PREPARED FOR
Eng. Mohammed Abdallah

PREPARED FOR
Dr. Moustafa Mahmoud

ARCHITECTURE
REPORT

RISC-IT
NO RISC NO FUN

Team 3

Team Members

Name	ID
Ahmed Hamed Gaber	9220027
Ahmed Hamdy Mohammed	9220032
Rawan Ahmed Foad	9220302
Somia Saad Ismail	9202666

Instructions Opcode

> *Note: The opcode uses 5 bits, allowing for 32 possible values. Out of these, 25 are utilized, while the remaining 7 are designated as unused and are implemented as NOP (no operation).*

One Operand Instructions

Instruction	Opcode (BIN)	Opcode (OCT)
RESET	00000	0 0
HLT	00001	0 1
NOP	00010	0 2
SETC	00011	0 3
NOT Rdst, Rsrc1	00100	0 4
INC Rdst, Rsrc1	00101	0 5
IN Rsrc1	00110	0 6
OUT Rsrc1	00111	0 7

Two Operands Instructions

Instruction	Opcode (BIN)	Opcode (OCT)
ADD Rdst,Rsrc1,Rsrc2	01000	1 0
SUB Rdst,Rsrc1,Rsrc2	01001	1 1
AND Rdst,Rsrc1,Rsrc2	01010	1 2
MOV Rdst,Rsrc1	01011	1 3
IADD Rdst,Rsrc1,Imm	01100	1 4

Memory Instructions

Instruction	Opcode (BIN)	Opcode (OCT)
PUSH Rsrc1	10000	2 0
POP Rdst	10001	2 1
STD Rsrc1,offset(Rsrc2)	10100	2 4
LDD Rdst,offset(Rsrc1)	10101	2 5
LDM Rdst, Imm	10110	2 6

Branch Instructions

Instruction	Opcode (BIN)	Opcode (OCT)
JZ Rdst	11000	3 0
JN Rdst	11001	3 1
JC Rdst	11010	3 2
JMP Rdst	11011	3 3
CALL Rdst	11100	3 4
RET	11101	3 5
INT index	11110	3 6
RTI	11111	3 7

Instruction Bits Details

Note: The instruction is 16 bit long. After decoding the instruction, we identify those that require immediate values. In the fetching stage (second cycle), the immediate value is then forwarded to the execution unit.

