Faculty of Engineering
Cairo University

Computer Architecture: Spring 2023

CMPN301

Phase 3: Design & Implementation

Team C_2

Submitted to

Dr. Mayada Hadhoud

Dr. Dina Tantawy

Eng. Aya Hossam

Date: May 22, 2023

Ahmed Ameen	1190071
Ali Hassan	2200011
Hazem Montasser Abdel Azeem	2200003
Yousef Mahmoud Gilany	4200342

I. Instruction format

I-Type	***R-Type***	***J-Type***
IN	NOT	JZ
OUT	INC	JC
LDM	DEC	JMP
IADD	ADD	CALL
LDD	SUB	RET
STD	AND	RTI
PUSH	OR	
POP	MOV	
NOP		
SETC		
CLRC		

R-Type: (1 word wide)

Opcode[31:29]	ALUOp[28:26]	Rs[25:23]	Rt[22:20]	Rd[19:17]	x[16]

I-Type: (2 words wide)

Opcode[31:26]	Rs[25:23]	Rt[22:20]	xxxx[19:16]
Immediate value[15:0]			

J-Type: (1 word wide)

Opcode[31:26]	Rs[25:23]	xxxx[22:16]

II. Control signal table

Control signals

1. Mem-read
2. Mem-write
3. ALU src
4. ALU op
5. Mem-to-reg
6. Branch
7. Jump
8. RegDst
9. RegWrite
10. **Port EN**
11. **Flag EN**

	Mem-read	Mem-write	ALU src	ALU op	Mem-to-reg	Branch	Jump	RegDst	RegWrite	Port EN	Flag EN
IN	0	0	0	xxx	0	0	0	0	1	1	0
OUT	0	0	0	xxx	0	0	0	0	0	1	0
LDM	0	0	0	xxx	0	0	0	0	1	0	0
IADD	0	0	1	001	0	0	0	0	1	0	1
LDD	1	0	1	xxx	1	0	0	0	1	0	0
STD	0	1	1	xxx	0	0	0	0	0	0	0
PUSH (-)	0	1	0	xxx	0	0	0	0	0	0	0

	Mem-read	Mem-write	ALU src	ALU op	Mem-to-reg	Branch	Jump	RegDst	RegWrite	Port EN	Flag EN
POP (+)	1	0	0	xxx	1	0	0	0	1	0	0
NOP	0	0	0	000	0	0	0	0	0	0	0
SETC	0	0	1	xxx	0	0	0	0	0	1	1
CLRC	0	0	0	xxx	0	0	0	0	0	1	1
NOT	0	0	0	011	0	0	0	1	1	0	1
INC	0	0	0	111	0	0	0	1	1	0	1
DEC	0	0	0	110	0	0	0	1	1	0	1
ADD	0	0	0	100	0	0	0	1	1	0	1
SUB	0	0	0	101	0	0	0	1	1	0	1
AND	0	0	0	001	0	0	0	1	1	0	1
OR	0	0	0	010	0	0	0	1	1	0	1
MOV	0	0	0	000	0	0	0	1	1	0	0

	Mem-read	Mem-write	ALU src	ALU op	Mem-to-reg	Branch	Jump	RegDst	RegWrite	Port EN	Flag EN
JZ	0	0	0	xx0	0	1	0	0	0	0	0
JC	0	0	0	xx1	0	1	0	0	0	0	0
JMP	0	0	0	0	0	0	1	0	0	0	0
CALL (-)	0	1	0	0	0	0	1	0	0	0	0
RET (+)	1	0	0	0	0	0	1	0	0	0	0
RTI (+)	1	0	0	0	0	0	1	0	0	0	1
INT (-)	0	1	0	xxx	0	0	1	0	0	0	1
RST (initial value)	0	0	0	xxx	0	1	1	0	0	0	0

- Add AND gate with NOT Regwrite and PORT EN to the Read data 1 output decoder
- STD and LDD have the target address from Read data 1
    - STD: Register to be stored will be from Read data 2
    - LDD: Register to be loaded in will be from the write back stage and its address from instruction[20-18] which selects the register from the registerfile
- PUSH instruction will take the register to be pushed from Read data 2

