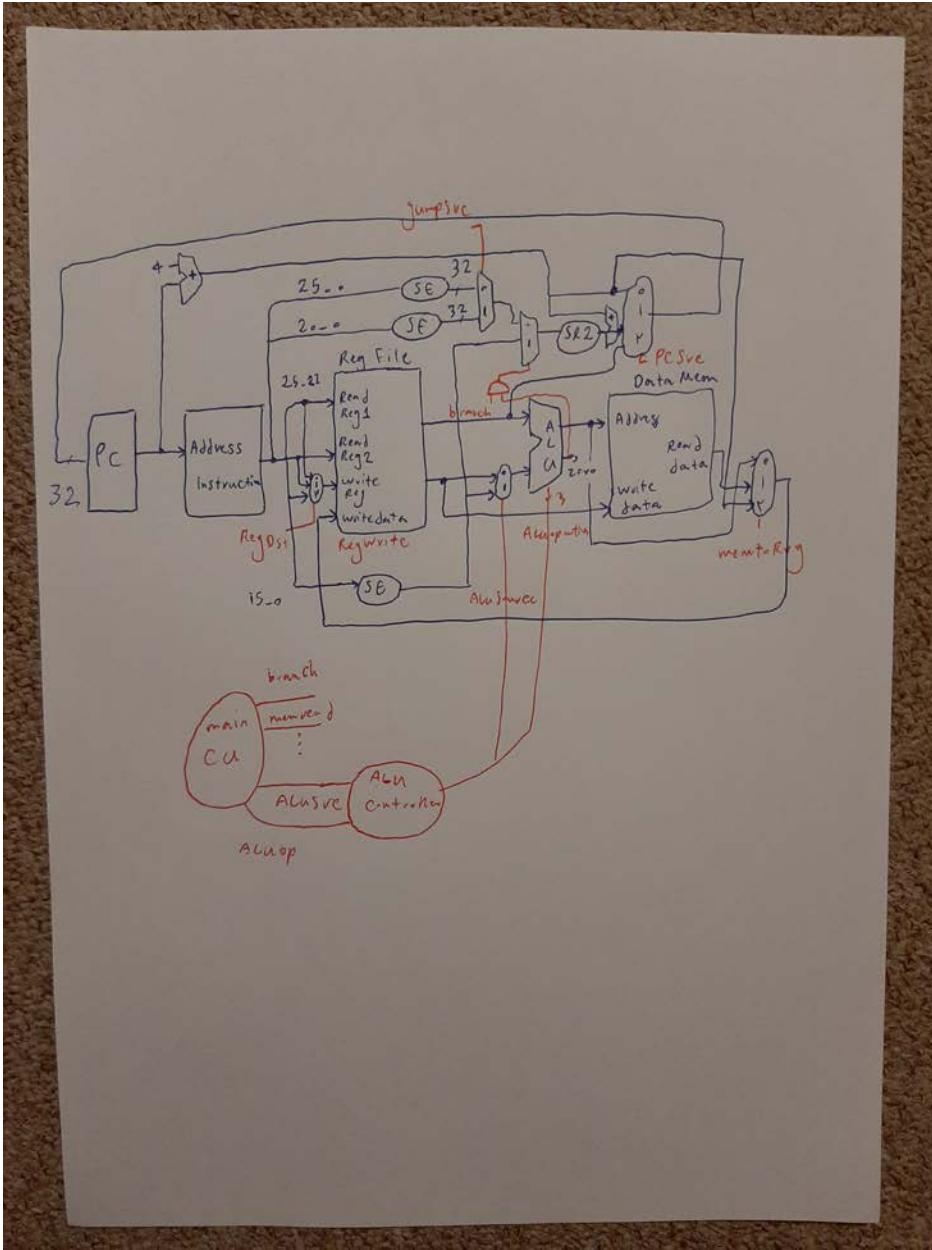


Single cycle MIPS's Data Path



Single cycle MIPS's Controller

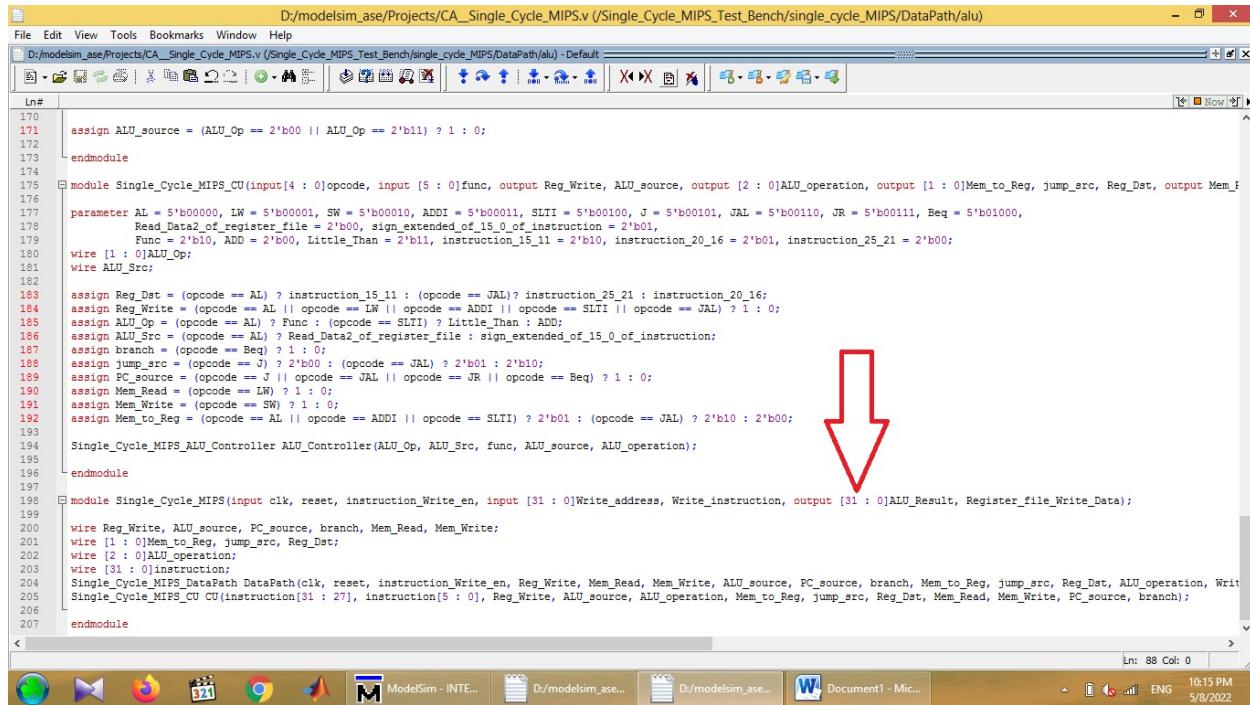
	RegDst	Regwrite	AluSrc	AluOp	jump	branch	PCsrc	MemRead	MemtoReg
op code									
ALU:00000	1	1	10	10	X	0	00	0	00
LW:00011	0	1	01	00	X	0	00	0	101
SW:00010	X	0	01	00	X	0	00	1	0
addi:00111	X	1	01	00	X	0	00	0	00
Slti:00100	X	1	01	11	X	0	00	0	XX
j:00101	X	0	XX	XX	0	0	01	0	00
jal:00110	X	0	XX	XX	1	0	11	0	XX
jr:00111	X	0	XX	XX	X	0	11	0	XX
beq:01000	X	0	XX	01	X	1	01	0	00

