ACS COLLEGE OF ENGINEERING

Kambipura, Mysore Road, Bengaluru – 560074

DEPARTMENT OF AEROSPACE ENGINEERING

AVIONICS AND INSTRUMENTATION LAB MANUAL

(18ASL77)

(Prescribed for VII – Semester Aerospace Engineering)

ACADEMIC YEAR 2021 – 2022

NAME OF THE FACULTY : _____

BRANCH : AEROSPACE ENGINEERING

SEMESTER &YEAR : VII & IV

ACADEMIC YEAR : 202 - 202

EXP.NO.	CALIBRATION AND MEASUREMENT WITH AIRSPEED INDICATOR
DATE:	

AIM

To Calibrate and measure the airspeed indicator.

APPARATUS REQUIRED

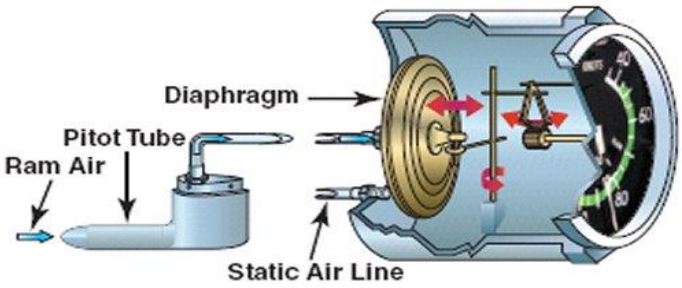
- Flight simulator
- Prepare 3D – Software

THEORY

Airspeed indicator, instrument that measures the speed of an aircraft relative to the surrounding air, using the differential between the pressure of still air (static pressure) and that of moving air compressed by the craft's forward motion (ram pressure); as speed increases, the difference between these pressures increases as well. Pressures are measured by a Pitot tube, a U-shaped apparatus with two openings, one perpendicular to the flow of air past the aircraft and one facing directly into the flow.

Air speed Indicator

Mercury or a similar liquid fills the bend in the tube, forming parallel columns balanced by the air pressure on each side. When static and ram pressure are equal, the columns have the same height. As the ram pressure increases, mercury on that side of the tube is pushed back and the columns become imbalanced. The difference between the two columns can be calibrated to indicate the speed; this value, called the indicated airspeed, may be given in knots, miles per hour, or other units. Since the airspeed indicator is calibrated at standard temperature and pressure, its readings are inaccurate at different temperatures and altitudes. An (uncorrected) indicated airspeed is still used to estimate an aircraft's tendency to stall. Instruments that electronically correct for altitudinal differences and temperature give the true airspeed, which is used to calculate the aircraft's position



TYPES OF AIRSPEEDS

Calibrated

Speed corrected for installation and instrument errors.

At high angle of attack, the pitot tube does not point straight into the relative wind, this tends to make the airspeed indicate lower than normal at low airspeeds.

Not usually a problem in cruise, usually we only worry about calibrate airspeed when we are converting to true airspeed.

True

The actual speed of your airplane is moving through undisturbed air.

On a standard day, Calibrated airspeed will be equal to TAS.

As density altitude increases, true airspeed increases for a given CAS or amount of power.

TAS can be calculated by using CAS with temperature and pressure on your E6B

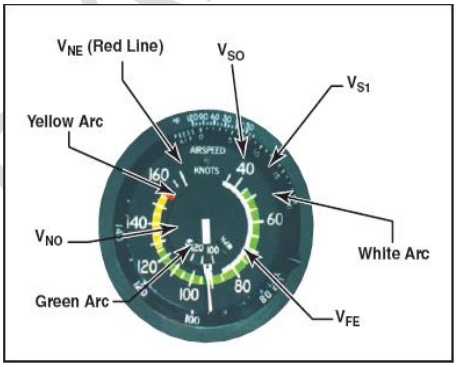
Equivalent

Calibrated airspeed corrected for adiabatic compressible flow at a particular altitude.

Above 200 kts and 20,000 feet air compresses in front the pitot tube causing abnormally high airspeeds.

Many flight computers are designed to compensate.

AIRSPEED INDICATOR SYMBOLOGY



White Arc – Flap operating Range
Green Arc – Normal Operations
Yellow Arc – Caution Area (Only use in smooth air)
Red Line – Never Exceed Speed
Vso – Stall speed in landing configuration
Vs – Stall speed in clean (flaps up) configuration
Vy – Climb speed for the max amount of height v. time
Vx- Climb speed for the max amount of height for distance
Vfe- Flap Extension speed: Flaps should not be used above this speed

Va – Design maneuvering/rough air speed: Speed at which abrupt full control inputs can be used without risking structural damage. Should never be exceeded in rough air. Changes with weight
Vno – Max structural Cruise speed
Vne – Never Exceed Speed.

CALIBRATION OF AIRSPEED INDICATOR IN FLIGHT SIMULATOR

An ASI is an extremely sensitive pressure gauge that must undergo the correct form of calibration to avoid damage to the instrument and inaccurate results. Due to the sensitivity of the gauge, it is in your best interest to have a professional calibrate your ASI. There have been many instances reported of individuals simply blowing into the ASI to apply pressure. This should NEVER be done and could cause serious damage and expensive consequences. If you choose to calibrate your ASI by using our Flight simulator, consider the following steps:

PROCEDURE

STEP 1: Switch on the Flight simulator
STEP 2: Run the P3D application (Ensure throttle in Ideal condition)
STEP 3: Synchronize the data from physical instrument to onscreen
STEP 4: Measure and record the initial temperature and pressure at the mean sea-level
STEP 5: Initiate the engine and increase the throttle at a required pace.
STEP 6: Reach the required Altitude in the given time
STEP 7: Record all three air speeds
STEP 8: Land the Flight and Check all inflight conditions

AIRSPEED INDICATOR ERRORS

- If the Pitot tube is blocked and the drain is open, speed will go to zero.
- If the Pitot tube is blocked and the drain is open, it will act as an altimeter.
- If the the Static vent is blocked, the airspeed will read higher than it should above altitude where it became blocked and lower than it should below.
- If all three all blocked, the needle will freeze.

RESULT

Thus, the air speed indicator is measured and calibrated.

EXP.NO.	CALIBRATION AND MEASUREMENT WITH ALTIMETER INDICATOR
DATE:	

AIM

To Calibrate and measure the altimeter indicator.

APPARATUS REQUIRED

- Flight simulator
- Prepare 3D – Software

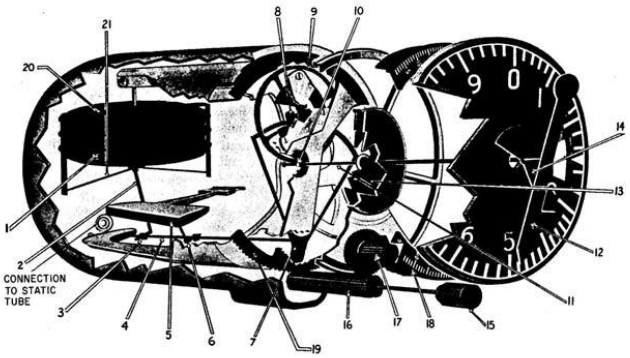
THEORY

It is one of the most important instruments especially while flying in conditions of poor visibility.

Altitude must be known for calculating other key parameters such as engine power, airspeed etc.

Altimeter works on the principle of barometer. In a sensitive altimeter there are three diaphragm capsule with two or three different dials each indicating different slab of altitude.

Altimeter should be compensated for atmosphere pressure change.



1. Diaphragm assembly
2. Diaphragm link
3. Diaphragm calibrating arm
4. Rocking shaft
5. Balance
6. Balanced connecting link and arm
7. Sector gear
8. Intermediate pinion gear and hairspring assembly
9. Intermediate wheel
10. Hand-staff pinion
11. Hand staff
12. Large pointer
13. Reducing gear train for small hand
14. Small hand
15. Barometric setting knob
16. Adjustable knob pinion
17. Idler gear and pinion
18. Barometric setting scale
19. Mechanism body gear
20. Thermometal compensator bracket
21. Push rods

Altimeter senses normal decrease in air pressure that accompanies an increase in altitude. The airtight instrument case is vented to the static port. With an increase in altitude, the air pressure within the case decreases and a sealed aneroid barometer (bellows) within the case expands. The barometer movement is transferred to the indicator, calibrated in feet and displayed with two or three pointers. Different types of indicators display indicated altitude in a variety of ways.

INDICATED ALTITUDE is read directly from the altimeter when set to current barometric pressure.

PRESSURE ALTITUDE is read from the altimeter when set to the standard barometric pressure of 29.92 in. Hg.

DENSITY ALTITUDE is the pressure altitude corrected for nonstandard temperature.

TRUE ALTITUDE is the exact height above mean sea level.

ABSOLUTE ALTITUDE is the actual height above the earth's surface.

PROCEDURE

STEP 1: Switch on the Flight simulator
STEP 2: Run the P3D application (Ensure throttle in Ideal condition)
STEP 3: Synchronize the data from physical instrument to onscreen
STEP 4: Adjust the barometric pressure manually based on the runway
STEP 5: Measure and record the initial temperature and pressure at the mean sea-level
STEP 6: Initiate the engine and increase the throttle at a required pace.
STEP 7: Reach the required Altitude in the given time
STEP 8: Record all the altitude
STEP 9: Land the Flight and Check all inflight conditions

ALTIMETER INDICATOR ERRORS

Position Error (installation), Scale Error (aneroids not assuming precise position), mechanical error (gearlinkage malfunction), Density error (non ICAO atmosphere) Hysteresis (different return path), Reversal error (rapid changes) o Altimeter functioning could be affected by Mountains, Down drafts and Turbulence.

RESULT

Thus, the altimeter indicator is measured and calibrated.

EXP.NO.	CALIBRATION AND MEASUREMENT WITH RATE OF CLIMB INDICATOR AND ATTITUDE INDICATOR
DATE:	

AIM

To Calibrate and measure the altimeter indicator.

APPARATUS REQUIRED

- Flight simulator
- Prepare 3D – Software

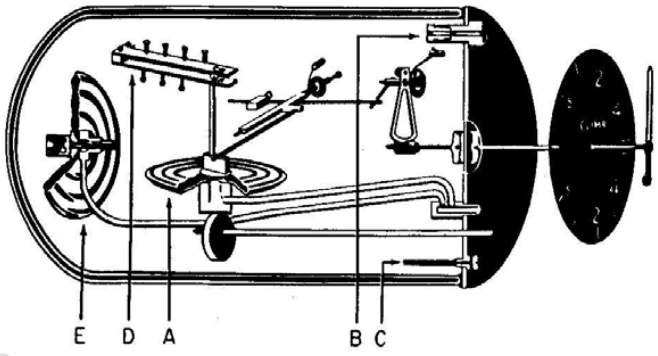
THEORY

Rate of climb indicator is also called vertical speed indicator which is again useful in blind flights.

