Digital Logic Optimization II

CSCI 2050U - Computer Architecture

Randy J. Fortier
@randy_fortier

OntarioTech UNIVERSITY

Outline

- Karnaugh maps
  - 2-variable
  - 3-variable
  - 4-variable

Karnaugh Maps - 2 Variable

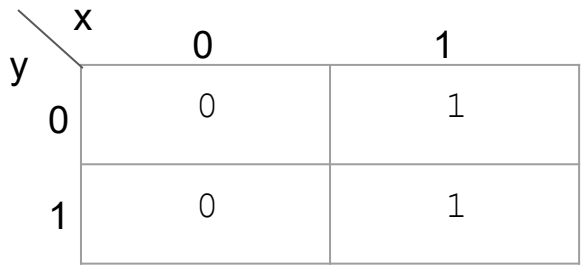
CSCI 2050U - Computer Architecture

OntarioTech UNIVERSITY

Karnaugh Maps (K-Maps)

- A K-map is a systematic way of simplifying Boolean algebraic expressions (in SOP form; alternatively truth tables)
  - First, we draw a sort of coiled-up truth table:

x	***y***	***xy + xy'***
0	0	0
0	1	0
1	0	1
1	1	1



Karnaugh Maps (K-Maps)

- Next, we look for rectangles that are made entirely of 1s
  - We look only for rectangles that are size 2^n (for any n)
  - We prefer rectangles that are larger over smaller rectangles

x	***y***	***xy + xy'***
0	0	0
0	1	0
1	0	1
1	1	1

y \ x	0	1
0	0	1
1	0	1

Karnaugh Maps (K-Maps)

- Next, we look for rectangles that are made entirely of 1s
  - This K-map has 4 spots
  - Does it have any rectangles of size 4? (e.g. 2x2)

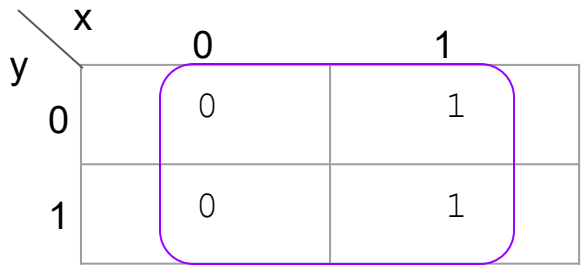
x	***y***	***xy + xy'***
0	0	0
0	1	0
1	0	1
1	1	1

y \ x	0	1
0	0	1
1	0	1

Karnaugh Maps (K-Maps)

- Next, we look for rectangles that are made entirely of 1s
  - This K-map has 4 spots
  - Does it have any rectangles of size 4? (e.g. 2x2) No (there is only one)

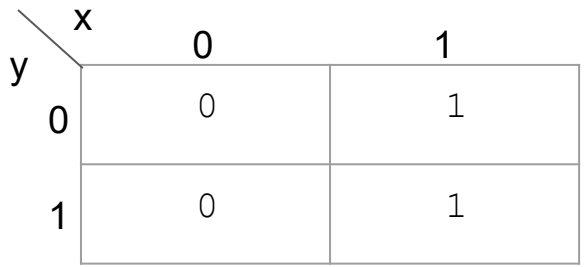
x	***y***	***xy + xy'***
0	0	0
0	1	0
1	0	1
1	1	1



Karnaugh Maps (K-Maps)

- Next, we look for rectangles that are made entirely of 1s
  - This K-map has 4 spots
  - Does it have any rectangles of size 2? (e.g. 1x2 or 2x1)

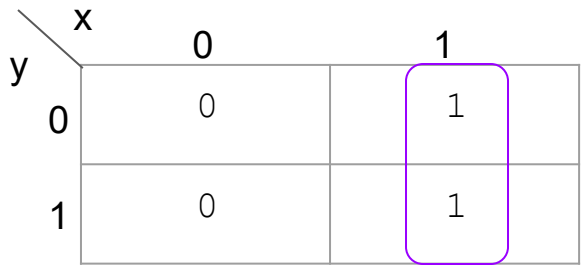
x	***y***	***xy + xy'***
0	0	0
0	1	0
1	0	1
1	1	1



Karnaugh Maps (K-Maps)

- Next, we look for rectangles that are made entirely of 1s
  - This K-map has 4 spots
  - Does it have any rectangles of size 2? (e.g. 1x2 or 2x1) Yes
  - Every rectangle can be represented by an expression
  - What is the expression in this case?

