Mohammad Alturk – 150108931
AKif Batur - 150111854

Marmara University
Faculty of Engineering
Department of Computer Engineering

Computer Organization 2013-2014
CSE338

Project#2

Mohammad Alturk - 150108931
AKif Batur - 150111854

CONTROL UNIT :

assign rformat=~|in;

assign lw=in[5]& (~in[4])&(~in[3])&(~in[2])&in[1]&in[0];

assign sw=in[5]& (~in[4])&in[3]&(~in[2])&in[1]&in[0];

assign beq=~in[5]& (~in[4])&(~in[3])&in[2]&(~in[1])&(~in[0]);

They are already defined we are getting the true opcode of their outputs

my new additinal instructions according to the given opcode

assign bmn=(~in[5])&(in[4])&~in[3]&(in[2])&(~in[1])&in[0];	// bmn 21	010101	I-type
assign balmn=(~in[5])&(in[4])&~in[3]&(in[2])&(in[1])&in[0];	// balmn 23	010111	I-type
assign bn=(~in[5])&(in[4])&in[3]&(~in[2])&(~in[1])&in[0];	// bn 25	011001	J-type
assign bneal=(in[5])&(~in[4])&in[3]&(in[2])&(~in[1])&in[0];	// bneal 45	101101	I-type
assign bgtz=(~in[5])&(~in[4])&~in[3]&(in[2])&(in[1])&in[0];	// bgtz 7	000111	I-type
assign jal=(~in[5])&(~in[4])&~in[3]&(~in[2])&(in[1])&in[0];	// jal 3	000011	J-type
assign ori=(~in[5])&(~in[4])&in[3]&(in[2])&(~in[1])&in[0];	// ori 13	001101	I-type

we are using the opcode here(Hint)

input [5:0] in;

if the opcode is (000000) →| (000000) ~→ 111111

or

if the opcode is (100000) →| (100000)¨ ~→ 011111

standard design fort he given signals :

Mohammad Alturk - 150108931
AKif Batur - 150111854

assign regdest=rformat;// **when I have 3 register {rt,rt,rd}**

assign alusrc=lw|sw|ori | bmn | balmn; // **pass from alusrc**

assign memtoreg=lw |balmn|bmn;// **read from memeory**

assign regwrite=rformat|lw|Link ;// **read from the memory (lw)and r type**

assign memread=lw|bmn|balmn ; // **read from the memory necessary data**

assign memwrite=sw; // **for write something to the memory**

assign branch=beq|balmn|bmn|bn|bneal; // **when the condition get ture**

assign aluop1=rformat|ori;// **using opcode to make the necessary operaiton or ori**

assign aluop0=beq|ori|bneal; // **make subtraction when its true or ori**

The additional design for new signals :

assign jump=jal;// // **jump used just for**

assign PcSource1 = bmn|balmn|bneal|beq|bgtz; // **I am giving all branch signals 11**

assign PcSource2 =bgtz|beq|bneal|bn|jal; // **I am giving all branch signals 11**

assign JumpNotZero=bmn|balmn|bneal|bn; // **all they work at not flag**

assign Flageski=balmn|bmn|bn; //**when i do not have label it Works**

assign Link=jal|balmn|bneal; //**just for link instruction**

Alucont :

we set the arithmetic control unit [2 bits] with necessary function .

if(~(aluop1|aluop0)) gout=3'b010;// **toplama 00**

if(aluop0&~(aluop1))gout=3'b110 ; // **çıkarma - branch 01**

if(aluop0 &aluop1)gout=3'b001; // **ori 11**

Mohammad Alturk – 150108931
AKif Batur - 150111854

if(~(aluop0) & aluop1) // **function code 10**

alu32:

in the Project we use only 2 flags : **Zero flag** and **Negative flag**

reg [1:0] flag;

in the Project , the given instructions which Work with zero flag use the not of zero flag , when zero flag not set it Works

and i am using the negative flag

jumpbranchunit:

we connect the result are taken from alucontrol with status register 2 bit for negative flag and zero flag , we distribute

assign flag = flageskikullanma ? flageski: flagyeni; // beq and bne would use flagyeni (defualt)

