

# Architectures and Algorithms for Synthesizable Programmable Logic Cores

Noha Kafafi, Kimberly Bozman, Steve Wilton

University of British Columbia  
Vancouver, B.C., Canada

This work is funded by Altera, Micronet, and NSERC

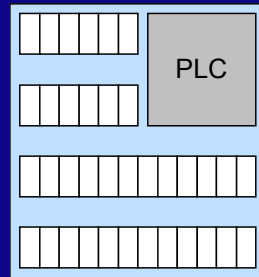
## *Configurable Logic Cores: Why?*

1. Accommodate complexity of current designs
2. Postpone some decisions until late in design cycle
3. Fast upgrade path for products
4. Can "patch up" design errors

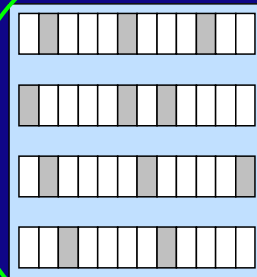
## *Soft Programmable Logic Cores:*

Use standard cells to implement PLC:

This work talks  
about  
this way



One way: Hard PLC



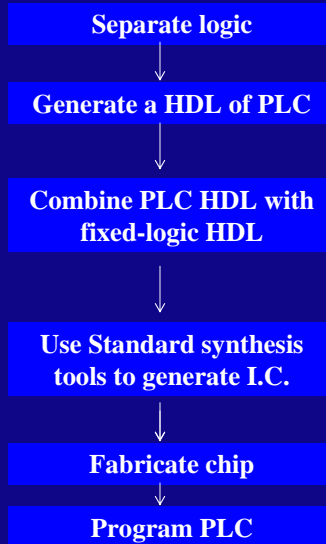
Alternative way: Soft PLC

## *Outline*

---

- Process
- New architectures
- CAD tool algorithms
- Results
- Future work
- Summary

How do we make a soft PLC?



### *Soft Programmable Logic Cores:*

#### **Advantages of using soft cores:**

- + Easy to integrate. Place and route with the rest of the ASIC
- + Very flexible, can generate exactly the core you need
- + Easy to migrate to smaller technologies

#### **Disadvantages:**

- Really inefficient compared to hard core (estimate 6-7x bigger)

#### **Our thought:**

It makes sense if you only want a small PLC (a few hundred gates, perhaps)

e.g. next state logic in state machine

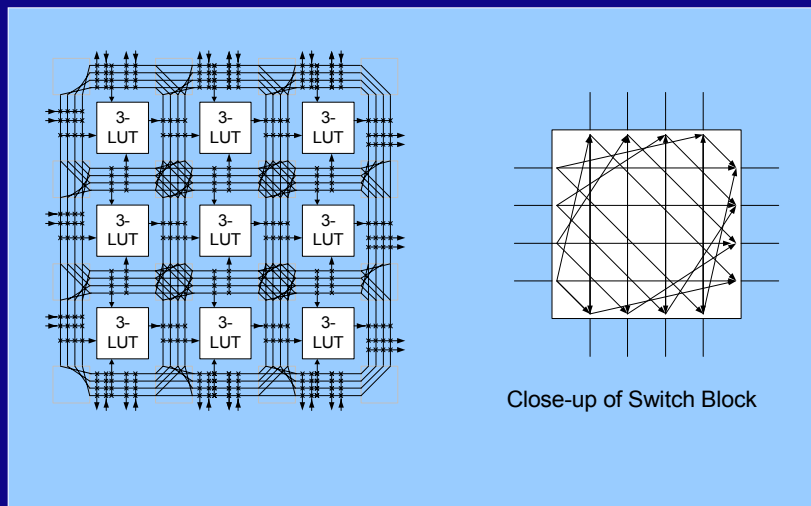
### Interesting Tid-bit:

When we synthesized our programmable logic core, we had all sorts of problems with combinational loops, but an un-programmed FPGA is full of them!

