Unit - V

Introduction to Digital signal processors

DSP functionalities - Circular buffering - DSP architecture - fixed and floating point architecture principles - programming - Application examples.

DSP functionalities :-

* Digital signal processing is carried out by mathematical operations.
* Digital signal processors are microprocessors specifically designed to handle digital signal processing tasks.

Circular Buffering :

* A circular buffer, circular queue, cyclic buffer or ring buffer is a data structure that uses a single, fixed size buffer as if it were connected to end to end.
* The circular buffer is well suited as a FIFO buffer while a standard non circular buffer is well suited as a LIFO buffer.
* A consequence of the circular buffer is that when it is full and a subsequent write is performed. then it starts overwriting the oldest data.

* Digital signal processors are designed to carry out FIR filters and similar techniques
* In real-time processing, the output signal is produced at the same time that the input signal is being acquired.
* Circular buffering is used in FIR filter implementation in real time
* Four parameters are needed to manage a circular buffer.
  → First there must be a pointer that indicates the start of the circular buffer in memory
  → second there must be a pointer indicating the end of the array. or a variable that holds its length.
  → Third the step size of the memory addressing must be specified.
  These three values define size and configuratio of the circular buffer.
  → Fourth value, the pointer to the most recent sample., must be modified as each new sample is acquired.

* DSPs should be optimized at managing circular buffers to achieve the highest possible execution speed.

Von Neumann Architecture :-

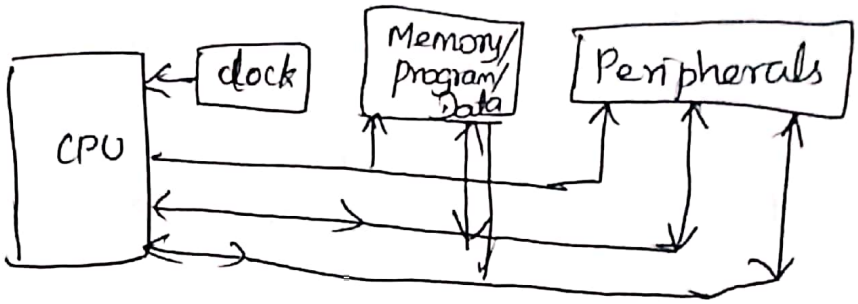


fig: Von Neumann Architecture

→ In 1946 John Von Neumann developed the first Computer architecture that allowed the computer to be programmed by codes residing in memory.

→ In this program instructions were stored in Read Only memory (ROM).

→ It is used in majority of microprocessors.

→ In a computer with Non Neumann architecture the CPU can be either reading an instruction or reading/writing data from/to memory

→ Both cannot occur at the same time since the instruction and data use the same signal pathways and memory.

* It consists of three buses
  * Data bus
  * the address bus
  * Control bus

Data bus :-

→ Transport data between CPU and its peripherals. It is bidirectional.

→ The CPU can read or write data in the peripherals.

Address Bus :-

→ It indicate which peripherals it wants to access and within each peripheral which specific register.

→ Address bus is unidirectional.

→ The CPU always writes the address, which is read by the peripherals.

Control Bus :-

* The bus carries signals that are used to manage and synchronize exchange between the CPU and its peripherals, as well as that indicates if the CPU wants to read or write the peripheral.

* The main characteristics of the Von Neumann architecure is that it only represents 1 bus system.
* The same bus carries all the information exchange between the CPU and the peripherals including the instruction code and data processed by the CPU.

Harvard Architecture:

* It is a relay-based computer which stored instruction on punched tape and data in relay latches.
* It physically separates memories for their instructions and data requiring dedicated buses for each of them.
* Instructions and operands can therefore be fetched simultaneously.
* Most DSP processors use a modified Harvard architecture with two or three memory buses. So it allows access to filter co-efficients and input signals in the same cycle.

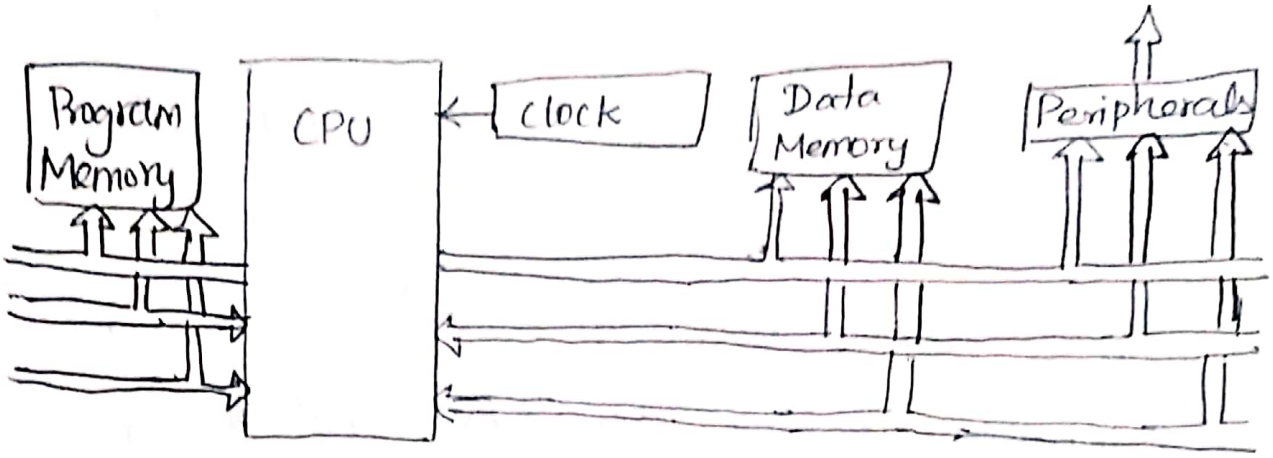


fig: Harvard Architecture

* It is capable of simultaneous reading an instruction code and reading or writting a memory or peripheral as part of the execution of the previous instruction.
* Since it has two memories it is not possible for the CPU to mistakenly write codes into the program memory and therefore compute the code while it is executing.
* It is less flexible.
* It needs two independent memory banks.
* The modified Harvard architecture used DSPs multiport memory that has separate bus systems for program memory and data memory and input/output peripherals