Level flights could be indicated by keeping the pointer on zero and subsequent changes are indicated in terms of ft/minute.

This is also differential pressure instrument -atmosphere and chamber pressure which is vented through a small capillary.

Response of VSI is rather sluggish and is also sensitive to temperature changes. Mechanical stops prevent damage due to steep dives or maneuvers.



RATE OF CLIMB INDICATOR

A: Diaphragm
B: Orifice Assembly
C: Capillary
D Springs
E: Valve

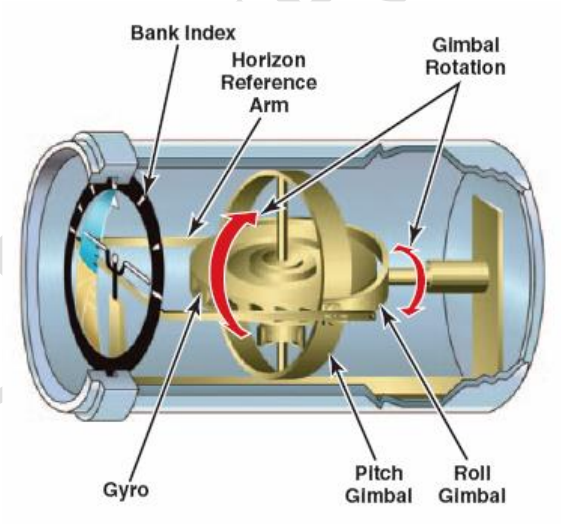
Vertical Speed Indicator (VSI) displays vertical component of an aircraft's flight path. It measures the rate of change of static pressure in terms of feet per minute of climb or descent. VSI compensates for changes in atmospheric density.

VSI is in a sealed case connected to the static line through a calibrated leak (restricted diffuser).

Diaphragm attached to the pointer by a system of linkages is vented to the static line without restrictions. With climb, the diaphragm contracts and the pressure drops faster than case pressure can escape through restructure, resulting in climb

ATTITUDE INDICATOR

Only instrument that gives immediate and direct indication of the airplane's pitch and bank attitude.



Operation

Gyro spins in the horizontal plane, mounted on dual gimbals that allow it to remain in the plane regardless of aircraft movement.

Pendulous vanes allow the attitude indicator erect itself on taxi.

PROCEDURE

STEP 1: Switch on the Flight simulator
STEP 2: Run the P3D application (Ensure throttle in Ideal condition)
STEP 3: Synchronize the data from physical instrument to onscreen
STEP 4: Adjust the barometric pressure manually based on the runway
STEP 5: Gradually increase the throttle and alter the AOA
STEP 6: During the climb measure the rate of change of altitude
STEP 7: Maintain the flight in steady cruise for several minutes
STEP 8: Gradually decrease the throttle and alter the AOA
STEP 9: During the decend measure the rate of change of altitude
STEP 10: Land the Flight and Check all inflight conditions

RESULT

Thus, the rate of climb indicator is measured and calibrated.

EXP.NO.	HALF ADDER AND FULL ADDER
DATE:	

AIM

To design half adder and full adder using basic logic gates and to verify the truth table

APPARATUS REQUIRED

Digital IC trainer kit	-	1
IC7408 (AND Gate)	-	1
IC7486 (XOR Gate)	-	1
IC7432 (OR Gate)	-	1
Connecting Wires	-	as required

THEORY

An adder is a digital circuit that performs addition of numbers. In many computers and other kindsof processors adders are used in the arithmetic logic units or ALU. They are also utilized in other parts of the processor, where they are used to calculate addresses, table indices, increment and decrement operators,and similar operations.

The half adder adds two single binary digits A and B. It has two outputs, sum (S) and carry (C).

The carry signal represents an overflow into the next digit of a multi-digit addition.

A full adder adds binary numbers and accounts for values carried in as well as out. A one-bit full adder adds three one-bit numbers, often written as A, B, and Cin; A and B are the operands, and Cin is a bit carried in from the previous less-significant stage. The full adder is usually a component in a cascade ofadders, which add 8, 16, 32, etc. bit binary numbers.

PROCEDURE

1. Make the connections as per the circuit diagram using the digital IC trainer kit and connecting wires

2. Switch ON the power supply

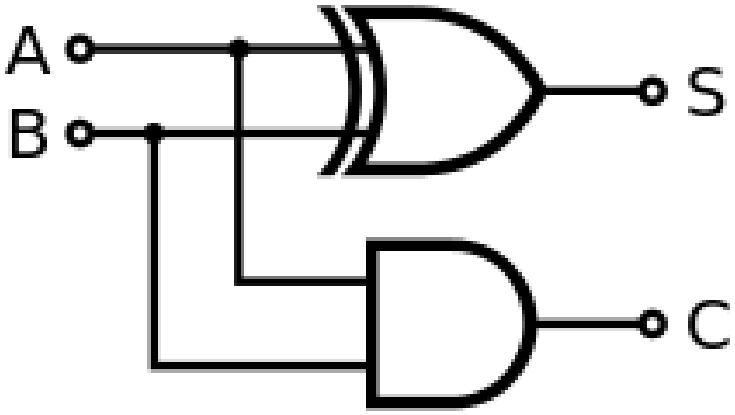
3. Note down the output values of sum and carry for various input combinations

4. Verify the truth table

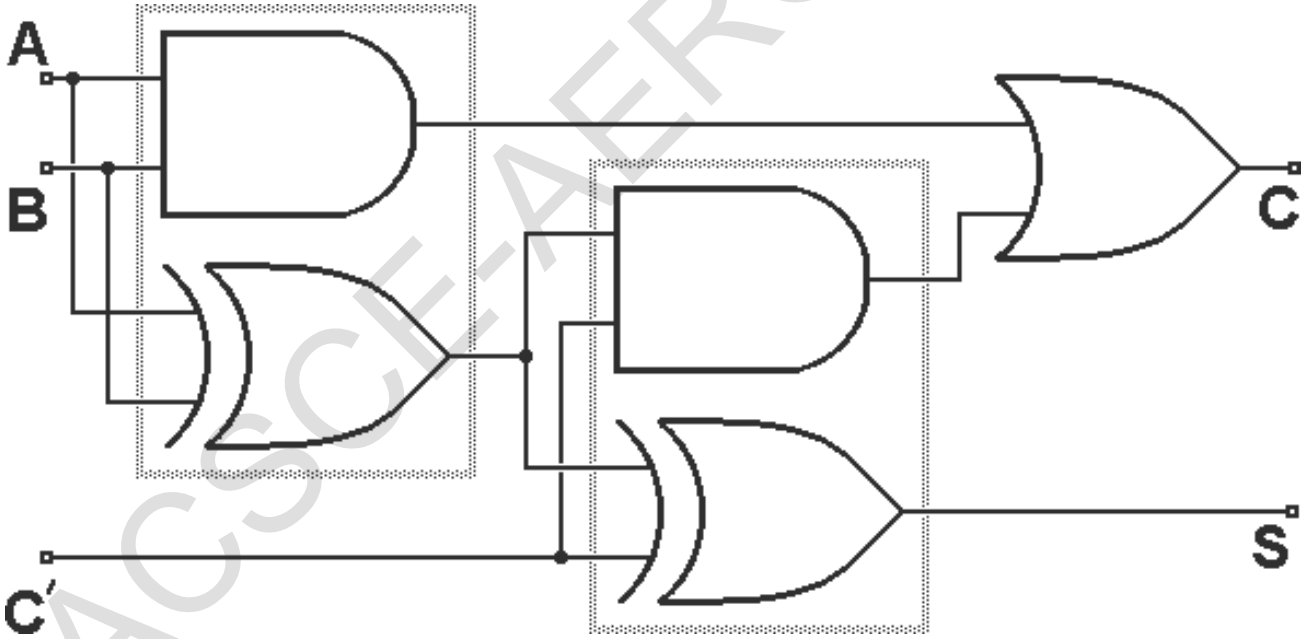
5. Similarly, repeat the procedure for full adder

CIRCUIT DIAGRAM

HALF ADDER



FULL ADDER



TRUTH TABLE

HALF ADDER

A	B	SUM, S	CARRY, C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

FULL ADDER

A	B	Cin	SUM, S	CARRY, C
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

RESULT

Thus, the half adder and full adder were designed using basic logic gates and the output was verified.

EXP.NO.	HALF SUBTRACTOR AND FULL SUBTRACTOR
DATE:	

AIM

To design half subtractor and full subtractor using basic logic gates and to verify the truth table.

APPARATUS REQUIRED

Digital IC trainer kit	-	1
IC7408 (AND Gate)	-	1
IC7486 (XOR Gate)	-	1
IC7432 (OR Gate)	-	1
IC7404 (NOT Gate)	-	1
Connecting Wires	-	as required

THEORY

A subtractor is a digital circuit that performs subtraction of numbers. In many computers and otherkinds of processors subtractors are used in the arithmetic logic units or ALU. They are also utilized in otherparts of the processor, where they are used to calculate addresses, table indices, increment and decrement operators, and similar operations.

The half Subtractor subtracts two single binary digits A and B. It has two outputs, difference (D) and borrow (B0). The borrow signal represents an overflow into the next digit of a multi-digit subtraction.

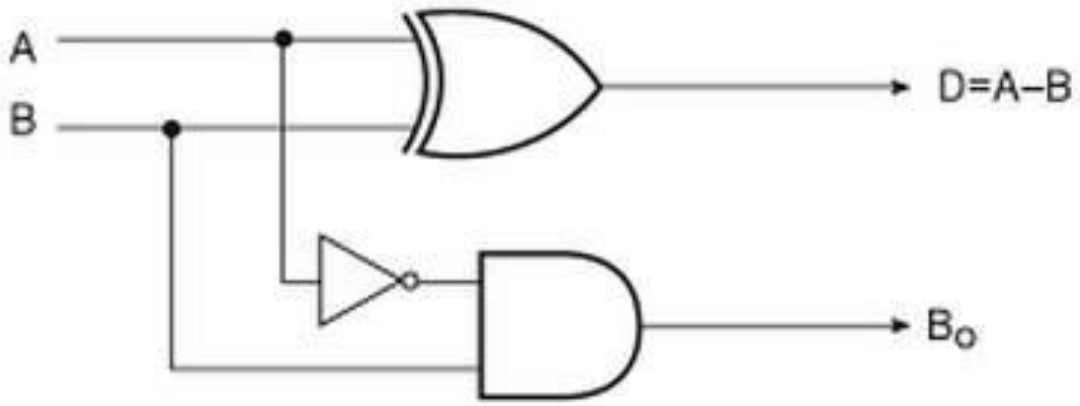
A full Subtractor subtracts binary numbers and accounts for values borrowed in as well as out. A one-bit full Subtractor subtracts three one-bit numbers, often written as A, B, and Cin; A and B are the operands, and Cin is a bit carried in from the previous less-significant stage. The full subtractor is usually a component in a cascade of subtractors, which subtract 8, 16, 32, etc. bit binary numbers.

