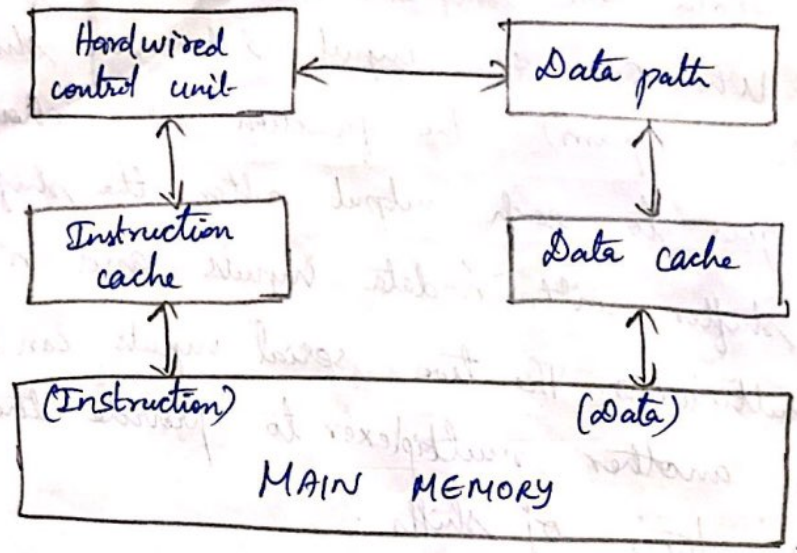
EXAMPLE ARCHITECTURES:

RISC – REDUCED INSTRUCTION SET COMPUTER

Reduced Instruction set Computer is a contrast processing unit to CISC (complex instruction set computer).

* RISC processors like Intel i860, SPARC, MIPS R3000, IBM RS/6000 have faster clock rates which range from 20 to 120 MHz.
* Most RISC processors use 32-bit instructions.
* They have limited (3 to 5) addressing modes
* Memory access cycle is broken into pipelined access operations
* A large register file and separate instruction and data caches are used.
* fixed length, easily decoded instruction format
* Single cycle instruction execution
* Hardwired rather than a microprogrammed control.

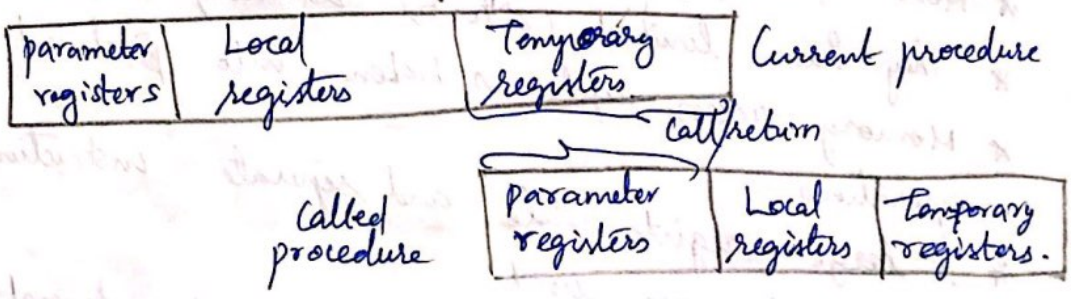


RISC Architecture with hardwired control & split instruction cache and data cache

RISC properties:

(1) Register – register operations :

* To implement register to register operations, RISC processors provide multiple set of registers. These registers are organized into overlapped windows and act as small, fast buffer for holding a subset of all variables that are most likely to be used.



Overlapping register window

2) One instruction per cycle:

* RISC processors executes one instruction per machine cycle.
* It is the time it takes to fetch 2 operands from registers, performs an ALU operation and stores result in a register.

3) Hardwired instructions:

* As RISC instructions are simple, they can be hardwired. They are executed faster than implemented with micro instructions as they do not require them and there to fetch data from registers

4) Reduced number of instructions:

* As RISC processor uses limited no-of instructions, it simplifies the design of control unit.

5) Simple Addressing modes:

* Almost all instructions use simple register addressing
* It does not support complex addressing modes.

