Case for Fast FPGA Compilation using Partial Reconfiguration

Dongjoon(DJ) Park, Yuanlong Xiao, Nevo Magnezi, and André DeHon

Implementation of Computation Group

University of Pennsylvania

Introduction

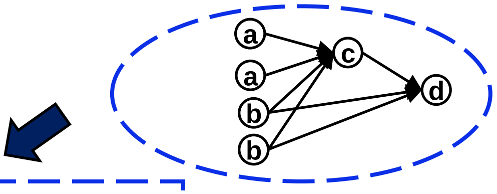
- **Problem**: FPGA’s long compilation

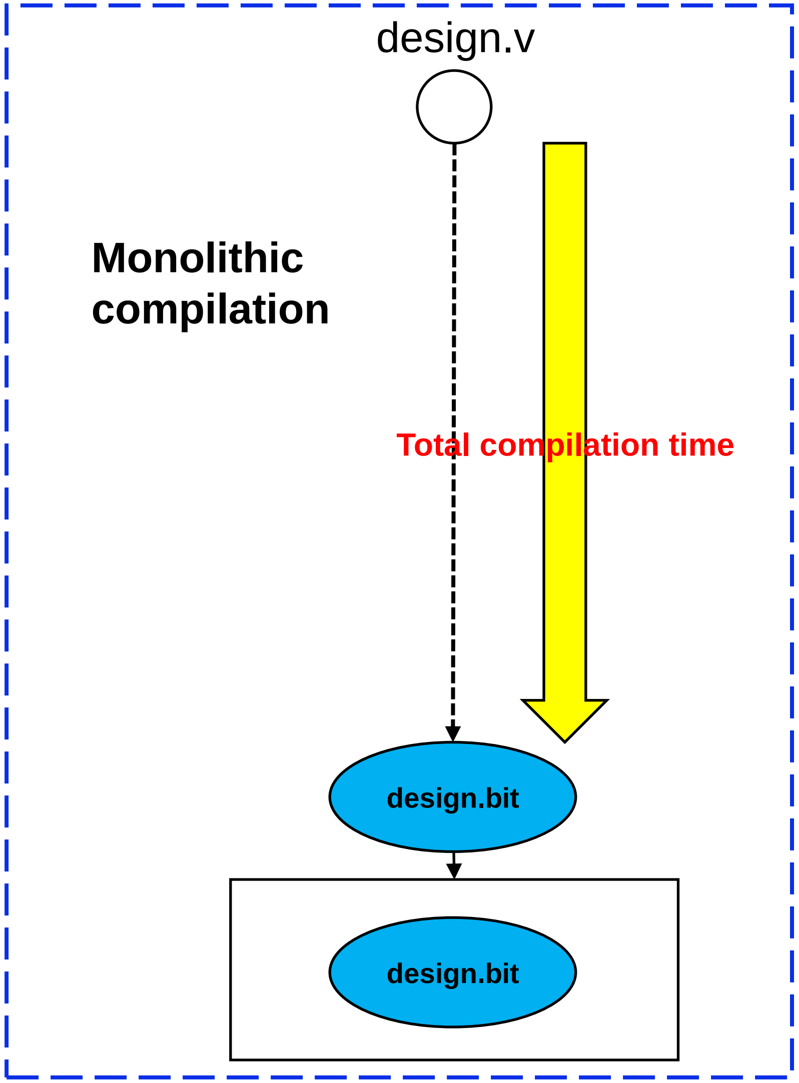
- **Methodology**: Divide-and-Conquer
  - Compile design blocks **in parallel**
  - **Partial Reconfiguration (PR)** for separate compilations
  - Connect design blocks through an **overlay network**

