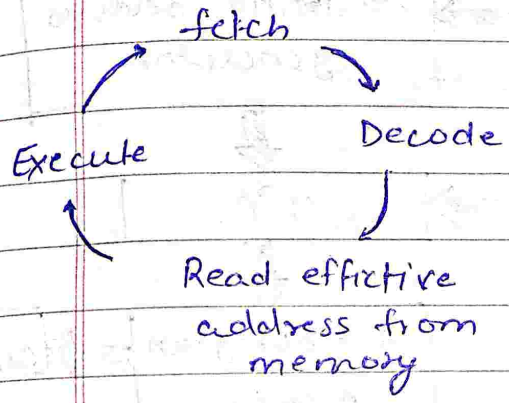
दिनांक - 17-03-2023

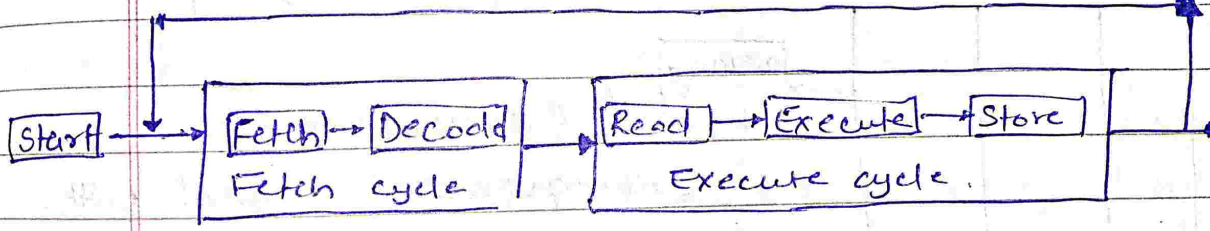
सी.पि.यू Data इंस्ट्रक्शन / ~~डेटा~~ साइकिल

Saathi

→ Instruction cycle.



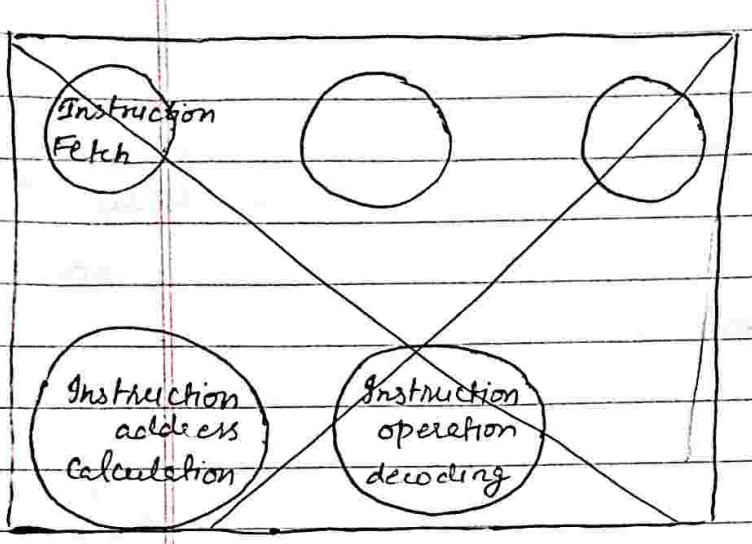
In the instruction cycle step 1 and 2 is mandatory for each and every cycle and step 3 to 5 that is called execute cycle, may be changed with each instruction. So, the total time, takes for that take to carry out five steps is called machine cycle or instruction cycle.



COA 23/03/2023

☆ Instruction set.