PROCEDURE

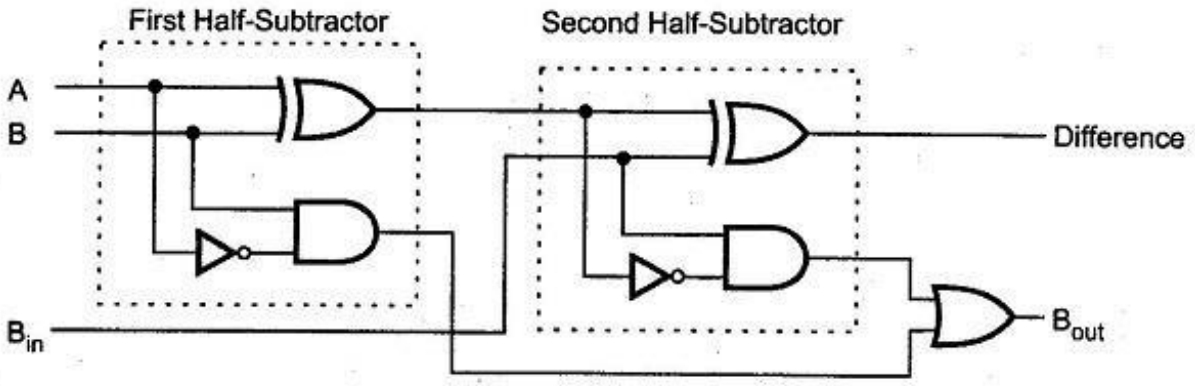
1. Make the connections as per the circuit diagram using the digital IC trainer kit and connecting wires
2. Switch ON the power supply
3. Note down the output values of sum and carry for various input combinations
4. Verify the truth table
5. Similarly, repeat the procedure for full subtractor

CIRCUIT DIAGRAM

HALF SUBTRACTOR



FULL SUBTRACTOR



TRUTH TABLE

HALF SUBTRACTOR

A	B	DIFFERENCE, D	BORROW, B0
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

FULL SUBTRACTOR

A	B	Cin	DIFFERENCE, D	BORROW, B0
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

RESULT

Thus, the half subtractor and full subtractor were designed using basic logic gates and the outputwas verified.

EXP.NO.	BINARY COMPARATOR
DATE:	

AIM

To study the binary comparator

APPARATUS REQUIRED

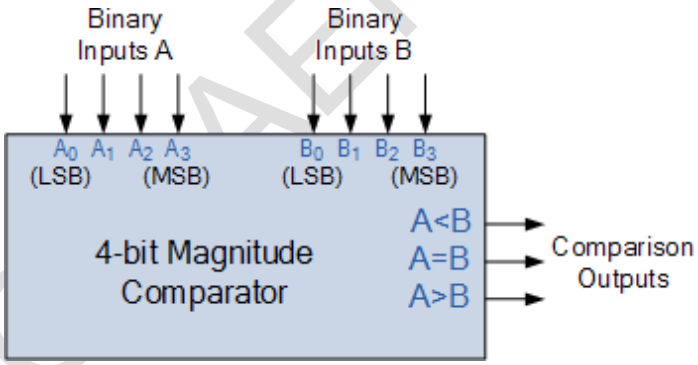
Binary Comparator Kit - 1

Connecting Wires - as required

THEORY

Digital or Binary Comparators are made up from standard AND, NOR and NOT gates that comparethe digital signals present at their input terminals and produce an output depending upon the condition of those inputs.

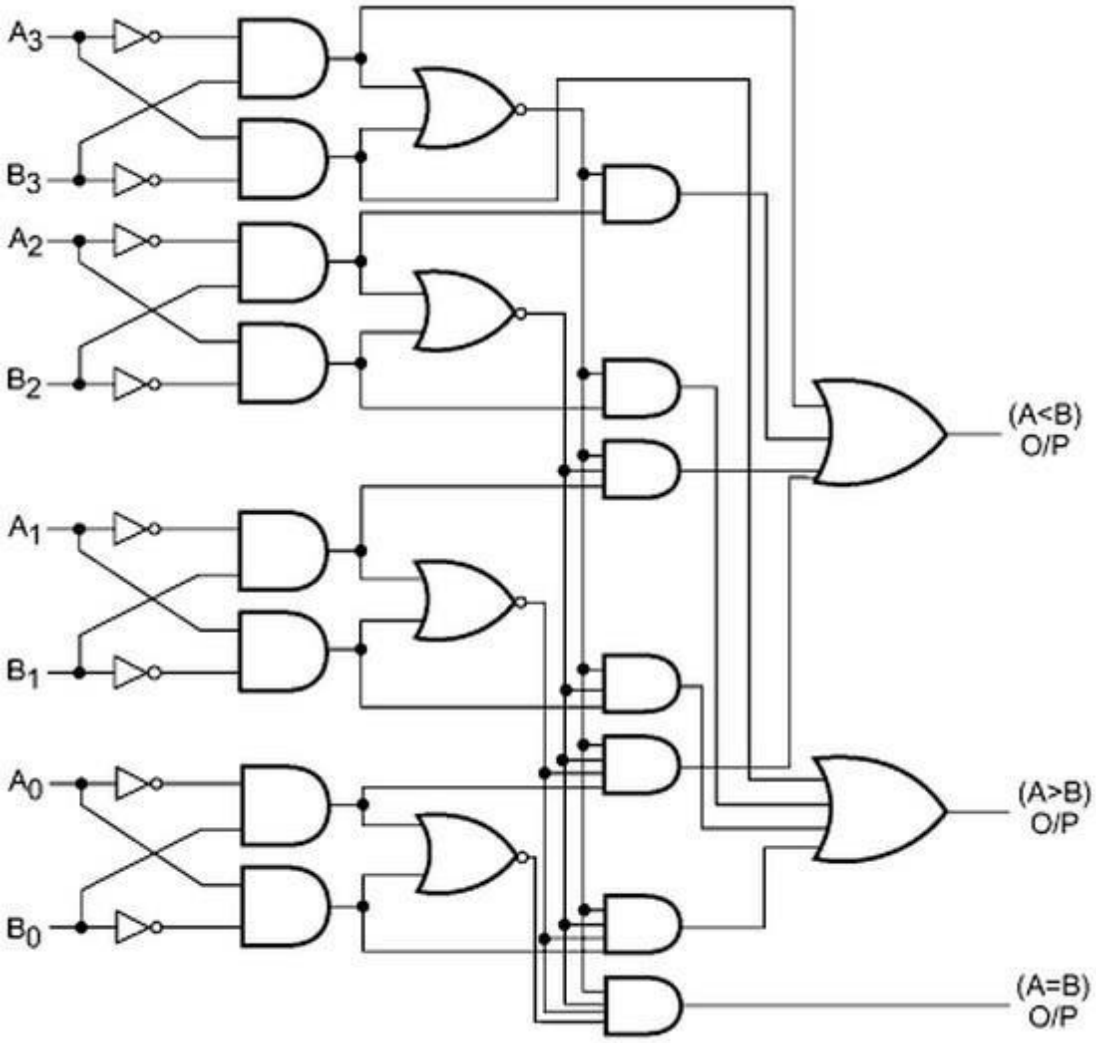
For example, along with being able to add and subtract binary numbers we need to be able to compare them and determine whether the value of input A is greater than, smaller than or equal to the valueat input B etc. The digital comparator accomplishes this using several logic gates that operate on the principles of Boolean Algebra.



PROCEDURE

1. Make the connections as per the circuit diagram using the digital IC trainer kit and connecting wires
2. Switch ON the power supply
3. Note down the output values for various input combinations
4. Verify the truth table
5. Similarly, repeat the procedure for full subtractor

CIRCUIT DIAGRAM



COMPARING INPUTS				OUTPUTS		
A_3, B_3	A_2, B_2	A_1, B_1	A_0, B_0	$Q_{A>B}$	$Q_{A<B}$	$Q_{A=B}$
$A_3>B_3$	X	X	X	H	L	L
$A_3<B_3$	X	X	X	L	H	L
$A_3=B_3$	$A_2>B_2$	X	X	H	L	L
$A_3=B_3$	$A_2<B_2$	X	X	L	H	L
$A_3=B_3$	$A_2=B_2$	$A_1>B_1$	X	H	L	L
$A_3=B_3$	$A_2=B_2$	$A_1<B_1$	X	L	H	L
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0>B_0$	H	L	L
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0<B_0$	L	H	L
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0=B_0$	L	L	H

RESULT

Thus, the magnitude comparator was studied.

EXP.NO.	SUM OF TWO 8-BIT NUMBERS USING 8085
DATE:	

AIM

To write a program to perform sum of two 8-bit numbers using 8085 microprocessor

APPARATUS REQUIRED

8085 Microprocessor kit, Power supply

ALGORITHM

Step 1: Start

Step 2: Get the first number and initialize carry

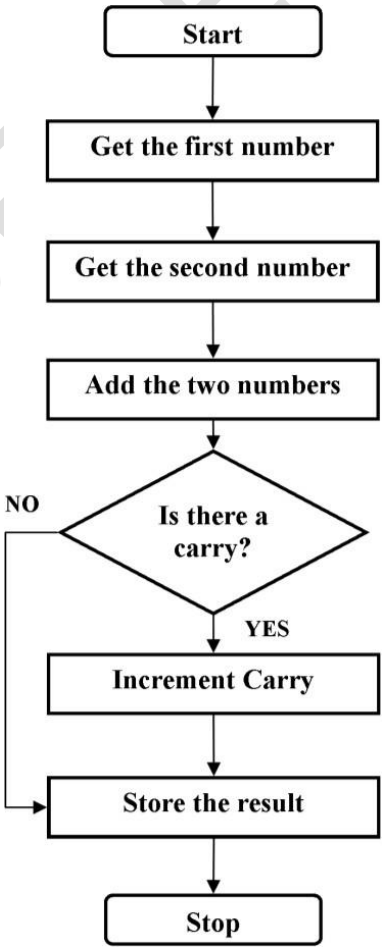
Step 3: Get the second number

Step 4: Add two numbers

Step 5: Store the result

Step 6: Stop

FLOW CHART



PROGRAM

ADDRESS	LABEL	MNEMONICS	HEX CODE	COMMENTS
4000		MVI C, 00H	0E	MOVE 00H TO REGISTER C
4001			00	
4002		LDA 4400H	3A	LOAD THE ACCUMULATOR WITH CONTENT OF 4400
4003			00	
4004			44	
4005		MOV B,A	47	MOVE THE CONTENT OF ACCUMULATOR TO B REGISTER
4006		LDA 4401H	3A	LOAD THE ACCUMULATOR WITH CONTENT OF 4401H
4007			01	
4008			44	
4009		ADD B	80	ADD CONTENT OF B REGISTER WITH THE CONTENT OF ACCUMULATOR
400A		JNC SUM	D2	JUMP ON NO CARRY TO LABEL 'SUM'
400B			0E	
400C			40	
400D		INR C	0C	INCREMENT C REGISTER BY 1
400E	SUM	STA 4402H	32	STORE THE CONTENT OF ACCUMULATOR TO 4402
400F			02	
4010			44	

4011		MOV A,C	79	MOVE THE CONTENT OF C REGISTER TO THE ACCUMULATOR
4012		STA 4403H	32	STORE THE CONTENT OF ACCUMULATOR TO 4403
4013			03	
4014			44	
4015		HLT	76	END OF PROGRAM

TABULATION

INPUT		**OUTPUT**	
ADDRESS	**DATA**	**ADDRESS**	**DATA**

MANUAL CALCULATION

RESULT

Thus the addition of two numbers was performed and the output was verified.

