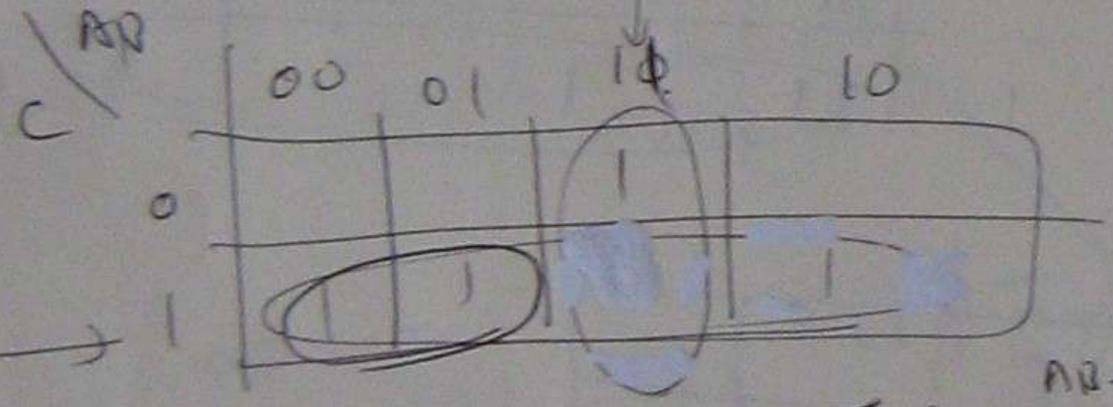
$\overline{A}=1 \rightarrow A=0,\ \overline{B}=1 \rightarrow B=0,\ C=1$

(or) all ABC

Inverted

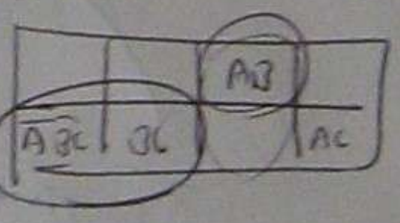
$A=1,\ B=1,\ C=1$

(OR) If $A=0 \Rightarrow \overline{A}=1,\ B=0 \Rightarrow \overline{B}=1,\ C=0 \Rightarrow \overline{C}=1$



Take A

$= C + AB$

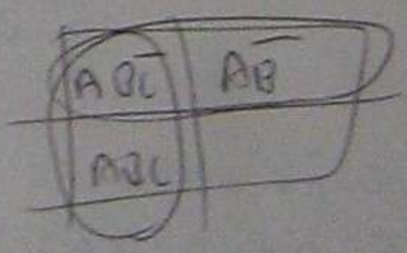


Take $AB + \overline{A}\,\overline{B}C + B\overline{B}$

$AB + C$

common factor $= C$

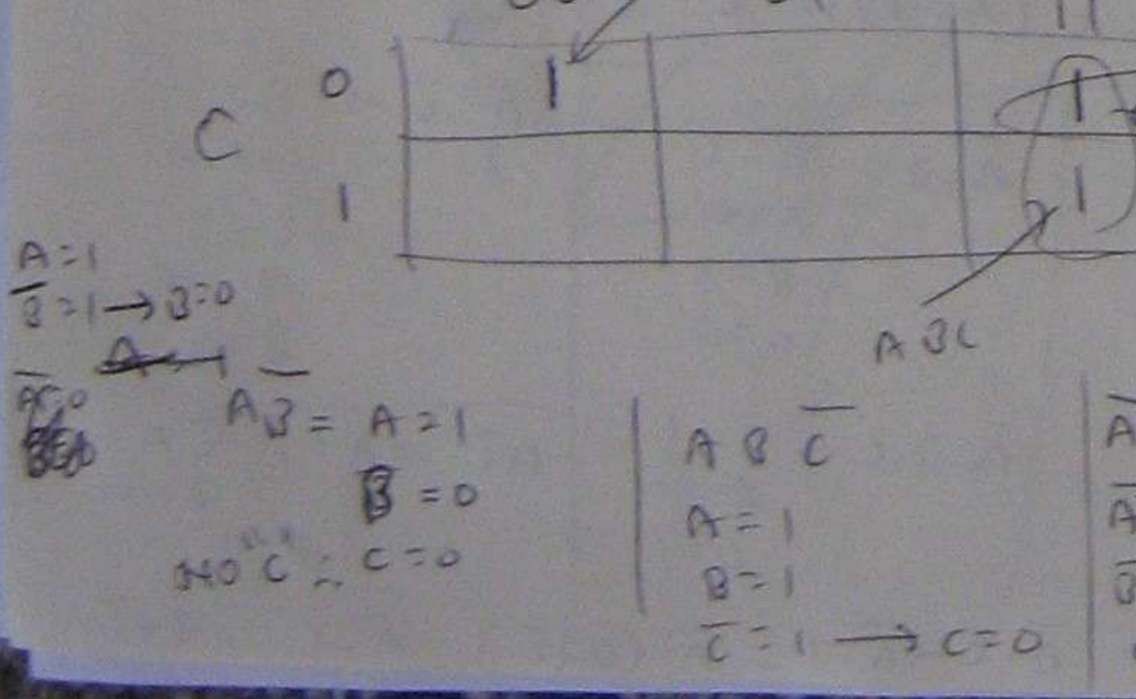
2(c) $A\overline{B} + A.B.\overline{C} + \overline{A}.\overline{B} + A.B.C$



Take $A\overline{B}\overline{C} + AB\overline{C} + A\overline{B}$

$A + \overline{B}$

$A\overline{B}$



$A=1$

$\overline{B}=1 \rightarrow B=0$

$\overline{A}=0$

$A\overline{B} = A=1,\ B=0,\ C=0$

$A\overline{B}\overline{C} = A=1,\ B=1,\ \overline{C}=1 \rightarrow C=0$

$\overline{A}.\overline{B} = \overline{A}=1 \rightarrow A=0,\ \overline{B}=1 \rightarrow B=0,\ C=0$

$ABC = A=1,\ B=1,\ C=1$

simplify $B + A \cdot B + \bar{A} \cdot B$

2(a) $A + B(A + \bar{A})$ $\quad A + \bar{A} = 1$

$A + B$

(b) $A \cdot B + \bar{A} \cdot \bar{B} \cdot C + AC + BC$

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

Use K map

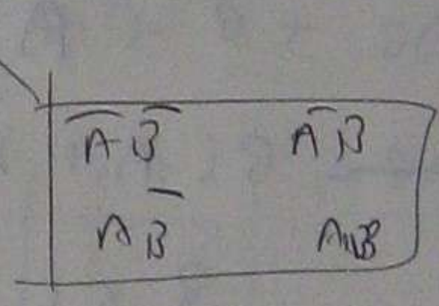
$AB + C$

(c) $A\bar{B} + AB\bar{C} + \bar{A} \cdot \bar{B} + ABC$

$\therefore A\bar{B} + \bar{A} \cdot \bar{B} + AB\bar{C} + ABC$

$\bar{B}(A + \bar{A}) + AB(C + \bar{C})$

$\bar{B} \times 1 + AB(1)$



$\bar{B} = 1 \quad A = 1$

$B = 0 \quad B = 1$

$A + \bar{B}$ //

(d) $\bar{A} \cdot \bar{C} + A \cdot \bar{B} \cdot C + AC + \bar{A}BC + \bar{A} \cdot C$

$\bar{A} \cdot \bar{C} + \bar{A}C + ABC + \bar{A}BC + AC$

$\bar{A}(\bar{C} + C) + BC(A + \bar{A}) + AC$

$\bar{A} + BC + AC$

$\bar{A} + C(A + B)$

Take common factor $\bar{A} + C$ //

// You take just comen factor

(e) $\bar{A} \cdot B + ABC + \bar{A} \cdot \bar{C} + A\bar{C}$

$\bar{A} \cdot B + ABC + \bar{C}(A + \bar{A})$

$\bar{A} \cdot B + ABC + \bar{C}$ $\quad B + \bar{C}$

$B(\bar{A} + AC) + \bar{C}$

(1) simplify following Boolen expression (12)

(a) $A + (A \cdot B) = A(1+B) = A$

(b) $A \cdot (\bar{A} + B) = A \cdot \bar{A} + AB$

$= 0 + AB = AB$

(c) $(A+B) \cdot (B+C)$

$AB + B \cdot B + \bar{A}C + B \cdot C$

$AB + B + \bar{A}C + BC$

~~AB~~ $B(A+1) + AC + BC$

$B \times 1 + AC + BC$

$B + AC + BC$

$AC + B + BC$

$AC + B(1+C)$

$AC + B \times 1$

$AC + B$ //

(d) $\overline{A \cdot B \cdot C} \cdot \overline{A\,B\,C} \quad \overline{A\,\bar{B}\,C}$

De morgan theorm

Invert Input $\overline{A\,B\,C} \quad \overline{A\,B\,C} \quad \overline{A\,\bar{B}\,C} = \overline{A\,B\,C} \quad \overline{A\,B\,C} \quad \overline{A\,\bar{B}\,C}$

Invert o/p & change design $= \overline{\overline{A\,B\,C}} + \overline{\overline{A\,B\,\bar{C}}} + \overline{\overline{A\,\bar{B}\,C}}$

$= \underline{ABC + AB\bar{C}} + A\,\bar{B}\,C$

$= AB(C + \bar{C}) + A\bar{B}C$

$= AB + A\bar{B}C$

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

$= AB + AC$

B/C	0	1
0	$\bar{B}\bar{C}$	$\bar{B}C$
1	$B\bar{C}$	BC

$\bar{B} = 1 \quad C = 1$

$B = 0$

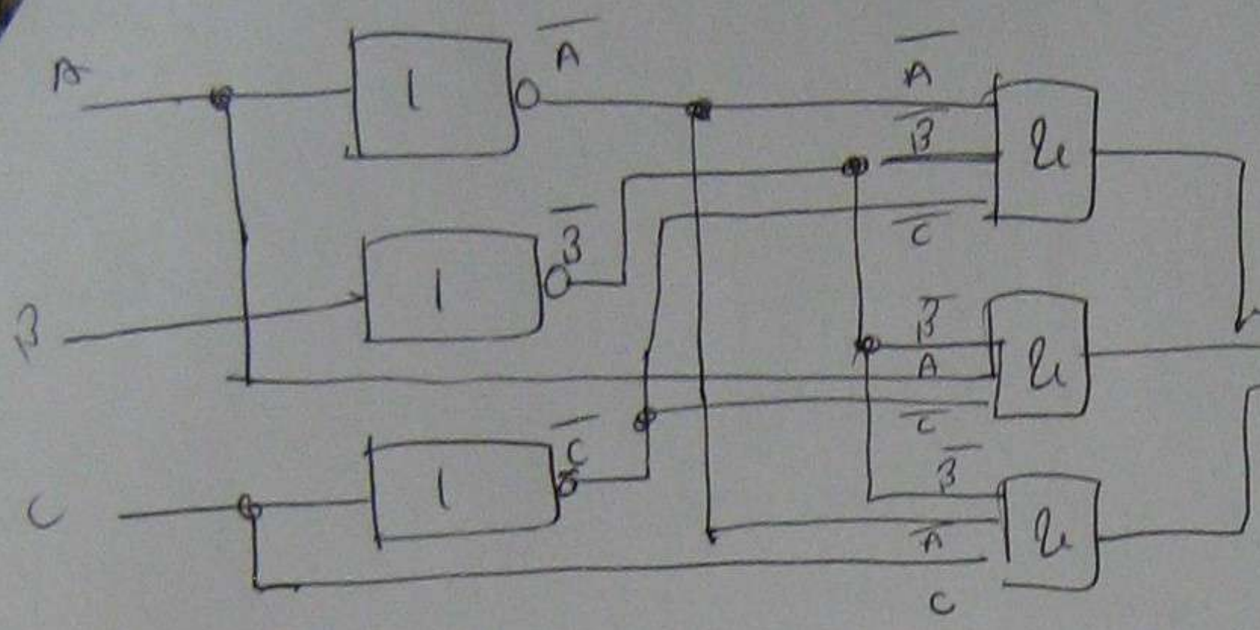
$B = B = 1$

$C = 0$

$\bar{B}C = \bar{B} = 1$

$B = 0 \; C = 1$

3) simplify & then redraw the ckt using 2 input NAND gates only



$\overline{A}\,\overline{B}\,\overline{C} + \overline{B}\,A\,\overline{C} + \overline{B}\,\overline{A}\,C$

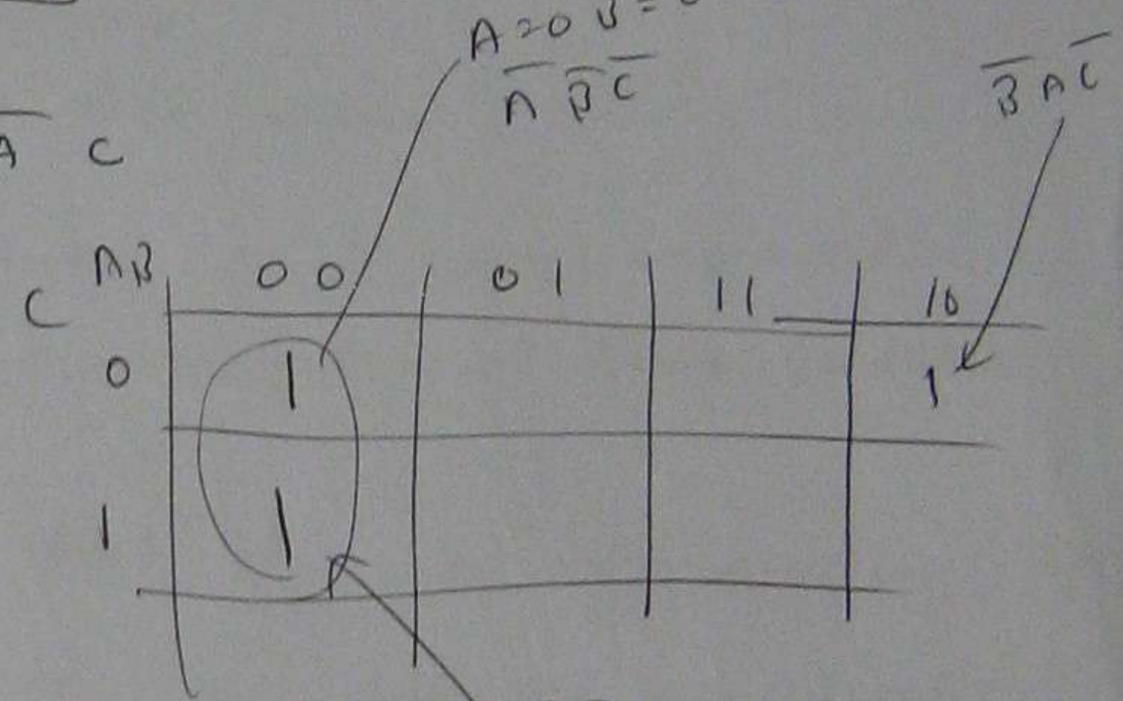
$\overline{B}\,\overline{C}\,(\overline{A}+A) + \overline{B}\,\overline{A}\,C$

$\overline{B}\,\overline{C} \times 1 + \overline{B}\,\overline{A}\,C$

$\overline{B}\,\overline{C} + \overline{B}\,\overline{A}\,C$

can't solve further

∴ use k graph



$\overline{A}\,\overline{B}\,\overline{C} \Rightarrow \overline{A}=1 \quad A=0$
$\overline{B}=1 \quad B=0$
$\overline{C}=1 \quad C=0$

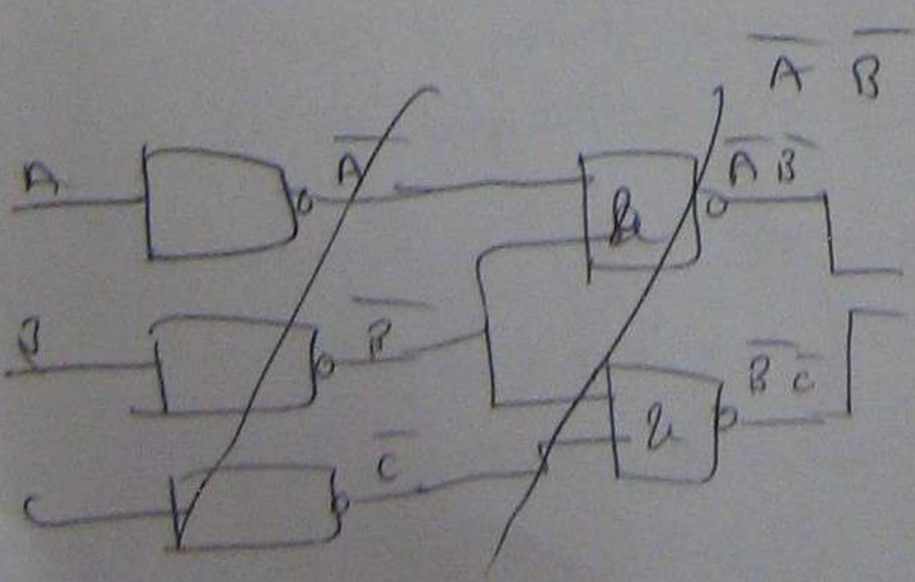
$\overline{B}\,A\,\overline{C} \Rightarrow \overline{B}=1 \rightarrow B=0$
$A=1$
$\overline{C}=1 \rightarrow C=0$

$\overline{B}\,\overline{A}\,C \rightarrow \overline{B}=1 \;\therefore B=0$
$\overline{A}=1 \;\therefore A=0$
$C=1$

$\therefore \overline{A}\,\overline{B}\,\overline{C} + \overline{B}\,\overline{A}\,C + \overline{B}\,A\,\overline{C}$

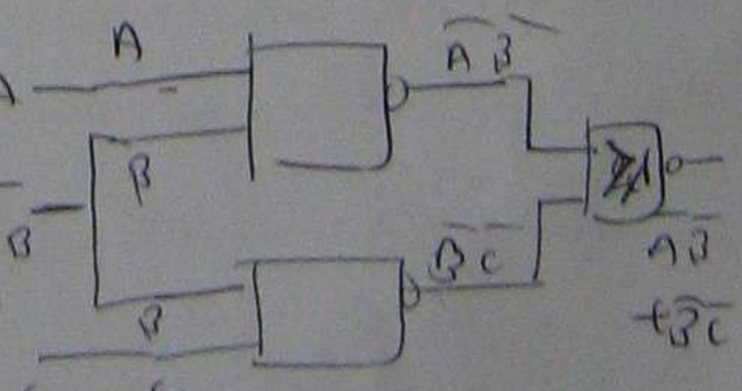
$\overline{A}\,\overline{B}\,(C+\overline{C}) + \overline{B}\,A\,\overline{C}$

$\overline{A}\,\overline{B} + \overline{B}\,A\,\overline{C}$



Take only

$\therefore \overline{A}\,\overline{B} + \overline{B}\,\overline{C}$



Week (1)

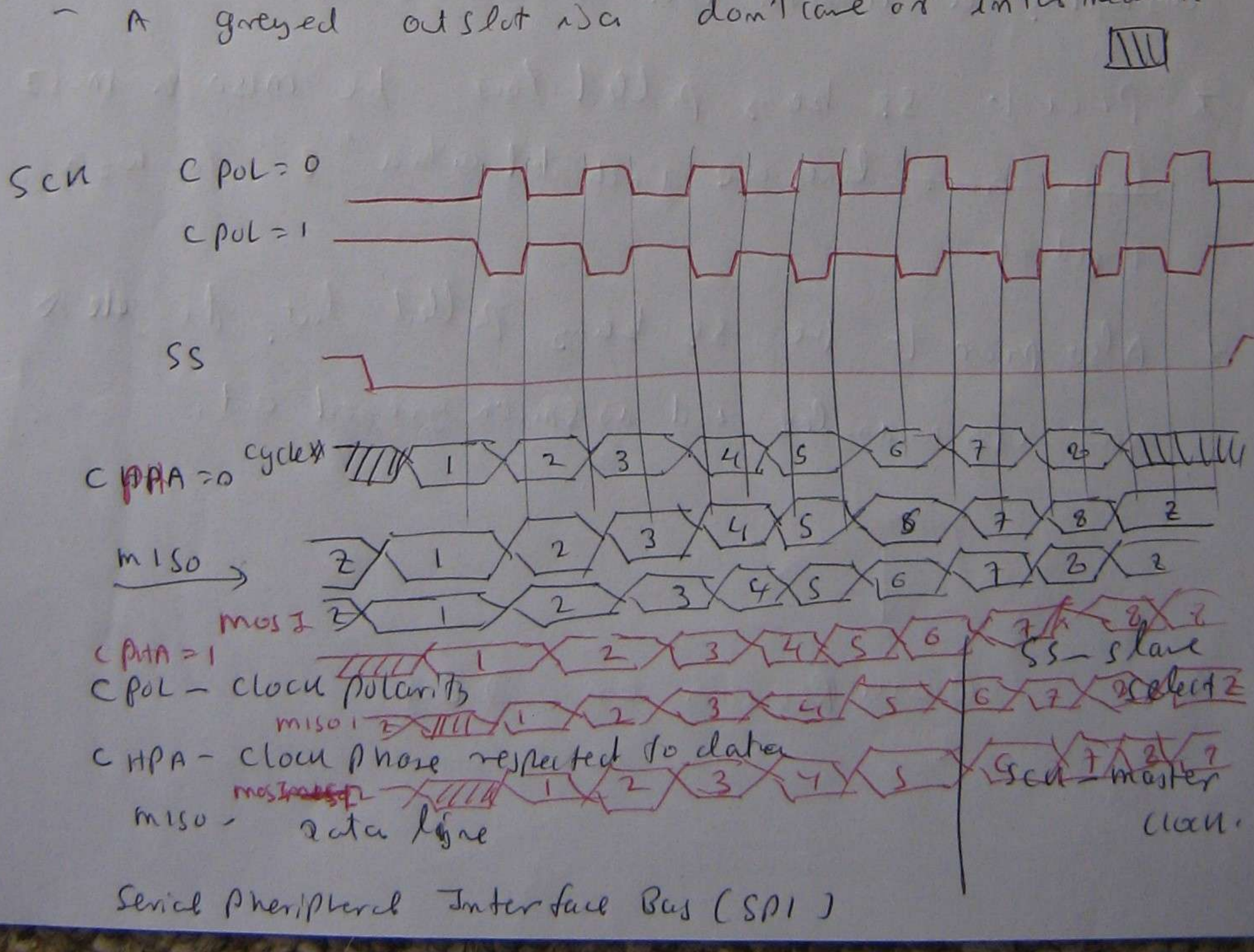
Timing diagrams

A digital Timing Diagram

A digital timing diagram is a representation of a set of signals in the time domain. A timing diagram can contain many rows, usually one of being the clock. It is a tool that is ubiquitous in digital electronics, hardware debugging and digital communication

most timing diagrams use the following conventions.

- Higher value is a logic one
- lower value is a logic zero
- A slot showing a high & low is an either or
- A 'Z' indicates high impedance
- A greyed out slot is a don't care or intermediate



Serial Peripheral Interface Bus (SPI)

SPI operates in the following way

- The master determines the appropriate CPOL & CPHA value
- The master pulls down the slave select (SS) line for a specific slave chip.
- The master clocks, sets at a specific frequency
- During each of a clock cycle, the transfer is full duplex
- The master writes on the MOSI line & reads the MISO line
- When finished, the master can continue with another byte transfer or pull SS high to end the transfer
- When a slave's SS line is high then both it's MISO and MOSI line should be high impedance so to avoid disrupting a transfer for a different slave.

* Prior to SS being pulled low, the MISO & MOSI ~~lines~~ ~~rows~~ lines are indicated with a "Z" for high impedance.
* Also prior to the SS being pulled low, the cycle now is meaningless and is shown greyed out.

+

Digital Electronics (2)

week (2)

Page 1 to 3

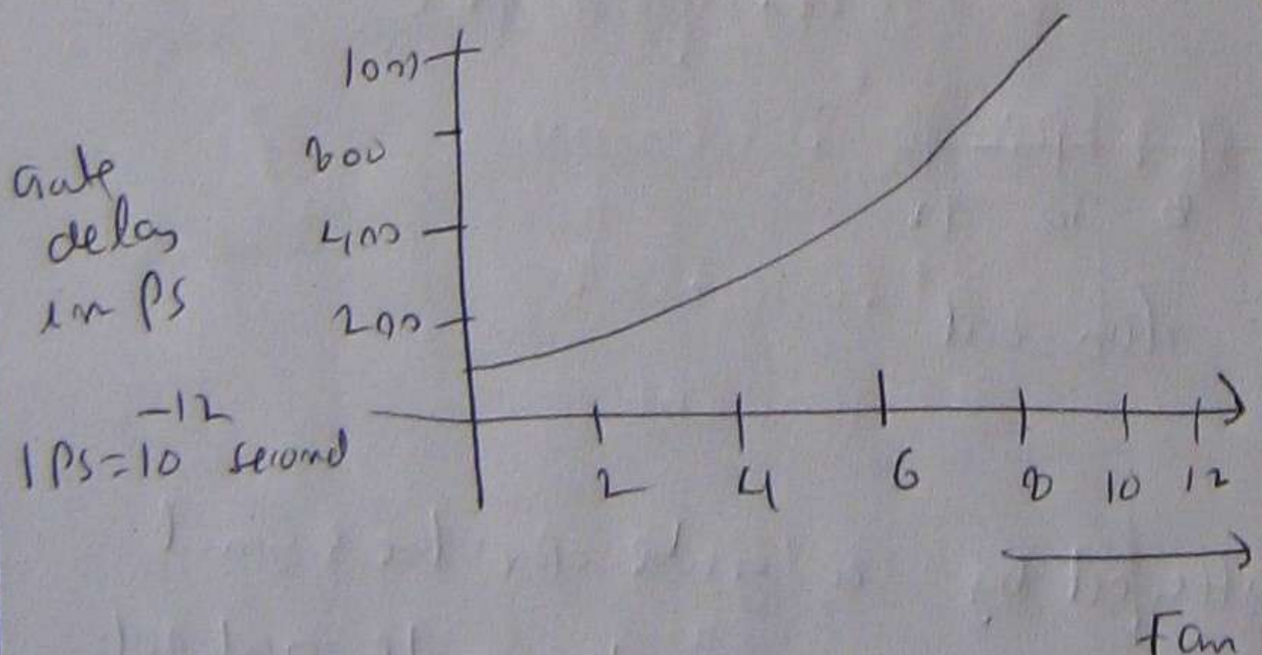
Operation of Discrete Equipments

Basic concept

Fan in
Fan-out
Noise margin
Power Dissipation
Gate delas
wire delas
Shear
voltage Threshold

Fan in is the number of inputs a gate has.

AND gate has ~~fan~~ fan-in of two
Three input NAND gate has fan-in of three.
NOT gate always has a fan-in of one.



Fan-out

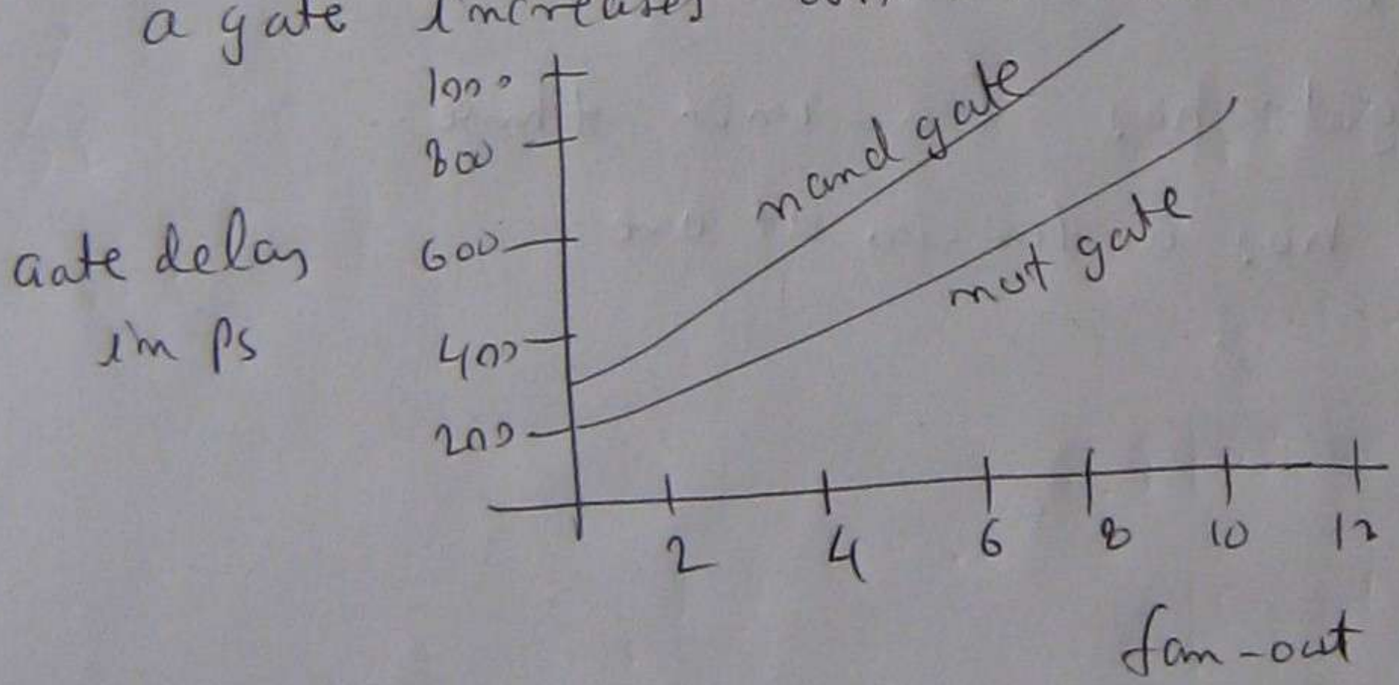
The number of gates that each gate can drive, while providing voltage levels in the guaranteed range. is called the standard load or fan out. The fan out really depends on the amount of electric current a gate can source or sink while driving other gates. The effect of loading a logic gate output with more than its rated

(2) wu2

fan-out has the following effects.

- In ~~low~~ LOW state, output voltage VOL may increase above VOL max
- In HIGH state, output voltage VOH may decrease below VOH min
- The operating temperature of the device may increase there by reducing the reliability of the device and eventually causing the device failure.
- output rise & fall times may increase beyond specifications.
- The propagation delay may rise above the specified value

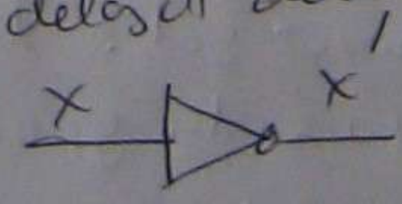
Normally as in the case of fan-in, the delay offered by a gate increases with the increase in fan-out



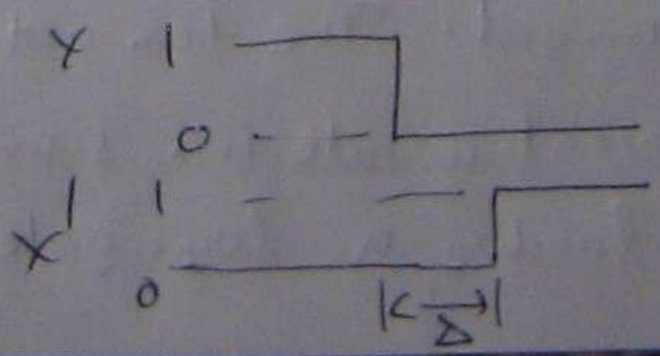
gate delay

gate delay is the delay offered by a gate for the signal appearing at its input. before it reaches the gate output

The figure shows a not gate with a delay of delta gate delay is also known as propagation delay



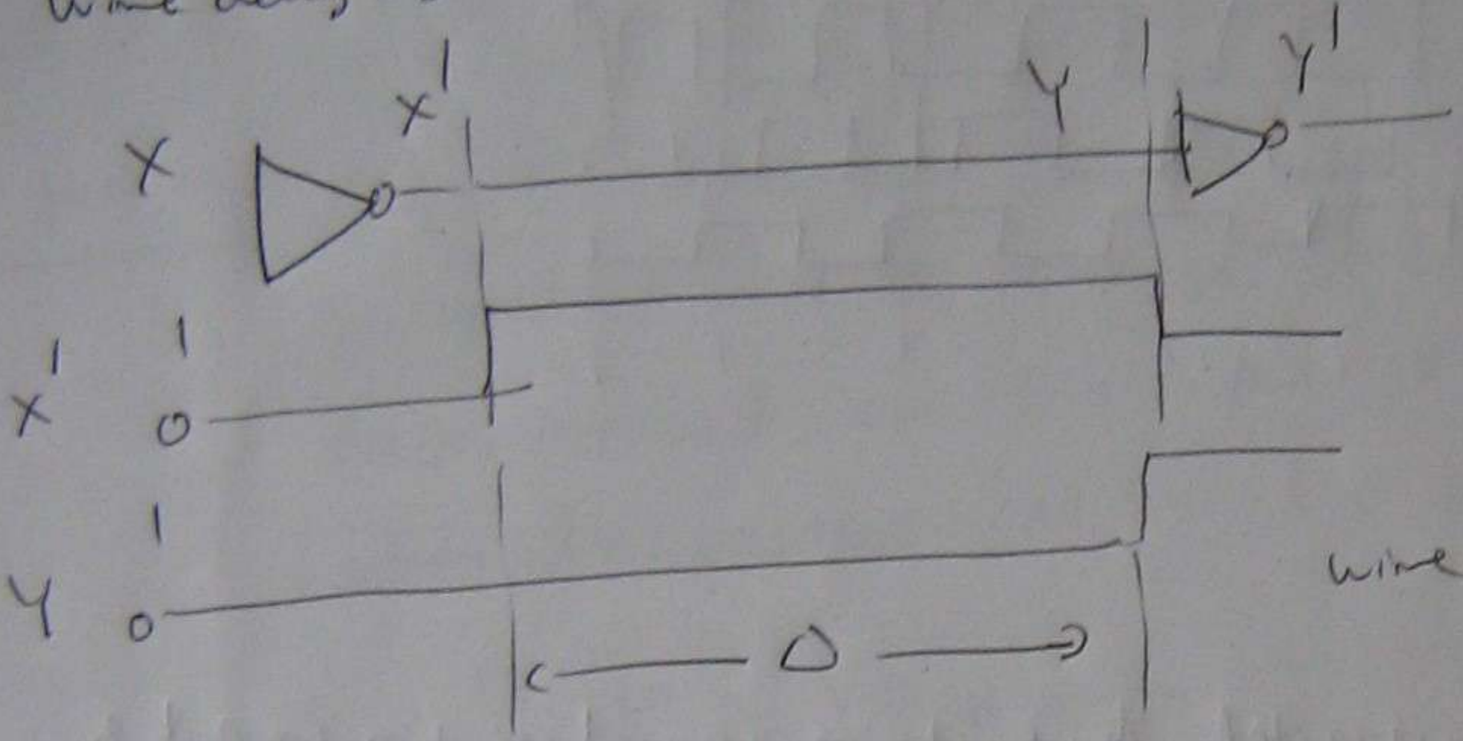
gate delay is rd the same for both transition.



Wire Delay

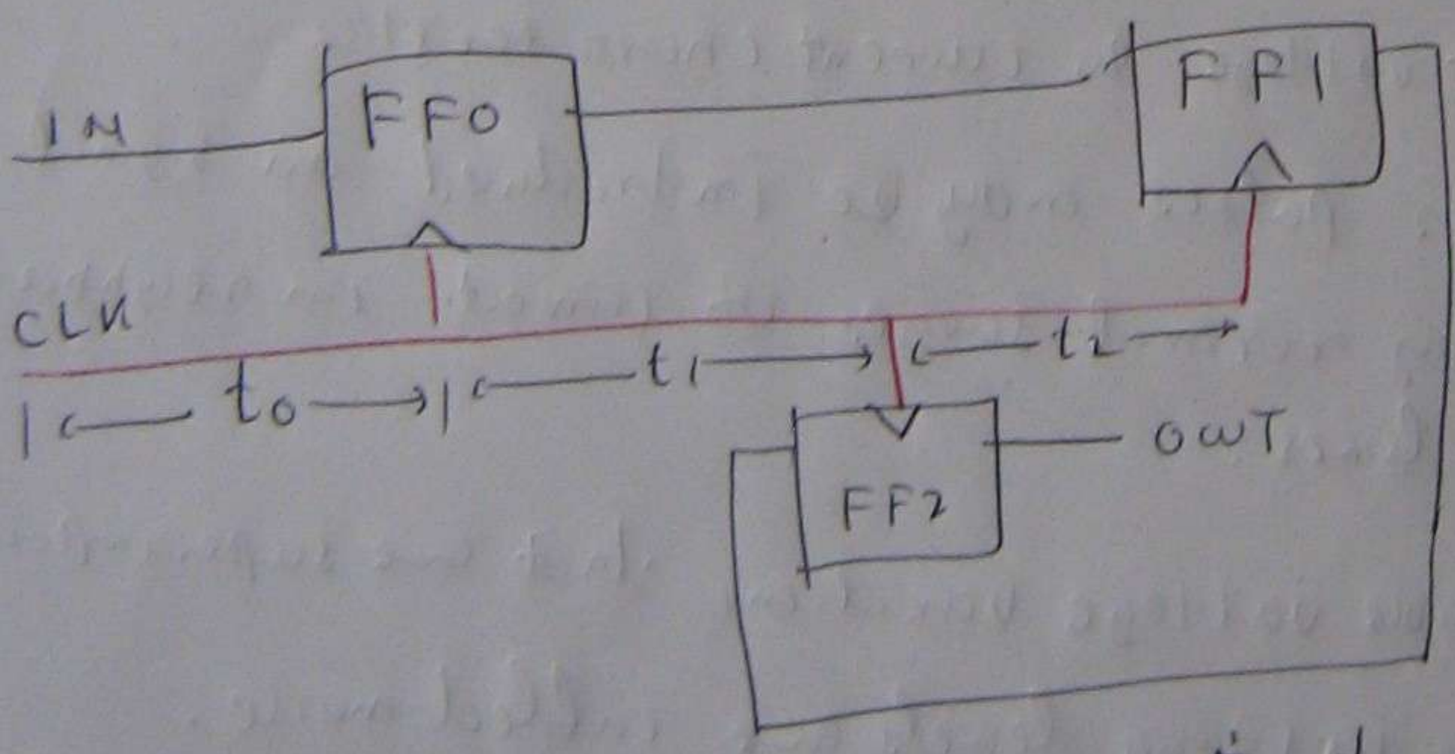
Gates are connected together with wires and these wires do delay the signal they carry. these delays become very significant when frequency increases.

wire delay is also known as transport delay or flight time.



Skew

The same signal arriving at different parts of the design with different phase is known as skew. Skew normally refers to CLOCK signal.



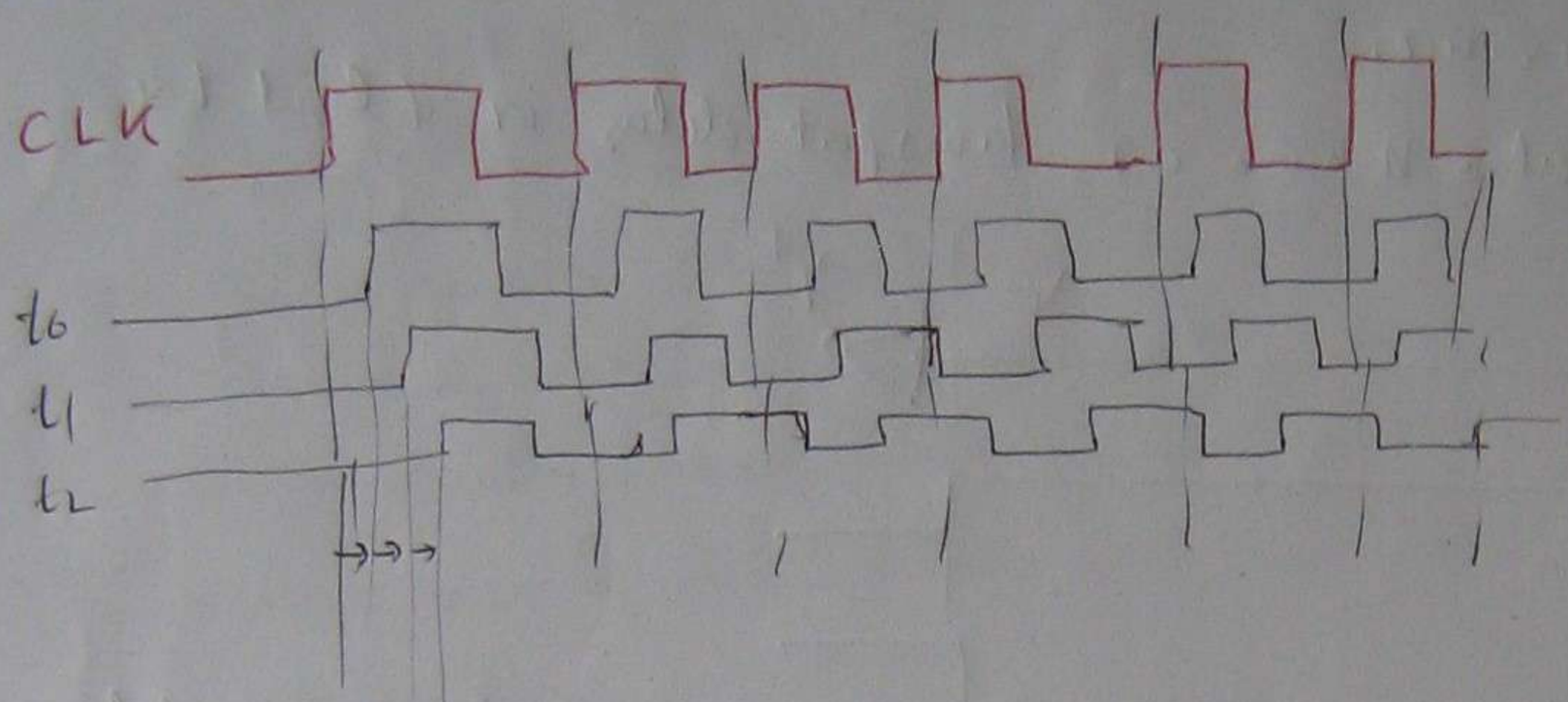
Clock signal reaches FF0 at t_0

FF1 at $t_0 + t_1 + t_2$

FF2 at $t_0 + t_1$

WK 7 (4)

The wave form below shows how clock looks at different parts of the design.



Noise margin

Gate circuits are constructed to sustain variations in input & output voltage levels. variations are the results of several factors.

- Batteries lose their full potential, causing the supply voltage to drop.
- High operating temperature may cause a drift in transistor voltage & current characteristics.
- Spurious pulses may be introduced on signal lines by normal surges of current in neighbouring supply lines.

All these undesirable voltage variations that are superimposed on normal operating voltage levels are called noise.

All gates are designed to tolerate a certain amount of noise on their input & output ports

The maximum noise voltage level that is tolerated by a gate is called noise margin.

LNM- Low Noise margin- The largest noise amplitude that is guaranteed not to change the output voltage level when superimposed on the input voltage of the logic gate (when this voltage is in the Low interval)

$$LNM = V_{ILmax} - V_{OLmax}$$

HNM High Noise margin- The largest noise amplitude that is guaranteed not to change the output voltage level if superimposed on the input voltage of the logic gate (when this voltage is in the HIGH interval)

$$HNM = V_{OH\ min} - V_{IH\ min}$$

tor (rise time)

The ~~time required~~ for

The counter of Fig.6 will count from 0 to 9 (10 states), and when the count reaches 10 (1010_2), it will be reset to zero by the NAND gate. This occurs because both inputs of the NAND gate are at a logic 1, causing it to output a logic 0, thereby asserting the reset inputs of all flipflops. Note that the counter will actually output a count of 1010 for a brief period, equal to the propagation delay of the NAND gate and the reset time of the flipflops, a disadvantage of this type of circuit.

COMMERCIAL ASYNCHRONOUS COUNTER ICS

Various types of counters are available in both the TTL and CMOS families. All TLL asynchronous counters contain two counters, which can be cascaded to give different modulos. CMOS counters vary and one type even has 24 stages. The following is a brief summary of some of the available asynchronous types.

TYPE	MODULUS	CLOCK FREQUENCY
74LS90	2 x 5	32MHz
74LS92	2 x 6	32MHz
74LS93	2 x 8	32MHz
74LS390	2 x 5	40MHz
4020	16384	10MHz (@ 5V)
4024	128	12MHz (@ 5V)

For further information consult manufacturers data books and recommended text.

****

ADVANCED CERTIFICATE IN APPLIED INDUSTRIAL ELECTRONICS
YEAR 1 DIGITAL PRINCIPLES
THEORY LESSON 16

Written by Peter Phillips, 1990

7 SEGMENT DISPLAY DEVICES

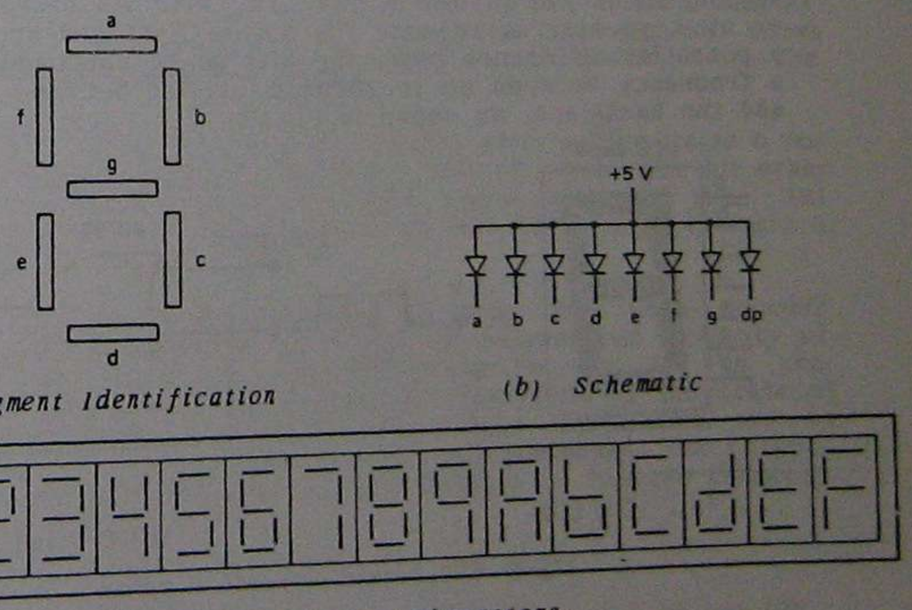
OBJECTIVES: At the end of this lesson you should be able to:
(a) Describe the operation of a common cathode and a common anode 7-segment LED display.
(b) Describe the operating principles, advantages and disadvantages of an LCD display.
(c) Describe the operation of a 7-segment decoder-driver IC.
(d) Describe the principles of multiplexing a 7 segment display.

1. INTRODUCTION

These notes examine the two most common types of displays; the 7-segment LED and LCD types. The ICs used to drive a 7 segment display are also discussed as is display multiplexing methods.

2. 7-SEGMENT LED DISPLAYS

A 7-segment LED display consists of a package containing an arrangement of LEDs and an optical system that provides a format suitable for displaying various alpha-numeric characters. There are two types of LED displays; the common anode and the common cathode. Fig.1 shows a common anode type, including the circuit, segment identification and displayed formats of the hex characters 0 to F.



(a) Segment Identification *(b) Schematic*

(c) Formats for the Hex Characters

Fig.1: *The Common Anode 7-segment LED Display*

The common cathode display uses the same segment identification and displayed character formats, but has the cathode of each internal LED connected to a common line. The common anode display is operated with an active low driver IC, usually with a series resistor between each segment and the driver. The common cathode type requires an active high driver. The series resistor is used to limit the LED current to around 10mA per segment, and for a 5 volt supply will have a value of approximately 330 ohms.

3. THE LCD DISPLAY

The Liquid Crystal Display (LCD) is an AC voltage driven device that operates by reflecting or absorbing light; it does not generate light. The most common type is the field-effect LCD, as it consumes the least power compared to other types, and is used primarily in portable equipment. The power consumption of an LCD is a few microwatts, and the brighter the ambient conditions the better, unlike the LED display which becomes harder to view with increasing ambient light. Also, the LED display consumes around 50mW of power per segment.

However, the LCD has several disadvantages when compared to the LED display, including:

(a) The need for backlighting in low-light ambient conditions, defeating the low power consumption benefit.
(b) A limited operating temperature.
(c) More expensive.
(d) Slower response.
(e) More difficult to multiplex.

Fig.2(a) shows the layout of the LCD, with its backplane common to all segments. A segment will absorb light and appear black, if a potential difference operating **with no DC component**, usually at a frequency between 30 to 200Hz is applied between the segment and the backplane, as shown in Fig.2(b).

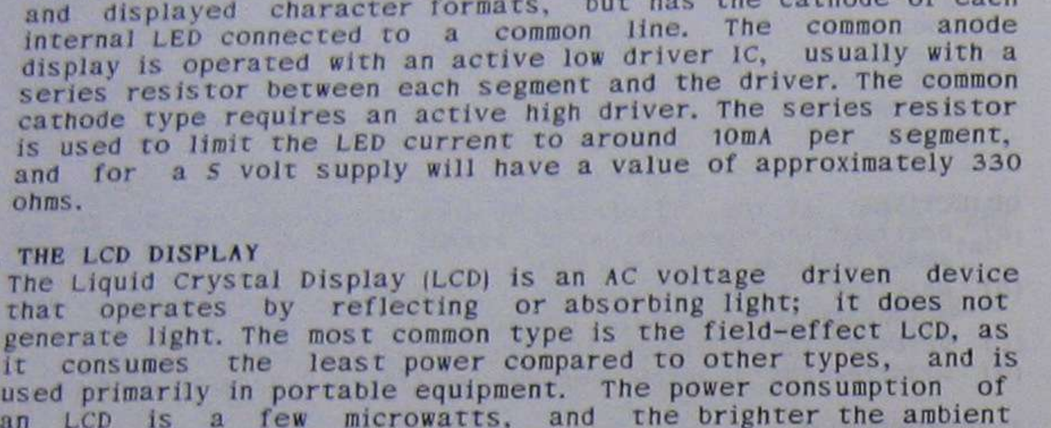


Fig.2: *The Liquid Crystal Display*

If the control input of Fig.2(b) is a 0, the output of the CMOS EX-OR gate will be in phase with the 40Hz signal, causing no potential difference between the backplane and the segment. A 1 on the control line causes the EX-OR gate to behave as an

inverter, creating a potential difference between the applied signals, as they are now 180° out of phase. The segment then absorbs light and appears black compared to the rest of the display.

4. 7-SEGMENT DECODER/DRIVERS

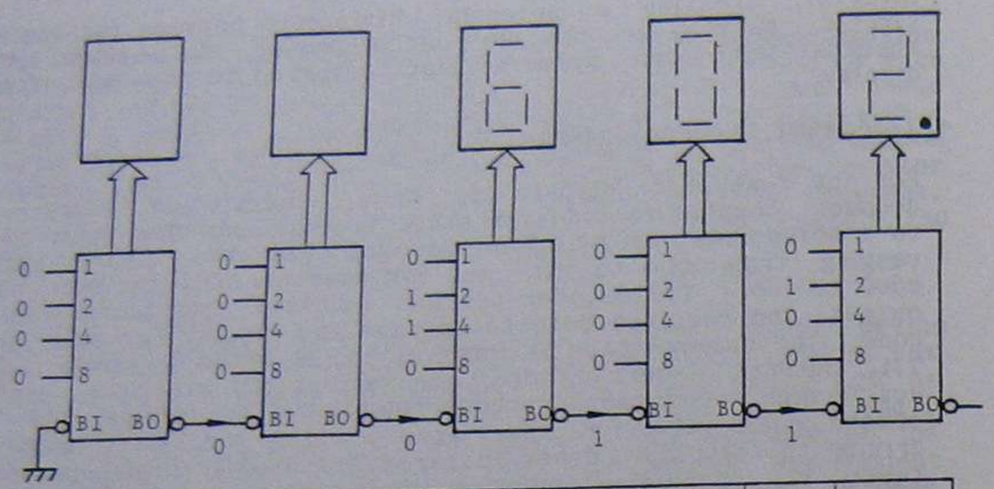
For any 7-segment display to show numerals or letters, a decoder coupled to a driver stage is required. The input codes to a hexadecimal to 7-segment decoder are the binary values ranging from 0000 to 1111, and for numerals only the BCD codes 0000 to 1001. The decoder section comprises gates connected to output the required code to operate the display for each input code. The driver stage is integral and is often a conventional TTL output circuit. There are various ICs available, either active high for common cathode displays, or active low for common anode varieties. Apart from decoding the binary input and driving the display, other features are usually incorporated in these ICs, including blanking of leading or trailing zeros, brightness control and a test for all segments.

4.1 ZERO BLANKING

This facility allows a multidigit number to be displayed without the leading zeros associated with an integer, and the trailing zeros in a decimal value. For example, the number 00602.04010 is more readable as 602.0401. To achieve this, terminals referred to as '*ripple blanking in*' (RBI) and '*ripple blanking out*' (RBO) are provided. Because the decoder/driver ICs for each character would have these pins interconnected, information from the RBO pin *ripples* through to the next RBI pin. Fig.3(a) shows how the integer (value left of the decimal point) of a 5 numeral display would have the RBI and RBO pins interconnected. Fig.3(b) shows the interconnections for a 5 digit decimal value. Note the following comments concerning Fig.3.

* The RBI terminal will blank the display when active (logic 0 for the example), only for the input code of 0000; for all other input codes it has no effect. Under these conditions (i.e., RBI = 0, and input code = 0000) the RBO pin will go to logic 0, otherwise it goes to logic 1.

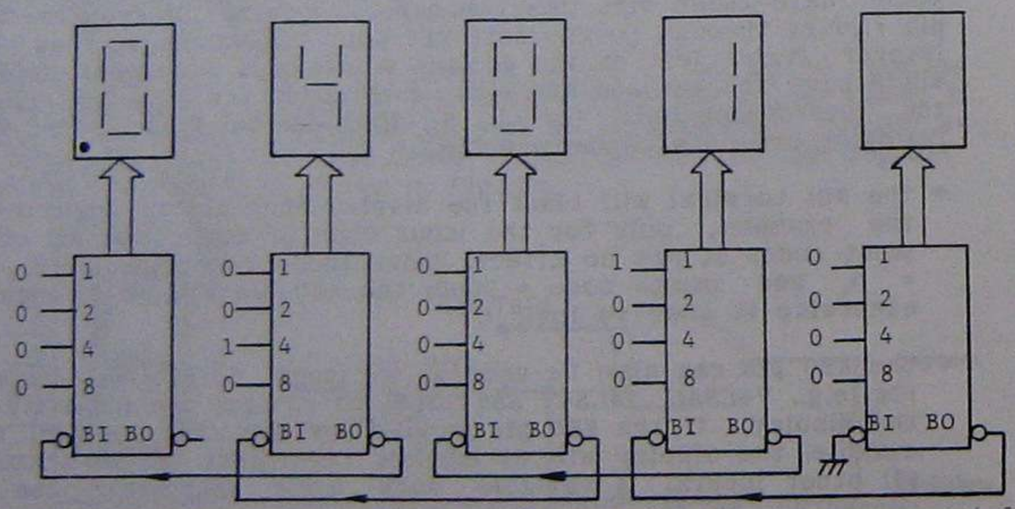
* The RBO pin can also be used as an **input** for some decoder ICs (e.g., 74LS46, 74LS47 and 9368) to control the intensity of the display. If the RBO pin is held low (for the ICs in the example) the display will extinguish, regardless of the state of all other inputs. A variable duty cycle oscillator can be connected to the RBO pin to pulse width modulate the output, which varies the intensity of the display; a feature commonly used on digital clocks.



INPUT CODE	0000		0000		0110		0000		0010
BI	0		0		0		1		1
BO	0		0		1		1		X
DISPLAY	blank		blank		lights		lights		lights

Figure 3(a) Leading Zero Blanking (active low)

* Note that the blanking information ripples from the MSD to the LSD.



INPUT CODE	0000		0100		0000		0001		0000
BI	1		1		1		0		0
BO	X		1		1		1		0
DISPLAY	lights		lights		lights		lights		blank

Figure 3(b) Trailing Zero Blanking (active low)

* Note that the blanking information ripples from the LSD to the MSD.

5. MULTIPLEXED LED DISPLAYS

Multiplexing is used to minimise the IC count for a multi-digit display and allows one decoder/driver IC to operate a number of displays. Fig.4 shows the basic principles of display multiplexing in which switches are used to:

(a) Connect the selected binary value to the decoder.
(b) Supply power to the display that must register the value.

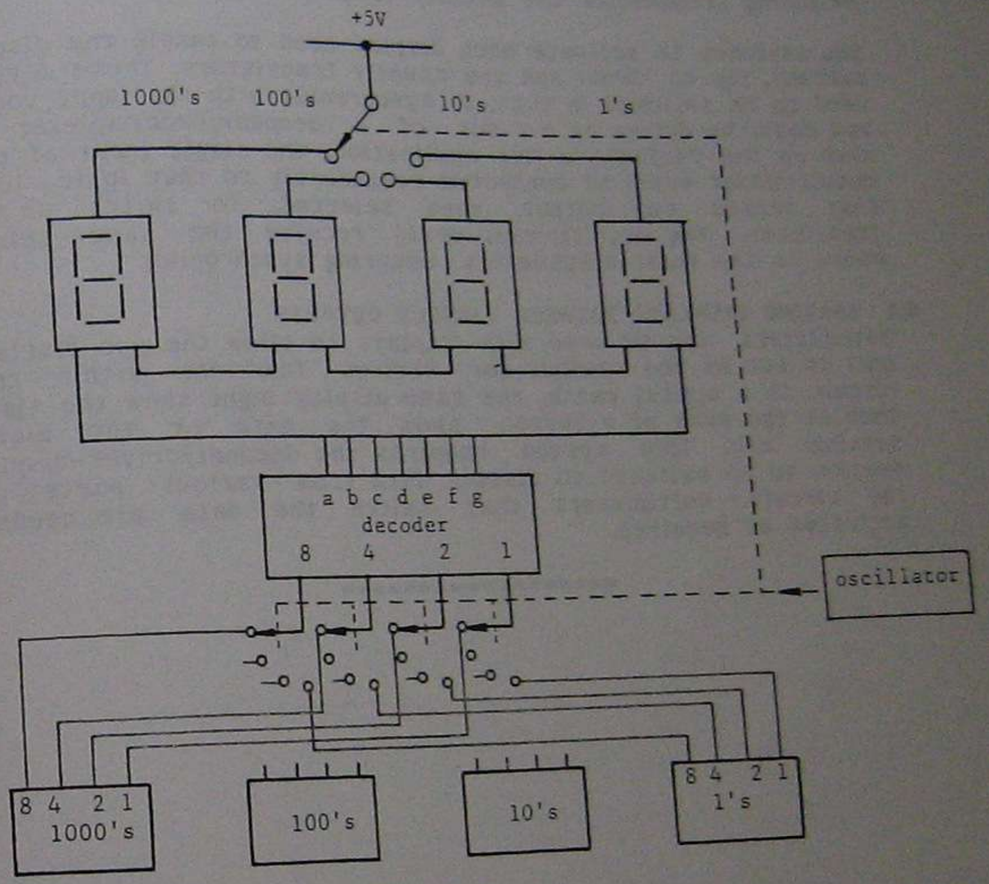


Fig.4: *Multiplexing 7-segment LED Displays*

Note that the decoder outputs are connected to each display in parallel, but because only one display is connected to the supply at a time, each display appears to be independent from the other. Conventional multiplexer ICs could be used to switch the inputs to the decoder. For the circuit of Fig.4, the logic equivalent of a 4 position, 4 pole switch is needed, and could comprise two, dual 4-input multiplexers such as the 74LS253. Each multiplexer in this case has 2 select lines, and would be connected to each receive the sequential series of 2-bit codes of 00, 01, 10 and 11. The oscillator must therefore be a circuit that outputs these codes at the required rate.

The switching frequency must be high enough to prevent flicker, and related to the number of displays being driven. Usually, it is uneconomical to multiplex less than 4 displays and becomes too complex for 10 or more. For example, a multiplexing frequency of 100Hz used in a 4 digit system allows each display to be on for 10mS, then off for 30mS. If 10 displays are used however, each display is on for 10mS, then off for 90mS, causing flicker. Switching frequencies are usually 100Hz or higher.

The switches to activate each display need to handle the display current, (up to 150mA) and are usually transistors. These devices need to be switched in turn, in synchronism with the input codes, and could be driven by a 1 out of 4 decoder/demultiplexer IC such as the 74LS139. In this application, the single input of the demultiplexer would be connected permanently to that logic level that causes the output, when selected, to switch on the transistor. The demultiplexer would receive the same select codes as the input multiplexers, ensuring synchronism.

5.1 SHARING DISPLAYS BETWEEN VARIOUS OUTPUTS

Multiplexers can be used with displays to allow the one display unit to act as the readout for various functions within the circuit. In a digital watch, the same display might show the time, then at the push of a button, show the date or the alarm setting, etc. This system requires the decoder/driver-display section to be switched to display data from various parts of the circuit. Multiplexers that switch the data are needed, activated as required.

****

Review Questions

1. Briefly describe an application of a DAC and an ADC.

_____

_____

_____

_____

2. Calculate the percent resolution of a 12-bit DAC.

_____

3. Calculate the full-scale output voltage of a single-polarity, 10-bit DAC that has a step size of 5mV.

_____

4. A single-polarity, 8-bit DAC has a full scale output of 5V. Calculate the output voltage if the input is set to 1Eh.

_____

5. A single-polarity, 8-bit ADC produces its highest output when 5.1V is applied to the input. Determine its resolution and calculate the maximum change of input voltage to cause the output to change.

_____

6. Explain 'ADC conversion time'.

_____

_____

Skill practice 8: Digital/Analog

Aim

In this exercise you will use commercial A
Digital to Analog Converters (DAC) to co
socket.

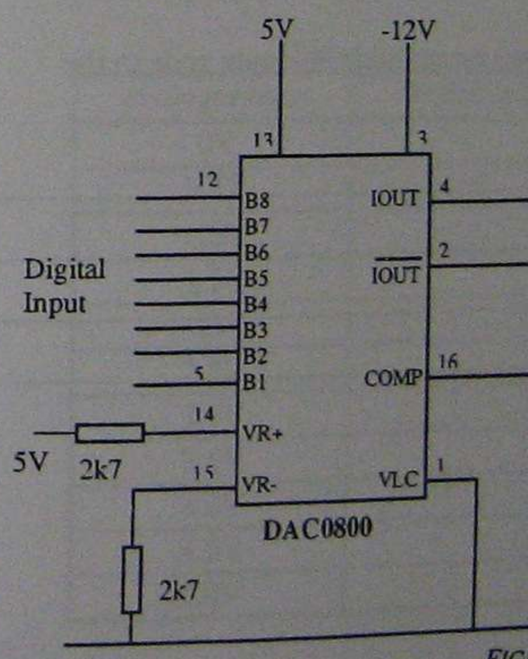
Equipment

- IC types: DAC0800, ADC0802
- resistors: 10k , 3 x 2k7
- 10k potentiometer
- capacitors: 150 pF, 10 nF
- digital bread-boarding system
- logic probe
- dual-trace oscilloscope
- voltmeter
- IC extractor.

Procedure

Part 1: Digital to analog conversion

- Connect the digital to analog conv



FIG

activated as required.

