A Configurable Architecture to Limit Wakeup Current in Dynamically-Controlled Power-Gated FPGAs

Assem Bsoul and Steve Wilton
{absoul, stevew}@ece.ubc.ca

System-on-Chip Research Group
Department of Electrical and Computer Engineering
University of British Columbia

Vancouver, B.C., Canada

Our FPGA Architecture

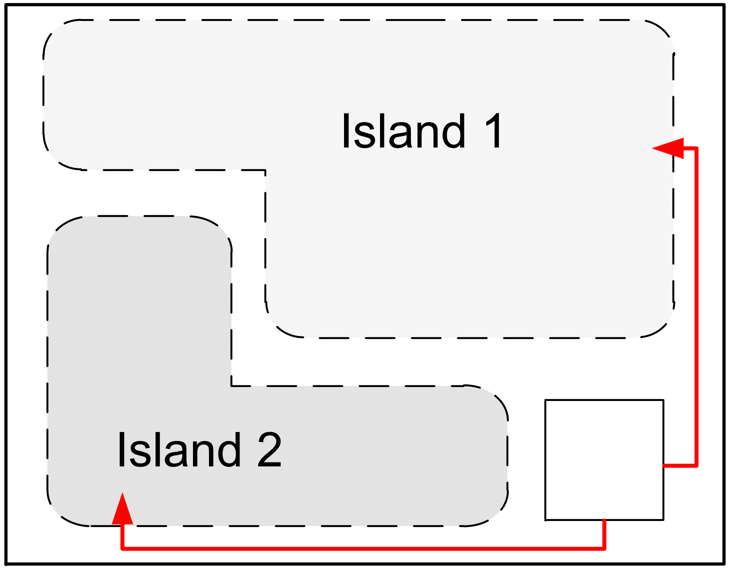
An FPGA architecture supporting dynamic power gating:

- Turn off regions at **run-time** to save power, with on-chip control

ASIC designers do this regularly

Challenges for an FPGA:

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



Motivation for new FPGA Arch.

High-end FPGAs are power hungry

- Entering an era where we can’t turn it all on at once!
- Large power leads to heat issues ➔ reliability, cooling solution

Mobile hand-held applications

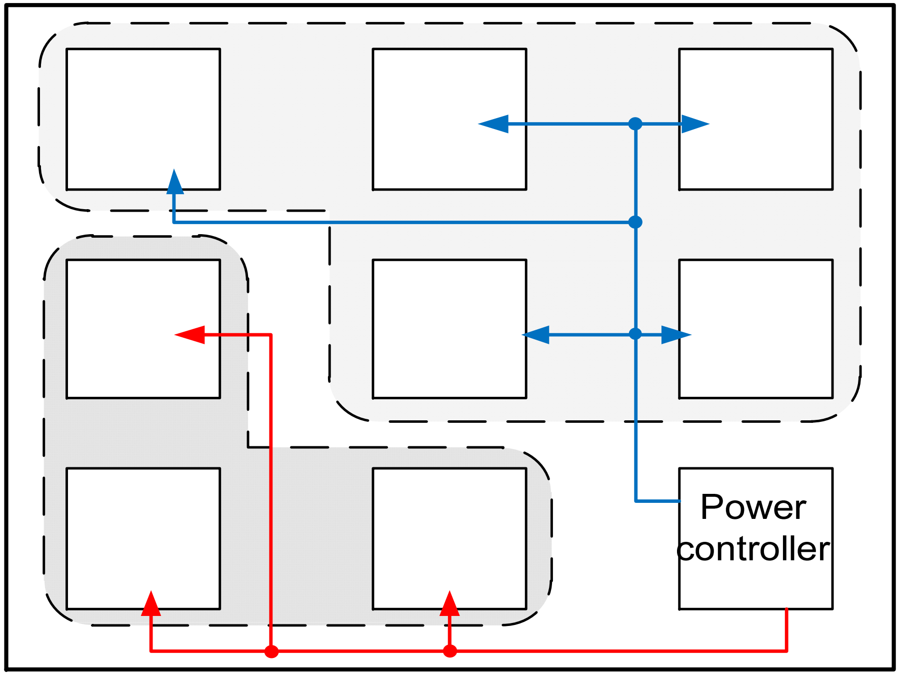
- Many applications have regions with long idle periods
- Could take advantage of this sort of architectural support

Our FPGA Architecture – Big Picture

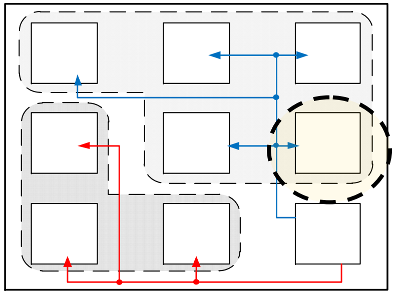
Divide FPGA device into power-controlled regions

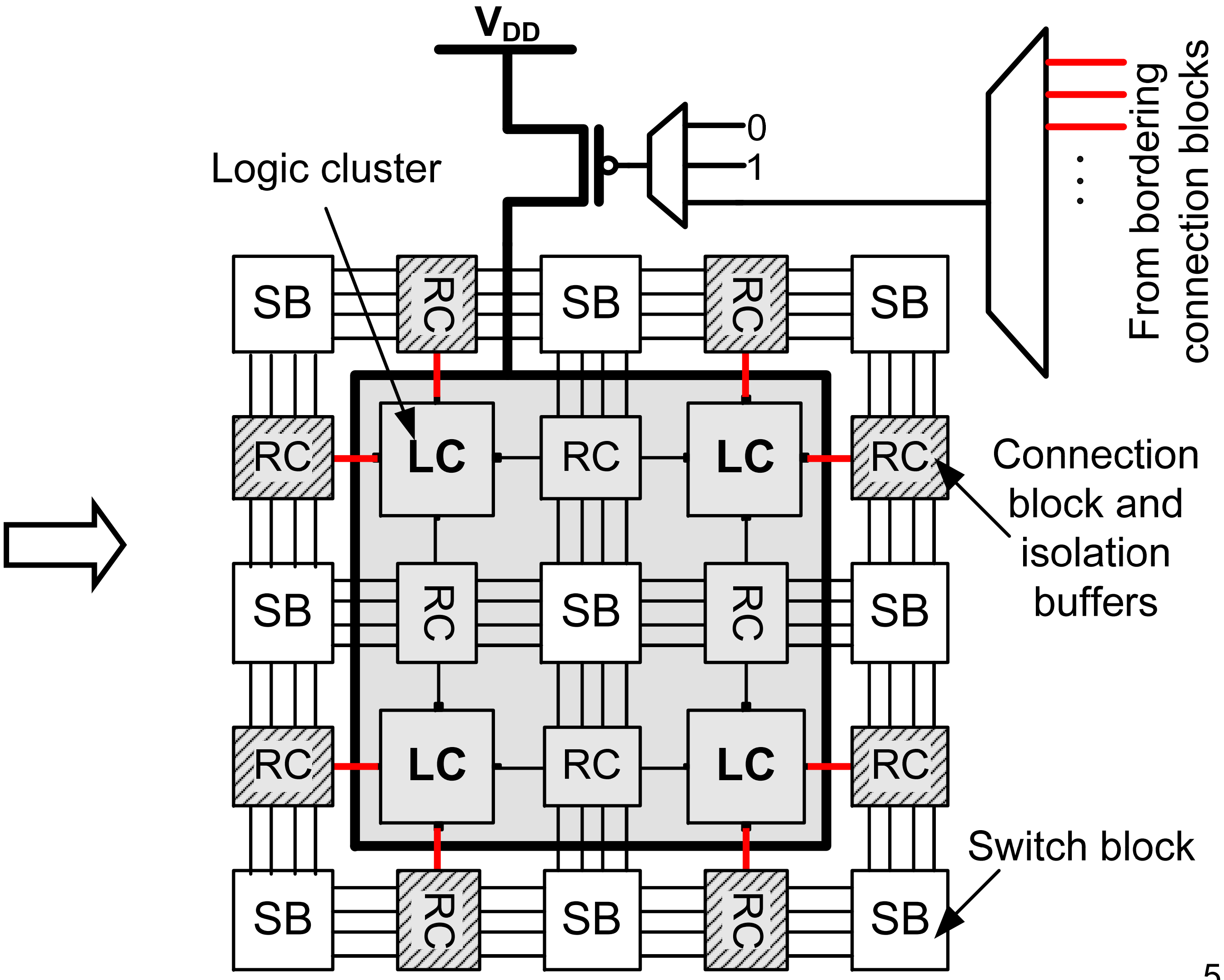
- Support dynamically-controlled sleep mode

