


Administrative Matters

- Homework #2 posted
 - Due 10/27, electronically
 - You can scan it and send that in...
- Midterm examination
 - November 1st 9:40AM sharp!
 - Close everything
 - Formula sheets will be given
- Nov 3rd Lecture cancelled
 - Work on your project/presentation!
- Student presentation
 - Nov 8th and Nov 10th
 - 30 minutes each student + 10 minutes Q&A



1

CS220: Synthesis of Digital System, Fall 05

Topic of presentation


- Scott
 - Project presentation
 - Reconfigurable Computing/Virtex/ ...
- Ryan
 - Project presentation
 - Embedded System Building Blocks/sensor net/...
- Eric
 - Project presentation
 - System level synthesis/Spark/...
- John
 - Quasi-Static Scheduling

- Nov 8th

- 9:40-10:20 Scott
- 10:20-11:00 Ryan

- Nov 10th

- 9:40-10:20 Eric
- 10:20-11 John



2

CS220: Synthesis of Digital System, Fall 05

Heuristic minimization

- Provide irredundant covers with “reasonably small” cardinality
- Fast and applicable to many real-life functions
- Avoid bottlenecks of exact minimization
 - Prime generation and storage
 - Covering

3

CS220: Synthesis of Digital System, Fall 05

Heuristic minimization principle

- Local minimum cover
 - Given initial cover
 - Make it prime
 - Make it irredundant
- Iterative improvement
 - Improve on cardinality by “modifying the implicants”
 - Basic operations
 - Expand
 - Reduce
 - Reshape
 - Irredundant

4

CS220: Synthesis of Digital System, Fall 05

Heuristic minimization operators

- Expand
 - Make implicants prime
 - Remove covered implicants
- Reduce
 - Reduce size of each implicant while preserving cover
 - Make it easier to be covered and removed
- Reshape
 - Modify implicant pairs
 - Expand one and reduce the other
- Irredundant
 - Make cover irredundant

5

CS220: Synthesis of Digital System, Fall 05

Example: Mini

• Given minterm table

0000 1	
0010 1	
0100 1	
0110 1	
1000 1	
1010 1	
0101 1	0**0 α
0111 1	1000 1
1001 1	1010 1
1011 1	0101 1
1101 1	0111 1
1101 1	1001 1
	1011 1
	1101 1

• Expand 0000 to prime $\alpha = 0**0$

- Drop 0100, 0010, 0110 from cover

• Expand 1000 to $\beta = **0*0$

- Drop 1010 from the cover

	0101 1	0**0 α
	0111 1	**0*0 β
	1001 1	
	1011 1	
	1101 1	

6

CS220: Synthesis of Digital System, Fall 05

Example: Mini

- Expand 0101 to $\gamma = 01^{**}$
 - Drop 0111 from the cover
- Expand 1001 to $\delta = 10^{**}$
 - Drop 1011 from the cover
- Expand 1101 to $\varepsilon = 1^{*}01$
- Cover is : $\{\alpha, \beta, \gamma, \delta, \varepsilon\}$
 - Prime, redundant, minimal w.r.t. SCC

0**0 α		0**0 α
*0*0 β		*0*0 β
01** γ	←	0101 1
1001 1		0111 1
1011 1		1001 1
1101 1		1011 1
		1101 1

0**0 α		0**0 α
*0*0 β		*0*0 β
01** γ		01** γ
10** δ	→	10** δ
1101 1		1*01 ε

CS220: Synthesis of Digital System, Fall 05 7

Example: Mini

- Reduce $\alpha = 0^{**}0$ to nothing
- Reduce $\beta = *0^{*}0$ to $\tilde{\beta} = 00^{*}0$
 - $10^{*}0$ is covered by δ
- Reduce $\varepsilon = 1^{*}01$ to $\tilde{\varepsilon} = 1101$
 - 1001 is covered by δ
- Cover is : $\{\tilde{\beta}, \gamma, \delta, \tilde{\varepsilon}\}$

*0*0 β		0**0 α
01** γ	←	*0*0 β
10** δ		01** γ
1*01 ε		10** δ
		1*01 ε

00*0 $\tilde{\beta}$		0**0 α
01** γ		*0*0 β
10** δ		01** γ
1*01 $\tilde{\varepsilon}$		10** δ
		1*01 ε

