CSEE 6861 CAD of Digital Systems
Handout: Lecture #3
2/4/16

Prof. Steven M. Nowick
nowick@cs.columbia.edu

Department of Computer Science (and Elect. Eng.)
Columbia University
New York, NY, USA

ESPRESSO Algorithm:
The "EXPAND" Step, continued

"EXPAND" Step: Expansion Direction (REVIEW)

Basic idea:

- Once ***cube c*** in ***cover F*** is selected for expansion, expand it in "good directions"

3 Heuristic Expansion Steps (in sequence): to expand a *cube c*

1. Expand ***c*** to completely contain (i.e. "swallow up") as many other cubes of ***cover F*** as possible

   ______- delete these completely-contained cubes immediately!

2. Once #1 done -- continue to expand ***c*** to partially overlap as many other cubes of ***cover F*** as possible

   - the goal is to "induce redundancies", which then get deleted in the IRRED step

3. Once #2 done -- continue to expand ***c*** into a "maximal size" prime implicant

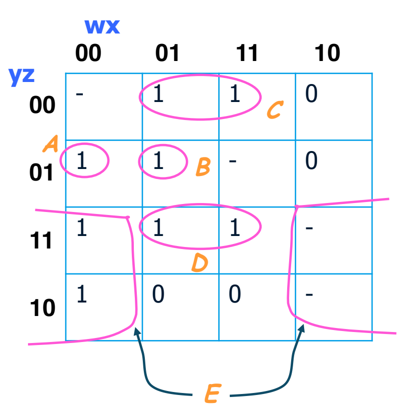
⇨ **New Focus: Step #2 and #3**

#3

"EXPAND" Step: Expansion Direction

Example #1: illustrates Step #1 + Step #2

Suppose cube A has been picked for expansion

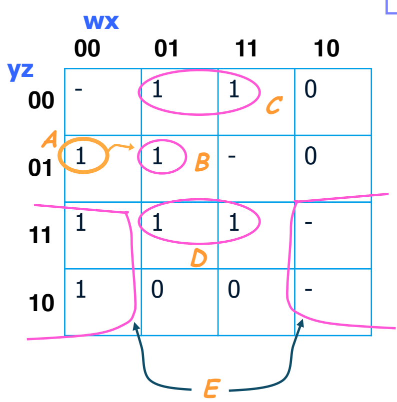


#4

“EXPAND” Step: Expansion Direction

Example #1 (cont.): illustrates Step #1 + Step #2

STEP #1: expand cube A to fully contain as many other cubes as possible

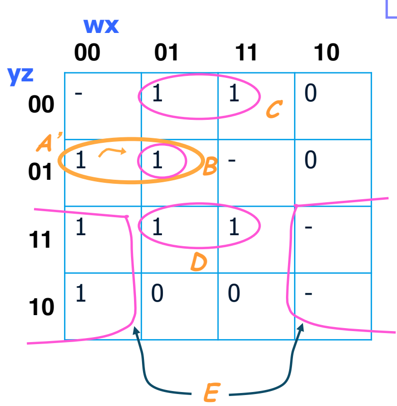


#5

“EXPAND” Step: Expansion Direction

Example #1 (cont.): illustrates Step #1 + Step #2

STEP #1: expand cube A to fully contain as many other cubes as possible



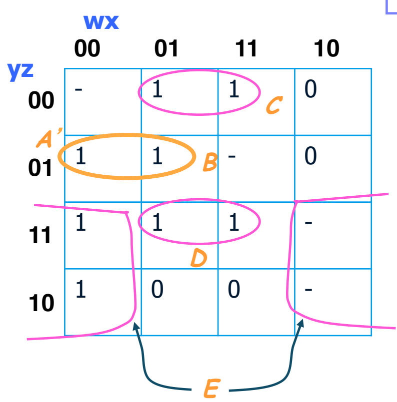
- expand cube A to fully contain cube B

#6

“EXPAND” Step: Expansion Direction

Example #1 (cont.): illustrates Step #1 + Step #2

STEP #1: expand cube A to fully contain as many other cubes as possible



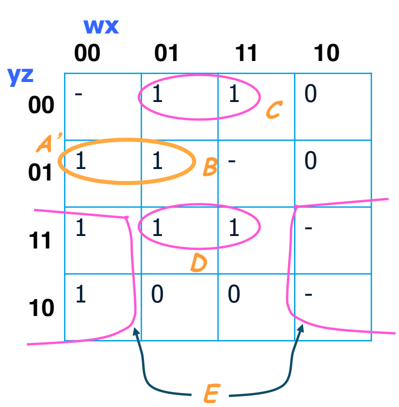
- expand cube A to fully contain cube B
- delete cube B

