Mohammed Hussein

key

جامعة بيرزيت
BIRZEIT UNIVERSITY

Faculty of Engineering & Technology – Electrical & Computer Engineering Department

First Semester 2017-2018 Digital Systems (ENCS234) Midterm Exam

Allowed Time: 90 minutes Total Marks: 100 Date: 03/12/2017

---

Student ID : key (Only your ID **without** your name and your instructor name)

| ABET Outcome | Question # | Full Mark | Student Mark |
|---|---|---|---|
| | Q#1 | 45 | |
| | Q#2 | 18 | |
| | Q#3 | 12 | |
| | Q#4 | 10 | |
| | Q#5 | 15 | |
| | **TOTAL** | **100** | |

**Note:** write your solution on the space provided. If you need more space, write on the back of the sheet containing the question.

Q1] Select the correct answer (**45 points, 3 points each**):

1) What is the number represented of the binary words 10010010, assuming the representation is in Two's complement

   (A) -110
   B. 110
   C. -109
   D. -108
   E. None

2) The number $(161)_{10}$ is equivalent to:

   A. $(11)_{16}$
   B. $(1A)_{16}$
   (C) $(A1)_{16}$
   D. $(AA)_{16}$
   E. None

3) The number $(55)_{8}$ is equivalent to:

   (A) $(2D)_{16}$
   B. $(D2)_{16}$
   C. $(B1)_{16}$
   D. $(1B)_{16}$
   E. None

4) Using BCD code, when a computer adds (01001001) 2 + (10000000) 2 the result of this addition:

   A. is a correct BCD number.
   B. must be corrected by adding (00000110) 2.
   (C) must be corrected by adding (01100000) 2.
   D. must be corrected by adding (01100110) 2.
   E. is wrong and can't be corrected.
   F. None

5) Even parity is:
   (A) an extra bit added to make the total number of ones even to detect odd number of errors.
   B. an extra bit added to make the total number of ones even to detect even number of errors.

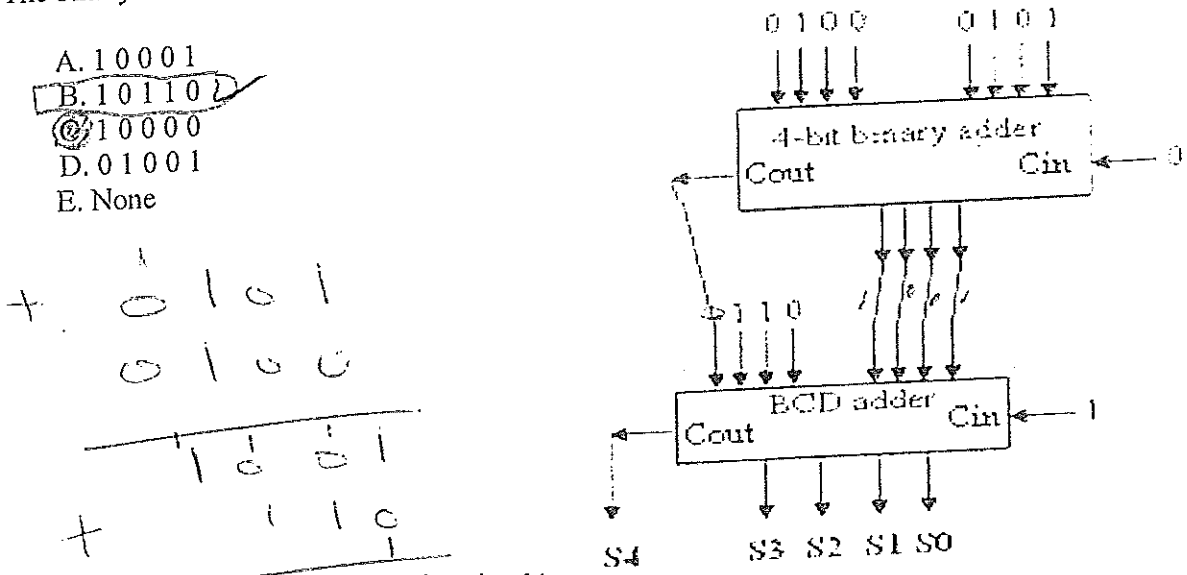
C. an extra bit added to make the total number of ones odd to detect odd number of errors.
D. an extra bit added to make the total number of ones odd to detect even number of errors.
E. an extra bit added to make the total number of ones even to detect any number of errors.