Overlay Network

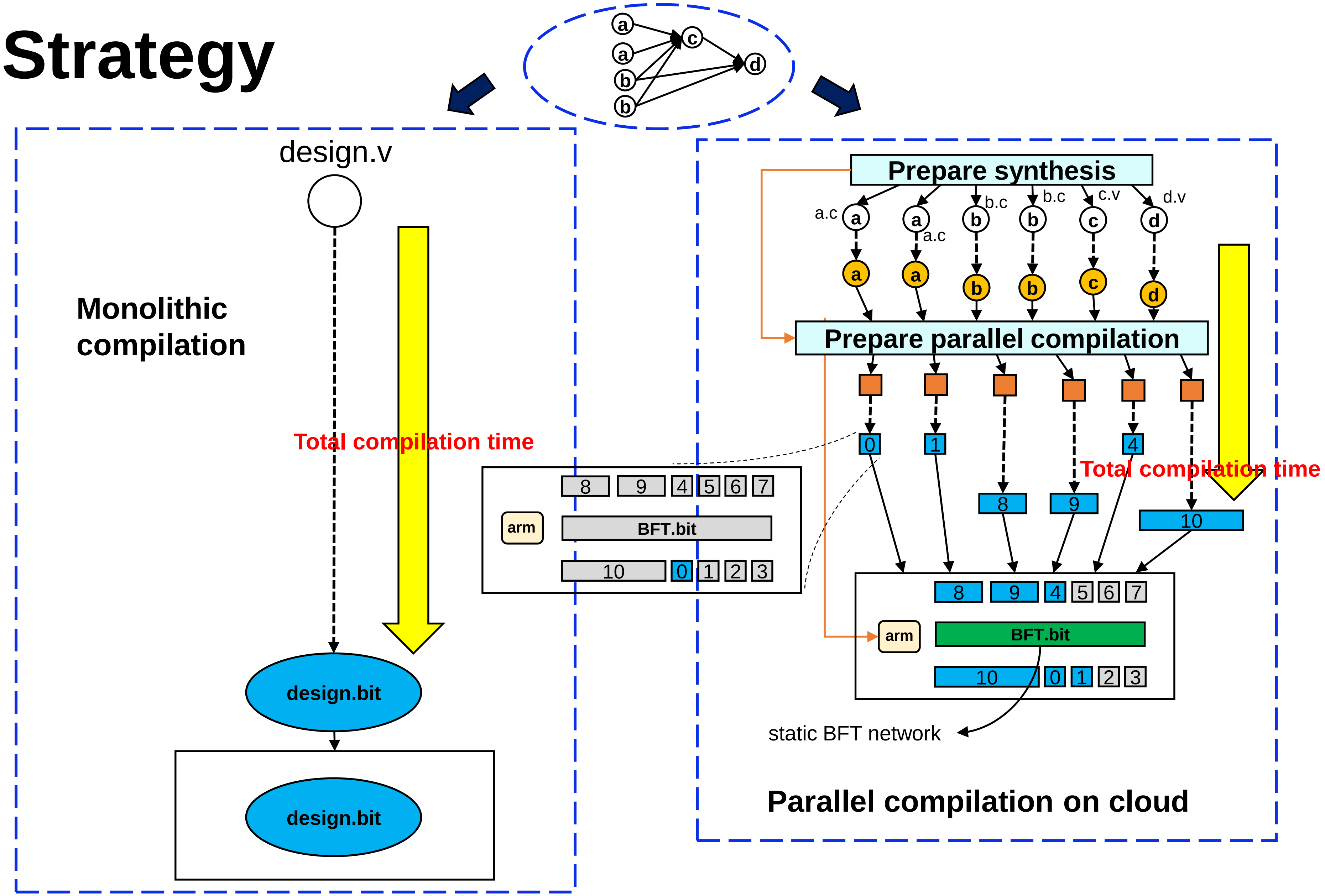
- Overlay network:
  Packet-switched Butterfly Fat Tree (BFT) network as a static design of PR
  - Support arbitrary connectivity among separately-compiled components
  - Fixed and pre-computed
    - not contribute to user-design mapping time
  - Packet switched
    - no need to configure overlay network