Use general-purpose routing fabric for control signals



Our FPGA Architecture – One Region



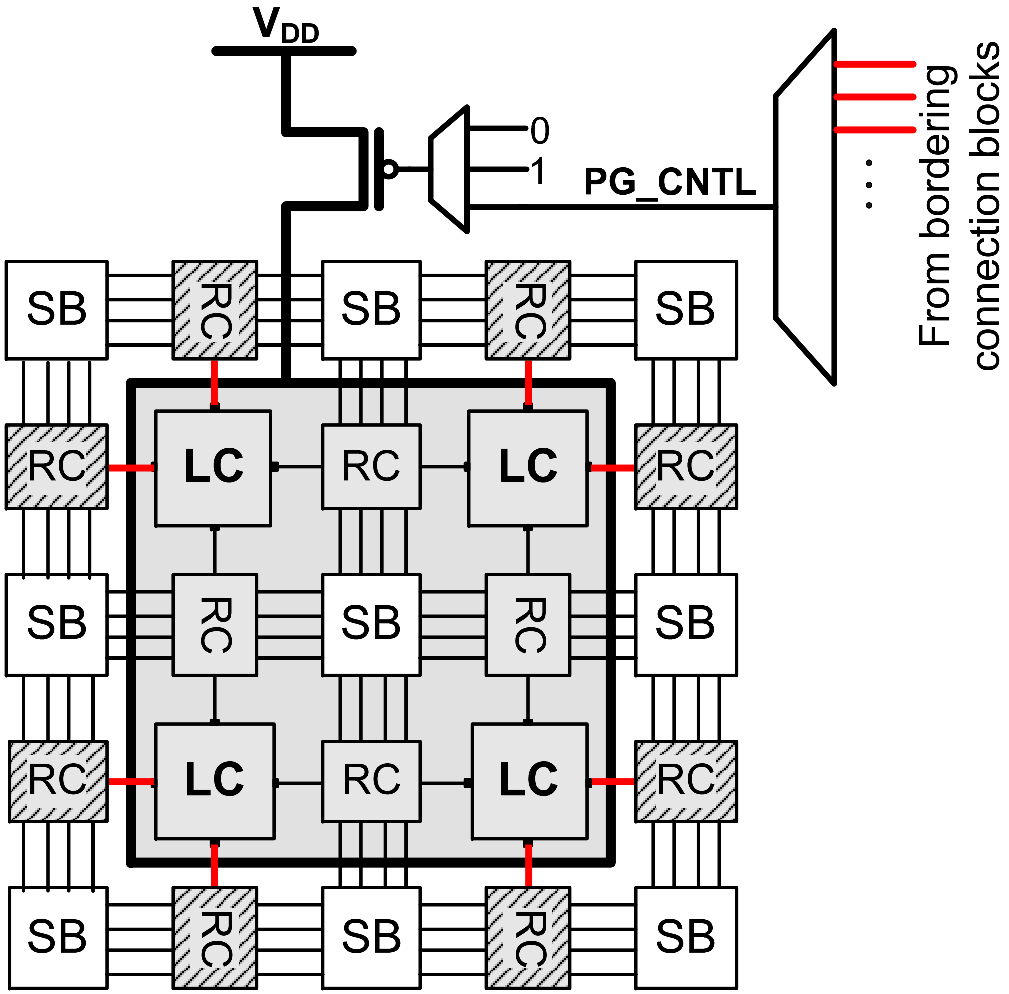


V_{DD}
0
1
Logic cluster
From bordering connection blocks
SB
RC
LC
Connection block and isolation buffers
Switch block

Our FPGA Architecture – One Region

Input pins from bordering routing channels are used to control region's power state.

Switch blocks are not power-gated ➔ a future work.



