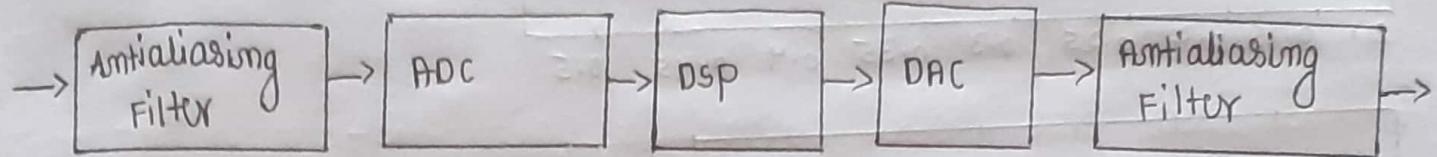


Explain the block diagram of a DSP system. also draw the typical in a DSP system

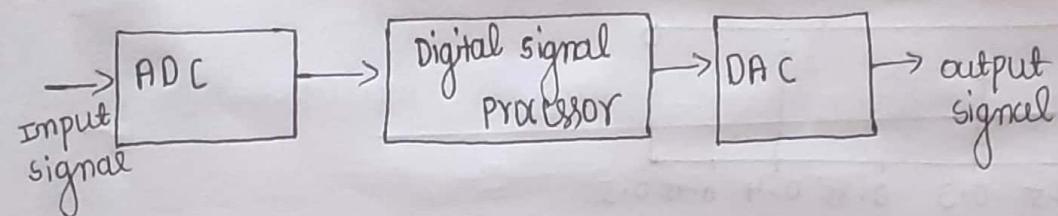
### A digital signal processing system



The block diagram of a DSP system

A computer or a processor is used for digital signal processing. Anti aliasing filter is a LPF which passes signal with frequency less than or equal to half the sampling frequency in order to avoid aliasing effect. similarly at the other end reconstruction filter is used to reconstruct the sample from the staircase output of the DAC

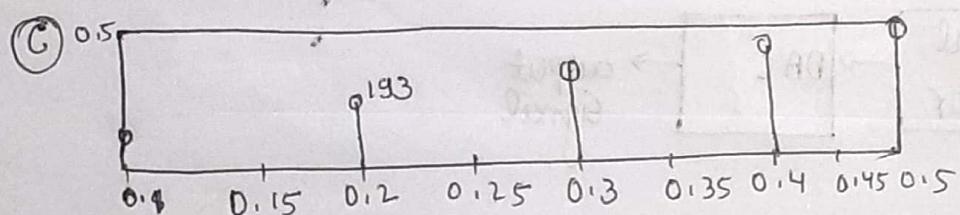
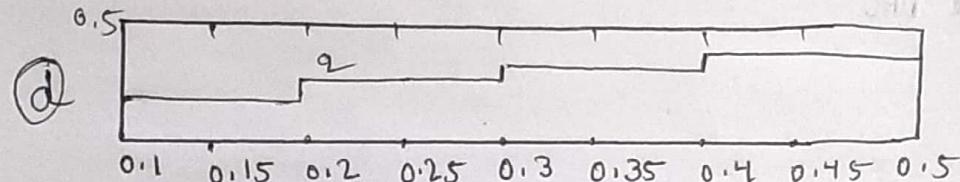
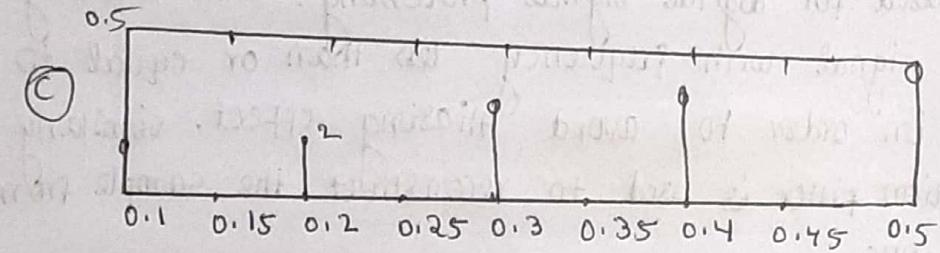
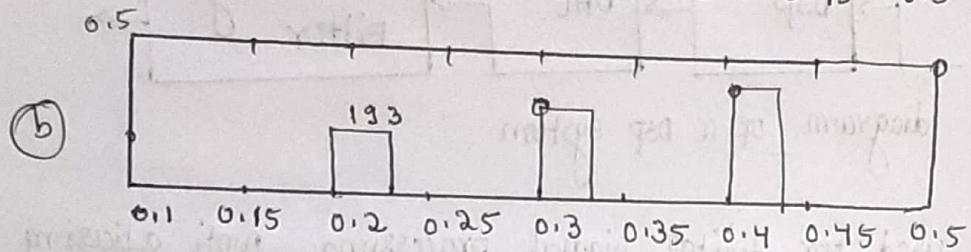
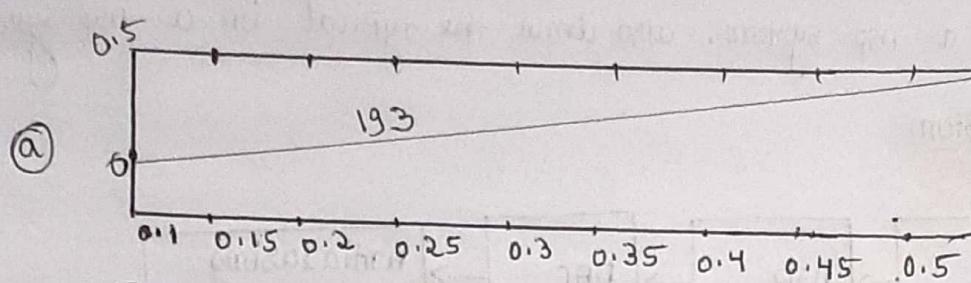
### Typical DSP system



