An FPGA Architecture Supporting Dynamically Controlled Power Gating

Assem A.M. Bsoul and Steven J.E. Wilton

Department of Electrical and Computer Engineering
University of British Columbia

Vancouver, B.C., Canada

What this talk is about

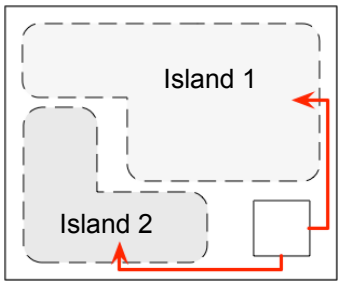
An FPGA Architecture supporting dynamic power gating:

- Turn off regions, at run-time, with on-chip control

ASIC designers do this regularly

Challenges for an FPGA:

- We don't know about application
- Routing for control signals
- Handling rush current in a programmable way



Motivation

High-end FPGAs are power-hungry

- Entering an era where we can't turn it all on at once!
- Need to selectively turn off regions when not being used
- Static control may not be enough...

Mobile hand-held applications

- Many applications have regions with long idle periods
- Could take advantage of this sort of architectural support

3

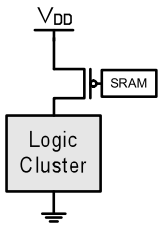
Relevant Work: Power Gating for FPGAs

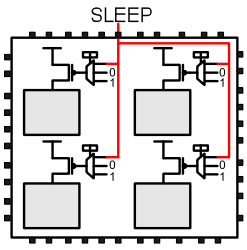
Available FPGA power gating is **statically-controlled**

– Unused FPGA parts are turned off at configuration time

Some proposals exist for dynamic control

– Exploit DR to turn FPGA blocks on/off at runtime
– Sleep transistor could come from off-chip





4

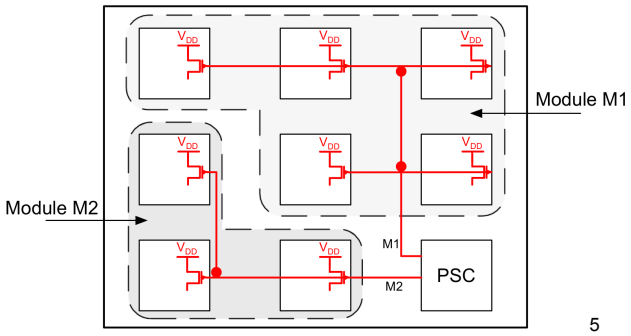
Our Architecture

Divide FPGA device into power-controlled regions.

- Support dynamically-controlled sleep mode.

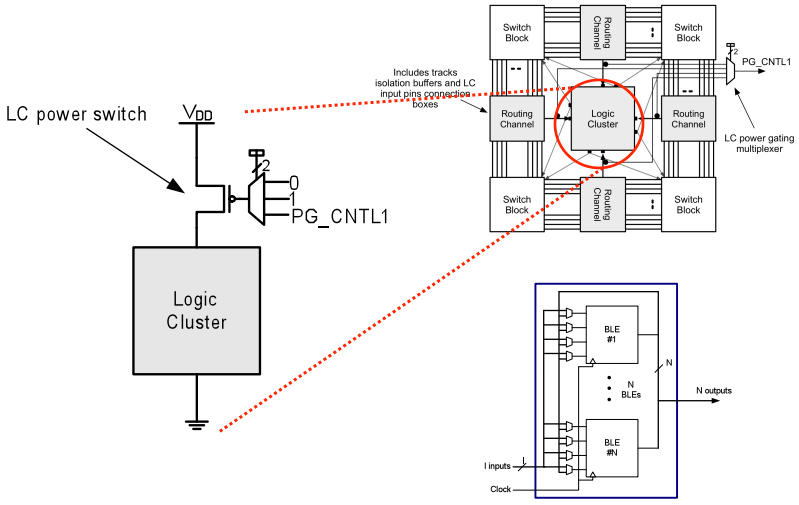
Use general-purpose routing fabric for control signals.

- Utilize unused input pins of logic clusters.

