



# CprE / ComS 583 Reconfigurable Computing

Prof. Joseph Zambreno  
Department of Electrical and Computer Engineering  
Iowa State University

Lecture #4 – FPGA Technology Mapping



## Quick Points

- Lectures are viewable for on-campus students via WebCT
- Class e-mail list created: [cpre583@iastate.edu](mailto:cpre583@iastate.edu)
- Less focus on interconnect theory
  - More on interconnects in actual devices
  - Read [AggLew94], [ChaWon96A], [Deh96A] for more details

August 31, 2006

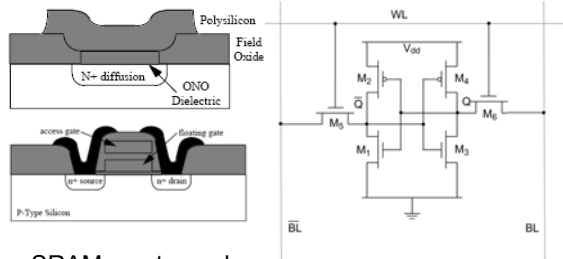
CprE 583 – Reconfigurable Computing

Lect-04.2



## Recap

- Various FPGA programming technologies (Anti-fuse, (E)EPROM, Flash, SRAM):



- SRAM most popular

August 31, 2006

CprE 583 – Reconfigurable Computing

Lect-04.3



## LUTs and Digital Logic

- $k$  inputs  $\rightarrow 2^k$  possible input values
- $k$ -LUT corresponds to  $2^k \times 1$  bit memory
  - Truth table is stored
  - $2^{2^k}$  possible functions –  $O(2^{2^k} / k!)$  unique

$$F = A_0A_1A_2 + \bar{A}_0A_1\bar{A}_2 + \bar{A}_0\bar{A}_1\bar{A}_2$$

$A_2$	$A_1$	$A_0$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	...	255
0	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	...	1
0	0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	...	0
0	1	0	0	0	0	0	1	1	1	0	0	0	0	1	1	1	1	1	...	1
0	1	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	...	0
1	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	...	0
1	0	1	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	...	0
1	1	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	...	0
1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	...	1

August 31, 2006

CprE 583 – Reconfigurable Computing

Lect-04.4



## Outline

- Recap
- General Routing Architectures
- FPGA Architectural Issues
- Early Commercial FPGAs
  - Xilinx XC3000
  - Xilinx XC4000
- Technology Mapping using LUTs

August 31, 2006

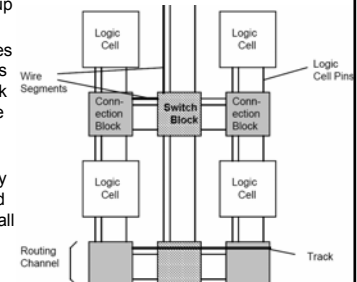
CprE 583 – Reconfigurable Computing

Lect-04.5



## General Routing Architecture

- A *wire segment* is a wire unbroken by programmable switches
- A *track* is a sequence of one or more wire segments in a line
- A routing *channel* is a group of parallel tracks
- A *connection block* provides connectivity from the inputs and outputs of a logic block to the wire segments in the channels
- A *switch block* is a block which provides connectivity between the horizontal and vertical wire segments on all four of its sides



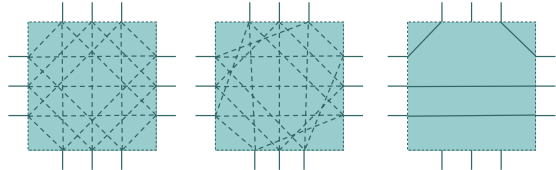
August 31, 2006

CprE 583 – Reconfigurable Computing

Lect-04.6

## Switch Boxes

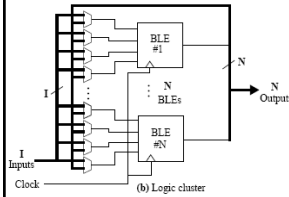
- $F_s$  – connections offered per incoming wire
- *Universal* switchbox can connect any set of inputs to their target output channels simultaneously
  - Build-able with  $F_s = 3$
  - Xilinx XC4000 switchbox is  $F_s = 3$  but not universal



