Becoming More Tolerant: Designing FPGAs for Variable Supply Voltage

Ibrahim Ahmed
Linda Shen
Vaughn Betz

The Edward S. Rogers Sr. Department
of Electrical & Computer Engineering
UNIVERSITY OF TORONTO

Technology Scaling: Transforming the World

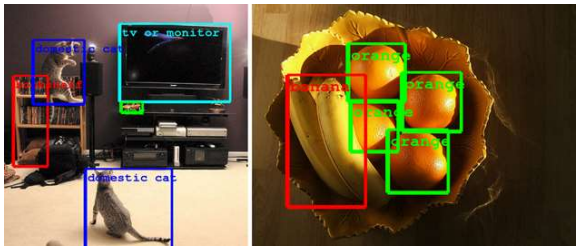
- Packing ever more computations on a single chip

Technology Scaling: Transforming the World

- Packing ever more computations on a single chip

Technology Scaling: Transforming the World

- Packing ever more computations on a single chip



Technology Scaling: The Other Side

- Huge energy demand
  - Data centers consumed 2% of total US electricity, 2014[a]
  - ICT sector to consume 9-20% of global electricity, 2025[b]

[a] N. Jones. How to stop data centres from gobbling up the worlds electricity. Nature, 561:163-166, 09 2018.
[b] A. Shehabi et al. United States Data Center Energy Usage Report. Lawrence Berkeley National Laboratory, Berkeley, California., 2016.

Technology Scaling: The Other Side

- Huge energy demand
  - Data centers consumed 2% of total US electricity, 2014[a]
  - ICT sector to consume 9-20% of global electricity, 2025[b]

- Many devices are power constrained
  - Mobile/edge
  - Cellular base station, satellites, etc.

Shelley, a self-driving Audi TT developed by Stanford University, uses the brains in the trunk to speed around a racetrack autonomously.

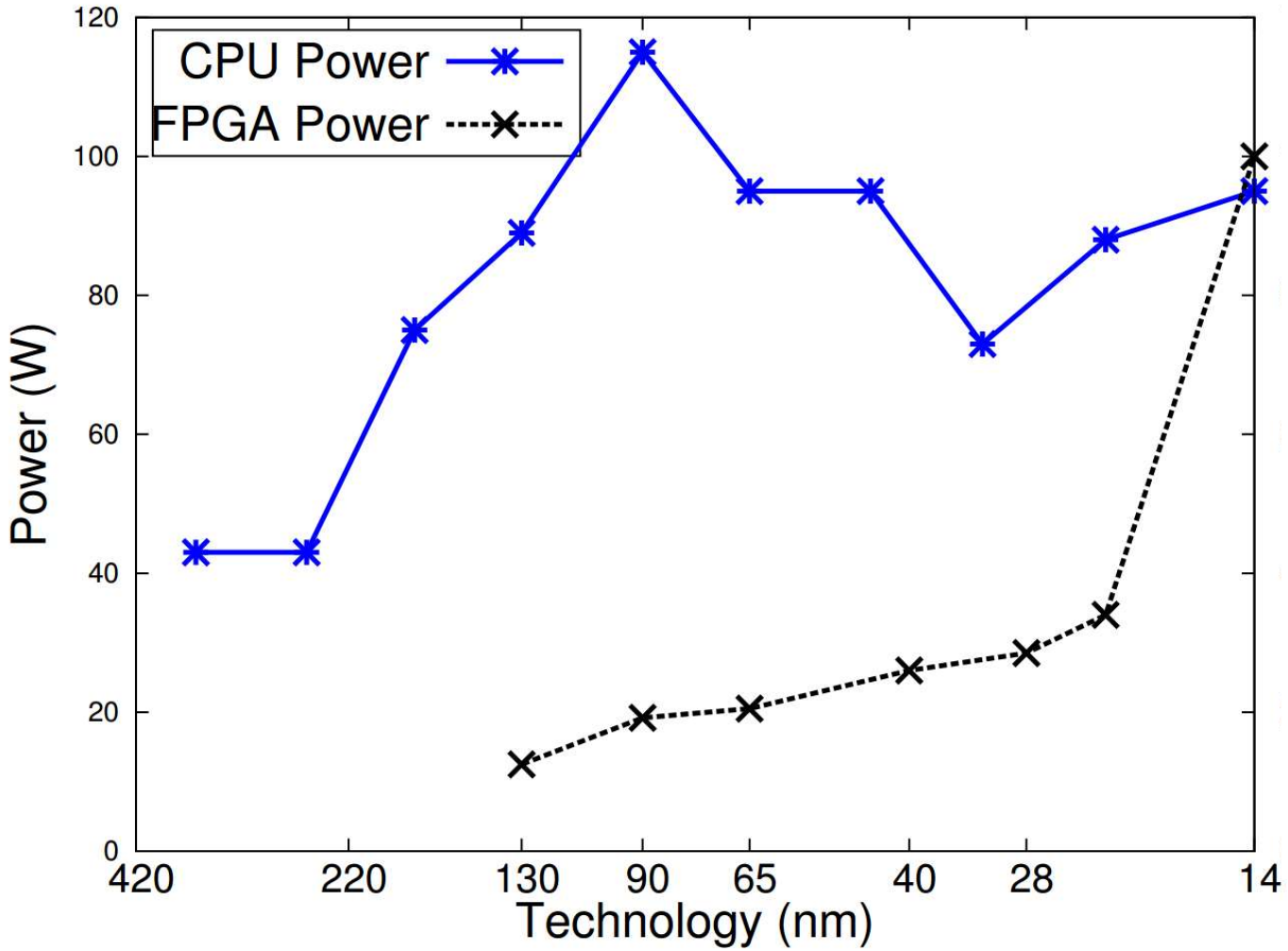
The Edward S. Rogers Sr. Department of Electrical & Computer Engineering
UNIVERSITY OF TORONTO

[a] N. Jones. How to stop data centres from gobbling up the worlds electricity. Nature, 561:163-166, 09 2018.
[b] A. Shehabi et al. United States Data Center Energy Usage Report. Lawrence Berkeley National Laboratory, Berkeley, California., 2016.

