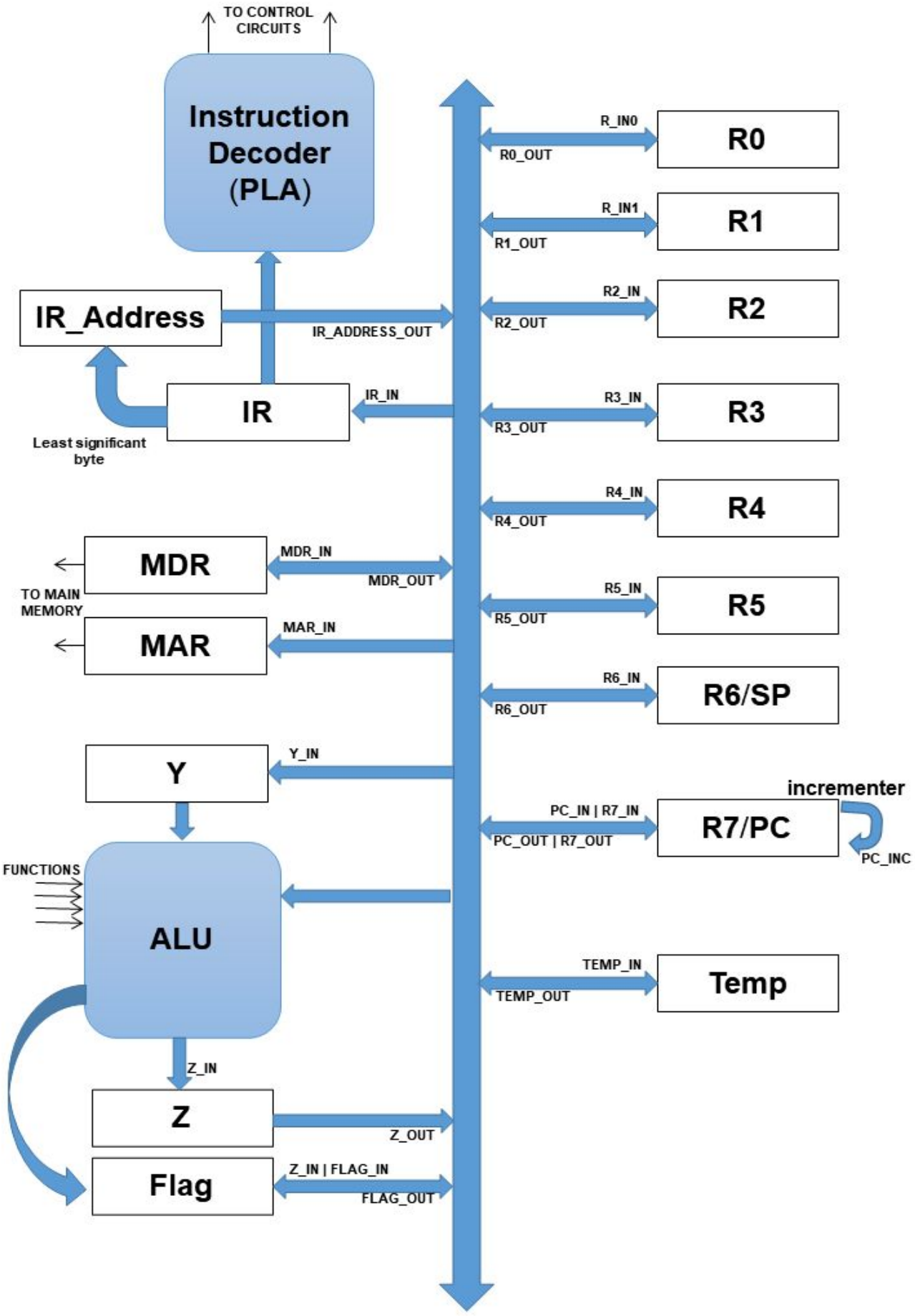
Cairo University - Faculty of Engineering
Computer Engineering Department
Computer Architecture (CMP301) – Fall 2020

Project phase two

Team 1

Name	**Sec**	**BN**
Hossam Alaa Ramadan Kamel	1	21
Muhammad Ahmad Hesham Mahmoud	2	15
Muhammad Alaa Abdel-Khaleq Ahmad	2	22
Mahmoud Amr Muhammad Refaat	2	23