AluSrc : 00 → 0
01 → 1
10 → func

Alu op : 00 → +
01 → -
10 → func
11 → little than

	Alu Controller	Alu Source
AluOp: 00	+	
AluOp: 01	-	0
AluSrc: 00		1
AluSrc: 01		
O.W.	+	0
add:	-	0
Sub:		
mult:	comp	0
Slt: 01	and	0
and: 01	or	0
or: 10		

Single cycle MIPS's Top level



```

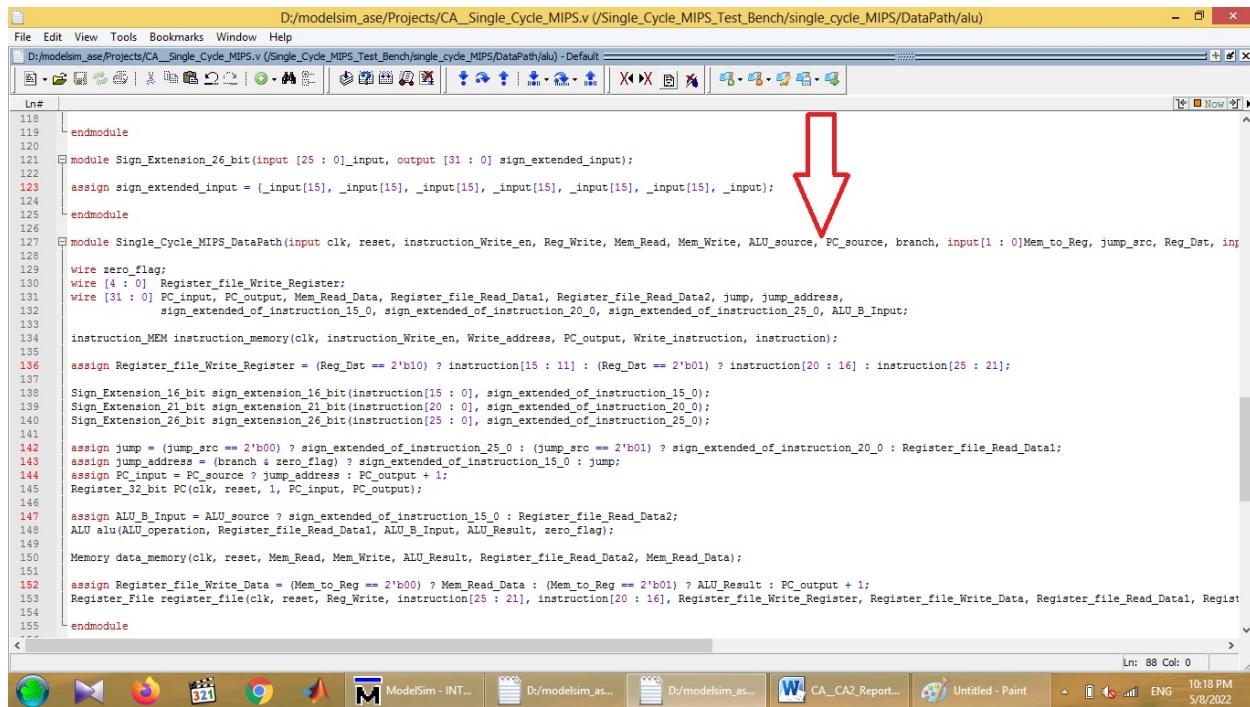
D:/modelsim_ase/Projects/CA_Single_Cycle_MIPS.v (/Single_Cycle_MIPS_Test_Bench/single_cycle_MIPS/DataPath/alu)
File Edit View Tools Bookmarks Window Help
D:/modelsim_ase/Projects/CA_Single_Cycle_MIPS.v (Single_Cycle_MIPS_Test_Bench/single_cycle_MIPS/DataPath/alu) - Default
Ln#
170
171 assign ALU_source = (ALU_Op == 2'b00 || ALU_Op == 2'b11) ? 1 : 0;
172
173 endmodule
174
175 module Single_Cycle_MIPS_CU(input[4 : 0]opcode, input [5 : 0]func, output Reg_Write, ALU_source, output [2 : 0]ALU_operation, output [1 : 0]Mem_to_Reg, jump_src, Reg_Dst, output Mem_I
176 parameter AL = 5'b00000, LW = 5'b00001, SH = 5'b00010, ADDI = 5'b00011, SLTI = 5'b00100, J = 5'b00101, JAL = 5'b00110, JR = 5'b00111, Beq = 5'b01000,
177 Read_Data2_of_register_file = 2'b00, sign_extended_of_15_0_of_instruction = 2'b01,
178 Func = 2'b10, ADD = 2'b00, Little_Than = 2'b11, instruction_15_11 = 2'b10, instruction_20_16 = 2'b01, instruction_25_21 = 2'b00;
179
180 wire [1 : 0]ALU_Op;
181 wire ALU_Src;
182
183 assign Reg_Dst = (opcode == AL) ? instruction_15_11 : (opcode == JAL) ? instruction_25_21 : instruction_20_16;
184 assign Reg_Write = (opcode == AL || opcode == LW || opcode == ADDI || opcode == SLTI || opcode == JAL) ? 1 : 0;
185 assign ALU_Op = (opcode == AL) ? Func : (opcode == SLTI) ? Little_Than : ADD;
186 assign ALU_Src = (opcode == AL) ? Read_Data2_of_register_file : sign_extended_of_15_0_of_instruction;
187 assign branch = (opcode == Beq) ? 1 : 0;
188 assign jump_src = (opcode == J || opcode == JAL) ? 2'b01 : 2'b10;
189 assign PC_Source = (opcode == J || opcode == JAL || opcode == JR || opcode == Beq) ? 1 : 0;
190 assign Mem_Read = (opcode == LW) ? 1 : 0;
191 assign Mem_Write = (opcode == SH) ? 1 : 0;
192 assign Mem_to_Reg = (opcode == AL || opcode == ADDI || opcode == SLII) ? 2'b01 : (opcode == JAL) ? 2'b10 : 2'b00;
193
194 Single_Cycle_MIPS_ALU_Controller(ALU_Op, ALU_Src, func, ALU_source, ALU_operation);
195
196 endmodule
197
198 module Single_Cycle_MIPS(input clk, reset, instruction_Write_en, input [31 : 0]Write_address, Write_instruction, output [31 : 0]ALU_Result, Register_file_Write_Data);
199
200 wire Reg_Write, ALU_source, PC_Source, branch, Mem_Read, Mem_Write;
201 wire [1 : 0]Mem_to_Reg, jump_src, Reg_Dst;
202 wire [2 : 0]ALU_operation;
203 wire [31 : 0]instruction;
204 Single_Cycle_MIPS_DataPath DataPath(clk, reset, instruction_Write_en, Reg_Write, Mem_Read, Mem_Write, ALU_source, PC_Source, branch, Mem_to_Reg, jump_src, Reg_Dst, ALU_operation, Writ
205 Single_Cycle_MIPS_CU(instruction[31 : 27], instruction[5 : 0], Reg_Write, ALU_source, ALU_operation, Mem_to_Reg, jump_src, Reg_Dst, Mem_Read, Mem_Write, PC_Source, branch);
206
207 endmodule

```

Ln: 88 Col: 0

ModelSim - INTE... D:/modelsim_ase... D:/modelsim_ase... Document1 - Mic... 10:15 PM 5/8/2022