****

Exercise

Q1 Describe the operation of a comman cathode & a comman anode 7 segmed LCD display (P80, 81)

Q2 Describe operation principle, advantages & disadvantages of LCD displays P81

Q3 Describe the operation of 7 segmet decoder - driver IC P82

Q4 Describe the principles of multiplexing a 7 segmet displays P84

Exercise

ADVANCED CERTIFICATE IN APPLIED INDUSTRIAL ELECTRONICS

YEAR 1 DIGITAL PRINCIPLES

Written by Peter Phillips 1990

TOPIC: THEORY ASSIGNMENT WEEK 15

---

1. Determine the number of flipflops required in a counter for each of the following modulo numbers:

   (a) Modulus 3 (b) Modulus 12 (c) Modulus 16

   (d) Modulus 36 (e) Modulus 64 (f) Modulus 144

2. Determine the modulus of the counter in Fig.1

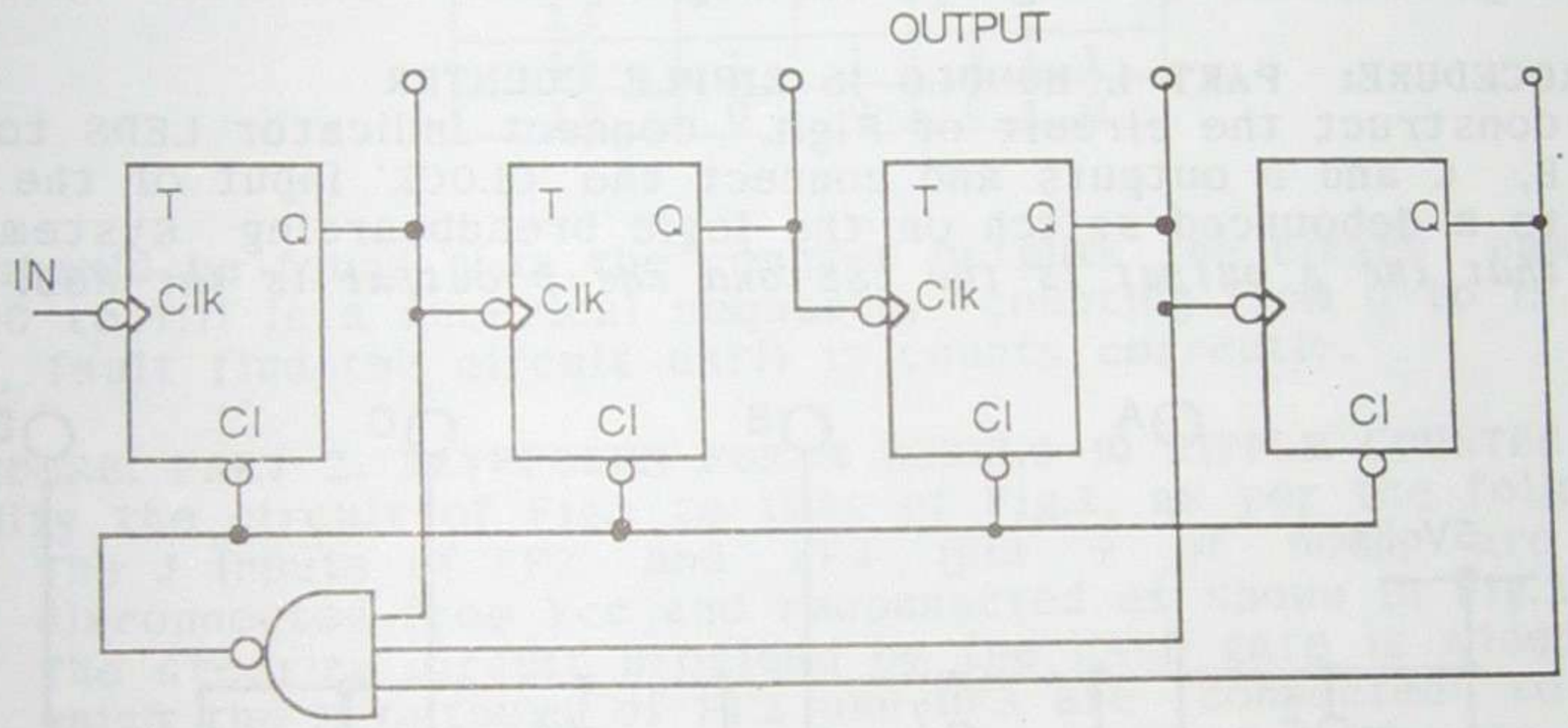


Fig.1

3. Draw the circuit of modulo 8 *count-down* ripple counter that uses JK flipflops.

Page 61

4. Give an advantage of a synchronous counter compared to an asynchronous counter.

P72

****

week ⑥

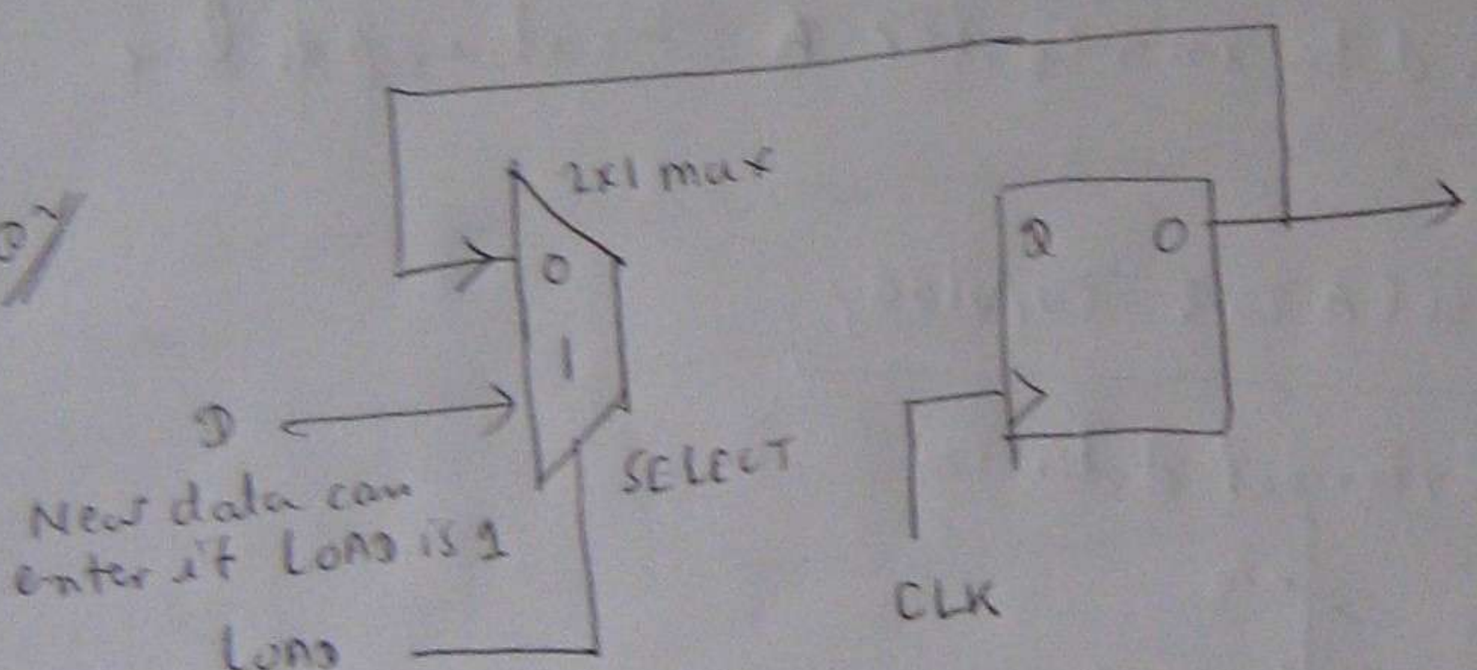
Digital sub system Data Transfer

use of simple registers to control the flow of data in computer system.

cpu

- Data flow section
  - Registers
  - Busses
  - Arithmetic / logic components
- control section (alu logic)
  - steering logic
  - counters for state sequencing

control pins - CLEAR
LOAD



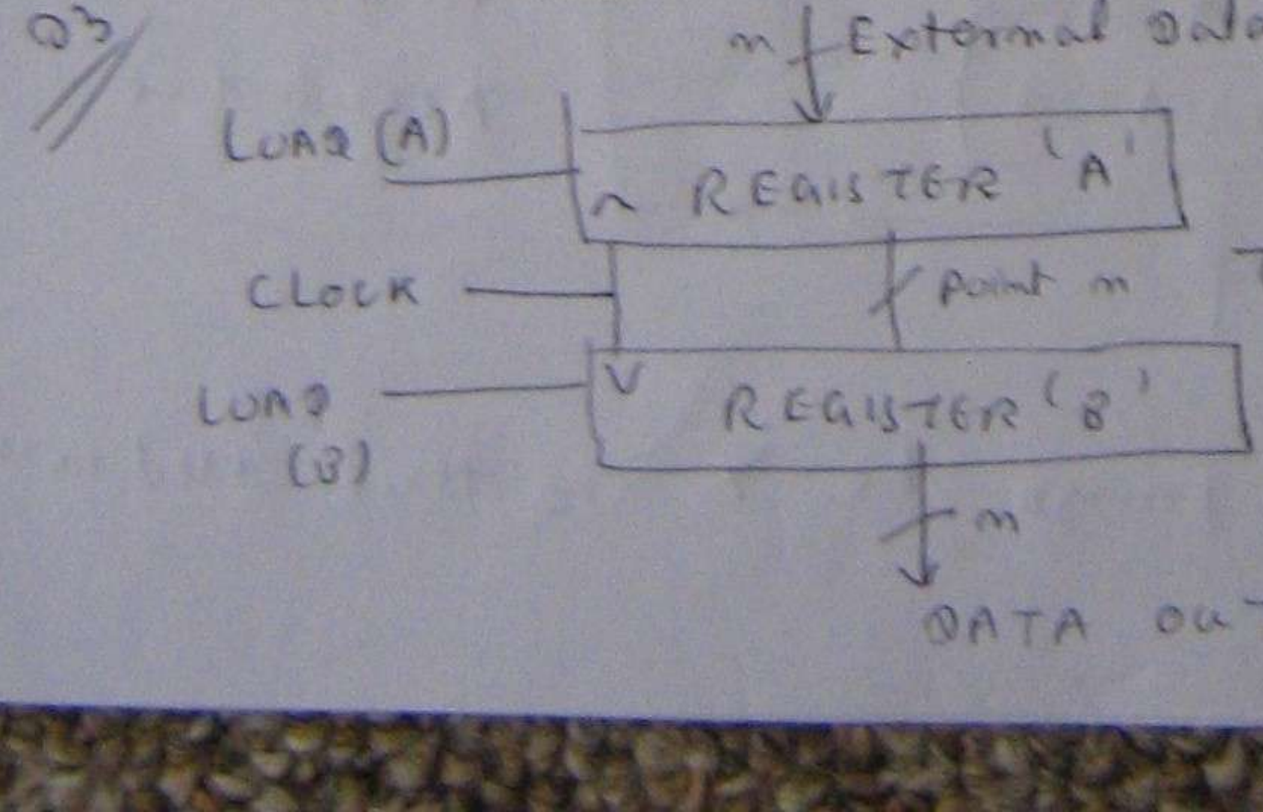
Basic D Register Bit

a mux feeding a D Flip Flop

The clock runs continuously If LOAD is 0. The flip flop just keeps re-storing its old data and so doesn't change.

If LOAD is 1, then new data will be stored.

Basic data flow connection



Transfer bus n bits wide parallel bus

THE SIMPLEST REGISTER TRANSFER ARCHITECTURE

Data will be loaded from the external data input bus into register A if LOAD A is high when clock edge arrives.

32 bit register

If load B High → when the clock edge arrives. whatever is at POINT X will be transferred into register B.

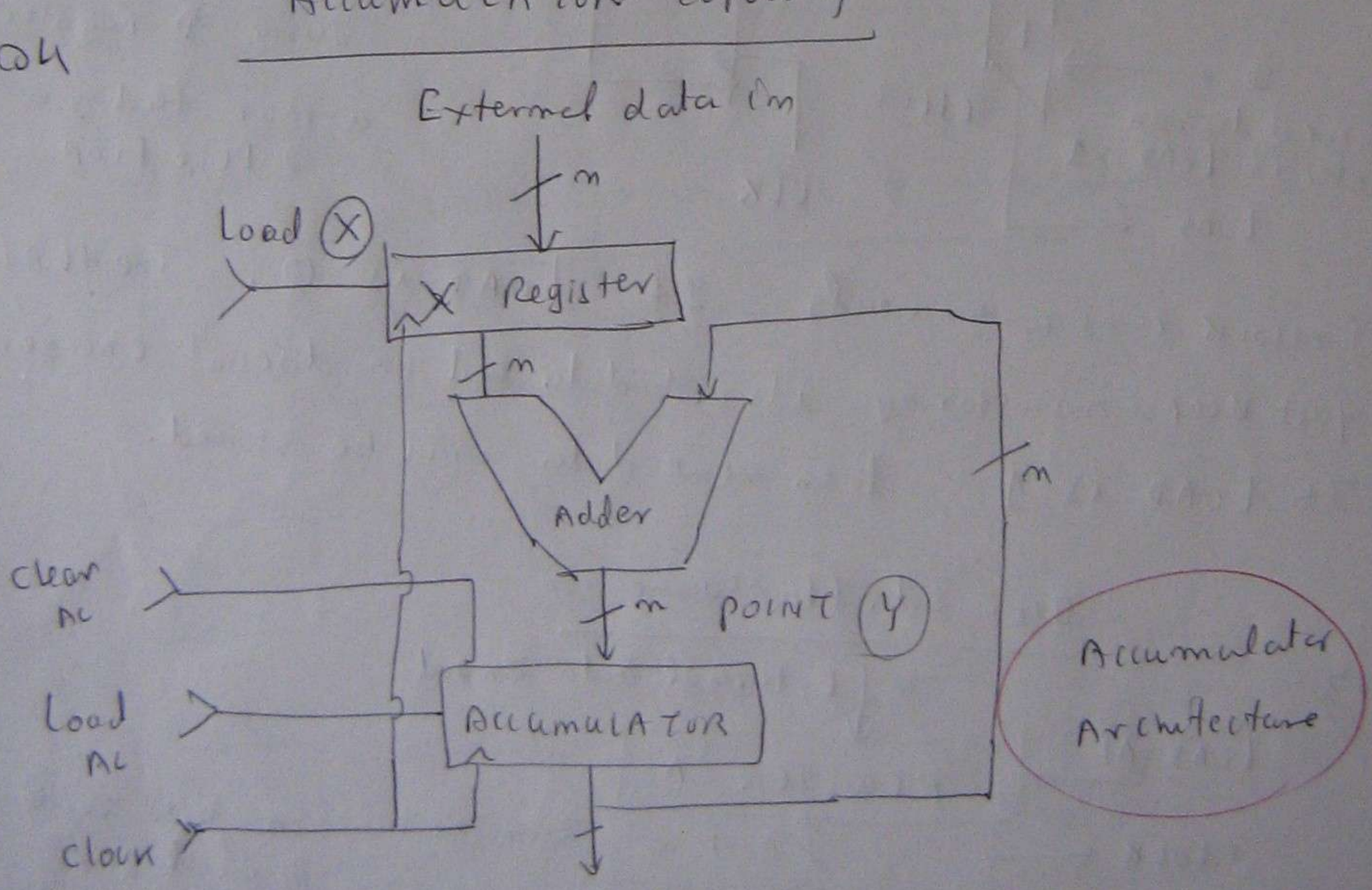
If load A and load B are both 1

when the clock edge arrives, external data goes into register A | will be loaded into register B.

and previous data in register A it is pipe line

ACCUMULATOR Topology

Q4



Here data is brought in to through the X register, is added into accumulator.

The algorithm for tracing the operation of any synchronous circuit is to determine the inputs to all flip flops (registers) when the rising edge of clock arrives

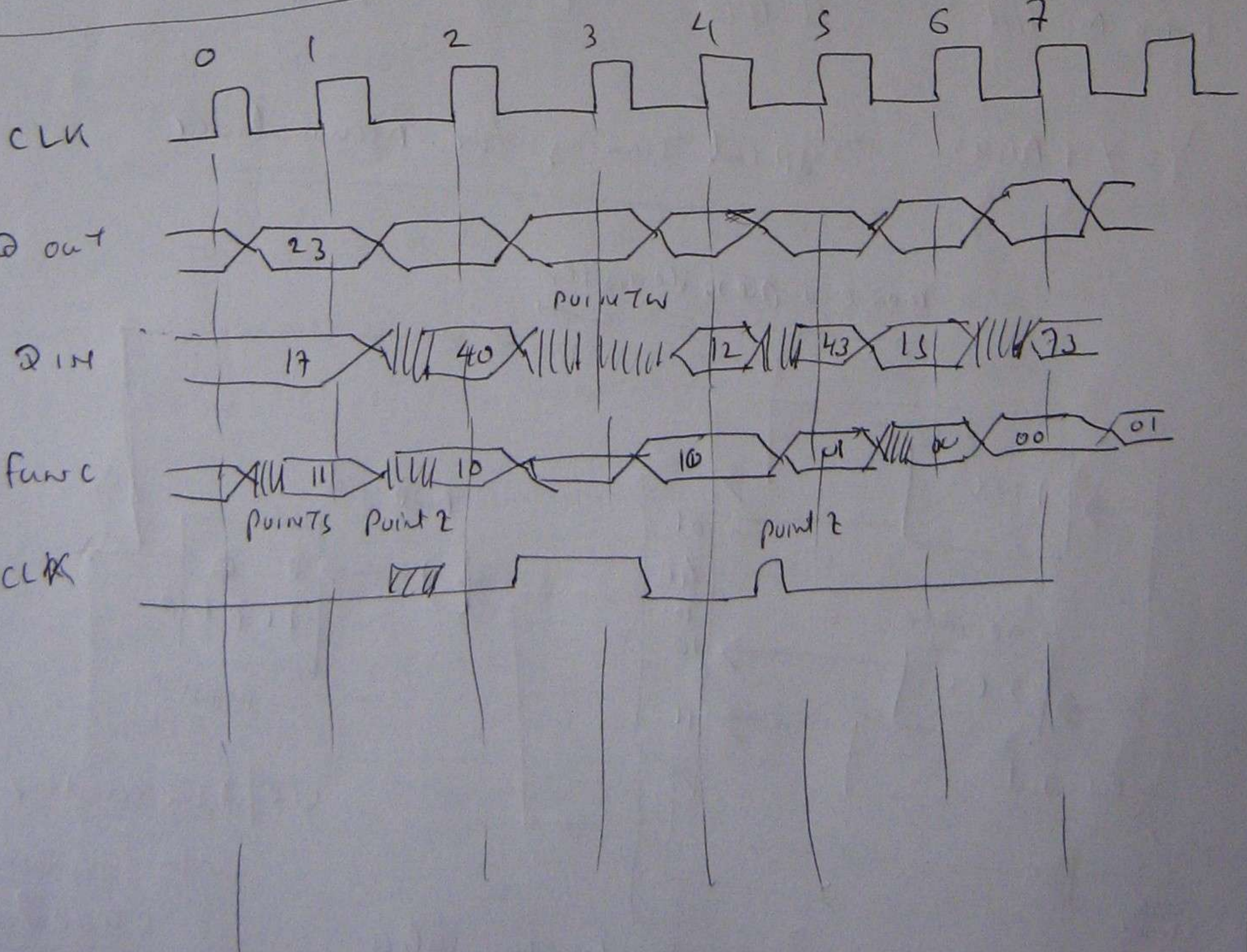
- 10 in Accum Register
- 27 at Exit Input when it is loaded in to X REG by rising edge 0.

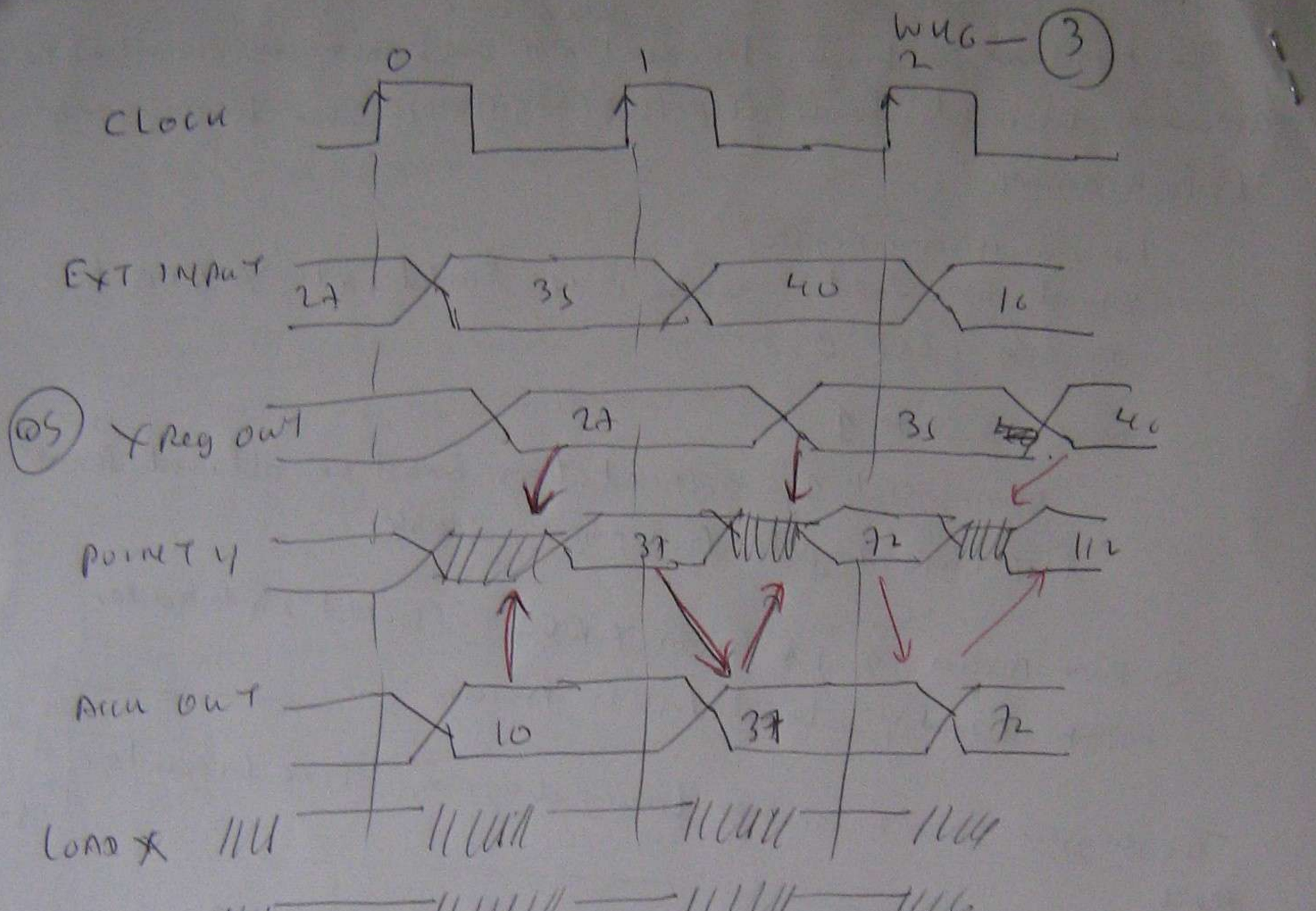
- Now in State 1

  During State 1, sufficient time must be allowed for all flip flops inputs to become stable.

- 10 from ACCUM & 27 from X REG flowed in to adder.
  37 at point Y loaded into ACCUM

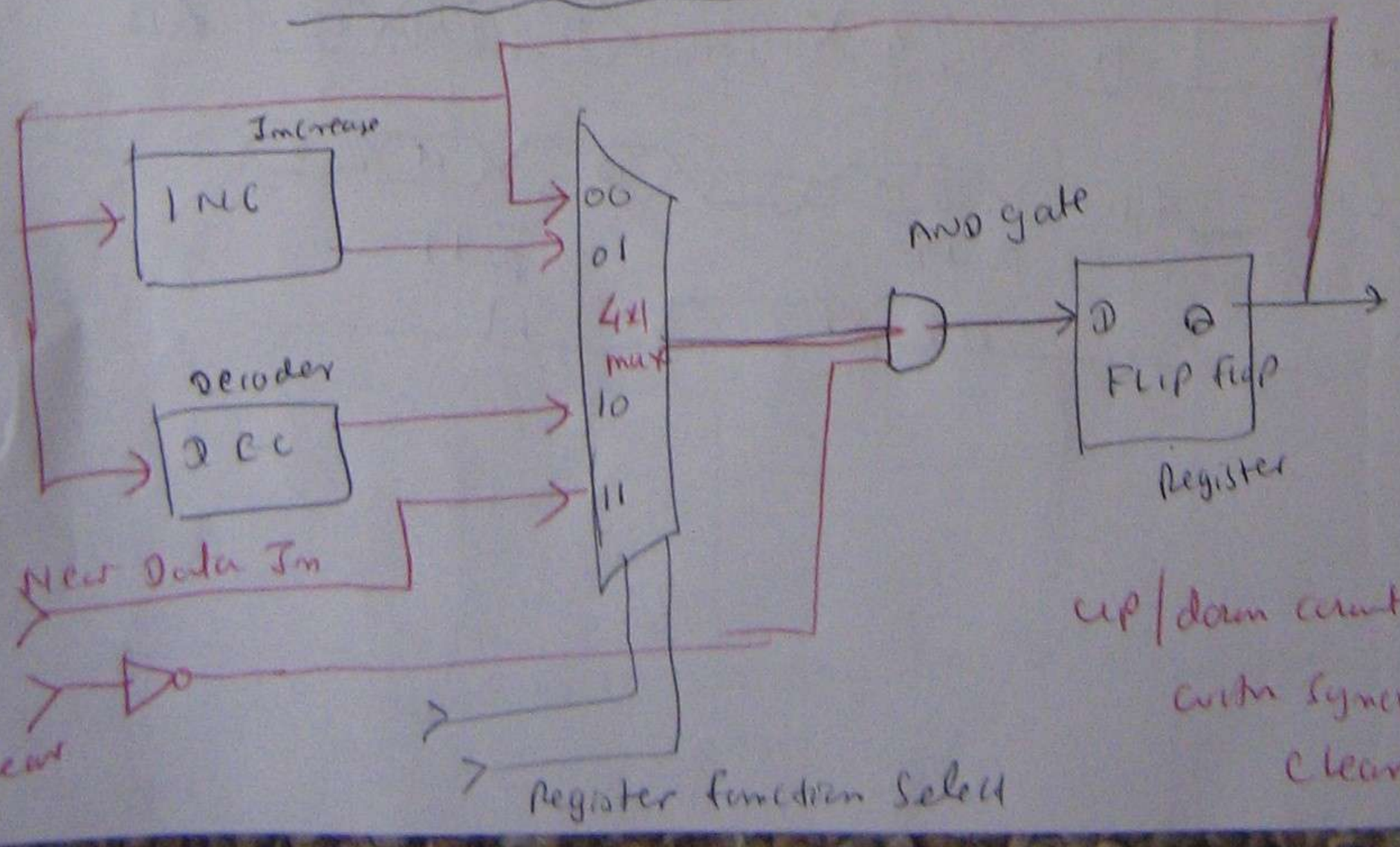
The diagram shows another timing diagram, Time for counter





Y = X + Accu: Typical Timing for Accumulator

more complex register



Week (6)
Exercise

Q1 Show the sections of CPU

Q2 Explain the Basic D Register / D Flip Flop with sketch

Q3 ~~Describe~~ Draw a basic data flow in simplest register transfer architecture

Q4 Explain accumulator topology

Q5 Sketch timing diagram how data are accumulated

ADVANCED CERTIFICATE IN APPLIED INDUSTRIAL ELECTRONICS

YEAR 1 DIGITAL PRINCIPLES

Written by Peter Phillips 1990

TOPIC: THEORY ASSIGNMENT WEEK 9

---

1. Convert the following:

   (a) 479_{10} to BCD = ...

   (b) 1001 0101 0111 0010 (BCD) to decimal = ...

   (c) 1001 0101 (BCD) to binary = ...

2. State a use for the following codes:

   (a) BCD p46

   (b) Gray p48

   (c) ASCII p47

3. Draw the circuit of a 1 of 4 decoder that has active high outputs. Fig 2 P 50

4. Determine the value of the parity bit required for the following codes:

   (a) 110 0111 - EVEN parity; parity bit logic level = ...

   (b) 000 0000 - EVEN " ; " " " " = ...

   (c) 100 0011 - ODD " ; " " " " = ...

   (d) 111 1110 - ODD " ; " " " " = ...

5. Briefly explain the difference between an encoder and a decoder.

p49 + P51

****

DE2 Week (4) Exercise

/

ADVANCED CERTIFICATE IN APPLIED INDUSTRIAL ELECTRONICS

YEAR 1 DIGITAL PRINCIPLES

Written by Peter Phillips 1990

TOPIC: THEORY ASSIGNMENT WEEK 10

---

1. Draw the block diagram of a DUAL 4 input multiplexer. Ps2

2. Draw the block diagram of a 1 of 8 demultiplexer. Ps6

3. Draw the logic circuit of a two input multiplexer. Ps3

4. Show, with a block diagram, how an 8 input multiplexer can implement truth table 1.

TRUTH TABLE 1

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

****

DE2 Week (8) Exercise

ADVANCED CERTIFICATE IN APPLIED INDUSTRIAL ELECTRONICS

YEAR 1 DIGITAL PRINCIPLES

Written by Peter Phillips 1990

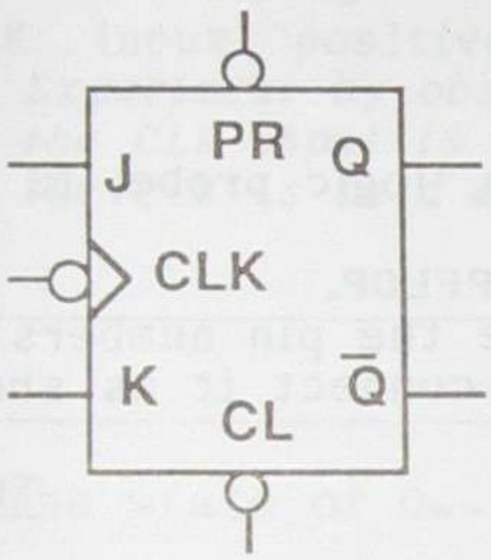
TOPIC: THEORY ASSIGNMENT WEEK 12

---

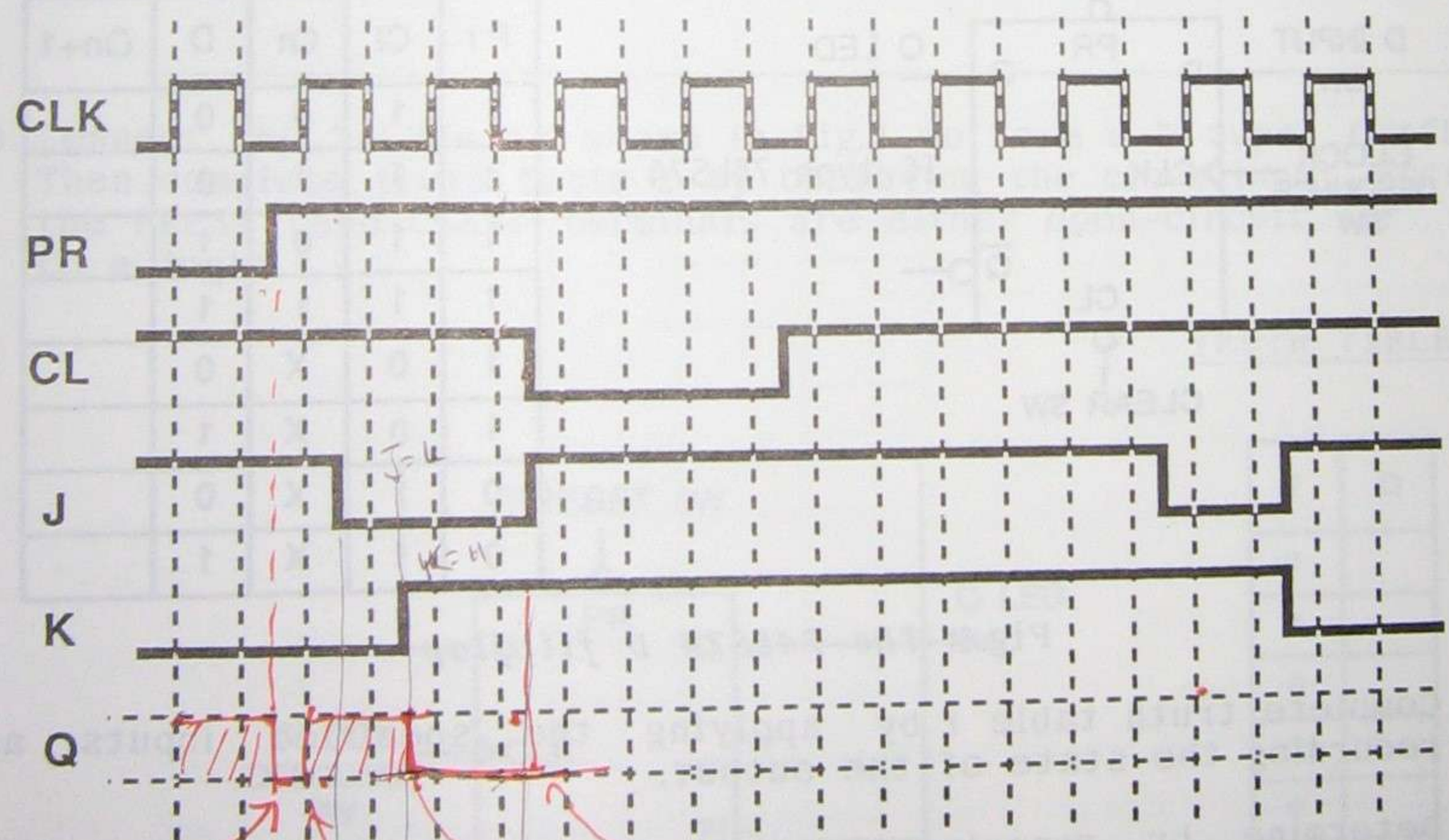
1. Draw diagrams showing how a D flipflop and a JK flipflop can be converted to a T flipflop. P65

2. What advantage does a JK flipflop have over an SR flipflop? P65

3. Draw the output (Q) waveform for the JK flipflop shown. P69



1/2 74LS76



PR L → Q H
CL H

PR - H
CL - H

CLH ↓

J - H
K - L
(H)

(i) PR = H
CL = H CL↓

(ii) J = H K = L Q = H

(iii) J = L
K = H
Q = L

PR = H
CL = L

(iv) Q = L

J = L
K = H
→ Q L

ADVANCED CERTIFICATE IN APPLIED INDUSTRIAL ELECTRONICS

YEAR 1 DIGITAL PRINCIPLES

THEORY LESSON 12

Written by Peter Phillips, 1990

THE D, JK AND T FLIPFLOPS

OBJECTIVES: At the end of this lesson you should be able to:

(a) Show how a clocked RS flip-flop can be altered to become a D flip-flop.
(b) Describe the operation of a JK flip-flop, and draw its truth table.
(c) Describe the difference between edge triggering and level triggering.
(d) Show how a JK flip-flop can be configured as a T and a D type flip-flop.
(e) Describe the master-slave principle as applied to flip-flops.

1. INTRODUCTION

In previous notes the basic SR flipflop was described. The SR flipflop is not a commonly used device, particularly in IC form, as it suffers from the limitation of producing an invalid output if both inputs are asserted. These notes describe the commercially available IC flipflop types, as well as describing the 'master-slave' principle used in some of these to overcome timing problems. The flipflops being examined are:

(a) the D flipflop.
(b) the T flipflop.
(c) the JK flipflop.

2. THE D FLIPFLOP

The SR flipflop has two inputs, in which asserting both can cause an invalid output. By connecting an inverter between the S and R inputs, the flipflop can be converted to the *D flipflop*, in which only one input is provided, as shown in Fig.1.

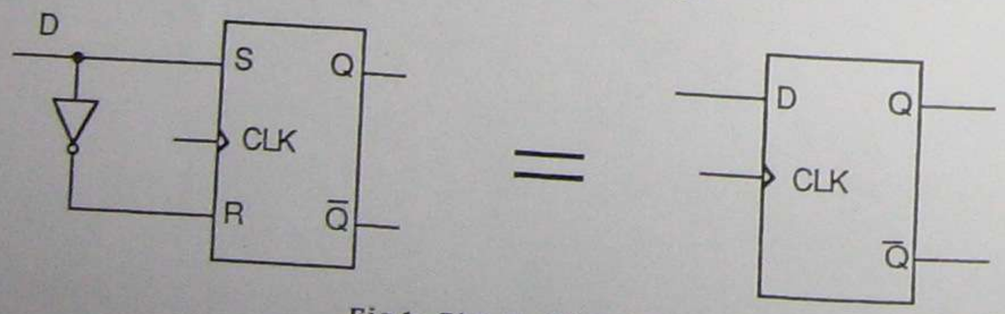


Fig.1: *The D flipflop*

The truth table for the D flipflop shows that the output equals the D input after the clock pulse. That is, after the clock pulse Q_{N+1} equals D, regardless of the previous state of Q. In other words, the D input is stored until the next clock pulse.

D FLIPFLOP TRUTH TABLE

	D	Q	Qn+1	
1	0	0	0	RESET
2	0	1	0	
3	1	0	1	SET
4	1	1	1	

3. THE TOGGLE (T) FLIPFLOP

Toggling is an action that causes the output of a flipflop to change between its two output states on each consecutive operation of the clock pulse. A push-on, push-off switch is a form of toggle. The flipflops of Fig.2 show how an SR and a D flipflop can be made to toggle with each successive clock pulse.

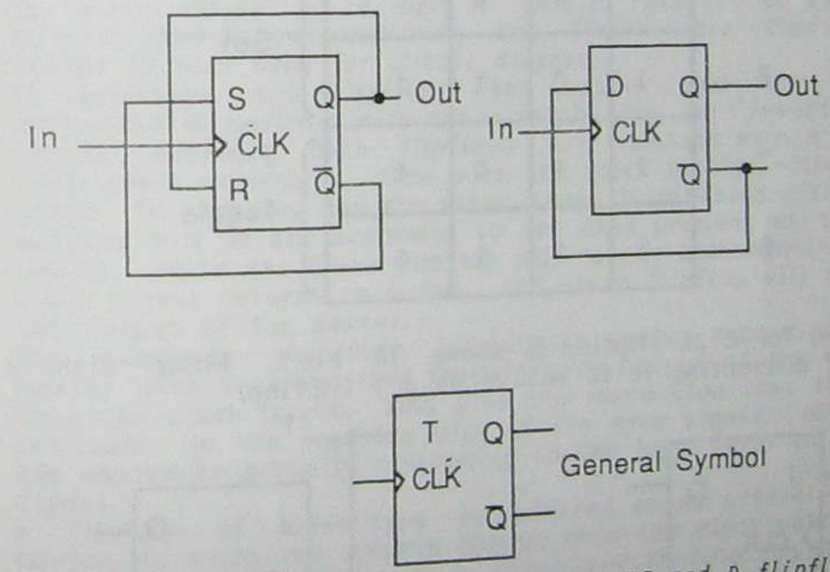


Fig.2: *The T (toggle) flipflop using SR and D flipflops*

In both cases, the output of the flipflop is fed back to the input, meaning the input data will be different for each clock pulse. The input is now the clock signal, and the output can be take from either output terminal. The output signal will be half the frequency of the input clock signal.

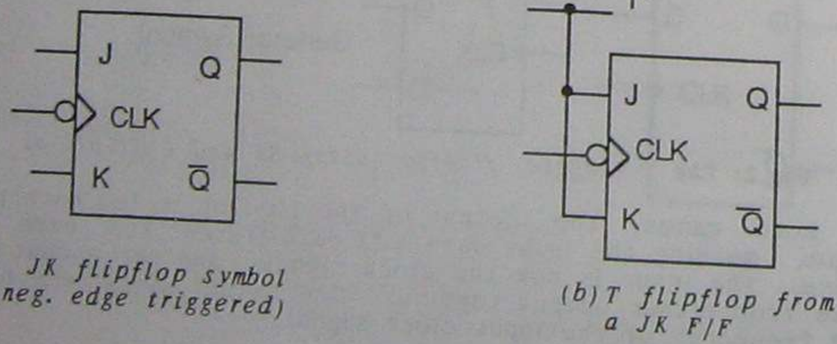
3. THE JK FLIPFLOP

The JK flipflop overcomes the problem of the possible invalid output condition inherent in the SR flipflop, while still retaining two inputs. The *SET* input is now the *J* input, and the *RESET* input is now the *K* input. However, if both the J and K inputs are held at a logic 1, the outputs will toggle, rather than become indeterminate, as shown in the truth table.

JK FLIPFLOP TRUTH TABLE

	J	K	Q	Qn+1	
1	0	0	0	0	No Change
2	0	0	1	1	
3	0	1	0	0	Reset
4	0	1	1	0	
5	1	0	0	1	Set
6	1	0	1	1	
7	1	1	0	1	Toggle
8	1	1	1	0	

The symbol for a JK flipflop is shown in Fig.3, along with a method of connecting it to behave as a T flipflop.



(a) JK flipflop symbol (neg. edge triggered)

(b) T flipflop from a JK F/F

Fig.3: *The JK flipflop*

4. MASTER-SLAVE FLIPFLOPS

Connecting conventional JK flipflops in tandem, as shown in Fig.4, can give problems due to time delays within each flipflop. The problem arises from the fact that although the flipflop changes on the edge of its clock signal, there is a small time interval before and after the clock edge during which the input data must be held stable. These times are called the *setup* and *hold* times, respectively.

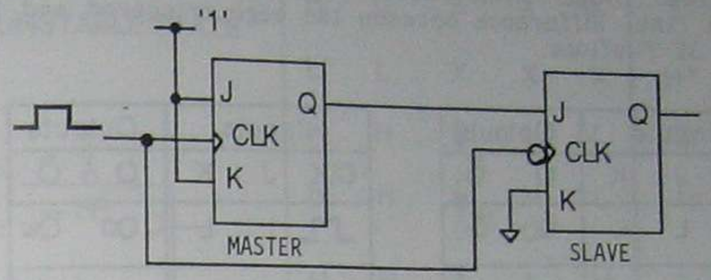


Fig.4: *JK flipflops in tandem*

In Fig.4, the J and K data (logic 1) is processed on the rising edge of the clock pulse. The output of the first flipflop feeds the input of the second, and both flipflops are clocked together. The problem occurs because the output of the first F/F is changing (toggling) when the clock pulse rises to a logic 1. This changing data is being fed to the second F/F, whose clock signal is also rising to a logic 1. This is referred to as the *race effect*, and is overcome with the *Master-Slave* configuration. *(Refer to text book for circuit diagram).*

In principle, two flipflops, the master and the slave are connected in series, with the slave having an inverted clock signal. Assuming both flipflops are enabled with a logic 1 at their clock terminals, then when the clock is high, the master stage is enabled and the slave stage is disabled. The master outputs will be set according to the data present at the master inputs, while the slave flipflop will remain unchanged. When the clock signal returns to a low, the slave flipflop will respond to the output of the master.

The sequence therefore is: data to master when clock is high, master data to slave (and therefore to output of the whole unit) when the clock is low. This give the impression that the output responds to the negative edge of the gate signal, even though the device is actually responding to the logic *level* of the gate signal.

A flipflop of this type is referred to as a *level-triggered* device, in which the outputs change when the clock pulse reaches a certain level. Thus, it is necessary to hold the input data steady while the clock is high as during this time the master can respond to its inputs. The slave will respond to the state of the master outputs as soon as the clock signal becomes a logic 0, meaning the final output will depend on the data present to the master inputs just prior to the clock signal's negative edge.

To overcome this problem, some flipflops utilise *edge triggering*, in which internal circuitry causes the output to respond only when the clock is actually changing. Thus the data needs only to be held constant for the short time that the clock is actually changing and may vary at all other times, without causing a change to the outputs. However, the clock must have a fast rise and fall time, usually within 5 microseconds. It must also be bounce free. Fig.5 illustrates how National Semiconductor illustrate the difference between the edge triggered and level triggered JK flipflops.

Inputs			Outputs	
CLK	J	K	Q	$\overline{Q}$
↓	L	L	Q_0	$\overline{Q}_0$
↓	H	L	H	L
↓	L	H	L	H
↓	H	H	TOGGLE	

EDGE TRIGGERED JK FLIP FLOP

Inputs			Outputs	
CLK	J	K	Q	$\overline{Q}$
⎍	L	L	Q_0	$\overline{Q}_0$
⎍	H	L	H	L
⎍	L	H	L	H
⎍	H	H	TOGGLE	

MASTER SLAVE JK FLIP FLOP

Fig.5: *Comparison of edge and level triggered JK flipflops*

Note: The master slave flipflop requires a complete clock cycle to transfer input data to the outputs for both types.

5. THE JK FLIPFLOP AS A D FLIPFLOP

The JK flipflop can be connected to behave as a D flipflop (one input) by connecting an inverter between the J and K inputs as shown for the SR flip in Fig.1.

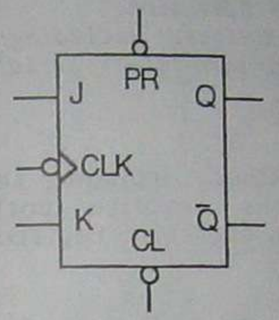
6. PRACTICAL CONSIDERATIONS

When first powered up, flipflops have an unpredictable output state. To allow control under power up, and to give overriding control for particular applications, manufacturers provide *asynchronous* inputs (inputs not controlled by the clock) that can set or clear the Q output. For example, the JK flipflop of Fig.6 has two asynchronous inputs labelled PRESET and CLEAR that are asserted with a logic 0. Thus, a low to the CLEAR input will clear Q to a 0, and a low to the PRESET will set Q to a 1. If both the PRESET and CLEAR inputs are asserted together, the output will be unstable, as in the SR flipflop. The truth table in Fig.6 lists how each input affects the flipflop. Note that Q_0 indicates that the Q output does not change.

X = DONT CARE

↓ = NEG CLK EDGE

H* = UNSTABLE STATE



Inputs					Outputs	
PR	CL	CLK	J	K	Q	$\overline{Q}$
L	H	X	X	X	H	L
H	L	X	X	X	L	H
L	L	X	X	X	H*	H*
H	H	↓	L	L	Q_0	$\overline{Q}_0$
H	H	↓	H	L	H	L
H	H	↓	L	H	L	H
H	H	↓	H	H	TOGGLE	
H	H	H	X	X	Q_0	$\overline{Q}_0$

Fig.6: *JK flipflop with asynchronous PRESET and CLEAR*

****

****

ADVANCED CERTIFICATE IN APPLIED INDUSTRIAL ELECTRONICS

YEAR 1 DIGITAL PRINCIPLES

THEORY LESSON 9

Written by Peter Phillips, 1990

ENCODERS AND DECODERS

OBJECTIVES: At the end of this lesson you should be able to:

(a) Convert values between decimal and BCD.
(b) Describe the difference between weighted and unweighted codes.
(c) Define various terms applicable to data transfer.
(d) State the use of the ASCII code.
(e) Determine the logic level of a parity bit required for odd and even parity within a given data byte.
(f) Describe the format and an application of the Gray code.
(g) Draw the circuit diagram of a 1 of 4 decoder, and describe its operation.
(h) Describe the operation of an encoder.

1. INTRODUCTION

Any data within a digital circuit is comprised of "1's" and "0's", although an output device may display the data in various forms, such as the octal or hexadecimal forms discussed previously. Often, input devices to a digital circuit, such as a keyboard or a digital transducer will input the binary information in a form other than straight binary; that is, as a *code*. To change the binary data codes from one form to another, decoders and encoders are used. The codes discussed are: the BCD code, the Gray code and the ASCII code. The decoders/encoders examined are: the '1 of 4' decoder, the '1 of 10' decoder, a BCD encoder and a priority encoder.

2. CODES

Some of the terms applicable to binary data are listed below:

(a) ***BIT:*** a single binary logic level of either 1 or 0.
(b) ***BYTE:*** a group of eight bits.
(c) ***NYBBLE:*** half a byte.
(d) ***WORD:*** a group of bytes.
(e) ***FORMAT:*** the specific arrangement of the data, such as the number of bits/byte, the number of bytes/word and the task allotted to each bit or byte.
(f) ***CHARACTER:*** the numerals 0 to 9 and all letters and symbols used by a typewriter.
(g) ***SERIAL DATA:*** a data transfer method in which each bit is fed sequentially over a single transmission line.
(h) ***PARALLEL DATA:*** a data transfer method where data is transferred over a number of lines.

Digital codes can be classified into two groups; weighted and unweighted. A weighted code is one in which the position of the binary bit determines its value, and evaluation of the code is a matter of adding the weights. Unweighted codes cannot be evaluated in this way.

2.1 THE BCD CODE

The *BINARY-CODED DECIMAL* code, (BCD or 8421 code) is a weighted code and uses the binary values of the decimal characters 0 to 9 only. Table 1 lists the BCD codes (*which are the same as the natural binary values*) of these 10 decimal characters.

TABLE 1

Decimal	BCD	Decimal	BCD
0	0000	5	0101
1	0001	6	0110
2	0010	7	0111
3	0011	8	1000
4	0100	9	1001

To convert a decimal number to BCD, write the 4 bit binary value of each character. For example:

23_{10} = 0010 0011 (BCD)

2579_{10} = 0010 0101 0111 1001 (BCD)

This code should not be confused with natural binary. For example, the number 16 is written in BCD as 0001 0110; in natural binary it is 10000.

Some characteristics of the BCD code are:
(a) The BCD code is more wasteful than straight binary.
(b) A number encoded in BCD is more easily read by a human operator than its equivalent in natural binary.
(c) The code uses the weighting values of 8421; natural binary uses an infinite number of weights.
(d) The following 4 bit binary numbers are invalid in the BCD code: *1010, 1011, 1100, 1101, 1110, 1111.* (i.e., the 4 bit binary codes for the decimal numbers 10 to 15).

2.2 OTHER WEIGHTED CODES

By altering the weighting of each bit position, the 10 decimal numerical characters can be represented by any code; not just 8421 BCD. Some of these codes are:

(a) The 2421 code. (b) The 5421 code. (c) The 7421 code.
(d) The Biquinary code. Weighting is 5-0 4-3-2-1-0 (Eg, 8 is 10 01000).

There are various advantages for each code, which are used in specific applications. Encoding a decimal value to any of these codes requires that the weighting values add up to that number. E.g., decimal 9 in the 7421 code is 1010. A multidigit number is expressed with each digit converted to its coded equivalent, as in 8421 BCD.

2.3 THE ASCII CODE

The codes discussed so far represent numerical values, and can be used in arithmetic operations. A widely used code for information transfer is the American Standard Code for Information Interchange (ASCII), which uses a 7 bit code for all the alpha-numeric characters on a typewriter, including codes for operations such as a line feed, space, etc. It is commonly used when sending text between computers, printers, wordprocessors etc. It is not used in mathematical operations, although all the numerical characters have an ASCII code. A computer keyboard will generate the ASCII code for each key press, and the computer will store this code (in 1's and 0's), but display it as an alpha-numeric character. The following table lists a few ASCII codes:

CHARACTER	ASCII CODE	
	BINARY	HEX
A	100 0001	41
B	100 0010	42
a	110 0001	61
0	011 0001	30
9	011 1001	39
space	010 0000	20

Because 7 bits are used, up to 128 discrete codes can be produced. The code 000 0000 ($00) is known as a NULL, and the highest code of 111 1111 ($7F) is called DELETE, which backspaces and erases the last character. Other alpha-numeric codes include the relatively old 5-bit Baudot code (used in teletypewriters), and the EBCIDIC and Hollerith codes occasionally used by IBM and Burroughs.

2.4 ERROR DETECTION

Digital data transmission has the advantage over analog data transmission in that extra information can be included within the data to enable the receiver to check for any errors. Apart from error detection, error *correction* is possible if enough error coding is included. The simplest form of error detection is the use of ***Parity.*** This method can be either of two forms; *odd or even parity.* In even parity, the transmitter ensures that all data transmitted contains an even number of logic 1's. This requires the inclusion of a '*parity bit*', which is set to either a 0 or a 1, depending on the number of 1's otherwise in the transmitted byte. For example, when transmitting the ASCII code, an 8th bit is added to the 7 required for the ASCII code. Any code with an odd number of 1's would need the parity bit set to a 1 for even parity, otherwise the receiver would reject the data byte. In this case, the parity bit for ASCII 'a' (110 0001) would be a 1 to give four 1's within the 8 bit byte.

Odd parity is that arrangement that guarantees an *odd* number of 1's in the binary byte. For the ASCII code of 'a', the parity bit would be 0, as the data has the required odd number of 1's. Because of its simplicity, error correction is not possible, and the loss of two 1's would not be detected.

2.5 THE GRAY CODE

All the codes so far discussed apply to either numerical or text representation. Another type of coding is used in digital *control* systems. Positional control is a common function required in industry, in which a controller must position a slide, or rotate a shaft according to a particular input signal. Some form of feedback is necessary to inform the controller circuitry that the correct position has been reached, and it is in this application that the Gray code is used.

The Gray code belongs to a class of codes called *minimum-change* codes. This means that each adjacent code has altered by only one bit. Table 2 shows possible Gray code values for the decimal numbers 0 to 15. A 4 bit Gray code is used as this satisfies the requirements for 16 discrete steps, but Gray codes can have as many bits as required to obtain the required resolution.

TABLE 3

Decimal	Binary	Gray Code
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

Fig.1 shows a 12 segment, 4 track Gray code optical disc that could be attached to a rotating shaft. Four light sensors (one per track) to detect the black (for 1) and white (for 0) are used to produce the digital output. Because the code must present a one bit change between all sectors, including sectors 0 and 11, a different code for values above 6 to that of table 3 is necessary, illustrating that there is no standard Gray code.

A shaft encoder that resolves angular rotation to more than one sixteenth of 360° would require more 'tracks' to give the number of bits required. If 12 tracks are encoded onto an optical disc, a resolution of 2^{12} (equals 4096), would result, giving an angular

displacement to within 0.1°. The feature of a one bit change between adjacent binary numbers prevents errors occurring when the encoded disc is changing between sectors. If a standard binary code was used, then a change for example between sectors 7, (0111) and 8 (1000) would result in all 4 bits changing. Any slight misalignment of the optical sensors could cause wrong digital codes to be produced, giving unpredictable results.

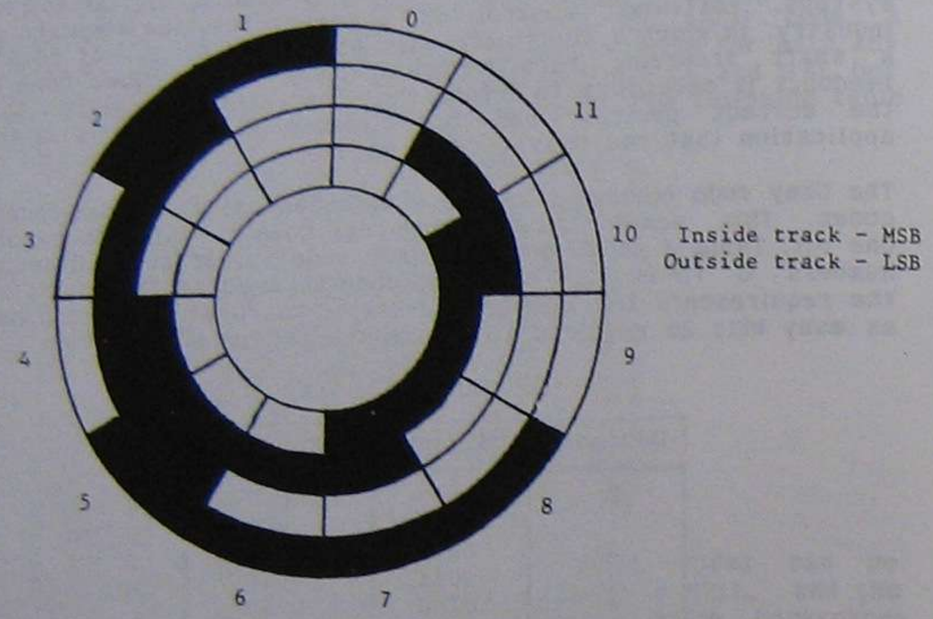


Fig.1: *12 Segment Optical Shaft Encoder*

3. DECODERS AND ENCODERS

For the purposes of these notes, a ***decoder*** is defined as a device that has a number of inputs and outputs, but activates only ***one output*** for a given input code. An ***encoder*** is a device that has a number of inputs and outputs, but produces a specific code at the output when only ***one input*** at a time is activated. Note however, that some decoders (those that drive a LED display for example) activate more than one output at a time. These are described in later notes.

3.1 DECODERS

There are a number of ICs listed as performing a decoding function, and the simplest of these is the 1 of 4 decoder. This decoder converts a 2 bit binary value to its decimal equivalent and is shown in Fig.2. It operates by sending one, *and only one*, output high for each of the four codes. Recall that an AND gate will produce an output of 1 only when all its inputs are a 1. In Fig.2, if the input is A = B = 0, then output 0 will be high, as the inverters place two 1's at the input of the top AND gate. This circuit decodes the binary numbers 00, 01, 10 and 11, by activating outputs 0, 1, 2 and 3 respectively.

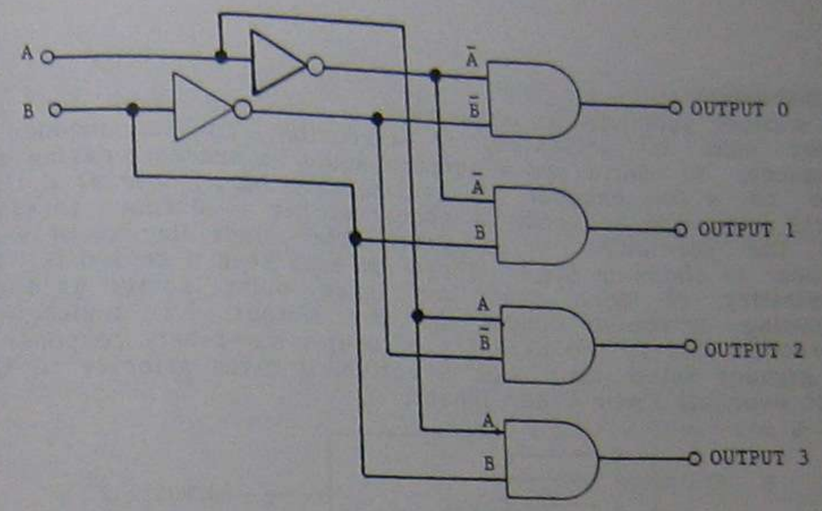


Fig.2: *A 1 of 4 Decoder*

By expanding the circuit, any number of outputs can be provided. The usual maximum is 16 outputs, which requires 4 inputs; making a 1 of 16 decoder. A common type of decoder is the 1 of 10; also known as a BCD decoder, as it will decode a BCD input back to decimal. This decoder requires 4 inputs and has 10 outputs. Because 4 inputs will allow 16 input codes, the BCD decoder must not respond to the binary inputs above decimal 9; that is, no outputs are activated for the invalid BCD codes. The 7442 IC is a TTL 1 of 10 BCD decoder. This IC has 'active low' outputs, which means the output goes to a logic 0 when selected; all others remain at a logic 1. This allows the outputs to be used to drive a load, because an output, when low, can sink more current than it can source when high. Note that inverter pairs are used at the input to ensure the fan in of the IC is maintained at one unit load. The operational block diagram of an active low BCD decoder is shown in Fig.3.

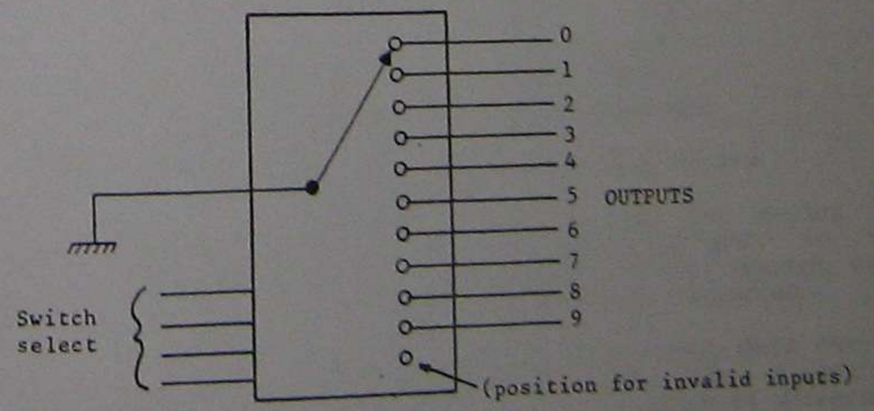


Fig.3: *Block Diagram of an Active Low BCD Decoder*

3.2 ENCODERS

An encoder produces at its outputs a specific code for each input that is activated. If a decimal to BCD encoder was required, 10 inputs and 4 outputs would be needed; making a 10 line to 4 line encoder. Activating each input, *one at a time*, would cause the BCD code to appear at the 4 output terminals for the particular value input. The block diagram of a BCD encoder is shown in Fig.4. A problem with such a device is the possibility of more than one input being active at a time producing erroneous codes at the output. A device that overcomes this is the *PRIORITY* encoder, as it only responds to the highest value input that is active: it gives priority to this input over all lower order inputs.

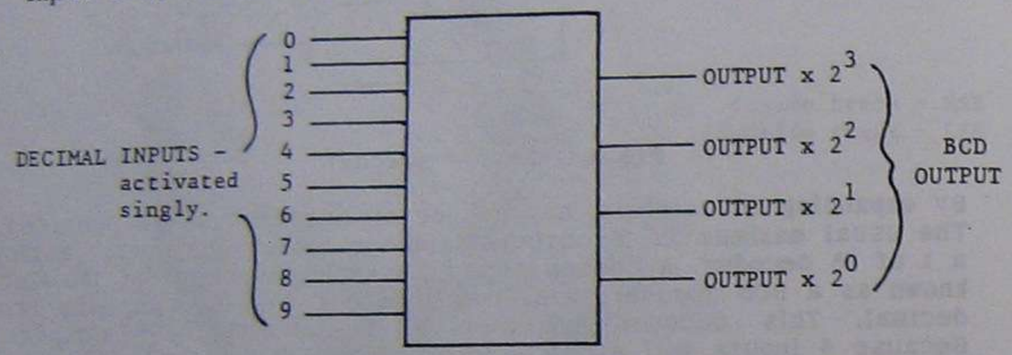


Fig.4: *Block Diagram of a BCD Encoder*

****

ADVANCED CERTIFICATE IN APPLIED INDUSTRIAL ELECTRONICS

YEAR 1 DIGITAL PRINCIPLES

THEORY LESSON 10

Written by Peter Phillips, 1990

MULTIPLEXERS

OBJECTIVES: At the end of this lesson you should be able to:

(a) *Draw the block diagram of a 4 way multiplexing system, and describe the principle of time division multiplexing.*
(b) *Draw the logic diagrams of a 4 way multiplexer and a 4 way demultiplexer.*
(c) *Show how a multiplexer is able to implement a given combinational logic truth table.*

1. MULTIPLEXERS AND DEMULTIPLEXERS

A special use of decoders/encoder ICs is *multiplexing*, which, in an electrical sense, means using one data line to transmit several signals; apparently simultaneously. Several methods of multiplexing are possible, including time division and frequency division multiplexing. Time division multiplexing refers to that arrangement in which each input is given access to the transmission line for a specific time interval, and the receiver is synchronised to receive the signal and connect it to its intended output for this period. The block diagram of Fig.1 shows the principles of time division multiplexing for 4 inputs.

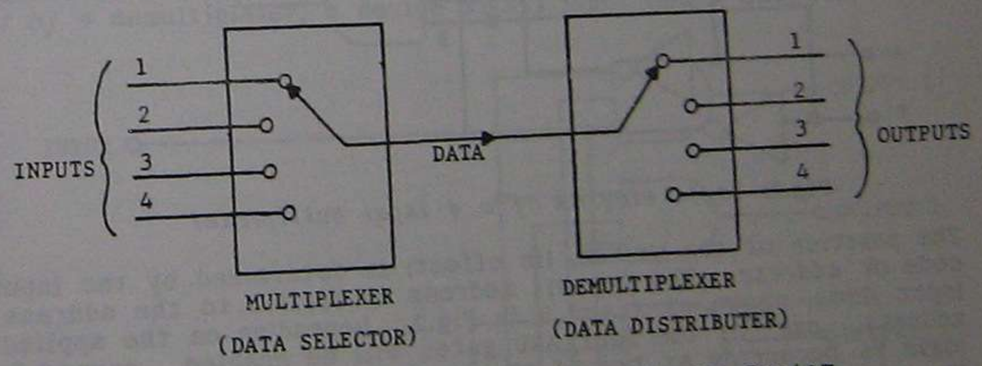


Fig.1: *Block Diagram of a 4 way Multiplexing System*

If both the 4-pole, single-way switches of Fig.1 are moving in synchronism, each input will be connected only to its corresponding output. If the switching speed is fast enough, each input/output pair will appear to be permanently connected.

Multiplexing is often used in computers to transmit data over a single line. In this application, a single wire, e.g., a printed circuit board track, is used to transmit a number of different signals from a variety of outputs to their particular input. This saves wires (or PCB tracks), but makes the circuit more complex.

Logic Device Terminal Characteristics

Logic Level | Supply voltage

Logic Level

Logic levels are the voltage levels for logic high and logic low.

V_{OHmin} – The minimum output voltage in HIGH state (logic 1)
V_{OHmin} is 2.4V for TTL
4.9V for CMOS

V_{OLmax} – maximum output voltage in Low state. (logic 0)

V_{OLmax} = 0.4V for TTL
0.1V for CMOS

V_{IHmin} – minimum Input voltage guaranteed to be recognised as logic 1 V_{IHmin} is 2V for TTL
3.5V for CMOS

V_{ILmax} – maximum input voltage guaranteed to be recognised as logic 0.

