Marking Scheme

Total No. of Questions – [04] Total No. of Printed Pages - [03]

U218-145(T1)

G.R. No.	

OCTOBER 2018/ IN-SEM (T1)

S. Y. B. TECH. (INFORMATION TECHNOLOGY) (SEMESTER - I)

COURSE NAME: DIGITAL ELECTRONICS AND LOGIC DESIGN

COURSE CODE: ITUA21175

(PATTERN 2017)

Time: [**1 Hour**] [Max. Marks: **30**]

(*) Instructions to candidates:

1) Answer Q.1 OR Q.2 and Q.3 OR Q.4.
2) Figures to the right indicate full marks.
3) Use of scientific calculator is allowed
4) Use suitable data where ever required

Q.1) a) Convert the following numbers:

i. $(2598.675)_{10} = (\ \ )_{16}$
ii. $(110101\ .101010)_{2} = (\ \ )_{8}$
iii. $(A72E)_{16} = (\ )_{8}$ **[6 marks]**

Marking Scheme: Step-by-step conversion [2 marks each]

b) Construct Hamming Code for the following 8-bits word. Use Even Parity.
i. 10101010
ii. 00000000
iii.11111111 **[6 marks]**

Marking Scheme: Hamming code format, show parity bit position, Step-by-step conversion [2 marks each]

c) Define Boolean algebra & Boolean Expression. Explain any 4 basic properties of Boolean algebra? **[4 marks]**

Marking Scheme: Definition Boolean algebra & Boolean Expression [2 marks], any 4 basic properties of Boolean algebra [2 marks].

OR

Q.2) a) Explain in short with respect to radix, symbols used, weight assigned to position, example, and applications of the following:

i) Binary Number System ii) Hexadecimal Number System

iii) Octal Number System **[6 marks]**

Marking Scheme: Explanation wrt radix, symbols used, weight assigned to position, example, and applications. [2 marks each]

b) Perform following arithmetic using 2's Complement:

i) $(7)_{10} - (11)_{10}$ ii) $(-7)_{10} - (11)_{10}$

iii) $(-7)_{10} + (11)_{10}$ **[6 marks]**

Marking Scheme: Step-by-step solution to the above arithmetic and correct answer. [2 marks each]

c) Minimize the 4-variable logic function using K-map technique.

f (A,B,C,D) = ABC'D + A'BCD + A'B'C' + A'B'D' + AC' + B' **[4 marks]**

Marking Scheme: 4 variable K-map, grouping 1's, and minimized expression.

Q.3) a) Explain Adder with Look-Ahead carry. Consider 4-bit adder and formulate Boolean expressions for the carry (C_0, C_1, C_2, C_3). **[6 marks]**

Marking Scheme: Explanation, Diagram, Boolean Expressions (C_0, C_1, C_2, C_3).

b) Design (Truth Table, Logic Function, Circuit Diagram) Full Adder using MUX IC 74153. **[4 marks]**

Marking Scheme: Truth Table, Logic Function, Circuit Diagram using MUX IC 74153.

c) Define Digital Comparators. Draw and discuss n-bit comparator. **[4 marks]**

Marking Scheme: Definition, Diagram, Working, and Example.

OR

Q.4) a) Design (Truth Table, K-map, Boolean expressions, Circuit Diagram) and draw the 4-bit Code Converter circuit for converting Excess-3 Code to BCD Code. **[6 marks]**

Marking Scheme: Truth Table, K-map, Boolean expressions, Circuit Diagram.

b) Design a Full Adder circuit using logic gates. Show Truth Table, K-map simplification and Circuit diagram. **[4 marks]**

Marking Scheme: Truth Table, K-map simplification and Circuit diagram.

c) Discuss the 8-bit Parity generator/checker circuit IC 74180. **[4 marks]**

Marking Scheme: 8-bit Parity generator/checker circuit IC 74180 diagram, working, Function table, example.

Total No. of Questions – [04] Total No. of Printed Pages - [02] 11

U218-145 (T1)

G.R. No.	

