6/8/19

Introduction To programmable DSP

Architecture features of DSP processors:

1) DSP processor should have multiple Register so that data can exchanged from Rigister to Registe fast. 2) DSP operations required multiple operands simultaneously Hence DSP processor should have multiple operand fetching capacitor.
3) DSP should have buffers to support the shifting operations.
4) The DSP processor should able to perform multiple and Accumulate operations very fast.
5) DSP processor should have multiple painters to support multiple operands, jumps and shift operation
6) DSP processor can be used with generator processor they should have multi processing capacity
7) To support DSP operations DSP processor should have on chip memory
8) for real time Applications interrupts and timers are Required. Hence DSP process should have powerfull interrupts and timers.

7/8/19

MAC (Multiply and Accumulator):

most of the operations in DSP involve array multiplication the operations such as convolution, correlation require mutiply and accumulate operations.

In real time applications the array multiplication and accumulation must be completed before next

sample of i/p comes. this requires very fast implementation of multiplication and accumulation.

For this purpose the dedicated H/W unit MAC is used is called multiplier and accumulator.

The complete MAC operation is executed in one clock cycle.

In the DSP processor the o/p of the multiplier is store into the product register. this product register contains are added to the Accumulator Register the central ALU.

The DSP process have the special instruction called MACD. ie; multiply and Accumulate with data shift.

MAM (Multiple Acesses Memory)

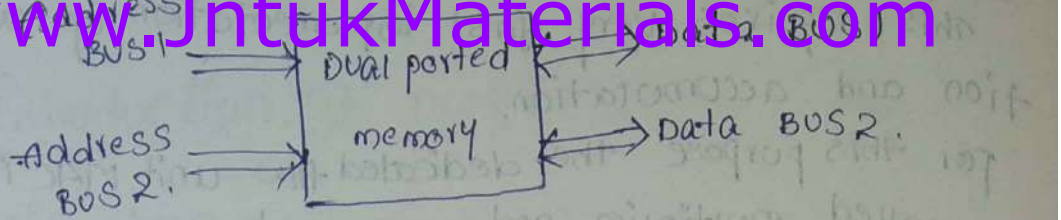
The MAM allows more than one memory Acesses in a single clock cycle. the Dual Acesses RAM. (DA-RAM) allows 2-memory Acesses in a single clock cycle.

The Dual Acesses RAM is connected to the DSP processes with 2 Address with 2 data buses independently. this gives 4 memory Acesses in a single clock period.

MPM (Multi ported Memory)

The MPM has the facility of interfacing multiple address and data buses. the following Fig shows the Dual ported memory. this memory has 2 Address buses and 2 data buses saperatly Interfaced.

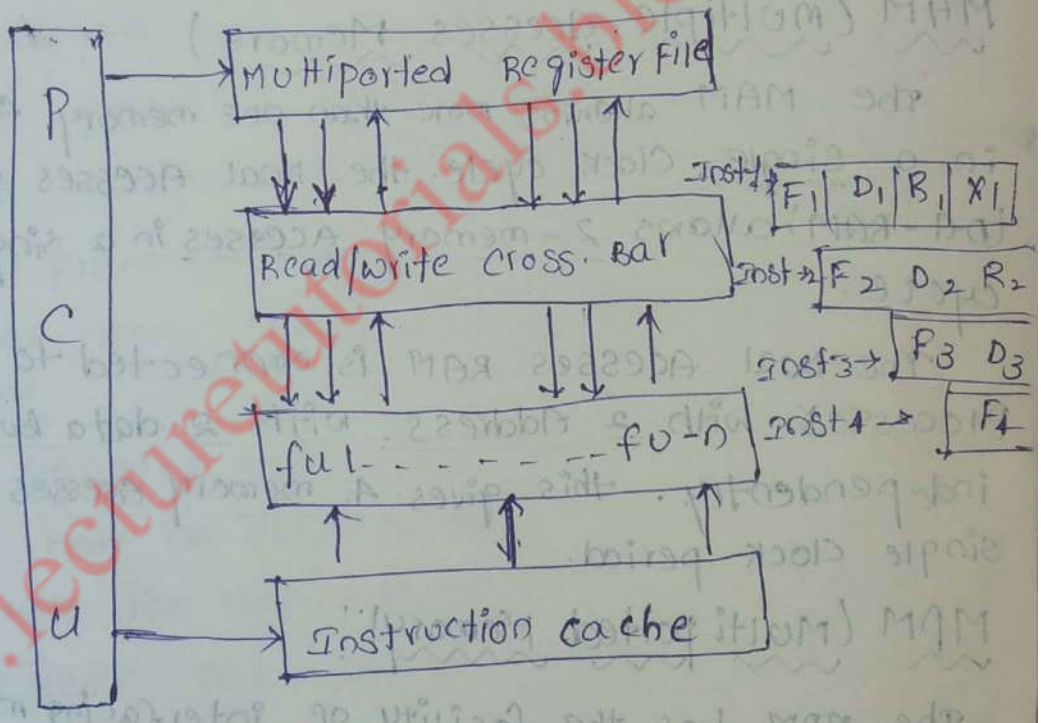
The Dual ported memory can allows 2 memory Acesses in a single clock period.



