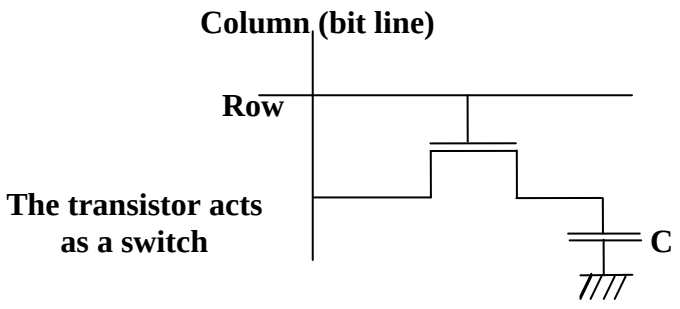
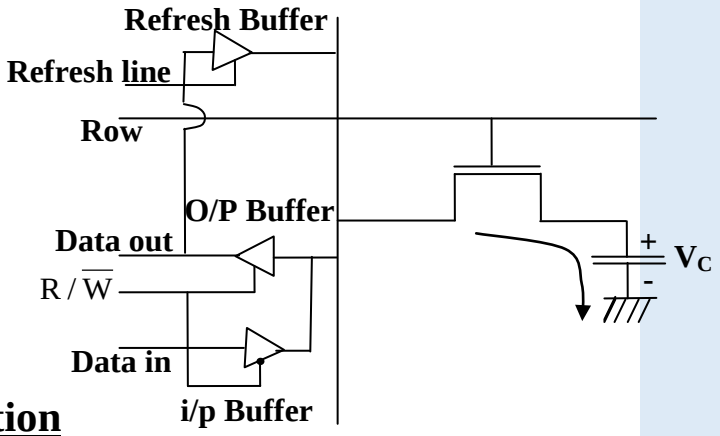
Lectures of Electrical Engineering Department

Subject Title: Digital Electronics

Class:3rd

Lecture Contents	**Lecture sequences:**	**First lecture**	**Instructor Name:**
	The major contents: **1-DRAM: definition and charactaristics** **2- DRAM operation** **3- Design of DRAM**		
	The detailed contents: **1- DRAM** • The memory cell is capacitor and MOSFET. • The main advantage is its higher density or more bits per package compared with SRAM because the memory cell is very simple compared with half of SRAM. Also the cost per bit is less in the case of DRAM.  **MOS DRAM Cell** • The process (Memory refresh) is done every 5-10 ms. • Data can be read much faster from SRAM than from DRAM. • SRAM stored data indefinitely as long as power is applied but DRAM cannot retain very long without the capacitors being recharged by a process called refreshing.		

2-DRAM Operation



1) Write Operation

$R/\overline{W}$ = Low, Row = High, Refresh line = Low

i/p buffer enable $D_{in} = 0, V_c = 0$

$D_{in} = 1, V_c = 5$ Volt

o/p buffer = disenable

2) Read Operation

$R/\overline{W}$ = High, Row = High, Refresh line = Low

o/p buffer enable $V_c = 0 \Rightarrow D_{out} = "0"$

$V_c = 5 \Rightarrow D_{out} = "1"$

i/p buffer disenable

3) Refresh Operation

$R/\overline{W}$ = High, Refresh line = High, Row = High

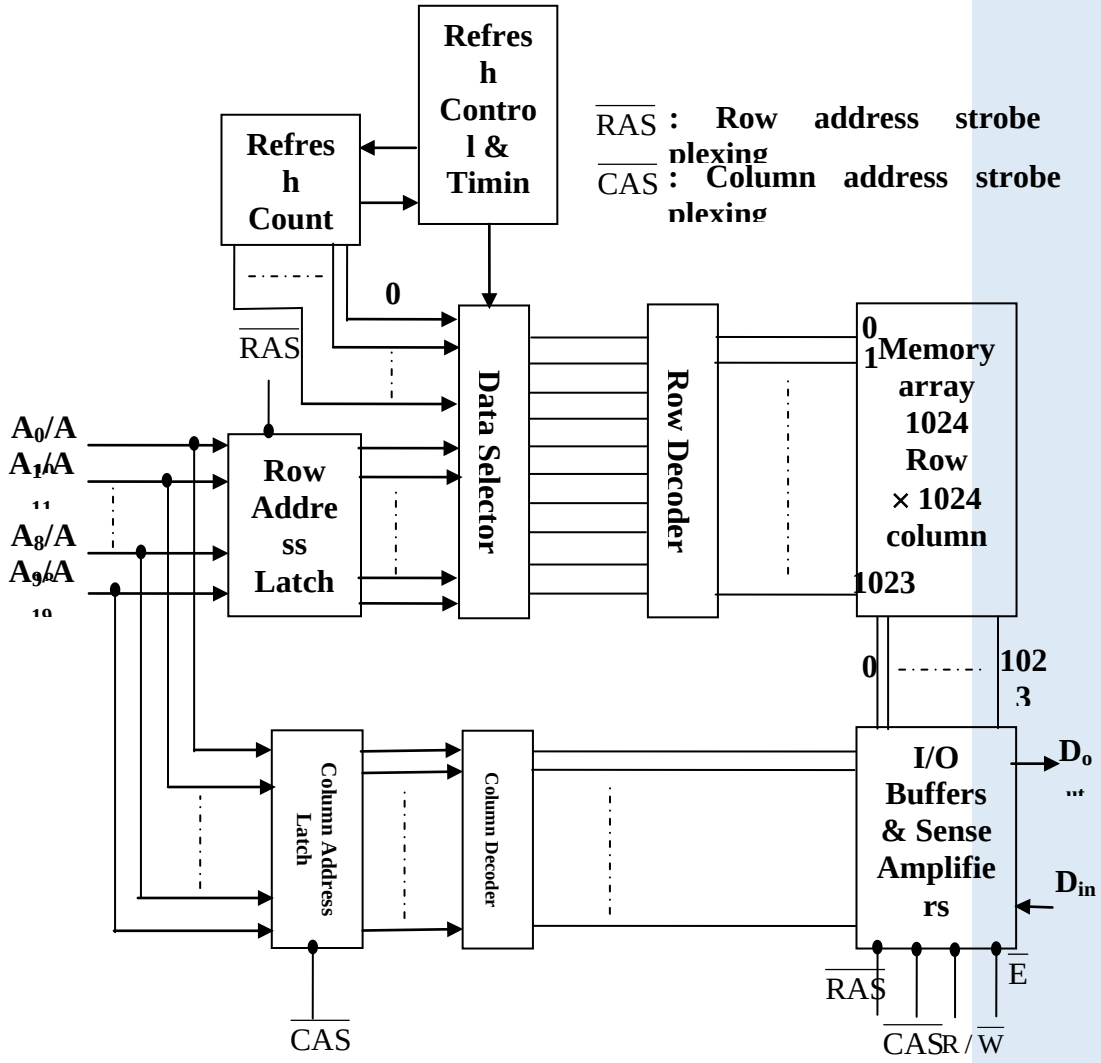
o/p buffer and refresh buffer enabled

Address Multiplexing

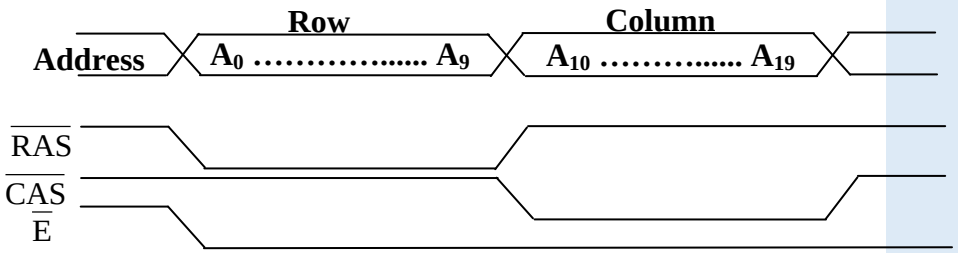
- DRAM use a technique called address multiplexing to reduce the no. of address lines and thus the number of I/O pins of the package (Chip).

3- Example

For 1 M * 1 bit DRAM. Draw the block diagram for the memory, Draw the timing for address multiplexing.



Simplified Block Diagram of 1 M × 1 DRAM



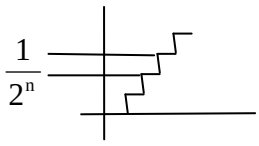
Timing diagram for address multiplexing

اسم التدريسي رقم الهاتف عنوان البريد الاليكتروني

Lectures of Electrical Engineering Department

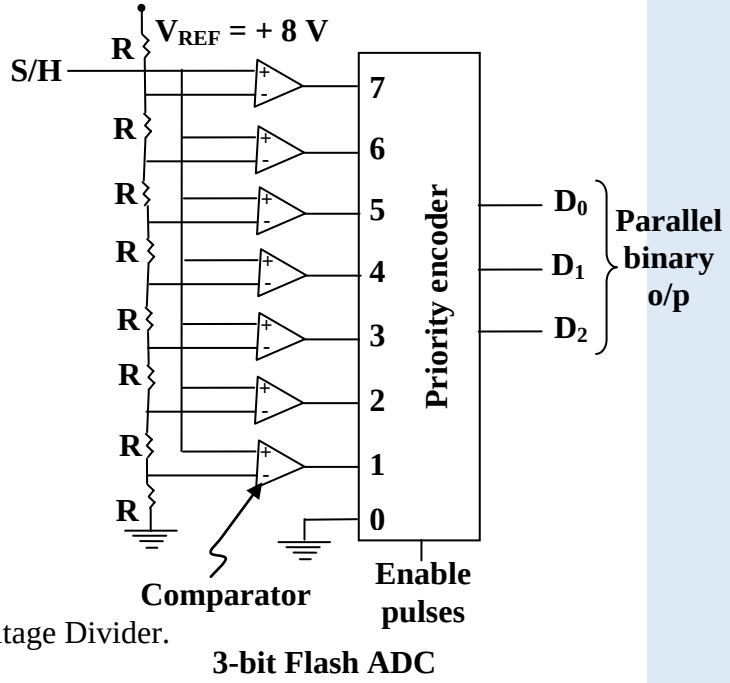
Subject Title: Digital Electronics

Class:3rd Electronics

Lecture Contents	**Lecture sequences:**	**second lecture**	**Instructor Name:**
	The major contents: **1-A/D Converter:Characteristics** **2-Flash A/D converter** **3-Successive-Approximation ADC** **4-Feedback Type ADC**		
	The detailed contents: **1- A/D Converter** The important ADC parameters are resolution and throughput Resolution: is the quantum of the i/p analogue voltage change required to increment its digital O/P from one Code to the Next higher code. An n-bit ADC can resolve one part in (2^n-1). It may be expressed as a percentage of full scale or in bits. The resolution of 8-bit ADC can be expressed as one part in 256 or as 0.39% of full scale or simply as 8-bit resolution. $$R = \frac{V_{ref}}{2^n}$$  **Throughput** Is the sampling rate an ADC can handle in unity of samples per second (SPC).		

2- Flash A/D Converter

(Parallel Simultaneous Converter)



3-bit Flash ADC

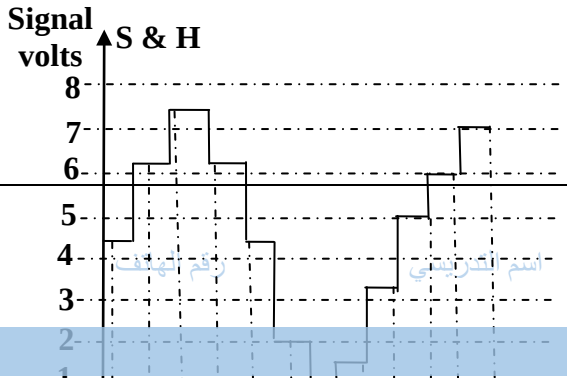
- R: Resistive Voltage Divider.
- No. of R = 2^n.
- No. of comparator = 2^n-1.
- It is chief advantage is that it provides a fast conversion time because of a high throughput measured in sample (Per) second (SPS).
- The large no. of comparators necessary for a reasonable size binary number is one of the disadvantage of this type.