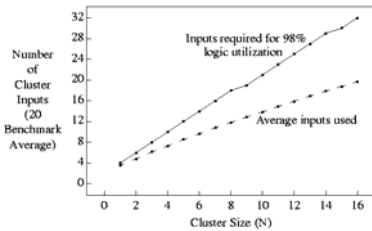
- Read [ChaWon96A] for more details

## Architectural Issues [AhmRos04A]

- What values of  $N$ ,  $I$ , and  $K$  minimize the following parameters?
  - Area
  - Delay
  - Area-delay product
- Assumptions
  - All routing wires length 4
  - Fully populated IMUX
  - Wiring is half pass transistor, half tri-state

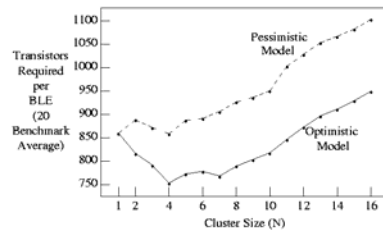


## Number of Inputs per Cluster



- Lots of opportunities for input sharing in large clusters [BetRos97A]
- Reducing inputs reduces the size of the device and makes it faster
- Most FPGA devices (Xilinx) have 4 BLE per cluster with more inputs than actually needed

## Logic Cluster Size



- Small block cluster more efficient
- Includes area needed for routing
- Smallest clusters (e.g. one BLE per cluster) not “CAD friendly”
- Most commercial devices have 4-8 BLEs per cluster

## Effect of N and K on Area

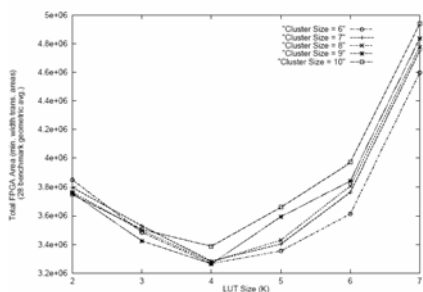


Fig. 8. Total Area for Clusters of Size 6 to 10

- Cluster size of  $N = [6-8]$  is good,  $K = [4-5]$

## Effect of N and K on Performance

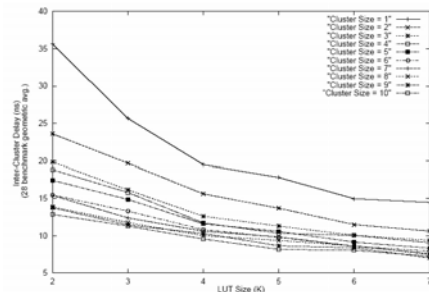


Fig. 17. Total Inter-Cluster Delay for Clusters of Size 1 to 10

- Inconclusive: Big  $K$  and  $N > 3$  value looks good

### Effect of N and K on Area-Delay

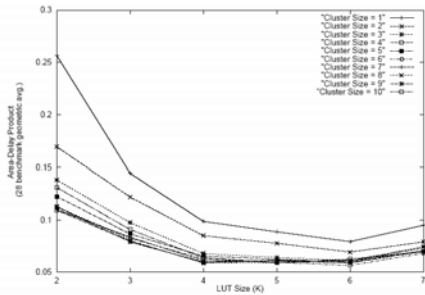


Fig. 20. Area-Delay Product for Clusters of Size 1 to 10

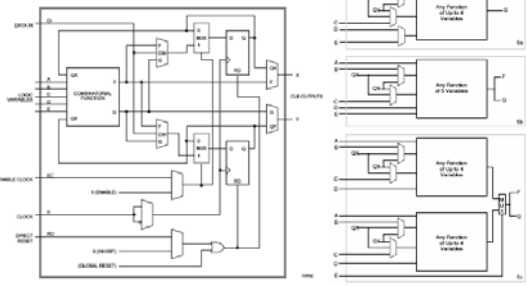
- K = 4-6, N = 4-10 looks OK

### Putting it All Together

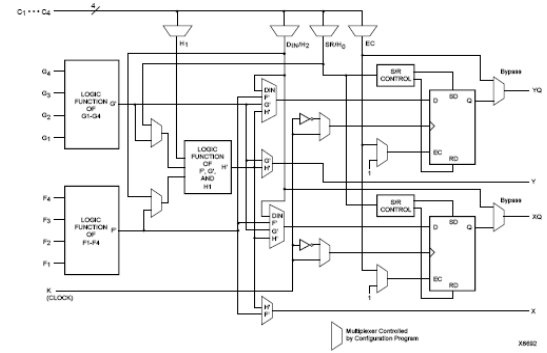
- Area:
  - LUT count decreases with  $k$  (slower than exponential)
  - LUT size increases with  $k$  (exponential logic area, ~linear interconnect area)
- Delay:
  - LUT depth decreases with  $k$  (logarithmic)
  - LUT delay increases with  $k$  (linear)
- Examples:
  - Xilinx XC3000 family
    - $F_8 = 3$
    - $I = 5$
    - $N = 2$
  - Xilinx XC4000 family
    - $F_8 = 3$
    - $I = 9$
    - $N \sim 2.5$

