Unit - V Introduction to Digital Signal Processors

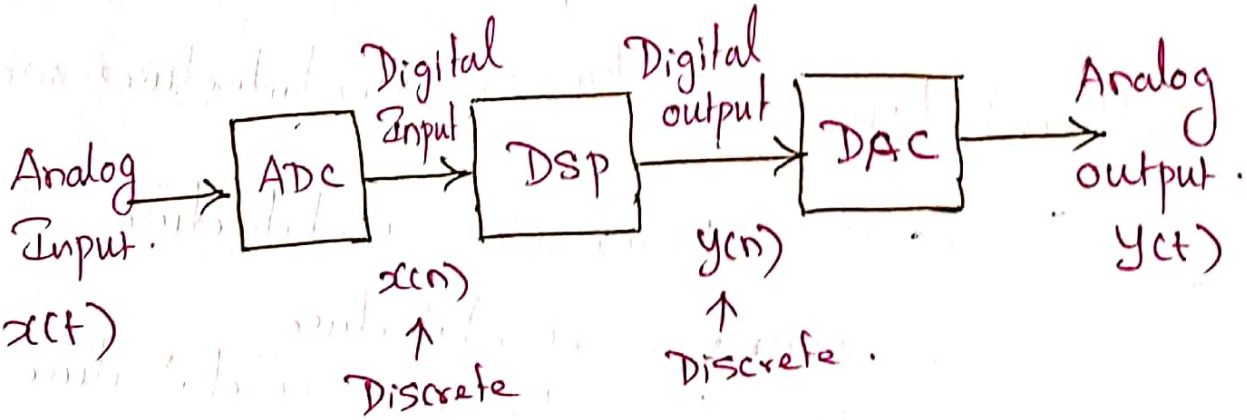
DSP functionalities - Circular buffering DSP architecture - Fixed and floating Point architecture Principles - Programming Application examples.

Dsp Processor :-

A digital Signal Processor (DSP) is a Specialized microProcessor (or a SIP block) with its architecture optimized for the operational needs of Digital Signal Processing.

Basic Block Diagram of Dsp Processor :-

The basic Block diagram of DSP Processor is shown in figure. which Consists of Analog to Digital convertor used to convert analog input signal to digital signal. and DSP Processor which



is usually a filter (FIR (or) IIR) which Perform. Operation and Digital to analog Convert Converts the Digital output to analog output.

Common features required for DSP Processor are as follows

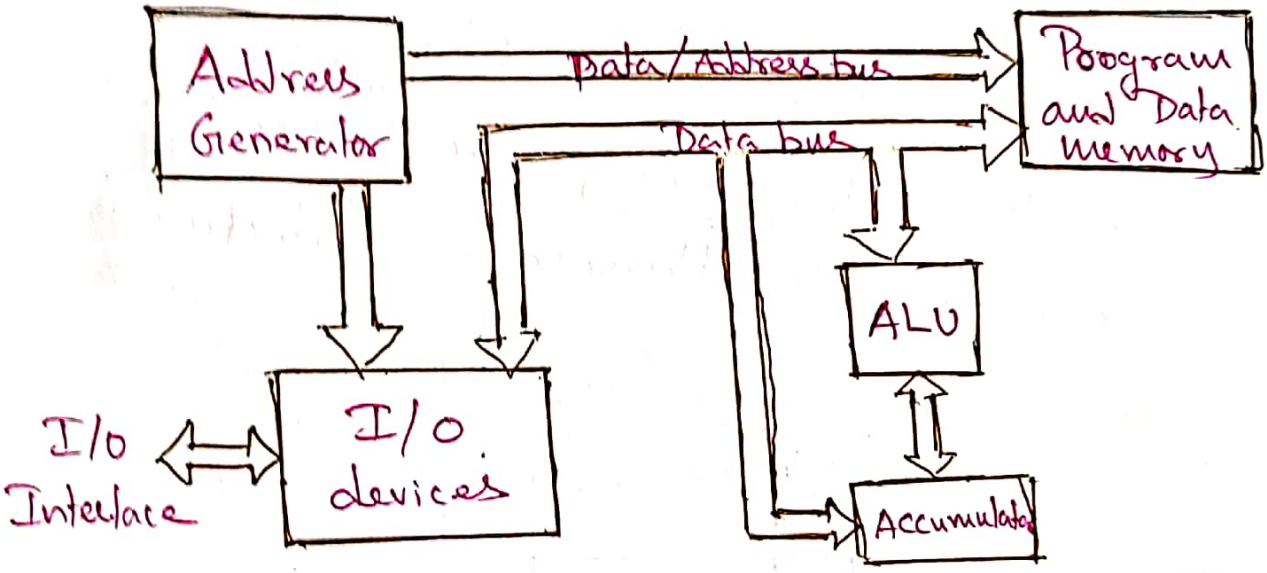
* Harvard Architecture
* Dedicated MAC [Multiplier - Accumulator]
* Pipelining
* Hardware Circular buffer.
* DMA.
* Saturation Arithmetic.
* Cache memory

