CSEE 6861 CAD of Digital Systems
Handout: Lecture #5
2/18/16

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

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

ESPRESSO: Advanced Steps
(i) Essentials #2
(ii) MAKE-SPARSE

ESPRESSO: "Essentials #2" Step

generating all essentials when <u>not</u> given all primes

"Essentials #2" Step

<u>Example: given **any** "prime cover" F of a Boolean function f</u>

(prime cover = cover using only prime implicants, i.e. fully-expanded cubes)

Q1. Does the cover include all essentials?

yz \ wx	00 A	01	11	10
00	1	1	0	0
01	0 B	1	1 C	0
11	0	0	1 D	1
10	0	0	0	0

"Essentials #2" Step

Example: given any "prime cover" F of a Boolean function f

Q1. Does the cover include all essentials?

yz \ wx	00 A	01	11	10
00	1	1	0	0
01	0 B	1	1 C	0
11	0	0	1 D	1
10	0	0	0	0

A1. YES! Every prime cover F of a function f includes all its essentials

"Essentials #2" Step

Example: given any "prime cover" F of a Boolean function f

Q2. Which cubes in the cover are essential?

yz \ wx	00 A	01	11	10
00	1	1	0	0
01	0 B	1	1 C	0
11	0	0	1 D	1
10	0	0	0	0

"Essentials #2" Step

Example: given any "prime cover" F of a Boolean function f

Q2. Which cubes in the cover are essential?

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

A B C D

Note: cover F does not necessarily include all primes!

missing prime!

#7

"Essentials #2" Step

Example: given any "prime cover" F of a Boolean function f

Q2. Which cubes in the cover are essential?

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

A B C D

essentials

#8

“Essentials #2” Step

Example: given any “prime cover” F of a Boolean function f

Q2. Which cubes in the cover are essential?

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

A B C D

not essentials

“Essentials #2” Step

Example: given any “prime cover” F of a Boolean function f

Q2. Which cubes in the cover are essential?

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

A B C D

… how to identify all essentials when cover F does not include all primes?

not essentials

"Essentials #2" Step

Example: given any "prime cover" F of a Boolean function f

Q2. Which cubes in the cover are essential?

yz \ wx	00 A	01	11	10
00	1	1	0	0
01	0 B	1	1 C	0
11	0	0	1 D	1
10	0	0	0	0

CASE ANALYSIS:
3 types of minterms in cubes

"Essentials #2" Step

Example: given any "prime cover" F of a Boolean function f

Q2. Which cubes in the cover are essential?

yz \ wx	00 A	01	11	10
00	1	1	0	0
01	0 B	1	1 C	0
11	0	0	1 D	1
10	0	0	0	0

CASE ANALYSIS:
3 types of minterms in cubes

Each ON-set minterm is:
⇨ *(i) covered by 2 or more cubes*

“Essentials #2” Step

Example: given any “prime cover” F of a Boolean function f

Q2. Which cubes in the cover are essential?

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

A B C D

CASE ANALYSIS:
3 types of minterms in cubes

Each ON-set minterm is:
(i) covered by 2 or more cubes
⇨ *(ii) covered by only 1 cube*

“Essentials #2” Step

Example: given any “prime cover” F of a Boolean function f

Q2. Which cubes in the cover are essential?

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

A B C D

CASE ANALYSIS:
3 types of minterms in cubes

Each ON-set minterm is:
(i) covered by 2 or more cubes
⇨ *(ii) covered by only 1 cube*

2 sub-cases: …

“Essentials #2” Step

Example: given any “prime cover” F of a Boolean function f

Q2. Which cubes in the cover are essential?

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

CASE ANALYSIS:
3 types of minterms in cubes

Each ON-set minterm is:
(i) covered by 2 or more cubes
⇨ *(ii) covered by only 1 cube*

2 sub-cases: …
(a) ON-set minterm “isolated”:
no neighboring cubes (only OFF-set)

#15

“Essentials #2” Step

Example: given any “prime cover” F of a Boolean function f

Q2. Which cubes in the cover are essential?

