# EEL 3701 – Digital Logic and Computer Systems

# Short Report Design Problem 2

# Greg Bolling

# UF ID 64911960

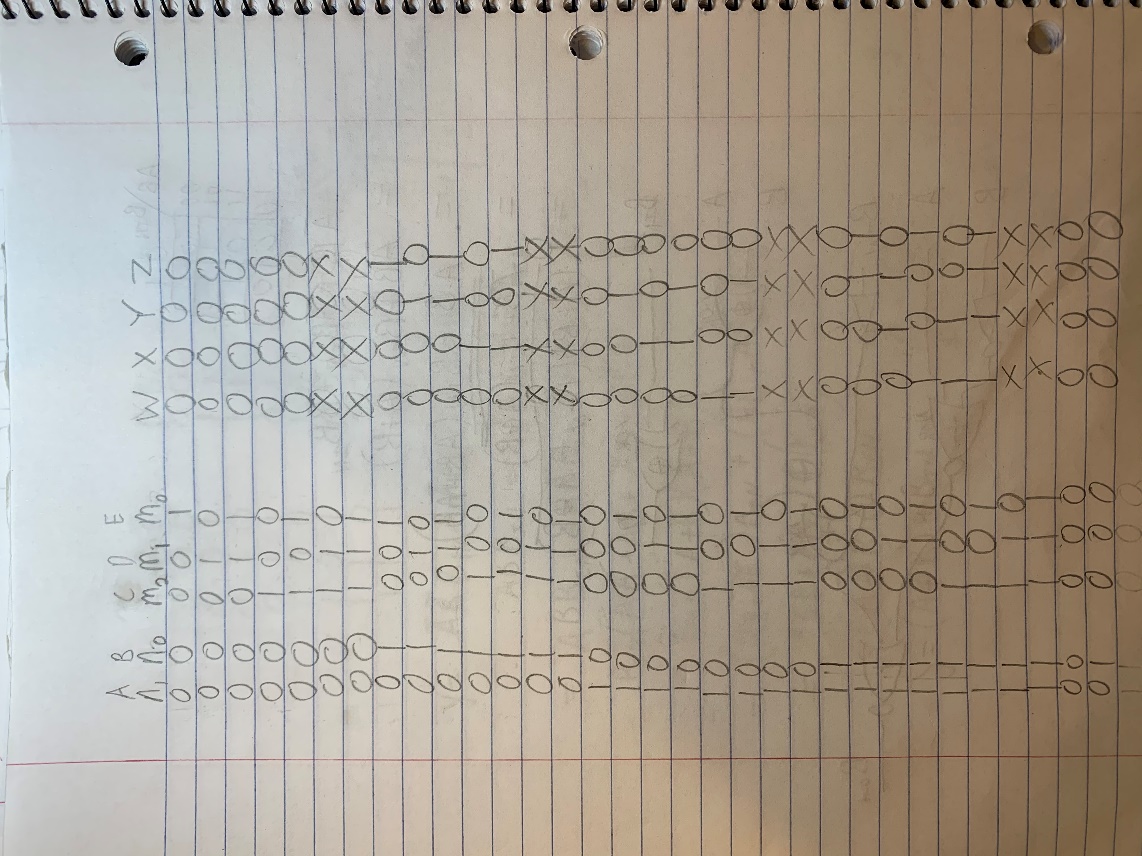
## Problem Statement

The goal of the design was to create a multiplication of a 2 bit number by a 3 bit number..