$V_{IL\,max}$ is 0.8V for TTL
1.5V for CMOS

Current Levels

I_{OHmax} = maximum current the output can source in HIGH state while still maintaining the output voltage above V_{OHmin}.

I_{OLmax} = maximum current the output can sink in Low state while still maintaining the output

voltage below $V_{OL\,max}$

$I_{I\,max}$ - The maximum current that flows in to an input in any state (1μA for CMOS)

t_r (rise time)

The time required for the output voltage to increase from $V_{IL\,max}$ to $V_{IH\,min}$

t_f (fall time)

The time required for the output voltage to decrease for $V_{IH\,min}$ to $V_{IL\,max}$

t_p (propagation delay)

The time between the logic transition on an input and the corresponding logic transition on the output for the logic gate. The propagation delay is measured at mid point.

Power Dissipation

Each gate is connected to a power supply. V_{CC} (V_{DD} in the case of CMOS). It draws a certain amount of current during its operation. Since each gate can be in a High, Transition or Low states, there are three different currents drawn from power supply.

Ps (Static Power Dissipation)

Power consumed when the output or input are not changing or rather when clock is turned off.
Normally static power dissipation is caused by leakage current. (transistor size below 90n-m, leakage current could be as high as 40% of Total Power dissipation)

Pd (Dynamic Power Dissipation)

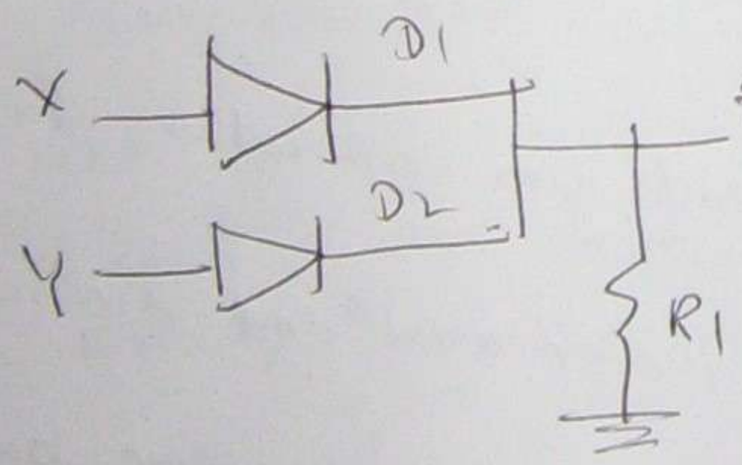
Power consumed during output and input transitions.
Pd - actual power consumed. Total power consumed by transistor + leakage current.

Thus

Total power dissipation = Static Power dissipation + dynamic Power dissipation

Diode Logic (DL)

Diode + Resistor



Diode needs forward bias to conduct

D_1 or D_2 gets forward bias, they conduct
when any diode conducts, (Z) output is high

Points
- Diode Logic suffers from voltage degradation from one stage to the next
- Diode Logic Only Permits OR and AND functions
- Diode Logic is used extensively but not in Integrated circuits.

Ps (Static Power Dissipation)

Power consumed when the output or input are not changing or rather when clock is turned off.
Normally static power dissipation is caused by leakage current. (transistor size below 90n-m, leakage current could be as high as 40% of Total Power dissipation)

Pd (Dynamic Power Dissipation)

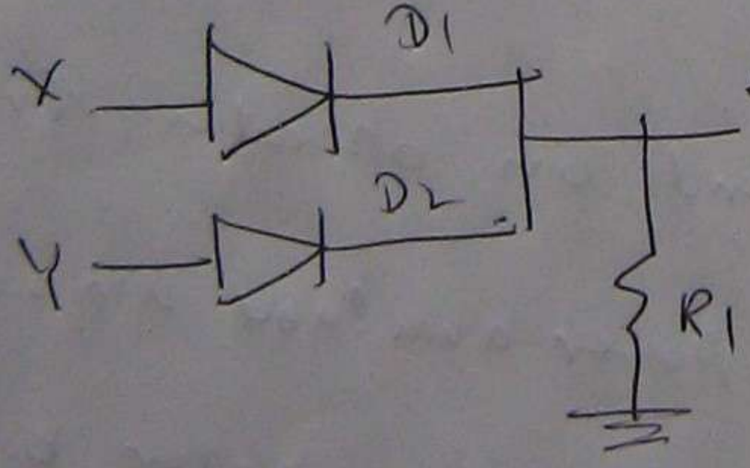
Power consumed during output and input transitions.
Pd - actual power consumed. Total power consumed by transistor + leakage current.

Thus

Total Power dissipation = Static Power dissipation + dynamic Power dissipation

Diode Logic (DL)

Diode + Resistor Diode needs forward bias to conduct



D_1 or D_2 gets forward bias, they conduct
when any diode conducts, (Z) output is high

Points
- Diode Logic suffers from voltage degradation from one stage to the next
- Diode Logic Only Permits OR and AND functions
- Diode Logic is used extensively but not in Integrated circuits.

I_{CCH} - current drawn during HIGH state

I_{CCT} - current drawn during HIGH to LOW
low to HIGH transition

I_{CCL} - current drawn during LOW state.

for TTL I_{CCT} - Transition current is negligible, in comparison to I_{CCH} and I_{CCL} if we assume that I_{CCH} and I_{CCL} are equal.

Average Power Dissipation = $\frac{V_{cc} \times (I_{CCH} + I_{CCL})}{2}$

For CMOS I_{CCH} and I_{CCL} are negligible.
In comparison to I_{CCT}.

Average Power dissipation is calculated as below

Average Power Dissipation = $V_{cc} \times I_{CCT}$

TTL Power dissipation does not depend on frequency of operation

CMOS Power dissipation depends on operation frequency current & Power available in battery $\simeq$ constant

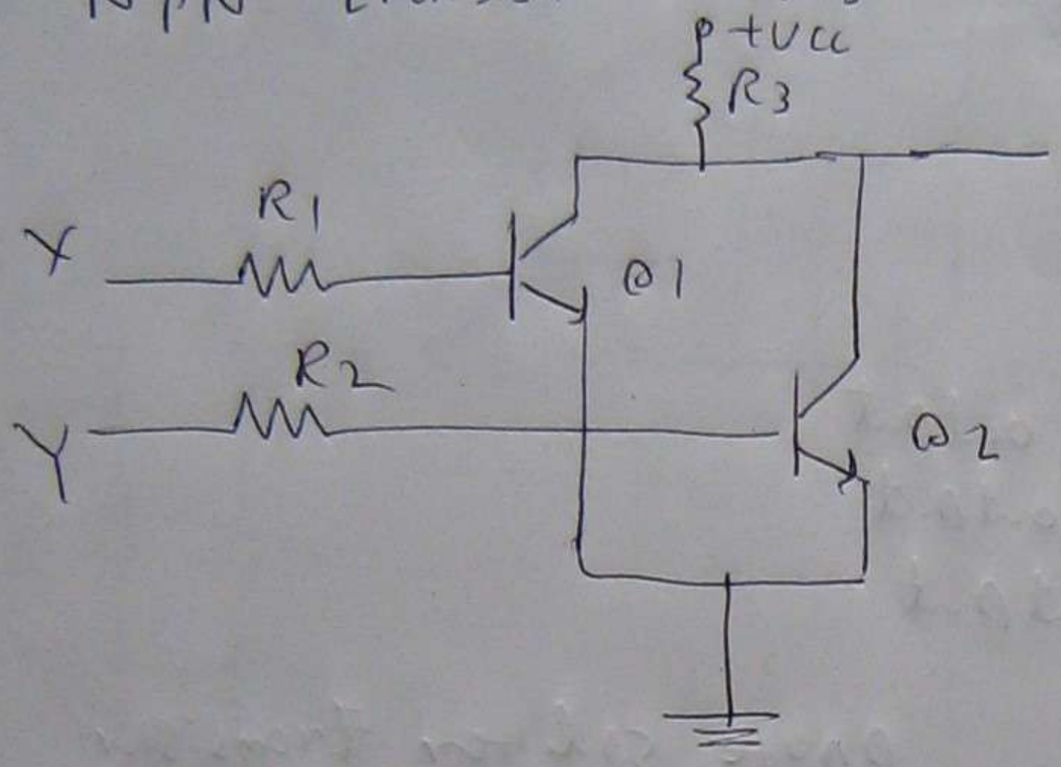
Power dissipation

The greater Power dissipation - the shorter battery life
Power dissipation is proportional to the heat generated by chip or system. Excessive heat dissipation may increase operating temperature and cause gate circuitry to drift out of its normal operating range.

Resistor Transistor Logic (RTL)

Resistor | Transistor.

NPN transistor — High at input causes output to be Low (Like inverter)



NOR Gate

When either input X or Y is high, the corresponding transistor goes to saturation and output is pulled to Low

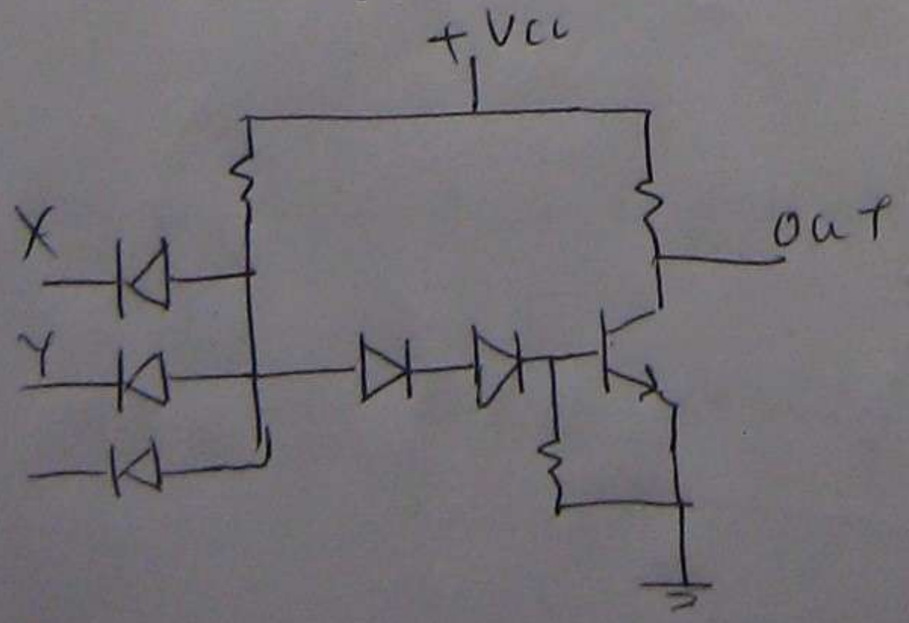
Diode Transistor Logic (DTL)

AND gate

— X, Y, Z input low, corresponding diode conduct current through 4.7k resistor.

— No current through the diode connected in series to transistor base: Transistor does not conduct. output is HIGH

— when X, Y, Z are high, the diodes in series conduct all driving transistor in to saturation, output is low



Transistor Transistor Logic

- Standard TTL
- High Speed TTL
- Low Power TTL
- Schottky TTL

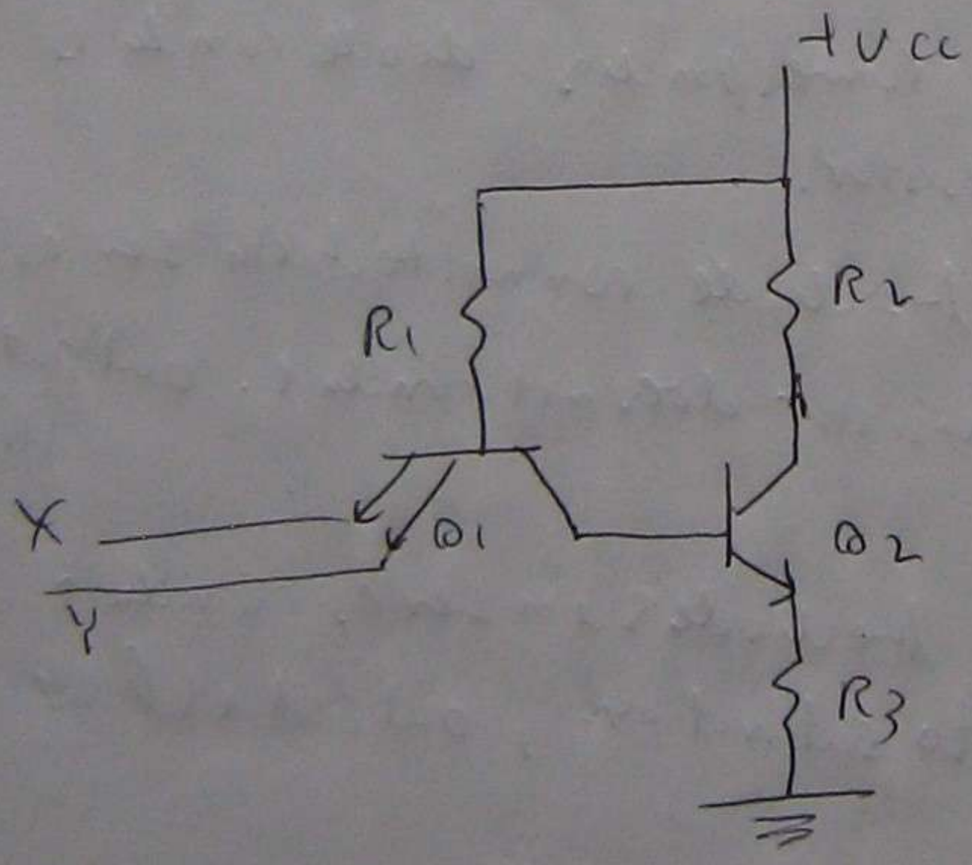
TTL configuration for outputs
- Totem - pole output
- Open collector output
- Tristate output

Input transistor & phase splitter transistor

Input stage → multi emitter transistor

When any input is driven low, the emitter base junction is forward biased and input transistor conducts.

This turn drives the phase splitter in to cut off

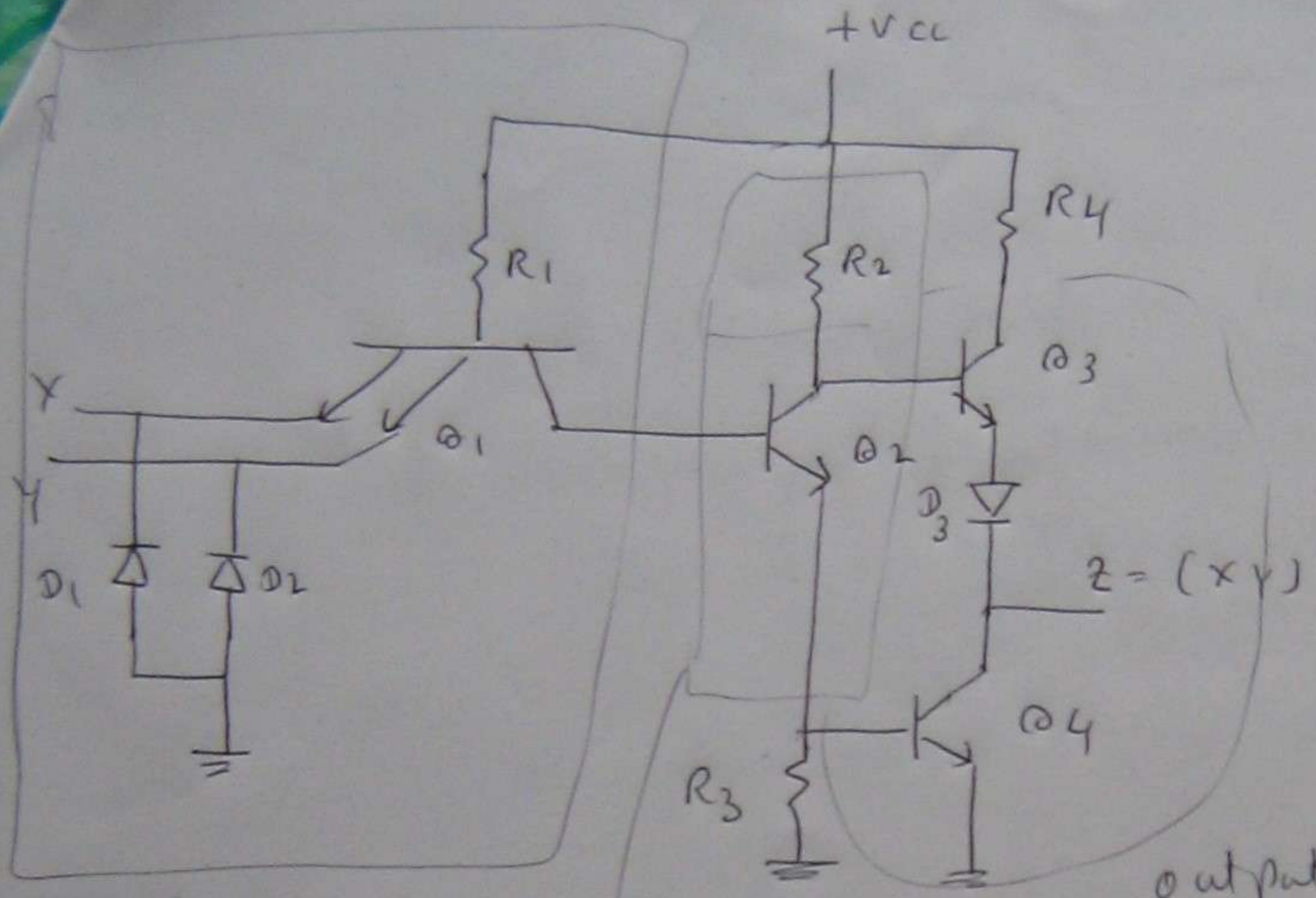


Totem - pole output

Totem pole NAND gate.
- Input Stage
- Phase Splitter Stage
- output stage

Totem pole NAND gate

WK 11—7



output stage Totempole because transistor Q_3 sits on upon Q_4

- Q_2 Provides complementary voltage to Q_3 and Q_4

Q_3 / Q_4	one conducts other cutoff

- Q_4 – pull down transistor – It pulls the output voltage down when it saturates & the other is cut off
- Q_3 – pull up transistor – It pulls the voltage up when it saturates & the other is cut off

D_1 D_2
Diode · D_3 – Protection diode – They conduct when there is large negative voltage at input shorting to ground.

Exercise

(1) How do you understand $V_{OH\,min}$, $V_{OL\,max}$, $V_{IH\,min}$ $V_{IL\,max}$, $I_{OH\,min}$ $I_{OL\,max}$

(2) Sketch ~~D-L~~, Diode logic, Resistor Transistor Logic

Diode Transistor Logic

(3) Explain operation of Totem pole Nand gate

(4) Express formula to calculate average power dissipation.

Week (11) (1)

Logic Device Terminal Characteristics
Logic Level | supply voltage

Logic Level

Logic Levels are the voltage levels for logic high and logic low.

V_{OHmin} - The minimum output voltage in HIGH state (logic 1) V_{OHmin} is 2.4V for TTL
4.9V for CMOS

V_{OLmax} - maximum output voltage in Low state. (logic 0)

V_{OLmax} = 0.4V for TTL
0.1V for CMOS

V_{IHmin} - minimum Input voltage guaranteed to be recognised as logic 1 V_{IHmin} is 2V for TTL
3.5V for CMOS

V_{ILmax} - maximum input voltage guaranteed to be recognised as logic 0.

V_{ILmax} is 0.8V for TTL
1.5V for CMOS

Current Levels

I_{OHmax} = maximum current the output can source in HIGH state while still maintaining the output voltage above V_{OHmin}.

I_{OLmax} = maximum current the output can sink in Low state while still maintaining the output

voltage below $V_{OL\,max}$

$I_{I\,max}$ - The maximum current that flows in to an input in any state (1μA for CMOS)

t_r (rise time)

The time required for the output voltage to increase from $V_{IL\,max}$ to $V_{IH\,min}$

t_f (fall time)

The time required for the output voltage to decrease for $V_{IH\,min}$ to $V_{IL\,max}$

t_p (propagation delay)

The time between the logic transition on an input and the corresponding logic transition on the output for the logic gate. The propagation delay is measured at mid point.

Power Dissipation

Each gate is connected to a power supply. V_{CC} (V_{DD} in the case of CMOS). It draws a certain amount of current during its operation. Since each gate can be in a High, Transition or Low states, there are three different currents drawn from power supply.

Ps (Static Power Dissipation)

Power consumed when the output or input are not changing or rather when clock is turned off.
Normally static power dissipation is caused by leakage current. (transistor size below 90n-m, leakage current could be as high as 40% of Total Power dissipation)

Pd (Dynamic Power Dissipation)

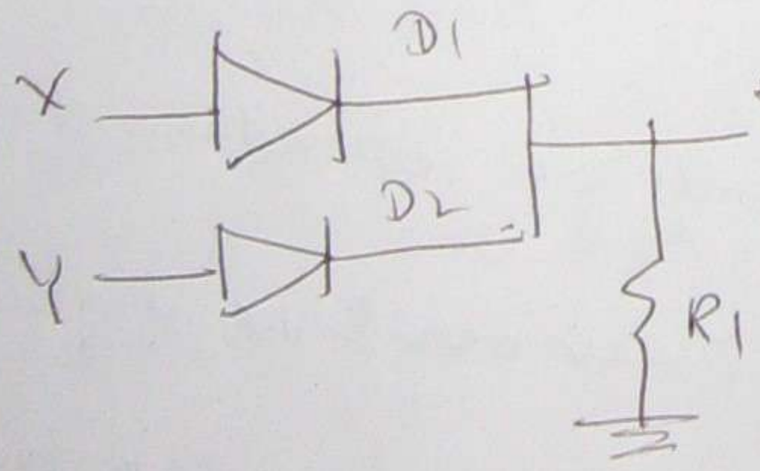
Power consumed during output and input transitions.
Pd - actual power consumed. Total power consumed by transistor + leakage current.

Thus

Total power dissipation = Static Power dissipation + dynamic Power dissipation

Diode Logic (DL)

Diode + Resistor Diode needs forward bias to conduct



D1 or D2 gets forward bias, they conduct
when any diode conducts, (Z) output is high

Points
- Diode Logic suffers from voltage degradation from one stage to the next
- Diode Logic Only Permits OR and AND functions
- Diode Logic is used extensively but not in Integrated circuits.

I_{CCH} - current drawn during HIGH state

I_{CCT} - current drawn during HIGH to LOW
LOW to HIGH transition

I_{CCL} - current drawn during LOW state.

for TTL I_{CCT} - Transition current is negligible.
in comparison to I_{CCH} and I_{CCL} if we assume that I_{CCH} and I_{CCL} are equal.

Average Power Dissipation = $\frac{V_{cc} \times (I_{CCH} + I_{CCL})}{2}$

For CMOS I_{CCH} and I_{CCL} are negligible.
Incomparison to I_{CCT}.

Average Power dissipation is calculated as below

Average Power Dissipation = $V_{cc} \times I_{CCT}$

TTL Power dissipation doesnot depend on frequency of operation.

CMOS Power dissipation depends on operation frequency
current & Power available in battery ≃ constant

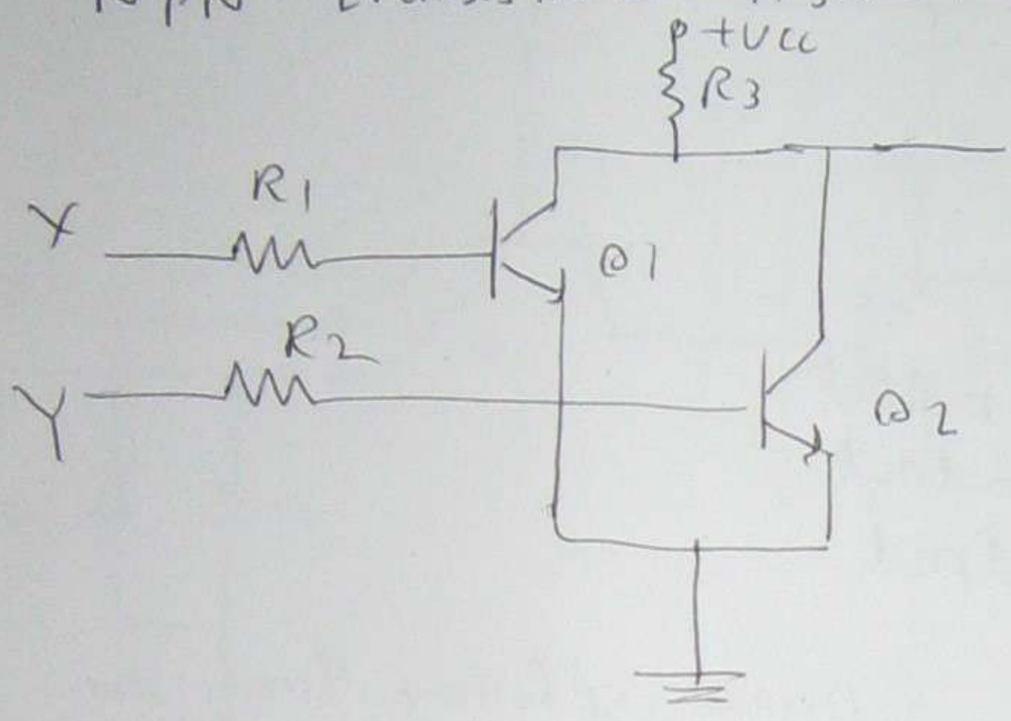
Power dissipation

The greater Power dissipation - the shorter battery life
Power dissipation is proportional to the heat generated by chip or system. Excessive heat dissipation may increase operating temperature and cause gate circuitry to drift out of its normal operating range.

Resistor Transistor Logic (RTL)

Resistor | Transistor.

NPN transistor - High at input causes output to be Low (Like inverter)



$Z = (X + Y)$

NOR Gate

When either input X or Y is high, the corresponding transistor goes to saturation and output is pulled to Low

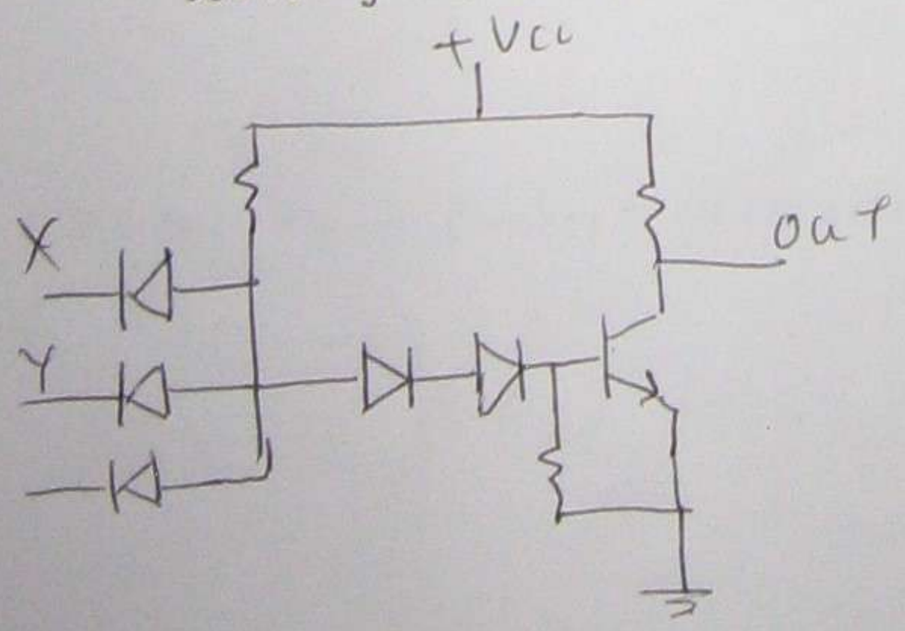
Diode Transistor Logic (DTL)

AND gate

— X, Y, Z input low, corresponding diode conducts current through 4.7k resistor.

— No current through transistor base: Transistor does not conduct. output is HIGH

— when X, Y, Z all are high, the diodes in series conduct driving transistor in to saturation, output is low



Transistor Transistor Logic

- Standard TTL
- High Speed TTL
- Low Power TTL
- Schottky TTL

TTL configuration for outputs

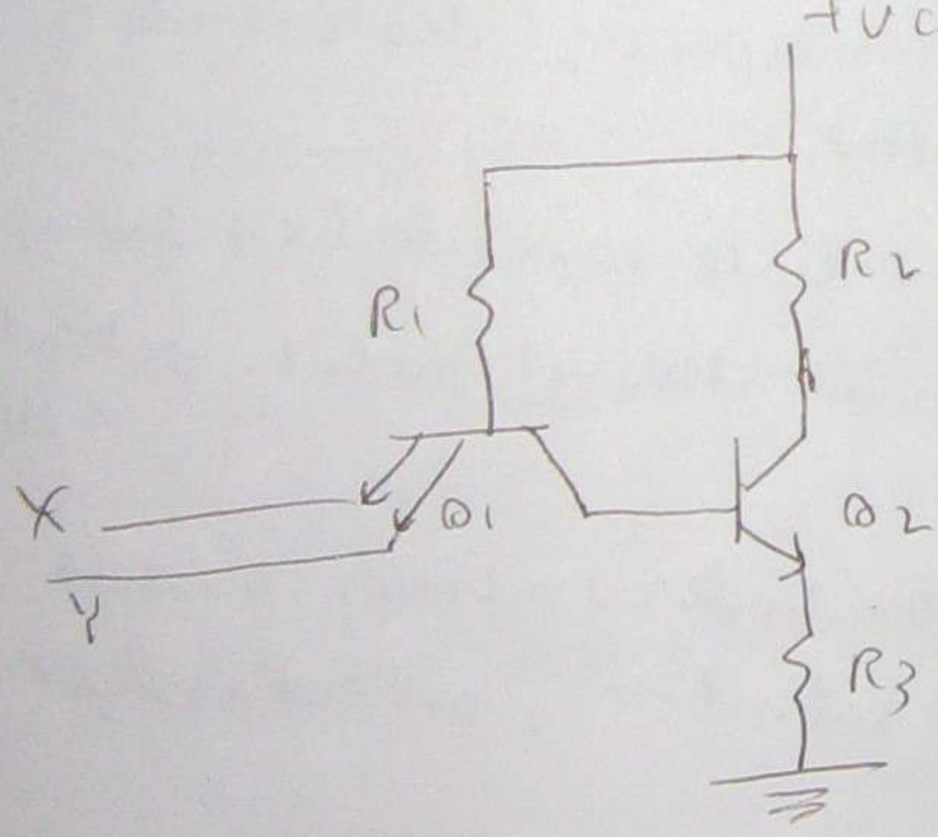
- Totem - pole output
- open collector output
- Tristate output

Input transistor & phase splitter transistor

Input stage → multi emitter transistor

When any input is driven low, the emitter base junction is forward biased and input transistor conducts.

This turn drives the phase splitter in to cut off



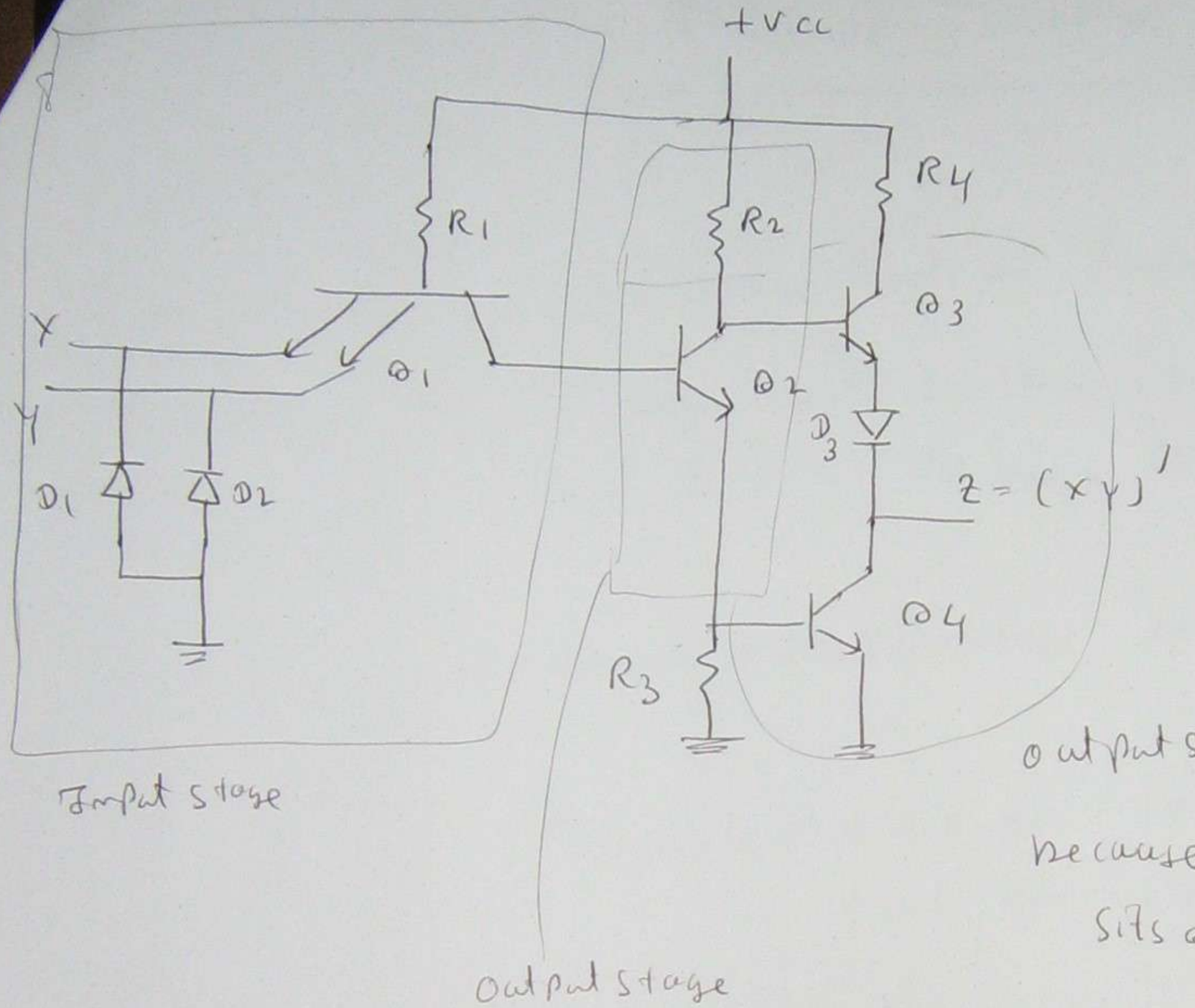
Totem - pole output

Totem pole NAND gate.

- Input Stage
- Phase Splitter Stage
- output stage

Totem pole NAND gate

WK 11—7



output stage Totempole

because transistor Q_3 sits ~~on~~ upon Q_4

- Q_2 provides complementary voltage to Q_3 and Q_4

  Q_3 | Q_4 — one conducts other cutoff

- Q_4 – pull down transistor – It pulls the output voltage down when it saturates & the other is cut off

  Q_3 – pull up transistor – It pulls the voltage up when it saturates & the other is cut off

  D_1 & D_2 Diode . D_3 – Protection diode – They conduct when there is large negative voltage at input shorting to ground.

Exercise

(1) How do you understand $V_{OH\,min}$, $V_{OL\,max}$, $V_{IH\,min}$ $V_{IL\,max}$, $I_{OH\,min}$ $I_{OL\,max}$

(2) Sketch ~~D-L~~, Diode Logic, Resistor Transistor Logic Diode Transistor Logic

(3) Explain operation of Totem pole Nand gate

(4) Express formula to calculate average power dissipation.

ADVANCED CERTIFICATE IN ELECTRONICS

ENCODERS

encoder produces at its outputs a specific code for each
ut that is activated. If a decimal to BCD encoder was
uired, 10 inputs and 4 outputs would be needed making a 10
to 4 line encoder. Activating each input, *one at a time*,
ld cause the BCD code to appear at the 4 output terminals
the particular value input. The block diagram of a BCD
oder is shown in Fig.4. A problem with such a device is the
sibility of more than one input being active at a time
ucing erroneous codes at the output. A device that
rcomes this is the *PRIORITY* encoder, as it only responds to
highest value input that is active it gives priority to this
over all lower order inputs.

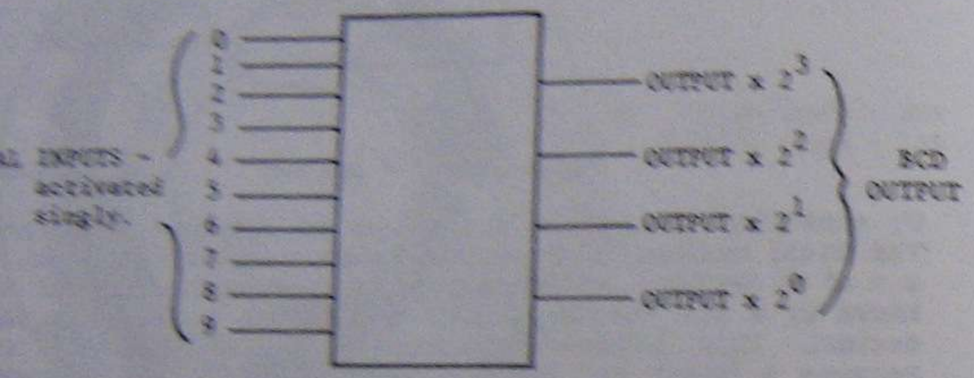


Fig.4: *Block Diagram of a BCD Encoder*

****

ADVANCED CERTIFICATE IN APPLIED INDUSTRIAL ELECTRONICS

YEAR 1 DIGITAL PRINCIPLES

THEORY LESSON 10

Written by Peter Phillips, 1990

MULTIPLEXERS

OBJECTIVES: At the end of this lesson you should be able to:

(a) *Draw the block diagram of a 4 way multiplexing system, and describe the principle of time division multiplexing.*
(b) *Draw the logic diagrams of a 4 way multiplexer and a 4 way demultiplexer.*
(c) *Show how a multiplexer is able to implement a given combinational logic truth table.*

1. MULTIPLEXERS AND DEMULTIPLEXERS

A special use of decoders/encoder ICs is *multiplexing*, which, in an electrical sense, means using one data line to transmit several signals: apparently simultaneously. Several methods of multiplexing are possible, including time division and frequency division multiplexing. Time division multiplexing refers to that arrangement in which each input is given access to the transmission line for a specific time interval, and the receiver is synchronised to receive the signal and connect it to its intended output for this period. The block diagram of Fig.1 shows the principles of time division multiplexing for 4 inputs.

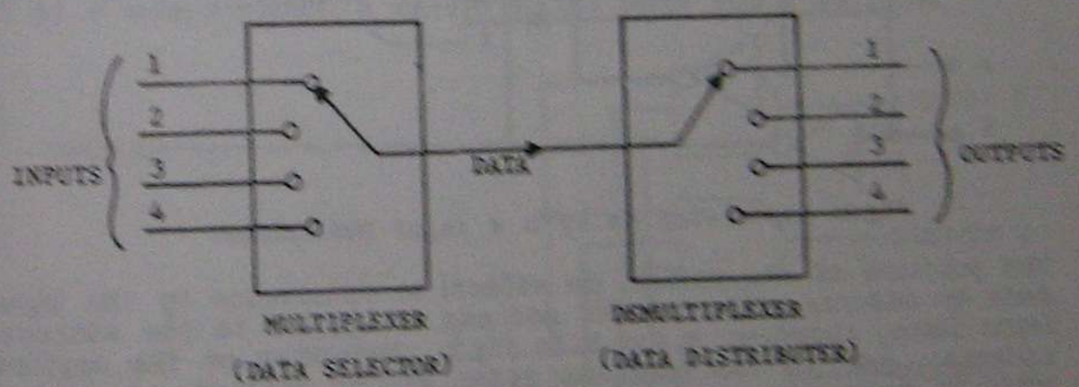


Fig.1: *Block Diagram of a 4 way Multiplexing System*

If both the 4-pole, single-way switches of Fig.1 are moving in synchronism, each input will be connected only to its corresponding output. If the switching speed is fast enough, each input/output pair will appear to be permanently connected.

Multiplexing is often used in computers to transmit data over a single line. In this application, a single wire, e.g., a printed circuit board track, is used to transmit a number of different signals from a variety of outputs to their particular input. This saves wires (or PCB tracks), but makes the circuit more complex.

The component parts are therefore the *multiplexer*, to switch each signal to the common line, and a *demultiplexer* to distribute the signals to their particular input.

1.1 MULTIPLEXERS (DATA SELECTORS)

A multiplexer or *data selector*, is a form of encoder, except it has only one output. The multiplexer of Fig.1 shows the principle of operation, in which a switch connects between the inputs to the single output. Analog multiplexers, used to switch a number of audio signals operate almost identically to the switch analogy, using analog switches to perform the switching function. In a digital multiplexer, because the only levels being switched are logic 1's and 0's, gates are used. The circuit of Fig.2 shows the logic diagram of a 4 input multiplexer.

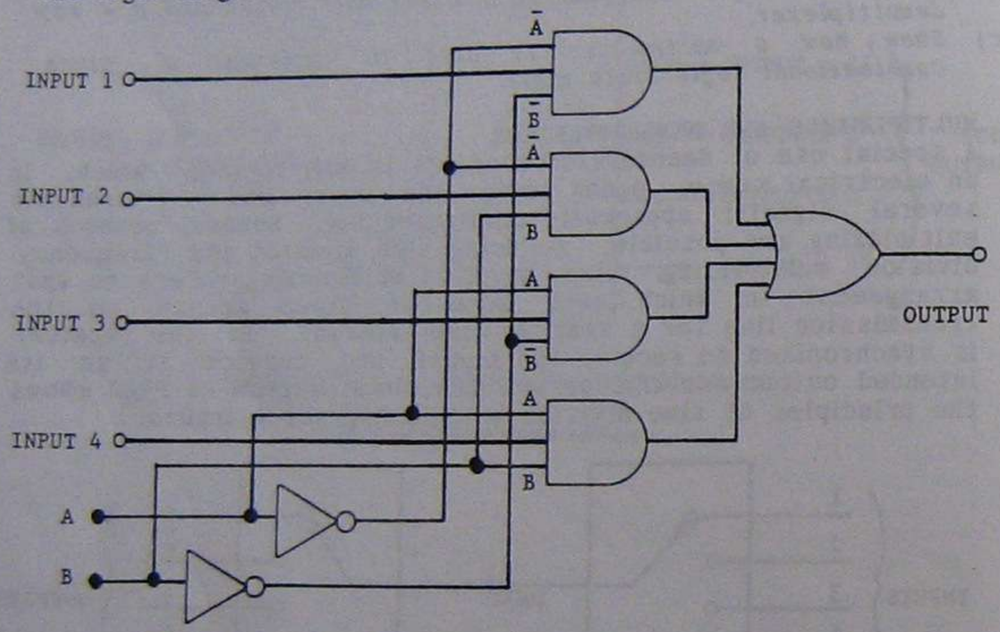


Fig.2: *Logic Diagram of a 4 input Multiplexer*

The position of the switch, (in effect) is determined by the input code or *address*. The two bit address is applied to the address input lines, shown as A and B in Fig.2. Depending on the applied address, *one* of the four AND gates will be enabled, caused by logic 1's occurring at two of the three inputs to that AND gate. The output of the AND gate will now follow its input signal. Note that the output of each AND gate is fed to an OR gate, the output of which follows the data at the selected input.

The number of address lines depends on the number of inputs. The 16 input version, type 74150 has 4 address lines. Other types include the QUAD 2-input device such as the 74LS257, which contains four by two input multiplexers. Other varieties are the 74LS153 which is a dual 4 input device and the 74LS151 or 74LS251 which are both 8 input multiplexers. The 74LS251 and the 74LS253 are dual 4 input multiplexers with three state outputs. Table 1 lists the more readily available TTL types.

TABLE 1. *MULTIPLEXERS*

TYPE	*DEVICE No.*	*OUTPUT*
Quad 2-input	74LS157	true output
Quad 2-input	74LS158	inverted output
Quad 2-input	74LS257	true output, 3S
Quad 2-input	74LS258	inverted output, 3S
Quad 2-input	74LS298	latched output, true
Dual 4-input	74LS153	true output
Dual 4-input	74LS253	true output, 3S
Dual 4-input	74LS352	inverted output
Dual 4-input	74LS353	inverted output, 3S
8-input	74LS151	inverted & true outputs
8-input	74LS251	inverted & true outputs, 3S
8-input	74LS152	inverted output only
16-input	74150	inverted output only

NOTE: true output = output not inverted, 3S = 3 state output.

1.2 DEMULTIPLEXING (DATA DISTRIBUTERS)

The demultiplexer is the opposite to a multiplexer, in that it receives the multiplexed data and redistributes it to the intended device. The block diagram of a demultiplexer is shown in Fig.1, depicted by a switch connecting the single input to one of four outputs. This device is a form of decoder, except that the input is also the data input. Therefore, like the multiplexer, the demultiplexer is a switching device. The position of the 'switch' is determined by the address code applied to address lines A and B as for the multiplexer. Fig.3 shows the logic diagram of a *1 of 4* demultiplexer, a device with 1 input and 4 outputs.

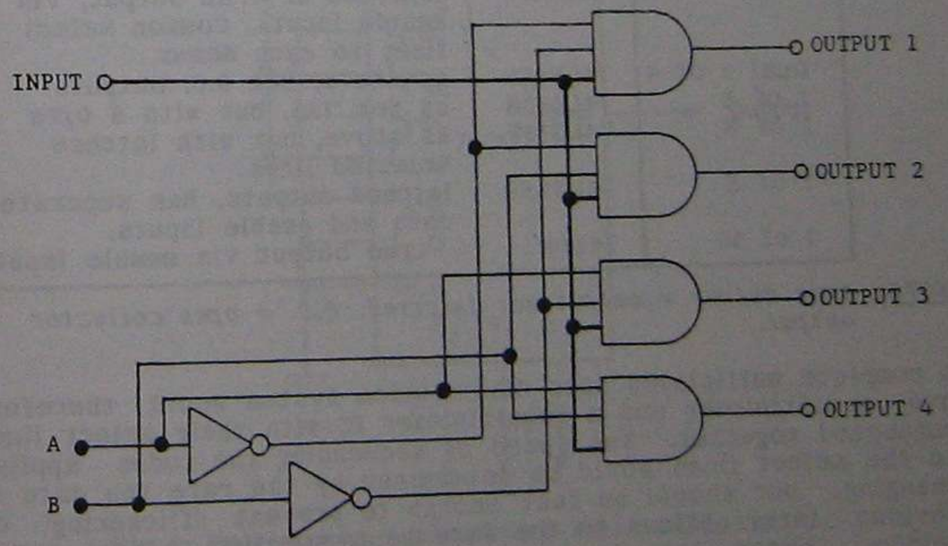


Fig.3: *Logic Diagram of a 1 of 4 Demultiplexer*

The component parts are therefore the *multiplexer*, to switch each signal to the common line, and a *demultiplexer* to distribute the signals to their particular input.

1.1 MULTIPLEXERS (DATA SELECTORS)

A multiplexer or *data selector*, is a form of encoder, except it has only one output. The multiplexer of Fig.1 shows the principle of operation, in which a switch connects between the inputs to the single output. Analog multiplexers, used to switch a number of audio signals operate almost identically to the switch analogy, using analog switches to perform the switching function. In a digital multiplexer, because the only levels being switched are logic 1's and 0's, gates are used. The circuit of Fig.2 shows the logic diagram of a 4 input multiplexer.

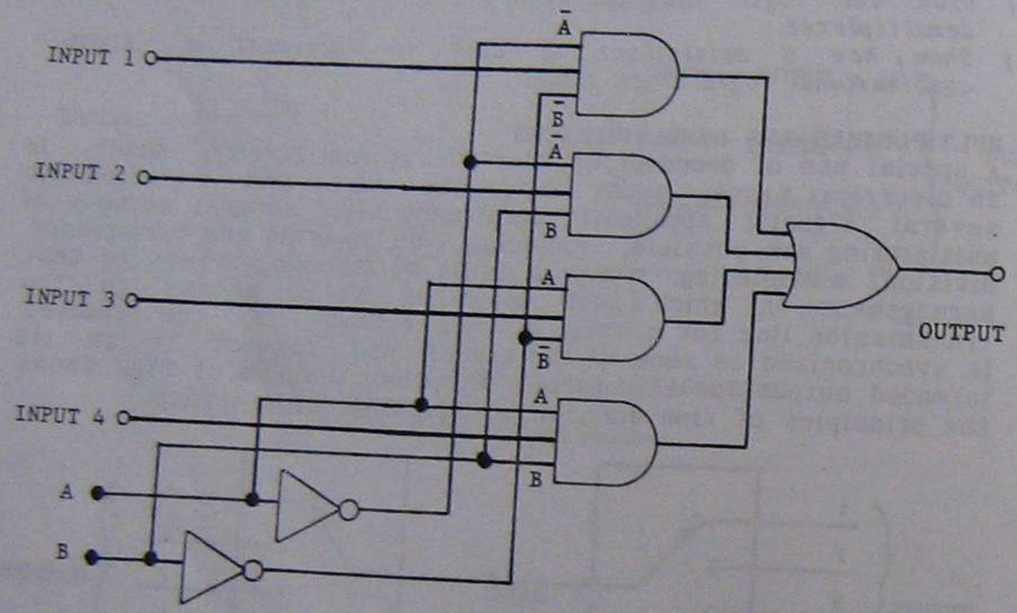


Fig.2: *Logic Diagram of a 4 input Multiplexer*

The position of the switch, (in effect) is determined by the input code or *address*. The two bit address is applied to the address input lines, shown as A and B in Fig.2. Depending on the applied address, *one* of the four AND gates will be enabled, caused by logic 1's occurring at two of the three inputs to that AND gate. The output of the AND gate will now follow its input signal. Note that the output of each AND gate is fed to an OR gate, the output of which follows the data at the selected input.

The number of address lines depends on the number of inputs. The 16 input version, type 74150 has 4 address lines. Other types include the QUAD 2-input device such as the 74LS257, which contains four by two input multiplexers. Other varieties are the 74LS153 which is a dual 4 input device and the 74LS151 or 74LS251 which are both 8 input multiplexers. The 74LS251 and the 74LS253 are dual 4 input multiplexers with three state outputs. Table 1 lists the more readily available TTL types.

TABLE 1. *MULTIPLEXERS*

TYPE	*DEVICE No.*	*OUTPUT*
Quad 2-input	74LS157	true output
Quad 2-input	74LS158	inverted output
Quad 2-input	74LS257	true output, 3S
Quad 2-input	74LS258	inverted output, 3S
Dual 4-input	74LS298	latched output, true
Dual 4-input	74LS153	true output
Dual 4-input	74LS253	true output, 3S
Dual 4-input	74LS352	inverted output
Dual 4-input	74LS353	inverted output, 3S
8-input	74LS151	inverted & true outputs
8-input	74LS251	inverted & true outputs, 3S
8-input	74LS152	inverted output only
16-input	74150	inverted output only

NOTE: true output = output not inverted, 3S = 3 state output.

1.2 DEMULTIPLEXING (DATA DISTRIBUTERS)

The demultiplexer is the opposite to a multiplexer, in that it receives the multiplexed data and redistributes it to the intended device. The block diagram of a demultiplexer is shown in Fig.1, depicted by a switch connecting the single input to one of four outputs. This device is a form of decoder, except that the input is also the data input. Therefore, like the multiplexer, the demultiplexer is a switching device. The position of the 'switch' is determined by the address code applied to address lines A and B as for the multiplexer. Fig.3 shows the logic diagram of a *1 of 4* demultiplexer, a device with 1 input and 4 outputs.

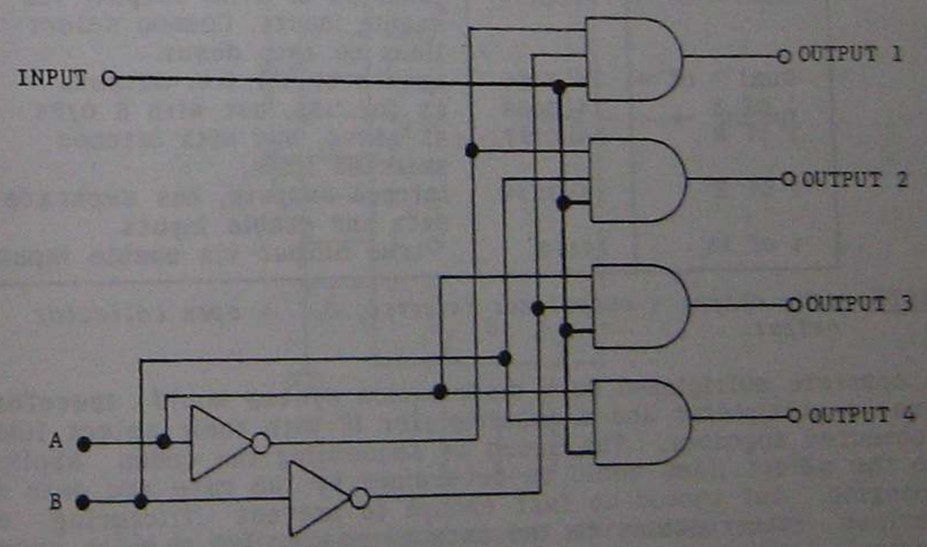


Fig.3: *Logic Diagram of a 1 of 4 Demultiplexer*

In Fig.3, the input line is connected to all the AND gates. However, only *one* AND gate at a time is selected, depending on the code applied to the A and B inputs. The selected AND gate will have a logic 1 at two of its inputs, and its output will now follow the input logic level. Thus, a logic 0 applied to both the A and B select lines will cause logic 1's to be present at two of the inputs of the top AND gate. This will allow output 1 to follow the input.

In practice, demultiplexers are a modified form of a decoder. As shown in the block diagram of Fig.3 in the theory notes for week 9, a decoder is a device that activates one output (usually by sending the selected output to a logic 0). In other words, there is no actual *input* line, rather the device simply sends the output low if that output is selected. However, a demultiplexer IC has another input line, usually called the *ENABLE* input. This input can be used to make the selected output follow input data, as the selected output will be a logic 0 if the enable input is active (usually with a logic 0). If the enable input is now bought to a logic 1, the selected output will become a logic 1. Thus, the data input terminal is generally labelled as the ENABLE input. Some ICs have two enable inputs, although only one is used for the input data when the IC is being used as a demultiplexer. The table below summarises some of the available TTL demultiplexer ICs.

TABLE 2. *DEMULTIPLEXERS*

TYPE	*DEVICE No.*	*OUTPUT*
Dual 1 of 4	74LS139	true output via enable input
Dual 1 of 4	74LS155	inverted or true output, via enable inputs. Common select lines to each demux.
Dual 1 of 4	74LS156	as above, but O.C. outputs
1 of 8	74LS138	as for '155, but with 8 O/Ps
1 of 8	74LS137	as above, but with latched selected lines.
1 of 8	74LS259	latched outputs, has separate data and enable inputs.
1 of 16	74154	true output via enable input

NOTE: true output = output not inverted, O.C. = open collector output.

A complete multiplexed data distribution system would therefore use a multiplexer and a demultiplexer IC with their select lines connected together. The speed of sequencing the codes applied to the select lines would be determined by the rate the data is changing, but should be fast enough to prevent 'flickering' or obvious interruptions to the data being supplied to each output device. Later lessons will show other applications of multiplexing.

2. OTHER USES FOR MULTIPLEXERS

Multiplexers (and demultiplexers) are useful in a range of applications other than simply transferring digital data along a common data line. A good use for a multiplexer is to implement a given truth table. Consider the truth table of Table 3.

TABLE 3

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

To implement this truth table with gates would require the designer to first extract the Boolean expression for the output. Then, to ensure the least number of gates, the expression would need to be examined for possible simplification. The final circuit could be implemented with NAND gates, but would be likely to use at least two or more ICs. A simpler and more direct method is to use a multiplexer. Because there are three inputs, giving 8 codes, an 8 input multiplexer is required. The A, B and C inputs are now connected to the select lines of the multiplexer, and each input is connected either to a 0 or a 1 as required. Thus, when the particular code for A, B and C occurs, the output (Z) will be determined by the logic level at the selected input. The solution to Table 3 is shown below in Fig.4.

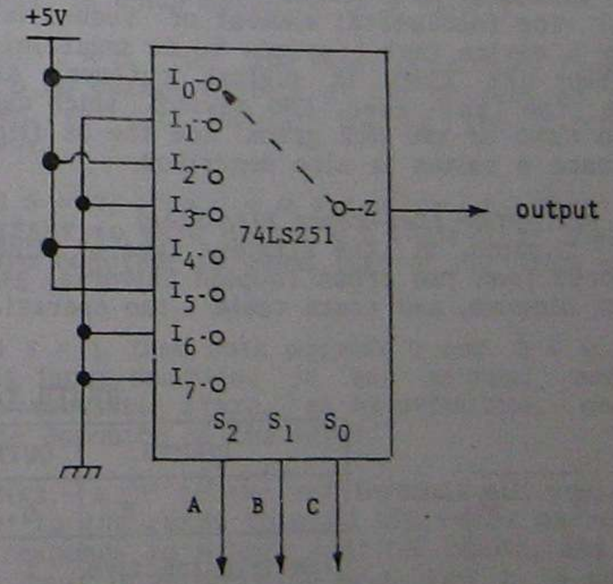


Fig.4: *Multiplexer implementing truth table 3.*

****

Schmitt Trigger Device

multivibrator is an electronic circuit used to implemented a variety of simple two state system such as oscillators, timers & flip flops.
- generate square wave, high level of harmonic,

3 types of multivibrator circuit

Astable circuit is not stable in either state. It continuously oscillates from one state to the other

monostable one of the state is stable. The other is not. The circuit will flip into the unstable state for a determined period. but will eventually return to the stable state.

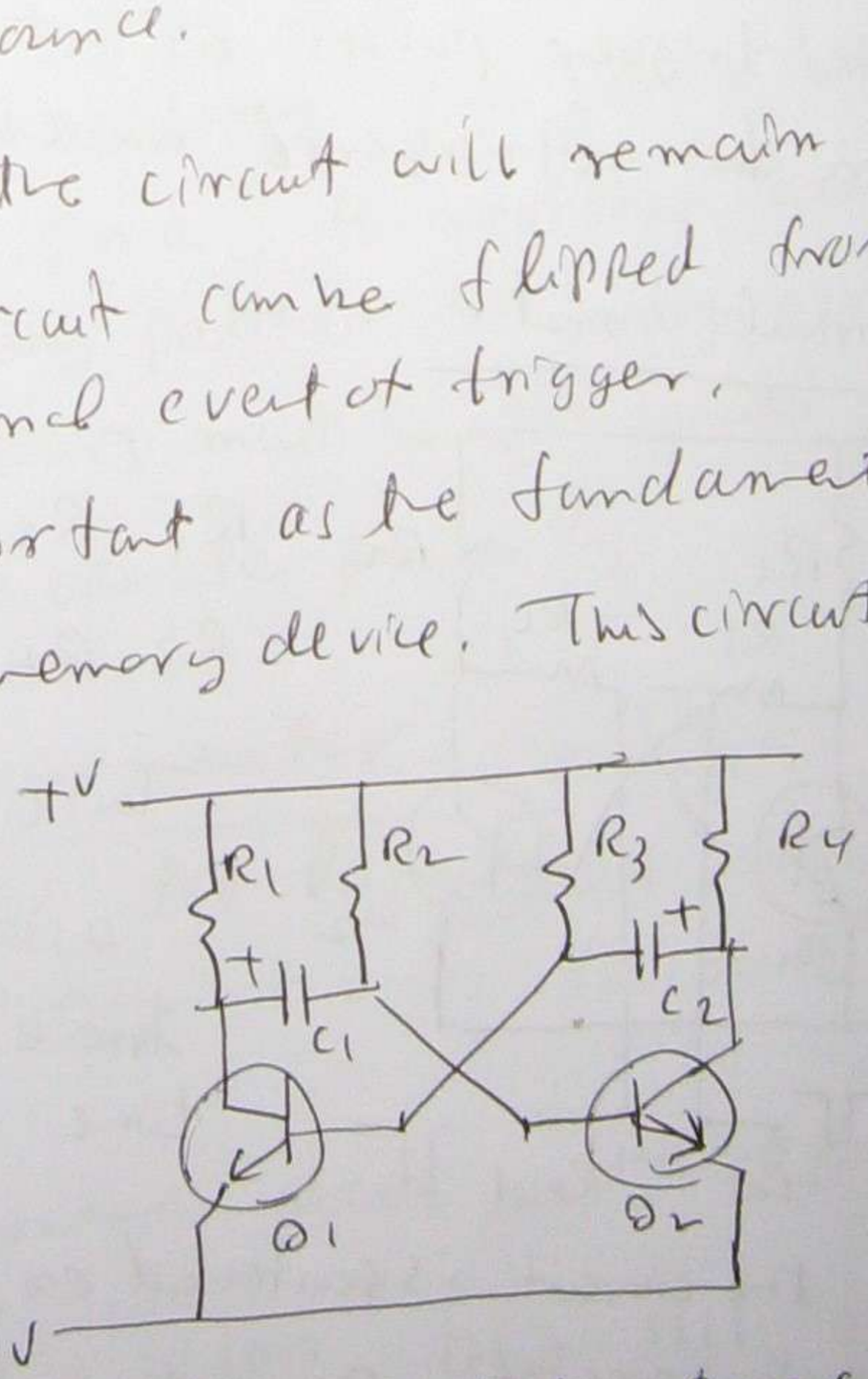
Such a circuit is useful for creating a timing period of fixed duration in response to some external event. A common application is also known as a one shot. This circuit is also is an eliminating switch bound.

Bistable in which the circuit will remain in either state indefinitely. The circuit can be flipped from one state to the other by an external event of trigger.

Such a circuit is important as the fundamental building block of a register or memory device. This circuit is also known as flip-flop

BJT astable multivibration

R_1 R_4 = 10K
R_2 R_3 = 150K
C_1 C_2 = 10μF
Q_1, Q_2 BC547 NPN transistor Switching

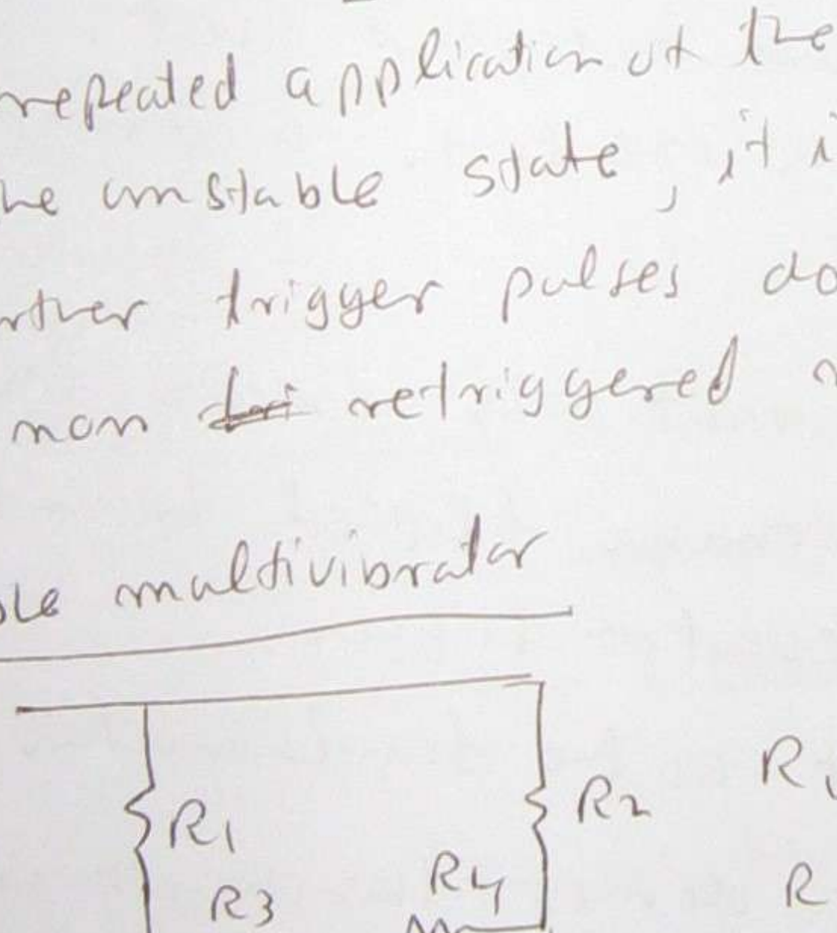


This circuit keeps one transistor switched on and the other switched off
Initially Q_1 is switched on, Q_2 is switched off

multivibrator frequency

$$f = \frac{1}{T} \simeq \frac{1}{\ln(2) \cdot RC} \simeq \frac{1}{0.693 (R_2 C_1 + R_3 C_2)}$$

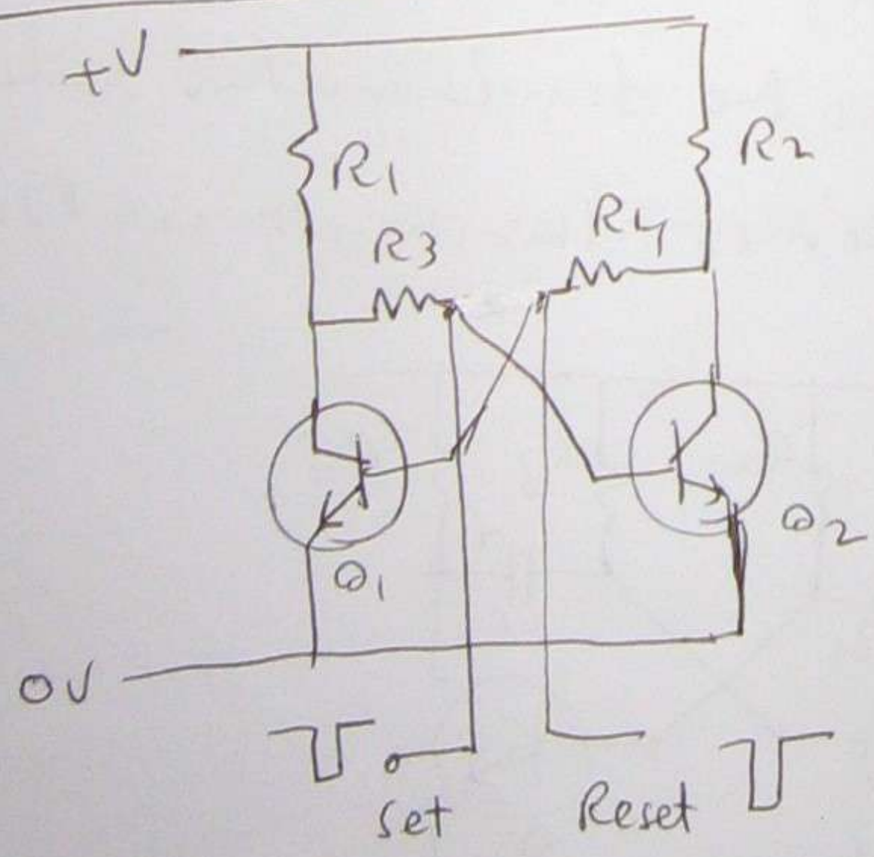
monostable vibrator



when triggered by an input pulse, a monostable multivibrator will switch to its unstable position for a period of time, and then return to its stable state.