Single cycle MIPS's Data Path



```

D:/modelsim_ase/Projects/CA_Single_Cycle_MIPS.v (/Single_Cycle_MIPS_Test_Bench/single_cycle_MIPS/DataPath/alu)
File Edit View Tools Bookmarks Window Help
D:/modelsim_ase/Projects/CA_Single_Cycle_MIPS.v (Single_Cycle_MIPS_Test_Bench/single_cycle_MIPS/DataPath/alu) - Default
Ln#
118
119 endmodule
120
121 module Sign_Extension_26_bit(input [25 : 0]_input, output [31 : 0] sign_extended_input);
122 assign sign_extended_input = (_input[15], _input[15], _input[15], _input[15], _input[15], _input[15], _input);
123
124 endmodule
125
126 module Single_Cycle_MIPS_DataPath(input clk, reset, instruction_Write_en, Reg_Write, Mem_Read, Mem_Write, ALU_source, PC_source, branch, input[1 : 0]Mem_to_Reg, jump_src, Reg_Dst, in
127
128 wire zero_flag;
129 wire [4 : 0] Register_file_Write_Register;
130 wire [31 : 0] PC_input, PC_output, Mem_Read_Data, Register_file_Read_Data1, Register_file_Read_Data2, jump, jump_address,
131 sign_extended_of_instruction_15_0, sign_extended_of_instruction_20_0, sign_extended_of_instruction_25_0, ALU_B_Input;
132
133 instruction_MEMORY instruction_memory(clk, instruction_Write_en, Write_address, PC_output, Write_instruction, instruction);
134
135 assign Register_file_Write_Register = (Reg_Dst == 2'b10) ? instruction[15 : 11] : (Reg_Dst == 2'b01) ? instruction[20 : 16] : instruction[25 : 21];
136
137 Sign_Extension_16_bit sign_extension_16_bit(instruction[15 : 0], sign_extended_of_instruction_15_0);
138 Sign_Extension_21_bit sign_extension_21_bit(instruction[20 : 0], sign_extended_of_instruction_20_0);
139 Sign_Extension_26_bit sign_extension_26_bit(instruction[25 : 0], sign_extended_of_instruction_25_0);
140
141 assign jump = (jump_src == 2'b00) ? sign_extended_of_instruction_25_0 : (jump_src == 2'b01) ? sign_extended_of_instruction_20_0 : Register_file_Read_Data1;
142 assign jump_address = (branch & zero_flag) ? sign_extended_of_instruction_15_0 : jump;
143 assign PC_input = PC_source ? jump_address : PC_output + 1;
144 Register_32_bit PC(clk, reset, 1, PC_input, PC_output);
145
146 assign ALU_B_Input = ALU_source ? sign_extended_of_instruction_15_0 : Register_file_Read_Data2;
147 ALU alu(ALU_operation, Register_file_Read_Data1, ALU_B_Input, ALU_Result, zero_flag);
148
149 Memory_data_memory(clk, reset, Mem_Read, Mem_Write, ALU_Result, Register_file_Read_Data2, Mem_Read_Data);
150
151 assign Register_file_Write_Data = (Mem_to_Reg == 2'b00) ? Mem_Read_Data : (Mem_to_Reg == 2'b01) ? ALU_Result : PC_output + 1;
152 Register_File register_file(clk, reset, Reg_Write, instruction[25 : 21], Register_file_Write_Register, Register_file_Write_Data, Register_file_Read_Data1, Registr
153
154 endmodule

```

Ln: 88 Col: 0

ModelSim - INTE... D:/modelsim_ase... D:/modelsim_ase... CA_CA2_Report... Untitled - Paint 10:18 PM 5/8/2022

Single cycle MIPS's CU (main controller)

```

D:/modelsim_ase/Projects/CA_Single_Cycle_MIPS.v (/Single_Cycle_MIPS_Test_Bench/single_cycle_MIPS/DataPath/alu)
File Edit View Tools Bookmarks Window Help
D:/modelsim_ase/Projects/CA_Single_Cycle_MIPS.v (/Single_Cycle_MIPS_Test_Bench/single_cycle_MIPS/DataPath/alu) - Default
Ln#
159 parameter ADD = 6'b000000, SUB = 6'b000001, Little_Than = 6'b000010, AND = 6'b000100, OR = 6'b000101, SLT = 6'b000011;
160
161 assign ALU_operation = (ALU_Op == 2'b00) ? ADD :
162     (ALU_Op == 2'b01) ? SUB :
163     (ALU_Op == 2'b11) ? Little_Than :
164     (func == ADD) ? ADD :
165     (func == SUB) ? SUB :
166     (func == SLT) ? Little_Than :
167     (func == AND) ? AND :
168     (func == OR) ? OR:
169     3'bz;
170
171 assign ALU_source = (ALU_Op == 2'b00 || ALU_Op == 2'b11) ? 1 : 0;
172
173 endmodule
174
175 module Single_Cycle_MIPS CU(input[4 : 0]opcode, input [5 : 0]func, output Reg_Write, ALU_source, output [2 : 0]ALU_operation, output [1 : 0]Mem_to_Reg, jump_src, Reg_Dst, output Mem_F
176
177 parameter AL = 5'b00000, LW = 5'b00001, SW = 5'b00010, ADDI = 5'b00011, SLTI = 5'b00100, J = 5'b00101, JAL = 5'b00110, JR = 5'b00111, Beq = 5'b01000,
178     Read_Data1_of_register_file = 2'b00, sign_extended_of_15_0_of_instruction = 2'b01,
179     Func = 2'b10, ADD = 2'b00, Little_Than = 2'b11, instruction_15_11 = 2'b10, instruction_20_16 = 2'b01, instruction_25_21 = 2'b00;
180 wire [1 : 0]ALU_Op;
181 wire ALU_Src;
182
183 assign Reg_Dst = (opcode == AL) ? instruction_15_11 : (opcode == JAL)? instruction_25_21 : instruction_20_16;
184 assign Reg_Write = (opcode == AL || opcode == LW || opcode == ADDI || opcode == SLTI || opcode == JAL) ? 1 : 0;
185 assign ALU_Op = (opcode == AL) ? Func : (opcode == SLTI) ? Little_Than : ADD;
186 assign ALU_Src = (opcode == AL) ? Read_Data1_of_register_file : sign_extended_of_15_0_of_instruction;
187 assign branch = (opcode == Beq) ? 1 : 0;
188 assign jump_src = (opcode == J) ? 2'b00 : (opcode == JAL) ? 2'b01 : 2'b10;
189 assign PC_source = (opcode == J || opcode == JAL || opcode == JR || opcode == Beq) ? 1 : 0;
190 assign Mem_Read = (opcode == LW) ? 1 : 0;
191 assign Mem_Write = (opcode == SW) ? 1 : 0;
192 assign Mem_to_Reg = (opcode == AL || opcode == ADDI || opcode == SLTI) ? 2'b01 : (opcode == JAL) ? 2'b10 : 2'b00;
193
194 Single_Cycle_MIPS_ALU_Controller ALU_Controller(ALU_Op, ALU_Src, func, ALU_source, ALU_operation);
195
196 endmodule