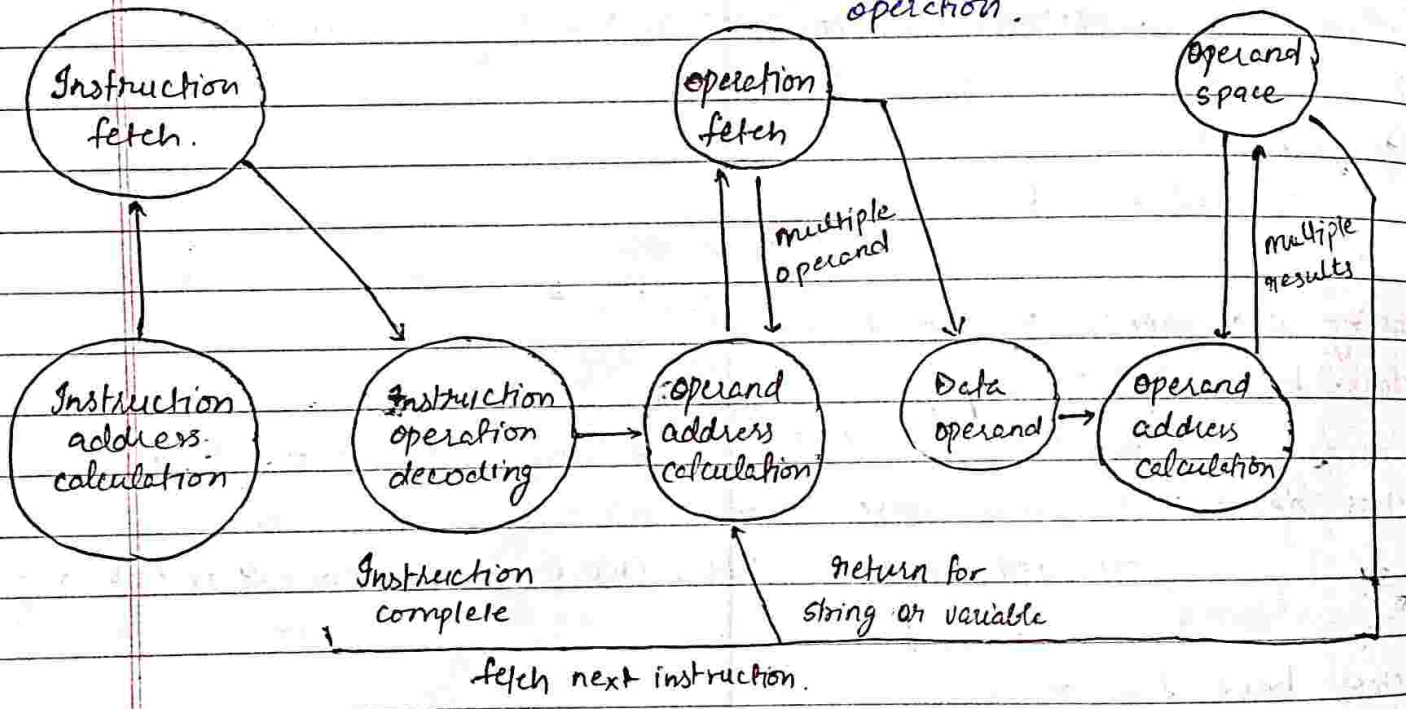
set of commands.



The instructions contains following elements :-

i) operation code - specify the operation to be performed. The operation is specified by binary code, or op code.

ii) Source operand reference. The operation may involve one or more source operand i.e. the operand that are inputs for operation.



INSTRUCTION CYCLE STATE DIAGRAM

iii) Result operand operation - the operation may produce a result.

iv) Next instruction reference - this tells the processor, where to ~~place~~ fetch the instruction after the ~~ex~~ execution of this instruction is complete.

Instruction format

ADD R, X

↑ op code ↑↗ oper

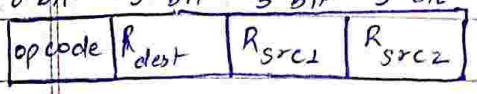
* Address format :-

Depend on the source operand reference, result operation ref. & next instruction reference there are four categories of machines or addresses or references.

i) Three address instruction
ii) Two address instruction
iii) One address instruction
iv) RISC address instructions.

1) Three address instruction

This type of Instruction contains 3 address fields.



Here, 2 address bits are used for source address and one address bit contains destination or result address.

ex:- ADD DEST, SRC1, SRC2.

Instruction.

comments :-

M(DEST) = M[SRC1] + M[SRC2] ;

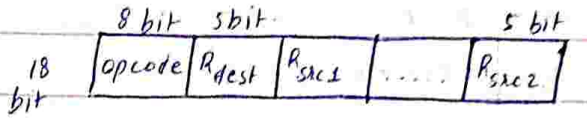
Q Convert the equation written in C language into the 3 address instruction format.