Power gating region, size 2x2

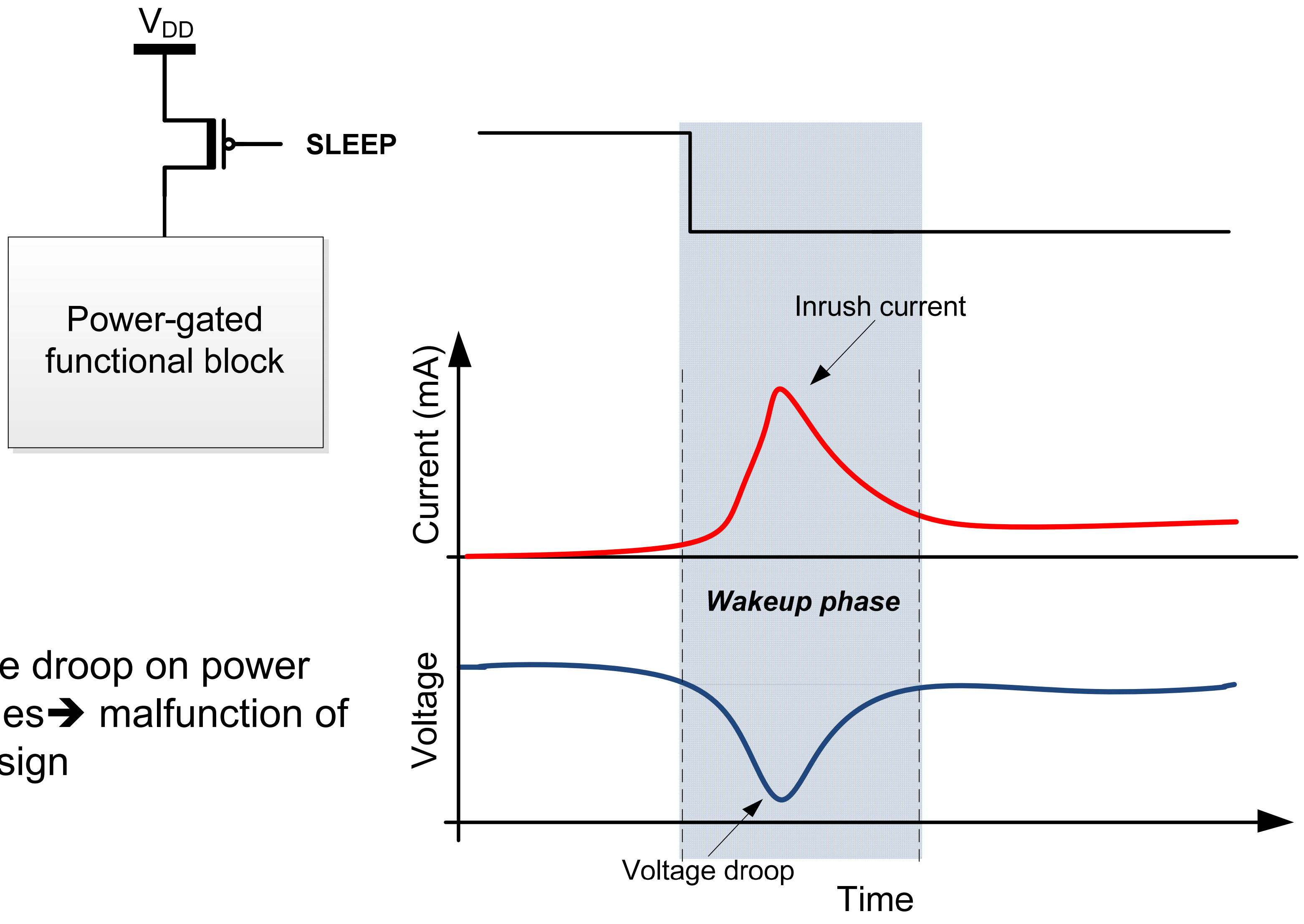
Our FPGA Architecture – Challenges

Challenges of the new FPGA architecture:

- **Inrush current during wakeup** ← **This talk**
- Power gating for switch blocks at run-time
- Routing of signals
- Mapping applications to our architecture (CAD)

What is the **inrush current** problem?

Inrush Current in Power Gating



Voltage droop on power grid lines➔ malfunction of the design

In power-gated ASIC designs, how the inrush current problem is solved?

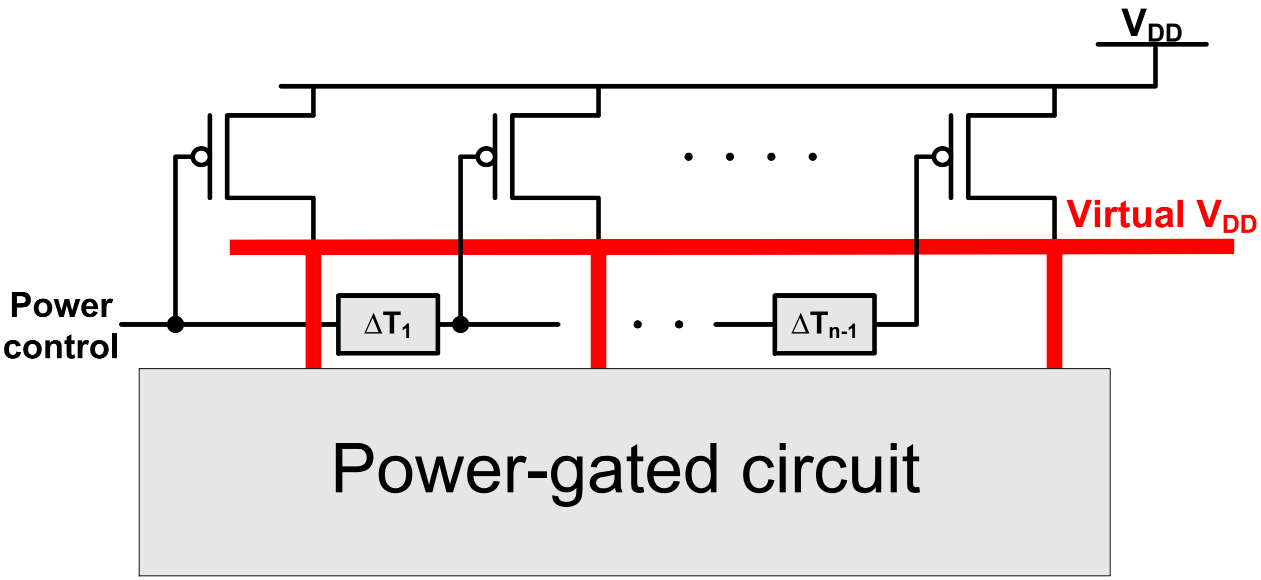
Related work – ASIC Domain

Turning on the power switch incrementally

- By controlling gate voltage of the sleep transistor (power switch).

Daisy chaining:

- A chain of parallel power switches – instead of one large switch
- Delay the wakeup of each stage to limit inrush current



How the problem is different in **our** FPGA?

Inrush current in ASIC vs. FPGA

In ASICs …

- Power gating is well known
- Application is known at fabrication time
- Inrush current requirements are known at design time

In FPGAs …

- Application is not known at fabrication time
- Sizes and locations of power-gated blocks are not known
- Inrush current requirements are not known

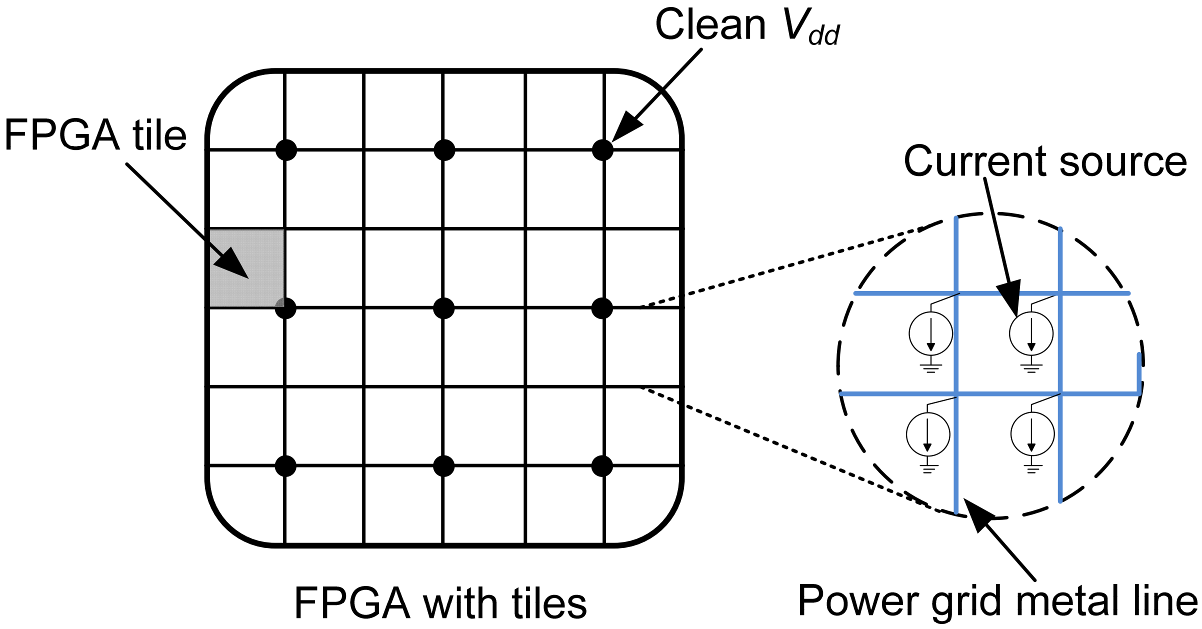
➔ ***The solution needs to be configurable!***

How serious the problem is in **our** FPGA?