With the help of dual port memory the program and data can be store in a single memory chip and they can be Acessed simultaneously.

The multi ported memories having increased no of pins larger chip area which makes it more expensive and large in size.

VLIW (very long Instruction word):



PCU → program control unit.

fu → function unit.

9/8/19. Sum of the DSP process use ver VLIW architecture such architecture consists of multiple no of ALU's, MAP units and shifter Registers.

The following fig shows the block diag of VLIW architecture.

The above Architecture consist of multi ported register file used for getting the operands and storing the results.

The Read/write cross bar provides parallel Random Acesses by functional units to the multiported Register file.

The function units work concurently with the load (or) store operation of the data b/w RAM and Register file.

The program control unit provides the algorithm that executes independent ||el operations.

The performance of VLIW Architecture depends on degree of parallelism.

Normally 8-functional units are preferred this number is limed by H/W cost of the multiported Register file and Read/write cross bar.

Pipelining:

without pipelining

Value of 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	
8.				I_2

With pipelining:

Value of T	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

9/8/19

Any instruction cycle can be split into the following instruction.

1) fetch: In this phase an instruction is fetch from the memory.

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

3) Read: An operand required for this instruction is fetch from the data memory.

4) execute: The operation is executed and results are stored at appropriate place.

Each of the above instructions can be saperately executed in different functional units. to the following fig shows 1) How the instruction is executed without pipeline.

2) from this fig. we can observe that when the instruction I_1 is in fetch, the other units such as

decode, Read and Execute are idle. this means each functional unit is busy only for 25% of the total time.

from the fig (2) shows the instruction execution with pipeline,

Here we can observe that instruction-1 is in decode phase. next instruction I_2 is fetched.

Similarly, when I_2 goes to decode phase, next instruction I_3 is fetched.

∴ we observed that all the functional units are executed 4 successful instructions at any time.

compare above two figs we can observed that 5 instructions are executed. the same time pipelining misused.

Special Addressing modes:

The conventional microprocess have addressing modes such as direct, indirect, immediate Addressing modes. The DSP process have additional addressing modes because of fetch execution is fast.

1) **Short immediate Addressing mode:**

The operand is specified using a short constant this short constant becomes a part of a single word instruction.

In DSP process 8-bit operand can be specified as one of the operand in single word instructions such as: add, subtract, Div, AND, OR, NAND.. etc.

2) **Short direct Addressing mode:**

The lower order Address of the operand is specified in the single word instruction in DSP process the lower 7-bits of the Address are specified as a part of instruction.

Higher 9-bits of the Address are stored the data

Phase point:

3) Memory Map Addressing mode:

The cpu and I/O registers are accessed are memory location this registers are maped in the starting phage (or) ending phage of the memory phase

The Phage-0 corresponds to the starting page of the memory space

4) Indirect Addressing mode:

The addressing of the operands are stored in the Indirect Address registers.

In DSP process such registers are called Auxiliary registers.

Any of these auxiliary registers can be updated in the operands fetched by these registers are being executed.

