B-TECH 4TH SEMESTER
END-TERM EXAMINATION

MAY 2021

✓✓✓✓✓
1, 2, 3, 4, 5
? ? ?

Subject: Computer Architecture and Organization - I
[CS 2202]

Date: 21st May 2021
Name: Abhiroop Mukherjee
Exam Roll No.: 510519109
G-Suite ID: 510519109.abhirup@students.iiests.ac.in
No. of Sheets Uploaded: 8

Q.1) m-addressed m/c → m-addressed m/c works on m explicit operands, if total no. of operands are n, then n-m ~~operands~~ operands are implicit, i.e.
→ Instruction only has m operands.

given Z = W + X - Y

I) 3 addressed

ADD	Z, W, X	[Z ← W+X]
SUB	Z, Z, Y	[Z ← Z-Y]

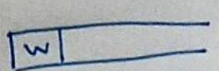
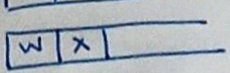
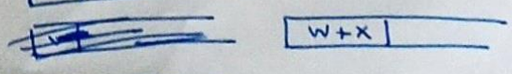
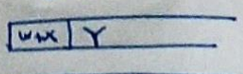
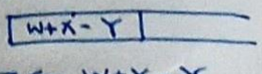
II) 2 addressed

ASS	Z, W	[Z ← W]
ADD	Z, X	[Z ← Z+X]
SUB	Z, Y	[Z ← Z-Y]

III) 1 addressed.

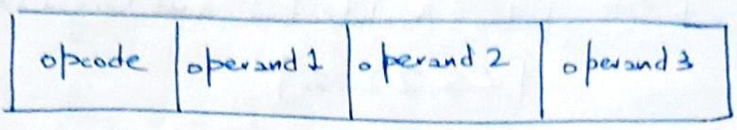
LOAD	W	[AC ← W]
ADD	X	[AC ← AC+X]
SUB	Y	[AC ← AC-Y]
STORE	Z	[Z ← AC]

IV) 0 addressed.

PUSH	W	
PUSH	X	
ADD		
PUSH	Y	
SUB		
POP	Z	Z ← W+X-Y

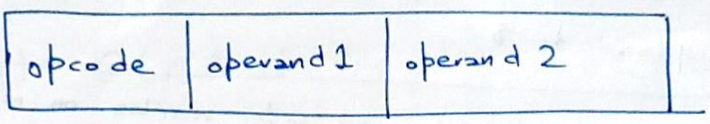
→ given op code 1 byte, an operand 2 byte

I) 3-addressed



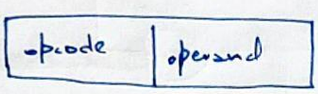
* $L_{inst} = 1 + 2 \times 3 = 7$ bytes

II) 2-addressed



$L_{inst} = 1 + 2 \times 2 = 5$ bytes

III) 1-addressed



$L_{inst} = 1 + 2 = 3$ bytes

IV) 0-addressed

→ when PUSH/POP ⇒ requires operand → 3 bytes

→ else 1 byte [only opcode]

∴ As m decreses, L_{inst} deceses, but at the same time, no. of instructio
to complete task increses

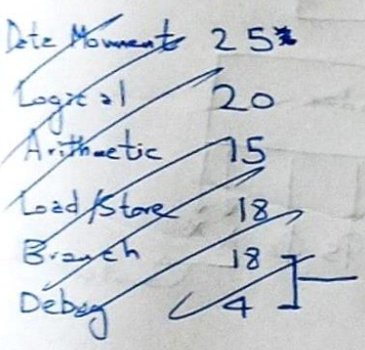
Frequency Dependant Opcode

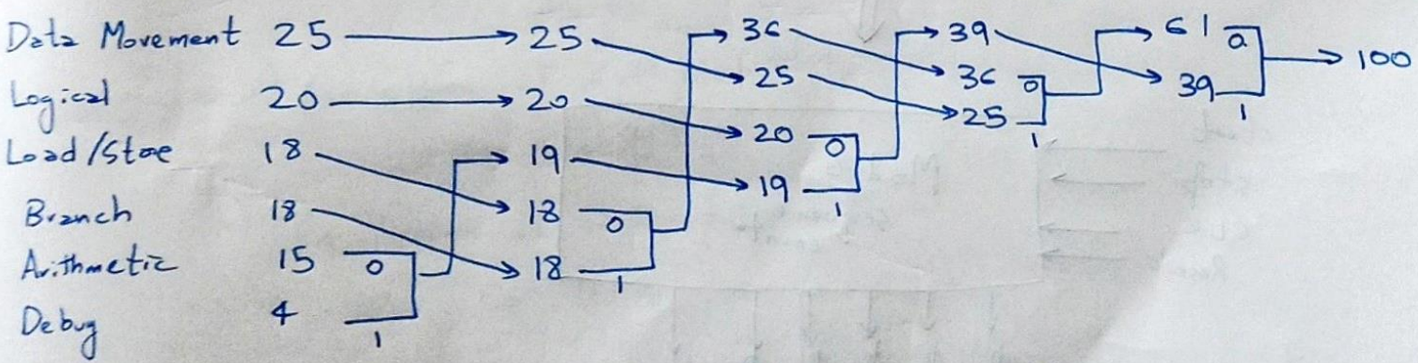
→ It has been observed that some instructions are used more frequently than others