DSP is a technique of performing the mathematical operations on the signals in digital domain.

As real time signals are analog in nature we need first convert the analog signal to digital then we have to process the signal in digital domain and again converted back to analog domain.

signals that occur in a typical DSP are



- ① continuous time signal    ② quantized signal    ③ sampled signal  
④ sampled data signal    ⑤ DAC output
- ⑥ mention the basic architecture features that should be provided by a programmable DSP devices.

## Basic Architectural Features

A programmable DSP device should provide instructions similar to a conventional microprocessor. The instruction set of a typical DSP device should include the following.

- a) Arithmetic operations such as ADD, SUBTRACT, MULTIPLY etc
- b) Logical operations such as AND, OR, NOT, XOR etc
- c) multiply and accumulate (MAC) operation
- d) Signal scaling operation

In addition to the above provisions, the architecture should also include

- a) on chip registers to store immediate results
  - b) on chip memories to store signal samples (RAM)
  - c) on chip memories to store filter coefficients (ROM)
- 3) what is the significance of shifter in DSP architecture? Explain the implementation of a 4-bit shift right barrel shifter.

### Shifter

Shifters are used to either scale down or scale up operands or the results.

- a. while performing the addition of  $N$  numbers each of  $n$  bits long the sum can grow up to  $n \log_2 N$  bits long
- b. similarly while calculating the product of two  $n$  bit numbers the product can grow up to  $2n$  bits long

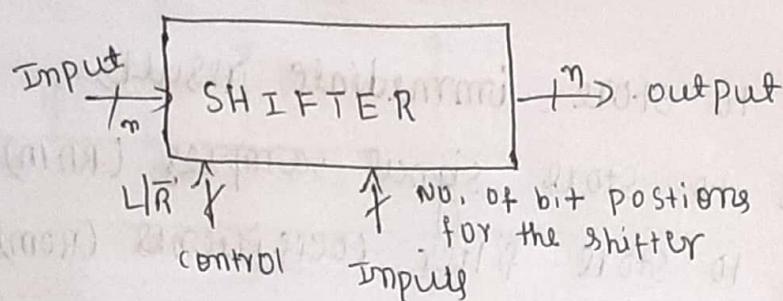
Q) Finally in case of addition of two floating-point numbers.

