Optimizing Our Design! (II)

Prof. Usagi

Recap: Boolean Laws/Theorems

	OR	AND
Associative laws	$(a+b)+c=a+(b+c)$	$(a\cdot b)\cdot c=a\cdot(b\cdot c)$
Commutative laws	$a+b=b+a$	$a\cdot b=b\cdot a$
Distributive laws	$a+(b\cdot c)=(a+b)\cdot(a+c)$	$a\cdot(b+c)=a\cdot b+a\cdot c$
Identity laws	$a+0=a$	$a\cdot 1=a$
Complement laws	$a+a'=1$	$a\cdot a'=0$
DeMorgan's Theorem	$(a+b)' = a'b'$	$a'b' = (a+b)'$
Covering Theorem	$a(a+b) = a+ab = a$	$ab + ab' = (a+b)(a+b') = a$
Consensus Theorem	$ab+ac+b'c = ab+b'c$	$(a+b)(a+c)(b'+c) = (a+b)(b'+c)$
Uniting Theorem	$a(b+b') = a$	$(a+b)\cdot(a+b')=a$
Shannon's Expansion	$f(a,b,c) = a'b' + bc + ab'c$ $f(a,b,c) = a\,f(1,b,c) + a'\,f(0,b,c)$	

Recap: How many "OR"s?

- For the truth table shown on the right, what's the minimum number of "OR" gates we need?

  A. 1
  B. 2
  C. 3
  D. 4
  E. 5

$F(A, B, C) =$

$A'B'C' + A'B'C + A'BC' + A'BC + AB'C' + ABC'$

$= A'B'(C'+C) + A'B(C'+C) + AC'(B'+B)$

$= A'B' + A'B + AC'$

$= A' + AC' = A'(1+C')+AC'$ **Distributive Laws**

$= A' + A'C' + AC'$

$= A' + (A'+A)C'$

$= A' + C'$

How can I know this!!!

Input			Output
A	B	C	
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Recap: Karnaugh maps

- Alternative to truth-tables to help visualize adjacencies
- Guide to applying the uniting theorem
- Steps
  - Create a 2-D truth table with input variables on each dimension, and adjacent column(j)/row(i) only change one bit in the variable.
  - Fill each (i,j) with the corresponding result in the truth table
  - Identify ON-set (all 1s) with size of power of 2 (i.e., 1, 2, 4, 8, …) and "unite" them terms together (i.e. finding the "common literals" in their minterms)
  - Find the "minimum cover" that covers all 1s in the graph
  - Sum with the united product terms of all minimum cover ON-sets

Recap: 2-variable K-map example

Input		Output
A	B	
0	0	**1**
0	1	**1**
1	0	**1**
1	1	**0**

		A′	A
	A \ B	0	1
B′	0	1	1
B	1	1	0

B′

A′

$$F(A, B) = A' + B'$$

Recap: 3-variable K-map?

- Reduce to 2-variable K-map — 1 dimension will represent two variables
- Adjacent points should differ by only 1 bit
  - So we only change one variable in the neighboring column
  - 00, 01, 11, 10 — such numbering scheme is so-called **Gray–code**

Input			Output
A	B	C	
0	0	0	**1**
0	0	1	**1**
0	1	0	**1**
0	1	1	**1**
1	0	0	**1**
1	0	1	**0**
1	1	0	**1**
1	1	1	**0**

		A′B′	**A′B**	**AB**	**AB′**
	C \ (A, B)	0,0	0,1	1,1	1,0
C′	0	1	1	1	1
C	1	1	1	0	0

C′

A′

F(A, B, C) = A′ + C′

Poll close in 1:30

Valid K-Maps

- How many of the followings are "valid" K-Maps?

(1)

	0,0	0,1	1,1	1,0
0	0	1	0	1
1	1	0	1	0

(2)

	0,1	1,1	1,0	0,0
0	1	0	1	0
1	0	1	0	1

(3)

	1,1	1,0	0,1	0,0
0	0	1	1	0
1	1	0	0	1

(4)

	0,0	0,1	1,0	1,1
0	0	1	1	0
1	1	0	0	1

A. 0
B. 1
C. 2
D. 3
E. 4

Valid K-Maps

- How many of the followings are "valid" K-Maps?