#7

“EXPAND” Step: Expansion Direction

Example #1 (cont.): illustrates Step #1 + Step #2

STEP #2: expand cube A', by 1 variable at a time, to overlap as many other cubes as possible

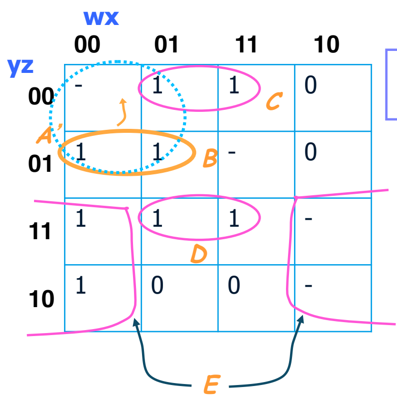


#8

"EXPAND" Step: Expansion Direction

Example #1 (cont.): illustrates Step #1 + Step #2

STEP #2: expand cube A', by 1 variable at a time, to overlap as many other cubes as possible



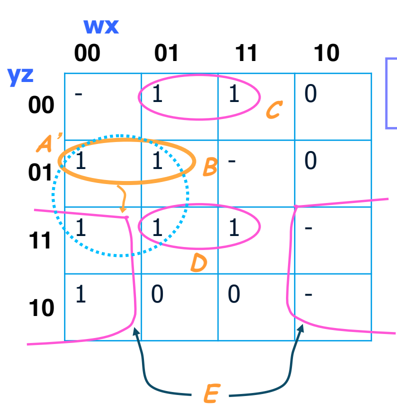
Option 1: expand cube A' in "z' dimension": overlaps 1 additional cube (C)

#9

"EXPAND" Step: Expansion Direction

Example #1 (cont.): illustrates Step #1 + Step #2

STEP #2: expand cube A', by 1 variable at a time, to overlap as many other cubes as possible



Option 2: expand cube A' in "y dimension": overlaps 2 additional cubes (D, E)

#10

"EXPAND" Step: Expansion Direction

Example #1 (cont.): illustrates Step #1 + Step #2

STEP #2: expand cube A', by 1 variable at a time, to overlap as many other cubes as possible

yz \ wx	00	01	11	10
00	-	1	1	0
01	1	1	-	0
11	1	1	1	-
10	1	0	0	-

Final choice = Option 2: expand cube A' in "y dimension"

#11

"EXPAND" Step: Expansion Direction

Example #2: illustrates Step #2 + Step #3

Suppose cube A has been picked for expansion

yz \ wx	00	01	11	10
00	-	0	-	0
01	-	-	1	-
11	-	1	1	1
10	0	1	1	1

#12

"EXPAND" Step: Expansion Direction

Example #2 (cont.): illustrates Step #2 + Step #3

STEP #2: expand cube A, by 1 variable at a time, to overlap as many other cubes as possible

yz \ wx	00	01	11	10
00	-	0	-	0
01	-	-	1	-
11	-	1	1	1
10	0	1	1	1

Final choice = expand cube A in "y dimension" ⇨ overlaps cubes B & C

#13

"EXPAND" Step: Expansion Direction

Example #2 (cont.): illustrates Step #2 + Step #3

STEP #2: expand cube A, by 1 variable at a time, to overlap as many other cubes as possible

yz \ wx	00	01	11	10
00	-	0	-	0
01	-	-	1	-
11	-	1	1	1
10	0	1	1	1

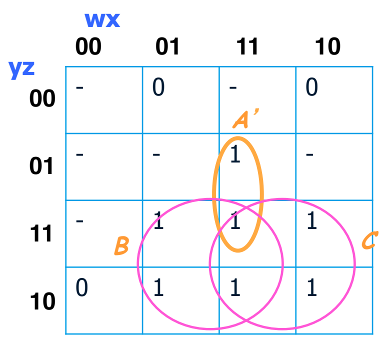
Final choice = expand cube A in "y dimension"

#14

"EXPAND" Step: Expansion Direction

Example #2 (cont.): illustrates Step #2 + Step #3

STEP #3: expand cube A' into a maximal size prime implicant

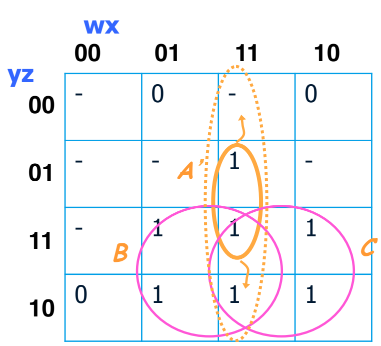


#15

"EXPAND" Step: Expansion Direction

Example #2 (cont.): illustrates Step #2 + Step #3

STEP #3: expand cube A' into a maximal size prime implicant



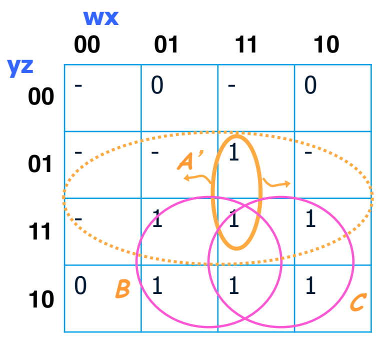
Option 1: expand cube A' in "z' dimension"
- A' becomes wx