OCTOBER 2018/ IN-SEM (T1)

S. Y. B. TECH. (Information Technology) (SEMESTER - I)

COURSE NAME: Digital Electronics and Logic Design

COURSE CODE: ITUA21175

(PATTERN 2017)

Time: [**1 Hour**] [Max. Marks: **30**]

(*) Instructions to candidates:

1) Answer Q.1 OR Q.2 and Q.3 OR Q.4.
2) Figures to the right indicate full marks.
3) Use of scientific calculator is allowed
4) Use suitable data where ever required

Q.1) a) Convert (step-by-step conversion) the following numbers:

i. $(2598.675)_{10} = (\ )_{16}$

ii. $(110101\ .101010)_{2} = (\ )_{8}$

iii. $(A72E)_{16} = (\ )_{8}$ **[6 marks]**

Marking Scheme: Step-by-step conversion [2 marks each]

Solution: i. $(2598.675)_{10} = (A26\ .\ AC)_{16}$

ii. $(110101\ .101010)_{2} = (65.52)_{8}$

iii. $(A72E)_{16} = (123456)_{8}$

i. Integer part

Divis	Dividend	Remain
16	2598	
16	162	6
16	10	2
16	10	A

Fractional part:

Decimal		Product	Integer part
0.675 x 16	=	10.8	10 (A)
0.8 x 16	=	12.8	12 (C)

ii. $(110101\ .101010)_{2}$ = (110 101. 101 010)

= $(65.52)_{8}$

iii. $(A\ 7\ 2\ E)_{16}$ = (1010 0111 0010 1110)

= 001 010 011 100 101 110

= (123456)₈

b) Construct Hamming Code for the following 8-bits word. Use Even Parity.
i. 10101010
ii. 00000000
iii. 11111111 **[6 marks]**

Marking Scheme: Hamming code format, show parity bit position, Step-by-step conversion [2 marks each]

Solution: i) 111101001010 ii) 000000000000 iii) 111011101111

Bit position →	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Encoded data →	P_1	P_2	d_1	P_4	d_2	d_3	d_4	P_8	d_5	d_6	d_7	d_8	d_9	d_{10}	d_{11}	P_{16}	d_{12}
i)	1	1	1	1	0	1	0	0	1	0	1	0					
ii)	0	0	0	0	0	0	0	0	0	0	0	0					
iv)	1	1	1	0	1	1	1	0	1	1	1	1					

$P_1 = d_1, d_2, d_4, d_5, d_7, d_9, d_{11}, d_{12}, d_{14}$
$P_2 = d_1, d_3, d_4, d_6, d_7, d_{10}, d_{11}, d_{13}, d_{14}$
$P_4 = d_2, d_3, d_4, d_8, d_9, d_{10}, d_{11}, d_{15}$

$P_8 = d_5, 6, 7, 8, 9, 10, 11$
$P_{16} = d_{12}, 13, 14, 15,$

c) Define Boolean algebra & Boolean Expression. Explain any 4 basic properties of Boolean algebra? **[4 marks]**

Marking Scheme: Definition Boolean algebra & Boolean Expression [2 marks], any 4 basic properties of Boolean algebra [2 marks].

Solution: i) A system of algebra that operates on Boolean variables. The binary nature of Boolean algebra makes it useful for analysis, simplification and design of logic circuits. The basic properties of Boolean algebra are commutative property, associative property and distributive property.

Commutative: The commutative property says that binary operations AND and OR may be applied left to right or right to left. (A AND B is the same as B AND A; A OR B is the same as B OR A.)

Associative: The associative property says that given three Boolean variables, they may be ANDed or ORed right to left or left to right. ((A AND B) AND C is the same as A AND (B AND C); (A OR B) OR C is the same as A OR (B OR C).)

Distributive: The distributive property says that given three Boolean variables, the first AND the result of the second OR the third is the same as the first AND the second OR the first AND the third. (A AND (B OR C) = (A AND B) OR (A AND C). Also, the first OR the result of second AND the third is the same as the first OR the second AND the result of the first OR the third. (A OR (B AND C) = (A OR B) AND (A OR C).)