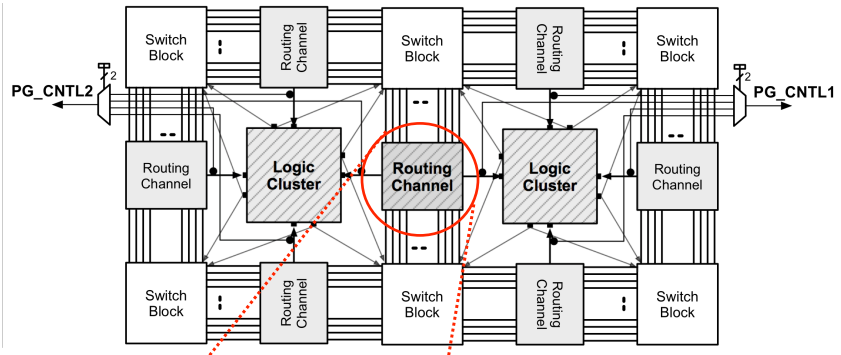


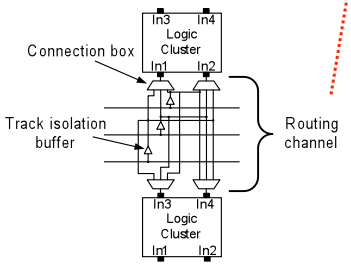
5

Basic PG Architecture – Logic Cluster



6

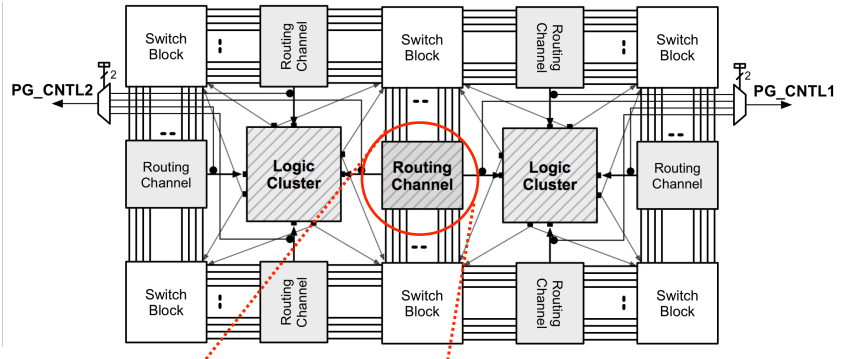


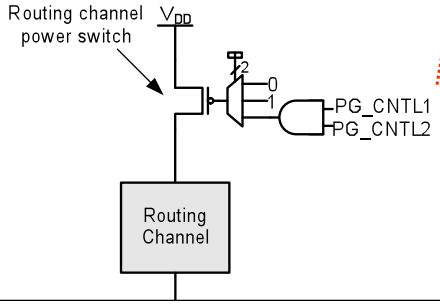


Re-use control signals from neighbouring logic

Turn off routing channel only when both neighbours are off

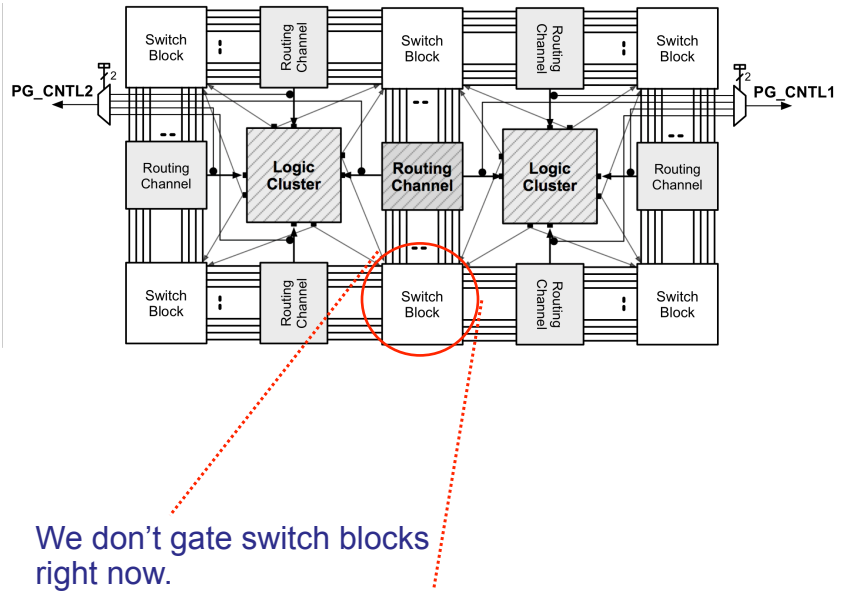
7





Interesting limitation: router can't use this channel for unrelated signals

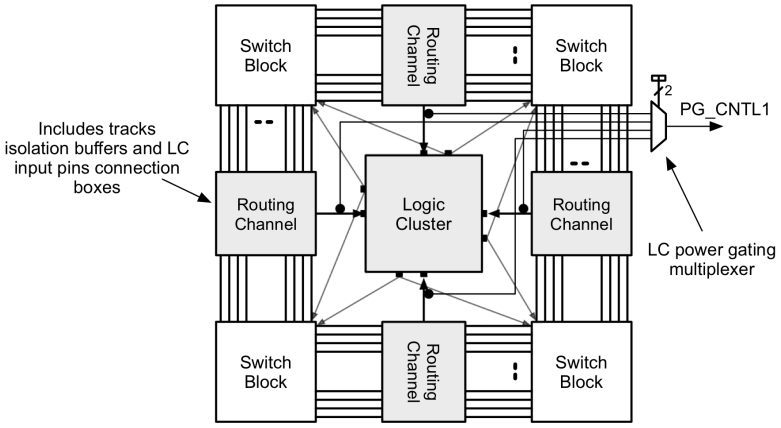
8



We don't gate switch blocks right now.
-> Interesting future work

9

Control Signals



Use general purpose routing fabric
Re-use existing input pins
- Pins are expensive.