A = B + C * D - E + F + A

Ans:

Instruction	comments.
MUL T, C, D	; T = C * D
ADD T, T, B	; T = B + C * D
SUB T, T, E	; T = B + C * D - E
ADD T, T, F	; T = B + C * D - E + F
ADD A, T, A	; A = B + C * D - E + F + A

2) Two address instruction



ex:- ADD DEST, SRC M(DEST) = [DEST] + [SRC]

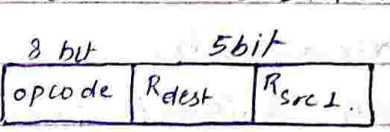
LOAD DEST, SRC M(DEST) = [SRC]

Q. A = B + C * D - E + F + A.

A.

Instruction	Comments.
LOAD T, C	; T = C .
MUL T, D	; T = C * D
ADD T, B	; T = B + C * D
SUB T, E	; T = B + C * D - E
ADD T, F	; T = B + C * D - E + F
ADD A, T	; A = B + C * D - E + F + A.

3) one address Instruction



ex;

ADD ADDR	ACC = ACC + M[ADDR]
LOAD ADDR	ACC = M[ADDR]

Q. A = B + C * D - E + F + A

A.

Instructions	comments
LOAD C	load C to the accumulator
MUL D	ACC = C * D

ADD B	ACC = B + C * D	
SUB E	ACC = B + C * D - E	
ADD F	ACC = B + C * D - E + F	
ADD A	ACC = B + C * D - E + F + A	
STORE A	store accumulator content to A.	

Saathi

Date ___/___/___

Q $A = B + C * D - E + F + A$

A

Instructions	Comments
PUSH E	PUSH ([E])
PUSH C	PUSH ([C])
PUSH D	PUSH ([D])
MUL	PUSH [[C] * [D])
PUSH B	PUSH ([B])
ADD	
SUB	PUSH([B + C * D] - [E])
PUSH F	
ADD	PUSH([B + C * D - E] + [F])
PUSH A	
ADD	PUSH(B + C * D - E + F + A)
POP A	

4) zero address instruction

8 bit | opcode |

Zero address instruction set, stacks are used to perform the operations, stacks supplies operands and recieve the result after the operation. The operation type of instruction do not need any address field in stack organisation computer. only special instructions load and scroll use are used for address field. This type of address instruction are refferred as stack machines.

for ex. HP3000

zero address instr. include only opcode without operand and the operand are used with stack operation i.e PUSH & POP.

for ex:

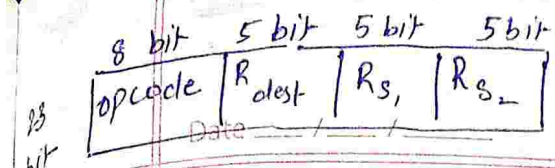
PUSH ADDR — PUSH ([ADDR])

ADD — PUSH (POP + POP)

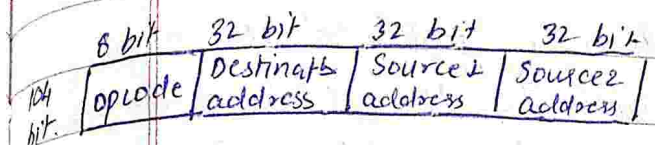
☆ RISC Instruction.

This type of instruction are restricted to use ~~and~~ of LOAD and store instruction when communicating with memory and CPU. All other instruction are executed in register in CPU without refferring to the memory. A program for a RISC type CPU consist of LOAD and STORE instruction, that has one memory and one register address and computation type instr^{n} that have 3 addresses with all 3 specified processor register.

Page No.



- Register address format



- Memory address format

ex: LOAD RD, ADDR RD = [ADDR]

STORE ADDR, RD

Q. A = B + C * D − E + F + A.

A

Instructions		Comments
LOAD	R1, B	R1 = [B]
LOAD	R2, C	R2 = [C]
LOAD	R3, D	R3 = [D]
LOAD	R4, E	R4 = [E]
LOAD	R5, F	R5 = [F]
LOAD	R6, A	R6 = [A]
MUL	R2, R2, R3	R2 = R2 * R3
ADD	R2, R2, R1	R2 = R2 + R1
SUB	R2, R2, R4	R2 = R2 − R4
ADD	R2, R2, R5	R2 = R2 + R5
ADD	R2, R2, R6	R2 = R2 + R6
STORE	A, R2	

Date - 24/03/2023 COA

5 marks

Saathi

☆ ADDRESSING MODES.