### Our solution:

We use uni-directional architectures.  
Feedback would have to be done outside the PLC

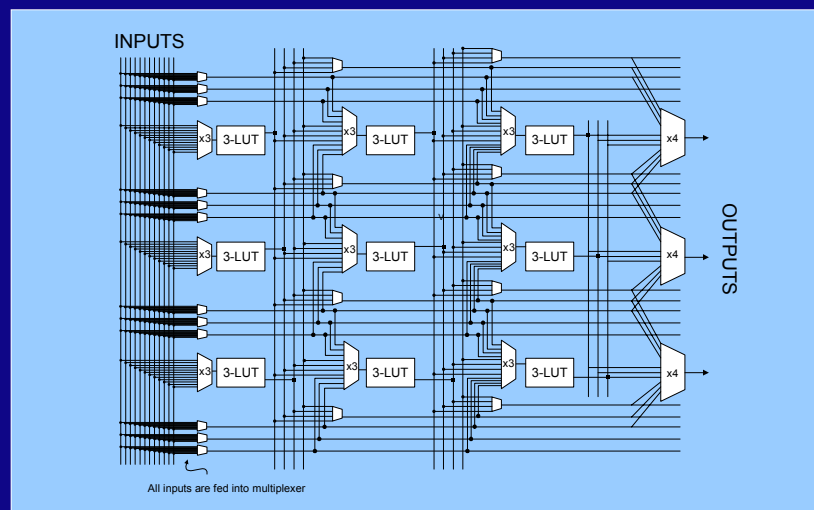
### *Directional Architecture:*



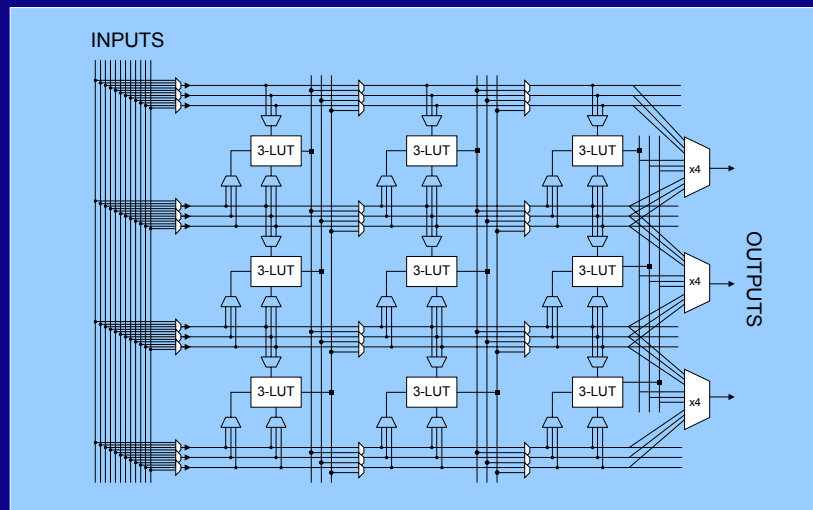
### A Few Interesting Observations:

1. Since we are only implementing small blocks, we can remove some flexibility
2. Since these blocks are hardwired to the rest of the chip, we still need lots of flexibility at the inputs and outputs
3. Each "tile" need not be identical

### Gradual Architecture:



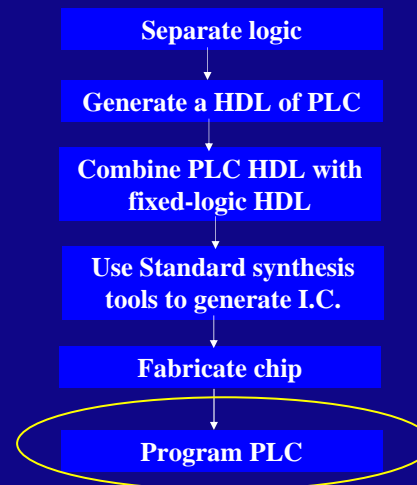
## Segmented Architecture:



## Computer -Aided Design Tools

Place and Route tools are needed to implement a user circuit on our core

- Need new algorithms for our architectures, to take into account:
  - directional aspect
  - new routing structure
- More details to come.....