Strategy


a
a
b
b
c
d


design.v
Monolithic compilation
Total compilation time
design.bit
design.bit

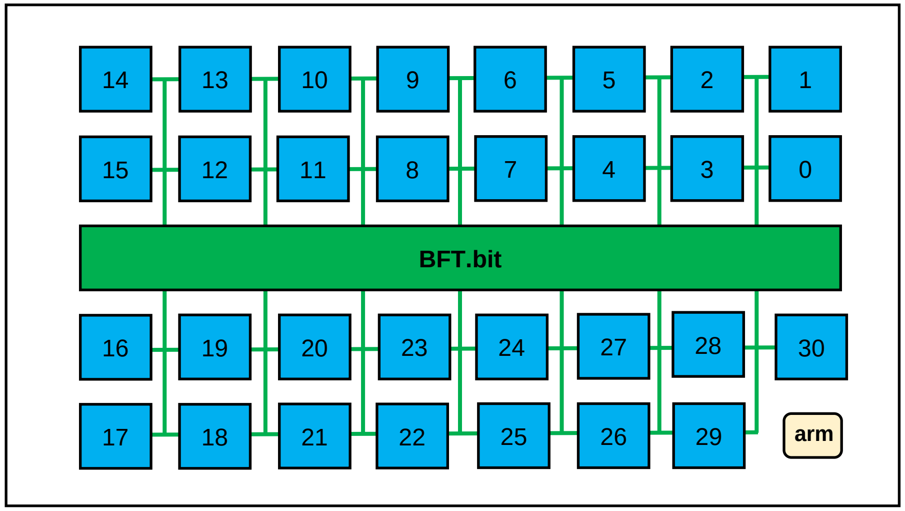
Strategy

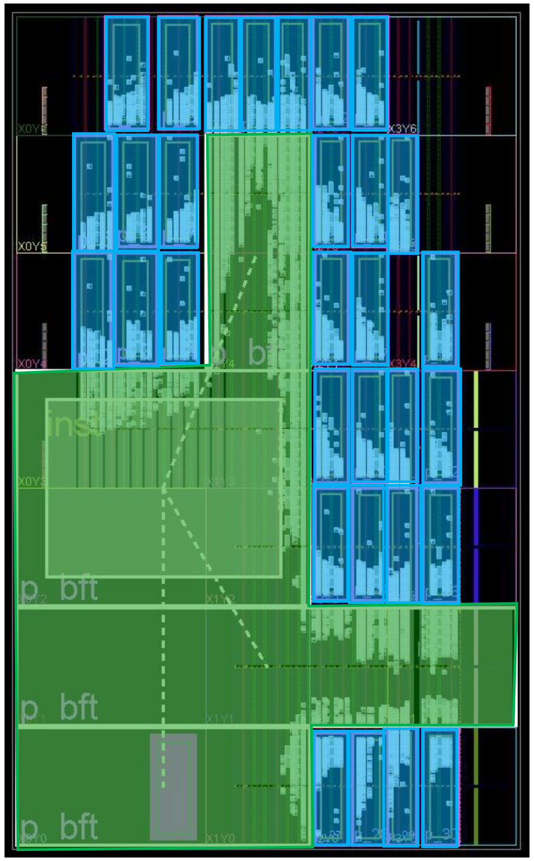


Penn University of Pennsylvania

Case study: multi-core design

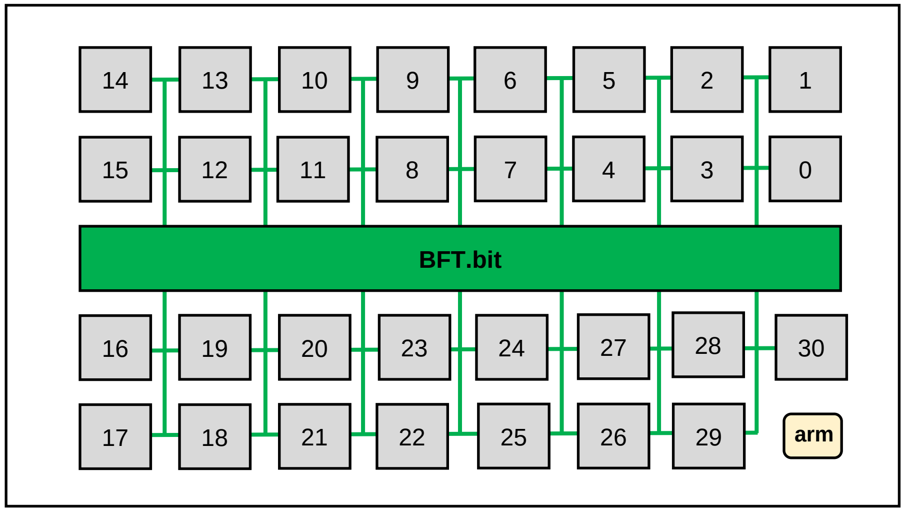
- An array of soft-cores (MicroBlaze processors)