10

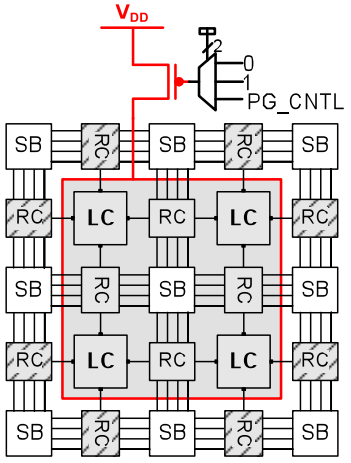
Area Overhead: Region Power Gating

Can adjust granularity

- Share a sleep transistor among tiles

Interesting tradeoff:

area vs. “CAD difficulty”



11

Rush Current

Problem: limit how much can be turned on at once

Possible solutions:

1. Expose it to the user
   - This is most familiar to an ASIC designer
2. Expose it to the CAD tool
3. Dedicated architectural support: programmable delay elements in turn-on circuits so they don’t turn on all at once

Right now, we are doing #1

12

Evaluation:

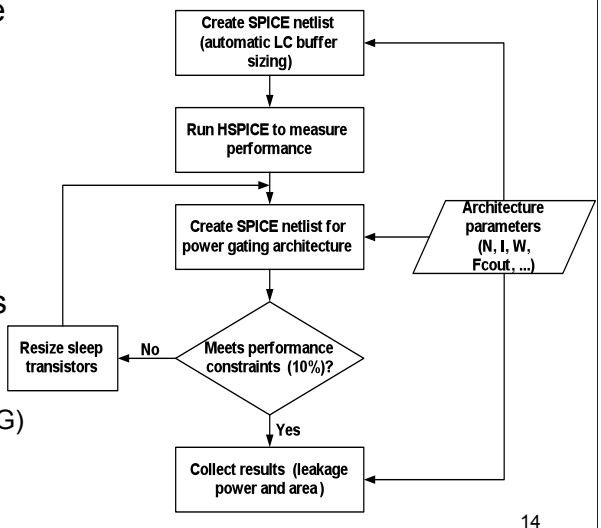
The Bad: Area, Delay, Leakage power overhead

The Good: Potential leakage reduction

13

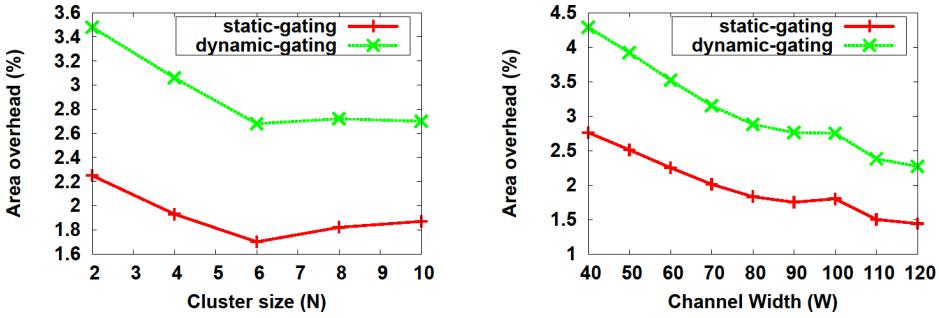
Experimental Setup

- Sweep architecture
  - N (cluster)
  - W (channel)
  - R (region)
- 45 nm PTM
- Three architectures
  - Ungated
  - Static-gating (SG)
  - Dynamic-gating (DG)



14

Area Overhead: Tile



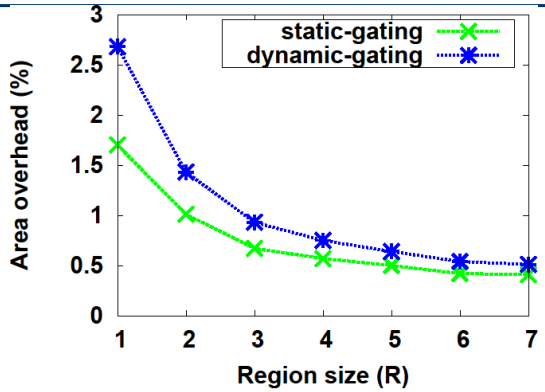
Area overhead compared to un-gated:

– **dynamic-gating 2.6%, static-gating 1.7% (N=6).**

Overhead for **dynamic-gating** is 33% than for **static-gating**.