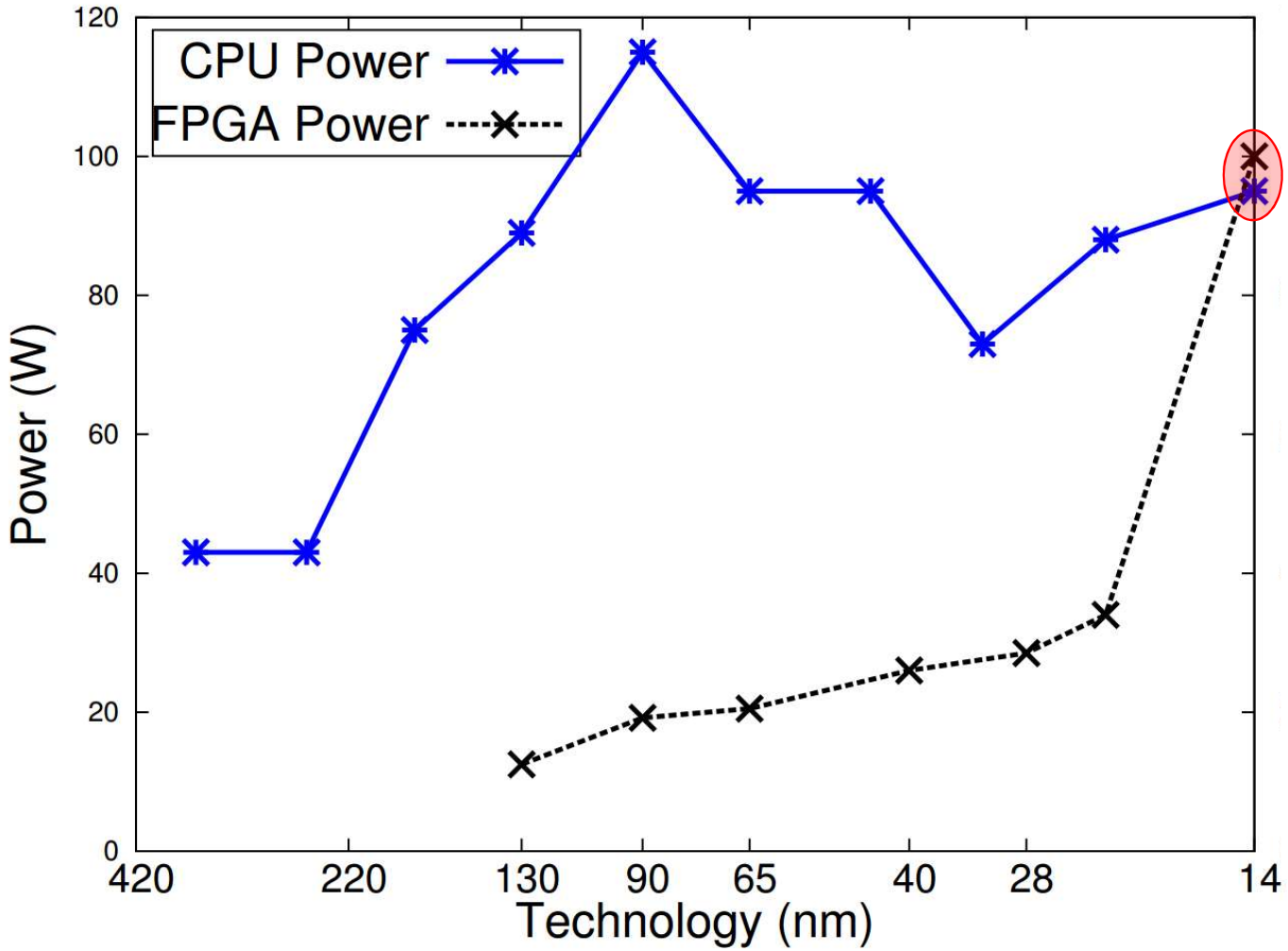
Moving Away from General-Purpose Processors

- FPGAs → trade-off between flexibility and efficiency
  - Users can build custom digital systems without the ASIC challenges
    - Not as power efficient as ASICs
  - Offer better performance/W than CPUs for many applications
  - Known to have lower absolute power than CPUs
  - Adopted in Microsoft, Baidu, and Amazon data centres