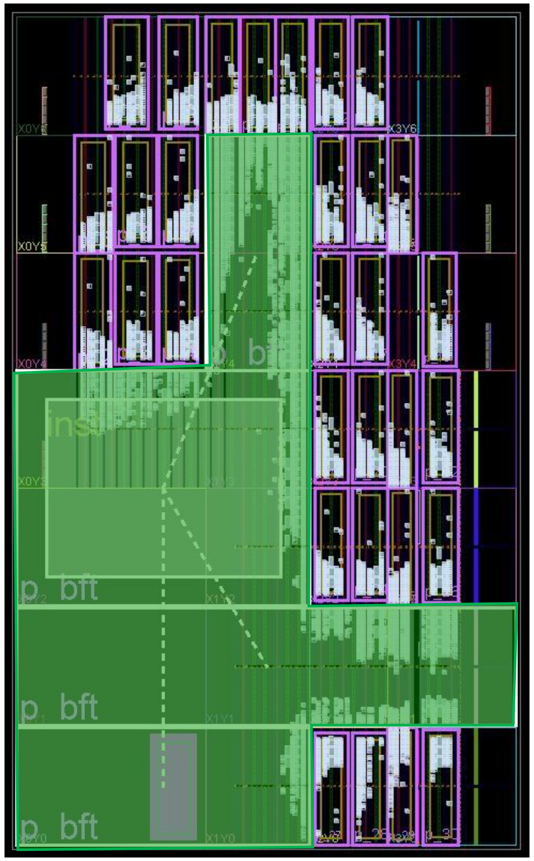




Case study: multi-core design

- An array of soft-cores (MicroBlaze processors)