15

Granularity Results: Area overhead



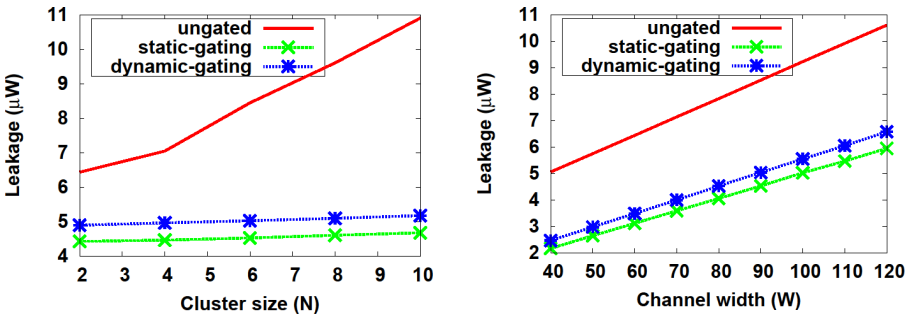
Area overhead compared to ungated:

– **dynamic-gating 0.75%, static-gating 0.57% (R=4).**

Area overhead decrease as R increase.

16

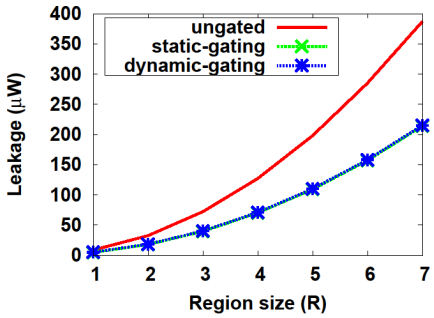
Leakage Overhead



- **Dynamic-gating** has **11%** more leakage than **static-gating** (N=6).
- Compared to **ungated**, **static-gating** and **dynamic-gating** reduce leakage in sleep mode by more than **40%** (SBs leakage is included).
- Leakage reduction increase with increased N (W).

17

Granularity Results: Leakage Overhead



- **Dynamic-gating** has **0.8%** more leakage than **static-gating** (R=4).
- Compared to **ungated**, **static-gating** and **dynamic-gating** reduce leakage in sleep mode by more than **44%**.
- Leakage reduction increase with increased R.

18

Delay overhead is 10% by design.

- We choose sleep transistors such that delay impact is no more than 10%
- Tradeoff: delay overhead vs. area overhead

19

Potential Leakage Reduction

Use a model that relates:

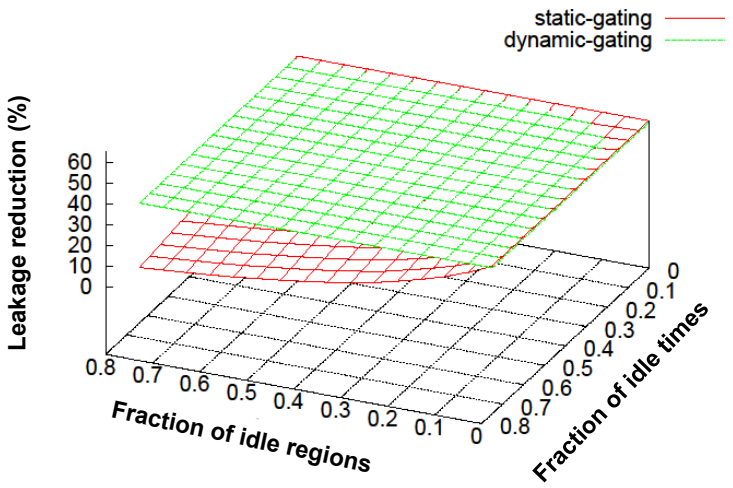
- Number and size of idle regions
- Proportion of the time idle regions can be turned off
- Size of the "power state controller"
- Potential slowdown of application

… to the energy savings of the architecture

Goal: can we bound how much leakage we can expect to save?

20

Leakage energy reduction compared
to **maximum potential savings**



Key result: we can save about 40% of unnecessary leakage

Summary

Dynamically controlled power gating is possible!

- can reduce 40% of unnecessary leakage
- Small area overhead, moderate delay overhead

Next steps:

- Need to turn off switch blocks
- This needs intelligent CAD tools
- Application mapping is tricky: how much can we automate?

22