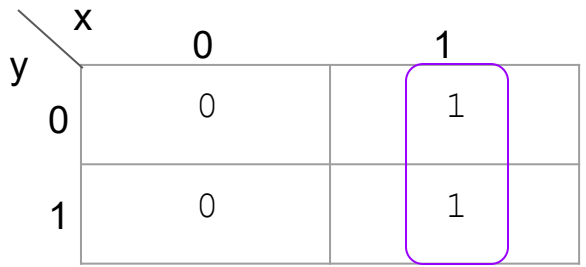
x	***y***	***xy + xy'***
0	0	0
0	1	0
1	0	1
1	1	1



Karnaugh Maps (K-Maps)

- Next, we look for rectangles that are made entirely of 1s
  - Both of these 1s represent different values for `y`
  - … but they also both represent the same value of `x`
  - In fact, the rectangle covers every situation where `x = 1`
  - The expression, therefore, is `x` (`x=0` would be `x'`)

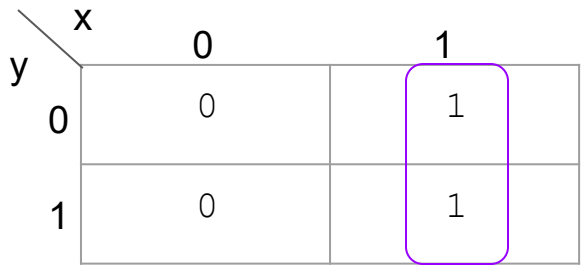
x	***y***	***xy + xy'***
0	0	0
0	1	0
1	0	1
1	1	1



Karnaugh Maps (K-Maps)

- Next, we look for rectangles that are made entirely of 1s
  - Since the rectangle we just found covers all of the 1s in the K-map, we are done
  - The simplified expression is, therefore, `f(x,y) = x`
- *Note: Rectangles can, and often do, overlap*

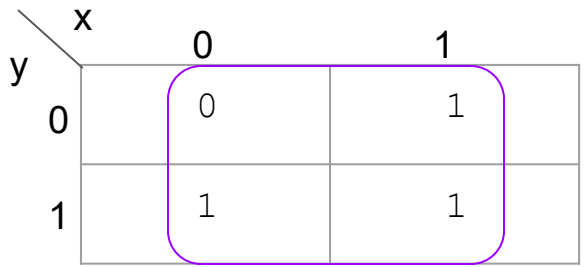
x	***y***	***xy + xy'***
0	0	0
0	1	0
1	0	1
1	1	1



Karnaugh Maps (K-Maps)

- Another example: `g(a,b) = xy + x'y + xy'`
  - Any rectangles of size 4?

x	***y***	***xy + x'y + xy'***
0	0	0
0	1	1
1	0	1
1	1	1



Karnaugh Maps (K-Maps)

- Another example: `g(a,b) = xy + x'y + xy'`
  - Any rectangles of size 4? No
  - Any rectangles of size 2?

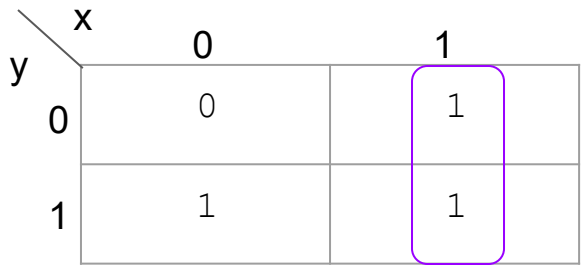
x	***y***	***xy + x'y + xy'***
0	0	0
0	1	1
1	0	1
1	1	1

y \ x	0	1
0	0	1
1	1	1

Karnaugh Maps (K-Maps)

- Another example: `g(a,b) = xy + x'y + xy'`
  - Any rectangles of size 4? No
  - Any rectangles of size 2? Yes
    - What is the expression for this rectangle?