(1) ✓

	0,0	0,1	1,1	1,0
0	0	1	0	1
1	1	0	1	0

(2) ✓

	0,1	1,1	1,0	0,0
0	1	0	1	0
1	0	1	0	1

(3)

	1,1	1,0	0,1	0,0
0	0	1	1	0
1	1	0	0	1

(4)

	0,0	0,1	1,0	1,1
0	0	1	1	0
1	1	0	0	1

A. 0
B. 1
C. 2
D. 3
E. 4

Recap: Minimum SOP for a full adder

- Minimum number of SOP terms to cover the "Out" function for a one-bit full adder?
  - A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 5

		A'B'	A'B	AB	AB'
	Out(A, B)	0,0	0,1	1,1	1,0
Cin'	0	0	1	0	1
Cin	1	1	0	1	0

Input			Output	
A	B	Cin	Out	Cout
0	0	0	**0**	**0**
0	1	0	**1**	**0**
1	0	0	**1**	**0**
1	1	0	**0**	**1**
0	0	1	**1**	**0**
0	1	1	**0**	**1**
1	0	1	**0**	**1**
1	1	1	**1**	**1**

Outline

- More on Karnaugh maps
- Design examples

Poll close in 1:30

Minimum SOP for a full adder

- Minimum number of SOP terms to cover the “Cout” function for a one-bit full adder?
  A. 1
  B. 2
  C. 3
  D. 4
  E. 5

Input			Output	
A	B	Cin	Out	Cout
0	0	0	**0**	**0**
0	1	0	**1**	**0**
1	0	0	**1**	**0**
1	1	0	**0**	**1**
0	0	1	**1**	**0**
0	1	1	**0**	**1**
1	0	1	**0**	**1**
1	1	1	**1**	**1**

Minimum SOP for a full adder

- Minimum number of SOP terms to cover the "Cout" function for a one-bit full adder?

  A. 1
  B. 2
  C. 3
  D. 4
  E. 5

	Input			Output	
	A	**B**	**Cin**	**Out**	**Cout**
	0	0	0	**0**	**0**
	0	1	0	**1**	**0**
	1	0	0	**1**	**0**
	1	1	0	**0**	**1**
	0	0	1	**1**	**0**
	0	1	1	**0**	**1**
	1	0	1	**0**	**1**
	1	1	1	**1**	**1**

		A'B'	**A'B**	**AB**	**AB'**
	Out(A, B)	**0,0**	**0,1**	**1,1**	**1,0**
Cin'	**0**	0	0	1	0
Cin	**1**	0	1	1	1

BCin **AB** **ACin**

Poll close in 1:30

Minimum SOP terms

- What's the minimum sum-of-products expression of the given truth table?
  A. A′B′C′ + A′BC′+ A′BC + AB′C′
  B. A′B′C + AB + AC
  C. AB′C′ + B′C′
  D. A′B + B′C′
  E. A′C′ + A′B + AB′C′

Input			Output
A	B	C	
0	0	0	**1**
0	0	1	**0**
0	1	0	**1**
0	1	1	**1**
1	0	0	**1**
1	0	1	**0**
1	1	0	**0**
1	1	1	**0**

Minimum SOP terms

- What's the minimum sum-of-products expression of the given truth table?
  A. A′B′C′ + A′BC′+ A′BC + AB′C′
  B. A′B′C + AB + AC
  C. AB′C′ + B′C′
  D. A′B + B′C′
  E. A′C′ + A′B + AB′C′

Input			Output
A	B	C	
0	0	0	**1**
0	0	1	**0**
0	1	0	**1**
0	1	1	**1**
1	0	0	**1**
1	0	1	**0**
1	1	0	**0**
1	1	1	**0**