6) Simple Instruction Format:

→ RISC processor provides simple instruction formats with fixed instruction length.
→ With fixed fields, opcode decoding and register operand addressing can occur simultaneously.

7) Instruction pipelining:

* RISC CPU contains several independent units that work in parallel.
* One of them fetches, other one decodes and other execute.

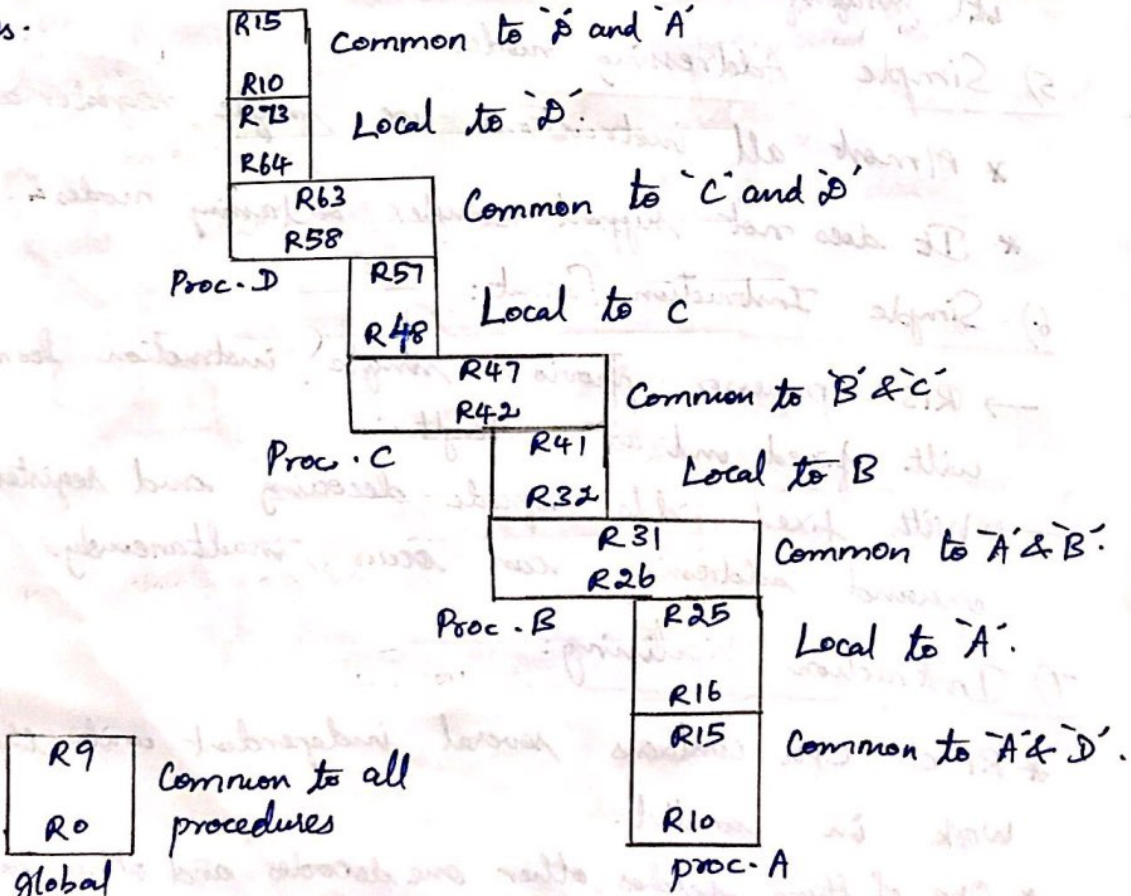
The instruction fetch phase fetches the instruction to be executed from memory and then execute phase performs ALU operation with register input and output to execute the instruction.

In case of load and store instructions, three phases are required

(1) Instruction fetch (I)
(2) Execute (E)
(3) Data Transfer (D)

(8) Overlapped Register windows:

A typical characteristic of some RISC processor is their use of overlapped register windows to provide the passing of parameters and avoiding restoring of register values.



Overlapped register windows.

