Name-Ashutosh soni

Id-2018UCP1505.

Que1:

Ans →.

In this question we check for. eligibility &. availaibility ~~of~~ required for the Instruction. to be executed.

Here bounds of loop are constants, therefore compiler will do the loop unrolling (If compiler won't then prefetcher will do) to increase the. instruction level parallelism.

clockcycle →	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
I1	S1	S1	S2	S3	S4.																		
I2			~~S1~~ S1	~~S2~~ S2	S2	S2	S3	S3	S4	S4													
I3				S1	S1	–	S2	–	S3	–	S4	S4	S4										
I4						S1	–	S2	S2	S3	S3	–	–	S4	S4								
I1							S1	S1	–	S2	–	S3	–	–	–	S4							
I2									S1	–	S2	S2	S2	S3	S3	–	S4	S4					
I3										S1	S1	–	–	S2	–	S3	–	–	S4	S4	S4		
I4												S1	S1	–	S2	S2	S3	S3	–	–	–	S4	S4

Here we see that from table that total 23 clock cycles are required for given four Instruction I1, I2, I3, & I4 to execute. tottally.

So the Ans is 23. for the above Instruction to run.

Name-Ashutosh Soni
Id-2018UCP1505

Que2:
Ans →.
There are three types of Hazards that can occur in a pipeline.

1) Data Hazard.
2) Structural Hazard.
3) Control Hazard.

Data Hazard:
This hazard because of data read & write from memory or from registers. This is further classified into three parts.

1). RAW (Read after Write) Data Hazard.

let
I1: R3 ← R1+R2.
I2: R5 ← R3+R4.

I1	IF	ID	OF	EX.	WB.	
I2.		IF	ID	OF	EX	WB.

Here when we use pipeline in Instruction 2. raw value of R4 is taken as it execute before write back.

2). WAR (Write after Read) Data Hazard.
of this type of Hazard takes place when we write data before reading. this type of Hazard chances are less until 4 stage pipelining.

3). WAW (Write after write) Data Hazard.
this type of Hazard takes place when we are writing at same memory or Register in both Instruction. chances of this hazard is also less until four stage pipelining.

Name-Ashutosh soni
Id-2018UCP1505

Structural Hazard:

This Hazard is cause because of conditional statements in the instruction.

```
I1 :   JUMPC 2020.
I2:    INC A
I3;    INC C.
```

In this type of sequence I2&I3 instruction are executed sequency & when. condition becomes true it go to 2020 but before that some instruction gets loadea & that time waste.

Control Hazard:

This type of Hazard cause because of limited resources in CPU.

```
I1      IF   ID   OF    EX   WB
I2           IF   ID    OF   EX   WB.
I3                IF    ID   OF   EX   WB
```

In I1& I3 both are calling memory at same time if no. of resources or buses are less as we know they are limited so in this case control Hazard comes in pictures.

Name-Ashutosh soni

Id-2018UCP1505

Que 4:

Ans →.

(a) mov ax, bx.

T1: MAR ← PC

T2: MDR. ← memory (instruction)

T3: IR ← MDR
PC ← PC+ 1

T4: ax ← bx

(b) MOV ax, 10h.

T1: MAR ← PC

T2: MDR ← memory content (~~10h~~) instruction

T3: IR ← MDR
PC ← PC+1.

T4: Ax ← 10h (IR).

ax gets 10h from. Instruction Register.

(c). MOV ax, [bx].

T1: MAR ← PC.

T2: MDR ← memory content (instruction)

T3: IR ← MDR.
PC ← PC+ 1

T4: MAR ← bx.

T5: MDR ← [bx]

T6: AX ← MDR.

Name- Ashutosh soni

Id-2018UCP1505.

(d) MOV ax, [4000H]

T1 : MAR ← PC.

T2 : MDR ← memory content (instruction).

T3 : IR ← MDR.
PC ← PC+1

T4 : MAR ← IR [4000H].

