

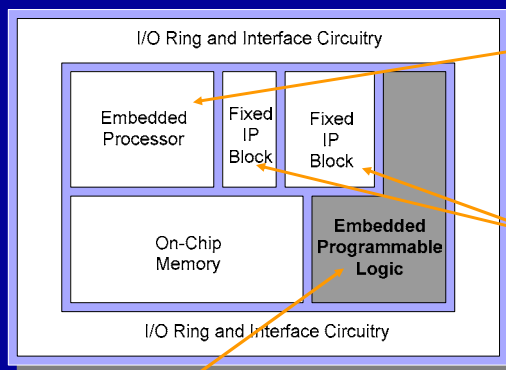


## Product-Term Based Synthesizable Embedded Programmable Logic Cores

Andy Yan,  
Dr. Steven Wilton

*SoC Research Lab, University of British Columbia  
Vancouver, BC Canada*

### Programmable IP in SoC Design



#### Processor:

- Functionality specified using software

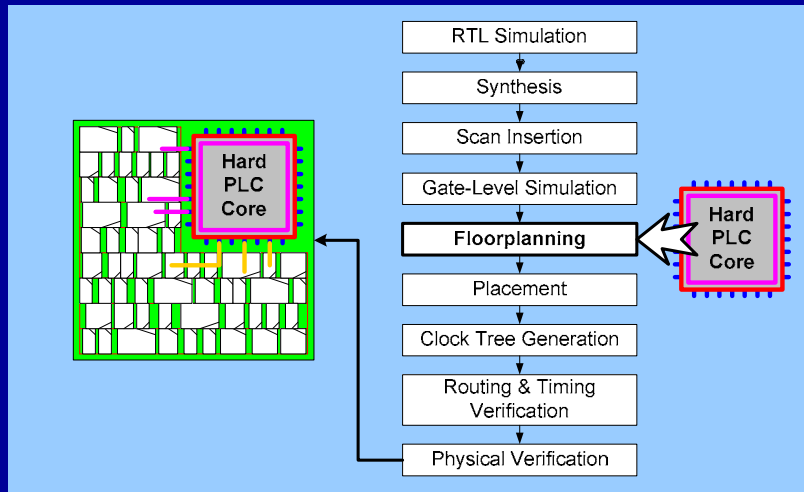
#### Fixed Logic:

- Functionality fixed at design time
- Little post-fab flexibility

#### Embedded Programmable Logic:

- Functionality specified through hardware configuration

## “Hard” Programmable IP Flow



3

## Soft Programmable Logic Cores

- **Conventional approach**
  - “Hard” FPGA layout provided by vendors
- **Our approach**
  - Synthesizable Programmable Logic Core (PLC)
  - “Soft”: HDL used to describe a PLC architecture, NOT to describe a particular user circuit
  - Synthesis required to translate RTL to gates

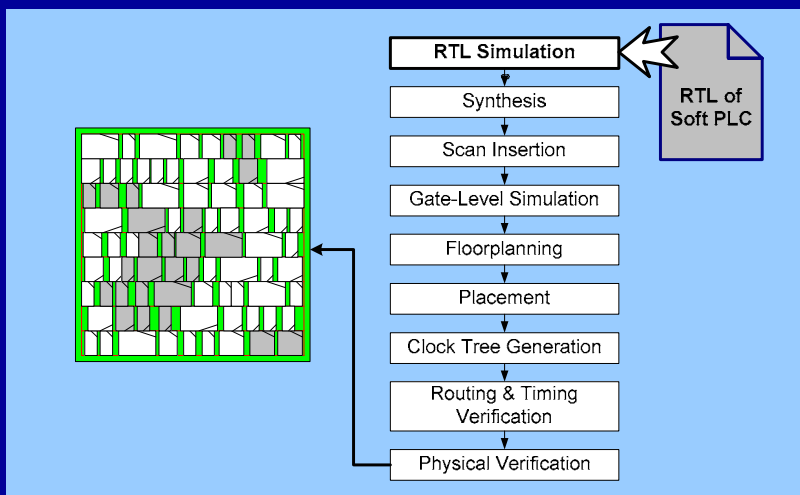
4

## Soft Programmable Logic Cores

- **Advantages**
  - Easy to integrate, reduces design time
  - Very flexible, can create the exact required core
  - Easy to migrate to smaller technologies
- **Disadvantages**
  - Inefficient compared to hard cores
- **Our thought**
  - Makes sense if you only want a small core (a few hundred gates, perhaps)  
e.g. next state logic in state machine