Identity: The identity property says that any value A AND the OR identity always returns A and that any value A OR the AND identity always returns A. (A AND 1 = A; A OR 0 = A.)

Complement: The complement property says that any value AND the compliment of that value equals the OR identity and that any value OR the compliment of that value equals the OR identity. (A AND (A') = 0; A OR (A') = 1.)

OR

Q.2) a) Explain in short with respect to radix, symbols used, weight assigned to position, example, and applications of the following:

i) Binary Number System ii) Hexadecimal Number System

iii) Octal Number System **[6 marks]**

Marking Scheme: Explanation w.r.t. radix, symbols used, weight assigned to position, example, and applications. [2 marks each]

Number System	**Base Radix**	**Symbols used**	**Weight assigned position**		**Example**
Binary	2	0 , 1	2^{i}	2^{-f}	1011.11
Octal	8	0,1,2,3,4,5,6,7	8^{i}	8^{-f}	3567.25
Decimal	10	0,1,2,3,4,5,6,7,8,9	10^{i}	10^{-f}	3974.98
Hexadecimal	16	0,1,2,3,4,5,6,7,8,9,A,B,C, D,E,F	16^{i}	16^{-f}	3FA9.56

b) Perform following arithmetic using 2's Complement:

i) $(7)_{10} - (11)_{10}$ ii) $(-7)_{10} - (11)_{10}$

iii) $(-7)_{10} + (11)_{10}$ **[6 marks]**

Marking Scheme: Step-by-step solution to the above arithmetic and correct answer.

Solution: [2 marks each]

i) (7) = 0111 (11) = 1011

2's compliment of (11) = 0101

0111 + 0101 = 1100 ; no carry generated, so the answer is –ve and in 2's comp. form.

ii) (-7) = 0111 (-11) = 1011
2's compliment of (-7) = 1001
2's compliment of (-11) = 0101

1001 + 0101 = 1110 ; sign bit is not same as operands, so there's a problem of overflow.

iii) (-7) = 0111 (11) = 1011
2's compliment of (-7) = 1001

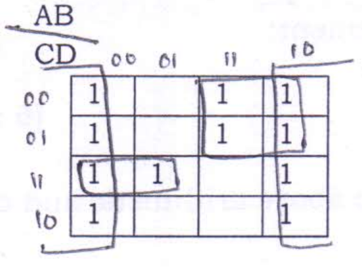
1001 + 1011 = 1 0100 ; discard the carry

c) Minimize the 4-variable logic function using K-map technique.
f (A,B,C,D) = ABC'D + A'BCD + A'B'C' + A'B'D' + AC' + B' **[4 marks]**

Marking Scheme: 4 variable K-map, grouping 1's, and minimized expression.

Solution:

1) Enter 1 in the cell with A=1, B=1, C=0, D=1 corresponding to the minterm ABC'D.
2) Enter 1 in the cell with A=0, B=1, C=1, D=1 corresponding to the minterm A'BCD.
3) Enter 1 in the two cell with A=0, B=0, C=0 corresponding to the minterm A'B'C'.
4) Enter 1 in the two cell with A=0, B=0, D=0 corresponding to the minterm A'B'D'.
5) Enter 1 in the four cell with A=1, C=0 corresponding to the minterm AC'.
6) Enter 1 in the eight cell with B=0 corresponding to the minterm B'.