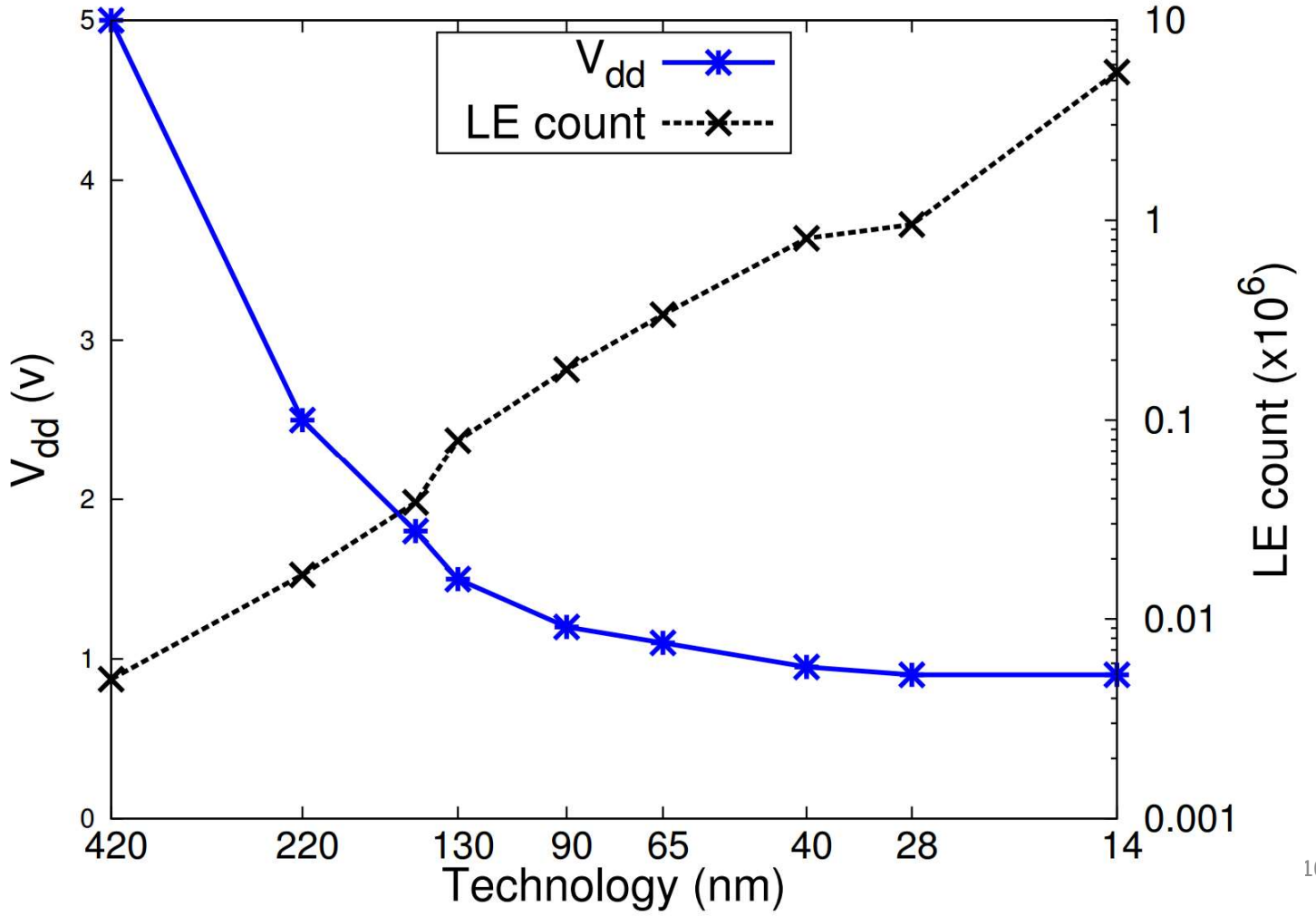
FPGA Power Consumption Challenge



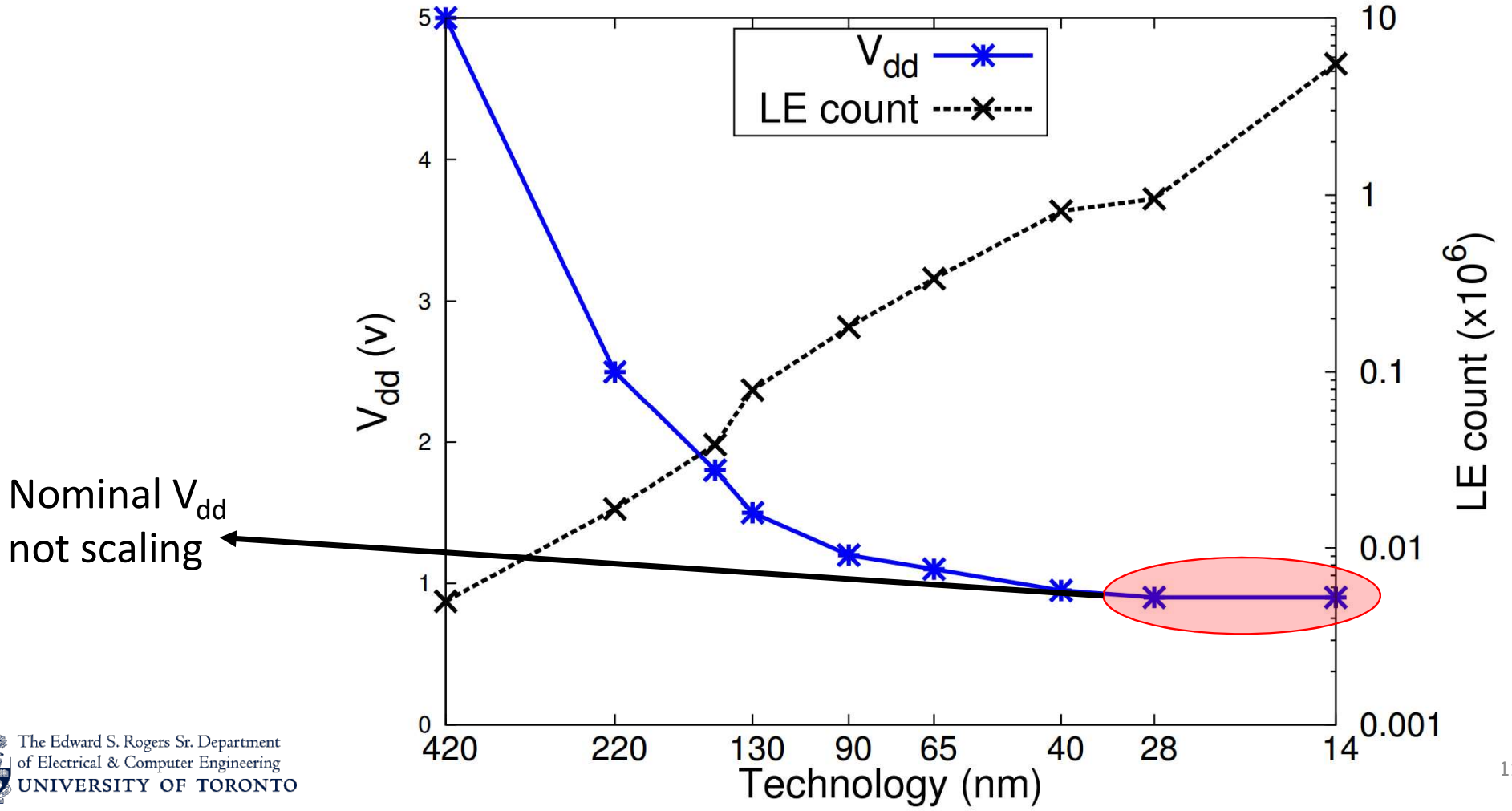
FPGA Power Consumption Challenge



What Happened?



What Happened?



Adaptive & Dynamic Voltage Scaling (DVS)

- Academic work on DVS
  - Set supply voltage (V_{dd}) dynamically → no longer fixed to nominal
  - Previous works have shown ~30% power reduction

Adaptive & Dynamic Voltage Scaling (DVS)

- Academic work on DVS
  - Set supply voltage (V_{dd}) dynamically → no longer fixed to nominal
  - Previous works have shown ~30% power reduction

- Intel SmartVID (adaptive voltage scaling)
  - Each FPGA stores it's own supply voltage value → determined during testing
  - Smart power supply sets the supply voltage based on the stored value

