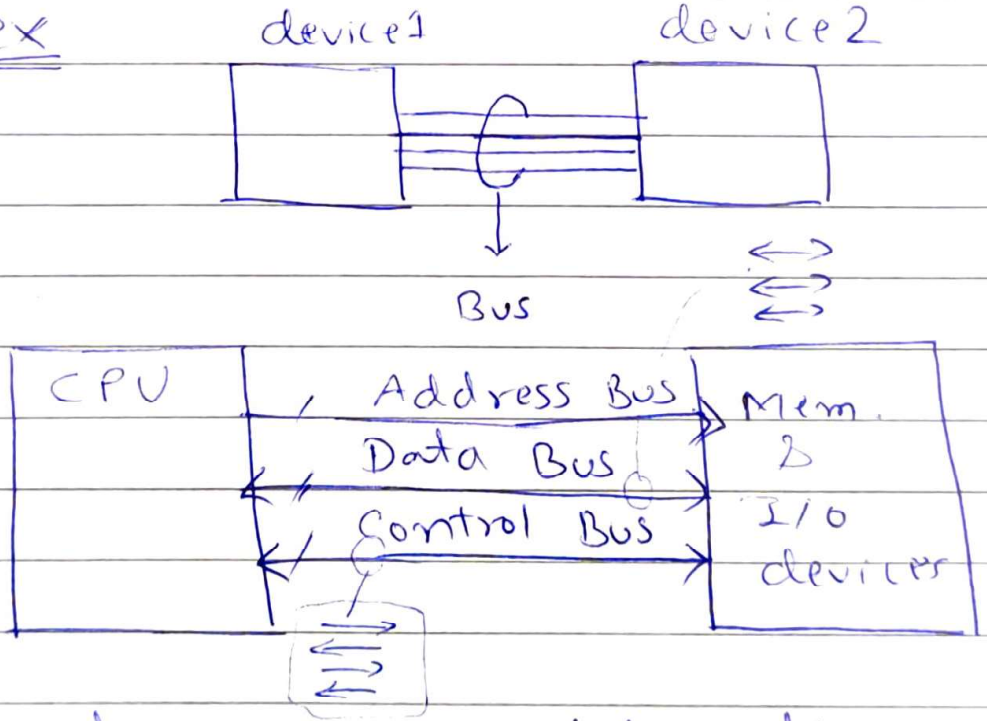
System Buses (Connection lines b/w two devices with comp. system)

- Address Bus
- Data Bus
- Control Bus

ex



Address Bus: unidirection, means only CPU can generate

Data bus :: Bi-directional (duplex).

Control Bus: bidirectional : every individual line is unidirectional
Interrupt signal by I/O → CPU

Mem can generate two type of signal for CPU
→ wait and → Ready
(CPU is operating at very fast speed but memory not operating at that speed)

(*) Memory Cycle Time
↳ Time from accepting a request till memory gets ready again to perform next operation

CPU Register

(*) Within CPU ~~are~~ small storage units are register

(*) While executing prog. inside CPU registers are used

→ General Pupose Registers (GPRs)
↳ Storing general content
↳ O/p from ALU
↳ i/p to ALU
↳ ~~store~~ temp. storage

→ Special content

→ Special Purpose Registers

1) Accumulator (AC)

2) ~~Program Counter~~
2) Program Counter

3) Instruction Register (IR)

4) Flag Register / Program Status Word (PSW)

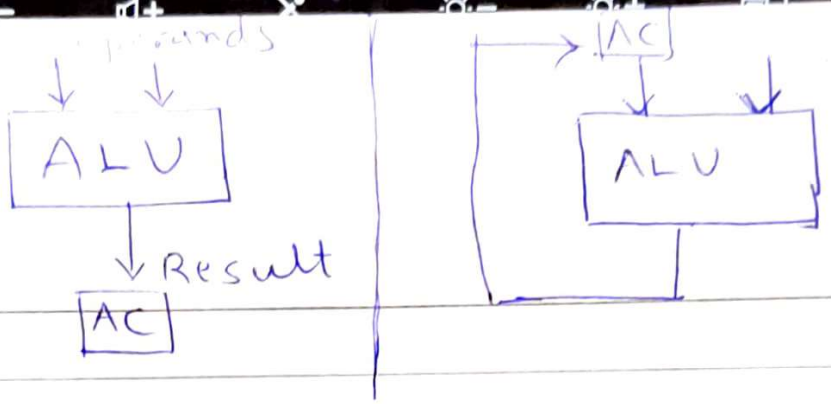
5) Address Register (AR) / Mem. Add. Register (MAR)

6) Data Register (DR) / MDR / M. Buffer Regis

* not full list, there are are others as well.