6) Given $F(x, y, z) = \prod(0, 3, 4, 7)$, and G is the complement of F, then:

A. $G(x, y, z) = \sum(0, 1, 2, 3, 5, 7)$
B. $G(x, y, z) = \prod(0, 3, 4, 7)$
C. $G(x, y, z) = \sum(1, 2, 5, 6)$
D. $G(x, y, z) = \sum(0, 3, 4, 7)$
E. None

7) The binary value of S4 S3 S2 S1 S0 in the circuit beside is

A. 1 0 0 0 1
B. 1 0 1 1 0
C. 1 0 0 0 0
D. 0 1 0 0 1
E. None




8) What is the result of Boolean expression simplification for $(B \oplus C) + (AB)'(A' + C')'$

A. $B \oplus C$
B. $A \oplus C$
C. $(B \oplus C)'$

D. $(A \oplus B)'$

E. None

9) In the shown K-map, the essential prime implicant is

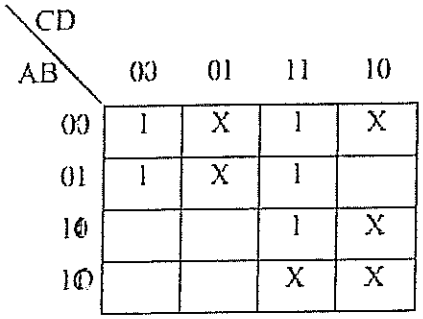
A. AC

B. CD

C. A' D

(D) A'C'

E. None




10) The output Y of the circuit computes

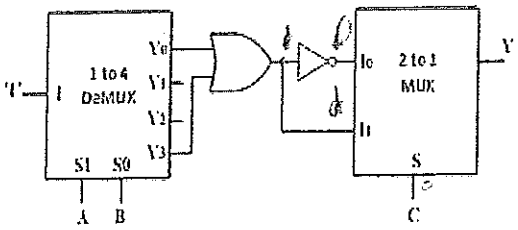
(A.) the sum bit of a full adder4

B. the carry bit of a full adder8

C. the complement of A

D. Complement of B

E. None



**11.** Which function F is implemented by using this 4X1 Multiplexer

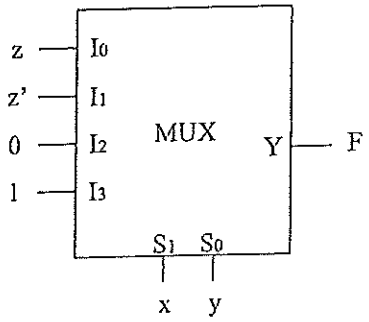
A. $F(x, y, z) = \sum(0, 3, 4, 5)$

B. $F(x, y, z) = \sum(1, 3, 6, 7)$

C. $F(x, y, z) = \sum(1, 2, 6, 7)$

D. $F(x, y, z) = \sum(1, 2, 5, 7)$

E. None




**12.** Implementation of full adder with two half adders and an ___ gate

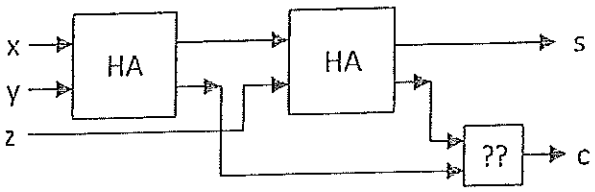
A. OR

B. NOR

C. XOR

D. XNOR

E. None




**13.** For the function F, the minimum product of sums expression

A. $F = w'x + w'y' + xy'z'$

B. $F = x'y + wx' + wz + wy$

C. $F = (x + y')(w' + x)(w' + z')(w' + y')$

D. $F = (w + x')(w + y)(x' + y + z)$

E. *None*

| yz \ wx | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | 1 | 1 | 0 |
| 01 | 1 | 1 | 0 | 0 |
| 11 | 0 | 1 | 0 | 0 |
| 10 | 0 | 1 | 0 | 0 |