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

CASE ANALYSIS:
3 types of minterms in cubes

Each ON-set minterm is:
(i) covered by 2 or more cubes
⇨ *(ii) covered by only 1 cube*

2 sub-cases: …
(b) ON-set minterm not “isolated”:
has neighboring adjacent cube(s)

#16

"Essentials #2" Step

Example: given any "prime cover" F of a Boolean function f

Q2. Which cubes in the cover are essential?

yz \ wx	00 A	01	11	10
00	1	1	0	0
01	0 B	1	1 C	0
11	0	0	1 D	1
10	0	0	0	0

Test for Case #2(b):
not in "isolated regions" =
has neighboring cubes

⇨ ***Grow "consensus" cubes:***
… between each adjacent cube pair

#17

"Essentials #2" Step

Example: given any "prime cover" F of a Boolean function f

Q2. Which cubes in the cover are essential?

yz \ wx	00 A	01	11	10
00	1	1	0	0
01	0 B	1	1 C	0
11	0	0	1 D	1
10	0	0	0	0

X = CONSENSUS (cube B, cube C)

Test for Case ii(b):
not in "isolated regions" =
Has Cube Neighbors

⇨ ***Grow "consensus" cubes:***
… between each adjacent cube pair
["adjacent" = distance-1]

#18

"Essentials #2" Step

Example: given any "prime cover" F of a Boolean function f

Q2. Which cubes in the cover are essential?

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

A B C D

X = CONSENSUS (cube B, cube C)

Test for Case #2(b):
not in "isolated regions" =
Has Cube Neighbors

⇨ ***Intuition: experiment to "grow" missing implicants***

- ***Consensus cube = "seed" which spans gap between adjacent cube pair (non-intersecting)***
- ***Idea: used to LOCALLY generate the "core" of missing primes in cover (consensus often is non-prime)***

#19

"Essentials #2" Step

Example: given any "prime cover" F of a Boolean function f

Q2. Which cubes in the cover are essential?

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

A B C D

X = CONSENSUS (cube B, cube C)

A2. If a cube w in cover F is not entirely covered by the union of:
(i) the set of other cubes in cover, and
(ii) the set of consensus cubes (i.e seeds of missing primes)
then it is essential!

#20

"Essentials #2" Step

Example: given any "prime cover" F of a Boolean function f

Q2. Which cubes in the cover are essential?

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

A B C D

X = CONSENSUS (cube B, cube C)

A2. If a cube w in cover F is not entirely covered by the union of:
(i) the set of other cubes in cover, and
(ii) the set of consensus cubes (i.e seeds of missing primes)
then it is essential!

EXAMPLE #1: Cube A Essential

#21

"Essentials #2" Step

Example: given any "prime cover" F of a Boolean function f

Q2. Which cubes in the cover are essential?

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

A B C D

X = CONSENSUS (cube B, cube C)

A2. If a cube w in cover F is not entirely covered by the union of:
(i) the set of other cubes in cover, and
(ii) the set of consensus cubes (i.e seeds of missing primes)
then it is essential!

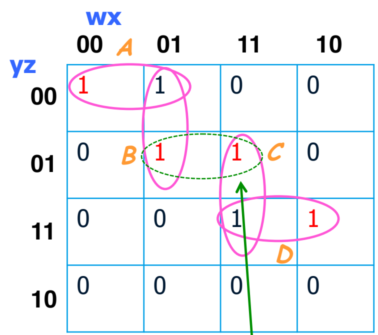
EXAMPLE #2: Cube C Not Essential

#22

"Essentials #2" Step

Example: given any "prime cover" F of a Boolean function f

Q2. Which cubes in the cover are essential?



