PipeRench : Power and Performance Evaluation of a Programmable Pipelined Datapath

Benjamin A. Levine and Herman H. Schmit

Dept. of Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA USA 15213
blevine@cmu.edu, herman@ece.cmu.edu

Electrical & Computer ENGINEERING

Acknowledgements

Research Sponsors

- DARPA
- NSF
- Northrop Grumman Electronic Systems
- Pittsburgh Digital Greenhouse

CMU Faculty

Seth Goldstein

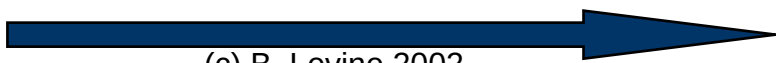
CMU Graduate Students

- Mihai Budiu
- Hari Cadambi
- Matt Moe
- R. Reed Taylor
- Andrew Tsai
- David Whelihan

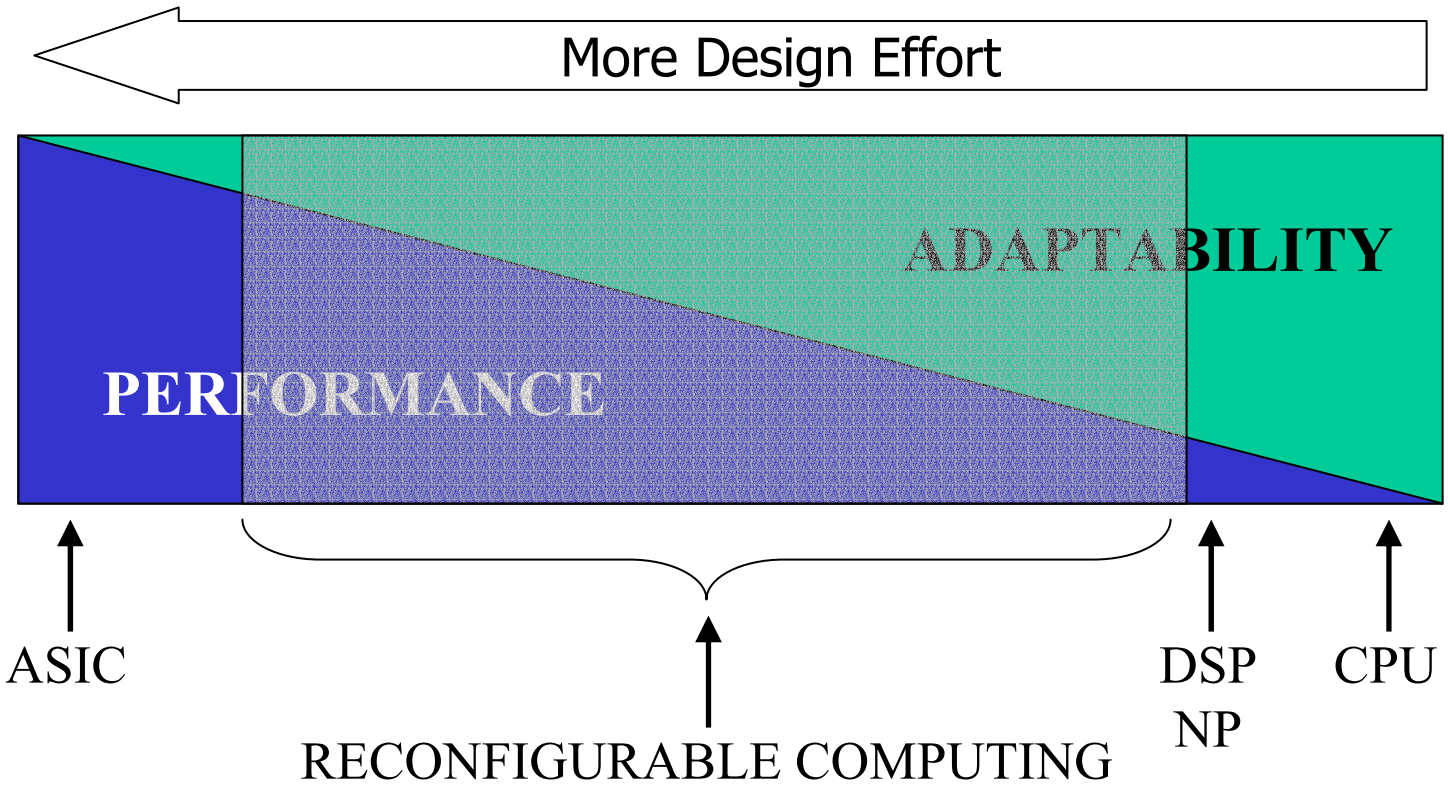
PipeRench **Is a Reconfigurable Computing Device**

Reconfigurable Computing Device = A Computing Device which can be reconfigured for each different application that it runs, by changing the functionality of its hardware and the way that its hardware is connected.

PipeRench **was developed by students and faculty at Carnegie Mellon.**



Why Reconfigurable Computing?



Why Reconfigurable Computing?

We want performance **and** adaptability:

- Performance of an ASIC

  Implement application as custom datapath to:
  - Increase parallelism.
  - Decrease memory traffic (through locality).
  - Increase performance.
  - Use less power.

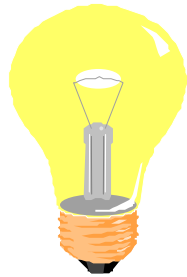
- Adaptability of a CPU.

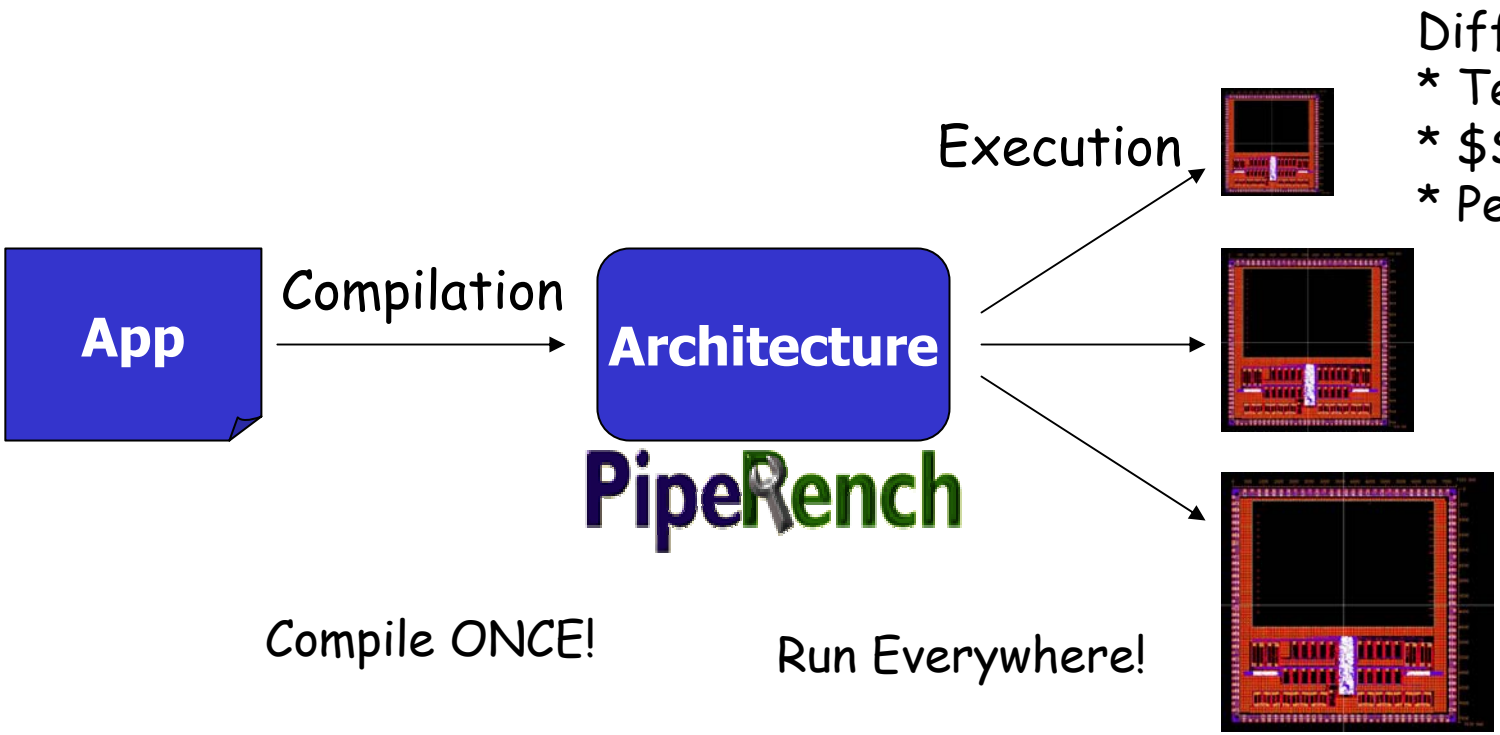
  Completely reprogram as needed for new applications.

Why NOT Reconfigurable Computing?

- FPGA design is more like HW than SW
  - No real C to FPGA yet, so must use HDL
- FPGA configuration is fixed to one FPGA
  - Must redesign to gain performance on larger FPGAs
  - Can't use design on FPGA with fewer resources.
  - Compares poorly to SW for microprocessors:
    - No portability
    - No scalability

Solution:
Use A Virtual Architecture



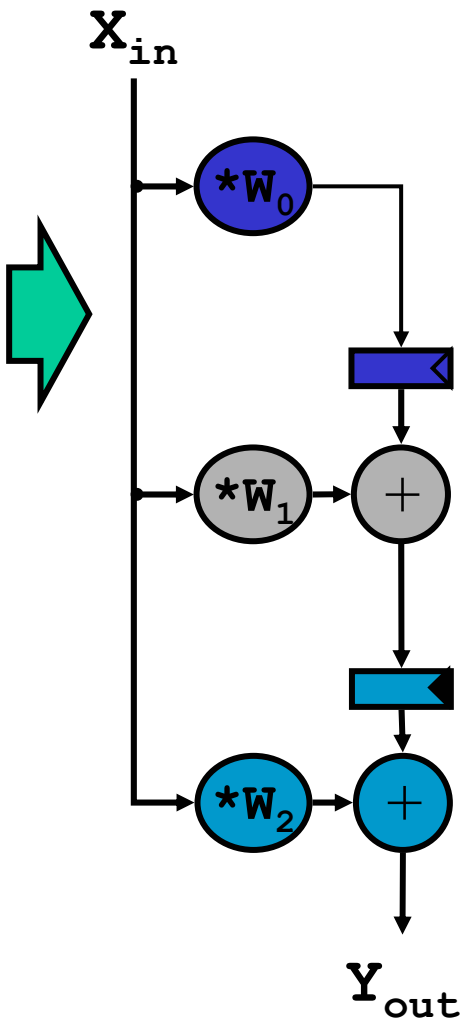


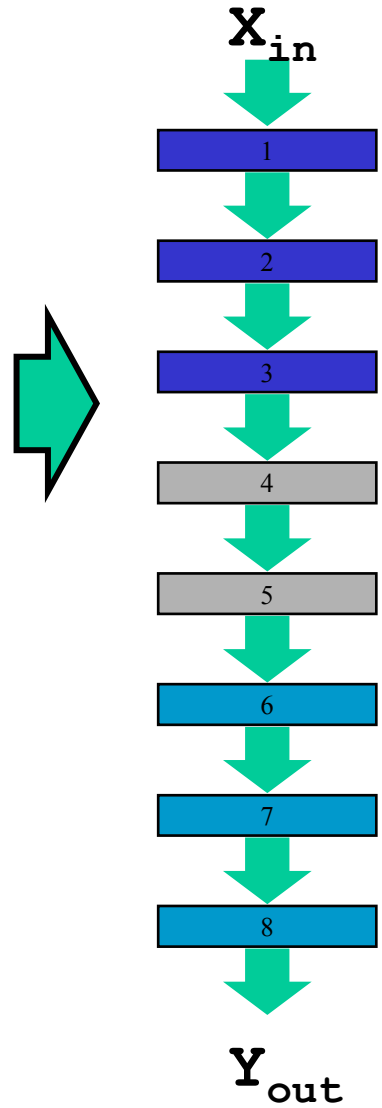
Virtual Architecture

- Compile to virtual machine
  - Makes compilation easier
  - Compile from high-level language (DIL)
  - Binaries decoupled from specific hardware
  - Scalable / Re-usable
- Restrict the model of computation to pipelined datapaths
  - Makes virtual architecture possible