### Barrel shifter

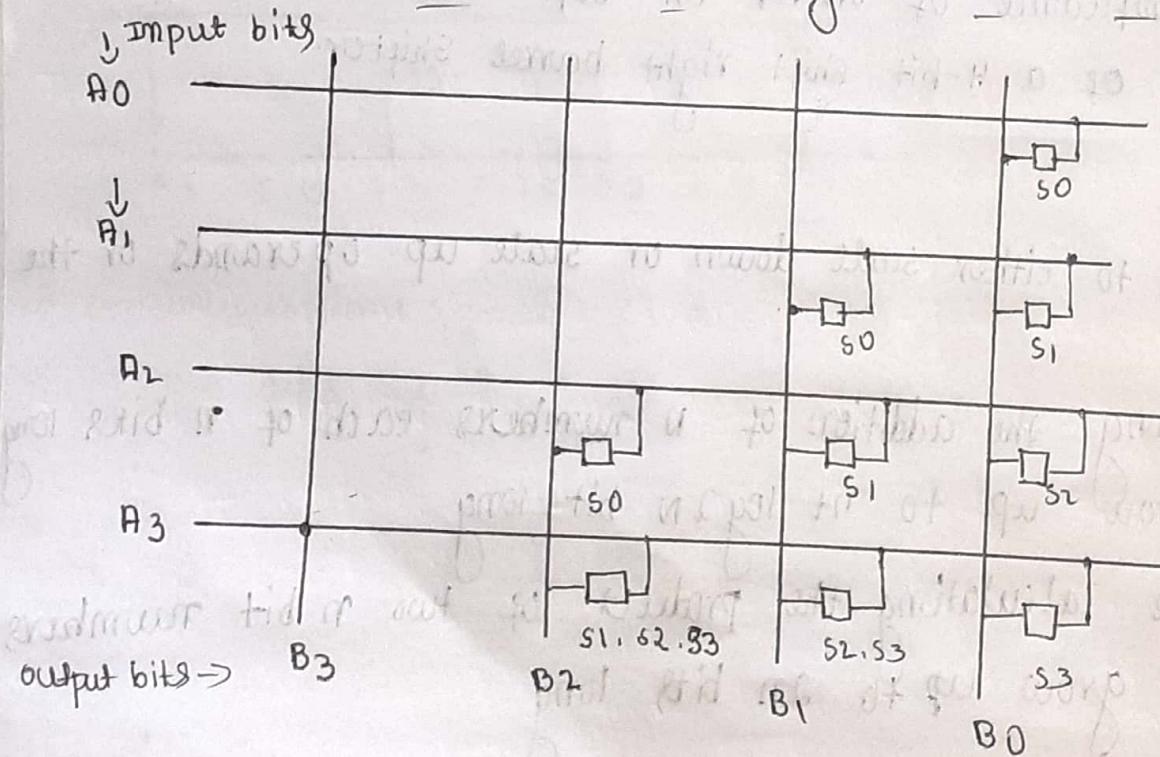
In conventional microprocessors normal shift registers are used for shift operation.

As it requires one clock cycle for each shift, it is not desirable for DSP application.

This can be accomplished using a barrel shifter.