* The system has a total of 74 registers.
* R0 through R9 are global registers that hold parameters shared by all procedures.
* Other 64 registers are divided into 4 windows to accomodate procedures A, B, C and D. Each register window has 10 local registers and two sets of six registers common to adjacent windows.
* Local registers are used for local variables.
* Common registers are used for exchange of parameters and results between adjacent procedures.
* Each procedure call activates a new register window by incrementing the pointer.

For example, if procedure 'A' calls procedure 'B'. Registers R26 through R31 are common to both procedures. So procedure 'A' stores the parameters for procedure 'B' in these registers. Procedure 'B' uses local registers R32-R41 for local variable storage. If procedure 'B' calls procedure 'C', it will pass the parameters through register R42 through R47.

In general organization of register windows will have the following relationships.

No. of Global registers = G
No. of local registers in each window = L.
No. of registers common to two windows = C.
No. of Windows = W

The no. of registers available for each window is calculated as follows:

Window size = L + 2C + G.

The total no. of registers needed in the processor is,

register file = (L + C)W + G.

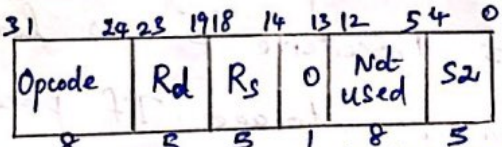
In the example given G = 10; L = 10; C = 6 & W = 4.

The window size is 10 + 12 + 10 = 32 registers and register file consists of (10 + 6) × 4 + 10 = 74 registers.

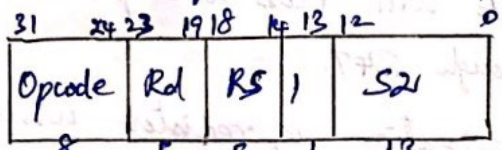
EXAMPLE RISC PROCESSOR

BERKELEY RISC-I

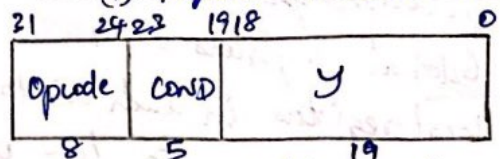
* Berkeley RISC-I is a 32-bit integrated circuit CPU.
* It supports 8, 16, (or) 32-bit data
* It has 32-bit instruction and a total of 31 instructions.
* It has a register file of 138 registers arranged into 10 global registers and 8 windows of 32 registers in each. The 32 registers in each window have an organization similar to the overlapped register windows.



(a) Register Mode (S2 - specifies a register)



(b) Register-immediate mode.



(c) PC relative mode

Berkeley RISC-I instruction formats

From the above instruction formats it is clear that 7 of the bits present the operation specification. 8th bit indicates whether to update the status bit after an ALU operation. For register - register instructions, 5-bit Rd field selects one of the 32 registers as destination for the result of operation. Operation is performed on the data stored in fields RS and S_2. (S_2 source register and sign-extended 13-bit constant)

For PC relative mode, it combines the last three fields to form a 19-bit relative address 'y' and is used primarily with JUMP and CALL instructions. The COND field replaces the Rd field for jump instructions and is used to specify one of the 16 possible branch conditions.

MIPS Architecture:

MIPS stands for Microprocessor without interlocked pipeline stages. It is a reduced instruction set Computer (RISC) developed by MIPS Technologies.

Registers:

MIPS-I has 32 number of 32-bit register (general purpose). Here last register is a link register.

Instruction formats:

Instructions are divided into 3 types: R, I & J. Every instruction starts with a 6-bit opcode. In