If repeated application of the input pulse maintains the circuit in the unstable state, it is called a retriggered monostable If further trigger pulses do not affect the period, the circuit is a non ~~ret~~ retriggered multivibrator

Bistable multivibrator



R_1 R_2 = 1K

R_3 R_4 = 10K

This circuit is similar to an astable multivibrator except that there is no charge or discharge time due to absence of capacitors

Hence, when the circuit is switched on, if Q_1 is on, its collector is at 0V. As a result, Q_2 gets switched off. This results in nearly +V volts being applied to base of Q_1 thus keeping it on

Thus the circuit remains stable in a single state continuously.

Similarly Q2 remains on continuously if it happens to get switched on first

Set → set Q1 on

Reset → reset it to off state

Ring counter

A ring counter is a type of counter composed of a circular shift register. The output of the last shift register is fed to the input of the first register.

There are two types of ring counters.

A straight ring counter or overlock counter connects the output of the last shift register to the first shift register input and circulates a single one (or zero) bit around the ring. For example, in a 4 register counter, with initial values of 1000, repeating pattern is 1000, 0100, 0010, 0001 note that one of the registers must be pre loaded with a 1 (or 0) in order to operate properly.

Twisted ring counter / Johnson counter connects the complement of the output of the last shift register to its input and circulates a stream of ones followed by zeros around the ring.

4 register with initial value 0000, the repeating pattern is 0000, 1000, 1100, 1110, 1111, 0011, 0001

4

4 bit ring counter sequences

Straight ring | overbeck counter | Twisted ring | Johnson counter

Clock Pulse	cycle	Q_0	Q_1	Q_2	Q_3
0	I	1	0	0	0
1		0	1	0	0
2		0	0	1	0
3		0	0	0	1
4	II	1	0	0	0
5		0	1	0	0
6		0	0	1	0
7		0	0	0	1
8	III	1	0	0	0

Clock Pulse	cycle	Q_0	Q_1	Q_2	Q_3
0	I	0	0	0	0
1		1	0	0	0
2		1	1	0	0
3		1	1	1	0
4	II	1	1	1	1
5		0	1	1	1
6		0	0	1	1
7		0	0	0	1
8	III	0	0	0	0

week (16)

Exercise

Q_1 sketch astable multivibrator & explain operation

Q_2 ——— monostable ———

Q_3 ——— bistable ———

Q_4 what is ring counter?

7											
2	111	1	0	0	0	6	11				

week (16)

Exercise

Q1 sketch astable multivibrator & explain operation

Q2 _____ monostable _____

Q3 _____ bistable _____

Q4 what is ring counter?

ADVANCED CERTIFICATE IN APPLIED INDUSTRIAL ELECTRONICS

YEAR 1 DIGITAL PRINCIPLES

THEORY LESSON 14

Written by Peter Phillips, 1990

SHIFT REGISTERS and DATA LATCHES

OBJECTIVES: At the end of this lesson you should be able to:

(a) Describe the operation of a digital latch.
(b) Draw the circuit of a serial-in parallel-out shift register that uses D or JK flipflops and describe its operation
(c) Describe a range of shift register functions, including: serial-in serial-out, parallel-in parallel-out, parallel-in serial out and the shift left/shift right functions.

1. INTRODUCTION

Flipflops are used in a variety of applications, including the latch and shift register functions. These notes describe both these functions as well as some of the latch and shift register ICs available.

2. LATCHES

A flipflop (either D or JK) is a storage device, capable of storing one bit of information. Working with one bit of information at a time is very slow and many digital systems process information in parallel. Typical parallel groupings are 4, 8, 16 or 32 bits called nibbles, bytes, words and long words respectively. Manufacturers produce a variety of ICs that incorporate a number of interconnected flipflops capable of storing a number of bits at the one time. These are called *latches* and have a common clock or enable for latching parallel information. Fig.1 shows the 74LS75, 4 bit latch, which comprises four D flipflops sharing a common enable (G) input.

TRUTH TABLE

(each latch)

Inputs		Outputs	
D	G	Q	$\overline{Q}$
L	H	L	H
H	H	H	L
X	L	Qo	$\overline{Qo}$

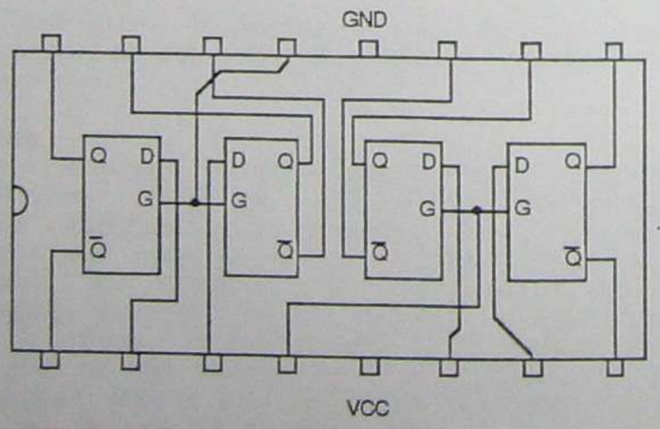


Fig.1: *74LS75 4-bit Latch*

The 74LS75 is used to store four parallel bits. Information present at each input (D) is transferred to the corresponding output (Q) when the enable (G) is high and the output will follow the data input as long as the enable remains high. (For this reason these devices are sometimes called TRANSPARENT LATCHES.) When the enable goes low, the information that was present at the data input at the time the transition occurred is retained at the Q output until the enable is permitted to go high. There is a minimum setup time of 20ns for this device which means that the D input must remain stable for at least this period before the gate is changed from a '1' to a '0'.

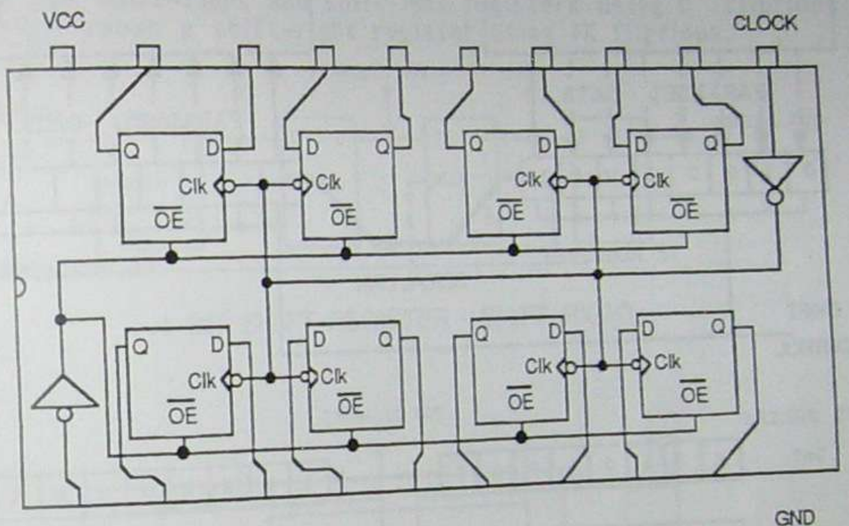


Fig.2: *74LS374 Octal D-TYPE Latch*

The 74LS374 (also the 74LS574), is often used for latching information in an 8 bit bus systems (bus = common lines). Note that this is an edge triggered device, used in microprocessor circuits which determine all timing from clock edges. It is used to latch output information from digital processing circuits.

3. SHIFT REGISTERS

Many digital applications require that binary data be shifted, either to the right or left. For example, if a binary number is shifted one position right, it is effectively divided by two. Fig.3 shows how the decimal value of 154 (10011010 in binary) becomes 77 (decimal) after one shift.

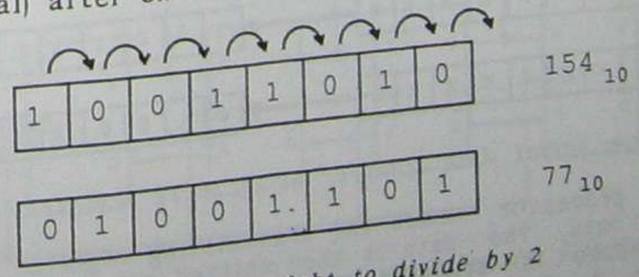


Fig.3: *Shift right to divide by 2*

Shift registers are also used in communications. Although digital processors usually work in parallel, data communications between processors over a distance is, for economic reasons done via two signal wires. This is called *serial* data transfer, in which each bit is transmitted one after another, as shown in Fig.4.

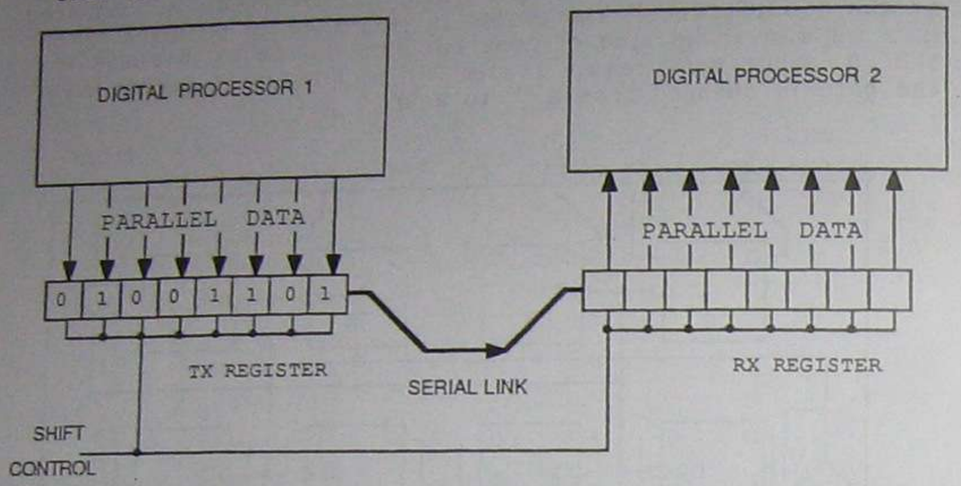


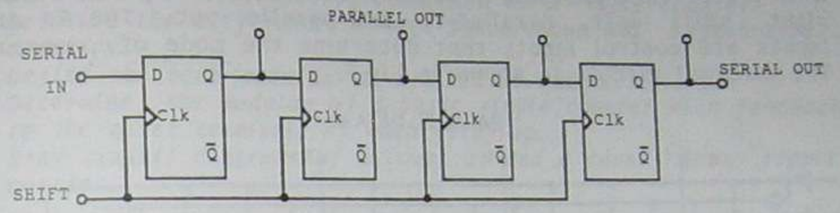
Fig.4: *Serial data transfer using shift registers*

In Fig.4, processor 1 loads the transmitting register with the 8 bits of data. The data is then shifted out one bit at a time until the data is transferred to the receive register of processor 2. This processor then performs a *parallel read* of

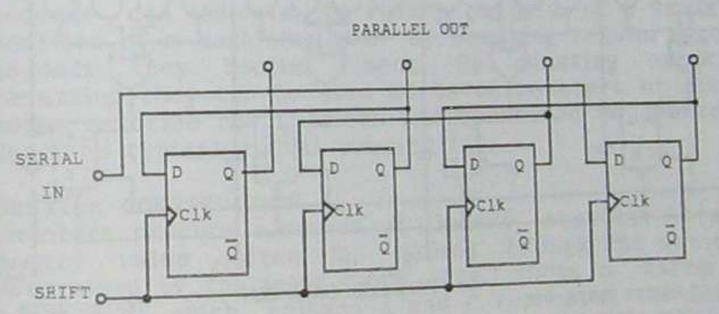
the receive register. The transmitting register is being used in the *parallel-in/serial-out* mode (PISO) while the receiving register is being used in the *serial-in/parallel-out* mode (SIPO). Shift registers can also be used for *parallel-in/parallel-out* (PIPO) and *serial-in/serial-out* (SISO) operations.

4. SHIFT REGISTER CONSTRUCTION

The shift register contains a number of D flipflops connected to a common clock. (D flipflops are made from SR & JK flipflops by connecting an inverter between the SET (J) and the RESET (K) inputs). Shifting occurs by pulsing the common clock line. Fig.5 shows the shift-right and shift-left registers using D flipflops and Fig.6 shows a shift-right register using JK flipflops.



4 BIT SHIFT REGISTER - SHIFT RIGHT



4 BIT SHIFT REGISTER - SHIFT LEFT

Fig.5: *Shift registers from D flipflops*

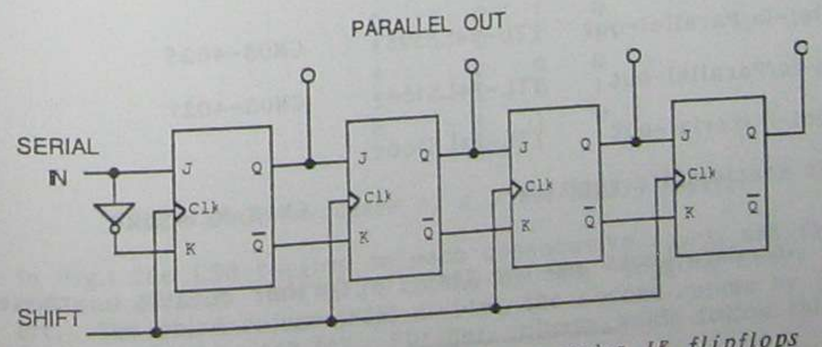


Fig.6 *A shift-right shift register using JK flipflops*

Note that the inverter in Fig.6 is only required on the first flipflop as an inverted output is available from each flipflop down the line. The parallel load function for a shift register can be achieved by using the preset and clear inputs of each flipflop, or it may use gating to the data inputs of each flipflop, in which the data input is temporarily disconnected from the preceding output.

6. COMMERCIAL SHIFT REGISTER ICS

The following is a list of some commercially available shift registers in IC form.

6.1 Parallel-In/Parallel-Out, bidirectional (TTL-74LS194, CMOS-4194).

A 16 pin IC that provides 4 registers which are capable of shift right, shift left, parallel-in and parallel-out. The S0 and SI inputs are control inputs that determine the mode of operation. The internal circuit is shown in Fig.7.

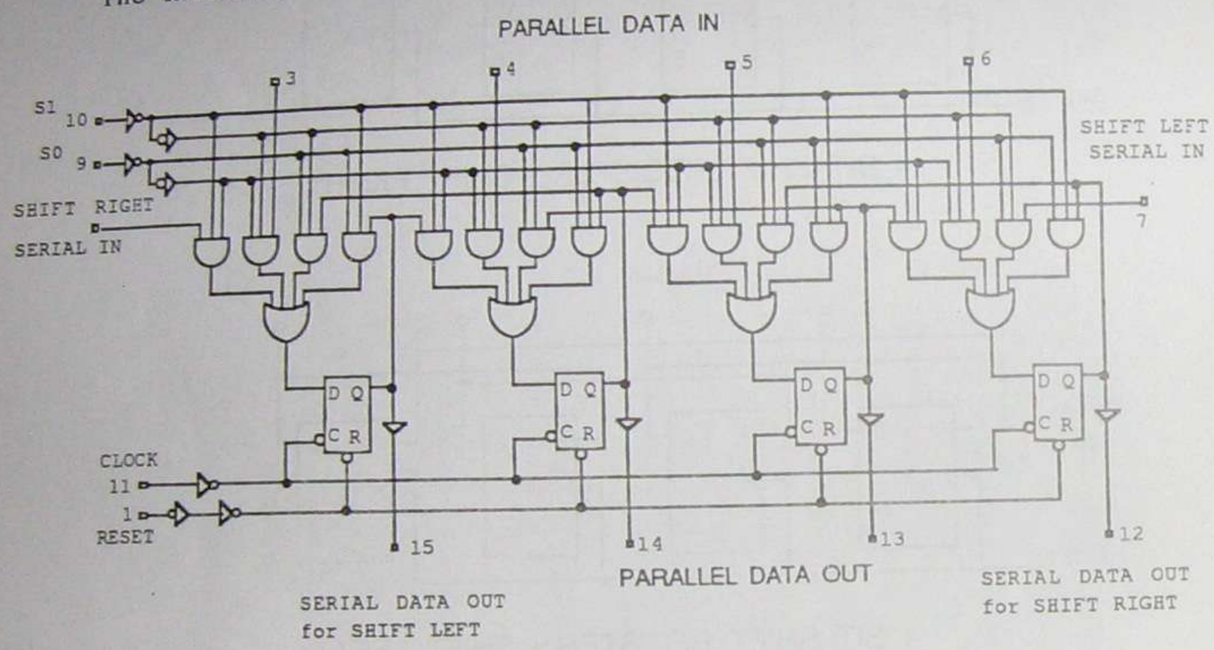


Fig.7: *Logic diagram of the '194*

6.2 Other types include the:

Parallel-in/Parallel-out	TTL-74LS195A	CMOS-4035
Serial-in/Parallel-out	TTL-74LS164	CMOS-4021
Parallel-in/Serial-out	TTL-74LS166	
128 bit static shift register		CMOS-MC14562B

Refer to data books and the text for further details on these ICs.

ADVANCED CERTIFICATE IN APPLIED INDUSTRIAL ELECTRONI

YEAR 1 DIGITAL PRINCIPLES

THEORY LESSON 15

Written by Peter Phillips, 1990

COUNTERS

OBJECTIVES: At the end of this lesson you should be able

(a) Describe an application for a digital counter.
(b) Show how JK flipflops are connected as a ripple count
(c) Describe the difference between synchronous and asyn counters.
(d) Define the term 'modulus' as applied to counters.
(e) Determine the modulus of a basic ripple counter with to the reset terminals of each flipflop.
(f) Draw circuit diagrams of a count up and a count dow counter.

1. INTRODUCTION

Counters can be used to count the number of conveyor, or if used to count the number of pulses f encoder can indicate the rotary position of a shaft position of a machining bed. By counting regular ever seconds they become timers. By counting events resetting they can be used as batch counters or divi notes describe how flipflops are connected as counte describe typical digital counter ICs.

2. COUNTER CONSTRUCTION

Counters produce a series of binary numbers, sta preset value (often 0), cycling through the serie determined by the input clock. Fig.1 shows a three count, in which columns A and C represent the lea most significant bits (MSB) respectively.

DECIMAL	C	B	A (LSB)
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

Fig.1: *Count sequence of a 3 bit binary counte*

In Fig.1 the LSB toggles on each consecutive count, column divides the first column by two, toggling bit. The third column (MSB) divides the second colu toggles every 4th bit. Further columns would fo

ADVANCED CERTIFICATE IN APPLIED INDUSTRIAL ELECTRONICS

YEAR 1 DIGITAL PRINCIPLES

THEORY LESSON 7

Written by Peter Phillips, 1990

LOGIC FAMILIES - PART 1

OBJECTIVES: At the end of this lesson you should be able to:

(a) Describe the internal circuit operation of a TTL NAND gate.
(b) List the various input and output currents and voltages that apply to the TTL logic family.
(c) Define the term 'Unit Load' as applied to the TTL family.
(d) Apply the TTL loading rules when interconnecting gates.

1. INTRODUCTION

There are two logic families in popular use:
1. TTL (Transistor - Transistor Logic)
2. CMOS (Complementary Metal Oxide Silicon - FET based)

Other logic families, such as DTL, RTL and ECL are available in limited configurations, and are not discussed in these notes. The TTL logic family is commonly used in industrial applications and comprises several varieties;

(a) TTL regular, now being superseded by other TTL types.
(b) TTL LS, currently the most widely supported family.
(c) TTL F, TTL H, TTL S, TTL L types which are available in limited IC varieties. The TTL F is the newest range, and the others have specialised applications.

Because the TTL family is more generally used, it is the main subject of these notes. The MOS family is also widely used, particularly in the area of highly complex ICs, such as microprocessors. A complete discussion of both families is beyond the scope of these notes. Instead, CMOS is treated only in a brief way to allow an appreciation of the technology.

In order that a particular type of logic IC can be used, it is necessary to know its limitations, power requirements, etc, and this is achieved by an examination of the internal circuitry of each type of family. These notes provide an outline only of the TTL regular, TTL LS, and MOS gates.

2. TTL REGULAR LOGIC

The TTL logic family is comprised mainly of the 54 and 74 series. The 74 range is the most common, and the 54 types are used for military specification applications, (i.e., -55°C to +125°C etc). Other types include the 9000, 9300 and 9600 ranges.

To be able to understand the operating limitations of a TTL IC, a knowledge of its internal circuit is required. The circuit of Fig.1 shows the schematic diagram for a 2 input NAND gate.

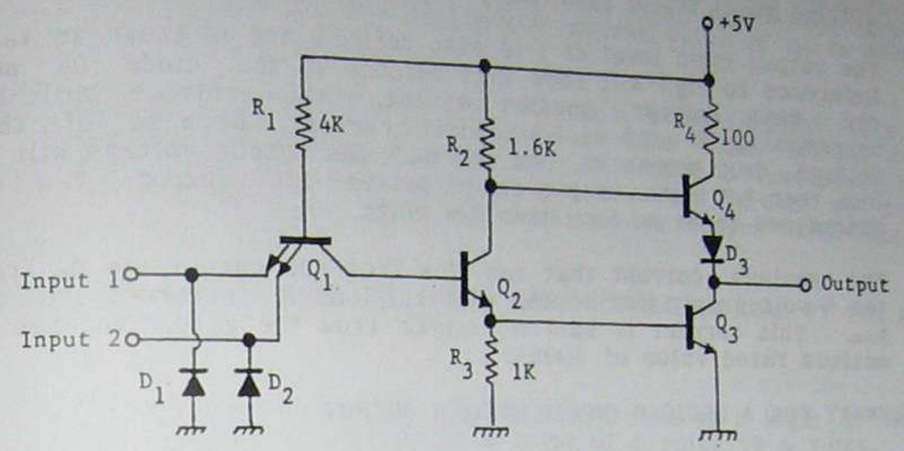


Fig.1: *Two Input TTL regular NAND Gate*

Note that Q_2 has 2 emitters, and is known as a multiple-emitter transistor. Diodes D_1 and D_2 are nearly always present at the inputs of TTL ICs, and are used to prevent high frequency ringing that can occur if the input signal has a fast rise time, or is coupled via a long lead. Precautions must be taken to prevent the input signal from being sufficiently negative to forward bias these diodes into destruction.

2.1 OPERATION WHEN A "0" IS APPLIED TO ONE INPUT

(1) If either or both inputs are connected to ground, Q_1 will be forward biased via R_1, and current will flow from the input(s) to ground. This current is referred to as I_{IL} (I with the input low) and is guaranteed not to exceed 1.6 mA. Note that the *<u>total</u>* I_{IL} flowing out of *<u>all</u>* the inputs of any one gate will not exceed 1.6 mA. The current is said to '*source*' from the IC, and the input device must therefore '*sink*' this current to ground.

(2) As a result of the current flowing in the base-emitter junction of Q_1, the voltage at the collector of Q_1 is virtually zero. This causes Q_2 to be turned off, and the collector voltage of Q_2 to be a maximum.

(3) The collector voltage at Q_2 causes Q_4 to turn on, and because there is no current flowing in Q_2, Q_3 is held off. The output is thus connected to the supply rail via Q_4 and is a logic 1; correct for a NAND gate.

For these conditions to occur, the input voltage to the gate must be a logic 0, known as V_{IL}. For TTL regular devices, the manufacturer guarantees that the gate will interpret any input voltage equal to, or less than 0.8 volts as a logic 0.

The output logic level of 1 is also defined, and is known as V_{OH}. Reference to Fig.1 will show that because of the diode D_3 and the base emitter junction of Q_4, this voltage will be approximately 5 - 1.2 = 3.8 volts. Further, because of the voltage drop across R_2, R_4 and Q_4, the output voltage will be less than 3.8 volts. For a fully loaded TTL output, V_{OH} is guaranteed to be no less than 2.4 volts.

The maximum current that can flow from the output via Q_4 from the 5 volt supply line to the external load is referred to as I_{OH}. This current is said to '*source*' from the gate, and has a maximum rated value of 800uA.

SUMMARY FOR A LOGIC 0 INPUT, LOGIC 1 OUTPUT

If the input voltage to a TTL regular gate is equal to or less than 0.8 volts (V_{IL}), a current (I_{IL}) with a value of 1.6mA or less will be sourced from the gate to ground, independent of the number of inputs connected to the logic 0. For a NAND gate (or inverter) the output will be a logic 1, with a guaranteed minimum voltage of 2.4 volts (V_{OH}), providing the current being taken (sourced) from the output (I_{OH}) is no greater than 800uA.

2.2 OPERATION WHEN A "1" IS APPLIED TO BOTH INPUTS

(1) With both inputs connected to +5V, the base voltage of Q_1 is less than the emitter voltage, and no base-emitter current can flow in Q_1. However, the collector voltage of Q_1 will be less than the base voltage, and the transistor Q_1 now conducts between base-collector. Thus, the input circuit now looks like the circuit shown in Fig.2.

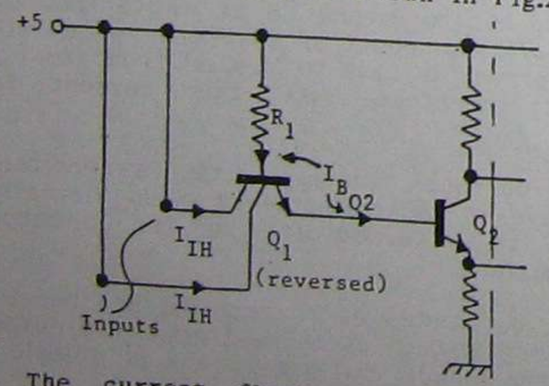


Figure 2

Equivalent circuit when both inputs are connected to +5V

(2) The current flowing from the +5V rail, via R_1 and the base-collector junction of Q_1, will now flow into the base of Q_2, turning Q_2 on.

(3) With Q_2 on, Q_4 will turn off, and Q_3 turn on. Thus, the output is now connected to ground via Q_3.

Note that:

(a) Because Q_1 is now "reversed", leakage current will flow into each input. This current is a '*sink*' current (flows into the IC) and is referred to as I_{IH}. Unlike the low input condition, I_{IH} flows in each input, rather than being distributed among them as for I_{IL}. The guaranteed maximum value of I_{IH} is 40uA per input.

(b) Because all the current for Q_2 is being supplied via R_1, allowing the inputs to 'float' has the same effect as connecting them to +5V. The only difference is that because the inputs are open-circuit, they are now prone to the effects of noise, which may cause random output variations.

AN OPEN CIRCUIT INPUT TO A TTL LOGIC IC WILL BE READ BY THE IC AS A LOGIC 1.

(c) The voltage (V_{IH}) that will be interpreted as a logic 1 for TTL regular is defined as a level of 2 volts to 5 volts.

(d) Because Q_3 is turned on, the output is at logic 0. The guaranteed maximum voltage that will be present at a fully loaded output when it is low is referred to as V_{OL}, and is specified at 0.4V or less.

(e) When low, an output can only 'sink' current to ground via the IC. The maximum allowed value (I_{OL}) is specified at 16mA for TTL regular; higher values may cause V_{OL} to exceed 0.4V.

SUMMARY FOR A LOGIC 1 INPUT, LOGIC 0 OUTPUT

When a TTL regular input is held at a voltage of 2V to 5V, it is at a logic 1, and a current (I_{IH}) of no more than 40uA will flow in (be sunk by) each input. For a NAND gate (or inverter) the output will be a logic 0, and providing the current being sunk by the output (I_{OL}) does not exceed 16mA, the voltage present at the output will not exceed 0.4V.

2.3 SUMMARY OF CURRENT AND VOLTAGE LEVELS FOR TTL LOGIC

(a) V_{IH}: *Minimum voltage level which is guaranteed to be interpreted as a logic "1" at the input* ..... = 2V

(b) V_{IL}: *Maximum voltage level which is guaranteed to be interpreted as a logic "0" at the input* ..... = 0.8V

(c) I_{IH}: *Maximum value of current required by each input that must be supplied by the driving source* ..... = 40uA

(d) I_{IL}: *Maximum value of current that will flow from any one or number of inputs from a single gate* ..... = 1.6mA

(e) V_{OH}: *Minimum output voltage for an output at logic "1", sourcing its maximum rated current* ..... = 2.4V

(f) V_{OL}: *Maximum output voltage for an output at logic "0", sinking its maximum rated value of current ..* = 0.4V

(g) I_{OH}: *The maximum value of current capable of being sourced by an output at logic "1", while ensuring V_{OH} is not less than 2.4 volts .....* = 800uA

(h) I_{OL}: *The maximum value of current capable of being sunk by an output at logic "0" without causing V_{OL} to exceed 0.4 volts.....* = 16 mA

Figs. 3 and 4 show how a TTL output drives a TTL input.

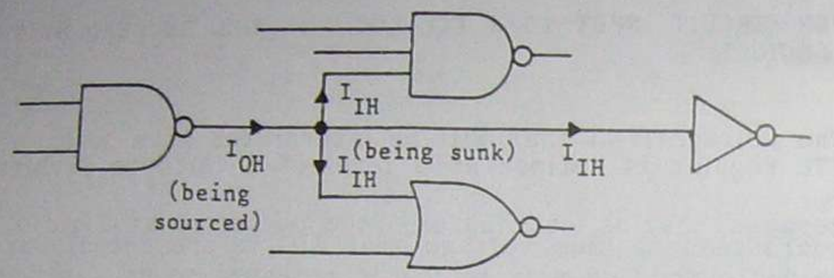


Fig.3: *TTL gate with its output at logic '1' and driving other TTL inputs*

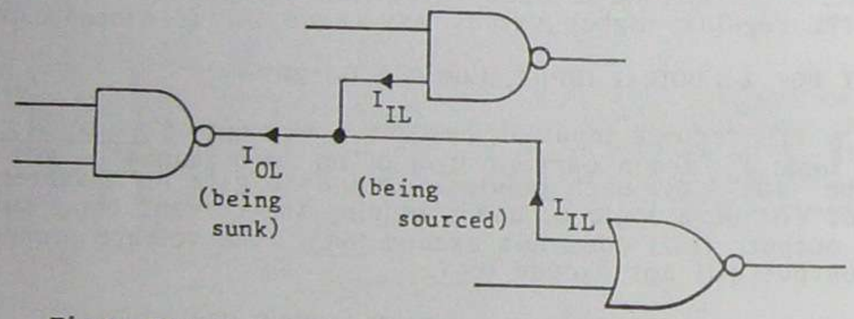


Fig.4: *TTL gate with its output at Logic '0' and driving other TTL inputs*

2.4 TTL LOADING RULES; THE UNIT LOAD

Examination of the current and voltage levels for TTL regular logic shows the following:

(a) An output at logic 0 is capable of sinking sufficient current to drive 10 inputs to logic 0.

(b) An output at logic 1 is capable of supplying (sourcing) sufficient current to drive 20 inputs to logic 1.

Because TTL regular technology is the basis for all other TTL types, the currents required by a TTL regular input are referred to as a UNIT LOAD.

A unit load is defined as an input that sources 1.6mA when low, and sinks 40uA when high.

A TTL regular output is said to have a FAN OUT, or loading factor of 10 unit loads when high, and 20 unit loads when low. Generally, only 10 individual gates would be connected to an output, even though 20 can be pulled low.

The unit load concept allows all IC types from the TTL family to be rated in terms of the number, or fraction of, a unit load they represent.

2.5 CHARACTERISTICS OF TTL LOGIC

TTL logic is widely used in lieu of conventional MOS logic for the following reasons:

(a) It can operate at higher speeds; typically 35Mhz.
(b) It has a relatively low output impedance, and can drive an external load.

However TTL regular logic has several problems; many of which are overcome by other derivatives of the TTL family. The main difficulty with TTL regular logic is the power required to operate it. The power supply for TTL regular logic must be carefully designed to ensure a constant 5 volts (within 5%), and must be free of any switching noise. The totem pole outputs of a TTL gate create a momentary short-circuit as they change state, causing spikes on the power supply. These can be eliminated by placing a 0.1uF de-spiking capacitor close to the IC, usually one for every 6 ICs. Other operating characteristics are:

(a) The *noise margin* (difference between the input logic level voltages required and the voltages available at the output) is 400mV for TTL regular. This means that noise on an input must not exceed 400mV if reliable operation is to be guaranteed.

(b) Voltages to a TTL input must always be 0.8V or less, or 2V or more. A voltage between these values may cause oscillation, or unpredictable logic levels at the output.

(c) Input and power supply leads should be as short as possible. A maximum of around 200mm is specified for an input lead length, depending on the frequency of the digital signal. If the power supply is remote from the digital circuit, on-board regulators should be used. Alternatively, heavy power supply wiring with plenty of local decoupling with tantalum capacitors can be used.

****

ADVANCED CERTIFICATE IN APPLIED INDUSTRIAL ELECTRONICS

YEAR 1 DIGITAL PRINCIPLES

THEORY LESSON 8

Written by Peter Phillips, 1990

LOGIC FAMILIES - PART 2

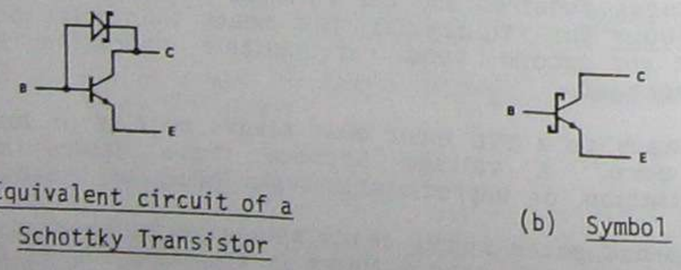
OBJECTIVES: At the end of this lesson you should be able to:

(a) List the input and output currents/voltages that apply to the TTL LS logic family.
(b) Determine the FAN IN and FAN OUT of various TTL sub families.
(c) List applications for open collector and three state logic ICs
(d) Compare the advantages and disadvantages of MOS to TTL logic.

1. TTL LS LOGIC

TTL LS (Low power Schottky) is the most widely supported TTL sub family of the TTL group and was introduced to overcome the various problems associated with TTL regular. As already described in previous notes, TTL regular logic requires considerable power to operate. Attempts to reduce this, as in the L (low power) variety result in a loss in operating speed, as well as output power. Examination of the TTL NAND gate internal circuit (Fig.1, notes for week 7) will show that the totem pole output stage consists of two transistors, driven by the input circuit. The values of the various resistors are chosen to ensure that worst-case transistors are fully saturated when on, by supplying sufficient base current. Often the base current required is excessive anyway, lowering the switch-off time as a result.

TTL LS logic uses Schottky transistors, which are conventional transistors with a Schottky diode connected between base and collector as shown in Fig.1.



(a) Equivalent circuit of a Schottky Transistor

(b) Symbol

Fig.1: *The Schottky Transistor*

A Schottky diode is a high speed switching diode that conducts when the voltage across it is around 0.3 volts. Because it is connected between the collector and emitter of the transistor the diode will conduct when the collector falls to approximately 0.3V, thereby providing a bypass for any base current above

that needed to saturate the transistor. This means the transistor switch-off time is increased, as only the minimum base charge must be removed. The effect is to increase the speed of operation, and TTL S (Schottky only) can operate at over 100MHz. The speed will be lowered if the internal resistor values are increased, and TTL LS incorporates the increased resistor values to bring the speed back to that of TTL regular.

1.1 CHARACTERISTICS OF TTL LS LOGIC

(a) Power consumption is approximately 20% that of TTL regular. A voltage supply of 5V is still required.
(b) Less switching noise is generated due to the reduced currents being switched. Also, less internal heat is generated, resulting in increased reliability.

The following summarises the input/output currents and voltages for the TTL LS sub-family. The corresponding values for TTL regular are shown in brackets for comparison.

1.2 SUMMARY OF CURRENT AND VOLTAGE LEVELS FOR TTL LS LOGIC

(a) V_{IH}: *Minimum voltage level that is guaranteed to be interpreted as a logic 1 at the input:* = **2V** (2V)

(b) V_{IL}: *Maximum voltage level that is guaranteed to be interpreted as a logic 0 at the input:* = **0.8V** (0.8V)

(c) I_{IH}: *Maximum current sunk by each input that must be supplied by the driving source:* = **20uA** (40uA)

(d) I_{IL}: *Maximum value of current that will flow from one or a number of inputs from a single gate:* = **0.4mA** (1.6ma)

(e) V_{OH}: *Minimum output voltage for an output at logic 1, sourcing its maximum rated current:* = **2.7V** (2.4V)

(f) V_{OL}: *Maximum output voltage for an output at logic 0, sinking its maximum rated current:* = **0.5V** (0.4V)

(g) I_{OH}: *The maximum value of current capable of being sourced by an output at logic 1, while ensuring V_{OH} is not less than 2.7 volts:* (2.4V for TTL reg) = **400uA** (800uA)

(h) I_{OL}: *The maximum value of current capable of being sunk by an output at logic 0 without causing V_{OL} to exceed 0.5 volts:* (0.4V for TTL reg) = **8mA** (16mA)

1.3 LOADING AND NOISE MARGIN FOR TTL LS

(a) A TTL LS input is defined as a 0.25 unit load when low, and a 0.5 unit load when driven high.

(b) A TTL LS output can source 400uA (half that of TTL regular) and can sink 8mA, (compared to 16mA for TTL regular). The *FAN OUT* (number of unit loads it can drive) is therefore 10

unit loads for a high, and 5 for a low output. However, because the *FAN IN* (multiple or fraction of a unit load represented by an input) of a TTL LS gate is lower, an LS output can drive 20 LS inputs. This compares favourably to TTL regular which drives (in effect) only 10 regular inputs.

(c) While V_{IL} and V_{IH} are the same as for TTL regular, (0.8V and 2V respectively), V_{OL} and V_{OH} for TTL LS are 0.5V and 2.7V respectively. This gives a noise margin of **300mV when low, and 700mV when high.**

Although TTL LS logic is the current standard, it suffers from several limitations. Paramount is the relatively high output impedance of the gate, which causes switching speeds to drop if loads with a higher than normal capacitance are involved. In some cases TTL LS will not always operate in conditions favourable to TTL regular. However, the low power consumption is its main feature, and TTL LS logic is widely used and readily available.

2. TTL F LOGIC

This type of logic incorporates a more sophisticated internal circuit, and according to the manufacturers (Fairchild), offers speeds up to 100MHz, with only a modest increase in the power requirements. It has a lower output impedance than LS, and can handle capacitive loads more reliably. It is possible this type of logic will become the standard, and already most IC configurations are available in the F technology. Unlike all other TTL groups, the F variety is susceptible to static discharge, due to the small size transistors used. Handling precautions similar to MOS are recommended. An indepth treatment of this sub family is beyond the scope of these notes.

3. SUMMARY OF THE TTL FAMILY

Table 1 compares the characteristics of the TTL family. The speed and power/gate listings are approximations, and provide a comparison only. (Reference - Fairchild data manual)

TABLE 1: *COMPARISON OF THE TTL VARIETIES*

Family TTL.	Speed (MHz)	Power /gate (mW)	Fan Out when HIGH (U.L)	Fan Out when LOW (U.L)	Input HIGH (U.L)	Input LOW (U.L)	Noise Margin (mV)	
							LOW	HIGH
Reg	35	10	20	10	1	1	400	400
H	50	22	12.5	12.5	1.25	1.25	400	400
L	3	1	4	2	1	0.5	500	400
S	125	19	25	12.5	1.25	1.25	300	700
LS	45	2	10	5	0.5	0.25	300	700
F	100	4	25	12.5	0.5	0.375	300	700

Noise margins assume driven input is the same type as the driving output, where;

$$\textit{Noise margin (LOW)} = V_{IL} - V_{OL}$$
$$\textit{Noise margin (HIGH)} = V_{IH} - V_{OH}$$

4. OPEN COLLECTOR AND THREE STATE LOGIC ICS

TTL logic ICs that have an output stage as shown in Fig.1 cannot usually be connected together, as the possibility of one output being at logic 0 and others being at logic 1 can create a short circuit across the power supply. Where it is required to have several outputs connected together, use is made of either open-collector or three state logic ICs.

4.1 OPEN COLLECTOR LOGIC

Open collector logic has an output stage such as that shown in Fig.2(a). This circuit is identical to that for a conventional TTL device, except the top transistor of the totem pole stage is omitted. For this reason, the output can only be pulled by Q_2 to ground, giving a logic 0. When Q_2 is off, the output is an open circuit, requiring a pull-up resistor if a logic 1 is required. This arrangement allows several outputs to be joined together, as there is no direct connection to the supply, preventing the possibility of a short circuit condition from occurring.

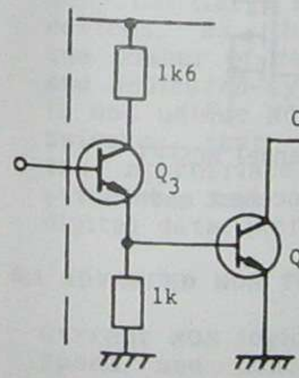


Fig.2(a): *Open collector output stage*

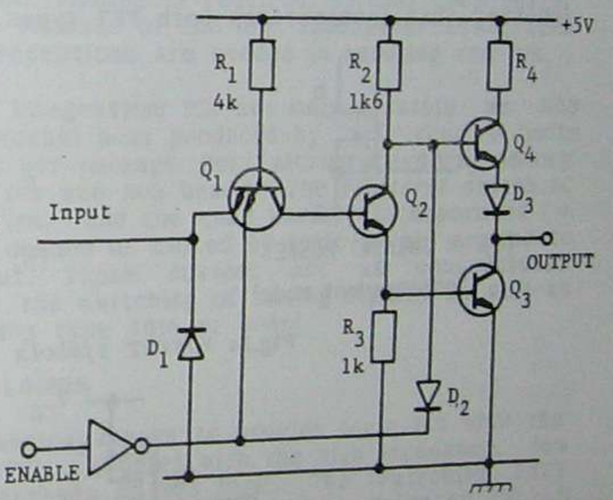


Fig.2(b): *Three state inverter*

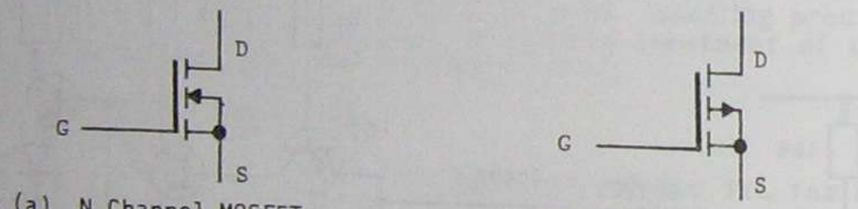
Open collector logic is no longer widely used, as the pull up resistor tends to slow the switching speed. Some open collector IC types feature high-voltage, high-current devices for Q_2, allowing interfacing to a load.

4.2 THREE STATE LOGIC

Three state logic ICs have an extra terminal as shown in Fig.2(b) that is used to turn off both output transistors. In Fig.2(b), if a logic 1 is applied to the ENABLE input, D_2 is turned on, bypassing to ground any bias current to Q_3 and Q_4, causing them to turn off. A logic 0 at the ENABLE input reverse biases D_2, making it an open circuit. This allows the IC to act normally when the outputs are *enabled*, but to become an open circuit, regardless of the previous output logic level, when *disabled*. Thus the output is capable of three states: logic 0, logic 1 and open circuit, unlike the open collector device, which really only has two states. A range of ICs are made with the three state logic feature, and they find application in computers and other data crunching circuits.

5. THE MOS LOGIC FAMILY

The MOS family are those logic ICs with internal circuits based on the enhancement mode MOSFET. This type of FET requires a 'forward' voltage to turn it on; positive with respect to the source voltage for the N channel device, negative (or less than the source terminal) for the P channel type. The MOS family is comprised of the NMOS, PMOS and CMOS types; NMOS having all N channel MOSFETS, PMOS all P channel MOSFETS, and CMOS incorporating both types. CMOS logic is the most common type of MOS logic, but LSI ICs often use either N or P MOS devices only. Fig.3 shows the N and P MOSFET symbols; Fig.4 a CMOS inverter that incorporates both FET types.



(a) N Channel MOSFET (enhancement mode)

(b) P Channel MOSFET (enhancement mode)

Fig.3: *MOSFET symbols*

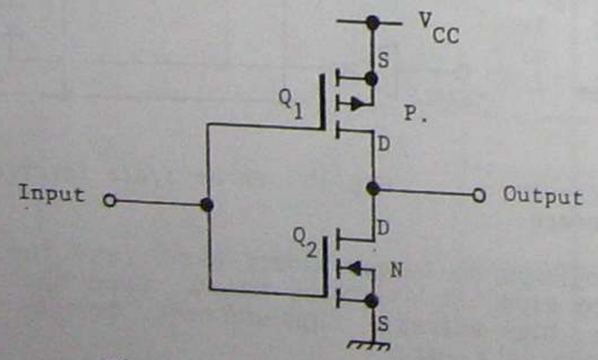


Fig.4: *The Basic CMOS Inverter*

In Fig.4, a logic 0 will cause Q_1 to conduct and Q_2 to turn off. The output voltage will equal Vcc, providing the load across the output is a high impedance (e.g., another CMOS input). If a logic 1 is applied, Q_2 will turn on, and Q_1 turn off. The output voltage will equal 0 volts, again assuming a high impedance load.

Because it is voltage operated (TTL is current driven), CMOS logic consumes little power. Standard MOS logic is relatively slow (4MHz), and cannot provide any output power to a load. It can operate from supply voltages ranging from +3 to +18 and the logic level input and output voltages for a CMOS IC depend on the supply voltage. Fairchild list the following values for a supply voltage of 5V:

V_{IL} = 1V; V_{IH} = 4V
V_{OL} = 0.05V; V_{OH} = 4.95V (for output currents less than 1uA)

In general, the output voltages of a CMOS IC will be virtually 0V and Vcc. Note that a TTL (or TTL LS) output cannot be directly connected to a CMOS input, as the V_{OH} value is not high enough to effectively drive the CMOS input high. If a pull-up resistor (2.2k or so) is connected from the output to Vcc, a TTL LS output can generally be connected to a CMOS input.

A CMOS output cannot normally drive a TTL input, (due to lack of drive current) although some brands of CMOS (Fairchild B series) are listed as being able to drive one TTL LS input. Where interfacing between the families is required, special CMOS buffer ICs should be used. Because of the high impedances associated with MOS logic, ESD precautions are needed in handling the ICs.

Many LSI (large scale integration) ICs are only possible as MOS devices, as the internal heat produced by TTL circuitry limits the number of devices per package. Most microprocessors, memory and dedicated-system ICs are MOS based. The *bilateral* switch IC is one unique MOS device, and the quad variety incorporates 4 switches that are opened or closed by logic levels applied to the appropriate input. These devices act as conventional switches, and allow the switching of analog signals as well as digital data. (Refer CMOS type 4016 or 4066)

5.1 ADVANCED MOS TECHNOLOGIES

Current MOS logic technology seeks to provide logic ICs with the speed and power of TTL, coupled with the high impedance, low power consumption advantages of MOS logic. The Fairchild FACT range of MOS logic can, according to Fairchild, drive 60 TTL LS inputs, with noise margins of 1.25 volts, at an operating power of around 0.1mW per gate. Other types manufactured by Fairchild include the HC range, (High power CMOS) with the speed and power capabilities of TTL LS. Many TTL compatible ICs are currently available in this and other MOS types. For this reason, the subject of MOS logic is clearly a broad one, and you are advised to consult manufacturers data for further information.

****

IC, usually one for every 6 ICs. Other operating characteristics are:

(a) The *noise margin* (difference between the input logic level voltages required and the voltages available at the output) is 400mV for TTL regular. This means that noise on an input must not exceed 400mV if reliable operation is to be guaranteed.

(b) Voltages to a TTL input must always be 0.8V or less, or 2V or more. A voltage between these values may cause oscillation, or unpredictable logic levels at the output.

(c) Input and power supply leads should be as short as possible. A maximum of around 200mm is specified for an input lead length, depending on the frequency of the digital signal. If the power supply is remote from the digital circuit, on-board regulators should be used. Alternatively, heavy power supply wiring with plenty of local decoupling with tantalum capacitors can be used.

****

Exercise

Q1 Explain operation of two inputs TTL regular Nand gate for Input 0 & 1

Q2 Explain characteristics of TTL Logic & Special requirements.

and dedicated-system ICs are MOS based. The *bilateral* switch is one unique MOS device, and the quad variety incorporates switches that are opened or closed by logic levels applied the appropriate input. These devices act as conventio switches, and allow the switching of analog signals as well digital data. (Refer CMOS type 4016 or 4066)

5.1 ADVANCED MOS TECHNOLOGIES

Current MOS logic technology seeks to provide logic ICs with speed and power of TTL, coupled with the high impedance, power consumption advantages of MOS logic. The Fairchild F range of MOS logic can, according to Fairchild, drive 60 TTL inputs, with noise margins of 1.25 volts, at an operating po of around 0.1mW per gate. Other types manufactured by Fairc include the HC range, (High power CMOS) with the speed and po capabilities of TTL LS. Many TTL compatible ICs are curren available in this and other MOS types. For this reason, subject of MOS logic is clearly a broad one, and you are adv to consult manufacturers data for further information.

****

week 13 Exercise

Q (1) Explain the operation of basic CMOS inverter

(2) what is the advantage of using schottky transistor

DE2 Exercize

ADVANCED CERTIFICATE IN APPLIED INDUSTRIAL ELECTRONICS

YEAR 1 DIGITAL PRINCIPLES

Written by Peter Phillips 1990

TOPIC: THEORY ASSIGNMENT WEEK 14

---

1. Explain, using circuit diagrams, the difference between a 4 bit latch and a 4 bit shift register. P69, 70

2. For the serial-in/parallel-out shift register circuit of Fig.1, complete the waveform diagrams below. Assume all Q outputs are initially at a logic 1.

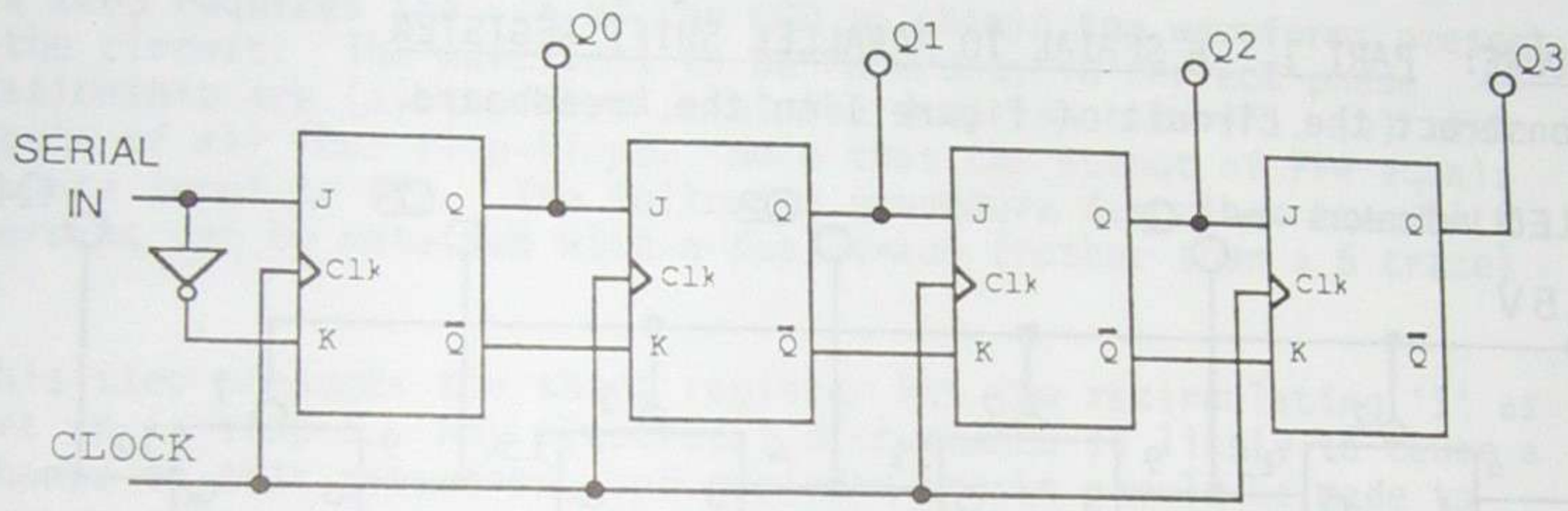
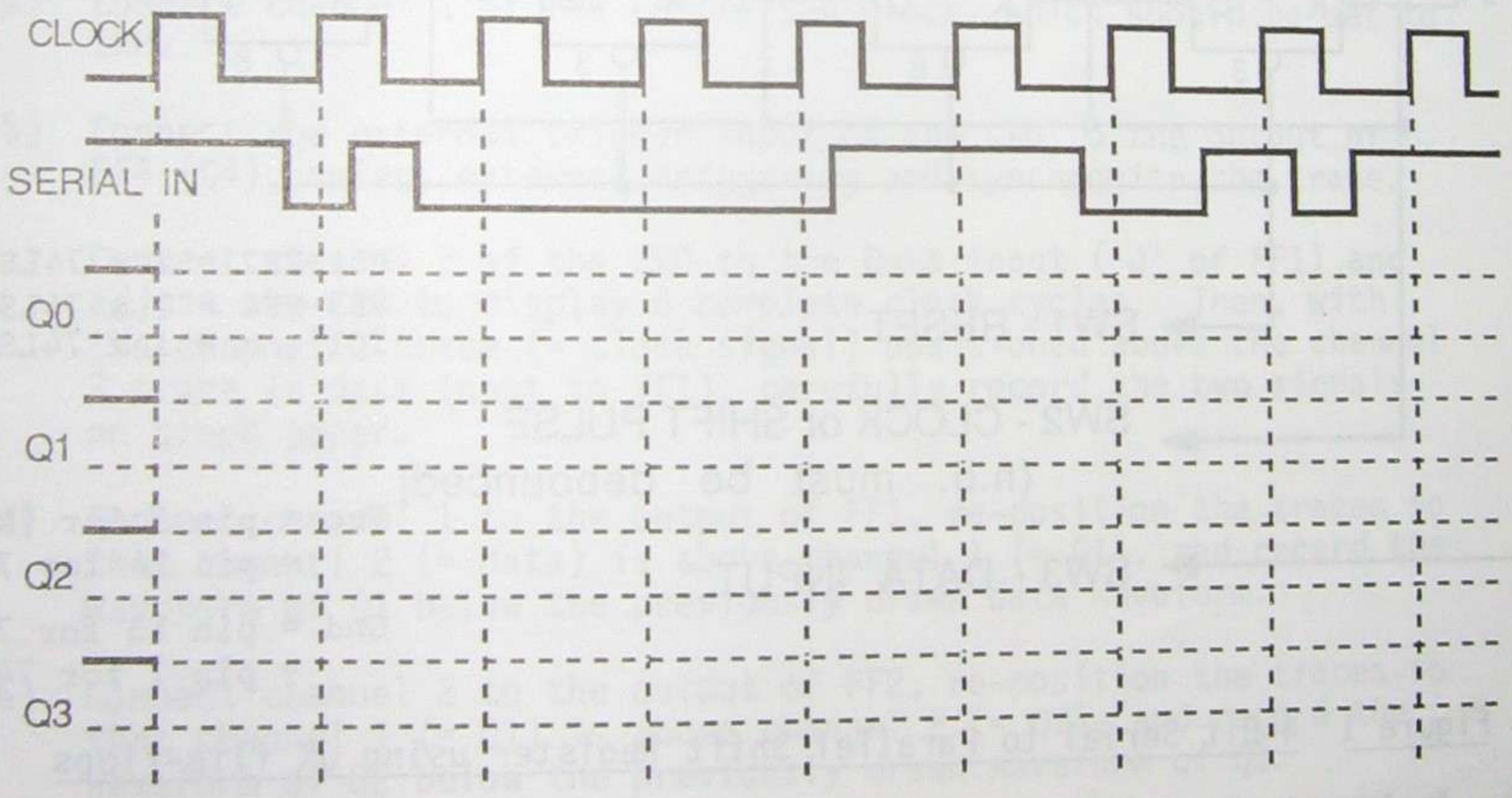


Fig.1



****

DE2 week (7) Exercise

ADVANCED CERTIFICATE IN APPLIED INDUSTRIAL ELECTRONICS

YEAR 1 DIGITAL PRINCIPLES

Written by Peter Phillips 1990

TOPIC: THEORY ASSIGNMENT WEEK 11

---

1. Draw the circuit of an SR flipflop implemented in NOR gates.

P 58

2. If the SET terminal of the circuit in Q1 has a logic 1 applied while the RESET terminal is held at logic 0, determine the state of the Q output.

P 57 Table 1

Output Q = 1

3. Describe the likely effect if the circuit of Q1 has a logic 1 applied to both the SET and RESET terminals.

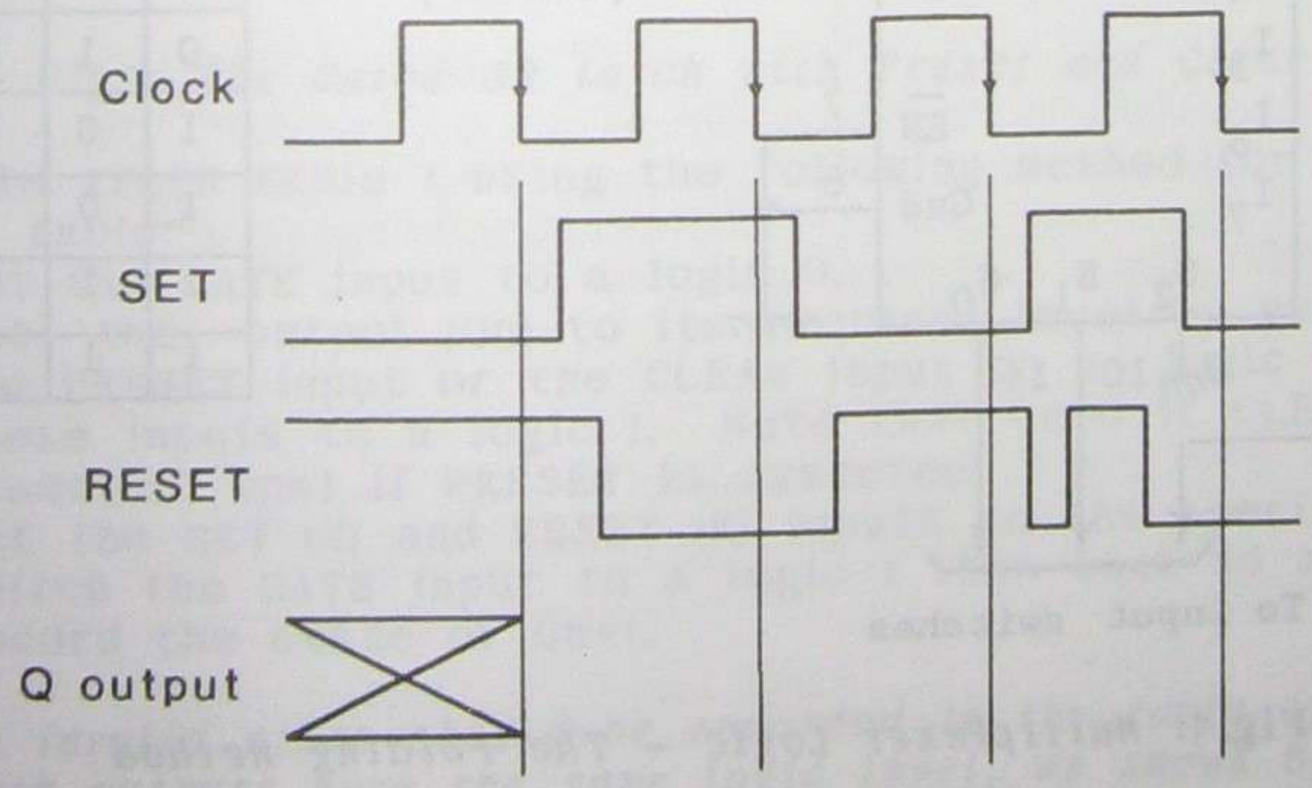
Invalid P 57 - Table 1

4. Draw the circuit of a switch debouncing circuit that uses NAND gates. Include values for any resistors used in the circuit.

P 59 | Fig 4

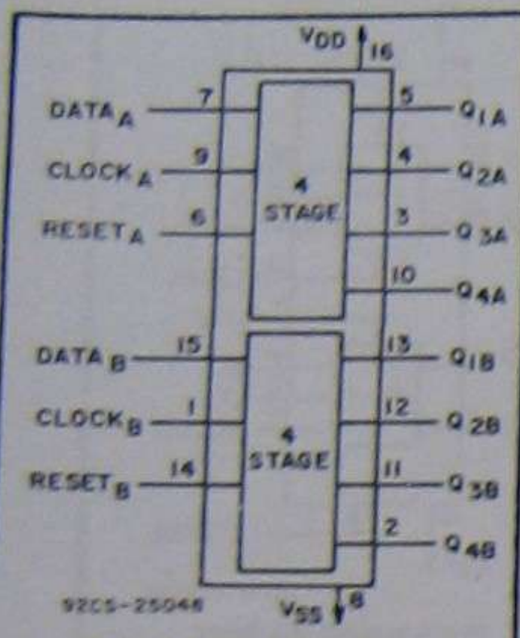
---

5. Complete the Q output waveform (below) for a negative edge triggered clocked SR flipflop. Assume active high inputs.



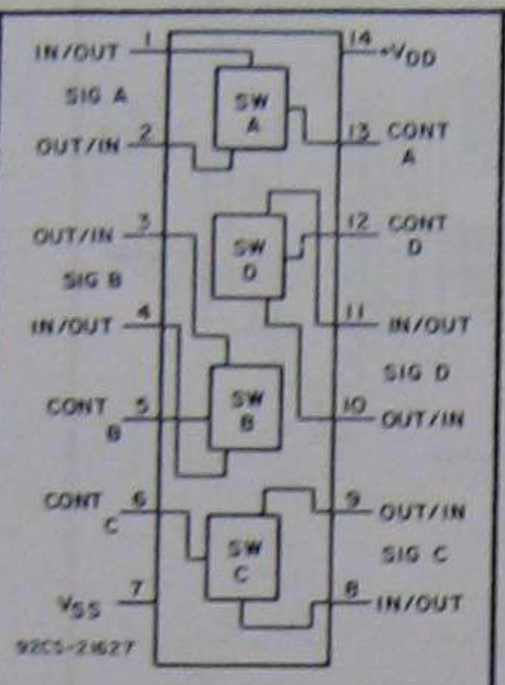
****

Functional Diagrams



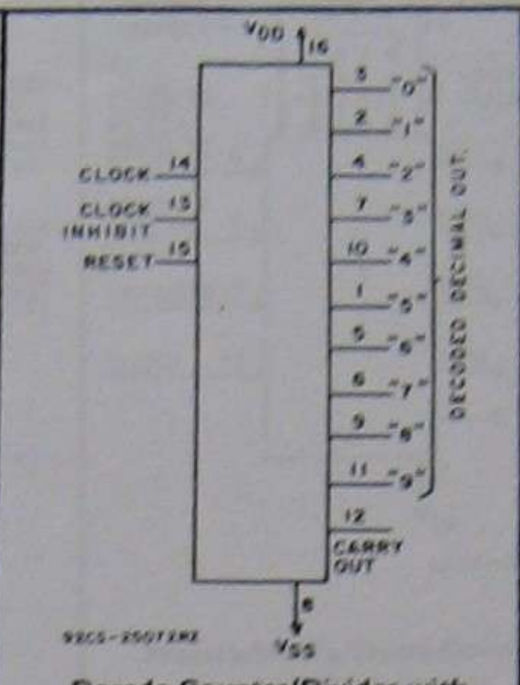
Dual 4-Stage Static Shift Register with Serial Input/ Parallel Output

CD4015A (File No. 943)
CD4015B (Prel.)