* It may also have multiple bus system for program memory alone or for data memory alone
* These multiple bus system increases complexity of the CPU, but allow it to access several memory locations simultaneously, there by increasing the data throughput between memory and CPU.

VLIW Architecture :-

* The Very Long Instruction Word processing increase the number of instructions that are processed per cycle.
* It is essentially a concatenation of several short instructions and require multiple execution units, running in parallel, to carryout the instructions in a single cycle.
* It executes multiples instructions/cycle and use simpler, regular instruction sets
* The VLIW processor consists of architecture that reads a relatively large group of instructions and executes them at the same time.
* The VLIW processor combines many simple instructions into a single long instruction word that uses different registers.

* A language compiler or pre-processor separates program instructions into basic operations that are performed by the processor in parallel.

* These operations are placed into a "Very long instruction word" that the processor can then disassemble, and then transfer each operation to an appropriate execution unit.

Advantages of VLIW architecture:

1. Increased performance
2. Better compiler targets
3. potentially easier to program
4. Potentially scalable.
5. Can add more execution units, allow more instructions to be packed into the VLIW instruction.

Disadvantages of VLIW architecture :-

1. New kind of programmer / compiler complexity
2. program must keep track of instruction scheduling.
3. Increased memory use
4. High Power Consumption
5. Misleading MIPS ratings

Multiply Accumulate Unit (MAC)

→ MAC operation is the basis of many digital signal processing algorithms like digital filtering.

→ The term "digital filter" refers to an algorithm by which a digital signal or sequence of numbers is transformed into a another sequence of numbers termed the output digital signal.

→ In general FIR filters are preferred in lower order solutions and since they don't employ feedback, they exhibit naturally bounded response.

→ They are simpler to implement and require one RAM location and one co-efficient for each other.

→ The FIR filters output is given by,

$$y(n) = \sum_{k=0}^{N-1} x(k)h(n-k)$$

→ The output of an FIR filter is simply a finite length weighted sum of the present and previous inputs to the filter.

* characteristics of a typical fixed point MAC include,

1. 16 X 16 bit 2's complement inputs
2. 16x16 bit multiplier with 32-bit product in 25 ns.
3. 32/40 bit accumulator.

* In the TMS 320C50, the FIR equation can be efficiently implemented using the instruction pair,

```
RPT  NMI
MACD HNMI,XNM1
```

* RPT NM1 operation :-

→ It loads the (N-1) into the repeat instruction counter, and causes the multiply-accumulate with data move (MACD) instruction following it to be repeated N times.

* MACD operation :-

→ Multiplies data sample x(n-k) in the data memory by the co-efficient, h(k) in the program memory.

→ Adds previous product to the accumulator

→ Implements the unit delay, symbolized by shifting the data sample x(n-k) up to update the tapped delay line.

MAC function :-

* The MAC speed applies both to finite Impulse response (FIR) and infinite impulse response (IIR) filters.

* The multiply-accumulate step performs the following:-

→ Reads a 16-bit sample data

→ Increments the sample data pointer by 2

→ Reads a 16-bit co-efficient

→ Increments the co-efficient register pointer by 2

→ Sign multiply (16-bit) data and co-efficient to yield a 32-bit result.

→ Adds the result to the contents of a 32-bit register pair for accumulate.

* The TMS320 C54x multiply-accumulate (MAC) unit Performs a 16x16 → 32 bit fractional multiply-accumulate operation in a single instruction cycle.

* The multiplier supports signed/signed multiplication, signed/unsigned multiplication, and unsigned/unsigned multiplication.

* Many instructions using the MAC unit can optionally specify automatic round-to-nearest rounding.

Architecture of TMS320C50:-

→ It is fabricated with CMOS integrated Circuit technology.

→ It is a fixed point, 16-bit processor running at 40MHz.

→ The single instruction execution time is 50nsec.

→ Its architectural design is based on the combination of advanced Harward architecture, on chip peripherals and onchip memory.

→ Onchip memory include 10k words of the RAM and 2k words of the ROM.

Functional Block-diagram:-

The functional blockdiagram of TMS320C5x can be divided into four sub blocks.

1. Bus Structure
2. Central processing Unit
3. Onchip Memory
4. Onchip Peripherals

1. Bus structure:-

Many DSP applications are accomplished using single-cycle multiply/accumulate instruction with a data move option.

