M.C.A. (First Semester) Examination,2013
Digital Electronics
Paper: Fifth

---

1. (i) 0.8125
X 2
1.6250
X 2
1.250
X 2
0.500
X 2
1.000

$(0.8125)_{10}=(.1101)_2$

---

ii) An integrated circuit (IC) fabricated on a die of a silicon semiconductor crystal, called a chip, containing the electronic components for constructing digital gates. The various gates are interconnected/ integrated inside the chip to form the digital integrated circuit. Examples of integrated circuits are encoder, decoder, multiplexer, de multiplexer depending on the integration of basic units.

---

iii) $(6D.3A)_{16} =(01101101.00111010)_2$

---

iv) Analog to digital converter converters anlog signal to digital signal and Digital to analog converter converts digital signal to analog signal.

---

v) In computer a real or floting point number is represented by mantissa and exponent. The first part , the mantissa is a signed fixed real number . The second part, the exponent indicates the position of the binary . For example the decimal number 3584.69 s represented in floting point representation as

Sign		Sign	
0	.358469	0	04
Mantissa		Exponent	

Vi)

$(5B.3A)_{16}= (01011011.00111010)_2= (001\ \ 011\ \ 011\ .\ \ 001\ \ 110\ \ \ 100)_2= (\ 1\ 3\ 3.\ 1\ 6\ 4)_2$

---

vii) SOP : If all the **MIN tems** are added or ORed together they form Sum of product equation.

Example : Y(**ABC**) = A'BC'+AB'C+ABC

POS: If all the **MAX terms** are multiplied or ANDes together they form Product of Sum equation

Example : Y(**ABC**) = (A+B+C')(A'+B+C)(A'+B'+C')

---

viii) The Short comings of SR flip flop is the undetermined condition (input 11) is never used because its output is undetermined and the input 00 produces no change in the output.

---

ix) Truth table: A table containing **all possible combinations** of the input variables and the corresponding values of the output variables is known as truth table. For n input variable it contains four (2^n) combinations of inputs and their corresponding outputs (2^n).

Example : Y=A.B
Total number of variables = 2
Total number of possible combination = 2^2=4

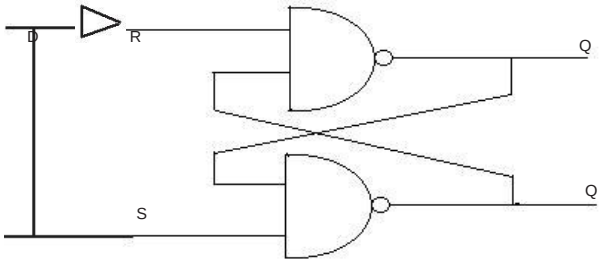
Inputs		Output
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

X) **Digital Encoder** more commonly called a **Binary Encoder** takes *ALL* its data inputs one at a time and then converts them into a single encoded output. An "n-bit" binary encoder has 2^n input lines and n-bit output lines

A **Decoder** is the exact opposite to that of an "Encoder" It is basically, a combinational type logic circuit that converts the binary code data at its input into one of a number of different output lines, one at a time producing an equivalent decimal code at its output. Therefore, if a binary decoder receives n inputs it activates one and only one of its 2^n outputs based on that input with all other outputs deactivated. A decoders output code normally has

SECTION-B

2.a) D Flip - flop: If the the design of Rs flop flop is modified to eliminate the possibility of a raced condition. the result is a new kind of flip - flop known as D flipflop



D	Q
0	0
1	1

Fig. No. 2.9

Fig. 2.9. shows one way to build a D latch (unlocked) because of the inverter, data bit D drives the S input of a NAND latch and compliment D drives the R input. Therefore, a high D sets the latch, and low D reset it. Observe, there is no race condition in this truth table. The inverter guarantees that S and R will always be in opposite states.

Therefore it is impossible to set up a race condition in the D latch.

Clocked RS flip – flop

Two different methods for constructing an RS flip – flop were discussed in the previous section with NOR gate and NAND gate realization. Both of these RS flip – flops or latches, are said to be "transparent "; that is any change in input at R or S is transmitted immediately to the output at Q and Q thus they acts as short term memory.

It is possible to store or clock the flip – flop in order to store information (set it or reset it) at any time, and then hold the stored information for any desired period of time. This flip – flop is called clocked RS flip – flop.

The circuit of a clocked RS flip – flop is shown in figure 2.6 with its symbol and truth table.