Implementation of 4-bit shift Right Barrel shifter

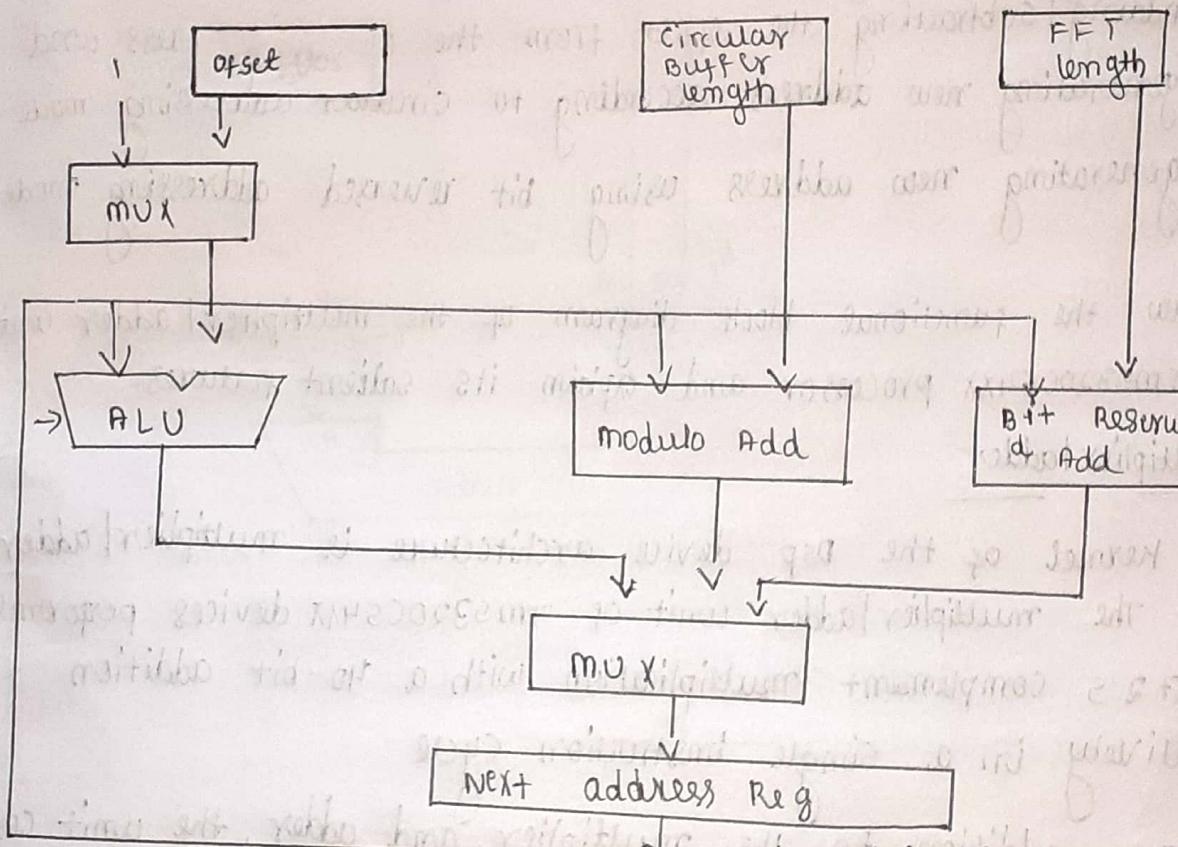


numbers.

Input	SHIFT (SWITCH)	Output ( $B_3, B_2, B_1, B_0$ )
A <sub>3</sub> A <sub>2</sub> A <sub>1</sub> A <sub>0</sub>	0 (S0)	A <sub>3</sub> A <sub>2</sub> A <sub>1</sub> A <sub>0</sub>
A <sub>3</sub> A <sub>2</sub> A <sub>1</sub> A <sub>0</sub>	1 (S1)	A <sub>3</sub> A <sub>2</sub> A <sub>1</sub> A <sub>0</sub>
A <sub>3</sub> A <sub>2</sub> A <sub>1</sub> A <sub>0</sub>	2 (S2)	A <sub>3</sub> A <sub>2</sub> A <sub>1</sub> A <sub>0</sub>
A <sub>3</sub> A <sub>2</sub> A <sub>1</sub> A <sub>0</sub>	3 (S3)	A <sub>3</sub> A <sub>2</sub> A <sub>1</sub> A <sub>0</sub>

is not desire-

- Q) with relevant block diagram explain address generation unit



### Address generation unit

The main job of the address generation unit is to generate the address of the operands required to carry out the operation to take required self address of intermediate.

operation. They have to work fast in order to satisfy the timing constraints. As the address generation unit has to perform some mathematical operations in order to calculate the operand address, it is provided with separate ALU.

Address generation typically involves one of the following operations.

- a. getting value from immediate operand, register or a memory location
- b. incrementing / decrementing the current address
- c. Adding / subtracting the offset from the current address
- d. Adding / subtracting the offset from the current address and generating new address according to circular addressing mode
- e. generating new address using bit reversed addressing mode