Priority Encoder

An encoder in which only the highest value input digital is encoded and any other active input is ignored.

Example

Determine the binary code output of 3-bit flash ADC for the i/p signal in figure and the encoder enable pulses shown consider V_{ref} = + 8 Volt.

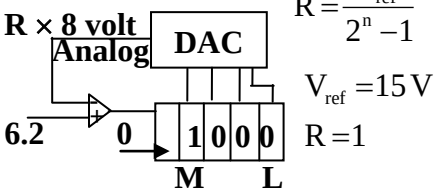


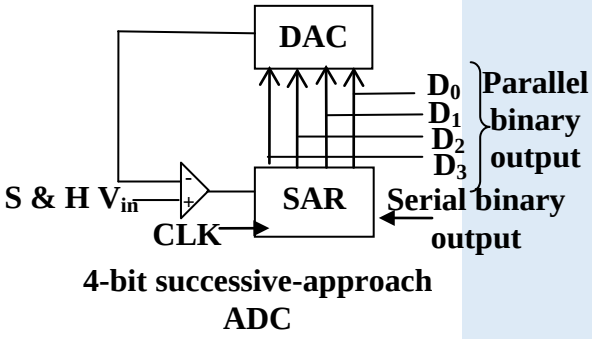
- The frequency of enable pulses and the number of bits in the binary code, determine the accuracy with which the sequence of binary codes represents the input of the ADC.
- There should be one enable pulse for each sampled level of the input signal.

3-2) Successive-Approximation ADC

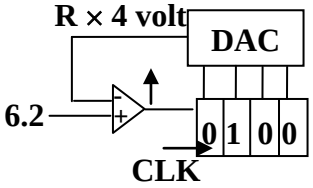
SAR: Successive Approximation Register.

1) MSB trial:

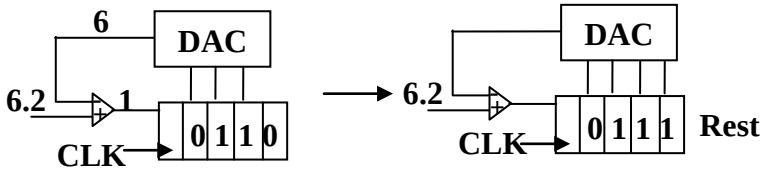




2) 2^2 trial:



3) 2^1 trial:



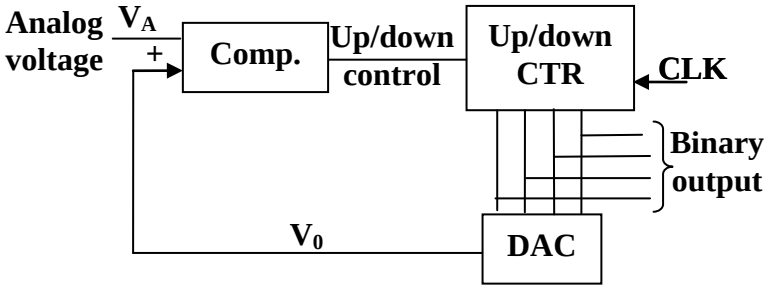
- The final binary is equal to (0110) that reciprocal to the analogue voltage (6.2 Volt).

If V_{ref} = 12 Volt.

$R = \frac{12}{16} = 0.75 \Rightarrow$ Code (1000) binary O/P that reciprocal analog voltage V_A = 6.2 Volt.

- It has much faster conversion time than the Dual Slope ADC but it slower than the flash ADC.

4-4) Feedback Type ADC



- Because of the ADC employs a ADC in its feed-back loop it is usually called a feed-back type ADC.
- The CTR counts either up or down depending on the binary level applied at its up-down control.
- The process continuous until the ADC O/P reaches the value of V_A at which point the comparator switches and stops the CTR.

Lectures of Electrical Engineering Department

Subject Title: Digital Electronics

Class: 3rd Electronics

Lecture Contents	**Lecture sequences:**	**Third lecture**	**Instructor Name:**
	The major contents: **1-Performance Characteristic of DAC** **2-DAC with binary-weighted resistors** **3-R-2R ladder DAC** **4-DAC Errors**		
	The detailed contents: **1- Performance Characteristic of DAC** **1) Resolution:** The resolution of DAC is the reciprocal of the no. of discrete steps in the output. This is of course is dependent on the no. of i/p bits. $R=\frac{1}{2^n-1}*100\%$ For example: a 4-bit DAC has a resolution of one part in 2^4-1 (One part of fifteen) expressed as a percentage $\left(\frac{1}{15}*100\%=6.67\%\right)$. The total no. of discrete steps equal to 2^n-1 where "n" is the number of bit resolution can be expressed also by the number of bits that are converted. **2) Accuracy** It is derived from a comparison of the actual output of DAC with the expected output. It is expressed as a percentage of a full scale of maximum O/P voltage. For		

example of converter has a full scale O/P of 10 Volt and the accuracy is (± 0.1%) then the maximum error for any O/P voltage is (10 * 0.001 = ± 10 mvolt).

$$\text{Accuracy} = \pm\frac{1}{2} * \frac{1}{2^n - 1} * 100\% = \pm\frac{1}{2^{n+1} - 2} * 100\%$$

For 6-bit DAC $\text{Accuracy} = \pm\frac{1}{2^7 - 2} * 100\% = \pm\ 0.79\%$

For 8-bit DAC $\text{Accuracy} = \pm\frac{1}{2^9 - 2} * 100\% = \pm\ 0.196\%$

3) Linearity

A linear error is a deviation from the ideal straight line output of a DAC.

A special case is an offset error which is the amount of O/P voltage when the i/p bits are all zeros.

4) Monotonicity

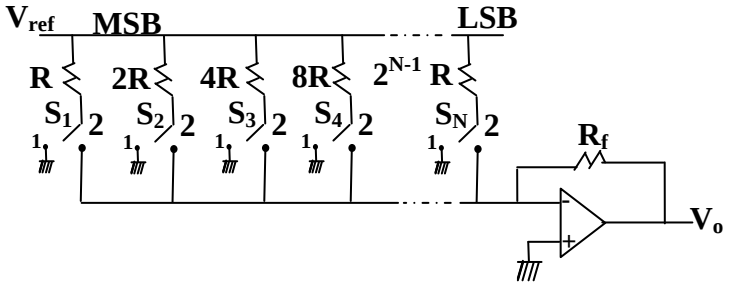
A DAC is monotonic if does not take any reverse steps when it is sequenced over its entire range of i/p bits.

5) Settling Time

It is the time between the switching of the digital inputs of the converter and the time when the O/P reaches its final value and remains within specific error band. It is the time required for the O/P to stabilize or change from its previous value to the new value corresponding to fresh digital inputs word. For a given converter the O/P does not change instantaneously when a change in the i/p occurs.

- General purpose DAC have a settling time of several "μS" while some of the high-speed DAC have a settling time of a few "nS".

2-



1) DAC with binary-weighed resistors

N; Number of bits b_1, b_2, b_3, … b_N = bits coefficient in the figure.

b_1 control S_1, b_2 control S_2 and so on.

$$i_o = \frac{V_{ref}}{R} b_1 + \frac{V_{ref}}{2R} b_2 + ... \frac{V_{ref}}{2^{N-1}R} b_n$$

The current through each resister is constant.

$$= \frac{2V_{ref}}{R}\left[\frac{b_1}{2} + \frac{b_2}{4} + \frac{b_3}{8} + \frac{b_N}{2^N}\right] \quad \text{for 4-bit}$$

$$= \frac{2V_{ref}}{R}[D] \qquad D_{max.} = \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} = 0.9$$

$$V_o = -i_o \times Rf = \frac{-2V_{ref}}{R} \frac{R}{2} \times D \qquad \text{let} \quad Rf = \frac{R}{2}$$

$$V_o = -V_{ref} \times D$$

Let R = 10 KΩ

MSB R_1 = 10 KΩ N = 4-bit

R_2 = 20 K

R_3 = 40 K

LSB R_4 = 80 K $R_{LSB} = 2^{N-1} R_{MSB}$

- One of the disadvantage of this type of the DAC is the number of difference resistor values. For a-8-bit DAC requires 8 resistors ranging from R to 128 R. This range of resistors requires tolerance making this type of DAC very difficult to mass-product.
- It should be noted that the accuracy of DAC depends on:-

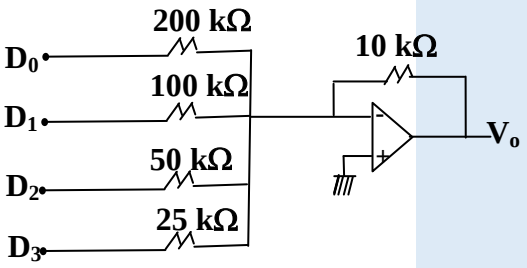
1) The accuracy of V_{ref}.
2) The precision of the binary weighted resistor.
3) The perfection of switches.

Example

Determine the output of the DAC in figure. If the sequence of 4-bit numbers shown below is applied to the inputs. The data inputs have a low value of zero volt and a high value of + 5v.

Solution

D_3	D_2	D_1	D_0	D	V_0 (Volt) = -4
0	0	0	0	0	0
0	1	0	1	0.3125	-1.25
0	0	1	1	0.1875	-0.75
0	0	1	0	0.125	-0.5
0	0	0	0	0	0
0	1	0	1	0.3125	-1.25
1	1	1	1	0.9375	-3.75
1	0	1	0	0.625	-.25



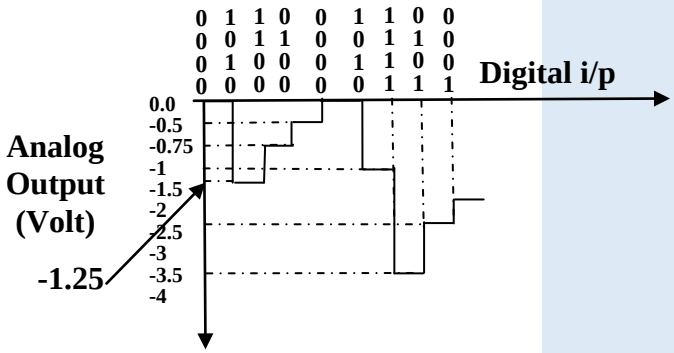
$$V_o = -2\frac{V_{ref}}{R} \times Rf \times D$$

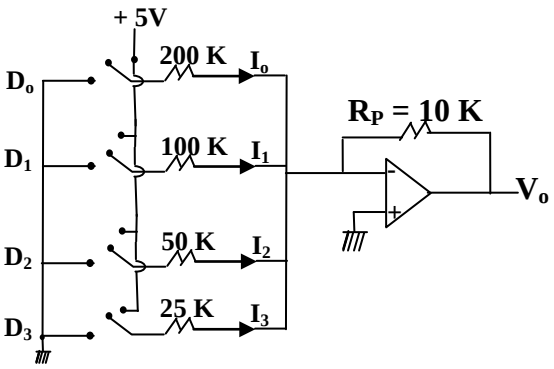
$$D = \frac{b_1}{2} + \frac{b_2}{4} + \frac{b_3}{8} + \frac{b_4}{16}$$

$$= \frac{0}{2} + \frac{1}{4} + \frac{0}{8} + \frac{1}{16}$$

$$= 0.3125$$

$$V_o = \frac{-2\times5}{25} \times 10 \times D$$





$$I_o = \frac{5}{200} = 0.025\,mA \Rightarrow V_o \equiv I_o\,Rf = 0.25 \text{ volt}$$