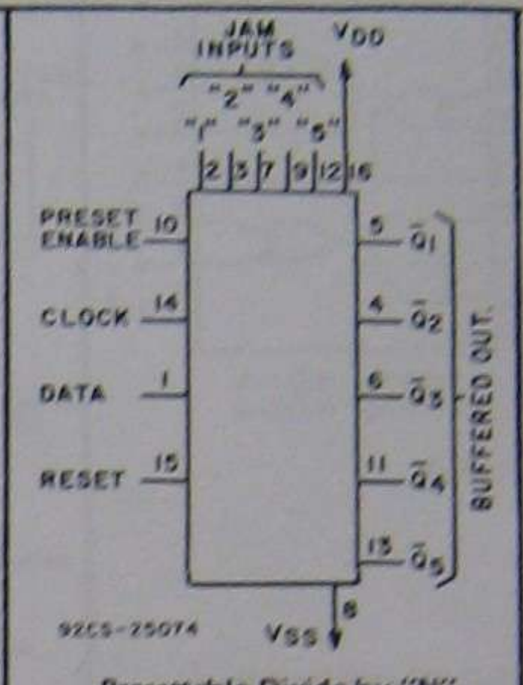
Quad Bilateral Switch

CD4016A (File No. 952)
CD4016B (File No. 953)



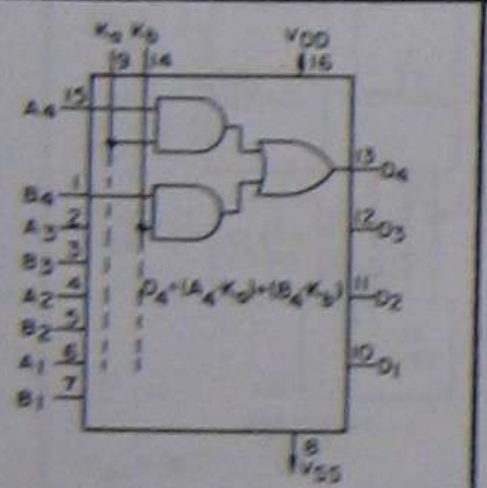
Decade Counter/Divider with 10 Decoded Decimal Outputs

CD4017A (File No. 927)
CD4017B (Prel.)



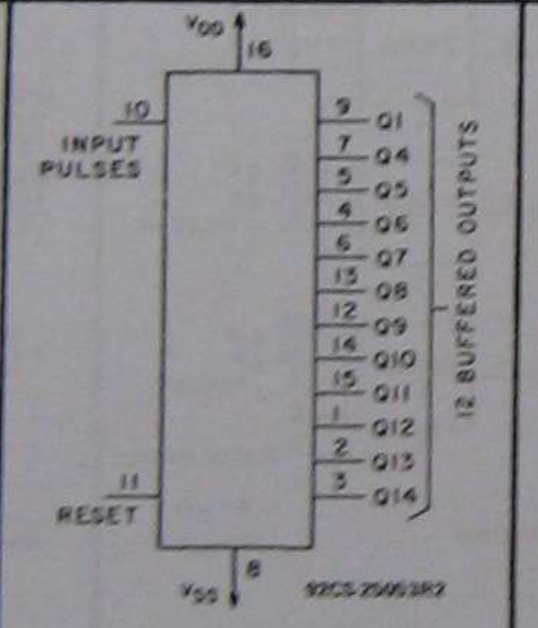
Presettable Divide-by-"N" Counter Fixed or Programmable

CD4018A (File No. 929)
CD4018B (Prel.)



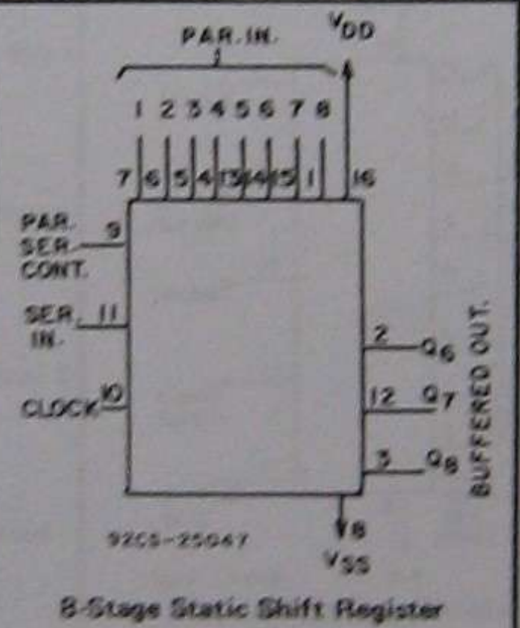
Quad AND/OR Select Gate

CD4019A (File No. 923)
CD4019B (Prel.)



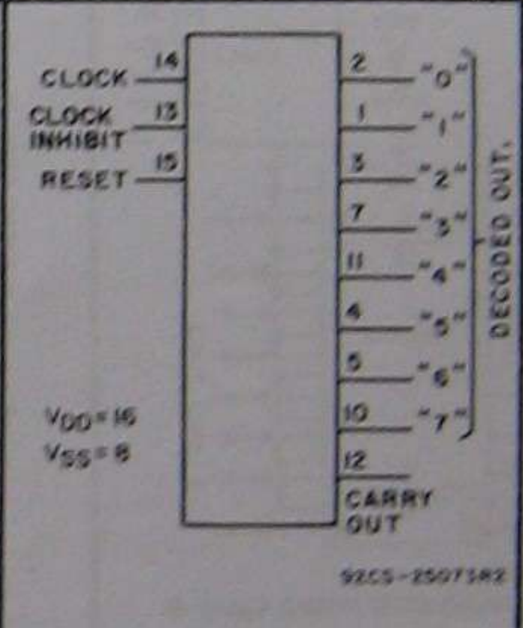
14-Stage Binary Ripple Counter

CD4020A (File No. 928)
CD4020B (Prel.)



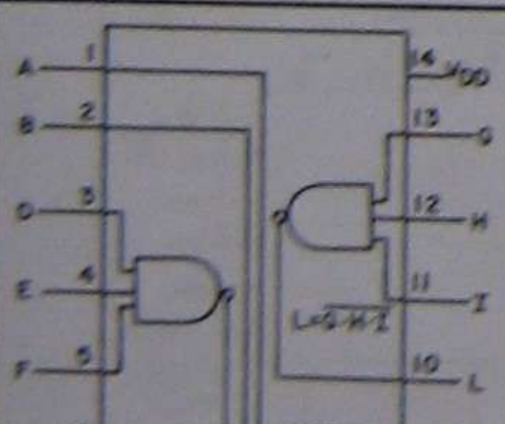
8-Stage Static Shift Register Asynchronous Parallel or Synchronous Serial Input/ Serial Output

CD4021A (File No. 933)
CD4021B (Prel.)



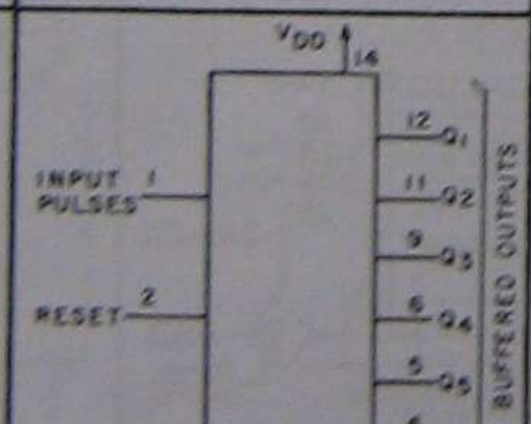
Divide-by-8 Counter/Divider with 8 Decoded Decimal Outputs

CD4022A (File No. 919)
CD4022B (Prel.)



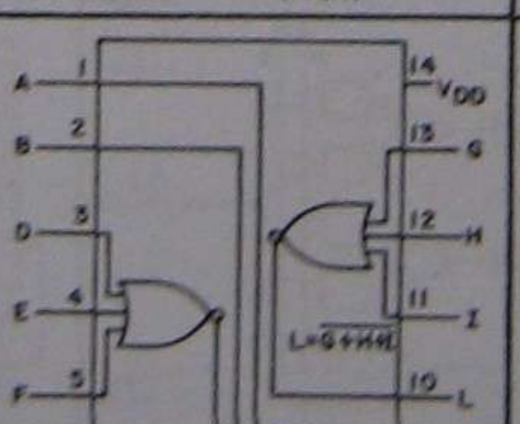
Triple 3-Input NAND Gate

CD4023A (File No. 946)
CD4023B (File No. 986)
CD4023UB (File No. 947)



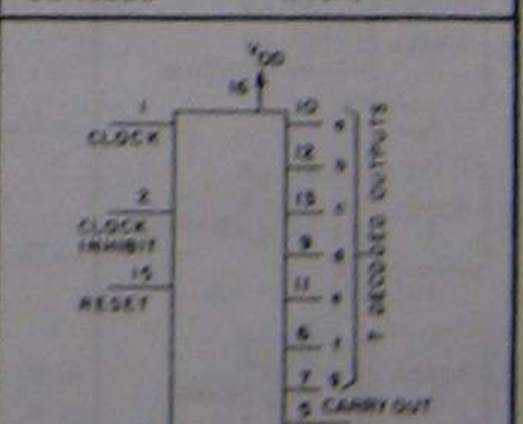
7-Stage Ripple-Carry Binary Counter/Divider

CD4024A (File No. 930)
CD4024B (Prel.)



Triple 3-Input NOR Gate

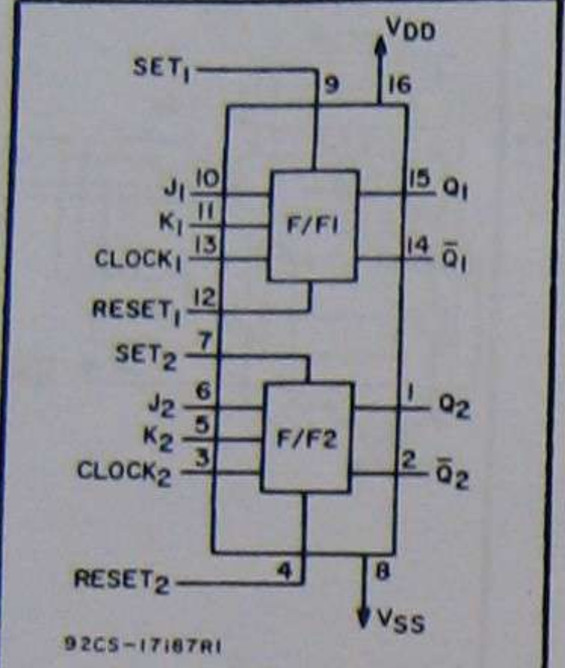
CD4025A (File No. 944)
CD4025B (File No. 985)
CD4025UB (File No. 945)



Decade Counter/Divider with 7-Segment Display Outputs and Display Enable

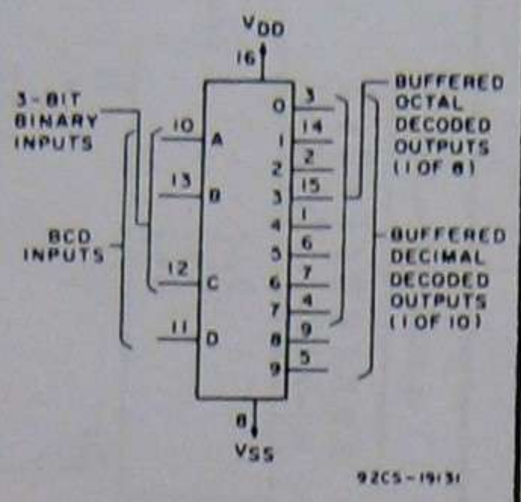
CD4026A (File No. 918)
CD4026B (Prel.)

Functional Diagrams



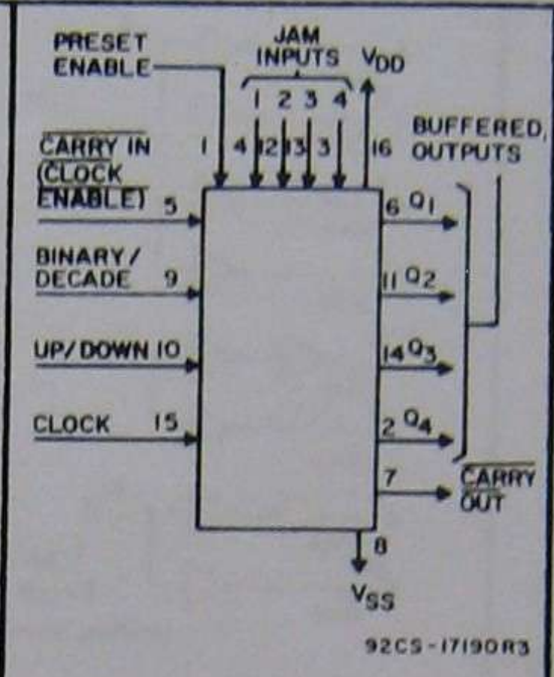
Dual J-K Master-Slave Flip-Flop with Set-Reset Capability

CD4027A (File No. 941)
CD4027B (File No. 942)



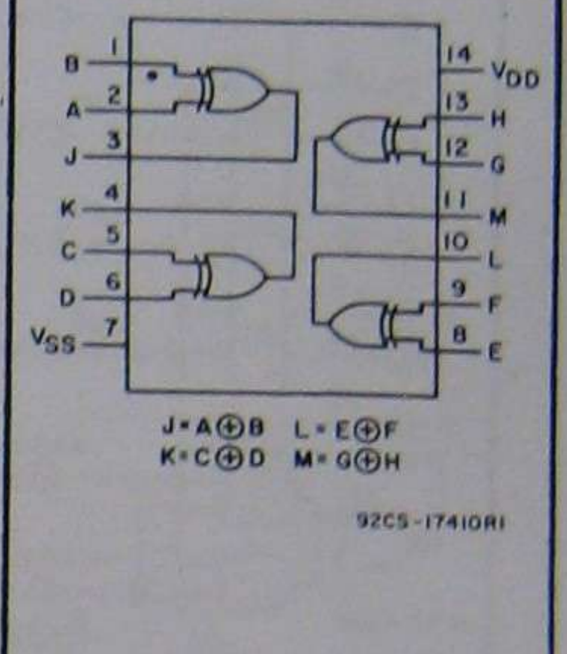
BCD-to-Decimal Decoder

CD4028A (File No. 937)
CD4028B (File No. 1016)



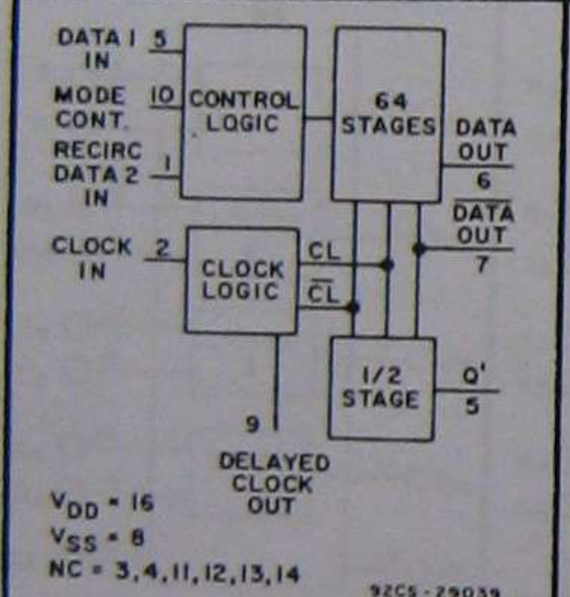
Presettable Up/Down Counter, Binary or BCD-Decade

CD4029A (File No. 931)
CD4029B (Prel.)



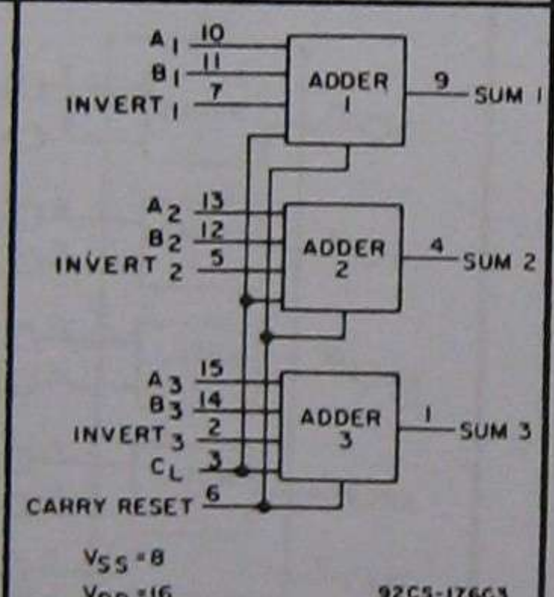
Quad Exclusive-OR Gate

CD4030A (File No. 932)
CD4030B (Prel.)



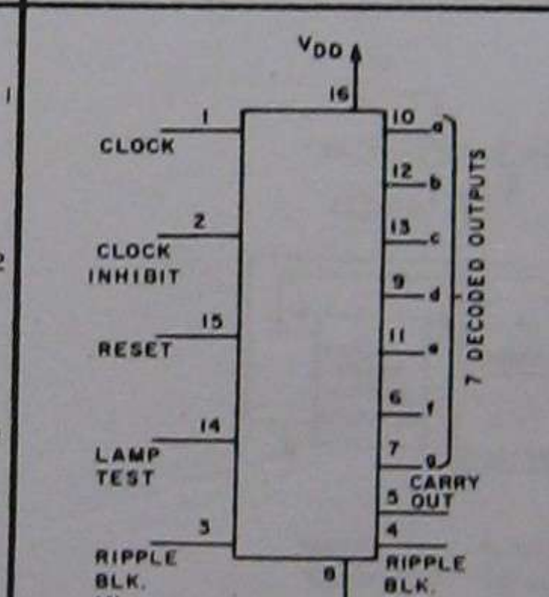
64-Stage Static Shift Register

CD4031A (File No. 569)
CD4031B (Prel.)



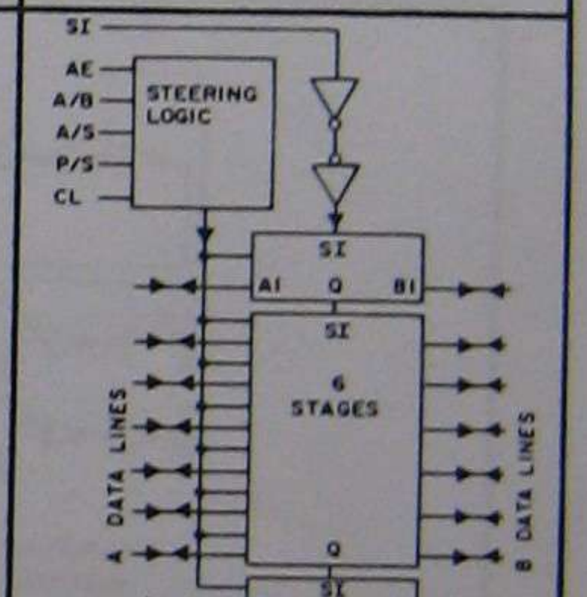
Triple Serial Adder Positive Logic

CD4032A (File No. 915)
CD4032B (Prel.)



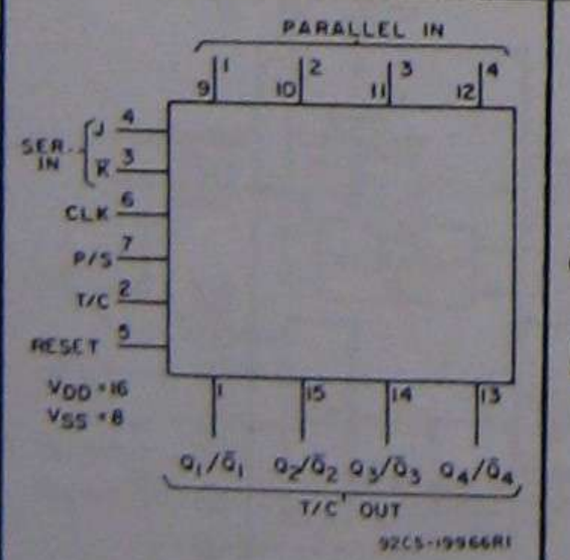
Decade Counter/Divider with 7-Segment Display Outputs and Ripple Blanking

CD4033A (File No. 918)
CD4033B (Prel.)



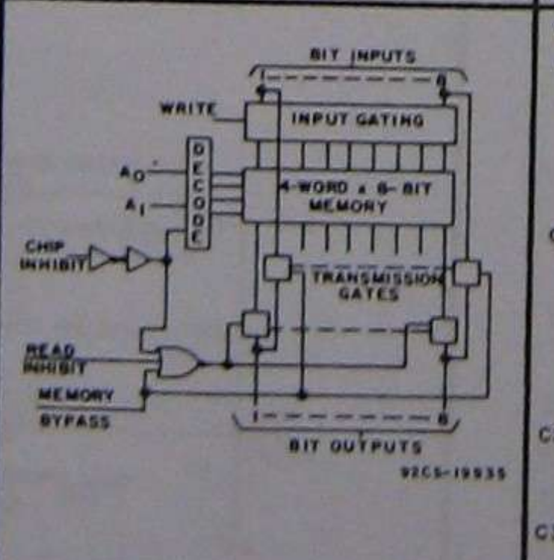
8-Stage Static Bidirectional Parallel/Serial Input/Output Bus Register

CD4034A (File No. 575)
CD4034B (Prel.)



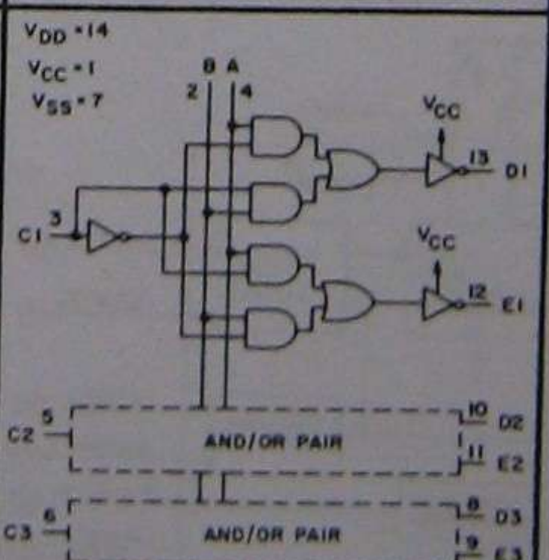
4-Stage Parallel In/Parallel Out Shift Register with J-K Serial Inputs and True/Complement Outputs

CD4035A (File No. 568)
CD4035B (Prel.)



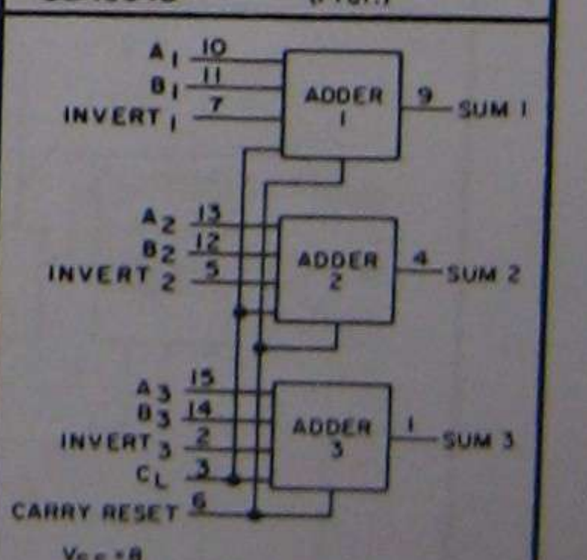
4-Word by 8-Bit Random Access NDRO Memory Binary Addressing

CD4036A (File No. 613)



Triple AND/OR Bi-Phase Pair

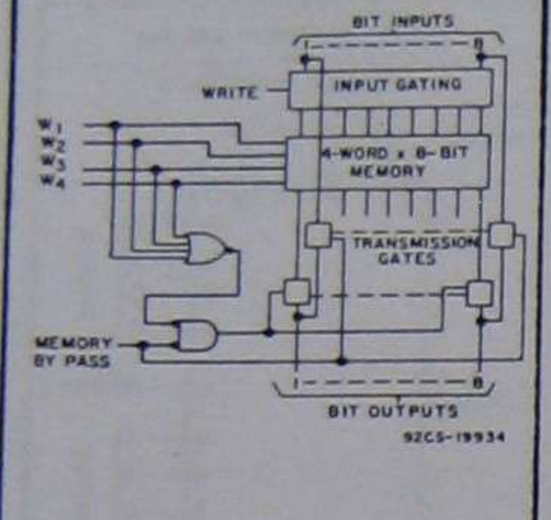
CD4037A (File No. 576)



Triple Serial Adder Negative Logic

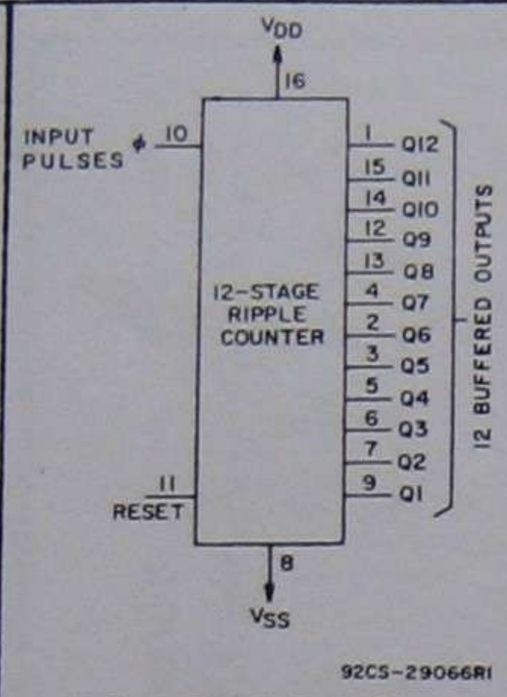
CD4038A (File No. 915)
CD4038B (Prel.)

Functional Diagrams



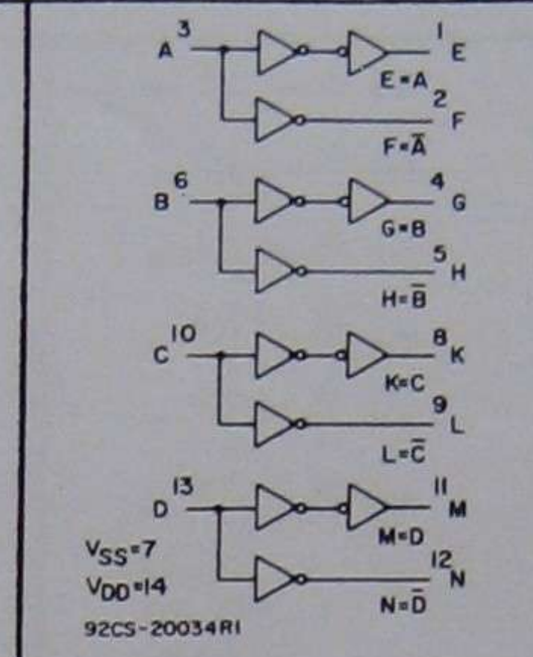
4-Word by 8-Bit Random Access NDRO Memory
Direct Word-Line Addressing

CD4039A (File No. 613)



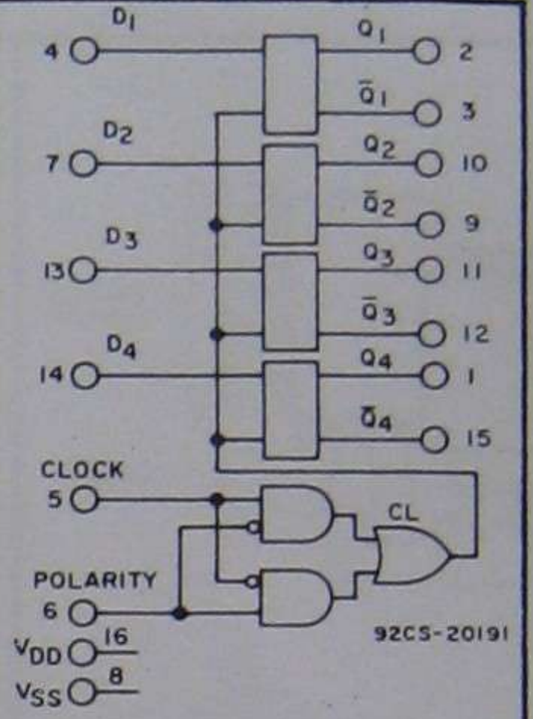
12-Stage Ripple-Carry Binary Counter/Divider

CD4040A (File No. 624)
CD4040B (Prel.)



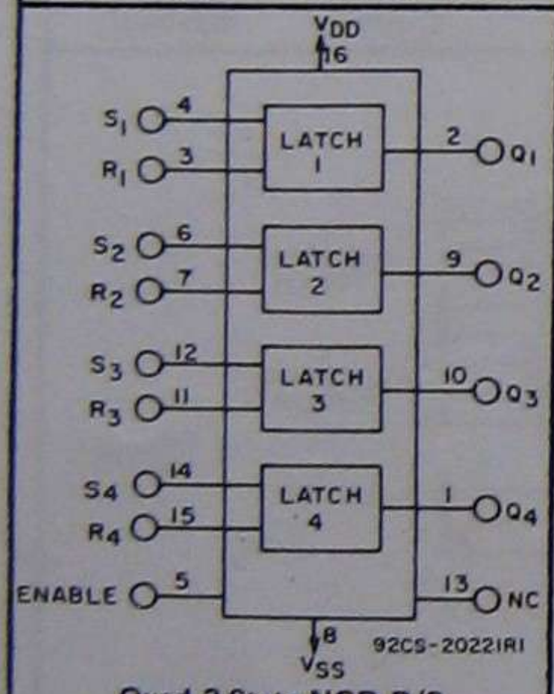
Quad True/Complement Buffer

CD4041A (File No. 572)
CD4041UB (File No. 934)



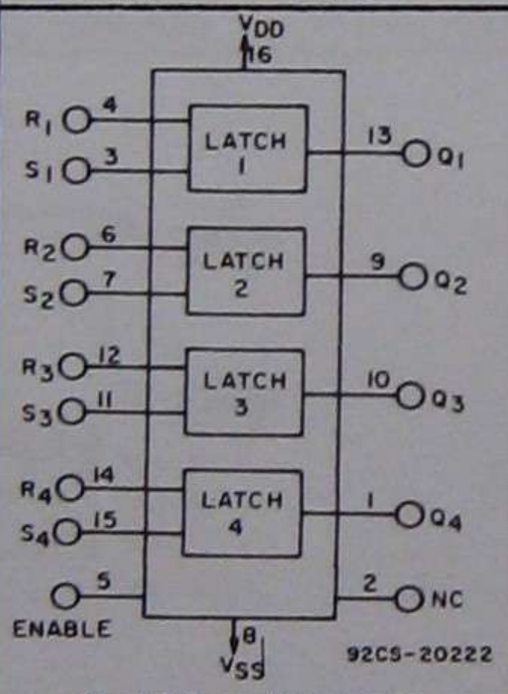
Quad Clocked "D" Latch

CD4042A (File No. 589)
CD4042B (File No. 954)



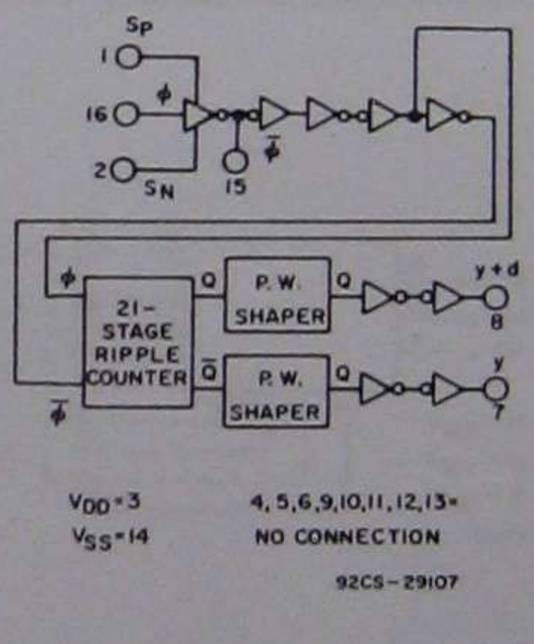
Quad 3-State NOR R/S Latch

CD4043A (File No. 590)
CD4043B (File No. 954)



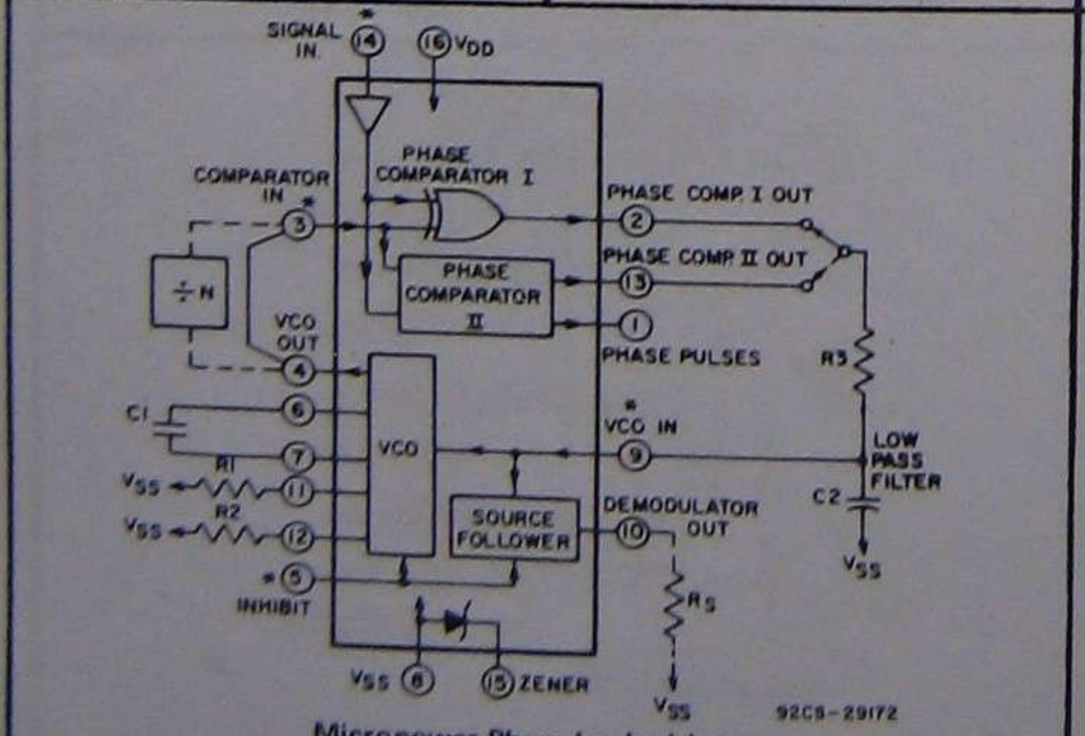
Quad 3-State NAND R/S Latch

CD4044A (File No. 590)
CD4044B (File No. 956)



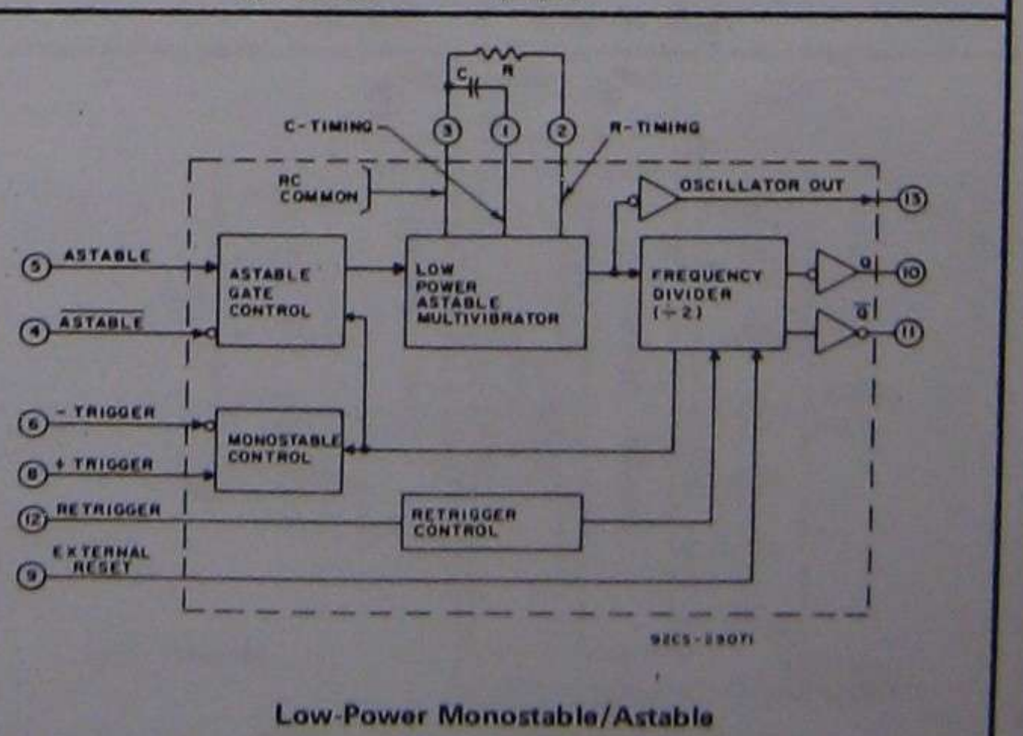
21-Stage Counter

CD4045A (File No. 614)
CD4045B (Prel.)



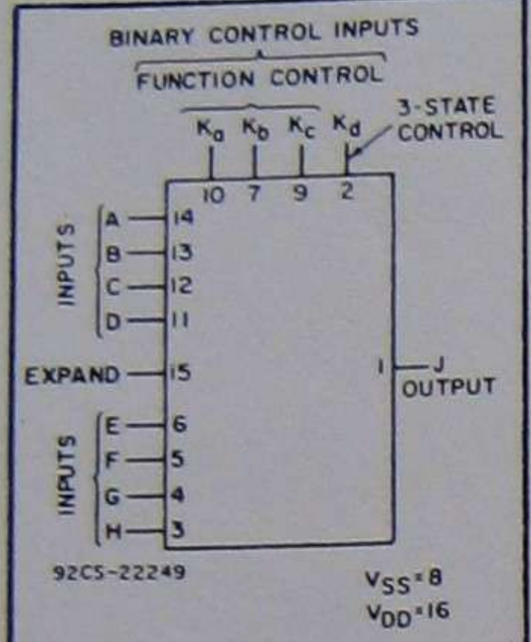
Micropower Phase-Locked Loop

CD4046A (File No. 637)
CD4046B (Prel.)



Low-Power Monostable/Astable Multivibrator

CD4047A (File No. 623)
CD4047B (Prel.)

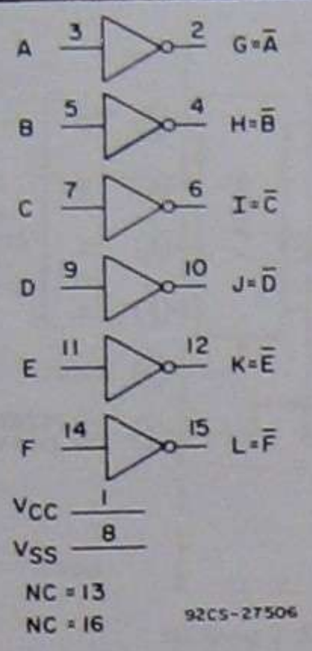


92CS-22249

V_{SS} = 8
V_{DD} = 16

Multi-Function Expandable 8-Input Gate

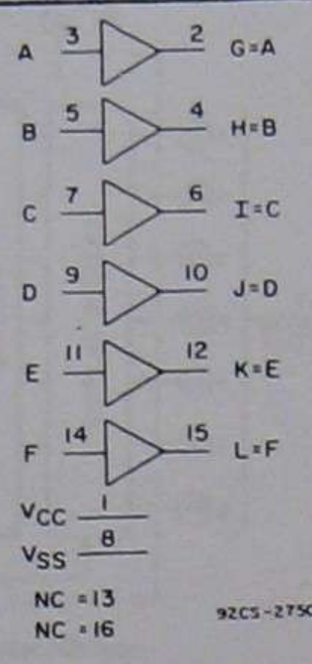
CD4048A (File No. 636)
CD4048B (Prel.)



92CS-27506

Hex Buffer/Converter Inverting Type

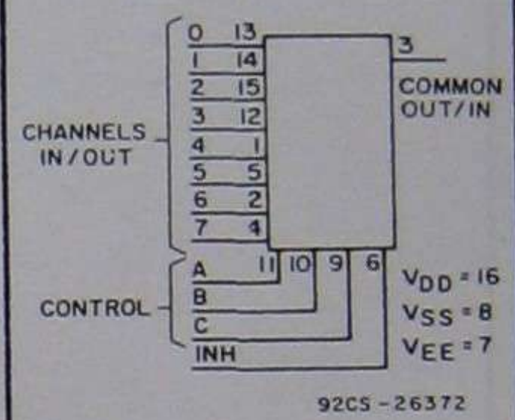
CD4049A (File No. 599)
CD4049UB (File No. 926)



92CS-27507

Hex Buffer/Converter Non-Inverting Type

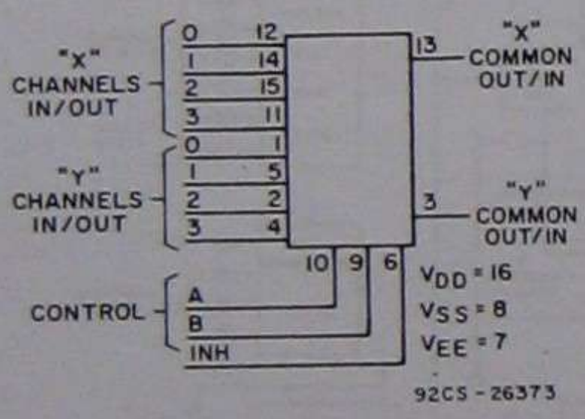
CD4050A (File No. 599)
CD4050B (File No. 926)



92CS-26372

Single 8-Channel Analog Multiplexer/Demultiplexer

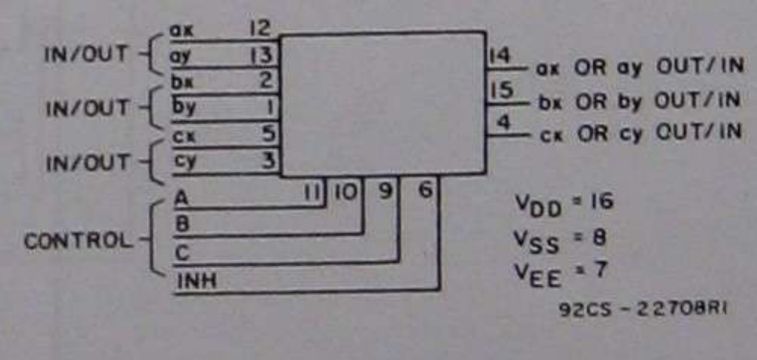
CD4051B (File No. 902)



92CS-26373

Differential 4-Channel Analog Multiplexer/Demultiplexer

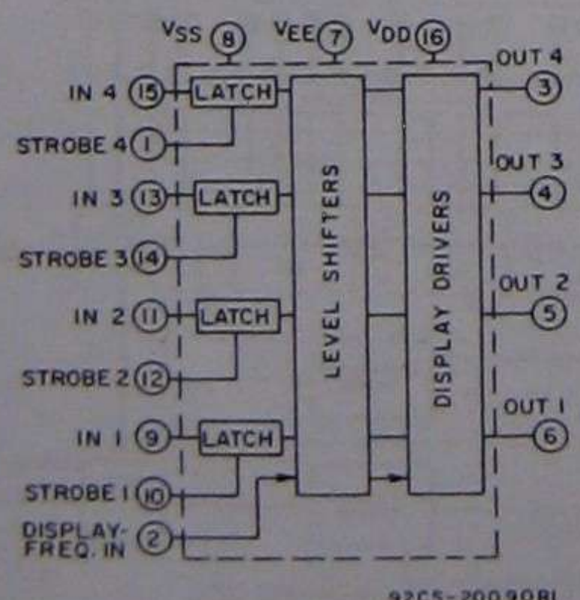
CD4052B (File No. 902)



92CS-22708RI

Triple 2-Channel Multiplexer/Demultiplexer

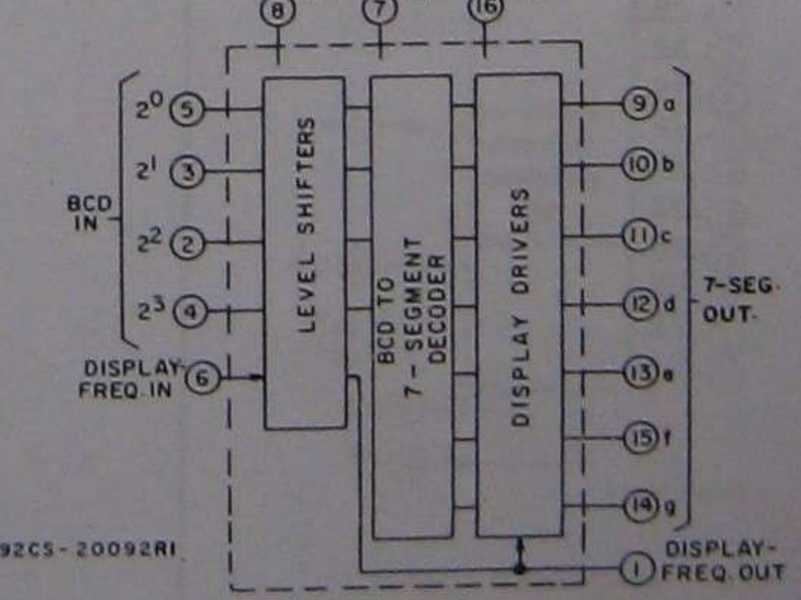
CD4053B (File No. 902)



92CS-20090RI

4-Segment Liquid-Crystal Display Driver

CD4054B (File No. 634)

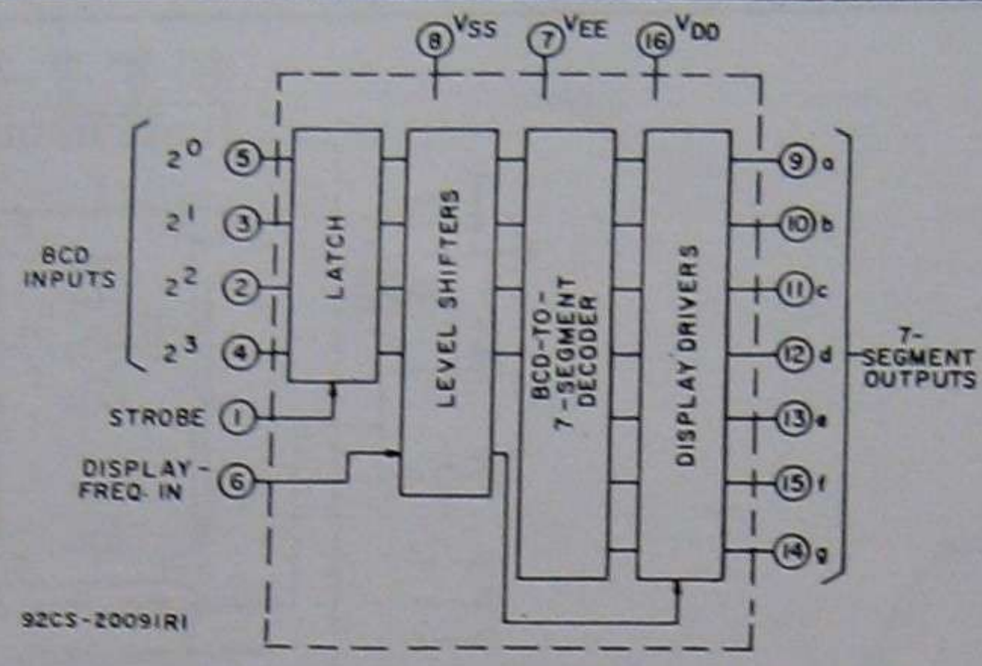


92CS-20092RI

BCD-to-7-Segment Decoder/Driver with "Display-Frequency" Output Liquid-Crystal Display Driver

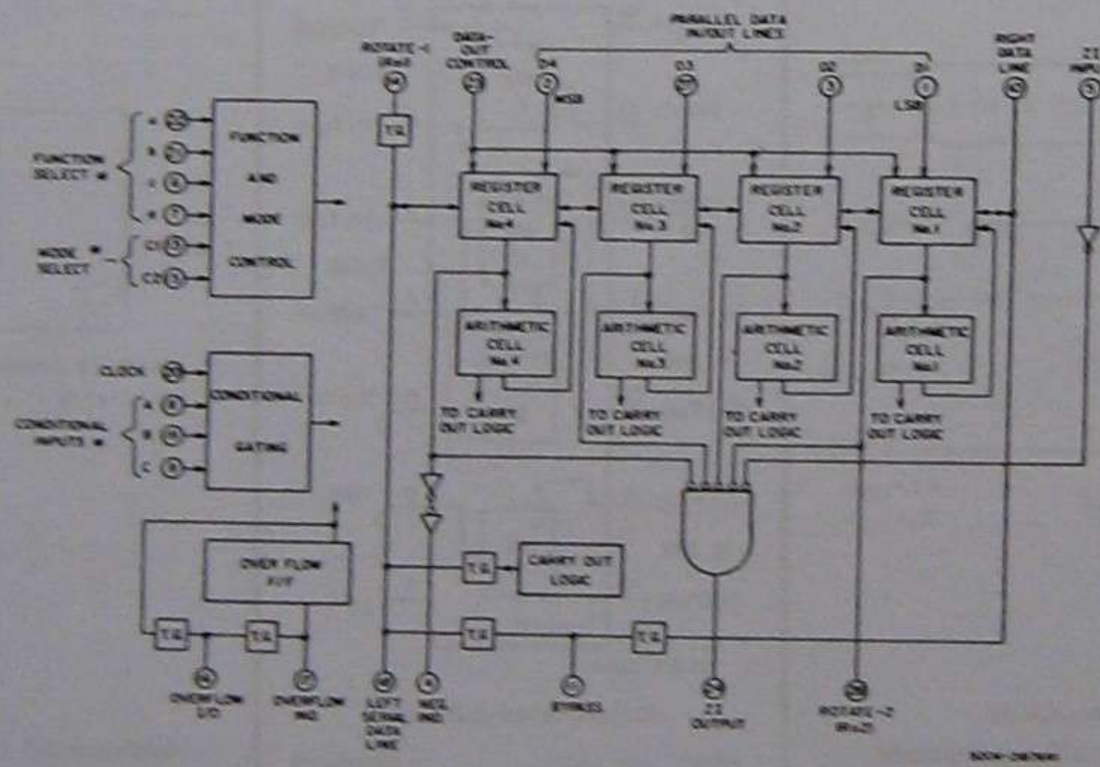
CD4055B (File No. 634)

Functional Diagrams



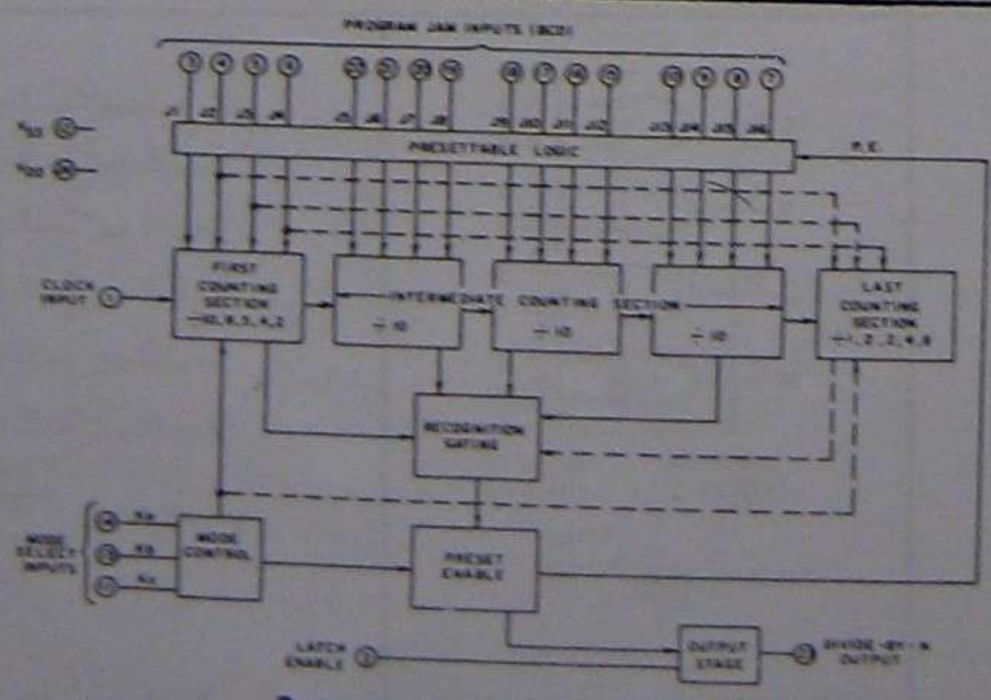
BCD-to-7-Segment Decoder/Driver with Strobed-Latch Function Liquid-Crystal Display Driver

CD4056B (File No. 634)



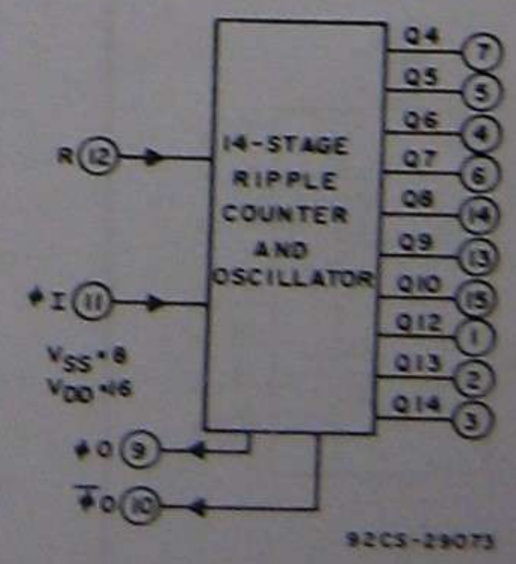
4-Bit Arithmetic Logic Unit

CD4057A (File No. 635)



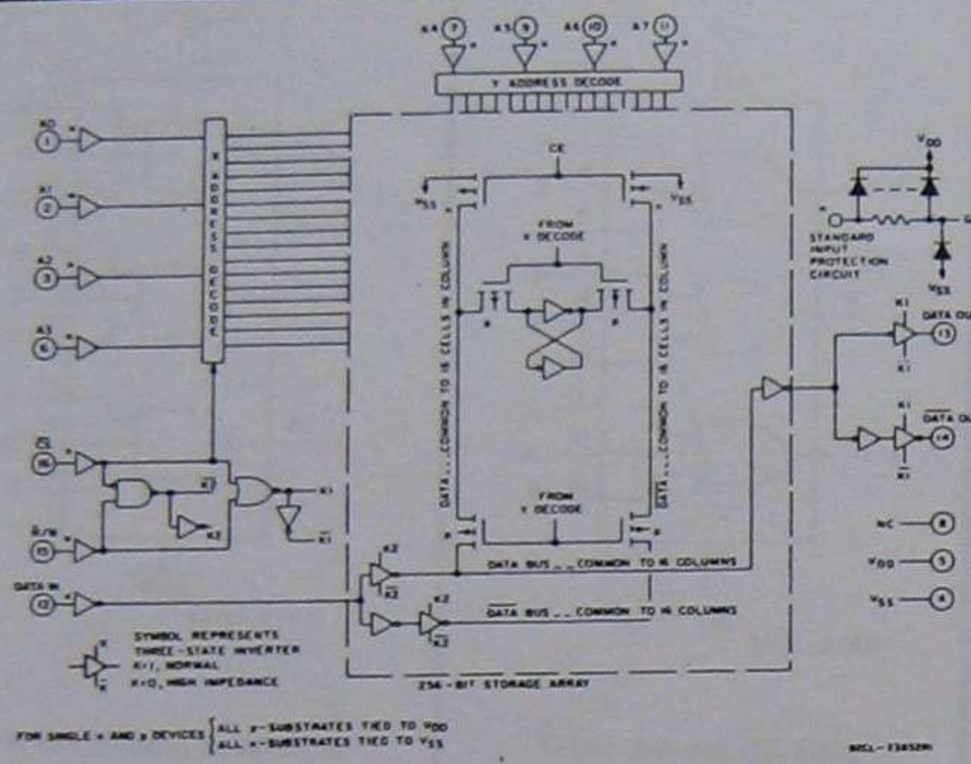
Programmable Divide-by-"N" Counter

CD4059A (File No. 898)



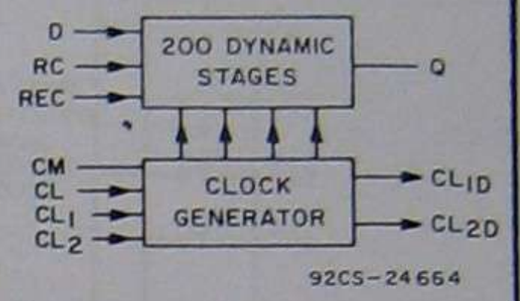
14-Stage Ripple-Carry Binary Counter/Divider and Oscillator

CD4060A (File No. 813)
CD4060B (Prel.)



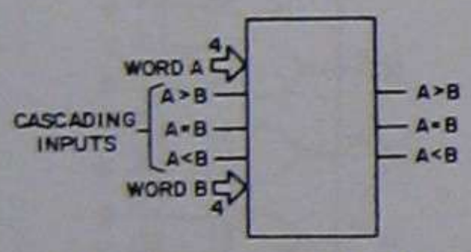
256-Word by 1-Bit Static Random-Access Memory

CD4061A (File No. 768)



200-Stage Dynamic Shift Register

CD4062A (File No. 816)

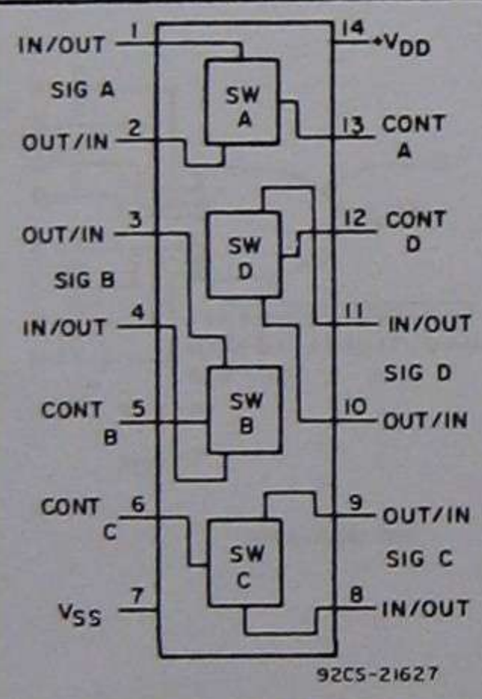


CD4063B FUNCTIONAL DIAGRAM

92CS-24516

4-Bit Magnitude Comparator

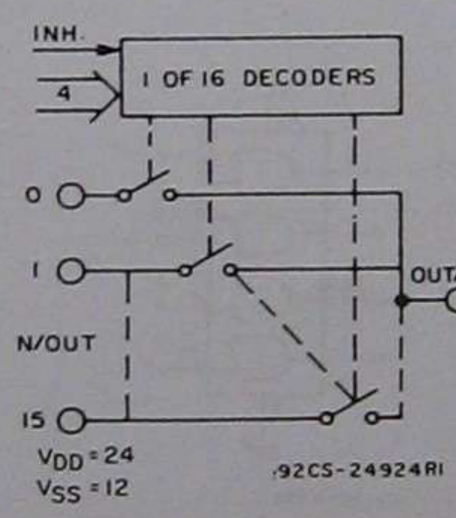
CD4063B (File No. 805)



92CS-21627

Quad Bilateral Switch

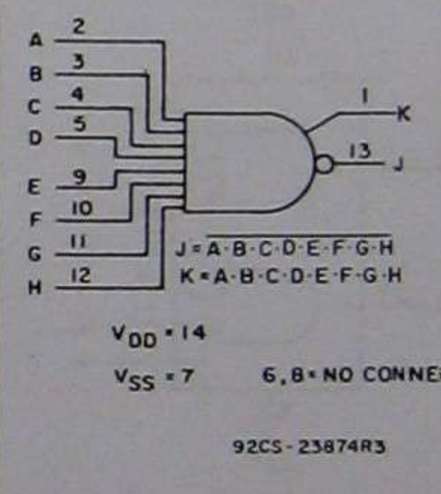
CD4066A (File No. 769)
CD4066B (Prel.)