```

Ln: 88 Col: 0

10:20 PM
5/8/2022

Single cycle MIPS's ALU controller

```

D:/modelsim_ase/Projects/CA_Single_Cycle_MIPS.v (/Single_Cycle_MIPS_Test_Bench/single_cycle_MIPS/DataPath/alu)
File Edit View Tools Bookmarks Window Help
D:/modelsim_ase/Projects/CA_Single_Cycle_MIPS.v (/Single_Cycle_MIPS_Test_Bench/single_cycle_MIPS/DataPath/alu) - Default
Ln#
136 assign Register_file_Write_Register = (Reg_Dst == 2'b10) ? instruction[15 : 11] : (Reg_Dst == 2'b01) ? instruction[20 : 16] : instruction[25 : 21];
137
138 Sign_Extension_16_bit sign_extension_16_bit(instruction[15 : 0], sign_extended_of_instruction_15_0);
139 Sign_Extension_21_bit sign_extension_21_bit(instruction[20 : 0], sign_extended_of_instruction_20_0);
140 Sign_Extension_26_bit sign_extension_26_bit(instruction[25 : 0], sign_extended_of_instruction_25_0);
141
142 assign jump = (jump_src == 2'b00) ? sign_extended_of_instruction_25_0 : (jump_src == 2'b01) ? sign_extended_of_instruction_20_0 : Register_file_Read_Data;
143 assign jump_address = (branch & zero_flag) ? sign_extended_of_instruction_15_0 : jump;
144 assign PC_input = PC_source ? jump_address : PC_output + 1;
145 Register_32_bit PC(clk, reset, 1, PC_input, PC_output);
146
147 assign ALU_B_Input = ALU_source ? sign_extended_of_instruction_15_0 : Register_file_Read_Data2;
148 ALU alu(ALU_Operation, Register_file_Read_Data, ALU_B_Input, ALU_Result, zero_flag);
149
150 Memory data_memory(clk, reset, Mem_Read, Mem_Write, ALU_Result, Register_file_Read_Data2, Mem_Read_Data);
151
152 assign Register_file_Write_Data = (Mem_to_Reg == 2'b00) ? Mem_Read_Data : (Mem_to_Reg == 2'b01) ? ALU_Result : PC_output + 1;
153 Register_File register_file(clk, reset, Reg_Write, instruction[25 : 21], instruction[20 : 16], Register_file_Write_Register, Register_file_Write_Data, Register_file_Read_Data, Register_file_Read_Data2, Register_file_Read_Data3);
154
155
156 endmodule
157
158 module Single_Cycle_MIPS_ALU_Controller(input [1 : 0]ALU_Op, input ALU_Src, input [5 : 0]func, output ALU_source, output [2 : 0]ALU_operation);
159
160 parameter ADD = 6'b000000, SUB = 6'b000001, Little_Than = 6'b000010, AND = 6'b000100, OR = 6'b000101, SLT = 6'b000011;
161
162 assign ALU_operation = (ALU_Op == 2'b00) ? ADD :
163     (ALU_Op == 2'b01) ? SUB :
164     (ALU_Op == 2'b11) ? Little_Than :
165     (func == ADD) ? ADD :
166     (func == SUB) ? SUB :
167     (func == SLT) ? Little_Than :
168     (func == AND) ? AND :
169     (func == OR) ? OR:
170     3'bz;
171
172 assign ALU_source = (ALU_Op == 2'b00 || ALU_Op == 2'b11) ? 1 : 0;
173
174 endmodule

```

Ln: 88 Col: 0

10:22 PM
5/8/2022

Single cycle MIPS's ALU module

Single cycle MIPS's Register file module

D:/modelsim_ase/Projects/CA_Single_Cycle_MIPS.v (/Single_Cycle_MIPS_Test_Bench/single_cycle_MIPS/DataPath/alu)

File Edit View Tools Bookmarks Window Help

D:/modelsim_ase/Projects/CA_Single_Cycle_MIPS.v (/Single_Cycle_MIPS_Test_Bench/single_cycle_MIPS/DataPath/alu) - Default

Ln#

```
56 always @ (address, Mem_Write) begin
57   for (1 = 0; i < Memory_Size; i = i + 1) begin
58     load[i] = 1'b0;
59   end
60   if (Mem_Write)
61     load[address] = 1'b1;
62   end
63
64   assign Read_Data = Mem_Read ? register[address] : 32'b0;
65
66 endmodule
67
68 module Register_File (input clk, reset, Reg_Write, input [4 : 0] Read_Register1, Read_Register2, Write_Register, input [31 : 0] Write_Data, output [31 : 0] Read_Data1, Read_Data2);
69
70   parameter Register_File_Size = 32;
71   wire [31 : 0] register [Register_File_Size - 1 : 0];
72   reg [Register_File_Size - 1 : 0] load;
73
74   genvar k;
75   generate
76     for (k = 0; k < Register_File_Size; k = k + 1) begin
77       Register_32_bit xx (clk, reset, load[k], Write_Data, register[k]);
78     end
79   endgenerate
80
81   reg [5 : 0];
82   always @ (Write_Register, Reg_Write) begin
83     for (1 = 0; i < Register_File_Size; i = i + 1) begin
84       load[i] = 1'b0;
85     end
86     if (Reg_Write)
87       load[Write_Register] = 1'b1;
88     end
89
90   assign Read_Data1 = register[Read_Register1];
91   assign Read_Data2 = register[Read_Register2];
92
93 endmodule
```



Ln: 88 Col: 0

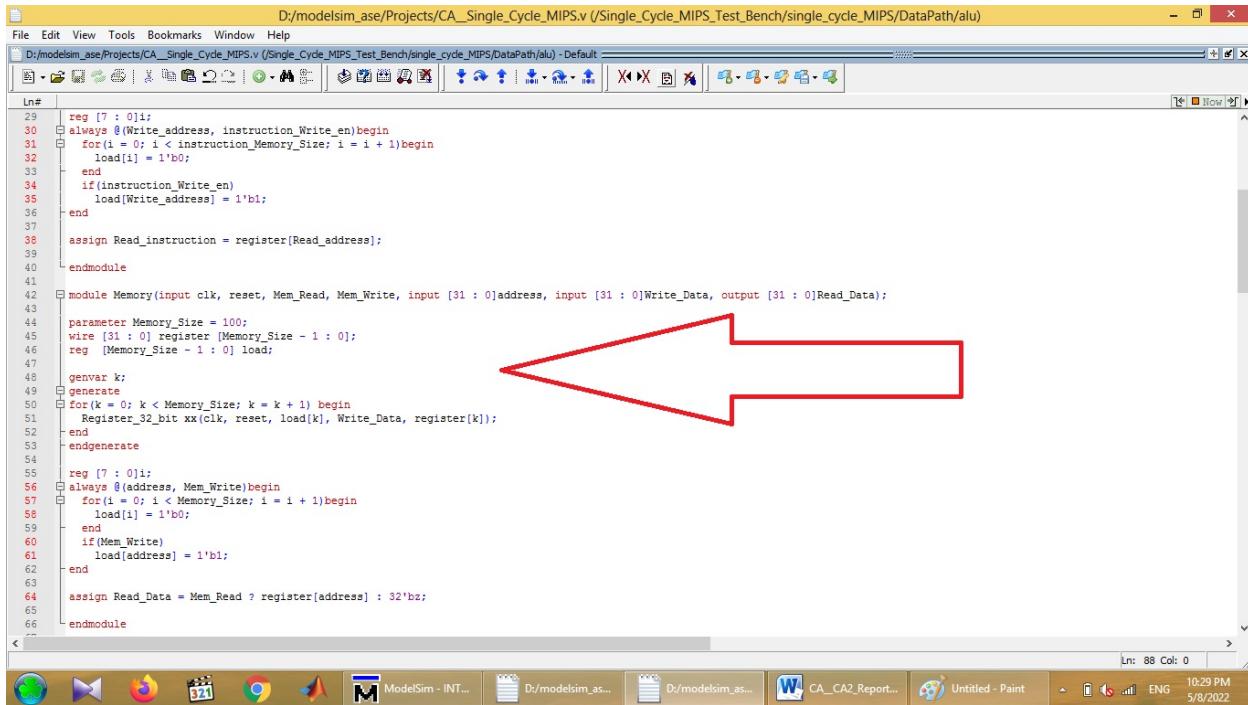
File Edit View Tools Bookmarks Window Help