Different types of DSP Architectures :-

1. Von - Neumann architecture
2. Harvard architecture
3. Modified Harvard architecture

Von - Neumann Architecture :-

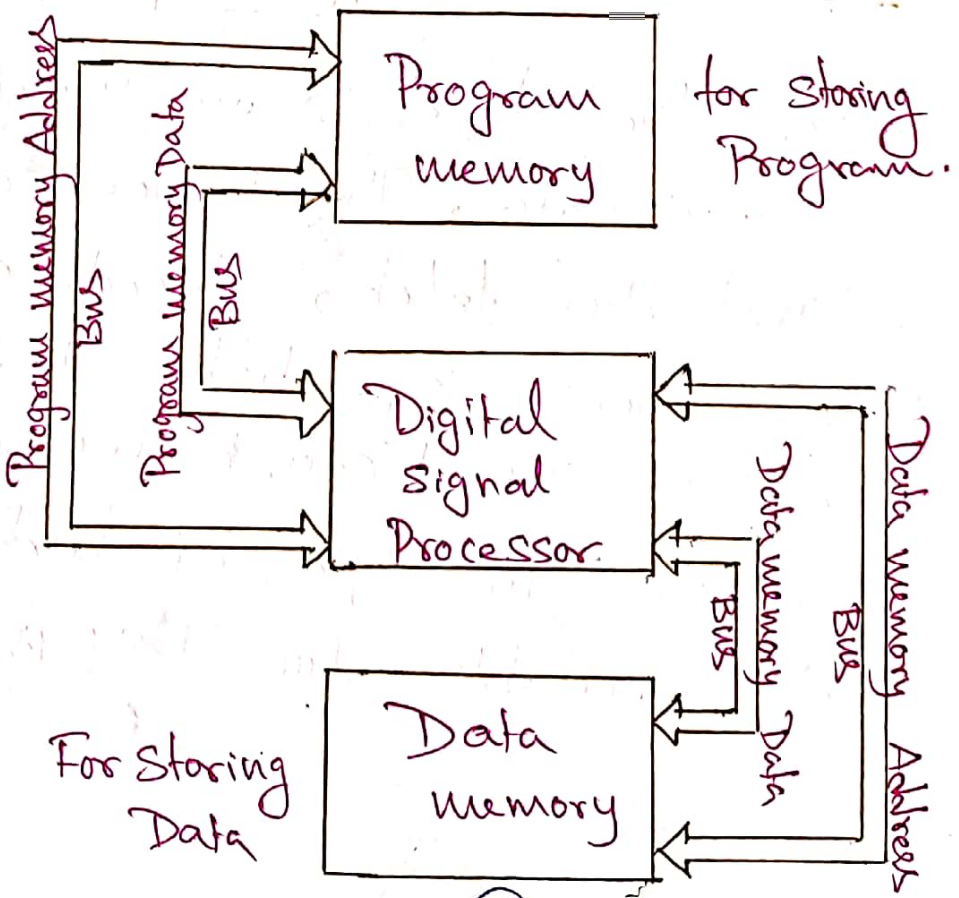
All the General Purpose Processor normally have this type of Architecture as shown in figure.



The architecture Shares Same memory for Program and data. The processors Perform instruction fetch, decode and execute operations Sequentially. In such Architecture, the Speed can be increased by pipelining. This type of architecture not much Suitable for DSP Processor

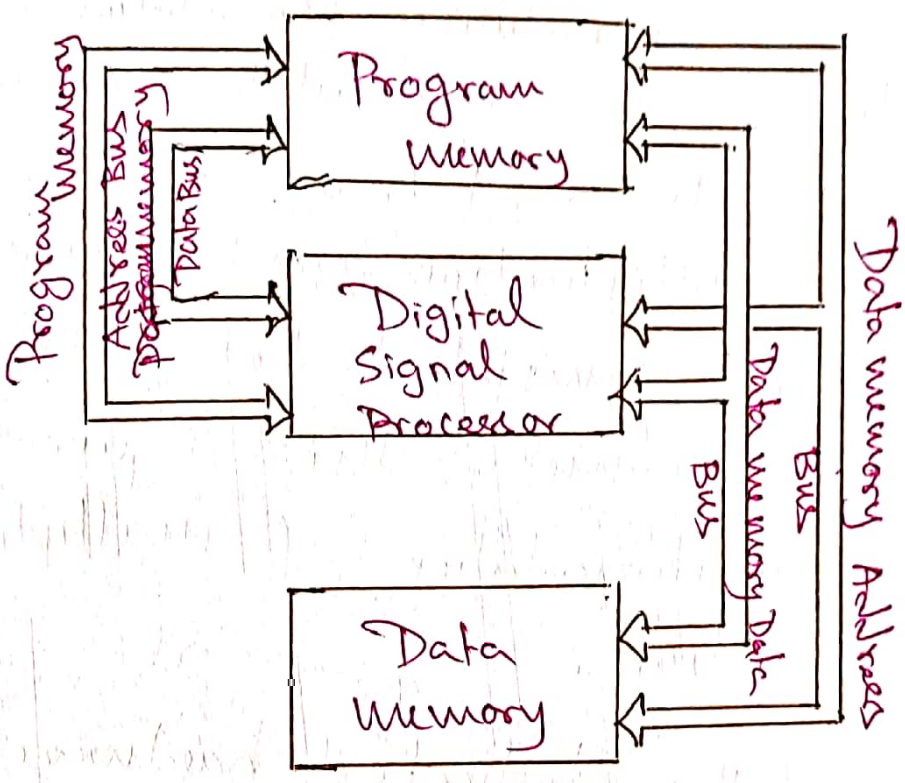
Harvard Architecture

The Harvard architecture has separate memories for Program and data. There are also separate, address and data buses for Program and data. Because of these separate on chip memories and internal buses, the speed of execution in Harvard architecture is high.



Modified Harvard Architecture

The block diagram shows the Modified Harvard Architecture.



The Data memory Address and Data buses are used to transfer the data from Program memory to data memory and Vice-Versa.

Normally the Program memory and data memory address are generated by Separate address generators.

Hardware Multiplier - Accumulator (MAC) Unit :