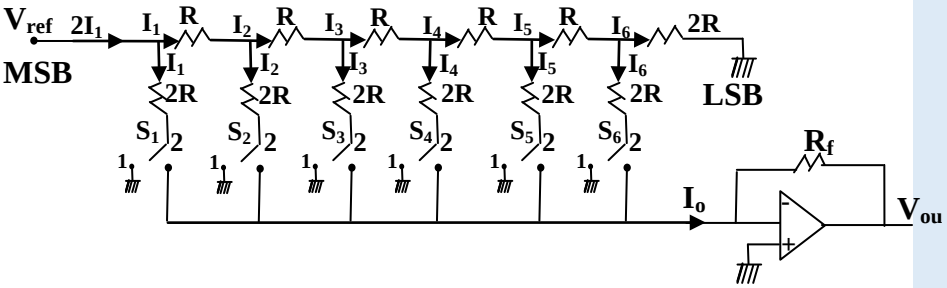
$$I_1 = \frac{5}{100} = 0.05\,mA \Rightarrow V_o \equiv I_1\,Rf = 0.5 \text{ volt}$$

$$I_2 = \frac{5}{50} = 0\,mA \Rightarrow V_o \equiv -1 volt$$

$I_3 = 0.2\,mA \Rightarrow V_o \equiv -2$ volt

D_3	D_2	D_1	D_0	في حالة
0	0	0	0	0
0	1	0	1	-1.25
0	0	1	1	- 0.25 - 0.5 = - 0.75 V

2) 3- R-2R Ladder DAC



$I_o = I_1 b_1 + I_2 b_2 + I_3 b_3 + ... + I_N b_N$ N: no. of bits

$I_1 = 2I_2, I_2 = 2I_3, I_3 = 2I_4$

$I_1 = 4I_3 = 8I_4 = 16I_5 = 32I_6, 2^{n-1} I_N$

$$I_o = I_1 b_1 + \frac{I_1 b_2}{2} + \frac{I_1 b_3}{4} + \frac{I_1 b_4}{8} ..... + \frac{b_N I_1}{2^{N-1}}$$

$I_o = 2I_1 * D$

$$I_1 = \frac{V_{ref}}{2R} \qquad V_{out} = -I_o * Rf$$

$$= -2R \frac{V_{ref} D}{2R} = Rf \quad \text{General}$$

If $Rf = R$ $\qquad V_{out} = -V_{ref} * D$ (only when Rf = R)

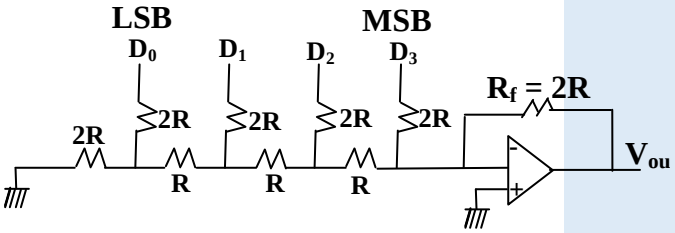
Analog O/P **Digital Input**

- R-2R ladder DAC overcomes one of the problem in the binary-weighted DAC is that it requires only two resistances (R, 2R) (i.e. Small Spread in resistance values). For this reason the R-2R DAC considered the more common type.

Example

Find the output voltage when:

1) Only $D_o = 1$.
2)
3) Only $D_1 = 1$.
4) Only $D_2 = 1$.
5) Only $D_3 = 1$.



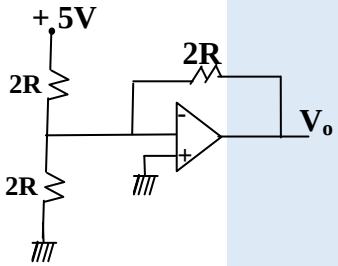
Solution

Method (1):-

1) $D_3 = 1 \quad D_o = D_1 = D_2 = 0$

$$I_o = \frac{5}{2R} \qquad \because V_o = I_o \ Rf$$

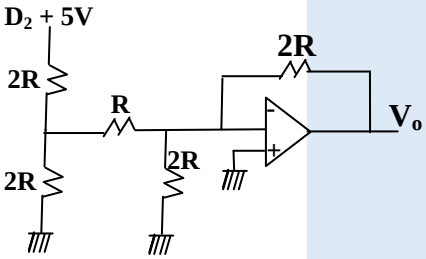
$$= \frac{-5}{2R} * 2R = -5 \text{Volt}$$

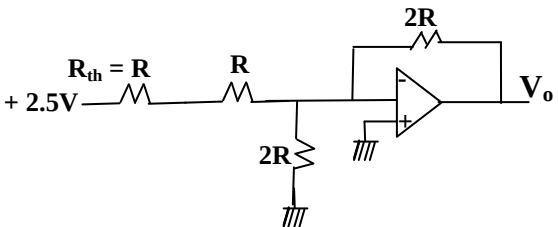


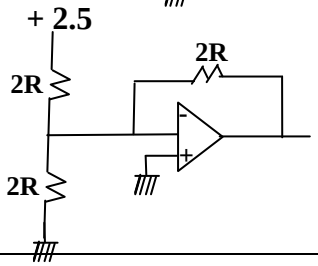
2) $D_2 = 1 \quad D_o = D_1 = D_3 = 0$

$$I_o = \frac{2.5}{2R} \qquad V_o = -i_o * Rf$$

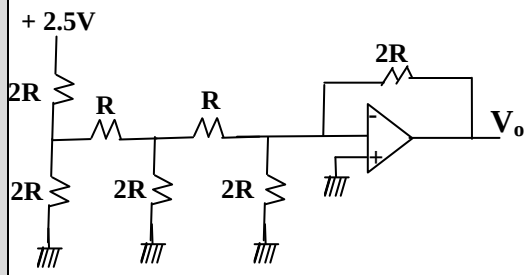
$$= \frac{-2.5}{2R} * 2R = 2.5 \text{Volt}$$

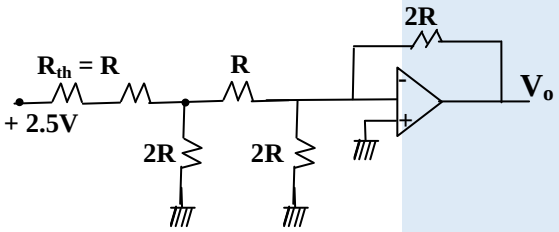






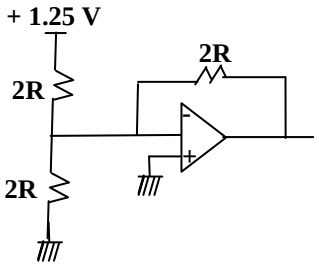
3) $D_1 = 1 \quad D_o = D_2 = D_3 = 0$





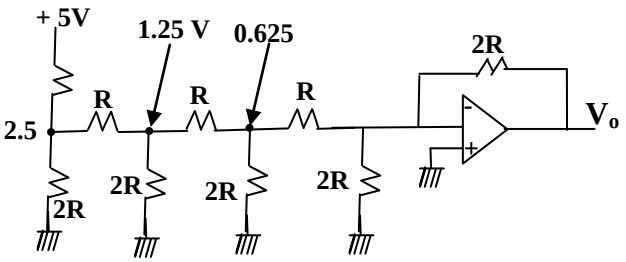
$$I_o = \frac{1.25}{2R} \Rightarrow V_o = \frac{-1.25}{2R} = 2R$$

$$= -1.25\,\text{Volt}$$



4) $D_o = 1 \quad D_1 = D_2 = D_3 = 0$

V_o = - 0.625 Volt



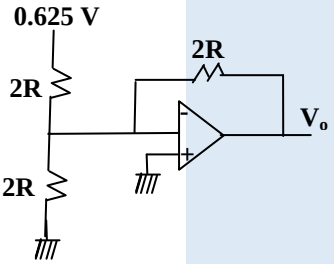
if $D_o = D_1 = D_2 = D_3 = 1$

V_o = (- 5) + (- 2.5) + (- 1.25) + (- 0.625)

=

if $D_o = D_3 = D_1 = 1, \; D_2 = 0$

V_o = -5 + - 1.25 + -0.625



Method (2):

$$V_o = -2\,\frac{V_{ref}}{2R} * D * Rf$$

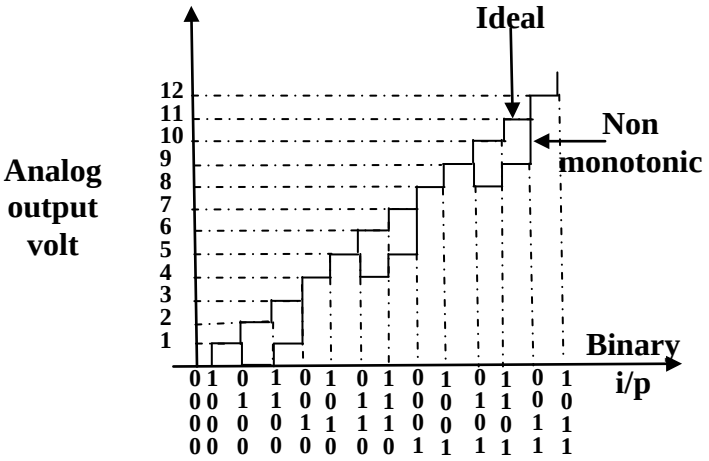
If $D_3 = 1 \quad D_1 = D_o = D_2 = 0 \qquad D = \frac{b_1}{2} + \frac{b_2}{4} + \frac{b_3}{8} + \frac{b_4}{16}$

$$V_o = -2*\frac{5}{2R}*\frac{1}{2}*2R \qquad = \frac{1}{2}+0+0+0$$

= - 5 Volt

4- DAC Errors

NON Monotonicity

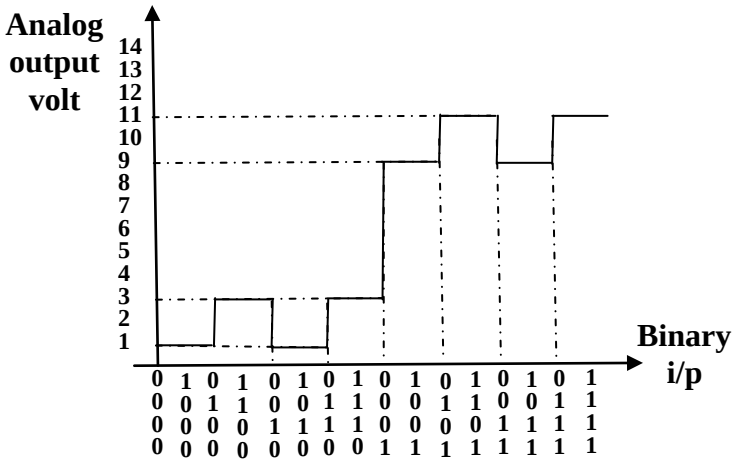


- In this particular case, the error occurs because 2^1 bit in binary code is "0" that is a short is causing the bit input-line to be stuck in low.

i/p				o/p (after)	o/p (before)
0	0	0	0	0	0
0	0	0	1	1	1
0	0	1	0	0	2
0	0	1	1	1	3
0	0	0	0	4	4
0	1	0	1	5	5
0	1	1	0	4	6
0	1	1	1	5	7
1	0	0	0	8	8
1	0	0	1	9	9
1	0	1	0	8	10
1	0	1	1	9	11
1	1	0	0	12	12

Example

A straight binary sequence is applied to a 4-bit DAC and the O/P in figure (1) is observed. What is the problem? Identify the error?