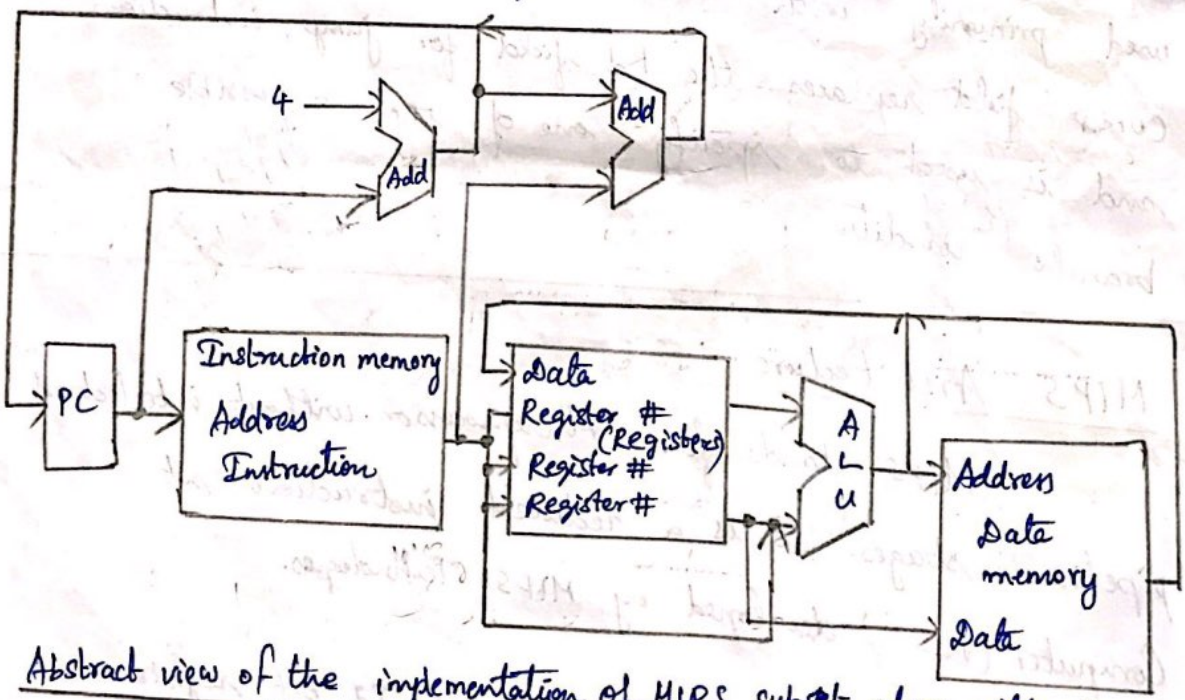
addition to opcode, R-type instructions specify three registers, a shift amount field & a function field.

I-type instructions specify 2 registers and a 16-bit immediate value.

J-type instructions follow the opcode with a 26-bit jump target.

Type	31	Format (bits)				0
R	Opcode (6)	rs (5)	rt (5)	rd (5)	Shamt (5)	funct (6)
I	opcode (6)	rs (5)	rt (5)	immediate (16)		
J	Opcode (6)	address (26)				

Instruction formats used in MIPS



Abstract view of the implementation of MIPS subset along with the functional units and major connections between them.