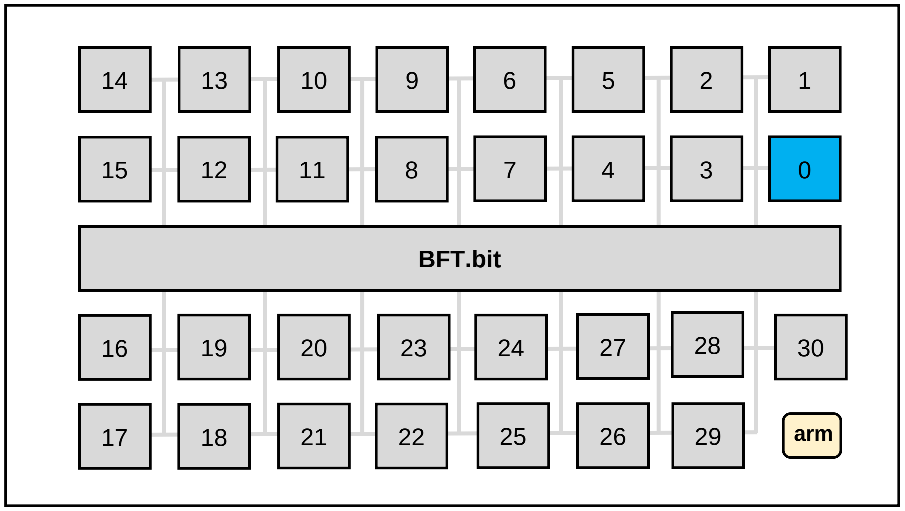


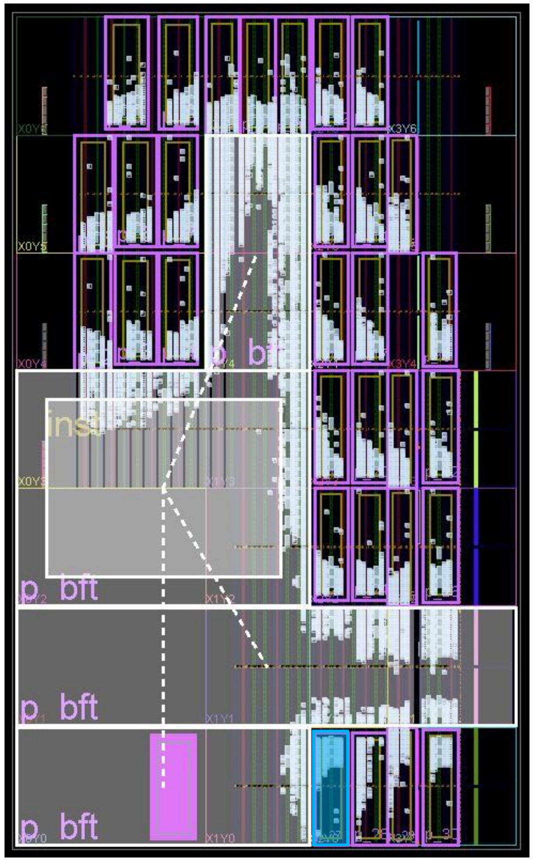


Penn
University of Pennsylvania

Case study: multi-core design

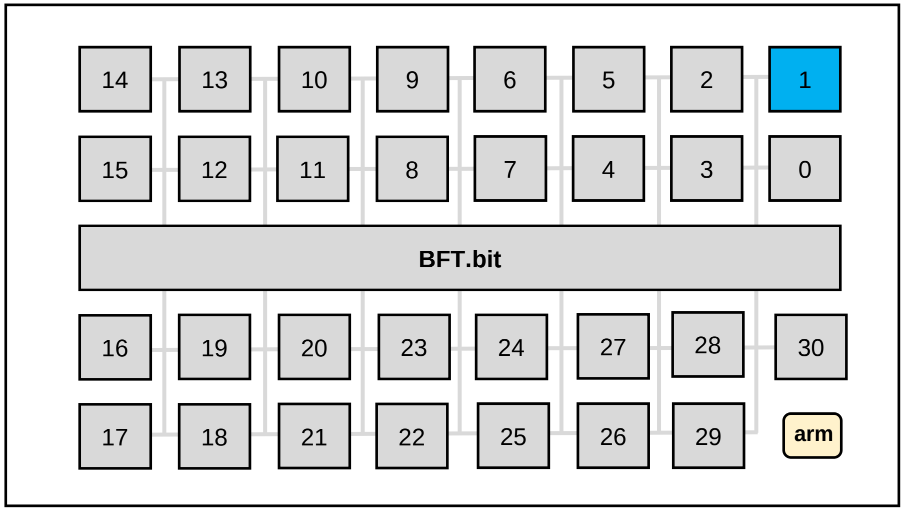
- An array of soft-cores (MicroBlaze processors)