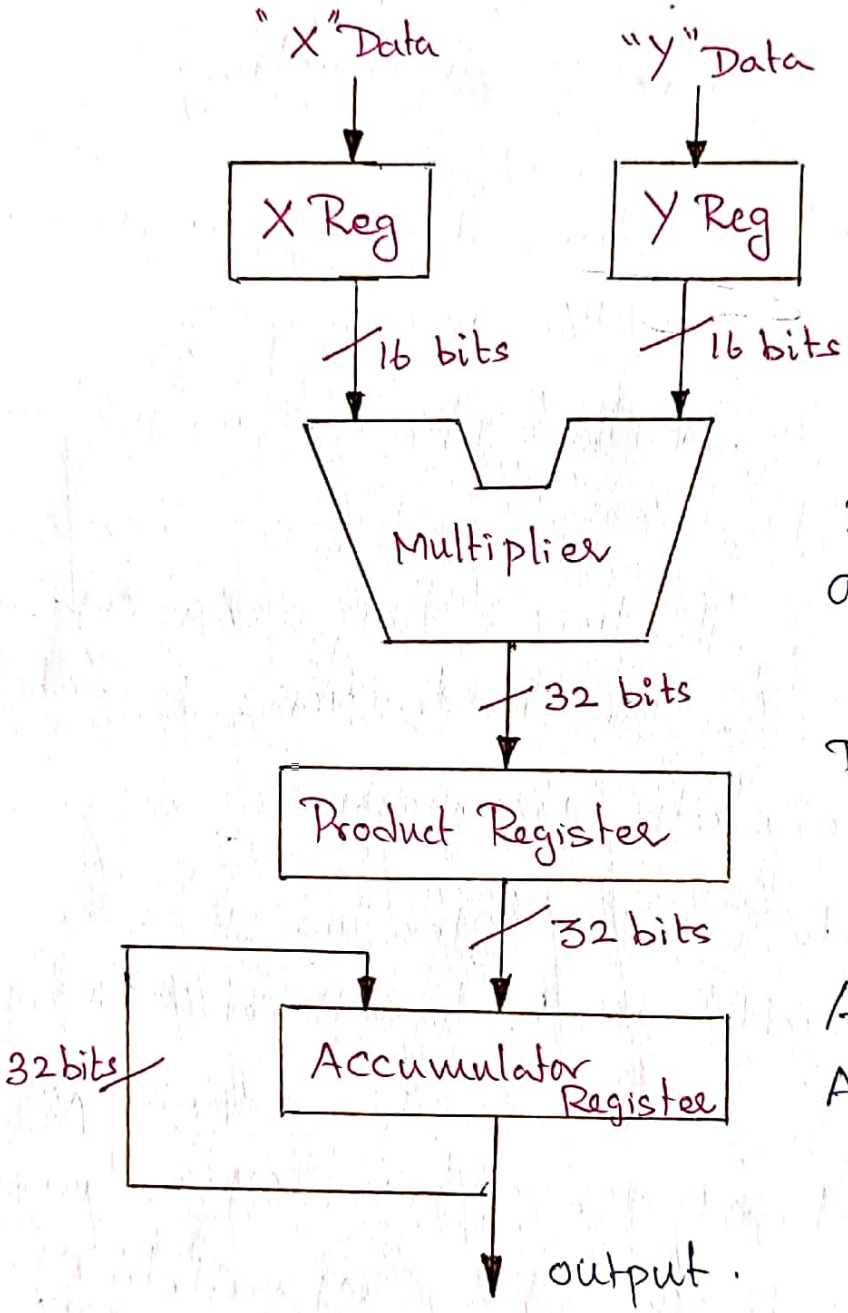
Most of the operations in DSP involve array multiplication. The operations such as Convolution, Correlation require multiply and accumulate operations.

In real time applications, the array multiplication and accumulation must be completed before next sample of input comes. This requires very fast implementation of multiplication and accumulation.

The dedicated hardware unit called MAC is used. It is called Multiplier-Accumulator (MAC).

The output of multiplier is stored into the product register. This product register contents are added to accu-mulator register Acc in central ALU

The Block diagram of typical MAC unit shown below.



MAC accepts two 16 bit data and Computes 32 bit Product in a Single Cycle.

Product Register Store the result.

Accumulator Register Accumulate the result.

The DSP Processors have a Special Instruction called MACD. This means multiply accumulate with data Shift.

Pipelining :-

Any instruction cycle can be split in following micro instructions:

1) Fetch: In this phase, an instruction is fetched from the memory

2) Decode: In this phase, an instruction is decoded

3) Read: In this phase, An operand required for the instruction is fetched from the data memory

4) Execute: The operation is executed and result are stored at appropriate place.

Each of the above operations are independent can be separately executed in different functional unit to speed up the processing is called Pipelining

Instruction execution without pipeline cycle.

Value of Clock T	Fetch	Decode	Read	Execute
1	I_1			
2		I_1		
3			I_1	
4				I_1
5	I_2			
6		I_2		
7			I_2	T
8				I_2

Note :- 8 T cycles required to complete 2 instructions.

Instruction execution with pipeline.

Clock T cycles	Fetch	Decode	Read	Execute.
1	I_1			
2	I_2	I_1		
3	I_3	I_2	I_1	
4	I_4	I_3	I_2	I_1
5	I_5	I_4	I_3	I_2
6		I_5	I_4	I_3
7			I_5	I_4
8				I_5

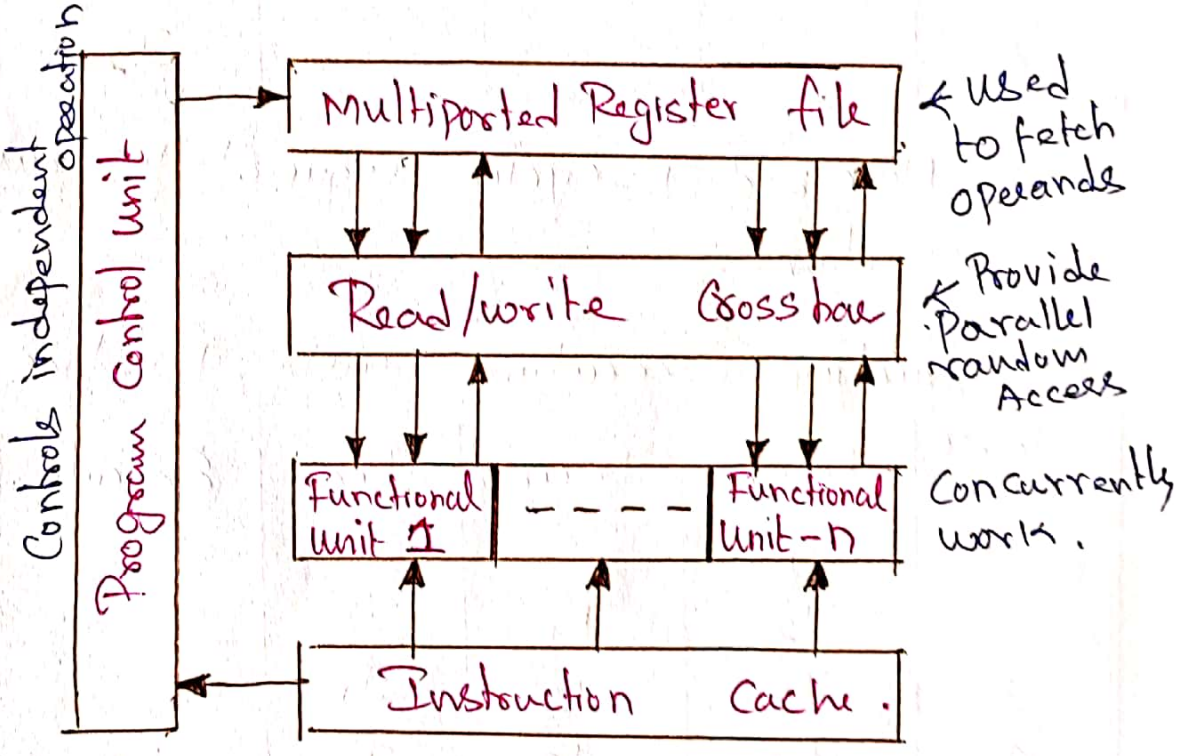
Note :- 5 Instructions are completed in 8T cycles.