Pipelined Datapaths

```
for (i=0; i<maxIn; i++)
{
   y[i]=0;
   for (j=0; j<Taps; j++)
   {
     y[i] += x[i+j]*w[j];
   }
}
```





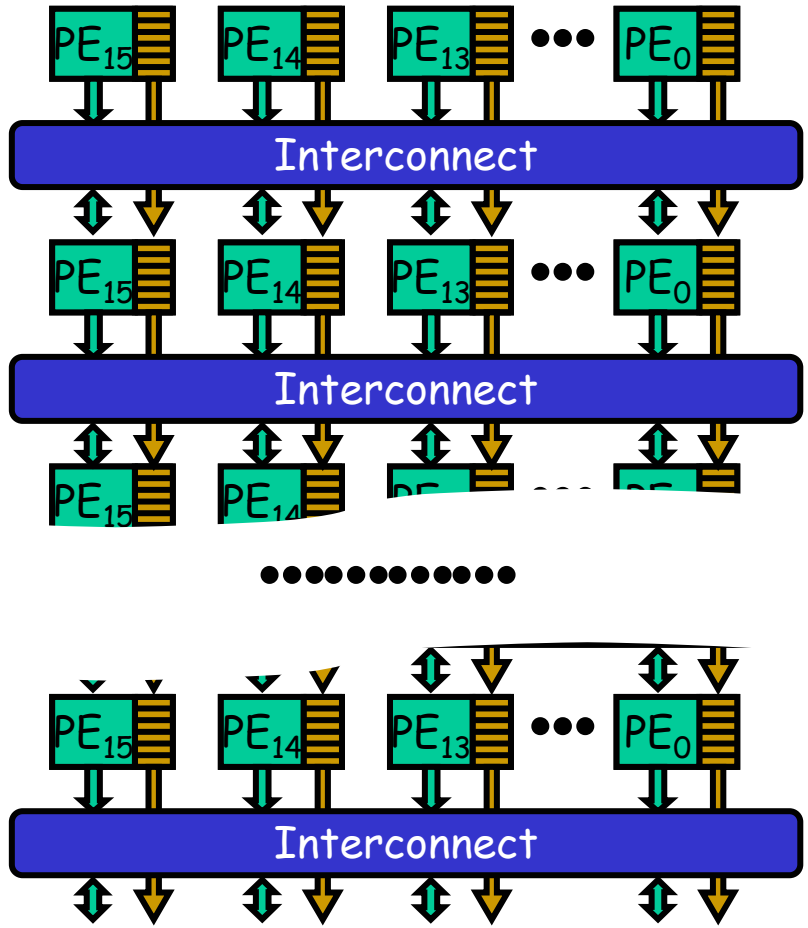
Lots of apps fit:

- DSP
- Image Processing
- Cryptography
- Packet Processing

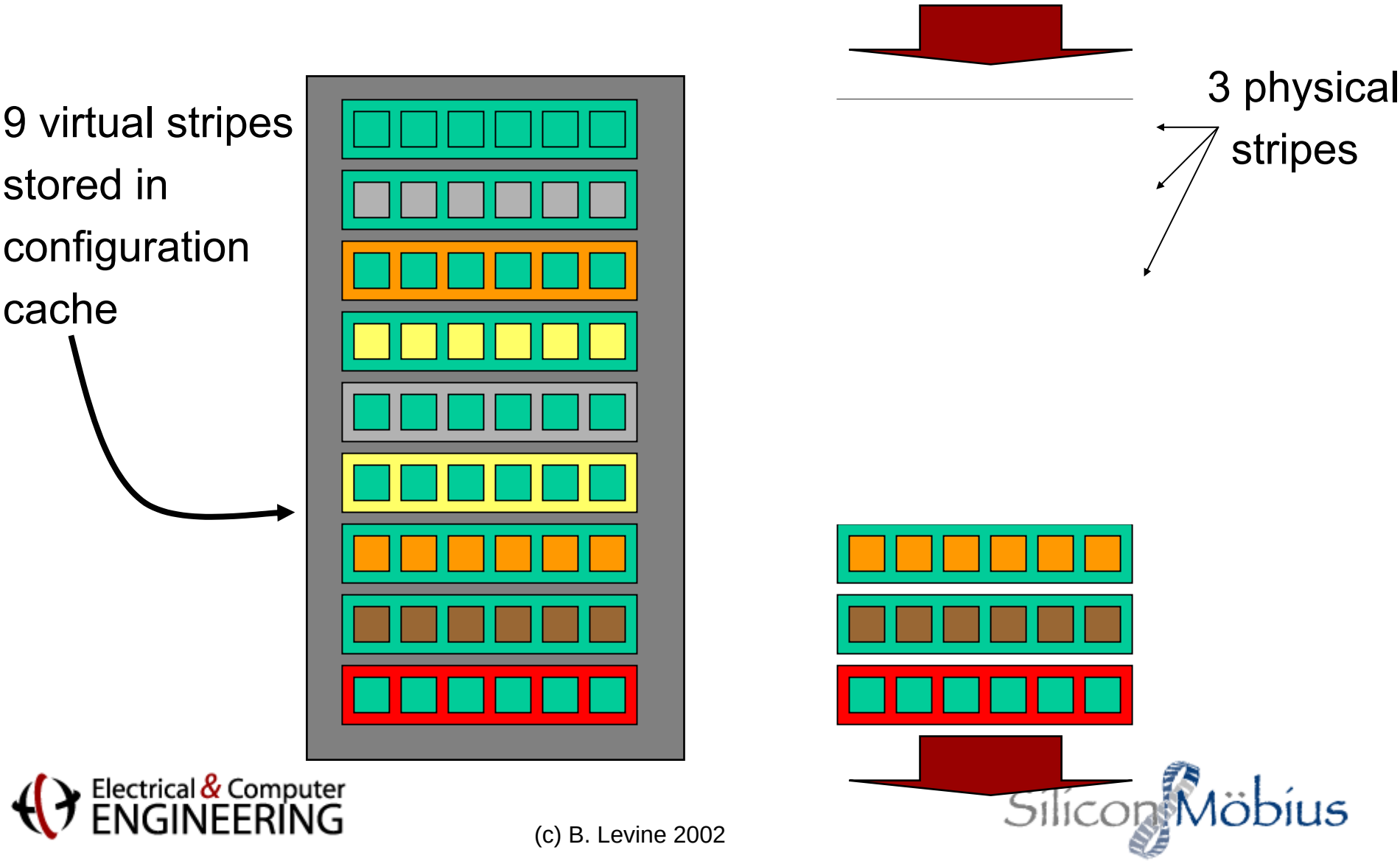
PipeRench Fabric

A programmable, pipelined data path containing:

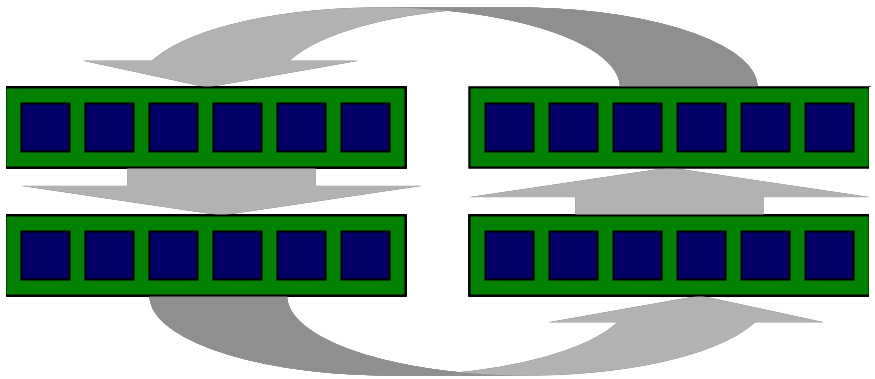
Processing elements
Local interconnect
Pass Registers
Unbounded Depth



Pipeline Virtualization

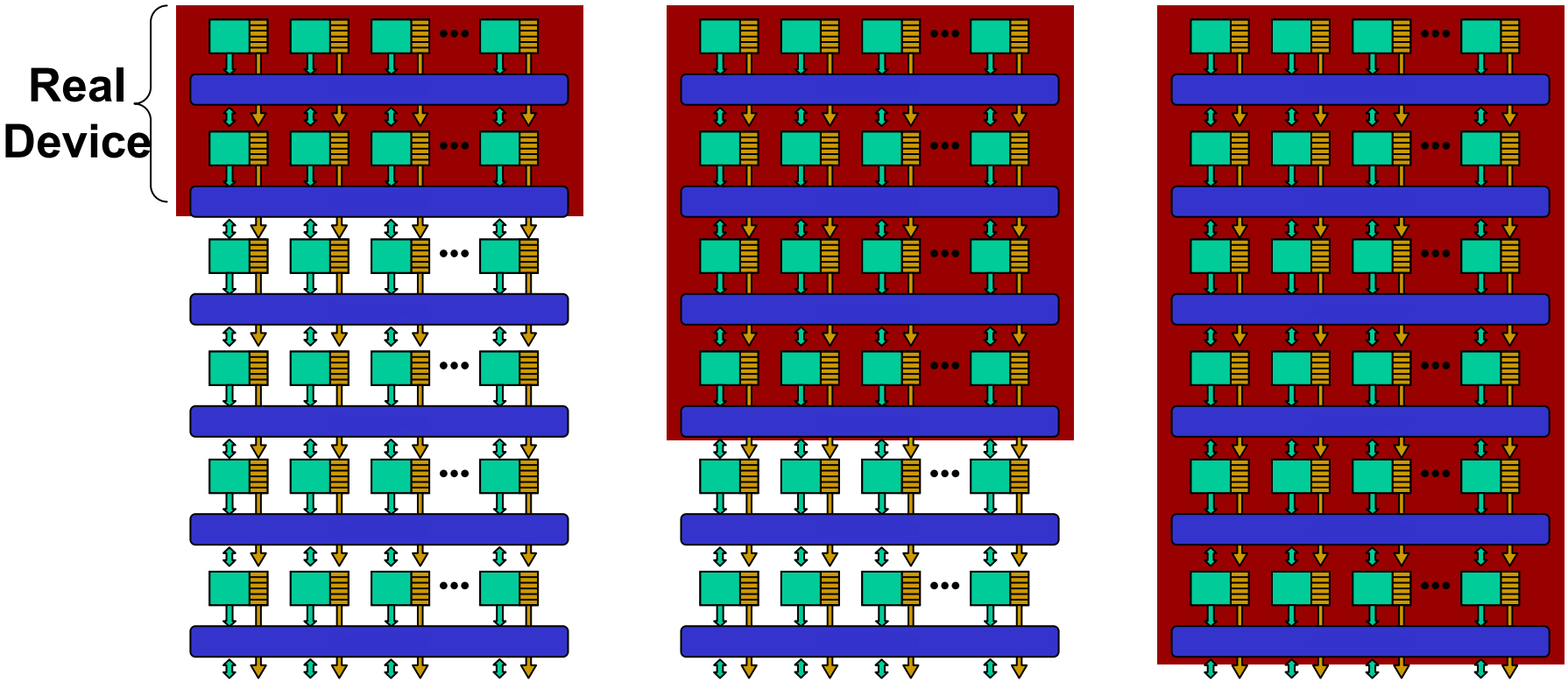


Pipeline Virtualization



- Stripes are connected in a ring.
- Data can always pass between adjacent virtual stripes in the physical fabric.