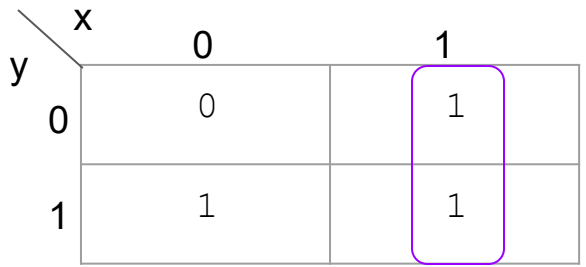
x	***y***	***xy + x'y + xy'***
0	0	0
0	1	1
1	0	1
1	1	1



Karnaugh Maps (K-Maps)

- Another example: `g(a,b) = xy + x'y + xy'`
  - Any rectangles of size 4? No
  - Any rectangles of size 2? Yes
    - What is the expression for this rectangle? **x**

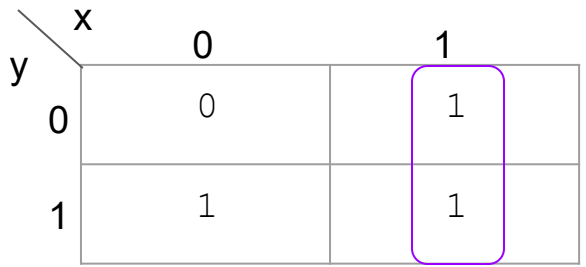
x	***y***	***xy + x'y + xy'***
0	0	0
0	1	1
1	0	1
1	1	1



Karnaugh Maps (K-Maps)

- Another example: `g(a,b) = xy + x'y + xy'`
  - Any rectangles of size 4? No
  - Any more rectangles of size 2?

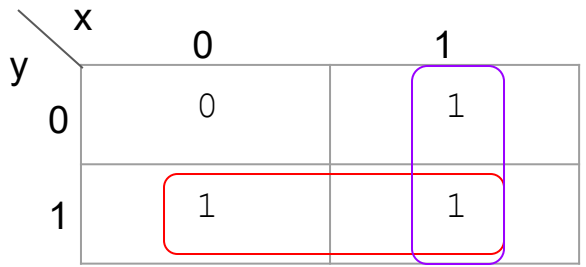
x	***y***	***xy + x'y + xy'***
0	0	0
0	1	1
1	0	1
1	1	1



Karnaugh Maps (K-Maps)

- Another example: `g(a,b) = xy + x'y + xy'`
  - Any rectangles of size 4? No
  - Any more rectangles of size 2? Yes
    - What is the expression for this rectangle? **y**

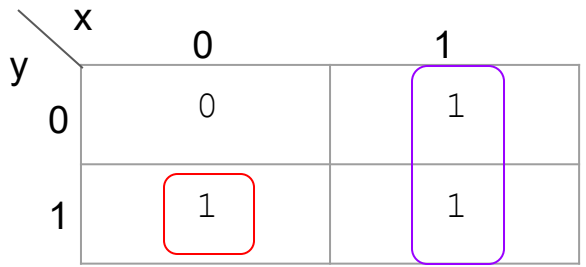
x	***y***	***xy + x'y + xy'***
0	0	0
0	1	1
1	0	1
1	1	1



Karnaugh Maps (K-Maps)

- Another example: `g(a,b) = xy + x'y + xy'`
  - Any rectangles of size 4? No
  - Any more rectangles of size 2? Yes
    - What if just added a 1x1 rectangle, instead?
    - This expression is **x'y** (this is not an improvement)

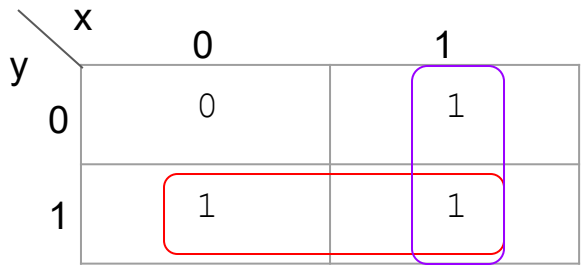
x	***y***	***xy + x'y + xy'***
0	0	0
0	1	1
1	0	1
1	1	1



Karnaugh Maps (K-Maps)

- Another example: `g(a,b) = xy + x'y + xy'`
  - Are we done?

