CO502 - Computer Architecture
Part 2- Report
Group 04

Hardware Units

1. ALU - Arithmetic Logic Unit

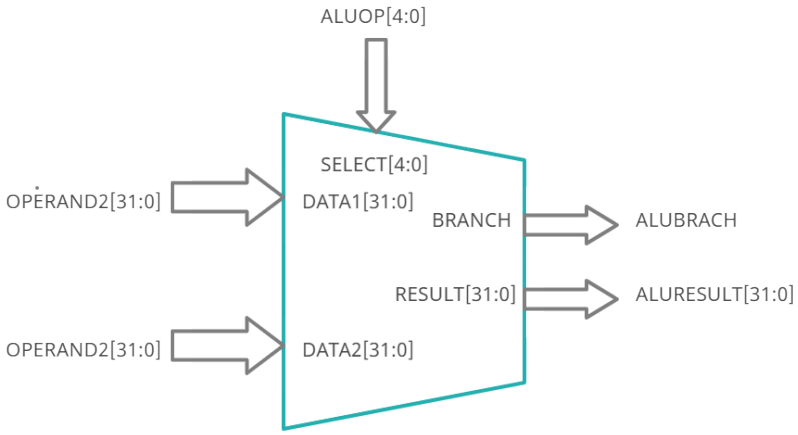

FIGURE 1 - ALU design

Inputs :

DATA1[31:0] - 32 bit operand 1
DATA2[31:0] - 32 bit operand 2
SELECT[4:0] - 5 bit ALUOp

Output :

RESULT[31:0] - 32 bit ALU Result
BRANCH - 1 bit Branch Result (which used for branch instructions)

In the alu the main function is getting 2 DATA inputs and do the operation according to the SELECT(which contain the alu opcodes) and give RESULT and BRANCH as output.

5 bits are assigned for ALUOp and 32 options for SELECT. And used 17 options for giving various RESULT outputs. And used another 6 for giving BRANCH output. If there is no RESULT or BRANCH output needed for an ALUOp it returns 0 as default.

The ALUOps assigned as Table 1 for operation,

ALUOP[4:0]	RESULT[31:0]	BRANCH
00000	RES_ADD	0
00001	RES_SLL	0
00010	RES_SLT	0
00011	RES_SLTU	0
00100	RES_XOR	0
00101	RES_SRL	0
00110	RES_OR	0
00111	RES_AND	0
01000	0	BR_BEQ
01001	0	BR_BNE
01010	0	0
01011	0	0
01100	0	RES_BLT
01101	0	RES_BGE
01110	0	RES_BLTU
01111	0	RES_BGEU
10000	RES_SUB	0
10001	RES_FWD	1
10010	0	0
10011	0	0
10100	0	0
10101	0	0
10110	0	0
10111	0	0
11000	RES_MUL	0
11001	RES_MULH	0
11010	RES_MULHSU	0
11011	RES_MULHU	0
11100	RES_DIV	0
11101	RES_REM	0
11110	RES_FWD	0
11111	RES_REMU	0

Table 1 : ALU Op codes

ALL the operation takes DATA2 which comes through the multiplexer therefore as initial plan set 2 time unit delay.

RESULT give in 32 bit binary format there for values which contain more than 32 bit send RESULT by negotiating extra bits. Therefore, It can't do a calculation where the result value exceeds the 32 bit and can't give the exact correct answers.

The ALU module is tested using a test bench and all the functions work properly in the ALU model test.

