OUTCOME #2: "an ability to analyze and design combinational logic circuits."

Multiple Choice – select the single most appropriate response for each question.
Note that "none of the above" MAY be a VALID ANSWER.

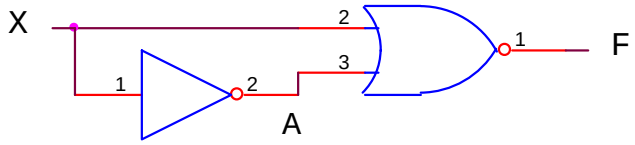
Place answers on the supplied BUBBLE SHEET only – nothing written here will be graded.

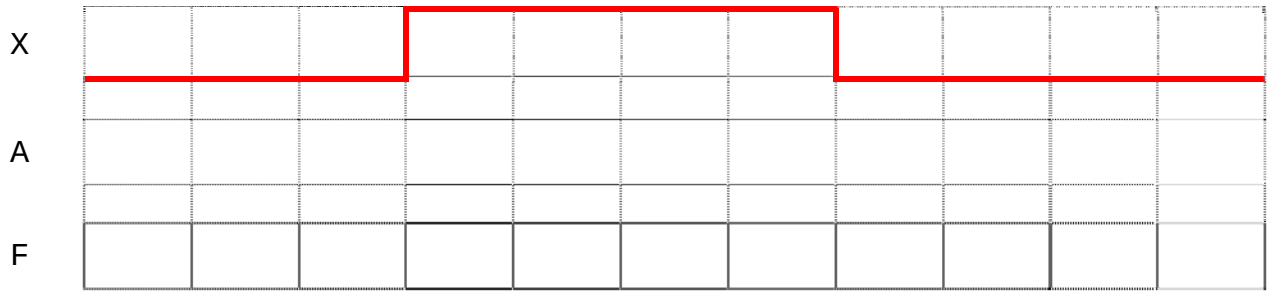
1. Most **logic minimization methods** are based on a generalization of:
   (A) the covering theorem
   (B) the combining theorem
   (C) the consensus theorem
   (D) DeMorgan's Law
   (E) none of the above

2. If the function **F(X,Y,Z)** is represented by the **ON set** $\sum_{X,Y,Z}$**(0,3,4,7)**, then the **dual** of this function $\mathbf{F^D(X,Y,Z)}$ is represented by the **ON set:**
   (A) $\sum_{X,Y,Z}$**(0,3,4,7)**
   (B) $\sum_{X,Y,Z}$**(1,2,4,6)**
   (C) $\sum_{X,Y,Z}$**(1,2,5,6)**
   (D) $\sum_{X,Y,Z}$**(1,2,4,7)**
   (E) none of the above

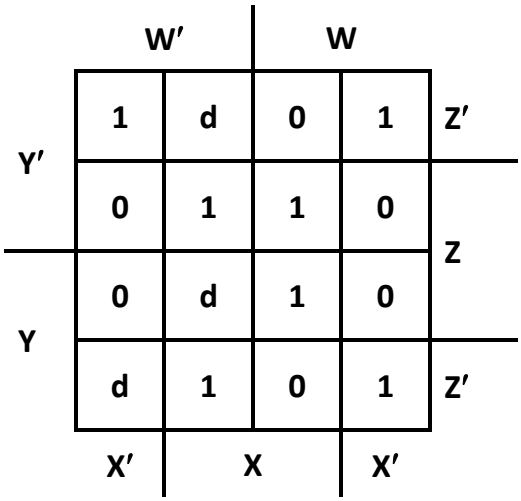
3. The **property** listed below that is **NOT** true for **XOR gates** is:
   (A) Associativity
   (B) commutivity
   (C) distributivity
   (D) idempotency
   (E) none of the above

4. The circuit shown below exhibits the following type of **hazard** when its input, X, transitions from **high-to-low**:
   (A) a static-0 hazard
   (B) a static-1 hazard
   (C) a dynamic low-to-high hazard
   (D) a dynamic high-to-low hazard
   (E) none of the above





The following K-map applies to questions 5 through 8:



5. Assuming the availability of **only true** input variables, the **fewest number of 2-input NAND gates** that are needed to realize this function is:

   (A) 6 (B) 7 (C) 8 (D) 9 (E) none of the above

6. Assuming the availability of **only true** input variables, the **fewest number of 2-input NOR gates** that are needed to realize this function is:

   (A) 6 (B) 7 (C) 8 (D) 9 (E) none of the above

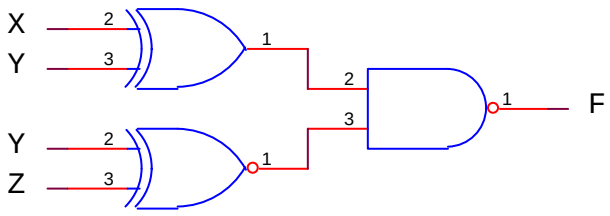
7. Assuming the availability of **only true** input variables, the **fewest number of 2-input open-drain NAND gates** that are needed to realize this function is:

   (A) 6 (B) 7 (C) 8 (D) 9 (E) none of the above

8. Assuming the availability of **only true** input variables, the **number of pull-up resistors** required to realize this function using **2-input open-drain NAND gates** is:

   (A) 2 (B) 3 (C) 4 (D) 5 (E) none of the above

The following circuit applies to questions 9 and 10:

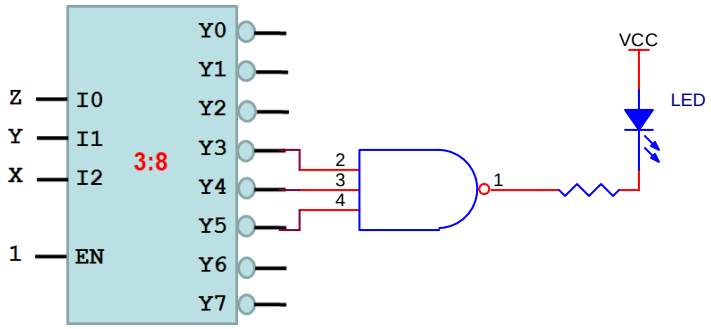


9. The **OFF set** of the function realized by this circuit is:
   (A) $\Pi_{x,y,z}$**(3,4)**
   (B) $\Pi_{x,y,z}$**(1,6)**
   (C) $\Pi_{x,y,z}$**(0,2,3,4,5,7)**
   (D) $\Pi_{x,y,z}$**(0,1,2,5,6,7)**
   (E) none of the above

10. The **cost** of a **minimal product of sums** realization of this function (assuming **both true and complemented variables** are available) would be:

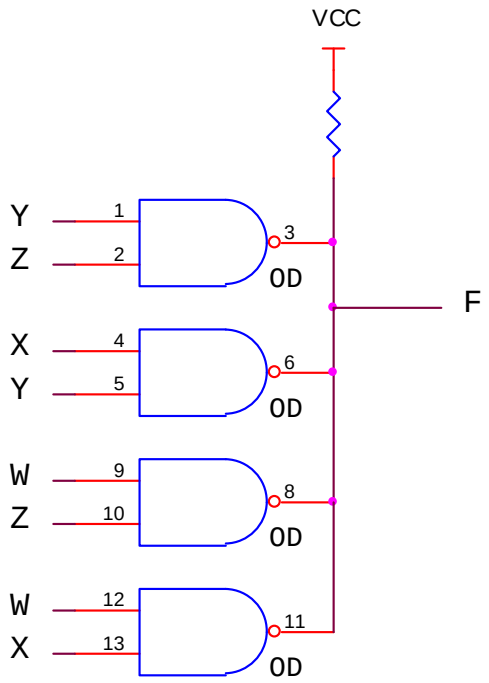
    (A) 10 (B) 11 (C) 12 (D) 13 (E) none of the above

The following circuit applies to question 11:



11. The **ON set** realized by this circuit, where the LED "on" condition corresponds to F(X,Y,Z)=1, is:
    (A) $\sum_{x,y,z}$ **(3,4,5)**
    (B) $\sum_{x,y,z}$ **(0,1,2,6,7)**
    (C) $\sum_{x,y,z}$ **(0,1,2)**
    (D) $\sum_{x,y,z}$ **(6.7)**
    (E) none of the above

The following circuit applies to questions 12 and 13:

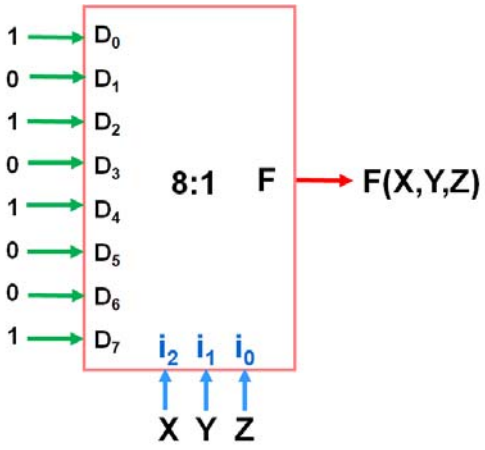


12. Expressed in minimum <u>sum-of-products</u> form, the function realized by this circuit is:
    (A) **W·Y + X·Z**
    (B) **W′·Y′ + X′·Z′**
    (C) **X·Y + Y·Z + W·Z + W·X**
    (D) **W′·Y′ + X′·Y·Z′ + X′·Y′·Z′**
    (E) none of the above

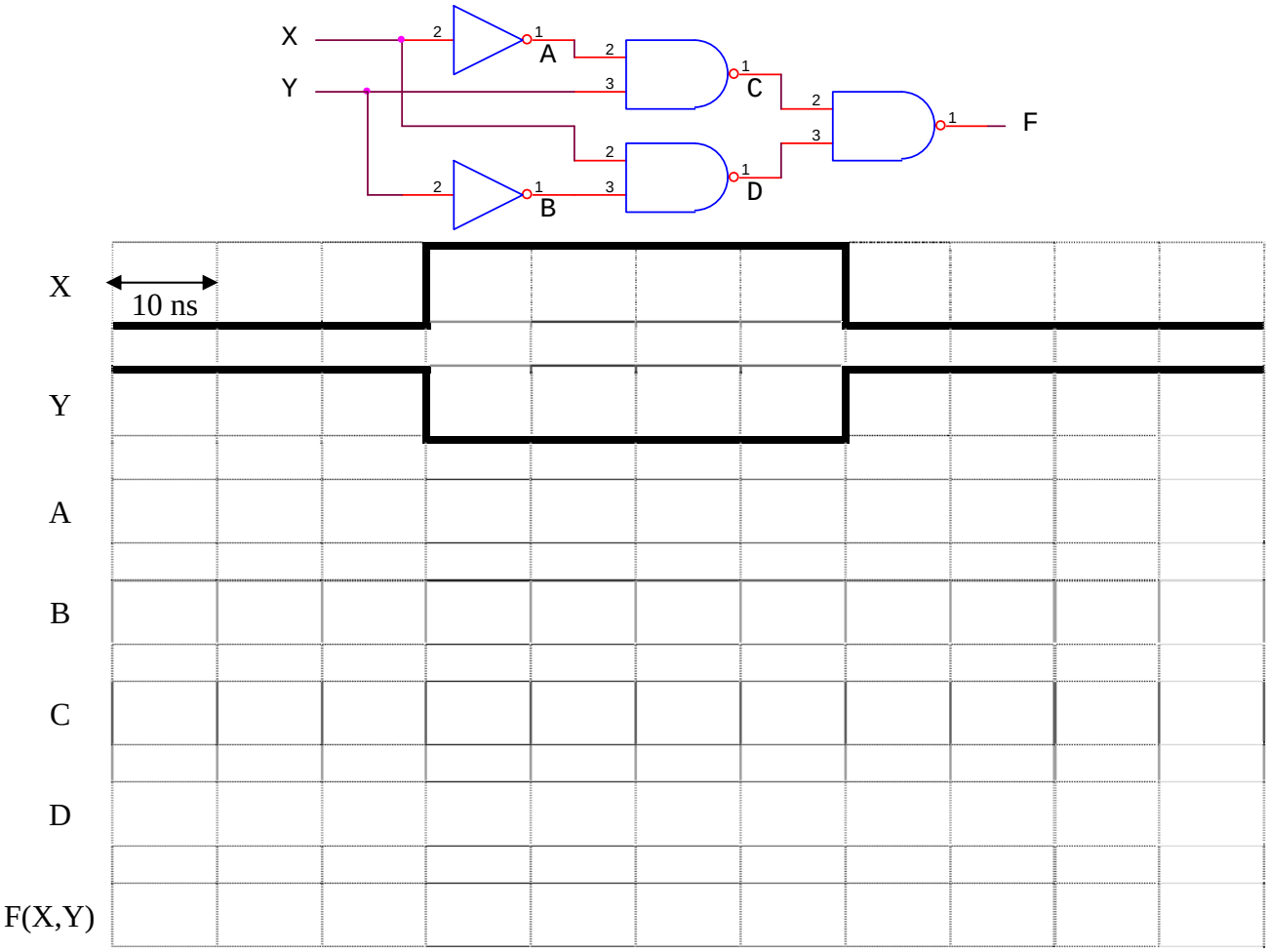
13. Expressed in minimum <u>product-of-sums</u> form, the function realized by this circuit is:
    (A) **(W+Y)·(X+Z)**
    (B) **(W′+Y′)·(X′+Z′)**
    (C) **(X+Y)·(Y+Z)·(W+Z)·(W+X)**
    (D) **(X′+Y′)·(Y′+Z′)·(W′+Z′)·(W′+X′)**
    (E) none of the above