Solution

Binary I/P				Analog O/P	Correct
2^3	2^2	2^1	2^0		
0	0	0	0	→ 1	→ 0
0	0	0	1	→ 1	→ 1
0	0	1	0	→ 3	→ 2
0	0	1	1	→ 3	→ 3
0	1	0	0	→ 1	→ 4
0	1	0	1	→ 1	→ 5
0	1	1	0	→ 3	→ 6
0	1	1	1	→ 3	→ 7
1	0	0	0	→ 9	→ 8
1	0	1	0	→ 11	→ 10
1	0	1	1	→ 11	→ 11
1	1	0	0	→ 9	→ 12
1	1	0	1	→ 9	→ 13
1	1	1	0	→ 11	→ 14
1	1	1	1	→ 11	→ 15

2^o stuck in high

2^o stuck in low

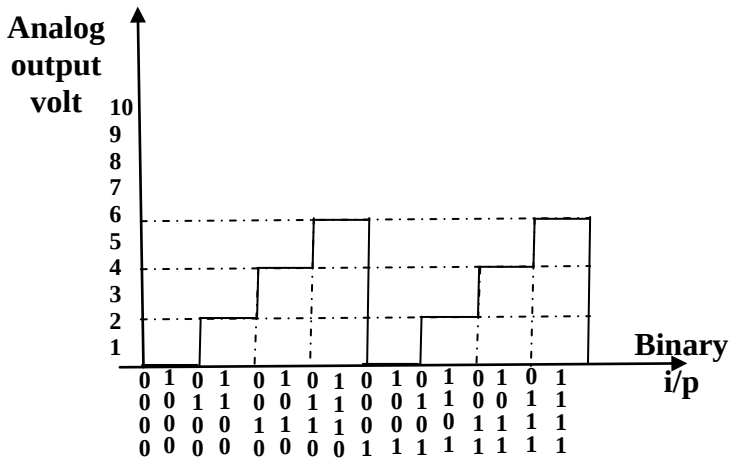
The error is non-monotonic

Example

If the first and the last bit are both stuck in low in a 4-bit DAC what will be the O/P? Draw it?

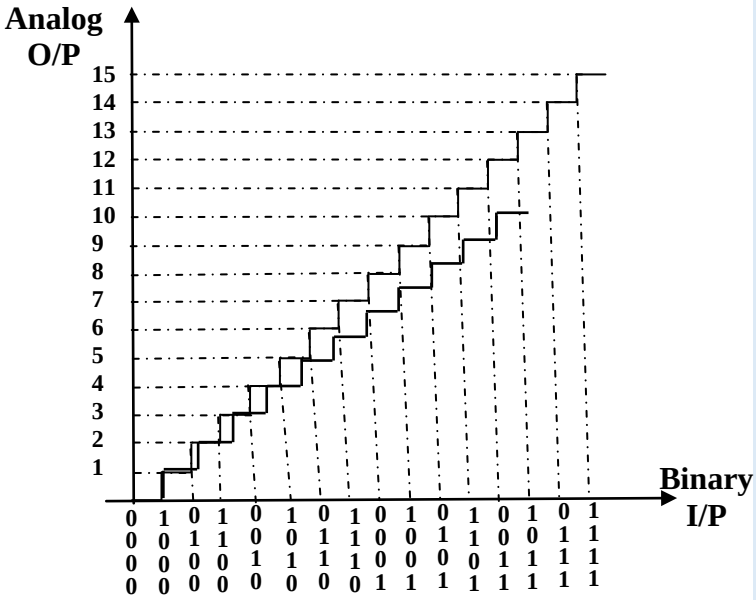
- Straight binary sequence that applied to.

Binary i/p	Analog o/p	Correct
0 0 0 0	0	0
0 0 0 1	0	1
0 0 1 0	2	2
0 0 1 1	2	3
0 1 0 0	4	4
0 1 0 1	4	5
0 1 1 0	6	6
0 1 1 1	6	7
1 0 0 0	0	8
1 0 0 1	0	9
1 0 1 0	2	10
1 0 1 1	2	11
1 1 0 0	4	12
1 1 0 1	4	13
1 1 1 0	6	14
1 1 1 1	6	15



2) Non Linearity

Figure shown illustrates non-linearity in which the step amplitude is less than it should be for certain input codes. This particular output could be caused by the 2^2 bit having an insufficient weight, perhaps because of a faulty input resistor. We could also see steps with amplitudes greater than normal if a particular binary weight were greater than it should be.

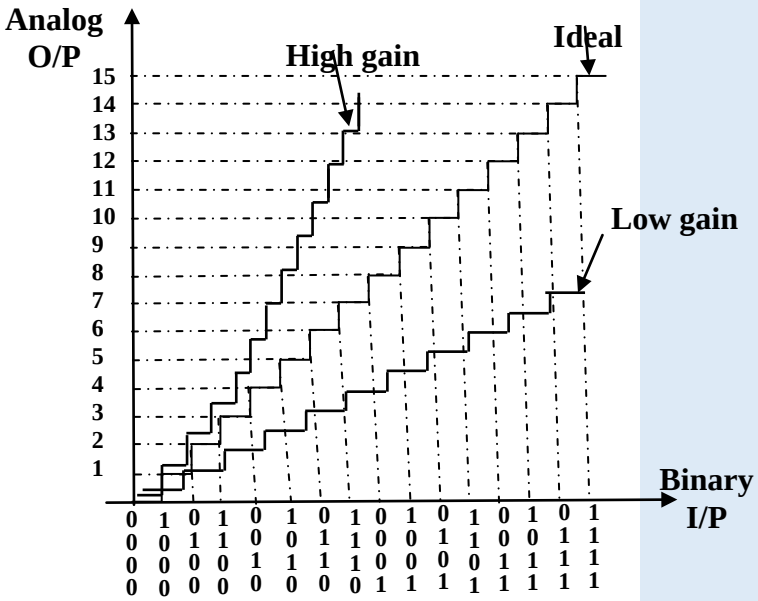


3) Low and High Gain

Output error caused by low or high gain are illustrated in figure.

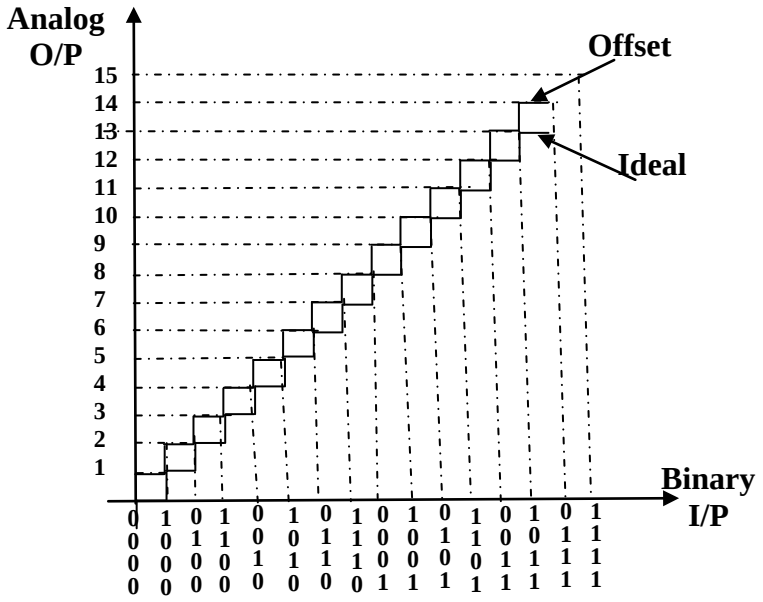
In this case of low gain all of the step amplitudes are less than ideal.

In this case of high gain, all of the step amplitudes are greater than ideal. This situation may be caused by a faulty feedback resistor in the OP. amp circuit.

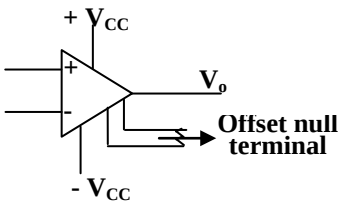


4) Offset Error

An offset error is illustrated in figure. Notice that when the binary input is (0000) the output voltage is nonzero and that this amount of offset is the same for all steps in the conversion. A faulty OP-amp be the culprit in this situation.



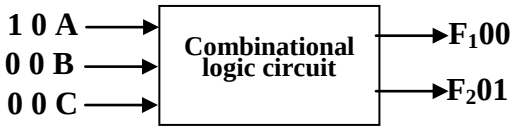
It is not possible to manufacture amplifier with perfectly matched transistor and resistors. The mismatch creates a dc. offset error in the O/P with no differential dc. i/p. The dc. o/p of an op-amp should ideally be zero volts with respect to ground. Any deviation from this is known as dc. offset error.

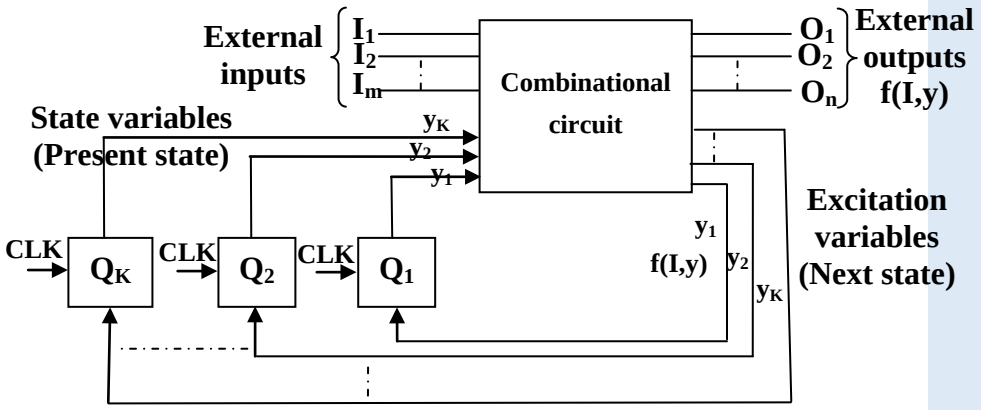


Lectures of Electrical Engineering Department

Subject Title: Digital Electronics

Class:3rd Electronics

Lecture Contents	**Lecture sequences:**	**Fourth lecture**	**Instructor Name:**
	The major contents: **1-Sequential Logic Circuits: Definition and Characteristics** **2- The types of sequential logic circuits** **3-General Design Procedure of Sequential Circuit**		
	The detailed contents: **1- Sequential Logic Circuits**  The logic circuits can be divided by two categories: 1) Combinational circuit is one where the O/P at any time depends upon only on the present combination of inputs at the point of time with total disregard. To the past state of the inputs. The logic gate is the most basic building block of combinational logic. 2) Sequential logic circuit: The output in it depends up on not only the present but also the past state of inputs. • Sequential circuit Comprises both logic gate and memory elements such as Flip-Flop.		



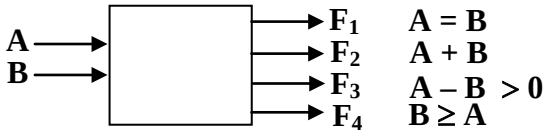
Block diagram of a sequential circuit (Synchronous)

- Present contents of memory elements is referred as the present state. The new contents of the memory elements are referred as next state and depend upon the external input and present state.
- Not all sequential circuit have i/p and O/P variables as in the general model just discussed. However, all have excitation variables and state variables. For example, N-bit gray code counter has no i/PS other than clock and no O/PS other than it's internal state (O/PS are taken off each F.F in the counter).

- **2-** Sequential circuits are divided into two main types:
  1) Synchronous sequential circuits: change their state and O/P values at discrete instants of time (clock signal). The memory elements used in synchronous circuits are usually Flip-Flop.
  2) Asynchronous: The transition from one state to another is initiated by the change in the primary i/ps, there is no external synchronization. The memory commonly used in it are time delayed devices.