ALU Operations

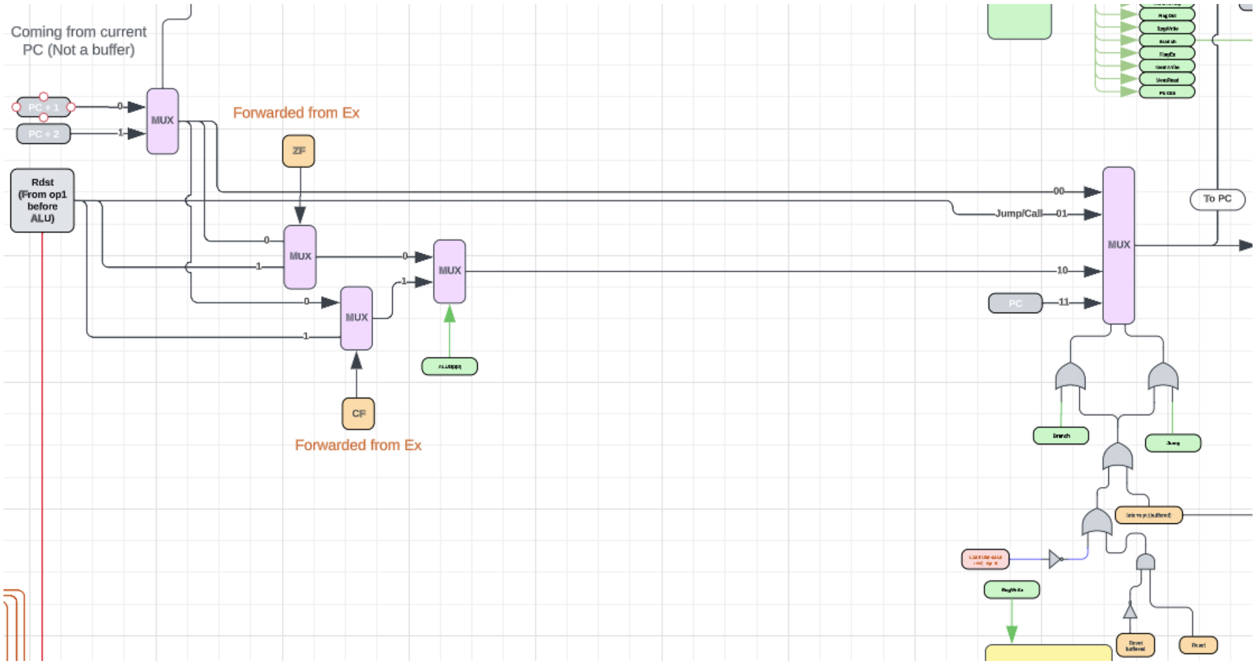
Operation	Inputs	Output	Opcode
NOP/Bypass	A	A	000
And	A, B	A&B	001
Or	A, B	A | B	010
Not	A	~A	011
Add	A, B	A + B	100
Sub	A, B	A - B	101
Dec	A	A - 1	110
Inc	A	A + 1	111

Note:

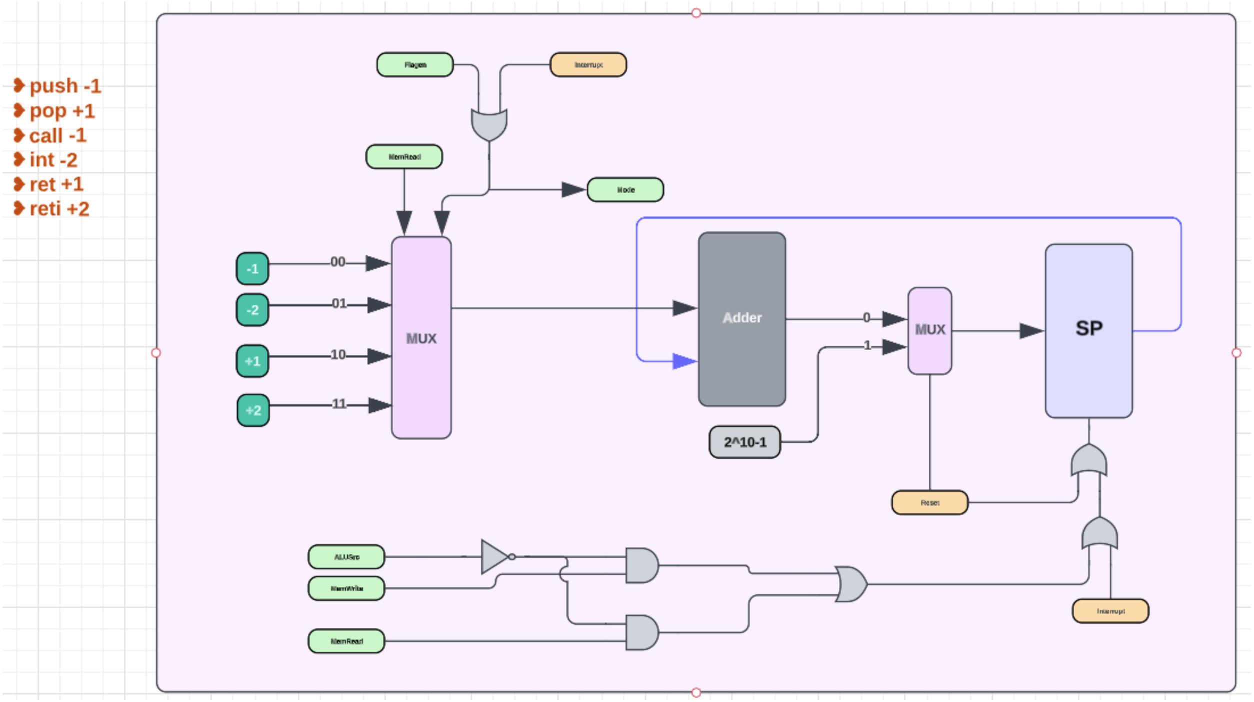
- ALU nop operation is used to bypass the left-hand side operand of the instructions in instructions such as MOV, STD
    - Ex: MOV R2,R5 -> output of ALU will be the value of R2, then R5 address is buffered till the write back to store R2 in R5.