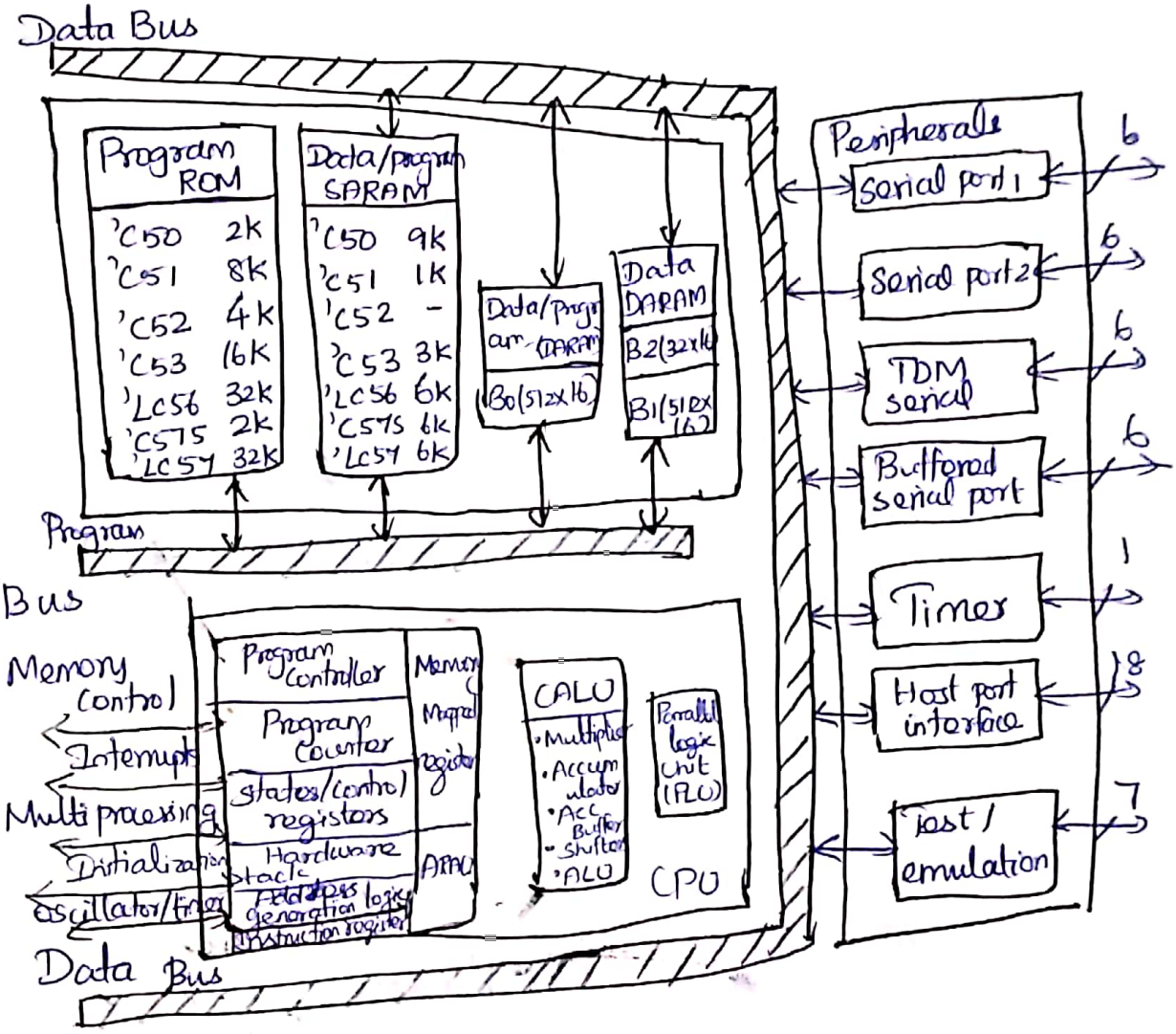
The C5x architecture has four buses:-

(i) program bus (PB)

(ii) Program address bus (PAB)

(iii) Data read bus (DB)

(iv) Data read address bus (DAB)



(i) PB :-

→ It carries the instruction code and immediate operands from program memory to the CPU.

(ii) PAB :-

→ It provides address to program memory space for both read and write.

DB :-

The data read bus interconnects various elements of the CPU to data memory space

DAB :-

The data read address bus provides the address to access the data memory space.

Central Processing unit :-

It consists of the following elements,

(i) central arithmetic logic unit (CALU)
(ii) Parallel logic Unit (PLU)
(iii) Auxilary register arithmetic unit (ARAU)
(iv) Memory mapped registers
(v) Program Controller.

(i) CALU :-

* It performs 2's complement arithmetic
* It consists of 16 bit x 16 bit parallel multiplier, 32 bit accumulator (ACC), 32-bit ACC buffer (ACCB), product register (PREG) and additional shifters.
* The 16x16 bit hardware multiplier is capable of computing a signed or an un-signed 32-bit Product in a single machine cycle.

* The 16-bit temporary register 0 (TREG0) holds one of the operands for the multiplier, and the other input is from the data bus or the program bus. The product register holds the product.

* The LT (Load TREG0) instruction normally loads TREG0 to provide one operand from the data bus and MPY instruction provides the second operand for multiplication operations.

* One input to ALU comes from accumulator and the other input from the product register of the multiplier, the accumulator buffer (ACCB) or the output of the scaling shifter.

Parallel Logic Unit (PLU)

* It is a second logic unit that executes logic operations on data without affecting the contents of the accumulator.

* It can directly set, clear, test or toggle multiplier bit in a status / control register on any data memory location.

* Auxilary Register Arithmetic Unit (ARAU):

* The C5x consists of a register file containing eight auxiliary register (AR0 - AR7) each of 16 bit length, a 3 bit auxiliary register pointer (ARP) and an unsigned 16-bit ALU.

x Auxiliary registers are used for indirect addressing of the data memory or for temporary data storage.

* The ARs and ARP can be loaded from data memory the ACC or the PREG or by an immediate operand, defined in the instruction.