FPGA Power Grid Modeling

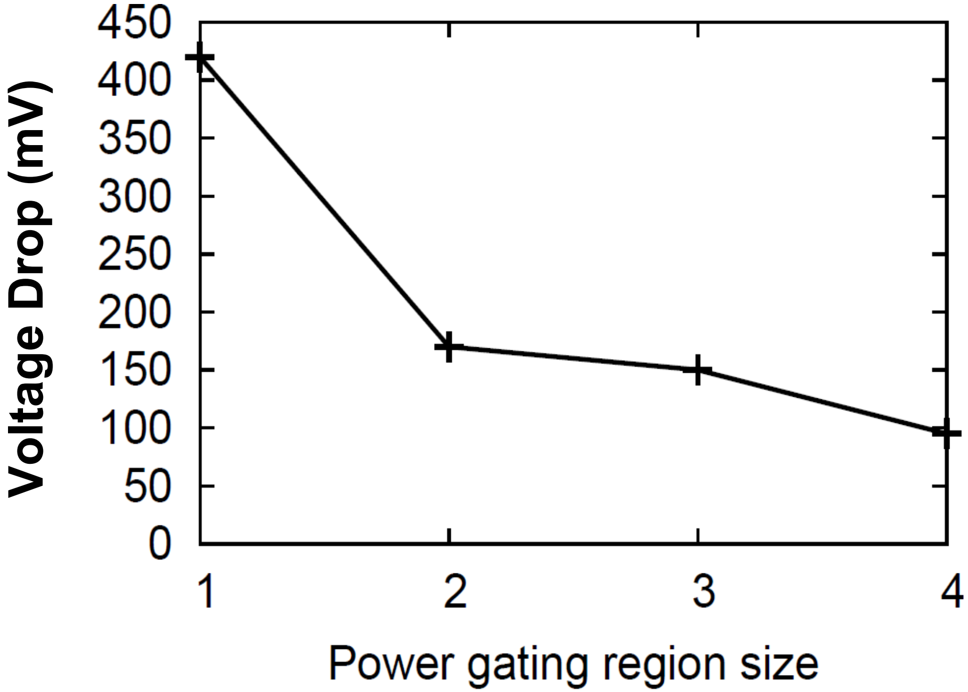
A model of the FPGA power grid to evaluate the effect of inrush current:

- Multiple metal layers that represent the power grid
- FPGA tiles modeled as a grid of current sources
- Obtain current values from power analysis of MCNC benchmarks (max average current per tile I_{max_tile} ≈ 400µA)



Effect of Inrush Current

Voltage drop due to inrush current in our FPGA architecture.



Finer granularity

➔ Voltage drop on power grid is larger than 100mV!

What are the possible solutions for **our** power-gated FPGA?

A Possible FPGA Solution

Ask the designer to take care of it!!

How?

Create a power controller that activates multiple signals in sequence to wakeup one functional block:

- Requires user experience and knowledge.
- Complicates design process.
- May result in power controller with large power consumption.

The Proposed Architecture …

A configurable architecture to limit inrush current.

Has two levels …

1) Fixed **intra-region** level:

- Ensures we can turn on individual regions safely.

2) Configurable **inter-region** level:

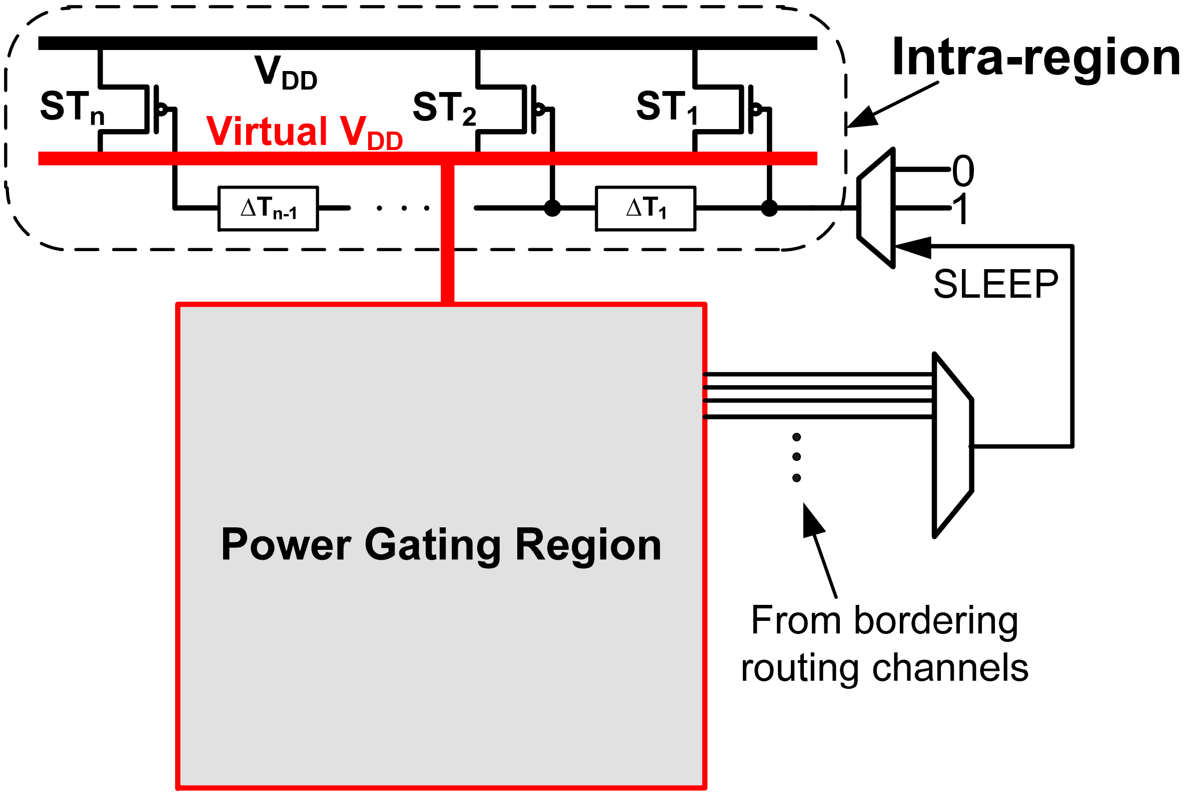
- Safely sequences the wakeup of a power-gated app.