Performance Scaling



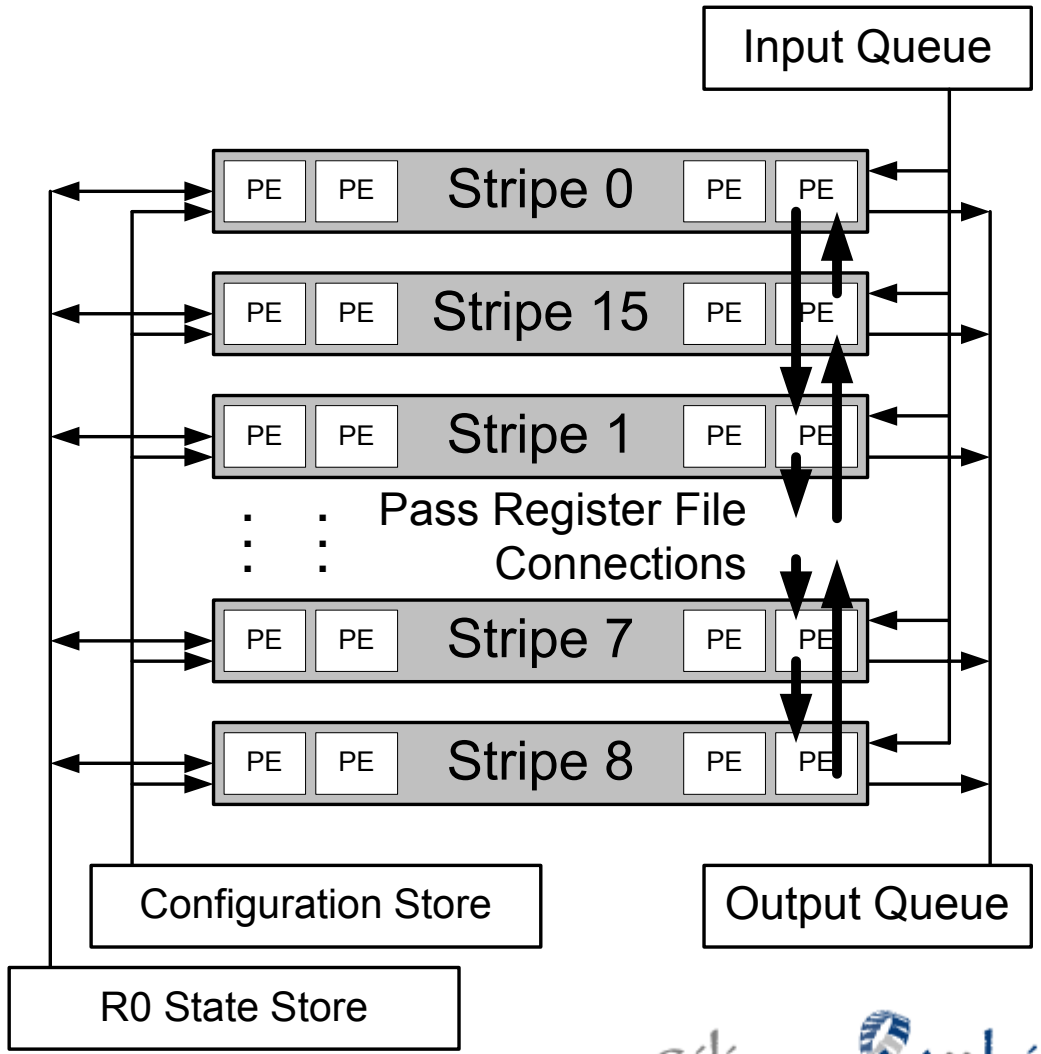
2 Stripes
2 Outputs / 6 Cycles

4 Stripes
4 Outputs / 6 Cycles

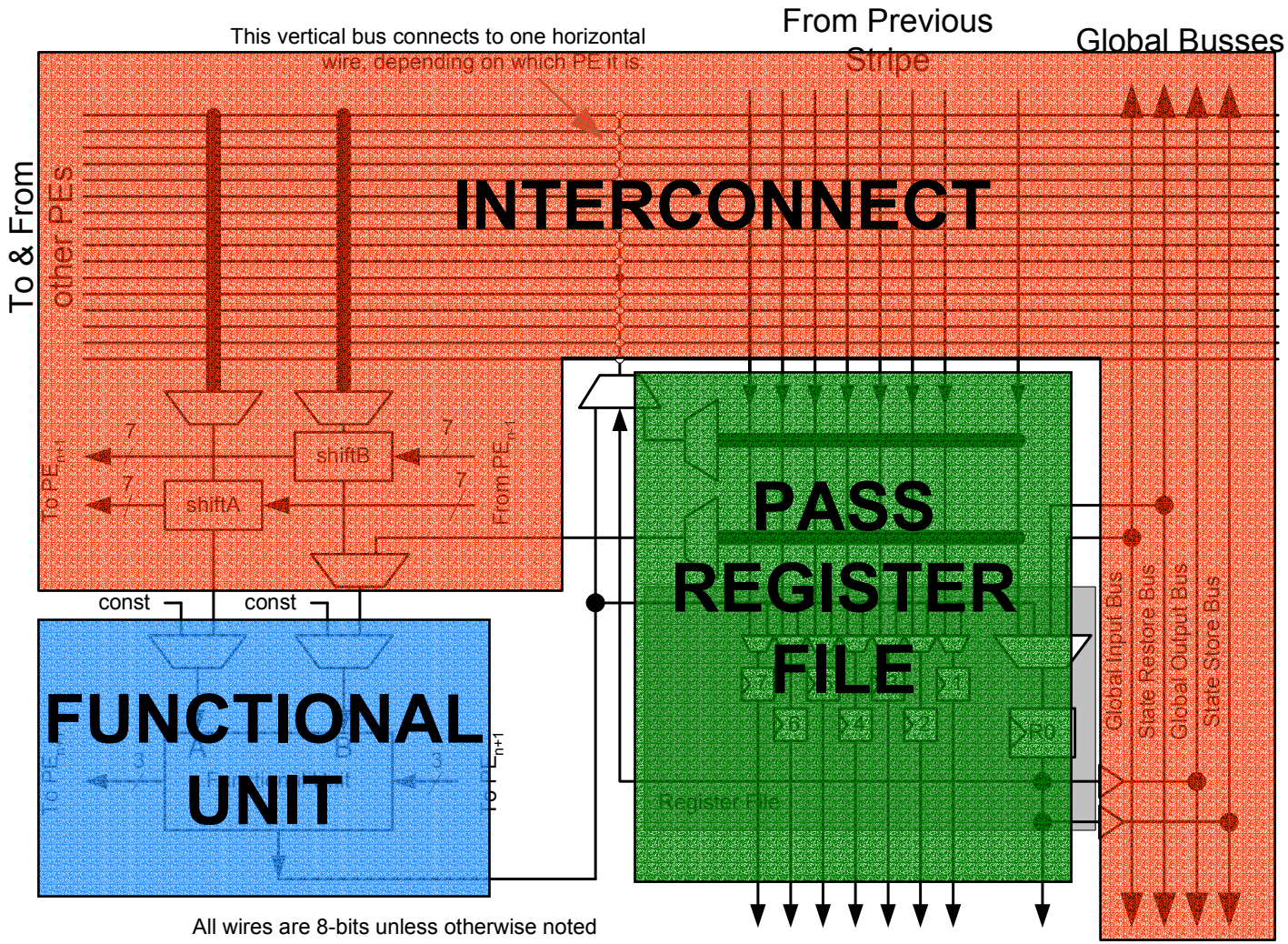
6 Stripes
6 Outputs / 6 Cycles

Chip Structure

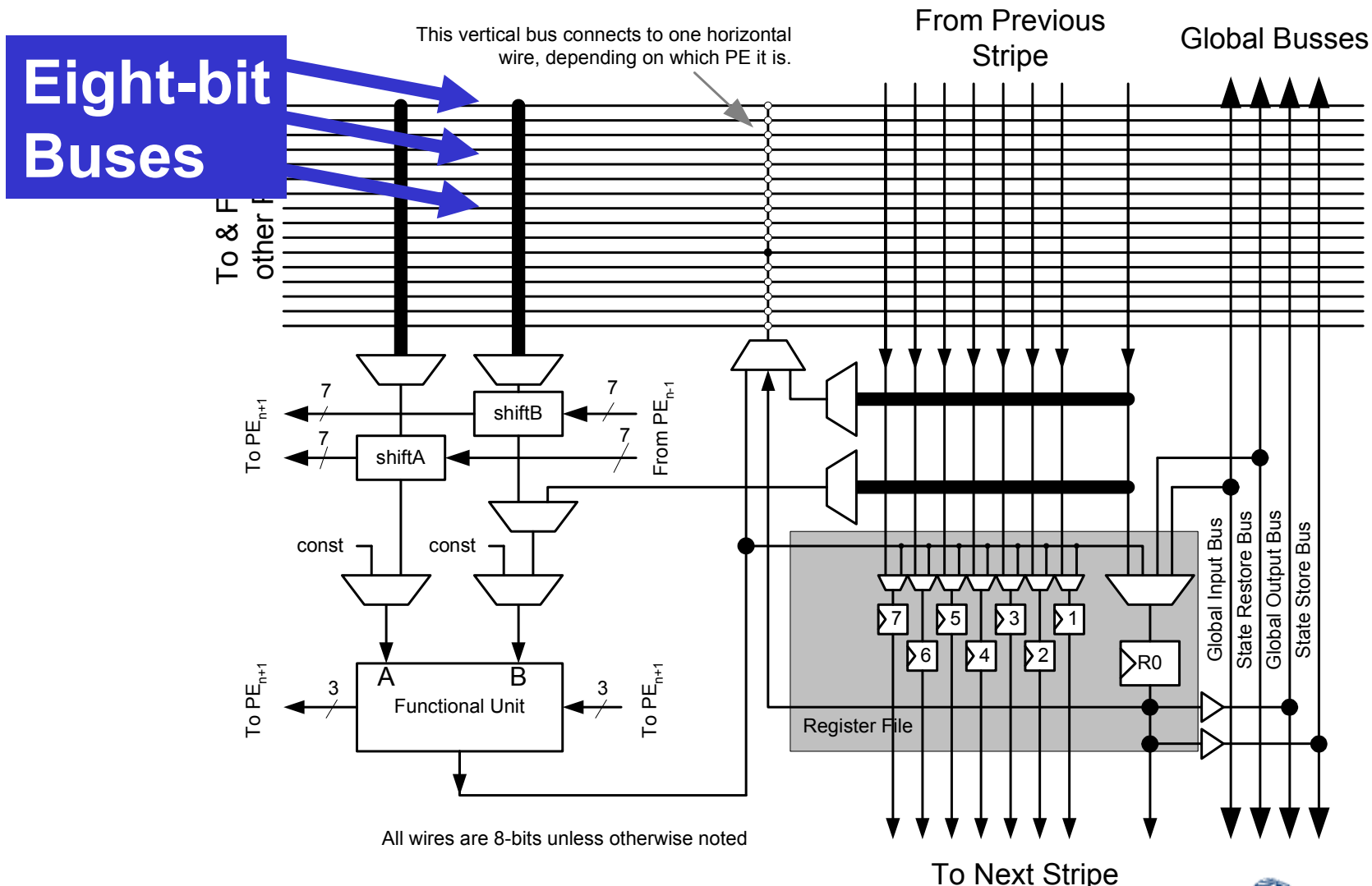
- Ring Structure
  - Interleaved
- Global Buses
  - Inputs
  - Outputs
  - Configuration
  - State Storage
- 16 Stripes
  - 16 8-bit PEs in each stripe