VLIW Architecture and Multiple ALU's

DSP Processor use Very Long Instruction Word (VLIW) Architecture.

VLIW Architecture consists of multiple number of ALUs, MAC units, Shifters etc..

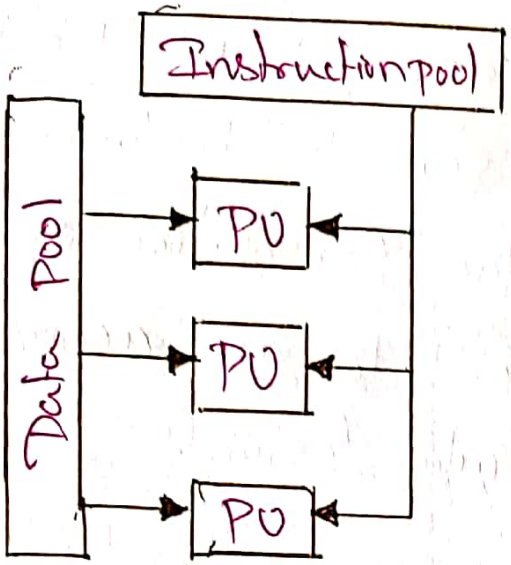
The block diagram of VLIW architecture shown below.



Normally 8-functional units are preferred. This number is limited by Hardware cost of the multiported register file and Crossbar switch.

Single Instruction Multiple Data (SIMD):-

The SIMD is the technique used to data level parallelism. The data is distributed across different parallel computing units. These computing units are called processing units. (PU)



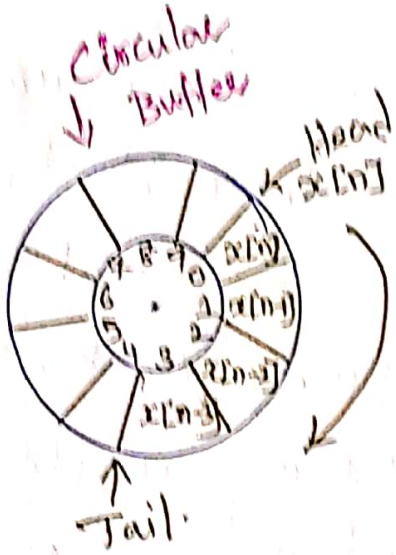
Advantages :-

* Single Instruction very efficiently Complete data process.
* Many Data in Single Instruction

Disadvantages :-

* All algorithms Cannot take Advantage of SIMD.
* Most of the Compilers do not generate SIMD instruction.
* Number representation mismatch may occur.

Hardware Circular Buffer %



Hardware Circular Buffer are required in DSP Processor to Perform the Circular Convolution operation

The Data are access in a Circular fashion.

Addressing in Sequence of

x(0) x(1) x(2) x(3) x(0) x(1) ....

Addressing Modes %

The addressing mode is the method of Specifying the data to be operated by an instruction.

The following Seven addressing modes %

* Direct addressing mode
* Memory mapped register addressing
* Indirect addressing mode
* Immediate addressing mode
* Dedicated register addressing
* Circular addressing
* Bit Reversed Addressing mode

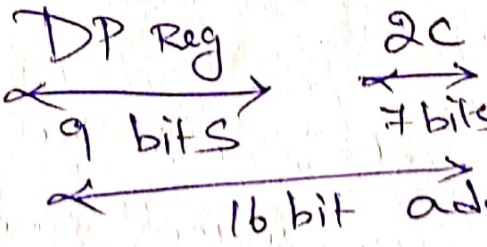
Direct Addressing Mode :-

In direct Addressing, the lower 7 bits of data memory address are Specified directly in the instruction itself. The upper 9 bits of the address will be the Content of data Memory Page Pointe (DP)

Example :-

ADDC 2Ch

address of data is generated by appending 9 bits of address for DP register and 2c 7 bits



Memory Mapped Register Addressing Mode :-

In Memory mapped register addressing the address of the memory mapped register can be Specified as direct address in the instruction.

Example :-

LAMM 16h ; Load accumulator with Content of memory mapped register mapped to address 16h.

Indirect Addressing Mode :-

In indirect addressing mode, the data memory address is Specified by the Content of one of the eight auxiliary registers AR0 - AR7

AR can be updated automatically either after or before the operand is fetched.

Example :

LACC *,0 ; Load the Content of data memory addressed by AR to accumulator without left shift.

Immediate Addressing Mode :-

In Immediate addressing, the data is specified as a part of the instruction itself.

Example :-

ADD #4Ah. 4A is data followed immediatly after opcode.

Dedicated Register Addressing Mode :-

In dedicated Register addressing mode, the address of one of the operands is specified by a dedicated CPU register BMAR (Block Move Address Register).