D:/modelsim_ase/Projects/CA_Single_Cycle_MIPS.v (/Single_Cycle_MIPS_Test_Bench/single_cycle_MIPS/DataPath/alu) - Default

ModelSim - INT... D:/modelsim_as... D:/modelsim_as... CA_CA2_Report... Untitled - Paint

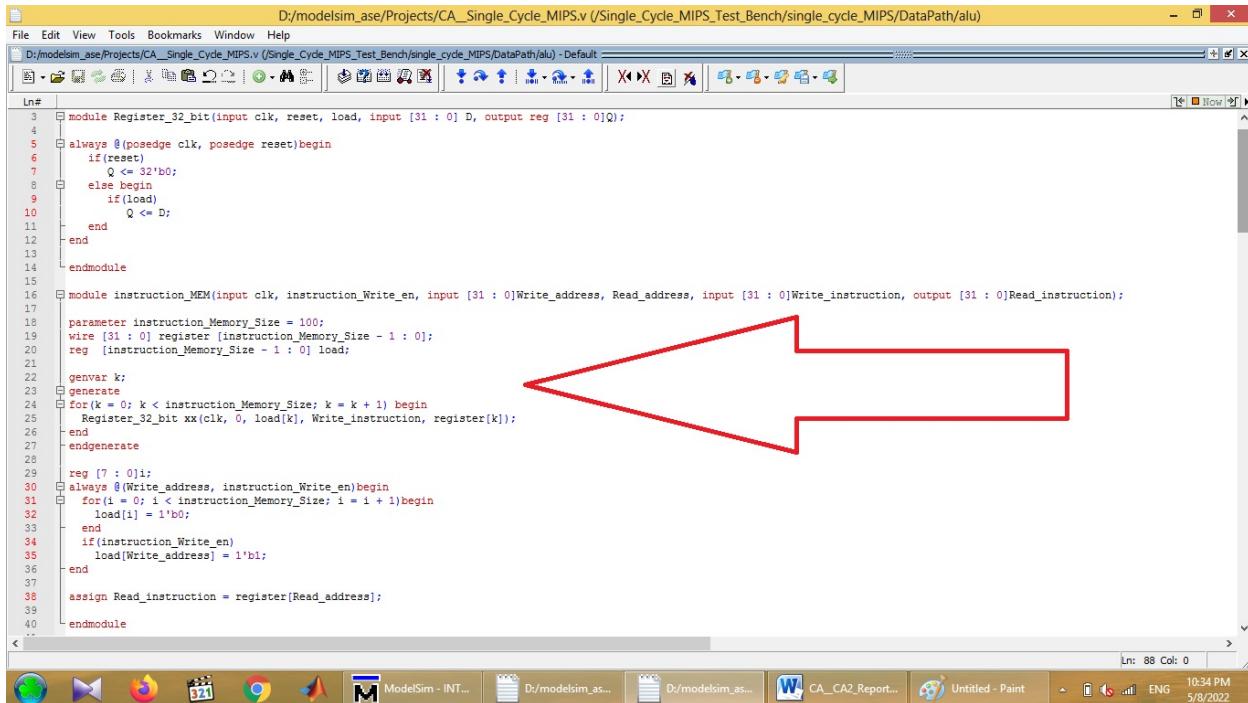
10:26 PM 5/8/2022

Single cycle MIPS's Memory module



```
D:/modelsim_ase/Projects/CA_Single_Cycle_MIPS.v (/Single_Cycle_MIPS_Test_Bench/single_cycle_MIPS/DataPath/alu)
File Edit View Tools Bookmarks Window Help
D:/modelsim_ase/Projects/CA_Single_Cycle_MIPS.v (/Single_Cycle_MIPS_Test_Bench/single_cycle_MIPS/DataPath/alu) - Default
Ln#
29 reg [7 : 0]i;
30 always @ (Write_address, instruction_Write_en)begin
31   for(i = 0; i < instruction_Memory_Size; i = i + 1)begin
32     load[i] = 1'b0;
33   end
34   if(instruction_Write_en)
35     load[Write_address] = 1'b1;
36 end
37
38 assign Read_instruction = register[Read_address];
39
endmodule
41
42 module Memory(input clk, reset, Mem_Read, Mem_Write, input [31 : 0]address, input [31 : 0]Write_Data, output [31 : 0]Read_Data);
43
parameter Memory_Size = 100;
wire [31 : 0] register [Memory_Size - 1 : 0];
reg [Memory_Size - 1 : 0] load;
47
genvar k;
48 generate
50 for(k = 0; k < Memory_Size; k = k + 1) begin
51   Register_32_bit xx(clk, reset, load[k], Write_Data, register[k]);
52 end
53 endgenerate
54
55 reg [7 : 0]i;
56 always @ (address, Mem_Write)begin
57   for(i = 0; i < Memory_Size; i = i + 1)begin
58     load[i] = 1'b0;
59   end
60   if(Mem_Write)
61     load[address] = 1'b1;
62 end
63
64 assign Read_Data = Mem_Read ? register[address] : 32'bz;
65
66 endmodule
< ...
Ln: 88 Col: 0
10:29 PM 5/8/2022
```

Single cycle MIPS's Instruction Memory module



```
D:/modelsim_ase/Projects/CA_Single_Cycle_MIPS.v (/Single_Cycle_MIPS_Test_Bench/single_cycle_MIPS/DataPath/alu)
File Edit View Tools Bookmarks Window Help
D:/modelsim_ase/Projects/CA_Single_Cycle_MIPS.v (/Single_Cycle_MIPS_Test_Bench/single_cycle_MIPS/DataPath/alu) - Default
Ln#
3 module Register_32_bit(input clk, reset, load, input [31 : 0] D, output reg [31 : 0]Q);
4
5 always @(posedge clk, posedge reset)begin
6   if(reset)
7     Q <= 32'b0;
8   else begin
9     if(load)
10       Q <= D;
11   end
12 end
13
14 endmodule
15
16 module instruction_MEMORY(input clk, instruction_Write_en, input [31 : 0]Write_address, Read_address, input [31 : 0]Write_instruction, output [31 : 0]Read_instruction);
17
parameter instruction_Memory_Size = 100;
wire [31 : 0] register [instruction_Memory_Size - 1 : 0];
reg [instruction_Memory_Size - 1 : 0] load;
19
genvar k;
20 generate
21 for(k = 0; k < instruction_Memory_Size; k = k + 1) begin
22   Register_32_bit xx(clk, 0, load[k], Write_instruction, register[k]);
23 end
24 endgenerate
25
26 reg [7 : 0]i;
27 always @ (Write_address, instruction_Write_en)begin
28   for(i = 0; i < instruction_Memory_Size; i = i + 1)begin
29     load[i] = 1'b0;
30   end
31   if(instruction_Write_en)
32     load[Write_address] = 1'b1;
33 end
34
35 assign Read_instruction = register[Read_address];
36
37
38 endmodule
< ...
Ln: 88 Col: 0
10:34 PM 5/8/2022
```