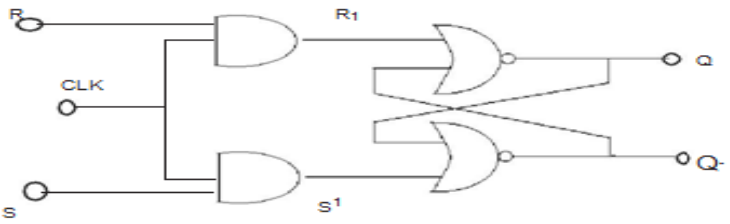


Fig. No. 2.6

CLK	R	S	Q
0	0	0	NC
0	0	1	NC
0	1	0	NC
0	1	1	NC
1	0	0	NC
1	0	1	1
1	1	0	0
1	1	1	* (Race)

clocked RS flip – flop x symbol and its truth table

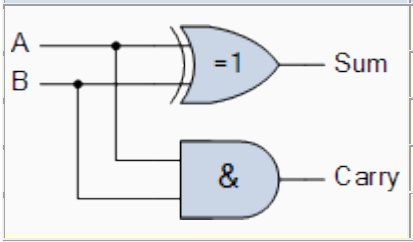
It consists of two additional and gates added at the input of R S flip – flop. In addition to control inputs R and S, there is a clock input CK.

The output of the two and circuits (S^1 and R^1) will be 0 as long as CK = 0. then the state of the flip – flop will remain unchanged and if S = 0 , R = 1 then S = 0 R = 1 and the flip – flop is reset to 0. On the other hand if S = 1 , R = 0 then S =1 R = 1 and the flip flop is set to 1. the presence of R^1 =1 and S^1 =1 , will however , results in an undetermined state.

2.b)

Half Adder : It is a combinational circuit which performs the arithmetic addition of two bits.

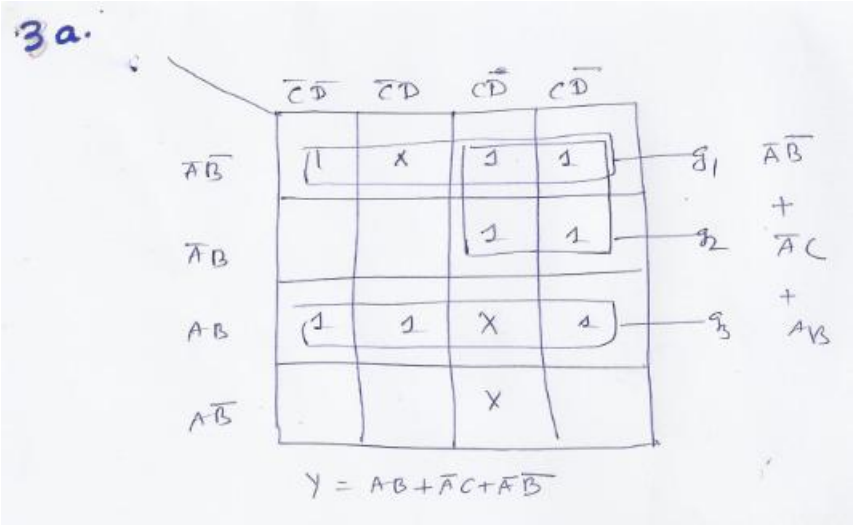
The Half Adder Circuit

Symbol	Truth Table			
	A	B	SUM	CARRY
	0	0	0	0
	0	1	1	0
	1	0	1	0
	1	1	0	1
Boolean Expression: Sum = A ⊕ B Carry = A . B				

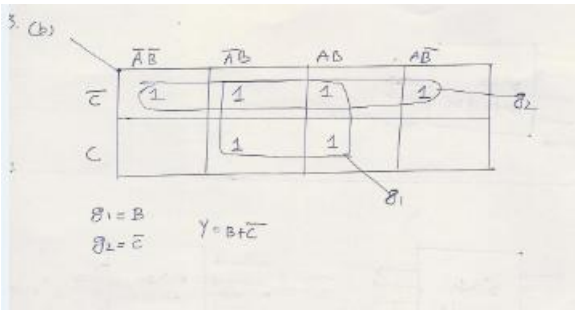
From the truth table we can see that the SUM (S) output is the result of the Ex-OR gate and the Carry-out (Cout) is the result of the AND gate. One major disadvantage of the Half Adder circuit when used as a binary adder, is that there is no provision for a "Carry-in" from the previous circuit when adding together multiple data bits.