Accumulator: (not specific to an architecture, generic point of view)

→ used to store result of ALU and sometimes to store one of the input for ALU



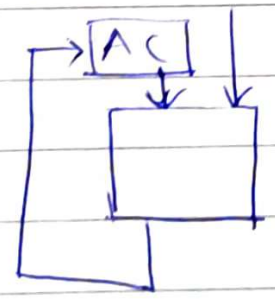
First o/p from ALU to Accumulator (AC)

(*) AC - base architecture → when one input is taken from AC and another i/p taken from somewhere else

Types of Architecture

(*) Based on input to ALU

1) AC - Based

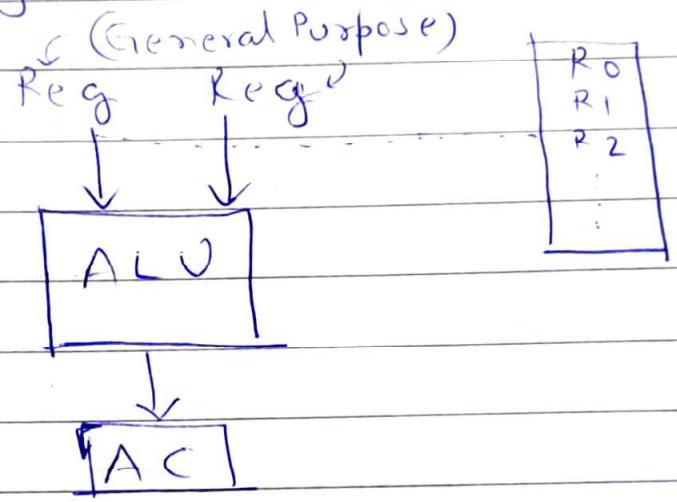


↳ instruction
↳ opcode operand
ex.

ADD R3 → AC ← AC + R3
MUL R1 → AC ← AC * R1

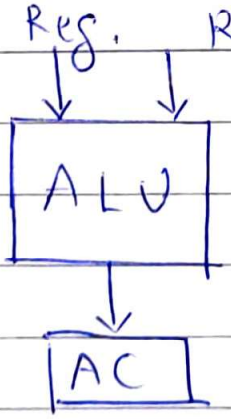
no need to mention AC in this architecture as one iput is taken from AC.

2) Register based architecture

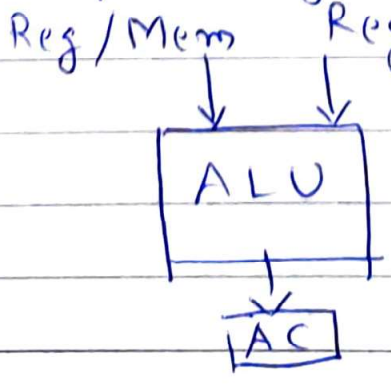


All inputs taken from register

3) Register - Mem. Architecture



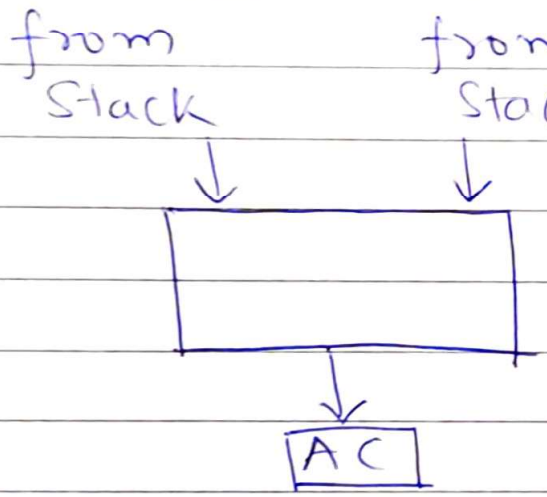
4) Complex System Architecture



Both i/p can be taken be from either Reg Or Mem.