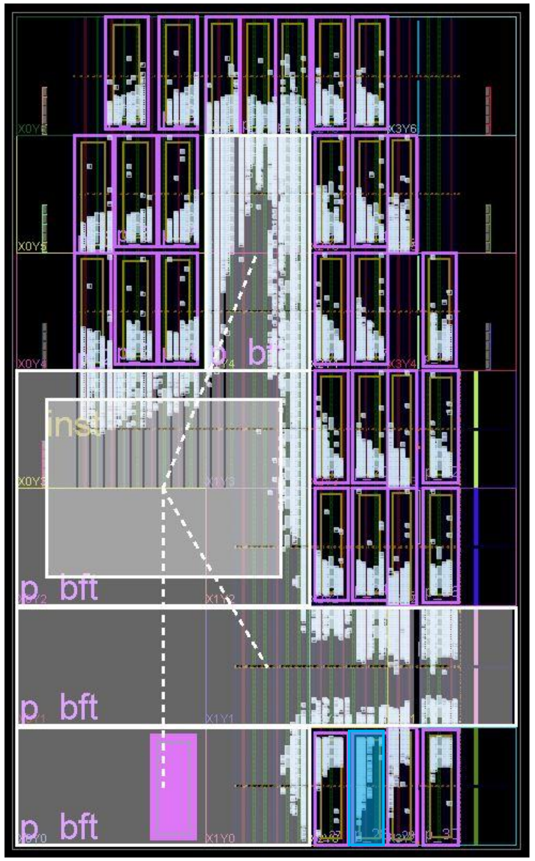




Case study: multi-core design

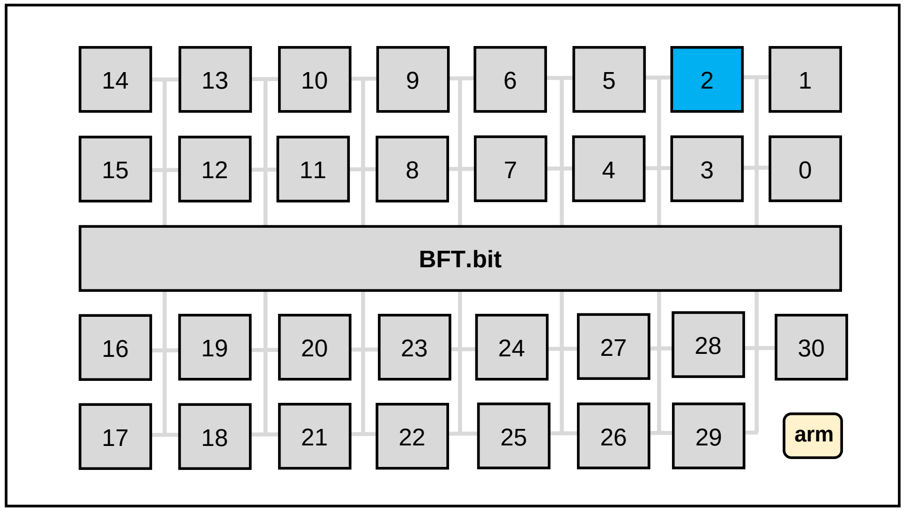
- An array of soft-cores (MicroBlaze processors)