#16

"EXPAND" Step: Expansion Direction

Example #2 (cont.): illustrates Step #2 + Step #3

STEP #3: expand cube A' into a maximal size prime implicant



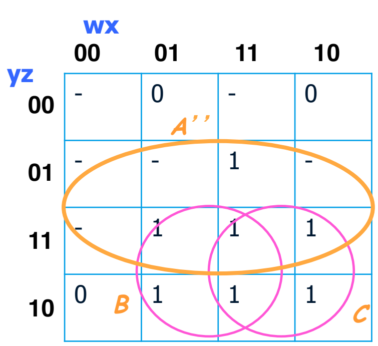
Option 2: expand cube A' in w'/x' dimensions
- A' becomes z

#17

"EXPAND" Step: Expansion Direction

Example #2 (cont.): illustrates Step #2 + Step #3

STEP #3: expand cube A' into a maximal size prime implicant



Final choice = Option 2
- 'larger' prime (A'' = z) has fewer literals

#18

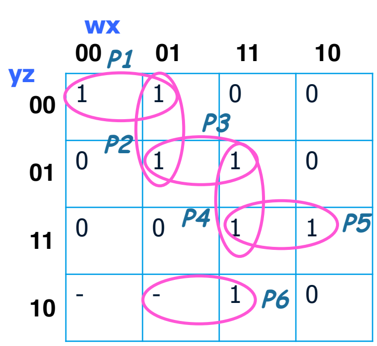
ESPRESSO Algorithm: The "IRREDUNDANT" Step

The "IRREDUNDANT" Step

Example Cover: *after "expand" = prime cover*

Given: ***cover F (after EXPAND)***

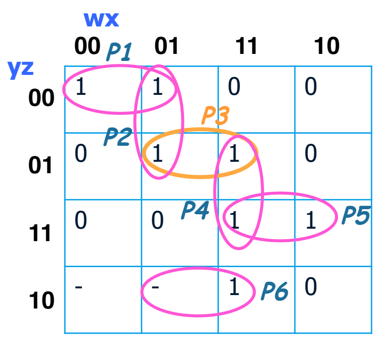
Goal: ***make F irredundant = delete max # of implicants while still maintaining a valid cover***



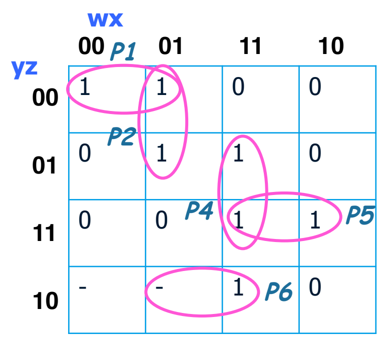
#20

The “IRREDUNDANT” Step

Approach #1 (SUBOPTIMAL!): *greedily remove one redundant cube at a time*



before:

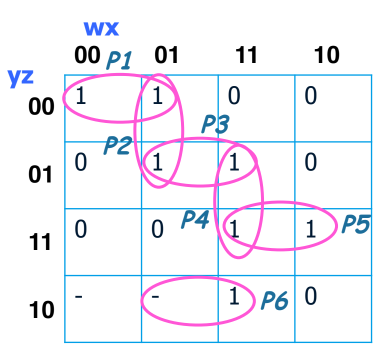


after deleting P3:
irredundant suboptimal cover

The “IRREDUNDANT” Step

Approach #2 (OPTIMAL!): *form & solve an exact optimization problem*

This is approach used in ESPRESSO…:



before:

Goal: set up a simplified PI table, and solve it exactly, to determine min # of cubes to keep (delete all others)

The "IRREDUNDANT" Step

Simplified PI Table:

prime implicants (columns); ON-set minterms (rows)

	P1	P2	P3	P4	P5	P6
0000	X					
0100	X	X				
0101		X	X			
1101			X	X		
1111				X	X	
1011					X	
1110						X

#23

The "IRREDUNDANT" Step

Approach: ***solve PI table exactly***

Simplified PI Table:

prime implicants (columns); ON-set minterms (rows)

	(P1)	P2	(P3)	P4	(P5)	(P6)
0000	X					
0100	X	X				
0101		X	X			
1101			X	X		
1111				X	X	
1011					X	
1110						X

Optimal Solution:
- *select {P1,P2,P5,P6}*
- *discard rest....*

#24

The "IRREDUNDANT" Step

An Issue:

- How can afford to compute an **exact solution** to **prime implicant table** *(like the expensive Quine-McCluskey method!)* in the inner loop of a fast heuristic algorithm *(espresso)?*

Justification:

- This is **not the same** as the QM method!