3-General Design Procedure of Sequential Circuit

1) Combinational circuit



A	B	F_1	F_2	F_3	F_4
0	0	1	0	0	1
0	1	0	1	0	1
1	0	0	1	1	0
1	1	1	0	0	1

2) Sequential circuit

External i/p ($X_1, X_2, \ldots X_n$)

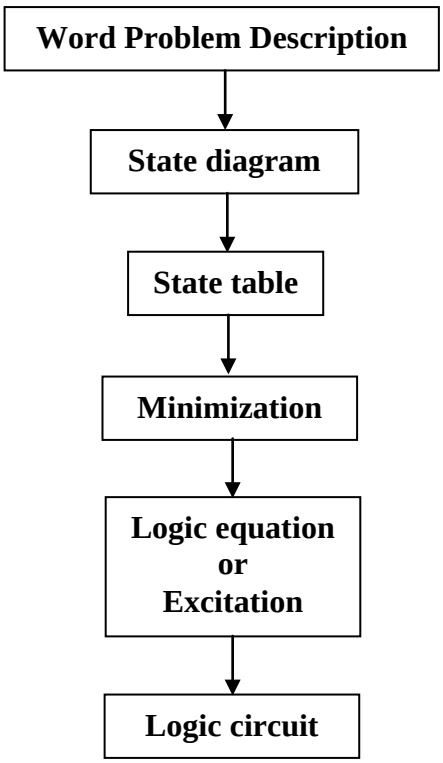
External o/p ($Z_1, Z_2, \ldots X_n$)

Present State ($y_1, y_2, \ldots y_K$)

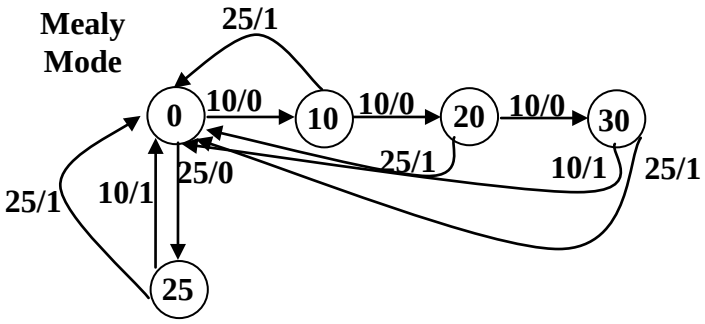
Next State ($Y_1, Y_2, \ldots Y_K$) } Memory

Examples

Construct the state diagram for a simplified circuit a soda vending Machine with a single input and single output. The input is one of two coins deposited (dimes or quarters). The O/P is a binary signal with "0" indicating no can of soda is released and 1 indicating a can is released. Assume the cost of a can of soda is 35 cent.

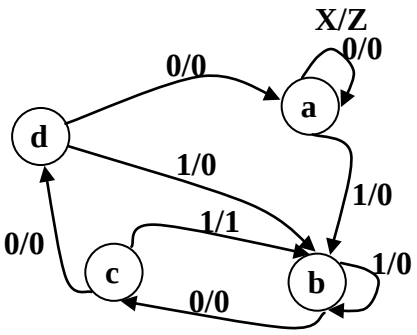


State diagram



تصميم الدوائر الالكترونية :

1- تنظم متطلبات التصميم على شكل مخطط يبين الانتقال من الحالة السابقة (P.S.) إلى الحالة اللاحقة (N.S.).
2- تحويل مخطط انتقال الحالات المتتالية إلى جدول يبين انتقال بين الحالات ويبين اخراجات باختلاف الحالات والادخالات.



3- عمل جدول تنظيمي (Implication Chart) لتقليص الحالات المتشابهة واختزالها إلى حالة واحدة.

P.S	NS/Z	
	X = 0	X = 1
a	a/o	b/o
b	c/o	b/o
c	d/o	b/1
d	a/o	b/o

4- نعيد كتابة الجدول من جديد بعد حالات الاختزال.

b	X		
c	X	X	
d	✓	a≠c	X
	a	b	c

P.S	NS/Z X = 0	X = 1
a	a/o	b/o
b	c/o	b/o
c	a/o	b/1

5- تحديد عدد المراجيح اللازمة في التصميم وفق المعادلة التالية.

For example

$2^{n-1} < NS \leq 2^n$

NS=3 when : NS: no. of states.

$\therefore n = 2$ $2 < 3 \leq 4$ n: no. of flop-flops

6- بعد اختيار عدد (F/F) نقوم بترميز الجدول.

7- تحديد نوع الـ T, RS, JK, D (F/F)

P.S	NS/Z X = 0	X = 1
00	00/0	01/0
01	11/0	01/0
11	00/0	01/1

P.S $Q_n(y)$	N.S $Q_{n+1}(y)$	D	J-K		R-S		T
0	0	0	0	X	X	0	0
0	1	1	1	X	0	1	1
1	0	0	X	1	1	0	1
						X	

Excitation table

Choose D-flip-flop in the design

8- نقوم بعمل خارطة كارنو لإيجاد (D_1 and D_2) و (Z) (كمعادلات).

P.S		X = 0		-		X = 1			
y_1	y_2	Y_1	Y_2	D_1	D_2	Y_1	Y_2	D_1	D_2
0	0	00/0		0	0	01/0		0	1
0	1	11/0		1	1	01/0		0	1
1	1	00/0		0	0	01/1		0	1

X \ y_1y_2	00	01	11	10	D_1
0	0	1	0	X	
1	0	0	0	X	

$$D_1 = \overline{X}\, y_1 \overline{y_2}$$

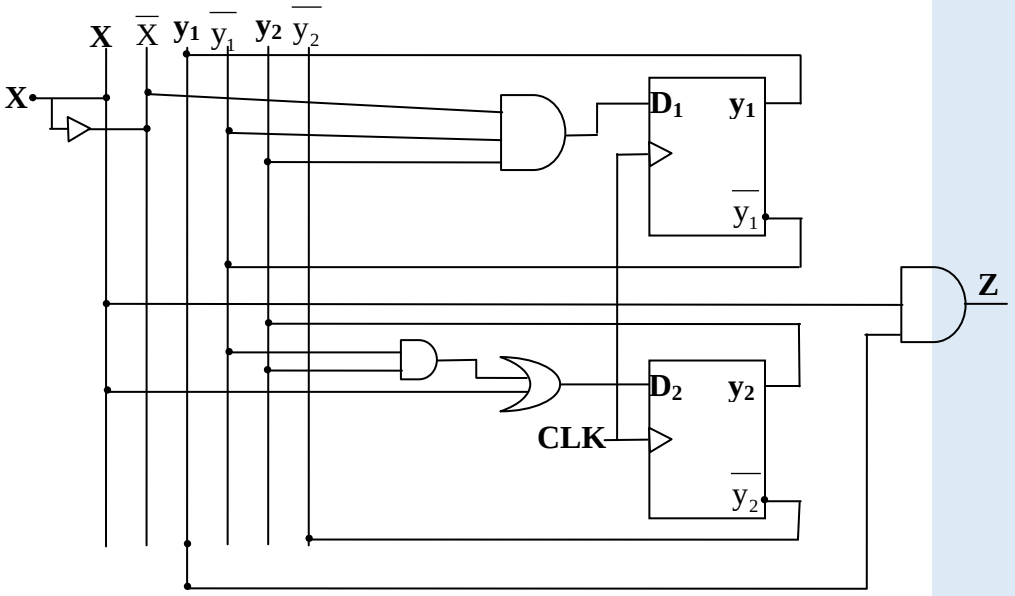
X \ y_1y_2	00	01	11	10	Z
0	0	0	0	X	
1	0	0	1	X	

$$Z = X\, y_1$$

X \ y_1y_2	00	01	11	10	D_2
0	0	1	0	X	
1	1	1	1	X	

$$D_2 = X + \overline{y_1}\, y_2$$

9- رسم الدائرة المنطقية.



Example

Minimize the transition table (State table) below and complete the design of the system (Use J-K-Flip-Flop) then draw the logic circuit.

B	X						
C	X	✓A=F					
D	~~A=F~~ ~~B=D~~	X	X				
E	X	~~A=F~~ ~~B=D~~	~~D=F~~	X			
F	✓A=F	X	X	~~B=D~~	X		
G	~~D=G~~ ~~A=F~~	X	X	~~A=F~~ ~~B=G~~ ~~D=G~~	X	~~A=F~~ ~~D=G~~	
H	X	~~B=C~~ ✓A=F	✓B=C	X	~~B=C~~ ~~D=F~~	X	X
	A	B	C	D	E	F	G

P.S	X_1X_2 00	X_1X_2 01	X_1X_2 10	X_1X_2 11
A	D/0	D/0	F/0	A/0
B	C/1	D/0	E/1	F/0
C	C/1	D/0	E/1	A/0
D	D/0	B/0	A/0	F/0
E	C/1	F/0	E/1	A/0
F	D/0	D/0	A/0	F/0
G	G/0	G/0	A/0	A/0
H	B/1	D/0	E/1	A/0

B = C = H

A = F

B = H

H = C

P.S	X_1X_2 00	X_1X_2 01	X_1X_2 10	X_1X_2 11
A	D/0	D/0	A/0	A/0
B	B/1	D/0	E/1	A/0
D	D/0	B/0	A/0	A/0
E	B/1	A/0	E/1	A/0
G	G/0	G/0	A/0	A/0

y_1	y_2	y_3	$X_1X_2 = 0$ $Y_1Y_2Y_3/Z$	$X_1X_2 = 01$ $Y_1Y_2Y_3/Z$	10 $Y_1Y_2Y_3/Z$	11 $Y_1Y_2Y_3/Z$
0	0	0	011/0	011/0	000/0	000/0
0	0	1	001/1	011/0	111/1	000/0
0	1	1	011/0	001/0	000/0	000/0
1	1	1	001/1	000/0	111/1	000/0
1	1	0	110/0	110/0	000/0	000/0

			00		01		10		11	
y_1	y_2	y_3	Y_1	J_1K_1	Y_1	J_1K_1	Y_1	J_1K_1	Y_1	J_1K_1
0	0	0	1	1X	1	1X	0	0X	0	0X
0	0	1	0	0X	1	1X	1	1X	0	0X
0	1	1	1	X0	0	X1	0	X1	0	X1
1	1	1	0	X1	0	X1	1	X0	0	X1
1	1	0	1	X0	1	X0	0	X1	0	X1

$\overline{X_1}=0$

y_1y_2 \ y_3X_2	00	01	11	10
00	0	0	0	0
01	X	X	0	0
11	X	X	X	X
10	X	X	X	X

$X_1=1$

y_1y_2 \ y_3X_2				
	0	0	0	1
	X	X	0	0
	X	X	X	X
	X	X	X	X

$$J_1=X_1\,\overline{X_2}\,y_3\,\overline{y_2}$$

$\overline{X_1}=0$

y_1y_2 \ y_3X_2				
	X	X	X	X
	X	X	X	X
	0	0	1	1
	X	X	X	X

$X_1=1$

y_1y_2 \ y_3X_2				
	X	X	X	X
	X	X	X	X
	1	1	1	0
	X	X	X	X

$$K_1=\overline{X_1}\,y_3+X_1\,\overline{y_3}+X_1X_2$$

			00		01		10		11	
y_1	y_2	y_3	Y_1	J_1K_1	Y_1	J_1K_1	Y_1	J_1K_1	Y_1	J_1K_1
0	0	0	1	1X	1	1X	0	0X	0	0X
0	0	1	0	0X	1	1X	1	1X	0	0X
0	1	1	1	X0	0	X1	0	X1	0	X1
1	1	1	0	X1	0	X1	1	X0	0	X1
1	1	0	1	X0	1	X0	0	X1	0	X1

$\overline{X_1}=0$

y_3X_2 / y_1y_2	00	01	11	10
00	1	1	1	0
01	X	X	X	X
11	X	X	X	X
10	X	X	X	X

$X_1=1$

y_3X_2 / y_1y_2				
00	0	0	0	1
01	X	X	X	X
11	X	X	X	X
10	X	X	X	X

$$J_2=X_1y_3\overline{X_2}+\overline{X_1}\,\overline{y_3}+\overline{X_1}X_2$$

y_3X_2 / y_1y_2				
00	X	X	X	X
01	X	X	1	0
11	0	0	1	1
10	X	X	X	X

y_3X_2 / y_1y_2				
00	X	X	X	X
01	X	X	1	1
11	1	1	1	0
10	X	X	X	X

$$K_2=y_3X_2+\overline{y_1}\,y_2X_1+X_2\,\overline{y_3}+\overline{X_1}\,y_3\,y_1$$