## More on Placement and Routing

### Placement:

- Used simulated annealing algorithm
- Normal FPGA algorithms based on wire length or critical path delay
- For our architecture, routing resources are very limited => minimize overuse of routing multiplexors
- Goal is to achieve "Placement for Routability"

### Routing:

- Turns out this is an easy problem
- Normal FPGA routers work well

## Placement Costs

$$\text{cost} = \sum \sum [\text{MAX}(0, \text{Occ}(x,y) - \text{Cap}(x,y) + \gamma)]$$

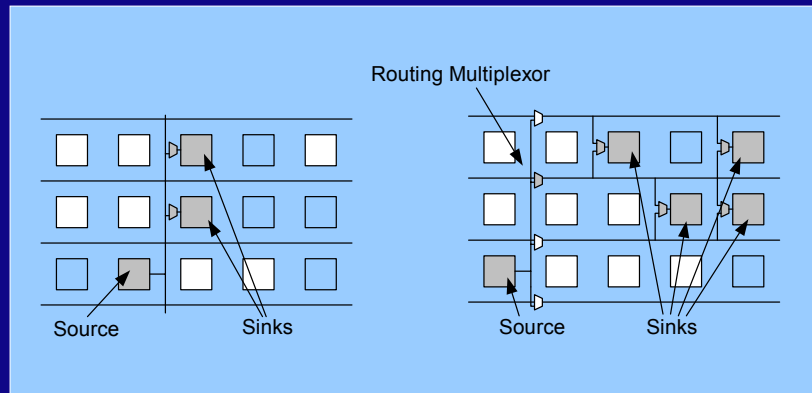
Occupancy of mux at location (x,y), is an estimate of how many nets would potentially use that routing mux

Capacity of mux at location (x,y) is equal to 1

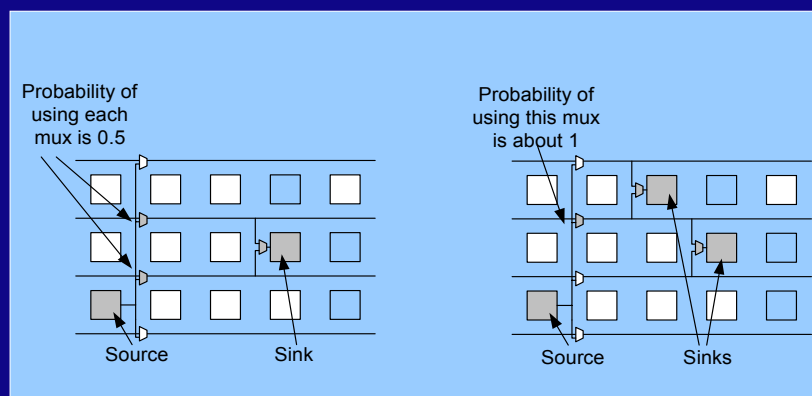
- For Segmented Architecture:

$$\text{cost} = \sum \sum \sum [\text{MAX}(0, \text{Occ}(x,y,z) - \text{Cap}(x,y,z) + \gamma)]$$

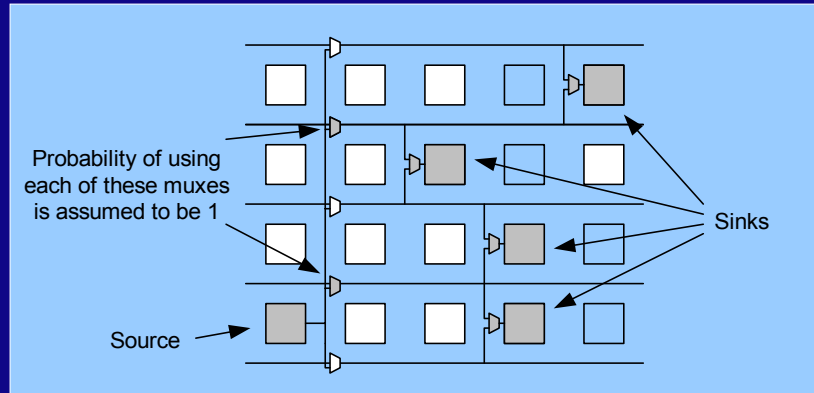
## Gradual Architecture: Some Good Placements