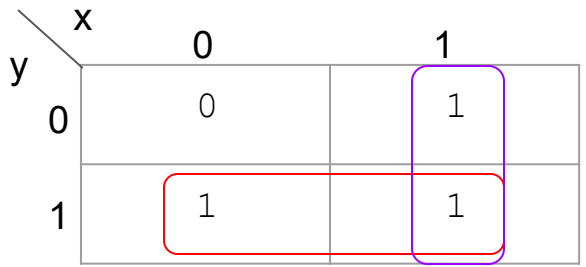
x	***y***	***xy + x'y + xy'***
0	0	0
0	1	1
1	0	1
1	1	1



Karnaugh Maps (K-Maps)

- Another example: `g(a,b) = xy + x'y + xy'`
  - Are we done? Yes, because all 1s have been covered by rectangles

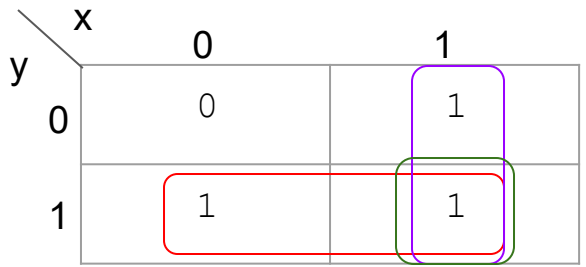
x	*y*	*xy + x'y + xy'*
0	0	0
0	1	1
1	0	1
1	1	1



Karnaugh Maps (K-Maps)

- Another example: `g(a,b) = xy + x'y + xy'`
  - Are we done? Yes, because all 1s have been covered by rectangles
  - Note: We could add another rectangle here, but that would be unnecessary since we have covered all 1s already

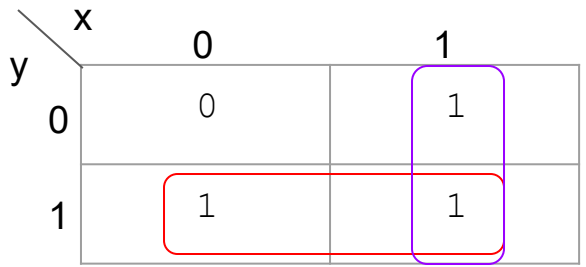
x	***y***	***xy + x'y + xy'***
0	0	0
0	1	1
1	0	1
1	1	1



Karnaugh Maps (K-Maps)

- Another example: `g(a,b) = xy + x'y + xy'`
  - Are we done? Yes, because all 1s have been covered by rectangles
  - Therefore, `g(a,b) = x + y`

x	***y***	***xy + x'y + xy'***
0	0	0
0	1	1
1	0	1
1	1	1



Karnaugh Maps - 3 Variable

CSCI 2050U - Computer Architecture

OntarioTech UNIVERSITY

Karnaugh Maps (K-Maps)

- K-maps for 3 variables have a notational quirk:

x	*y*	*z*	*xy + x'z + yz*
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

z \ xy	00	01	11	10
0	0	0	1	0
1	1	1	1	0

Karnaugh Maps (K-Maps)

- K-maps for 3 variables have a notational quirk:

x	*y*	*z*	*xy + x'z + yz*
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Note the unusual order, here:

z \ xy	00	01	11	10
0	0	0	1	0
1	1	1	1	0

Karnaugh Maps (K-Maps)