EXP.NO.	DIFFERENCE OF TWO 8-BIT NUMBERS USING 8085
DATE:	

AIM

To write a program to execute difference of two 8-bit numbers using 8085 microprocessor

APPARATUS REQUIRED

8085 Microprocessor kit, Power supply

ALGORITM

Step 1: Start

Step 2: Get the first number

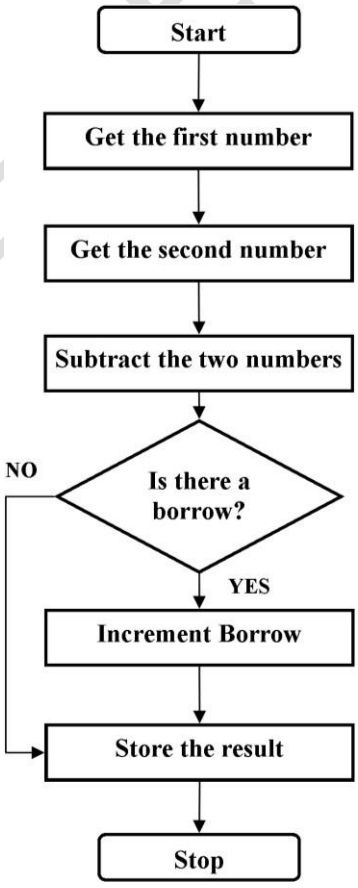
Step 3: Get the second number

Step 4: Subtract second number from first number

Step 5: Store the result

Step 6: Stop

FLOW CHART



PROGRAM

ADDRESS	LABEL	MNEMONICS	HEX CODE	COMMENTS
4000		MVI C, 00H	0E	MOVE 00H TO REGISTER C
4001			00	
4002		LDA 4400H	3A	LOAD THE ACCUMULATOR WITH CONTENT OF 4400
4003			00	
4004			44	
4005		MOV B,A	47	MOVE THE CONTENT OF ACCUMULATOR TO B REGISTER
4006		LDA 4401H	3A	LOAD THE ACCUMULATOR WITH CONTENT OF 4401H
4007			01	
4008			44	
4009		SUB B	80	SUBTRACT CONTENT OF B REGISTER WITH THE CONTENT OF ACCUMULATOR
400A		JNC DIFF	D2	JUMP ON NO CARRY TO LABEL 'DIFF'
400B			0E	
400C			40	
400D		INR C	0C	INCREMENT C REGISTER BY 1
400E	DIFF	STA 4402H	32	STORE THE CONTENT OF ACCUMULATOR TO 4402

400F			02	
4010			44	
4011		MOV A,C	79	MOVE THE CONTENT OF C REGISTER TO THE ACCUMULATOR
4012		STA 4403H	32	STORE THE CONTENT OF ACCUMULATOR TO 4403
4013			03	
4014			44	
4015		HLT	76	END OF PROGRAM

TABULATION

INPUT		**OUTPUT**	
ADDRESS	**DATA**	**ADDRESS**	**DATA**

MANUAL CALCULATION

RESULT

Thus the program was executed for the difference of 2 numbers in 8085 microprocessor and the output was verified.

EXP.NO.	INTERFACING ADC WITH 8085 MICROPROCESSOR
DATE:	

AIM

To write a program to interface ADC card with 8085 microprocessor and store digital data in memory.

APPARATUS REQUIRED

- 8085 Microprocessor kit,
- Power supply,
- Analog to Digital Converter Card

ALGORITHM

Step 1: Initialize the Channel

Step 2: Transfer the converter data output to the data port selected

Step 3: Address is cached in the decoder on low to high transition

Step 4: Conversion is done on falling edge

Step 5: ADC is selected and delay is given

Step 6: After end of conversion, get the data through the input port address

Step 7: Check for end of conversion

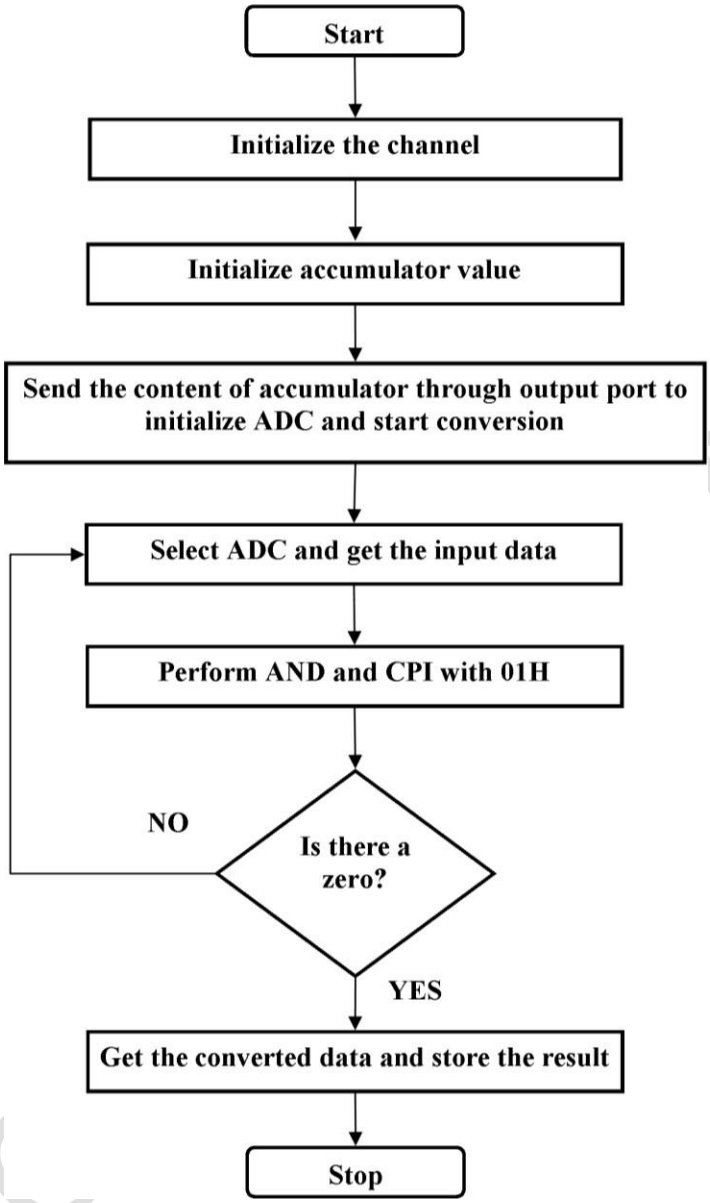
Step 8: If accumulator = 0 (end of conversion), proceed to next step; otherwise go to step 6

Step 9: Get the decoded data through the input port

Step 10: Store the result

Step 11: End of program

FLOW CHART


Start
Initialize the channel
Initialize accumulator value
Send the content of accumulator through output port to initialize ADC and start conversion
Select ADC and get the input data
Perform AND and CPI with 01H
Is there a zero?
NO
YES
Get the converted data and store the result
Stop

PROGRAM

ADDRESS	LABEL	MNEMONICS	HEX CODE	COMMENTS
8200		MVI A,98H	3E	Move 98H to accumulator
8202			00	
8202		OUT 23H	D3	Output the content of accumulator through port D3
8204		MVI A,00H	3E	
8206		OUT 21H	D3	
8208			00	
8208		MVI A,01H	3E	Channel in address 01
820A		OUT 22H	D3	Written in Port c
820C		MVI A,01H	3E	Make ale bit 1
820E		OUT 21H	D3	Written in port B
8210		MVI A,03H	3E	Make start conversation bit 1
8212		OUT 21H	D3	Written in port B
8214		MVI A,00H	3E	Make ale and Start bit 0
8216		OUT 21H	D3	Written in port B
8218		IN 22H	DB	Read port C
821A		ANI 10H	E6	To check for EQC-PC4 bit

821C		JNZ 8218	C2	If no, read again
821F		MVI A,04H	3E	Make OE bit 1
8221		OUT 21H	D3	Written in port b
8223		IN 20H	DB	
8225		MOV B,A	47	
8226		CALL 8400	CD	
8229				
8229		JMP 8208	C3	jump on no zero to label ‘loop’
822C		ORG 8400H	00	
8400			00	
8400	DELAY	MVI C,03H	03	
8402	DELAY 1	LXI D,FFFFH	11	Load 16 bit value in de register pair
8405	DEC	DCX D	1B	Decrement de register pair
8406		MOV A,D	7A	Move the content of D
8407		ORA E	B3	Or with the content of E
8408		JNZ 8405	C2	Jump not zero to decrement
840B		DCR C	0D	
840C		JNZ 8402	C2	

840F		RET	C9	Back to place where delay was called
8410	END			

TABULATION

ANALOG INPUT	DIGITAL OUTPUT

RESULT

Thus, the program to interface ADC with 8085 microprocessor was implemented and the outputwas verified.

EXP.NO.	INTERFACING DAC WITH 8085 MICROPROCESSOR
DATE:	

AIM

To write a program to interface DAC card with 8085 microprocessor and generate square, triangular andsawtooth waveforms.