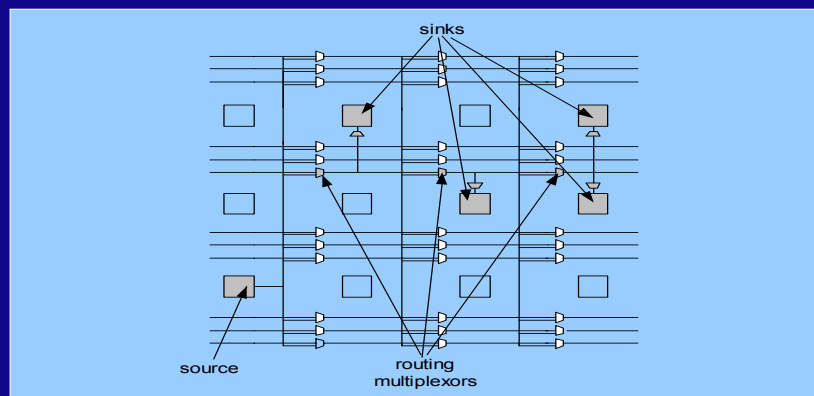
## Gradual Architecture: Estimating Mux Usage During Placement



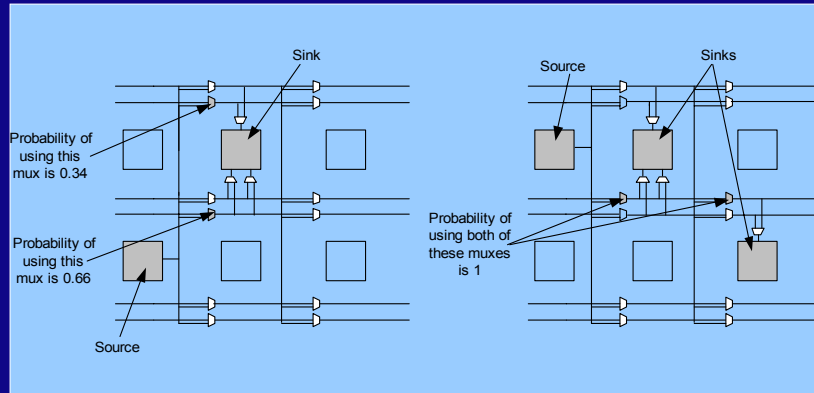
## *Gradual Architecture: Estimating Mux Usage During Placement*



## *Segmented Architecture: Some Good Placements*

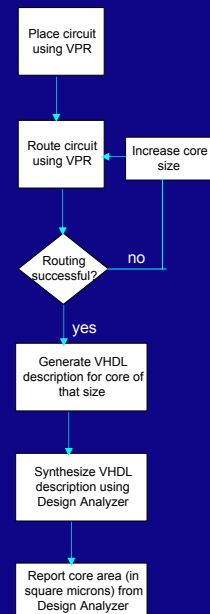


## Segmented Architecture: Estimating Mux Usage During Placement



## Experimental methodology

- Used 20 MCNC benchmark circuits
- Found the minimum sized core on which a circuit placed and routed successfully
- Synthesized a core of that size and obtained area from Design Analyzer



## Measurements and Estimates:

### Experimental Area Results:

- Results:
  - Gradual Architecture is 19% more dense than Directional Architecture
  - Gradual Architecture is 25% more dense than Segmented Architecture

### Area Estimate:

- Compared to the same size hard-FPGA, our soft FPGA is about 6.4x less dense

## Combinational Circuits:

From Hutton et al:



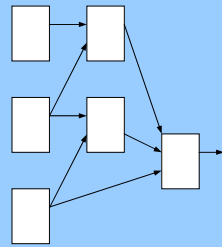
suar5.blif



sqrt8ml.blif

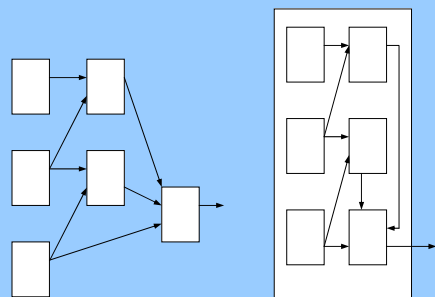
Combinational logic naturally has a “triangular” shape