* Index Register:

* The 16-bit index register (INDX) is used by the ARAU as a step value to modify the address in the ARs during indirect addressing.

x The INDX can be used to increment or decrement the address in steps larger than 1.

* Auxiliary Register compare register (ARCR)

→ The 16-bit ARCR is used for address boundary Comparision

→ It limits blocks of data and supports logical comparisions between the current AR and ARCR in conjunction with the CMPR instruction.

Block Move address Register (BMAR) :-

* The 16-bit BMAR holds an address value of the source destination space of a block move.

* The BMAR can also hold the address of an operand in program memory for a multiply accumulate operation.

Block Repeat Registers :- (RPTC, BRCR, PASR, PAER) -

RPTC :- The 16-bit repeat count register hold the repeat count in a repeat single operation and is loaded by the RPT and RPTC instructions.

BRCR :- The 16-bit block repeat counter register holds the count value for the block repeat feature.

PASR : The block repeat program address start register indicates the 16-bit address where the repeated block of code starts.

PAER : The block repeat Program Address End Register indicates the 16-bit address where the repeated block of code ends.

Auxiliary Registers (AR0 - AR7)

The eight 16-bit auxiliary registers (AR0-AR7)

can be accessed by the CALU and modified by the ARAU or the PLU

* It provides 16-bit address for indirect addressing to data space.

* The ARs can also be used as general purpose registers or counters.

Instruction Register (IPEG)

* It holds the opcode of the instruction being executed.

Interrupt Register (IMP, IFR).

* It masks specific interrupts at required time individually.

Status Register

* It contain status and control bits for the CPU.

Memory Mapped Registers :-

* The 'C5x has 96 registers mapped into page 0 of the data memory space (00-5Fh)

* This memory mapped register space contains various control and status registers including those for CPU, serial port, timer and software wait state generator.

Program Controller :

* It contains logic circuitary that decodes the operational instructions, manages the CPU pipeline, stores the status of CPU operations and

decodes the conditional operations. It consists of the following elements

(i) Program Counter
(ii) Status and Control registers
(iii) Hardware Stack
(iv) Address generation logic
(v) Instruction register.

On-chip Memory :-

The 'C5x architecture has a total memory address range of 224k words x 16 bits. The memory space is divided into four memory segments.

64k - Word - Program Memory space
64k - Word - local data memory space
64k - Word - input / output ports
32k - Word - Global data memory space.

The large on-chip memory of 'C5x includes,
(i) Program read only memory
(ii) Data / program single access RAM (SARAM)
(iii) Data / program dual access RAM (DARAM).

On-chip Peripherals :-

The on-chip peripherals interface connected to the 'C5x CPU include,

(i) Clock generator
(ii) Hardware timer
(iii) Software programmable wait state generators
(iv) General purpose I/O pins
(v) Parallel I/O ports
(vi) Serial port interface
(vii) Buffered serial port
(viii) Time-division multiplexed (TDM) serial port
(ix) Host port interface
(x) User unmaskable interrupts

Applications of DSP :-

1. Speech coding :-

* Depending on the application, the sampling rate F for speech will normally lie in the range 6-20kHz.

* Before sampling the speech signal is passed through an anti-aliasing filter to remove the frequency components above F/2.

* It is then sampled at a rate >F, the Nyquist rate.

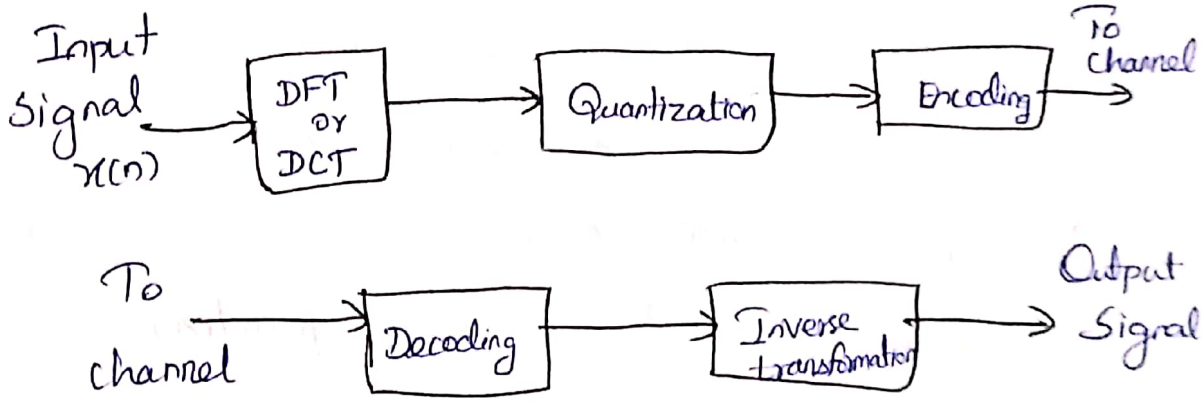
* The sampled signal is represented by a 12-bit code for high quality speech processing applications.

The different methods of speech coding techniques are

1. Waveform coding → (i) Pulse code Modulation (PCM)
   (ii) Adaptive pulse code Modulation (APCM)
   (iii) Differential Pulse code Modulation (DPCM)
   (iv) Adaptive Differential pulse code Modulation (ADPCM)
   (v) Delta Modulation (DM)
   (vi) Adaptive Delta Modulation (ADM)

2. Frequency domain coding → (i) Transform Coding
   (ii) Adaptive Transform coding
   (iii) Subband coding

Transform coding :-



*In which a block of input samples is taken and linearly transformed using DFT or DCT computation via a fast fourier transform (FFT) algorithm.