2. Register File

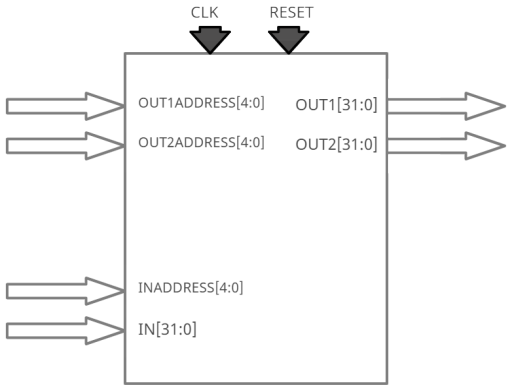

FIGURE 2 : Register file design

Inputs:

OUT1ADDRESS [4:0] - address of OUT1 (data which read from the register file)
OUT2ADDRESS [4:0] - address of OUT2 (data which read from the register file)
INADDRESS[4:0] - address of IN (Data which write on the register file)
IN [31:0] - The which need to write on register file
WRITEANABLE - the signal to enable the writing data on register. This is 1 when need to write on register
CLK - clock
RESET- result signal

Outputs:

OUT1 [31:0] - Data value 1 read from the register according to the OUT1ADDRESS
OUT1 [31:0] - Data value 1 read from the register according to the OUT1ADDRESS

The register file can keep 32 data values of 32 bits (4 Bytes).
After 2 time units Writing will happen to the Register.