One Operand Instructions

15→11	10→8	7→5	4→2	1→0
Opcode	Rsrc1	Rdst	-	-

two Operands Instructions

15→11	10→8	7→5	4→2	1→0
Opcode	Rsrc1	Rsrc2	Rdst	-

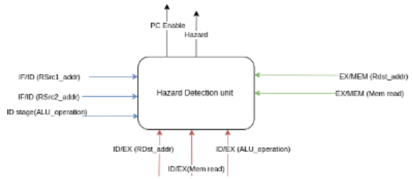
Memory Instructions

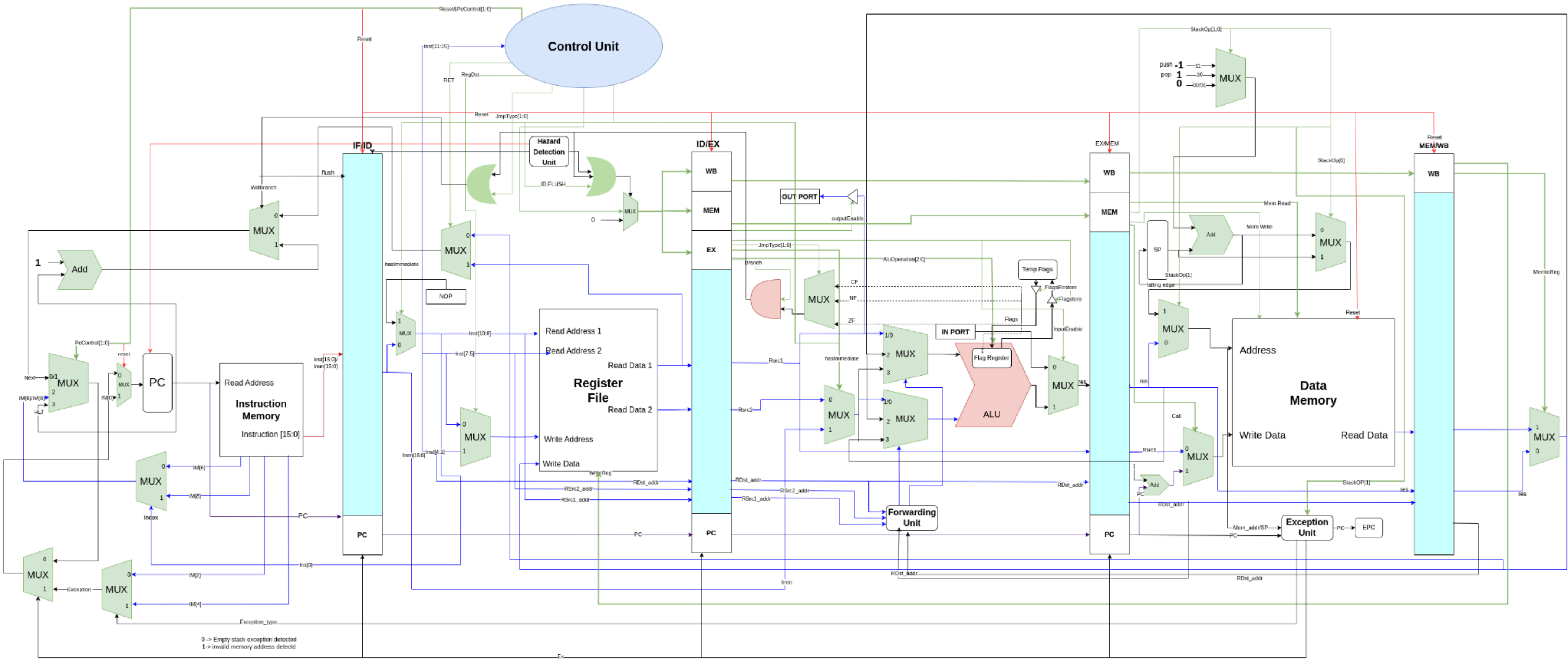
15→11	10→8	7→5	4→2	1→0
Opcode	Rsrc1	Rsrc2/-	-	-

Jump Instructions

15→11	10→8	7→5	4→2	1→0
Opcode	Rsrc1	-	-	index/-

Schematic Diagram





Control Unit Signals

Signal	Bits	Signal	Bits	Signal	Bits
MemtoReg	1	Reset	1	MemRead	1
RET	1	AluOperation	3	MemWrite	1
PcCntrl	2	hasImediate	1	FlagRestore	1
WriteReg	1	Branch	1	FlagStore	1
call	1	RegDst	1	InEnable	1
StackOp	2	IF Flush	1	OutEnable	1
JmpType	2	ID Flush	1	-	-

Pipeline Registers Size

Stage	Bits Details	Required	Size
IF/ID	PC[15:0] + Ins/Imm[15:0]	32	32
ID/EX	Rsrc1_addr[2:0] + Rdst_addr[2:0] + Rsrc1[15:0] + Sig[17:0] + PC[15:0] + Rsrc2_addr[2:0] + Rsrc2[15:0]	74	128
EX/MEM	Sig[6:0]+Rsrc1[15:0] + Res[15:0] + Rdst_addr[2:0] + PC[15:0]	58	64
MEM/WB	Sig[1:0]+Mem[15:0] + Res[15:0] + Rdst_addr[2:0]	37	64

Pipelined Hazards & Solution

Structural Hazards

Decode & Writeback

We made the write back be performed during the first half of the clock cycle, while decode occurs during the second half.

Stack

In the memory stage, stack data is read during the first half of the clock cycle, and any required modifications to the stack are performed in the second half.

Data Hazards

- Full-forwarding(ALU-ALU, Memory-ALU)
- Fetch-ALU forwarding (Immediate Values)

Control Hazards

Branch Prediction

We assume that branches are not taken by default. Depending on the instruction, we determine whether our prediction was correct during either the fetch or execute stages.

Cases of misprediction detection:

- In the fetch stage, we flush the preceding instruction.
- In the execute stage, we flush the two preceding instructions.