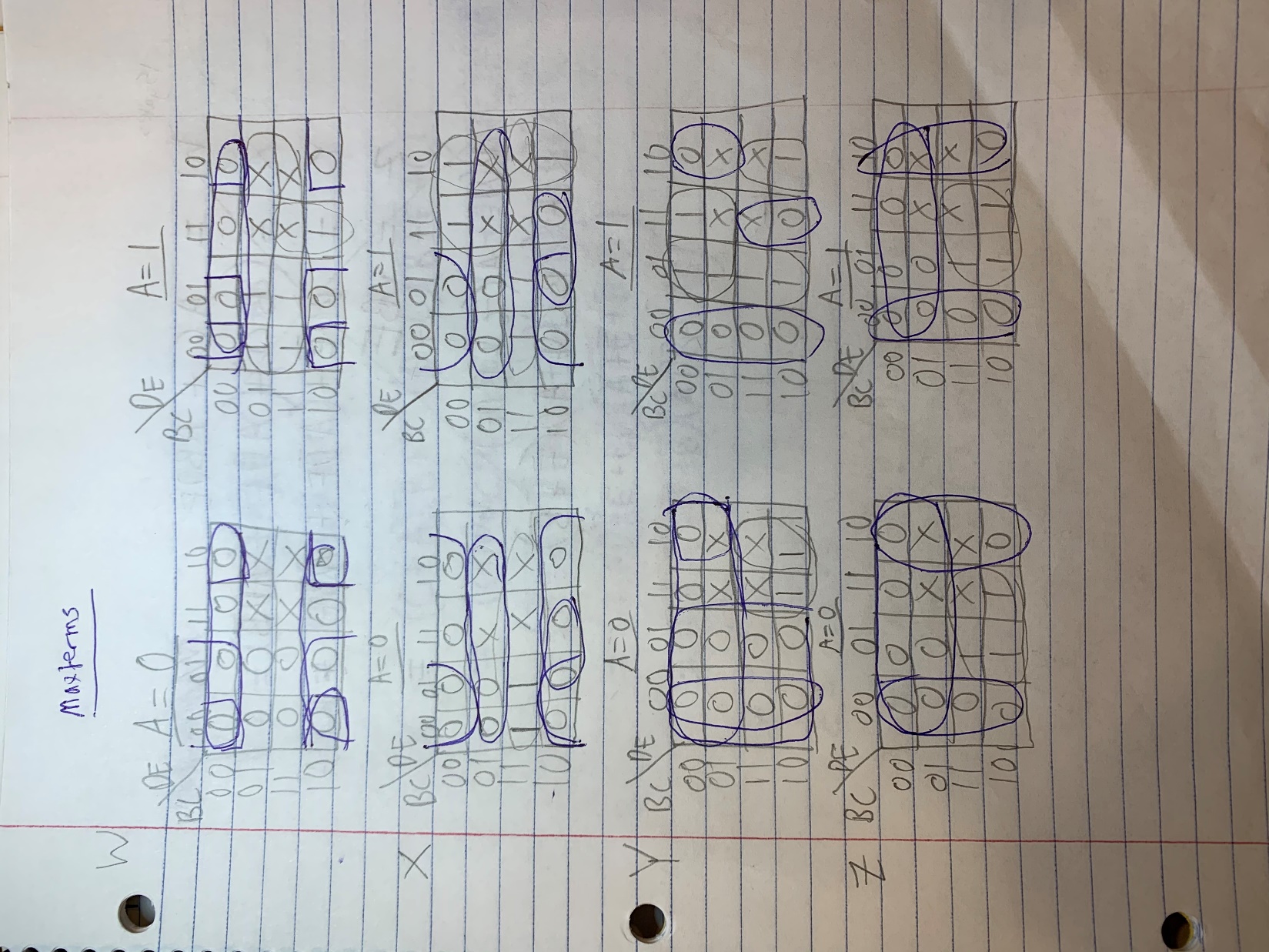
## Design

I implemented this design as a Product of Sums to make sure it was a NOR-NOR implementation. This meant using K-Maps and circling the zeros. I tried Sum of Products first but that didn’t work too well. I wrote the equations from the desin implementation and tested them in VHDL using Switch 1 and 2 for the first 2-bit value and Switch 3,4,5 for the next three bit value.