5

## “Soft” Programmable IP Flow



6

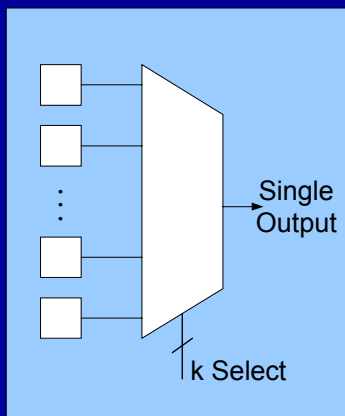
## Our Contribution

- **Previous architecture design**
  - Gradual Architecture [Kafafi et al., FPGA '03]
  - Basic logic element: Lookup-tables (LUT)
- **Propose new architectural family**
  - Basic logic element: Product-term array block
  - 35% density improvement
  - 72% speed improvement

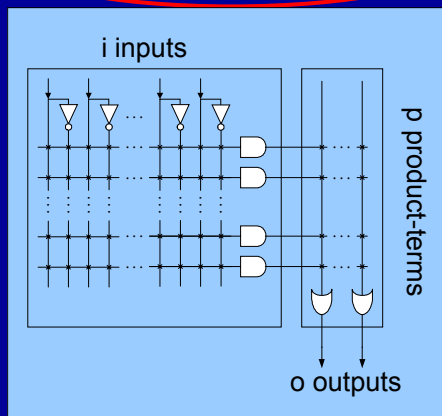
7

## Basic Logic Elements

Lookup-Table (LUT)



Product-term Block (PTB)



8

## Architectural Requirements

- **Area and delay minimization**
  - Large capacity product-term blocks (PTB) and shallow core depth
- **Simple placement and routing**
  - “Full connectivity” routing fabric
- **Flexible and scalable architecture**
  - Architecture parameter definitions and optimizations

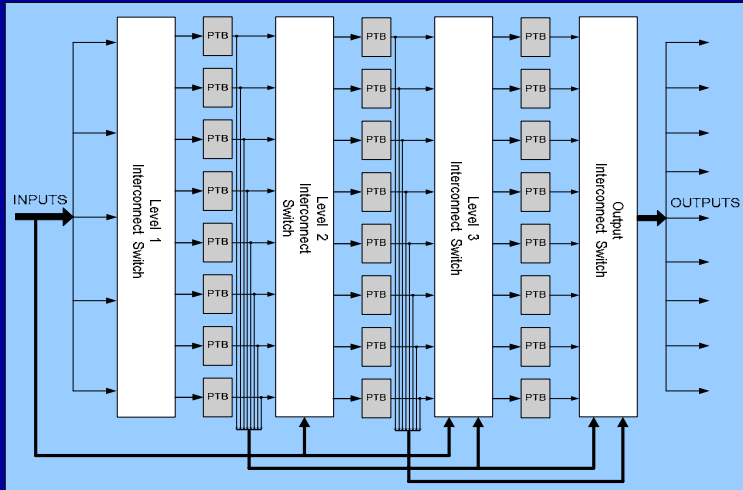
9

## Synthesizable PTB Architecture

- Product-term blocks (PTBs) arranged in several levels
- Unidirectional signal flow to avoid combinational loops in un-programmed fabric
- Outputs of PTBs in one level can only be connected to inputs in subsequent levels
- 2 interconnect strategies:
  - Rectangular and Triangular PTB architecture