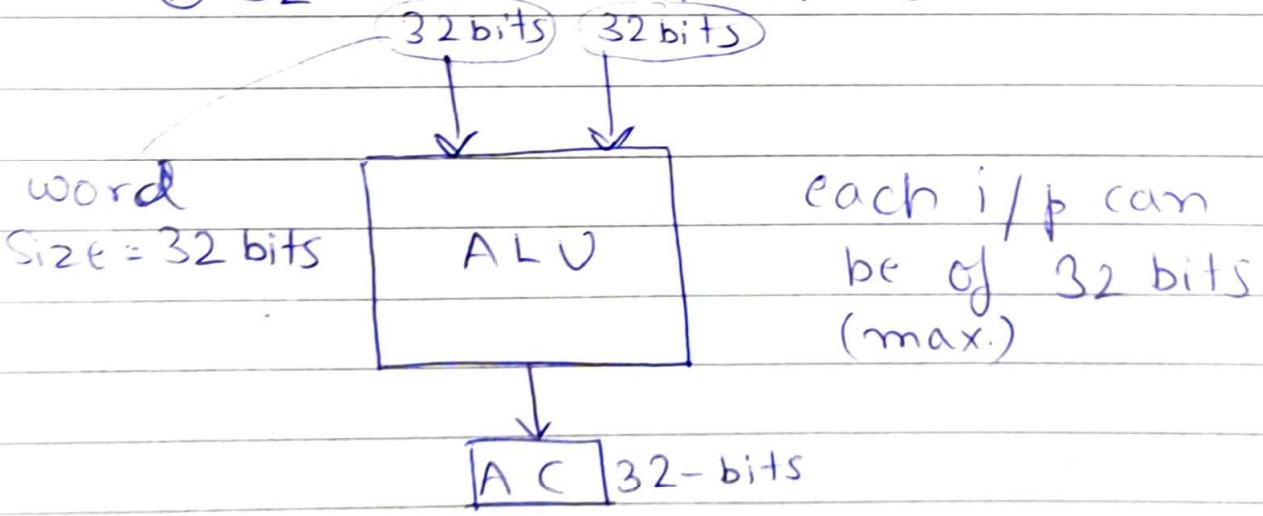
can be in memory

5) Stack Based



Note

* 32-bit architecture

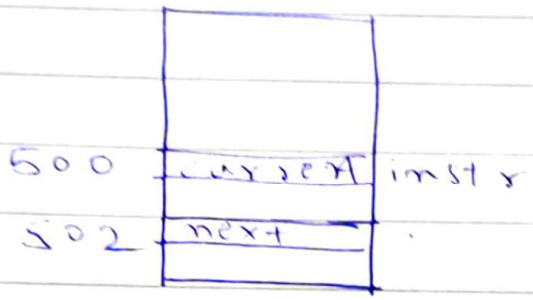


* If result of two i/ps is more than 32 bit then extra bit is stored in carry flag register

* Similarly for 64 bits architecture
  → word size = 64 bits

Program Counter

Store address of next instruction

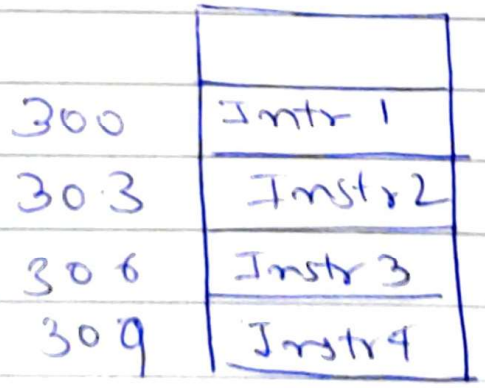


Q) A CPU has 24-bits instruction. A program starts at address 300 (in decimal) which of the following is a legal program counter value?

