# A Synthesizable Datapath-Oriented Embedded FPGA Fabric

Steven J.E. Wilton[1], Chun Hok Ho[2], Philip Leong[3], Wayne Luk[2], Brad Quinton[1]
[1]University of British Columbia
[2]Imperial College London
[3]Chinese University of Hong Kong

(This work was performed at Imperial College London)

## What this talk is about

A new FPGA Fabric that is:

Embedded: Embed this in an ASIC, not part of a stand-alone FPGA

Synthesizable: can be synthesized using normal ASIC tools and implemented in standard cells

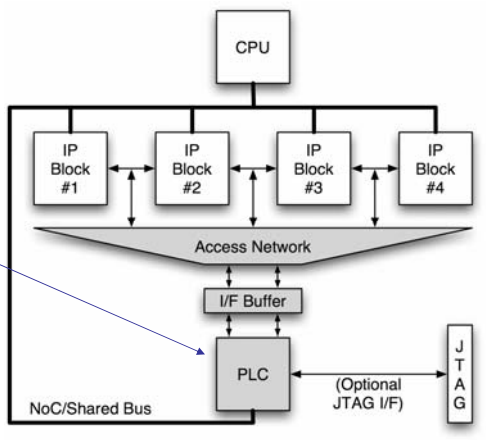
Datapath-Oriented: focus on bus-based (numeric) applications

## Motivation: Embedded Debug

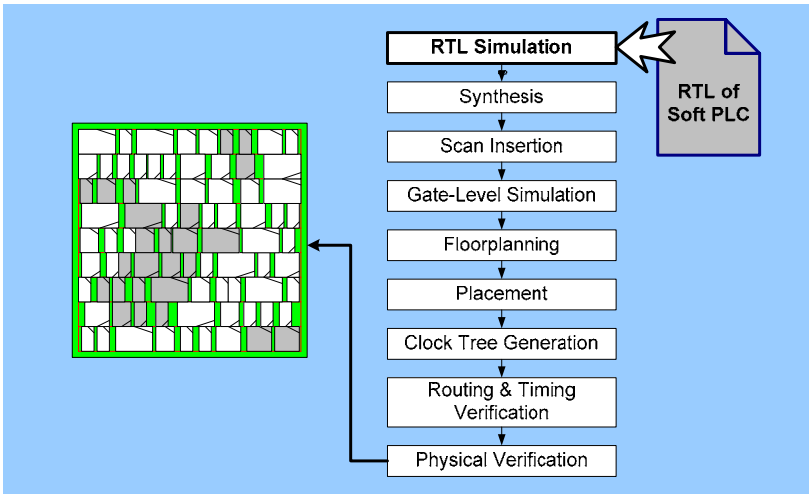
Embed a small amount of programmable logic onto an ASIC

- Use this logic to observe and/or control internal signals
- Perform simple data collection/monitoring operations

This talk: Architecture of this block




## Synthesizable "Soft" FPGA Cores



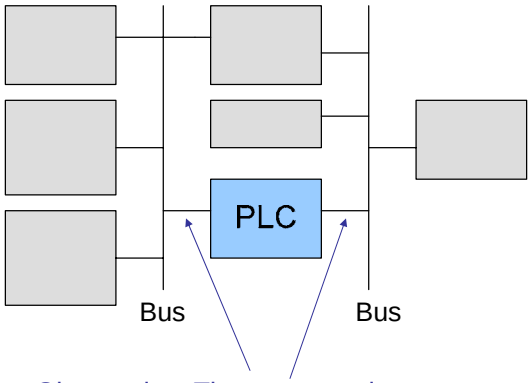
## Implications of Being Synthesizable

Observation 1: To make it truly synthesizable, must avoid combinational loops in the unprogrammed fabric