Information involved in any operation performed by the CPU needs to be addressed. Most of instructions have operands so some way is needed to specify where they are stored so, thats why we need the addressing mode. To perform any operation we have to give corresponding instructions to the microprocessor. In each instruction programmer has to specify 3 things.

i) operation to be performed.
ii) address of the source data.
iii) address of the destination result

The method by which the address of source of data or the address of destination of result is given in the instructions is called addressing mode. The term addressing mode refers to the way in which operand of the instruction is specified. The rules of interpreting or modifying the address field of instruction is called addressing mode.

☆ Effective address.

The address of operand produced by the application of the rule for interpreting or modifying the

Date __/__/____

Saathi

address field of the instructions before the operand is actually referenced. The no. of different addressing modes are explained below :-

The type of addressing modes :-

1) Direct addressing mode (absolute)

In the direct addressing mode the address of the operand is given in the instruction itself.

For ex :- LDA 2500H

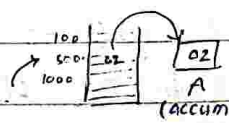
where LDA (load accumulator) is the operation & 2500H is the address of the source and accumulator is address of the destination.

In the instruction load the continues memory location 2500H in accumulator

2) Indirect addressing mode

In the indirect addressing mode the address field of the instruction gives the address at which the effective is stored in the memory

ex. LOAD [1000], R1

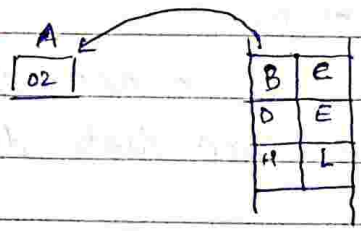


where LOAD is operation and in bracket 1000 is the memory location where address is available of content.

R1 is the destination register.

3) Register addressing mode

In the register addressing mode the operand is in general purpose register. for ex. mov A, B



B has 02

where, mov is the operation.

B is the source, A is the destination the instruction move the contents of B to register A

Register

4) Indirect addressing mode.

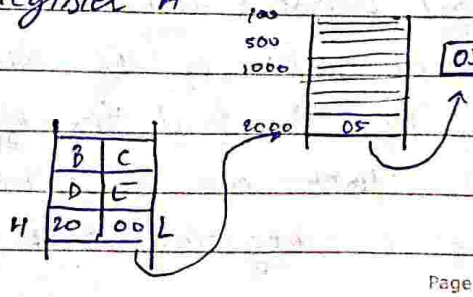
In the register indirect addressing mode the address of the operand specified by a HL register pair. for ex. move A, M

where, move is the operation.

M is memory location specified by HL register pair.

A is the destination address.

In the example, the instruction move of the data content from the memory location specified by HL pair to the destination register A



Page No.

Saathi

Date __/__/____

5) Immediate addressing mode
In the immediate addressing mode the operand is specied within the instruction itself. -
for ex. LDA # 1000
where, LDA is opeiation and 1000 is consant immediate data.
or
LOAD R1 #1000 R1 [1000]
or (Implied)

6) Implicit ^ addressing mode
The implicitt addressing mode says. If the source of the data as well as address of the destination of result is fixed there is the no need to give any operand along with the instructions.
for ex. CMA
CMA is complement of accumulator
ACC ← $\overline{ACC}$

7) Relative addressing mode.
The relative addressing mode provides the effective address after adding the contents of PC. with the address part (offset address) of the instruction. The offset address is added to the current value of PC during the execution to find the effective address. Offset can be either positive or negative. Such address is usually used for branched instruction.
Formula.:
E.A = address part of the instruction + the content of the P.C.
ex LDA & Address.
where LDA is operation and &Address is the address ~~of~~ to find the effective address with addition value of PC.
ACC ← M [ACC + PC]

8) Indexed addressing mode.
In the indexed addressing mode the content of Indexed registei is added to the address part of the instruction to obtain the effective address. ~~Index~~
Indexed register is the special registes in the CPU in the application of array, the distance betiseen the beginning address and the address of the index value stored in the registes.
for ex. LDA ADRS (R1)
where
ADRS is the
ACC ← M[ACC + IR]