---

14. The multiplexer circuit shown realizes the following **OFF set:**
    (A) $\prod_{X,Y,Z}$ **(0,2,4,7)**
    (B) $\prod_{X,Y,Z}$ **(0,1,6,7)**
    (C) $\prod_{X,Y,Z}$ **(1,3,4,7)**
    (D) $\prod_{X,Y,Z}$ **(1,3,5,6)**
    (E) none of the above



The following circuit and timing chart apply to questions 15 and 16. Assume each gate has a propagation delay of 10 ns; use the chart provided as a worksheet.



15. Steady-state (static) analysis of the function realized by this circuit for the inputs provided predicts that the output F(X,Y) should:
    (A) always be low
    (B) always be high
    (C) be equal to the input X
    (D) be equal to the input Y
    (E) none of the above

16. Dynamic analysis of the output F(X,Y) reveals that:
    (A) a static "0" hazard will be produced whenever the inputs X and Y transition simultaneously
    (B) a static "1" hazard will be produced whenever the inputs X and Y transition simultaneously
    (C) a static "0" hazard will be produced only when input X transitions from low-to-high and input Y transitions from high-to-low
    (D) a static "1" hazard will be produced only when input X transitions from high-to-low and input Y transitions from low-to-high
    (E) none of the above

The **ispLever-Generated Reports** below apply to questions 17 through 19.

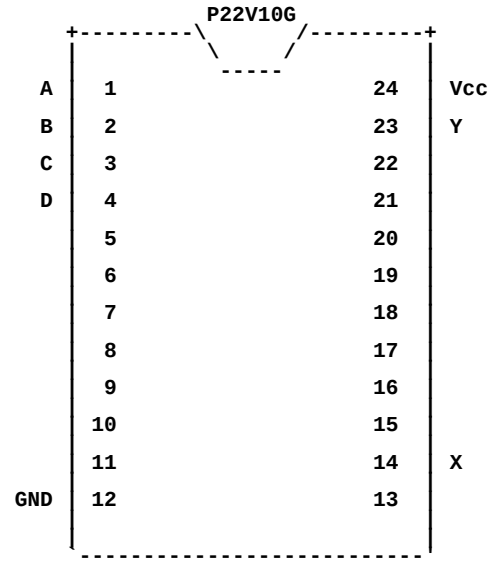
REDUCED EQUATION REPORT:

```
 P-Terms   Fan-in  Fan-out  Type  Name (attributes)
---------  ------  -------  ----  ----------------
   4/4        4        1    Pin   X
   3/4        4        1    Pin   Y
=========
   7/8          Best P-Term Total: 7
                      Total Pins: 6
                     Total Nodes: 0
              Average P-Term/Output: 3


Positive-Polarity Equations:

X = !B&!D&A&!C # B&D&A&!C # !B&!D&!A&C  # B&D&!A&C;

Y = B&D # !A&!C # A&C;

Reverse-Polarity Equations:

!X = B&!D # !B&D # !A&!C # A&C;

!Y = !D&A&!C # !B&A&!C # !D&!A&C # !B&!A&C;
```

