Penedo B. Thibeaud 40212017

COEN 311 Section W (Computer Organization and Software)
Assignment 2
Due Friday October 6, 2023

Note: Please submit only one PDF file through Moodle.

In this assignment we want to design and measure the performance of a new processor (CPU) with 16 general purpose Registers (R0, …., R15), 16-bit data bus and 16-bit address bus.
Note: Memory is by default byte organized.

Question 1)
Draw the internal organization of this CPU by showing the widths of all registers as well as all external and internal buses. Also show the byte organized memory and its interface to the CPU.

Question 2)
Write an assembly program to evaluate the following expression for this CPU. Ignore the remainder part of the division process and assume the multiplication result is 16 bit instead of 32 bit.

```
x = ((a + b) / (a + c)) * (100 * a)
```

The instruction set of this processor consists of the following instructions:

```
ADD  Ri, Rj, Rk  ; add Rj to Rk and put result in Ri
MUL  Ri, Rj, Rk  ; multiply Rj by Rk and put result in Ri
DIV  Ri, Rj, Rk  ; divide Rj by Rk and put result in Ri
MOVE Ri, #100    ; move constant value #100 in Ri
LOAD Ri, M[X]    ; load (copy) content of Memory at address
                   X into Ri
STORE M[X], Ri   ; store (copy) Ri into Memory location at
                   address X
```

Note that in the equation above a = M[A]. It represents content of memory location at address A. Same applies for B, C, X.

Question 3)
How many bits (in binary) would be required to encode instructions for this processor? Explain how.

Question 4)
Using encoding bits from Question 3) above, provide possible binary encoding for the following instructions:

a) `ADD R5, R1, R2`
b) `MOVE R4, #100`
c) `LOAD R3, M[C] (where C is memory address 2004)`

Question 5)
Provide micro-instructions in the form of detailed flowchart for the execution of the instructions in Question 4).

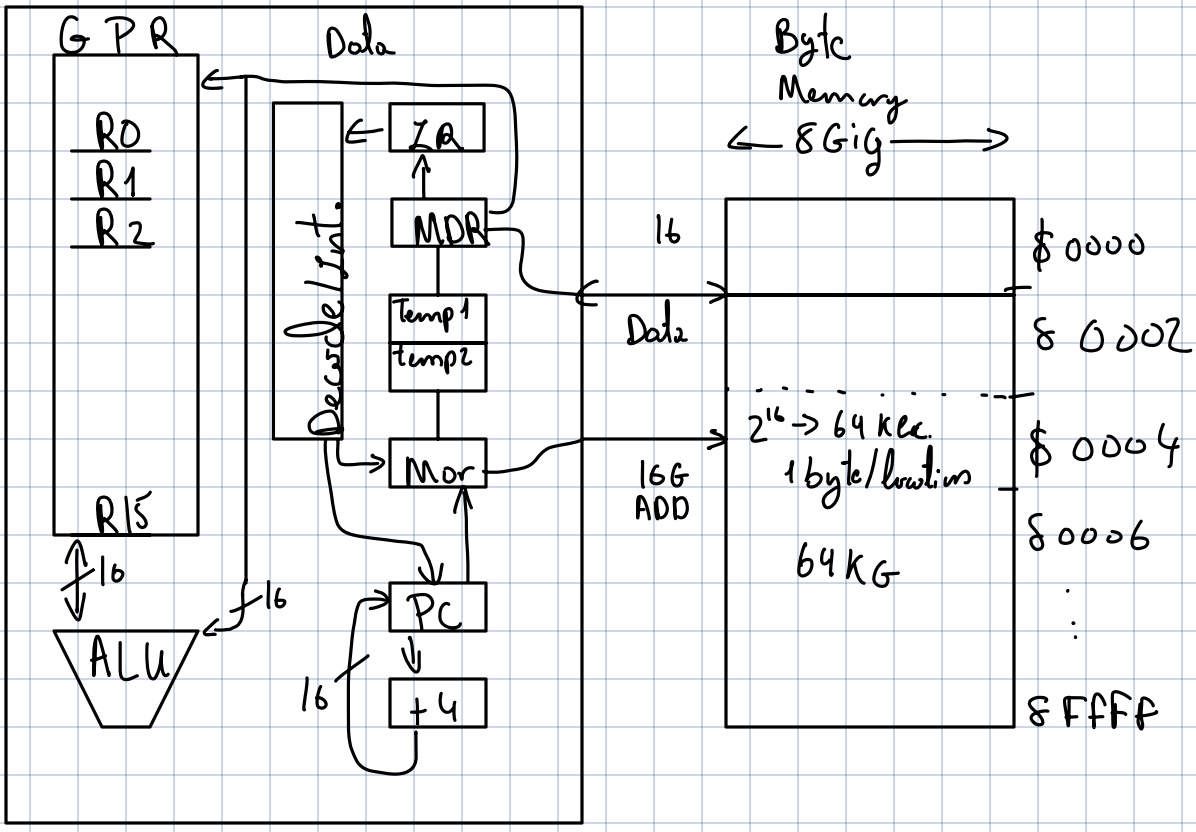
a) `ADD R5, R1, R2`
b) `MOVE R4, #100`
c) `LOAD R3, M[C] (where C is memory address 2004)`