Observation 2: Each tile need not be identical

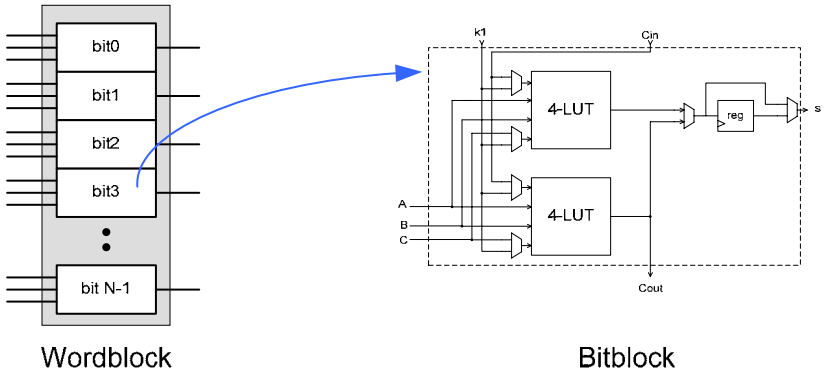
## Implications of being datapath-oriented:

Use it when the PLC is connected to a bus:




Observation: These connections are permanently tied to the bus signals, and we know this when the ASIC is designed

## Logic Architecture

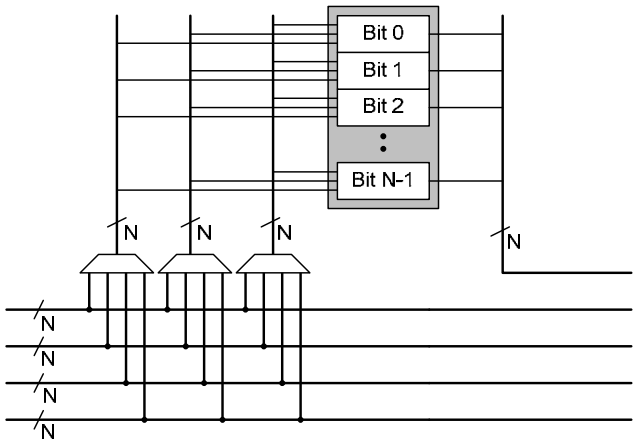



Wordblock Bitblock

Key point:

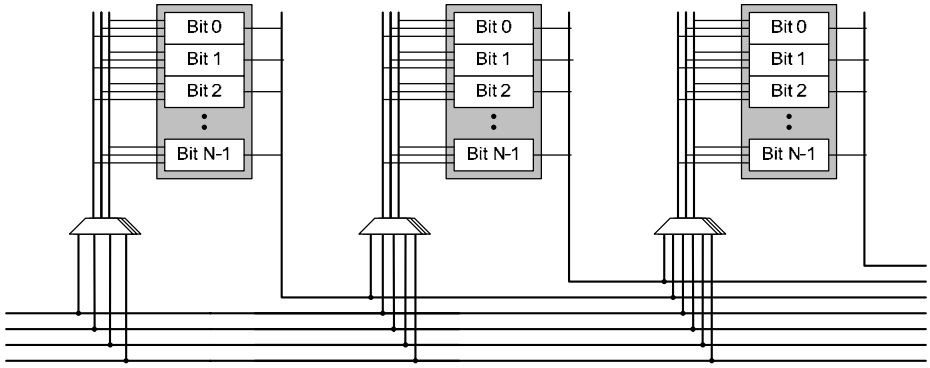
- All bitblocks within a wordblock share same set of configuration bits
- Means all bitblocks implement the same function