ispLEVER operators:
AND - & OR - #
NOT - ! XOR - $

CHIP REPORT:

```
X = B&D&!A&C # !B&!D&!A&C
    # B&D&A&!C # !B&!D&A&!C;

Y = B&D # A&C # !A&!C;
```

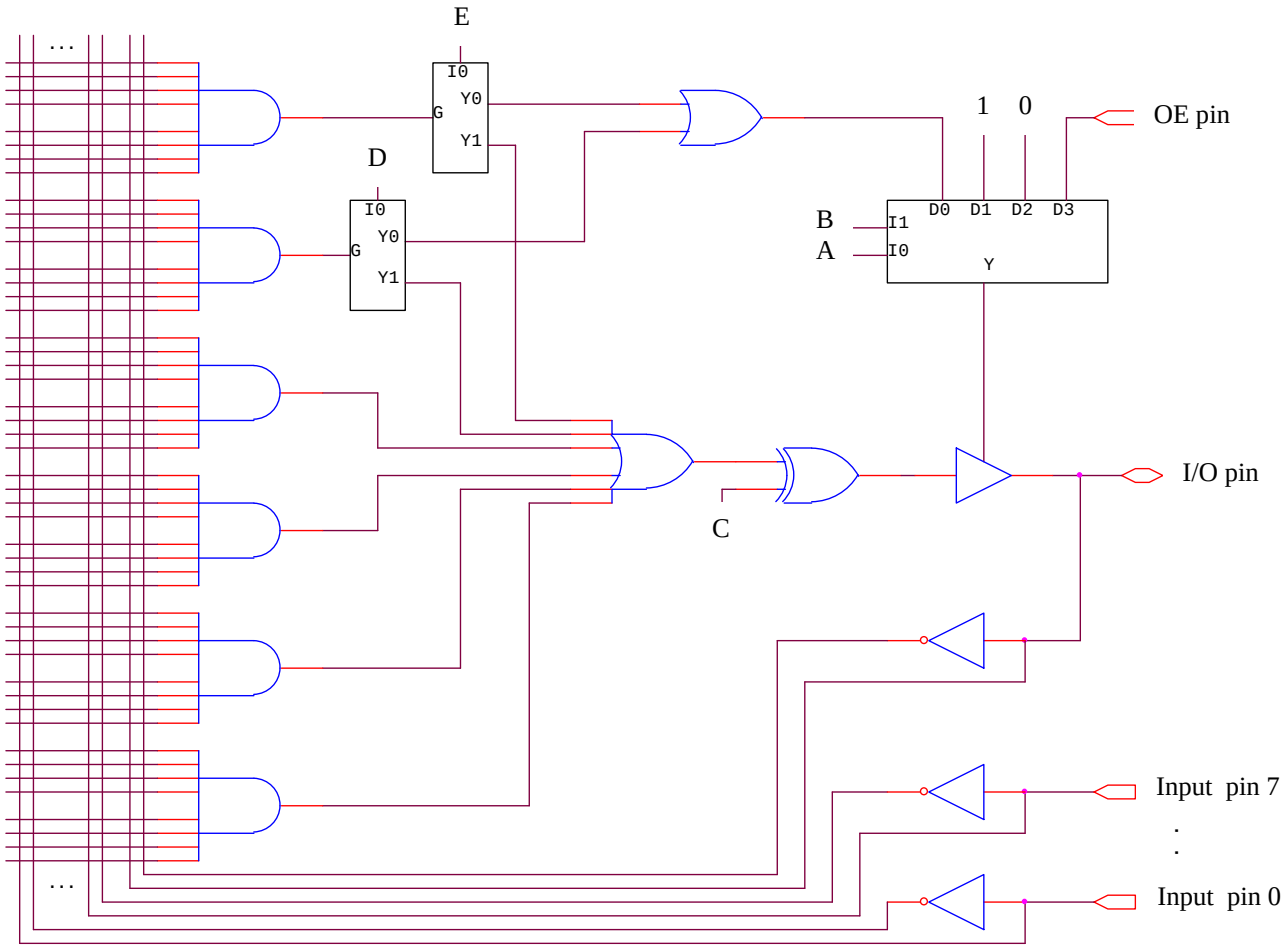


17. The **total number of P terms** used by this Verilog program is:
    (A) 3
    (B) 4
    (C) 7
    (D) 8
    (E) none of these