		A′B′	A′B	AB	AB′
	C (A, B)	0,0	0,1	1,1	1,0
C′	0	1	1	0	1
C	1	0	1	0	0

B′C′

A′B

4-variable K-map

- Reduce to 2-variable K-map — both dimensions will represent two variables
- Adjacent points should differ by only 1 bit
  - So we only change one variable in the neighboring column
  - Use Gray-coding — 00, 01, 11, 10

		A′B′	A′B	AB	AB′
		00	01	11	10
C′D′	00	1	0	0	0
C′D	01	1	0	0	0
CD	11	0	0	0	0
CD′	10	1	0	0	1

A′B′C′

B′CD′

F(A, B, C) = A′B′C′+B′CD′

Poll close in 1:30

4-variable K-map

- What's the minimum sum-of-products expression of the given K-map?
  A. B′C′ + A′B′
  B. B′C′D′ + A′B′ + B′C′D′
  C. A′B′CD′ + B′C′
  D. AB′ + A′B′ + A′B′D′
  E. B′C′ + A′CD′

		A′B′	A′B	AB	AB′
		00	01	11	10
C′D′	00	1	0	0	1
C′D	01	1	0	0	1
CD	11	0	0	0	0
CD′	10	1	1	0	0

4-variable K-map

- What′s the minimum sum-of-products expression of the given K-map?
  A. B′C′ + A′B′
  B. B′C′D′ + A′B′ + B′C′D′
  C. A′B′CD′ + B′C′
  D. AB′ + A′B′ + A′B′D′
  E. B′C′ + A′CD′

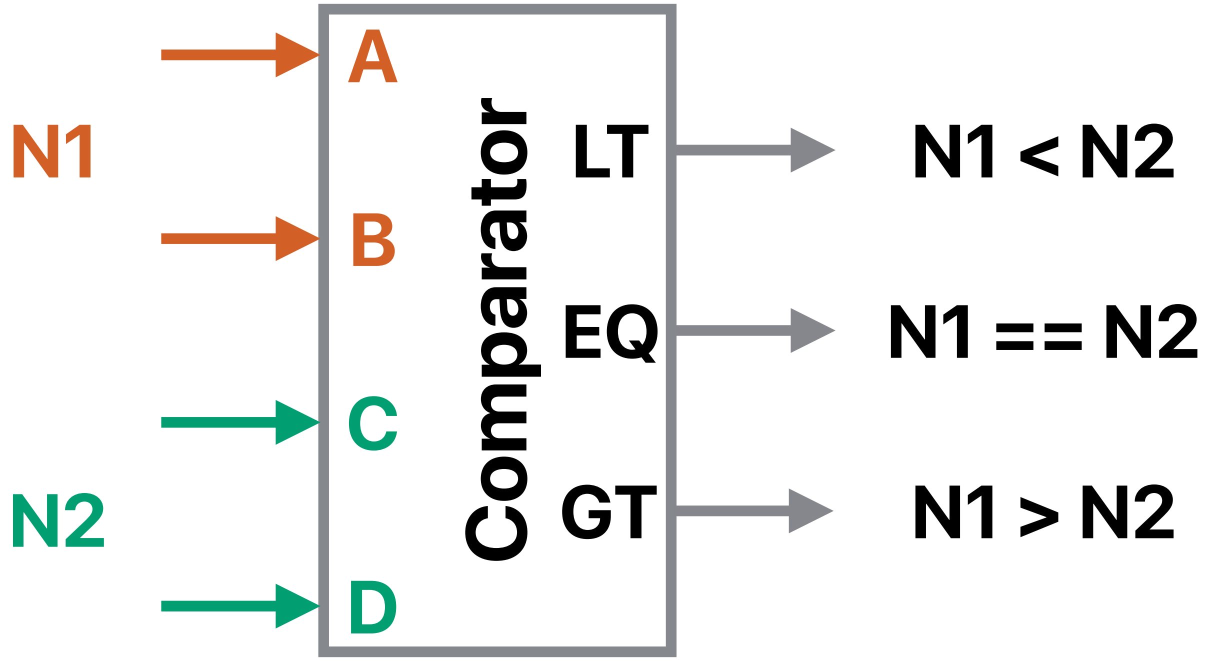
		A′B′	A′B	AB	AB′
		00	01	11	10
C′D′	00	1	0	0	1
C′D	01	1	0	0	1
CD	11	0	0	0	0
CD′	10	1	1	0	0