92CS-24924R1

16-Channel Multiplexer/Demultiplexer

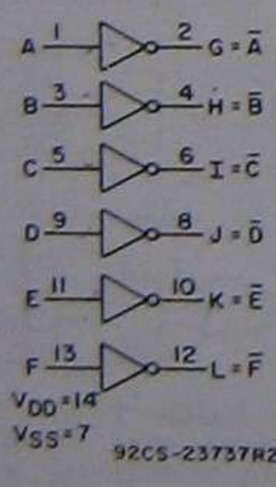
CD4067B (File No. 909)



92CS-23874R3

8-Input NAND/AND Gate

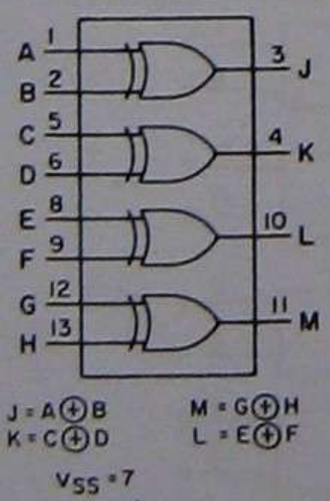
CD4068B (File No. 809)



92CS-23737R2

Hex Inverter

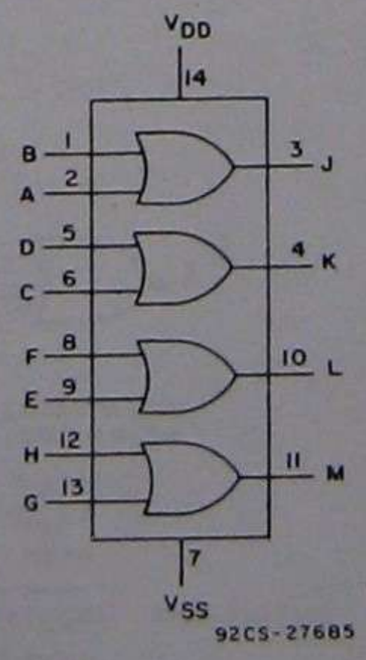
CD4069UB (File No. 804)



92CS-24566R2

Quad Exclusive-OR Gate

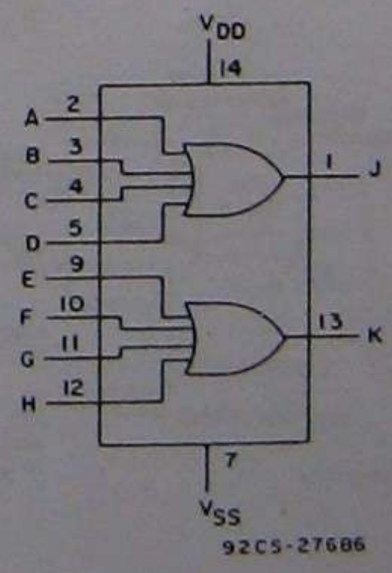
CD4070B (File No. 910)



92CS-27685

Quad 2-Input OR Gate

CD4071B (File No. 807)

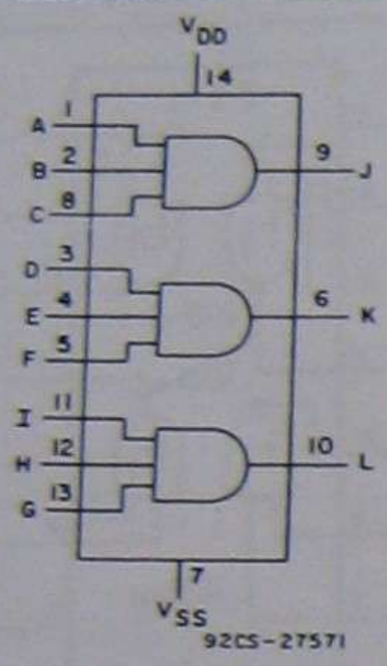


92CS-27686

Dual 4-Input OR Gate

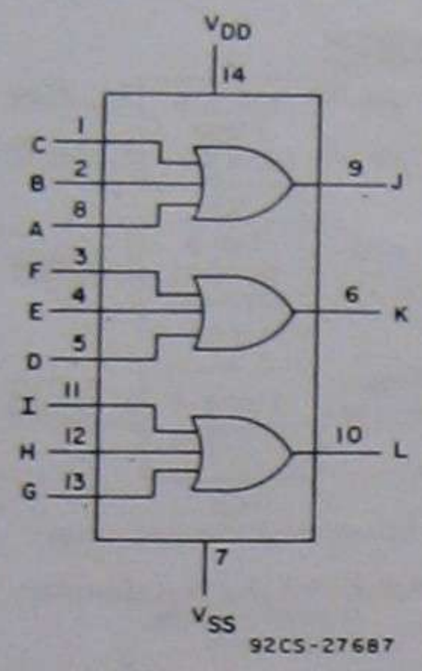
CD4072B (File No. 807)

Functional Diagrams



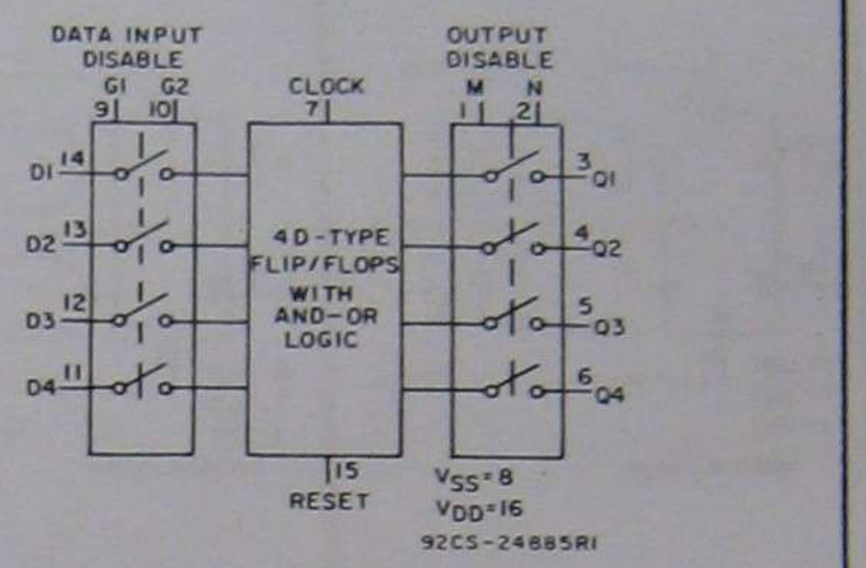
Triple 3-Input AND Gate

CD4073B (File No. 806)



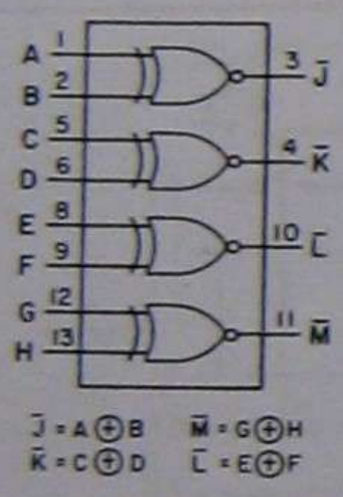
Triple 3-Input OR Gate

CD4075B (File No. 807)



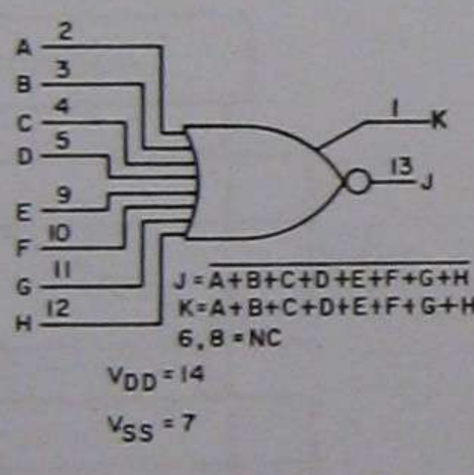
4-Bit D-Type Register

CD4076B (File No. 903)



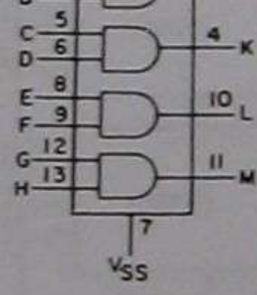
Quad Exclusive-NOR Gate

CD4077B (File No. 910)



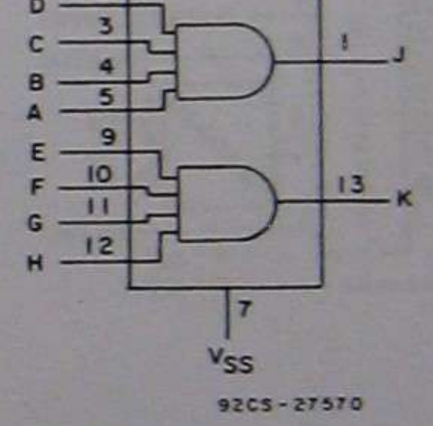
8-Input NOR/OR Gate

CD4078B (File No. 810)



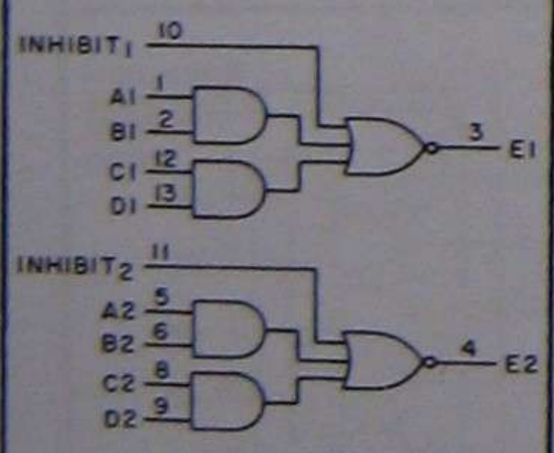
Quad 2-Input AND Gate

CD4081B (File No. 806)



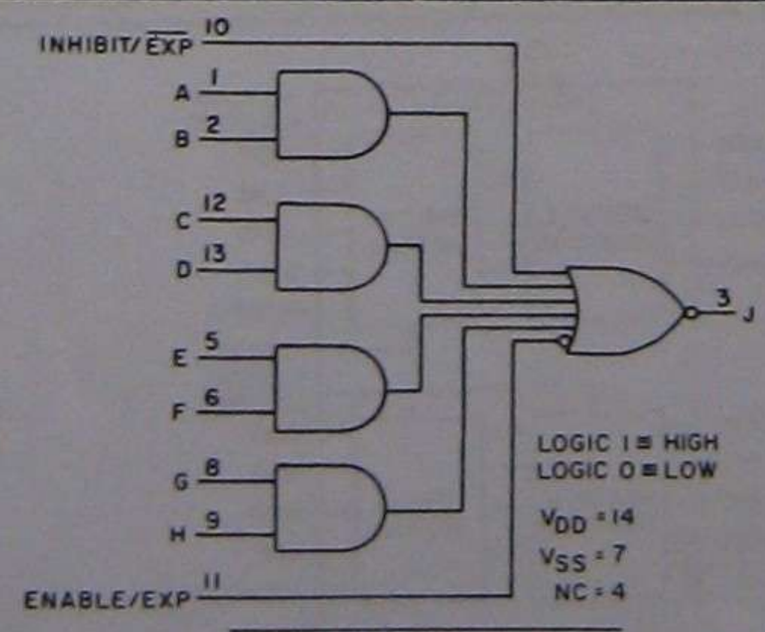
Dual 4-Input AND Gate

CD4082B (File No. 806)



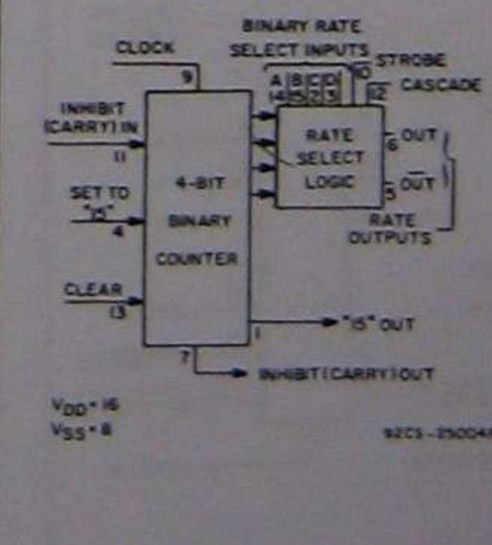
Dual 2-Wide, 2-Input AND-OR-INVERT (AOI) Gate

CD4085B (File No. 811)



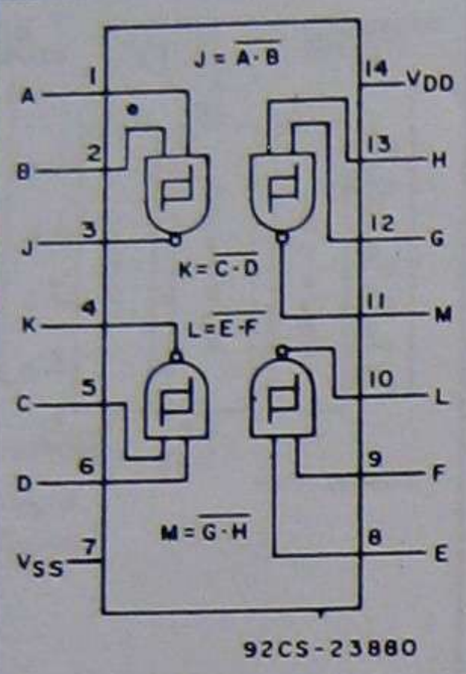
Expandable 4-Wide, 2-Input AND-OR-INVERT (AOI) Gate

CD4086B (File No. 812)



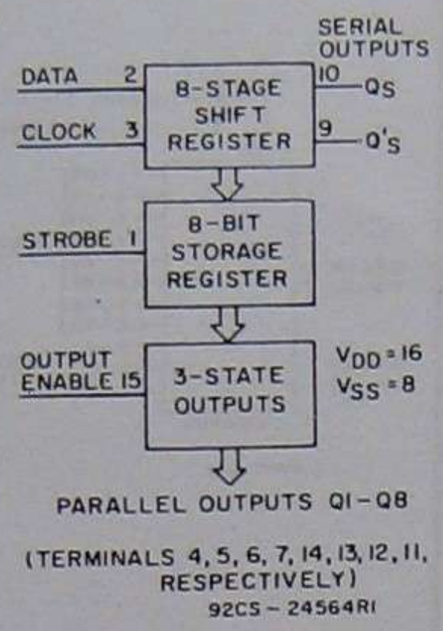
Binary Rate Multiplier

CD4089B (File No. 1003)



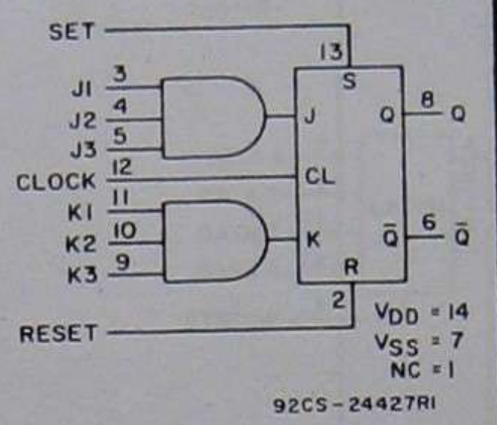
Quad 2-Input NAND Schmitt Trigger

CD4093B (File No. 836)



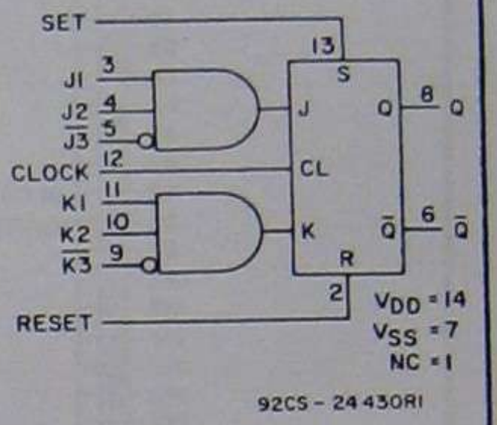
8-Stage Shift-and-Store Bus Register

CD4094B (File No. 869)



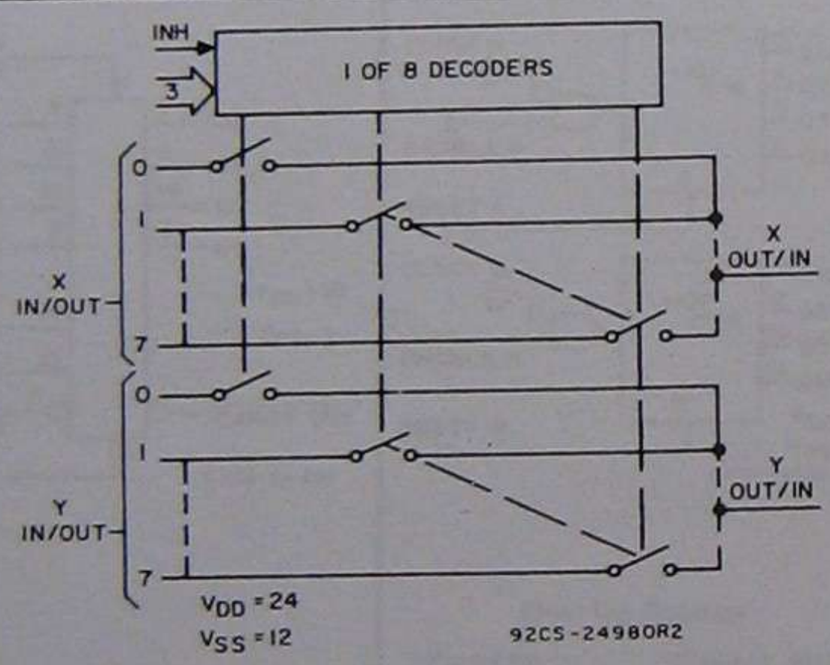
Gated J-K Master-Slave Flip-Flop, Non-Inverting Inputs

CD4095B (File No. 879)



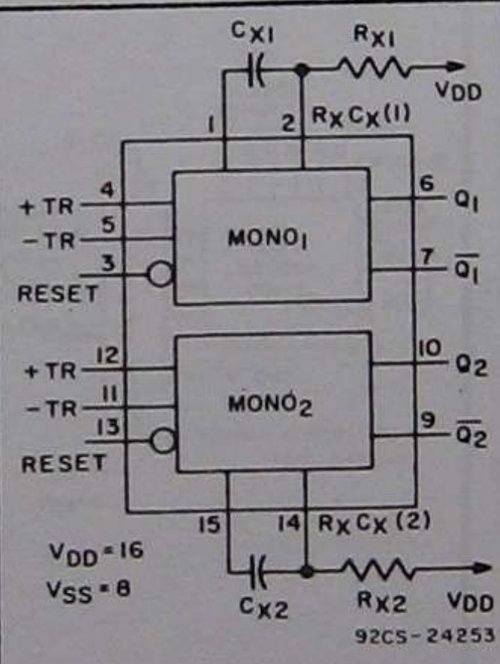
Gated J-K Master-Slave Flip-Flop, Inverting and Non-Inverting Inputs

CD4096B (File No. 879)



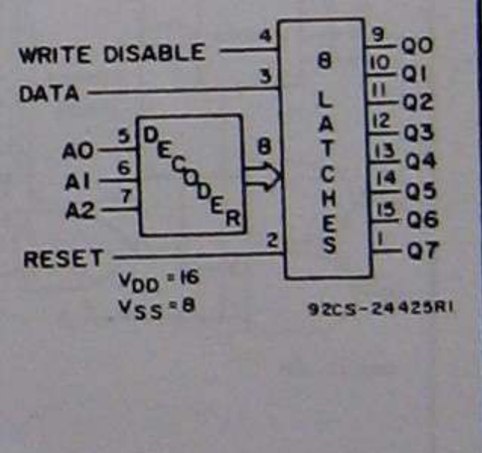
Differential 8-Channel Multiplexer/Demultiplexer

CD4097B (File No. 909)



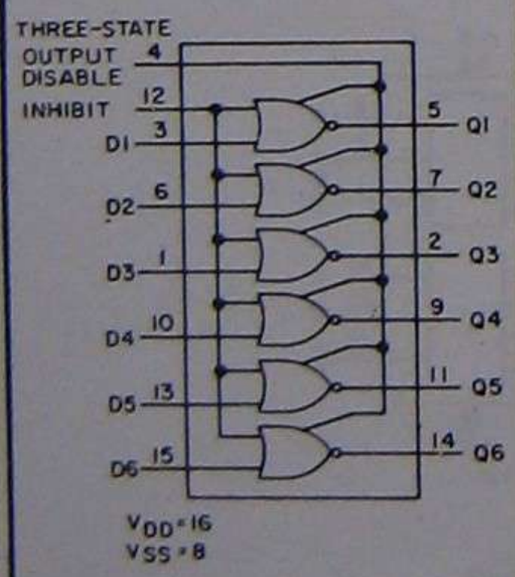
Dual Monostable Multivibrator

CD4098B (File No. 979)



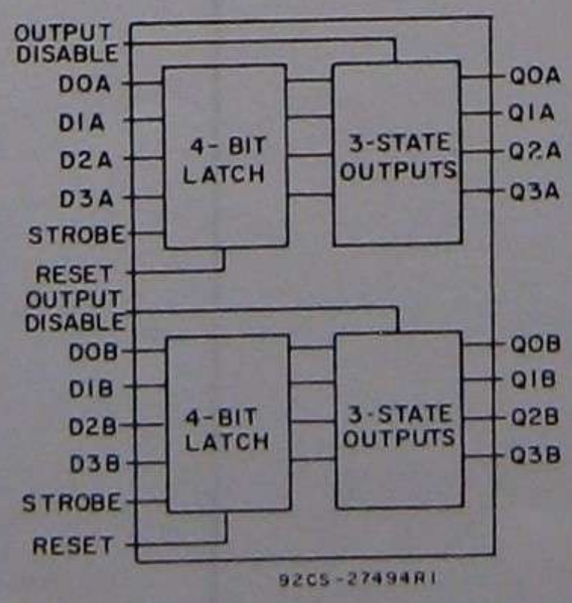
8-Bit Addressable Latch

CD4099B (File No. 948)



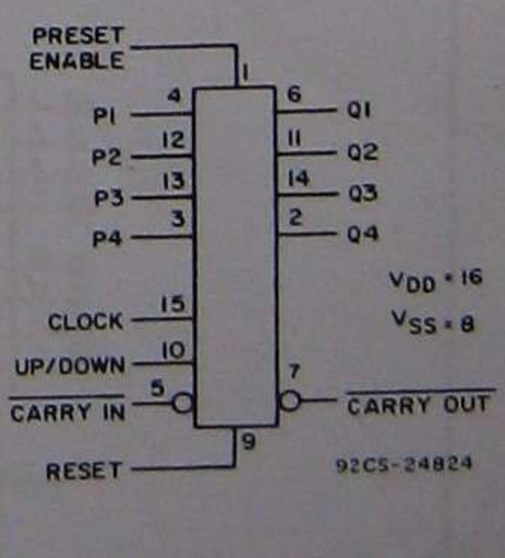
Strobed Hex Inverter/Buffer

CD4502B (File No. 1002)



Dual 4-Bit Latch

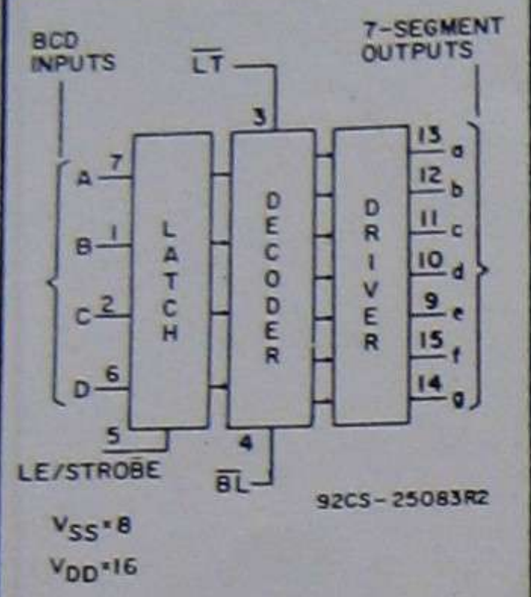
CD4508B (File No. 1009)



BCD Presettable Up/Down Counter

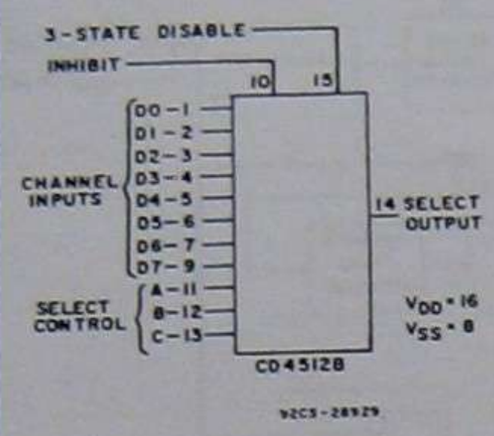
CD4510B (File No. 899)

Functional Diagrams



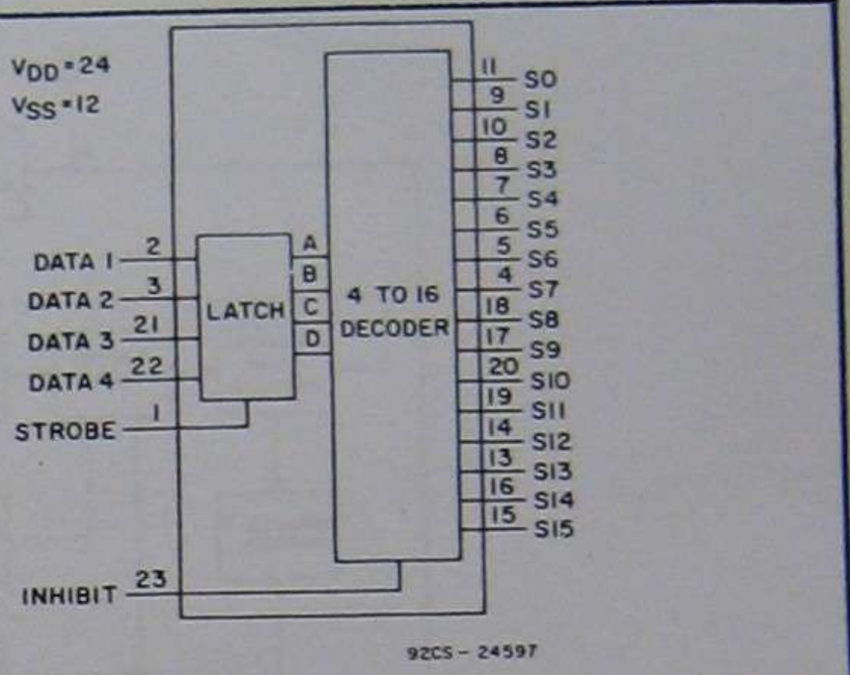
BCD-to-7-Segment Latch Decoder Driver

CD4511 (File No. 901)



8-Channel Data Selector

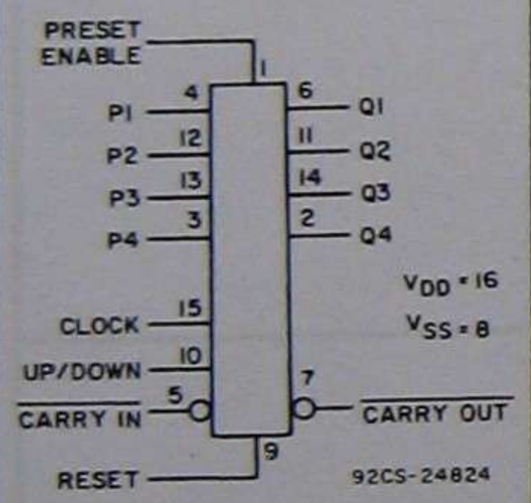
CD4512B (Prel.)



4-Bit Latch/4-to-16 Line Decoder

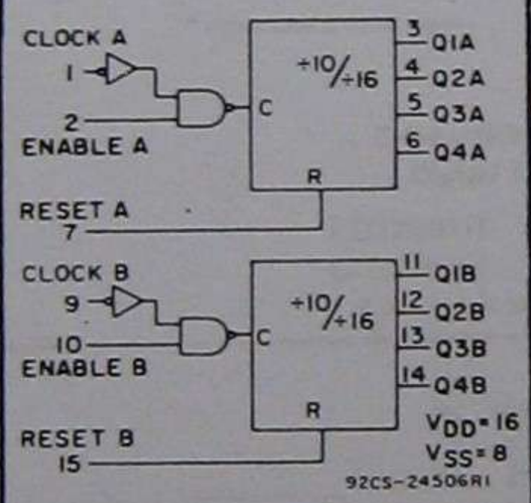
CD4514B (File No. 814) Output "High" on Select

CD4515B (File No. 814) Output "Low" on Select



Binary Presettable Up/Down Counter

CD4516B (File No. 899)



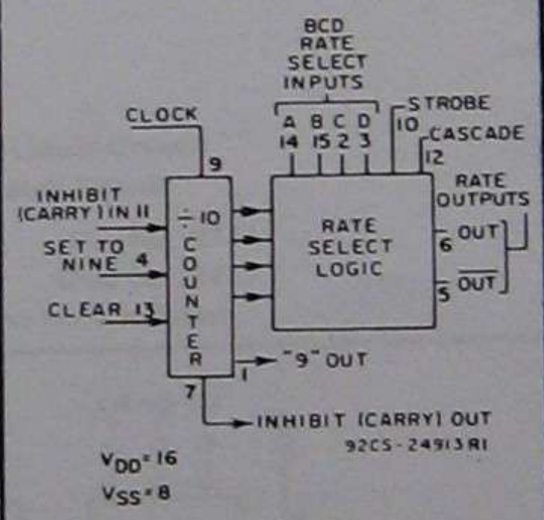
Dual Up Counter

CD4518B (File No. 808)

BCD

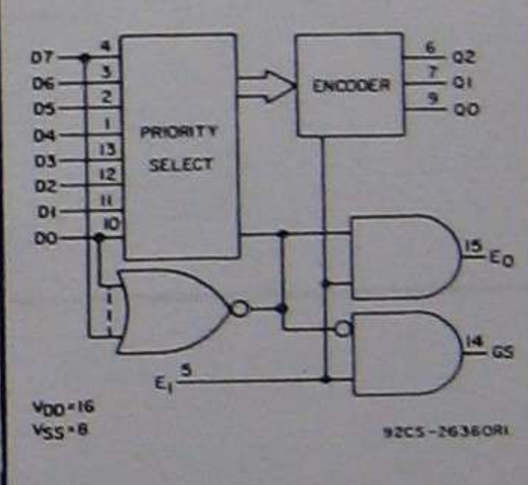
CD4520B (File No. 808)

Binary



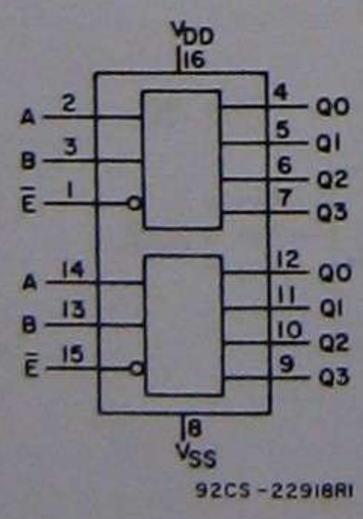
BCD Rate Multiplier

CD4527B (File No. 1006)



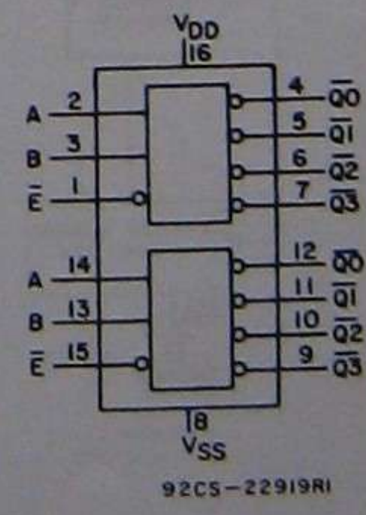
8-Bit Priority Encoder

CD4532B (File No. 876)



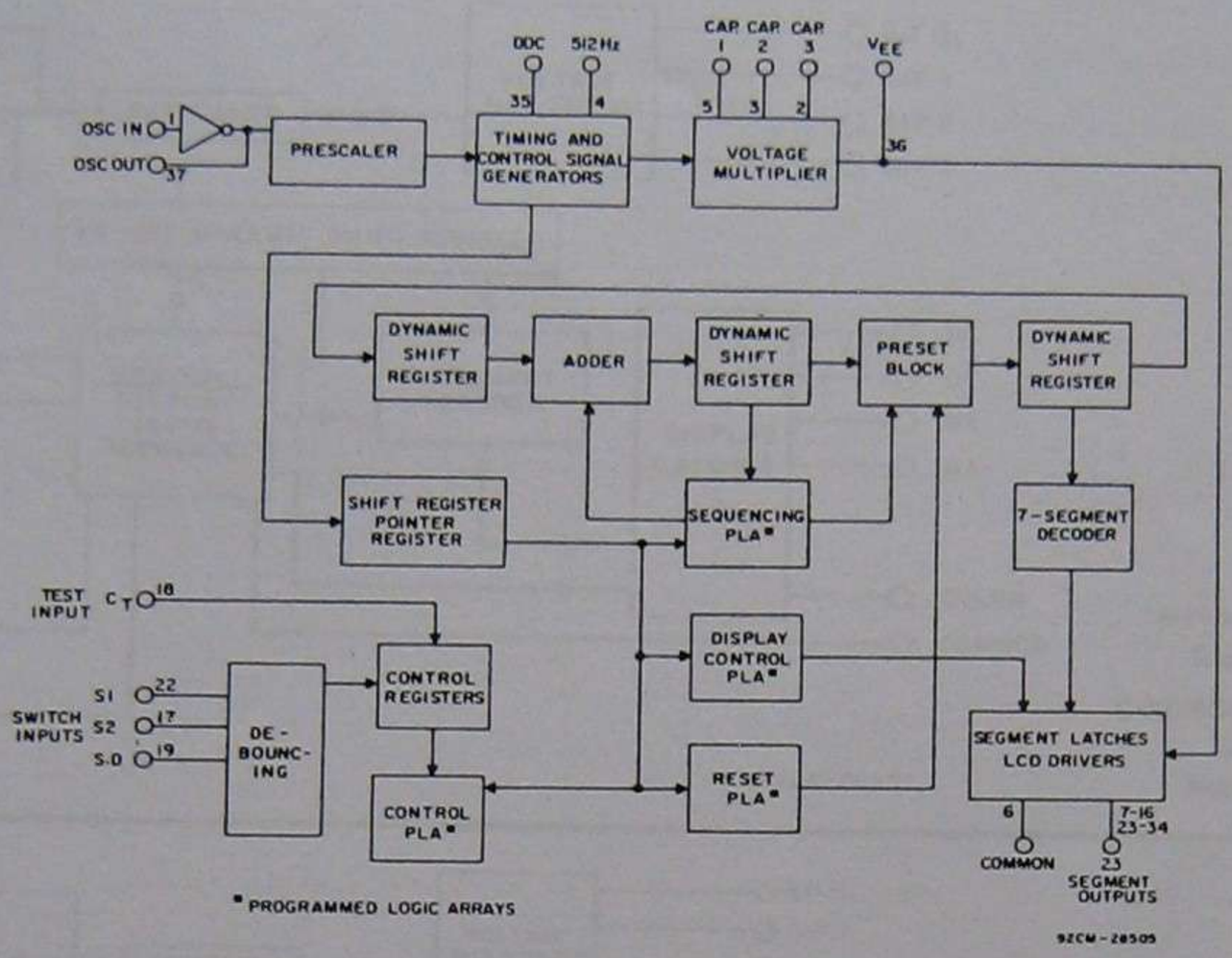
Dual Binary-to-1-of-4 Decoder/Demultiplexer Output "High" on Select

CD4555B (File No. 858)



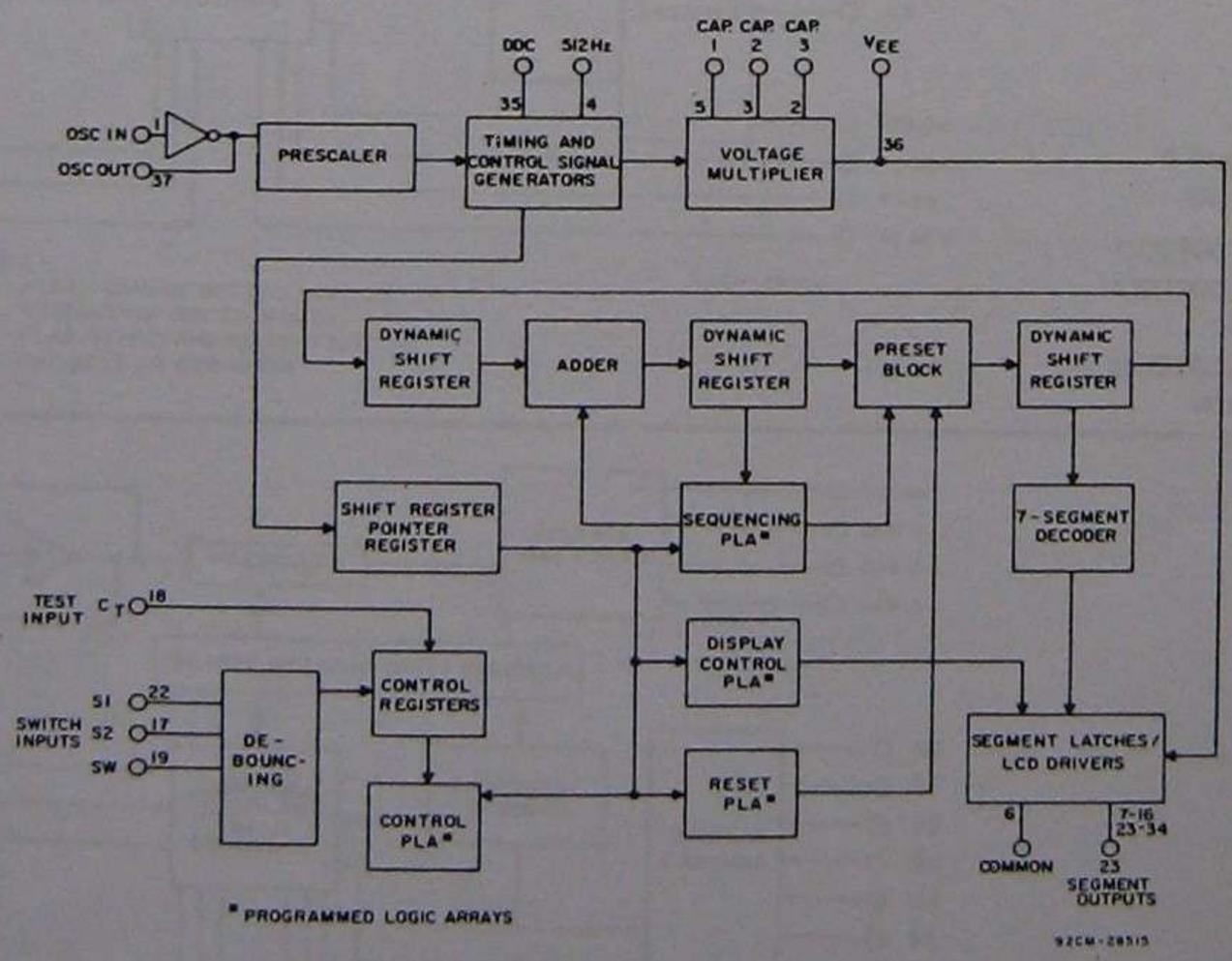
Dual Binary-to-1-of-4 Decoder/Demultiplexer Output "Low" on Select

CD4556B (File No. 858)



5-Function Liquid-Crystal Digital Watch Circuit

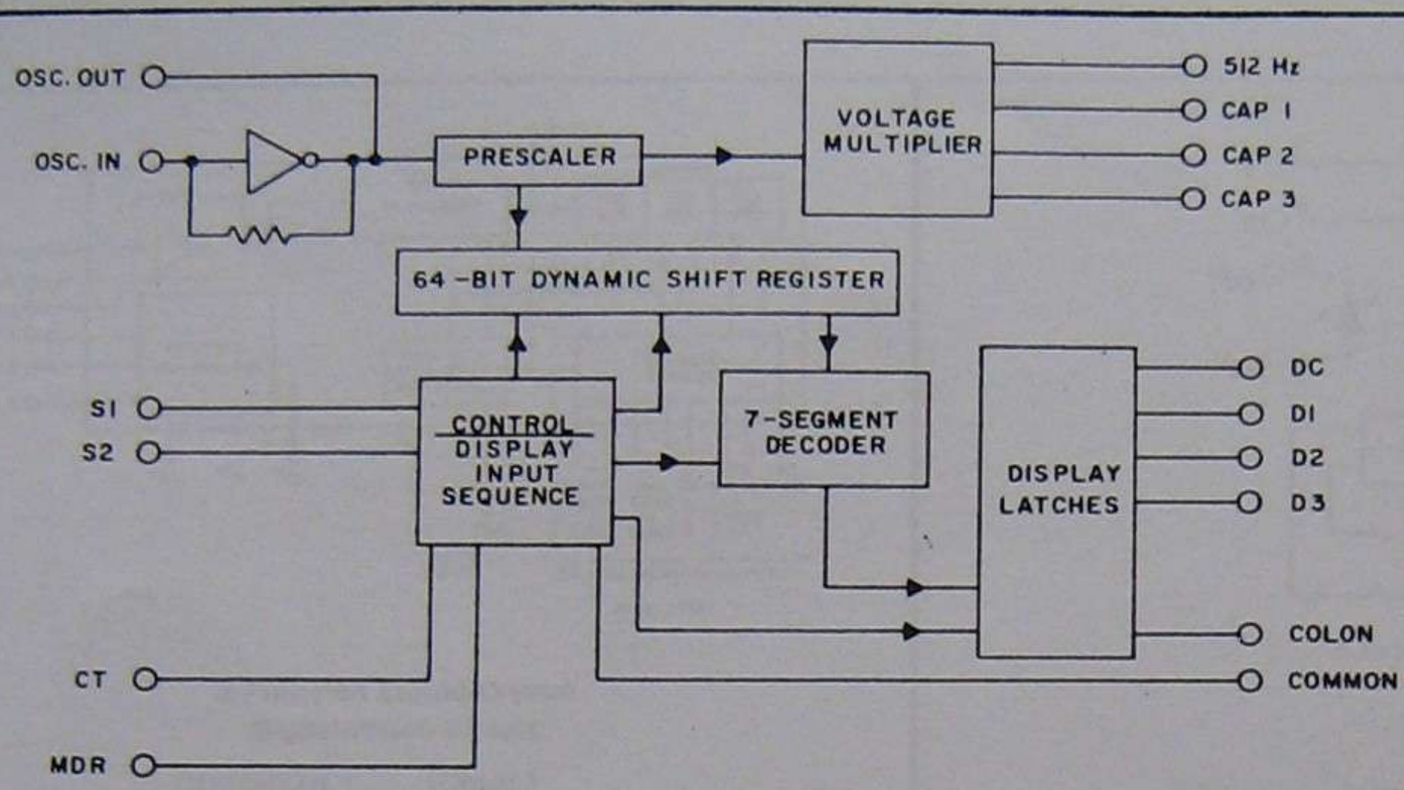
CD22001H (File No. 971)
CD22002H (File No. 971)
Automatic and Manual Reset



6-Function Liquid-Crystal Digital-Watch Circuit with Stopwatch

CD22003H (File No. 972)
Automatic and Manual Reset
Automatic Shutdown for Power Conservation

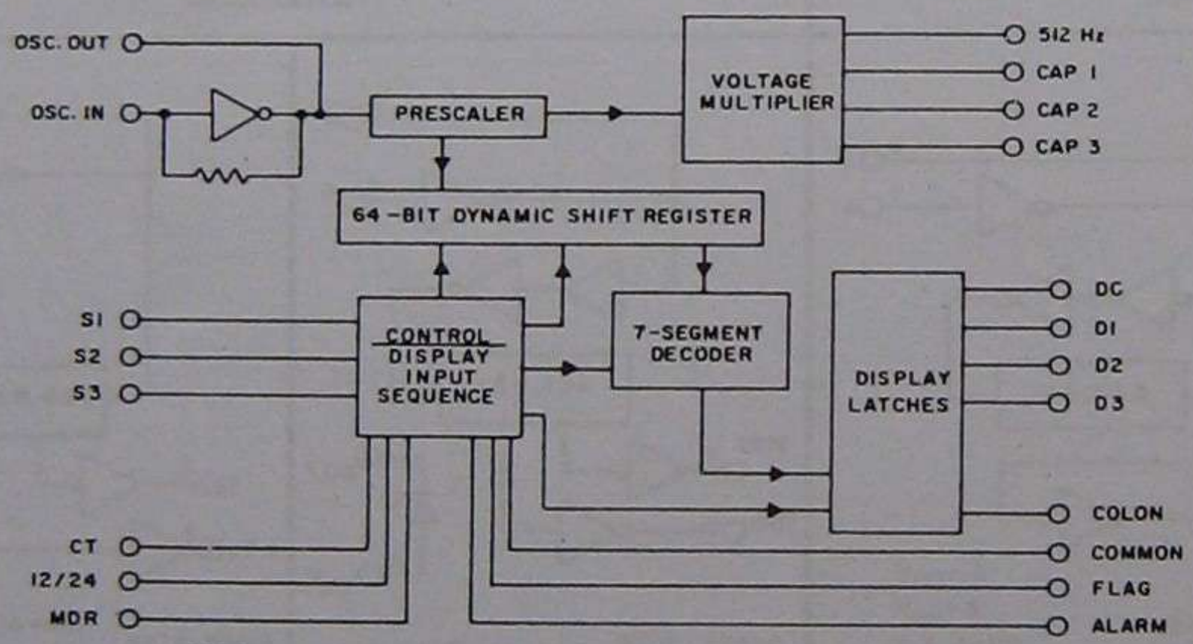
Functional Diagrams



92CM-29487

6-Function Liquid-Crystal Digital-Watch Circuit

CD22006H (Objec.)
4-Year Calendar
Automatic Shutdown

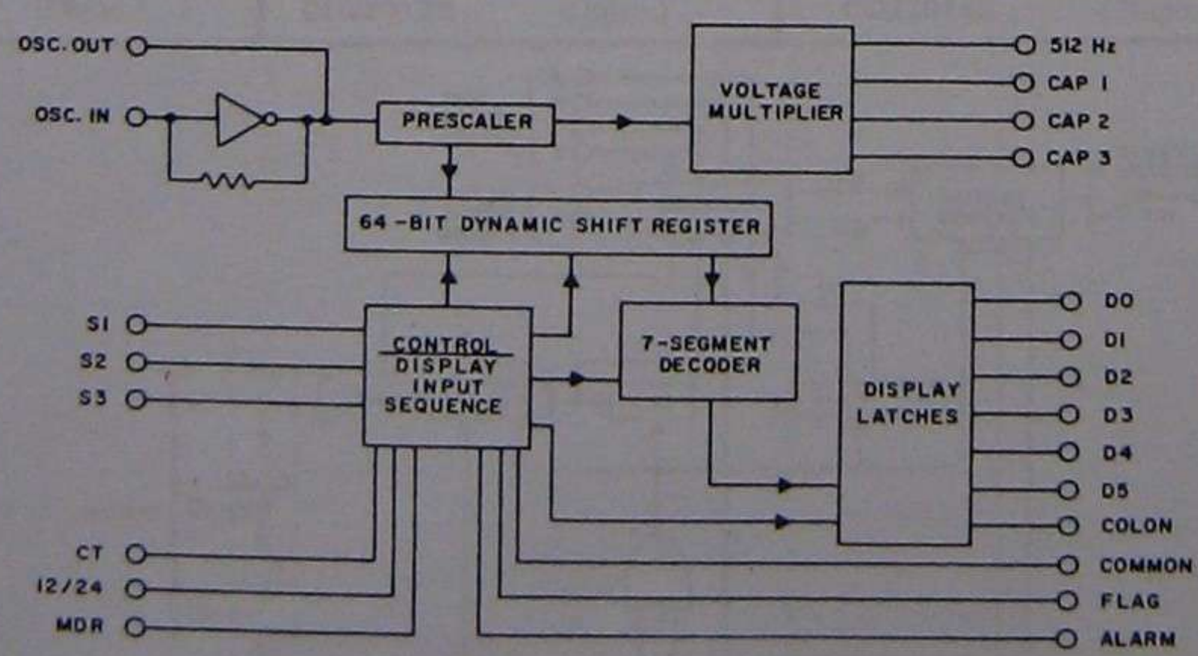


92CM-29490

NOTE:
ALARM OUTPUT NOT INCLUDED ON CD22007VIH AND CD22007H; FLAG OUTPUT AND S3 INPUT NOT INCLUDED ON CD22007H

6-Function Liquid-Crystal Digital-Watch Circuit

CD22007H (Objec.)
CD22007VIH (Objec.)
With Stopwatch
CD22007V2H (Objec.)
With 24-Hour Alarm

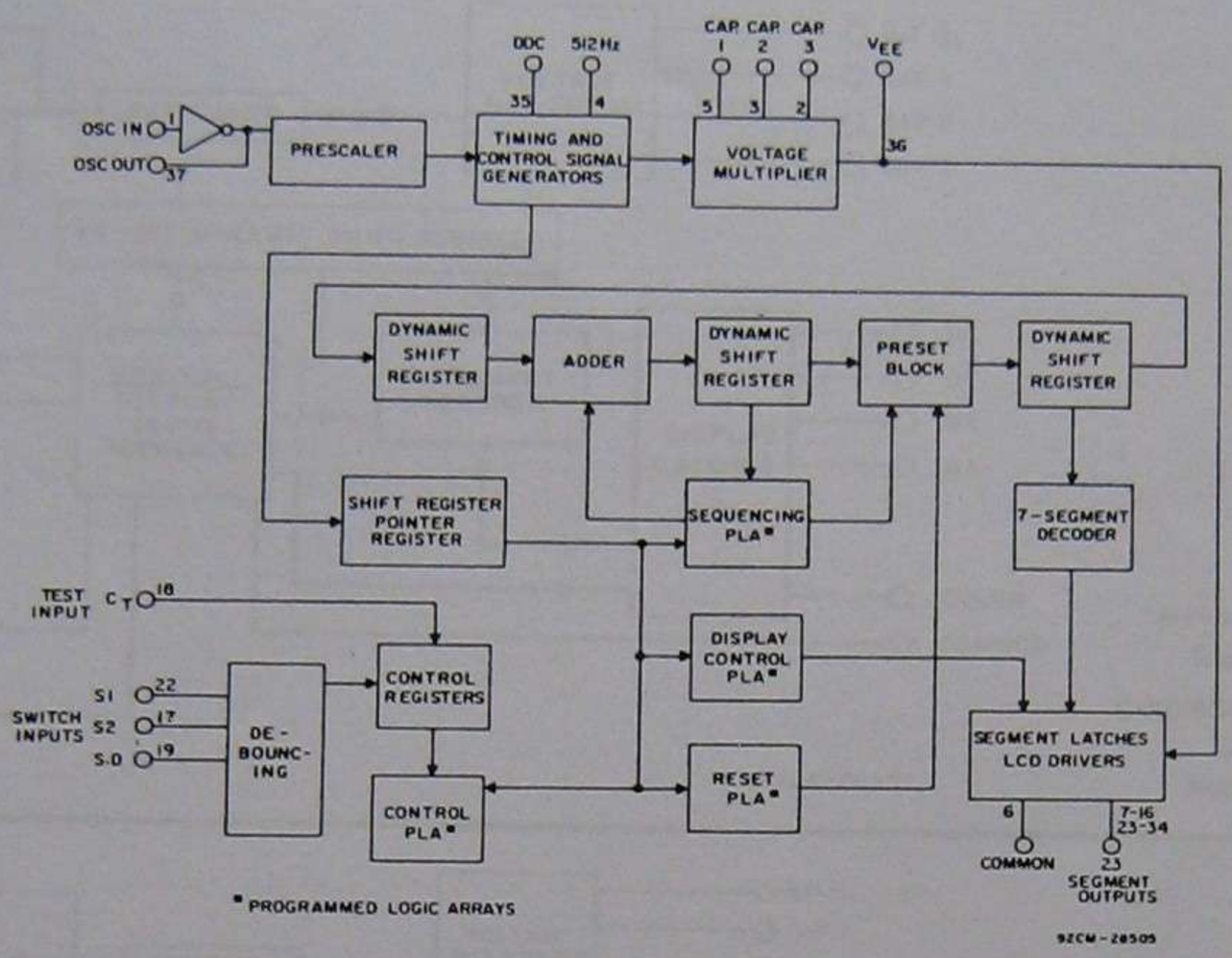


92CM-29484

NOTE:
ALARM OUTPUT NOT INCLUDED ON CD22008VIH AND CD22008H; FLAG OUTPUT AND S3 INPUT NOT INCLUDED ON CD22008H.

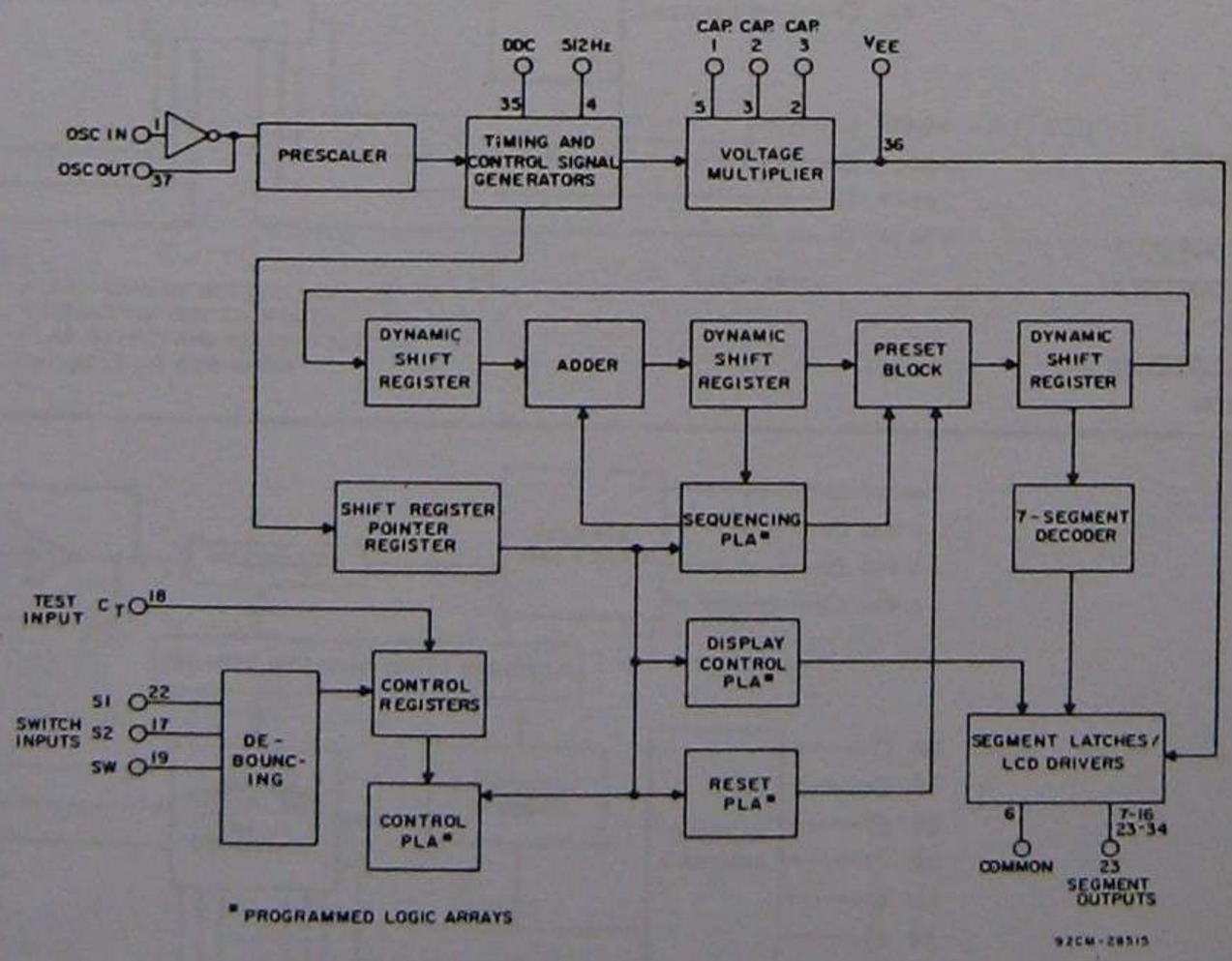
6-Function Liquid-Crystal Digital-Watch Circuit

CD22008H (Objec.)
CD22008VIH (Objec.)
With Stopwatch
CD22008V2H (Objec.)
With 24-Hour Alarm



5-Function Liquid-Crystal Digital Watch Circuit

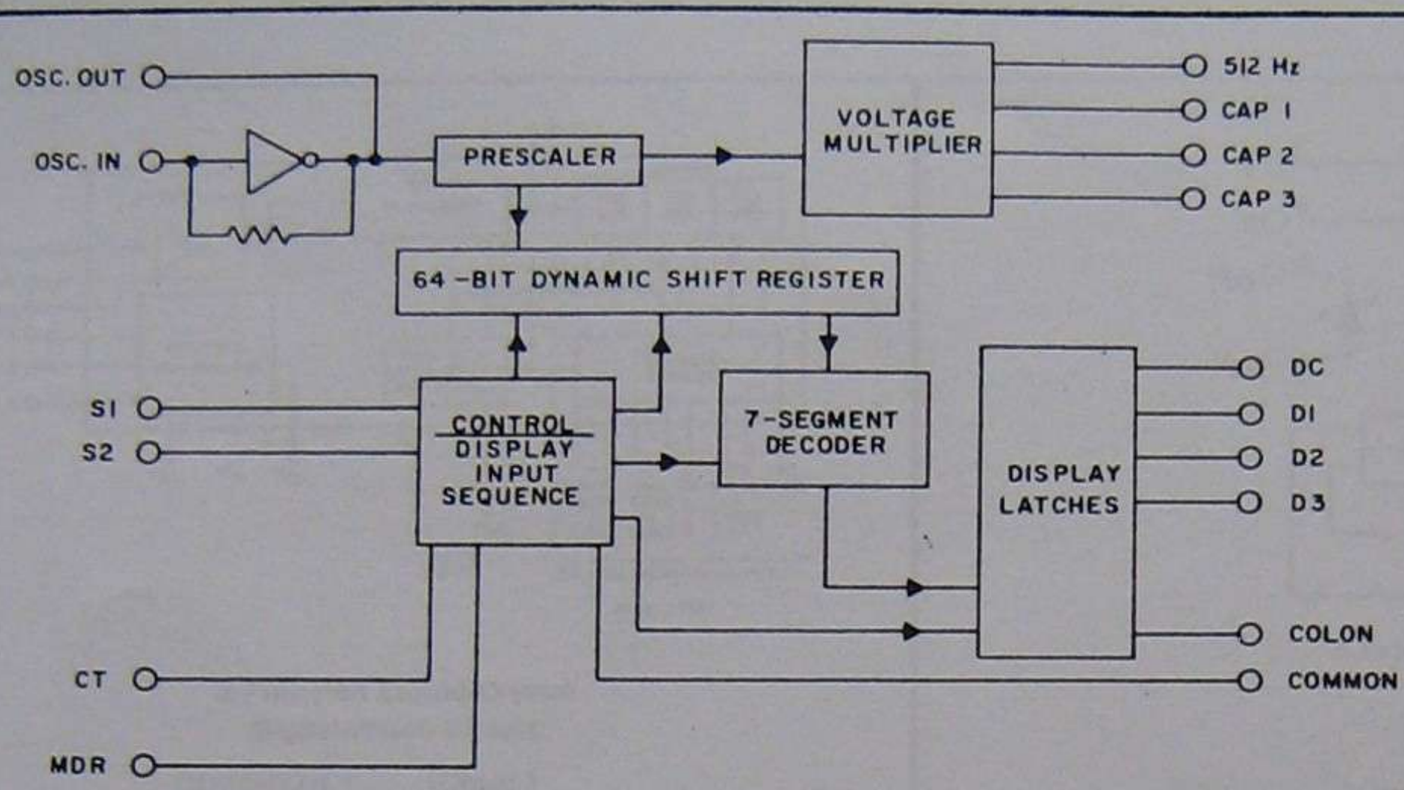
CD22001H (File No. 971)
CD22002H (File No. 971)
Automatic and Manual Reset



6-Function Liquid-Crystal Digital-Watch Circuit with Stopwatch

CD22003H (File No. 972)
Automatic and Manual Reset
Automatic Shutdown for Power Conservation

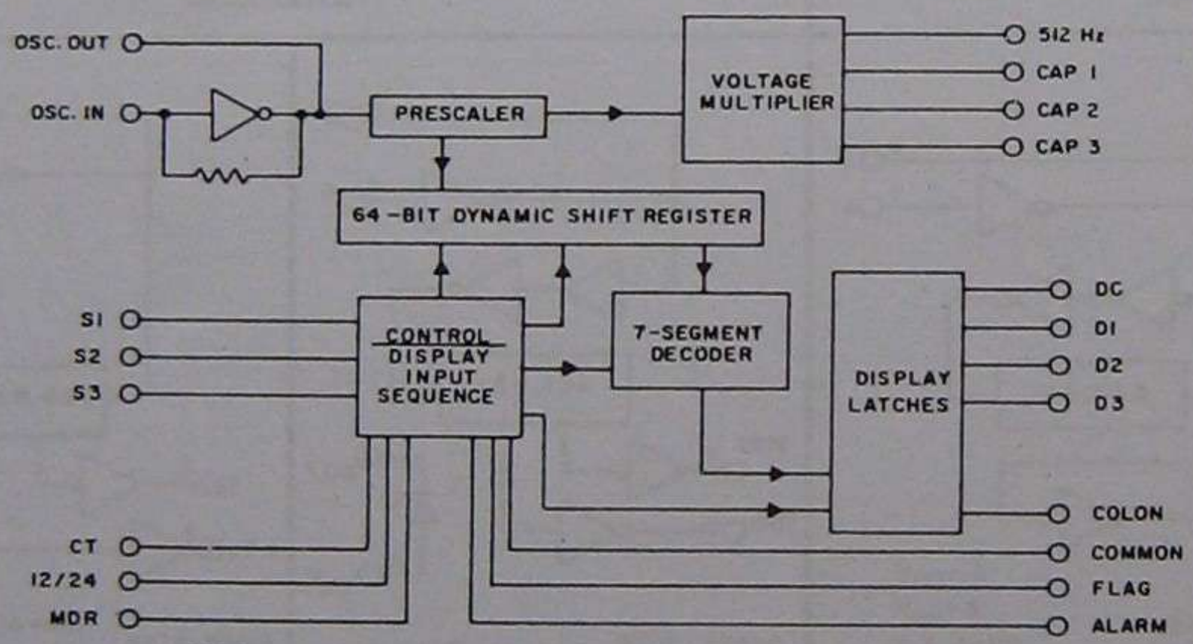
Functional Diagrams



92CM-29487

6-Function Liquid-Crystal Digital-Watch Circuit

CD22006H (Objec.)
4-Year Calendar
Automatic Shutdown

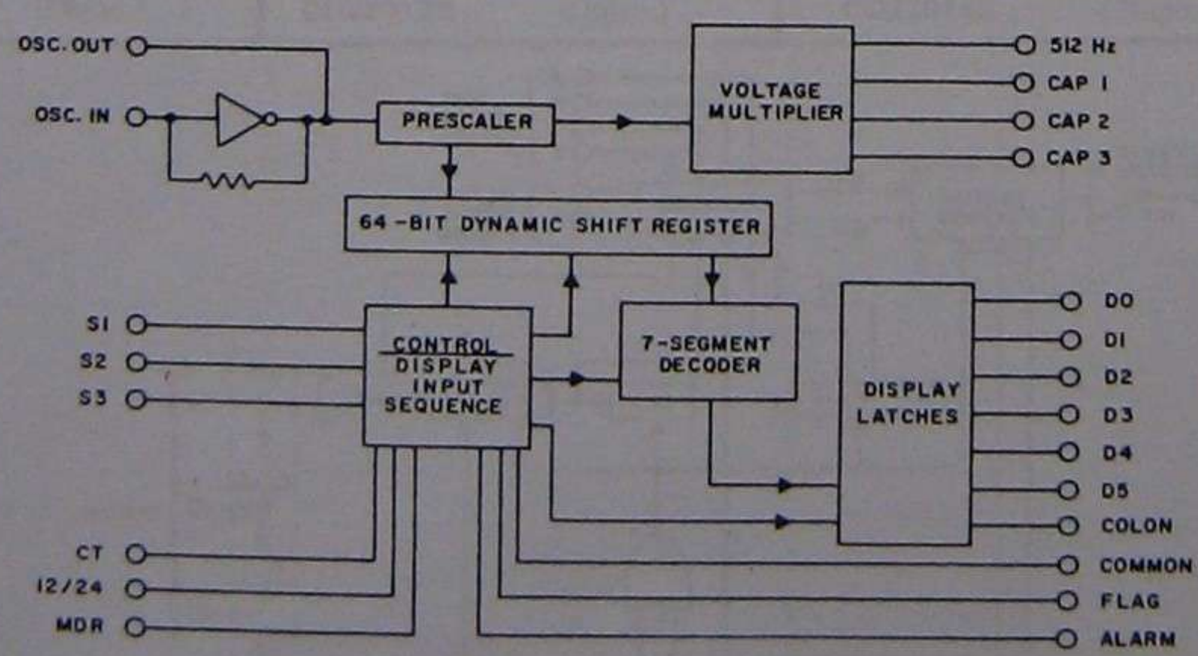


92CM-29490

NOTE:
ALARM OUTPUT NOT INCLUDED ON CD22007VIH AND CD22007H; FLAG OUTPUT AND S3 INPUT NOT INCLUDED ON CD22007H

6-Function Liquid-Crystal Digital-Watch Circuit

CD22007H (Objec.)
CD22007VIH (Objec.)
With Stopwatch
CD22007V2H (Objec.)
With 24-Hour Alarm



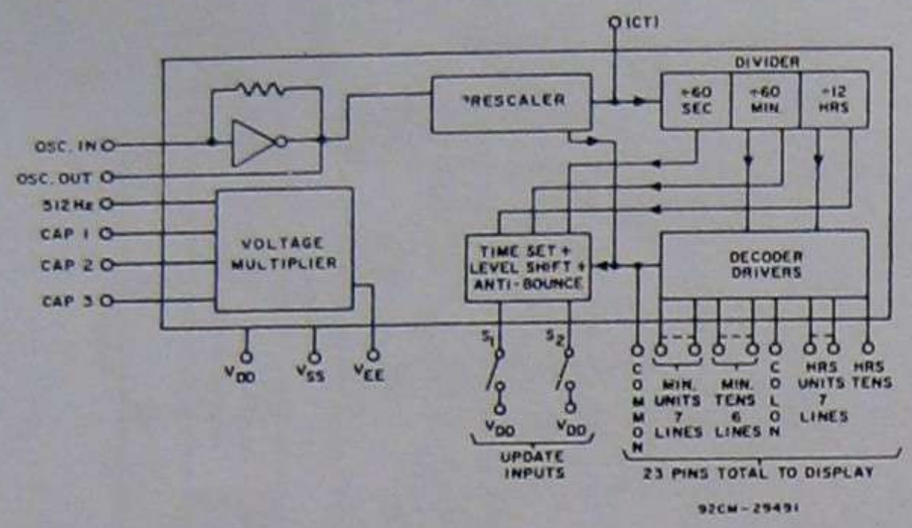
92CM-29484

NOTE:
ALARM OUTPUT NOT INCLUDED ON CD22008VIH AND CD22008H; FLAG OUTPUT AND S3 INPUT NOT INCLUDED ON CD22008H.