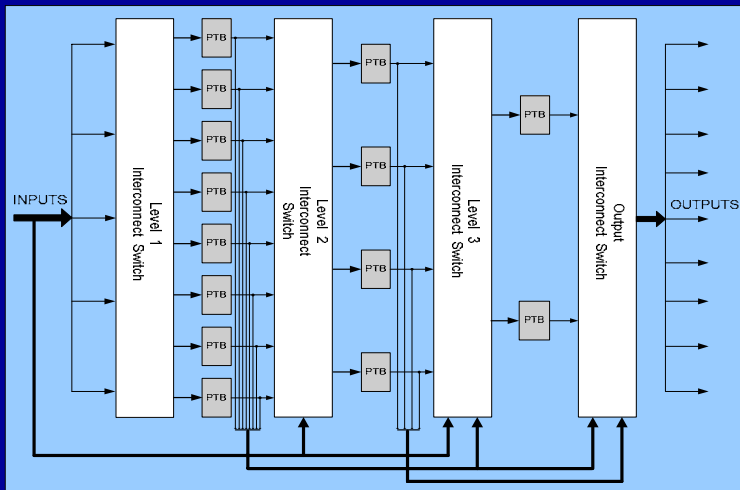
10

## Rectangular PTB Architecture



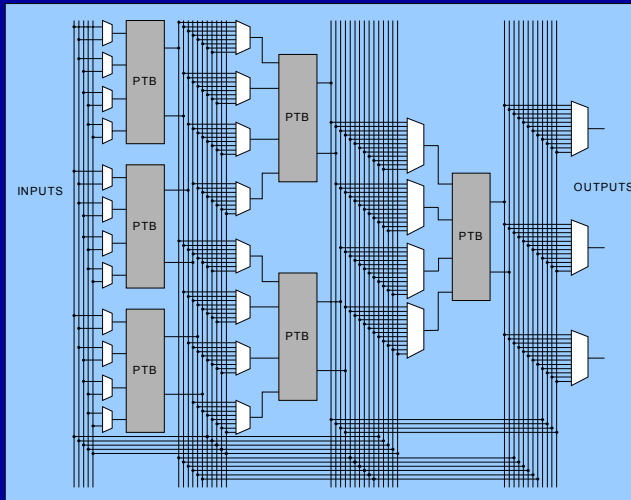
11

## Triangular PTB Architecture



12

## Detailed View of Interconnect Fabric



- Very flexible, not restrictive
- Easy P&R tools
- We can do better though

13

## Parameter Optimization

### Two Parameter Classes:

- **High-Level Parameters**
  - Specified by SoC core user / VLSI designer
  - Used to identify a specific core in a programmable library
- **Low-Level Parameters**
  - Not specified by SoC core user / VLSI designer
  - Used to describe specific characteristics of library
  - Determined through architectural experimentation

14

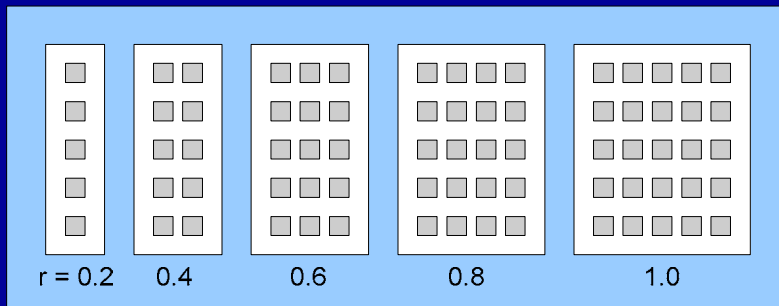
## Architectural Parameters

High-Level Parameters	Low-Level Parameters
Number of Primary Inputs Pins	Number of Inputs per PTB ( $l$ )
Number of Primary Output Pins	Number of Product-terms per PTB ( $p$ )
Number of Product-term blocks (PTBs)	Number of Outputs per PTB ( $o$ )
	Ratio of PTBs in Neighboring Levels ( $r, \alpha$ )