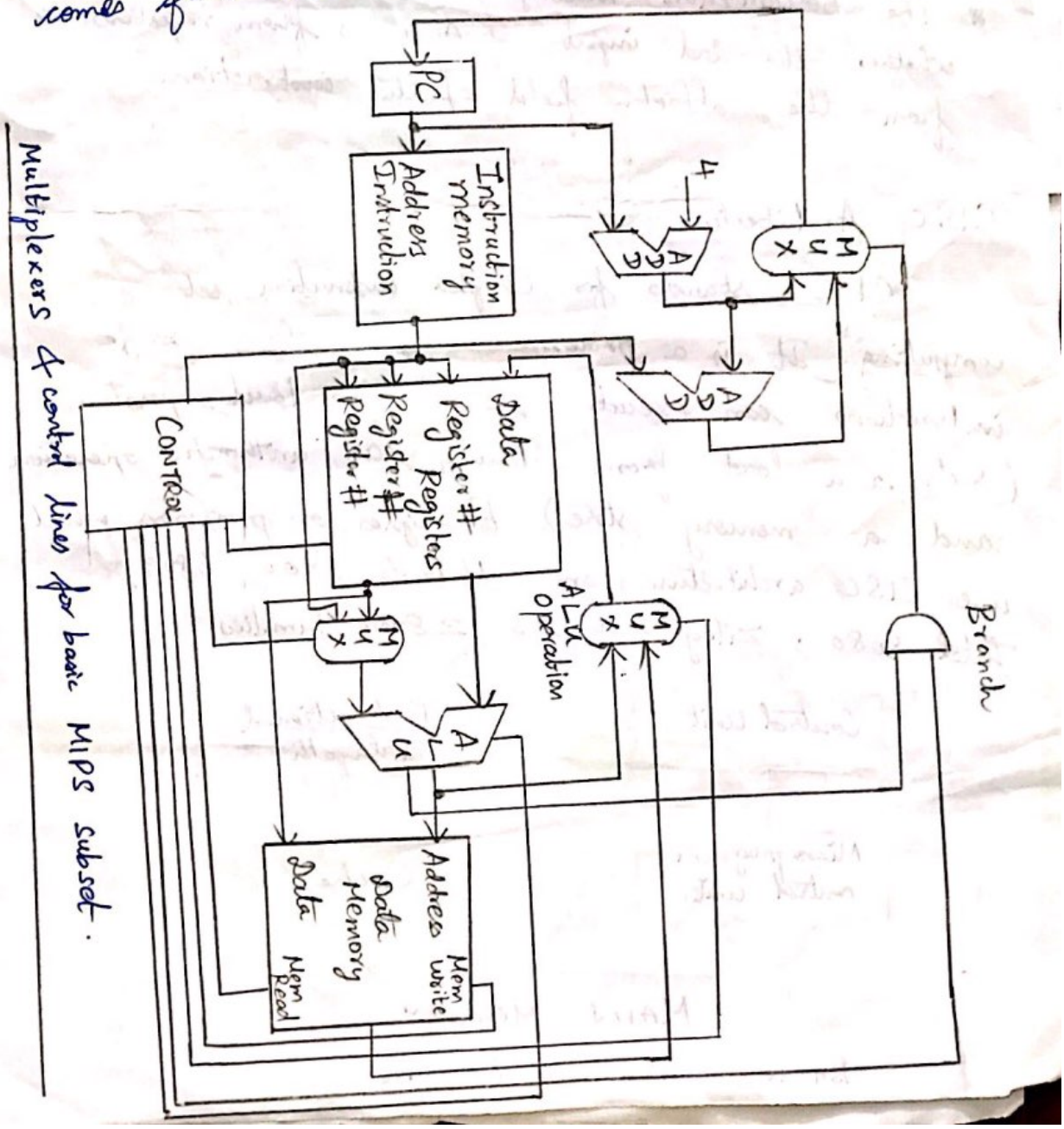
* By using program counter (PC), all instructions begin to supply the instruction address to the instruction memory.

* After the instruction is fetched, the register operands needed by an instruction are specified by fields of that instruction.

* Register operands can be operated to compute a memory address.

* If the instruction is an arithmetic-logical instruction, the result from the ALU must be written on to a register.

* Branches are in need of the ALU output to determine the next instruction address. The instruction address comes from the ALU (or) from an adder.

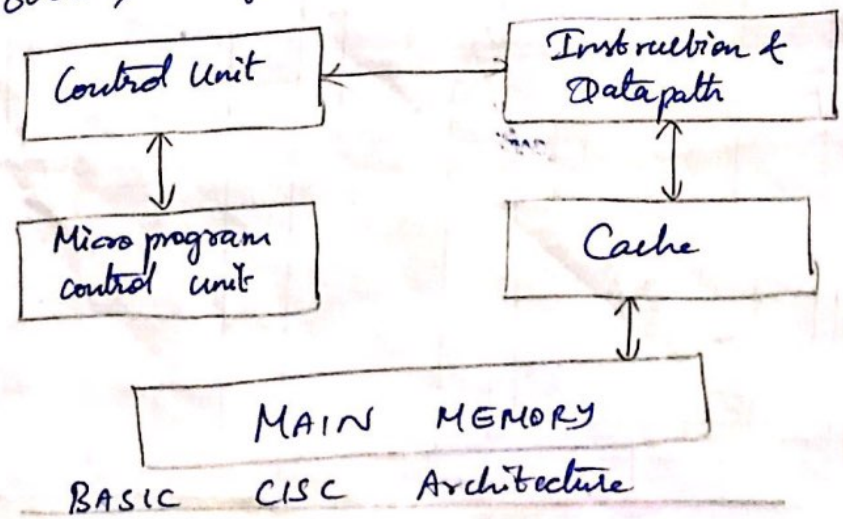


The previous figure shows the datapath of abstract view of MIPS subset with three multiplexers and few control lines for the major functional units. Control unit determines how to arrange the control lines for the functional units and multiplexer.