Single cycle MIPS's Sign Extensions module

Single cycle MIPS's Register module

D:/modelsim_ase/Projects/CA_Single_Cycle_MIPS.v (/Single_Cycle_MIPS_Test_Bench/single_cycle_MIPS/DataPath/alu) - Default

```
File Edit View Tools Bookmarks Window Help
D:/modelsim_ase/Projects/CA_Single_Cycle_MIPS.v (/Single_Cycle_MIPS_Test_Bench/single_cycle_MIPS/DataPath/alu) - Default
Ln# 1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38

`timescale 1ns/1ns

module Register_32_bit(input clk, reset, load, input [31 : 0] D, output reg [31 : 0]Q);
always @ (posedge clk, posedge reset)begin
    if(reset)
        Q <= 32'b0;
    else begin
        if(load)
            Q <= D;
    end
end
endmodule

module instruction_MEMORY(input clk, instruction_Write_en, input [31 : 0]Write_address, Read_address, input [31 : 0]Write_instruction, output [31 : 0]Read_instruction);
parameter instruction_Memory_Size = 100;
wire [31 : 0] register [instruction_Memory_Size - 1 : 0];
reg [Instruction_Memory_Size - 1 : 0] load;
genvar k;
generate
for(k = 0; k < instruction_Memory_Size; k = k + 1) begin
    Register_32_bit xx(clk, 0, load[k], Write_instruction, register[k]);
end
endgenerate

reg [7 : 0]i;
always @ (Write_address, instruction_Write_en)begin
    for(i = 0; i < instruction_Memory_Size; i = i + 1)begin
        load[i] = 1'b0;
    end
    if(instruction_Write_en)
        load[Write_address] = 1'b1;
end
assign Read_instruction = register[Read_address];
```

Single cycle MIPS's Test Bench

D:/modelsim_ase/Projects/CA_Single_Cycle_MIPS_TestBench.v (/Single_Cycle_MIPS_Test_Bench)

```
File Edit View Tools Bookmarks Window Help
D:/modelsim_ase/Projects/CA_Single_Cycle_MIPS_TestBench.v (Single_Cycle_MIPS_Test_Bench) - Default
Ln# 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39

module Single_Cycle_MIPS_Test_Bench();
reg clk, reset, instruction_Write_en;
reg [31 : 0]Write_address, Write_instruction;
wire [31 : 0]ALU_Result, Register_file_Write_Data;

Single_Cycle_MIPS single_cycle_MIPS(clk, reset, instruction_Write_en, Write_address, Write_instruction, ALU_Result, Register_file_Write_Data);

always #50 clk = ~clk;
initial begin
    clk = 0; reset = 0; instruction_Write_en = 1; Write_address = 0;

    #100 Write_instruction = 32'b000110000000000010000000000000110; // ADDI R1, R0, 6
    #100 Write_address = Write_address + 1;
    #100 Write_instruction = 32'b000110000000000010000000000000111; // ADDI R2, R0, 7
    #100 Write_address = Write_address + 1;
    #100 Write_instruction = 32'b0001100000000000110000000000001000; // ADDI R3, R0, 8
    #100 Write_address = Write_address + 1;
    #100 Write_instruction = 32'b0001100000000000100000000000001001; // ADDI R4, R0, 9
    #100 Write_address = Write_address + 1;
    #100 Write_instruction = 32'b00011000000000001010000000000001010; // ADDI R5, R0, 10
    #100 Write_address = Write_address + 1;
    #100 Write_instruction = 32'b00011000000000001100000000000001011; // ADDI R6, R0, 11
    #100 Write_address = Write_address + 1;
    #100 Write_instruction = 32'b00011000000000001110000000000001100; // ADDI R7, R0, 12
    #100 Write_address = Write_address + 1;
    #100 Write_instruction = 32'b00011000000000001000000000000001101; // ADDI R8, R0, 13
    #100 Write_address = Write_address + 1;
    #100 Write_instruction = 32'b00011000000000001010000000000001110; // ADDI R9, R0, 14
    #100 Write_address = Write_address + 1;
    #100 Write_instruction = 32'b00011000000000001011000000000001111; // ADDI R10, R0, 15
    #100 Write_address = Write_address + 1;
    #100 Write_instruction = 32'b000110000000000010110000000000010000; // ADDI R11, R0, 16
    #100 Write_address = Write_address + 1;
    #100 Write_instruction = 32'b000110000000000011000000000000010001; // ADDI R12, R0, 17
    #100 Write_address = Write_address + 1;

```

D:/modelsim_ase/Projects/CA_Single_Cycle_MIPS_TestBench.v (/Single_Cycle_MIPS_Test_Bench)

File Edit View Tools Bookmarks Window Help

D:/modelsim_ase/Projects/CA_Single_Cycle_MIPS_TestBench.v (/Single_Cycle_MIPS_Test_Bench) - Default

Ln#

```
38 #1000 Write_instruction = 32'b00011000000011000000000000000010001; // ADDI R12, R0, 17
39 #1000 Write_address = Write_address + 1;
40 #1000 Write_instruction = 32'b000000000000100011011111000000; // ADD R13, R1, R2
41 #1000 Write_address = Write_address + 1;
42 #1000 Write_instruction = 32'b00000000100000110111011111000001; // SUB R14, R4, R3
43 #1000 Write_address = Write_address + 1;
44 #1000 Write_instruction = 32'b0000000010100110011111111000011; // SLT R15, R5, R6
45 #1000 Write_address = Write_address + 1;
46 #1000 Write_instruction = 32'b0000000011010001000011111000100; // AND R16, R7, R8
47 #1000 Write_address = Write_address + 1;
48 #1000 Write_instruction = 32'b0000000011010101000111111000101; // OR R17, R9, R10
49 #1000 Write_address = Write_address + 1;
50 #1000 Write_instruction = 32'b00000000101100010000000000001010; // SLTI R18, R11, 10
51 #1000 Write_address = Write_address + 1;
52 #1000 Write_instruction = 32'b00010000101100010000000000001000; // SW R5, R17, 80 R5 = 10
53 #1000 Write_address = Write_address + 1;
54 #1000 Write_instruction = 32'b000010001011000100000000001010000; // IW R5, R19, 80 R5 = 10
55 #1000 Write_address = Write_address + 1;
56 #1000 Write_instruction = 32'b00101000000000000000000000111100; // J 60
57 #1000 Write_address = Write_address + 1;
58 #1000 Write_instruction = 32'b0001100000001010000000000000110001; // ADDI R20, R0, 25
59 #1000 Write_address = 60;
60 #1000 Write_instruction = 32'b0001100000001010000000000000010111; // ADDI R20, R0, 23
61 #1000 Write_address = Write_address + 1;
62 #1000 Write_instruction = 32'b0011001010100000000000000000110010; // JAL R21, 50
63 #1000 Write_address = Write_address + 1;
64 #1000 Write_instruction = 32'b00011000000010110000000000000011010; // ADDI R22, R0, 26
65 #1000 Write_address = 50;
66 #1000 Write_instruction = 32'b0001100000001011000000000000011000; // ADDI R22, R0, 24
67 #1000 Write_address = Write_address + 1;
68 #1000 Write_instruction = 32'b0001100000001011000000000000101000; // ADDI R23, R0, 40
69 #1000 Write_address = Write_address + 1;
70 #1000 Write_instruction = 32'b0011101110000000000000000000000000; // JR R23
71 #1000 Write_address = Write_address + 1;
72 #1000 Write_instruction = 32'b00011000000011000000000000000100000; // ADDI R24, R0, 32
73 #1000 Write_address = 40;
74 #1000 Write_instruction = 32'b00011000000011000000000000000101011; // ADDI R24, R0, 43
75
```