A) 400 B) 500

C) 600 D) 700

A) 24 bits → 3 Bytes



So answer is C) 600 as it's multiple of 3

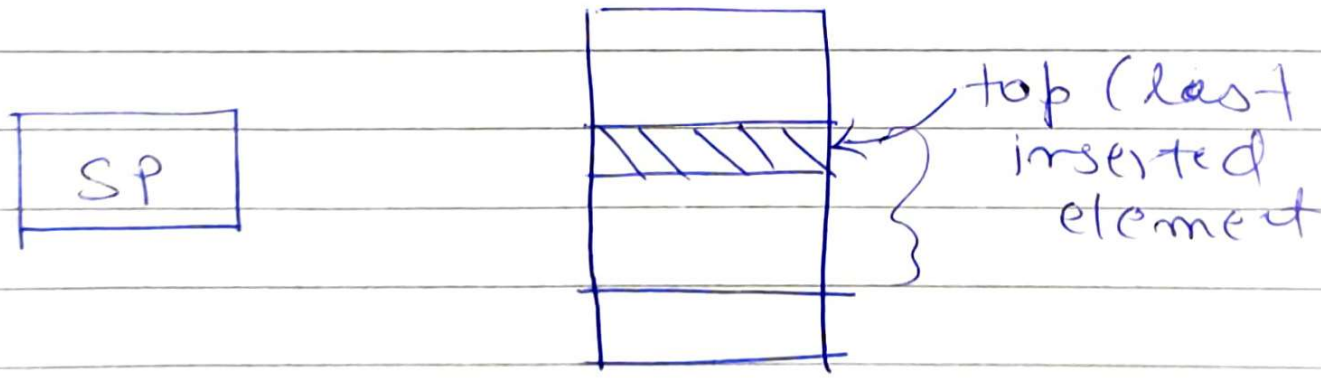
Instruction Register

used to store current instruction which CPU is executing.

Stack Pointer

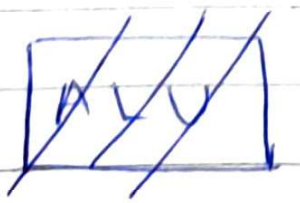
in the memory for execution purpose there should be a stack

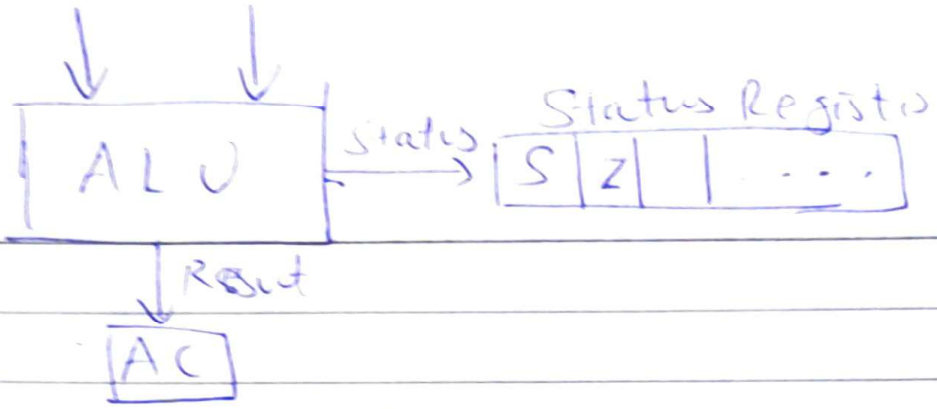
LIFO



→ used to store address of the top element of stack

Flag Register (Status Register)





→ 8055 will have diff. Status Register

→ 8086 will have diff.

and so on.

S = Sign

Z = Zero

* if ALU result is '0' then Z = 1

if ALU ≠ 0 then Z = 0

* S { 0 — +ve ; 1 — -ve }

why to set status register ?

→ It stores the Status of ALU results

→ to check conditions.