Key Observation: columns in this table are only the primes in *current cover,* -- *not all the primes of the function!*

Hence, there is no prime generation step, and often a (much) smaller table!

#25

The "IRREDUNDANT" Step

Approach #2 *(OPTIMAL)*: final solution

yz \ wx	00	01	11	10
00	1	1	0	0
01	0	1	1	0
11	0	0	1	1
10	-	-	1	0

P1, P2, P3, P4, P5, P6

before:

yz \ wx	00	01	11	10
00	1	1	0	0
01	0	1	1	0
11	0	0	1	1
10	-	-	1	0

P1, P3, P5, P6

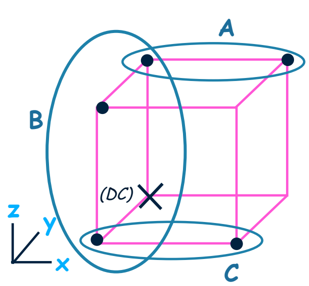
after: irredundant optimal cover

#26

ESPRESSO Algorithm:
The "REDUCE" Step

The "REDUCE" Step

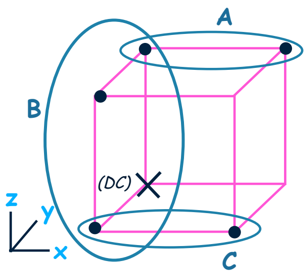
Example Cover: *after "expand" and "irredundant" = prime irredundant cover*



#28

The "REDUCE" Step

Example Cover: *after "expand" and "irredundant" = prime irredundant cover*



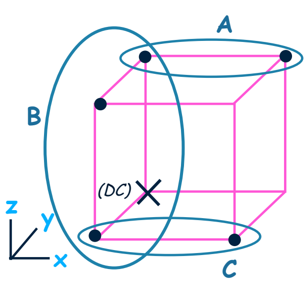
Given: cover F (after EXPAND/IRRED)

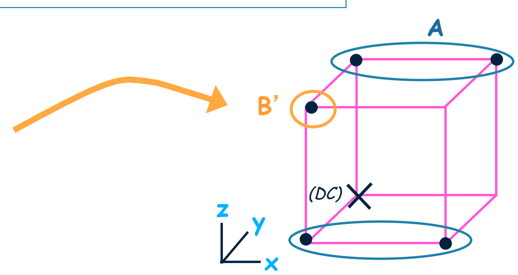
Goal: maximally reduce all cubes of cover F, in some order, while still maintaining a valid cover

#29

The "REDUCE" Step

Key Observation: reduce operation is ***order-dependent***



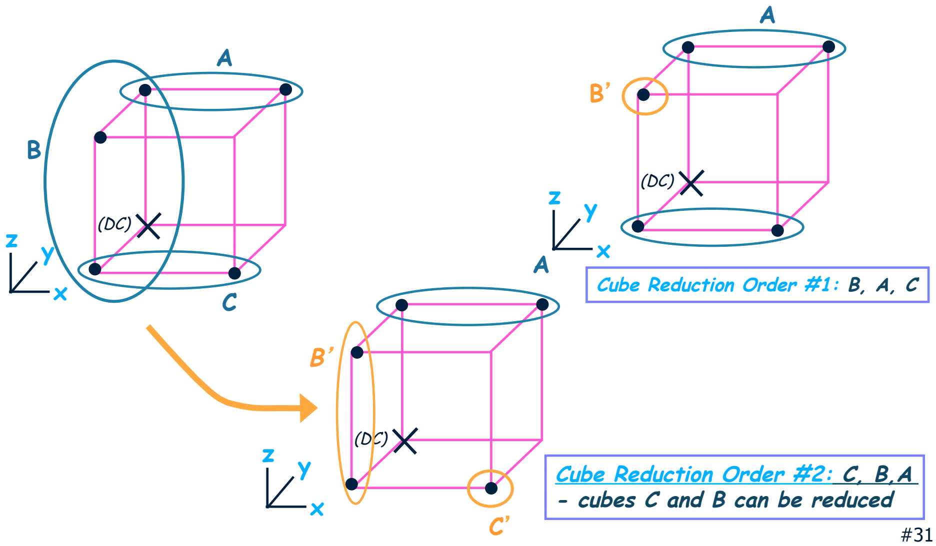


Cube Reduction Order #1: B, A, C
- only B can be reduced

#30

The "REDUCE" Step

Key Observation: reduce operation is ***order-dependent***



#31