15

## Low-Level Parameter Optimization

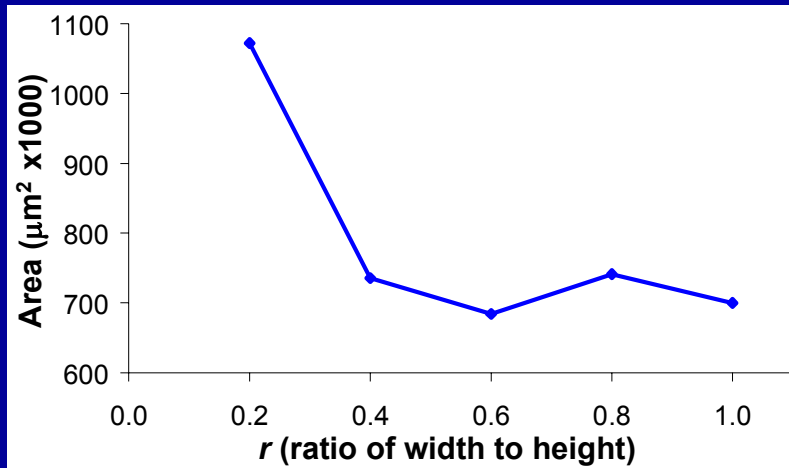
### Rectangular PTB Architecture



$r$  = ratio of width to height

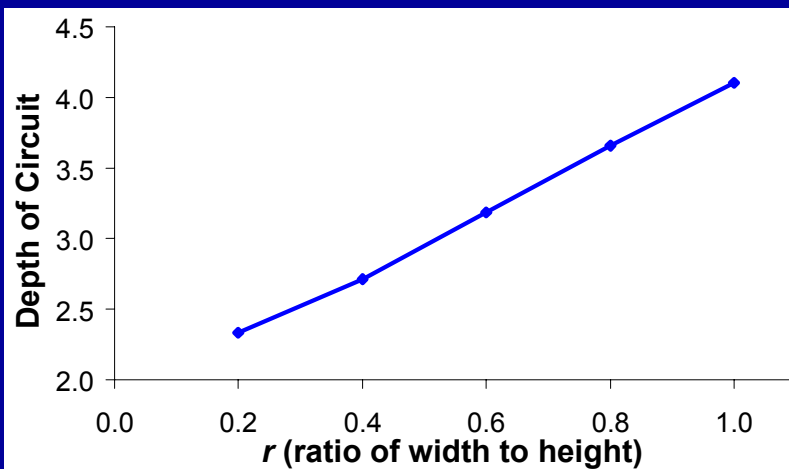
16

## Rectangular PTB Architecture Area



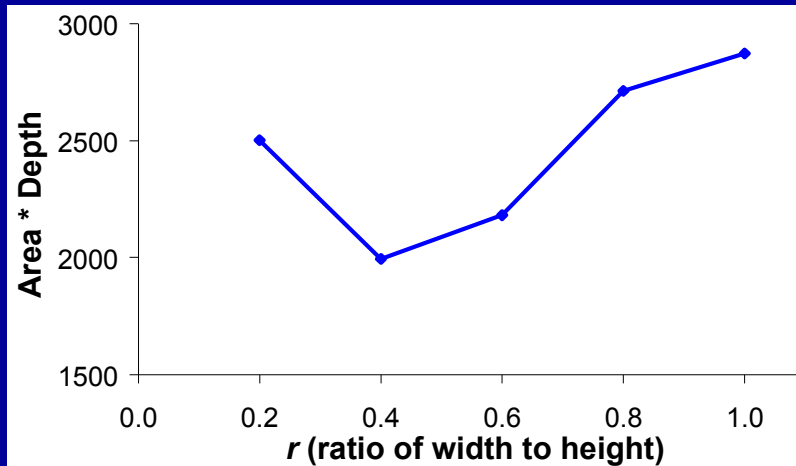
17

## Rectangular PTB Architecture Depth



18

## Rectangular PTB Architecture



19

## Low-Level Parameter Optimization

### Triangular PTB Architecture

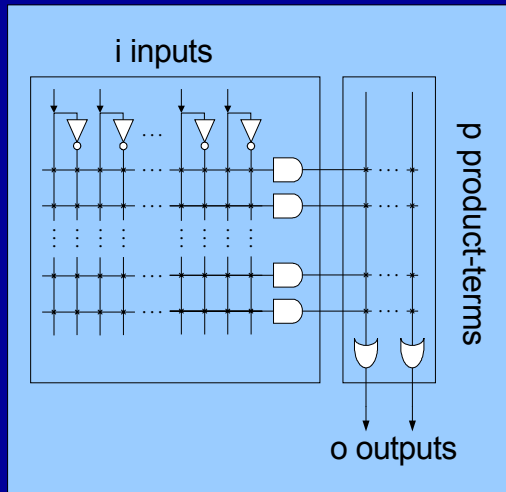


$\alpha$  = number of PTB drop-off factor

- 19% improvement in area-delay

20

## Other Low-Level Parameters



### Product-Term Block Parameters:

- input  $i = 12$
- product-term  $p = 9$  or  $18$  depending on size of circuit
- output  $o = 3$