3. Control Unit

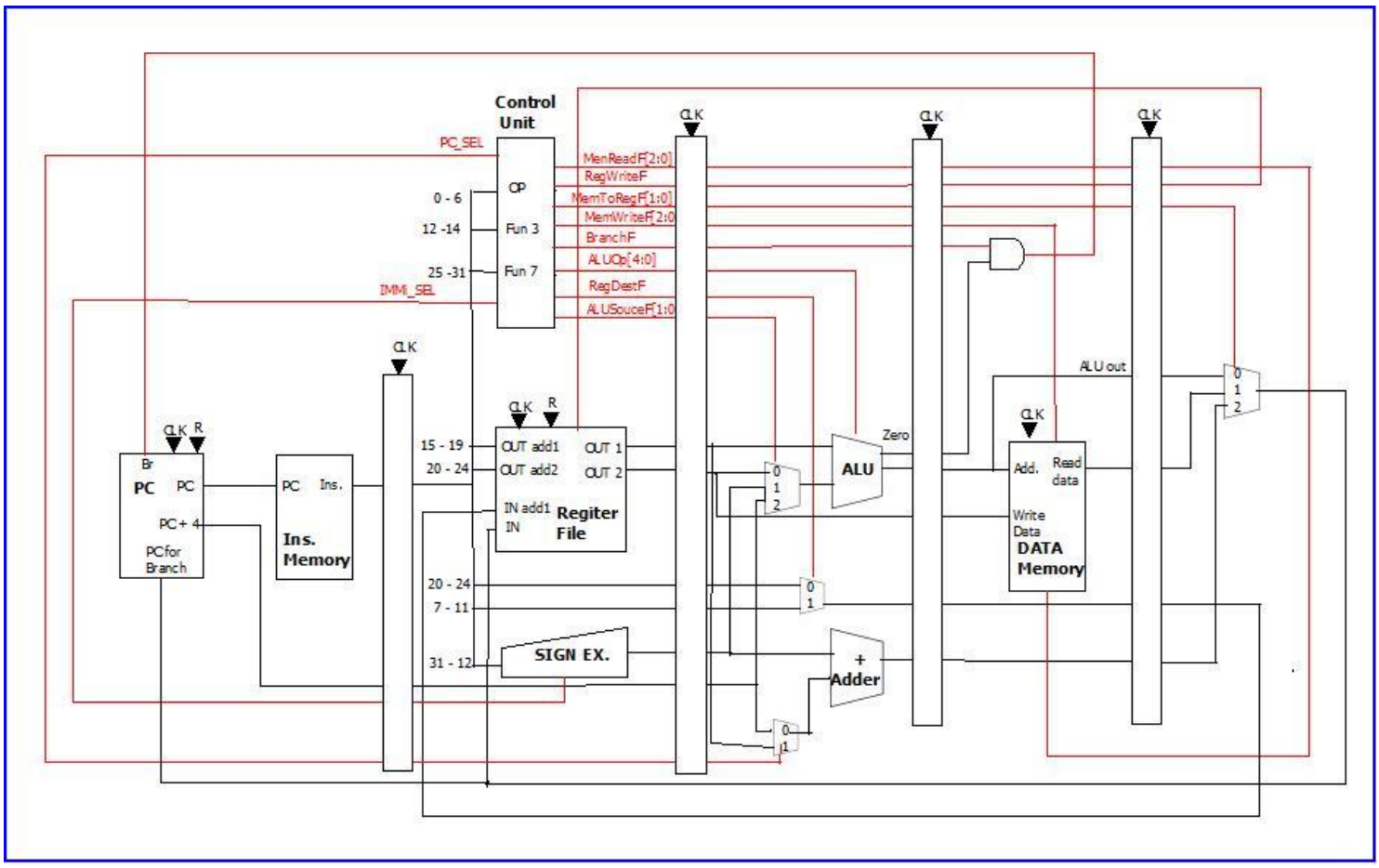

FIGURE 3 : Pipeline CPU

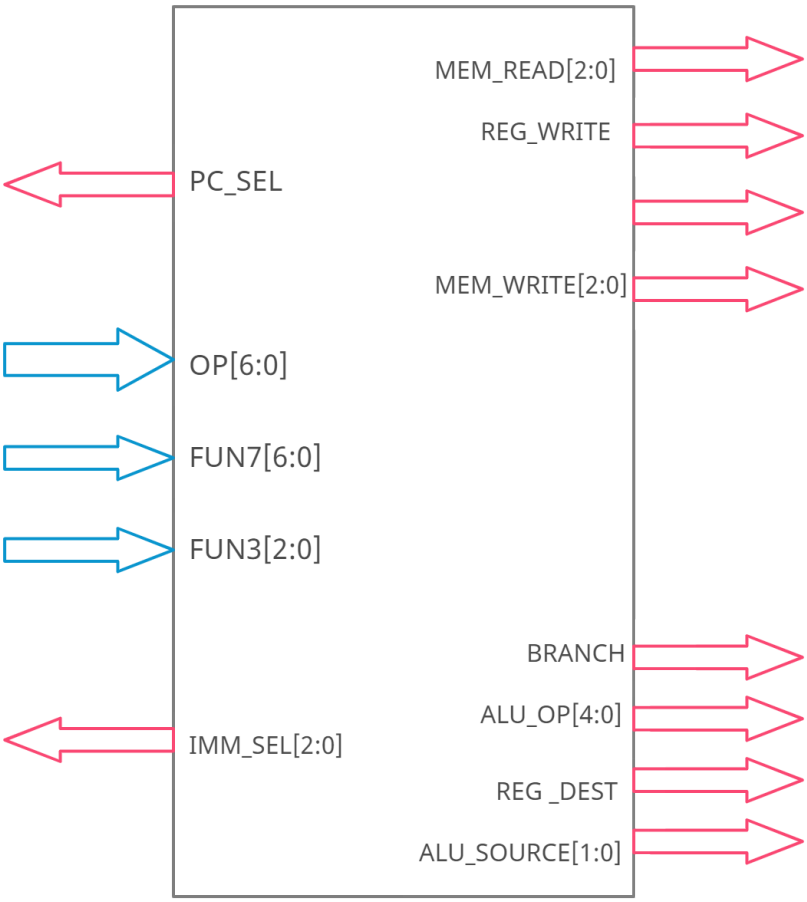

FIGURE 4 : Control unit design

Inputs :

OPCODE[6:0]	- opcode of the instruction
FUN7[6:0]	- function 7 of the instruction
FUN3[2:0]	-function 3 of the instruction

Outputs :

- Memory Read Flag
- Register write Flag
- Memory Write flag
- Memory to register flag
- Branch flag
- Alu opcode
- Register destination flag

Alu source flag
Immediate value selector
PC selector

There are 10 control signals which are controlled by this unit.

