Homework 3

Q1)

a) 7200 rpm

1 min = 7200 rotations

1 sec = $\frac{7200}{60}$ = 120 rotations

1 rotation data access = 32 × 16 KB

Peak transfer rate = 120 (rps) × 32 × 16 KB.

$60 \times 2 \times 2^5 \times 2^4$ KBps

60×2^{10} KBps (2^{10} KB = 1MB)

= 60 MBps → Ans

b) for reading 1 sector = Seek Time + Rotational latency + Transfer time

1 sector = 16 KB

(Peak) Transfer rate = 60 MBps

Time = $\frac{16 \text{ KB}}{60 \times 2^{10} \text{ KBps}}$ = $\frac{2^4}{60 \times 2^{10}}$ secs

= 0.00026 secs

= 0.26 ms

Total time = Seek time + Rotational latency + Transfer time = $3 + \frac{1}{2} \times \frac{1}{120} \times 10^3 + 0.26$ ms

= 7.42 ms → Ans

c) 4MB = 4×2^{10} = 4096 KB

3 sectors = 3 × 16 KB = 48 KB

No of sectors = $\frac{4096}{16}$ = 256 sectors

Data transfer for 256 sectors = $256 \times 0.26 = 66.56$ ms

No of contiguous sectors = 3

Contiguous Sector changes = $\frac{256}{3} = 85.33$ (or 86)

Time for contiguous sector changes = $86 \times$ (Seek time + Rotational latency)

$$= 86 \times \left(3\text{ ms} + \frac{1}{2} \times \frac{1}{120} \times 10^3 \text{ ms}\right)$$

$$= 86\,(3 + 4.16)$$

$$= 86 \times 7.16 = 616.33 \text{ ms}$$

Total = $615.78 + 66.56 = 682.38$ ms

↳ Ans

Q2)

a) No of instructions = $8 - 1 = 7$

bits = $\log_2 7 \approx 3$ bits

Biggest instruction — LUI/LLI

LUI rd, imm

3 bit | 3 bit | 8 bit

$3 + 3 + 8 = 14$ bits

Immediate : −128 to 127

= 256 numbers

bits = $\log_2 256 = 8$ bits

Registers = 8

bits = $\log_2 8 = 3$ bits

Q2)

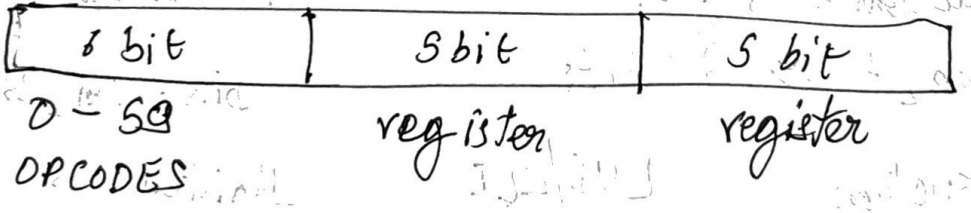
a) No of instruct^{n} = 60 + 30 + 3 + 26 = 119

bits = $\log_2 119 \approx 7$ bits

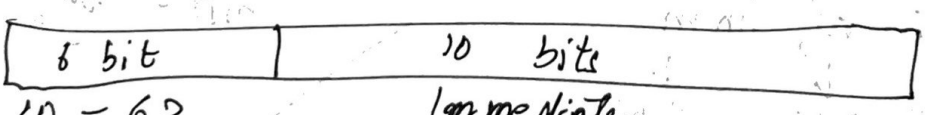
But 2 register instructions take 5 + 5 = 10 bits (Since there are 32 registers so we need $\log_2 32 = 5$ bits to access each)

Since machine is 16 bits & 2 register instruction take 10 bits so OPCODE can have max 6 bits. But we need 7 bits for OPCODE for 119 instructions. Hence, it is not possible without expanding OPCODES.