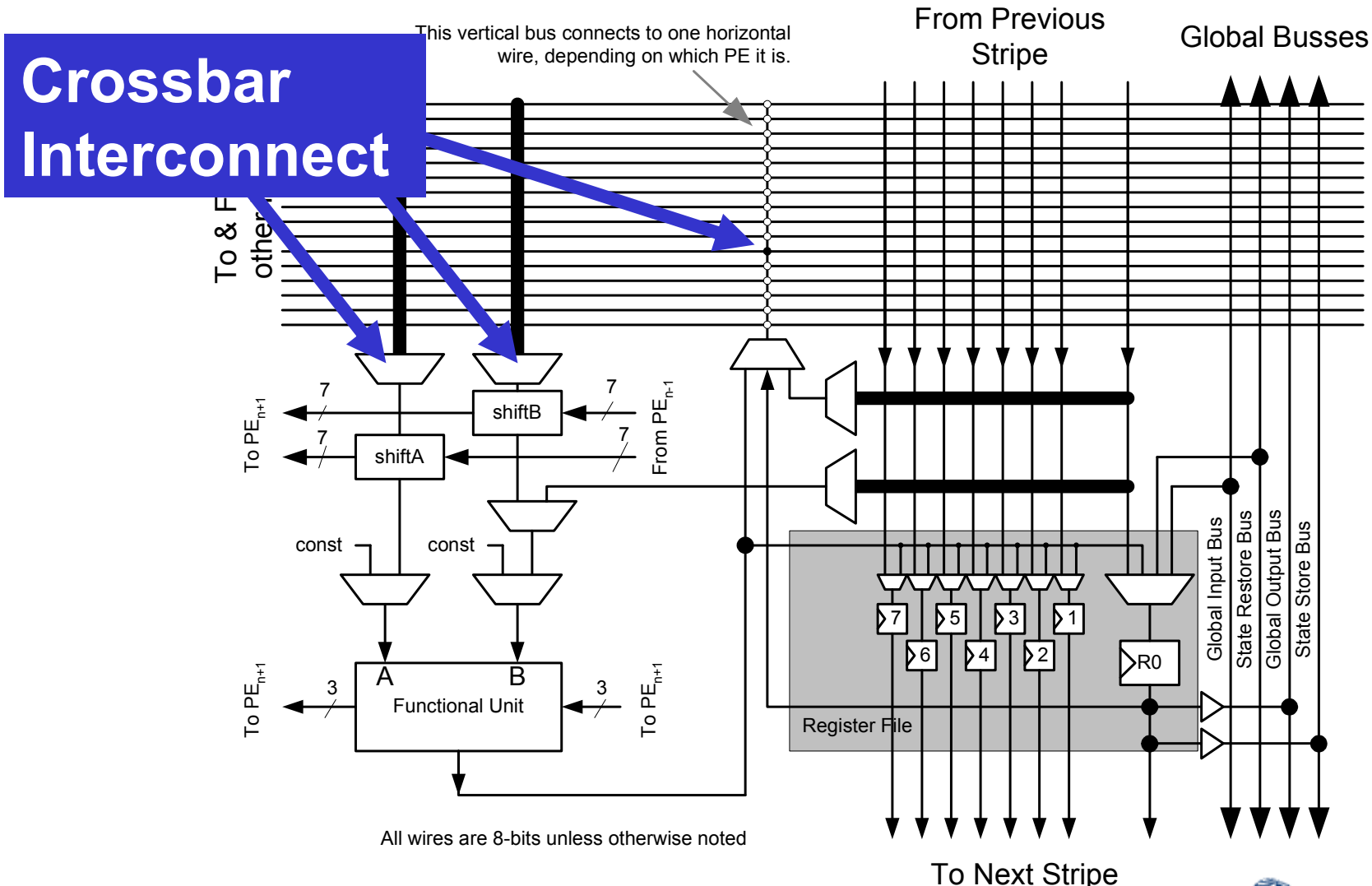
PE Architecture



PE Architecture

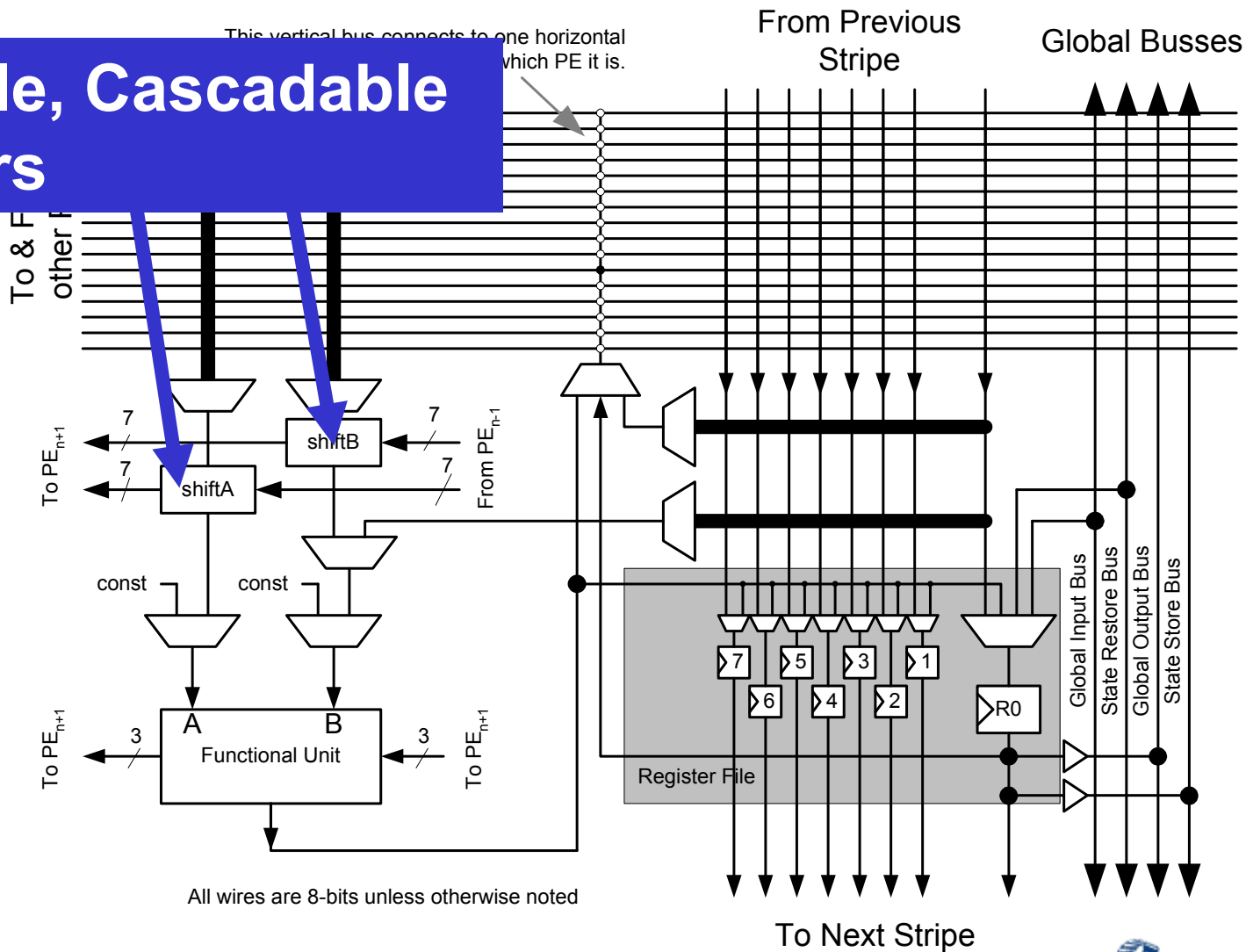


PE Architecture



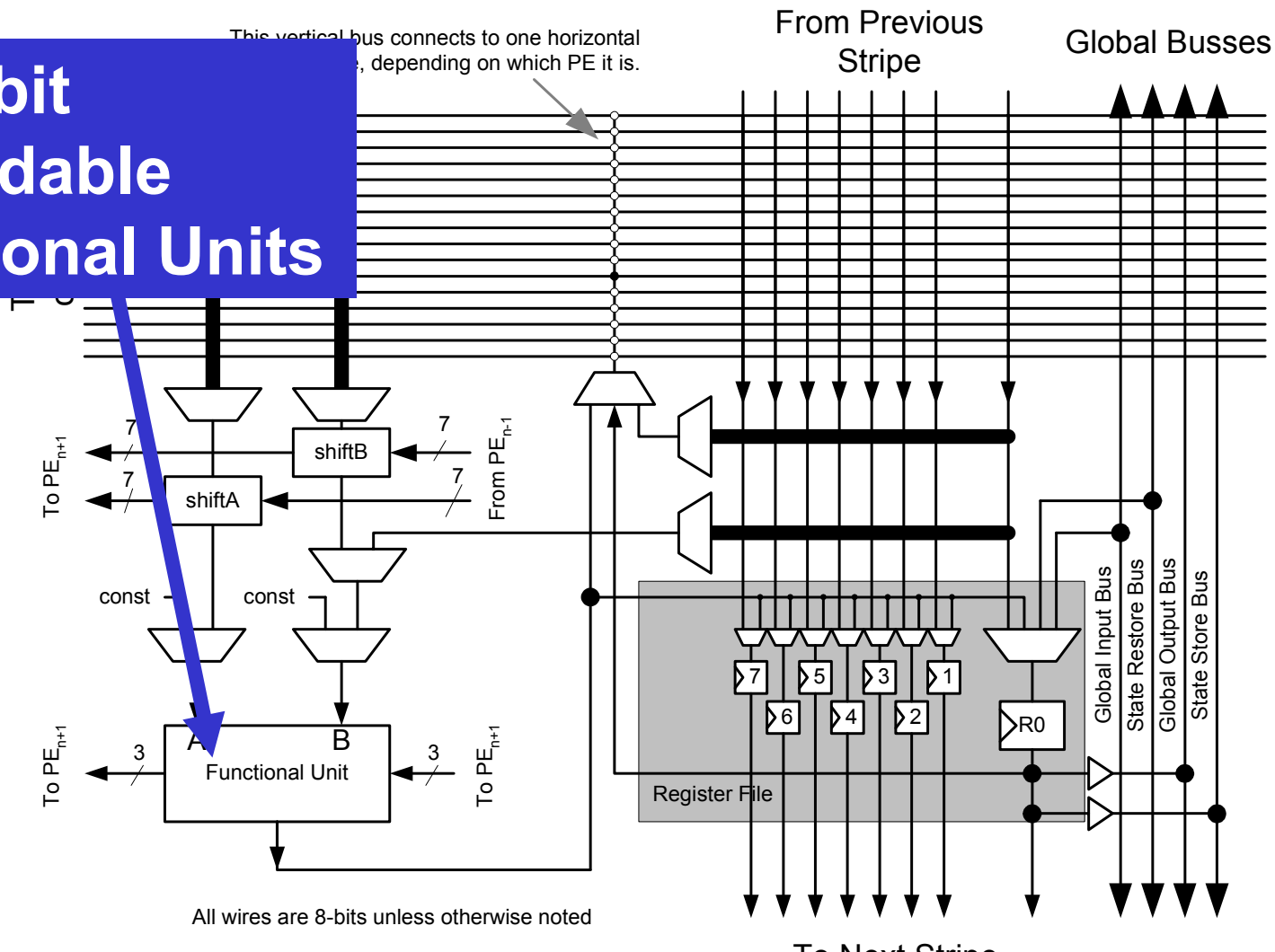
PE Architecture

Flexible, Cascadable Shifters

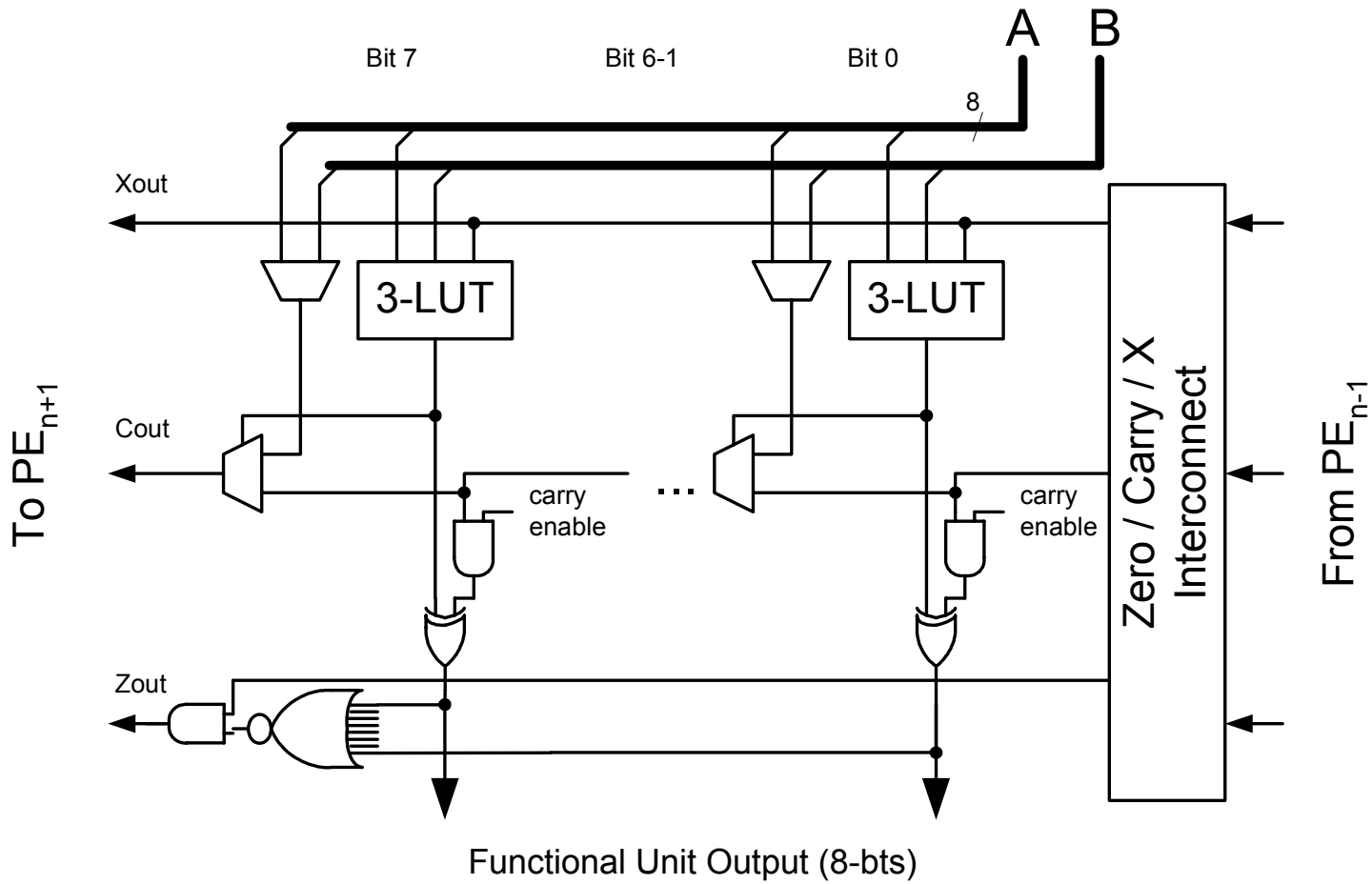


PE Architecture

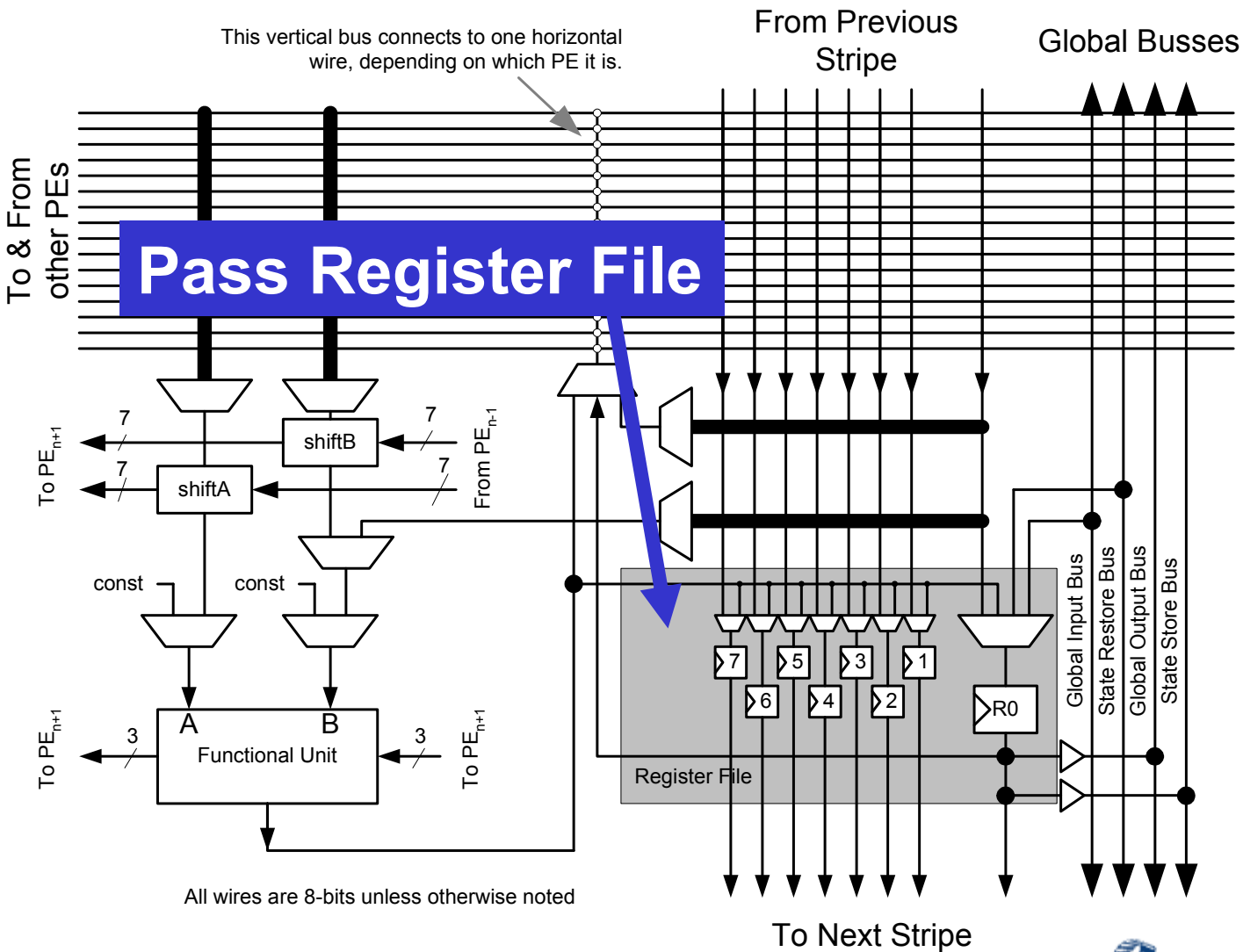