The auxiliary registers are incremented (or) decremented Automatically by the value specified. in offset Registers. these offset registers called index registers

5) Bit-Reversal Addressing modes:

for the calculation of FFT the i/p data is required in bit-Reversal order. there is no need to re-shuffle the data in bit reversal sequence

The serially arranged the data in the memory given to the processor in bit reversal mode. with the help of Bit-reversal addressing mode

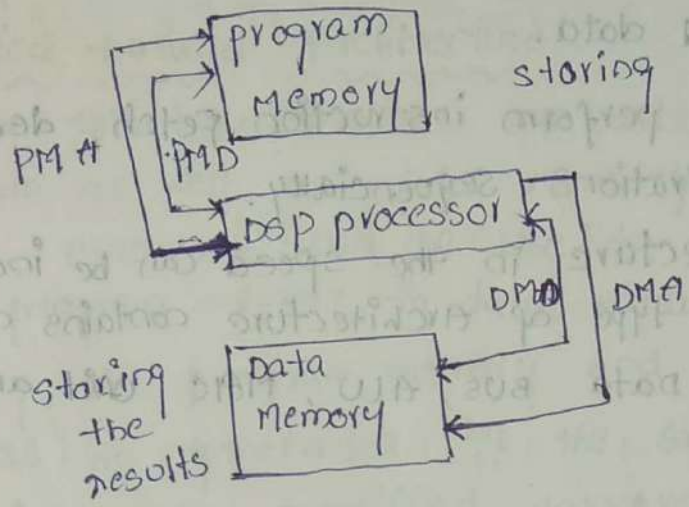
6) Circular Addressing mode:

In this mode the data stored in the memory can be read (or) write in circular fashion.

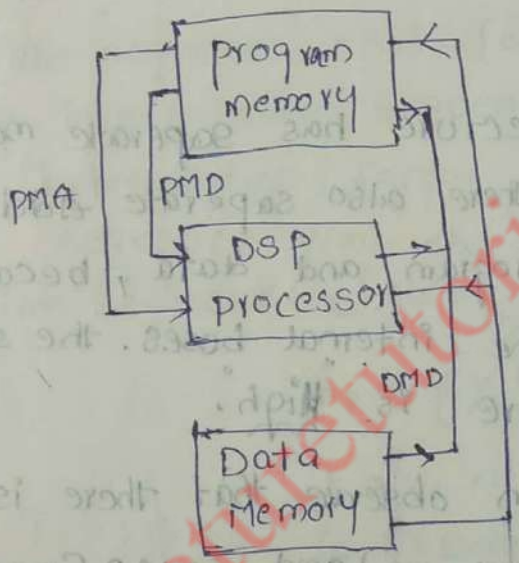
The memory is organized as a circular buffer the begining and ending addresses of the circular

buffer are continuously monitor.

12/3/19 Harvard Architecture:-



Modified Harvard Architecture:-



MACD instruction performs multiply and accumulate in the memory accesses are required.

1) Fetch MACD Instruction from program memory
2) fetch one of the operands. from program memory.
3) fetch second operand from data memory.
4) Data Memory write.

If this instruction is executed with von neumann Architecture. It requires 4 block cycles but Harvard and modified Harvard requires less no of clock cycles.

Von Neuman Architecture:

The General purpose process normally we have this type of Architecture the Architecture × same memory for program and data.

The process perform instruction fetch, decode, Read and execute operations sequencially.

such Architecture in the speed can be increased by pipelining this type of Architecture contains common Address and Data BUS, ALU, MAC Unit and I/O devices.

This type of Architecture is not suitable for DSP processes.

Harvard Architecture:

the Harvard Architecture has saperate memory for program and data there also saperate Address and data buses for program and data, because of these saperate on chip × memory the internal buses. the speed of execution Architecture is High.

from the fig we can observe that there is PMA BUS [program memory Address] and PMD [program memory Data] BUS saperate from program.

Similarly, there is saperate data memory Data (DMD) BUS and data memory Address [DMA] BUS.

The DSP includes various Registers, Address generators and ALU. . .

The PMD BUS is used to get instructions from the program memory and DMD is used to exchange operands and results from Data memory.

the Instruction code from program memory and operands from data memory can be search simultaneously is spanalle/ operation increases the speed.

Modified Harvard Architecture:

From this Architecture one set of bus to accesses program as well as data memory.

The DMD bus can be use to transfer the data from program memory to data memory. and vice-versa Normally, the program memory and data memory address on generated by the saperated address generators, this modified Harvard Architecture. used in several programmable DSP.