* The top multiplexer controls what value replaces the PC
* The middle multiplexer is used to steer the output of ALU (or) output of data memory for writing into the register file.
* The bottom most multiplexer is used to determine whether the 2nd input of ALU is from registers (or) from the offset field of the instruction.

CISC Architecture:

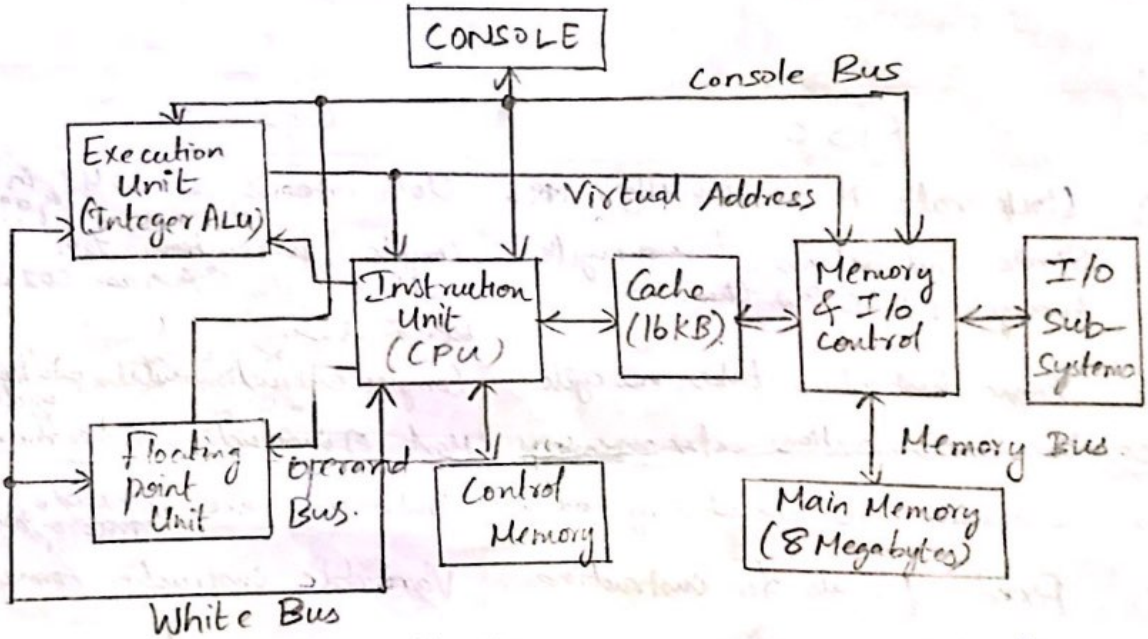
CISC stands for Complex instruction set computing. It is a processor design, where single instructions can execute several low-level operations (such as a load from memory, an arithmetic operation and a memory store). Examples of processors which use CISC architecture are Motorola 6800, 6809, Intel 8080; Zilog Z80, Z8, Z8000 families.



BASIC CISC Architecture

Example Architecture : VAX 8600 CISC processor.

VAX 8600 is a CISC processor developed by digital Equipment Corporation in 1985. This machine implemented a typical CISC architecture with microprogrammed control. Instruction set contained 300 instructions with 20 different addressing modes.



The VAX 8600 CPU, a CISC processor Architecture.