- Let's continue this example
  - Are there any size 8 rectangles? (remember, the rectangles must be of size 2^n)

z \ xy	00	01	11	10
0	0	0	1	0
1	1	1	1	0

Karnaugh Maps (K-Maps)

- Let's continue this example
  - Are there any size 8 rectangles? No
  - Are there any size 4 rectangles?

z \ xy	00	01	11	10
0	0	0	1	0
1	1	1	1	0

Karnaugh Maps (K-Maps)

- Let's continue this example
  - Are there any size 8 rectangles? No
  - Are there any size 4 rectangles? No

z \ xy	00	01	11	10
0	0	0	1	0
1	1	1	1	0

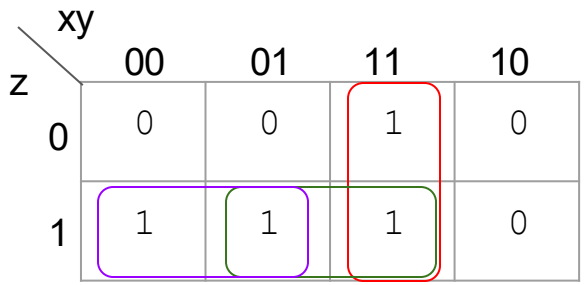
Karnaugh Maps (K-Maps)

- Let's continue this example
  - Are there any size 8 rectangles? No
  - Are there any size 4 rectangles? No
  - Are there any size 2 rectangles?

z \ xy	00	01	11	10
0	0	0	1	0
1	1	1	1	0

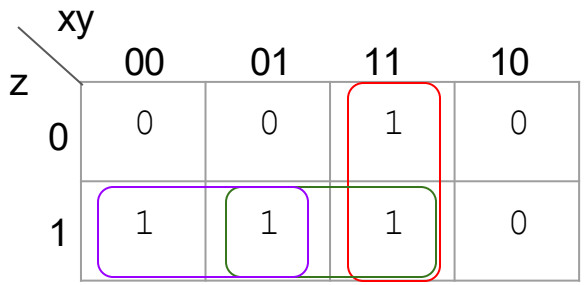
Karnaugh Maps (K-Maps)

- Let's continue this example
  - Are there any size 8 rectangles? No
  - Are there any size 4 rectangles? No
  - Are there any size 2 rectangles? Yes
    - **xy**
    - **x′z**
    - **yz**



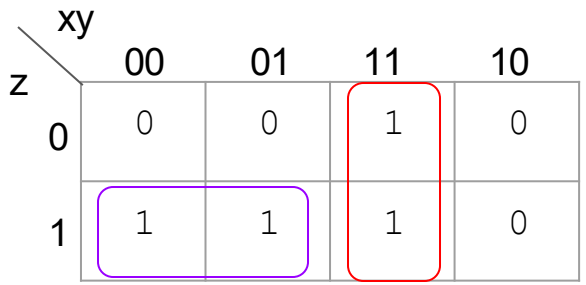
Karnaugh Maps (K-Maps)

- Let's continue this example
  - Are there any size 8 rectangles? No
  - Are there any size 4 rectangles? No
  - Are there any size 2 rectangles? Yes
    - **xy**
    - **x′z**
    - **yz**
    - Right?



Karnaugh Maps (K-Maps)

- Let's continue this example
  - Are there any size 8 rectangles? No
  - Are there any size 4 rectangles? No
  - Are there any size 2 rectangles? Yes
    - **xy**
    - **x′z**
    - **yz**
    - Right? No! The term **yz** is unnecessary
    - `h(x,y,z) = xy + x′z`



Karnaugh Maps - 4 Variable

CSCI 2050U - Computer Architecture

OntarioTech UNIVERSITY

Karnaugh Maps (K-Maps)

- 4-variable K-maps look like this:
  - `f(a,b,c,d) = abc'd' + abc'd + abcd + abcd' + a'bc'd + a'bcd + a'b'c'd + ab'c'd`
- *Note: We will not do K-maps for any expressions beyond 4 variables*

cd \ ab	00	01	11	10
00	0	0	1	0
01	1	1	1	1
11	0	1	1	0
10	0	0	1	0

Karnaugh Maps (K-Maps)