* Then the transformation of the signal is efficiently coded by assigning bits to transform co-efficients
* At receiver an inverse transformation is used to reconstruct the speech signal.
* The number of bits assigned to each transform co-efficient is proportional to its corresponding spectral energy value.

2) Sub-band coding :-

* In which the input signal is first split into number of non-overlapping frequency bands by bandpass filters.
* The output of each BPF is decimated or down sampled by a factor M.
* It can achieved by retaining every M^{th} sample of the filter output and discarding M-1 samples.
* The output of the decimator is quantized using the techniques like PCM, DPCM etc. and transmitted.
* At the receiver the inverse sequence of events takes place.

* The subbands are demultiplexed and decoded and then each subband signal is interpolated by inserting zeros to replace the samples discarded during decimation at the transmitter.
* The output of the interpolator is appled to bandpass filters.
* All bandpass filter outputs are summed together to reproduce the original signal.

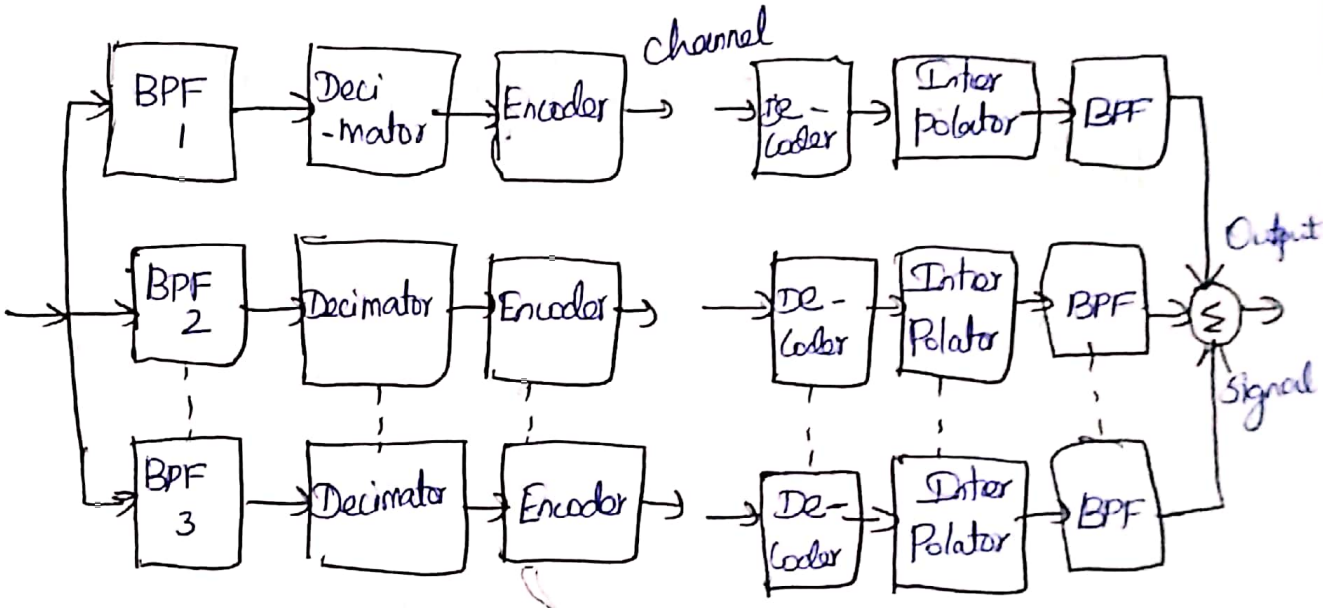
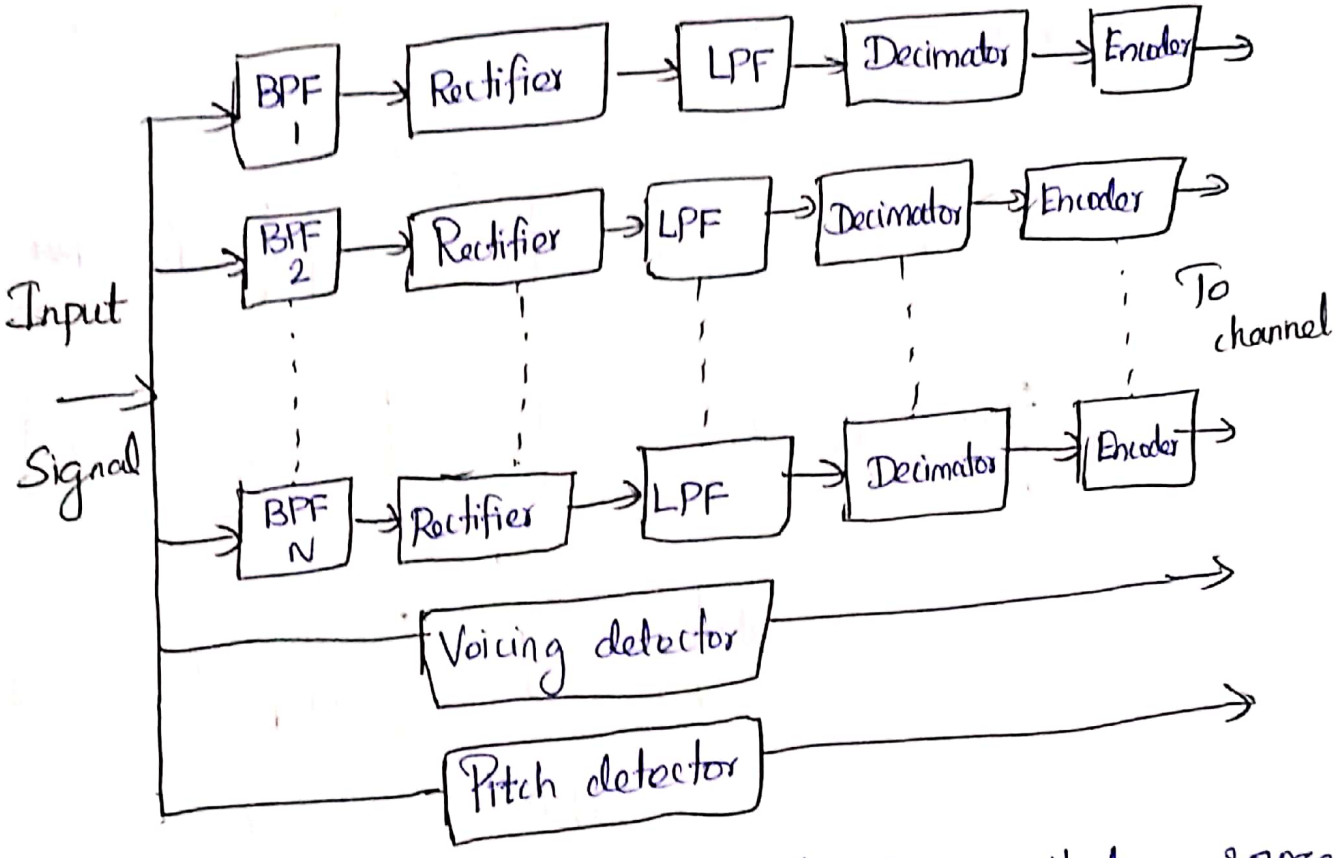