B′C′

A′CD′

Design Example: 2bit comparator

Two-bit comparator



Input				Output		
A	B	C	D	LT	EQ	GT
0	0	0	0	0	1	0
0	0	0	1	1	0	0
0	0	1	0	1	0	0
0	0	1	1	1	0	0
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	1	0	0
0	1	1	1	1	0	0
1	0	0	0	0	0	1
1	0	0	1	0	0	1
1	0	1	0	0	1	0
1	0	1	1	1	0	0
1	1	0	0	0	0	1
1	1	0	1	0	0	1
1	1	1	0	0	0	1
1	1	1	1	0	1	0

We'll need a 4-variable Karnaugh map for each of the 3 output functions

Poll close in 1:30

LT?

- What’s the minimum SOP presentation of LT?
  A. A′B′D′ + AC′ + BCD
  B. A'B'D + A'C + B′CD
  C. A'B'C'D' + A'BC'D + ABCD + AB′CD′
  D. ABCD + AB′CD′ + A′B′C′D′ + A′BC′D
  E. BC'D' + AC' + ABD'

Input				Output		
A	B	C	D	LT	EQ	GT
0	0	0	0	0	1	0
0	0	0	1	1	0	0
0	0	1	0	1	0	0
0	0	1	1	1	0	0
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	1	0	0
0	1	1	1	1	0	0
1	0	0	0	0	0	1
1	0	0	1	0	0	1
1	0	1	0	0	1	0
1	0	1	1	1	0	0
1	1	0	0	0	0	1
1	1	0	1	0	0	1
1	1	1	0	0	0	1
1	1	1	1	0	1	0

LT?

- What's the minimum SOP presentation of LT?
  A. A′B′D′ + AC′ + BCD
  B. A'B'D + A'C + B′CD
  C. A'B'C'D' + A'BC'D + ABCD + AB′CD′
  D. ABCD + AB′CD′ + A′B′C′D′ + A′BC′D
  E. BC'D' + AC' + ABD'

		A′B′	A′B	AB	AB′
		00	01	11	10
C′D′	00	0	0	0	0
C′D	01	1	0	0	0
CD	11	1	1	0	1
CD′	10	1	1	0	0

A′B′D
A′C
B′CD

Input				Output		
A	B	C	D	LT	EQ	GT
0	0	0	0	0	1	0
0	0	0	1	1	0	0
0	0	1	0	1	0	0
0	0	1	1	1	0	0
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	1	0	0
0	1	1	1	1	0	0
1	0	0	0	0	0	1
1	0	0	1	0	0	1
1	0	1	0	0	1	0
1	0	1	1	1	0	0
1	1	0	0	0	0	1
1	1	0	1	0	0	1
1	1	1	0	0	0	1
1	1	1	1	0	1	0

Poll close in 1:30

GT?

- What's the minimum SOP presentation of GT?
  A. A′B′D′ + AC′ + BCD
  B. A'B'D + A'C + B′CD
  C. A'B'C'D' + A'BC'D + ABCD + AB′CD′
  D. ABCD + AB′CD′ + A′B′C′D′ + A′BC′D
  E. BC'D' + AC' + ABD'

Input				Output		
A	B	C	D	LT	EQ	GT
0	0	0	0	0	1	0
0	0	0	1	1	0	0
0	0	1	0	1	0	0
0	0	1	1	1	0	0
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	1	0	0
0	1	1	1	1	0	0
1	0	0	0	0	0	1
1	0	0	1	0	0	1
1	0	1	0	0	1	0
1	0	1	1	1	0	0
1	1	0	0	0	0	1
1	1	0	1	0	0	1
1	1	1	0	0	0	1
1	1	1	1	0	1	0

GT?