X = CONSENSUS (cube B, cube C)

A2. Algorithmic Formulation: ESSEN #2*

Given cover F, including cube 'e' (to check if e essential):

1. Remove 'e' from F:
   G = F – {e}
2. Compute consensus of 'e' and each cube in G:
   H = consensus{e, G}
3. Formulate containment problem:
   Check if e <= G U H

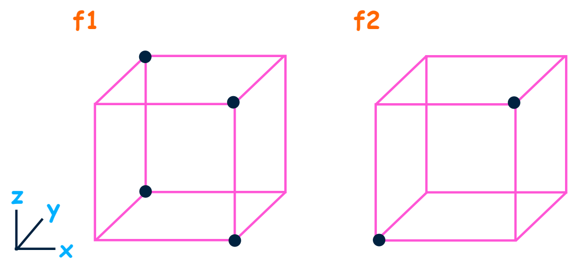
**equivalent to Hachtel/Somenzi, Theorem 5.4.1 (p. 204)*

#23

ESPRESSO: "Make-Sparse" Step

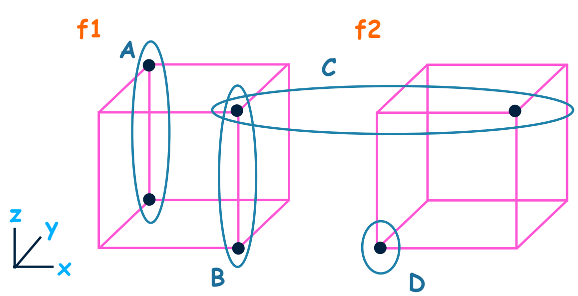
Multi-Output Minimization: Example #1

Multi-Output Function



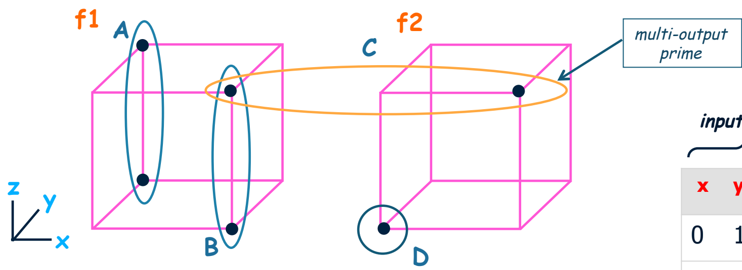
Multi-Output Minimization: Example #1

Cover #1A: min-cost cover, using ONLY multi-output primes



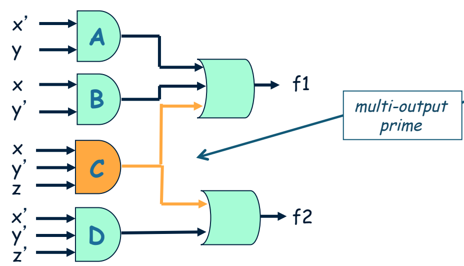
Multi-Output Minimization: Example #1

Cover #1A: using ONLY multi-output primes



input part			output part	
x	y	z	f1	f2
0	1	-	1	0
1	0	-	1	0
1	0	1	1	1
0	0	0	0	1

PLA Representation of Cover = "cubical complex"



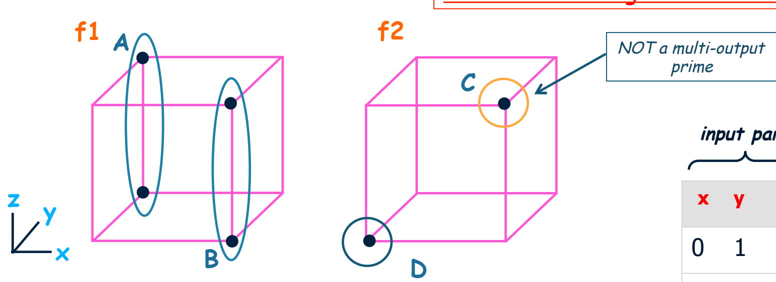
corresponding 2-level implementation

COST = 15 gate inputs

#27

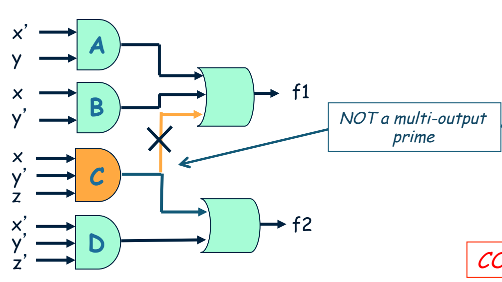
Multi-Output Minimization: Example #1

Cover #1B: using some NON-PRIMES!