fig: Subband coding

3. Channel Vocoder :-

* It consists of both an analyser and a synthesizer.
* The analyzer consists of a number BPF with impulse response $w(n)\cos w_i n$ followed by a FWR and LPF.

* The rectifier and LPF serve as an envelope detector.
* The resulting information together with the amplitude of the signal forms the representations for the speech signal..
* These parameters are sampled and quantized for transmission



* At the receiver a wideband excitation source is generated using voiced/unvoiced signal.
* The generator is used to switch in a random voice or a pulse generator with the fundamental frequency of the pulse generator.

* The received channel signals are used to modulate the amplitude of the excitation signal, which excites the corresponding BPF.
* The bandpass filter outputs are summed to produce the speech signal.

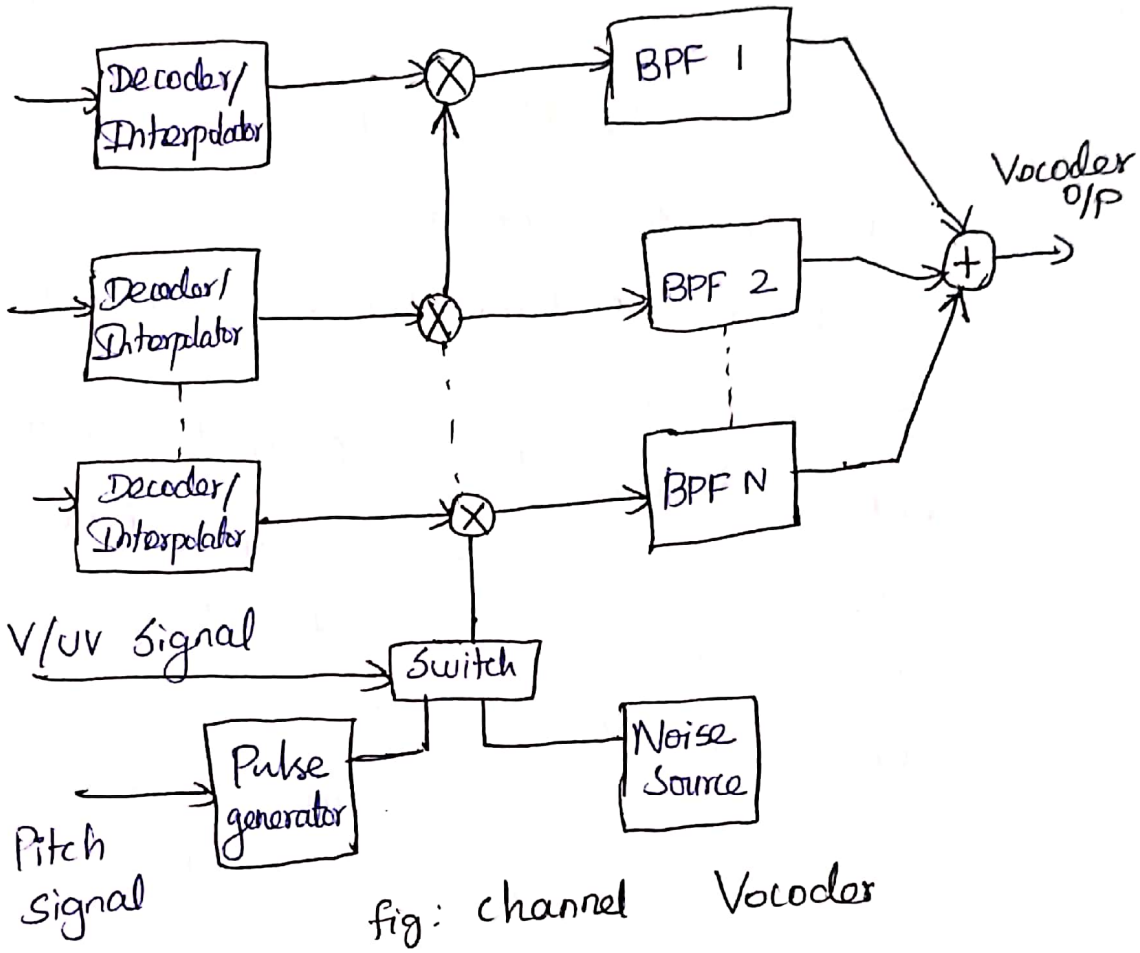


fig: channel Vocoder

4) Homomorphic Vocoder :-

* In which, the cepstrum is computed once in every 10-20 sec.
* The Vocal tract cepstral co-efficients are separated from the excitation co-efficients by a linear filtering operation.