- What’s the minimum SOP presentation of GT?
  A. A′B′D′ + AC′ + BCD
  B. A'B'D + A'C + B′CD
  C. A'B'C'D' + A'BC'D + ABCD + AB′CD′
  D. ABCD + AB′CD′ + A′B′C′D′ + A′BC′D
  E. BC'D' + AC' + ABD'

		A′B′	A′B	AB	AB′
		00	01	11	10
C′D′	00	0	1	1	1
C′D	01	0	0	1	1
CD	11	0	0	0	0
CD′	10	0	0	1	0

BC'D'

AC′

ABD′

Input				Output		
A	B	C	D	LT	EQ	GT
0	0	0	0	0	1	0
0	0	0	1	1	0	0
0	0	1	0	1	0	0
0	0	1	1	1	0	0
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	1	0	0
0	1	1	1	1	0	0
1	0	0	0	0	0	1
1	0	0	1	0	0	1
1	0	1	0	0	1	0
1	0	1	1	1	0	0
1	1	0	0	0	0	1
1	1	0	1	0	0	1
1	1	1	0	0	0	1
1	1	1	1	0	1	0

Poll close in 1:30

EQ?

- What's the minimum SOP presentation of EQ?
  A. A'B'D' + AC' + BCD
  B. A'B'D + A'C + B'CD
  C. A'B'C'D' + A'BC'D + ABCD + AB'CD'
  D. A'BCD' + A'B'CD' + A'B'C'D' + A'BC'D
  E. BC'D' + AC' + ABD'

Input				Output		
A	B	C	D	LT	EQ	GT
0	0	0	0	0	1	0
0	0	0	1	1	0	0
0	0	1	0	1	0	0
0	0	1	1	1	0	0
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	1	0	0
0	1	1	1	1	0	0
1	0	0	0	0	0	1
1	0	0	1	0	0	1
1	0	1	0	0	1	0
1	0	1	1	1	0	0
1	1	0	0	0	0	1
1	1	0	1	0	0	1
1	1	1	0	0	0	1
1	1	1	1	0	1	0

EQ?

- What's the minimum SOP presentation of EQ?
  A. A'B'D' + AC' + BCD
  B. A'B'D + A'C + B'CD
  C. A'B'C'D' + A'BC'D + ABCD + AB'CD'
  D. A'BCD' + A'B'CD' + A'B'C'D' + A'BC'D
  E. BC'D' + AC' + ABD'

		A'B'	A'B	AB	AB'
		00	01	11	10
C'D'	00	1	0	0	0
C'D	01	0	1	0	0
CD	11	0	0	1	0
CD'	10	0	0	0	1

Input				Output		
A	B	C	D	LT	EQ	GT
0	0	0	0	0	1	0
0	0	0	1	1	0	0
0	0	1	0	1	0	0
0	0	1	1	1	0	0
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	1	0	0
0	1	1	1	1	0	0
1	0	0	0	0	0	1
1	0	0	1	0	0	1
1	0	1	0	0	1	0
1	0	1	1	1	0	0
1	1	0	0	0	0	1
1	1	0	1	0	0	1
1	1	1	0	0	0	1
1	1	1	1	0	1	0

Don't cares!

Incompletely Specified Functions

- Situations where the output of a function can be either 0 or 1 for a particular combination of inputs
- This is specified by a don't care in the truth table
- This happens when
  - The input does not occur. e.g. Decimal numbers 0… 9 use 4 bits, so (1,1,1,1) does not occur.
  - The input may happen but we don't care about the output. E.g. The output driving a seven segment display – we don't care about illegal inputs (greater than 9)

Don't care

B \ A	0	1
0	0	0
1	1	X

K-Map with "Don't Care"s

You can treat "X" as either 0 or 1
— depending on which is more advantageous

	(A, B) / C	A′B′ 0,0	A′B 0,1	AB 1,1	AB′ 1,0
C′	0	1	0 X 1	1	1
C	1	1	1	0	0

C′ AC′

A′B′ A′C A′C

If we treat the "X" as 0?