In this addressing mode a block of data can be moved to another location

Example :-

BLDD BMAR 6Fh; the Source address is Content of BMAR. Destination address is DP, Reg and 6F

Circular Addressing :-

The Circular addressing is Similar to indirect addressing. This addressing mode allows the Specified memory buffer to be accessed sequentially with a pointer which wraps around like circle.

The Four circular buffer registers namely,

CBSR1: } Circular Buffer Start address Register
CBSR2: }

CBER1: } Circular Buffer End address Register.
CBER2: }

Bit Reversed Addressing Mode :-

In bit reversed addressing, the address is incremented / decremented in the bit reversal order. and address is specified by AR Register.

Example :-

MAC 14F0h, *BR0+

Content Program memory is multiplied by data memory and added to the Acc Register

DSP Architecture

Types of DSP Architecture

* Floating Point Arithmetic Processor
  TMS 320C3X
* Fixed Point Arithmetic Processor
  TMS 320C 5X
  TMS 320C 54X.

Fixed Point DSP Architecture :-

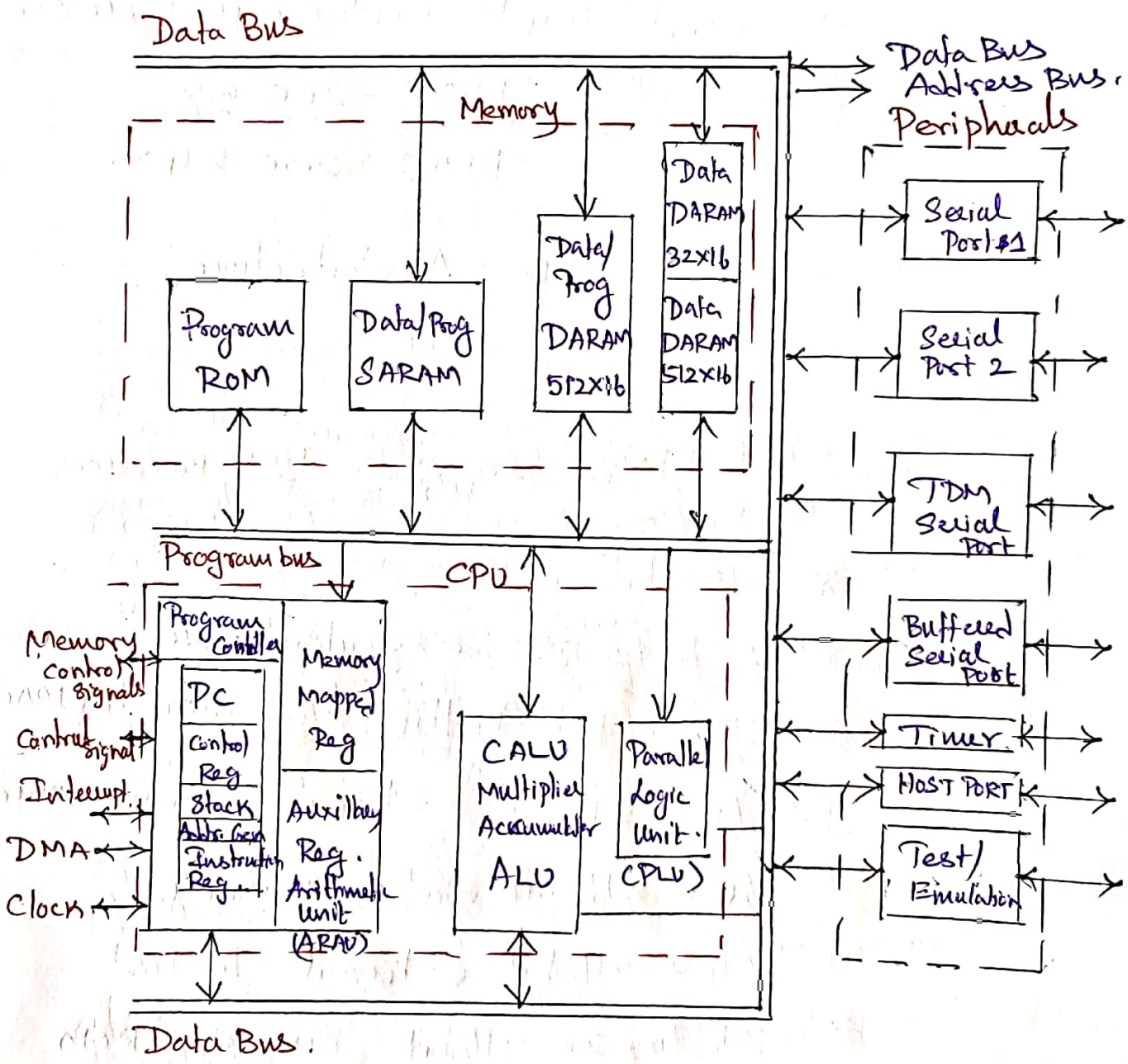
TMS 320 C 5X :-

Features of TMS 320C 5X DSP Processors

* 16-bit CPU
* 20 to 50 ns execution time
* 16 x 16 bit Multiply / Accumulate (MAC)
* 64 K x 16 bit external Program memory
* 64 K x 16 bit data memory
* 64 K x 16 bit external I/O Port
* 2K to 32K x 16 bit on-chip PROM
* 4 to to 9K x 16 bit onchip RAM.

* Synchronous, TDM and buffered Serial Ports
* Inbiilt PLL (phase lock loop)
* JTAG Ports

Block diagram of TMS 320 C5X Processor.



The TMS 3205x Processors have an advanced version of Harvard architecture with Separate buses for Program and data, which facilitate Simultaneous access of Program and data.

The Architecture of TMS 3205x Processor can be broadly divided into three major areas. They are.

* CPU (Central Processing Unit)
* Memory
* Peripherals.

CPU [Central Processing Unit] functional Units are :-

* PLU (Parallel Logic Unit)
* Central Arithmetic logic unit (CALU)
* Memory mapped Register (MMR).

Parallel Logic unit (PLU).

The PLU is an additional logic unit, that Perform logical operations without affecting accumulator (or) Product Register.