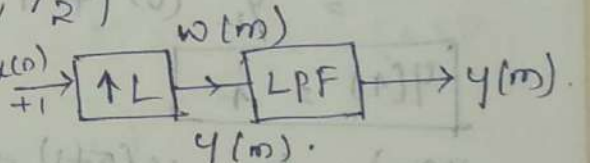
14/3/19 **

p) The sequence $x(n) = [0, 2, 4, 6, 8]$ is interpolated using interpolation sequence $b_K = [1/2, 1, 1/2]$ and the interpolation factor is 2 to find. the interpolated sequence $y(m)$. the s

Sol) Given sequence $x(n) = [0, 2, 4, 6, 8]$.

$b_K = [1/2, 1, 1/2]$

$L = 2, -1, 0$



(L-1) zero's.

$w(m) = [0, 0, 2, 0, 4, 0, 6, 0, 8]$.

$y(m) = \sum b_K w(m-K)$.

$y(0) = b_{K=0} w(m+1) + b_K w(m-0) + b_K w(m-1)$

$y(0) = b_{-1} w(0+1) + b_0 w(0-0) + b_1 w(0-1)$

$y(0) = 1/2 (0) + 1 (0) + 1/2 (0)$

$y(0) = 0$

$y(1) = b_K w(1+1) + b_K w(1-0) + b_K w(1-1)$

$y(1) = b_{-1} w(2) + b_0 w(1) + b_1 w(0)$

$y(1) = 1$

$y(2) = b_K w(2+1) + b_K w(2-0) + b_K w(2-1)$

$= b_{-1} w(3) + b_K w(2) + b_K w(1)$

$= \frac{1}{2}(0) + 1(2) + \frac{1}{2}(0)$

$y(2) = 2$

$y(3) = b_K w(3+1) + b_K w(3-0) + b_K w(3-1)$

$= b_K w(4) + b_0 w(3) + b_{+1} w(2)$

$= \frac{1}{2}(4)^2 + 1(0) + \frac{1}{2}(2)$

$y(3) = 2+1 \Rightarrow y(3) = 3$

$y(4) = b_K w(4+1) + b_K w(4-0) + b_K w(4-1)$

$= b_{-1} w(5) + b_0 w(4) + b_{+1} w(3)$

$= \frac{1}{2}(0) + 1(4) + \frac{1}{2}(0)$

$y(4) = 4$

$y(5) = b_K w(5+1) + b_K w(5-0) + b_K w(5-1)$

$= b_{-1} w(6) + b_0 w(5) + b_{+1} w(4)$

$= \frac{1}{2}(6)^3 + 1(0) + \frac{1}{2}(4)^2$

$y(5) = 3+2$

$y(5) = 5$

$y(6) = b_K w(6+1) + b_K w(6-0) + b_K w(6-1)$

$= b_{-1} w(7) + b_0 w(6) + b_{+1} w(5)$

$= \frac{1}{2}(0) + b_0(6) + \frac{1}{2}(0)$

$y(6) = 6$

$y(7) = b_K\, w(7+1) + b_K\, w(7-0) + b_K\, w(7-1)$

$= b_{-1}\, w(8) + b_0\, w(7) + b_{+1}\, w(6)$

$= \frac{1}{2}(8)^4 + 1(0) + \frac{1}{2}(8)^3$

$y(7) = 4 + 3$

$\boxed{y(7) = 7}$

$y(8) = b_K\, w(8+1) + b_K\, w(8-0) + b_K\, w(8-1)$

$= b_{-1}\, w(9) + b_0\, w(8) + b_{+1}\, w(7)$

$= \frac{1}{2}(0) + 1(8) + \frac{1}{2}(0)$

$\boxed{y(8) = 8}$

$y(n) = \{0, 1, 2, 3, 4, 5, 6, 7, 8\}$

2) The sequence $x(n) = [0, 3, 6, 9]$ is interpolated using. interpolation sequence $b_K = [1/3, 2/3, 1, 2/3, 1/3]$. and the interpolation factor is "3" find the Interpolated sequence $y(n)$.

sol) Given data.