$F(A,B,C)=A'B'+A'C+AC'$

If we treat the "X" as 1?

$F(A,B,C) = C' + A'C$

Poll close in 1:30

4-input K-Maps with Don't Cares

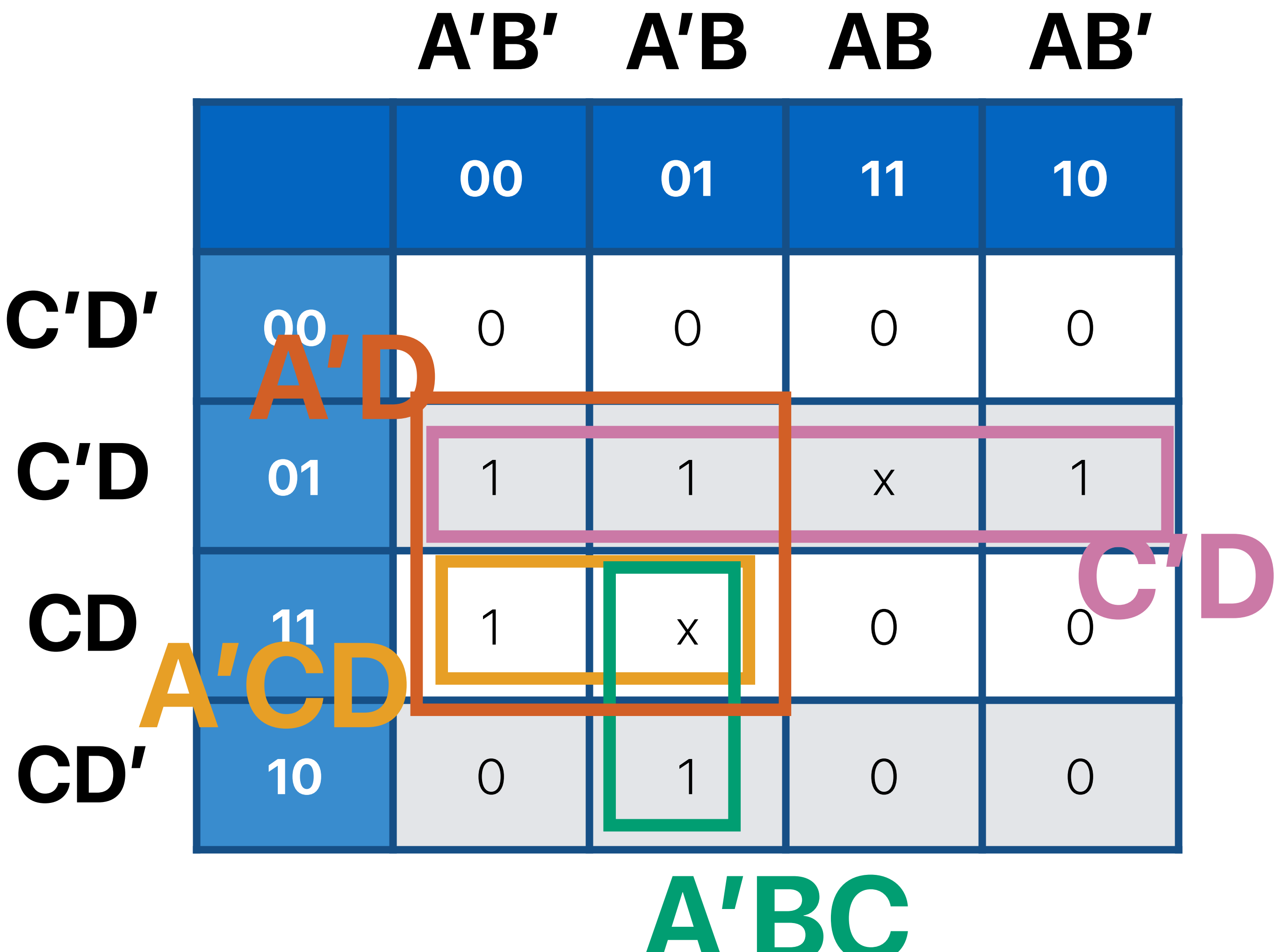
- How many of the following could be a valid for the given K-map?
  1. C′D + A′CD + A′BC
  2. C′D + A′D + A′BC
  3. C′D + A′CD
  4. A′D + B′C′D + A′BCD′