Fixed *Intra-region* Architecture

Intra-region Architecture

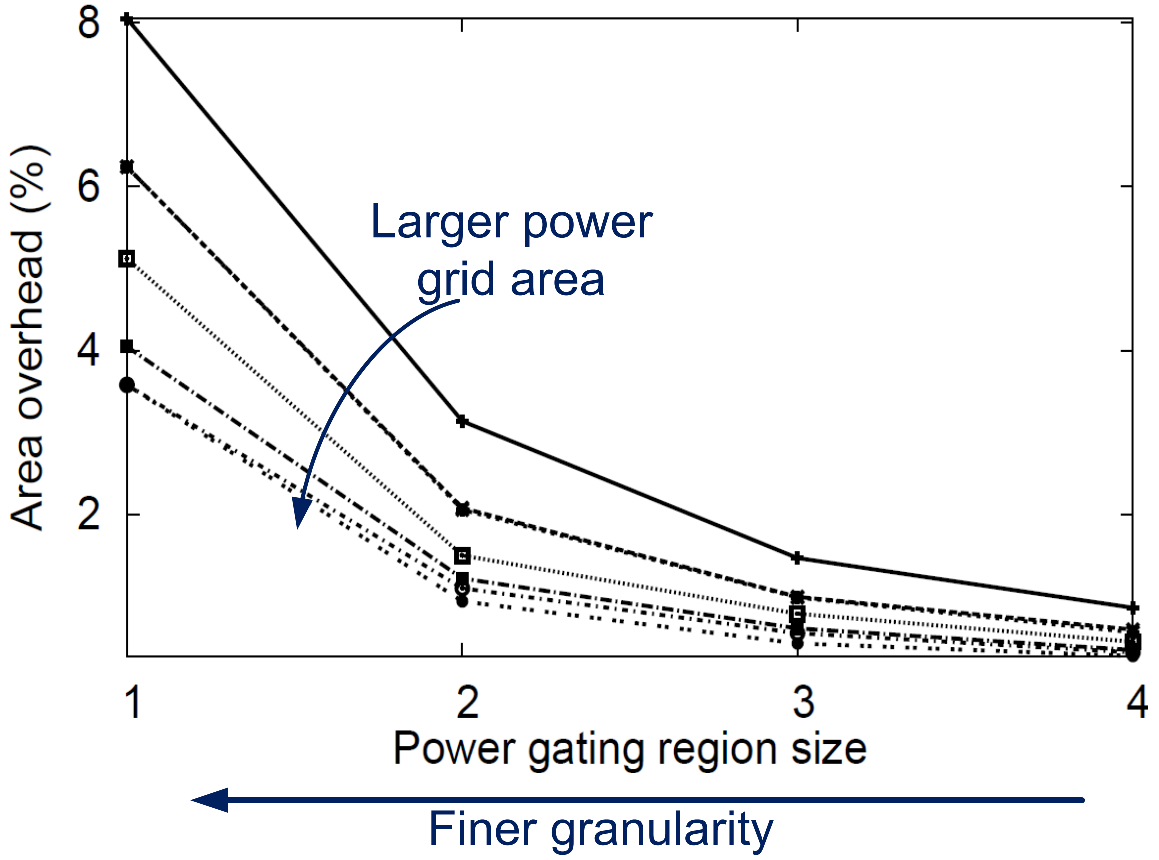
Use parallel sleep transistors (STs) and sequence wakeup using **fixed** delay elements.

Sizes of STs and delay elements is based on maximum allowed current.



A Tradeoff exists between area overhead of intra-region level and power grid metal area.

Area Overhead - Intra-Region Level



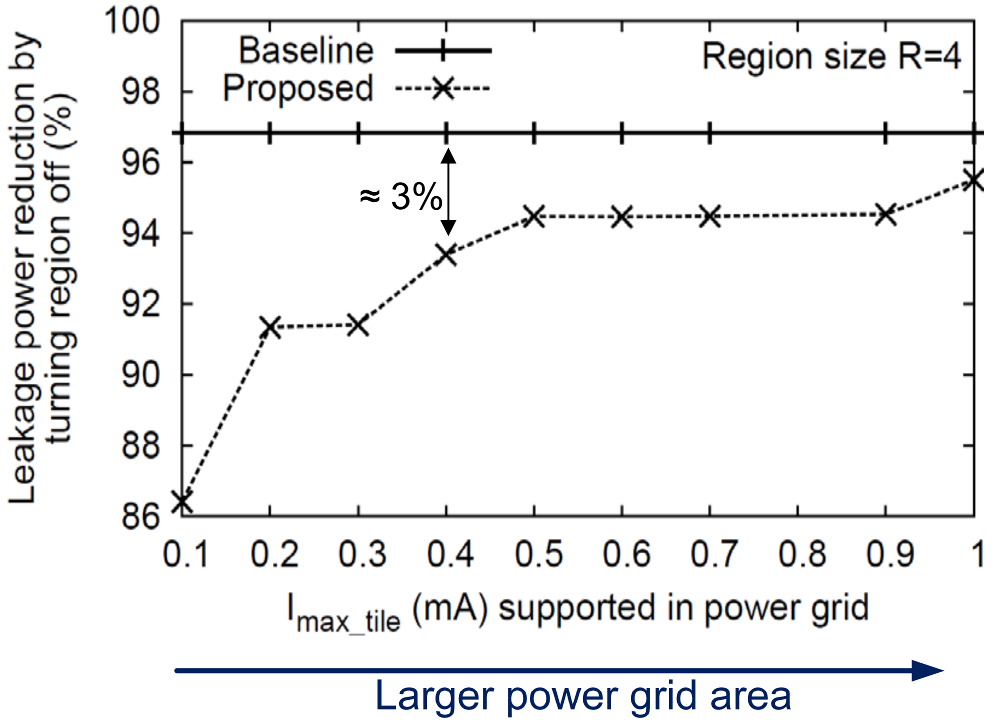
Larger power grid area

Finer granularity

- ➔ Area overhead is smaller for larger region sizes.
- ➔ Less than 2% for region size 3x3, and 1% for region size 4x4.

Leakage Power Savings

Baseline ≡ power gating arch. **without** inrush handling.



➔ Power savings:

≈ 97% for baseline

≈ 94% with intra-region architecture @ *Imax_tile* = 400μA

Configurable *Inter-region* Architecture

Inter-region Architecture

Intra-region level should solve the problem (theoretically!)

However …

In practice more current might be drawn due to:

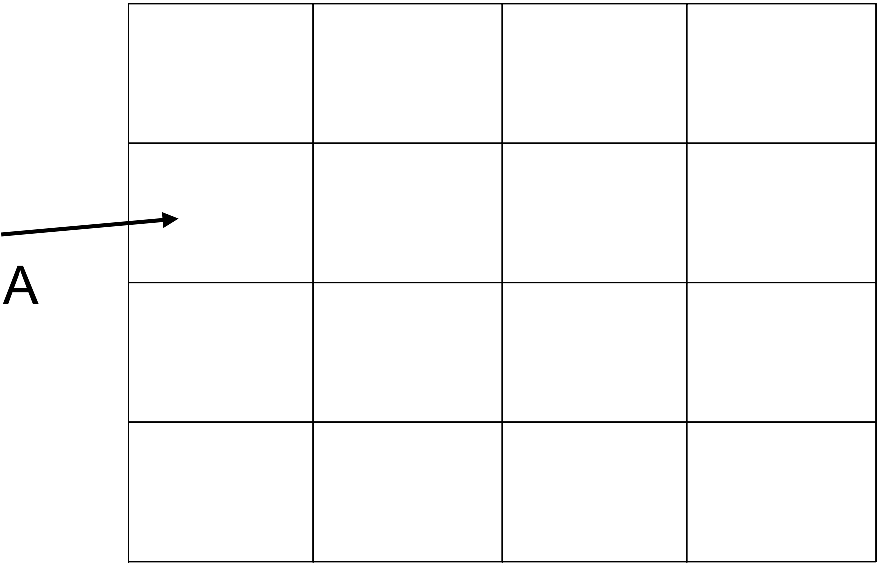
- Switching of unrelated signals in routing.
- Switching of inputs to logic clusters partially turned on.