D:/modelsim_ase/Projects/CA_Sinlge_Cycle_MIPS_TestBench.v (/Single_Cycle_MIPS_Test_Bench)

```

File Edit View Tools Bookmarks Window Help
D:/modelsim_ase/Projects/CA_Sinlge_Cycle_MIPS_TestBench.v (/Single_Cycle_MIPS_Test_Bench) - Default
Ln# 47
48 #1000 Write_address = Write_address + 1;
49 #1000 Write_instruction = 32'b00000001001010100011111000101; // OR R17, R9, R10
50 #1000 Write_address = Write_address + 1;
51 #1000 Write_instruction = 32'b00100001011000100000000000001010; // SLTI R18, R11, 10
52 #1000 Write_address = Write_address + 1;
53 #1000 Write_instruction = 32'b00010000101100010000000000001000; // SW R5, R17, 80 R5 = 10
54 #1000 Write_address = Write_address + 1;
55 #1000 Write_instruction = 32'b00001000101100110000000000001010000; // LW R5, R19, 80 R5 = 10
56 #1000 Write_address = Write_address + 1;
57 #1000 Write_instruction = 32'b0010100000000000000000000000111100; // J 60
58 #1000 Write_address = Write_address + 1;
59 #1000 Write_instruction = 32'b000110000010100000000000000011001; // ADDI R20, R0, 25
60 #1000 Write_address = 60;
61 #1000 Write_instruction = 32'b000110000010100000000000000010111; // ADDI R20, R0, 23
62 #1000 Write_address = Write_address + 1;
63 #1000 Write_instruction = 32'b0011001010101000000000000000110010; // JAL R21, 50
64 #1000 Write_address = Write_address + 1;
65 #1000 Write_instruction = 32'b000110000010100000000000000011010; // ADDI R22, R0, 26
66 #1000 Write_address = 50;
67 #1000 Write_instruction = 32'b000110000010100000000000000011000; // ADDI R22, R0, 24
68 #1000 Write_address = Write_address + 1;
69 #1000 Write_instruction = 32'b0001100000101110000000000000101000; // ADDI R23, R0, 40
70 #1000 Write_address = Write_address + 1;
71 #1000 Write_instruction = 32'b0011010110000000000000000000000000; // JR R23
72 #1000 Write_address = Write_address + 1;
73 #1000 Write_instruction = 32'b0001100000110000000000000000100000; // ADDI R24, R0, 32
74 #1000 Write_address = 40;
75 #1000 Write_instruction = 32'b0001100000110000000000000000101011; // ADDI R24, R0, 43
76 #1000 reset = 1; instruction_Write_en = 0;
77 #200 reset = 0;
78
79 #7000 $stop;
80
81 end
82
83 endmodule
84

```

Ln: 45 Col: 0

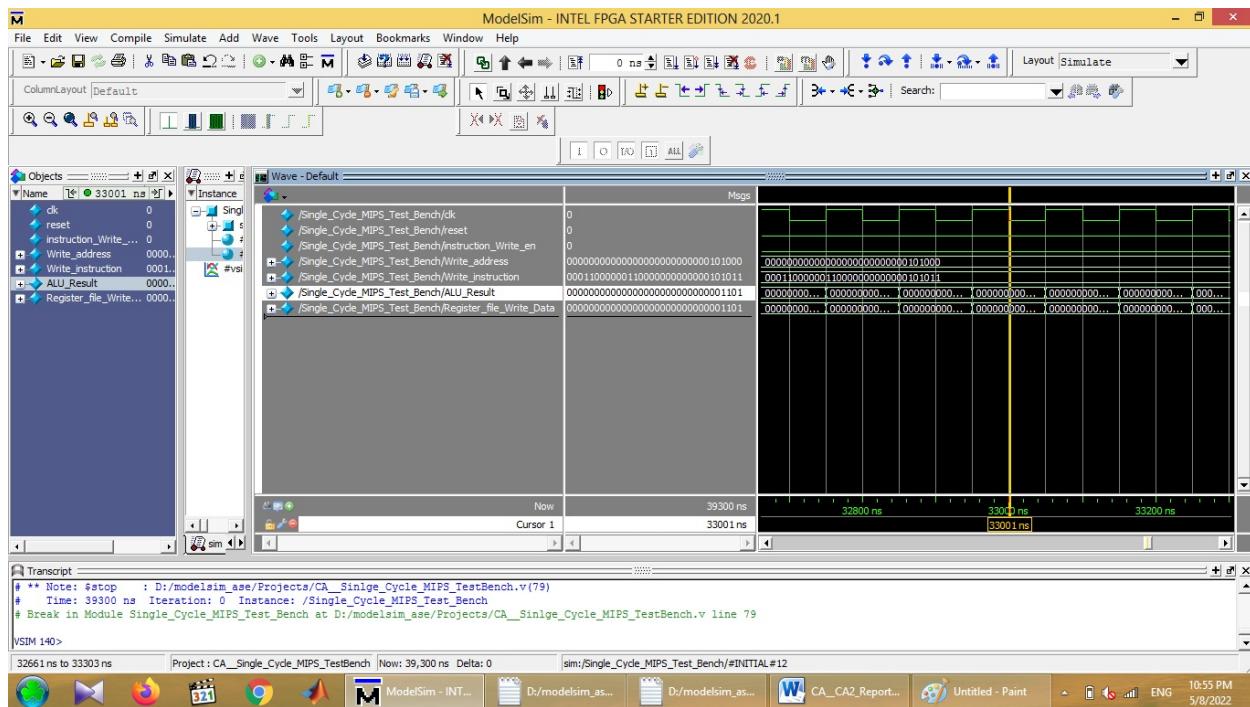
File Edit View Tools Bookmarks Window Help

D:/modelsim_ase/Projects/CA_Sinlge_Cycle_MIPS_TestBench.v (/Single_Cycle_MIPS_Test_Bench) - Default

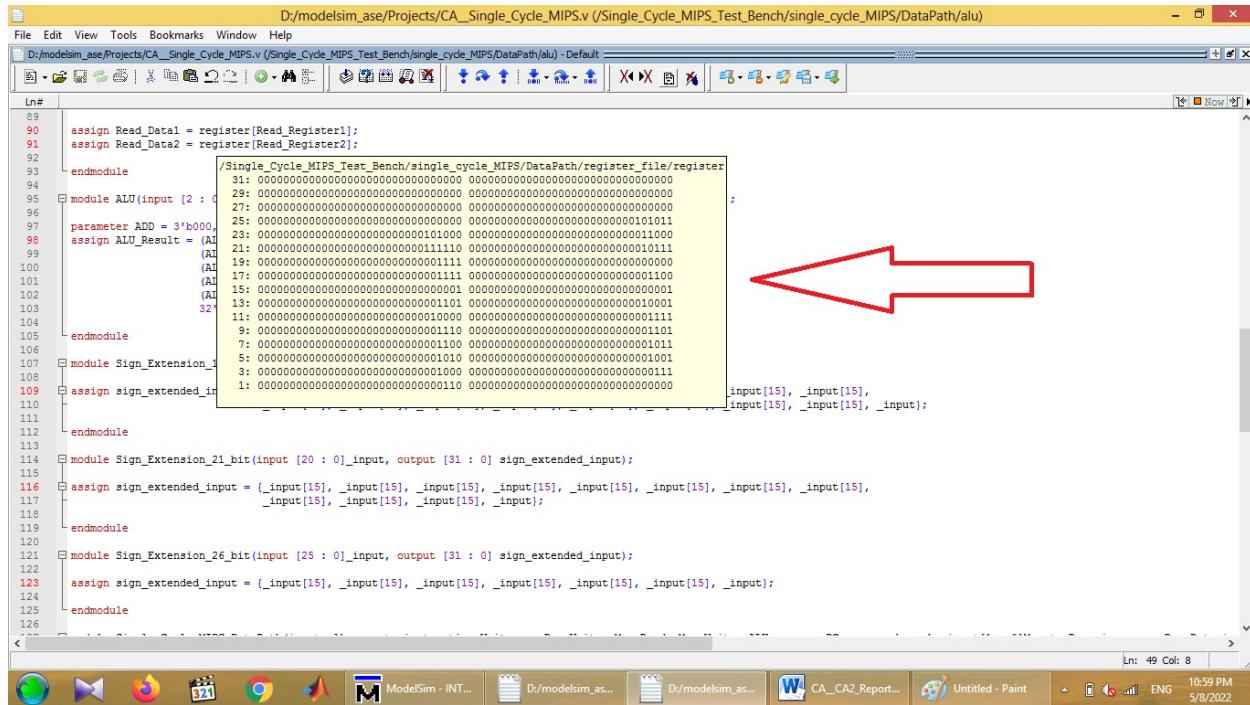
ModelSim - INT... D:/modelsim_ase... D:/modelsim_ase... CA_CA2_Report... Untitled - Paint

