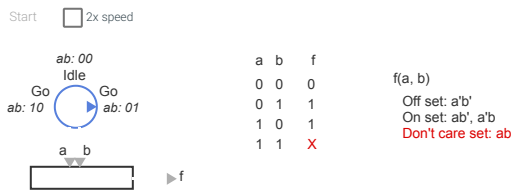
12.10 Don't cares

Incompletely specified functions

An ***incompletely specified function*** does not define an output value for every input combination. Ex: A 3-position knob may set 2 inputs to 00, 01, or 10. Combination 11 is not possible and thus f is not specified for that combination.

All possible minterms of a function can be divided into an ***on set*** (function outputs 1), ***off set*** (function outputs 0), and ***don't care set*** (function output is not specified). In a truth table or on a K-map, don't care minterms are indicated with an X.

PARTICIPATION ACTIVITY 12.10.1: An incompletely specified function.



PARTICIPATION ACTIVITY 12.10.2: Incompletely specified functions.

Indicate whether the function f having inputs a, b is completely specified.

1) f is 1 for a'b', 1 for a'b, 1 for ab', and 0 for ab.
   - ○ Complete
   - ○ Incomplete

2) f is 1 for a'b', and 0 otherwise.
   - ○ Complete
   - ○ Incomplete

3) f is 1 for a'b', and for ab.
   - ○ Complete
   - ○ Incomplete

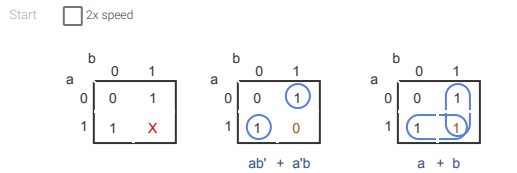
4) f is 1 for a'b', 1 for a'b, and 0 for ab'.
   - ○ Complete
   - ○ Incomplete

5) For a'b', f is 1. For ab, the output value doesn't matter. All other combinations output 0.
   - ○ Complete
   - ○ Incomplete

Minimizing with don't care minterms

Due to the nature of digital circuits, a circuit will output either 0 or 1 for every input value combination. Thus, even for an incompletely specified function, a designer must still choose whether to output 0 or 1 for each don't care minterm. Commonly, designers make the choice that yields a minimized circuit.

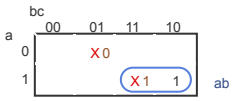
PARTICIPATION ACTIVITY 12.10.3: Minimizing with don't care minterms: Choosing a value that minimizes circuit size.



When drawing circles on a K-map, a designer can choose whether an X (don't care) should be 0 or 1. If outputting a 1 allows for a larger circle, then 1 is a better choice, leading to fewer literals in a term. Otherwise, outputting 0 is a better choice, leading to fewer terms.

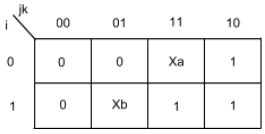
PARTICIPATION ACTIVITY 12.10.4: For don't cares (X's), designers choose to output 1 if that enables a larger circle (and thus smaller term), else choose to output 0 to yield fewer circles (and thus fewer terms).

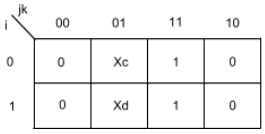
Start 2x speed



PARTICIPATION ACTIVITY 12.10.5: Minimizing with don't care minterms.

Indicate whether each X should be chosen to output 0 or 1.





1) Xa
   - 0
   - 1

2) Xb
   - 0
   - 1

3) Xc
   - 0
   - 1

4) Xd
   - 0
   - 1

Warning

Many designers recommend avoiding incompletely-specified functions (don't cares) except in rare cases. Even though an input combination should never appear, the combination might possibly appear, even briefly—maybe due to electrical noise, due to glitches while switching, or during startup. If the output really does matter in that case, then the designer should choose a safe value. Ex: For the earlier three-position knob example, if f = 1 turns on a device, then 11 is not really a don't care, because the device should not turn on. So f should be set to 0 for input 11, just in case a 11 accidentally appears.

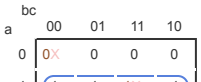
Example: Electronic die

A die can display values 1-6. When rolled, a circuit lights an LED if a high-number is rolled, defined as 4-6. If the die's value is input to the circuit in binary as 3 bits, then combinations 000 and 111 (0 and 7) can't occur. Thus, 000 and 111 are don't care minterms.

PARTICIPATION ACTIVITY 12.10.6: Electronic die: 0 and 7 are don't care minterms.

Start 2x speed

a	b	c	f
0	0	0	X
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	X



PARTICIPATION ACTIVITY 12.10.7: Don't cares example: Electronic die.

Consider the example above.

1) If the designer makes the X in cell 111 a 0, what is f after covering the 1's with circles?
   - ab'
   - ac
   - ab' + ac'

2) If the designer makes the X in cell 111 a

1, what is f after covering the 1's with circles?

- a'
- a
- ab' + ab

3) If the designer makes the X in cell 000 a 1, what additional term would result?

- b'c'
- None

Provide feedback on this section