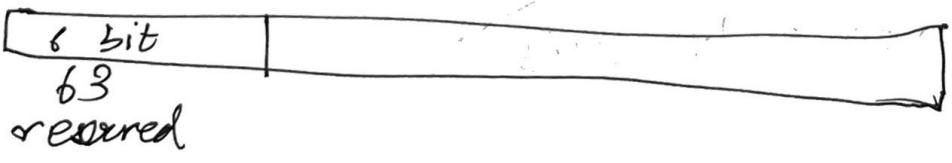
b) * 60 instructions with 10 bits for register & 6 OPCODE bits
* 3 instructions with 10 bits for immediate & 6 OPCODE bits
* 1 ~~for reser~~ instruct^{n} for reserved value



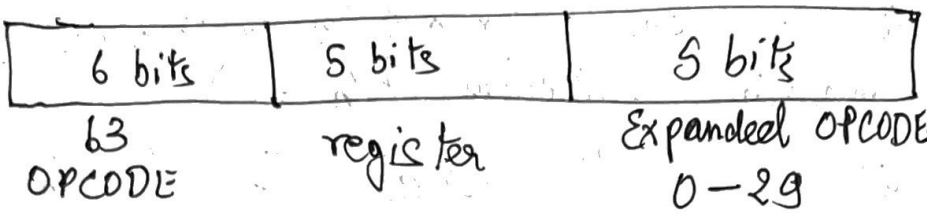
Type A instructions



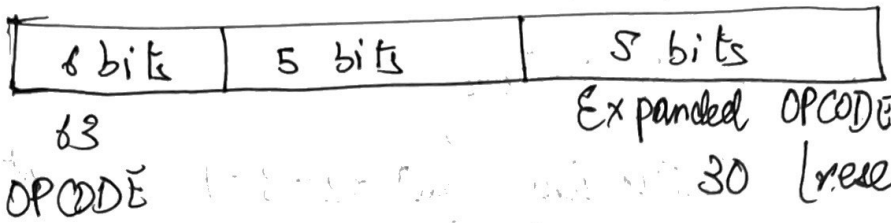
Type C instructions



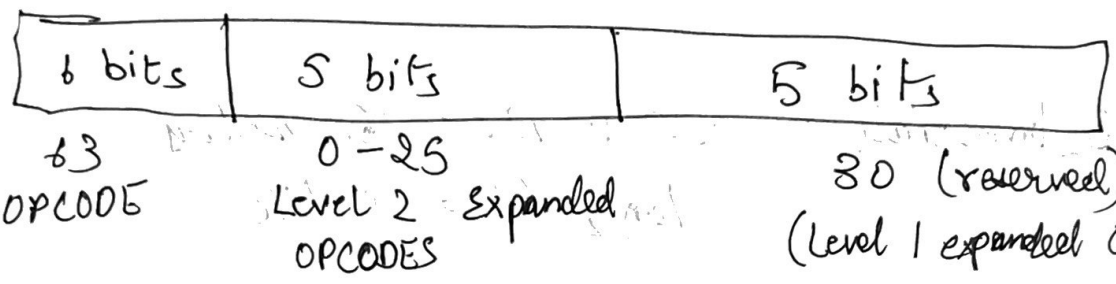
* 30 instructions with 1 register (on the reserved value 63)
* 1 instruction reserved again (for no operand instructions)
* 1 value unused



(Type B instructions)



*) 26 instructions with no operand (on the reserved value of 63 on OPCODE & 30 on level 1 expanded OPCODE)



c) No operand instructions have 10 bits free & level 1 expanded OPCODE has 1 in free space (for value 31) & level 2 expanded OPCODE has 6 free values (for values 26-31 on the level reserved value of 30)

∴ Total no of instructions possible

$1 \times 2^5 + 6 \times 2^0$

$= 32 + 6 =$ 38 instructions → Ans

d) Since there is 1 instruction with 3 registers (15 bits) so only 1 bit left for OPCODE. If we use value 0 for that instruction & value 1 for reserved value then we

have 5 bits free for 2 register instructions (total = 60) & 10-bit immediate instructions (total = 3)

∴ We will need to reduce Type A instructions or Type C instructions as we have only 5 bits (32 values)

Reducing Type A instructions

Type A – 28 instructions (32 − 3 − 1 = 28 out of 60) → Ans
Type B – 30 instructions
Type C – 3 instructions
Type D – 26 instructions

Q3) a) Immediate : −128 to 127 = 256 numbers

$\log_2 256 = 8$ bits

Registers : 8

$\log_2 8 = 3$ bits

Instructions : 8 − 1 = 7

$\log_2 7 \approx 3$ bits

Total = 8 + 3 + 3 = 14 bits → Ans

b) Since we need as few bits as possible so it's a total of 14 bits

BEQ rs, rt, addr

3 bit 3bit 3 bit → 14 − (3 + 3 + 3) = 5 bits (32 values)

Since it can branch both backwards & forward so it can jump from −16 to 15.