  A. 0
  B. 1
  C. 2
  D. 3
  E. 4

		A′B′	A′B	AB	AB′
		00	01	11	10
C′D′	00	0	0	0	0
C′D	01	1	1	x	1
CD	11	1	x	0	0
CD′	10	0	1	0	0

4-input K-Maps with Don't Cares

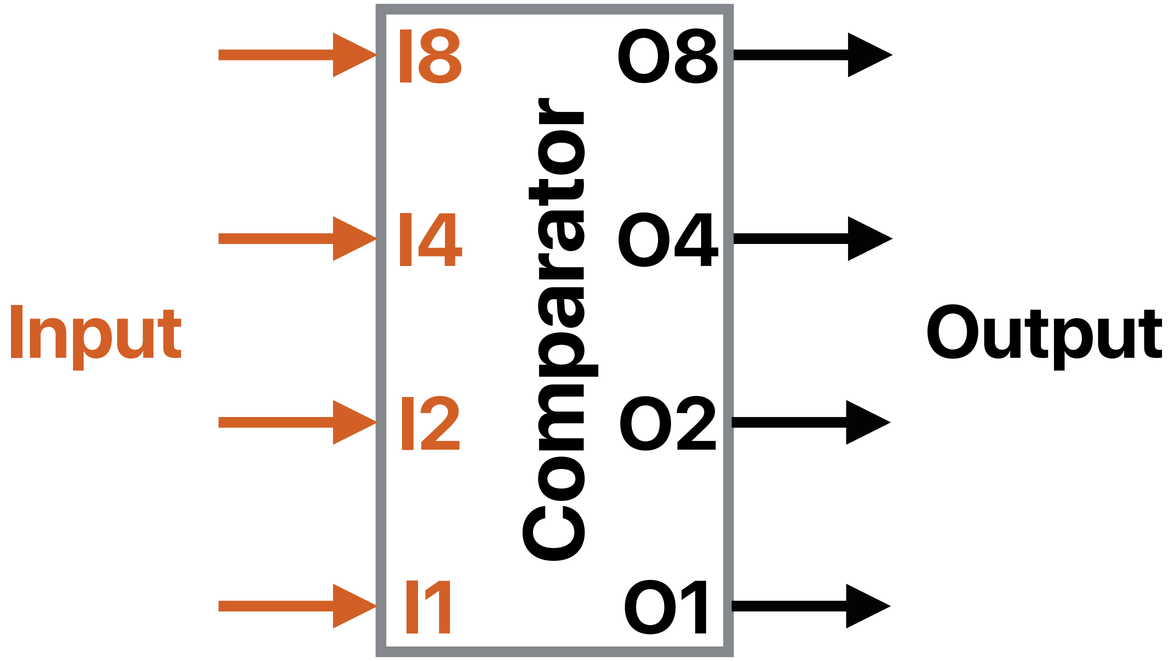
- How many of the following could be a valid for the given K-map?
  ① C′D + A′CD + A′BC
  ② C′D + A′D + A′BC
  ③ C′D + A′CD
  ~~④~~ A′D + B′C′D + A′BCD′
  A. 0
  B. 1
  C. 2
  D. 3
  E. 4



Design examples — BCD + 1

BCD+1 — Binary coded decimal + 1

- 0x0 — 1
- 0x1 — 2
- 0x2 — 3
- 0x3 — 4
- 0x4 — 5
- 0x5 — 6
- 0x6 — 7
- 0x7 — 8
- 0x8 — 9
- 0x9 — 0
- 0xA — 0xF — Don't care



