VE370 HW3

Pan Chongdan 516370910121

1. a. jr instruction. should use $ra instead of $v0

2a. int function (int a, int b, int c)

```
int i=a+b;
if (c!=0) return i;
i=b-a;
```

3a 101

4a Executable File Header

Text size 440 Hex

Data size 90 Hex

Text Segement	Address	instruction
	00400000 Hex	lbu $a0, 8000($gp)
	00400004 Hex	jal 0.040 0000 Hex
	...	
	0040c440 Hex	sw $a1, 8090($gp)
	00400444 Hex	jal 00400440 Hex

Data Segement	Address	Symbol
	10000000 Hex	X
	10000004 Hex	B
	10000090 Hex	Y
	10000094 Hex	A

5. a Yes, the text size and data size can't exceed 440 Hex and 90 Hex

6 a. Since the text segement has address for each instruction, the addres must be within the text size, or branch and jump instruction may jump out of the allocated size and cause problems for assemblers.

Problem 7

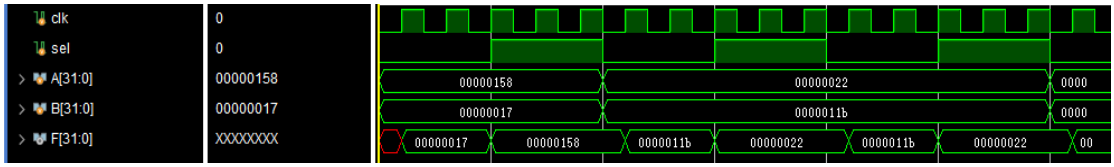
Verilog code

```
module HW3_1(clock,A,B,F,sel);
    input [31:0]A;
    input [31:0]B;
    input sel;
    input clock;
    output [31:0]F;
    reg [31:0]F;
    always @(posedge clock)begin
        if(sel==1)begin
            F<=A;
        end
        else begin
            F<=B;
        end
    end

endmodule
```

testbench

```
module HW3_1_TEST();
reg clk,sel;
reg [31:0]A;
reg [31:0]B;
wire [31:0]F;
HW3_1 UUT(clk,A,B,F,sel);
initial begin
#0 clk=0;A=344;B=23;sel=0;;
#100 A=34;B=283;
#200 A=0;B=10;
end
always #10 clk=~clk;
always #50 sel=~sel;
endmodule
```



Problem8

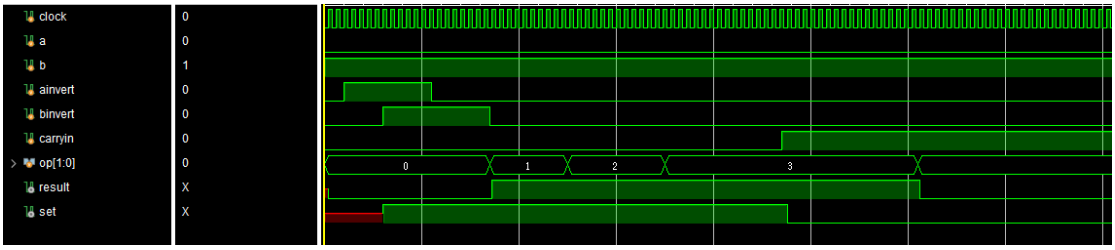
Verilog Code

```
module HW3_2(clock,a,b,ainvert,binvert,result,carryin,op,set);
input clock;
input a;
input b;
input ainvert;
input binvert;
input carryin;
input [1:0]op;
output result;
output set;
reg result,set,a1,b1;
always @(posedge clock)begin
    if(ainvert==1)a1=~a;
    if(binvert==1)b1=~b;
    set=a1+b1+carryin;
    if(op==0)result=(a&&b);
    else if (op==1)result=(a||b);
    else if (op==2)result=set;
    else if(a<b)result=1;
    else result=0;
end
endmodule
```

testbench

```
module HW3_2_TEST();
 reg clock;
 reg a;
 reg b;
 reg ainvert;
 reg binvert;
 reg carryin;
 reg [1:0]op;
 wire result,set;
 HW3_2 UUT(clock,a,b,ainvert,binvert,result,carryin,op,set);
initial begin
 #0 clock=0;ainvert=0;binvert=0;op=0;a=0;b=1;carryin=0;
 #10 ainvert=~ainvert;
 #20 binvert=~binvert;
 #25 ainvert=~ainvert;
 #30 binvert=~binvert;op=1;
 #40 op=2;
 #50 op=3;
 #60 carryin=1;
 #70 op=4;

end
 always #2 clock=~clock;
endmodule
```



Problem 9

Verilog code

```
module HW3_3(RR1, RR2, WR, WD, RD1, RD2, write);
input [4:0]RR1;
input [4:0]RR2;
input [4:0]WR;
input [31:0]WD;
input write;
output [31:0]RD1;
output [31:0]RD2;
reg [31:0]RD1;
reg [31:0]RD2;
reg [31:0]REG[31:0];
always @(*)begin
    RD1='bz;
    RD2='bz;
    if(write)REG[WR]<=WD;
    else if(!write)begin
        RD1<=REG[RR1];
        RD2<=REG[RR2];
    end
end
endmodule
```

testbench

```
module HW3_3_TEST();
reg [4:0]RR1;
reg [4:0]RR2;
reg [4:0]WR;
reg [31:0]WD;
reg write;
wire [31:0]RD1;
wire [31:0]RD2;
HW3_3 UUT(RR1, RR2, WR, WD, RD1, RD2, write);
initial begin
#0 write=1;WR=0;WD=0;RR1=0;RR2=0;
#5 WR=1;WD=7;
#10 WR=3;WD=100;
#25 write=0;RR1=1;RR2=3;WR=5;WD=9;
#30 RR2=1;RR1=3;
end
endmodule
```

1