## Routing Architecture




Key point: Signals are routed as buses

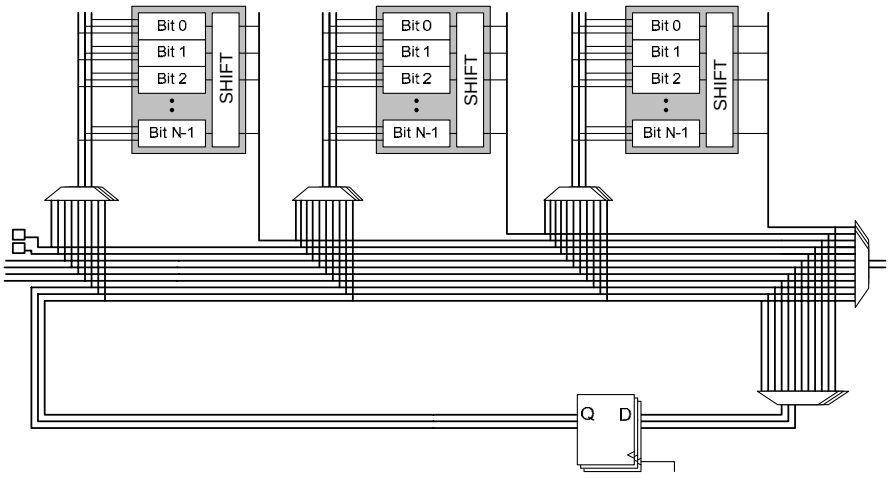
## Routing Architecture



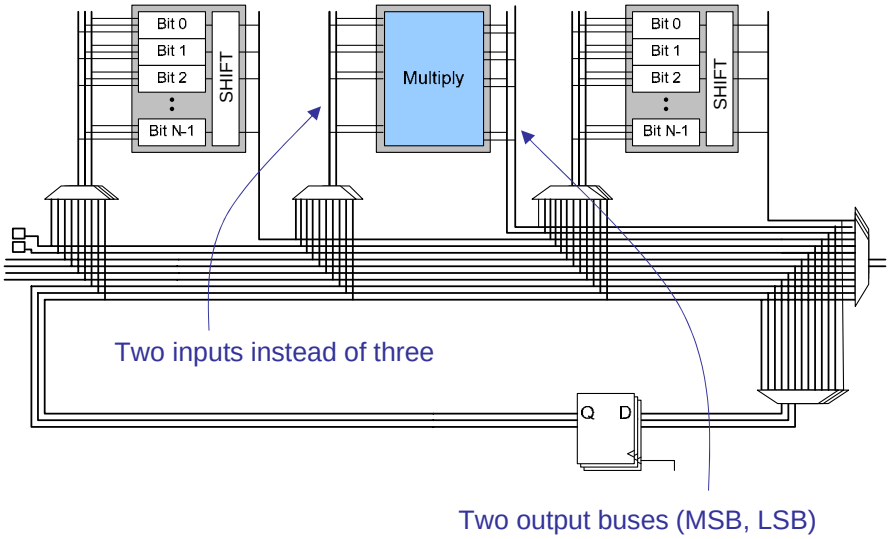

Key point: - Linear array of wordblocks

- Number of buses goes up as we go to the right

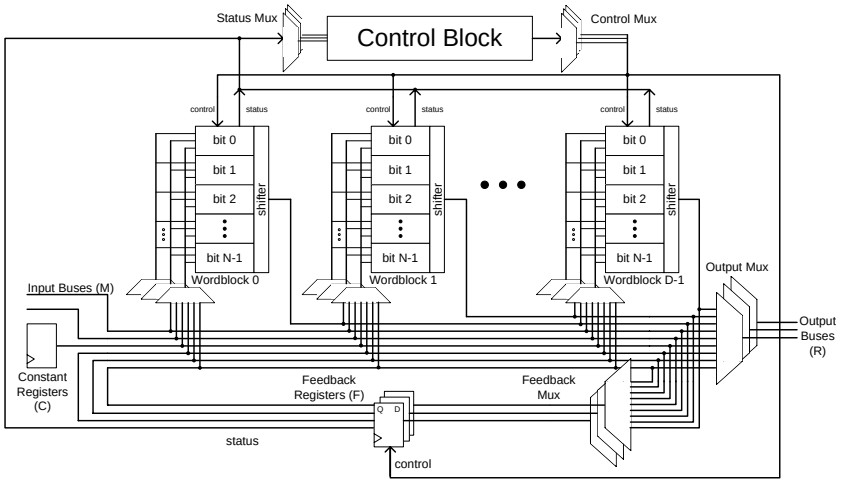
## Datapath Architecture



# Multipliers



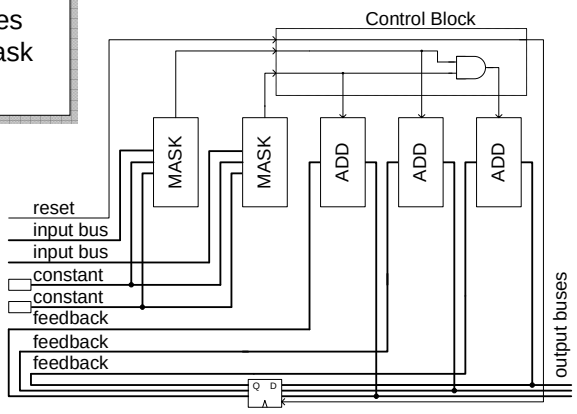

# Add a Control Block




Control block is based on P-term fine-grained synthesizable core

## Example Mapping

Monitor two buses:
- Count the number of times each bus matches a mask
  - includes don't care bits
- Count the number of times both buses match the mask at the same time



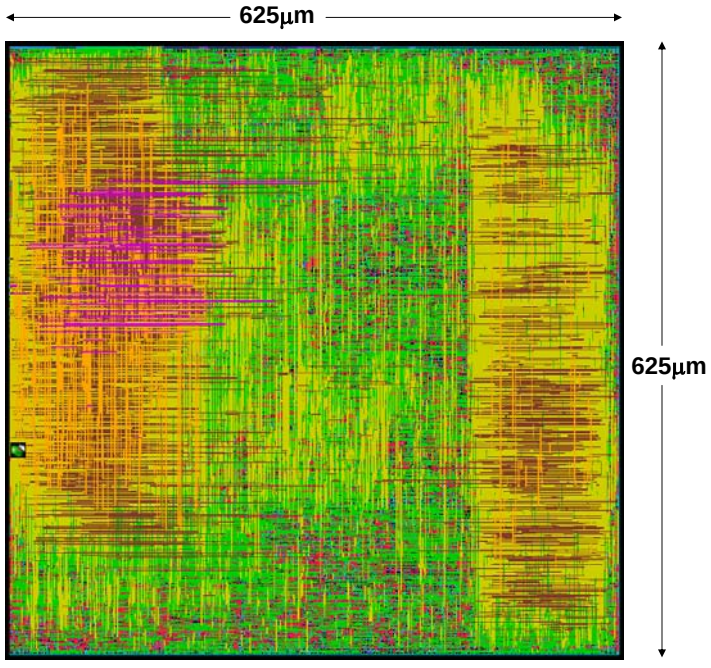

Interesting Questions:

1. How do the various architectural parameters affect density?
2. How does this compare to a fine-grained architecture?

| Bench-Mark | Datapath (ours) | Fined-Grain (PTerm) | ASIC | Fine-Grain/ Datapath | Datapath/ ASIC |
|---|---|---|---|---|---|
| fbly | 332,091 | 132,339,335 | 9,300 | 399 | 35.7 |
| dotv3 | 225,518 | 65,534,780 | 6,575 | 291 | 34.3 |
| dscg | 325,029 | 116,271,968 | 9,473 | 358 | 34.3 |
| fir4 | 307,154 | 130,971,120 | 9,843 | 426 | 31.2 |
| egcd | 3,778,611 | 22,776,474 | 10,420 | 6.02 | 363 |
| momul | 486,654 | 11,448,589 | 7,097 | 23.5 | 68.5 |
| median | 194,654 | 10,733,962 | 4,420 | 55.1 | 44 |
| debug1 | 119,286 | 1,302,928 | 3,484 | 10.9 | 34 |

Key result 1: Significantly better than fine-grained architecture

Key result 2: Overhead roughly the same as FPGA/ASIC



## Conclusions

Our architecture is 6 to 426 x more efficient than fine-grained architecture

But, this is only for datapath-oriented circuits.

However, this is ok:

- In an SoC, we know, when the chip is designed, whether the inputs are buses or bits
- If there are buses, use this architecture
- If there are not buses, use a fine-grained architecture

Final thought: using this architecture, the overhead is similar to that of a normal FPGA. People already accept this!