APPARATUS REQUIRED

- 8085 Microprocessor kit,
- Power supply,
- Cathode Ray Oscilloscope,
- Digital to Analog Converter Card

PROGRAM

ADDRESS	LABEL	MNEMONICS	HEX CODE	COMMENTS
8200	START	MVI A, 80H	3E	Port A,B – Output Port C(U),(I) - Input
8202			00	
8202		OUT23H	D3	Written in CWR
8204			00	
8204		MVI B, 00H	06	Initialize b reg to zero
8206		MOV A,B	78	Mov b reg content to a reg
8207		OUT 20H	D3	Transfer a reg content to Port A
8209		INR B	04	Increment b reg content
820A		CALL 8400	CD	A delay
820D		JMP 8206	C3	Repeat sending data to Port A

8210			00	
8210		ORG 8400H	00	
8400	DELAY	MVI C,02	0E	Move value 2 to C reg
8402	DELAY 1	LXI D,FFFF	11	Load 16 bit value in de reg pair
8405	DEC	DCX D	1B	Decrement de register pair
8406		MOV A,D	7A	Mov the content of D to A
8407		ORA E	B3	Or with the content of E
8408		JNZ 8405	C2	Jump not zero to decrement
840B		DCR C	0D	Decrement c reg content
840B		JNZ 8402	C2	Jump not zero to load c reg with the value
840F		RET	C9	Back to place ehere delay was called
8410	END			

RESULT

Thus the program to interface DAC with 8085 microprocessor and to generate square and triangular waveforms was performed and the output was verified.

EXP.NO.	MULTIPLEXER AND DEMULTIPLEXER
DATE:	

AIM

To design and implement multiplexer and demultiplexer and to verify the truth table.

APPARATUS REQUIRED

Digital IC trainer kit	-	1
IC74151 (MUX)	-	1
IC74138 (DEMUX)	-	1
Mono Pulse Generator	-	1
Address Generator	-	1
Connecting Wires	-	as required

THEORY

The multiplexer, shortened to “MUX” or “MPX”, is a combinational logic circuit designed to switch one of several input lines through to a single common output line by the application of a control signal. Multiplexers operate like very fast acting multiple position rotary switches connecting or controllingmultiple input lines called “channels” one at a time to the output.

Multiplexers, or MUX’s, can be either digital circuits made from high speed logic gates used to switch digital or binary data or they can be analogue types using transistors, MOSFET’s or relays to switchone of the voltage or current inputs through to a single output.

In digital electronics, multiplexers are also known as data selectors because they can “select” eachinput line, are constructed from individual Analogue Switches encased in a single IC package as opposed to the “mechanical” type selectors such as normal conventional switches and relays.

They are used as one method of reducing the number of logic gates required in a circuit design or when a single data line or data bus is required to carry two or more different digital signals.

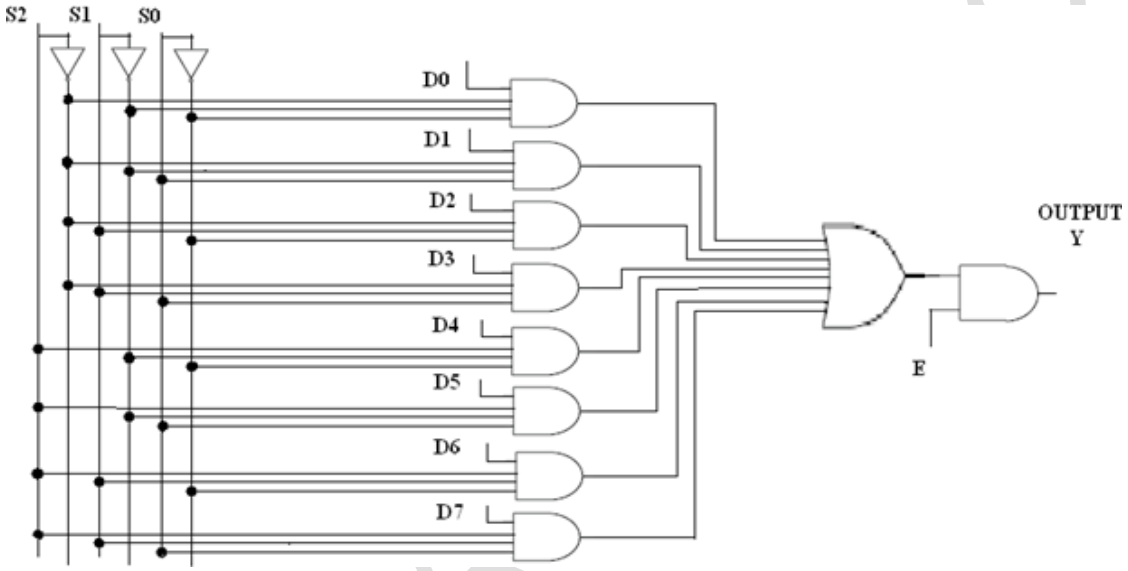
The demultiplexer takes one single input data line and then switches it to any one of a number of individual output lines one at a time. The demultiplexer converts a serial data signal at the input to a parralleldata at its output lines.

PROCEDURE

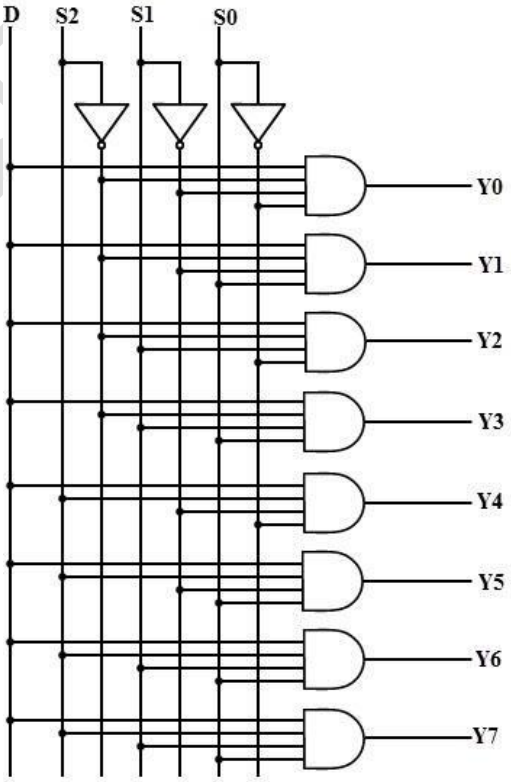
1. Make the connections as per the circuit diagram using the digital IC trainer kit and connecting wires
2. Switch ON the power supply
3. Note down the output values for various input combinations
4. Verify the truth table

CIRCUIT DIAGRAM

MULTIPLEXER



DEMULTIPLEXER



TRUTH TABLE

MULTIPLEXER

S2	S1	S0	OUTPUT Y
0	0	0	D0
0	0	1	D1
0	1	0	D2
0	1	1	D3
1	0	0	D4
1	0	1	D5
1	1	0	D6
1	1	1	D7

DEMULTIPLEXER

Data Input	Select Inputs			Outputs							
D	S_2	S_1	S_0	Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0
D	0	0	0	0	0	0	0	0	0	0	D
D	0	0	1	0	0	0	0	0	0	D	0
D	0	1	0	0	0	0	0	0	D	0	0
D	0	1	1	0	0	0	0	D	0	0	0
D	1	0	0	0	0	0	D	0	0	0	0
D	1	0	1	0	0	D	0	0	0	0	0
D	1	1	0	0	D	0	0	0	0	0	0
D	1	1	1	D	0	0	0	0	0	0	0

RESULT

Thus, MUX and DEMUX were designed and implemented and their output was verified.

EXP.NO.	ENCODER AND DECODER
DATE:	

AIM

To study the encoder and decoder and their operation

APPARATUS REQUIRED

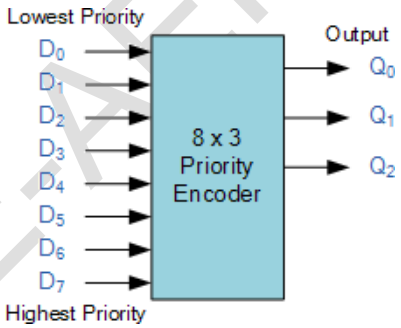
Encoder & Decoder trainer kit - 1

Connecting Wires - as required

THEORY

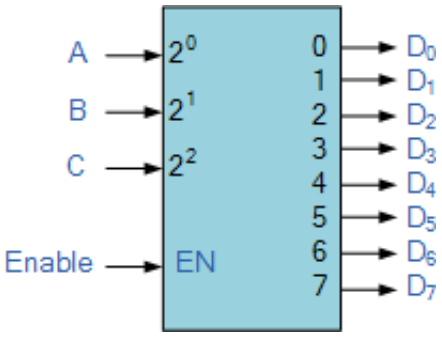
ENCODER

An encoder is a digital circuit that performs inverse operation of a decoder. An encoder has 2ninputlines and n output lines. In encoder the output lines generates the binary codecorresponding to the input value. In octal to binary encoder it has eight inputs, one for each octaldigit and three output that generate the corresponding binary code. In encoder it is assumed thatonly one input has a value of one at any giventime otherwise the circuit is meaningless. Here, when all inputs are zero the outputs are zero. The zero outputs can also begenerated when D0 = 1.



DECODER

A decoder is a multiple input multiple output logic circuit which converts coded input into coded output where input and output codes are different. The input code generally has fewer bits than the output code. Each input code word produces a different output code word i.e there is one to one mapping can be expressed in truth table. In the block diagram of decoder circuit the encoded information is present as n input producing 2npossible outputs. 2noutput values are from 0through output 2n-1