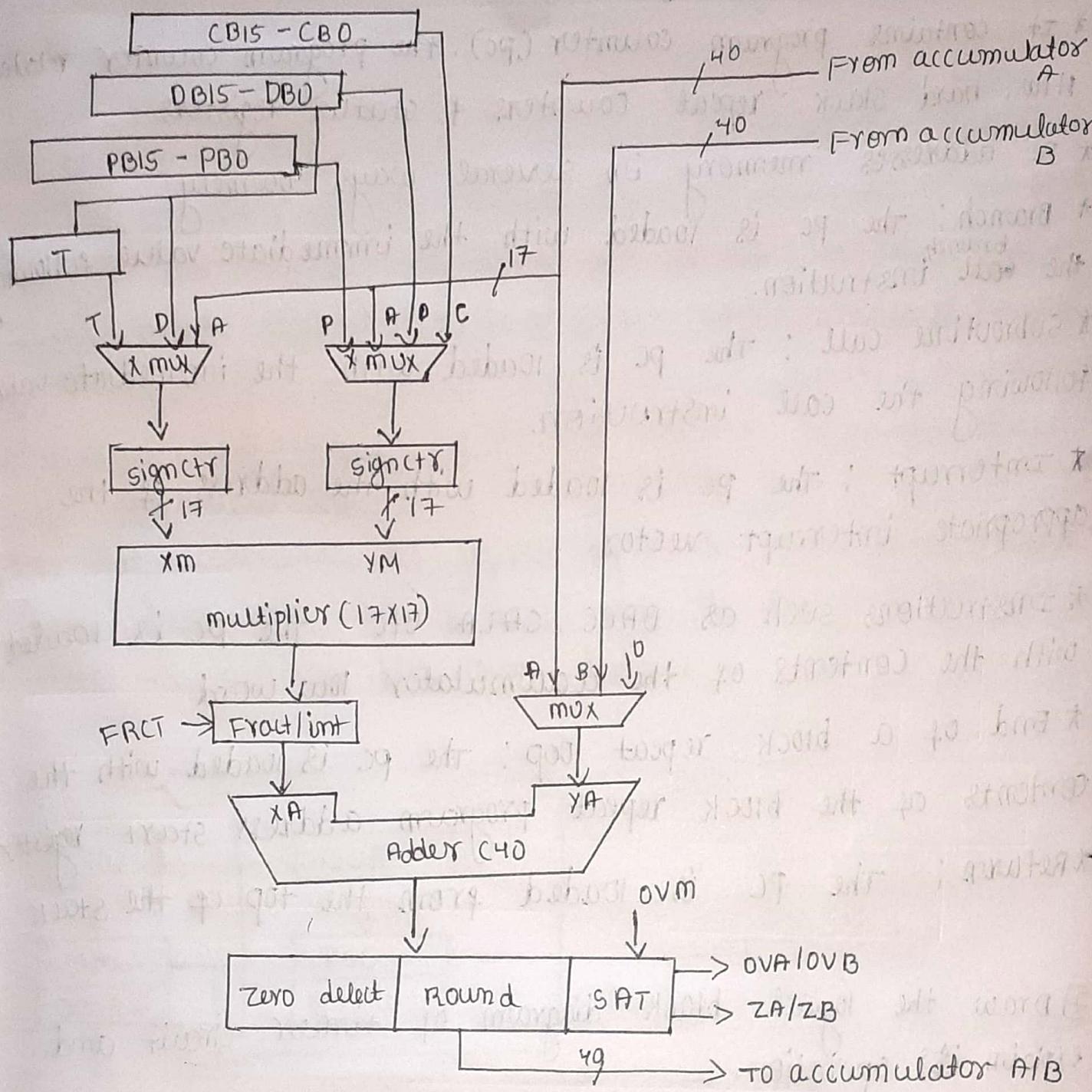
5) Draw the functional block diagram of the multiplier / adder unit of TMS320C54XX processor and explain its salient features.

### multiplier ladder

The kernel of the DSP device architecture is multiplier / adder unit. The multiplier / adder unit of TMS320C54XX devices performs  $17 \times 17_2$ 's complement multiplication with a 40-bit addition effectively in a single instruction cycle.

In addition to the multiplier and adder, the unit consists of control logic for integer and fractional computations and a 16-bit temporary storage register  $T$ .

The compare select & store unit (CSSU) is a hardware unit specifically incorporated to accelerate the add / compar / select operation.



Functional diagram of the multiplier ladder unit of TMS320C54XX processor

6] write a note on program control

- \* It contains program counter (pc), the program counter related Hlw, hard stack, repeat counters & status registers.
- \* PC addresses memory in several ways namely:
- \* Branch: The PC is loaded with the immediate value following the ~~call~~<sup>branch</sup> instruction.
- \* Subroutine call: The PC is loaded with the immediate value following the call instruction.
- \* Interrupt: The PC is loaded with the address of the appropriate interrupt vector.
- \* Instructions such as BACC, CALA etc: The PC is loaded with the contents of the accumulator low word.
- \* End of a block repeat loop: The PC is loaded with the contents of the block repeat program address start register.
- \* Return: The PC is loaded from the top of the stack.

7) Draw the logical block diagram of timer circuit and explain its operation.