6-Function Liquid-Crystal Digital-Watch Circuit

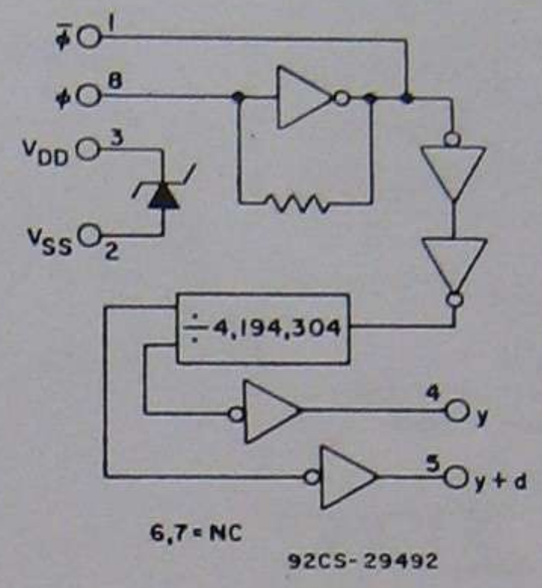
CD22008H (Objec.)
CD22008VIH (Objec.)
With Stopwatch
CD22008V2H (Objec.)
With 24-Hour Alarm

Functional Diagrams



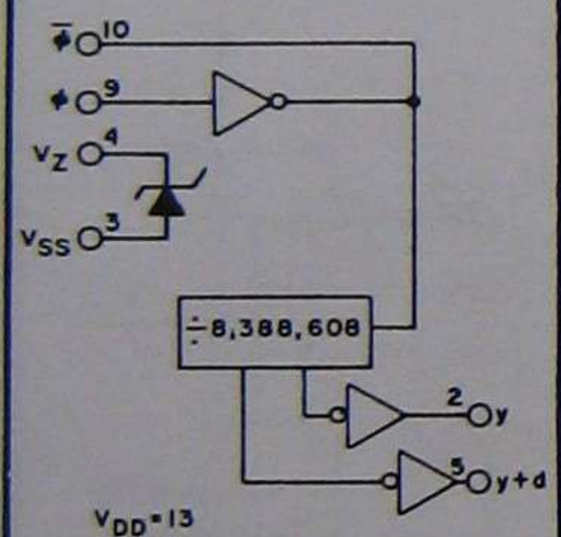
2-Function Liquid-Crystal Digital-Watch Circuit

CD22009H (Objec.)
CD22009RH (Objec.)
Mirror-Image of CD22009H



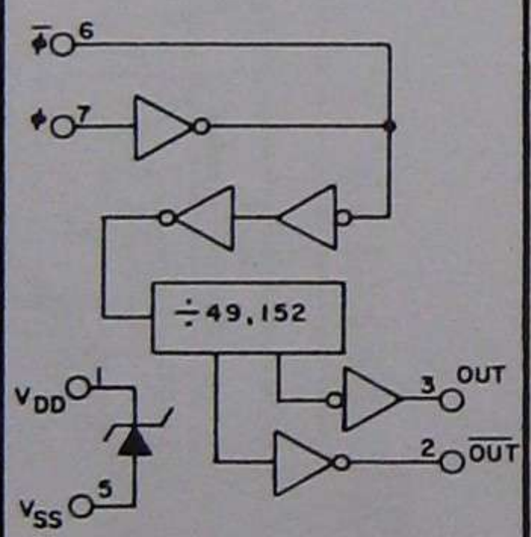
Quartz Analog Clock Circuit

CD22011E (Objec.)



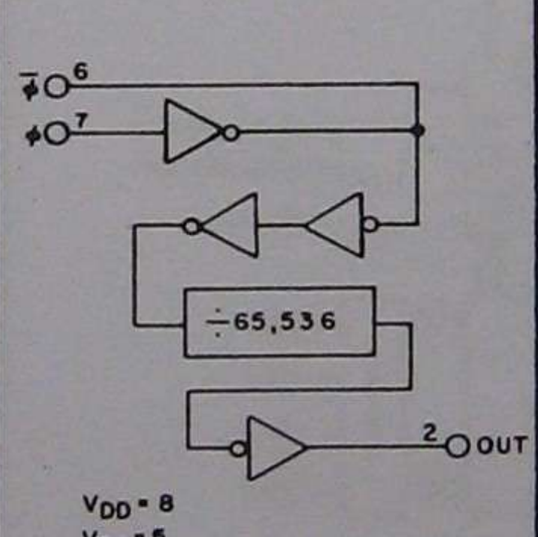
Quartz Analog Clock Circuit

CD22012E (Objec.)



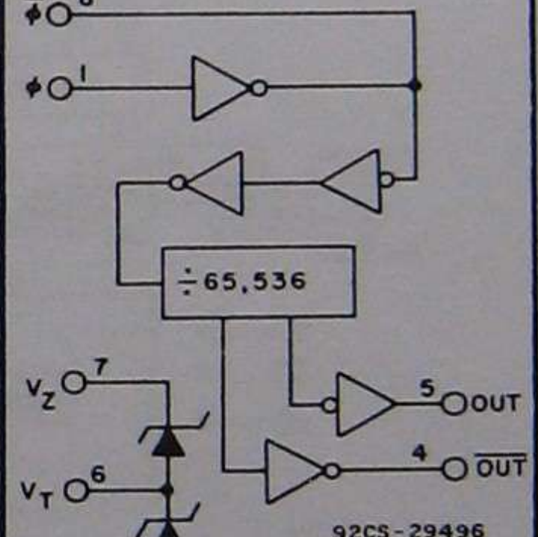
Quartz Analog Clock Circuit

CD22013E (Objec.)



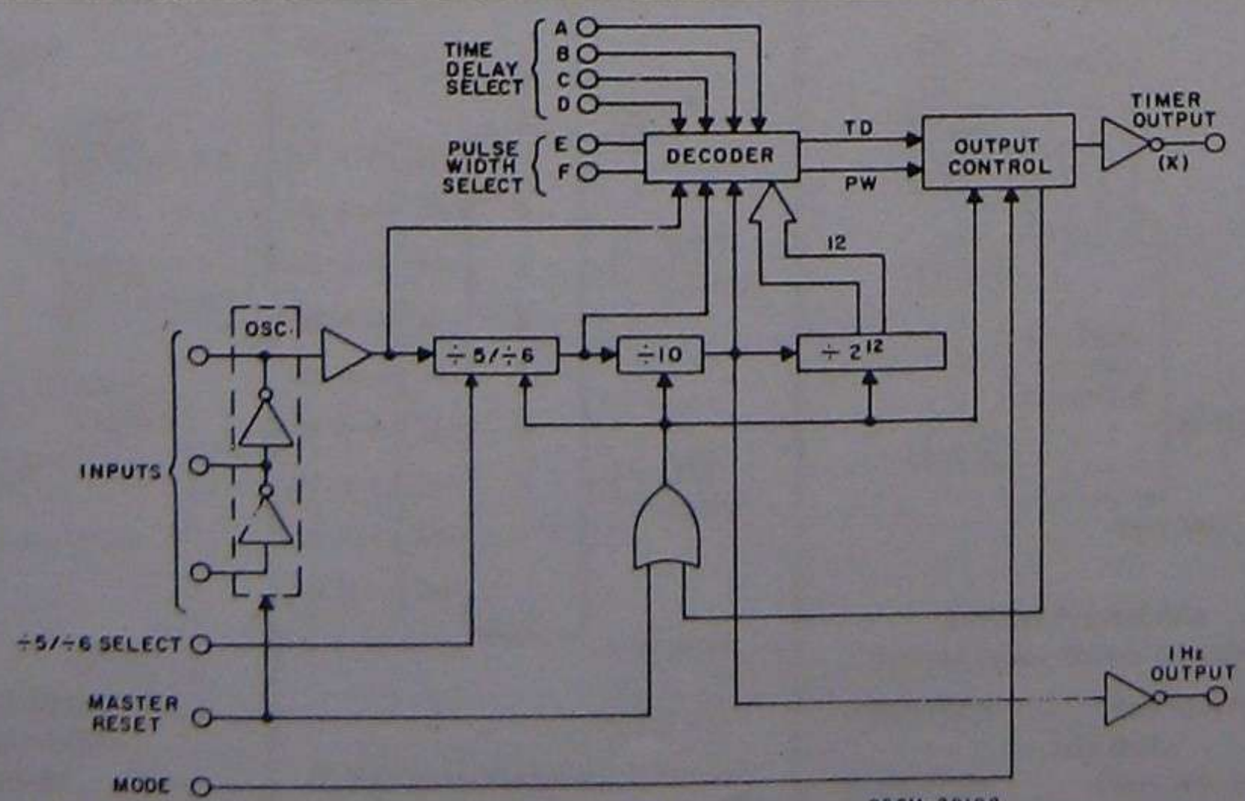
Quartz Analog Clock Circuit

CD22014E (Objec.)



Quartz Analog Clock Circuit

CD22015E (Objec.)

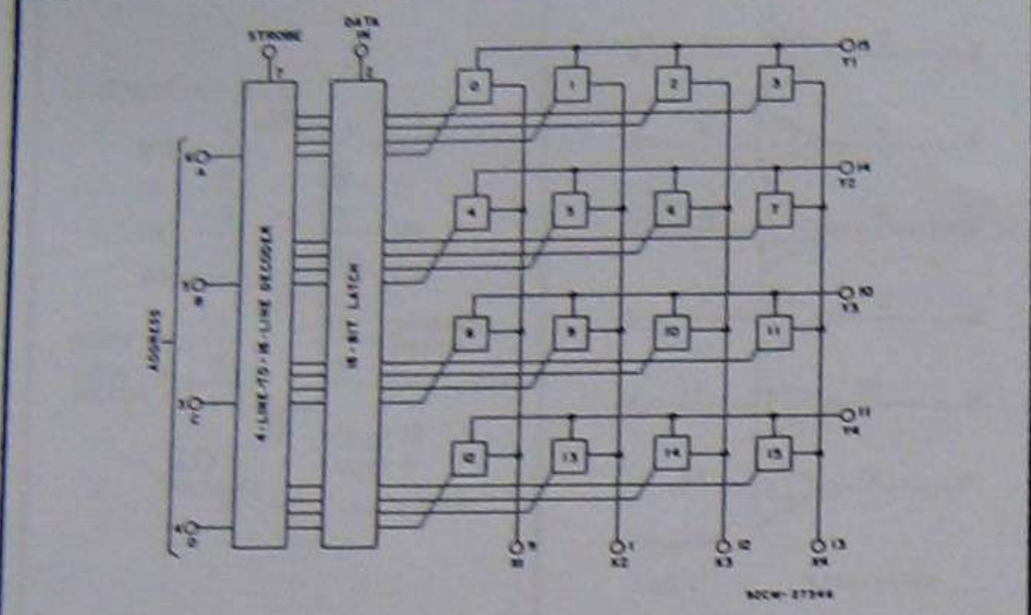


M = "0" (SINGLE TRANSITION)
M = "1" (SINGLE PULSE)
M = (X) (RECYCLE)

Industrial Timer Circuit

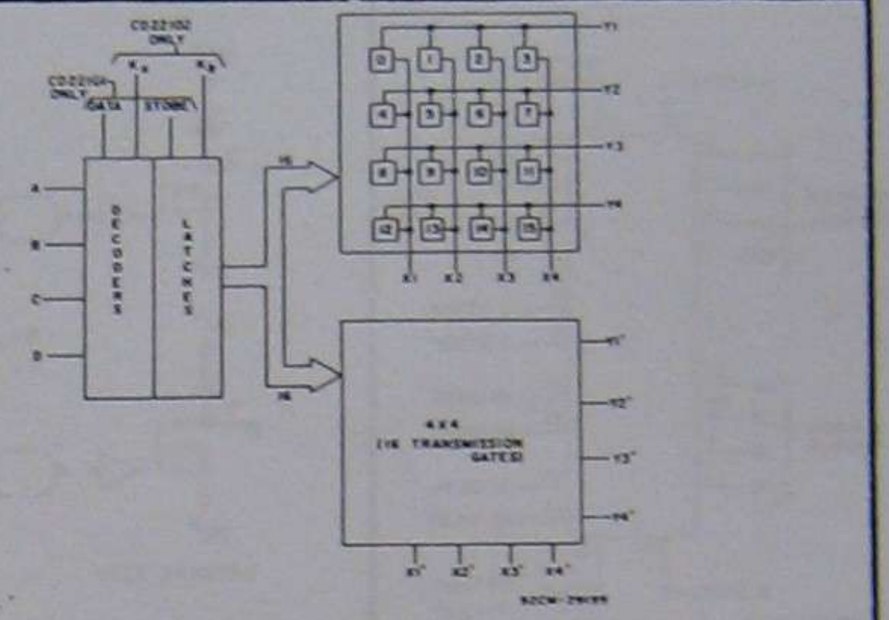
CD22017E (Objec.)

Functional Diagrams



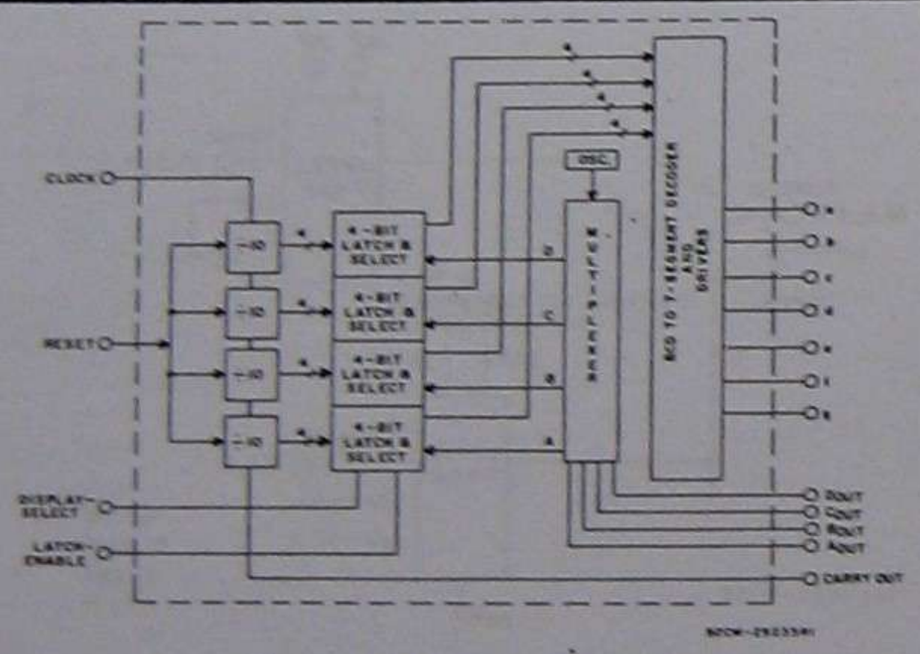
4-by-4 Crosspoint Switch
with Control Memory

CD22100 (Prel.)



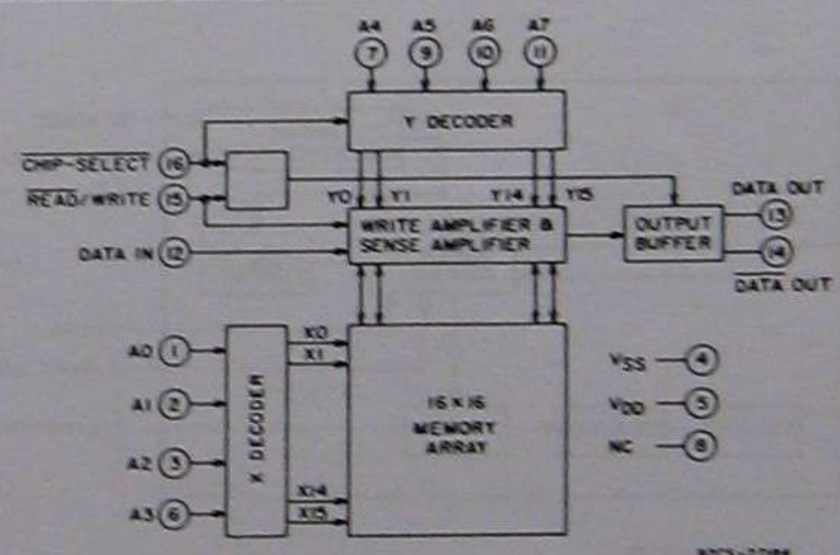
4-by-4-by-2 Crosspoint
Switch with
Control Memory

CD22101 (Objec.)
CD22102 (Objec.)



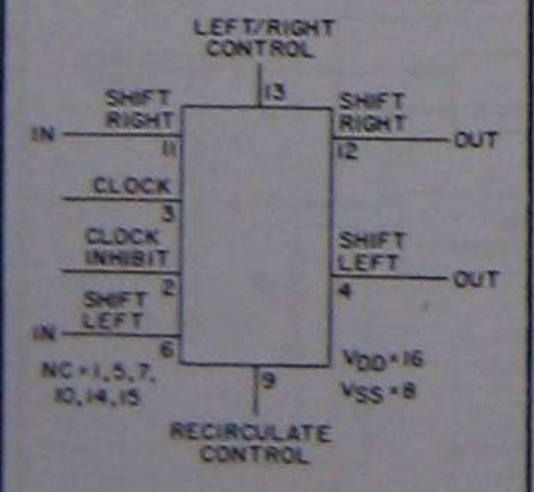
Counter and 7-Segment Decoder
with Multiplexer Driver

CD22925	(Objec.)	CD22927	(Objec.)
CD22926	(Objec.)	CD22928	(Objec.)



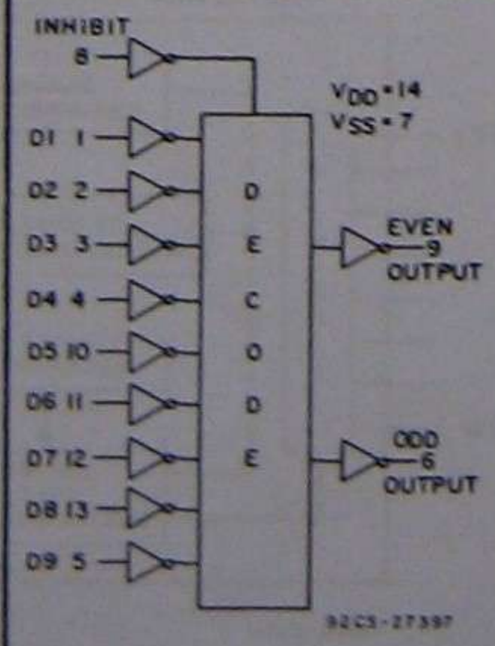
256-Word by 1-Bit Static
Random-Access Memory

CD40061 (Prel.)
CD40061A (Prel.)



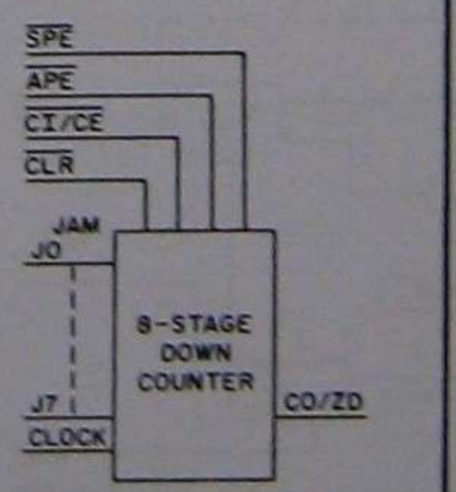
32-Stage Static
Left/Right Shift
Register

CD40100B (File No. 980)



9-Bit Parity Generator/Checker

CD40101B (File No. 1000)



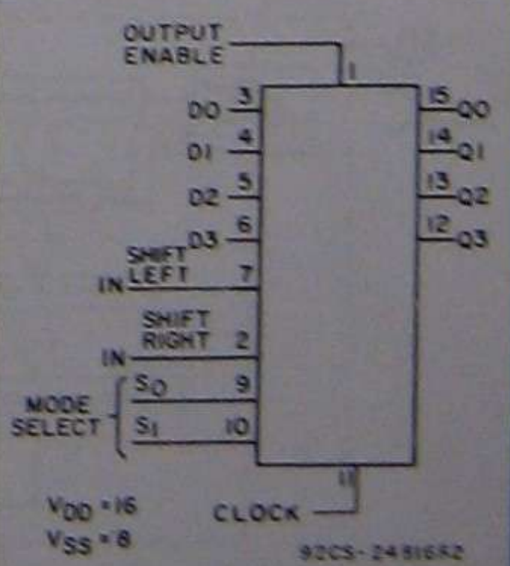
8-Stage Presettable
Synchronous Down Counter

CD40102B (File No. 984)

2-Decade BCD

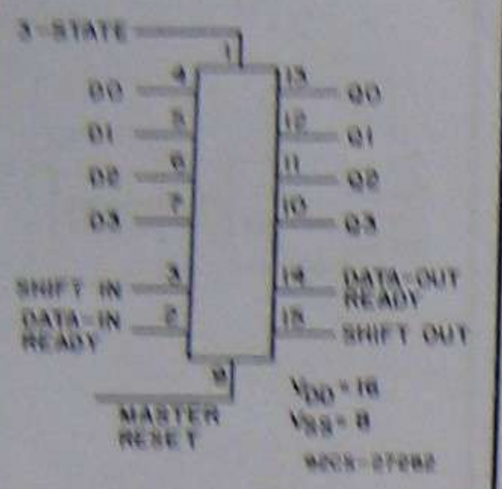
CD40103B (File No. 984)

8-Bit Binary



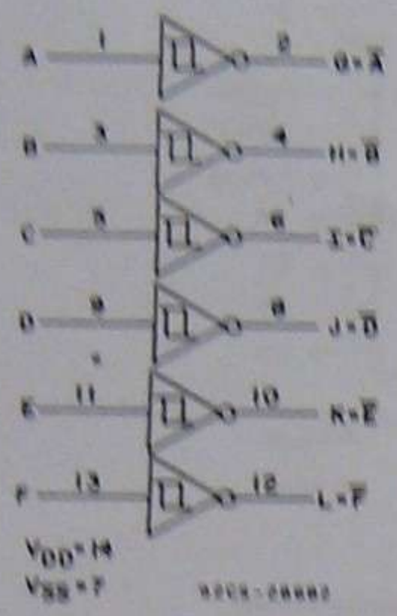
4-Bit Bidirectional
Universal Register

CD40104B (Prel.)



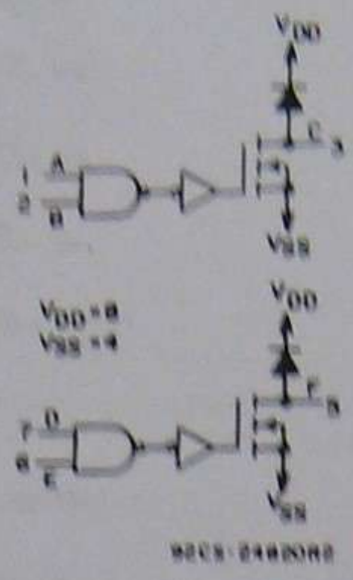
FIFO Register
4-Bits Wide by 16-Bits Long

CD40105B (Prel.)



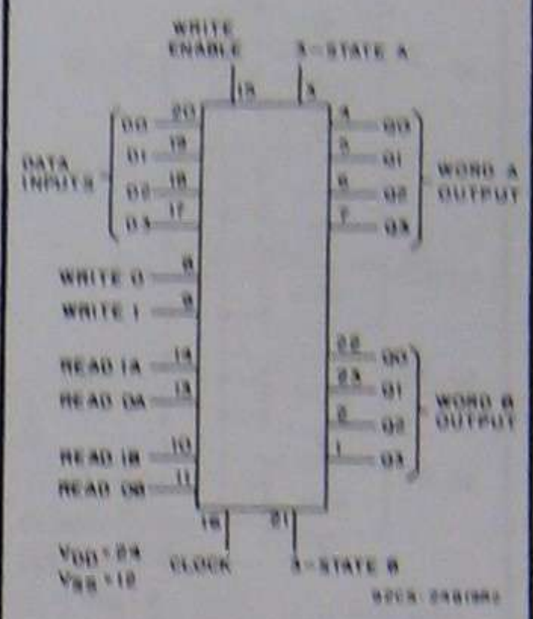
Hex Schmitt Trigger

CD40106B (File No. 1017)



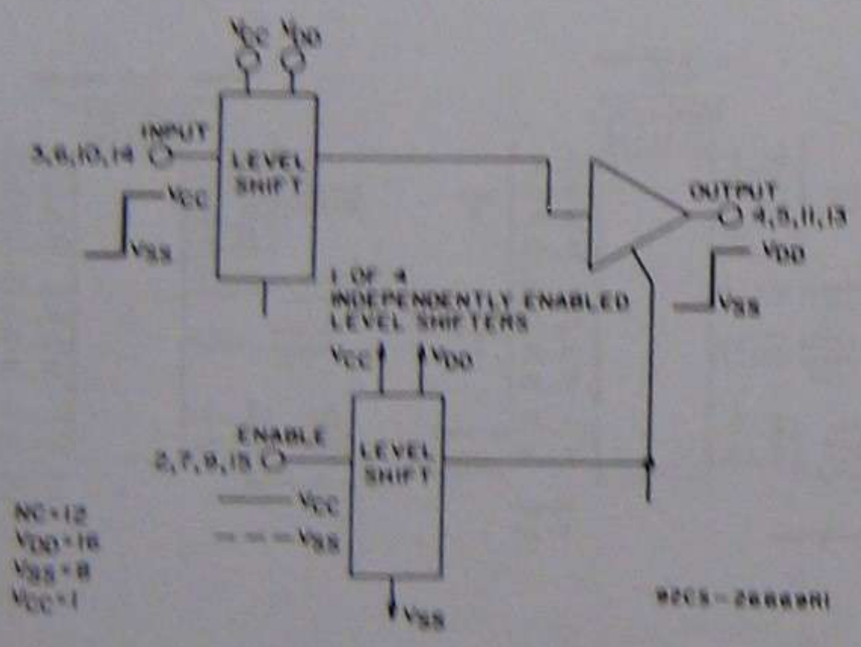
Dual 2-Input NAND
Buffer/Driver

CD40107B (File No. 1015)



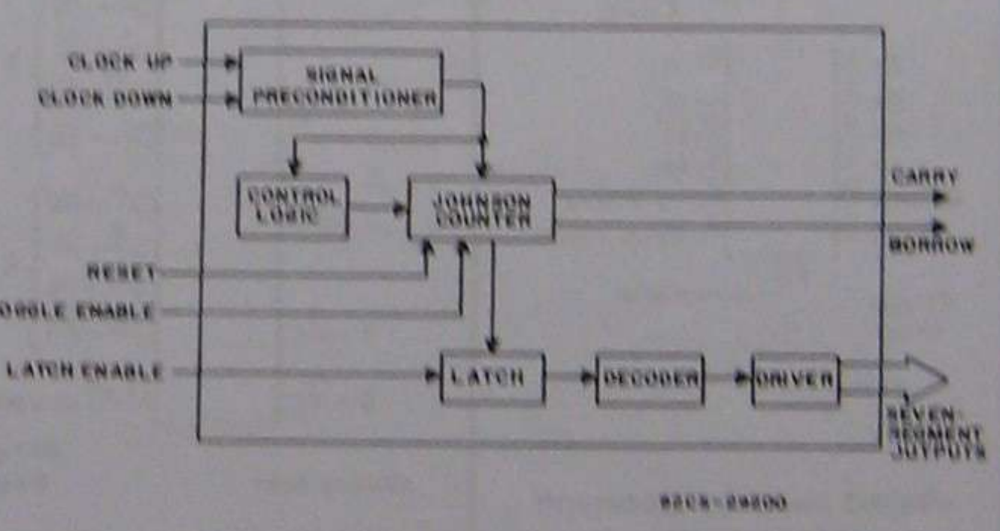
4-by-4 Multiport Register

CD40108B (File No. 1011)



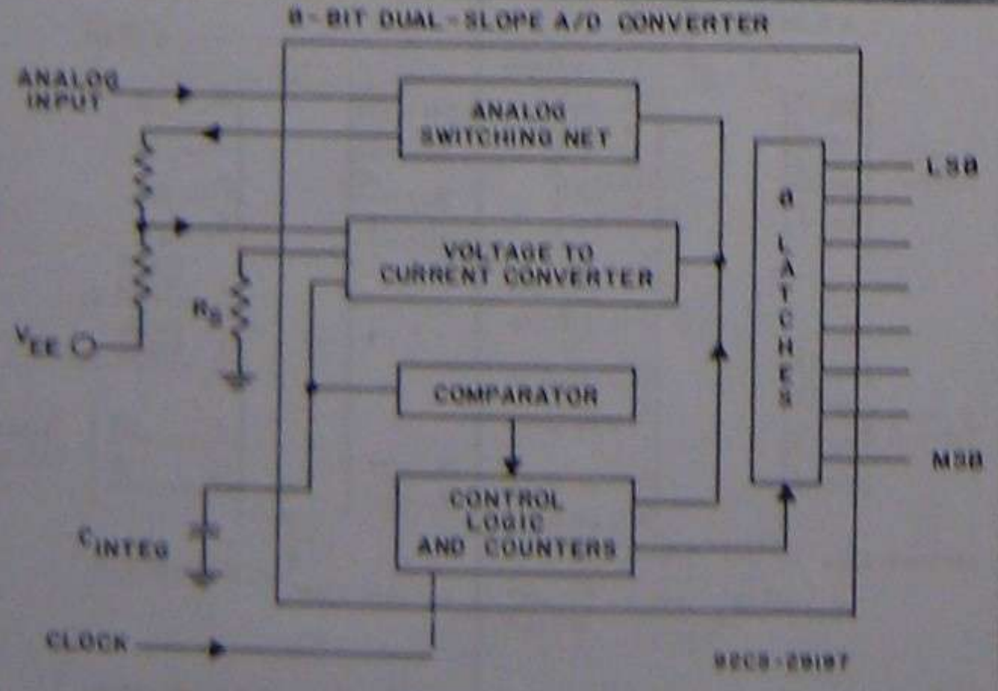
Quad Low-to-High
Voltage Level Shifter

CD40109B (File No. 1018)



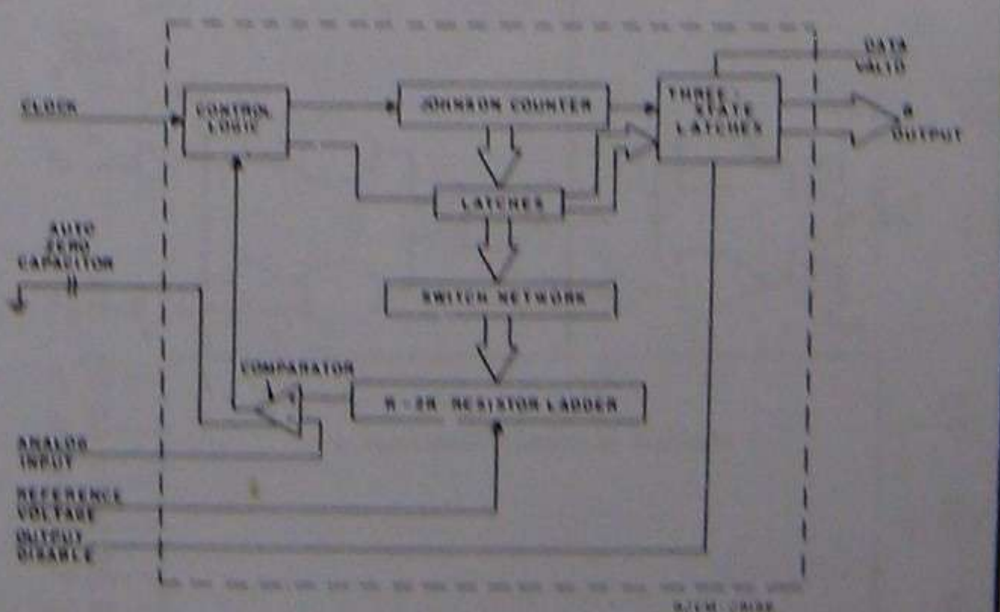
Decade Up-Down Counter/
Decoder/Latch/Driver

CD40110B (Objec.)



8-Bit Dual Slope
A/D Converter

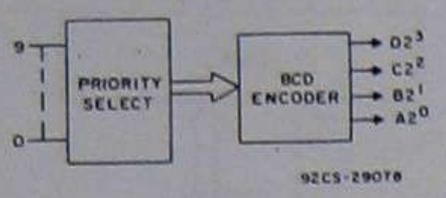
CD40112B (Objec.)



Successive Approximation
8-Bit A/D Converter

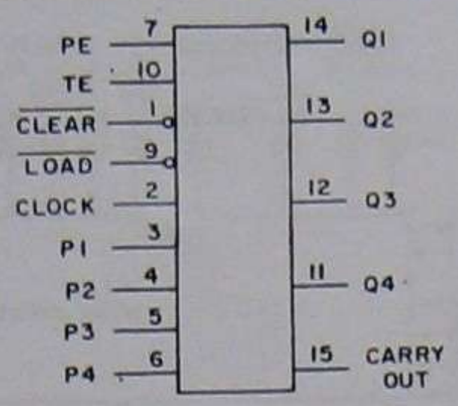
CD40113B (Objec.)

Functional Diagrams



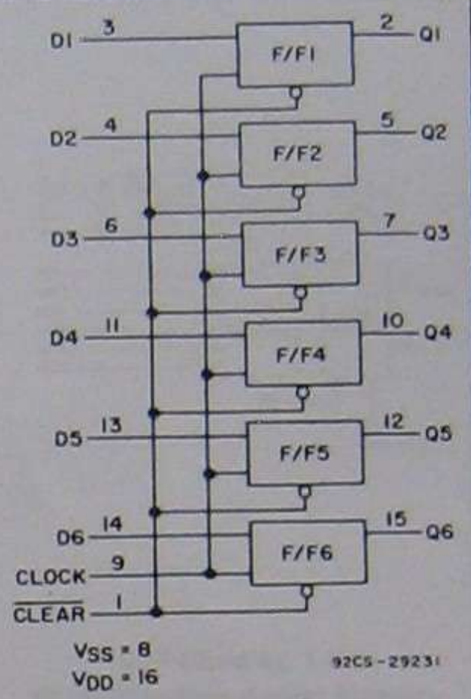
10-Line-to-4-Line
BCD Priority Encoder

CD40147B (Objec.)



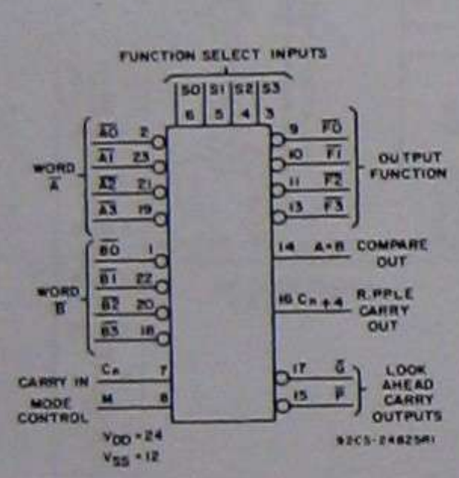
Synchronous 4-Bit Counter

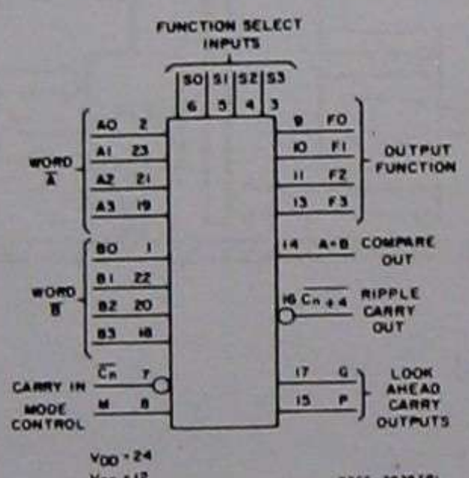
CD40160B (Prel.) Decade with Asynchronous Clear
CD40161B (Prel.) Binary with Asynchronous Clear
CD40162B (Prel.) Decade with Synchronous Clear
CD40163B (Prel.) Binary with Synchronous Clear



Hex "D" Type Flip-Flop

CD40174B (Prel.)

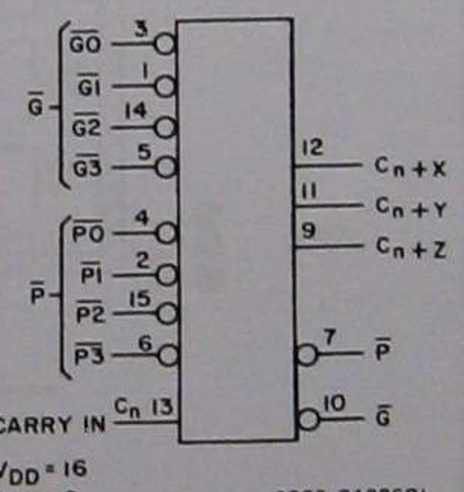




4-Bit Arithmetic Logic Unit

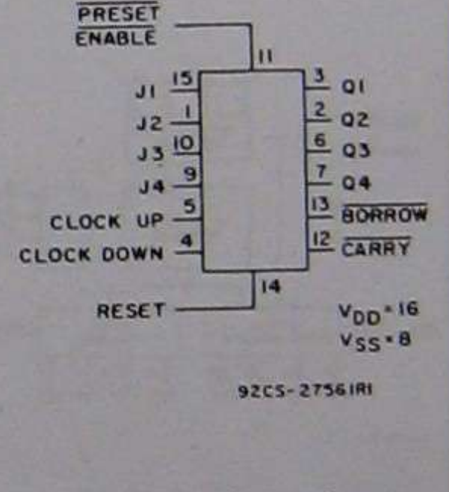
Active-Low Data Active-High Data

CD40181B (Prel.)



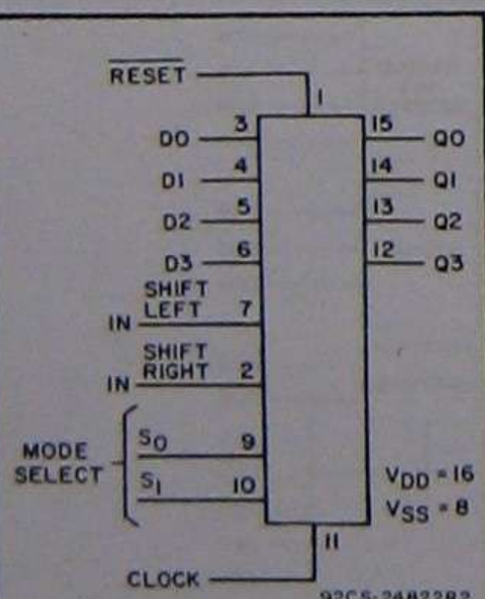
Look-Ahead Carry Generator

CD40182B (File No. 1008)



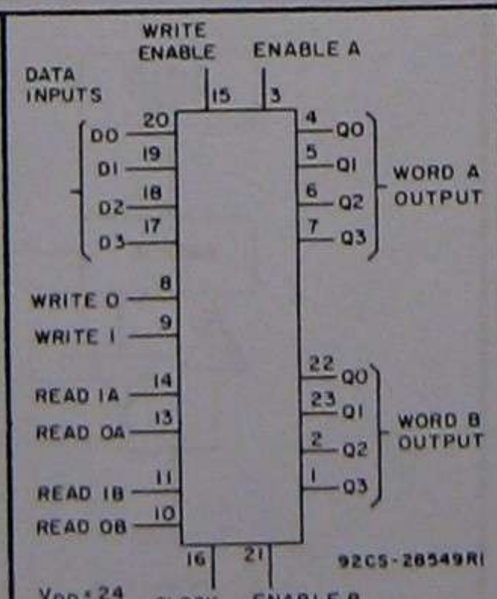
Presettable Up/Down Counter
(Dual Clock with Reset)

CD40192B (File No. 993) BCD
CD40193B (File No. 993) Binary



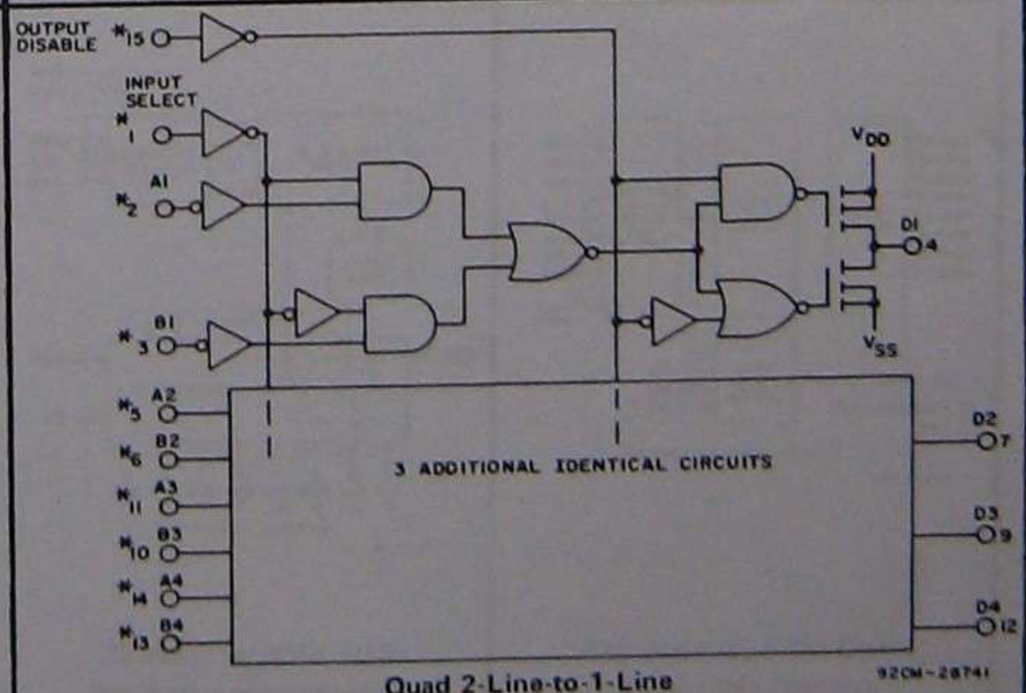
4-Bit Bidirectional
Universal Shift Register

CD40194B (Prel.)



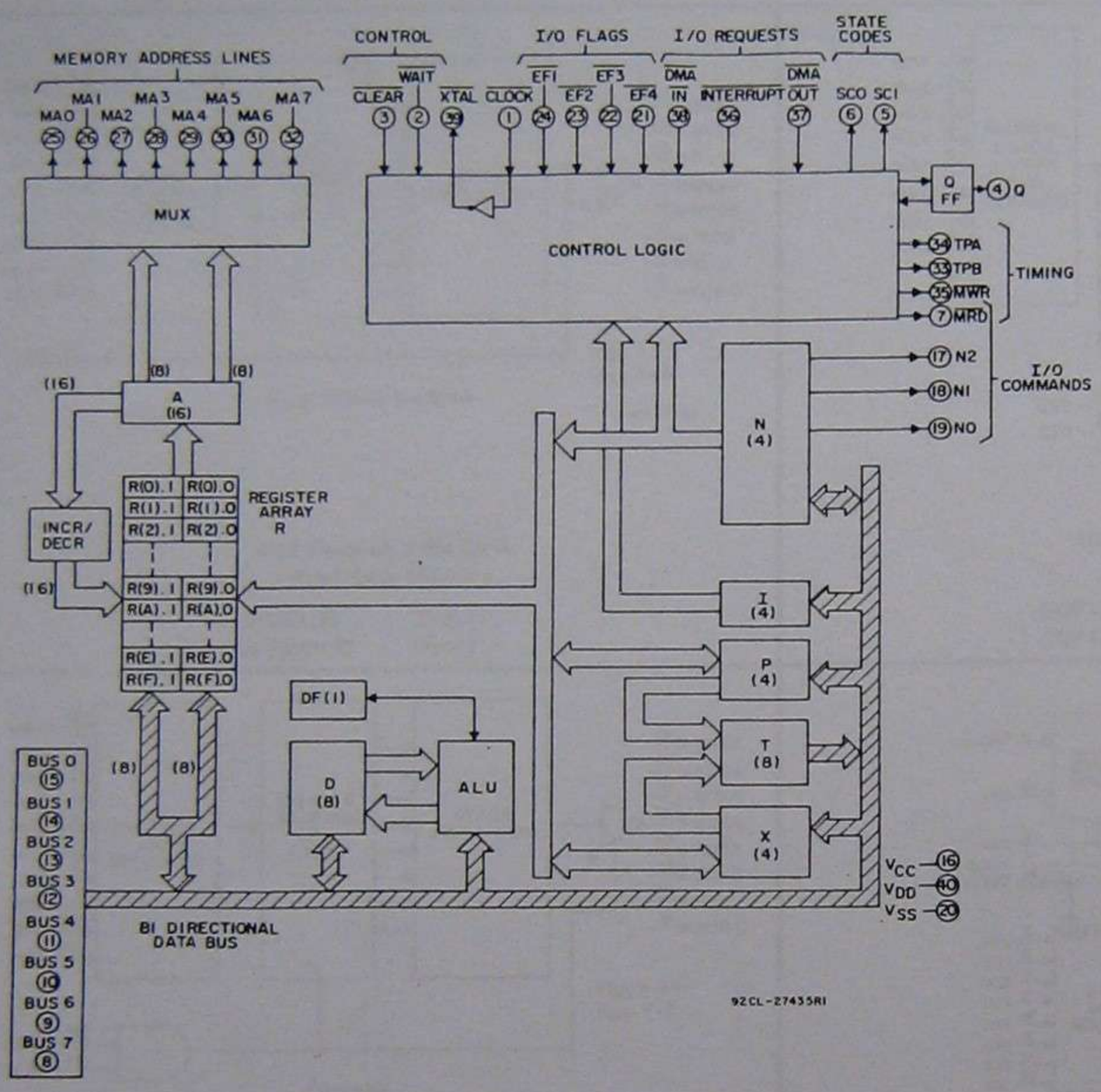
4-by-4 Multiport Register

CD40208B (File No. 1007)



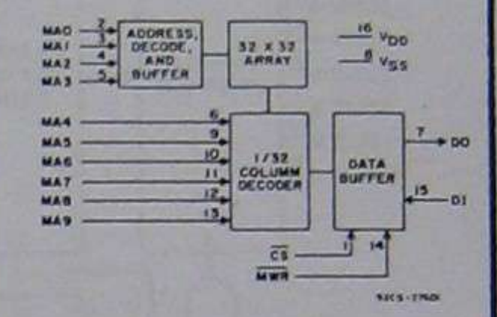
Quad 2-Line-to-1-Line
Data Selector/Multiplexer

CD40257B (File No. 982)



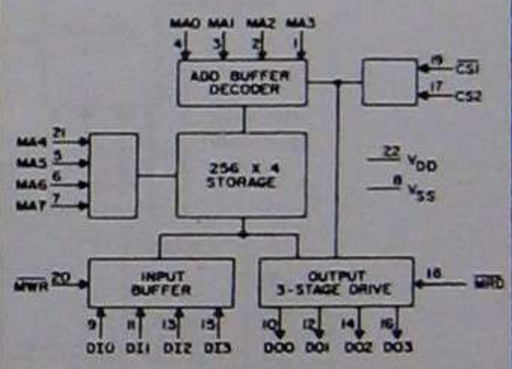
8-Bit Registered-Oriented Central Processing Unit

CDP1802C (Prel.)
CDP1802CD (Prel.)



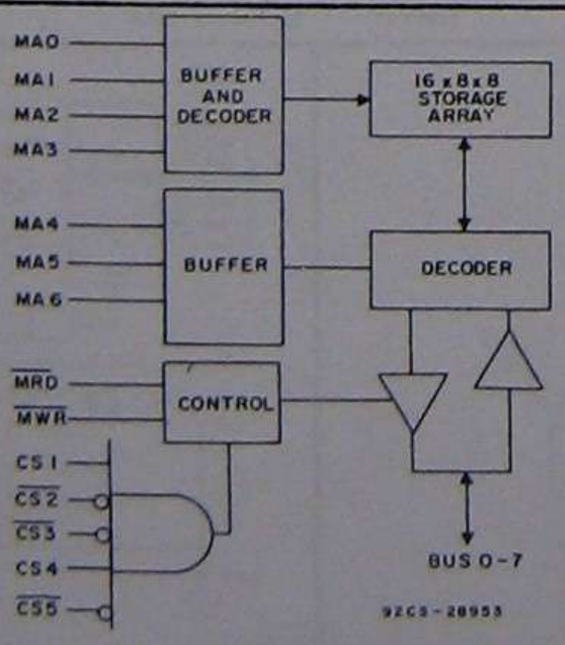
1024-Word by 1-Bit Static Random-Access Memory

CDP1821SD (Prel.)
CDP1821SCD (Prel.)



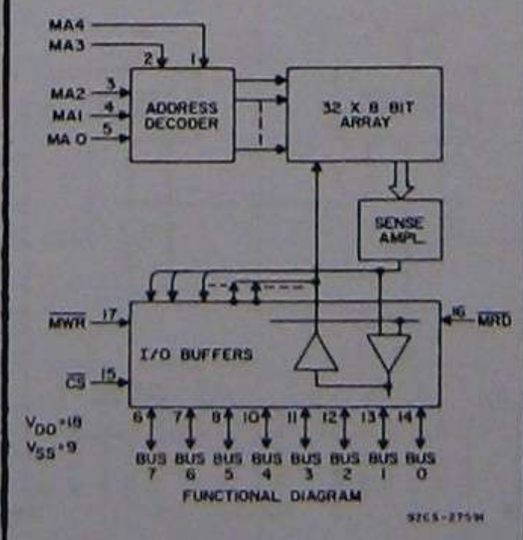
256-Word by 4-Bit Static Random-Access Memory

CDP1822SD (Prel.)
CDP1822SCD (Prel.)



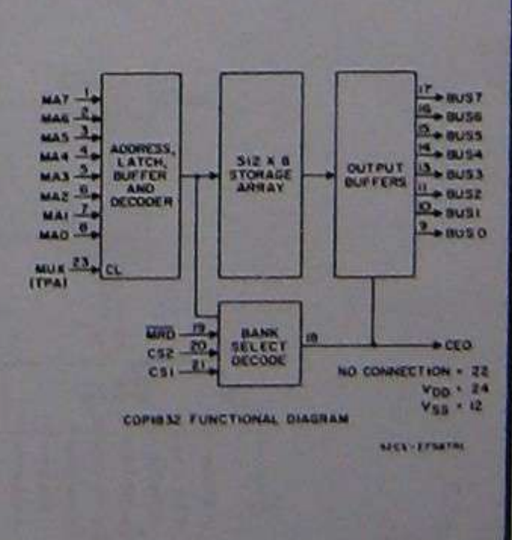
128-Word by 8-Bit Static Random-Access Memory

CDP1823SD (Prel.)
CDP1823SCD (Prel.)



32-Word by 8-Bit Static Random-Access Memory

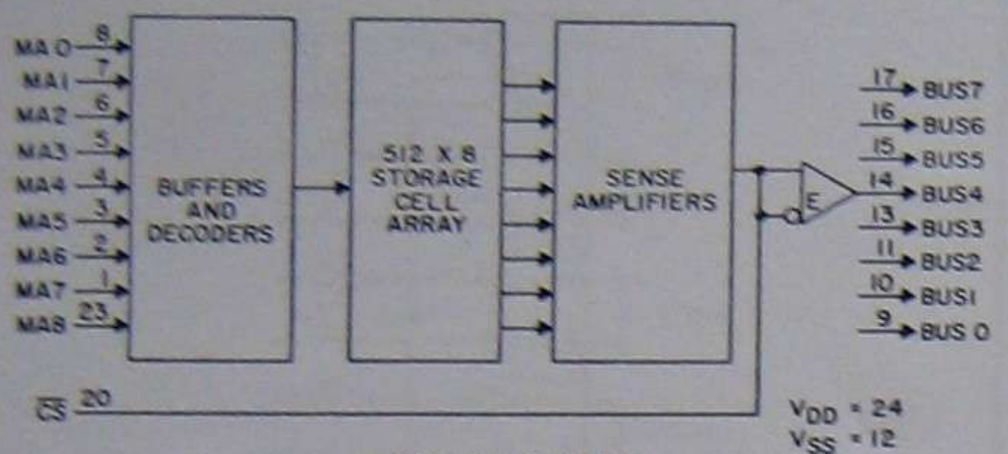
CDP1824D (Prel.)
CDP1824CD (Prel.)



512-Word by 8-Bit Static Read-Only Memory

CDP1831D (Prel.)
CDP1831CD (Prel.)

Functional Diagrams

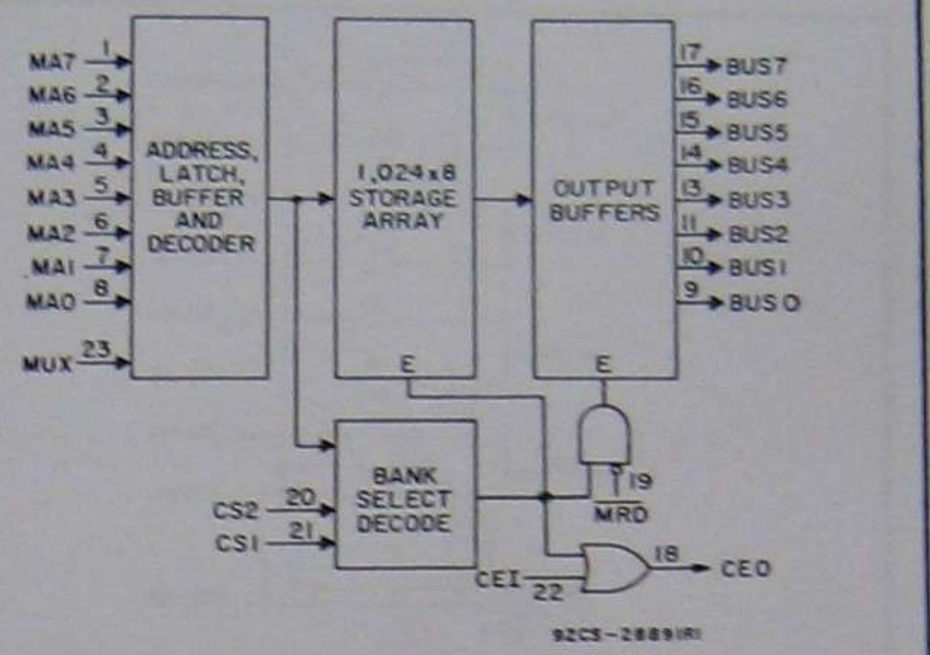


FUNCTIONAL DIAGRAM

92CS-27581

512-Word by 8-Bit Static Read-Only Memory

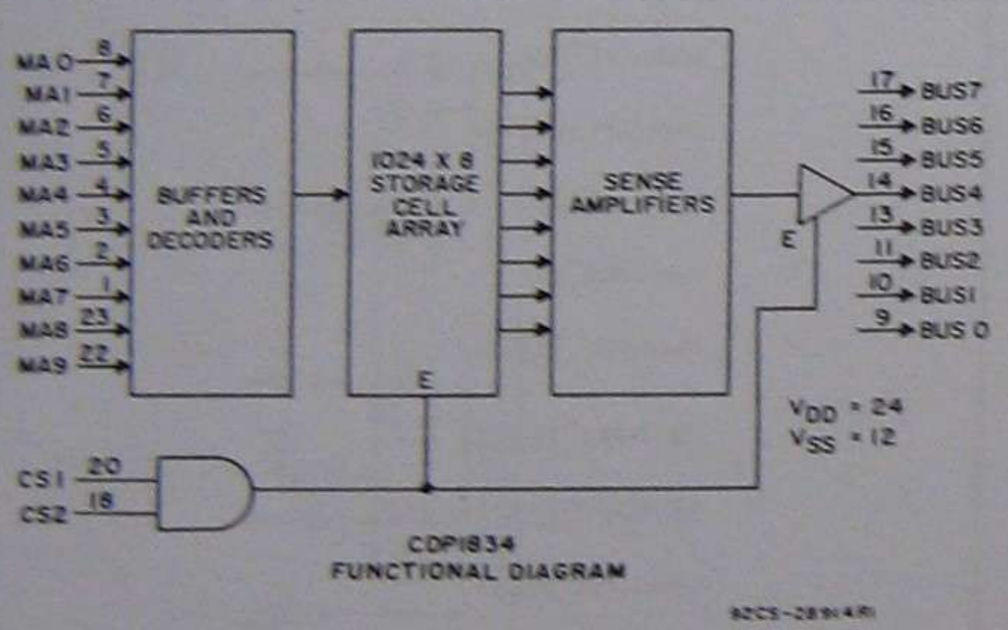
CDP1832D (Prel.)
CDP1832CD (Prel.)



92CS-2889(R)

1024-Word by 8-Bit Static Read-Only Memory

CDP1833C (Prel.)
CDP1833CD (Prel.)

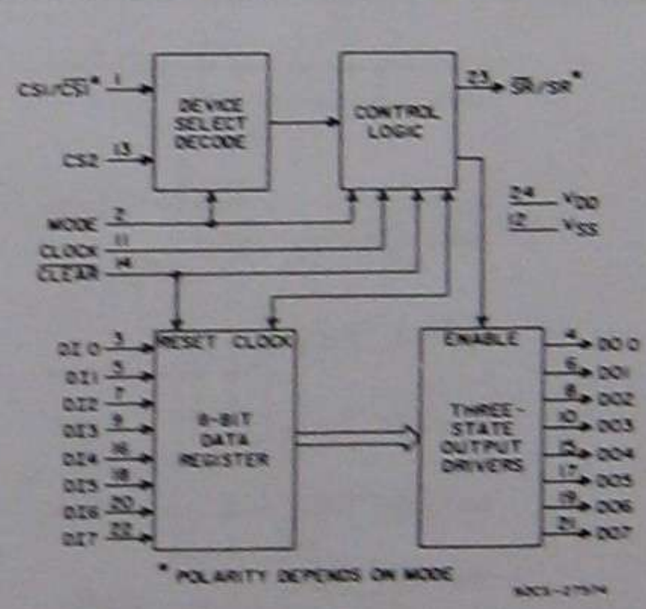


CDP1834
FUNCTIONAL DIAGRAM

92CS-28914R1

1024-Word by 8-Bit Static Read-Only Memory

CDP1834D (Prel.)
CDP1834CD (Prel.)

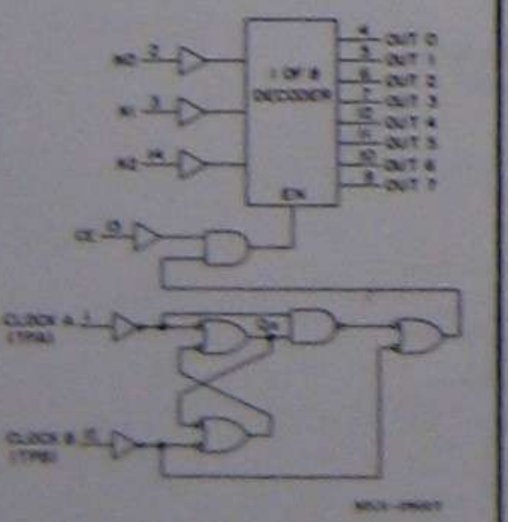


* POLARITY DEPENDS ON MODE

92CS-27574

8-Bit Input/Output Port

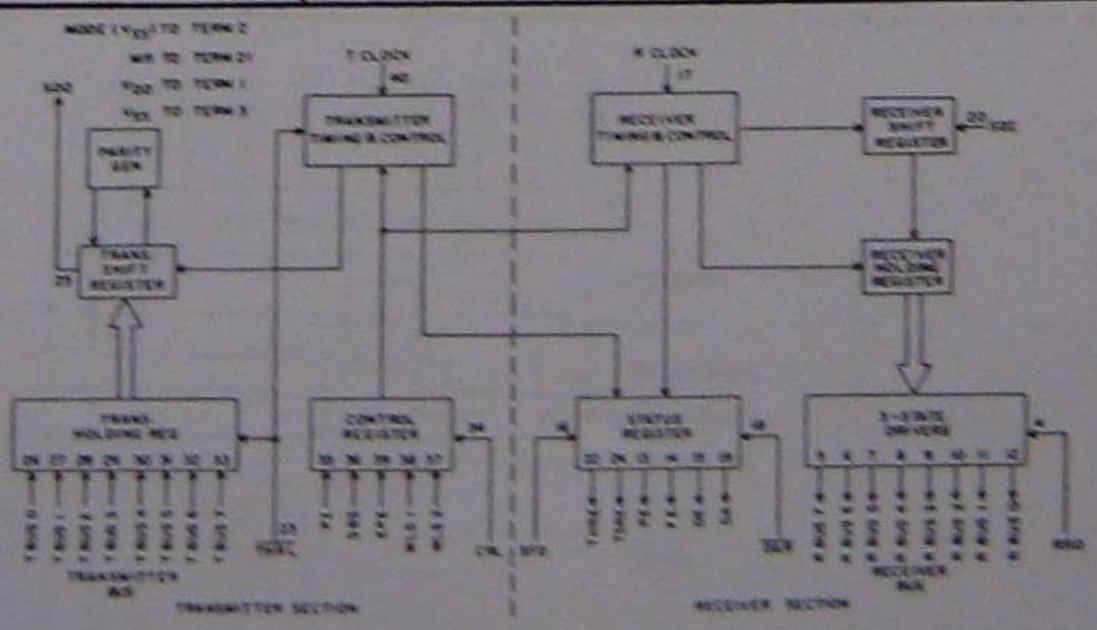
CDP1852D (Prel.)
CDP1852CD (Prel.)



92CS-29001

N-Bit 1 of 8 Decoder

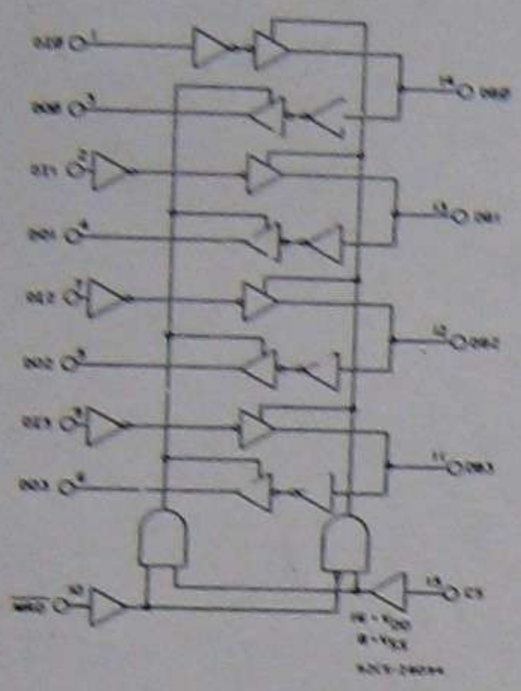
CDP1853D (Prel.)
CDP1853CD (Prel.)



Universal Asynchronous Receiver/ Transmitter (UART) (Standard Mode 0 Shown)

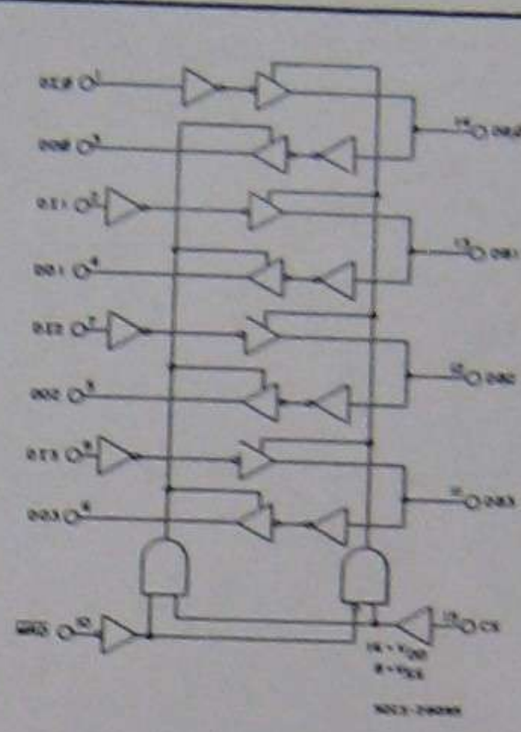
CDP1854C (Prel.)
CDP1854CD (Prel.)

Functional Diagrams



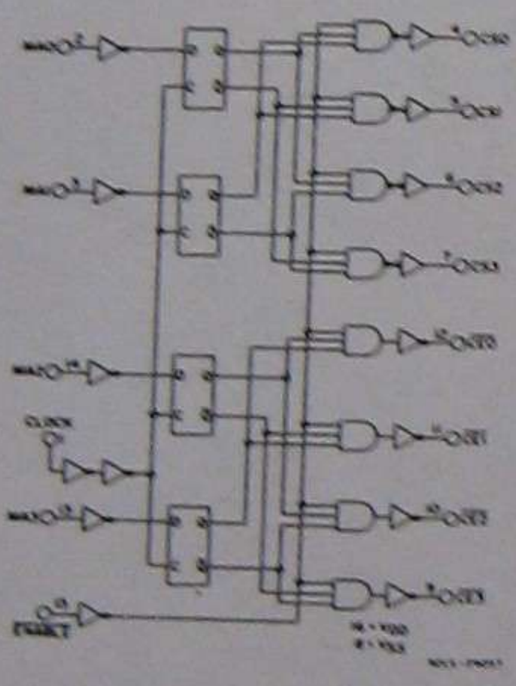
4-Bit Bus Buffer/Separator

CDP1856D (Prel.)
CDP1856CD (Prel.)