T5 : MDR ← [4000H]

T6 : ax ← MDR

Qu5:

Ans ⇒

DMA. controller transfer 32bits (4 byte).

input device transfer data =
9600 bytes per second.

CPU fetching & executing instruction rate
= 2,000,000 ins per second.

1 instruction
$= \frac{1}{2,000,000}$

CPU slow down rate =?

input device transfer rate = 9600 bytes per second

1 byte $= \frac{1}{9600}$ second.

4 byte transfer time = $4 * \frac{1}{9600}$ second.

Name-Ashutosh.Soni

Id-2018UCP1505

CPU slow down time =

$$\frac{1/2{,}000{,}000}{4 * 1/9600} \times 100$$

$$= \frac{9600}{4 * 2{,}000{,}000} = 0.0012$$

in % = 0.0012 X 100

= 0.12%

So, there will be 0.12% CPU slow down due to DMA activity.

Que 6:-

Ans:-

(a) Direct mapped Cache.

cache memory size = 16 KB.

Block size = 256 bytes.

Main memory size = 128 KB.

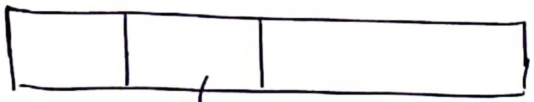
Lets assume, memory is Byte addressible.

$128\text{ KB} = 2^7 \times 2^{10}$ Bytes

$= 2^{17}$ Bytes

$\boxed{\text{NO. of bits in Physical Address} = 17}$.

Name- Ashutosh Soni
Id- 2018UCP1505



Block size = 256 Bytes
= 2^8 Bytes.

$\boxed{\text{No. of Bits in block. offset = 8.}}$

No. of Bits in line Number =
Cache size / line size
$= \frac{16\,KB}{256\,Bytes} = \frac{~~2^{14}~~\ 2^{14}}{2^8} = 2^6$ lines

$\boxed{\text{No. of Bits in line NO = 6}}$.

No. of bits in tag =
Total bits - line No Bits - offset (Block) Bits.

= 17 - (6 + 8)
= 17 - 14 = $\underline{3\text{ Bits}}$.

$\boxed{\text{No of bits in tag = 3}}$

Name-Ashutosh soni
Id- 2018UCP1505

(ii) Tag directory size
= No. of tags. X tag size.
= No. of line in cache x No. of Bits in tag
$= 2^6 \times 3.$
$= 64 \times 3 = 192$ Bits.

$\boxed{\text{size of tag directory} = 192 \text{ bits}}$

(b) fully Associative mapped cache.

Total bits we found = 17.

and no. of bits in block offset = 8.

(i) No. of bits =
Total bits - No. of bits in block offset.
$= 17 - 8 = 9$ bits.

No. of lines = cache size / line size.

$= \frac{16 \cancel{x} KB}{256 \text{ Bytes}} = \frac{2^{14}}{2^8} = 2^6$ lines

(ii) Tag directory size.
$= 2^6 \times 9$
$= 64 \times 9.$
$= 576$ Bits.

$\boxed{\text{size of tag directory} = 576 \text{ bits}}$

Scanned by CamScanner

Name - Ashutosh Soni
Id - 2018UCP1505

Que 7:
Ans -),

RISC	CISK.
1) Fixed size Instruction ~~as size is~~	1) Multiple type Instruction so size may vary here of Instruction.
2). Here code size is large as it doesnot support complex Instructions	2) Here code size is small as it supports complex Instructions.
3) Here a Instruction is executed in single clock pulse	3) Here multiple clock pulse may required as. depending on Instruction Complexity.
4). It focuses on Software	4) It focuses on Hardware
5). It requires more. number of Registers.	5). Requires less number of registers.
6) Used only Hardwired control unit	6) uses both Hardwired & micro programmed. control unit.

Scanned by CamScanner

Name-Ashutosh soni

Id-2018UCP1505

Que 8

Ans :→

Given,

Im. CON.

main memory access time = 100ns.

cache is 10 times faster than main memory.

