**Various ways MUX code**

module various\_mux(

input wire [3:0]a\_i,

input wire [3:0]sel\_i,

output wire y\_ter\_o,

output logic y\_case\_o,

output logic y\_ifelse\_o,

output logic y\_loop\_o,

output wire y\_aor\_o

);

assign y\_ter\_o = sel\_i[0] ? a\_i[0] :

sel\_i[1] ? a\_i[1] :

sel\_i[2] ? a\_i[2] :

a\_i[3];

always\_comb begin

case(sel\_i)

4'b0001: y\_case\_o = a\_i[0];

4'b0010: y\_case\_o = a\_i[1];

4'b0100: y\_case\_o = a\_i[2];

4'b1000: y\_case\_o = a\_i[3];

default: y\_case\_o = 1'bx;

endcase

end

always\_comb begin

if(sel\_i[0]) y\_ifelse\_o = a\_i[0];

else if(sel\_i[1]) y\_ifelse\_o = a\_i[1];

else if(sel\_i[2]) y\_ifelse\_o = a\_i[2];

else if(sel\_i[3]) y\_ifelse\_o = a\_i[3];

else y\_ifelse\_o = 1'bx;

end

always\_comb begin

y\_loop\_o = 0;

for(int i=0; i<4; i++)begin

y\_loop\_o = (sel\_i[0] & a\_i[0]) | y\_loop\_o;

end

end

assign y\_aor\_o = (sel\_i[0] & a\_i[0]) |

(sel\_i[1] & a\_i[1]) |

(sel\_i[2] & a\_i[2]) |

(sel\_i[3] & a\_i[3]);

Endmodule

**Test Bench Code**

module tb\_various\_mux ();

logic [3:0] a\_i;

logic [3:0] sel\_i;

// Output using ternary operator

logic y\_ter\_o;

// Output using case

logic y\_case\_o;

// Ouput using if-else

logic y\_ifelse\_o;

// Output using for loop

logic y\_loop\_o;

// Output using and-or tree

logic y\_aor\_o;

// Module instance

various\_mux day13(.\*);

// Stimulus

initial begin

for (int i =0; i<32; i++) begin

a\_i = $urandom\_range(0, 4'hF);

sel\_i = 1'b1 << $urandom\_range(0, 2'h3); // one-hot

#5;

end

$finish();

end

// VCD

initial begin

$dumpfile("day13.vcd");

$dumpvars(0, tb\_various\_mux);

end

endmodule

**Simulation**