III. Schematic diagram & Dataflow

1. PC update circuit



2. SP update circuit



IV. Pipeline stages design

Fetch stage (IF/ID)

Inputs	Ouputs

Instruction[31:0]	IF.Instruction[31:26] = ControlUnitInput
	IF.Instruction[25:23] = ReadAddr1
	IF.Instruction[22-20] = ReadAddr2
	IF.Instruction[19:17] = WriteAddr
	IF.Instruction[15:0] = ImmediateValue to SignExt

PC[15:0]	IF.PC[15:0]

Decode stage (ID/EX)

Inputs	Ouputs
ControlUnitOutput[10:0]	ID.ControlUnitOutput[10:0]
RegisterFile.ReadData1[15:0]	ID.ReadData1 to MUX circuit (OUT port or ALU read data 1)
RegisterFile.ReadData2[15:0]	ID.ReadData2 to MUX circuit (Immediate or ALU read data 2)
IF.Instruction[19:17](WriteAddr)	ID.Instruction[19:17]
IF.Instruction[22:20](ReadAddr2)	ID.Instruction[22:20]
Inst[15:0]	ID.Inst[15:0] to ● Immediate or ALU read data 2 and split to EX buffer
IF.PC[15:0]	ID.PC[15:0]
Forwarded SP[15:0]	ID.SP[15:0]

Note: PC[15:0] is forwarded back to the PC.

Execute stage (EX/MEM1)

Inputs	Ouputs
ID.Instruction[19:17](WriteAddr)	EX.Instruction[19:17](WriteAddr)
ID.Instruction[22:20](ReadAddr2)	EX.Instruction[22:20](ReadAddr2)
ID.ControlUnitOutput[10:0]	EX.ControlUnitOutput[10:0]
ID.Inst[15:0] to ● Immediate or ALU read data 2	EX.Inst[15:0] to ● Immediate or ALU read data 2
ID.PC[15:0]	EX.PC[15:0]
ALU.Result[15:0]	EX.ALUResult to data memory MUX (from SP or ALU result)
FlagRegister[2:0]	EX.FlagRegister[2:0]
ID.ReadData2[15:0]	EX.ReadData2 to Write data memory (Push/load or call/int)
ID.SP[15:0]	EX.SP[15:0] to data memory MUX (from SP or ALU result)
Output from decoder that selects between portOut or alu operand	EX.PortOut

Note:

- Memory read & write signals and write data are forwarded to data memory from Execute stage.

Memory 1 stage (MEM1/MEM2)

Inputs	Ouputs
EX.Instruction[19:17](WriteAddr)	MEM1.Instruction[19:17](WriteAddr)
EX.Instruction[22:20](ReadAddr2)	MEM1.Instruction[22:20](ReadAddr2)
EX.ControlUnitOutput[10:0]	MEM1.ControlUnitOutput[10:0]
EX.Inst[15:0] to ● Immediate or ALU read data 2	MEM1.Inst[15:0]
EX.PC[15:0]	MEM1.PC[15:0]
EX.FlagRegister[2:0]	MEM1.FlagRegister[2:0]
EX.ALUResult[15:0]	MEM1.ALUResult

EX.SP[15:0]	MEM1.SP[15:0] to data memory MUX (from SP or ALU result)
EX.PortOut	MEM1.PortOut

Note:

- 2 MUXes that are input to data for memory read address & write data will be before M1/M2 stage and their output will directly enter the memory without entering MEM1/MEM2 stage

Memory 2 stage (MEM2/WB)

Inputs	Ouputs
MEM1.PortOut	Out_port from processor
WB Address: 3-Bit address to be input for write address in register file. It comes from mux that selects between MEM1.Instruction[19:17](WriteAddr) & MEM1.Instruction[22:20](ReadAddr2)	MEM2.WB Address: 3-Bit address to be input for write address in register file
DataMemory.ReadData to DataMemOut to WBDecoder	MEM2.WBDecoderOut to RegisterFile.WriteData and Flag update circuit and PC update circuit