Page No.

these are questions ⬇️

Date

E.A = Address part of the instruct^{n} & + the content of the I.R

PC = 250

R_1 = 250

ACC

Memory

	addr	data
250	opcode	mode
251	address or data = 500	
252	next instruct^{n}	
400	700	
500		
	400	
700		
800	300	
900	200	

9) Base register addressing

In the base registers addressing the content of base register is added to the address part of the instruction to ~~find out~~ obtain the effective address.

E.A = address part of the instruct^{n} + contents of BR.

LDA ADRS (R_2)

ACC ← M[ACC + R1]

OP code; load to accumulator

Assume that table and find different effective addresses and content of accumulator for diff. type of addressing modes

10) Auto increment/ decrement addressing.

This is similar to the register indirect addressing mode except the ~~th~~ register is. increment/ decrement after or before ~~th~~ its value is used to access the memory. When the address stored in the register refers to the table of data into the memory. So, it is necessary to increment/ decrement the register after access to the table.

★ Instruction types.

1) Data movement.
2) Arithmetic or/and logical
3) Flow control
4) Input output

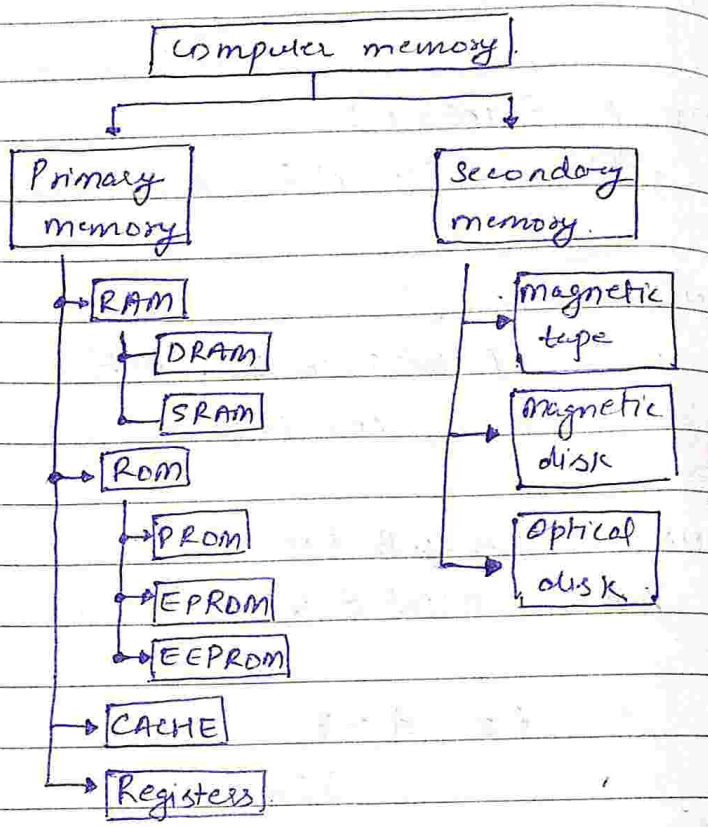
Assgn write ~~short~~ note on

Q. ~~what is~~ RISC and SISC processor. Find the diff. b/w these two

Page No.

