**DATAFLOW MODELLING**

**NAME: RITHIKA KATHIRVEL**

**ROLL NUMBER: B200055CS**

1.

module MUX2X1\_DATAFLOW(y\_out,in\_y,in\_y\_,in\_s);

output y\_out;

input in\_y,in\_y\_,in\_s;

assign y\_out=~(~(in\_y&~(in\_s&in\_s))&~(in\_y\_&in\_s));

endmodule

2.

module DEMUX\_DATA(y\_out,y\_out\_1,in\_y\_,in\_selec);

output y\_out,y\_out\_1;

input in\_y\_;

input in\_selec;

assign y\_out=~((~(in\_y\_&(~(in\_selec&in\_selec))))&(~(in\_y\_&(~(in\_selec&in\_selec)))));

assign y\_out\_1=~((~(in\_y\_&in\_selec))&(~(in\_y\_&in\_selec)));

endmodule

3.

module sixteenbit\_data\_MUX(y\_out,in\_y,in\_y\_,in\_selec);

output[15:0]y\_out;

input[15:0]in\_y;

input[15:0]in\_y\_;

input in\_selec;

MUX2X1\_DATAFLOW test[15:0](.y\_out(y\_out),.in\_y(in\_y),.in\_y\_(in\_y\_),.in\_selec(in\_selec));

Endmodule

4.

module mux4X1\_data(out,i0,i1,i2,i3,s1,s0);

input[15:0] i0,i1,i2,i3;

input s1,s0;

output[15:0]out;

wire[15:0] mux1,mux2;

sixteenbit\_data\_MUX test\_mux\_1(.y\_out(mux1),.in\_y(i0),.in\_y\_(i1),.in\_selec(s1));

sixteenbit\_data\_MUX test\_mux\_2(.y\_out(mux2),.in\_y(i2),.in\_y\_(i3),.in\_selec(s1));

sixteenbit\_data\_MUX test\_mux\_3(.y\_out(out),.in\_y(mux1),.in\_y\_(mux2),.in\_selec(s0));

endmodule

5.

module mux\_16bit\_8x1\_data(out\_data\_y,inp1,inp2,inp3,inp4,inp5,inp6,inp7,inp8,s0,s1,s2);

output[15:0]out\_data\_y;

input[15:0]inp1,inp2,inp3,inp4,inp5,inp6,inp7,inp8;

input s0,s1,s2;

wire[15:0] t1,t2;

mux4X1\_data test\_1(.out(t1),.i0(inp1),.i1(inp2),.i2(inp3),.i3(inp4),.s1(s0),.s0(s1));

mux4X1\_data test\_2(.out(t2),.i0(inp5),.i1(inp6),.i2(inp7),.i3(inp8),.s1(s0),.s0(s1));

MUX2X1\_DATAFLOW test\_3(.y\_out(out\_data\_y),.in\_y(t1),.in\_y\_(t2),.in\_s(s2));

endmodule

6.

module DEMUX\_1X4\_data(out\_y\_1,out\_y\_2,out\_y\_3,out\_y\_4,in\_1,in\_s1,in\_s2);

output out\_y\_1,out\_y\_2,out\_y\_3,out\_y\_4;

input in\_1,in\_s1,in\_s2;

wire t0,t1;

DEMUX\_DATA(.y\_out(t0),.y\_out\_1(t1),.in\_y\_(in\_1),.in\_selec(in\_s1));

DEMUX\_DATA(.y\_out(out\_y\_1),.y\_out\_1(out\_y\_2),.in\_y\_(t0),.in\_selec(in\_s2));

DEMUX\_DATA(.y\_out(out\_y\_3),.y\_out\_1(out\_y\_4),.in\_y\_(t1),.in\_selec(in\_s2));

endmodule

7.

module DEMUX\_1X8\_DATA(out1,out2,out3,out4,out5,out6,out7,out8,in1,ins1,ins2,ins3);

output out1,out2,out3,out4,out5,out6,out7,out8;

input in1,ins1,ins2,ins3;

wire t0,t1,t2,t3,t4,t5;

DEMUX\_DATA test\_1(.y\_out(t0),.y\_out\_1(t1),.in\_y\_(in1),.in\_selec(ins1));

DEMUX\_DATA test\_2(.y\_out(t2),.y\_out\_1(t3),.in\_y\_(t0),.in\_selec(ins2));

DEMUX\_DATA test\_3(.y\_out(t4),.y\_out\_1(t5),.in\_y\_(t1),.in\_selec(ins2));

DEMUX\_DATA test\_4(.y\_out(out1),.y\_out\_1(out2),.in\_y\_(t2),.in\_selec(ins3));

DEMUX\_DATA test\_5(.y\_out(out3),.y\_out\_1(out4),.in\_y\_(t3),.in\_selec(ins3));

DEMUX\_DATA test\_6(.y\_out(out5),.y\_out\_1(out6),.in\_y\_(t4),.in\_selec(ins3));

DEMUX\_DATA test\_7(.y\_out(out7),.y\_out\_1(out8),.in\_y\_(t5),.in\_selec(ins3));

endmodule