Hence, branch ahead = 15 addresses → Answer

c) LUI & LLI store in 2 halves of register & each can store 256 numbers

Total bits = $\log_2 256 \times 256 = \log_2 256 + \log_2 256$

$= 8 + 8 = 16$ bits

↳ Ans

d) LW & SW use indirect addressing (meaning addresses are stored in registers which are accessed through values these registers)

Total no of addresses indexed = Size of register

$= 2^{16}$ addresses

↳ Ans

Q4)

a) 500 MHz

$= 500 \times 10^6$ instructions$^{\text{cycles}}$ per second

Trap = 1000 cycles

500×10^6 cycle → 1 sec

1000 cycle → $\frac{1000}{500 \times 10^6}$

$= \frac{1}{500 \times 10^3}$

$= \frac{2}{1000 \times 10^3} = 0.002$ ms

$= 0.0002$ ms or 2 Ms

↳ Ans

$\simeq 0.002$ ms or 2 Ms

Time spent trapping for disk access

$$= \frac{0.002}{8+0.002} \times 100$$

$$= 0.024\%$$

↳ Ans

Time spent trapping for network access

$$\frac{0.002}{4+0.002} \times 100$$

$$= 0.049\%$$

↳ Ans

b) Network access time = 4ms − $\frac{40}{100} \times 4$

$$= 4 - 1.6$$

$$= 2.4 \text{ ms}$$

Trap = 0.002 ms Total = 2.4 + 0.002 = 2.402 ms

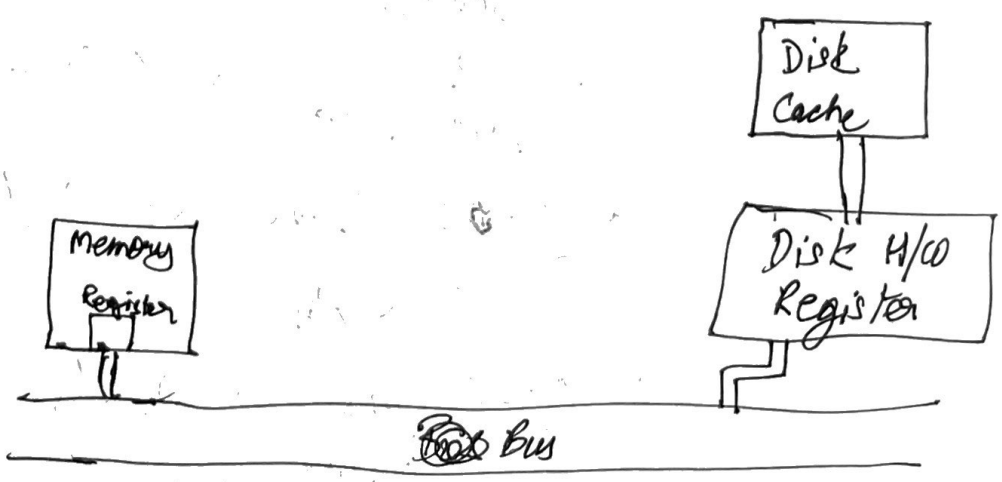
Percentage reduction

$$= \frac{4.002 - 2.402}{4.002} \times 100$$

$$= \frac{1.6}{4.002} \times 100 = 39.98\%$$

↳ Ans

Q5)