- Truth tables can sometimes have *don't care* conditions
- When this happens, we can treat it as either a 0 or a 1
  - Whichever is more convenient for finding our rectangles
- In this example, the ? should be a 1 since it enables this 1x4 rectangle

cd \ ab	00	01	11	10
00	0	0	1	0
01	1	?	1	1
11	0	1	1	0
10	0	0	1	0

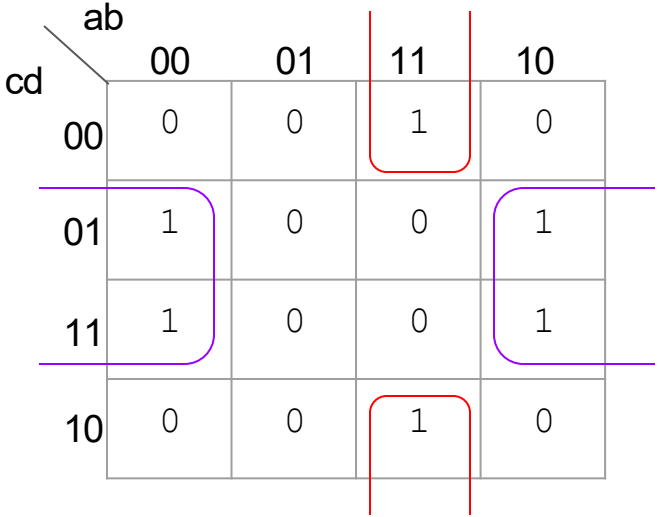
Karnaugh Maps (K-Maps)

- K-maps have one other confusing aspect for students
  - The positions in this table are adjacent because the values next to each other (horizontally or vertically) differ by exactly one bit
  - Notice that the left and right columns for 3 or 4-variable k-maps (and the top and bottom rows) also differ by exactly one bit

cd \ ab	00	01	11	10
00	0	0	1	0
01	1	0	0	1
11	1	0	0	1
10	0	0	1	0

Karnaugh Maps (K-Maps)

- K-maps have one other confusing aspect for students
  - As a consequence, these are valid rectangles:



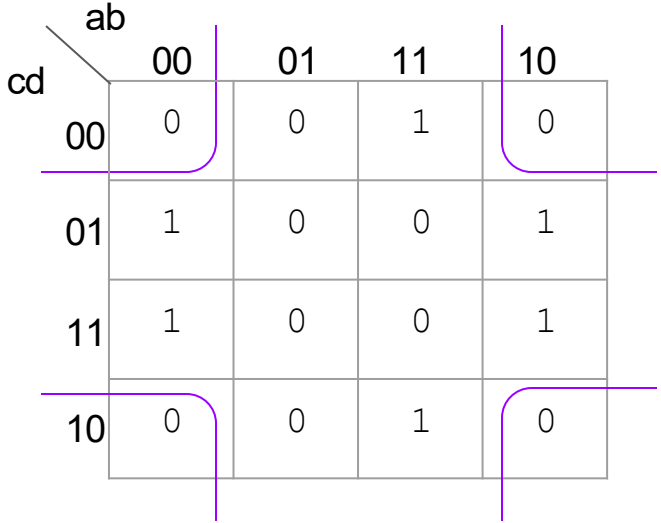
Karnaugh Maps (K-Maps)

- K-maps have one other confusing aspect for students
  - As a consequence, these are valid rectangles:

cd \ ab	00	01	11	10
00	1	1	0	1
01	0	0	0	0
11	1	0	0	1
10	1	1	1	1

Karnaugh Maps (K-Maps)

- K-maps have one other confusing aspect for students
  - As a consequence, these are valid rectangles
  - … and so is this:



Wrap-Up

- Karnaugh maps
  - 2-variable
  - 3-variable
  - 4-variable

What is next?

- Finite state machines
- Oscillators (clock)
- Latches