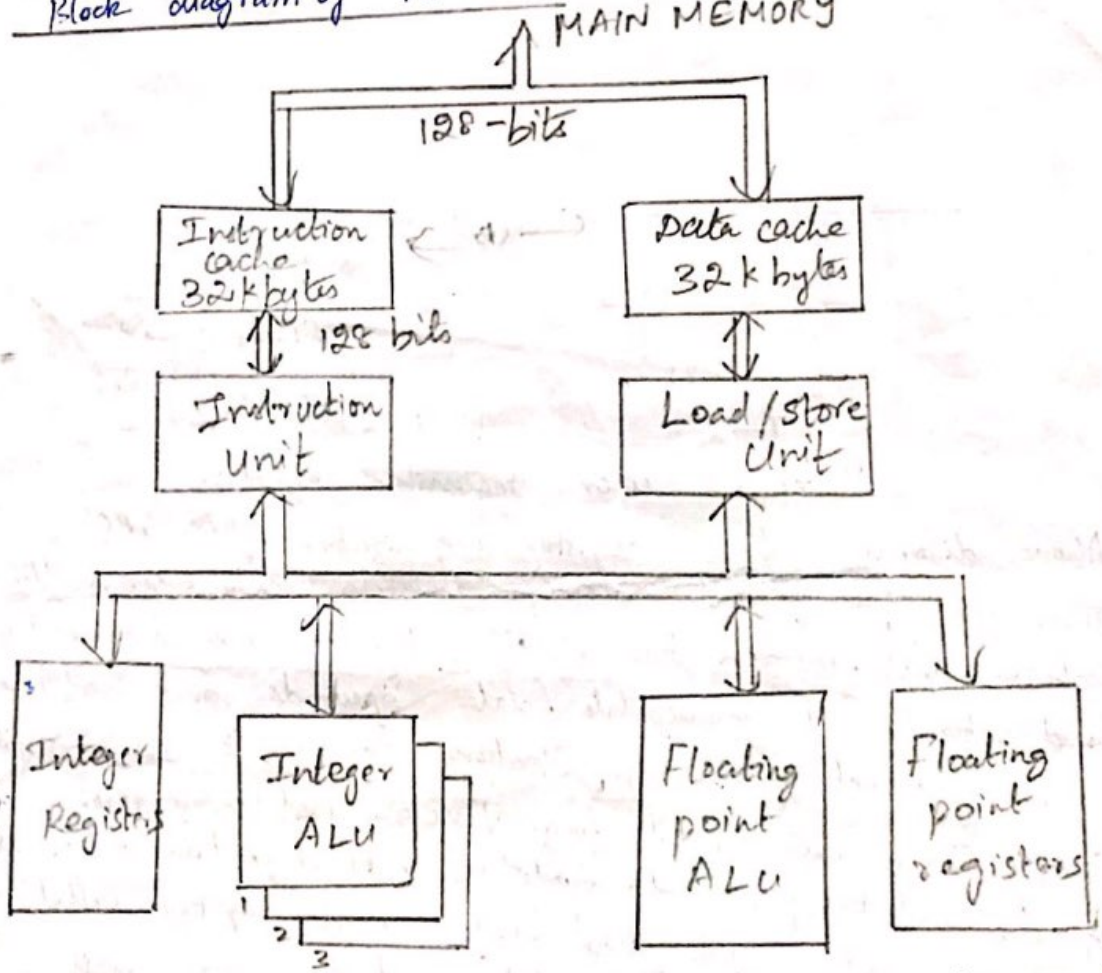
A translation lookaside buffer (TLB) was used in memory control unit for fast generation of a physical address from a virtual address. Both integer & floating point units were pipelined. Pipelining heavily relied on cache hit ratio & on minimal branching to avoid damage to pipeline flow. Instruction cycle varies from 2 cycles to 20 cycles. (Eg) Multiply & divide might tie up the execution unit for a large no. of cycles.

	RISC	CISC
1)	Clock rate is 50-150 MHz in 1993	Clock rate is 33-50 MHz in 1992
2)	Simple instructions take one cycle Average CPI is less than 1.5.	Complex instructions take multiple cycles. Average CPI is bet. 2 & 15.
3)	Simple instructions takes one cycle	Complex instructions takes multiple cycles.
4)	Very few instructions refer memory.	Most of instructions refer memory
5)	Instructions are executed by hardware	Instructions are executed by microprogram.
6)	Fixed formats for instructions	Variable instruction formats
7)	Few instructions	Many instructions
8)	Few addressing modes & most instructions have Register-register addressing modes.	Many addressing modes.
9)	Complexed addressing modes are synthesized by software	Supports complex addressing modes
10)	Multiple register sets	Single register sets.
11)	Highly pipelined	Less or No pipelined
12)	Complexity is in the compiler	Complexity is in the microprogram.