→ If we could, by any ways, reduce the no. of words to represent frequently used opcode, we can reduce average word length of program [hence reducing size of program]

→ We do this with the help of Hoffman Encoding, most frequently used instructions are encoded with fewer number of bits.

→ given frequencies → Data Movement 25%.
→ Logical 20%.
→ Arithmetic 15%.
→ Load/Store 18%.
→ Branch 18%.
→ Debug 4%.





So, using this heap, we get :

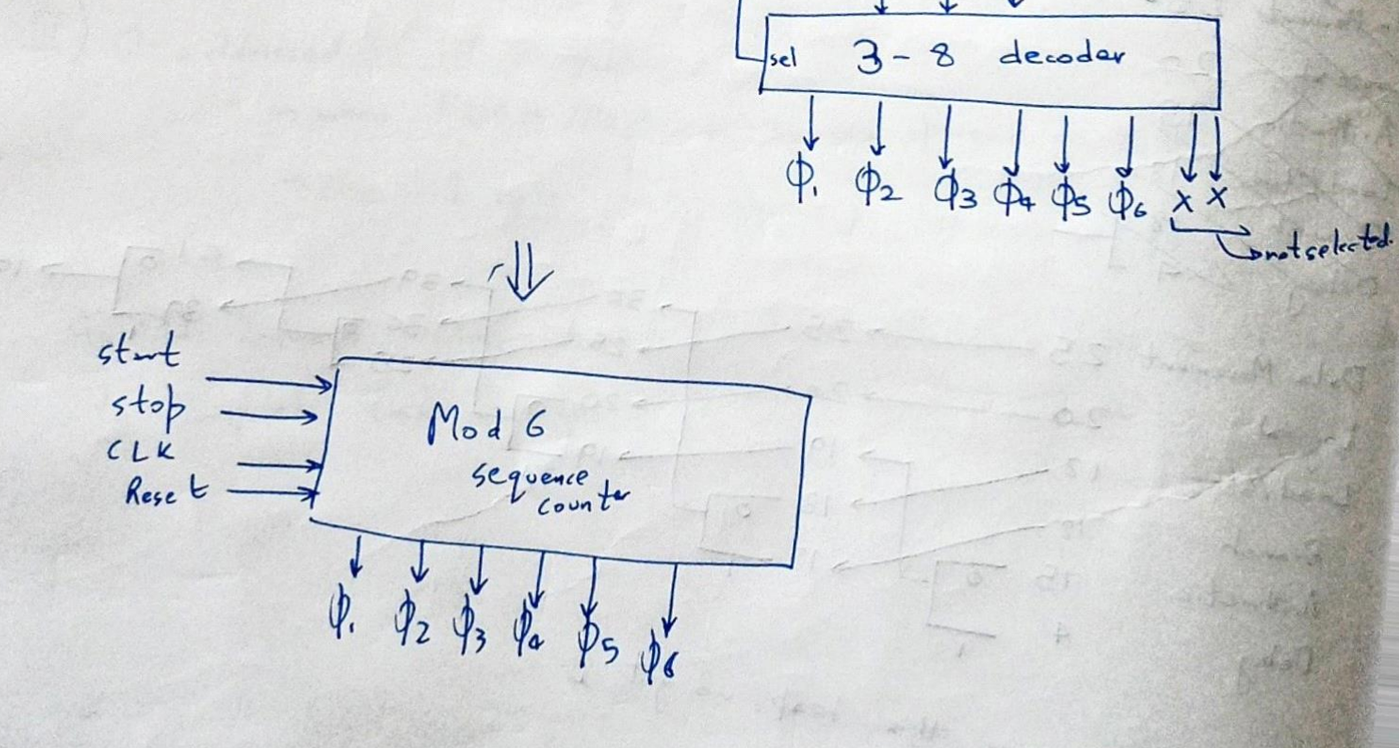
Data Movement : 0 1
Logical : 1 0
Load/Store : 0 0 0
Branch : 0 0 1
Arithmetic : 1 1 0
Debug : 1 1 1