// bmn balmn baln use flageski

assign exor = JumpNotZero ^^ flag[0];

// with **jump not zero** and **bne** would set **bne**

// with not setting **jump not zero** and **beq** would set **beq**

// with **jump not zero** and **bmn** would set **bmn**

// with **jump not zero** and **balmn** would set **balmn**

Mohammad Alturk - 150108931
AKif Batur - 150111854

// with **jump not zero** and **baln** would set **baln**

// with **jump not zero** and **bneal** would set **bneal**

assign noor = ~(flag[0] | flag[1]);

// with **ZeroFlag** and **NegativeFlag** would set

assign whichone = Bgtz ? noor : exor; ı am using control signal (Bgtz) to let Bgtz pass or not

i am controling all the branch instruction with andout gate

assign andout= branch && whichone;

Muxes:

2 x 1 mux for save the address of jal

2 x 1 mux for save the pc+4 in (ra 31 register)

2 x 1 mux for ori zero extend

2 x 1 mux for control the bmn balmn and baln which the look for always to the last Z flag's value defualt(0)

2 x 1 mux for bgtz

assign orout= andout | jump; // let branch and jump instruction pass according to the given signals from pcsorurce1 and pcsource2

assign pcsourceout1=pcsource1 &&orout; gather all the gate results according to the given signals

assign pcsourceout2=pcsource2&& orout; which are selected in pcsource2 ve pcsource1

Mohammad Alturk - 150108931
AKif Batur - 150111854

Test Part

bmn I-type opcode=21 bmn imm16($rs) if Status [Z] = 0, branches to address found in memory

[immdiate + rs]

0x 04 + 0x12 it is going to search for 0d22.addresss memory → 00002200

0x08 + 0x12 it is going to search for 0d32.addresss memory →01240000

0x02 + 0x07 it is going to search for 0d09.addresss memory →00001000

Opcode R1 $0 immediate

010101 00 001 00000 00000000 00000100

hex → 542000 04

hex → 542000 08

hex → 542000 02

Explanation : the funciton of this intruction to add [immediate + R1's values] and branch to this address in the memory

balmn I-type opcode=23 balmn $rt, imm16($rs) if Status [Z] = 0, branches to address found in memory, link address is stored in $rt (which defaults to 31)

[immdiate + rs]

0x 05 + 0x02 it is going to search for 0d7.addresss memory → 05000000

0x04 + 0x04 it is going to search for 0d7.addresss memory → 00000010

hex → 5C210005

hex →5C210004

explanation : after implement it save rt to pc +4 00000004

Mohammad Alturk – 150108931
AKif Batur - 150111854

bn J-type opcode=25 bn Target if Status [Z] = 0, branches to pseudo-direct address (formed

as j does)

[immediate]*2

Hex →64000003 0x0000000C

Hex →64000008 0x00000020

Explanation : the function of this instruction to branch to pesudo –direct address immediately , comparing with normal branch actuatlly it does not have another difference just it branh directly when the zero flag is set

bneal I-type opcode=45 bneal $rs, $rt, Target if R[rs] != R[rt], branches to PC-relative address (formed as beq & bne do), link address is stored in register 31

if rs is not equal rt jump label to PC-relative

Hex → B4220003 + pc+4 0x00000010

Hex → B4220005 + pc+4 0x00000018

Bgtz
if rs is greater than zero jump label to PC-relative

Hex → 1C200007 + pc+4 **0x00000020**

Jal

Jump to pseudo address

Hex → 0C000012 0x00000048

Mohammad Alturk – 150108931
AKif Batur - 150111854

ORI

[immdiate | rs]

0x12 0x12

Hex → 34220012 00000012