PROCEDURE

1. Make the connections as per the circuit diagram using the Universal Shift Registers trainer kit andconnecting wires

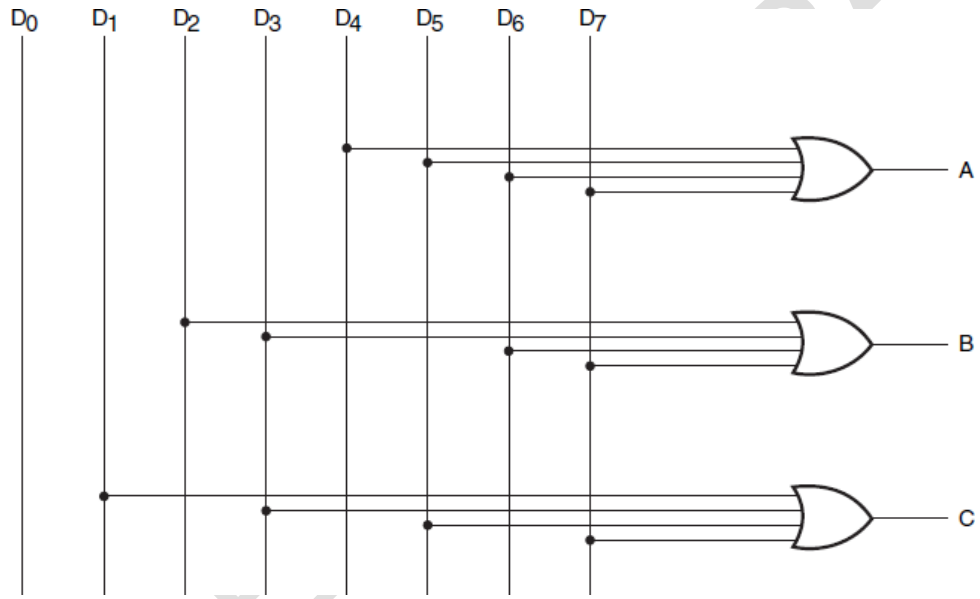
2. Switch ON the power supply

3. Note down the output values for various input combinations

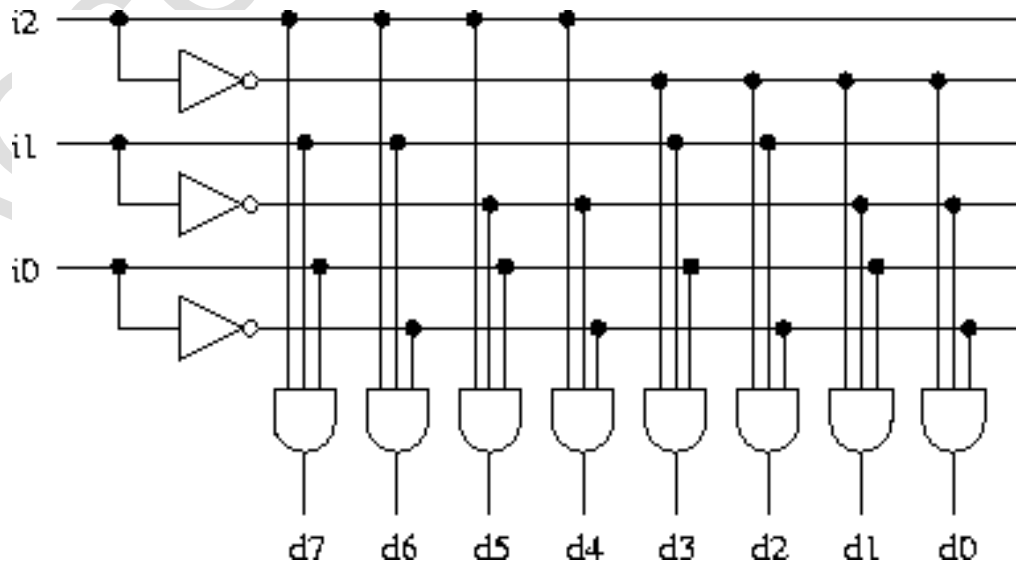
4. Verify the output

CIRCUIT DIAGRAM

ENCODER



DECODER



TRUTH TABLE

ENCODER

INPUT								OUTPUT		
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Q_2	Q_1	Q_0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	X	0	0	1
0	0	0	0	0	1	X	X	0	1	0
0	0	0	0	1	X	X	X	0	1	1
0	0	0	1	X	X	X	X	1	0	0
0	0	1	X	X	X	X	X	1	0	1
0	1	X	X	X	X	X	X	1	1	0
1	X	X	X	X	X	X	X	1	1	1

DECODER

INPUT			OUTPUT							
Q_2	Q_1	Q_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

RESULT

Thus, the encoder and decoder were studied and the output was verified.

EXP.NO.	STUDY OF MIL STD 1553B DATA BUS
DATE:	

MIL STD 1553B Data Bus

MIL-STD-1553 is a military standard that defines the electrical and protocol characteristics for a data bus. A data bus is used to provide a medium for the exchange of data and information between various systems. It is similar to what the personal computer and office automation industry has dubbed a Local Area Network(LAN).

MIL-STD-1553B defines the term Time Division Multiplexing (TDM) as "the transmission of information from several signal sources through one communications system with different signal samples staggered in time to form a composite pulse train." This means that data can be transferred between multiple avionics units over a single transmission media, with the communications between the different avionics boxes taking place at different moments in time, hence time division.

Elements of MIL STD 1553B Data Bus

They are:

- The transmission media.
- Remote terminals.
- Bus controllers.
- Bus monitors.

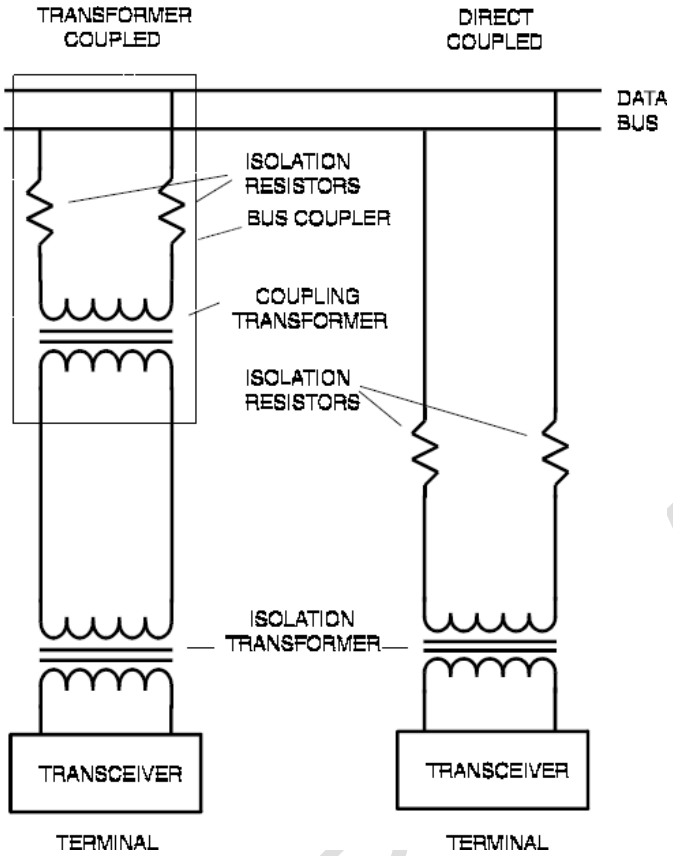
Transmission Media

The transmission media, or data bus, is defined as a twisted shielded pair transmission line consisting of the main bus and a number of stubs. There is one stub for each terminal connected to the bus. The main data bus is terminated at each end with a resistance equal to the cable's characteristic impedance (plus or minus two percent). This termination makes the data bus behave electrically like an infinite transmission line.

Coupling Method

There are two types of coupling, (i) direct coupling and (ii) transformer coupled

The primary difference between the two is that the transformer coupled method uses an isolation transformer for connecting the stub cable to the main bus cable. In both methods, two isolation resistors areplaced in series with the bus. In the direct-coupled method, the resistors are typically located within the terminal, whereas in the transformer-coupled method, the resistors are typically located with the coupling transformer in boxes called data bus couplers.



Another difference between the two coupling methods is the length of the stub. For the direct-coupled method, the stub length is limited to a maximum of one foot. For the transformer-coupled method, the stubcan be up to a maximum of twenty feet long. Therefore for direct-coupled systems, the data bus must be routed in close proximity to each of the terminals, whereas for a transformer-coupled system, the data bus may be up to twenty feet away from each terminal.

Remote Terminals

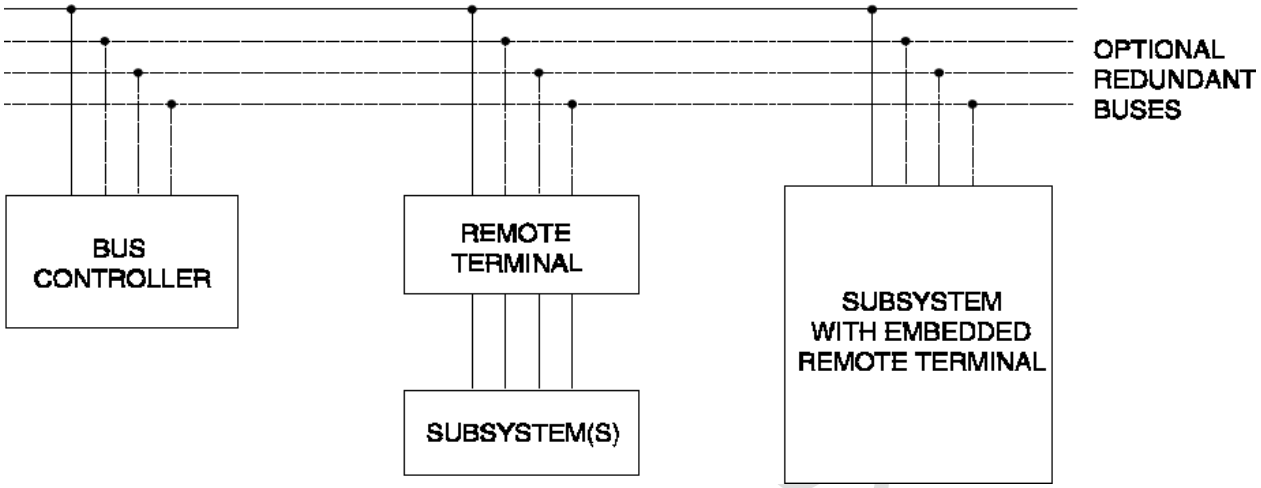
Remote terminals are defined within the standard as “All terminals not operating as the bus controller or asa bus monitor”. Therefore if it is not a controller, monitor, or the main bus or stub, it must be a remote terminal. The remote terminal comprises the electronics necessary to transfer data between the data bus andthe subsystem.

A remote terminal typically consists of a transceiver, an encoder/decoder, a protocol controller, a buffer ormemory, and a subsystem interface. In a modern black box containing a computer or processor, the subsystem interface may consist of the buffers and logic necessary to interface to the computer's address, data, and control buses.

But a remote terminal must be more than just a data formatter. It must be capable of receiving and decodingcommands from the bus controller and responding accordingly. It must also be capable of buffering a message worth of data, detecting transmission errors and performing validation tests upon the data, and reporting the status of the message transfer.

A remote terminal must follow the protocol defined by the standard. It can only respond to commands received from the bus controller (i.e., it speaks only when spoken to). When it receives a

valid command, it must respond within a very small, closely defined amount of time. If a message doesn't meet the validityrequirements defined, then the remote terminal must invalidate the message and discard the data (not allowit to be used by the subsystem). In addition to reporting status to the bus controller, most remote terminalstoday are capable of providing some level of status information to the subsystem.



Bus Controller

The bus controller is responsible for directing the flow of data on the data bus. While several terminals maybe capable of performing as the bus controller, only one bus controller may be active at a time. The bus controller is the only one allowed to issue commands onto the data bus.

The commands may be for the transfer of data or the control and management of the bus (referred to as mode commands). Typically, the bus controller is a function that is contained within some other computer,such as a mission computer, a display processor, or a fire control computer.