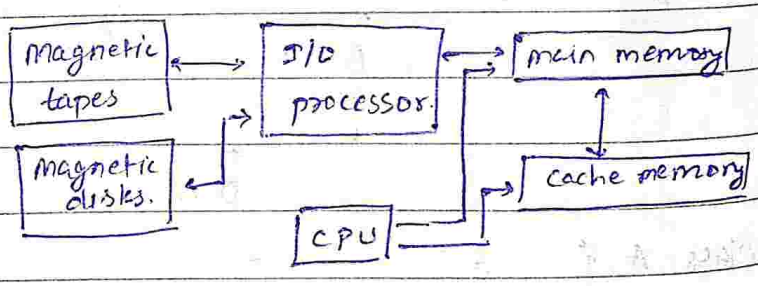
COA - 21-03-2023

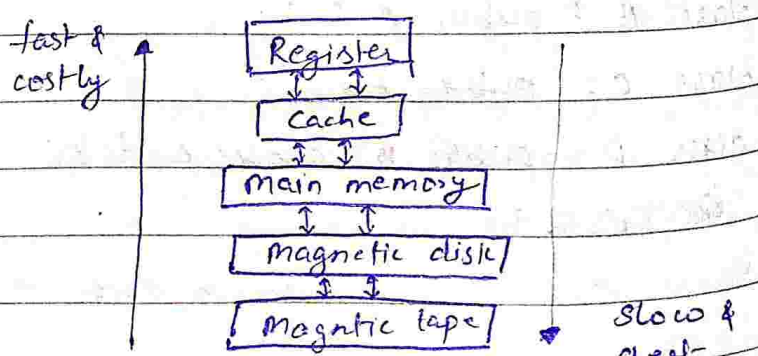
☆ Memory organisation.



Assign Q. what is diff. bet^n DRAM & SRAM?

☆ Memory Hierarchy.





Saathi

Q How many 128 bit RAM chip are needed to provide memory capacity of 2048 byte?

Sol $\frac{2048}{128} = \frac{2^{11}}{2^7} = 2^4 = 16$

Q How many lines of address are must be used to access 2048 byte of memory?

Sol

Q How many lines must be decoded for chip-select. Specify the size of the decoder.

Sol 4 lines must be decoded for chip select.
decoder size ~ 4 X 16

Q Your RAM will be 512 X 8 MB and you need the capacity of 512 gB.
Find out the ram size, No. of RAM

Sol

$512\ GB \Rightarrow \frac{512 \times 1024\ MB}{512 \times 8\ MM}$

$= \frac{2^{10}}{2^3} = 2^7 = 128$ RAM chips.

Total capacity = 512 GB = 512X1024

$= 2^9 \times 2^{10} = 2^{19}$

Available size

Date __/__/____

06/04/2023.

Saathi

Q Find the total memory capacity and determine the no. of address lines. I/O lines, that can be stored in specific memory.

1) 64 K × 8 bits.

I/O → 8 lines

address line → $2^6 \times 2^{10} = 2^{16}$

$\doteq 16$.

memory capacity = 64 K

2) 16 M × 32

I/O lines → 32

address lines → $2^4 \times 2^{10} \times 2^{10} = 2^{24}$

$= 24$

Memory capacity = 16×2^{24}

$= 16 \times 4 = 64$ M.

3) 4 GB × 64

I/O lines → 64

address lines = $4 \times 2^{10} \times 2^{10} \times 2^{10}$

$= 2^2 \times 2^{10} \times 2^{10} \times 2^{10}$

$= 32$

memory capacity = 4×2^{30}

$= 24$ GB

☆ Memory Interleaving

☆ Cache-Mapping functions.

→ Direct Mapping

→ Associated Mapping Cache

→ Block-set associated mapping

☆ Replacement Algorithm.

→ LRU replacement policy

→ FIFO replacement policy

→ Random replacement policy

Q Consider the following reference string of pages made by the processor

7 0 1 2 0 3 0 4 2 3

0 3 1 2 0

Frame size is 3

P_1	7	7	[7]	2	2	2	[2]	4	4	[4]	0	0	0	0	0
P_2		0	0	0	[0]	3	3	[3]	2	2	2	[2]	1	1	1
P_3			1	1	1	[1]	0	0	[0]	3	3	3	[3]	2	2
	x	x	x	x	H	x	x	x	x	x	x	H	x	x	H

No. of hit = 3

Page No.

Date ___/___/_____

Saathi

Q Reference string

7 0 1 2 0 3 0 4 2 3 0 3 1 2 0

P_1	7	7	[7]	2	2	2
P_2		0	0	0	0	0
P_3			1	1	[1]	3
					H	

4, 7, 5, 7, 6, 7, 10, 4, 8, 5, 8, 6, 8, 11, 4, 9, 5, 9, 6, 9, 12, 4, 7, 5, 7

P_1	4	4	4	[4]	6	6	6
P_2		7	7	7	7	7	7
P_3			5	5	5	[5]	10
P_4				7	7	7	7
				H		H	

ye questions khud solve kar lena...

Page No. ____