## *Non-Rectangular Cores:*



Original Circuit

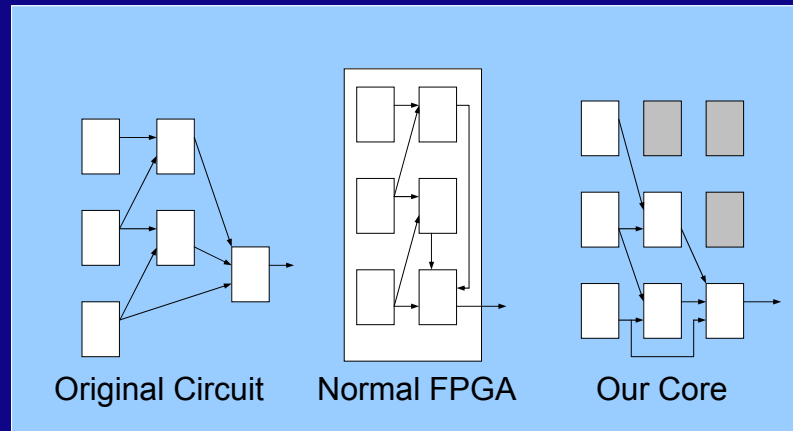
## *Non-Rectangular Cores:*



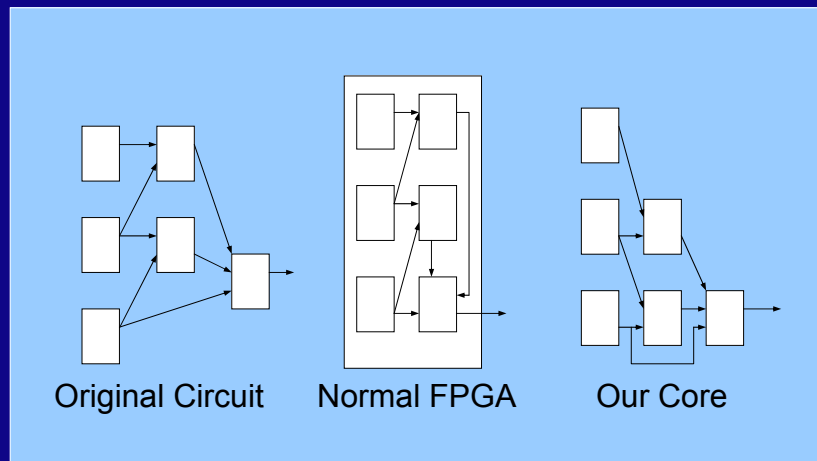
Original Circuit

Normal FPGA

## Non-Rectangular Cores:



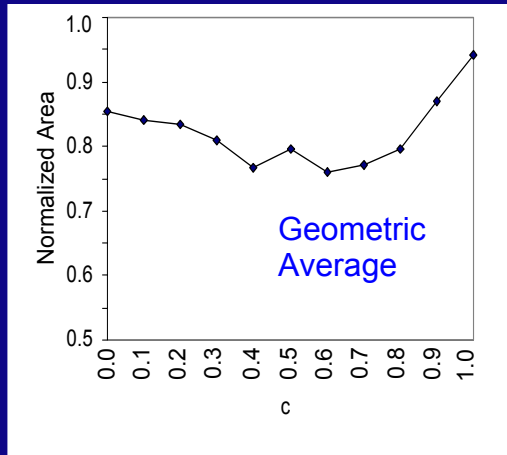
## Non-Rectangular Cores:



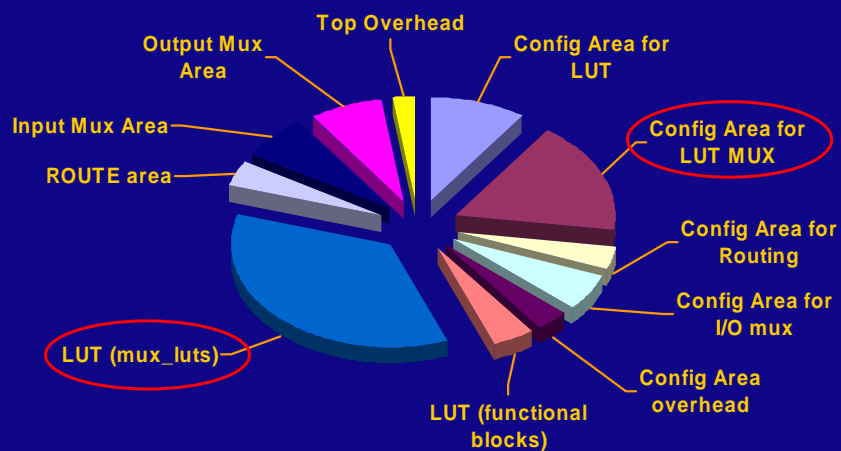
**Remember:** Since we are synthesizing these cores with standard cells, the actual layout will not be triangular