There are three types of bus controller architectures,

- Word Controller – transfers one word at a time (old)
- Message Controller – transfers one message at a time
- Frame Controller – capable of processing multiple messages in a sequence specified by a hostcomputer (latest)

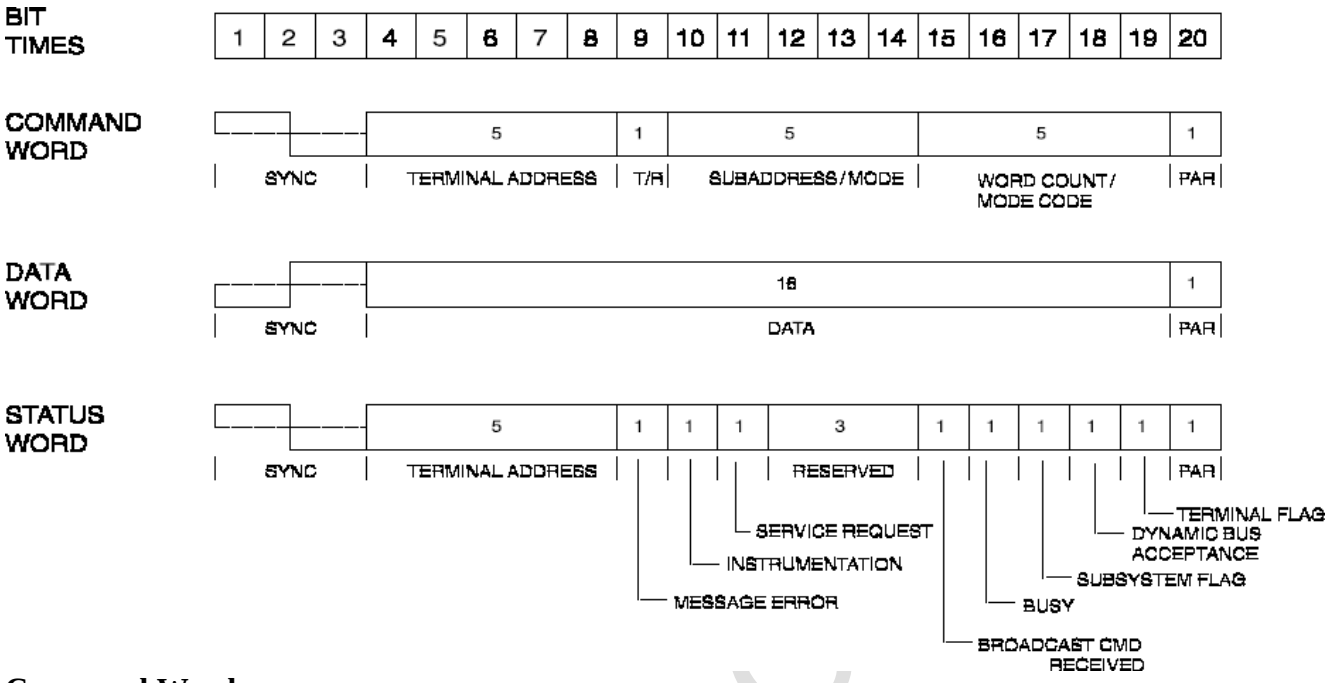
Bus Monitor

A bus monitor is a terminal that listens (monitors) to the exchange of information on the data bus. The standard strictly defines how bus monitors may be used, stating that the information obtained by a bus monitor be used "for off-line applications (e.g., flight test recording, maintenance recording or mission analysis) or to provide the back-up bus controller sufficient information to take over as the bus controller."

Word Types

- Command Word

- Data Word
- Status Word



Command Word

The Command Word (CW) specifies the function that a remote terminal is to perform. Only the active buscontroller transmits this word. The word begins with command sync in the first three bit times. Five bit Terminal Address (TA) field (bit times 4-8) states to which unique remote terminal the command is intended(no two terminals may have the same address).

The next bit (bit time 9) makes up the Transmit/Receive (T/R) bit. This defines the direction of informationflow and is always from the point of view of the remote terminal. A transmit command (logic 1) indicatesthat the remote terminal is to transmit data, while a receive command (logic 0) indicates that the remote terminal is going to receive data. The only exceptions to this rule are associated with mode commands.

The next five bits (bit times 10-14) make up the Subaddress (SA)/Mode Command bits. Logic 00000B or 11111B within this field is decoded to indicate that the command is a Mode Code Command.

The next five bit positions (bit times 15-19) define the Word Count (WC) or Mode Code to be performed.If the Subaddress/Mode Code field is 00000B or 11111B, then this field defines the mode code to be performed. If not a mode code, then this field defines the number of data words to be received or transmitteddepending on the T/R bit. A word count field of 00000B is decoded as 32 data words.

The last bit (bit time 20) is the word parity bit. Only odd parity is used.

Data Word

The Data Word (DW) contains the actual information that is being transferred within a message. The first three-bit time contains data sync. This sync pattern is the opposite of that used for command and status words and therefore is unique to the word type. Data words can be transmitted by either a remote terminal(transit command) or a bus controller (receive command).

The next sixteen bits of information are left to the designer to define. The last bit is parity.

Status Word

A remote terminal in response to a valid message transmits only the status word (SW). The status word is used to convey to the bus controller whether a message was properly received or to convey the state of theremote terminal (i.e., service request, busy, etc.).

The first five bits (bit times 4-8) of the information field are the Terminal Address (TA).

The next bit (bit time 9) is the Message Error (ME) bit. This bit is set by the remote terminal upon detection of an error in the message or upon detection of an invalid message (i.e. Illegal Command) to the terminal.

The Instrumentation bit (bit time 10) is provided to differentiate between a command word and a status word (remember they both have the same sync pattern). The instrumentation bit in the status word is alwaysset to logic "0". If used, the corresponding bit in the command word is set to logic 1.

The Service Request bit (bit time 11) is provided so that the remote terminal can inform the bus controllerthat it needs to be serviced. This bit is set to logic "1" by the subsystem to indicate that servicing is needed.

Bit times 12-14 are reserved for future growth of the standard and must be set to logic "0".The bus controller should declare a message in error if the remote terminal responds with any of these bits set in itsstatus word.

The Broadcast Command Received bit (bit time 15) indicates that the remote terminal received a valid broadcast command. On receiving a valid broadcast command, the remote terminal sets this bit to logic "1"and suppresses the transmission of its status words.

The Busy bit (bit time 16) is provided as a feedback to the bus controller as to when the remote terminal isunable to move data between the remote terminal electronics and the subsystem in compliance to a command from the bus controller.

The Dynamic Bus Control Acceptance bit (bit time 18) informs the bus controller that the remote terminalhas received the Dynamic Bus Control Mode Code and has accepted control of the bus. The bus controller,on receiving the status word from the remote terminal with this bit set, ceases to function as the bus controller and may become a remote terminal or bus monitor.

The Terminal Flag bit (bit time 19) informs the bus controller of a fault or failure within the remote terminalcircuitry (only the remote terminal). A logic "1" shall indicate a fault condition.

The last bit (bit time 20) is used for parity.

EXP.NO.	STUDY OF PAM
DATE:	

Generation of PAM

Pulse amplitude modulation is the basic form of pulse modulation in which the signal is sampled at regular and each sample is made proportional to the amplitude of the modulating signal at the sampling instant.

The Fig1 shows the generation of PAM signal from the sampler which has two inputs i.e. modulating signal and sampling signal or carrier pulse.

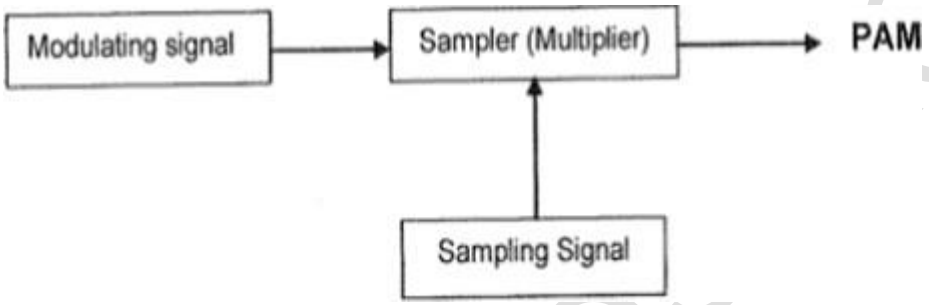


Fig1. Generation of PAM signal

Thus the amplitude of the signal is proportional to the modulating signal through which information is carried. This is Pulse amplitude modulation signal.

Fig2 shows the spectrum of pulse amplitude modulated signal along with the message signal and the sampling signal which is the carrier train of pulses with the help of the waveform plotted in time domain.

Pulse Modulation may be used to transmitting analog information, such as continuous speech signal or data.

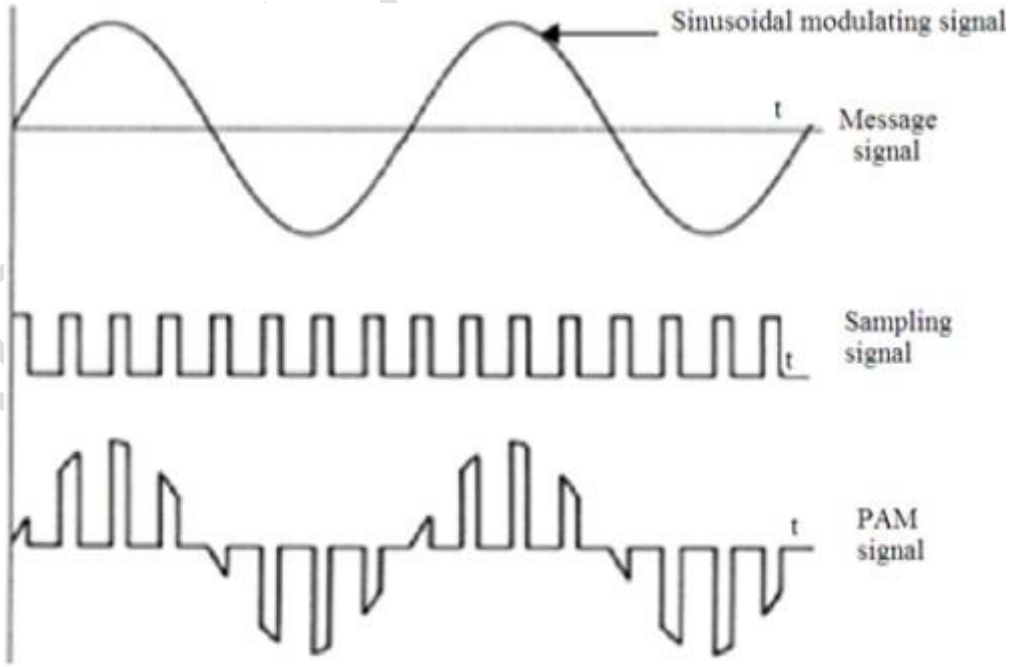


Fig2. Spectrum of PAM signal

Demodulation of PAM

For Demodulation of the Pulse Amplitude Modulated signal, PAM is fed to the low pass filter as shown in Fig3 below.