Which is application specific!

➔Instead of turning on a complete power-gated application at once, turn on **one** power gating region at a time.

Inter-region Architecture

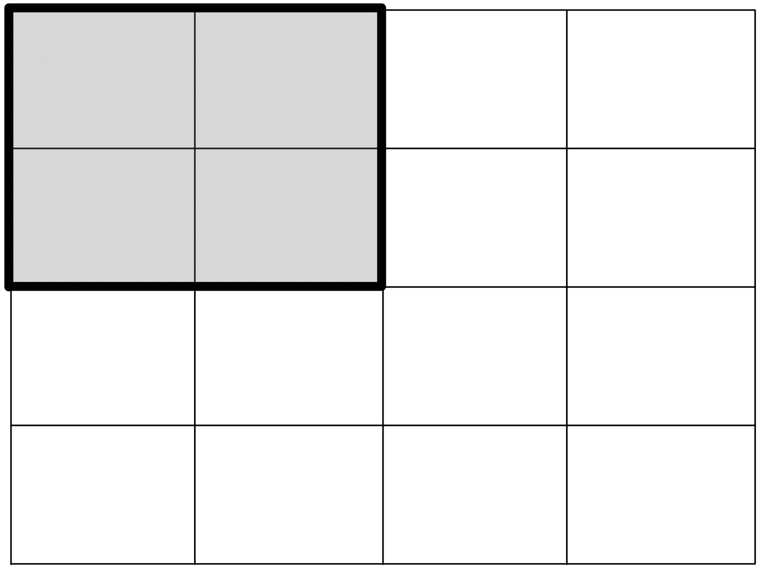
FPGA chip with power gating regions.



Power gating region in FPGA

Inter-region Architecture

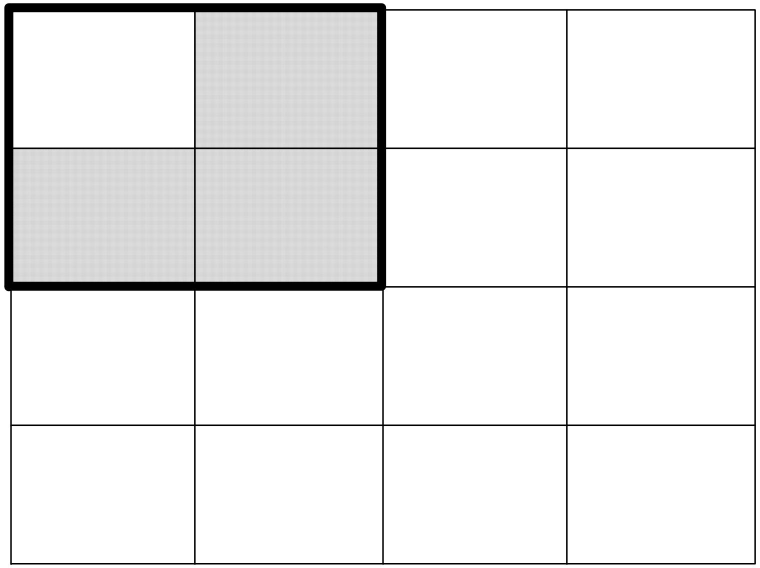
A power-gated app. is mapped to one or more regions.



Inter-region Architecture

During wakeup: turn on one region at a time.

After 1 x ΔT

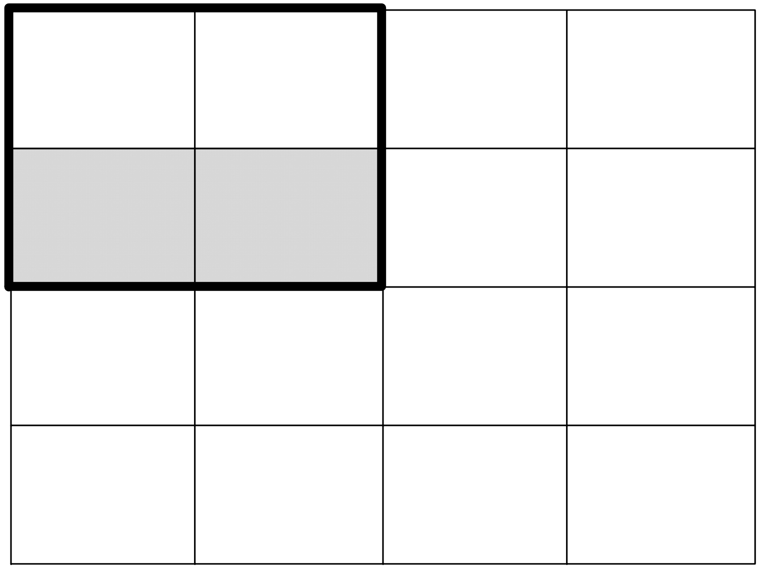


Delay is inserted before the turn on of next region.

Inter-region Architecture

During wakeup: turn on one region at a time.

After 2 x ΔT

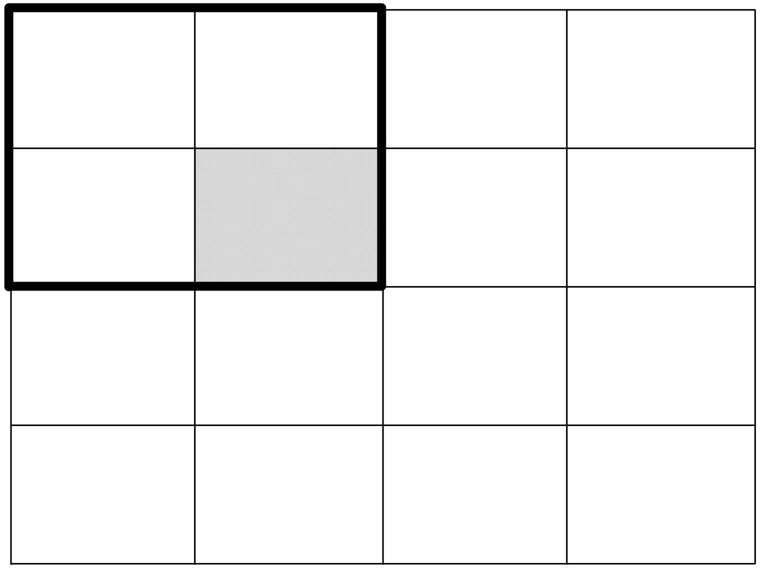


Delay is inserted before the turn on of next region.

Inter-region Architecture

During wakeup: turn on one region at a time.