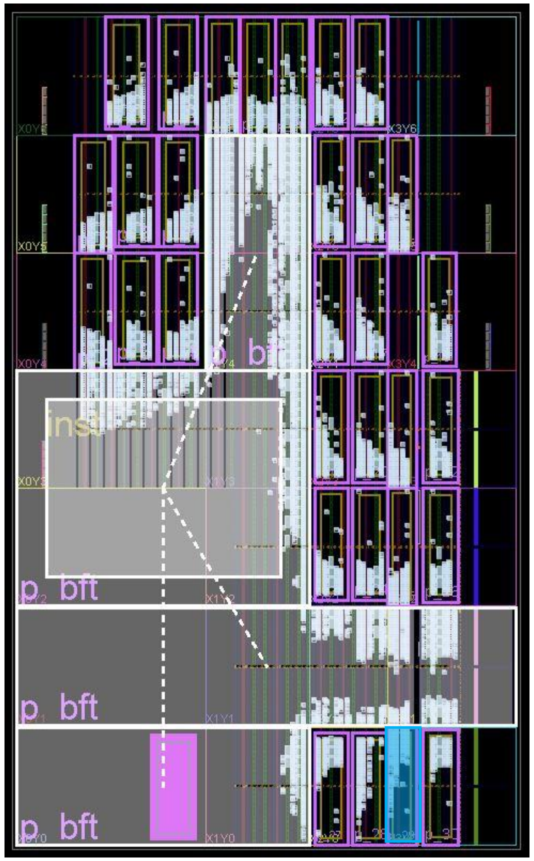




Case study: multi-core design

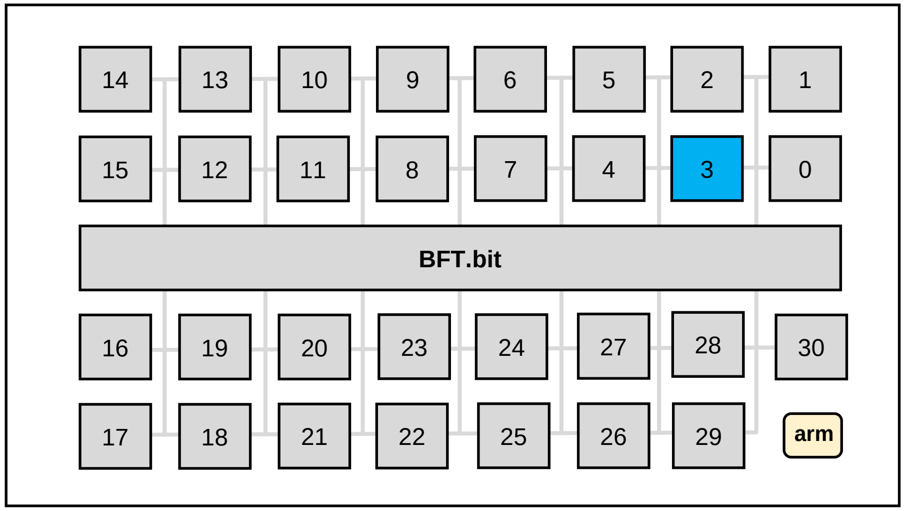
- An array of soft-cores (MicroBlaze processors)