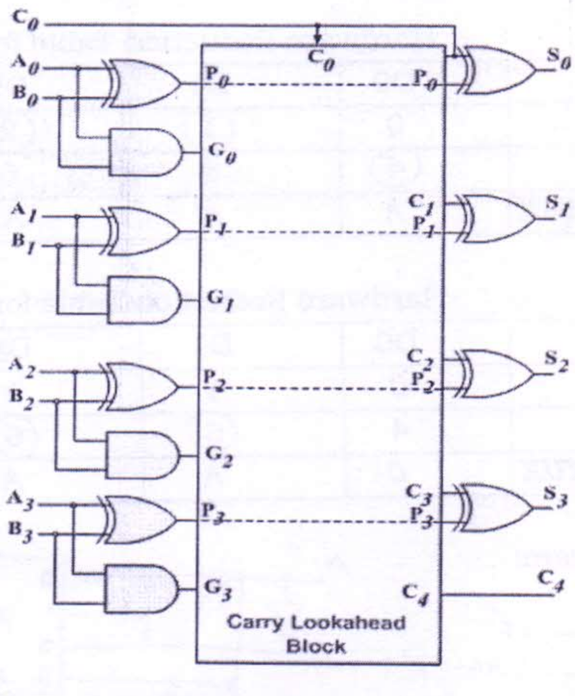
CD \ AB	00	01	11	10
00	1		1	1
01	1		1	1
11	1	1		1
10	1			1

Y = B' + AC' + A'CD

Q.3) a) Explain Adder with Look-Ahead carry. Consider 4-bit adder and formulate Boolean expressions for the carry (C_0, C_1, C_2, C_3). **[6 marks]**

Marking Scheme: Explanation, Diagram, Boolean Expressions (C_0, C_1, C_2, C_3).

- Solution: In this circuit, the 2 internal signals Pi and Gi are given by:
  - Pi = Ai ⊕Bi ....(1)
  - Gi = Ai . Bi ....(2)
- The output sum and carry can be defined as :
  - Si = Pi ⊕ Ci ....(3)
  - Ci +1 = Gi + Pi.Ci ....(4)
- Gi is known as the carry Generate signal since a carry (Ci+1) is generated whenever Gi=1, regardless of the input carry (Ci).
- Pi is known as the carry propagate signal since whenever Pi =1, the input carry is propagated to the output carry, i.e., Ci+1 = Ci (note that whenever Pi =1, Gi =0).
- Computing the values of Pi and Gi only depend on the input operand bits (Ai & Bi) as clear from the Figure and equations.
- Thus, these signals settle to their steady-state value after the propagation through their respective gates.
- The Boolean expression of the carry outputs of various stages can be written as follows:
  - C1 = G0 + P0C0
  - C2 = G1 + P1C1 = G1 + P1 (G0 + P0C0)
    = G1 + P1G0 + P1P0C0
  - C3 = G2 + P2C2 = G2 + P2G1 + P2P1G0 + P2P1P0C0
  - C4 = G3 + P3C3
    = G3 + P3G2 + P3P2G1 + P3P2P1G0 + P3P2P1P0C0



b) Design (Truth Table, Logic Function, Circuit Diagram) Full Adder using MUX IC 74153. **[4 marks]**

Marking Scheme: Truth Table, Logic Function, Circuit Diagram using MUX IC 74153.

Solution: Truth Table

Inputs			S	C
A	B	C		
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Sum $= \sum m\ (1, 2, 4, 7)$

Carry $= \sum m\ (3, 5, 6, 7)$

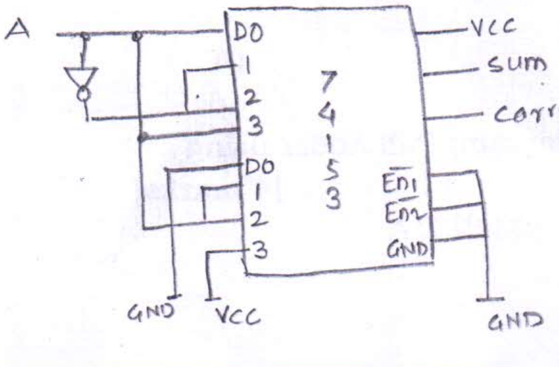
Hardware Reduction Table for Sum:

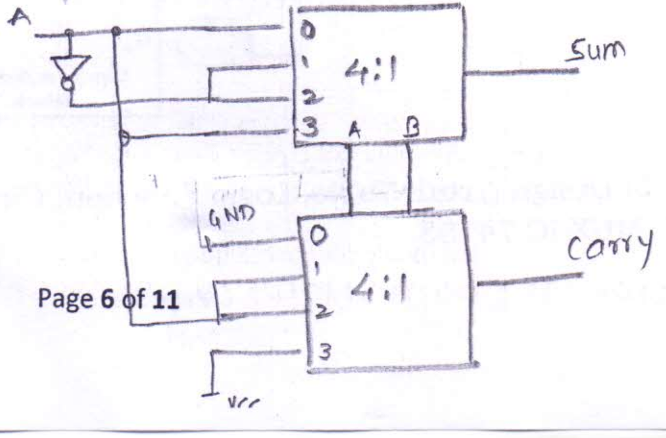
	D0	D1	D2	D3
A!	0	1	2	3
A	4	5	6	7
Input to MUX	*A*	*A!*	*A!*	*A*

Hardware Reduction Table for Carry:

	D0	D1	D2	D3
A!	0	1	2	3
A	4	5	6	7
Input to MUX	*0*	*A*	*A*	*1*

Circuit diagram:





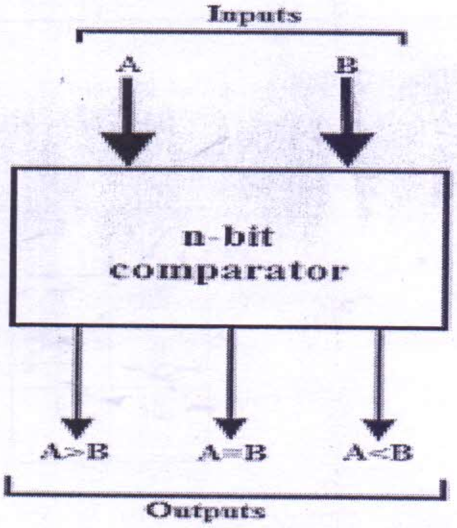
c) Define Digital Comparators. Draw and discuss n-bit comparator. **[4 marks]**

Marking Scheme: Definition, Diagram, Working, example.

Solution: Comparators can be designed for comparing multibit numbers. It receives two n-bit numbers A and B as input and the outputs are A > B, A = B, and A < B. Depending on the magnitude of the two numbers, one of the

output will be HIGH.

n-bit comparator:



OR

Q.4) a) Design (Truth Table, K-map, Boolean expressions, Circuit Diagram) and draw the 4-bit Code Converter circuit for converting Excess-3 Code to BCD Code. **[6 marks]**

Marking Scheme: Truth Table, K-map, Boolean expressions, Circuit Diagram.

Solution: Truth Table

INPUT (EXCESS-3 CODE)				OUTPUT (BCD CODE)			
E_3	E_2	E_1	E_0	B_3	B_2	B_1	B_0
0	0	0	0	X	X	X	X
0	0	0	1	X	X	X	X
0	0	1	0	X	X	X	X

0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	1
0	1	0	1	0	0	1	0
0	1	1	0	0	0	1	1
0	1	1	1	0	1	0	0
1	0	0	0	0	1	0	1
1	0	0	1	0	1	1	0
1	0	1	0	0	1	1	1
1	0	1	1	1	0	0	0
1	1	0	0	1	0	0	1
1	1	0	1	X	X	X	X
1	1	1	0	X	X	X	X
1	1	1	1	X	X	X	X