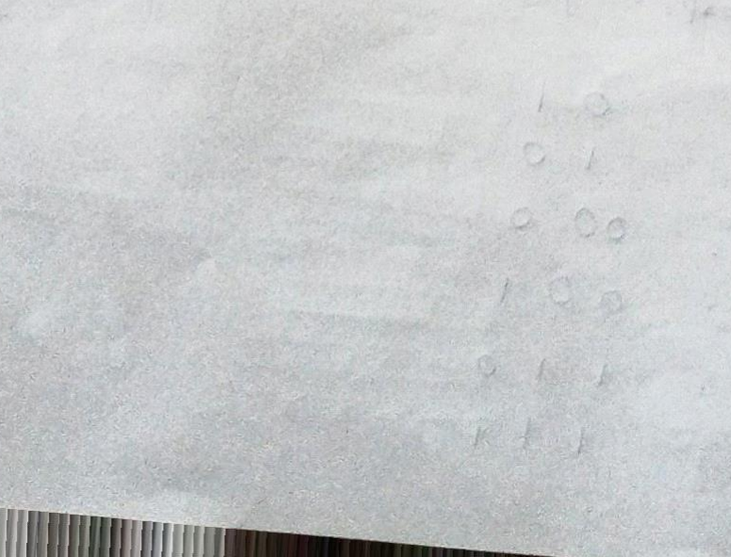
3) given figure, have to make CU using sequence

→ From the figure, we see that every instruction i_1, i_2, i_3, i_4 t
6 steps to complete, so we can use a mod 6 coun
to drive the CU

→ given each control signal takes time t, Let us define six phases $\phi_1, \phi_2, \phi_3, \phi_4, \phi_5, \phi_6$, each separated by t
t, with assumption that a control signal will ~~executed~~
start get executed at the start of a phase

→ Figure of Mod 6 Sequence Counter





- now from the figure, we can make following observations

i) C_0 will happen at the start of every execution cycle

$$\therefore C_0 = \phi_1$$

ii) C_1 happens at second step of execution cycle, and also ~~when~~ in fifth instruction for i_1, i_3 and i_4

