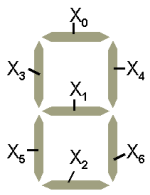
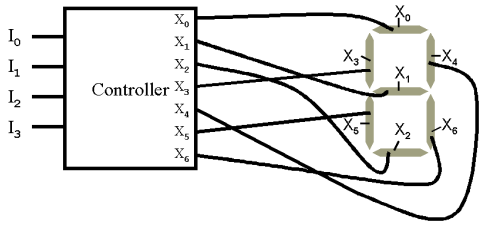
Sample Design: 7 Segment Controller

Build a circuit that accepts as input a binary encoded integer between 0 and 9 inclusive and outputs control lines that can be attached to a 7 segment display

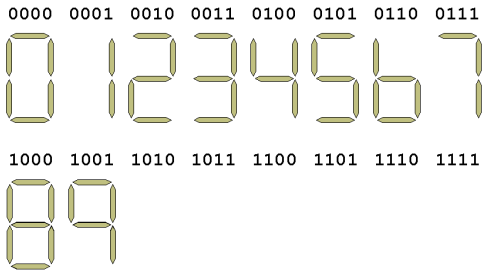


1



2

Defining the desired outputs



3

Design Steps

1. Define each output as a boolean function.
   - Truth table makes sense for this problem.
2. Derive SOP expression for each output.
   - We can do this easily given a truth table.
3. Minimization
   - Karnaugh Maps
4. Draw the circuit

4

Truth Table for X_0

- X_0 controls the top segment, which must be turned on (1) for the numbers: 0,2,3,5,7,8,9

$I_0I_1I_2I_3$:

0000, 0010, 0011, 0101, 0111, 1000, 1001

0 2 3 5 7 8 9

- X_0 should be off (0) for the numbers: 1,4,6

$I_0I_1I_2I_3$:

0001, 0100, 0110

1 4 6

5

I_0	I_1	I_2	I_3	X_0
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1

I_0	I_1	I_2	I_3	X_0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	?
1	0	1	1	?
1	1	0	0	?
1	1	0	1	?
1	1	1	0	?
1	1	1	1	?

6

Input patterns 1010 -> 1111

- It's never defined which displays should be turned on for many possible input patterns.
- We don't have to worry about this – we just make sure that for the defined input patterns, the outputs turned on are correct.

7

Deriving SOP

- One *term* for each row of the truth table that has the output on (1).
- We have 7 rows that are set to 1.
- This will be a big expression!

8

SOP terms

- To come up with a single term, we AND together all the inputs, although some will be negated.
- The inputs that have the value 0 in the row we are looking at will be negated.
- The resulting term will (AND) will be a 1 only if the inputs correspond to our row.

9

I_0	I_1	I_2	I_3	X_0
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1

SOP term derivation from a row in the truth table

I_0 must be a 0, so we write $\overline{I_0}$
I_1 must be a 0, so we write $\overline{I_1}$
I_2 must be a 1, so we write I_2
I_3 must be a 0, so we write $\overline{I_3}$

The term is: $\overline{I_0} \bullet \overline{I_1} \bullet I_2 \bullet \overline{I_3}$ or simply: $\overline{I_0}\,\overline{I_1} I_2 \overline{I_3}$

10

Repeat for all other rows that are 1 in the truth table.

- We end up with 7 terms, and we OR them all together to get an expression for X_0

$$X_0 = \overline{I_0}\,\overline{I_1}\,\overline{I_2}\,\overline{I_3} + \overline{I_0}\,\overline{I_1} I_2 \overline{I_3} + \overline{I_0}\,\overline{I_1} I_2 I_3 + \overline{I_0} I_1 \overline{I_2} I_3 + \overline{I_0} I_1 I_2 I_3 + I_0 \overline{I_1}\,\overline{I_2}\,\overline{I_3} + I_0 \overline{I_1} I_2 I_3$$

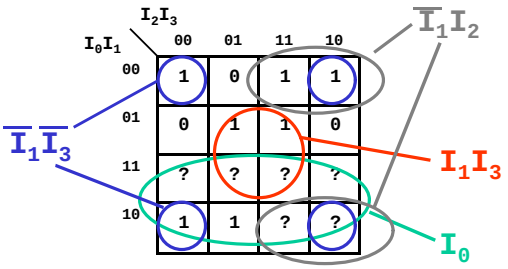
11

Minimization: K-Map

I_0I_1 \ I_2I_3	00	01	11	10
00	1	0	1	1
01	0	1	1	0
11	?	?	?	?
10	1	1	?	?

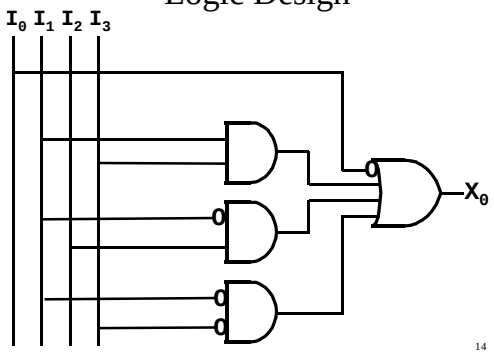
12

Reducing



13

Logic Design



14

6 more segments (outputs)

- So far we have only dealt with X_0, we still need to handle X_1 – X_6!
- Nothing is different – just derive each truth table, SOP expressions, minimize and come up with logic diagram…

15