For example, suppose we want to add together two 8-bit bytes of data, any resulting carry bit would need to be able to "ripple" or move across the bit patterns starting from the least significant bit (LSB). The most complicated operation the half adder can do is "1 + 1" but as the half adder has no carry input the resultant added value would be incorrect. One simple way to overcome this problem is to use a **Full Adder** type binary adder circuit.

3.a

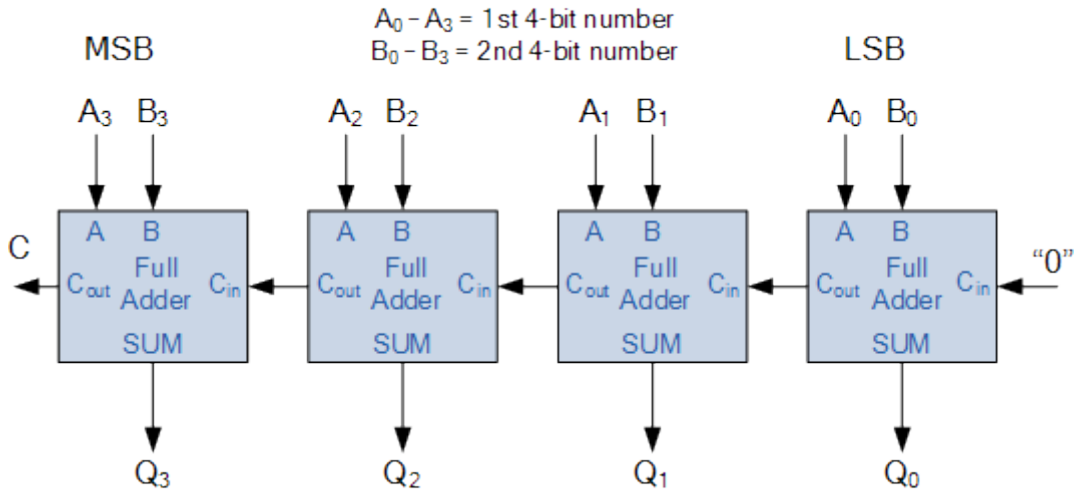


3.b



4.The 4-bit Binary Adder

A 4-bit Binary Adder



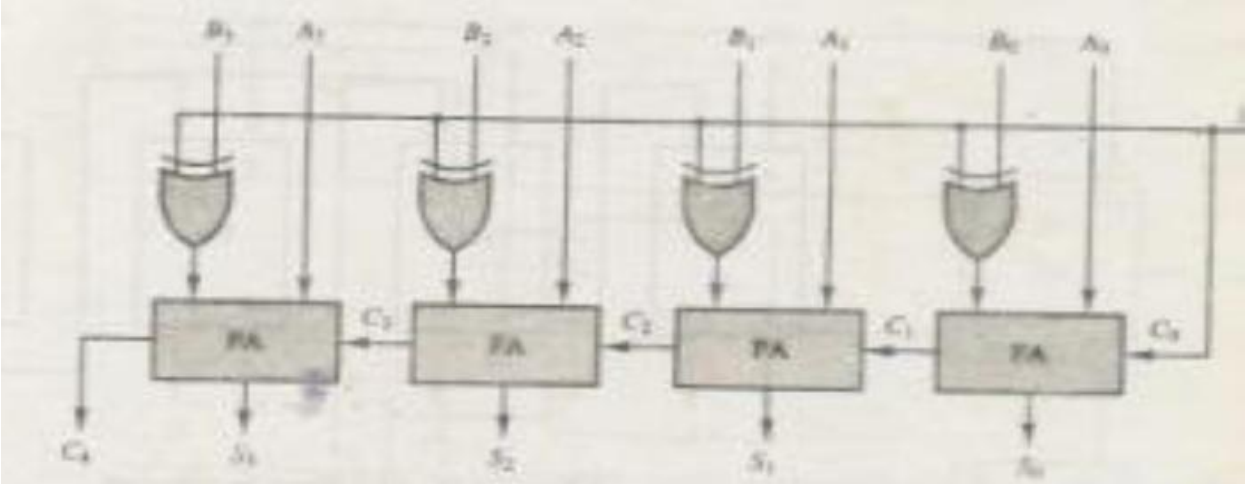
One main disadvantage of "cascading" together 1-bit **binary adders** to add large binary numbers is that if inputs A and B change, the sum at its output will not be valid until any carry-input has "rippled" through every full adder in the chain. Consequently, there will be a finite delay before the output of a adder responds to a change in its inputs resulting in the accumulated delay especially in large multi-bit binary adders becoming prohibitively large. This delay is called **Propagation delay**. Also "overflow" occurs when an n-bit adder adds two numbers together whose sum is greater than or equal to 2^n