It consists of 3 memory mapped registers

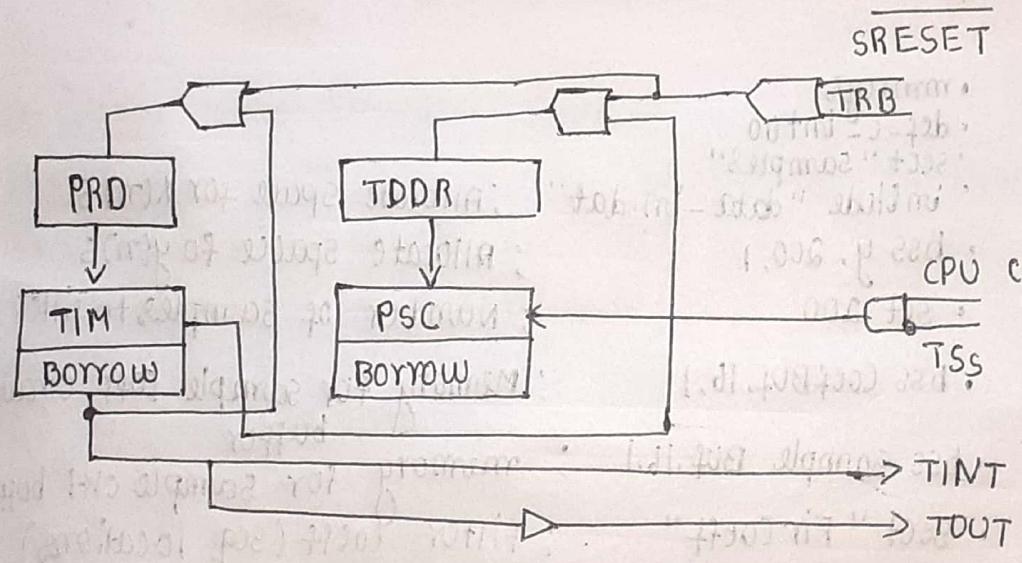
- \* The time register (TIM)
- \* Time period register (PRD)
- \* Time control register (TCR)
  - pre scaler block (PSC)
  - TDDR (Time Divider Down ratio)

The time register (TIM) is a 16-bit memory-mapped register that decrements at every pulse from the prescaler block (PSC).

The time period register (PRD) is a 16-bit memory-mapped register whose contents are loaded onto the TIM whenever the TIM decrements to zero or the device is reset (SRESET).

The timer can also be independently reset by using the TRB signal. The timer control register (TCR) is a 16-bit memory-mapped register that contains status and control bits.

The prescaler block is also an on-chip counter. Whenever the prescaler bits count down to 0, a clock pulse is given to the TIM register that decrements the register by 1. The TDDR bits contain the divide-down ratio which is loaded onto the prescaler block after each time the prescaler bits count down to 0.



Logical block diagram of timer circuit

8] write an assembly language program for TMS320C54XX processor to implement an FIR filter

Solution

Program to implement an FIR filter

It implements the following equation:

$$y(n) = h(N-1) \times x(n-(N-1)) + h(N-2) \times x(n-(N-2)) + \dots + h(1) \times x(n-1) + h(0) \times x(n)$$

where  $N$  = number of filter coefficients = 16

$h(N-1), h(N-2), \dots, h(0)$  etc are filter coefficients (15 numbers)

The coefficients are available in file: coeff-fir.dat

$x(n-(N-1)), x(n-(N-2)), \dots, x(n)$  are signal samples (integers)

The input  $x(n)$  is received from the data file: data-in.dat

The computed output  $y(n)$  is placed in a data buffer.

	• mmyregs	
	• def_c_int00	
in samples	• sect "samples"	
	• include "data-in.dat"	; Allocate space for $x(n)$ s
out samples	• bss y, 200, 1	; Allocate space for $y(n)$ s
sample cnt	• set 200	; Number of samples to filter
	• bss CoefBuf, 16, 1	; Memory for sample coeff, circular buffer
	• bss Sample Buf, 16, 1	; memory for sample clk buffer
	• sect "Fir coeff"	; Filter coeff (seq locations)
Fir Coeff	• include "coeff-fir.dat"	
	• set 15	; N - 1

```

    .text
-C-int00:    STM # OUT samples, AR6
              RPT # Sample Cnt
              ST # 0, * AR6 +
              STM # IN samples, AR5
              STM # OUT samples, AR6
              STM # Sample Cnt, AR4

CALL fir-unit
SSBX SA M

loop:      LD * AR5 + A
          CALL fir-filter
          STH A, 1, * AR6 +
          BANZ loop, * AR4 -
          NOP
          NOP
          NOP

```

### FIR Filter Initialization Routine

; This routine sets AR2 as the pointer for the sample circular buffer  
; AR3 as the pointer for coefficient circular buffer,  
; BK = Number of filter taps - 1  
; AR0 = 1 = circular buffer pointer increment

fir-unit:

ST # COEFBUF, AR3

ST # Sample BUF, AR2

STM # NM1, BK

RPT # NM1

MVPPD # FIRDEF, \* AR3 + Y1

RPT # NM1 - 1

ST # 0h, \* AR2 + Y1



## STM #1, ARO

RET  
NOP  
NOP  
NOP

## FIR FILTER Routine

; Enter with A = the current sample  $x(n)$  - an integer, AR2 pointing to the location for the current sample  $x(n)$ , and AR3 pointing to the 15 coefficient (N-1). Exit with A =  $y(n)$  as a 15 number.

fir-filter:

STL A, \*AR2 + 0%  
RPTZ A, # Nm1  
MAC \*AR3 + 0%, \*AR2 + 0%, A

RET

NOP

NOP

NOP

end

⑨ write a program for implementation of the decimation filter for TMS320C54XX processor.

## Implementation of decimation filter

It implements the following equation:

$$y(m) = h(4)x(3n-4) + h(3)x(3n-3) + h(2)x(3n-2) + h(1)x(3n-1) + h(0)x(3n)$$

followed by the equation

$$y(n+1) = h(4) \times (3n-1) + h(3) \times (3n) + h(2) \times (3n+1) + h(1) \times (3n+2) + h(0) \times (3n+3)$$

and so on a decimation factor of 3 and a filter length of 5.

- mmregs

- dec\_c\_imt00

In samples

- sect "samples"

out samples

- #include "data-in.dat"

samplecnt

- bss y, 80, 1

- set 24 D

- sect "Fircoeff"

Fircoeff

- include "coeff-dec.dat"

Nm1

- set 4

- bss coeffbuf, 5, 1

- bss sample Buf, 5, 1

- t ex1

-c-imt00:

STM # outsamples, AR6

RPT # sample cnt

ST # 0, \*AR6 +

STM # insamples AR5

STM # outsamples AR6

STM # Samplecnt, AR4

CALL dec-init

loop:

CALL dec-filter

STH A, 1, \*AR6 +

BPNZ loop, \*AR4 -

nop

nop

nop



## Decimation filter Initialization Routine

This routine sets AR2 as the pointer for the sample circular buffer, and AR3 as the pointer for coefficient circular buffer  
BK = Number of filter taps ; AR0=1=circular buffer pointer increment.

dec\_init:

ST # CoeffBuf, AR3

ST # Sample Buf, AR2

STM # Nm1, BK

RPT # Nm1

MVPPD # FirCoeff, \*AR3+%

RPT # Nm1

ST # 0h, \*AR2+%

STM # 1, AR0;

RET

nop

nop

nop

## FIR Filter Routine

Enter with A = x(n), AR2 pointing to the circular sample buffer + AR3 to the circular coeff buffer, AR0=1,