↳ ex. (a > b) → CPU will a-b in ALU and status of Result is +ve then a > b condition is true

Program Status Word

In some CPUs

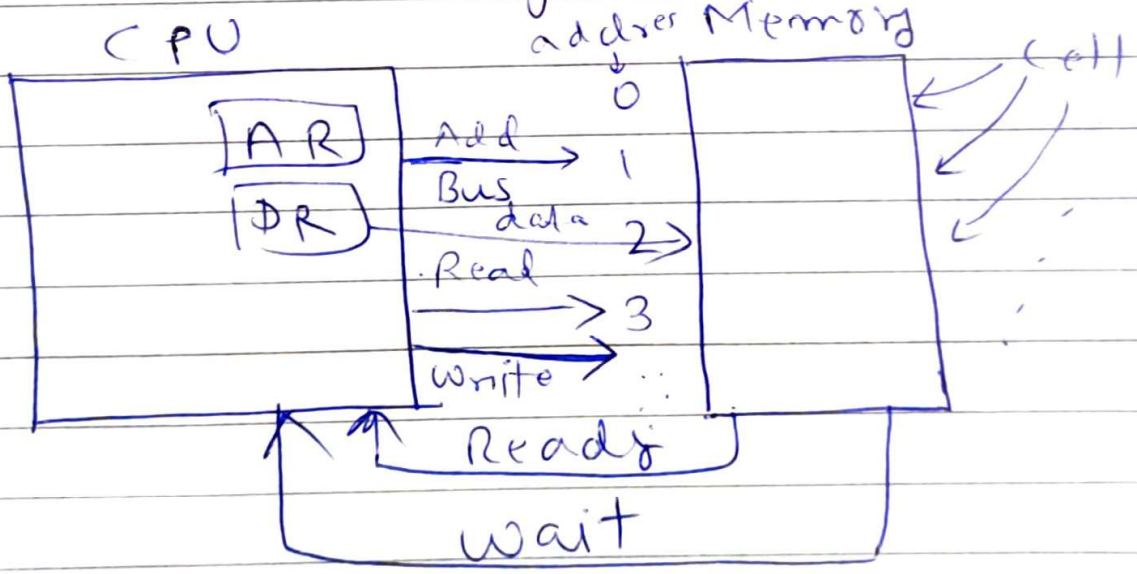
In some CPUs

Status Reg.

AC + Status Reg.

MAR : Send address to memory

MDR : Send data to memory (memory write) and receive data from memory (mem. read)



Mem read → CPU sends add to mem using add bus

→ CPU enables read control signal for memory

→ Mem. performs read opr. & sends the data to CPU using data bus

Mem write

1) CPU send add to mem. using add bus

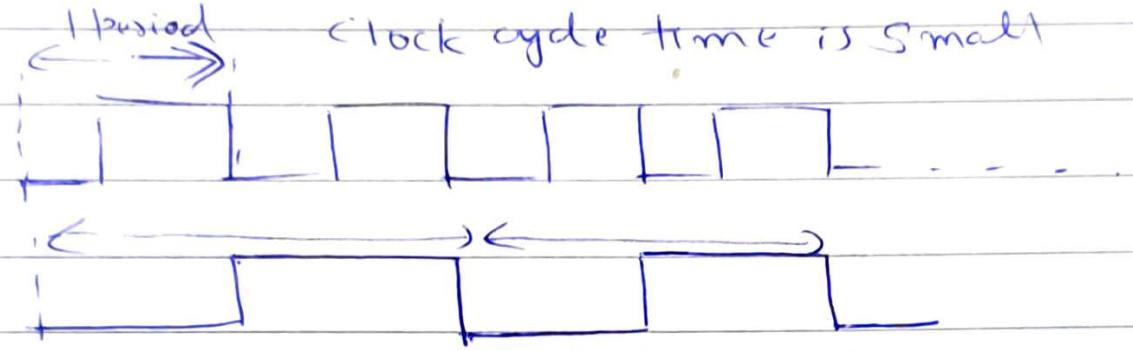
2) CPU sends data to mem. using data bus

3) CPU sends enabled write signal to mem.

4) Mem. perform write opr. on provided add.

Software Architecture

CPU Performance



For 1st clock

$$F = 1\ GHz = 10^9\ Hz$$

$$T = \frac{1}{f} = \frac{1}{10^9}\ sec = 1\ nsec.$$

For 2nd clock

$$f = 500\ MHz = 500 \times 10^6\ Hz$$

$$T = \frac{1}{f} = \frac{1}{500 \times 10^6}\ sec$$

$$= 2\ msec.$$

So clock 1 is faster

Clock rate f depends on

→ clock is becoming faster

→ CPU organization is getting better

ex) ADD R1, R2

→ Fetch PC out MAR. Read . . . (1 clock cycles)

→

→

- May be on a diff. CPU with diff. organization less no. of micro instructions are required

- Speed with which a computer executes programs is affected by the design of its hardware and its machine language instruction.