K-Map For Reduced Boolean Expressions Of Each Output

$B3 = E3E2 + E3E1E0$

E3E2 \ E1E0	00	01	11	10
00	X	0	1	0
01	X	0	X	0
11	0	0	X	1
10	X	0	X	0

$B2 = \overline{E2}\,\overline{E1} + E2E1E0 + E3E1\overline{E0}$

E3E2 \ E1E0	00	01	11	10
00	X	0	0	1
01	X	0	X	1
11	0	1	X	0
10	X	0	X	1

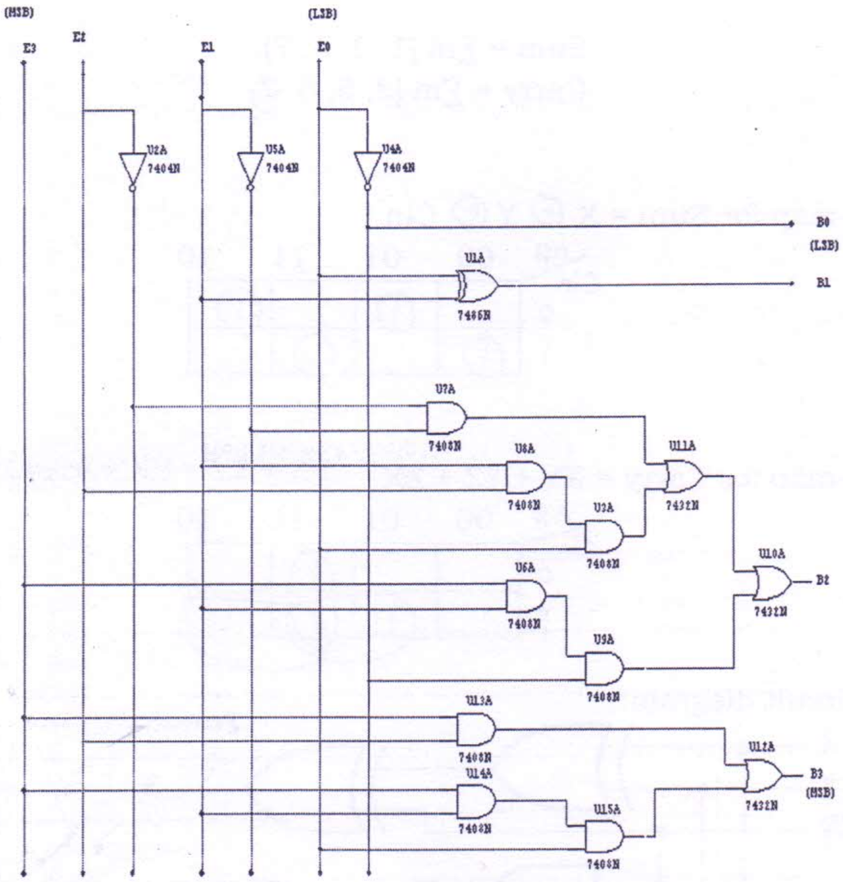
$B1 = \overline{E1}E0 + E1\overline{E0}$

E3E2 \ E1E0	00	01	11	10
00	X	0	0	0
01	X	1	X	1
11	0	0	X	0
10	X	1	X	1

$B0 = \overline{E0}$

E3E2 \ E1E0	00	01	11	10
00	X	1	1	1
01	X	0	X	0
11	0	0	X	0
10	X	1	X	1

Circuit Diagram



b) Design a Full Adder circuit using logic gates. Show Truth Table, K-map simplification and Circuit diagram. **[4 marks]**

Marking Scheme: Truth Table, K-map simplification and Circuit diagram.

Solution: Truth Table

Inputs			Sum	Carry
A	B	C		
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1

1	1	1	1	1

Sum = ∑m (1, 2, 4, 7)
Carry = ∑m (3, 5, 6, 7)

