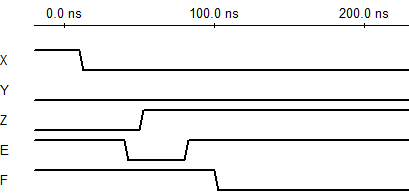
1)a)



1)b)

`timescale 1ns / 1ps

module HW1\_1b(e, f, a, b, c, d);

input a, b, c, d; //define inputs

output e, f; //define outputs

wire x, y, z; //intermediate wires

not NOT2(x, a); //assign x wire

and AND2(y, a, b); //assign y wire

or OR2(z, y, c); //assign z wire

or OR1(e, x, z); //assign e output

nand NAND2(f, z, d); //assign f output

Endmodule

1)c)

`timescale 1ns / 1ps

module HW1\_1c(e, f, a, b, c, d);

input a, b, c, d; //define inputs

output e, f; //define outputs

assign e = ~a | ((a & b) | c); //dataflow for output e. Not a or z

assign f = ~(d & ((a & b) | c)); //dataflow for output f. D nand z

endmodule

1)d)

`timescale 1ns / 1ps

module HW1\_1d(e, f, a, b, c, d);

input a, b, c, d; //define inputs

output e, f; //define outputs

reg e, f; //make output reg to be assigned in always block

always@(a or b or c or d) //sensitive to change of inputs

begin

e = ~a | ((a & b) | c); //assign output e. Not A or Z

f = ~(d & ((a & b) | c)); //assign output f. D nand z

end

Endmodule

2)