Eight-bit Cascadable Functional Units



Functional Unit Architecture



PE Architecture

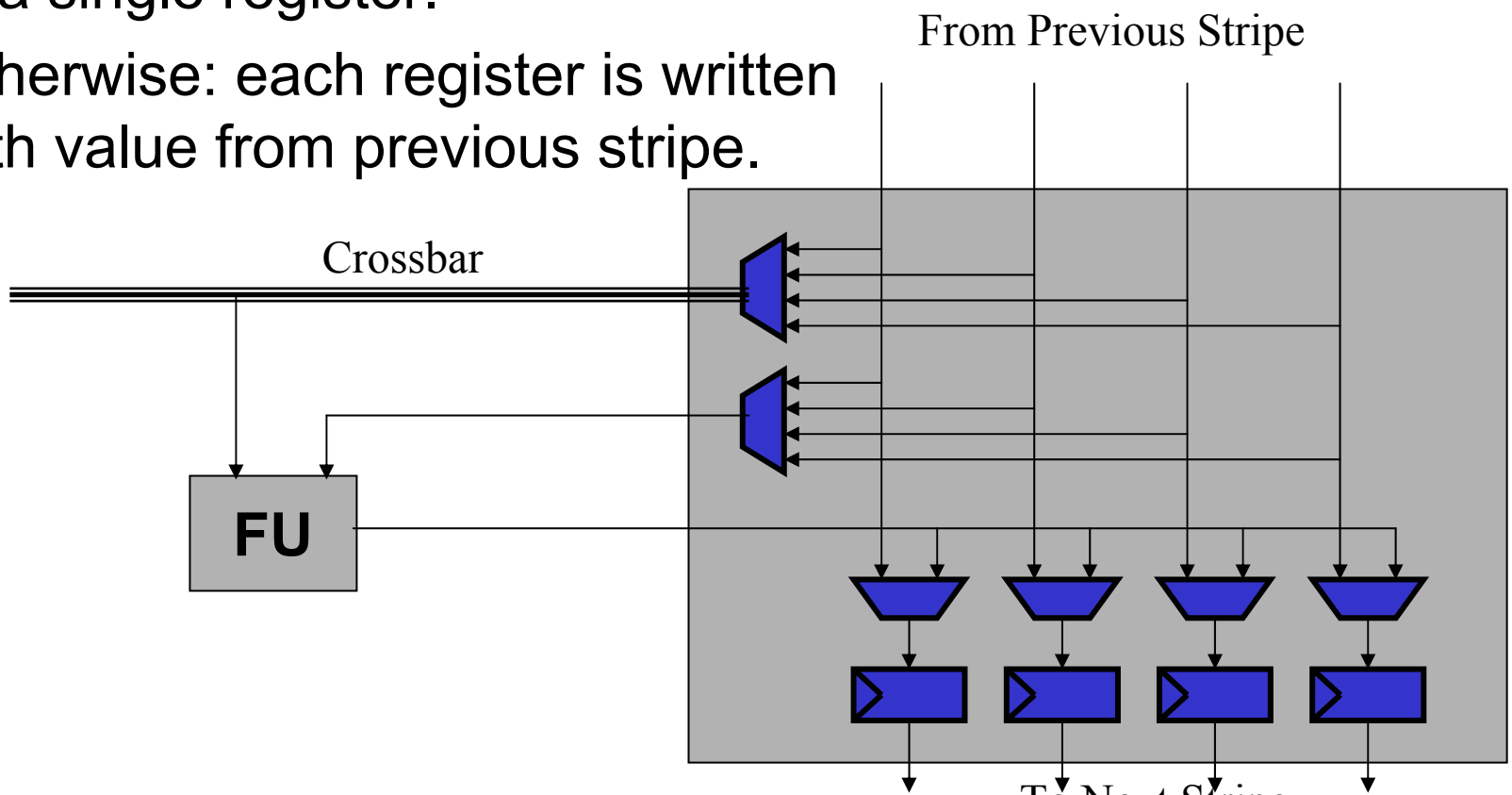


Electrical & Computer ENGINEERING

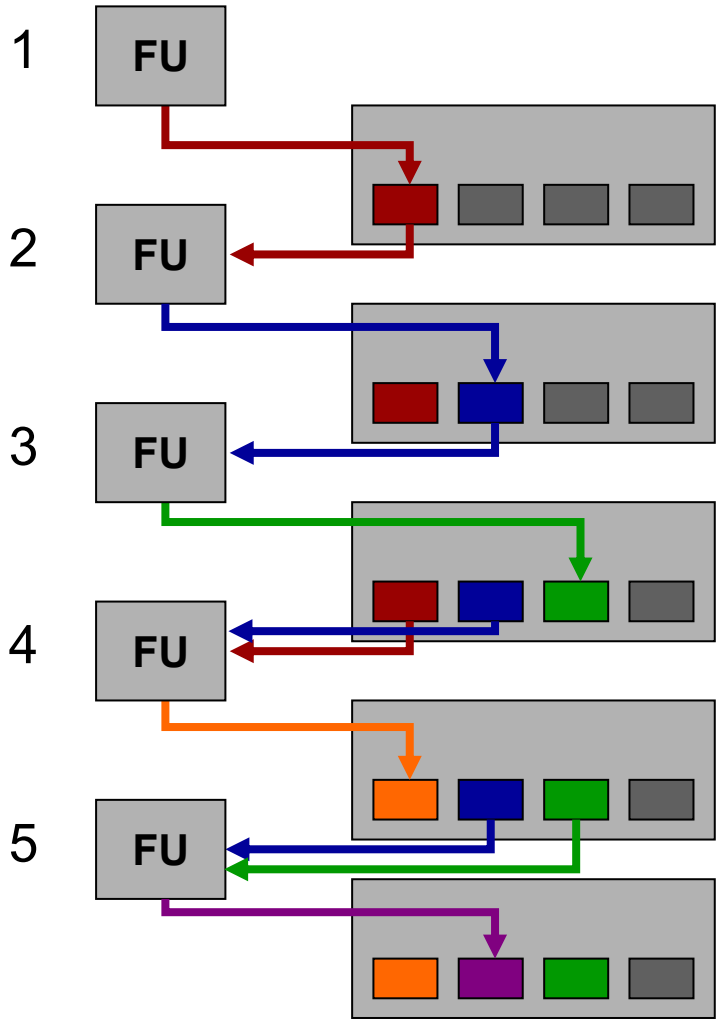
Silicon Möbius

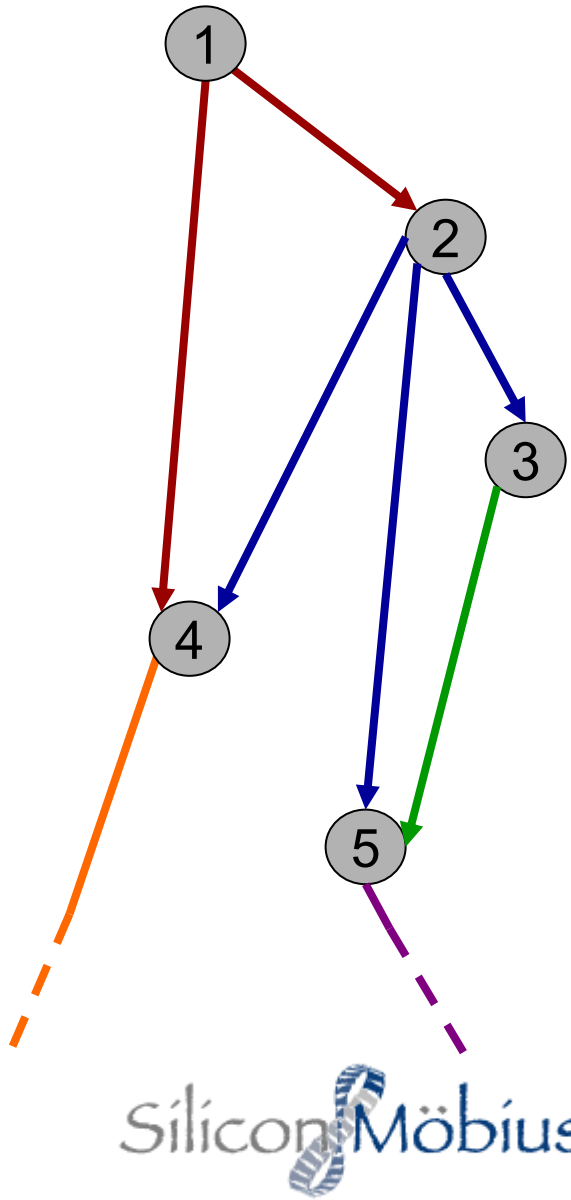
Pass Register File

- Two values can be read in each stripe.