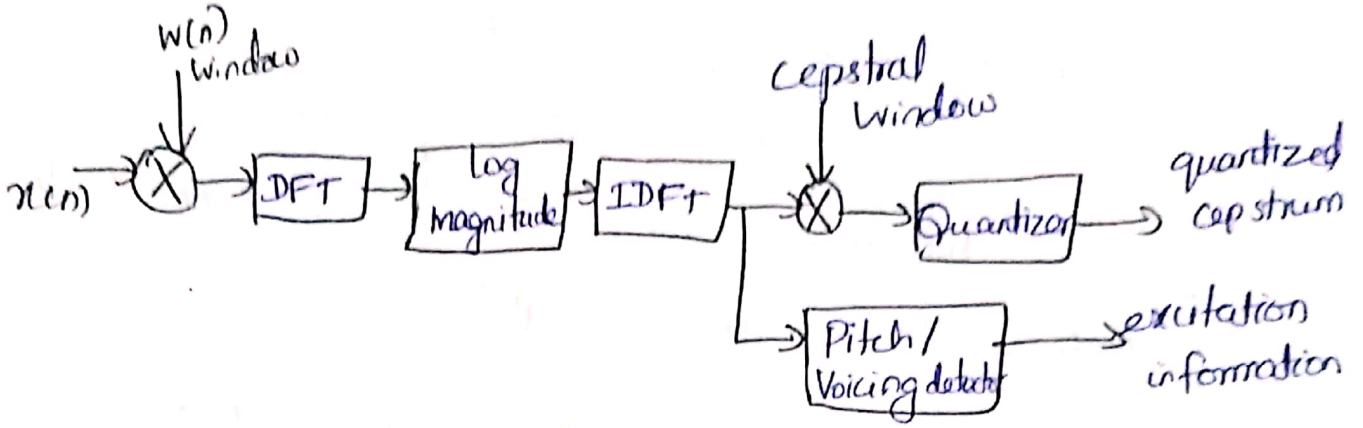


fig: Homomorphic vocoder.

* The excitation information together with the quantized cepstrum values are transmitted to the synthesizer.
* At synthesizer the vocal tract cepstral coefficients are fourier transformed, exponentiated and inverse fourier transformed to produce the vocal tract impulse response.
* By convolving this impulse response with a voiced / unvoiced signal the original speech is reconstructed.

5) Digital processing of Audio Signals:-

* In the digital tape recorder each input signal is applied to a LPF which eliminates high frequency noise.

* The output from the filter is sampled and converted into digital words.
* The bit streams are multiplexed. In addition bit for timing, for correcting errors, parity checking and block scaling are introduced.
* A modulator is used to convert the composite bit-stream into a series of analog pulses.
* During reproduction the signal is decoded unpacked and distributed to each output.

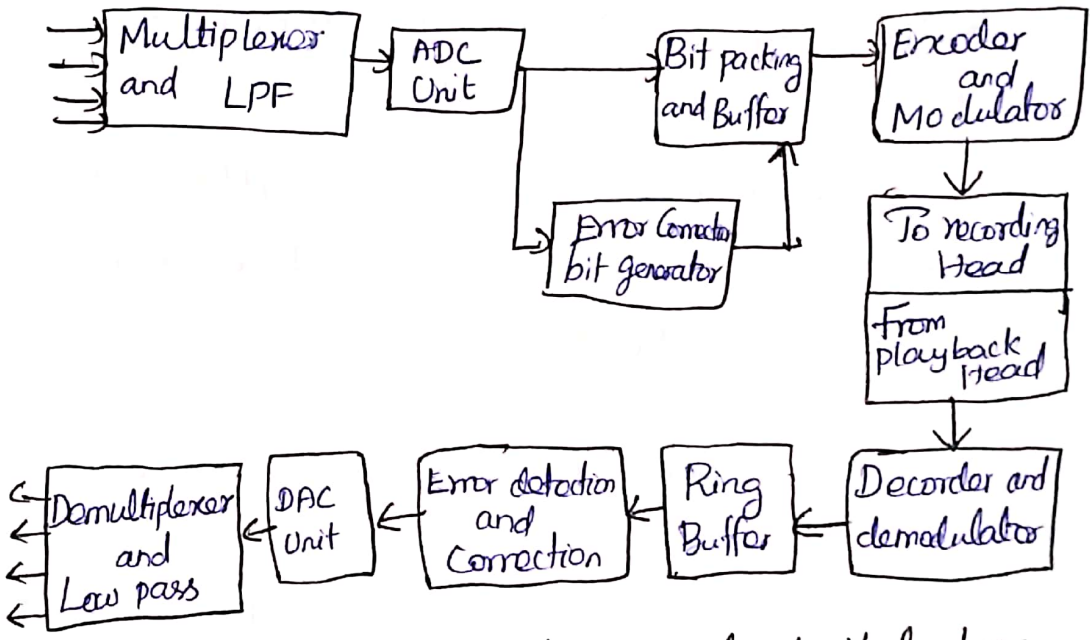


fig: Block diagram of digital tape recorder system

6) Radar signal processing :-

* It consists of a data processing system, control unit, signal generator, transmitting and receiving antenna, matched filter.

