```
\\\\main module
module alu(output reg [31:0]out=0,output reg greaterr,lesserr,equalityy,
input [2:0] opcode,input [31:0]a,b,input clk );
wire [31:0]mu,add,sub,z;
wire greater,lesser,equality;
multt m1(mu,a,b);
add a1(add,a,b,clk,0);
add a2(sub,a,b,clk,1);
comp c1(greater,lesser,equality,a,b,clk);
div d1(z,a,b,clk);
always@(posedge clk)
begin
case(opcode)
3'b000:out=add;
3'b001:out=sub;
3'b010:out=mu;
3'b011:out=z;
3'b100:{greaterr,lesserr,equalityy}={greater,lesser,equality};
endcase
if(opcode!=3'b100)
{greaterr,lesserr,equalityy}=0;
if(opcode==3'b100)
out=0;
end
endmodule
\\\ MULTIPLIER
module multt(output [31:0]mu,input [31:0]a,b);
wire [47:0]muu;
assign muu[47:0]=({1'b1,a[22:0]}) * ({1'b1,b[22:0]});
assign mu[22:0]=(muu[47]==1)?muu[46:24]:muu[45:23];
assign mu[30:23]=(muu[47]==1)?(((a[30:23]) + (b[30:23]))+(10000001)+(00000001))
:((a[30:23]) + (b[30:23]))+(10000001);
assign mu[31]=a[31]^b[31];
endmodule
\\\ADDER-SUBTRACTOR
module add(as,a,b,clk,op);
input clk,op;
input [31:0]a,b;
output reg [31:0]as=0;
reg s1,s2,s3;
reg [7:0]e1,e2,e3;
reg [22:0]m1,m2;
reg [7:0]exp_diff=0;
reg [24:0]m3=0;
```

```
reg count=0;
reg [4:0]index=23;
reg [4:0]i=23;
always@(posedge clk)
begin
if(op)
begin
s1=a[31];s2=(~b[31]);
end
else
begin
s1=a[31];s2=b[31];
end
e1=a[30:23];e2=b[30:23];
m1=a[22:0];m2=b[22:0];
if(e1==e2)
begin
e3=e1;
if(s1==s2)
begin
m3 = {1'b1,m1}+{1'b1,m2};
s3 = s1;
as[30:23]=(m3[24]==1'b1)?e3+1:e3;
as[22:0]=(m3[24]==1'b1)?m3[23:1]:m3[22:0];
end
else
begin
m3 = (m1>m2)?{1'b1,m1}-{1'b1,m2}:{1'b1,m2}-{1'b1,m1};
s3 = (m1>m2)?s1:s2;
count=0;
index=23;
for(i=23;i>0;i=i-1)
begin
if(m3[i]==1 && count==0)
begin
index=i;
count=count+1;
end
end
m3[23:0]=m3[23:0]<<(23-index);
as[22:0]=m3[22:0];
as[30:23]=e3-(23-index);
end
end
```

```
else
begin
exp_diff=(e1>e2)?(e1-e2):(e2-e1);
e3=(e1>e2)?e1:e2;
if(s1==s2)
begin
if(a[30:0]>b[30:0])
begin
m2={1'b1,m2[22:0]}>>exp_diff;
m3 = {1'b1,m1}+{1'b0,m2};
end
else
begin
m1={1'b1,m1[22:0]}>>exp_diff;
m3 = {1'b0,m1}+{1'b1,m2};
end
s3 = s1;
as[30:23]=(m3[24]==1'b1)?e3+1:e3;
as[22:0]=(m3[24]==1'b1)?m3[23:1]:m3[22:0];
end
else
begin
if(a[30:0]>b[30:0])
begin
m2={1'b1,m2[22:0]}>>exp_diff;
m3 = {1'b1,m1}-{1'b0,m2};
s3 = s1;
end
else
begin
m1={1'b1,m1[22:0]}>>exp_diff;
m3 = {1'b1,m2}-{1'b0,m1};
s3 = s2;
end
count=0;
index=23;
for(i=23;i>0;i=i-1)
begin
if(m3[i]==1 && count==0)
begin
index=i;
count=count+1;
end
end
```

```
m3[23:0]=m3[23:0]<<(23-index);
as[22:0]=m3[22:0];
as[30:23]=e3-(23-index);
end
end
as[31]=s3;
end
endmodule
\\\COMPARATOR
module comp(greater,lesser,equality,a,b,clk);
input clk;
input [31:0]a,b;
reg [31:0]compp;
reg s1,s2;
output reg greater,lesser,equality;
always@(posedge clk)
begin
greater=0;lesser=0;equality=0;
s1=a[31];s2=b[31];
if(s1==s2)
begin
compp=a[30:0]-b[30:0];
if (compp[30:0]==0)
equality=1'b1;
else
begin
if(compp[31]==1)
begin
if(s1==0 &s2==0)
lesser=1'b1;
else
greater=1'b1;
end
else
begin
if(s1==0 &s2==0)
greater=1'b1;
else
lesser=1'b1;
end
end
end
else
if(s1==1 &s2==0)
```

```
lesser=1'b1;
else
greater=1'b1;
end
endmodule
\\\\DIVISION
module div(z,a,b,clk);
input clk;
input [31:0]a,b;
output reg [31:0]z;
wire [24:0]diff1,diff2,diff3;
reg [7:0]m;
reg [22:0]count;
assign diff1={1'b1,a[22:0]}-{{1'b1,b[22:0]}*0};
assign diff2={1'b1,a[22:0]}-{{1'b1,b[22:0]}*1};
assign diff3={1'b1,a[22:0]}-{{1'b1,b[22:0]}*2};
always@(posedge clk)
begin
z[31]=a[31]^b[31];
m=a[30:23]-b[30:23];
z[30:23]=m+8'd127;
if(diff2[24]==1)
count=diff1[22:0];
else
begin
if(diff3[24]==1)
count=diff2[22:0];
else
count=0;
end
z[22:0]=count;
end
endmodule
```