One solution is to generate the carry-input signals directly from the A and B inputs rather than using the ripple arrangement above. This then produces another type of binary adder circuit called a **Carry Look Ahead Binary Adder** were the speed of the parallel adder can be greatly improved using carry-look ahead logic.

Adder-Subtractor:

The subtraction of binary numbers can be done most continently by means of complements . The subtraction A-B can be done by taking the 2's complement of B and adding it to A. The 2's complement can be obtained by taking the 1's complement and adding one to the least significant pair of bits. The I's complement can be implemented with inverter and a one can be adds to the sum through the input carry.

The addition and substraction operation can be combined into one common circuit by including an exclusive- OR gate with each full-adder. A 4 bit adder substractor circuit is as follows. The mode input M controls the operation. When M=0 the circuit is an adder and when M=1 the circuit becomes a sunstractor. Each exclusive OR gate receives input M and one of the inputs B. When M=0, we have B ex-OR 0=B, The full adder receives the v alue of B, The input carry is 0 and the circuit performs A plus B .When M=1. We have B ex-Or 1 =B' and C0=1. The b inputs are all complemented and a 1 is addedthrough the input carry.



A= A3 A2 A1 A0 (0 0 1 1)

B= B3 B2 B2 B0 (0 0 1 0)

Addition : (1^{st} full adder)

Input: A0=1 B0=0 C0=1

Output : S0=1 C1=0

(2^{nd} full adder)