18. The **chip report** indicates that the fitter program chose the following forms of the reduced equations to burn into the PLD:
    (A) the positive polarity forms of both X and Y
    (B) the reverse polarity forms of both X and Y
    (C) the positive polarity form of X and the reverse polarity form of Y
    (D) the reverse polarity form of X and the positive polarity form of Y
    (E) none of the above

19. A possible **Verilog source form** of the equation for **X** is:
    (A) `X = (B ^ D) & (A ~^ C);`
    (B) `X = (B ^ D) | (A ~^ C);`
    (C) `X = (B ~^ D) & (A ^ C);`
    (D) `X = (B ~^ D) | (A ^ C);`
    (E) none of the above

The **Macrocell Reference Figure** below applies to questions 20 through 22.



20. The possibilities for controlling the tri-state buffer enable do **not** include:
    (A) always OFF
    (B) always ON
    (C) controlled by an expression comprised of one product term
    (D) controlled by an expression comprised of two product terms
    (E) none of the above

21. If **D=0** and **E=1**, the **maximum number** of **product terms** that can be utilized by a function realized by the macrocell I/O pin is:

    (A) 2 (B) 4 (C) 5 (D) 6 (E) none of these

22. To realize the **positive polarity equation** of a function with an **active low** output, the following settings should be used:
    (A) **C=0, B=0, A=1**
    (B) **C=0, B=1, A=0**
    (C) **C=1, B=0, A=1**
    (D) **C=1, B=1, A=0**
    (E) none of the above

The following Verilog program applies to questions 23 through 25:

23. The number of equations generated by this program is:
    (A) 2
    (B) 4
    (C) 8
    (D) 16
    (E) none of the above

24. When **TEN** is negated, **S[1]=1,** and **S[0]=1,** the outputs **Y_z[3:0]** will:
    (A) all be zero
    (B) all be one
    (C) all be Hi-Z
    (D) be equal to the inputs **A[3:0]**
    (E) none of the above

25. When **TEN** is asserted, **S[1]=1,** and **S[0]=0,** the outputs **Y_z[3:0]** will:
    (A) be equal to the inputs **A[3:0]**
    (B) be equal to the inputs **B[3:0]**
    (C) be equal to the inputs **C[3:0]**
    (D) be equal to the inputs **D[3:0]**
    (E) none of the above

```
/* Mid-Size Multiplexer */

module midmux(TEN,S,A,B,C,D,Y_z);

  input wire TEN;
  input wire [1:0] S;
  input wire [3:0] A, B, C, D;
  output tri [3:0] Y_z;
  wire [3:0] Y;

  assign Y_z = TEN ? Y : 4'bZZZZ;

  always @ (S) begin
    Y = 4'b0000;
    case (S)
      2'd0: Y = A;
      2'd1: Y = B;
      2'd2: Y = C;
      2'd3: Y = D;
    endcase
  end
endmodule
```

The following Verilog program applies to questions 26 and 27:

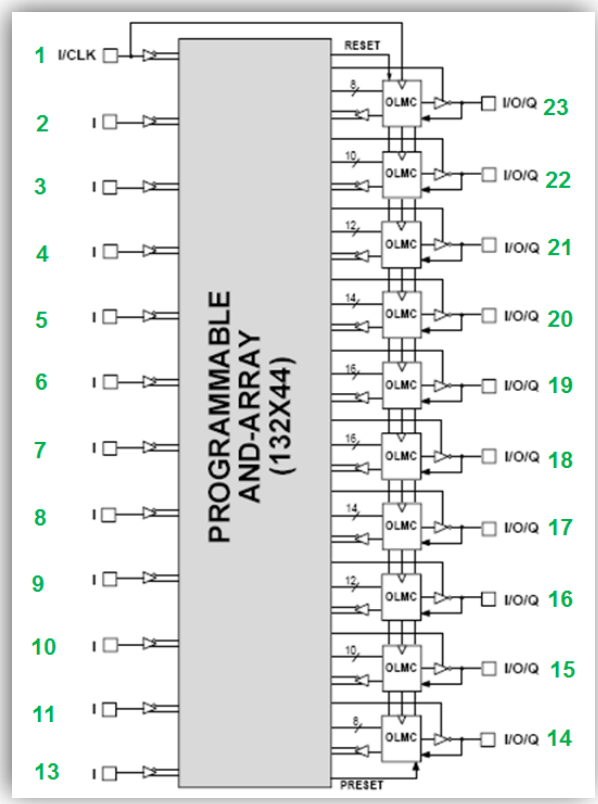
26. If input **I[0]** is **asserted** and *all the other inputs* are **negated**, the output produced will be:
    (A) **E_z[1]=0, E_z[0]=0, G=0**
    (B) **E_z[1]=0, E_z[0]=0, G=1**
    (C) **E_z[1]=Hi-Z, E_z[0]=Hi-Z, G=Hi-Z**
    (D) **E_z[1]=Hi-Z, E_z[0]=Hi-Z, G=1**
    (E) none of the above

27. If inputs **I[1], I[2],** and **TEN** are **asserted** and *all the other inputs* are **negated**, the output produced will be:
    (A) **E_z[1]=0, E_z[0]=0, G=0**
    (B) **E_z[1]=0, E_z[0]=1, G=1**
    (C) **E_z[1]=1, E_z[0]=0, G=1**
    (D) **E_z[1]=Hi-Z, E_z[0]=Hi-Z, G=1**
    (E) none of the above

```
/* Mid-Size Priority Encoder */
module prienc(I, E_z, G, TEN);
  input wire [3:0] I;
  input wire TEN;
  output tri [1:0] E_z;
  output wire G;
  reg [2:0] EG;
  always @ (I) begin
    casez (I)
      4'b0000:  EG = 3'b000;
      4'b0001:  EG = 3'b001;
      4'b001?:  EG = 3'b011;
      4'b01??:  EG = 3'b101;
      4'b1???:  EG = 3'b111;
    endcase
  end
  assign G = EG[0];
  assign E_z = TEN ? EG[2:1] : 2'bzz;
endmodule
```

The following Verilog program and 22V10 diagram apply to questions 28 through 30.

```
/* Big XOR Attempt on 22V10 */

module bigxor (I, BXOUT);

  input wire [11:0] I;
  output wire BXOUT;

  assign BXOUT = ^I;

endmodule
```



Pins 14 & 23 – 8 P-terms each
Pins 15 & 22 – 10 P-terms each
Pins 16 & 21 – 12 P-terms each
Pins 17 & 20 – 14 P-terms each
Pins 18 & 19 – 16 P-terms each

28. Just when you thought the exam was safe from your well-intentioned *"best friend from another major"* (BFFAM), you discover a Verilog program they have written for a 22V10. It looks like this time your BFFAM is attempting to create a 12-variable XOR function. The **number of product terms** required to realize the equation for **`BXOUT`**, given the Verilog program **<u>as written</u>**, is:
    (A) 24
    (B) 1024
    (C) 2048
    (D) 4096
    (E) none of these

29. As expected, the Verilog program targeting a 22V10 **doesn't fit (<u>as written</u>)** because:
    (A) the pin numbers are not specified
    (B) there are an insufficient number of input pins
    (C) there are an insufficient number of I/O (macrocell) pins
    (D) there are an insufficient number of P-terms
    (E) none of the above

30. The Verilog program shown could be **modified** to successfully realize the "Big XOR" function on a single 22V10 by:
    (A) adding an explicit pin assignment for **`BXOUT`**
    (B) adding explicit pin assignments for *all* the input and output variables
    (C) **splitting the equation for `BXOUT`** into two 5-variable XOR functions (assigned to two specific macrocells), the **outputs of which** could then be XORed with the **other two inputs** using a third macrocell (i.e. using a total of 3 macrocells)
    (D) *changing their degree option,* as there is **no way** to implement a 12-variable XOR function on a 22V10, and consequently there is **no way** to successfully modify the Verilog program!
    (E) none of the above