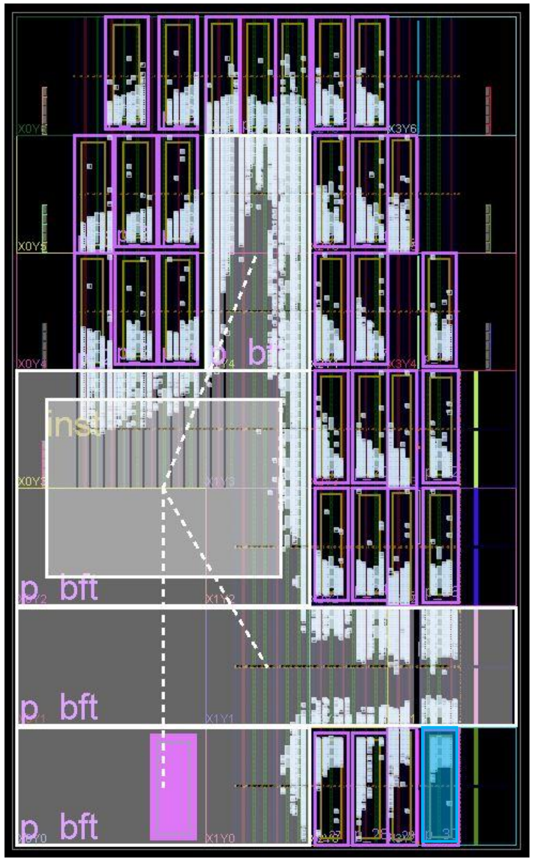




Case study: multi-core design

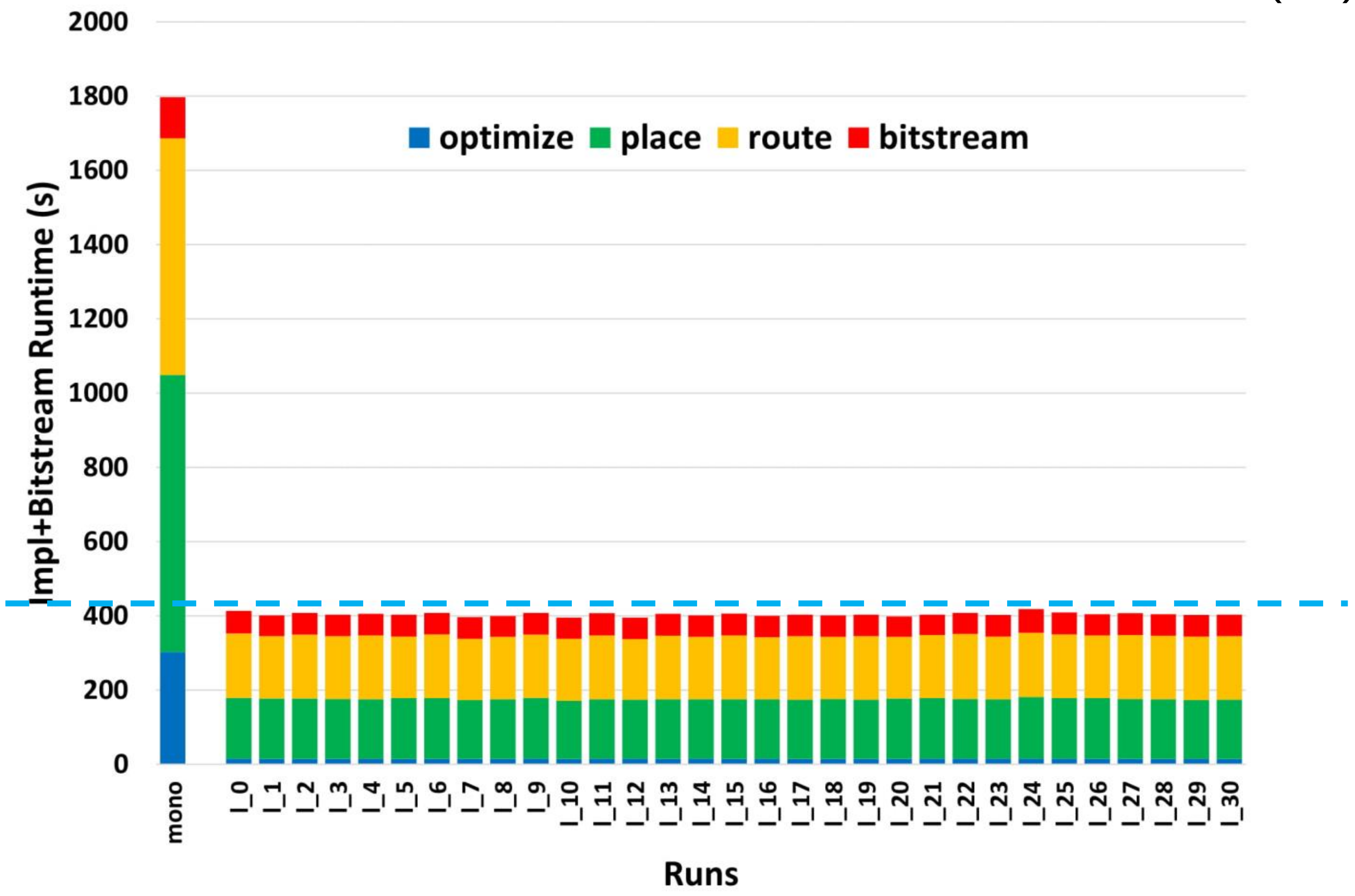
- An array of soft-cores (MicroBlaze processors)





Case study: multi-core design

microBlaze(v.0)



Case study: multi-core design

- Result

Runtime(secs)	microBlaze(v.0)		microBlaze(v.1)		microBlaze(v.2)	
	Mono	Parallel	Mono	Parallel	Mono	Parallel
Synthesis	3171	287	3118	283	2510	235
Impl+bitstream	**1797**	**418**	**1692**	**413**	**1283**	**398**

4.30x

4.10x

3.22x

Conclusion

- Showed **4x compile time speedup** using the current tool's existing facilities
- Work-in-progress
  - Tool(Xilinx Vivado) challenges
  - More automation in the flow and more benchmarks
  - Optimization in the overlay and PR architecture

Penn
UNIVERSITY of PENNSYLVANIA