FPGA	Arria 10	Stratix 10	Agilex
Range (V)	0.85-0.9	0.8-0.94	0.6-1

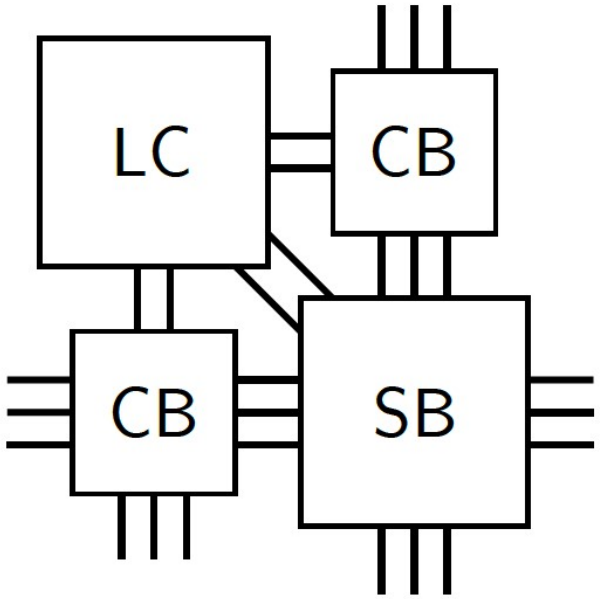
Rethinking FPGAs for Variable Supply Voltage

- FPGAs moving away from fixed nominal-V_{dd} operation
- But, FPGAs have always been designed for fixed-V_{dd}
- Goals:
  - Evaluate the delay sensitivity of existing FPGA circuits to V_{dd}
  - Design FPGAs that are better suited for variable V_{dd}

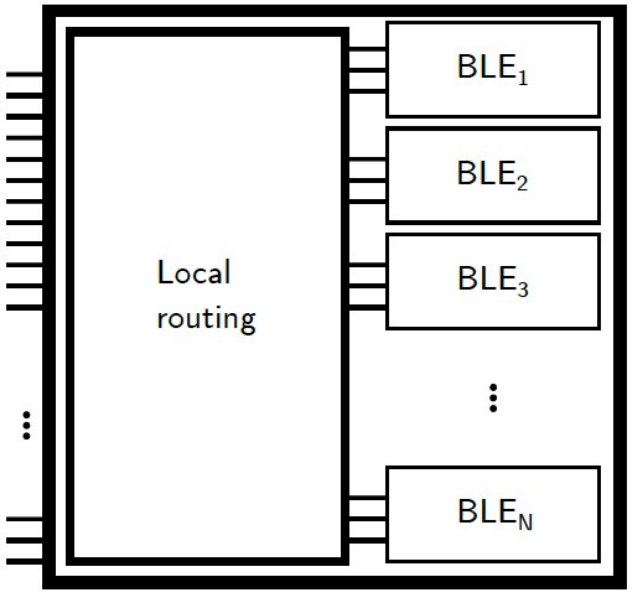
Outline

- Background
- Analyzing Existing FPGA building blocks (logic and routing)
- VPR analysis over benchmarks
- Designing new LUTs
- Summary and Future Work

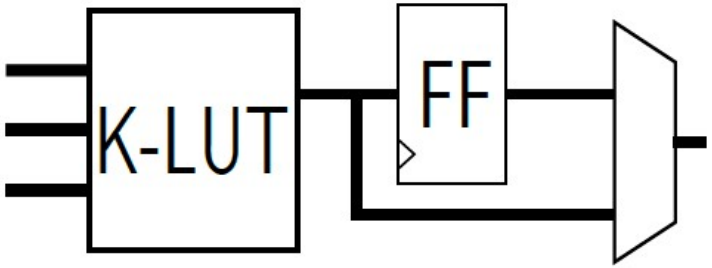
Background: Island-style FPGA Architecture



