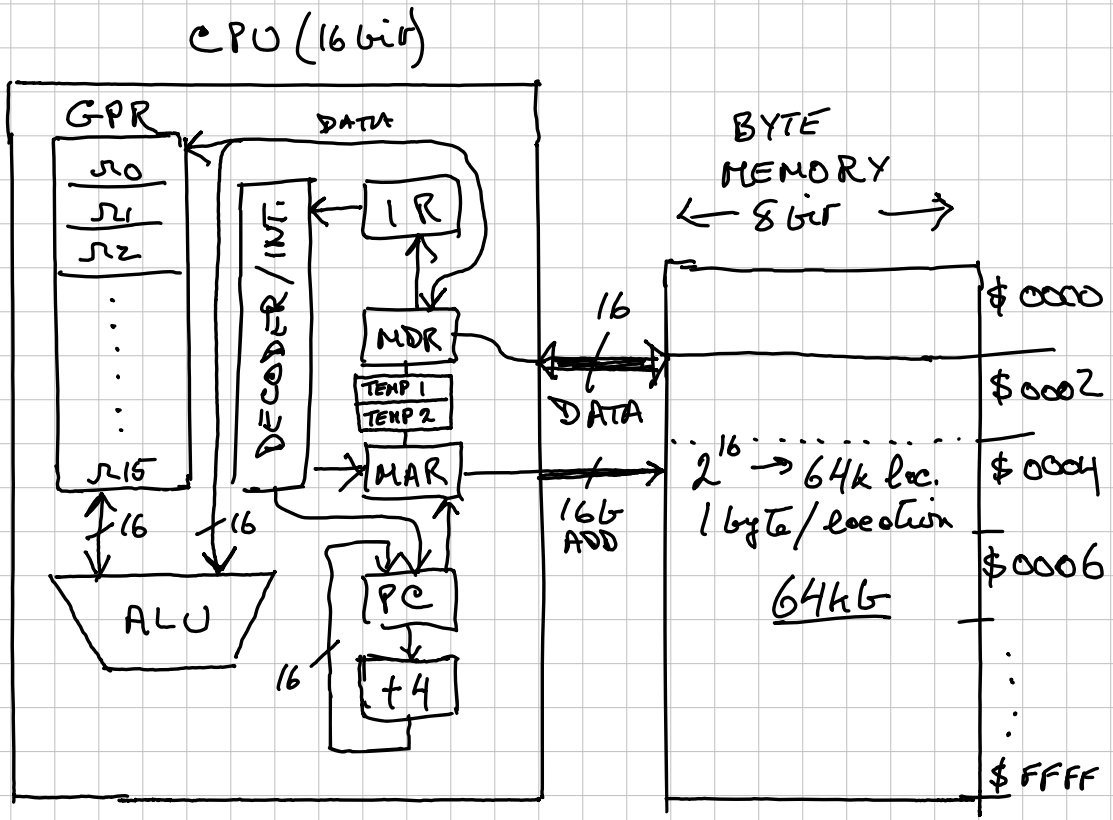
David CRONIN 28840024
COEN 311 - Assig 2



Registers TEMP 1 and TEMP 2 are for complex instructions

David CRONIN 28840024
COEN 311 - Assig2

Q2:

$$x = ((a+G)/(c+d)) * (100 * a)$$

reg: r_0 r_1 r_2 r_3 r_4 r_0

r_5 = r_0, r_1 ; r_6 = r_2, r_3 ; r_7 = r_4, r_0 ; r_8 = r_5, r_6 ; r_9 = r_8, r_7

$R_0 \leftarrow a$	LOAD R_0, M[a]
$R_1 \leftarrow G$	LOAD R_1, M[G]
$R_2 \leftarrow c$	LOAD R_2, M[c]
$R_3 \leftarrow d$	LOAD R_3, M[d]
$R_4 \leftarrow \#100$	MOVE R_4, #100
$R_5 \leftarrow R_0 + R_1$	ADD R_5, R_0, R_1
$R_6 \leftarrow R_2 + R_3$	ADD R_6, R_2, R_3
$R_7 \leftarrow R_1 * R_4$	MUL R_7, R_1, R_4
$R_8 \leftarrow R_5 / R_6$	DIV R_8, R_5, R_6
$R_9 \leftarrow R_8 * R_7$	MUL R_9, R_8, R_7
RAM $\leftarrow R_9$	STORE M[x], R_9

Daniel CRONIN 28840024
COEN 311 - Assig 2

CONSTRAINTS

Q3) • 16 bit DATA bus • 16 bit ADD bus.
• 6 instructions • 2 TEMP registers

Consider LOAD Ri, M[X]

if X is a 16 bit address
and Ri is one of 16 requires 4 bits.
and LOAD is one of 6 requires 3 bits
minimum instruction length is 22 bits
but GPR is only 16 bits wide
Therefore complex instructions of 32 bits
are required:

1010XXXX . 0100XXXX $2004

OP code (8 bits) | Ri (8 bits) | ADDRESS (16 bits)

← 32 bit complex instruction →

David CRONIN 28840024
COEN 311 - Assig 2

Q4)

a) ADD R_5 R_1 R_2 → $ A512

This is a single 16 bit instruction

1010 0101 0001 0010 XXXX
A 5 1 2

16 bits 16 bits

b) MOVE R_4 #100 → $ B464 (#100 = $64)

0001 0100 0110 0100

simple 16 bit instruction limited to 0 → 255 #

complex 32 bit instruction #0 → 65.5K

David CRONIN 28840024
COEN 311 - Assig2

Q4

c) LOAD R_3, M[$2004] →

complex instruction requiring 32 bits

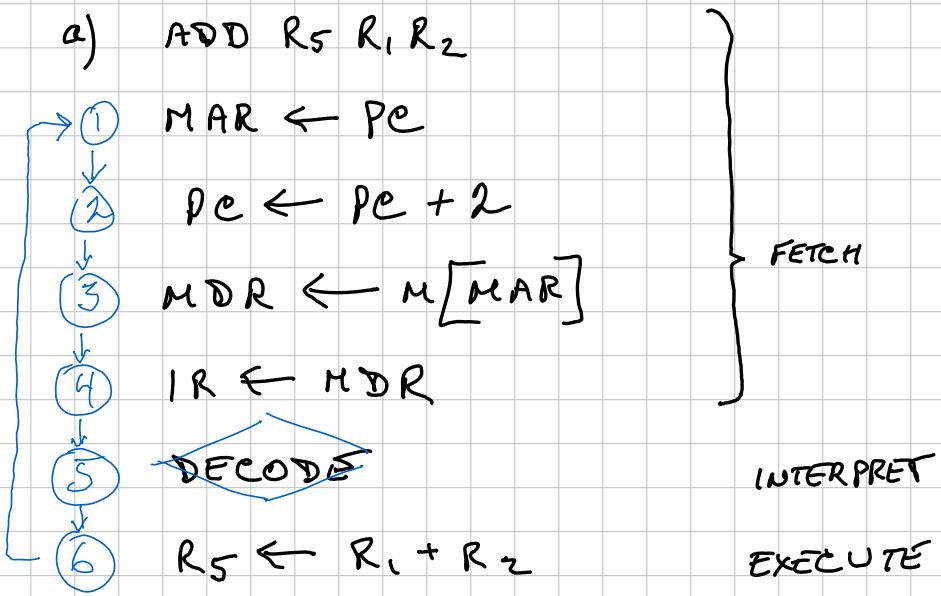
$ C X 3 X 2 0 0 4

1100 XXXX 0011 XXXX 0010 0000 0000 0100

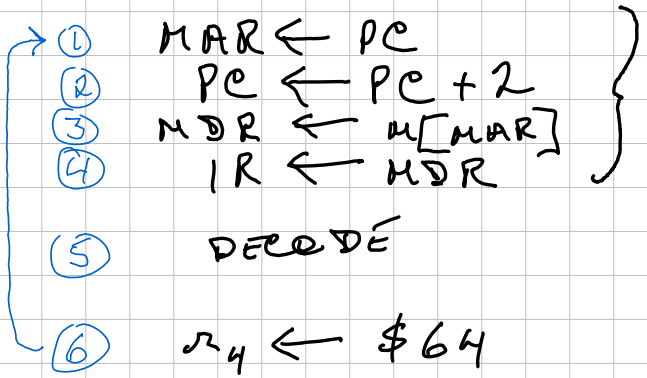
David CRONIN 28840024
COEN 311 - Assig2

Q5) executed as 16 bit instruction

a) ADD R_5 R_1 R_2



b) MOVE R_4 #100 → $B464



executed as 16 bit instruction

David CRONIN 28840024
COEN 311 - Assig 2

Q5 c) LOAD R3, M[$2004]

This is a complex 32 bit instruction

lower 16 bits are the memory address $2004

Fetch memory addr.

PC ← PC + 2

MAR ← PC

MDR ← M[MAR]

R3 ← MDR

Q6) ALL: DECODE 1cc FETCH 12cc INCR 2cc

a) ADD Ri, Rj, Rk 3cc → 18cc 9ns

b) MUL Ri, Rj, Rk 5cc → 20 10ns

c) DIV Ri, Rj Rk 5cc ↗

d) MOVE Ri, #100 2cc → 17 8.5ns

LOAD Ri, M[A] 11cc → 26 13ns

STORE M[A], Ri 11cc ↗

2GHz → .5ns.