14. For the shown Demux, the Boolean function is

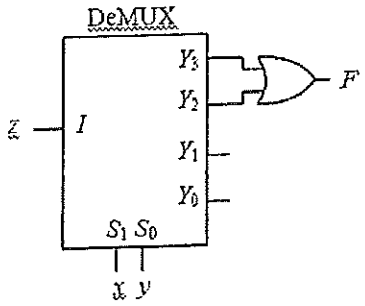
A. F = x' y' z + x y z

B. F = x' y z + x y z

(C) F = x y' z + x y z

D. F = x y' z' + x y z

E. None




15. For the shown circuit the function F is

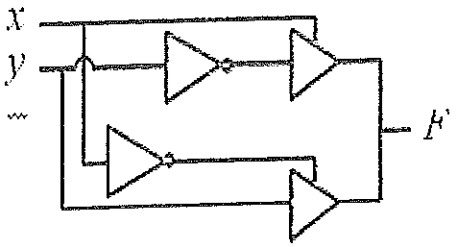
A. F = X AND Y

B. F = X OR Y

(C.) F = X XOR Y

D. F = X XNOR Y

E. None




| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|
| A | C | A | C | A | D | B | A | D | A |

| 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|
| C | A | C | C | C |

**Q2] 18 points**

Simplify using QM Tabulation method the following function

$$F(A,B,C,D) = \sum m(0,1,2,3,5,7,8,10,14,15)$$

1st-1

| minterm | A | B | C | D |
|---|---|---|---|---|
| ✓ 0 | 0 | 0 | 0 | 0 |
| ✓ 1 | 0 | 0 | 0 | 1 |
| ✓ 2 | 0 | 0 | 1 | 0 |
| ✓ 8 | 1 | 0 | 0 | 0 |
| ✓ 3 | 0 | 0 | 1 | 1 |
| ✓ 5 | 0 | 1 | 0 | 1 |
| ✓ 10 | 1 | 0 | 1 | 0 |
| ✓ 7 | 0 | 1 | 1 | 1 |
| ✓ 14 | 1 | 1 | 1 | 0 |
| 15 | 1 | 1 | 1 | 1 |

1st-2

| minterm | A | B | C | D |
|---|---|---|---|---|
| ✓ (0,1) | 0 | 0 | 0 | — |
| ✓ (0,2) | 0 | 0 | — | 0 |
| ✓ (0,8) | — | 0 | 0 | 0 |
| ✓ (1,3) | 0 | 0 | — | 1 |
| ✓ (1,5) | 0 | — | 0 | 1 |
| ✓ (2,3) | 0 | 0 | 1 | — |
| ✓ (2,10) | — | 0 | 1 | 0 |
| ✓ (8,10) | 1 | 0 | — | 0 |
| ✓ (3,7) | 0 | — | 1 | 1 |
| ✓ (5,7) | 0 | 1 | — | 1 |
| (10,14) | 1 | — | 1 | 0 |
| (7,15) | — | 1 | 1 | 1 |
| (14,15) | 1 | 1 | 1 | — |

1st-3

| minterm | A | B | C | D |
|---|---|---|---|---|
| 0,1,2,3) | 0 | 0 | — | — |
| 0,2,8,10) | — | 0 | — | 0 |
| 1,3,5,7) | 0 | — | — | 1 |

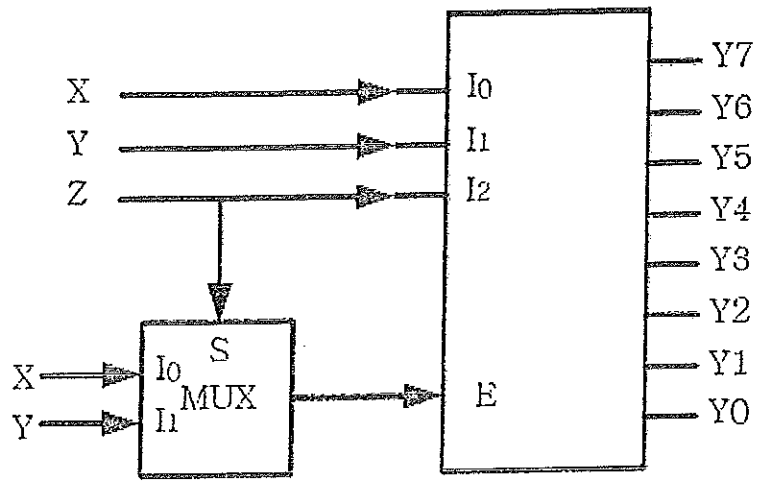
| | 0 ✓ | 1 ✓ | 2 ✓ | 3 ✓ | 5 ✓ | 7 ✓ | 8 ✓ | 10 ✓ | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0,1,2,3 | X | X | X | X | | | (X) | X | | |
| 0,2,8,10 | X | | X | | | | | | | |
| 1,3,5,7 | | X | | X | (X) | X | | X | X | |
| 10,14 | | | | | | X | | | X | X |
| 7,15 | | | | | | ✓ | | | | |
| 14,15 | | | | | | | | | | |

