Combinational logic CKT

* Outputs and inputs values only
  + Inputs through combinational logic CKT lead to the output
  + EX of a circuit on the board
  + Truth table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| abc | t1 | t2 | t3 | t4 | f1 | F2 |
| 000 | 0 | 0 | 0 | 0 | 0 | 0 |
| 001 | 0 | 0 | 1 | 0 | 0 | 1 |
| 010 | 0 | 0 | 1 | 0 | 0 | 1 |
| 011 | 0 | 1 | 1 | 1 | 1 | 1 |
| 100 | 0 | 0 | 0 | 0 | 0 | 0 |
| 101 | 0 | 0 | 1 | 0 | 0 | 1 |
| 110 | 1 | 0 | 1 | 1 | 1 | 1 |
| 111 | 1 | 0 | 1 | 1 | 1 | 1 |

* Design Procedure
  + Problem is stated
  + # of avail inputs that is needed and # of outputs
  + Assign letter symbols to the inputs and outputs
  + Truth table that defines the relationship between the ins and outs
  + Simplfy the boolean expression for each output is obtained
  + Logic diagrams is drawn
  + Ex: Design a ckt – BCD to gray code conversion (Ch 1 pg 22)
    - A B C D (inputs) W X Y Z (outputs)
    - Truth tables

|  |  |
| --- | --- |
| A B C D | W X Y Z |
| 0000 | 0000 |
| 0001 | 0001 |
| 0010 | 0011 |
| 0011 | 0010 |
| 0100 | 0110 |
| 0101 | 0111 |
| 0110 | 0101 |
| 0111 | 0100 |
| 1000 | 1100 |
| 1001 | 1101 |
| 1010 | XXXX |
| 1011 | XXXX |
| 1100 | XXXX |
| 1101 | XXXX |
| 1110 | XXXX |
| 1111 | XXXX |

* + - Kmaps (Map1 = W = A)(Map 2 = X = A + B ) (Map 3 = Y = BC' + B'C) (Map 4 = Z = C'D + CD' )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ab\cd | 00 | 01 | 11 | 10 |
| 00 |  |  |  |  |
| 01 |  |  |  |  |
| 11 | x | x | x | x |
| 10 | 1 | 1 | x | x |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ab\cd | 00 | 01 | 11 | 10 |
| 00 |  |  |  |  |
| 01 | 1 | 1 | 1 | 1 |
| 11 | x | x | x | x |
| 10 | 1 | 1 | x | x |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ab\cd | 00 | 01 | 11 | 10 |
| 00 |  |  | 1 | 1 |
| 01 | 1 | 1 |  |  |
| 11 | x | x | x | x |
| 10 |  |  | x | x |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ab\cd | 00 | 01 | 11 | 10 |
| 00 |  | 1 |  | 1 |
| 01 |  | 1 |  | 1 |
| 11 | x | x | x | x |
| 10 |  | 1 | x | x |

* + - Drew a ckt design on the board. Uses 2 exclusive or gates and one or gate.
* Adders
  + Half-Adder – A CKT that adds 2 bits and gives sum and Carry
    - Truthtable

|  |  |
| --- | --- |
| X y | (Carry) (Sum) |
| 00 | 0 0 |
| 01 | 01 |
| 10 | 01 |
| 11 | 10 |

* + - S = x'y + xy' - x (exclusive or) y
    - C = xy
    - Drew the CKT design on board. Use one exclusive or and 1 and for x and y to S. and 1 and gate for x and y to C
  + Full adder – combo ckt -> adds 3 bits => sum and carry
    - Truthtable

|  |  |
| --- | --- |
| X y z | (Carry) (Sum) |
| 000 | 0 0 |
| 001 | 0 1 |
| 010 | 0 1 |
| 011 | 1 0 |
| 100 | 0 1 |
| 101 | 1 0 |
| 110 | 1 0 |
| 111 | 1 1 |

* + - Kmaps ( Map1 = S = x'y'z + x'yz' + xy'z' + xyz) (Map 2 = C = yz + xz + xy)

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

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

* + - Drew CKT on board. S uses 6 inverters and 4 and gates and then 1 big or gate. C uses 3 and gates then one big or gate
  + Full Adder => 2 half adders & 1 or gate
  + Binary Adder
    - A : A3 A2 A1 A0
    - B: B3 B2 B1 B0

LAB 4

Design a counter 0 -> 3 -> 0

|  |  |
| --- | --- |
| Ins | Outs |
| A B | X Y |
| 0 0 | 0 1 |
| 0 1 | 1 0 |
| 1 0 | 1 1 |
| 1 1 | 0 0 |