CS220: Synthesis of Digital System, Fall 05 8

Example: Mini

- Reshape $\{\tilde{\beta}, \delta\}$ to $\{\beta, \tilde{\delta}\}$
 - Where $\tilde{\delta} = 10^{*}1$
- Expand $\tilde{\delta} = 10^{*}1$ to $\delta = 10^{**}$
- Expand $\tilde{\varepsilon} = 1101$ to $\varepsilon = 1^{*}01$
- Cover is : $\{\beta, \gamma, \delta, \varepsilon\}$
 - Prime, Irredundant

*0*0 β		00*0 β
01** γ	←	01** γ
10** δ		10** δ
1001 $\tilde{\varepsilon}$		1101 ε

*0*0 β		00*0 β
01** γ		01** γ
10** δ	→	10** δ
1001 $\tilde{\varepsilon}$		1*01 ε

CS220: Synthesis of Digital System, Fall 05 9

Testability

- Manufacturing testing for fabrication defect
 - Stuck at fault model
- Two-level cover is fully testable
 - If it is prime and irredundant
- E.g. $a'b' + b'c + ac + ab$
 - ac stuck at 0 is not testable
 - It is redundant
 - In fact, some logic minimization algorithm use ATPG as core
 - $a'b' + b'c + ab$ is fully testable

CS220: Synthesis of Digital System, Fall 05 10

Matrix representation of logic covers

- Used in logic minimizers
- Different formats
- Usually one row per implicant
- Symbols: 0,1,*...

10	11	11	10
10	01	11	11
01	10	11	11
01	11	10	01

CS220: Synthesis of Digital System, Fall 05 11

The positional cube notation

- Encoding scheme

∅	00
0	10
1	01
*	11
- Operations
 - Intersection – AND
 - Union – OR
- ∅ symbol means implicant is void and should be removed
- * means implicant is full in that variable
 - Don't care

CS220: Synthesis of Digital System, Fall 05 12

Example

- $f = a'd' + a'b + ab' + ac'd$

10	11	11	10
10	01	11	11
01	10	11	11
01	11	10	01

- $a'd' \text{ AND } a'b = 10\ 01\ 11\ 10 = a'bd'$
- $a'd' \text{ AND } ab' = 00\ 10\ 11\ 10 = \emptyset$
 - Can be removed

13

Multi-value input functions

- Input variables can have many values
- Representations
 - Literals: set of valid values
 - Sum of products of literals
 - Let's say a, b are binary, but f can be 0, 1, or 2
 - $a^{(0)}b^{(0)}f^{(0)} + a^{(0)}b^{(1)}f^{(2)} + a^{(1)}b^{(0)}f^{(2)} + a^{(1)}b^{(1)}f^{(0,1)}$
- Extension of positional cube notation

10	10	100
10	01	001
01	10	001
01	01	110

14

Multiple output functions

- Representation by its characteristic function
 - $y = f(x)$ can be written as $\chi(x,y) = 1$
- $f_0 = a'b' + ab$; $f_1 = ab$; $f_2 = ab' + a'b$

10	10	100
10	01	001
01	10	001
01	01	110

- Represented in as multi-value input function
 - $a^{(0)}b^{(0)}f^{(0)} + a^{(0)}b^{(1)}f^{(2)} + a^{(1)}b^{(0)}f^{(2)} + a^{(1)}b^{(1)}f^{(0,1)}$
- Input/output parts are encoded together
 - No need to talk about multiple output functions anymore
 - Simply transform it to multi-value input single Boolean output functions

15

Notations for operations between functions

- Size of a literal is the number of 1's in its field
 - 110 size 2
- Size of implicant is the product of sizes of its literals
 - 110 11 size 4
- Intersection
 - Compute by bitwise product
 - 110 11 and 100 01 is 100 01
- Supercube of two implicants is the smallest containing cube
 - Compute by bitwise or
 - Supercube(10 10 100; 10 01 001) is 10 11 101
- Distance of two implicants is the number of empty fields in their bitwise and
 - Distance (110 11; 100 01) is 0; Distance (110 11; 001 01) is 1

16

Cofactor

- First test to see if there is intersection
 - Then perform $\alpha_1 + \beta_1' \dots \alpha_n + \beta_n'$

E.g. $f = x'y' + xy$

	10	10
	01	01

- Take cofactor w.r.t. $x = [01\ 11]$
- First row - void, no intersection
- Second row 11 01 = y
 - = f_a
- Now we can take cofactor w.r.t. any cube (multi-value!)

17