Input A1= 1 B1=1 c1=0

Output : S1 =0 C1=1 ……

5a. $(1)_8 + (4)_8$

Binary values of the above 001 + 100 = 101 in binary

Again converting into octal $(101)_2=(5)_8$

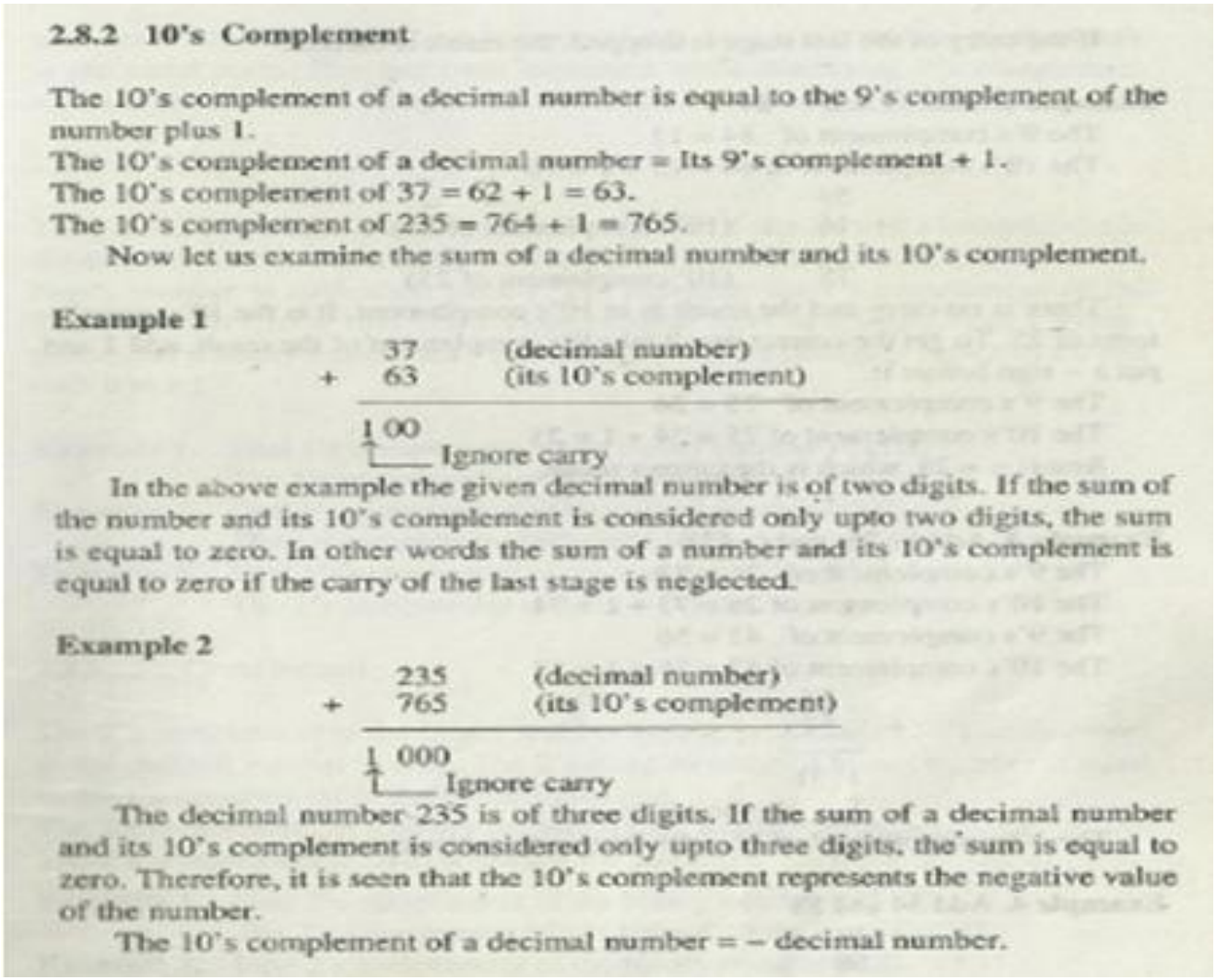
$(5)_{16} + (3)_{16}$

Binary values of the above 101 + 011 = 1000

Again converting into octal (1000) = $(8)_{16}$

5.b 9's Complement and 10's complement

To form the 9's complement of decimal number each digit of a decimal number is subtracted from 9. The result so obtained is known as 9's complenent of the number. For example the 9's complenent of 37 is (99-37)=62 . The 9's complement of 235 is (999-235)=764



2.8.2 10's Complement

The 10's complement of a decimal number is equal to the 9's complement of the number plus 1.

The 10's complement of a decimal number = Its 9's complement + 1.

The 10's complement of 37 = 62 + 1 = 63.

The 10's complement of 235 = 764 + 1 = 765.

Now let us examine the sum of a decimal number and its 10's complement.

Example 1

```
   37      (decimal number)
+  63      (its 10's complement)
-----
 1 00
 ↑
 └── Ignore carry
```

In the above example the given decimal number is of two digits. If the sum of the number and its 10's complement is considered only upto two digits, the sum is equal to zero. In other words the sum of a number and its 10's complement is equal to zero if the carry of the last stage is neglected.

Example 2

```
   235     (decimal number)
+  765     (its 10's complement)
-------
 1 000
 ↑
 └── Ignore carry
```

The decimal number 235 is of three digits. If the sum of a decimal number and its 10's complement is considered only upto three digits, the sum is equal to zero. Therefore, it is seen that the 10's complement represents the negative value of the number.

The 10's complement of a decimal number = – decimal number.

---

6. (Any three differences of each)

Static RAM : 1. Used for cache memory.

2.Very cost effective.

3. Access time is faster

4. less storage space

Dynamic Ram 1. Used for main memory

2. Cost is less than SRAM

3. Access time is less as compared to SRAM

4. Storage capability is high.

Primary memory: 1. Access speed is high.

2. Less storage capacity.

3. Primary memories are internal memory.

4. May be volatile or non volatile.

Secondary memory: 1. Access speed is slow

2. High storage capacity

3. Secondary memories are external.

4. Secondary memories are non volatile memories.

Main memory: 1. Storage capacity is high

2. High access speed

3.Constructed from DRAMs

4. Stores currently used data.

Cache memory: 1. Storage capacity is low

2. low access speed

3. Constructed from SRAM

4. Stores frequently used data.

Multiplexer: 1. It is a digital circuit which takes n input data line and selects one out of them or produces one output line.

2.Many input and one output.

De multiplexer: 1.It is a digital circuit which takes one data input line and places the data on any of 2^n output lines.

2.One input line and many output line.