After 3 x ΔT

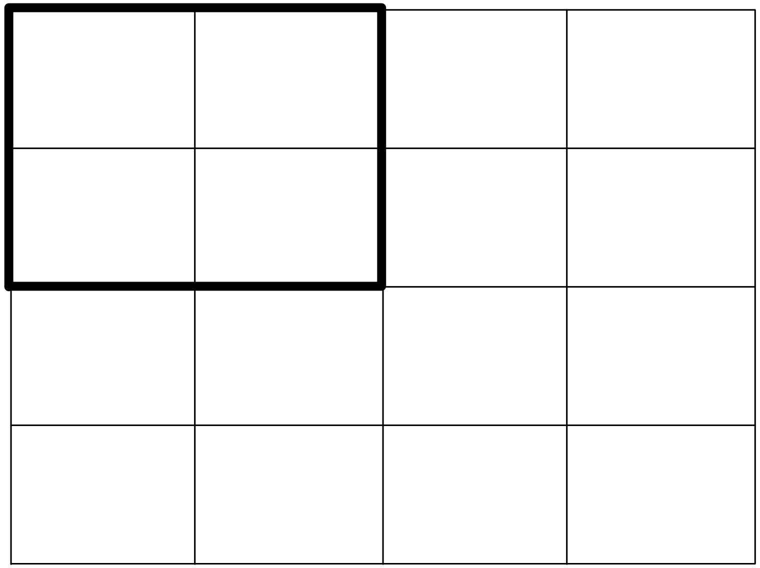


Delay is inserted before the turn on of next region.

Inter-region Architecture

During wakeup: turn on one region at a time.

After 4 x ΔT



Inter-region Architecture

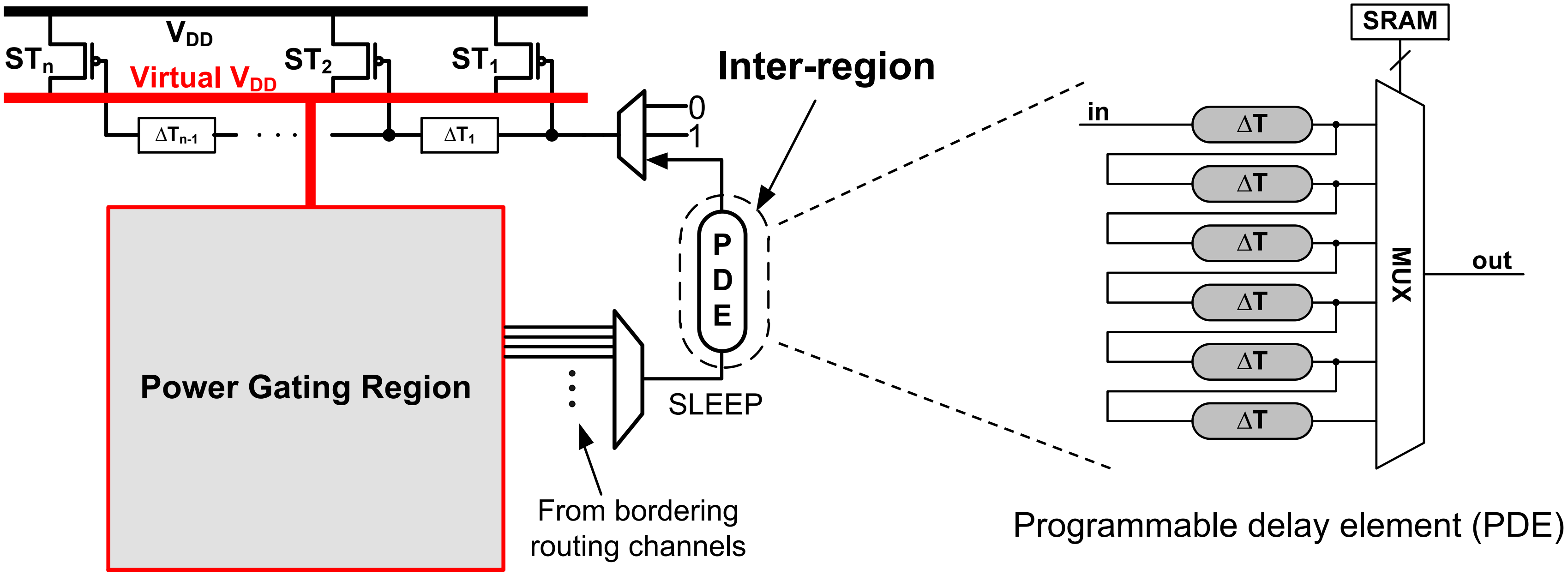
We need to insert delays between the turn-on of regions.

However … we don’t know:

- the shape, size, and location of a power-gated app. on FPGA.
- how many delays are required before turning on a region.

➔ Use programmable delay elements (PDEs).

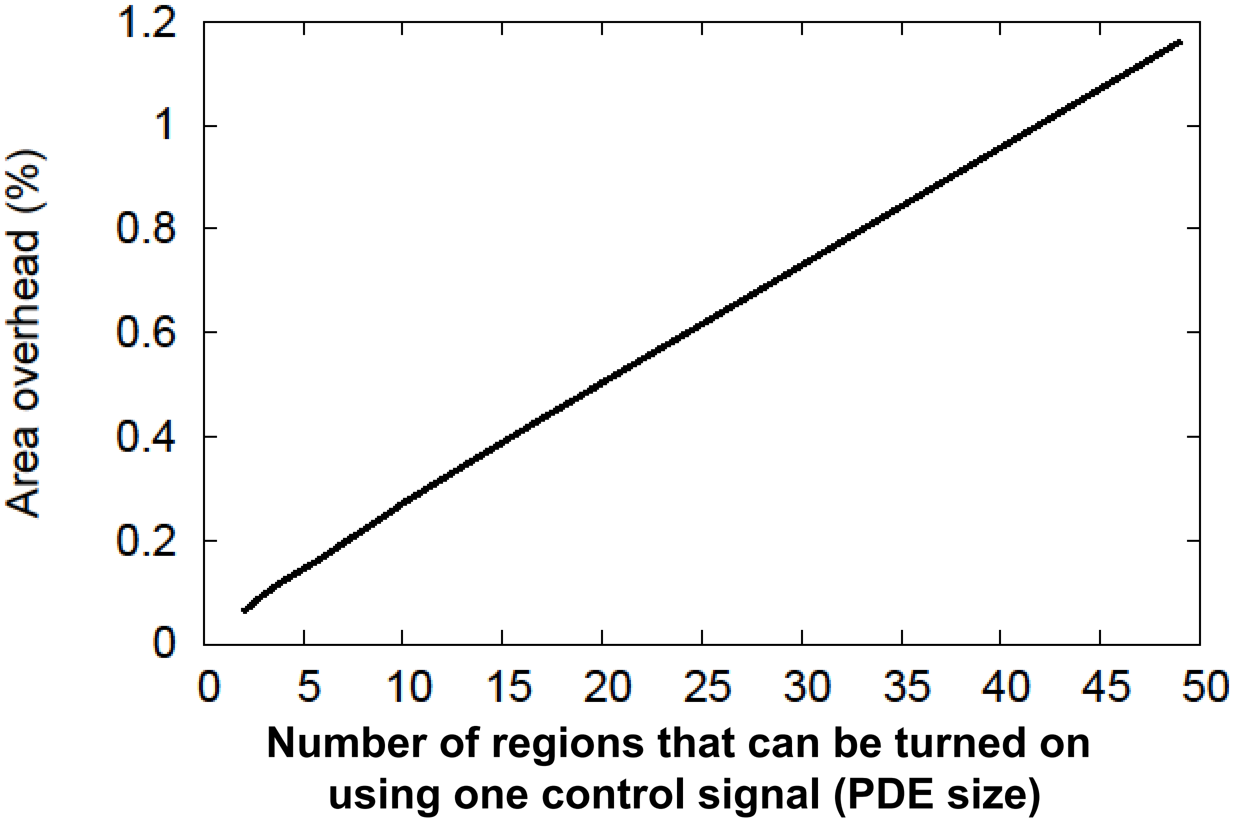
Inter-region Architecture



Programmable delay element (PDE)

