Prime Number Design Example

Problem Statement: Implement a minimized (SOP) combinational logic circuit to detect if a BCD digit is prime (0,1,2,3,5,7).

1. **Step:** Determine Inputs and Outputs

Inputs:	Outputs:
b_3,b_2,b_1,b_0	Let P=1 indicate Prime
b_3 -BCD MSB, b_0 -BCD LSB	Let P=0 indicate Not Prime

2. **Step:** Derive Truth Table and create outputs

Row	b_3	b_2	b_1	b_0	BCD	**P**
0	0	0	0	0	0	1
1	0	0	0	1	1	1
2	0	0	1	0	2	1
3	0	0	1	1	3	1
4	0	1	0	0	4	0
5	0	1	0	1	5	1
6	0	1	1	0	6	0
7	0	1	1	1	7	1
8	1	0	0	0	8	0
9	1	0	0	1	9	0
10	1	0	1	0	-	X
11	1	0	1	1	-	X
12	1	1	0	0	-	X
13	1	1	0	1	-	X
14	1	1	1	0	-	X
15	1	1	1	1	-	X

X – means “don't care”. These cases should never happen.

3. Step: Determine simplified Boolean expression.

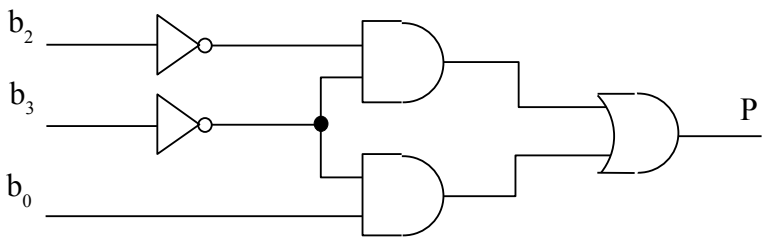
K-Map:

$b_1b_0 \backslash b_3b_2$	00	01	11	10
00	1	0	X	0
01	1	1	X	0
11	1	1	X	X
10	1	0	X	X

$P = \bar{b}_3\bar{b}_2 + \bar{b}_3 b_0$ or $P = \bar{b}_3\bar{b}_2 + b_2 b_0$

4. **Step:** Implement

choosing $P = \bar{b}_3\bar{b}_2 + \bar{b}_3 b_0$



Don't need b_1.

Full design would require specific IC, timing analysis, ...

5. **Step:** Verify

At any step, the designer might have to go back to step 1 and redo parts of the process. This is an iterative process.