$x(n) = [0, 3, 6, 9]$

$b_K = [1/3, 2/3, 1, 2/3, 1/3]$

$-2 \quad -1 \quad \uparrow_0 \quad +1 \quad +2.$

$L-1 = 0$

$3-1 \Rightarrow 2.$

$w(n) = \{0, 0, 0, 3, 0, 6, 0, 0, 9\}$

$y(n) = \sum b_K\, w(n-K)$

$y(n) = b_K\, w(n+2) + b_K\, w(n+1) + b_K\, w(n-0) + b_K\, w(n+1) + b_K(n-2).$

$y(0) = b_{-2}\, w(2) + b_{-1}\, w(1) + b_0\, w(0) + b_1\, w(-1) + b_2(-2)$

$y(0) = 0$

$y(1) = b_{-2}\, w(3) + b_{-1}\, w(2) + b_0\, w(1) + b_1\, w(0) + b_2\, w(-1)$

$= 1/3\,(3) + 2/3\,(0) + 1\,(0) + 2/3\,(0) + 1/3\,(0)$

$y(1) = 1$

$y(2) = b_{-2}\, w(4) + b_{-1}\, w(3) + b_0\, w(2) + b_1\, w(1) + b_2\, w(0)$

$= 1/3\,(0) + 2/3\,(3) + 1\,(0) + 2/3\,(0) + 1/3\,(0)$

$y(2) = 2$

$y(3) = b_{-2}\, w(5) + b_{-1}\, w(4) + b_0\, w(3) + b_1\, w(2) + b_2\, w(1)$

$= 1/3\,(0) + 2/3\,(0) + 1\,(3) + 2/3\,(0) + 1/3\,(0)$

$y(3) = 3$

$y(4) = b_{-2}\, w(6) + b_{-1}\, w(5) + b_0\, w(4) + b_1\, w(3) + b_2\, w(2)$

$= 1/3\,(6)^2 + 2/3\,(0) + 1\,(0) + 2/3\,(3) + 1/3\,(0)$

$= 2 + 2.$

$y(4) = 4$

$y(5) = b_{-2}\, w(7) + b_{-1}\, w(6) + b_0\, w(5) + b_1\, w(4) + b_2\, w(3)$

$= 1/3\,(0) + 2/3\,(3)^2 + 1\,(0) + 2/3\,(0) + 1/3\,(3)$

$= 4 + 1$