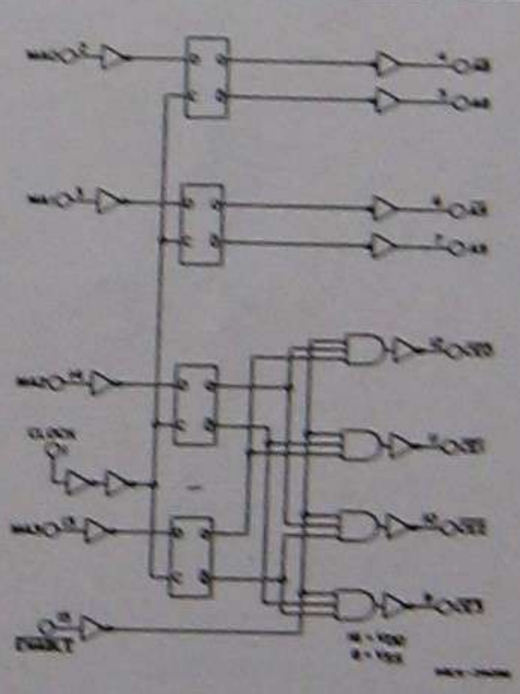
4-Bit Bus Buffer/Separator

CDP1857D (Prel.)
CDP1857CD (Prel.)



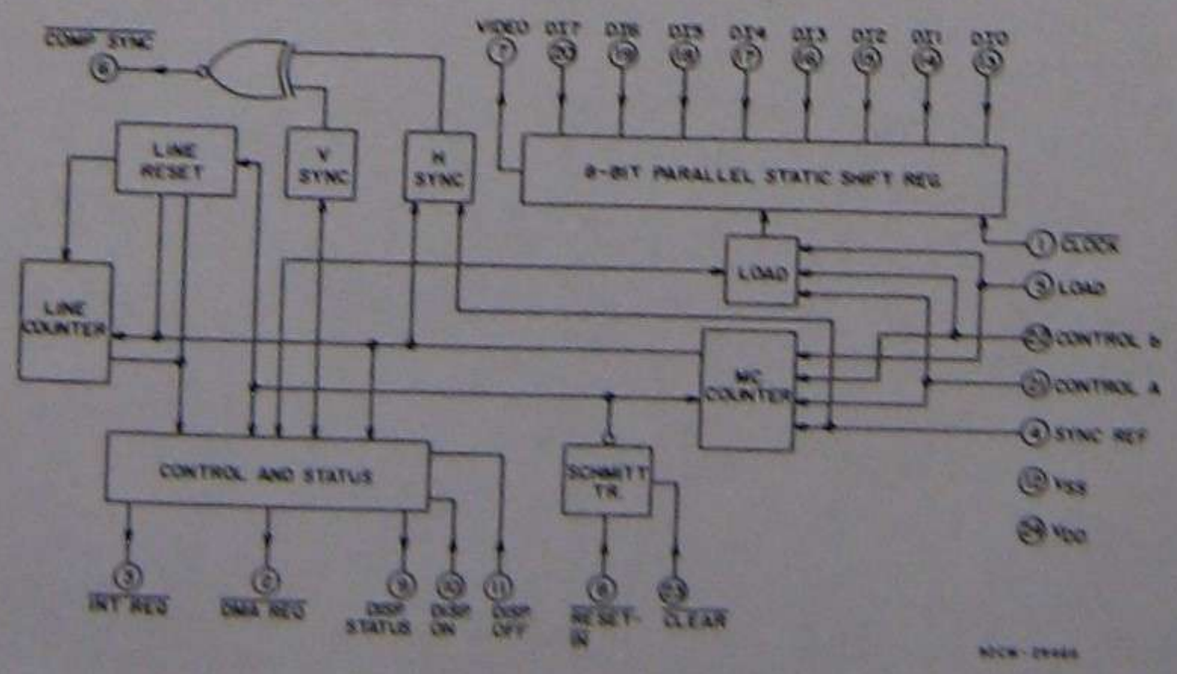
4-Bit Latch with Decode

CDP1858C (Prel.)
CDP1858CD (Prel.)



4-Bit Latch with Decode

CDP1859D (Prel.)
CDP1859CD (Prel.)



Video Display Controller

CDP1861CD (Prel.)

/

4001 4×2I/P NOR

4069 Hex Inv

4011 4×2I/P NAND

4023 Triple 3 I/P NAND

4025 " " NOR

4071 4×2 I/P OR

4082 4×2 I/P AND
81

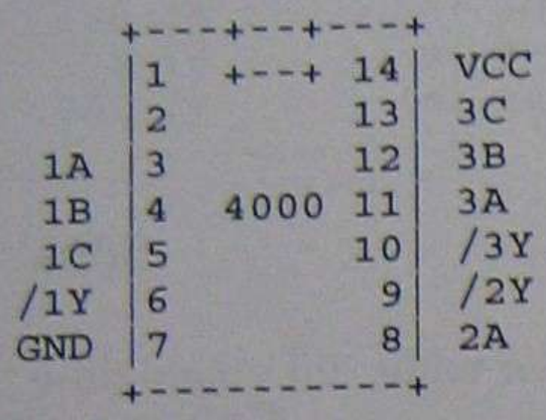
4070 Ex-OR

4000 series CMOS IC's: 4000...4049

The old but still much-used 4000 series logic IC's.

4000

Dual 3-input NOR gates.



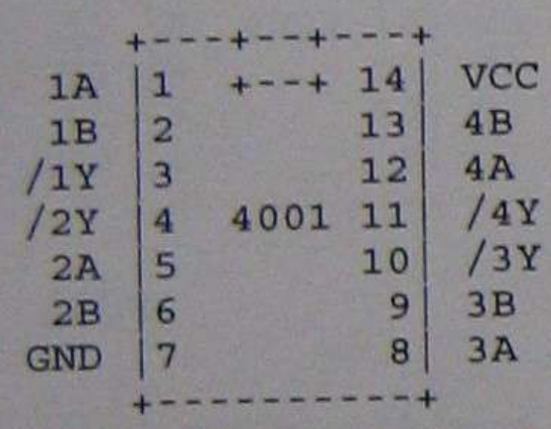
$/1Y=\overline{1A+1B+1C}$

$/2Y=\overline{2A}$

$/3Y=\overline{3A+3B+3C}$

4001

Quad 2-input NOR gates.

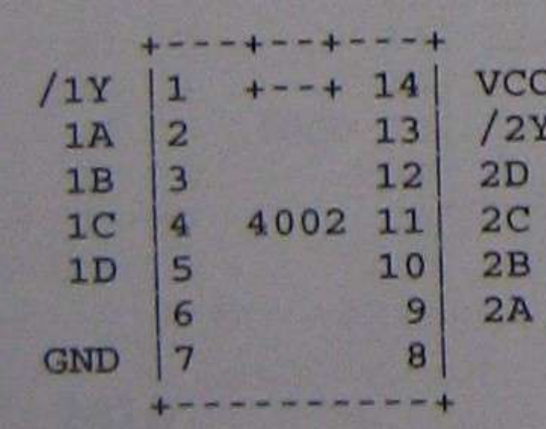


A	B	/Y
0	0	1
0	1	0
1	0	0
1	1	0

$/Y = \overline{A+B}$

4002

Dual 4-input NOR gates.

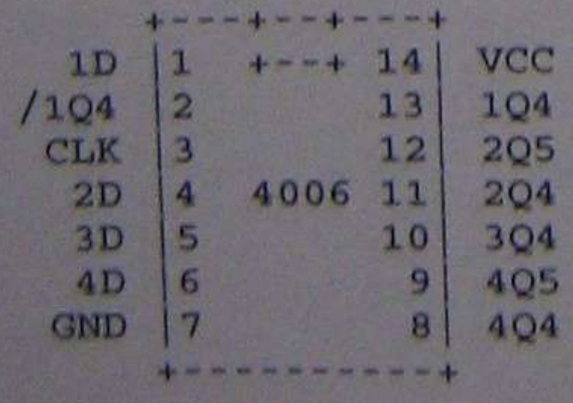


A	B	C	D	/Y
0	0	0	0	1
0	0	0	1	0
0	0	1	X	0
0	1	X	X	0
1	X	X	X	0

$/Y = \overline{(A+B+C+D)}$

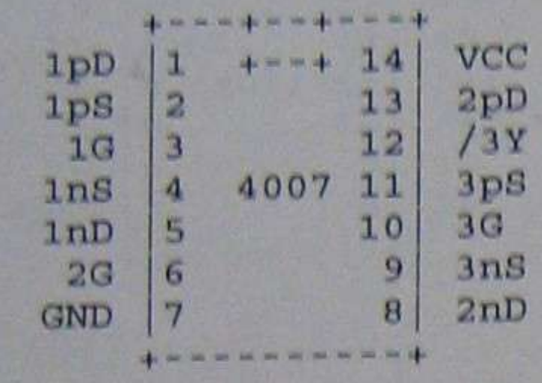
4006

Dual 4-bit and dual 5-bit serial-in serial-out shift registers with common clock.



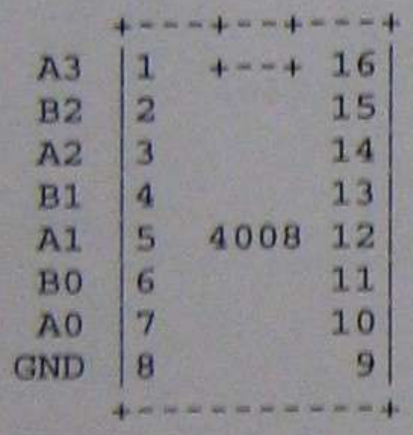
4007

Dual complementary CMOS pair and unbuffered inverter.
For use as simple inverters, connect 1pS=3pS=VCC, 1nS=3nS=GND, 1pD=1nD=/1Y and 2pD=2nD=/2Y.



4008

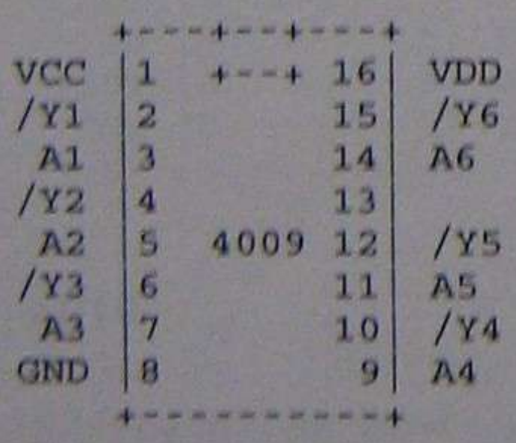
4-bit binary full adder with fast carry.



S=A+B+CIN

4009

Hex inverters with level shifted outputs.
VDD may not be lower than VCC.

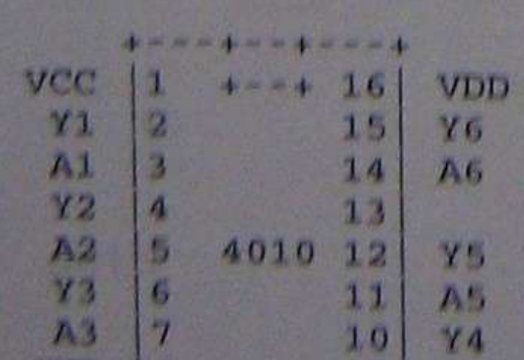


A	/Y
0	1
1	0

$/Y = \overline{A}$

4010

Hex buffers with level shifted outputs.
VDD may not be lower than VCC.

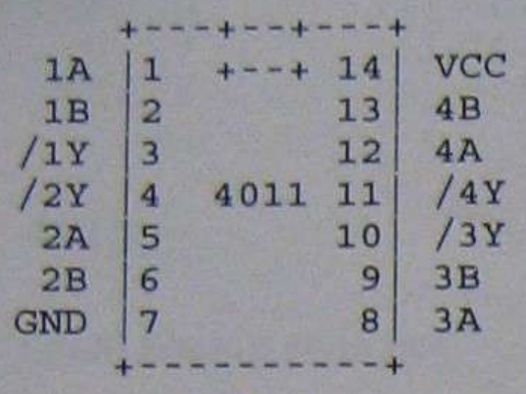


A	Y
0	0
1	1

Y = A

4011

Quad 2-input NAND gates.

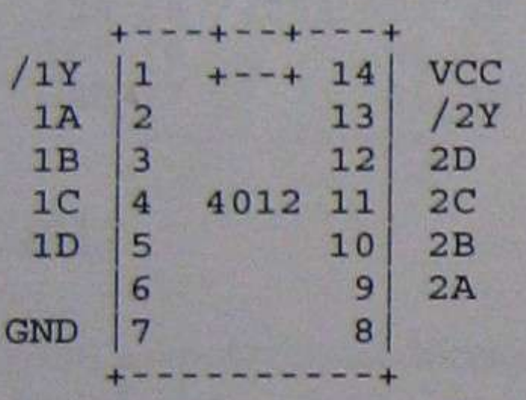


A	B	/Y
0	0	1
0	1	1
1	0	1
1	1	0

$/Y = \overline{AB}$

4012

Dual 4-input NAND gates.

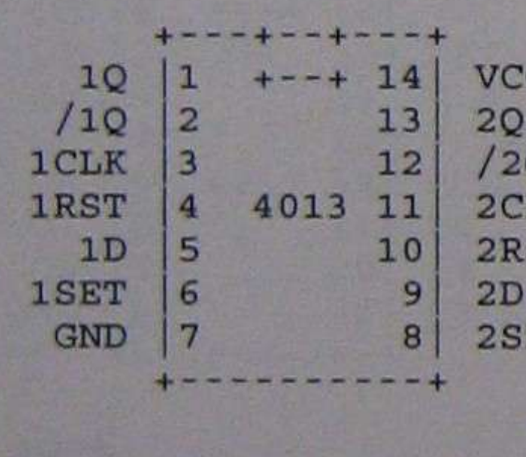


A	B	C	D	/Y
0	X	X	X	1
1	0	X	X	1
1	1	0	X	1
1	1	1	0	1
1	1	1	1	0

$/Y = \overline{ABCD}$

4013

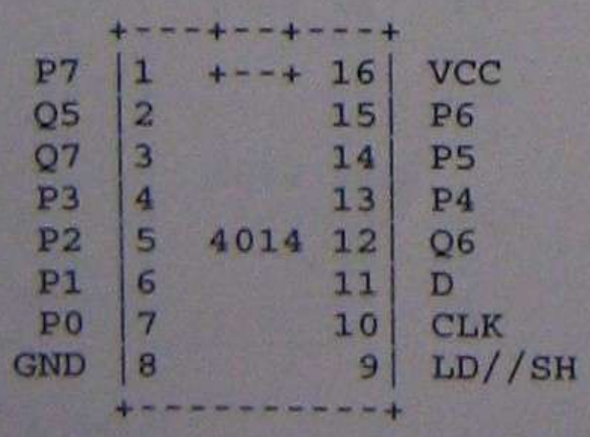
Dual D flip-flop with set and reset.



D	CLK	SET	RST	Q	/Q
X	X	0	1	0	1
X	X	1	0	1	0
X	X	1	1	1	1
0	/	0	0	0	1
1	/	0	0	1	1
X	!/	0	0	-	-

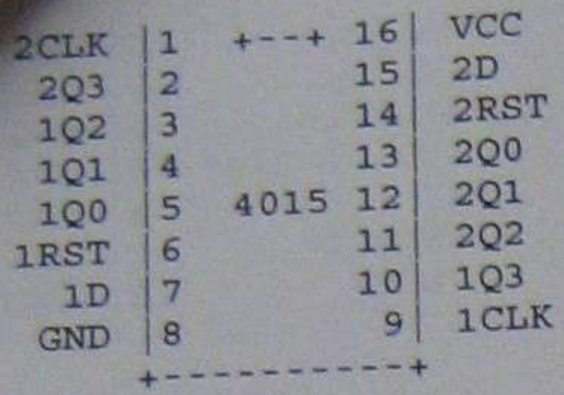
4014

8-bit parallel-in serial-out shift register with three parallel outputs.



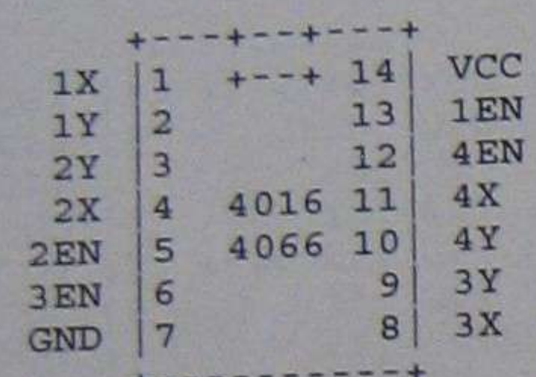
4015

Dual 4-bit serial-in parallel-out shift register with asynchronous reset.



4016

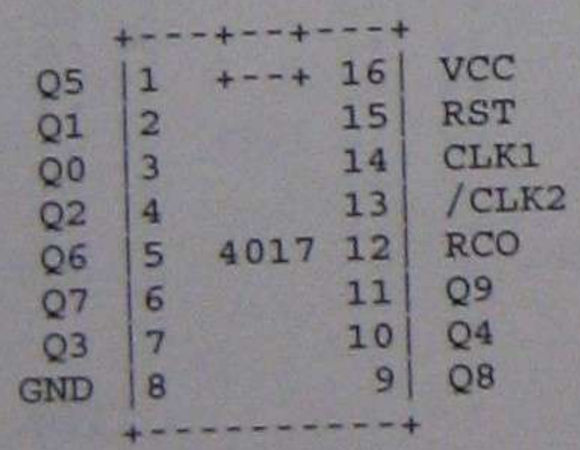
Quad analog switches.



4017

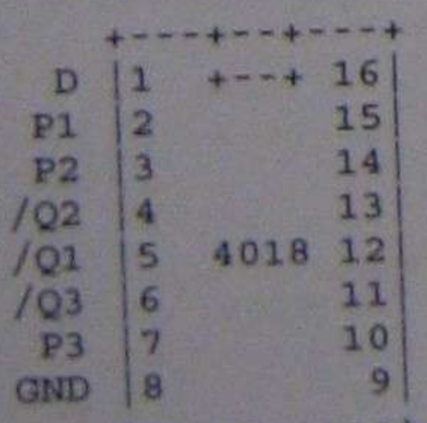
4-bit asynchronous decade counter with fully decoded outputs, reset and both active high and active low clocks.

The two CLK inputs are ANDed together, so that either can be used as clock or clock enable.



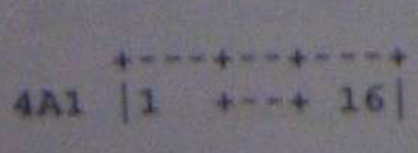
4018

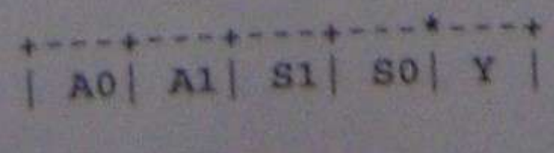
5-stage (divide by 2,4,6,8 or 10) Johnson counter with preset inputs.



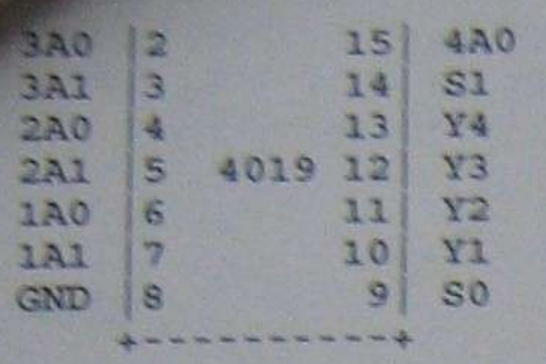
4019

8-to-4 line noninverting data selector/multiplexer with OR function.





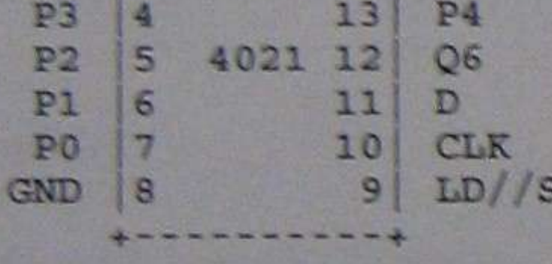
Y=S0.A0+S1.A1



X	X	0	0	0
X	0	0	1	0
0	X	1	0	0
X	1	X	1	1
1	X	1	X	1

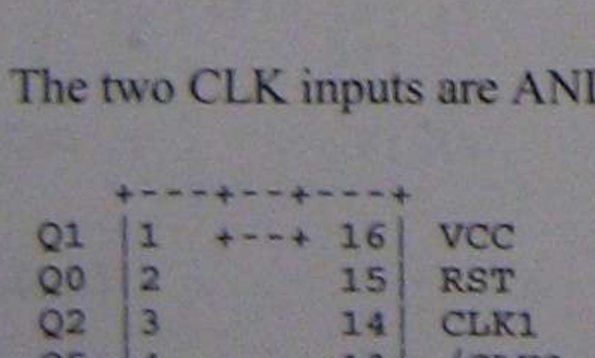
4020

14-bit asynchronous binary counter with reset.
Q1 and Q2 outputs missing.



4021

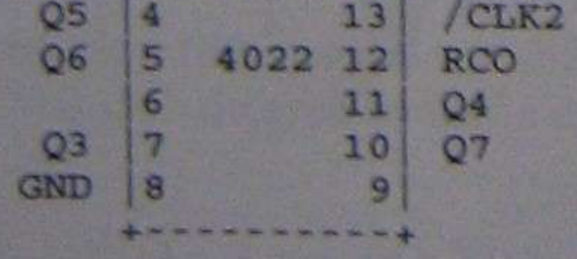
8-bit parallel-in serial-out shift register with asynchronous load input and three parallel outputs.



4022

3-bit asynchronous binary counter with fully decoded outputs, reset and both active high and active low clocks.

The two CLK inputs are ANDed together, so that either can be used as clock or clock enable.



4023

Triple 3-input NAND gates.

```
1SET |7          10| 2J
 GND |8           9| 2SET
     +--------------+
```

0	1	/	0	0	0	1
1	0	/	0	0	1	0
1	1	/	0	0	/Q	Q
X	X	!/	0	0	-	-

4028

1-of-10 noninverting decoder/demultiplexer.

```
     +---+--+---+
 Y4 |1  +--+ 16| VCC
 Y2 |2       15| Y3
 Y0 |3       14| Y1
 Y7 |4       13| S1
 Y9 |5  4028 12| S2
 Y5 |6       11| S3
 Y6 |7       10| S0
GND |8        9| Y8
    +----------+
```

S3	S2	S1	S0	Y0	Y1	...	Y9
0	0	0	0	1	0	0	0
0	0	0	1	0	1	0	0
.	.	.	.	0	0	.	0
1	0	0	1	0	0	0	1
1	0	1	X	0	0	0	0
1	1	X	X	0	0	0	0

4029

4-bit synchronous binary/decade up/down counter with preset and ripple carry output.

```
     +---+--+---+
  PE |1  +--+ 16| VCC
  Q4 |2       15| CLK
  P4 |3       14| Q3
  P1 |4       13| P3
/RCI |5  4029 12| P2
  Q1 |6       11| Q2
/RCO |7       10| U//D
 GND |8        9| B//D
     +----------+
```

4030

Quad 2-input XOR gates.

```
     +---+--+---+
 1A |1  +--+ 14| VCC
 1B |2       13| 4B
 1Y |3       12| 4A
 2Y |4  4030 11| 4Y
 2A |5       10| 3Y
 2B |6        9| 3B
GND |7        8| 3A
    +----------+
```

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

$$Y = A \$ B = (A.\overline{B}) + (\overline{A}.B)$$

4031

64-bit serial-in serial-out shift register with multiplexed inputs.
Y is Q63 delayed by half a cycle (i.e. clocked on falling edge).

```
     +---+--+---+
   E |1  +--+ 16| VCC
 CLK |2       15| D
     |3       14|
     |4       13|
   Y |5  4031 12|
 Q63 |6       11|
/Q63 |7       10| E//D
 GND |8        9| CLKout
     +----------+
```

```
 1A |1   +--+ 14| VCC
 1B |2       13| 3C
 2A |3       12| 3B
 2B |4  4023 11| 3A
 2C |5       10| /3Y
/2Y |6        9| /1Y
GND |7        8| 1C
    +----------+
```

A	B	C	/Y
0	X	X	1
1	0	X	1
1	1	0	1
1	1	1	0

/Y = ABC

4024

7-bit asynchronous binary counter with reset.

```
    +---+--+---+
/CLK |1  +--+ 14| VCC
 RST |2       13|
  Q6 |3       12| Q0
  Q5 |4  4024 11| Q1
  Q4 |5       10|
  Q3 |6        9| Q2
 GND |7        8|
    +----------+
```

4025

Triple 3-input NOR gates.

```
    +---+--+---+
 1A |1  +--+ 14| VCC
 1B |2       13| 3C
 2A |3       12| 3B
 2B |4  4025 11| 3A
 2C |5       10| /3Y
/2Y |6        9| /1Y
GND |7        8| 1C
    +----------+
```

A	B	C	/Y
0	0	0	1
0	0	1	0
0	1	X	0
1	X	X	0

$/Y = \overline{A+B+C}$

4026

4-bit asynchronous decade counter with 7-segment decoder/common-cathode LED driver, display enable, ripple carry, reset and both active high and active low clocks.

```
    +---+--+---+
CLK1 |1  +--+ 16| VCC
/CLK2|2       15| RST
 DEI |3       14| YC'
 DEO |4       13| YC
  CO |5  4026 12| YB
  YF |6       11| YE
  YG |7       10| YA
 GND |8        9| YD
    +----------+
```

4027

Dual J-K flip-flops with set and reset.

```
    +---+--+---+
  1Q |1  +--+ 16| VCC
 /1Q |2       15| 2Q
1CLK |3       14| /2Q
1RST |4       13| 2CLK
  1K |5  4027 12| 2RST
  1J |6       11| 2K
    +----------+
```

J	K	CLK	SET	RST	Q	/Q
X	X	X	1	1	1	1
X	X	X	1	0	1	0
X	X	X	0	1	0	1
0	0	/	0	0	-	-

```
 /INV |2        15| Q1
   /K |3        14| Q2
    J |4        13| Q3
  RST |5  4035  12| P3
  CLK |6        11| P2
LD//SH|7        10| P1
  GND |8         9| P0
      +----------+
```

4038

Triple negative-edge-triggered serial adder.
Each section can be used to add long binary words, one bit on each clock cycle. CRST resets the internal carry flip-flop after one clock delay. The INV inputs can be used to invert the sum output (giving a 1's-complemented result).

```
     +---+--+---+
   3S |1  +--+ 16| VCC
3INV  |2       15| 3A
/CLK  |3       14| 3B
   2S |4       13| 2A
2INV  |5  4038 12| 2B
CRST  |6       11| 1B
1INV  |7       10| 1A
  GND |8        9| 1S
      +----------+
```

4040

12-bit asynchronous binary counter with reset.

```
     +---+--+---+
Q11  |1  +--+ 16| VCC
 Q5  |2       15| Q10
 Q4  |3       14| Q9
 Q6  |4       13| Q7
 Q3  |5  4040 12| Q8
 Q2  |6       11| RST
 Q1  |7       10| /CLK
GND  |8        9| Q0
     +----------+
```

4041

Quad buffers with complementary outputs.

```
     +---+--+---+
 1Y  |1  +--+ 14| VCC
/1Y  |2       13| 4A
 1A  |3       12| /4Y
 2Y  |4  4041 11| 4Y
/2Y  |5       10| 3A
 2A  |6        9| /3Y
GND  |7        8| 3Y
     +----------+
```

```
+---*---+---+
| A | Y |/Y |
+===*===+===+
| 0 | 0 | 1 |
| 1 | 1 | 0 |
+---*---+---+
```

Y = A

4042

4-bit transparent latch with selectable latch enable polarity and complementary outputs.

```
     +---+--+---+
 Q3  |1  +--+ 16| VCC
 Q0  |2       15| /Q3
     +----------+
```

```
+---+---+---*---+---+
| LE| LP| D | Q |/Q |
+===+===+===*===+===+
```

```
+----------+
```

4032

Triple serial adder.
Each section can be used to add long binary words, one bit on each clock cycle. CRST resets the internal carry flip-flop after one clock delay. The INV inputs can be used to invert the sum output (giving a 1's-complemented result).

```
      +---+--+---+
   3S |1  +--+ 16| VCC
 3INV |2       15| 3A
  CLK |3       14| 3B
   2S |4       13| 2A
 2INV |5  4032 12| 2B
 CRST |6       11| 1B
 1INV |7       10| 1A
  GND |8        9| 1S
      +----------+
```

4033

4-bit asynchronous decade counter with 7-segment decoder/common-cathode LED driver, ripple blanking, ripple carry, reset and both active high and active low clocks.

```
      +---+--+---+
  CLK1 |1  +--+ 16| VCC
 /CLK2 |2       15| RST
   RBI |3       14| LT
   RBO |4       13| YC
    CO |5  4033 12| YB
    YF |6       11| YE
    YG |7       10| YA
   GND |8        9| YD
       +----------+
```

4034

8-bit bidirectional universal shift register with common serial input, dual parallel I/O ports and selectable synchronous/asynchronous parallel load.

```
      +-----+--+-----+
    A7 |1    +--+   24| VCC
    A6 |2            23| B7
    A5 |3            22| B6
    A4 |4            21| B5
    A3 |5            20| B4
    A2 |6            19| B3
    A1 |7    4034    18| B2
    A0 |8            17| B1
   ENA |9            16| B0
     D |10           15| CLK
 B//A  |11           14| ASY//SY
   GND |12           13| LD//SH
       +---------------+
```

B//A	LD//SH	A0..7	B0..7
0	1	in	out
1	1	out	in
0	0	Z	out
1	0	out	Z

4035

4-bit inverting/noninverting universal shift register with J-/K inputs and asynchronous reset.

```
      +---+--+---+
   Q0 |1  +--+ 16| VCC
```

```
/Q0 |3        14| D3
D0  |4        13| D2
LE  |5  4042 12| /Q2
LP  |6        11| Q2
D2  |7        10| Q1
GND |8         9| /Q1
    +----------+
```

0	0	0	0	1
0	0	1	1	0
1	0	X	-	-
1	1	0	0	1
1	1	1	1	0
0	1	X	-	-

4043

Quad 3-state S-R latches with overriding set.

```
    +---+--+---+
1Q  |1  +--+ 16| VCC
2Q  |2       15| 1R
2R  |3       14| 1S
2S  |4       13|
OE  |5  4043 12| 4S
3S  |6       11| 4R
3R  |7       10| 4Q
GND |8        9| 3Q
    +----------+
```

S	R	OE	Q
X	X	0	Z
0	0	1	-
0	1	1	1
1	0	1	0
1	1	1	1

4044

Quad 3-state S-R latches with overriding reset.

```
    +---+--+---+
1Q  |1  +--+ 16| VCC
    |2       15| 4S
2S  |3       14| 4R
2R  |4       13| 2Q
OE  |5  4044 12| 4R
3S  |6       11| 4S
3R  |7       10| 4Q
GND |8        9| 3Q
    +----------+
```

S	R	OE	Q
X	X	0	Z
0	0	1	-
0	1	1	1
1	0	1	0
1	1	1	0

4045

21-bit asynchronous binary counter with oscillator and reset input.
Only two 3% duty cycle outputs (180` out of phase) from the last counter stage are available. Can be used to generate a 1Hz clock signal using a 2.097152MHz crystal. P and N MOSFET source connections from the oscillator inverter are brought out of the package to allow the use of source resistors, but usually pS=VCC and nS=GND.

```
    +---+--+---+
pS  |1  +--+ 16| X1
nS  |2       15| X0
VCC |3       14| GND
    |4       13|
    |5  4045 12|
    |6       11|
QA  |7       10|
QB  |8        9|
    +----------+
```

4046

Phase Locked Loop.

```
      +---+--+---+
PCPout|1  +--+ 16| VCC
```

```
PC1out |2        15| Zener
 PCinB |3        14| PCinA
VCOout |4        13| PC2out
   /EN |5  4046  12| R2
   C1A |6        11| R1
   C1B |7        10| SFout
   GND |8         9| VCOin
       +----------+
```

4047

Low-power astable/monostable multivibrator with oscillator output.

```
       +---+--+---+
 Cext  |1  +--+ 14|  VCC
 Rext  |2       13|  OSC
RCext  |3       12|  RETRIG
 /AST  |4  4047 11|  /Q
  AST  |5       10|  Q
  /TR  |6        9|  RST
  GND  |7        8|  TR
       +----------+
```

4048

3-state 8-input multifunction gate.

```
       +---+--+---+
    Y  |1  +--+ 16|  VCC
   OE  |2       15|  X
    A  |3       14|  H
    B  |4       13|  G
    C  |5  4048 12|  F
    D  |6       11|  E
   S1  |7       10|  S2
  GND  |8        9|  S0
       +----------+
```

S2	S1	S0	OE	Output function
X	X	X	0	Z
0	0	0	1	8-input NOR
0	0	1	1	8-input OR
0	1	0	1	2-wide 4-input OR-AND
0	1	1	1	2-wide 4-input OR-NAND
1	0	0	1	8-input AND
1	0	1	1	8-input NAND
1	1	0	1	2-wide 4-input AND-NOR
1	1	1	1	2-wide 4-input AND-OR

4049

Hex inverters with high-to-low level shifter inputs.

```
       +---+--+---+
  VCC  |1  +--+ 16|
  /Y1  |2       15|  /Y6
   A1  |3       14|  A6
  /Y2  |4       13|
   A2  |5  4049 12|  /Y5
  /Y3  |6       11|  A5
   A3  |7       10|  /Y4
  GND  |8        9|  A4
       +----------+
```

A	/Y
0	1
1	0

$/Y = \overline{A}$

4066

Quad analog switches.

```
       +---+--+---+
   1X  |1  +--+ 14|  VCC
   1Y  |2       13|  1EN
   2Y  |3       12|  4EN
```

```
PC1out |2        15| Zener
 PCinB |3        14| PCinA
VCOout |4        13| PC2out
   /EN |5  4046  12| R2
   C1A |6        11| R1
   C1B |7        10| SFout
   GND |8         9| VCOin
       +----------+
```

4047

Low-power astable/monostable multivibrator with oscillator output.

```
       +---+--+---+
 Cext  |1  +--+ 14|  VCC
 Rext  |2       13|  OSC
RCext  |3       12|  RETRIG
 /AST  |4  4047 11|  /Q
  AST  |5       10|  Q
  /TR  |6        9|  RST
  GND  |7        8|  TR
       +----------+
```

4048

3-state 8-input multifunction gate.

```
       +---+--+---+
    Y  |1  +--+ 16|  VCC
   OE  |2       15|  X
    A  |3       14|  H
    B  |4       13|  G
    C  |5  4048 12|  F
    D  |6       11|  E
   S1  |7       10|  S2
  GND  |8        9|  S0
       +----------+
```

S2	S1	S0	OE	Output function
X	X	X	0	Z
0	0	0	1	8-input NOR
0	0	1	1	8-input OR
0	1	0	1	2-wide 4-input OR-AND
0	1	1	1	2-wide 4-input OR-NAND
1	0	0	1	8-input AND
1	0	1	1	8-input NAND
1	1	0	1	2-wide 4-input AND-NOR
1	1	1	1	2-wide 4-input AND-OR

4049

Hex inverters with high-to-low level shifter inputs.

```
       +---+--+---+
  VCC  |1  +--+ 16|
  /Y1  |2       15|  /Y6
   A1  |3       14|  A6
  /Y2  |4       13|
   A2  |5  4049 12|  /Y5
  /Y3  |6       11|  A5
   A3  |7       10|  /Y4
  GND  |8        9|  A4
       +----------+
```

A	/Y
0	1
1	0

$/Y = \overline{A}$

4066

Quad analog switches.

```
       +---+--+---+
   1X  |1  +--+ 14|  VCC
   1Y  |2       13|  1EN
   2Y  |3       12|  4EN
```

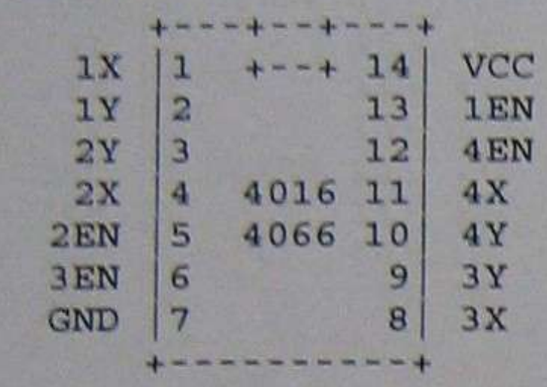
Back to the GBCM index.

4000 series CMOS IC's: 4050...4099

The old but still much-used 4000 series logic IC's.

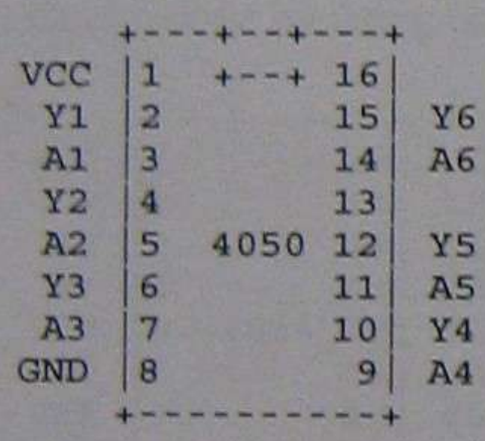
4016

Quad analog switches.



4050

Hex buffers with high-to-low level shifter inputs.

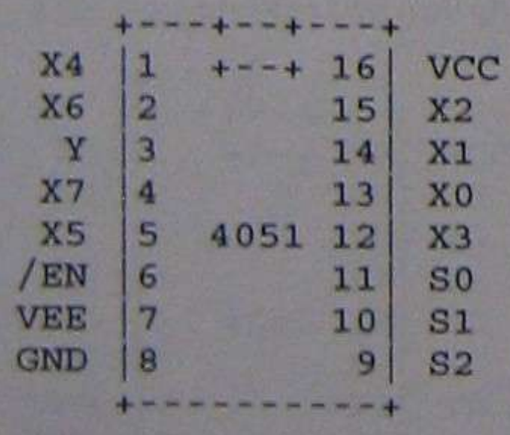


A	Y
0	0
1	1

Y = A

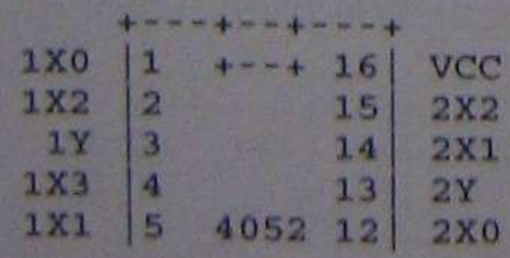
4051

8-to-1 line analog multiplexer/demultiplexer with dual power supply.
VEE supply may not be more positive than GND.



4052

8-to-2 line analog multiplexer/demultiplexer with dual power supply.
VEE supply may not be more positive than GND.



```
/EN |6          11| 2X3
VEE |7          10| S0
GND |8           9| S1
    +--------------+
```

4053

Triple 2-to-1 line analog multiplexer/demultiplexer with dual power supply.
VEE supply may not be more positive than GND.

```
    +---+--+---+
1X0 |1  +--+ 16| VCC
1X1 |2       15| 1Y
2X1 |3       14| 3Y
 2Y |4       13| 3X1
2X0 |5  4053 12| 3X0
/EN |6       11| 3S
VEE |7       10| 1S
GND |8        9| 2S
    +----------+
```

4054

Quad level shifters/LCD drivers with input latches.
A level-shifted inverse of the P (phase) input should be connected to the backplane of the LCD; this can be done by using one section of the 4054 with A=0 and LE=1.

```
    +---+--+---+
1LE |1  +--+ 16| VCC
  P |2       15| 1A
 1Y |3       14| 2LE
 2Y |4       13| 2A
 3Y |5  4054 12| 3LE
 4Y |6       11| 3A
VEE |7       10| 4LE
GND |8        9| 4A
    +----------+
```

LE	A	R
0	X	-
1	0	0
1	1	1

$Y = R\bar{\$}P$

4055

BCD to 7-segment decoder/LCD driver.
The Po (phase) output should be connected to the backplane of the LCD.

```
    +---+--+---+
 Po |1  +--+ 16| VCC
 A2 |2       15| YF
 A1 |3       14| YG
 A3 |4       13| YE
 A0 |5  4055 12| YD
 Pi |6       11| YC
VEE |7       10| YB
GND |8        9| YA
    +----------+
```

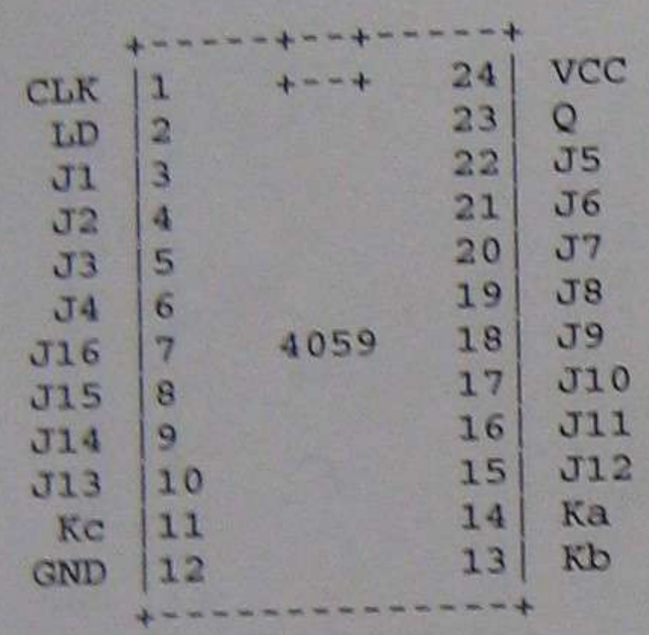
4056

BCD to 7-segment decoder/LCD driver with input latches.
A level-shifted inverse of the P (phase) input should be connected to the backplane of the LCD.

```
    +---+--+---+
 LE |1  +--+ 16| VCC
 A2 |2       15| YF
```

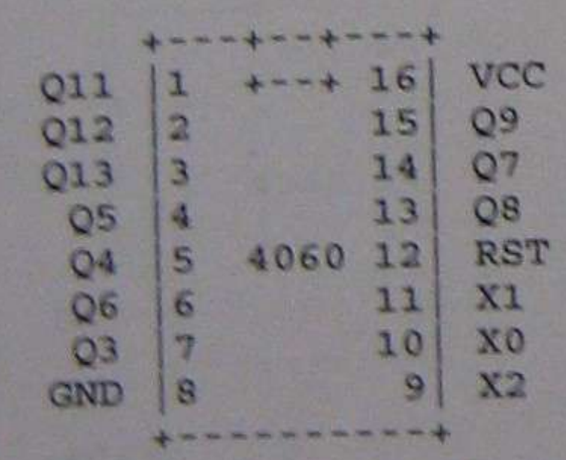
4059

Divide by N counter.
Ka, Kb, Kc are the modulus (divide by number) of the 1st and last counting sections. N can range from 3 to 15999. The down-counter is preset by 15 jam inputs.



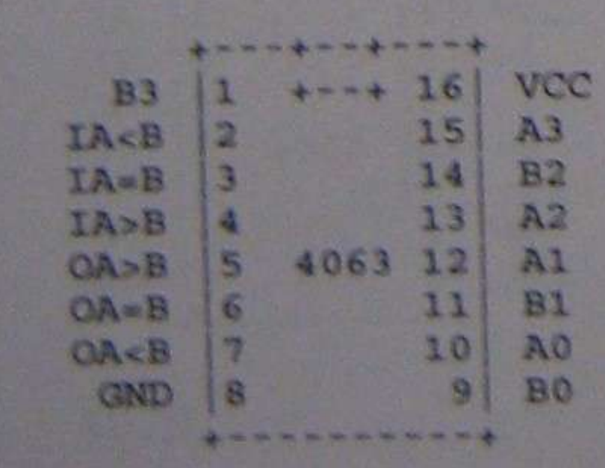
4060

14-bit asynchronous binary counter with oscillator and reset input.
Q0,Q1,Q2 and Q10 outputs are missing.



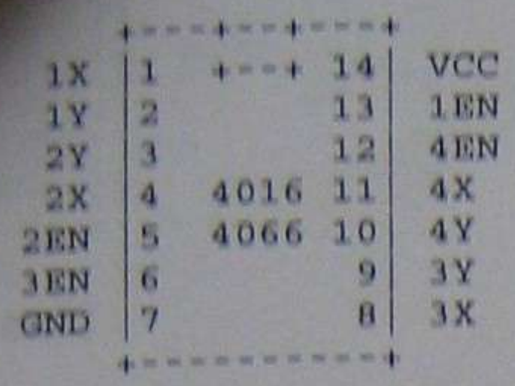
4063

4-bit noninverting magnitude comparator with cascade inputs.



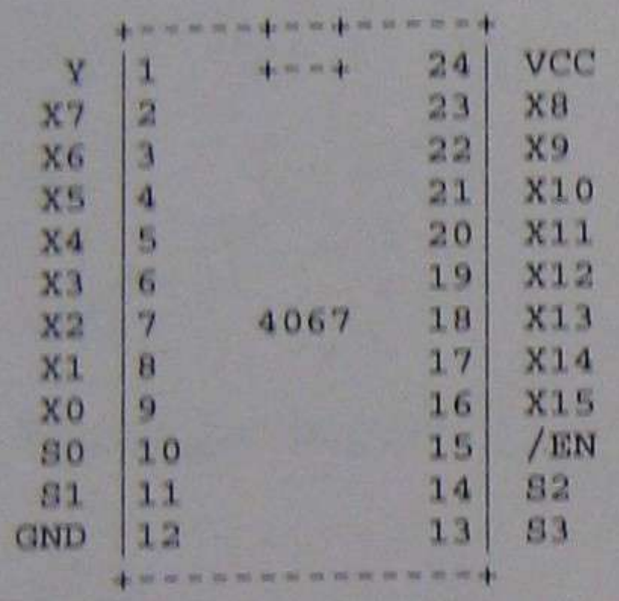
4066

Quad analog switches.



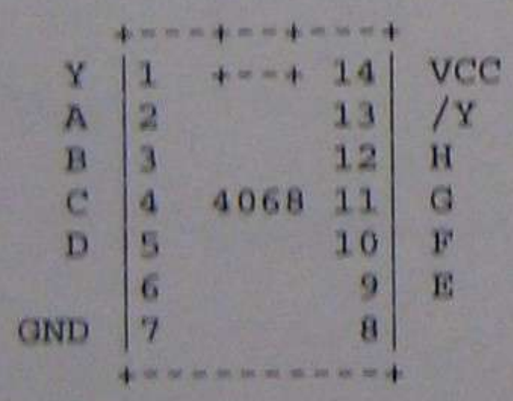
4067

16-to-1 line analog multiplexer/demultiplexer.



4068

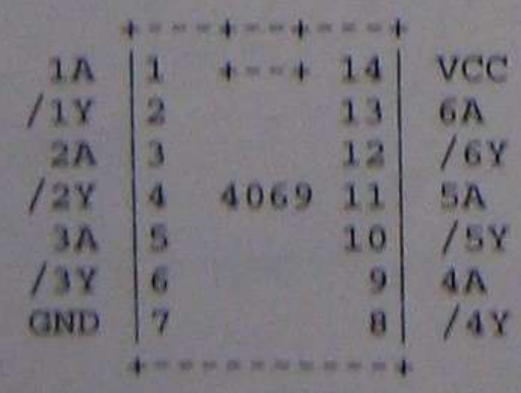
8-input AND/NAND gate with complementary outputs.



Y = ABCDEFGH

4069

Hex inverters.

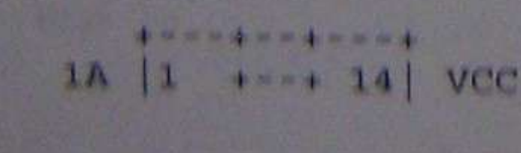


A	/Y
0	1
1	0

$\overline{Y} = \overline{A}$

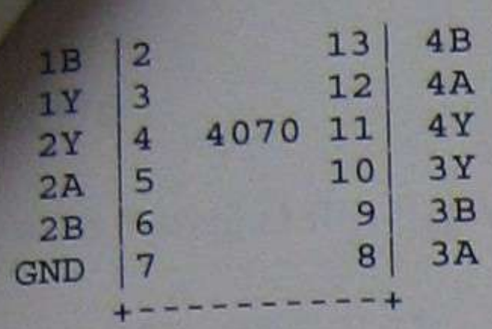
4070

Quad 2-input XOR gates.



A	B	Y

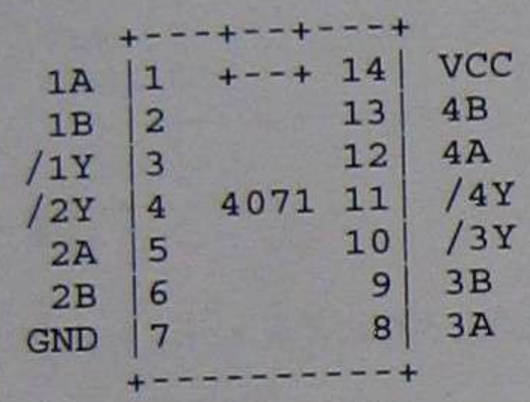
$Y = A \oplus B = (A . \overline{B}) + (\overline{A} . B)$



0	0	0
0	1	1
1	0	1
1	1	0

4071

Quad 2-input OR gates.

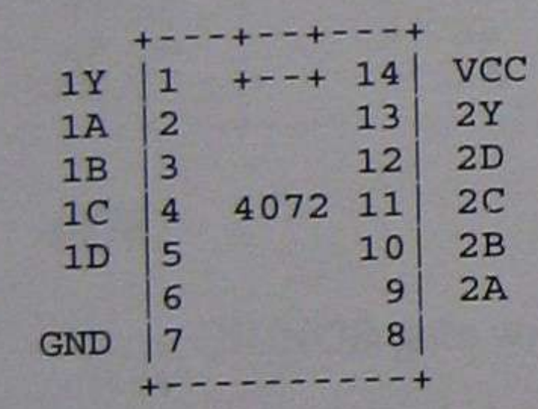


A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

Y = A+B

4072

Dual 4-input OR gates.

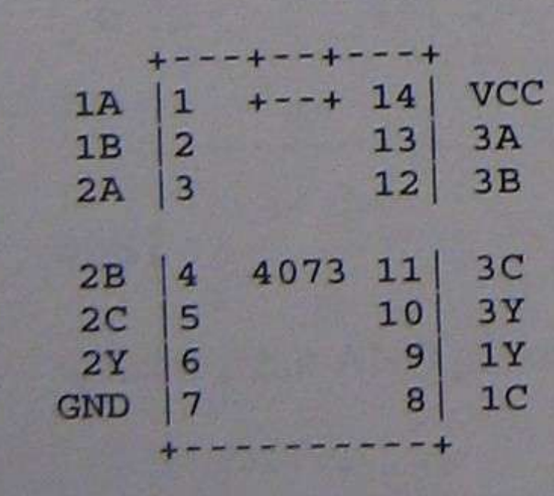


A	B	C	D	/Y
0	0	0	0	0
0	0	0	1	1
0	0	1	X	1
0	1	X	X	1
1	X	X	X	1

Y = A+B+C+D

4073

Triple 3-input AND gates.

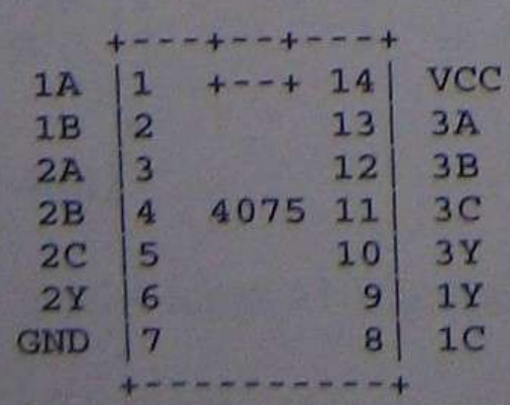


A	B	C	Y
0	X	X	0
1	0	X	0
1	1	0	0
1	1	1	1

Y = ABC

4075

Triple 3-input OR gates.

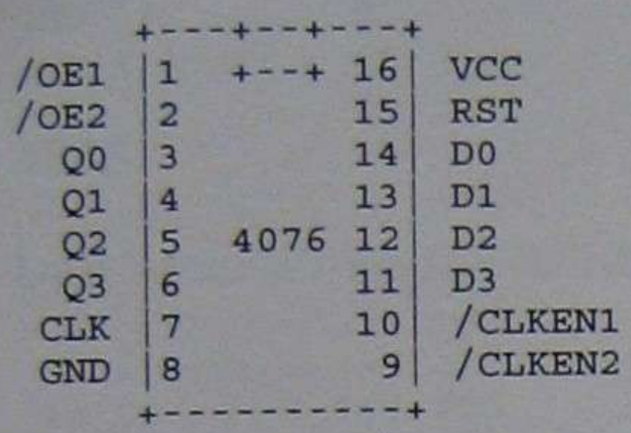


A	B	C	Y
0	0	0	0
0	0	1	1
0	1	X	1
1	X	X	1

Y = A+B+C

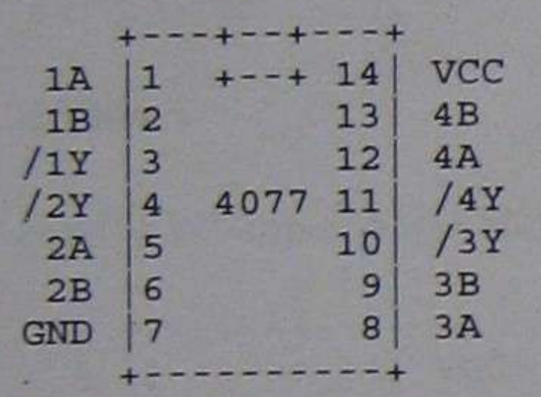
4076

4-bit 3-state D flip-flop with reset, dual clock enables and dual output enables.



4077

Quad 2-input XNOR gates.

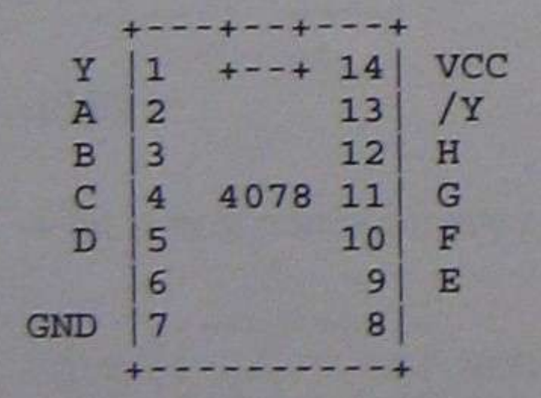


A	B	/Y
0	0	1
0	1	0
1	0	0
1	1	1

$Y = A\overline{\$}B = (\overline{A}.\overline{B}) + (A.B)$

4078

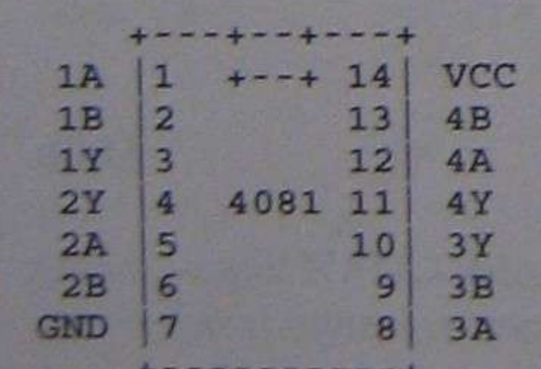
8-input OR/NOR gate with complementary outputs.



$Y=A+B+C+D+E+F+G+H$

4081

Quad 2-input AND gates.



A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

$Y = AB$

4082

Dual 4-input AND gates.

```
 1Y |1   +--+ 14| VCC
 1A |2       13| 2Y
 1B |3       12| 2D
 1C |4  4082  11| 2C
 1D |5       10| 2B
    |6        9| 2A
GND |7        8|
    +----------+
```

A	B	C	D	Y
0	X	X	X	0
1	0	X	X	0
1	1	0	X	0
1	1	1	0	0
1	1	1	1	1

$Y = ABCD$

4085

Dual 3-wide 2/1-input AND-NOR gates.

```
     +---+--+---+
 1A |1   +--+ 14| VCC
 1B |2       13| 1D
/1Y |3       12| 1C
/2Y |4  4085  11| 1E
 2A |5       10| 2E
 2B |6        9| 2D
GND |7        8| 2C
    +----------+
```

$/Y = \overline{AB+CD+E}$

4086

6-wide 2/1-input AND-NOR gate.

```
     +---+--+---+
  A |1   +--+ 14| VCC
  B |2       13| H
 /Y |3       12| G
    |4  4086  11| K
  C |5       10| J
  D |6        9| F
GND |7        8| E
    +----------+
```

$/Y = \overline{AB+CD+EF+GH+J+/K}$

4089

4-bit synchronous binary rate multiplier.

```
     +---+--+---+
Q15 |1   +--+ 16| VCC
 D2 |2       15| D1
 D3 |3       14| D0
SET |4       13| RST
 /Q |5  4089  12| CASC
  Q |6       11| CIN
COUT|7       10| STB
 GND|8        9| CLK
    +----------+
```

4093

Quad 2-input NAND gates with schmitt-trigger inputs.
0.9V typical input hysteresis at VCC=+5V and 2.3V at VCC=+10V.

```
     +---+--+---+
 1A |1   +--+ 14| VCC
 1B |2       13| 4B
/1Y |3       12| 4A
/2Y |4  4093  11| /4Y
 2A |5       10| /3Y
 2B |6        9| 3B
    +----------+
```

A	B	/Y
0	0	1
0	1	1
1	0	1
1	1	0

$/Y = \overline{AB}$

```
GND |7          8| 3A          +---+---*---+
    +-----------+
```

4094

8-bit 3-state serial-in parallel-out shift register with output latches.
Q7' is Q7 delayed by half a cycle (i.e. clocked on falling edge).

```
     +---+--+---+
  LE |1  +--+ 16| VCC
   D |2       15| OE
 CLK |3       14| Y4
  Y0 |4       13| Y5
  Y1 |5  4094 12| Y6
  Y2 |6       11| Y7
  Y3 |7       10| Q7
 GND |8        9| Q7'
     +----------+
```

4095

J-K flip-flop with triple ANDed J an K inputs, set and reset.

```
     +---+--+---+
     |1  +--+ 14| VCC
 RST |2       13| SET
  J1 |3       12| CLK
  J2 |4  4095 11| K3
  J3 |5       10| K2
  /Q |6        9| K1
 GND |7        8| Q
     +----------+
```

J1.J2.J3	K1.K2.K3	CLK	SET	RST	Q	/Q
X	X	X	1	1	0	0
X	X	X	1	0	1	0
X	X	X	0	1	0	1
0	0	/	0	0	-	-
0	1	/	0	0	0	1
1	0	/	0	0	1	0
1	1	/	0	0	/Q	Q
X	X	!/	0	0	-	-

4096

J-K flip-flop with triple ANDed J an K inputs (one inverted), set and reset.

```
     +---+--+---+
     |1  +--+ 14| VCC
 RST |2       13| SET
  J1 |3       12| CLK
  J2 |4  4096 11| K1
 /J3 |5       10| K2
  /Q |6        9| /K3
 GND |7        8| Q
     +----------+
```

J1.J2./J3	K1.K2./K3	CLK	SET	RST	Q	/Q
X	X	X	1	1	0	0
X	X	X	1	0	1	0
X	X	X	0	1	0	1
0	0	/	0	0	-	-
0	1	/	0	0	0	1
1	0	/	0	0	1	0
1	1	/	0	0	/Q	Q
X	X	!/	0	0	-	-

4097

16-to-2 line analog multiplexer/demultiplexer.

```
      +------+--+------+
   1Y |1     +--+ 24| VCC
 1X7  |2         23| 2X0
 1X6  |3         22| 2X1
 1X5  |4         21| 2X2
 1X4  |5         20| 2X3
 1X3  |6         19| 2X4
```

```
1X2 |  7          4097          15 |  2X5
1X1 |  8                        17 |  2Y
1X0 |  9                        16 |  2X6
 S0 | 10                        15 |  2X7
 S1 | 11                        14 |  S2
GND | 12                        13 |  /EN
```

4098

Dual monostable multivibrator, retriggerable, resettable.

```
1Cext  |  1          16 |  VCC
1RCext |  2          15 |  2Cext
 1RST  |  3          14 |  2RCext
 1TR   |  4          13 |  2RST
/1TR   |  5  4098    12 |  2TR
 1Q    |  6          11 |  /2TR
/1Q    |  7          10 |  2Q
GND    |  8           9 |  /2Q
```

4099

1-of-8 addressable latch with reset.

```
Q7  |  1          16 |  VCC
RST |  2          15 |  Q6
 D  |  3          14 |  Q5
/WR |  4          13 |  Q4
 A0 |  5  4099    12 |  Q3
 A1 |  6          11 |  Q2
 A2 |  7          10 |  Q1
GND |  8           9 |  Q0
```

Back to the GHCM index.

Review Questions

1. Simplify the following Boolean expressions using Boolean postulates (ie. theorems)

   (a) $A + (A.B)$

   (b) $A.(\overline{A} + B)$

   (c) $(A + B) . (B + C)$

   (d) $\overline{\overline{A.B.C} . \overline{A.B.\overline{C}} . \overline{A.\overline{B}.C}}$

2. Simplify the following Boolean expressions using Karnaugh maps.

   a) $A + A.B + \overline{A}.B$

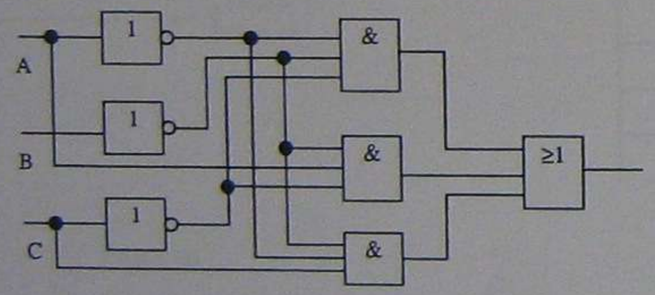
   b) $A.B + \overline{A}.\overline{B}.C + A.C + B.C$

   c) $A.\overline{B} + A.B.\overline{C} + \overline{A}.\overline{B} + A.B.C$

   d) $\overline{A}.\overline{C} + A.B.C + A.C + \overline{A}.B.C + \overline{A}.C$

   e) $\overline{A}.B + A.B.C + \overline{A}.\overline{C} + A.\overline{C}$

3. Simplify and then redraw the following circuit using 2-input NAND gates only.



Skill Practice 4: Digital Logic Simplification

Aim

In this exercise you will construct basic digital circuits on a bread-boarding socket, and use a logic probe to measure output logic levels and record them in a truth table. You will then determine the Boolean expression for the output to be simplified using Boolean postulates or Karnaugh maps.

Equipment

- IC types: 74LS00 x 2, '02, '04, '08, '11, and '32
- Digital bread-boarding system
- Logic probe
- IC extractor.

Procedure

- Using your data sheets, assign pin numbers to the circuit to be constructed.
- Construct the circuit and connect the inputs to the logic switches.
- Write the Boolean expression for the constructed circuit.
- Apply each binary combination to the logic switches and record the resulting output in the truth table.
- Use Boolean postulates and Karnaugh maps to simplify the output expression.

Part 1

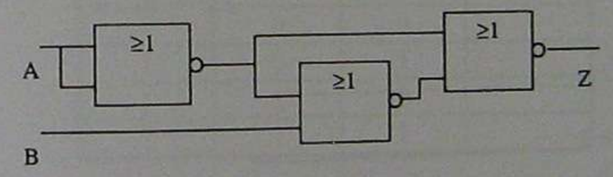


Figure 1

A	B	Z

Truth table

Boolean expression of figure 1: _____

Simplified expression of figure 1: _____

Review Questions

1. Explain 'triboelectric' and 'induction' electrostatic charging.

2. List the environmental conditions in the workplace that would contribute to the generation of static charges.

3. List materials found in the workplace that would contribute to the generation of static charges.

4. List the three common myths regarding ESD.

5. What typical value of electrostatic voltage be generated by walking across a vinyl floor in an air-conditioned room?

6. What value of resistance is used to terminate a static wrist-strap to the ground potential?

7. How can an electronic component be damaged by ESD while stored in a 'pink poly' bag?