Representative FPGA tile

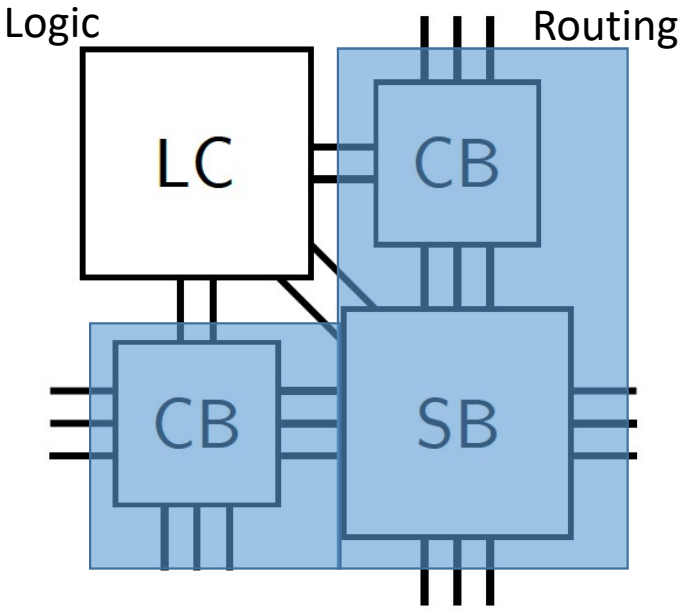


Logic Cluster (LC)

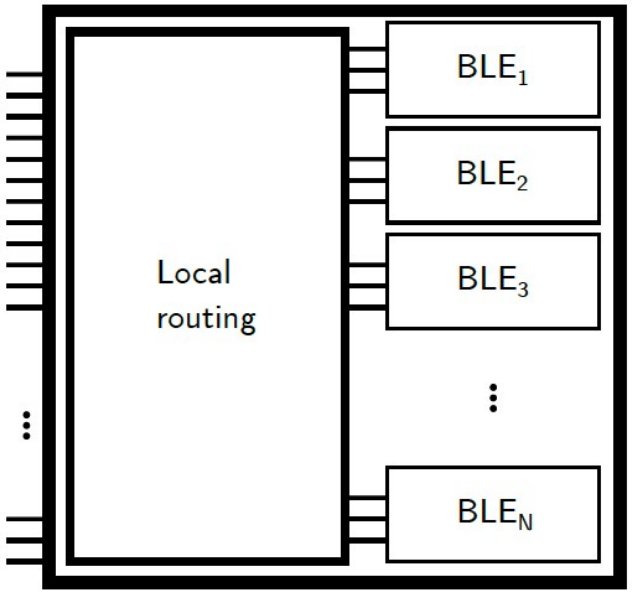


Basic Logic Element (BLE)

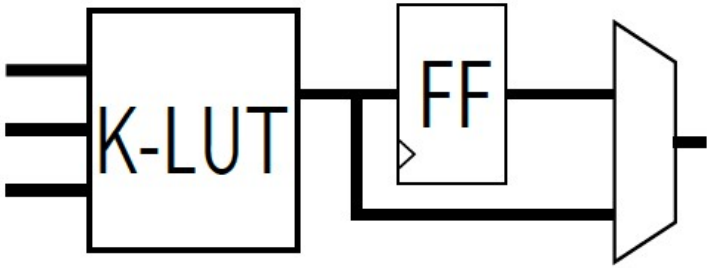
Background: Island-style FPGA Architecture



Representative FPGA tile

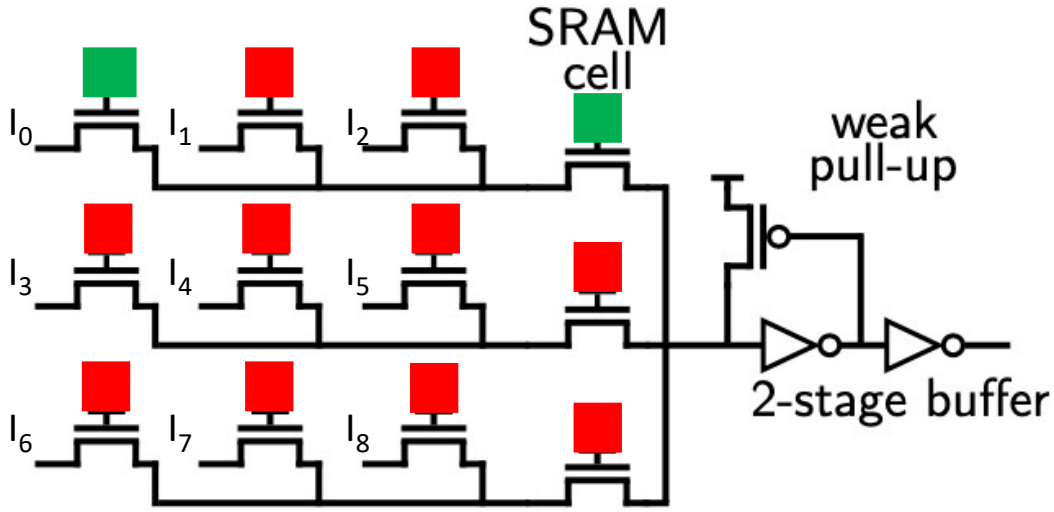


Logic Cluster (LC)



Basic Logic Element (BLE)