input part			output part	
x	y	z	f1	f2
0	1	-	1	0
1	0	-	1	0
1	0	1	0	1
0	0	0	0	1

PLA Representation of Cover = "cubical complex"



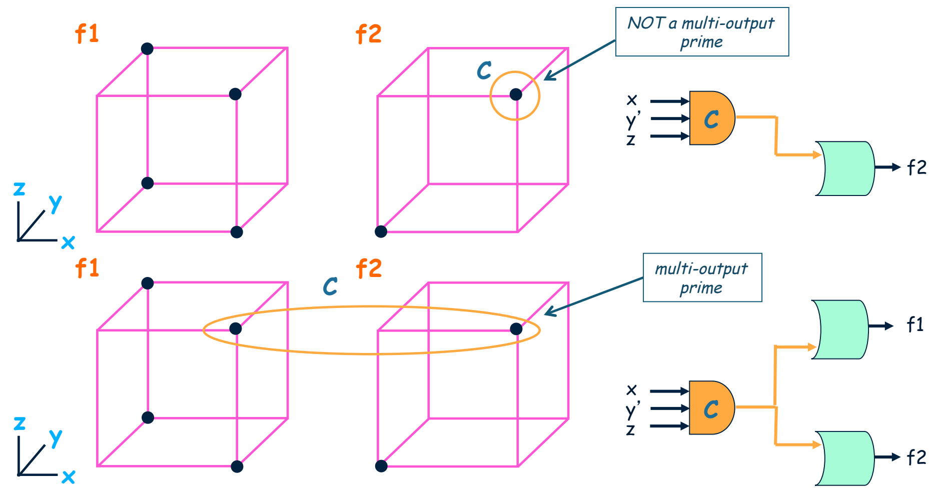
corresponding 2-level implementation

COST = 14 gate inputs = better!

#28

Observation

Peculiar feature of multi-output primes (spanning multiple functions):



⇨ **Spanning more functions: (i) increases coverage, BUT (ii) also increases cost!**

#29

ESPRESSO Strategy: using MAKE-SPARSE (simple version)

Basic idea:

- Initial ESPRESSO goal: body of algorithm always 'expands' to multi-output primes
- Post-processing step (at end of algorithm) = **MAKE-SPARSE**
  - heuristically reduce final cover cost by DELETING UNNECESSARY OUTPUT CONNECTIONS
- Basic version of "MAKE-SPARSE":
  -- this step is a form of multi-output "REDUCE" =
  eliminate unnecessary AND-gate wire fanouts
  -- also known as "reduce output parts"

⇨ **See previous slides:**
- before MAKE-SPARSE: Cover #1A (cost = 15 gate inputs)
- after MAKE-SPARSE: Cover #1B (cost = 14 gate inputs)

#30

MAKE-SPARSE: advanced version

Used in Espresso-II:

- Key Idea: improved "MAKE-SPARSE" can involve new expansion too!

Step #1. Use restricted "reduce": "REDUCE OUTPUT PARTS"

for each AND-gate, eliminate unnecessary output connections

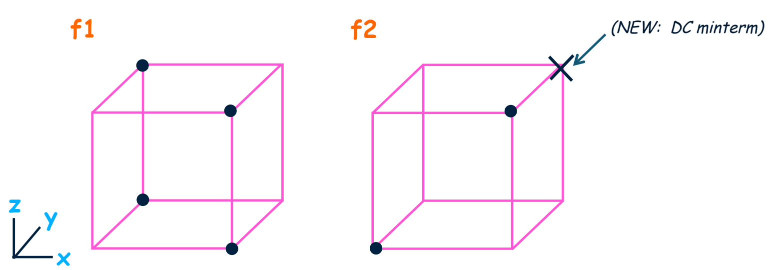
Step #2. Use restricted "expand": "EXPAND INPUT PARTS"