10:54 PM 5/8/2022

Simulation output of Single cycle MIPS



Register file after execution of instructions



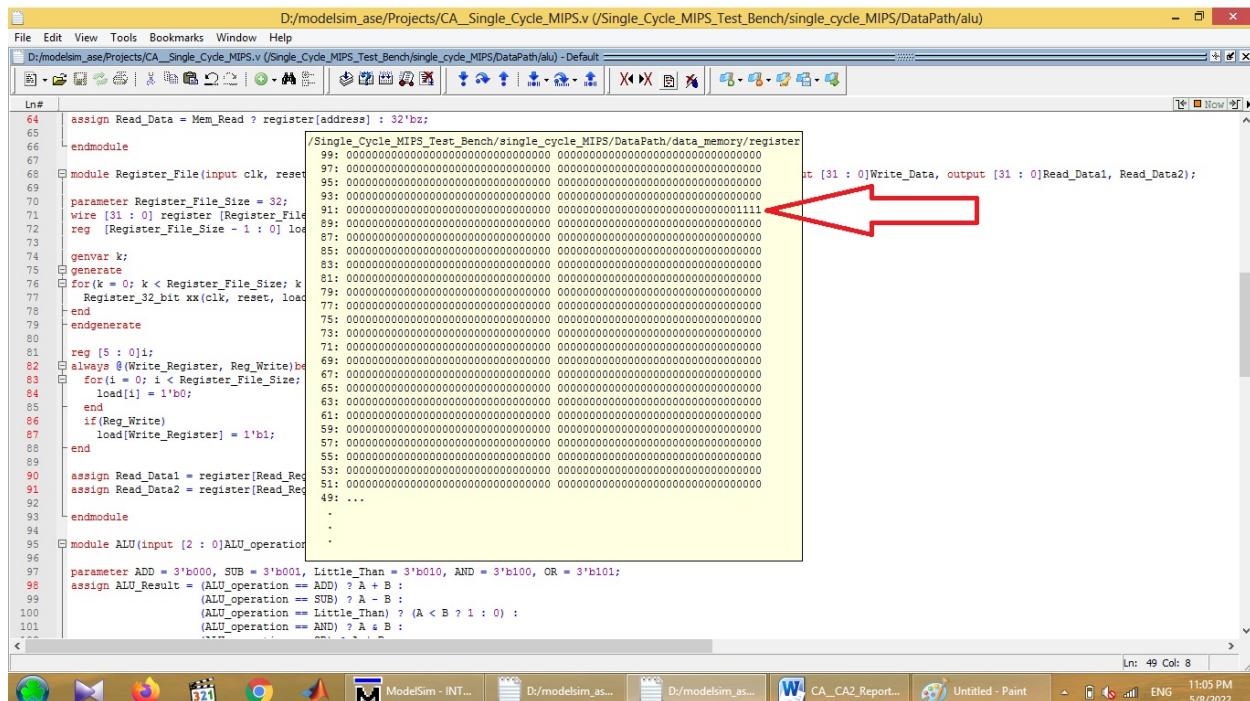
```

D:/modelsim_ase/Projects/CA_Single_Cycle_MIPS.v (/Single_Cycle_MIPS_Test_Bench/single_cycle_MIPS/DataPath/alu)
File Edit View Tools Bookmarks Window Help
D:/modelsim_ase/Projects/CA_Single_Cycle_MIPS.v (/Single_Cycle_MIPS_Test_Bench/single_cycle_MIPS/DataPath/alu) - Default
Ln# 69
69 assign Read_Data1 = register[Read_Register1];
70 assign Read_Data2 = register[Read_Register2];
71
72 endmodule
73
74 module ALU(input [2 : 0]ALU_operation;
75 parameter ADD = 3'b000, Little_Than = 3'b010, AND = 3'b100, OR = 3'b101;
76 assign ALU_Result = (ALU_operation == ADD) ? A + B :
77 (ALU_operation == SUB) ? A - B :
78 (ALU_operation == Little_Than) ? (A < B ? 1 : 0) :
79 (ALU_operation == AND) ? A & B :
80 (ALU_operation == OR) ? A | B : 0;
81
82 endmodule
83
84 module Sign_Extension_1_bit(input [15 : 0]input, output [31 : 0]sign_extended_input);
85 assign sign_extended_input = input[15], input[15];
86
87 endmodule
88
89 module Sign_Extension_21_bit(input [20 : 0]input, output [31 : 0]sign_extended_input);
90 assign sign_extended_input = _input[15], _input[15];
91
92 endmodule
93
94 module Sign_Extension_26_bit(input [25 : 0]input, output [31 : 0]sign_extended_input);
95 assign sign_extended_input = _input[15], _input[15];
96
97 endmodule
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126

```

Ln: 49 Col: 8 10:59 PM 5/8/2022

Data Memory after execution of instructions



```

D:/modelsim_ase/Projects/CA_Single_Cycle_MIPS.v (/Single_Cycle_MIPS_Test_Bench/single_cycle_MIPS/DataPath/alu)
File Edit View Tools Bookmarks Window Help
D:/modelsim_ase/Projects/CA_Single_Cycle_MIPS.v (/Single_Cycle_MIPS_Test_Bench/single_cycle_MIPS/DataPath/alu) - Default
Ln# 64
64 assign Read_Data = Mem_Read ? register[address] : 32'bz;
65
66 endmodule
67
68 module Register_File(input clk, reset);
69 parameter Register_File_Size = 32;
70 wire [31 : 0] register [Register_File_Size - 1 : 0] load;
71 reg [Register_File_Size - 1 : 0] load;
72
73 genvar k;
74 generate
75 for(k = 0; k < Register_File_Size; k)
76 Register_32_bit xx(clk, reset, load);
77 end
78 endgenerate
79
80 reg [5 : 0];
81 always @ (Write_Register, Reg_Write);
82 for(i = 0; i < Register_File_Size;
83 begin
84 load[i] = 1'b0;
85 end
86 if(Reg_Write)
87 load[Write_Register] = 1'b1;
88 end
89
90 assign Read_Data1 = register[Read_Reg1];
91 assign Read_Data2 = register[Read_Reg2];
92
93 endmodule
94
95 module ALU(input [2 : 0]ALU_operation;
96 parameter ADD = 3'b000, SUB = 3'b001, Little_Than = 3'b010, AND = 3'b100, OR = 3'b101;
97 assign ALU_Result = (ALU_operation == ADD) ? A + B :
98 (ALU_operation == SUB) ? A - B :
99 (ALU_operation == Little_Than) ? (A < B ? 1 : 0) :
100 (ALU_operation == AND) ? A & B :
101 (ALU_operation == OR) ? A | B : 0;
102
103 endmodule
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126

```

Ln: 49 Col: 8 11:05 PM 5/8/2022

Find maximum element of 20 element's array assembly Test Bench

Register file after execution of above instructions