---

7 A multiplexer is a combinational circuit which selects binary information from one of many inputs and direct it into a single output line.The input selection is controlled by a set of selection lines.It takes 2^n input lines and one output lines

16-to-1 multiplexer:

A 16 to 1 multiplexer has sixteen inputs and a 4 selection lines. At a time one of the input appears at the output depending on the selection line. Following figure shows the block diagram of 16 X 1 multiplexer.

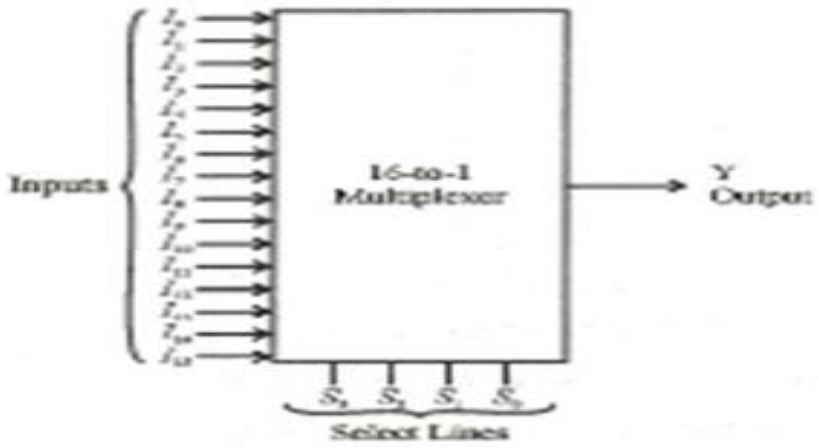
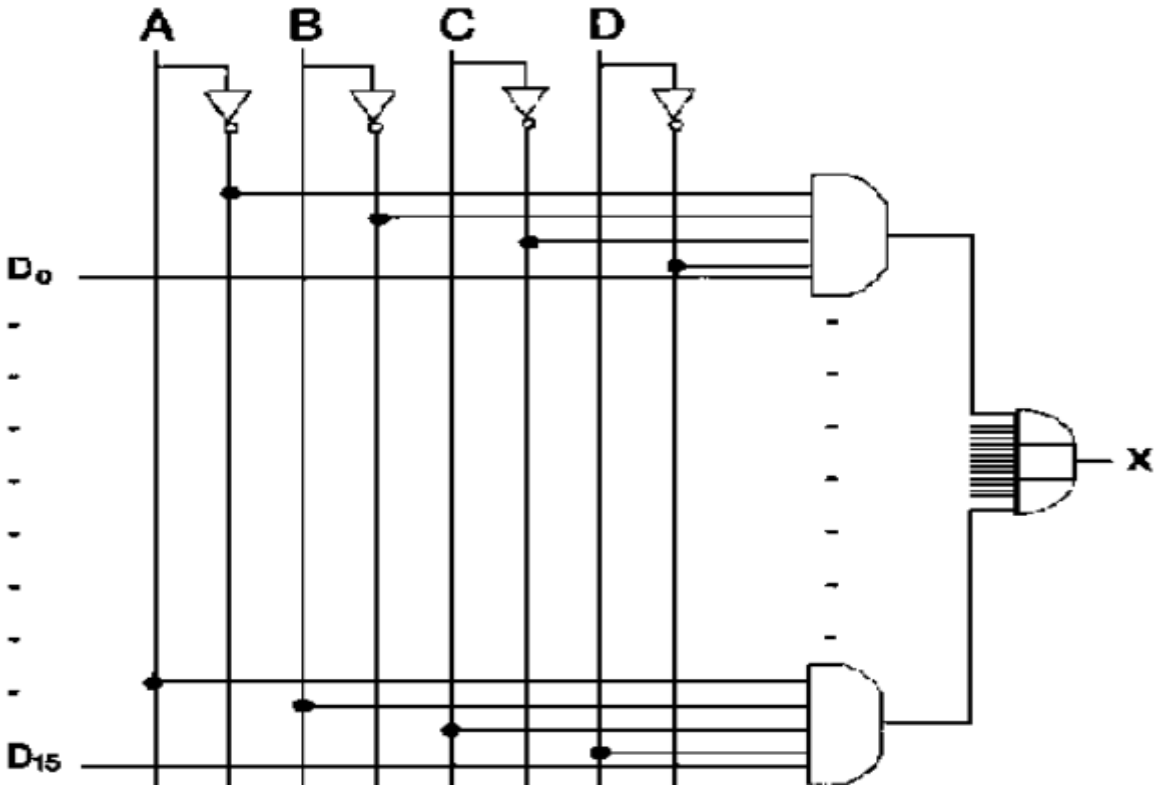


Fig. 7.25 Block Diagram of 16-to-1 Multiplexer

Y=So’ S1’ S2’ S3’ +……+S0S1S2S3



16 X 1 multiplexer

.