for each AND-gate, expand if possible within current outputs

#31

Multi-Output Minimization: Example #2

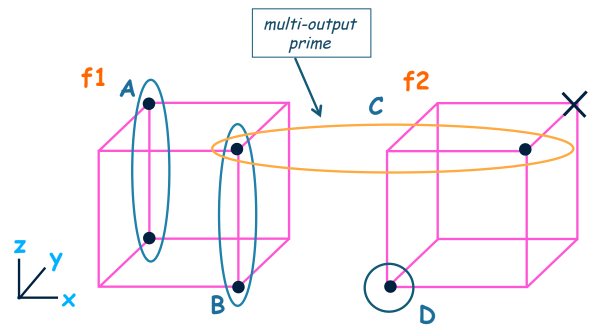
Multi-Output Function: illustrates advanced MAKE-SPARSE operation



#32

Multi-Output Minimization: Example #2

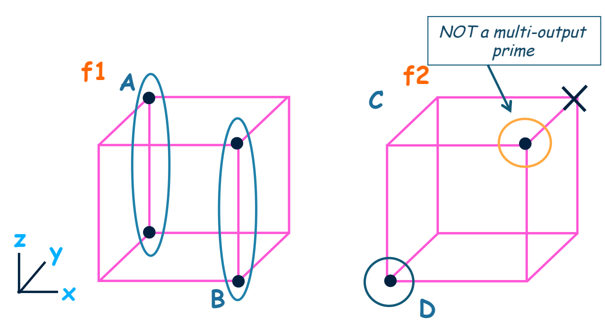
Cover #2A: initial min-cost cover, using ONLY multi-output primes



#33

Multi-Output Minimization: Example #2

STEP #1 -- Cover #2B: after RESTRICTED REDUCE of output part
= "REDUCE OUTPUT PART"



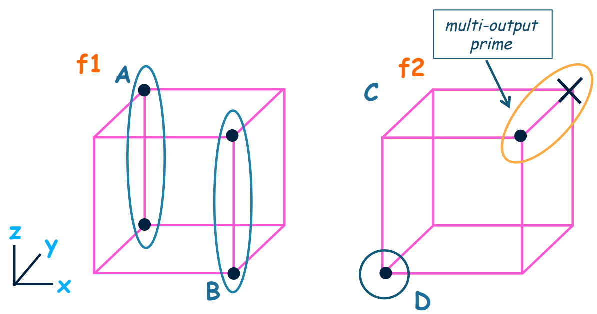
#34

Multi-Output Minimization: Example #2

STEP #2 -- Cover #2C: after RESTRICTED EXPAND of input part

- no expansion to new outputs, only **within current outputs**!

= "EXPAND INPUT PART"

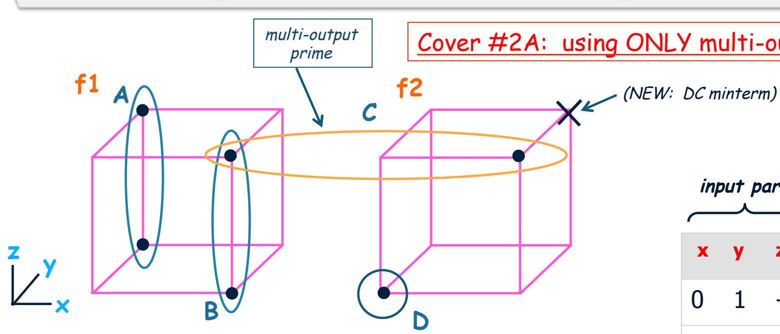


FINAL COVER

#35

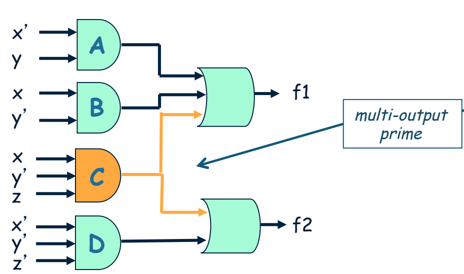
Multi-Output Minimization: Example #2 -- details