Hit ratio for read request = 0.92.

85% Read request

write Request = 100-85 = 15%.

Tavg = hit rate * time to access + (1-hit rate) * miss penality.

<u>for read request.</u>

Tavg = ~~[illegible]~~ (0.92 * 10 + 0.08 * 100)

~~[illegible]~~ = 9.2 + 8 = 17.2

let suppose hit ratio for write request same as hit ratio of read request.

<u>for write request</u>

Tavg = 0.92 * 110 + (0.08 * 100)

= 101.2 + 8 = 109.2.

if 0 then

T ~~avg~~ = 0 ~~[illegible]~~

Total time avg.

= (0.85) * 17.2 + (0.15) * 109.2

= ~~[illegible]~~ 31 ns. Ans

Scanned by CamScanner

Name- Ashutosh soni

Id-2018UCP1505

ques 3.

Ans 3 Restoring Division

$8 = (1000)_2$ $\quad$ $4 = (0100)_2$

$-8 = (1000)_2$ $\quad$ $-4 = (1100)_2$.

	Accumulator	Dividend	Divisor	Operation
1	0000	1000	0100	LS.
	0001.	000 –		sub m.
	1100			
	1101			
	0001	0000		Restore.
2	0010.	000 –		LS.
	1100			sub m
	1110			
	0010	0000		Restore,
3.	0100	000 –		LS,
	1100			sub m
	0000	0001		
4.	0000.	001 –.		LS.
	1100			sub – m
	1100			
	$\underbrace{0000}_{\text{Rem.}}$	$\underbrace{0010}_{\text{Quotient}}$		Restore.

Remainder $= (0000)_2 = 0$

Quotient $= (0010)_2 = 2$.

Name Ashutosh Soni
Id - 2018UCP1005

Que10
Ans →

For MIPS usually 32 bits wide
components of MIPC architecture.

- Memory.
- Other component of the data path
- Control unit.

Major components of data path

- Program counter (PC).
- Instruction Register (IC).
- Register file.
- Arithmetic logic unit (ALU).
- Memory.

Program counter: - A sequence of machine,
instruction in the text segments.

Register that stores the address of next instruction
to fetch &

- also known as instruction Pointer

In MIPS each instruction is 32 bits long.

So PC+4 as 3~~2bits~~ =

Instruction Register :-
Register that holds the instruction.
currently being decoded.

Name - Ashutosh soni

Id - 2018UCP1505

Arithmetic & Logical Unit (ALU).

→ · Implement binary arithmetic operations & logical operations

Input:

- operands - 2 X 3 2 bits
- operation : control signal

output:

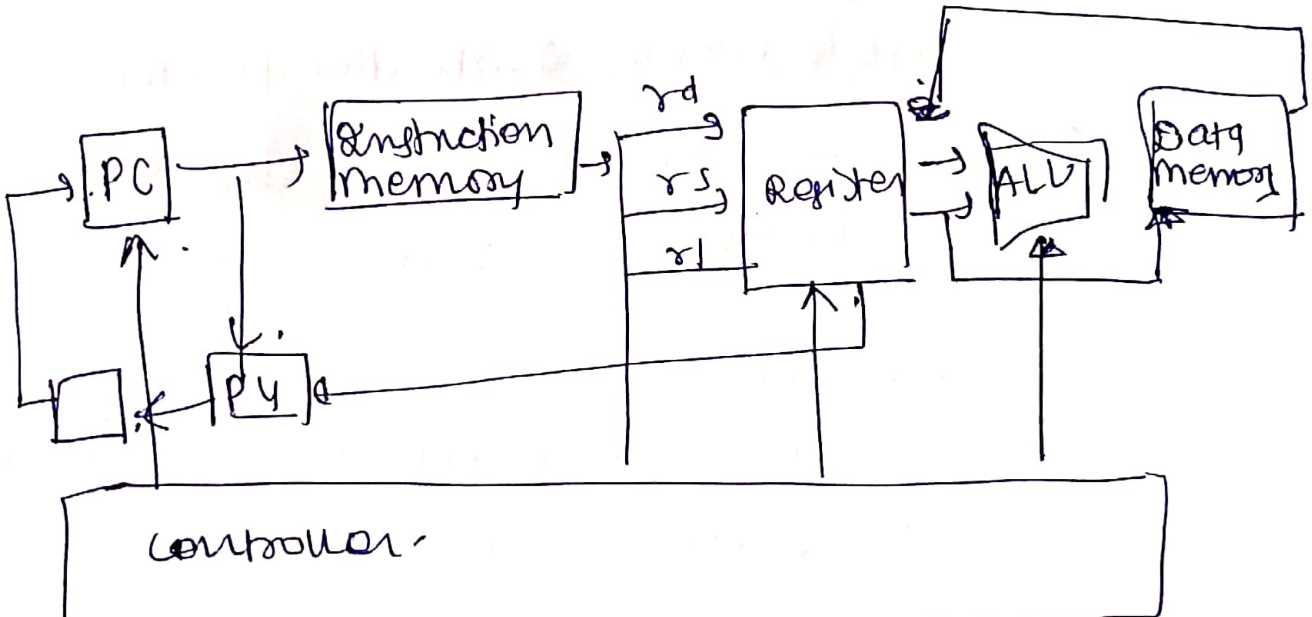
result : 1 X 6 4 bits.

Status : Conditional signal.

Control unit :

controls component of datapath to implement FDXycle
(Fetch Decode.)

- Inputs → Conditional signals
- output : control signal.



Name-Ashutosh Soni
Id-2018UCP1505

Que 11:

(a) Micro Instruction format and Nano Programming.

Microinstruction:

~~Micro~~ one symbolic representation of bits pattern. They consists of 128 bits and those bits are broken down into 30 functional field, each of those field consists of one or more bits and groped in five mayor categories.

Nano Programming:

This is a microinstruction. is in primary control-store memory it then has the control signals generated for each microinstruction. using a secondary control store memory. The output word from the secondary memory. is called Nano-Instruction.

(b) Interrupt: It is a signal from a device attached to computer or from program within the computer. that requires operating system to figure out what to do next. Devices and programs occasionally needs CPU services but we want predict when so the interaction with CPU each device or program is allowed to give Interrupt.

The interrupt are of two types:

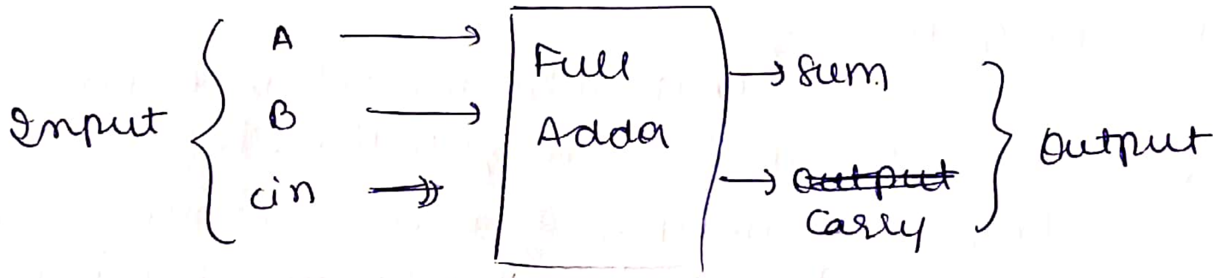
1) Hardware Interrupt: If a signal for the processor is from external device or hardware it is called Hardware Interrupt.