(P.S)			X_1X_2 = 00		01		10		11	
y_1	y_2	y_3	Y_3	J_3K_3	Y_3	J_3K_3	Y_3	J_3K_3	Y_3	J_3K_3
0	0	0	1	1X	1	1X	0	0X	0	0X
0	0	1	1	X0	1	X0	1	X0	0	X1
0	1	1	1	X0	1	X0	0	X1	0	X1
1	1	1	1	X0	0	X1	1	X0	0	X1
1	1	0	0	0X	0	0X	0	0X	0	0X

$\overline{X_1}=0$

y_3X_2 / y_1y_2				
00	1	1	X	X
01	X	X	X	X
11	0	0	X	X
10	X	X	X	X

$X_1=1$

y_3X_2 / y_1y_2				
00	0	0	X	X
01	X	X	X	X
11	0	0	X	X
10	X	X	X	X

$$J_3=\overline{X_1}\,\overline{y_1}$$

y_1y_2 \ y_3X_2	00	01	11	10
00	X	X	0	0
01	X	X	0	0
11	X	X	1	0
10	X	X	X	X

y_1y_2 \ y_3X_2	00	01	11	10
00	X	X	1	0
01	X	X	1	1
11	X	X	1	0
10	X	X	X	X

$$K_3 = X_2\, y_1 + X_1 X_2 + X_1 \overline{y_1}\, y_2$$

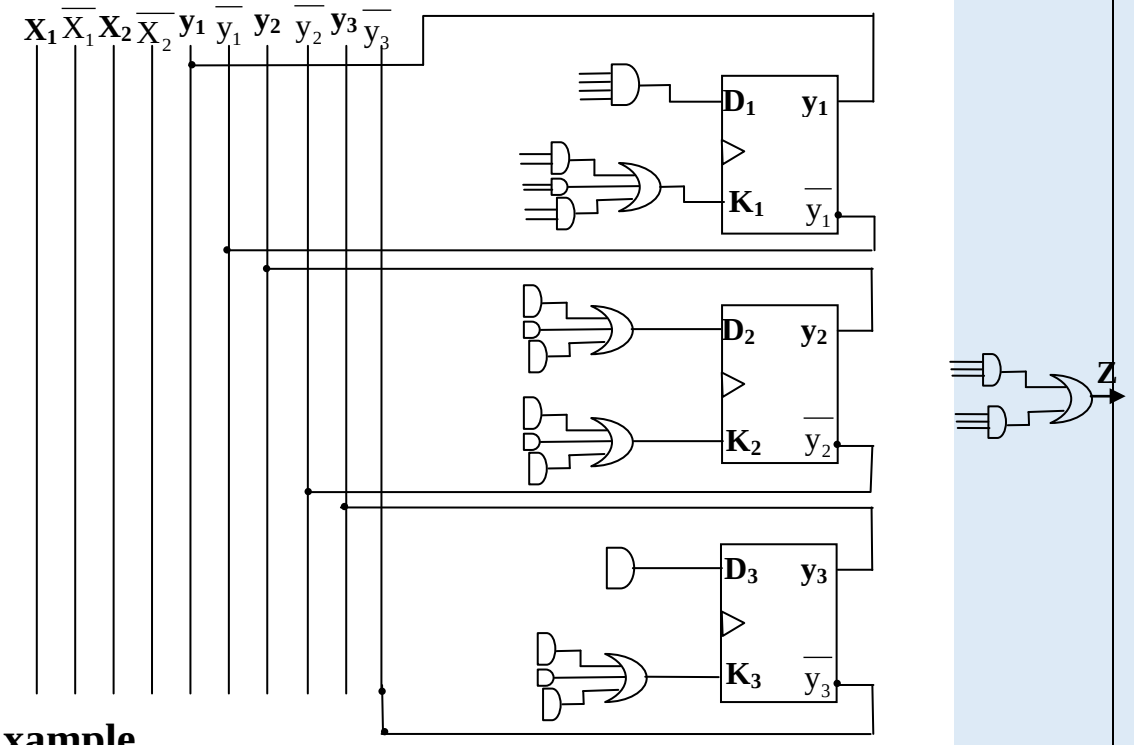
$\overline{X_1} = 0$

y_1y_2 \ y_3X_2	00	01	11	10
00	0	0	0	1
01	X	X	0	0
11	0	0	0	1
10	X	X	X	X

$X_1 = 1$

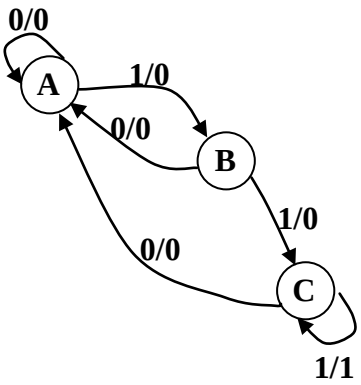
y_1y_2 \ y_3X_2	00	01	11	10
00	0	0	0	1
01	X	X	0	0
11	0	0	0	1
10	X	X	X	X

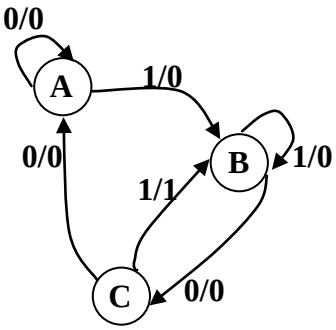
$$Z = y_1\, y_3\, \overline{X_2} + \overline{y_2}\, y_3\, \overline{X_2}$$

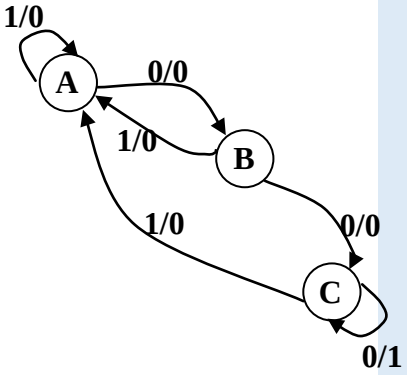


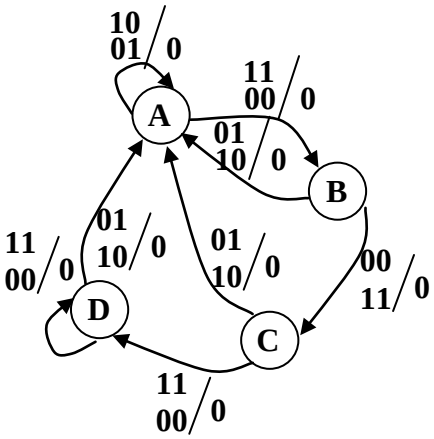
Example

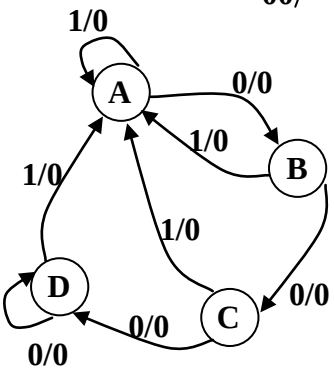
Design a sequential logic circuit to detect the sequence (111) whenever occur. Draw the circuit. Using T-F.F.

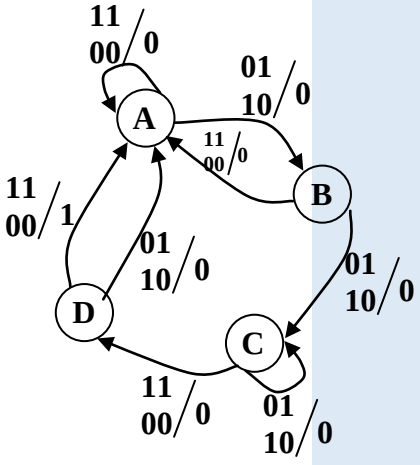












Example

Design a sequential logic circuit under the following condition:

1) The circuit will get O/P (1) for every Group of four pulses that starts with zero and ends at (1) but the second bit is the similar to third bit.

	2) The analysis of the first group, which will began when it is starts the first zero in the sequence of input. 3) The analysis of the next group will not start until the end of the group of four pulses whether it provides an output or not. 4) Use any F.F in the design.

Lectures of Electrical Engineering Department

Subject Title: Digital Electronics

Class: 3rd Electronics

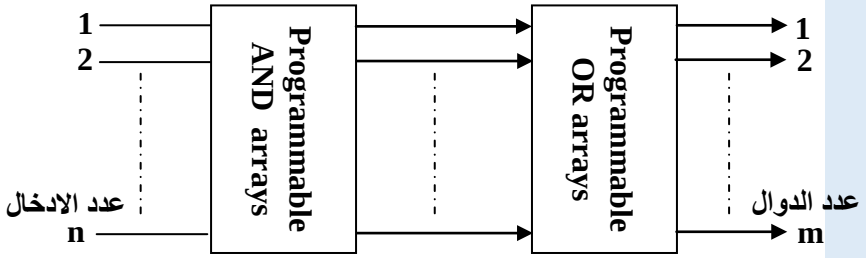
Lecture Contents	**Lecture sequences:**	**Fifth lecture**	**Instructor Name:**
	The major contents: **1-Function Implementation Using PLD'S** **2-Types of PLD'S** **3-Design of PLD'S**		
	The detailed contents: **1- Function Implementation Using PLD'S** **PLD'S:** Programmable Logic Devices. **PLD'S:** A logic devices is one in which the logic function is programmed by the user. In some cases can be reprogrammed many times. **2- Types of PLD'S** 1- **SPLD**: (Simple PLD) are the least complex form of PLD'S. The categories of SPLD'S are: **PAL:** (Programmable Array Logic). **GAL:** (Generic Array Logic) **PLA:** (Programmable Logic Array) **PROM:** (Programmable Read-only Memory) 2- **CPLD:** (Complex PLD) have a much higher capacity than SPLD, Permitting more complex logic circuits to be programmed into them. 3- **FPGAs:** (Filed Programmed Gate Arrays) are different from SPLD and CPLD.		

FPGAs: Consists of an array of anywhere from 64 to thousands of logic gate groups.

PLA

Is an "SPLD" that consists of a programmable "AND" array and programmable "OR" arrays.

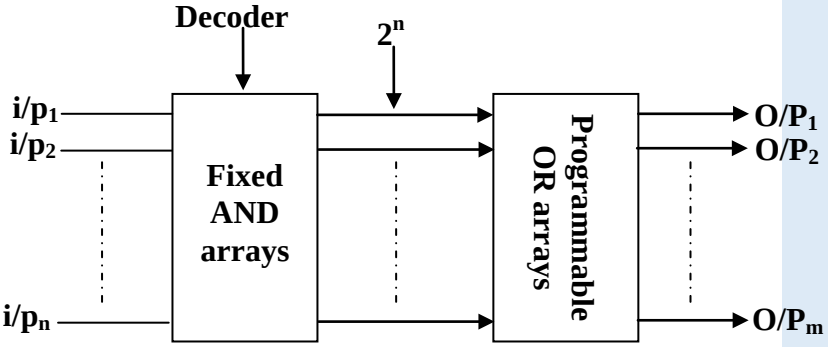
The PLA is called an (FPLA) (Filed Programmable Logic Array) because the user is the field not the manufacture programs it.



Block Diagram of a PLA

PROM

Consists of a set of fixed (non programmable) AND gates connected as a decoder and programmable OR arrays. Each AND gate has "n" inputs and each OR gate has (2^n) inputs. The no. of OR gates is equal to the word size (m) that stored in memory.