- PE can write one new value to a single register.
- Otherwise: each register is written with value from previous stripe.

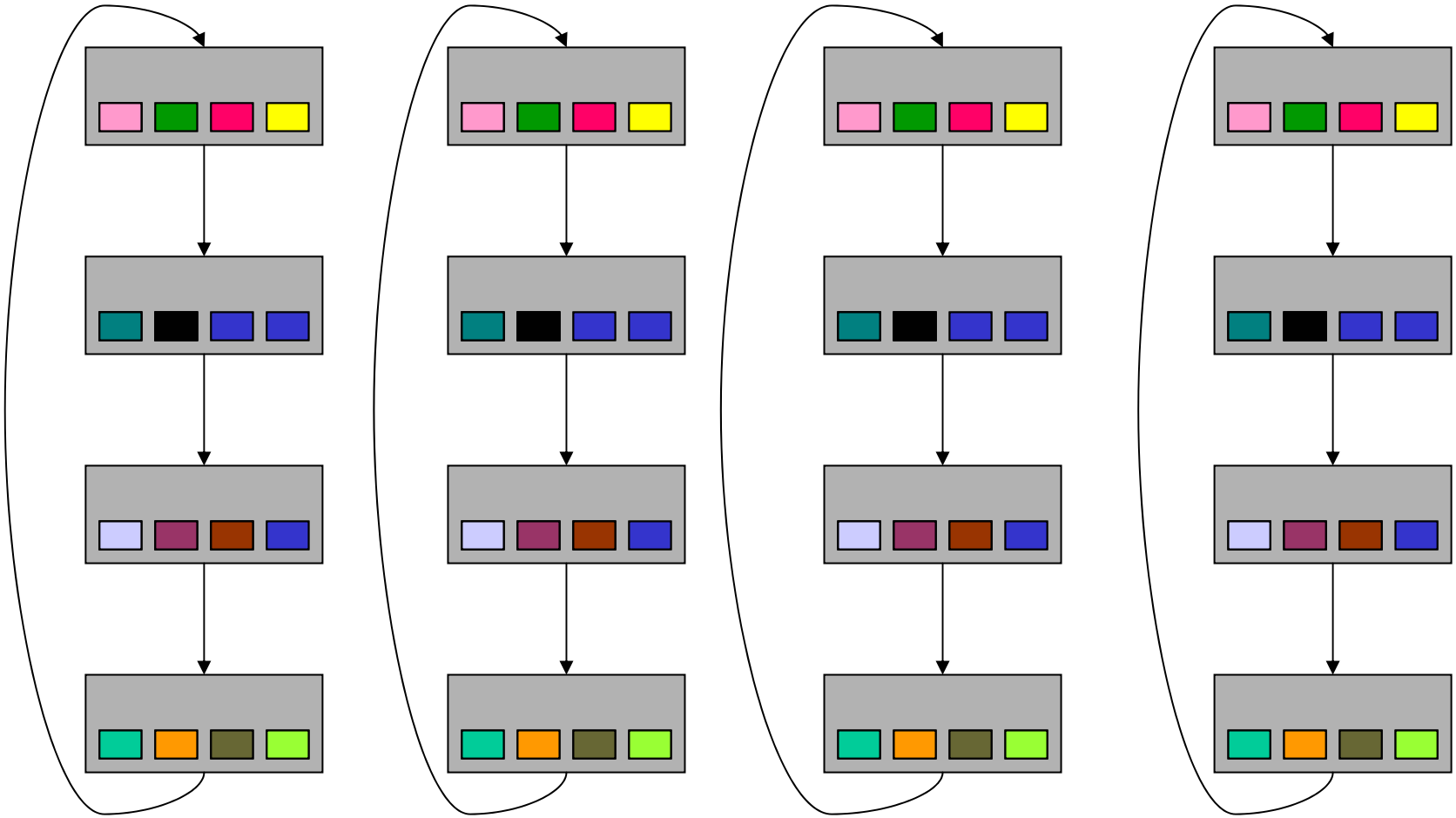


Pass Register File Operation



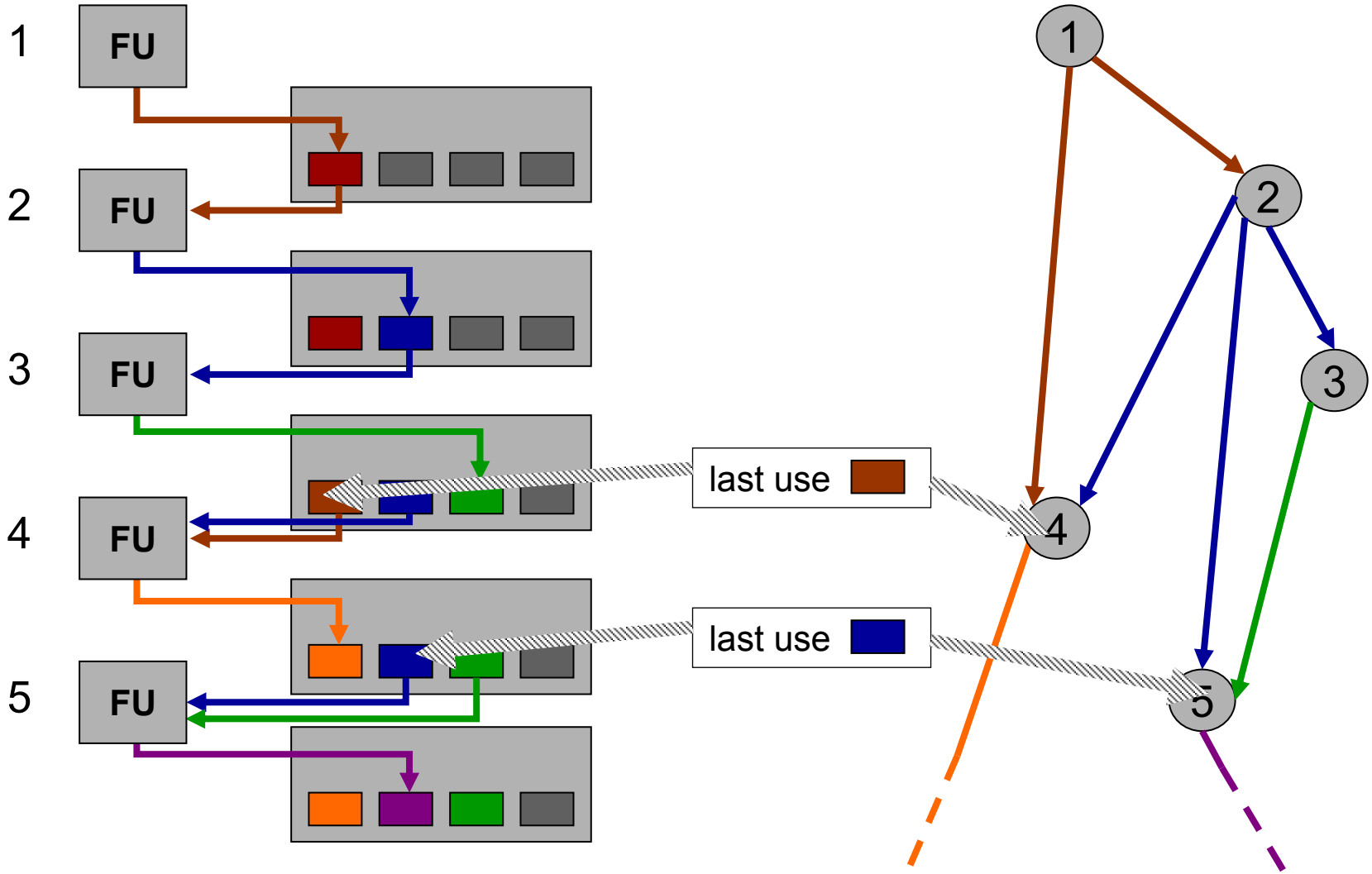


Pass Register Problem



Old register values cycle through fabric endlessly.
Extra switching consumes power.