Background: Conventional FPGA Routing MUX

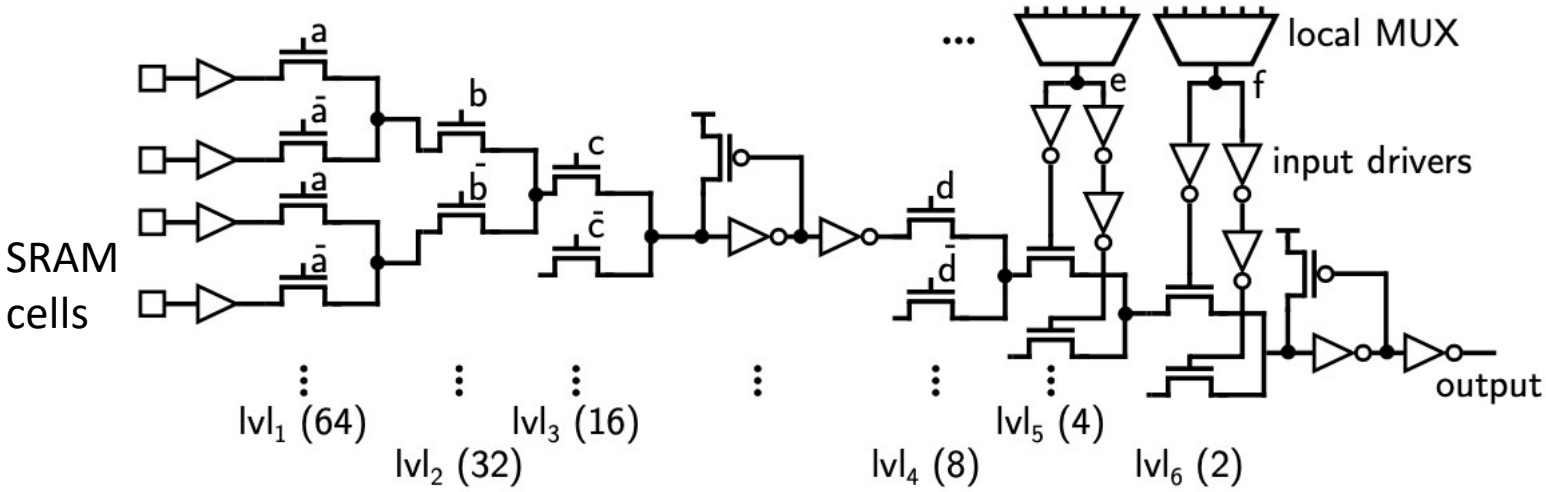


9-input two-stage multiplexer

SRAM cell storing 1

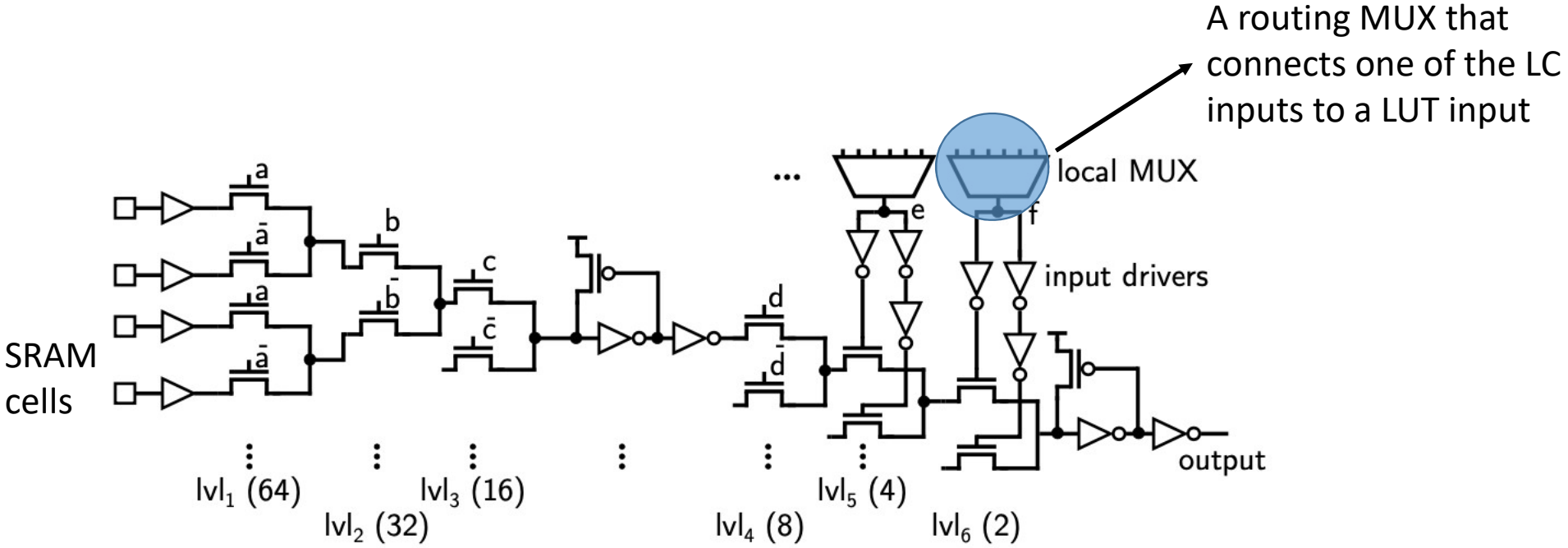
SRAM cell storing 0

Background: Conventional LUT Circuitry



Tree-based 6-input LUT multiplexer

Background: Conventional LUT Circuitry

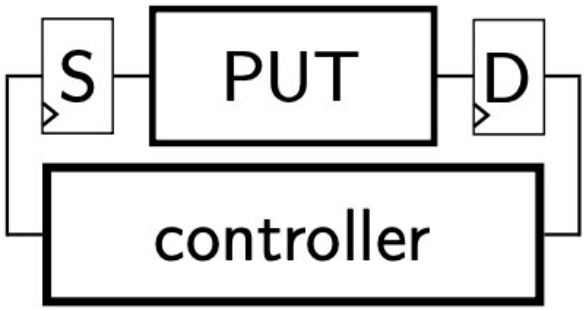


Tree-based 6-input LUT multiplexer

Outline

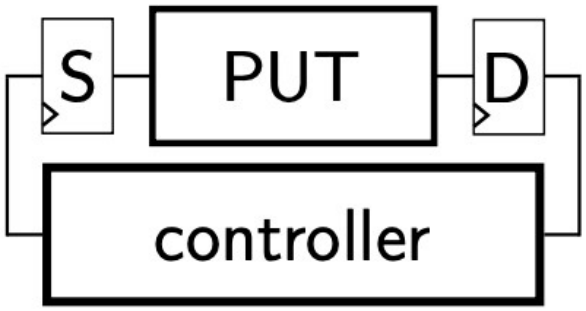
- Background
- Analyzing Existing FPGA building blocks (logic and routing)
- VPR analysis over benchmarks
- Designing new LUTs
- Summary and Future Work