The low pass filter eliminates high frequency ripples and generates the demodulated signal which has its amplitude proportional to PAM signal at all time instant.

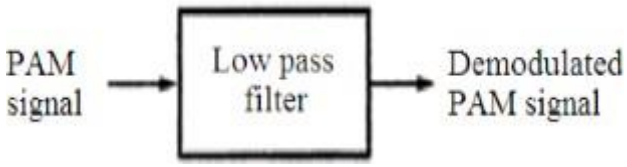


Fig3. PAM detector

This signal is then applied to an inverting amplifier to amplify its signal level to have the demodulated output with almost equal amplitude with the modulating signal.

The Fig4 below shows the modulated and demodulated PAM signal.

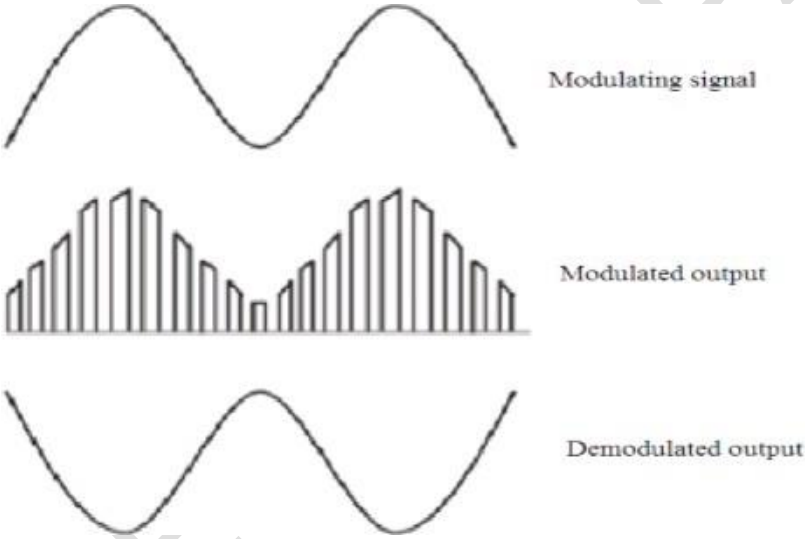


Fig4. Modulation and demodulation of PAM

Generation of PWM

PWM signal can be generated by using a comparator, where modulating signal and sawtooth signal form the input of the comparator. It is the simplest method for PWM generation.

The PWM generation is explained with the help of the Fig5 given below.

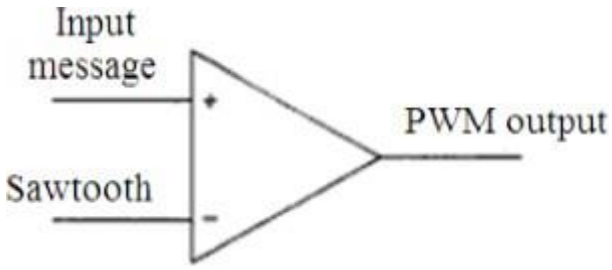


Fig5. PWM generation by a comparator

As shown in the figure, one input of the comparator is fed by the input message or modulating signal and the other input by a sawtooth signal which operates at carrier frequency.

Considering both ±ve sides, the maximum of the input signal should be less than that of sawtooth signal.

The comparator will compare the two signals together to generate the PWM signal at its output as shown in the third waveform of Fig6.

The rising edges of the PWM signal coincides with the falling edge of the sawtooth signal.

When the sawtooth signal is at the minimum value which is less than the minimum of the input signal, then the positive input of the comparator is at higher potential which gives the comparator output as positive.

When the sawtooth signal rises and is at the maximum value, the negative input of the comparator is at higher potential, which will produce the comparator output to be negative.

Thus the input signal magnitude determines the comparator output and its potential, which then decides the width of the pulse generated at the output.

In other words we can say that the width of the pulse generated signal is directly proportional to the amplitude of the modulating signal.

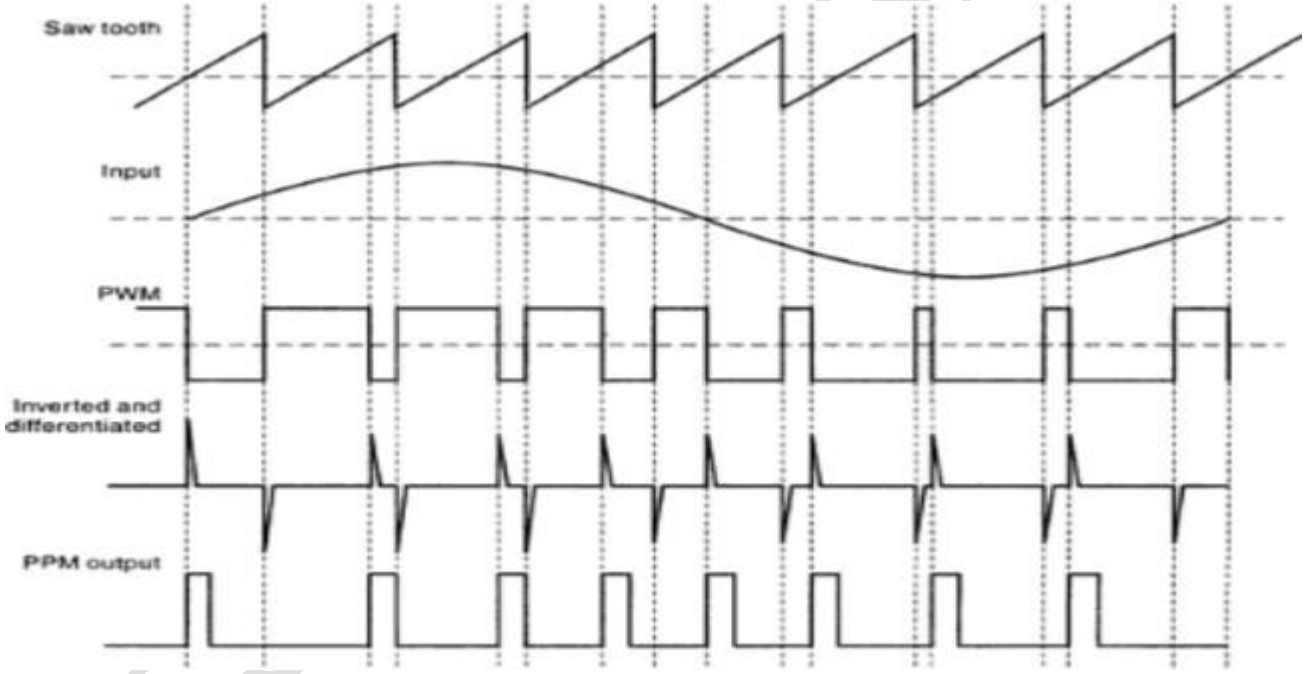


Fig6. PWM and PPM signal generation

Generation of PPM

PPM signal can be generated with the help of PWM as shown in Fig7 below.

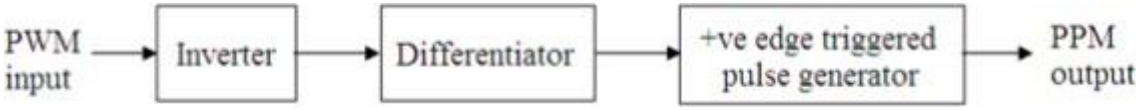


Fig7. PPM generation from PWM

The PWM signal generated above is sent to an inverter which reverses the polarity of the pulses.

This is then followed by a differentiator which generates +ve spikes for PWM signal going from High to Low and -ve spikes for Low to High transistion. The spikes generated are shown in the fourth waveform of Fig8.

These spikes are then fed to the positive edge triggered pulse generator which generates fixed width pulses when a +ve spike appears, coinciding with the falling edge of the PWM signal.

Thus PPM signal is generated at the output which is shown in the fifth waveform of Fig8.where pulse position carry the message information.

Demodulation of PWM and PPM

For PWM demodulation, put a ramp at the +ve edge which will stop at the arrival of –ve edge.

The ramp will attain different heights in each cycle since the widths are different and the heights attained are directly proportional to the pulse width and in turn the amplitude of the message signal.

This is then passed through a low pass filter where it will follow the envelop i.e. the message signal, which produces the demodulated signal at the output.

For PPM demodulation, ramp is used which starts at the +ve edge of the one pulse and stops at the +ve edge of the next pulse.

Thus the height of the generated ramp is determined by the delay between the pulses which indirectly follows the amplitude of the modulating signal.

This is then passed through a low pass filter which filters the envelop information as the demodulated signal.

The modulation and demodulation waveforms of PWM and PPM signals are shown in Fig8.

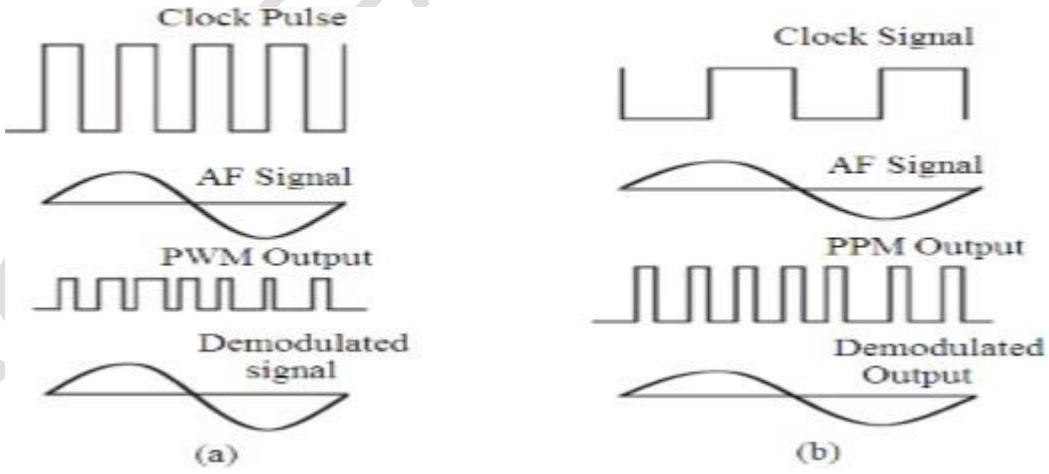


Fig8. Modulation and Demodulation of (a) PWM and (b) PPM