Logical operations like set, clear, ~~AND~~, test toggle etc.,

Central Arithmetic Logic Unit (CALU)

The Central Arithmetic logic unit Contains 32 bit ALU, 32 bit Accumulator, 32 bit Accumulator buffer, 16x16 MAC, 32 bit Product Register, 16 bit left and Right barrel Shifter.

Memory-Mapped Registers

The TMS320C5x has 96 nos of 16-bit Memory Mapped Registers and they are mapped into Page-0 of data memory Space.

The memory mapped registers includes various Control and status register for CPU, Serial port, and timers

Auxiliary Register Arithmetic Unit (ARAU)

The ARAU Contains eight 16-bit Auxiliary registers (AR0-AR7), a 3-bit Auxiliary Register Pointer (ARP), a 16bit index Register (INDX).

The auxiliary ~~resig~~ registers Can also be Used as general-purpose registers.

Program Controller :-

The Program Controller Contains logic circuits that decodes the instructions, they manages the CPU, Pipline, Stores the Status of CPU operations and decodes the Conditional operations.

The Program Counter (PC) used to hold the next instruction address.

On-chip Memory in TMS320C5x.

Program ROM :-

It is used to store the Program Code for a Specific application during manufacturing of chip itself. It is 2K to 32K word size.

Data/ Program Dual Access RAM (DARAM).

It Contains 3 blocks

The Block B_0 – 512 x 16 bit – Data/ Program

B_1 – 512 x 16 bit – Data

B_2 – 32 x 16 bits – Data.

Data/Program Single-Access RAM (SARAM):-

The Variables models of TMS320C5x has 1K words to 9K words of SARAM.

On-chip Peripherals of TMS320C5x.

The various on-chip Peripherals of TMS320C5x Processors are

* Clock Generator.
* Hardware Timer
* Software Programmable wait state Generator
* Parallel I/O Ports
* Host Port
* Serial Ports.

Clock Generator :-

The Clock Generator of the TMS320C5x Processor consists of an internal oscillator and Phase Locked Loop (PLL) circuit. It is used to generate Clock Pluses for CPU. It defines the Speed of CPU.

Hardware Timer :-

A 16 bit hardware timer with a 4-bit Prescaler is available in TMS320C5x This Programmable timer generates Clock at a rate that is between 1/2 and 1/3 of the machine Cycle rate.

It Contains three Registers to Control and operate the timer.

* Timer Control Register (TCR)
* Time Counter Register (TIM)
* Time Period Register (PRD).

Software Programmable Wait state Generators:-

The TMS320C5x Processor has Software-Programmable wait-state generators, which can insert/generate wait-states in external bus cycles for interfacing with slow/speed external memory (or) I/o devices.

Parallel I/O Ports :-

The TMS320C5x has 64K I/O address space. It is addressed by the IN (or) OUT instructions.

HOST Port interface (HPI) :-

The HPI is an 8-bit Parallel I/O Port used to interface with host processor.

Serial ports :-

Three different kinds of Serial Ports are available in TMS320C5x They are

* Time-Division Multiplexed (TDM) Serial
* Buffered Serial Port (BSP)
* General Serial port.

TDM serial Port can be used for Serial Communication between multiple processors.

BSP serial port Contains a full-duplex double-buffered serial Port Interface (SPI) and Auto-Buffering Unit (ABU).