Analyzing Existing FPGAs: Block-level (Silicon Measurements)

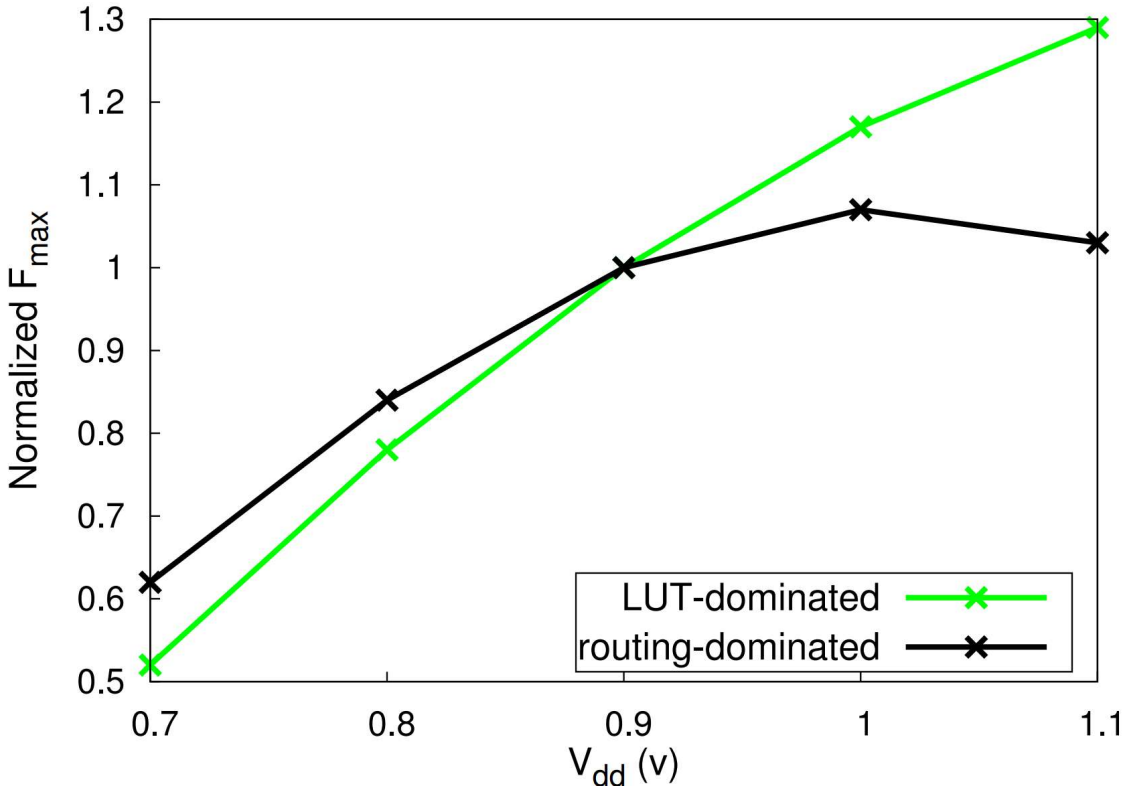


Setup to measure path delays on a Stratix V FPGA

Analyzing Existing FPGAs: Block-level (Silicon Measurements)