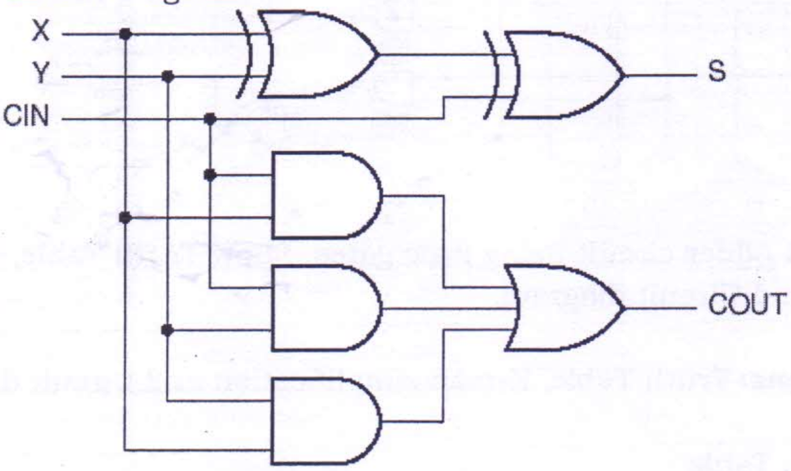
k-map for Sum = X ⊕ Y ⊕ Cin

Cin \ AB	00	01	11	10
0		1		1
1	1		1	

k-map for Carry = XY + YZ + ZX

Cin \ AB	00	01	11	10
0			1	
1		1	1	1

Circuit diagram:



c) Discuss the 8-bit Parity generator/checker circuit IC 74180. **[4 marks]**

Marking Scheme: 8-bit Parity generator/checker circuit IC 74180 diagram, working, Function table, example.

Solution:

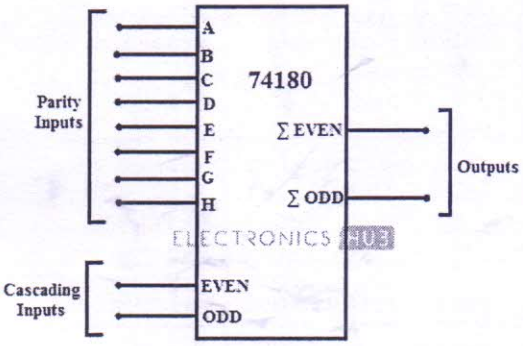
A parity generator is a combinational logic circuit that generates the parity bit in the transmitter. On the other hand, a circuit that checks the parity in the receiver is called parity checker. A combined circuit or devices of parity generators and parity checkers are commonly used in digital systems to detect the single bit errors in the transmitted data word.

The sum of the data bits and parity bits can be even or odd. In even parity, the added parity bit will make the total number of 1s an even amount whereas in odd parity the added parity bit will make the total number of 1s odd amount.

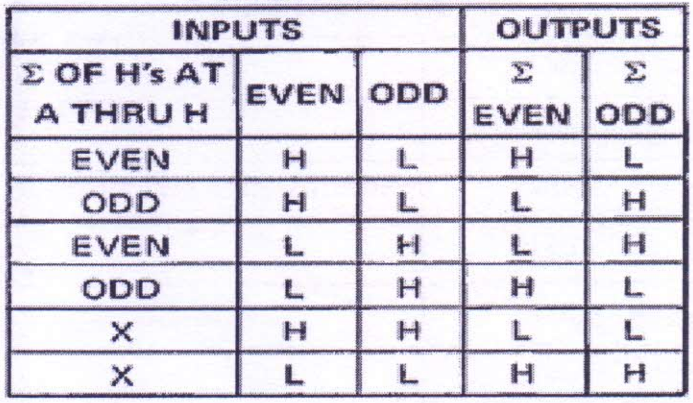
It is a 9-bit parity generator or checker used to detect errors in high speed data transmission or data retrieval systems. The figure below shows the pin diagram of 74180 IC.

This IC can be used to generate a 9-bit odd or even parity code or it can be used to check for odd or even parity in a 9-bit code (8 data bits and one parity bit).

This IC consists of eight parity inputs from A through H and two cascading inputs. There are two outputs even sum and odd sum. In implementing generator or checker circuits, unused parity bits must be tied to logic zero and the cascading inputs must not be equal.



FUNCTION TABLE



INPUTS			OUTPUTS	
Σ OF H's AT A THRU H	EVEN	ODD	Σ EVEN	Σ ODD
EVEN	H	L	H	L
ODD	H	L	L	H
EVEN	L	H	L	H
ODD	L	H	H	L
X	H	H	L	L
X	L	L	H	H

H = high level, L = low level, X = irrelevant