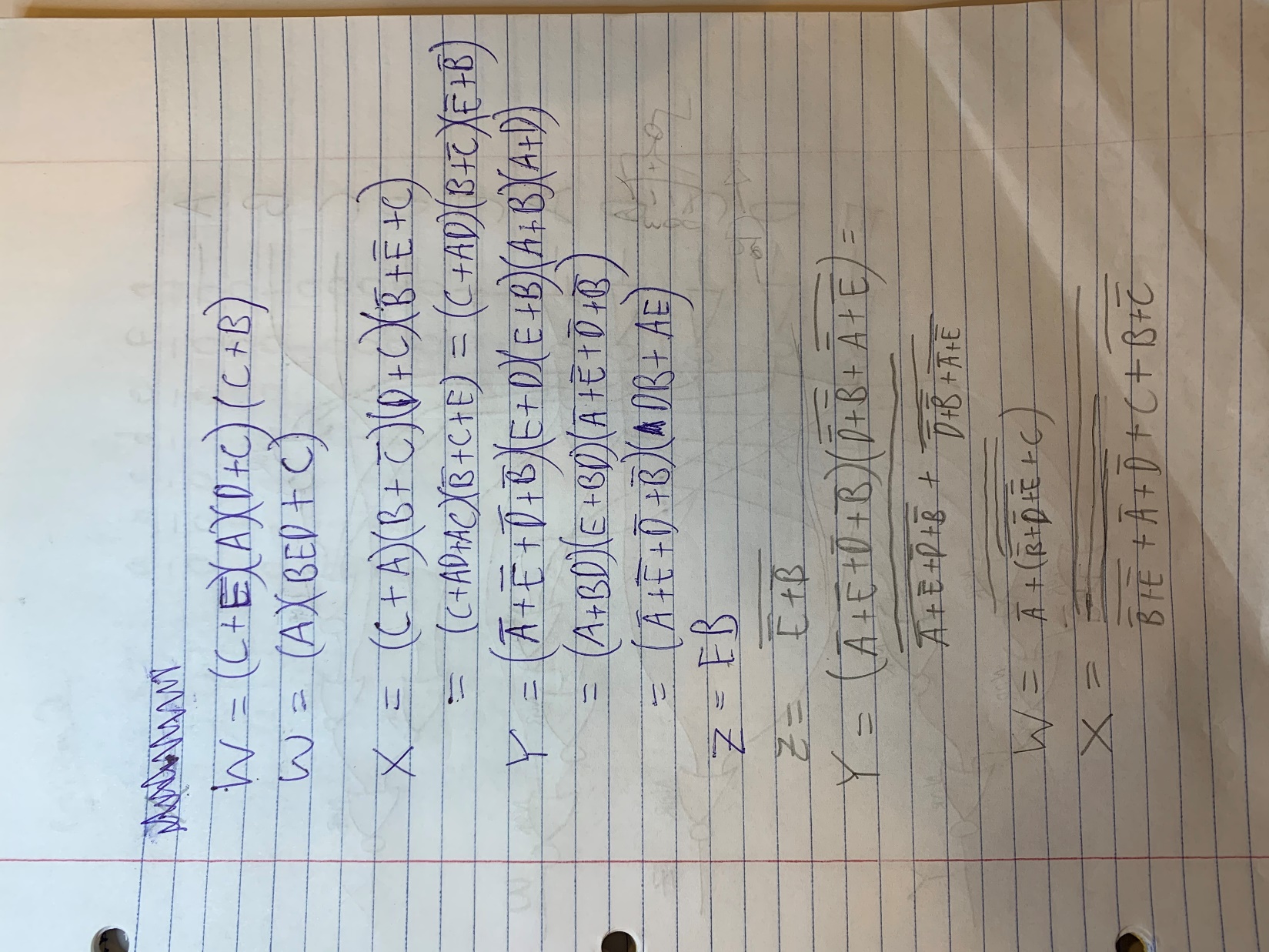
Truth Table:



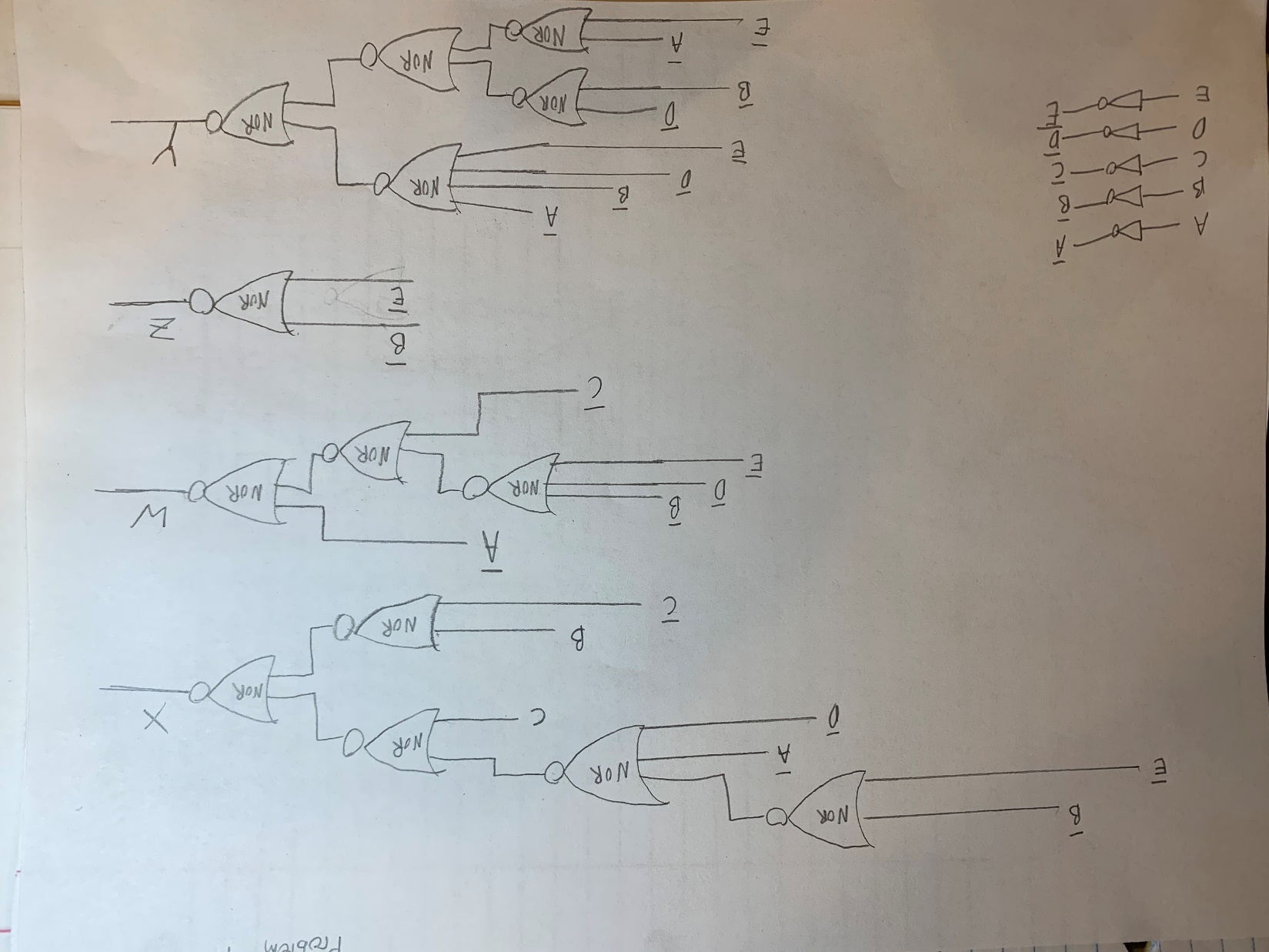
K-Maps



Equations and Boolean Reduction



Schematic



## Implementation

I implemented this with the board we have from class. It was done in VHDL and tested as per the video.

The VHDL Code for the NOR gates is below.

-------------------------------------------------------------------------------

-- Author: Greg Bolling

-- Date: March 12, 2020

-- Class; EEL 3701

-- Assignment: DesignProblem 1 Top Level Entity

-- Class: 12368

-- Section 7441

-- PI Name: Savvas Ferekides

-------------------------------------------------------------------------------

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.numeric\_std.all;

use ieee.std\_logic\_unsigned.all;

entity DesignProblem1 is

port (

Switch1 : in std\_logic; -- A

Switch2 : in std\_logic; -- B

Switch3 : in std\_logic; -- C

Switch4 : in std\_logic; -- D

Switch5 : in std\_logic; -- E

LEDOut : out std\_logic\_vector(3 DOWNTO 0) -- WXYZ

);

end DesignProblem1;

architecture AttachAll of DesignProblem1 is

signal w : std\_logic;

signal w1 : std\_logic;

signal w2 : std\_logic;

signal x : std\_logic;

signal x1 : std\_logic;

signal x2 : std\_logic;

signal x3 : std\_logic;

signal y : std\_logic;

signal y1 : std\_logic;

signal y2 : std\_logic;

signal y3 : std\_logic;

signal y4 : std\_logic;

signal z : std\_logic;

signal z1 : std\_logic;

signal A : std\_logic;

signal B : std\_logic;

signal C : std\_logic;

signal D : std\_logic;

signal E : std\_logic;

begin

A <= Switch1;

B <= Switch2;

C <= Switch3;

D <= Switch4;

E <= Switch5;

-- W

LEDOut(3) <= (not A) nor w1; -- one nor gate

w1 <= C nor w2; -- one nor gate

w2 <= not(((not B) or (not D)) or (not E)); -- one NOR gate

-- X

LEDOut(2) <= x2 nor x3;

x1 <= not (z1 or (not A) or (not D));

x2 <= B nor (not C);

x3 <= c nor x1;

-- Y

LEDOUt(1) <= y1 nor y4;

y1 <= not( (not A) or (not E) or (not D) or (not B) );

y2 <= (not D) nor (not B);

y3 <= (not A) nor (not E);

y4 <= y2 nor y3;

-- Z

LEDOut(0) <= z1;

z1 <= (not E) nor (not B);

end AttachAll;