.

.

8a. 4-Bit binary counter:

A register that goes through a predetermined sequence of steps up on the application of input pulses is known as counters.A counter that follows binary sequence is called binary counter. A n bit counter is a register of n Flip flops and associated gates. A binary counter goes through the sequence of binary number 0000,0001,….1111. Every lower order bit is complemented after the next sequence . Ex- Binary count 0111 to 1000 is obtained by a) complementing the lower order bit b) complementing the second order bit because the first two bits of 0111 is 1 and complementing 4^{th} bit because all previous bits are 1

A 4 bit binary counter can be implemented by

By using JK FF. JK FF has the property of toggling when the inpts are 11 and No change when the inputs are 00. In addition the counter is controlled by using an enable input that turns the counter on or off. If the JK input is maintained at 00 there is no change in output. Wnen the enable input is 1 The lower order bit changes to 1 and rest of the bits doesnot change giving rise to the sequence 0001. The next clock pulses

results 0010……..1111.

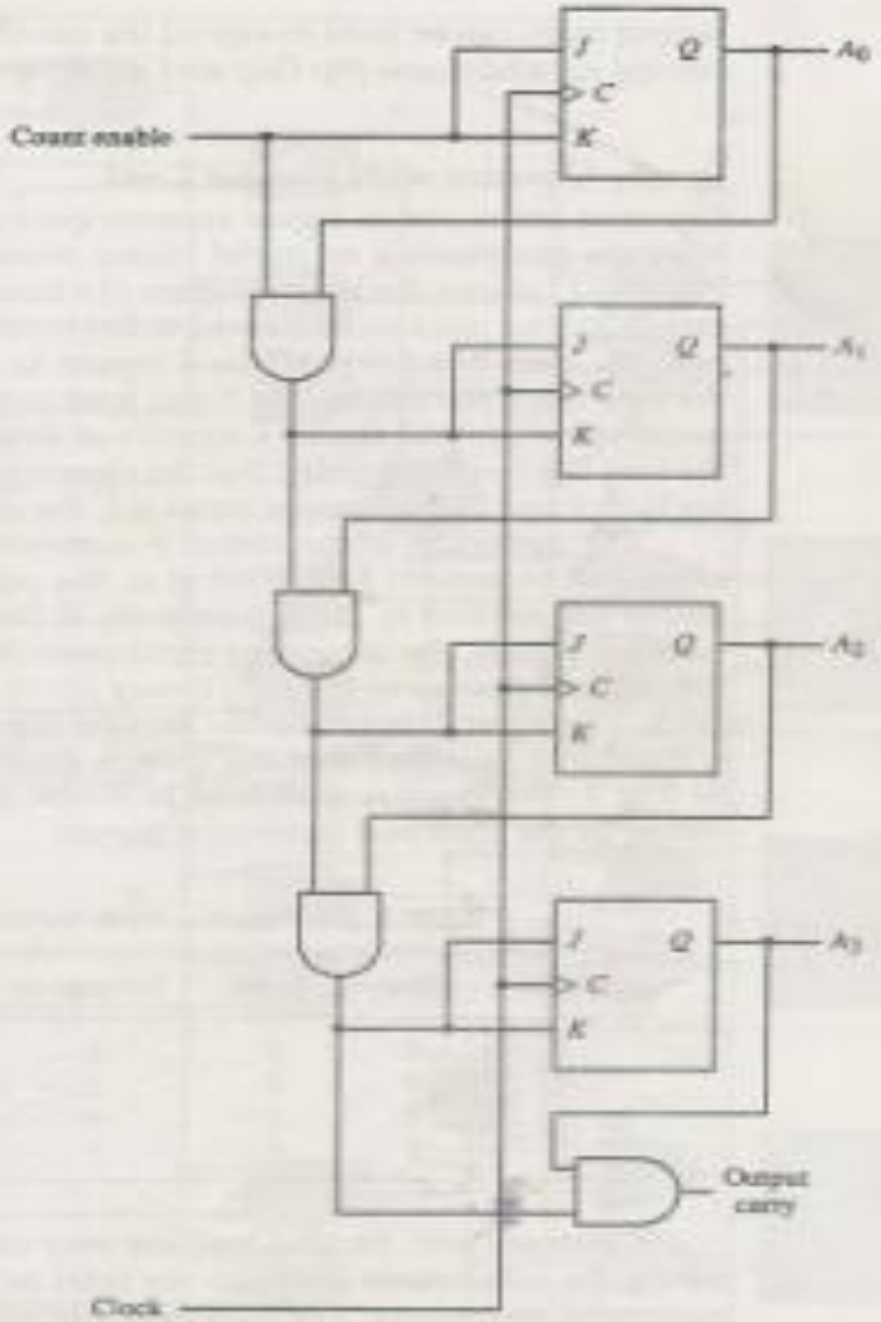


Figure 2-10 4-bit synchronous binary counter.

at 0 and the output of the counter does not change. The first stage A_0 is complemented when the counter is enabled and the clock goes through a positive transition. Each of the other three flip-flops are complemented when all previous least significant flip-flops are equal to 1 and the count is enabled. The chain of AND gates generate the required logic for the J and K inputs. The

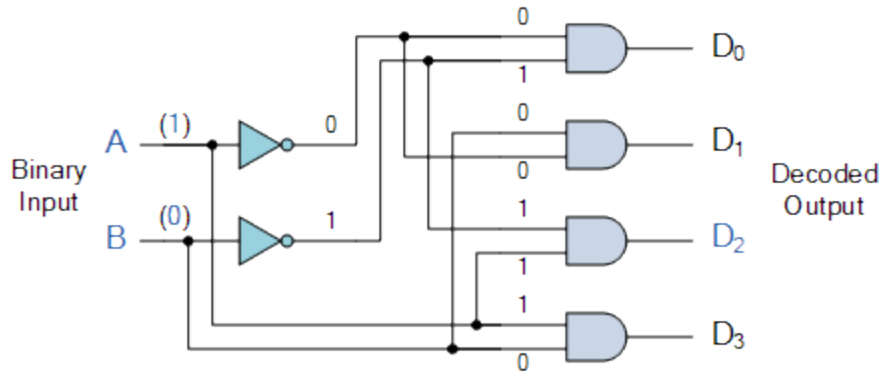
8. b. 2X4 Binary Decoder

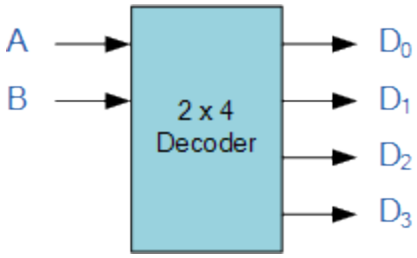