$F = -0-0 + 0--1 + 111-$

$= BD' + A'D + ABC$

**Q3] 12 points**

For the shown logic diagram, determine for each input combination of X, Y and Z the value of each output, Y7 through Y0.




| X | Y | Z | Y7 | Y6 | Y5 | Y4 | Y3 | Y2 | Y1 | Y0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

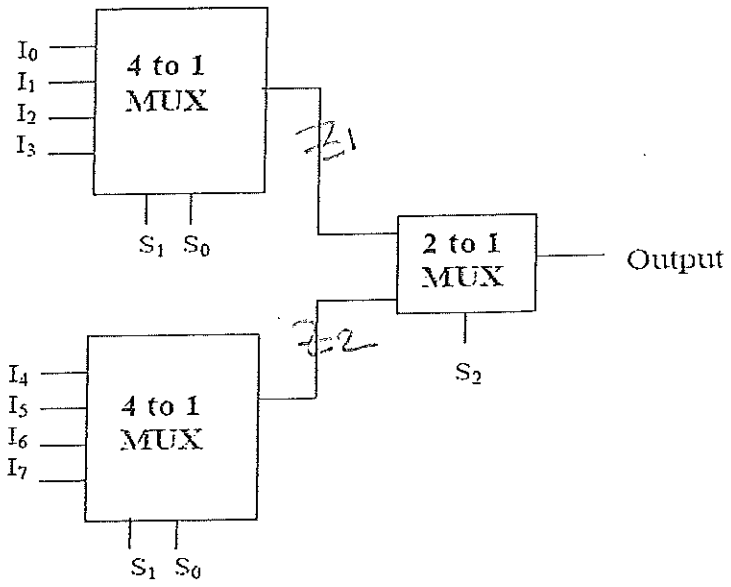
**Q4] 10 points**

Many offices and buildings use combination locks to control entry. As the design engineer of the *Wonderful Door Security Company,* you are asked to implement a door security system by using a card reader. There **are four inputs to the card reader**: inputs X, Y, and Z are used to validate the correct door code, and input V is used to check if the card reader is still valid. After the card reader is being read by the system, there **are three outputs** to this system: alarm (A), door open (D), and Error (E). Door (D) will only open when the decimal value of the binary inputs (x, y, z) is odd (عدد فردي) **AND** the card reader is valid. The Error (E) signal goes on when the code on the card is correct (i.e. decimal value equal to odd) but the card is no longer valid. Finally, the alarm (A) will trigger when the code is incorrect (i.e. decimal value equal to even). **Derive the truth table only**

| X | Y | Z | V | A | D | E |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 | 0 |

**Q5] 15 points**

For the system shown in the following figure




1. **(5 points)** Write a Verilog HDL code to describe the module mux4×1
2. **(4 points)** Write a Verilog HDL code to describe the module mux2×1
3. **(6 points)** Write a Verilog HDL code to describe the whole system structurally from its subsystems

(1)

```
module mux41 ( A,B,C,D,X,Y,F);
input A,B,C,D,X,Y;
output F;
wire w1,w2;
wire w3,w4,w5,w6;
not G1(w1,X);
not G2(w2,Y);
and G3(w3,A,w1,w2);
and G4(w4,B,w1,Y);
and G5(w5,C,Y10,w2);
and G6(w6,D,X,Y);
or G7(F,w3,w4,w5,w6);
```

// X selection
Y

// A
B
C
D inputs

[2]

```
module mux21(A,B,S,Y);
output Y;
input A,B,S;
wire W1,W2,W3;
not G1(W1,S);
and G2(W2,A,W1);
and G3(W3,B,S);
or G4(Y,W2,W3);
endmodule
```

[3]

```
module wholesystem(I0,I1,I2,I3,I4,I5,I6,I7,S0,S1,S2,Output);
output Output;
input I0,I1,I2,I3,I4,I5,I6,I7,S0,S1,S2;
wire Z1,Z2;

mux41 G1(I0,I1,I2,I3,S1,S0,Z1);
mux41 G2(I4,I5,I6,I7,S1,S0,Z2);
mux21 G3(Z1,Z2,S2,Output);
endmodule
```