$$\therefore C_1 = \phi_2 + \phi_5 [i_1 + i_3 + i_4]$$

iii) C_2 happens at third step

$$C_2 = \phi_3$$

iv) C_3 happens at third step,

$$C_3 = \phi_3$$

~~v~~) v) C_4 happens at fourth step

$$C_4 = \phi_4$$

vi) C_5 happens at sixth step of i_1

$$C_5 = \phi_6 i_1$$

vii) C_6 happens at fifth step of i_2

$$\therefore C_6 = \phi_5 i_2$$

viii) C_7 happens at sixth step of i_2

$$C_7 = \phi_6 i_2$$

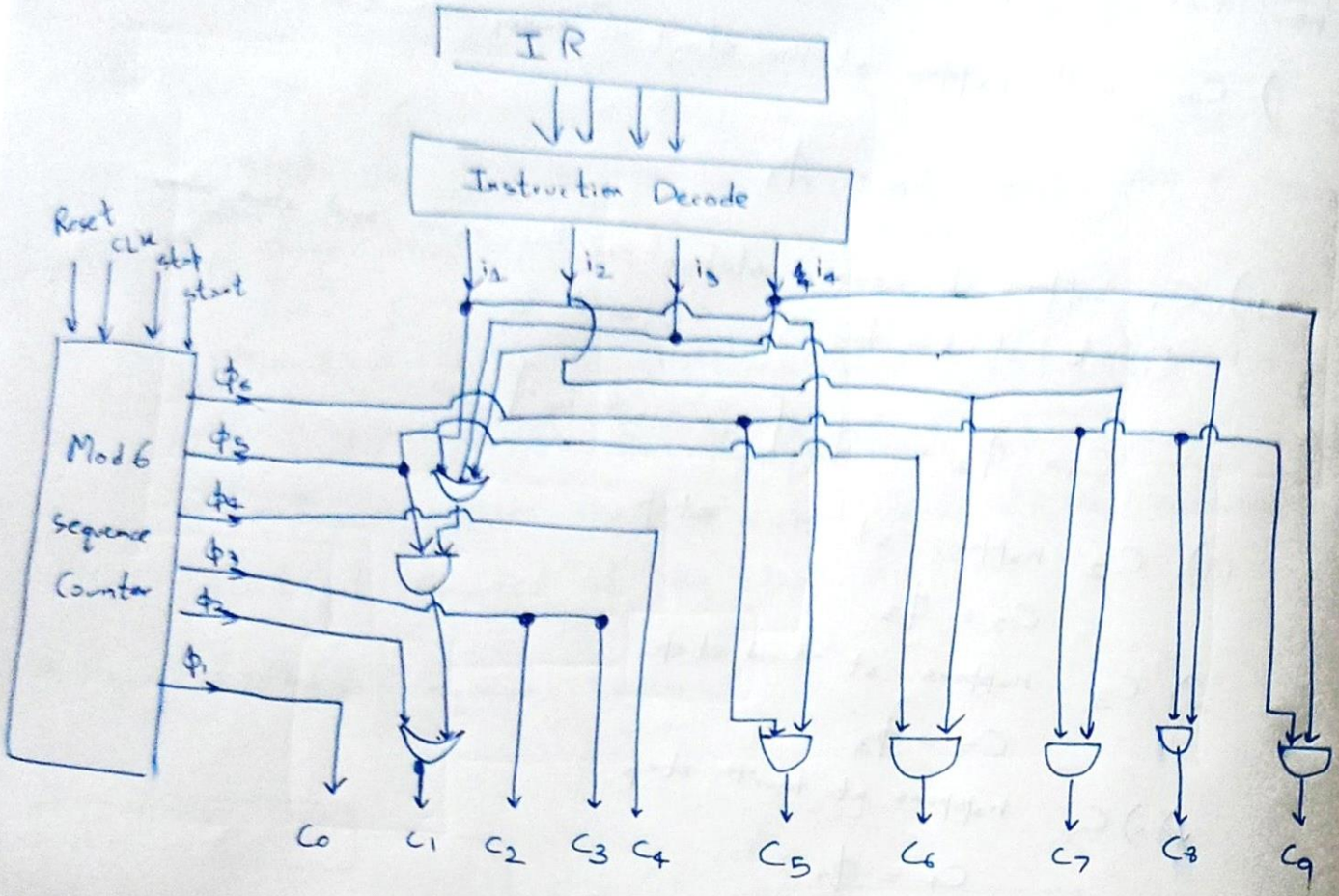
~~ix~~ ix) C_8 happens at sixth step of i_3

~~C_8~~ $C_8 = \phi_6 i_3$

x) C_9 happens at sixth step of i_4

$$C_9 = \phi_6 i_4$$

→ Using these observation, we can make the following CU



Q4) 2) given the following

i) RAM refreshes 32 times per ms

ii) Each refresh 100 ns

iii) Memory Cycle requires 200 ns

To find % of memory's total operating time dedicated to refreshes.

→ Time used up in Refresh ⇒ ~~32 x 100 ns~~ ⇒ ~~32 x 1 ms x 100 ns~~

= ~~32~~

[nano = 10^{-9}, ~~milli~~ = 10^{-3}]

→ Refreshes count = 32×10^{-6} times per ns

→ Refresh time = 100 ns

Time taken for ~~ef~~ refres = $32 \times 10^{-6} \times 100$ ns per ns

= 32×10^{-4} ns in one ns

→ ~~32×10^{-4}~~ x 200 ns in 200 ns

= 0.64 ns in 200 ns

→ So total time = 200 + 0.64 ns

∴ % of time dedicated to refreshes = $\frac{0.64}{200+0.64} \times 100\ \%$

$= 0.318\ \%$

→ given 32 times per ms refresh

no. of memory cycles per 1 ms = $\frac{1\text{ms}}{200\text{ns}} = \frac{10^6\text{ns}}{200\text{ns}} = 5000$ times

Time taken for refresh = 32 × 1000 ns

operation time taken for refresh = $\frac{32 \times 1000\text{ns}}{5000 \times 250\text{ns}}$

$= \frac{3200\text{ns}}{1.25 \times 10^6\text{ns}}$

$= 2.56 \times 10^{-3}$

% operation time = $2.56 \times 10^{-3} \times 100\ \%$

$= 0.256\ \%$

Q4) b) $0000_h = 0$ (dec)

$0FFF_h = 2^{12} - 1$ (dec)] → difference → 2^{12} → 4K

$8000_h = 2^{15}$ (dec)

$9FFF_h = [2^{15} + 2^{13}] - 1$ (dec)] → difference → 2^{13} → 8 K

given 16 bit address bus

Address of	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	------	A_0	hex	Chip
LS loc	0	0	0	0	0	0	---	0	0000h	4K byte
MS loc	0	0	0	1	1	1		1	3FFF$_h$	
LS loc	1	0	0	0	0	0		0	8000 h	8K byte
MS loc	1	0	0	1	1	1		1	9FFF$_h$	

so, it's interfacing will be as follows