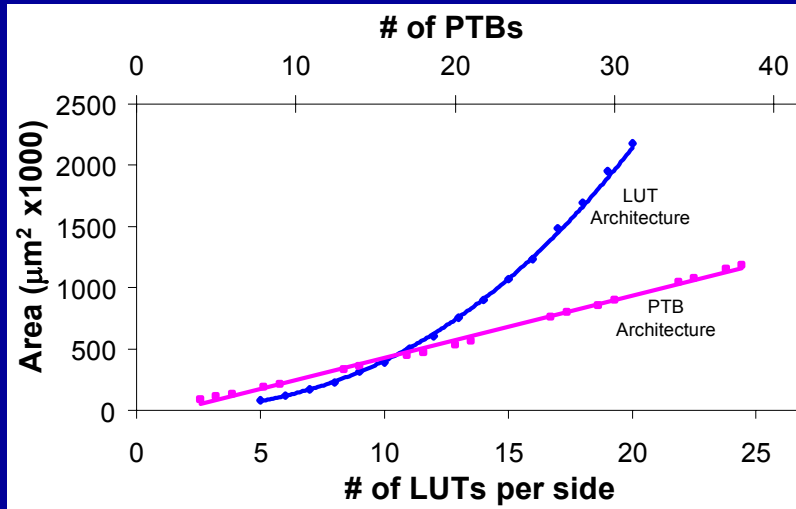
21

## Comparison to LUT-based Architecture

- 35% area improvement, 72% delay improvement
- Gains mainly from larger circuits (more than 50 equivalent 4-LUTs)
- **Factors:**
  - PTB-based architecture has larger and fewer logic blocks
  - PTB-based architecture routing fabric simpler and depth of core shallower

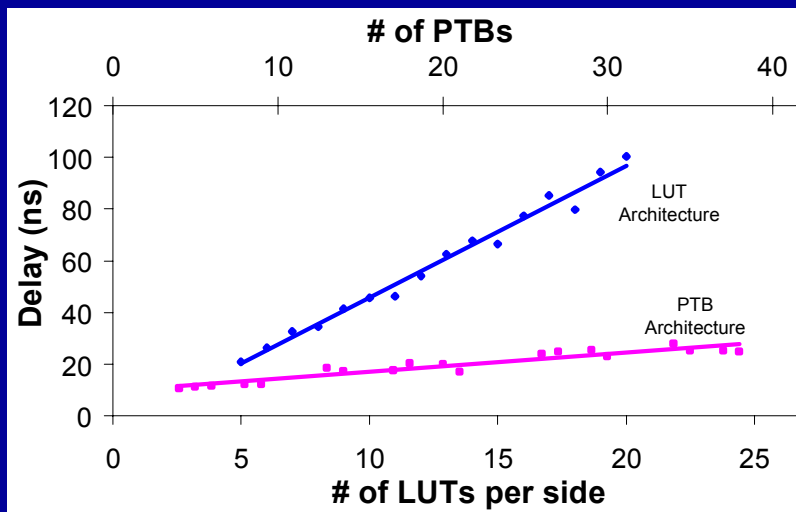
22

## Comparison to LUT-based Architecture



23

## Comparison to LUT-based Architecture



24

## **Summary**

---

- Presented a product-term based synthesizable programmable logic device
- Investigated effects of various architectural parameters
- Optimal product-term block (PTB) parameters:
  - input  $i = 10$
  - product-term  $p = 9$  or  $18$   
depending on size of circuit
  - output  $o = 3$

25

## **Summary**

---

- Compared product-term architecture to lookup-table based device
- Overall, 35% smaller and 72% faster
- Primarily due to reduction in amount of circuitry needed to route signals

26