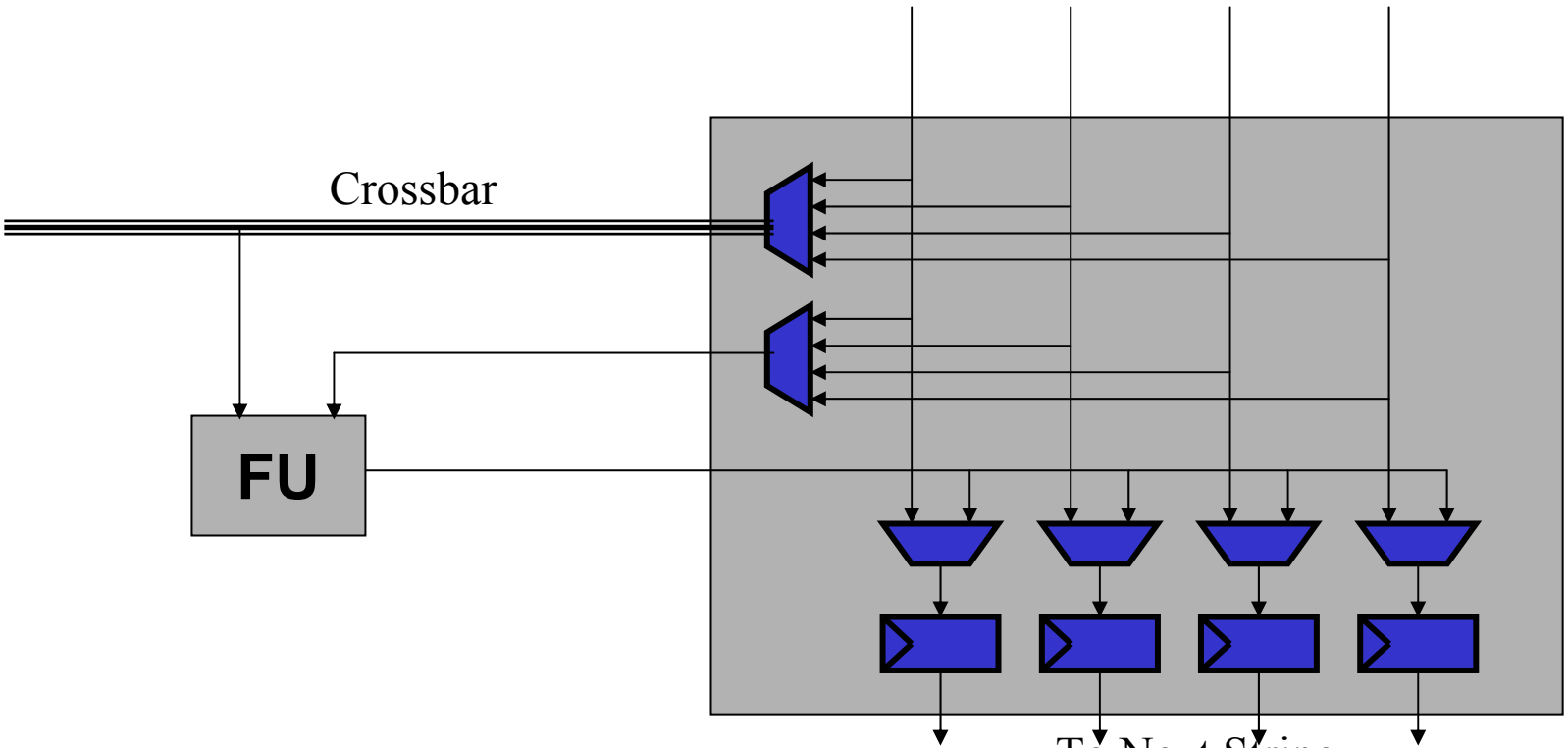
Pass Register File Operation



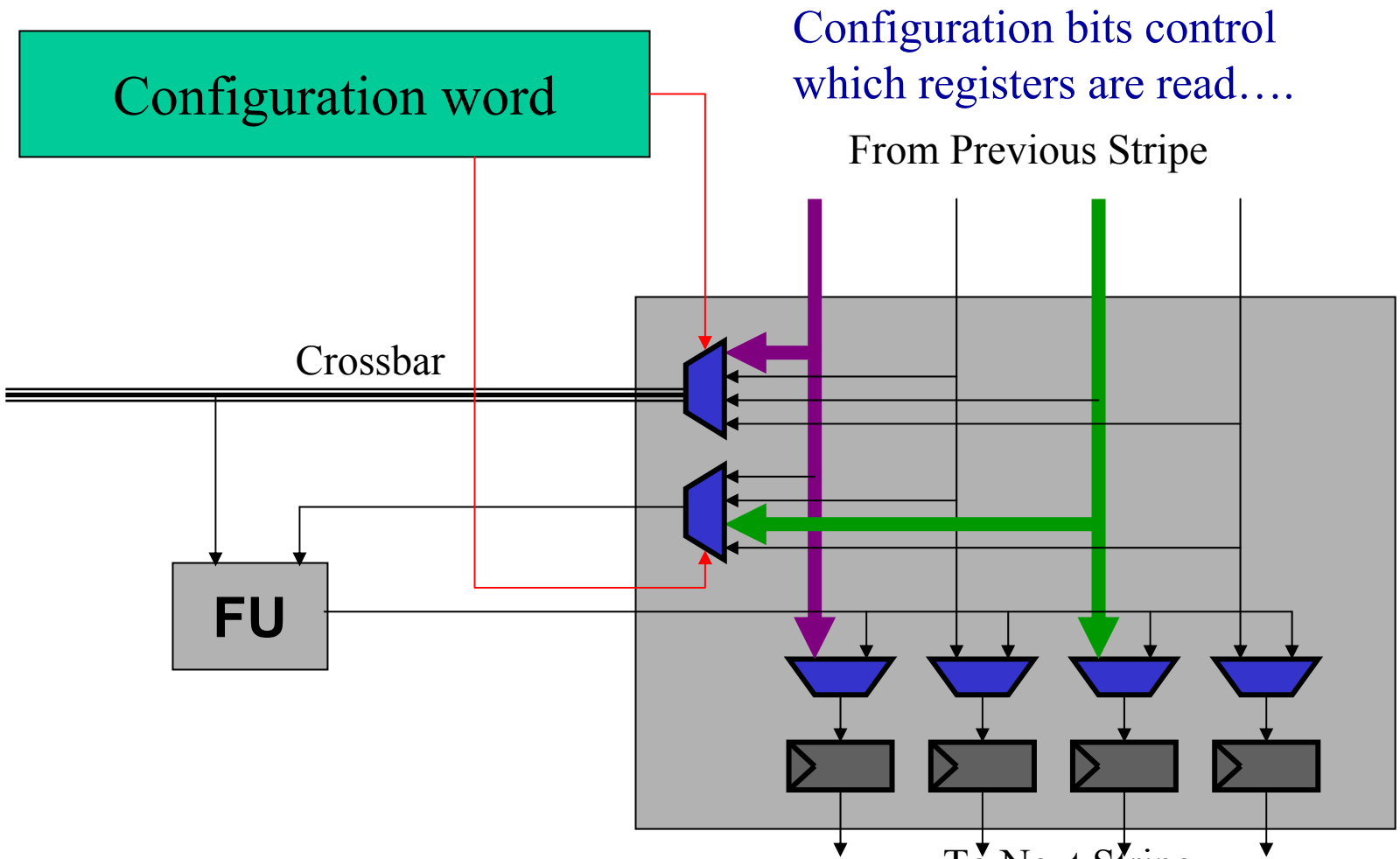
Solution: Register Kill

Configuration word

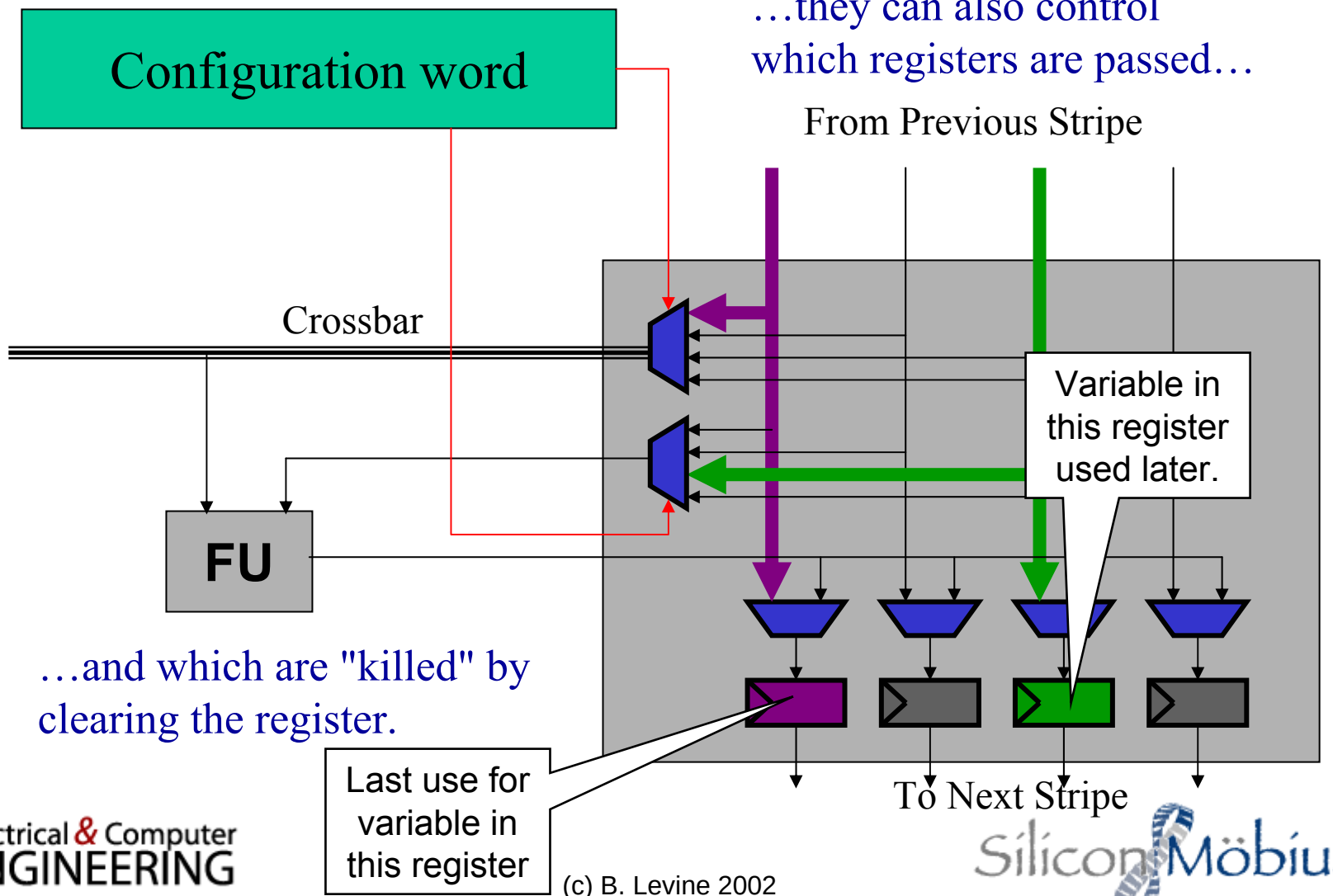
Configuration bits control which registers are read….