$y(5) = 5$

$y(6) = b_{-2} w(8) + b_{-1} w(7) + b_0 w(6) + b_1 w(5) + b_2 w(4)$

$= 1/3\,(0) + 2/3\,(0) + 1\,(6) + 2/3\,(0) + 1/3\,(0)$

$\boxed{y(6) = 6}$

$y(7) = b_{-2} w(9) + b_{-1} w(8) + b_0 w(7) + b_1 w(6) + b_2 w(5)$

$= 1/8\,(9)^3 + 2/3\,(0) + 1\,(0) + 2/3\,(6)^2 + 1/3\,(0)$

$y(7) = 3 + 4$

$\boxed{y(7) = 7}$

$y(8) = b_{-2} w(10) + b_{-1} w(9) + b_0 w(8) + b_1 w(7) + b_2 w(6)$

$= 1/3\,(0) + 2/3\,(9)^3 + 1\,(0) + 2/3\,(0) + 1/3\,(6)^2$

$y(8) = 6 + 2$

$\boxed{y(8) = 8}$

$y(9) = b_{-2} w(11) + b_{-1} w(10) + b_0 w(9) + b_1 w(8) + b_2 w(7)$

$= 1/3\,(0) + 2/3\,(0) + 1\,(9) + 2/3\,(0) + 1/3\,(0)$

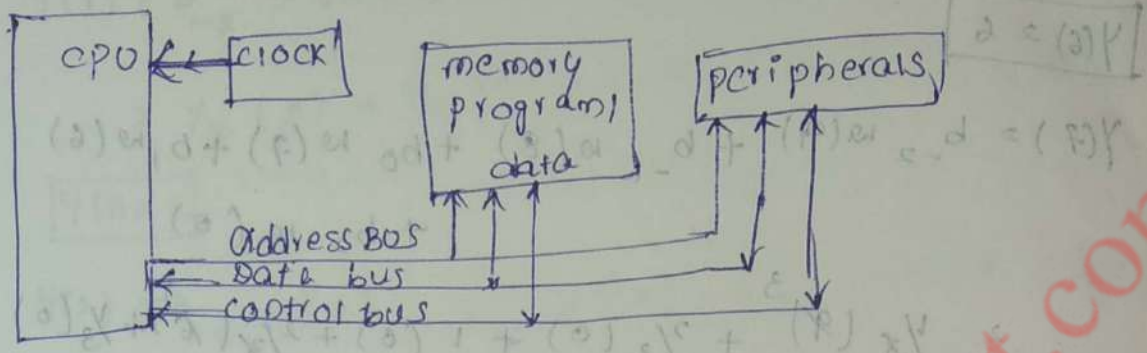
$\boxed{y(9) = 9}$

$y(n) = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$.

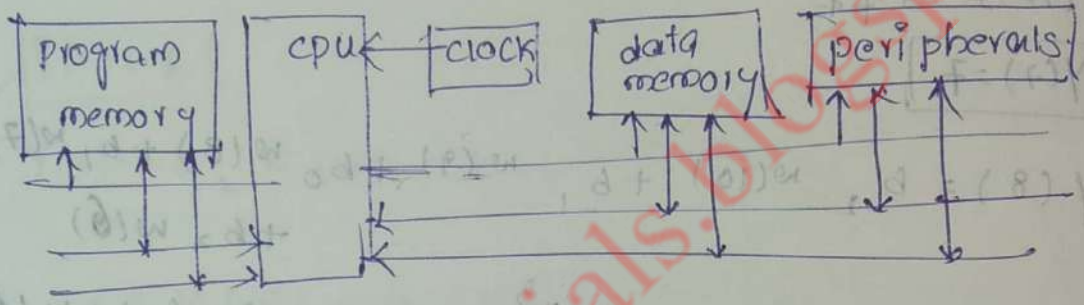
⟹ The DSP's are classified into

1) General purpose 2) Special purpose DSPs.

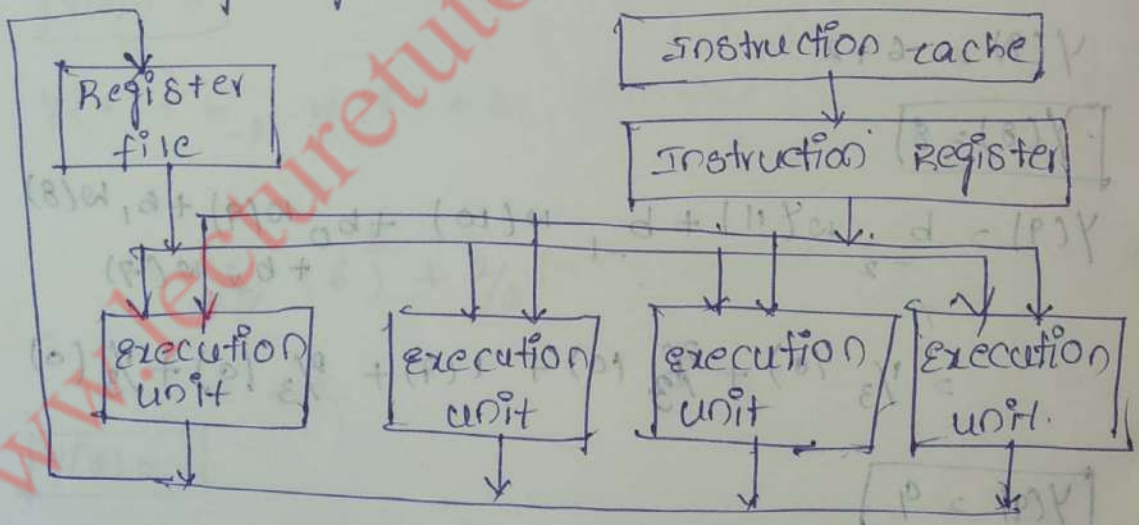
Von Neumann Architecture:



2) Harvard Architecture:



3) VLIW (very long Instruction word) Architecture:



Architecture of TMS320C50:

→ It is divided into 4 sub blocks.

1) Bus structure 2) CPU 3) Onchip memory 4) On chip peripherals.