- Since most programs are written in HLL so performance of compiler also affects the performance. (that translates programs into machine language).

- Compiler + machine instruction set + hardware in Synch

- Processor time: sum of processor time needed to execute the program.

Processor Clock

$$R = \frac{1}{P}$$

* Processor circuits are controlled by a timing signal called a 'clock'

* Clock defines regular time intervals called clock cycles.

* Machine instruction

ADD R1 R2

↳ PC_{OUT} MAR_{IN} READ WAIT

each of → step is executed in one clock cycle.

length P of one clock cycle

$$R = \frac{1}{P}$$

Cycles per second → Hz

Million → Mega (M)

Billion → Giga (G)

500 million/sec → 500 MHz

1250 " /sec → 1.25 GHz

$\frac{1}{500\ MHz} = 2\ ns$

$\frac{1}{1250\ GHz} = 0.8\ ns$

Basic Performance Equation

$$T = \frac{N \times S}{R}$$

T : processor time required to execute a program that has been prepared in some HLL

N : Machine language instr.

S : avg. no. of basic steps needed to execute one machine instruction is S, where each basic step is completed in one clock cycle.

R : Clock Rate R / Sec.

$$T = \frac{N \times S}{R}$$

→ whatever process is running is user mode

→ process running is in kernel mode

Hardwire design

Processor shld have faster (chp character)

Add A, B

→ executed using faster adder circuit

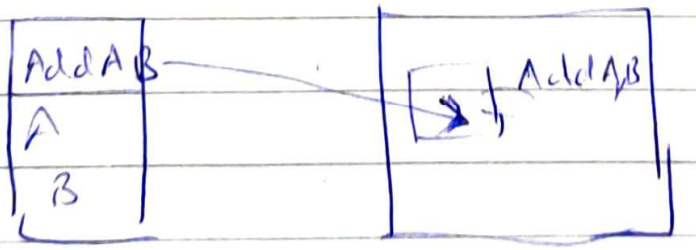
→ Since memory access is there so it shld be faster (mem access time)

Fetch, Decode, OF, Execute, Write

faster

Fetch, Decode, ~~Execute~~ Write

→ can be saved by using cache



To execute one step is performed in one clock cycle → 1 step

1 cycle / instruction

1 CPI

Instruction Set

Add A R_1 = S = 5

Add A B = S = 6

→ F — 1 clock — S = 6 — 2 clock — S = 12
E
D F
E
W

If 2 GHz (0.5 ns)
for 4 GHz (0.25 ns)

→ Same hardware / mem mgmt etc.

→ F → 2 clock
E → " "
:
W → 2 " — S = 12

(*) Reduced P but S becomes double
So T remains same

(*) How to reduce T → better hardware so P increases so that each step is also completed in one cycle only

$T = N \times S \times P$

or

$T = \frac{N \times S}{R}$

$P = \frac{1}{R}$

$C = A + B$

ISA

P1 (CISC) (complex instruction set computer)

ADD A B C

$A + B \rightarrow C$

mem(A) + mem(B), one to (C)

P2 (RISC (reduced instruction set computer) (not complex instruction allowed)

MOV A R0

MOV B R1

ADD R0 R1

MOV R1 C

There are 1/2/3 address instruction

ADD A [Accumulator]

ADD A B

ADD A BC

P3 [one mem. operand used]

MOV A, R0

ADD B R0

MOV R0, C

P4 (Single operand)

LOAD A

Add B [accumulator contains A+B]

STORE C [accumulator → C]

[load/Store used not MOVE]

(X) All 4 have diff ISA

P1 – P4, i3, i5, i7
AMD

(A) 90S → CISC
(A) Now mostly RISC

→ Which one is better

Hardware, memory, clock rate, cache same for CISC, RISC
→ P is same

We are concerned for "N"

P1 → N = 1
P2 → N = 4
P3 → N = 3
P4 → N = 3

(P1) Fetch
Dec
OFA
OFB
EX
WC
6 steps

NXS = 6 · 1 × 6
= 6

(P3) F, D, OF — S = 3

NXS
= 3 × 3
= 9

which one is better

$\Rightarrow \frac{6}{R}$ vs $\frac{9}{R}$

suppose for P1

100 (assembly lang instructions)

avg S = CPI
= 6

100 × 6

for P2

N = 300
S = CPI = 3
N × S = 900

so CISC is better than RISC

But nowadays RISC is preferred with the help of pipelining.