A **Decoder** is the exact opposite to that of an "Encoder" we looked at in the last tutorial. It is basically, a combinational type logic circuit that converts the binary code data at its input into one of a number of different output lines, one at a time producing an equivalent decimal code at its output. **Binary Decoders** have inputs of 2-bit, 3-bit or 4-bit codes depending upon the number of data input lines, and a n-bit decoder has 2^n output lines.

Therefore, if a binary decoder receives n inputs (usually grouped as a binary or Boolean number) it activates one and only one of its 2^n outputs based on that input with all other outputs deactivated. A decoders output code normally has more bits than its input code and practical "binary decoder" circuits include, 2-to-4, 3-to-8 and 4-to-16 line configurations.

A *Binary Decoder* converts coded inputs into coded outputs, where the input and output codes are different and decoders are available to "decode" either a Binary or BCD (8421 code) input pattern to typically a Decimal output code. Commonly available BCD-to-Decimal decoders include the TTL 7442 or the CMOS 4028. An example of a 2-to-4 line decoder along with its truth table is given below. It consists of an array of four NAND gates, one of which is selected for each combination of the input signals A and B.

A 2-to-4 Binary Decoders.





A	B	D_0	D_1	D_2	D_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

In this simple example of a 2-to-4 line binary decoder, the binary inputs A and B determine which output line from D0 to D3 is "HIGH" at logic level "1" while the remaining outputs are held "LOW" at logic "0" so only one output can be active (HIGH) at any one time. Therefore, whichever output line is "HIGH" identifies the binary code present at the input, in other words it "de-codes" the binary input

and these types of binary decoders are commonly used as **Address Decoders** in microprocessor memory applications.

D0=A’B’

D1=A’B

D2=AB’

D3=AB