### XC3000 Logic Block

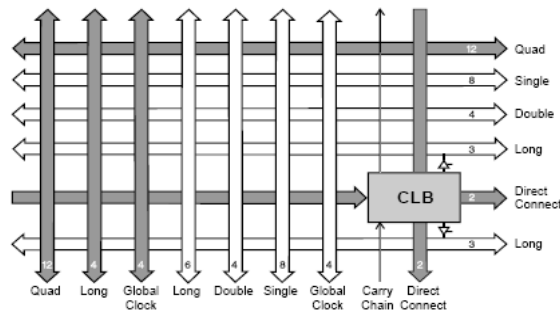
- 5-LUT, or two 4-LUTs



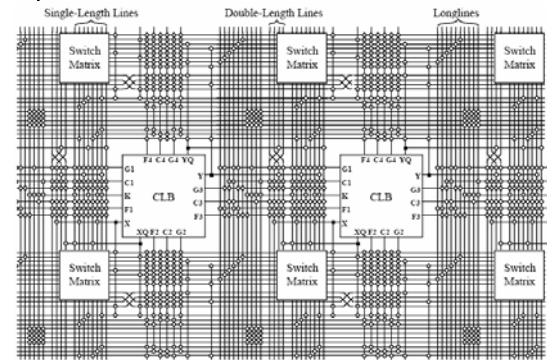
### XC4000 Logic Block



### XC4000 Routing Structure

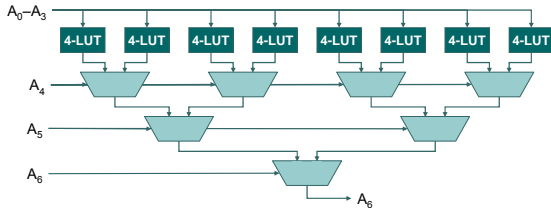


### XC4000 Routing Structure (cont.)



### LUT Computational Limits

- $k$ -LUT can implement  $2^{2^k}$  functions
  - Given  $n$  such  $k$ -LUTs, can implement  $(2^{2^k})^n$
  - Since 4-LUTs are efficient, want to find  $n$  such that  $(2^{2^4})^n \geq 2^{2^M}$
- Example – implementing a 7-LUT with 4-LUTs:



August 31, 2006 CprE 583 – Reconfigurable Computing Lect-04.19

### LUT Computational Limits (cont.)

- How much computation can be performed in a table lookup?
- Upper bound (from previous) –  $n \leq 2^{M-3}$
- Need  $n$  4-LUTs to cover a  $M$ -LUT:

$$(2^{2^4})^n \geq 2^{2^M}$$

$$n \log(2^{2^4}) \geq \log(2^{2^M})$$

$$n 2^4 \log(2) \geq 2^M \log(2)$$

$$n 2^4 \geq 2^M$$

$$n \geq 2^{M-4}$$

- Adding upper bound –  $2^{M-4} \leq n \leq 2^{M-3}$

August 31, 2006 CprE 583 – Reconfigurable Computing Lect-04.20

### LUTs Versus Memories

- Can also implement  $(2^{2^k})^w$  as a single large memory with  $k$  inputs and  $w$  outputs
- Large memory advantage – no need for interconnect and only one input decoder required
- Consider a 32K x 8bit memory ( $170M \lambda^2$ , 21ns latency)
  - $w = 8$
  - $k = 16$  (or 2 8-bit inputs to address  $2^{16}$  locations)
  - Can implement an 8-bit addition or subtraction
- Xilinx XC3042 – 288 4-LUTs ( $180M \lambda^2$ , 13ns CLB delay)
- **15-bit parity calculation:**
  - 5 4-LUTs (<2% of XC4032) –  $3.125M \lambda^2$
  - Entire SRAM –  $170M \lambda^2$
- **7-bit addition:**
  - 14 4-LUTs (<5% of XC4032) –  $8.75M \lambda^2$
  - Entire SRAM –  $170M \lambda^2$