Pipelining

ex

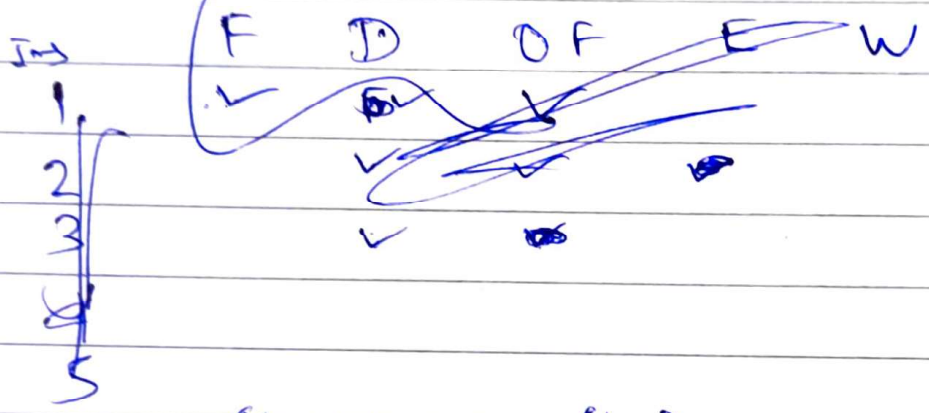
P2 → MOV
Add
MOV
} Simultaneous execution

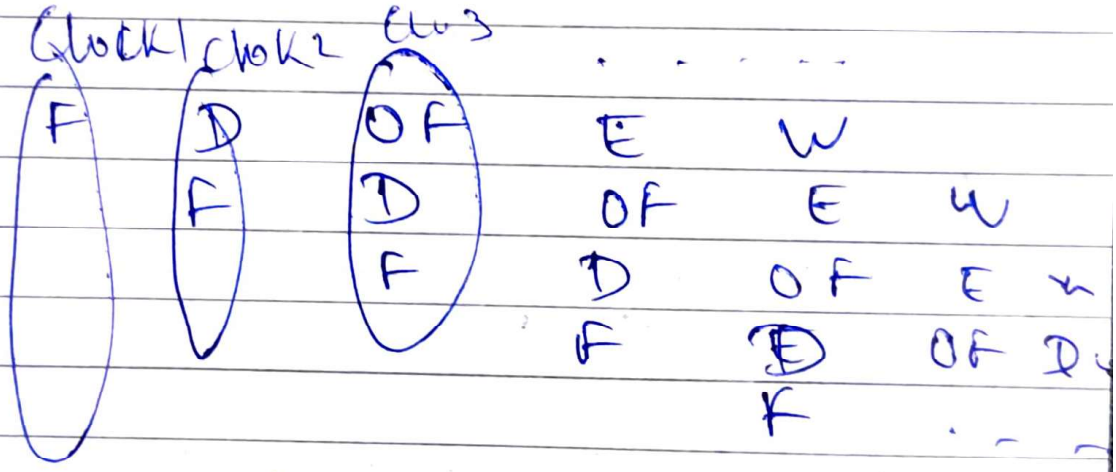
Sequence of steps

(F) - D - OF - E - W

~~one~~ → first instr. in first clock cycle

→ next decode





↳ S = 5

For 100 instruction program after 5th clock cycle

↳ .

effective value of 'S' = 1

Ex F D E

F D E

F D E

F D E

non pipeline ⇒ 10 × 3

pipeline → 12.