Cover #2A: using ONLY multi-output primes



input part			output part	
x	y	z	f1	f2
0	1	-	1	0
1	0	-	1	0
1	0	1	1	1
0	0	0	0	1

PLA Representation = "cubical complex"



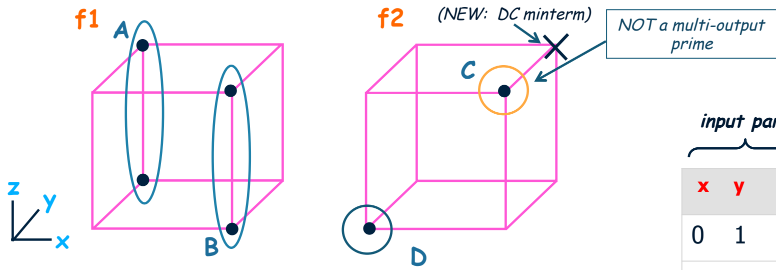
corresponding 2-level implementation

COST = 15 gate inputs

#36

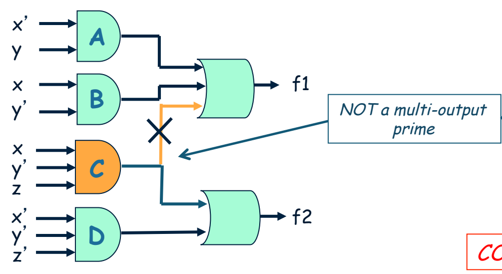
Multi-Output Minimization: Example #2 -- details

STEP #1 -- Cover #2B: REDUCE OUTPUT PART



	input part			output part
x	y	z	f1	f2
0	1	-	1	0
1	0	-	1	0
1	0	1	0	1
0	0	0	0	1

PLA Representation of Cover
= "cubical complex"



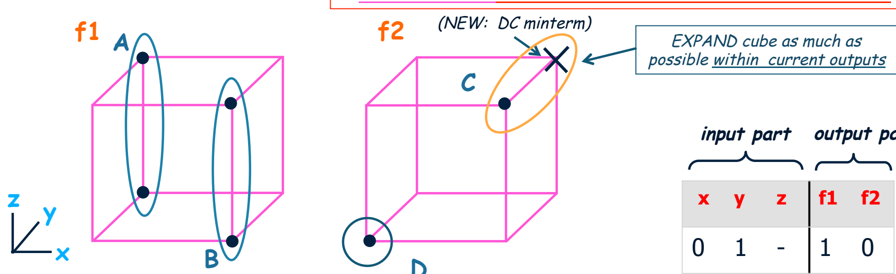
corresponding 2-level implementation

COST = 14 gate inputs = better!

#37

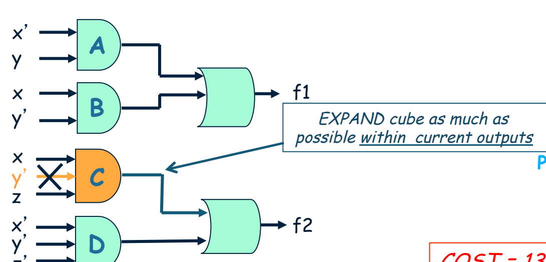
Multi-Output Minimization: Example #2 -- details

STEP #2 -- Cover #2C: EXPAND INPUT PART



	input part			output part
x	y	z	f1	f2
0	1	-	1	0
1	0	-	1	0
1	-	1	0	1
0	0	0	0	1

PLA Representation of Cover
= "cubical complex"



corresponding 2-level implementation

COST = 13 gate inputs = even better!

#38