Modules instantiated in second alu program:

module trans(

input [3:0] a,

output [3:0]b

);

assign b=a;

endmodule

module incre(

input [3:0] a,

output [3:0] s,

output Carry4

);

wire [3:0] b=4'b0000;

wire cin1=1'b1;

wire Carry1,Carry2,Carry3;

fulladder fa\_0( a[0], b[0], cin1, s[0], Carry1);

fulladder fa\_1( a[1], b[1], Carry1, s[1], Carry2);

fulladder fa\_2( a[2], b[2], Carry2, s[2], Carry3);

fulladder fa\_3( a[3], b[3], Carry3, s[3], Carry4);

Endmodule

module addition(

input [3:0] a,

input [3:0] b,

output [3:0] s,

output Carry4

);

wire cin1=1'b0;

wire Carry1,Carry2,Carry3;

fulladderr fa\_0( a[0], b[0], cin1, s[0], Carry1);

fulladderr fa\_1( a[1], b[1], Carry1, s[1], Carry2);

fulladderr fa\_2( a[2], b[2], Carry2, s[2], Carry3);

fulladderr fa\_3( a[3], b[3], Carry3, s[3], Carry4);

Endmodule

module addcarry(

input [3:0] a,

input [3:0] b,

output [3:0] s,

output Carry4

);

wire cin1=1'b1;

wire Carry1,Carry2,Carry3;

fulladd fa\_0( a[0], b[0], cin1, s[0], Carry1);

fulladd fa\_1( a[1], b[1], Carry1, s[1], Carry2);

fulladd fa\_2( a[2], b[2], Carry2, s[2], Carry3);

fulladd fa\_3( a[3], b[3], Carry3, s[3], Carry4);

Endmodule

module addcomple(

input [3:0] a,

input [3:0] d,

output [3:0] s,

output Carry4

);

wire cin1=1'b0;

wire [3:0]b;

assign b=~d;

wire Carry1,Carry2,Carry3;

fulladde fa\_0( a[0], b[0], cin1, s[0], Carry1);

fulladde fa\_1( a[1], b[1], Carry1, s[1], Carry2);

fulladde fa\_2( a[2], b[2], Carry2, s[2], Carry3);

fulladde fa\_3( a[3], b[3], Carry3, s[3], Carry4);

Endmodule

module sub(

input [3:0] a,

input [3:0] d,

output reg [3:0]df

);

wire cin1=1'b1;

wire [3:0]b,m;

assign b=~d;

wire [3:0]s;

wire Carry1,Carry2,Carry3,Carry5;

negone negone1(s,m,Carry5);

fulladder fa\_0( a[0], b[0], cin1, s[0], Carry1);

fulladder fa\_1( a[1], b[1], Carry1, s[1], Carry2);

fulladder fa\_2( a[2], b[2], Carry2, s[2], Carry3);

fulladder fa\_3( a[3], b[3], Carry3, s[3], Carry4);

always@(a,d,cin1,Carry4,m,s)

begin

case(Carry4)

0:df=m;

1:df=s;

endcase

end

Endmodule

module negone(

input [3:0] n,

output [3:0] s,

output Carry4

);

wire [3:0]b=4'b0000;

wire [3:0]a;

assign a=~n;

wire cin1=1'b1;

wire Carry1,Carry2,Carry3;

fulladder fa\_0( a[0], b[0], cin1, s[0], Carry1);

fulladder fa\_1( a[1], b[1], Carry1, s[1], Carry2);

fulladder fa\_2( a[2], b[2], Carry2, s[2], Carry3);

fulladder fa\_3( a[3], b[3], Carry3, s[3], Carry4);

Endmodule

module decre(

input [3:0] a,

output reg [3:0] df

);

wire cin1=1'b1;

wire [3:0] d=4'b0001;

wire [3:0]b,m;

assign b=~d;

wire [3:0]s;

wire Carry1,Carry2,Carry3,Carry5;

negone negone1(s,m,Carry5);

fulladder fa\_0( a[0], b[0], cin1, s[0], Carry1);

fulladder fa\_1( a[1], b[1], Carry1, s[1], Carry2);

fulladder fa\_2( a[2], b[2], Carry2, s[2], Carry3);

fulladder fa\_3( a[3], b[3], Carry3, s[3], Carry4);

always@(a,d,cin1,Carry4,m,s)

begin

case(Carry4)

0:df=m;

1:df=s;

endcase

end

endmodule

module negone(

input [3:0] n,

output [3:0] s,

output Carry4

);

wire [3:0]b=4'b0000;

wire [3:0]a;

assign a=~n;

wire cin1=1'b1;

wire Carry1,Carry2,Carry3;

fulladder fa\_0( a[0], b[0], cin1, s[0], Carry1);

fulladder fa\_1( a[1], b[1], Carry1, s[1], Carry2);

fulladder fa\_2( a[2], b[2], Carry2, s[2], Carry3);

fulladder fa\_3( a[3], b[3], Carry3, s[3], Carry4);

Endmodule

module andd(

input [3:0] a,

input [3:0] b,

output [3:0] Y

);

assign Y=a&b;

endmodule

module orr(

input [3:0] a,

input [3:0] b,

output [3:0] Y

);

assign Y=a|b;

endmodule

module exorr(

input [3:0] a,

input [3:0] b,

output [3:0] Y

);

assign Y=a^b;

endmodule

module comple(

input [3:0] a,

output [3:0] Y

);

assign Y=~a;

endmodule

module shiftl(

input [3:0] a,

output [3:0] Y

);

assign Y[3]=a[2];

assign Y[2]=a[1];

assign Y[1]=a[0];

assign Y[0]=1'b0;

endmodule

module shiftr(

input [3:0] a,

output [3:0] Y

);

assign Y[3]=1'b0;

assign Y[2]=a[3];

assign Y[1]=a[2];

assign Y[0]=a[1];

endmodule