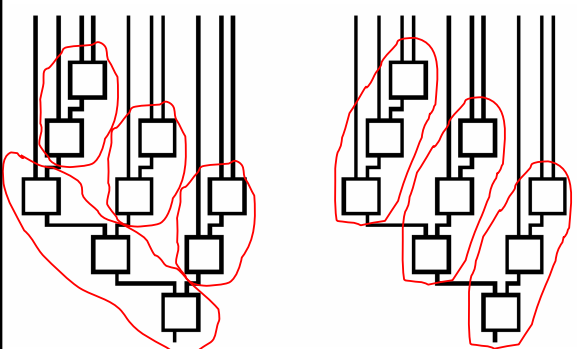
August 31, 2006 CprE 583 – Reconfigurable Computing Lect-04.21

### LUT Technology Mapping

- Task: map netlist to LUTs, minimizing area and/or delay
  - Similar to technology mapping for traditional designs
  - Library approach not feasible –  $O(2^{2^k} / k!)$  elements in library
  - In general it is NP-hard

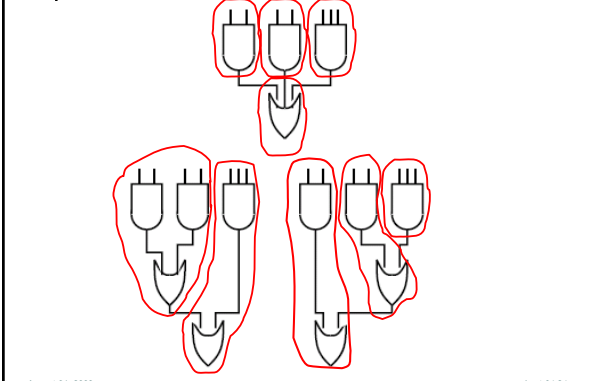
August 31, 2006 CprE 583 – Reconfigurable Computing Lect-04.22

### Area vs. Delay Mapping



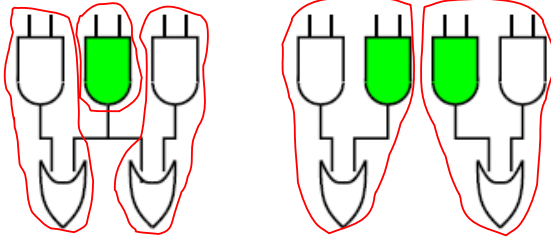
August 31, 2006 CprE 583 – Reconfigurable Computing Lect-04.23

### Decomposition



August 31, 2006 CprE 583 – Reconfigurable Computing Lect-04.24

### Why Replicate?

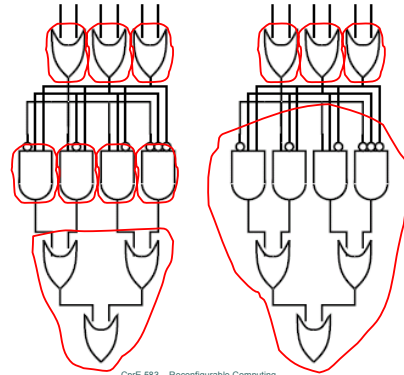


August 31, 2006

CprE 583 – Reconfigurable Computing

Lect-04.25

### Reconvergence

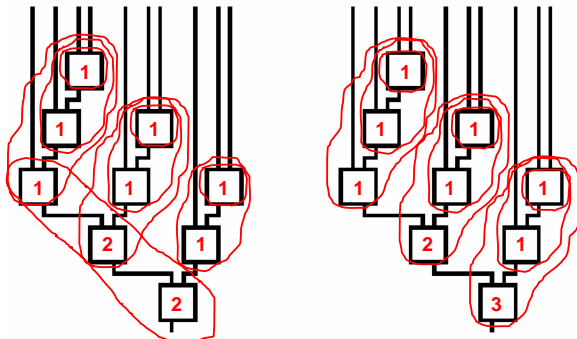


August 31, 2006

CprE 583 – Reconfigurable Computing

Lect-04.26

### Dynamic Programming



August 31, 2006

CprE 583 – Reconfigurable Computing

Lect-04.27

### Summary

- FPGA design issues involve number of logic blocks per cluster, number of inputs per logic block, routing architecture, and  $k$ -LUT size
- Can build  $M$ -LUT with  $n$   $k$ -LUTs where  $2^{M-3} \leq n \leq 2^{M-4}$
- Large LUTs generally inefficient
- Technology mapping is simplified because of 4-LUT properties
  - Techniques – decomposition, replication, reconvergence, dynamic programming
  - Area- or delay-optimal mapping still NP hard

August 31, 2006

CprE 583 – Reconfigurable Computing

Lect-04.28