## Non-Rectangular Cores:

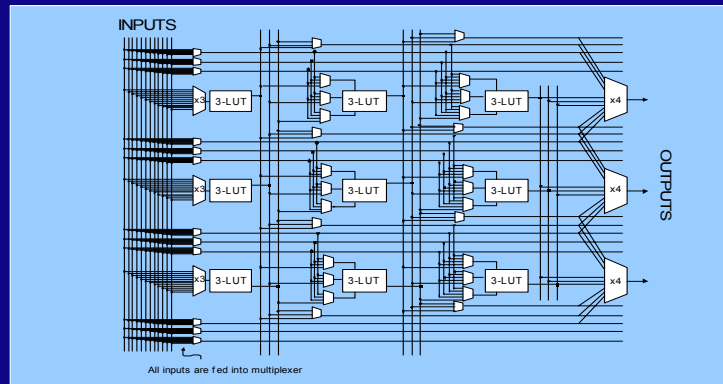
- As  $c$  increases, cores are more triangular => less area, but eventually core size increases and area increases again
- Using a  $c$  value of 0.6 results in 11% area savings, on average



## Soft-PLC Area Distribution

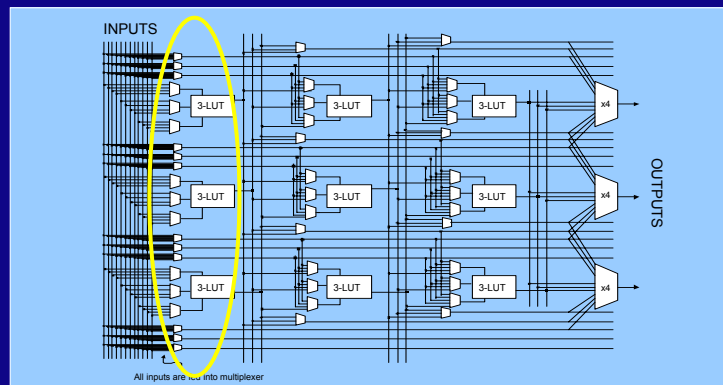


## Further Gradual Architecture Optimizations:



- Result:
  - 8.9% reduction in average area

## Further Gradual Architecture Optimizations:



- Result:
  - 12% increase in average area



## *Future Work*

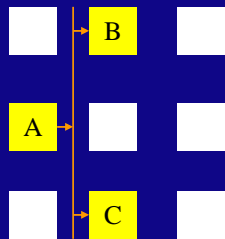
---

- Investigate speed of our core
- Investigate power implications of our core
- Add new cells to the standard cell library

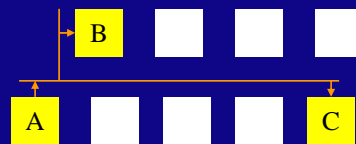
## *Future Work*

---

**Back Annotation of Delays:** After physical layout, give actual wire delay information to VPR for accurate delay driven placements



VPR assumes equal  
wire lengths  
 $A \rightarrow B = A \rightarrow C$



After physical layout  
actual wire lengths vary  
 $A \rightarrow B \ll A \rightarrow C$

## *Summary*

---

Soft Cores are viable!

Compared to a hard-core, 6.4 x less dense

Our Gradual Soft Core Architecture is 19% more dense than Directional Architecture

Our Gradual Soft Core Architecture is 25% more dense than Segmented Architecture

We've built a real chip (it has been fabricated and is now being tested)

## *For more details...*

---

- Paper:

N. Kafafi, K. Bozman, S.J.E. Wilton, "Architectures and Algorithms for Synthesizable Embedded Programmable Logic Cores", in the ACM International Symposium on Field-Programmable Gate Arrays, Feb 2003.

- Patent:

S.J.E Wilton, K. Bozman, N. Kafafi, J. Wu, "Method For Constructing An Integrated Circuit Device Having Fixed And Programmable Logic Portions And Programmable Logic Architecture For Use Therewith", U.S. Patent, submitted August 2003.

Licensed by Altera Corporation.