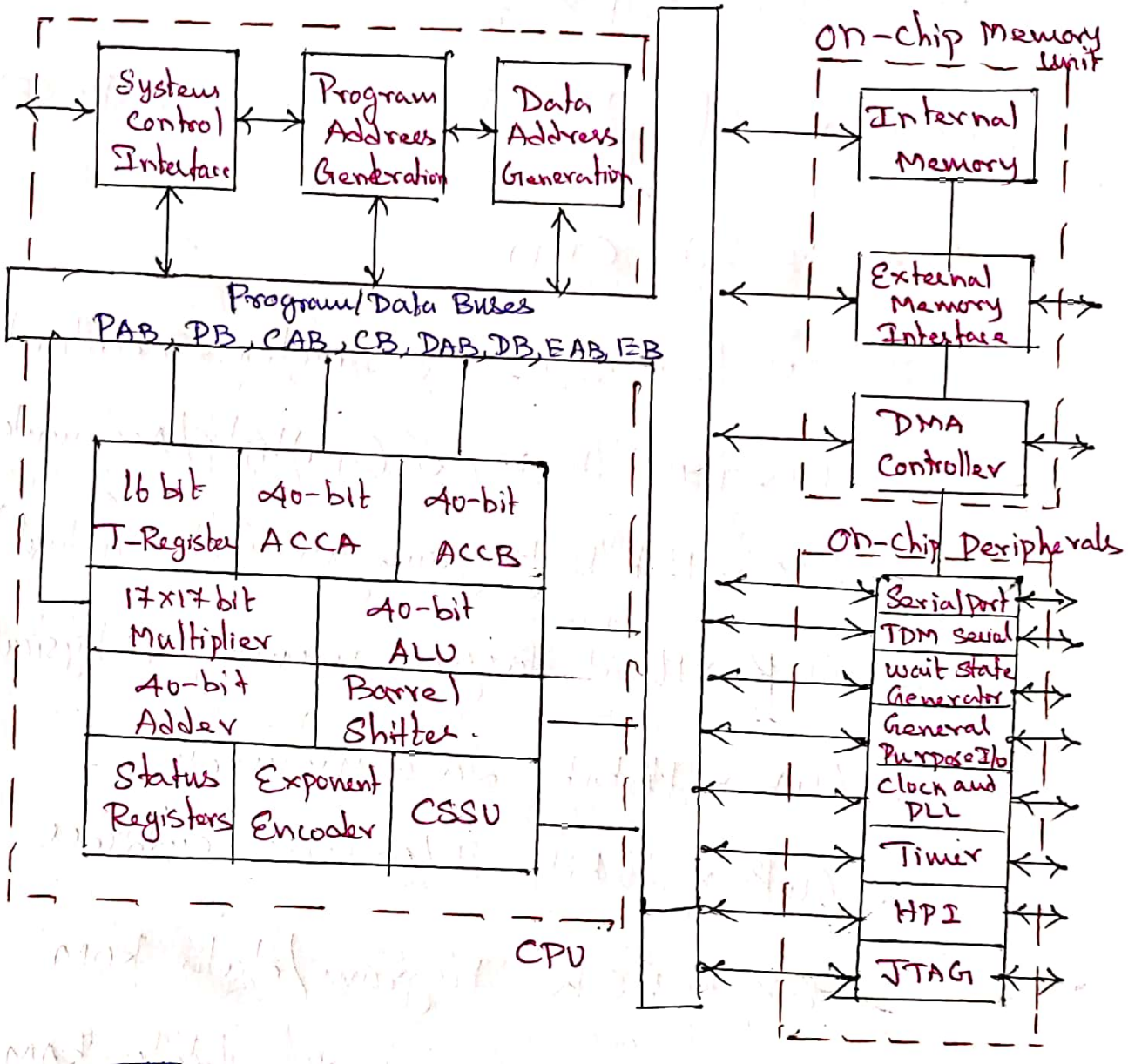
Fixed Point DSP Architecture :-

TMS 320c54x :-

Features of TMS 320C54x DSP Processors

* 16 bit CPU
* 25 to 8.3 ns instruction execution time.
* 17x17 bit MAC (Multiply/Accumulate)
* 8M x 16 bit Program memory (virtual)
* 64K x16 bit Program memory (physical)
* 64K x 16bit data memory
* 64K x 16 bit I/O Ports address
* 2K to 48K Program/data ROM
* 5K to 32K x16bit Program/data RAM
* JTAG Ports
* Programmable timer and PLL
* Synchronous TDM and buffered serial ports.

Block diagram of TMS320C54x



The TMS320C54x processors employ an advanced, modified Harvard architecture that maximizes processing power by providing four pairs of separate bus structure, three pairs for data memory and one pair for program memory.

The 4 Pairs (or) 8-internal buses of TMS320C54x Processors are,

PB : Program Bus
PAB : Program Address Bus
} used for Program access

CB : C Bus
CAB : C Address Bus
DB : D Bus
DAB : D Address Bus
} used for Data access

EB : E Bus
EAB : E Address Bus.
} Data memory

The architecture can be broadly divided into three major areas.

They are

* CPU [Central Processing Unit]
* on-chip memory unit
* on-chip Peripherals.

CPU Functional units

The functional units of CPU Consists of.

* 40 bit ALU [Arithmetic Logic Unit]
* 40 bit Accumulators [Acc A and Acc B]
* Barrel Shifter
* 17 x 17 multiplier
* 40 bit adder
* CSSU (Compare, Select and Store unit)
* exponent encoder
* Status register.
* Data address generator
* Program address generator.

Arithmetic Logic unit [ALU]

The 40 bit ALU can perform a wide range of arithmetic and logical operations.

The ALU can function as two 16-bit ALU's and perform two 16-bit operations simultaneously when ST1 is set.

Accumulators :-

The CPU has two 40-bit accumulators referred to as accumulator ACCA and accumulator ACCB.

The accumulators are divided into three parts :

* Guard bits (bits 32-39)
* A high-order word (bits 16-31)
* A low-order word (bits 0-15).

The Guard bits are used as headmargin for computation to prevent overflow.

Barrel Shifter :-

The 40-bit barrel shifter can perform 0 to 31 bits left shift, 0 to 16 bits right shift

It is used to realize the following operations.

* Prescaling
* Logical Shifting

* Normalizing the Accumulator
* Post scaling

It can handle 16 bit / 32-bit / 40 bit operands.

Multiplier / Adder

The multiplier / adder unit consists of 17 x 17 bit multiplier, 40-bit adder, zero detector, rounder etc.,

It is used to perform MAC operation. and an ALU operation in parallel.

Compare, Select and Store Unit (CSSU):-

The CSSU is an dedicated unit to perform add / compare / select operations in order to support butterfly algorithms,

Data / Program address Generator:-

ARAU0, ARAU1, AR0-AR7, are used to generate Data address

9 bit Data Page Pointer is used as upper 9 bits of Data memory address is used as lower to generate address

The Program Counter (PC) is a 16-bit register which holds the next instruction Address.

On-Chip Memory

The Three different types of on-chip memory are.

* Mask-Programmable ROM
* Single Access RAM [SARAM]
* Dual Access RAM [DARAM]

Mask-on-Chip ROM :-

2K to 48K words size maskable ROM It is used to store the Program Code and data for a specific application during manufacturing.

On-Chip DARAM :-

5K to 10K word on-chip DARAM accessed twice per machine cycle.

On-chip SARAM :-

24K words on chip SARAM are organized as three blocks of 8K words used to store data.

On-chip peripherals :-

The various on-chip Peripherals are

* Wait-state generator
* Parallel I/O Ports
* DMA Controller
* Host Port Interface
* Serial Ports
* Timer.
* PLL
* General Purpose I/O

Wait-state Generator :-

It is used to insert/generate wait states in external bus cycles for interfacing with slow/speed external memory (or) I/O devices.

Parallel I/O Ports :-

It has 64 K I/O address Space. It is addressed by using IN (or) OUT Instructions.

Host Port interfacing (HPI) :-

HPI is an 8 bit parallel I/O Port used to interface with host Processor.

Serial ports :-

The following four types of Serial Ports

* Synchronous Serial port
* TDM Serial Port.
* Buffered Serial Port
* Multichannel Buffere Serial Port.

DMA Controller :-

It is used to access memory Directly between internal and External memory.

PLL :-

Phase Lock loop used to generator Internal clock frequency

Timer :-

It has 16-bit timer with a 4-bit Prescaler. It is used for Controlling timing operations.

General-purpose I/o Pins :-

It has two general purpose I/o pins. They are

$\overline{BIO}$ – Branch control input Pin.

XF – External flag

XF Pin can be used to Signal External devices.

$\overline{BIO}$ pin can be used as alternative interrupt.