Block Diagram of a PROM

no. of AND = 2^n

no. of input of each AND = n

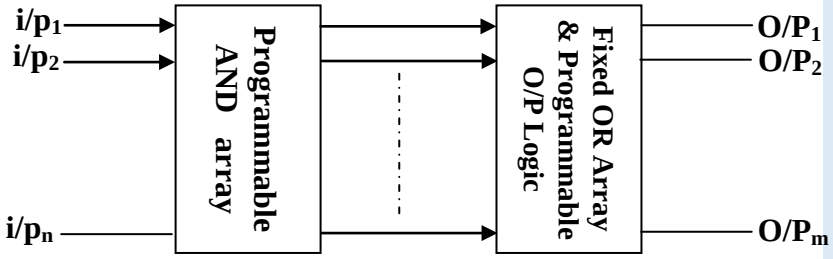
no. of OR = no. of word size = m

no. of input in each OR gate = no. of AND gate

GAL

Has a reprogrammable AND array and a fixed OR array with programmable O/P logic. The two main differences between GAL and PAL devices are (a) GAL is reprogrammable (b) GAL has programmable output.

The GAL can be reprogrammed again and again because it uses E^2CMOS technology instead of bipolar technology and fusible links.



Block Diagram of a GAL

E^2CMOS

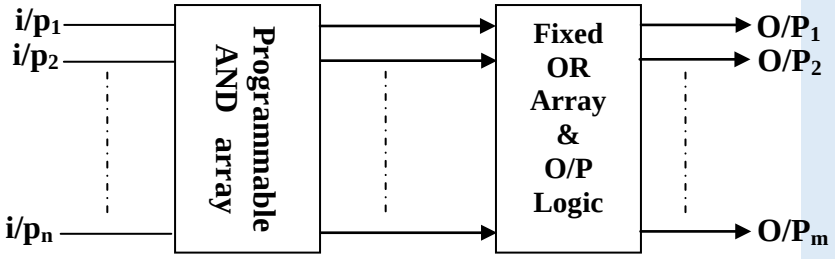
Electrically erasable complementary metal oxide semiconductor.

PAL

A PAL is an SPLD that was developed to overcome certain disadvantages of the PLA, such as longer-delays due to the additional fusible links that result from using two programmable arrays and more circuit complexity but is not as flexible as the PLA. The PAL is the most common one-time programmable (OTP) logic device and is implemented with bipolar technology (TTL or ECL).

ECL: Emitter Coupled Logic.

TTL: Transistor Transistor Logic.



Block Diagram of a PAL

increment its digital O/P from one Code to the Next higher code. An n-bit ADC can resolve one part in (2^n-1). It may be expressed as a percentage of full scale or in bits. The resolution of 8-bit ADC can be expressed as one part in 256 or as 0.39% of full scale or simply as 8-bit resolution.

3- Example

Implement the following functions by using (1) PROM (2) PLA technique (AND-OR-NOT).

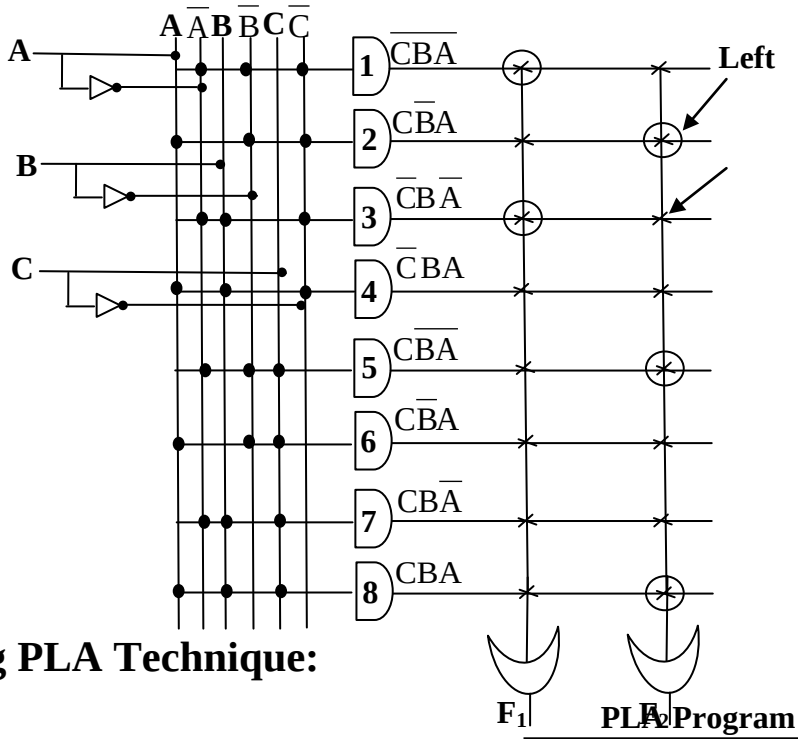
$$F_1 = \sum 0,2, \qquad F_2 = \sum 1,4,7$$

Solution

(*) Indicates a hard-wired interconnection.

(x) Indicates an intact (or unprogrammed) fusible link or interconnection.

1) PROM



By Using PLA Technique:

C \ BA	00	01	11	10
0	1	0	0	1
1	0	0	0	0

$F_1 = \overline{A}\,\overline{C}$

$\overline{F_1} = A + C$

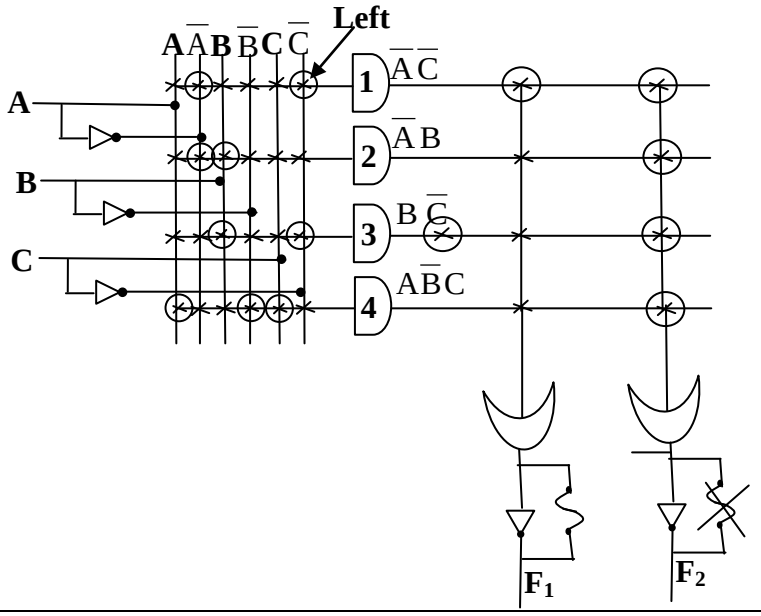
C \ BA	00	01	11	10
0	0	1	0	0
1	1	0	1	0

$F_2 = A\overline{B}\overline{C} + \overline{A}B\,C + ABC$

$\overline{F_2} = \overline{A}B + B\,\overline{C} + \overline{C}\,\overline{A} + A\overline{B}C$

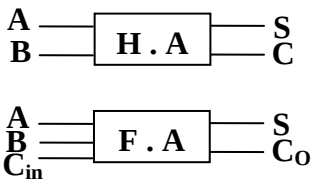
PLA Program Table

Product term		Inputs C	B	A	Outputs F_1	$\overline{F_2}$
$\overline{A}$	$\overline{C}$	0	-	0	1	1
$\overline{A}$	B	-	1	0	-	1
B	$\overline{C}$	0	1	-	-	1
A	$\overline{B}$ C	1	0	1	-	1



Example

Show the logic arrangement of PLA (AND-OR) required to implement binary full adder.



A \ BCi	00	01	11	10
0	0	1	0	1
1	1	0	1	0

$$S = \bar{A}\bar{B}C_i + \bar{A}B\bar{C}_i + A\bar{B}\bar{C}_i + ABC_i$$

$$\bar{S} = \bar{A}\bar{B}\bar{C}_i + A\bar{B}C_i + \bar{A}BC_i + AB\bar{C}_i$$

A	B	C_{in}	S_m	C_O
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

A \ BCi	00	01	11	10
0	0	0	1	0
1	0	1	1	1

$$C_o = AC_i + AB + BC_i$$

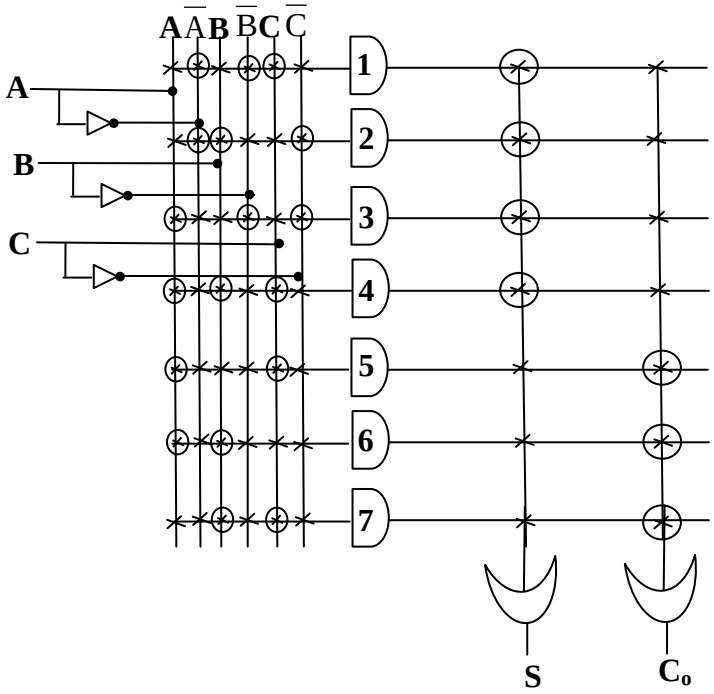
$$\overline{C_o} = \bar{A}\bar{B} + \bar{B}\bar{C}_i + \bar{B}\bar{C}_i$$

PLA (AND – OR – NOT) ⟵ **F**

PLA (AND – OR) ⟵ **F**

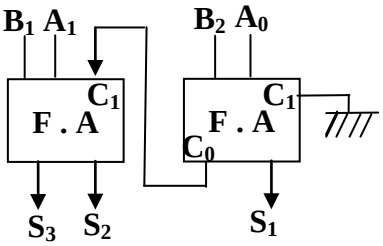
Product ???			Inputs A	B	C_i	Outputs S	C_o
$\bar{A}$	$\bar{B}$	C_i	0	0	1	1	-
$\bar{A}$	B	$\bar{C}_i$	0	1	0	1	-
A	$\bar{B}$	$\bar{C}_i$	1	0	0	1	-
A	B	C_i	1	1	1	1	-
A	C_i		1	-	1	-	1
A	B		1	1	-	-	1
B	C_i		-	1	1	-	1

PLA Program Table



Example

Construct a large circuit using PLA (AND-OR-NOT) and PAL that adds two binary numbers (A and B). Each binary number is composed of two bits (i.e. A = A_1A_0, B = B_1B_0),



B_1	B_0	A_1	A_0	S_3	S_2	S_1
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	1	0
0	0	1	1	0	1	1
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	0	1	1
0	1	1	1	1	0	0
1	0	0	0	0	1	0
1	0	0	1	0	1	1
1	0	1	0	1	0	0
1	0	1	1	1	0	1
1	1	0	0	0	1	1
1	1	0	1	1	1	0
1	1	1	1	1	0	1
1	1	1	1	1	1	0

A_1A_0 / B_1B_0

B_1B_0 \ A_1A_0	00	01	11	10
00	0	1	1	0
01	1	0	0	1
11	1	0	0	1
10	0	1	1	0

$$S_1 = \overline{B_0} A_0 + B_0 \overline{A_0}$$

A_1A_0 / B_1B_0

B_1B_0 \ A_1A_0	00	01	11	10
00	0	0	1	1
01	0	1	0	1
11	1	0	1	0
10	1	1	0	0

$$S_2 = \overline{B_0}\ \overline{B_1}\ A_1 + A_1 \overline{A_0}\ \overline{B_1} + \overline{A_1} A_0 \overline{B_1} B_0 + A_0 A_1 B_0 B_1 + \overline{B_0} B_1 \overline{A_1} + \overline{A_0}\ \overline{A_1} B_1$$