POWER PC :-

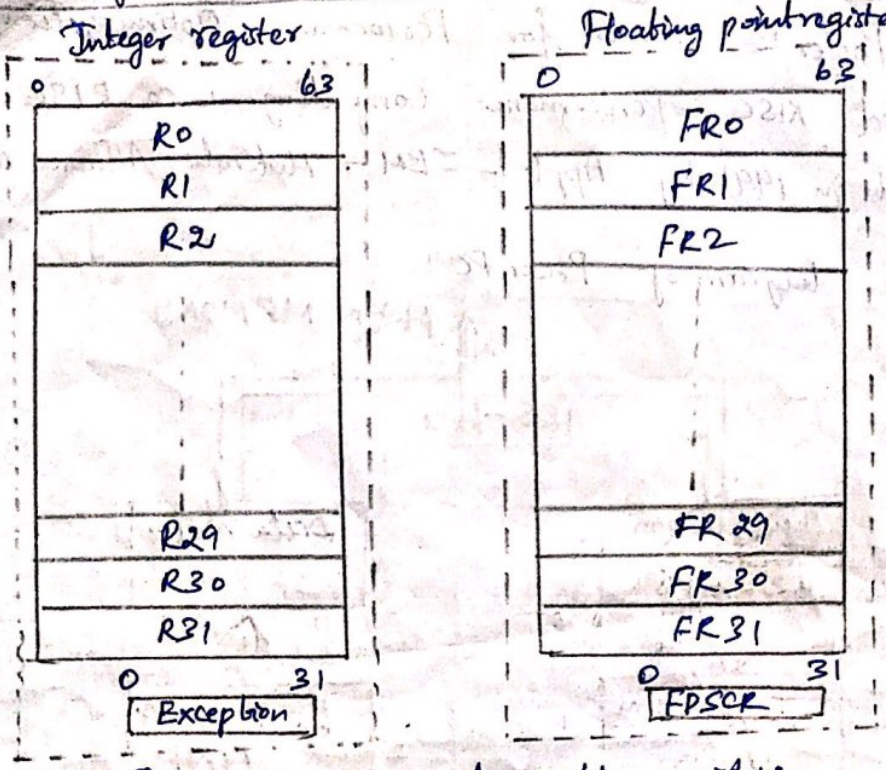
PowerPC stands for Performance Optimization with Enhanced RISC – Performance Computing, is a RISC architecture created in 1991 by Apple – IBM – Motorola Alliance called AIM.

Block diagram of Power PC :



Above shows the block diagram of PowerPC 620. It has 2 caches – instruction cache and data cache. Instead of single ALU, the powerPC 620 has 3 integer ALUs to handle integer and logic operations and one floating point ALU to perform floating point operations. It also has an instruction unit whose purpose is to fetch instructions from the main memory and send them to integer and floating point unit for execution.

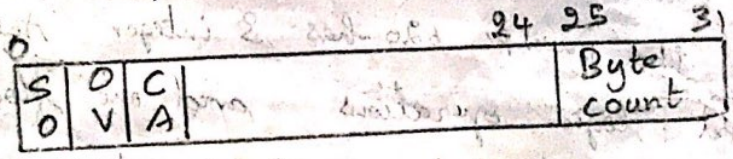
PowerPC register Organization:



Power PC user Accessible registers.

Above diagram shows register organization of powerPC. There are 32 general purpose registers R0 to R31. in the integer unit. Each has 64-bits. Registers may be used to load, store and manipulate data operands and used for register indirect addressing. Floating point unit has 32 registers. FR0 to FR31, each 64-bit long. These are used for floating point operations. Floating point status & control register (FPSCR) contains control & status bits

Integer unit has one (or) more register called exception register. It is used to indicate exceptions in integer arithmetic operations.



Exception register

Byte count: specifies number of bytes to be transferred by load / store using indexed instruction

SO: Summary overflow
OV: Overflow
CA: Carry.