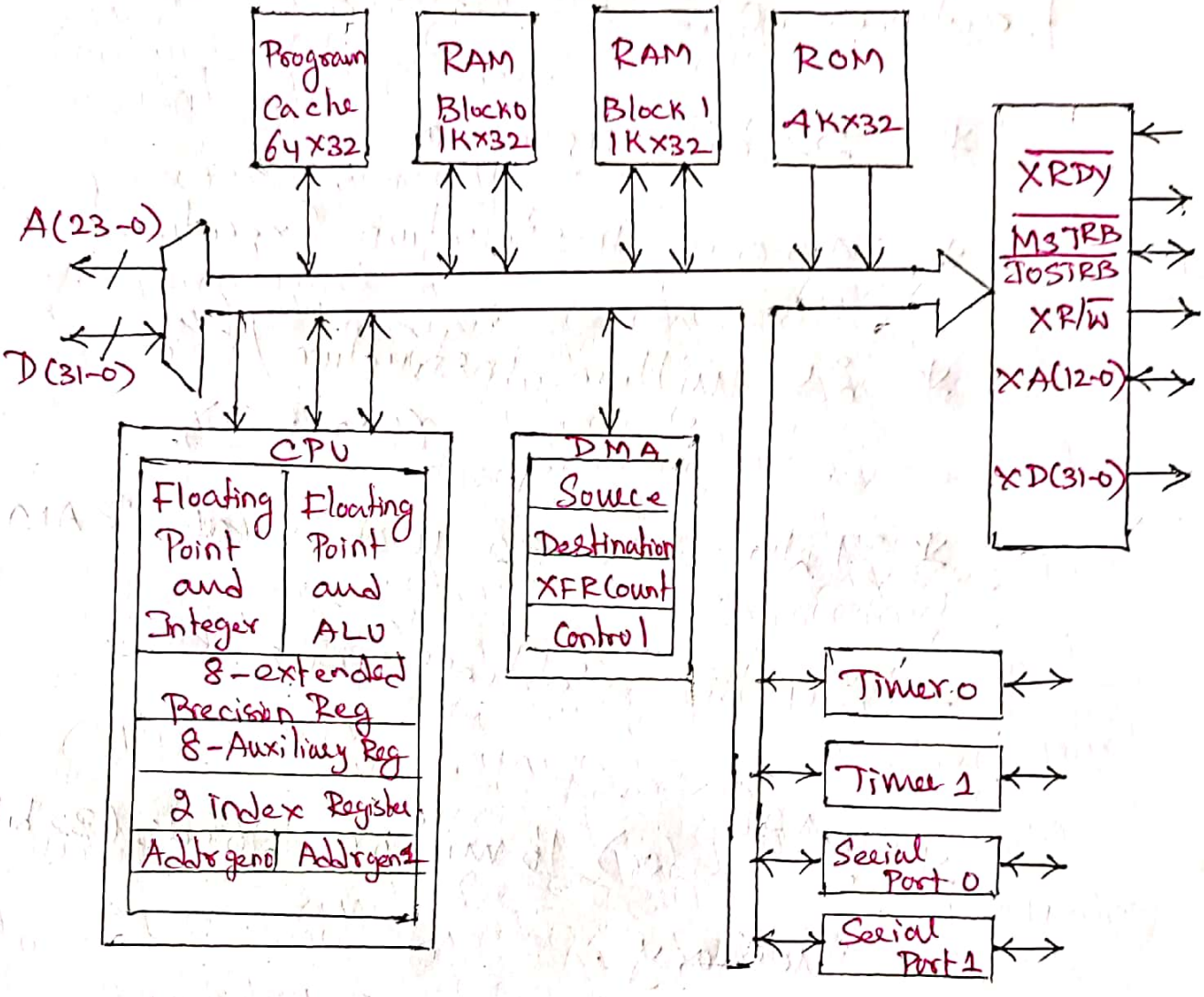
Floating Point DSP Processor Architecture :-

TMS 320C 3X :-

Features of TMS 320C 3X DSP Processors

* 32 bit Processor
* 40ns Instruction Execution time
* 25 million instruction / sec (MIPS)
* Two 1K x32 words of RAM
* 24 bit address bus.
* 2^{24} (or) 16 million words (32 bit) memory.
* One Serial Port.
* Third Gen floating point Processor.
* 50 million fp ops/sec (MFLOPS)

Block diagram of TMS320C3x



The ~~to~~ floating Point DSP Processors key features are high Precision floating Point arithmetic.

The floating Point arithmetic minimizes co-efficient quantization errors, overflow and round off errors.

The data width is 32-bits. It uses 40-bit and 80-bit accumulators. It Contains on chip memorys.

on-chip memorys :-

* Program cache 64K x 32 bits
* RAM Block 0 1K x 32 bits
* RAM Block 1 1K x 32 bits
* ROM 4K x 32 bits

CPU Block Conbaing floating Point multiplier and Adder, floating Point ALU. and 8-extended Precision Register, 8-AR [AR0-AR7] Auxiliary registers, 2-index register.

It Contain two Address generator. and 12 (control) registers.

DMA is used to access memory memory and I/O Ports directly.

Peripherals :-

It Consist of two timers of 32-bits Timer0 and Timer 1. and Two 16 bit Serial port - Serial port 0 and Serial port 1.

Applications :-

* Digital audio Processing
* Data Communication.
* Industrial Automation.
* Digital Image Processing.
* Compression

write an assembly language Program to add two numbers

```
.ps    0b00h        // origin of Program 0b00 addr
.entry              //Initialize Program Counter
CLRC SXM            //clear sign extension.
LAR  AR1, #1101h    //AR1 = 1101 (Address Data1)
LAR  AR2, #1102h.   // AR2 = 1102 (Address) Data2
LACC *+, 0, AR1 .   //Load ACC
LAR  AR3, #1103h.   // AR3 = 1103 (Address Result)
```

```
ADD *+, 16, AR2.   // Add Acc and AR2
SACL *1, 0, AR3    // Stor Result in AR3
RET                Return
.end.              end.
```

Replace ADD to SUB, MUL, DIV for Subtraction, Multiplication, Division Program.