Input				Output			
I8	I4	I2	I1	O8	O4	O2	O1
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	1	1
0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	1
1	0	0	1	0	0	0	0
1	0	1	0	X	X	X	X
1	0	1	1	X	X	X	X
1	1	0	0	X	X	X	X
1	1	0	1	X	X	X	X
1	1	1	0	X	X	X	X
1	1	1	1	X	X	X	X

K-maps

Input				Output			
I8	I4	I2	I1	O8	O4	O2	O1
0	0	0	0	**0**	**0**	**0**	**1**
0	0	0	1	**0**	**0**	**1**	**0**
0	0	1	0	**0**	**0**	**1**	**1**
0	0	1	1	**0**	**1**	**0**	**0**
0	1	0	0	**0**	**1**	**0**	**1**
0	1	0	1	**0**	**1**	**1**	**0**
0	1	1	0	**0**	**1**	**1**	**1**
0	1	1	1	**1**	**0**	**0**	**0**
1	0	0	0	**1**	**0**	**0**	**1**
1	0	0	1	**0**	**0**	**0**	**0**
1	0	1	0	**X**	**X**	**X**	**X**
1	0	1	1	**X**	**X**	**X**	**X**
1	1	0	0	**X**	**X**	**X**	**X**
1	1	0	1	**X**	**X**	**X**	**X**
1	1	1	0	**X**	**X**	**X**	**X**
1	1	1	1	**X**	**X**	**X**	**X**

O8

		I8'I4'	I8'I4	I8I4	I8I4'
		00	01	11	10
I2'I1'	00	0	0	X	1
I2'I1	01	0	0	X	0
I2I1	11	0	1	X	X
I2I1'	10	0	0	X	X

O4

		I8'I4'	I8'I4	I8I4	I8I4'
		00	01	11	10
I2'I1'	00	0	1	X	0
I2'I1	01	0	1	X	0
I2I1	11	1	0	X	X
I2I1'	10	0	1	X	X

O2

		I8'I4'	I8'I4	I8I4	I8I4'
		00	01	11	10
I2'I1'	00	0	0	X	0
I2'I1	01	1	1	X	0
I2I1	11	0	0	X	X
I2I1'	10	1	1	X	X

O1

		I8'I4'	I8'I4	I8I4	I8I4'
		00	01	11	10
I2'I1'	00	1	1	X	1
I2'I1	01	0	0	X	0
I2I1	11	0	0	X	X
I2I1'	10	1	1	X	X

5-variable KMaps?

	000	001	011	010	110	100	101	111
00	0	0	X	1	0	0	1	X
01	0	0	X	0	0	0	1	X
11	0	1	X	X	X	1	0	X
10	0	0	X	X	X	0	1	X

AB'CD'

6-variable KMap?

	000	001	011	010	110	100	101	111
000	0	0	X	1	0	0	1	X
001	0	0	X	0	0	0	1	X
011	0	1	X	X	X	1	0	X
010	0	0	X	X	X	0	1	X
110	1	1	X	X	1	1	X	X
100	0	0	X	0	1	1	X	1
101	1	1	X	0	0	0	X	0
111	0	0	X	X	0	0	X	X

AC′DF′

Announcement

- Assignment 1 due 4/14
  - Submit on zyBooks.com directly — all challenge questions up to 2.2
- Lab 2 due 4/16
  - Watch the video and read the instruction BEFORE your session
  - There are links on both course webpage and iLearn lab section
  - Submit through iLearn > Labs
- Reading quiz 4 due 4/21 **BEFORE** the lecture
  - Under iLearn > reading quizzes
- Assignment 2 due 4/23
  - Submit on zyBooks.com directly — all challenge questions 2.3-3.4
- Lab 3 due 4/30
  - Watch the video and read the instruction BEFORE your session
  - There are links on both course webpage and iLearn lab section
  - Submit through iLearn > Labs

Electrical
Computer Science
Engineering

120A

つづく