Instruction	Type	MEMread	MEMWrite	Memorytoreg	REGwrite	ALUsource	Branch	Registerdestination	IMM_SEL	PC_SEL
LB	I-Type	1	0	1	1	1	0	1	0	x
LH	I-Type	2	0	1	1	1	0	1	0	x
LW	I-Type	3	0	1	1	1	0	1	0	x
LBU	I-Type	4	0	1	1	1	0	1	0	x
LHU	I-Type	5	0	1	1	1	0	1	0	x
ADDI	I-Type	0	0	0	1	1	0	1	0	x
SLTI	I-Type	0	0	0	1	1	0	1	0	x
SLTIU	I-Type	0	0	0	1	1	0	1	0	x
XORI	I-Type	0	0	0	1	1	0	1	0	x
ORI	I-Type	0	0	0	1	1	0	1	0	x
ANDI	I-Type	0	0	0	1	1	0	1	0	x
SLLI	I-Type	0	0	0	1	1	0	1	1	x
SRLI	I-Type	0	0	0	1	1	0	1	1	x
SRAI	I-Type	0	0	0	1	1	0	1	1	x

SB	S-Type	0	1	x	0	1	0	x	2	x
SH	S-Type	0	2	x	0	1	0	x	2	x
SW	S-Type	0	3	x	0	1	0	x	2	x
SBU	S-Type	0	4	x	0	1	0	x	2	x
SHU	S-Type	0	5	x	0	1	0	x	2	x
ADD	R-Type	0	0	0	1	0	0	1	x	x
SUB	R-Type	0	0	0	1	0	0	1	x	x
SLL	R-Type	0	0	0	1	0	0	1	x	x
SLT	R-Type	0	0	0	1	0	0	1	x	x
SLTU	R-Type	0	0	0	1	0	0	1	x	x
XOR	R-Type	0	0	0	1	0	0	1	x	x
SRL	R-Type	0	0	0	1	0	0	1	x	x
SRA	R-Type	0	0	0	1	0	0	1	x	x
OR	R-Type	0	0	0	1	0	0	1	x	x
AND	R-Type	0	0	0	1	0	0	1	x	x
MUL	R-Type	0	0	0	1	0	0	1	x	x
MULH	R-Type	0	0	0	1	0	0	1	x	x
MULHSU	R-Type	0	0	0	1	0	0	1	x	x
MULHU	R-Type	0	0	0	1	0	0	1	x	x
DIV	R-Type	0	0	0	1	0	0	1	x	x
REM	R-Type	0	0	0	1	0	0	1	x	x
REMU	R-Type	0	0	0	1	0	0	1	x	x
LUI	U-type	0	0	0	1	1	0	1	3	x
AUIPC	U-type	0	0	2	1	x	0	1	3	x
JALR	I-Type	0	0	0	1	2	1	1	3	0
JAL	J-Type	0	0	0	1	2	1	1	3	1
BEQ	B-Type	0	0	2	0	0	1	x	2	0
BNE	B-Type	0	0	2	0	0	1	x	2	0
BLT	B-Type	0	0	2	0	0	1	x	2	0
BGE	B-Type	0	0	2	0	0	1	x	2	0
BLTU	B-Type	0	0	2	0	0	1	x	2	0

BGEU	B-Type	0	0	2	0	0	1	x	2	0

Table 2 : Instruction with control signals

https://docs.google.com/spreadsheets/d/1OVitoZ5JZzUCN-YxDwMBUPv7XC754ZeRF_KaZLkNERE/edit?usp=sharing

Follow this link for control signals with all ALU op codes.(Sheet 3)

- To process S type and J type instructions 2 new control signals are added(immediate value selector and PC selector).Since there are few types of immediate value range selections,a new Immediate value selector control signal is there and which will be processed in the Sign extender unit.
- Also there connection is made between ALU and the PC value,because of JAL and JALR instructions.Because they write PC value to the source register.
- Also there is another wire to connect PC adder and ALU output to process branch instruction and U-type instructions.
- There are no time delays for the control unit signal generating.
- And also signals are generated asynchronously.