Jan 2021


TO CONTROL CIRCUITS
Instruction Decoder (PLA)
IR_Address
IR_ADDRESS_OUT
IR
IR_IN
Least significant byte
MDR
MDR_IN
MDR_OUT
TO MAIN MEMORY
MAR
MAR_IN
Y
Y_IN
FUNCTIONS
ALU
Z_IN
Z
Z_OUT
Flag
Z_IN | FLAG_IN
FLAG_OUT
R_IN0
R0_OUT
R0
R_IN1
R1_OUT
R1
R2_IN
R2_OUT
R2
R3_IN
R3_OUT
R3
R4_IN
R4_OUT
R4
R5_IN
R5_OUT
R5
R6_IN
R6_OUT
R6/SP
incrementer
PC_IN | R7_IN
PC_OUT | R7_OUT
R7/PC
PC_INC
TEMP_IN
TEMP_OUT
Temp

Instruction Set Architecture:

Addressing modes

Register	000
Auto Increment	010
Auto Decrement	100
Indexed	110
Indirect Register	001
Indirect Auto Incr.	011
Indirect Auto Dec	101
Indirect Indexed	111

Registers

R0	000
R1	001
R2	010
R3	011
R4	100
R5	101
R6	110
R7	111

2 Operands Operations

OP code				Source addressing mode			Source register			Destination addressing mode			Destination register		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

OP codes

MOV	0001
ADD	0010
ADC	0011
SUB	0100
SBC	0101
AND	0110
OR	0111
XOR	1000
CMP	1001

1 Operand Operations

1	0	1	1	OP code				X	X	Destination Addressing mode			Destination register		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

OP codes

INC	0001
DEC	0010
CLR	0011
INV	0100
LSR	0101
ROR	0110
ASR	0111
LSL	1000
ROL	1001

Branch Operations

1	1	0	1	OP CODE				Offset							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

OP codes

BR	0001
BEQ	0010
BNE	0011
BLO	0100
BLS	0101
BHI	0110
BHS	0111

Special Operations

1	1	1	1	OP CODE				X	X	X	X	X	1	1	0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

OP codes

HLT	0001
NOP	0010
JSR	0011
RTS	0100

-Note:

JSR instruction syntax: **(labelX MUST NOT be followed by a colon)**

```
-JSR labelX          #jumps to address of labelX
-instruction_i

-.........

-.........
 -labelX
     - ......
     -........
     - RTS           #returns to address of instruction instruction_i
```

Micro instructions:

Control Word Grouping:

F0 (8 bits)	**F1 (3 bits)**	**F2 (3 bits)**	**F3 (2 bits)**
Next address field	000: NO_ACTION 001: PC_out 010: MDR_out 011: Z_out 100: R_src_out 101: R_dst_out 110: Temp_out 111: IR_Address_out	000: NO_ACTION 001: PC_in 010: IR_in 011: Z_in 100: R_src_in 101: R_dst_in	00: NO_ACTION 01: MAR_in 10: MDR_in 11: Temp_in

F4 (1 bit)	**F5(1bit)**	**F6 (4 bits)**	**F7 (2 bits)**
0: NO_ACTION 1: Y_in	0: NO_ACTION 1: Pc_inc	0000: BUS + 1 0001: BUS - 1 0010: Y + BUS 0011: Y+BUS + carry 0100: Y - BUS 0101: Y- BUS - carry 0110: Y & BUS 0111: Y | BUS 1000: LSR(Y) 1001: ROR(Y) 1010: Y^BUS 1011: ASR(Y) 1100: LSL(Y) 1101: ROL(Y) 1110: BUS 1111: INV(Y)	00: NO_ACTION 01: Read 10: Write

F8 (3 bits)	**F9 (1 bits)**	**F10 (1 bits)**
000: NO_ACTION 001: Or_dst 010: Or_indsrc 011: Or_inddst 100: Or_result 101: Or_branch 110: Or_operation	0: NO_ACTION 1: Pla_out	0: NO_ACTION 1: Halt

Analysis:

You can refer to the file (analysis.xlsx) for the memory access analysis , clock cycles analysis.

$$\text{CPI} = \frac{\sum Clock\ Cycles\ of\ all\ instructions}{Number\ of\ instructions} = \frac{6874}{660} = 10.4152$$

-Note: We've changed the assembler a little bit, so we'll attach the file with the required deliverables called (assembler.cpp)

-Note: You can find the actual micro instructions and how they will be distributed in the control store in the attached excel file called (Control Store.xlsx)

-Note: You can find the Instructions FlowChart in the attached file called (Instructions FlowChart.pdf)

Workload distribution :

Names	Workload
Hossam Alaa	-IR Register Structure -Assembler -ALU component -Registers -PLA -Tristate buffer -Components Integration
Muhammad Alaa	
Mahmoud Amr Mohamed Refaat	-Control store design -Bit ORing circuits -Program counter (as a counter) -Generic decoder -ROM component -RAM component -Components Integration
Muhammad Ahmad Hesham	