* The Signal generator provides test signal used by the processor.
* This signal is converted into RF range and then amplified by the modulator.
* The RF signal is transmitted towards the target by the transmitting antenna.
* The reflected energy from the target can be used to measure the target characteristics including position.
* The signal received is applied to a demodulator to convert it to baseband.
* The baseband signal is applied to a signal processor which performs three major signal processing functions.

They are,

1. Matched filtering for the received waveforms
2. Thresholding that reduces the data volume and rate to values that are appropriate for the DPS.
3. Estimation of target position in range, angle and velocity.

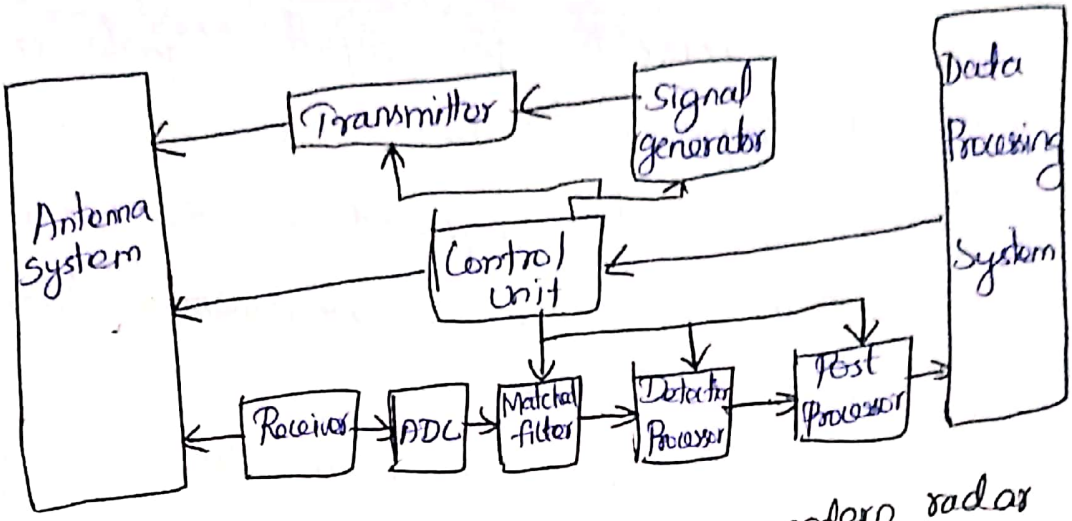


fig : Block diagram of a modern radar system.

DSP based measurement System:-

* DSP measurement systems have the capability of implementing different functions on the same hardware architecture.

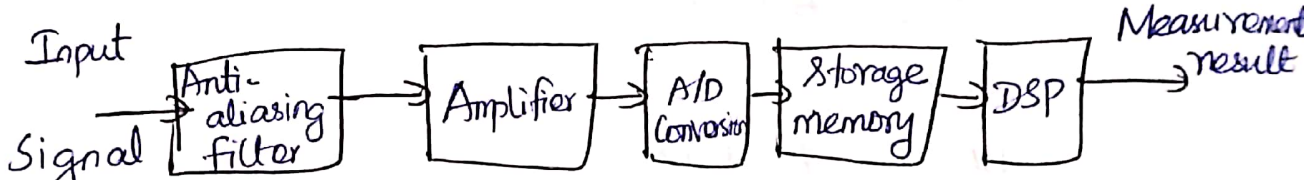


fig: Block diagram of a DSP-based measurement system.

Anti-aliasing filter :-

→ The antialiasing filter removes the possible aliasing effects.

Amplifier :-

→ It amplifies the output of the antialising filter.

A/D Conversion unit :-

It converts the input signals into a digital representation, with the resolution and sampling rate required by the correct execution of the measurement algorithm.

DSP Unit :-

It performs the measurement algorithm by the specified time slot.

Example programms :-

Ex.1. To write a program for integer multiplication.

```
LDP     #100H
LACC    #037AH,0
SACL    0000,0
LACC    #012EH,0
SACL    0001,0
LT      0000          ; T=37A
MPY     001           ; P=37A*12E=0004 19EC
PAC                   ; ACC=0004 19EC
SACL    0002,0
SACH    0003,0
H:      B       H
.END
```

* It stores the data 37A at data memory 8000 and the data 12E at the data memory address 8001.

* The instruction LT OP1(0000h) loads the first operand from dma 8000 to T register.

* The instruction MPY OP2(0001h) multiplies the content of dma 8001 with the T-register.

* The result is stored in P-register.

* The instruction PAC transfers the content of P-register to accumulator.

* The SACL and SACH instruction store the lower and higher order results in dma 8002 and 8003.

5.2. To find the two's complement of a given number.

```
LDP     #100H
LACL    #5          ; LOAD 5 TO ACC
CMPL                ; COMPLEMENT ACC
ADD     #1
SACL    0000,0      ; STORE 2's COMPLEMENT RES
H:      B   H
```

* In which the accumulator is loaded with data 5.
* The CMPL instruction complements the accumulator value and the result is FFFA.
* Now add one with accumulator which gives the result as FFFB.
* This is the two's complement of 5 and the result is stored in data memory location 8000.