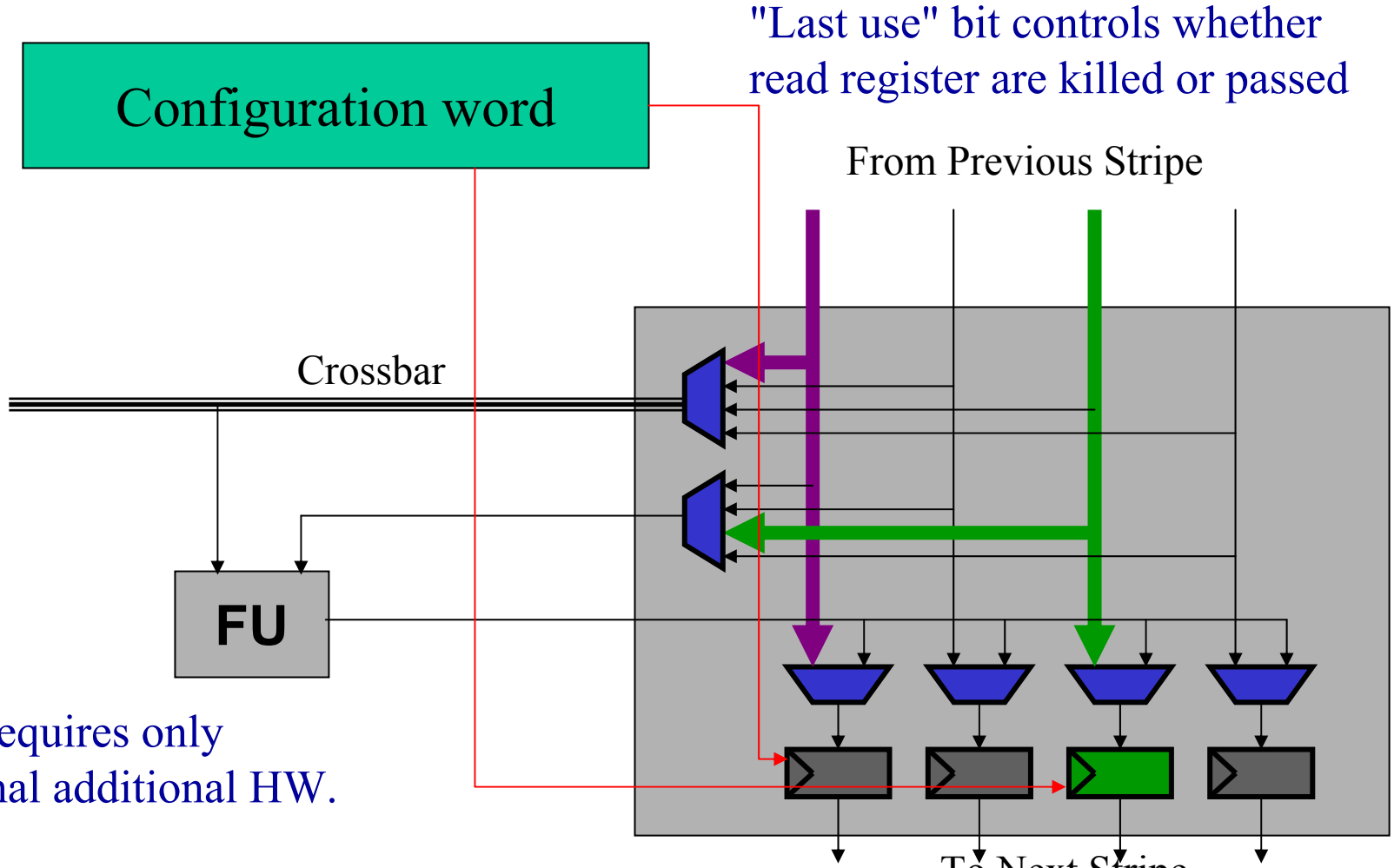
Solution: Register Kill



Solution: Register Kill



Solution: Register Kill



This requires only minimal additional HW.

Implemented Hardware Design

- Industrial Partner: ST Microelectronics
- Process Technology: 0.18 micron, 6 metal

  3.65 million transistors

  56 sq. mm die

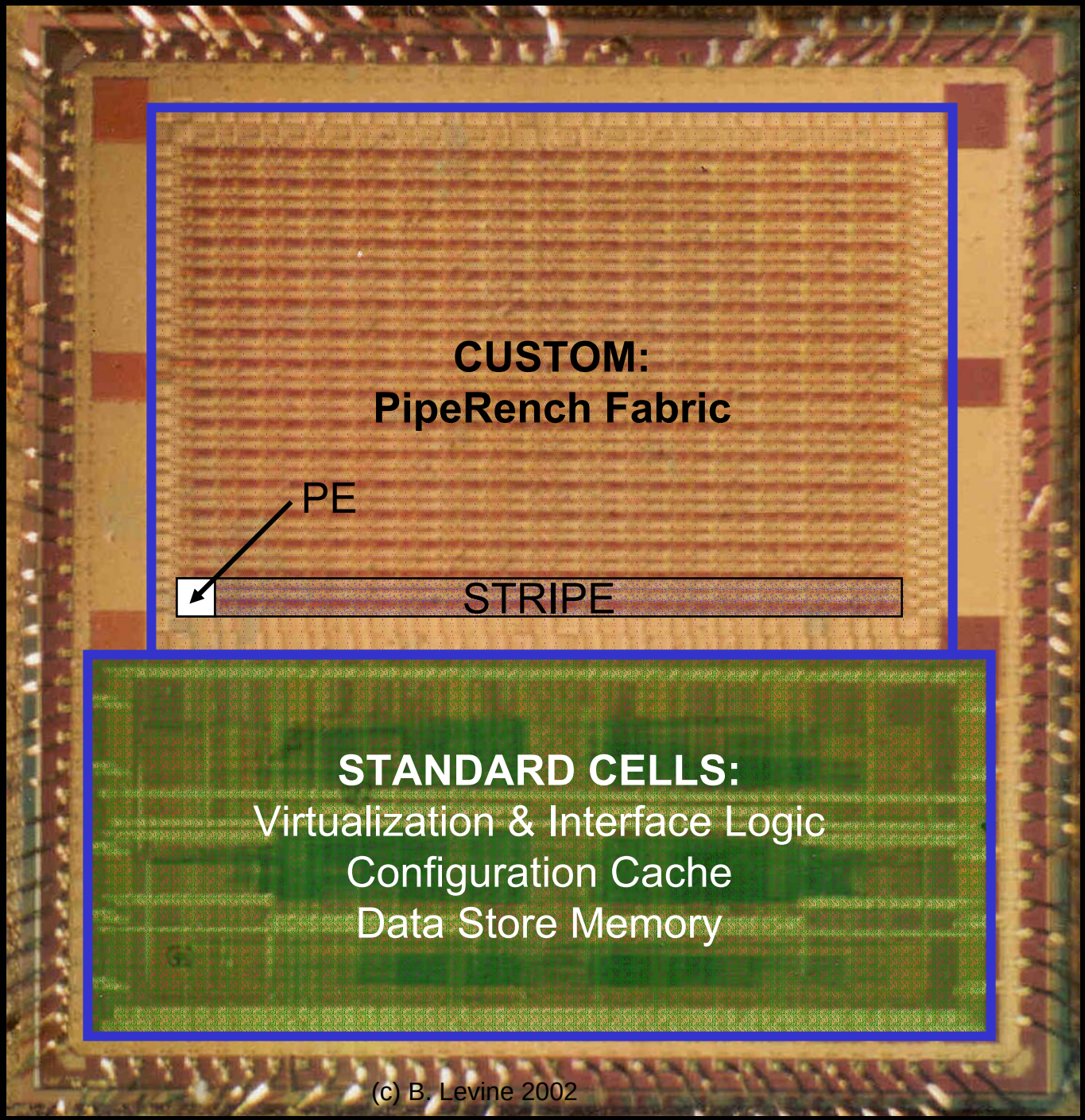
  120 MHz fabric operation

  60 MHz I/O operation

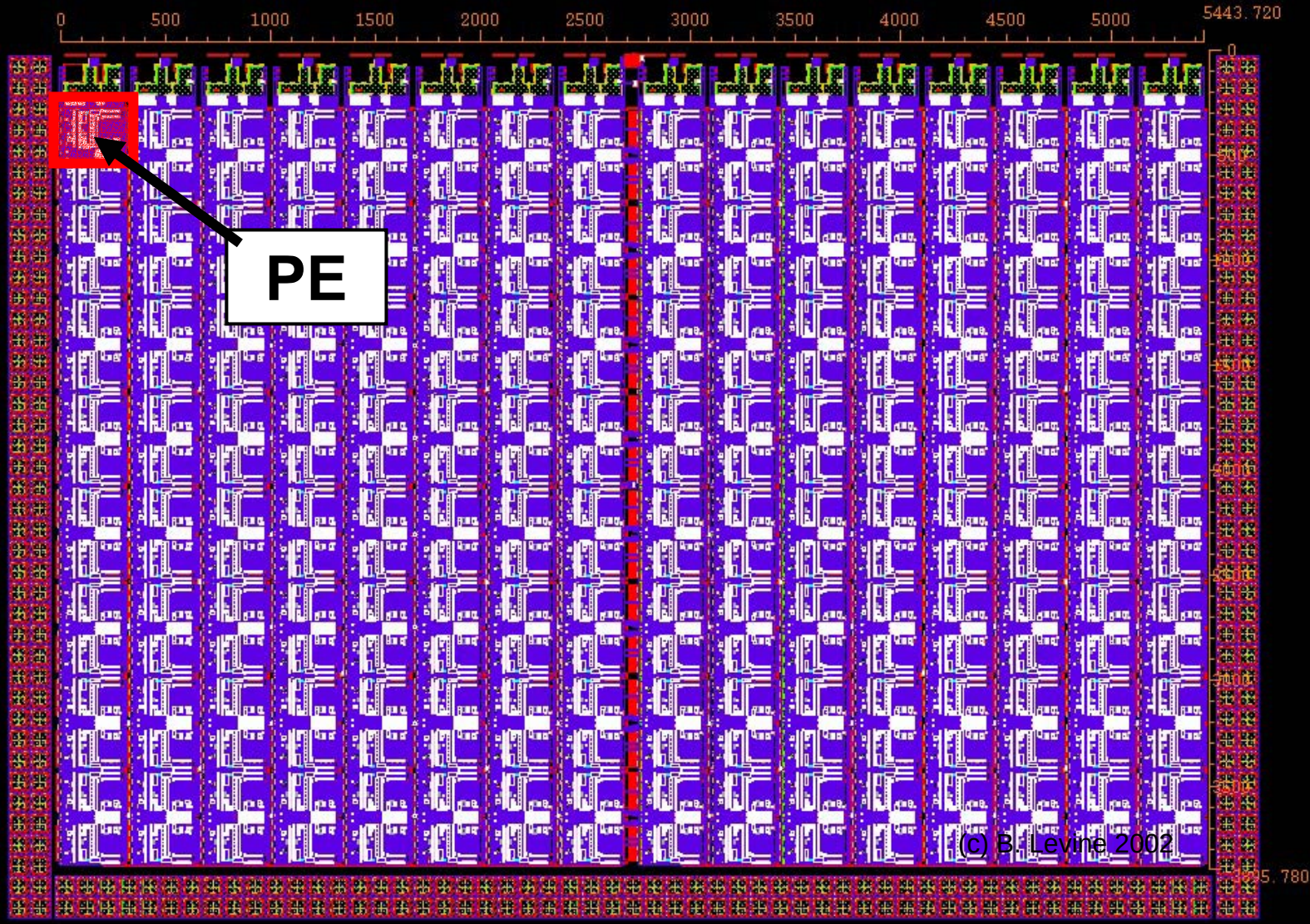
  < 3W power

  Reconfigure entire fabric in 133ns.

  Switch applications in 8ns.



Fabric Layout



PE Layout

PipeRench Status

- Chips running in our lab at CMU.
- Applications running at:
  - 132 MHz fabric speed
  - 66 MHz I/O speed.
- Decimation and correlation kernel from SAR ATR algorithm running on HW:
  - At 66 MHz, processes radar imagery at over 50 M Pixels/s.
  - consumes less than 800 mW

Register Kill Effectiveness

- Worst case power consumption due to pass register switching:

  ~ 150 mW @ 120 MHz.

  > 25% of the total chip core power consumption.

  ~ 50% fabric power consumption.
- Actual power savings varies by application, but can approach worst case.
- Required only trivial hardware modification and a simple additional assembler pass.

Other Performance

Encryption

- IDEA Encryption: 450 Mbps
  - Key is compiled into hardware
  - Compilation (including P&R) takes less than one minute
- Comparison:
  - 800 MHz Pentium III Xeon: 75.4 Mbps

Filtering

- 40 Tap 16-bit FIR Filter
  - 41.8 MSPS
- Comparable to high-end DSPs
  - Much lower clock rate
  - Without a full multiplier

  (taps are compiled into hardware)

Conclusions

- A practical virtual machine for pipelined programmable datapaths is possible.
- Virtual hardware $\Rightarrow$ physical hardware:
  - Completely self-managed on chip at run-time.
  - Enabled by fast incremental reconfiguration.
- Virtual architecture allows:
  - Easier compilation
  - Forward compatibility / Scalability
- Implemented chip has high performance and low power requirements

Future of PipeRench

- CMU has granted an exclusive worldwide license for PipeRench to SiliconMöbius.
- SiliconMöbius is commercializing PipeRench
  - Producing a family of binary-compatible chips for streaming applications like packet processing, cryptography, DSP, image and video processing.
  - Expanding the application model to include a wider range of applications.

http://www.siliconmobius.com info@siliconmobius.com