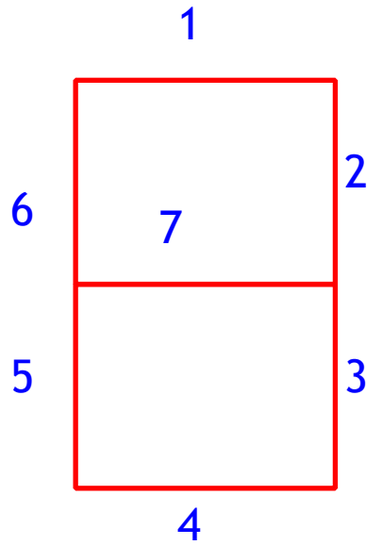
7-Segment Display

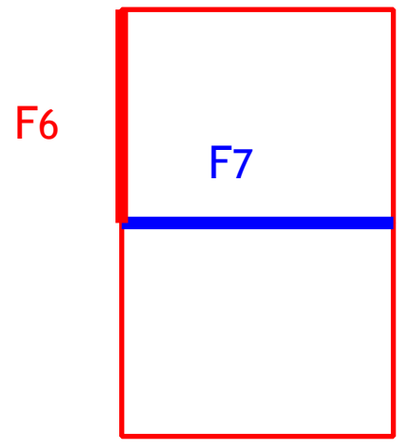
CLASS 13

L E D - 7-segment Display of Decimal Digits

Each segment lights up when the digit we want to create **requires it.**

Here it is:





We will focus on the segments, and write one function for each of the 7 segments, e.g. for:

F_i = 0 <=> segment i is off

F_i = 1 <=> segment i is on

	x	y	z	t	F_6
0	0	0	0	0	1
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	1
5	0	1	0	1	1
6	0	1	1	0	1
7	0	1	1	1	0
8	1	0	0	0	1
9	1	0	0	1	1
10	1	0	1	0	d
.	1	0	1	1	d
.	1	1	0	0	d
.	1	1	0	1	d
.	1	1	1	0	d
15	1	1	1	1	d

F_7 ← HW

What should we do with the function for inputs 10-15, which should never occur in our display? Do we care about the values F_6 gets for those inputs? NO

We therefore don't give a value of 0 or 1 for F_6 for those inputs. We will instead use the letter d ('don't care')

These d's, we will use to our advantage when minimizing the function. **NOTE**: The function we create will have to give a value of 0 or 1 for every possible input-occurring or not.

2^6 possibilities, for d = 0 or d = 1

We minimize 2^6 functions at once!

We put the d's on the K map, with the 1's.

Whether that value will be 0 or 1 will be established so that the function gets the minimal minimal form. It's simpler than it sounds :-)

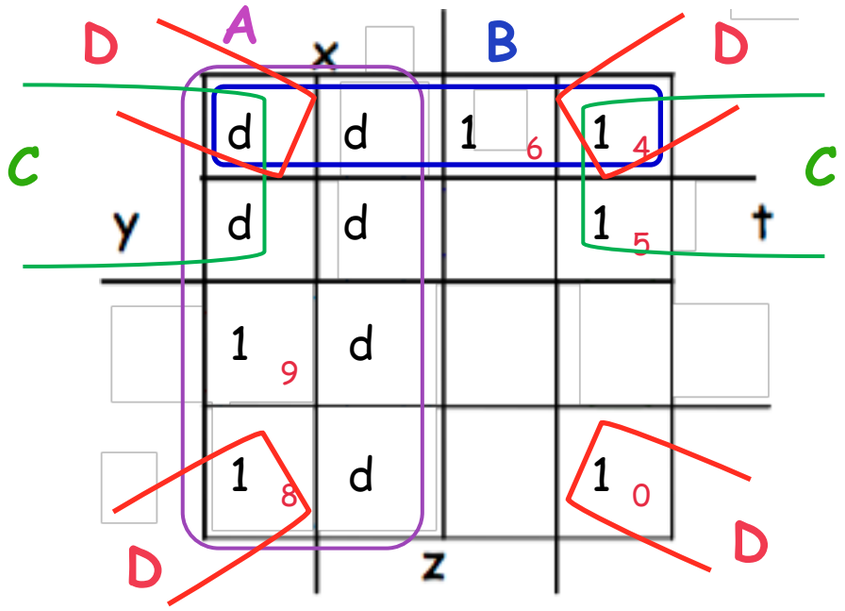
We use the d's to our advantage:

1) when forming implicants, then d = 1, as we want larger implicants- only if they cover at least one 1.

2) when performing the covering, we don't have to cover the d's, so d = 0 outside the minimal form.

Let's draw K map + form the prime implicants as a hint for HW 15.1:

From previous page we have:



Prime implicants:

	x	y	z	t	F_6
0	0	0	0	0	1
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	1
5	0	1	0	1	1
6	0	1	1	0	1
7	0	1	1	1	0
8	1	0	0	0	1
9	1	0	0	1	1
10	1	0	1	0	d
.	1	0	1	1	d
.	1	1	0	0	d
.	1	1	0	1	d
.	1	1	1	0	d
15	1	1	1	1	d

HW 15.1

Finish this by going on to finding all minimal forms for F_6.

HW 15.2

Find all minimal forms for F_7.