FDE FDE FDE FDE 10

Pipeline of CISC = 100/5 = 10% = 100

RISC = 300/5 = 300 = 1 = 300

→ but we have to see on some other aspects as well in pipelining

→ talk about hazards
data hazard
(Stalls)

→ logical correctness of program is maintained

For CISC

Add A, B, C
Add E, R1

F - D - OF - OF - E - W
F D - - OF - E W

3 Stalls

For RISC (P3)

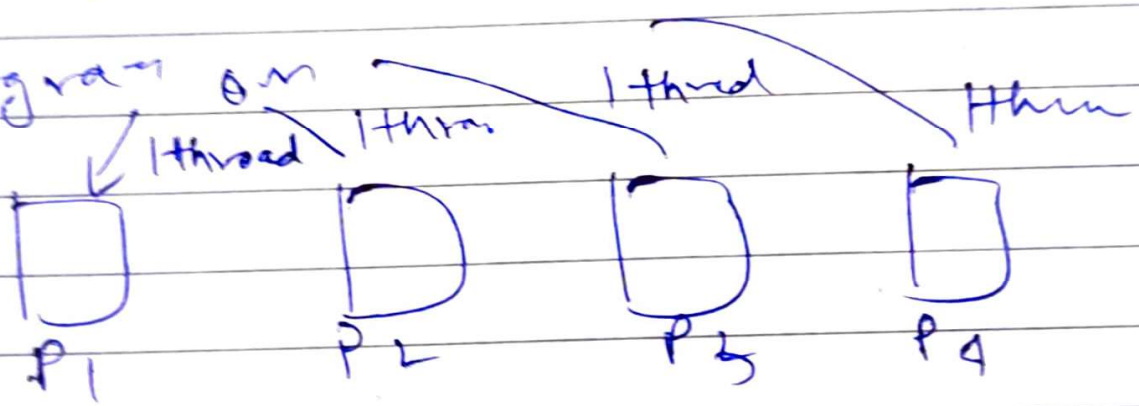
F D OF
F D OF E
F D - W
1 Stall

Easy to hard hazard (Stall) in RISC (simpler instruction)

④ Logical correctness → done by compiler

→ This is how ISA affects performance

Superscalar opr



→ multithreading (Covered in OS)

S can be reduced to 1 by using pipelining on thread ~~or~~ and cores

we can reduce S < 1 also using multicore processor

it can be done in RISC than in CISC

$$T = \frac{N \times S}{R} \quad , \quad N \times S \times P$$

Compiler

→ Reduce N we need to have stable machine instruction set

→ Compiler shld use it in good way.

→ optimizing compiler
↳ reduces N x S

→ No. of cyles of is dependent not only on the choice of instructions but also on the way they are ordered in the program. (chapter 8)

→ Compiler re-arranges program instructions to achieve better performance. without effecting the result of computation

→ Compiler shld be closely linked to the processor architecture.

→ Usually compiler and processor are designed at the same time with both designers in proper communication with each other.

→ Ultimate goal → reduce total no. of clock cycles.

Performance Measurement

→ Computer designers use performance estimates to evaluate the effectiveness of new features

→ Not only "T" but other features also affect the performance

→ For proper comparision. Standardized programs must be used.

Benchmark :→ performance measure is the time it takes a computer to execute a given benchmark

(*) Initially some artificial programs were used for standardizing benchmarks.

↳ but they do not stand in right standard when compared with real world applications.

→ Non-profit organization called System Performance Evaluation Corporation (SPEC) publishes representative application programs for different application domains together with test results for many commercially available computers.

→ For general purpose computers a suite of benchmarks.
↳ programs was selected in 1989
↳ modified in 1995 and 2000.

Programs:
→ game playing
→ compiler and database applica
→ numerically intensive programs in astrophysics and quantam chemistr

→ Program run testing PC and running time on a real PC is measured. (no simulations)

→ Program also run of reference PC

SPEC 95 → SUN SPARCstation 10/40

SPEC 2000 → Ultra SPARC 10 work station.

SPEC Rating

$$= \frac{\text{Running time on ref. PC}}{\text{Running time on PC under test}}$$

SPEC rating of 50 means PC unter test is 50 times as fast as Ultra SPARC 10 for a particular ~~test~~ benchmark.

Test run for all programs in SPEC Suite

→ Geometric mean of the results is computed

$$\text{SPCE Rating for } i \text{ prog} = \left(\prod_{i=1}^{n} \text{SPEC}_i \right)^{1/n}$$

SPEC measure is combination of all ~~measures~~ factors affecting performance such as compiler, OS, processor, mem of system.