Question 6)
Assuming an execution on a 2 GHz CPU, where every step requires a number of clock cycles (cc) as shown below, calculate the execution time (in ns) for every instruction in the instruction set as given in Question 5).

Internal Register Transfers	1 cc
PC Increment	2 cc
Decoding	1 cc
Memory Access	10 cc
Addition	3 cc
Multiplication	5 cc
Division	5 cc

In this assignment we want to design and measure the performance of a new processor (CPU) with 16 general purpose Registers (R0, …., R15), 16-bit data bus and 16-bit address bus.
Note: Memory is by default byte organized.

Question 1)
Draw the internal organization of this CPU by showing the widths of all registers as well as all external and internal buses. Also show the byte organized memory and its interface to the CPU.



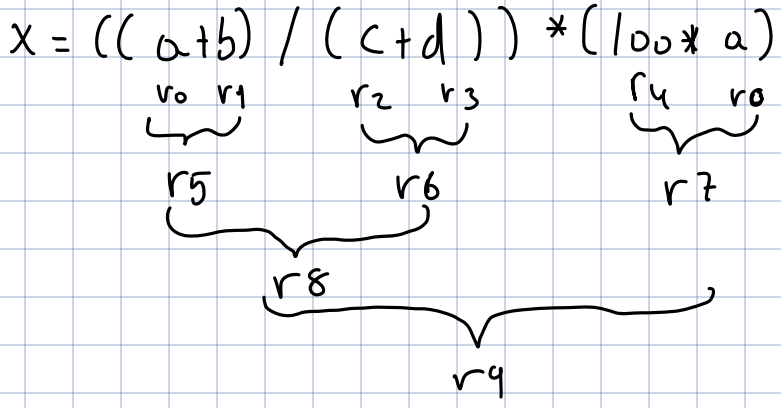
Register temp1 and temp2 are
bo complex inventory

Question 2)
Write an assembly program to evaluate the following expression for this CPU. Ignore the remainder part of the division process and assume the multiplication result is 16 bit instead of 32 bit.

```
x = ((a + b) / (a + c)) * (100 * a)
```

The instruction set of this processor consists of the following instructions:

```
ADD  Ri, Rj, Rk   ; add Rj to Rk and put result in Ri
MUL  Ri, Rj, Rk   ; multiply Rj by Rk and put result in Ri
DIV  Ri, Rj, Rk   ; divide Rj by Rk and put result in Ri
MOVE Ri, #100     ; move constant value #100 in Ri
LOAD Ri, M[X]     ; load (copy) content of Memory at address
                    X into Ri
STORE M[X], Ri    ; store (copy) Ri into Memory location at
                    address X
```