Name - Ashutosh Soni
Id - 2018UCP1505

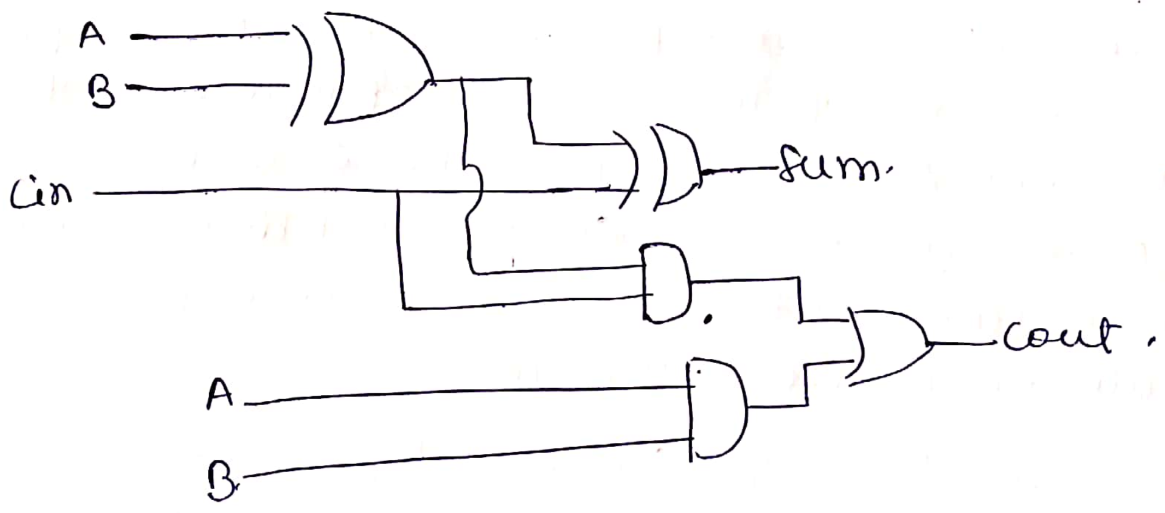
2) ~~Software~~ Interrupt : when a interrupt caused by a special instruction in the instruction set or by exception condition in processor itself, then it is called software Interrupt.

c) Full Adder.

full adder is the adder which adds the three input and produces the output sum and carry.



$$Sum = cin \oplus (A \oplus B)$$
$$Carry = AB + cin (A \oplus B)$$



Scanned by CamScanner

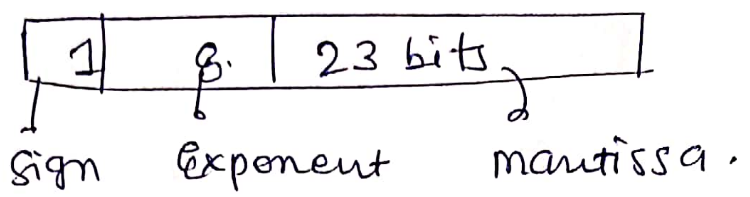
Name - Ashutosh Soni
Id - 2018UCP1505

(d) Floatingpoint Number Representation

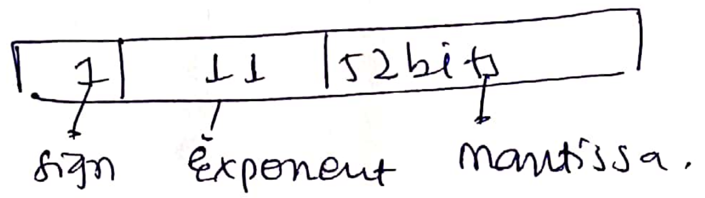
$\pm$ significant x Base$^{\pm \text{ exponent}}$.

Two Representation Techniques

① Single precision (32 bits)



② Double precision (64 bits)



Normalization

1.__ x Base$^{\pm \text{ exponent}}$

No more than 1 digit before decimal.

Name-Ashutosh Soni
Id-2018UCP1505

Que 3.
Ans ⇒

a) Using State Table method.