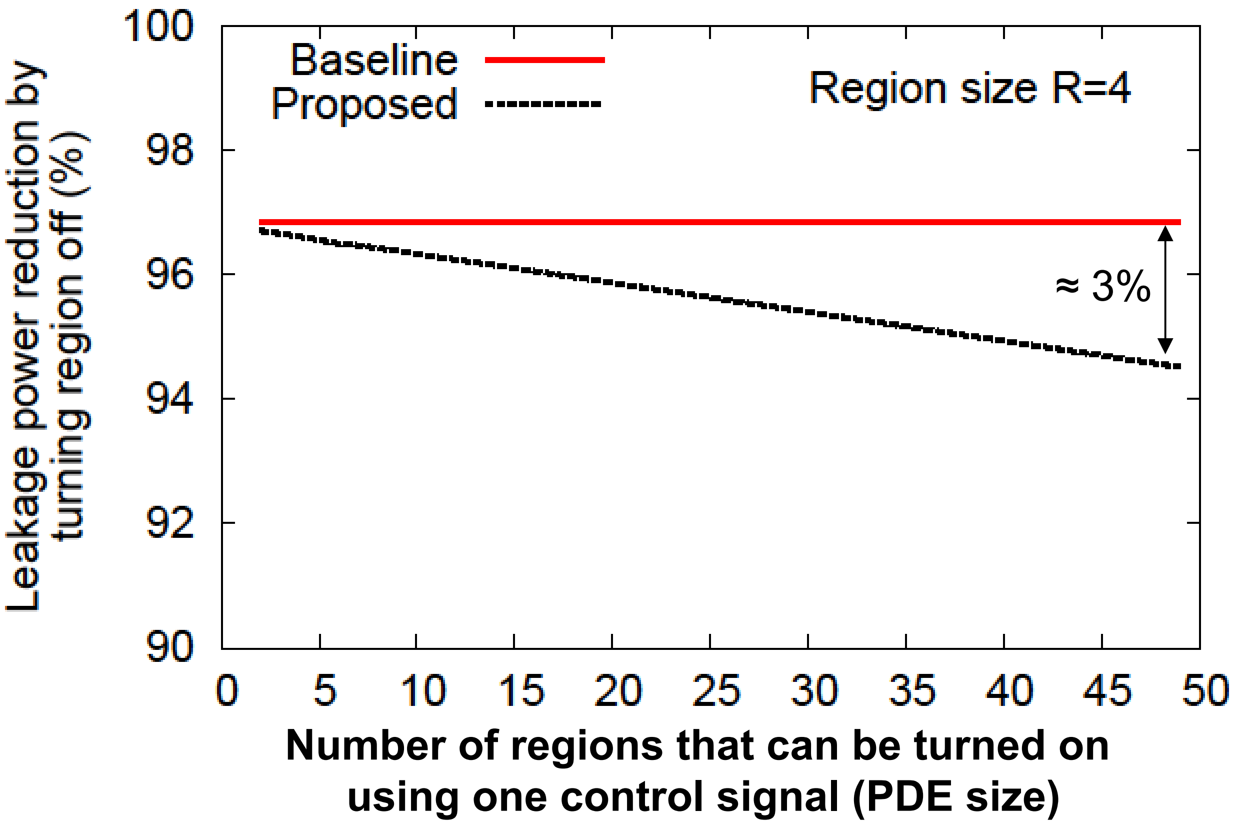
PDE size (number of inputs) ≡ the largest power-gated app. that can be turned on using one control signal.

Area Overhead – Inter-Region



Less than 1.2% area overhead!

Leakage Power Savings – Inter-Region



➔ Power savings:

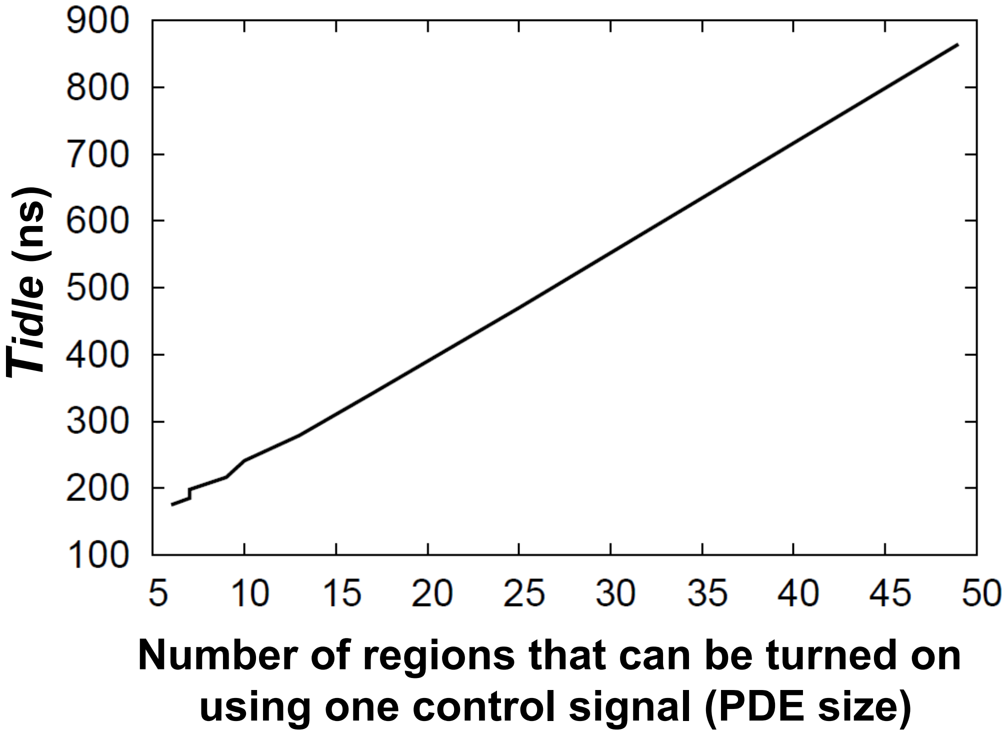
≈ 97% for baseline

≈ 94% with inter-region architecture @ PDE size = 49

Achieving Energy Savings

Tidle: minimum idle time of an app. in order to achieve energy savings.

Using a model to find *Tidle*:



For *clma* (775 tiles ≡ PDE size 49), *Tidle* ≈ 900 ns.

- Equivalent to 450 cycles on 500 MHz clock!

Summary

Inrush current is a problem in DCPG FPGAs.

A configurable architecture is proposed.

- A fixed intra-region level
- A configurable inter-region level

Area and power overheads are small:

- Less than 2% area overhead
- More than 90% leakage power savings during sleep mode (compared to 97% savings without inrush handling circuit)

Future Work

Power gating for switch blocks at run-time.

- They consume more than 50% of the leakage!

Inrush current.

- Revisiting for switch boxes power gating.

Routing control signals:

- Some switch boxes are turned off.
- The router should be aware of the new constraints.

Mapping applications to our architecture:

- Automatically detect blocks in an app. that can be power-gated.
- Automatically generate power controllers.