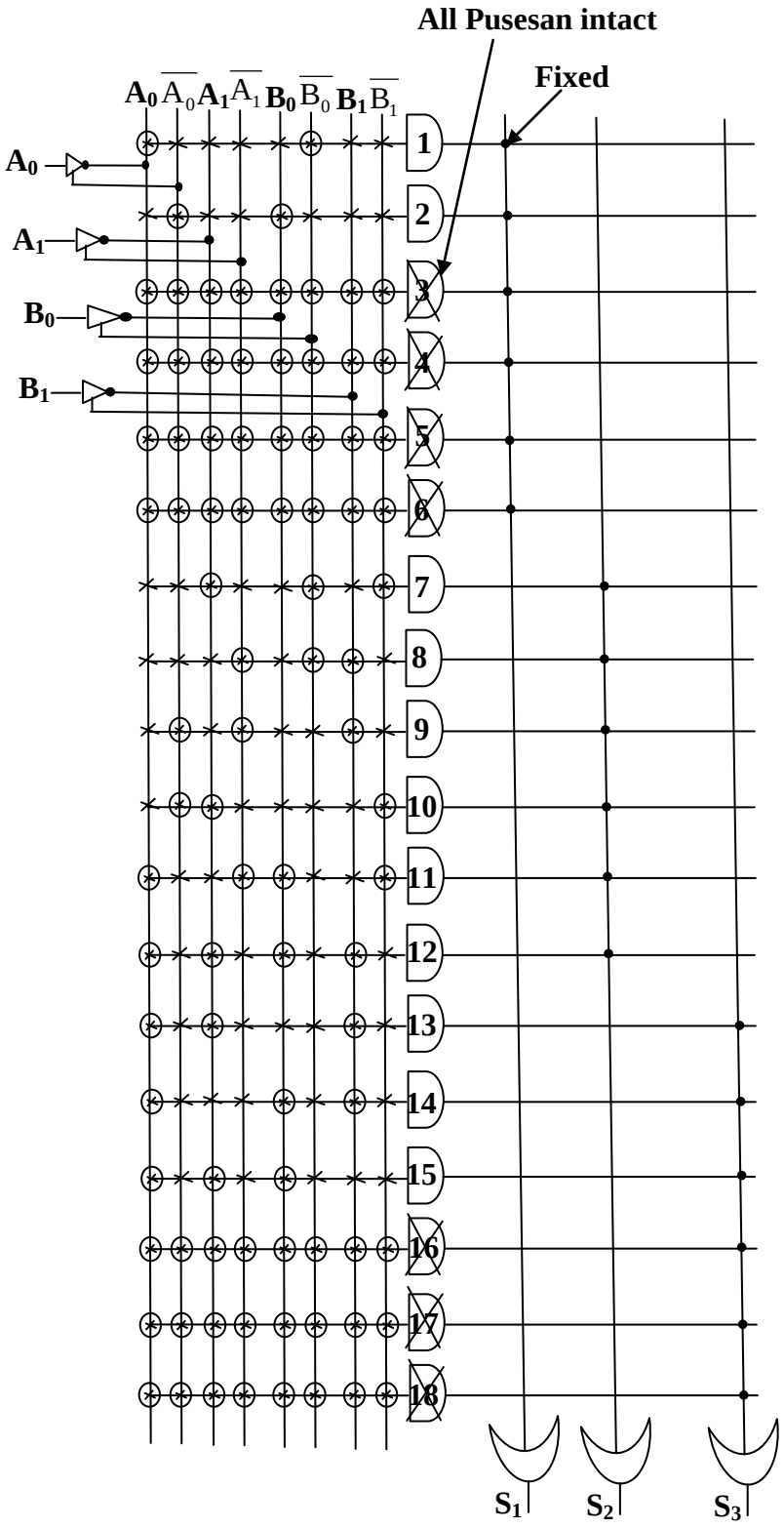
B_1B_0 \ A_1A_0	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	0	1	1	1
10	0	0	1	1

$$S_3 = B_1 A_1 + A_0 A_1 B_0 + B_0 B_1 A_0$$

[o. of AND gate = no. of outputs * max. product terms in the logic equation for the output = 3 × 6 = 18

PAL Program table

Product terms		AND i/pS B_1	B_0	A_1	A_0	Output
1.	$\overline{B_0}$ A_0	-	0	-	1	$S = \overline{B_0} A_0 + B_0 \overline{A_0}$
2.	B_0 $\overline{A_0}$	-	1	-	0	
3.	———	-	-	-	-	
4.	———	-	-	-	-	
5.	———	-	-	-	-	
6.	———	-	-	-	-	
7.	A_1 $\overline{B_0}$ $\overline{B_1}$	0	0	1	-	$S_2 = A_1 \overline{B_0}\,\overline{B_1} +$
8.	B_1 $\overline{B_0}$ $\overline{A_1}$	1	0	0	-	$B_1 \overline{B_0} A_1 + \overline{A_0}\,\overline{A_1} B_1$
9.	$\overline{A_0}$ $\overline{A_1}$ B_1	1	-	0	0	$+ A_1 \overline{A_0}\,\overline{B_1} + A_0 \overline{A_1} B_0 \overline{B_1}$
10.	A_1 $\overline{A_0}$ $\overline{B_1}$	0	-	1	0	$+ A_0 A_1 B_0 B_1$
11.	A_0 $\overline{A_1}$ B_0 $\overline{B_1}$	0	1	0	1	
12.	$A_0 A_1 B_0$ B_1	1	1	1	1	
13.	B_1 A_1	1	-	1	-	$S_3 = B_1 A_1$
14.	B_0 B_1 A_1	1	1	-	1	$+ B_0 B_1 A_0$
15.	A_0 A_1 B_0	-	1	1	1	$+ A_0 A_1 B_0$
16.	———	-	-	-	-	
17.	———	-	-	-	-	
18.	———	-	-	-	-	


All Pusesan intact
Fixed
A_0 $\overline{A_0}$ A_1 $\overline{A_1}$ B_0 $\overline{B_0}$ B_1 $\overline{B_1}$
A_0
A_1
B_0
B_1
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
S_1
S_2
S_3

Example

Use PAL to implement the following functions:

$$A(x,y,z)=\sum(1,2,4,6)$$

$$B(x,y,z)=\sum(0,1,3,6,7)$$

$$C(x,y,z)=\sum(1,2,4,6,7)$$

$$D(x,y,z)=\sum(1,2,3,5,7)$$

After using K – map, the equations for the outputs will be as following:

$$A=(X\overline{Z}+Y\overline{Z}+\overline{X}\,\overline{Y}\,Z)$$

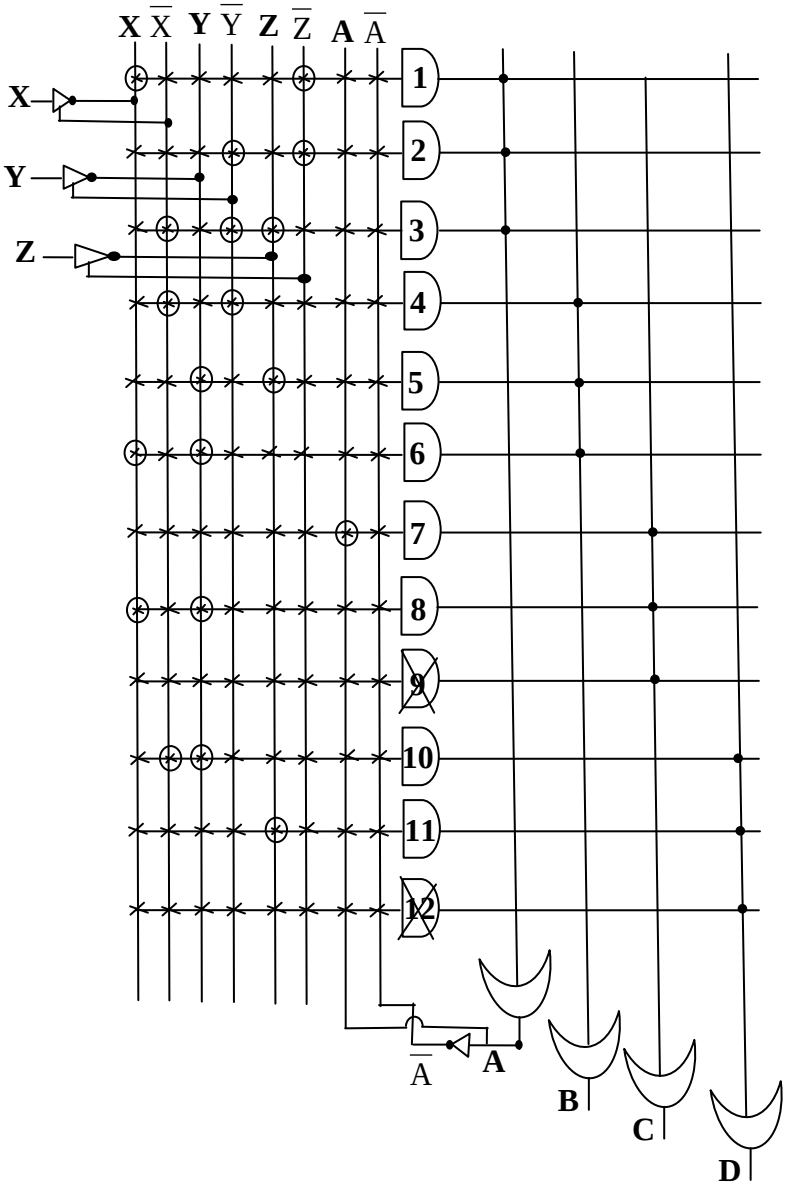
$$B=(\overline{X}\,\overline{Y}+YZ+XY)$$

$$C=(X\overline{Z}+Y\overline{Z}+\overline{X}\,\overline{Y}\,Z+XY)\Rightarrow C=A+XY$$

$$D=(\overline{X}\,Y+Z)$$

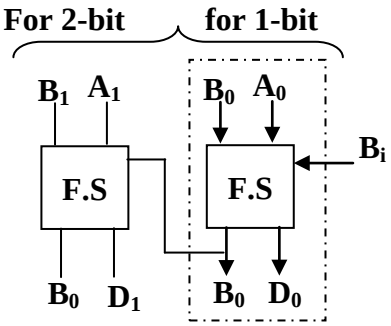
Program PAL Table

Product terms		Input X	Y	Z	A	Output
1.	$X\ \overline{Z}$	1	-	0	-	$A=X\overline{Z}+Y\overline{Z}+\overline{X}\,\overline{Y}\,Z$
2.	$Y\ \overline{Z}$	-	1	0	-	
3.	$\overline{X}\ \overline{Y}\ Z$	0	0	1	-	
4.	$\overline{X}\ \overline{Y}$	0	0	-	-	$B=\overline{X}\,\overline{Y}+YZ+XY$
5.	$Y\ Z$	-	1	1	-	
6.	$X\ Y$	1	1	-	-	
7.	A	-	-	-	1	$C=A+XY$
8.	$X\ Y$	1	1	-	-	
9.	———	-	-	-	-	
10.	$\overline{X}\ Y$	0	1	-	-	$D=\overline{X}Y+Z$
11.	Z	-	-	1	-	
12.	———	-	-	-	-	



Full Substructure

A	B	B_i	D	B_0
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1



A \ BB_i	00	01	11	10
0		1		1
1	1		1	

$$D = A\overline{B}\,\overline{B_i} + \overline{A}\overline{B}B_i + ABB_i + \overline{A}B\overline{B_i}$$

A \ BB_i	00	01	11	10
0		1	1	1
1			1	

$$B_o = \overline{A}B + \overline{A}B_i + BB_i$$