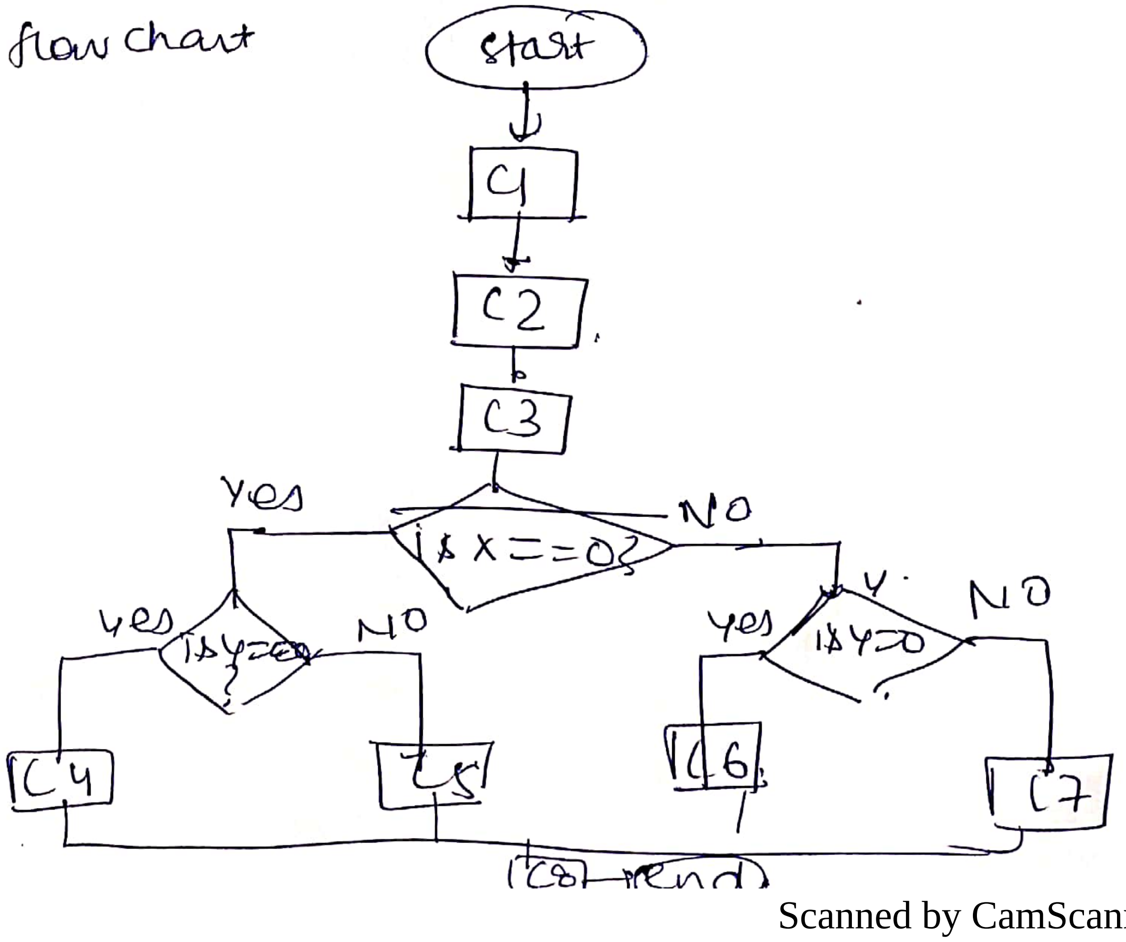
Instructions.

T-States.	compare	Jump	assign	compute
T_1	Z_{C_1}	$Z_{J,1}$	Z_{ai}	Z_{cp1}
T_2	Z_{C_2}	Z_{J2}	Z_{a2}	Z_{cp2}
⋮				
T_n	Z_{cn}	Z_j	Z_{ar}	Z_{cpn}

Zc represent the control signals generated in the state T_1 by instruction compare

b) Using Delay Element method

flow chart


start
C1
C2
C3
is x==0?
yes
NO
is y==0?
yes
NO
is y==0
yes
NO
C4
C5
C6
C7
C8 end