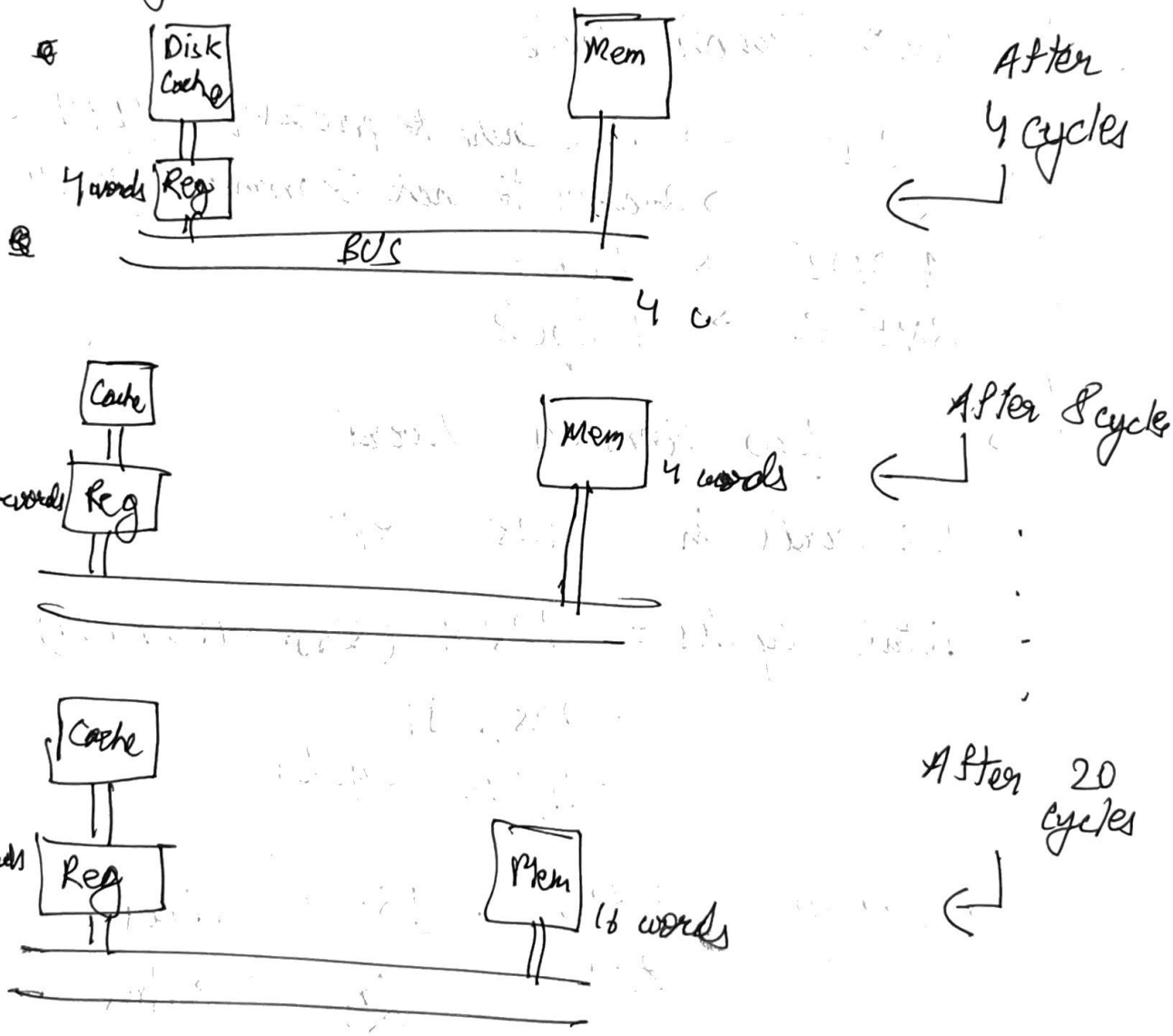
Disk cache & Disk H/w register can be pipelined.

Initially 4 cycles (for 4 words) to go to H/w registers.

Next 4 cycles ⇒ 4 cycles to transfer 4 words on the bus

(By this time, 4 words can come into the H/w register from disk cache. This step is 'PIPELINED')

Total cycles to transfer 16 words (Diagramatically)



Next 80 cycles will be for memory write. During this time, bus cannot be stalled and 4 words are stored in the H/w register of disk. Hence, next 16 words takes exactly 16 cycles because there is no

need for initial 4 cycles
Total words = 128 (512 bytes)
Total access times = $\frac{128}{16}$ = 8
of memory

Of 8, 1st takes 20 + 80 = 100 cycles
Next 7, 7 × (16 + 80) = 7 × 96 = 672 cycles
Total = 672 + 100 = 772 cycles
↳ Ans

Q6) Write through cache

LW → Disk cache to processor (4+1 cycles)
SW → Processor to cache to memory (80+4 cycles)
ADDI → 1 cycle
BNE 2 → 1 cycle

↳ 1 loop transfers 1 word

128 words in 128 loops

Total cycles = 128 × (84+3+1+1)
= 128 × 91
= 11,648 cycles

Speedup = $\frac{11648}{772}$ = 15.08 times
↳ Ans