Exit with A=y(n) as q15-number

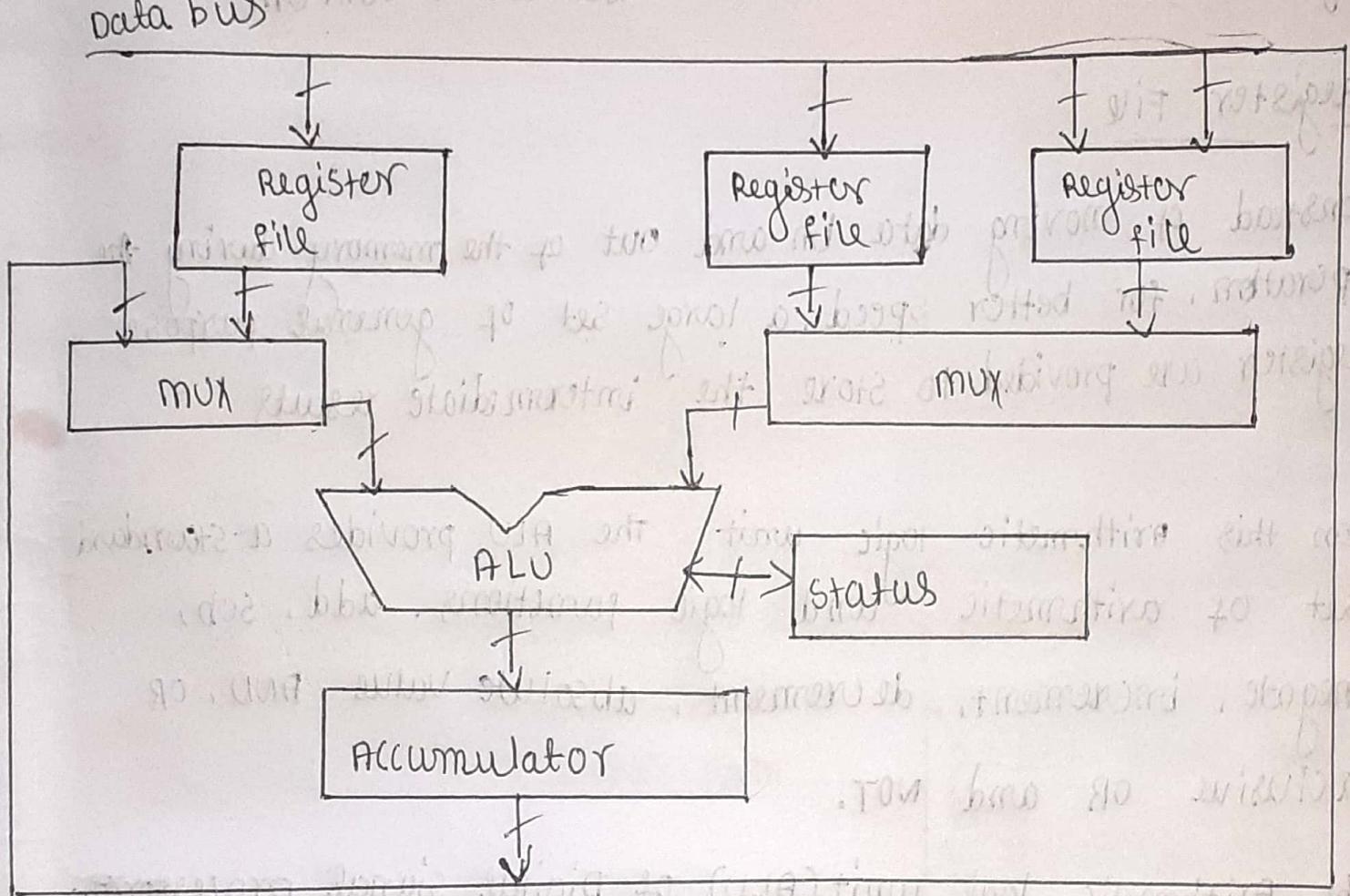


dec-filter:

```
LD *AR5+A  
STL A, *AR2+0%.  
LD *AR5+A  
STL A, *AR2+0%.  
LD *AR5+A  
STL A, *AR2+0%.  
RPTZ A, #NMI  
MAC AR3+0%, *AR2+0%,A  
RET  
nop  
nop  
nop  
.end
```

10 Explain the register pointer updating algorithm for circular buffer

II) with relevant block diagram. explain the various features of arithmetic and logic unit of DSP processor.



Arithmetic Logic unit of a DSP

A typical DSP device should be capable of handling arithmetic instruction like ADD, SUB, INC, DEC etc. and logical operation like AND, OR, NOT, XOR etc.

### Status Flags

ALU includes circuitry to generate status flags after arithmetic and logic operation.

## overflow management

Depending on the status of overflow and sign flags, the saturation logic can be used to limit the accumulator content.

## Register File

Instead of moving data in and out of the memory during the operation, for better speed, a large set of general purpose register are provided to store the intermediate results.

In this arithmetic logic unit the ALU provides a standard set of arithmetic and logic functions, add, sub, negate, increment, decrement, absolute value AND, OR Exclusive OR and NOT.

The Arithmetic logic unit (ALU) of Digital signal processor(DSP) performing arithmetic and logic operations.

Two divide primitives are also provided. The ALU takes two 16-bit inputs.

12] compare the architectural features of TMS320C25 and DSP56000 processor

Architectural feature	TMS320C25	DSP56000
1] Data representation formats	16-bit fixed	24-bit fixed point
2] Hardware multiplier	16x16	24x24
3] ALU	32 bits	56 bits
4] Internal buses	16-bit program bus 16-bit data bus	* 24-bit program bus 2x 24-bit data buses 24-bit global data bus
5] External buses	16-bit Program / data bus	* 24-bit program / data bus
6] On-chip Memory	544 words RAM 4K words ROM	* 512 words PROM 2x 256 words data RAM 2x 256 words data ROM
7] off-chip memory	64K words program 64K words data	64K words program 2x 64K words data

13] Draw the functional block diagram of barrel shifter of TMS320C54XX processor and explain the significance of each block.