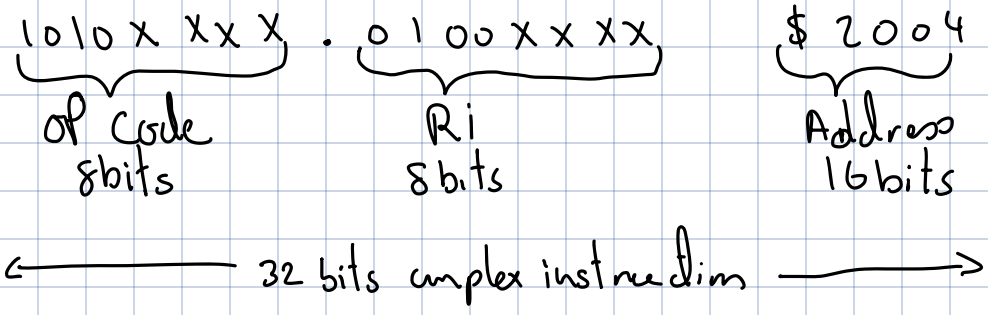
$r_0 \leftarrow a$	Load	r_0	, M[a]
$r_1 \leftarrow b$	Load	r_1	, M[b]
$r_2 \leftarrow c$	Load	r_2	, M[c]
$r_3 \leftarrow d$	Load	r_3	, M[d]
$r_4 \leftarrow \#100$	Move	r_4	, #100
$r_5 \leftarrow r_0 + r_1$	ADD	r_5	, r_0, r_1
$r_6 \leftarrow r_2 + r_3$	ADD	r_6	, r_2, r_3
$r_7 \leftarrow r_4 \times r_0$	Mul	r_7	, r_4, r_0
$r_8 \leftarrow r_5 / r_6$	DIV	r_8	, r_5, r_6
$r_9 \leftarrow r_8 * r_7$	Mul	r_9	, r_8, r_7
$r_{Am} \leftarrow r_9$	Store	M[X]	, r_9

Question 3)
How many bits (in binary) would be required to encode instructions for this processor? Explain how.

- 16 bit data bus
- 10 bit addr bus
- 6 instructions (Load, Move, ADD, Mul, Div, Store)
- 2 temp register

let Load Ri, M[x]

if x is a 16 bit address and Ri is one of 16 required 4 bits and Load is one of 6 require 2 bits.
Minimum instruction lenght is 22 bits but GPR is only 16 bits wide therefore complex instruction of 32 bits are required;



Question 4)

Using encoding bits from Question 3) above, provide possible binary encoding for the following instructions:

a) `ADD R5, R1, R2`

b) `MOVE R4, #100`

c) `LOAD R3, M[C] (where C is memory address 2004)`

a) ADD R5, R1, R2 → $A512 → 1010 0101 0001 0010

A 5 1 2

16 bits

b) Move R4, #100 → $B464 → 0001 0100 0110 0100

- Simple 16bit limited to 0 to 255
- complex 32bit instruction from 0 to 65,5k

c) Load R3, M[$2004] → C X 3 X 2 0 0 4

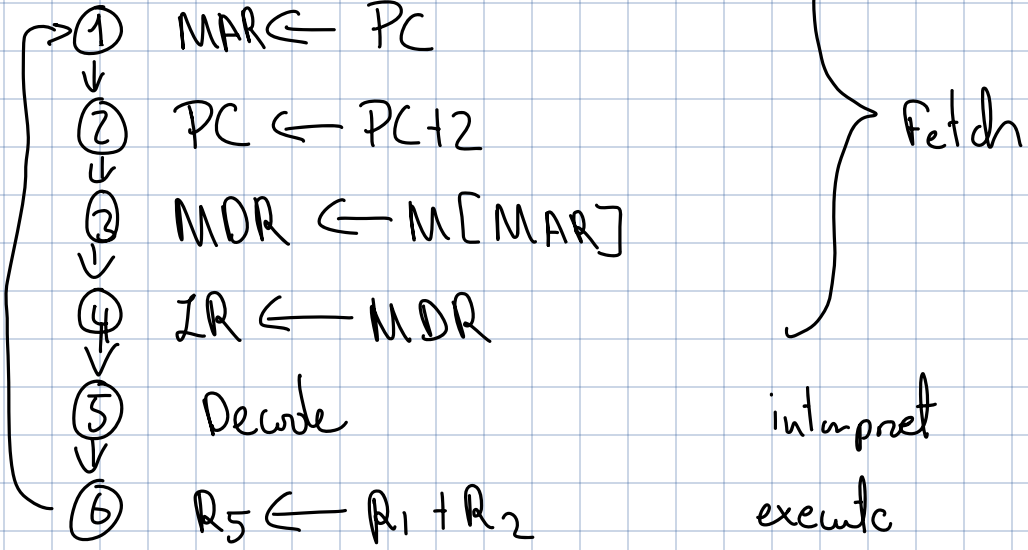
1100 XXXX 0011 XXXX 0010 0000 0000 0100

32 bits

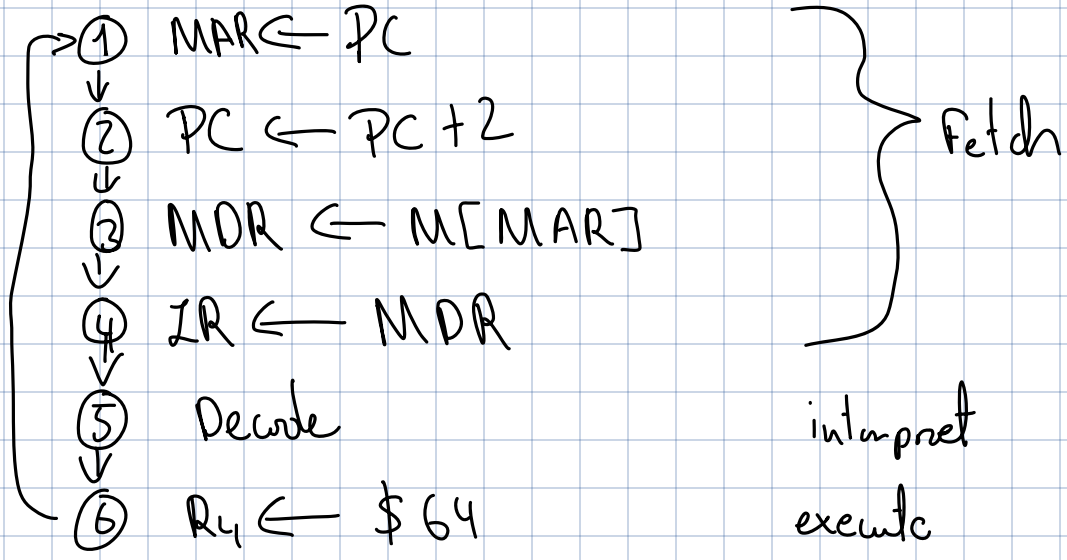
Question 5)
Provide micro-instructions in the form of detailed flowchart for the execution of the instructions in Question 4).

a) `ADD R5, R1, R2`
b) `MOVE R4, #100`
c) `LOAD R3, M[C]` (where C is memory address 2004)

a) ADD R_5, R_1, R_2



b) Move R_4 #100 → $B464



c) Load R3, M[$2004]

- this is a complex 32 bit interpretation
- lower 16 bits are the memory
- address $2004

PC ← PC + 2

MAR ← PC

MDR ← M[MAR]

R3 ← MDR

} Fetch memory add.

Question 6)

Assuming an execution on a 2 GHz CPU, where every step requires a number of clock cycles (cc) as shown below, calculate the execution time (in ns) for every instruction in the instruction set as given in Question 5).

Internal Register Transfers	1 cc
PC Increment	2 cc
Decoding	1 cc
Memory Access	10 cc
Addition	3 cc
Multiplication	5 cc
Division	5 cc

All: Decode 1cc Fetch 12 cc 1 NCR 2cc

a) ADD R_i, R_j, R_k 3cc → 18cc 9ns

b) Mul R_i, R_j, R_k 5cc → 20 10ns

c) Div R_i, R_j, R_k 5cc ↗

d) Move R_i, #100 2cc → 17 8,5 ns

Load R_i, M[A] 11cc → 26 13 ns

store M[A], R_i 11 cc ↗

2GHz → .5ns