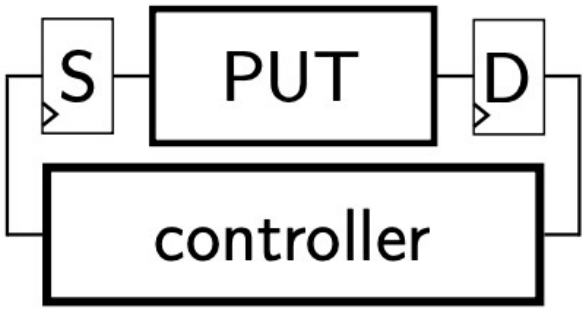


Setup to measure path delays on a Stratix V FPGA

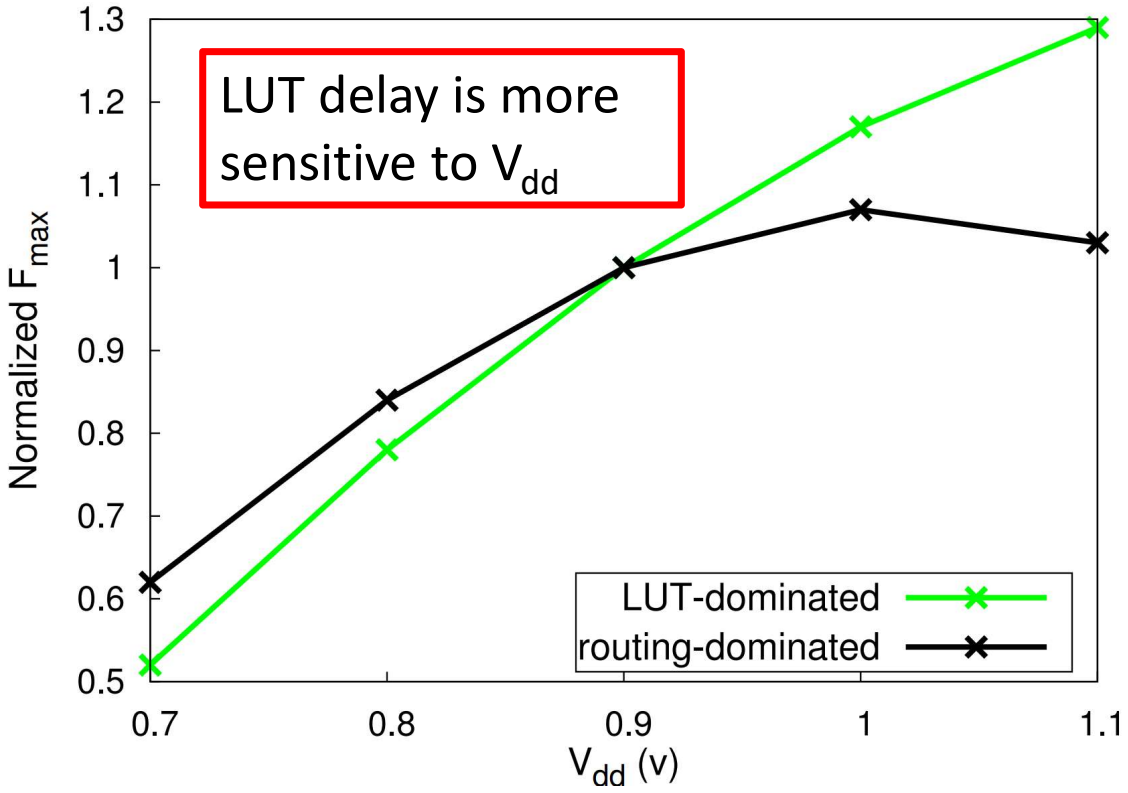


Measuring different types of paths on Stratix V

Analyzing Existing FPGAs: Block-level (Silicon Measurements)

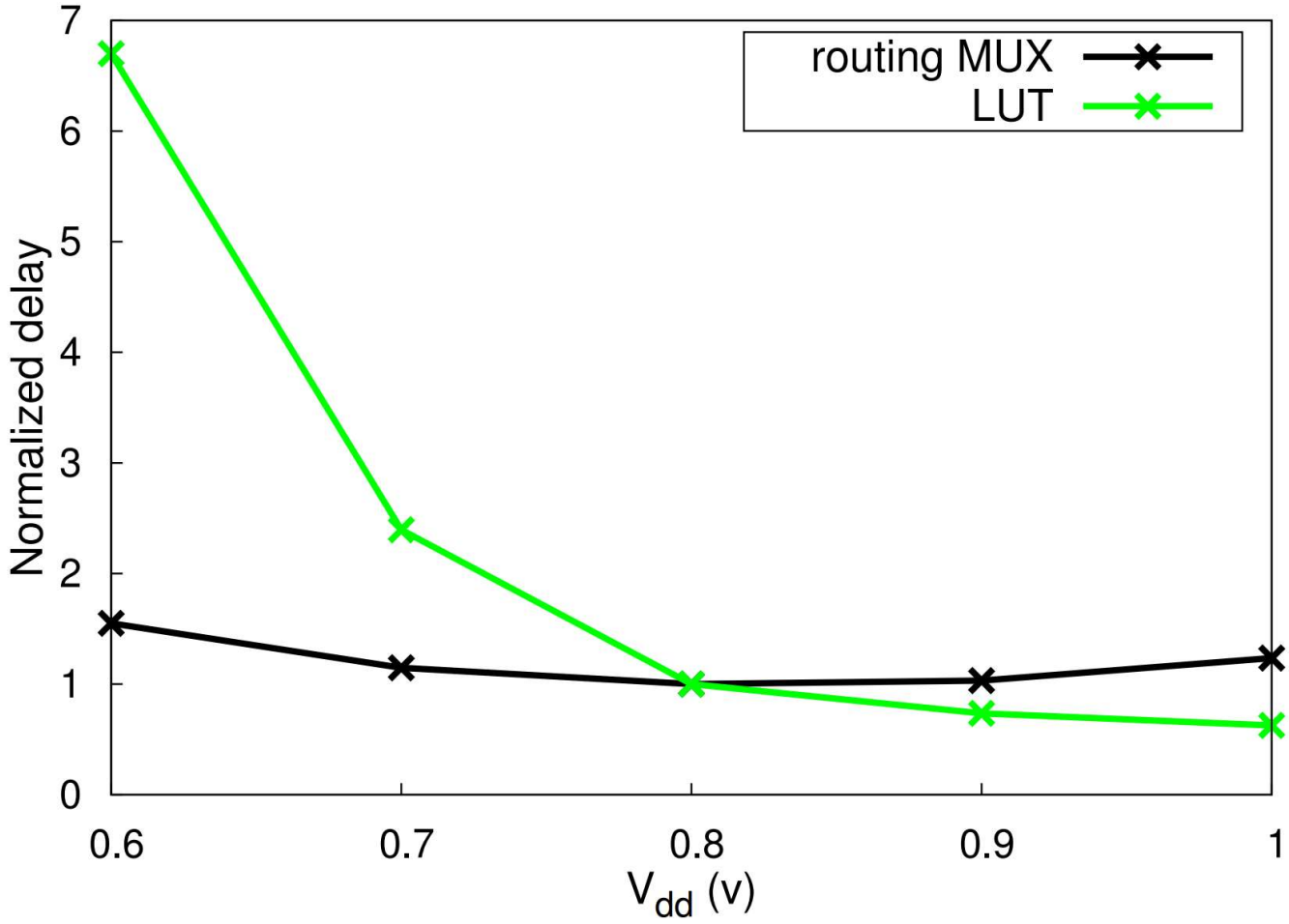


Setup to measure path delays on a Stratix V FPGA

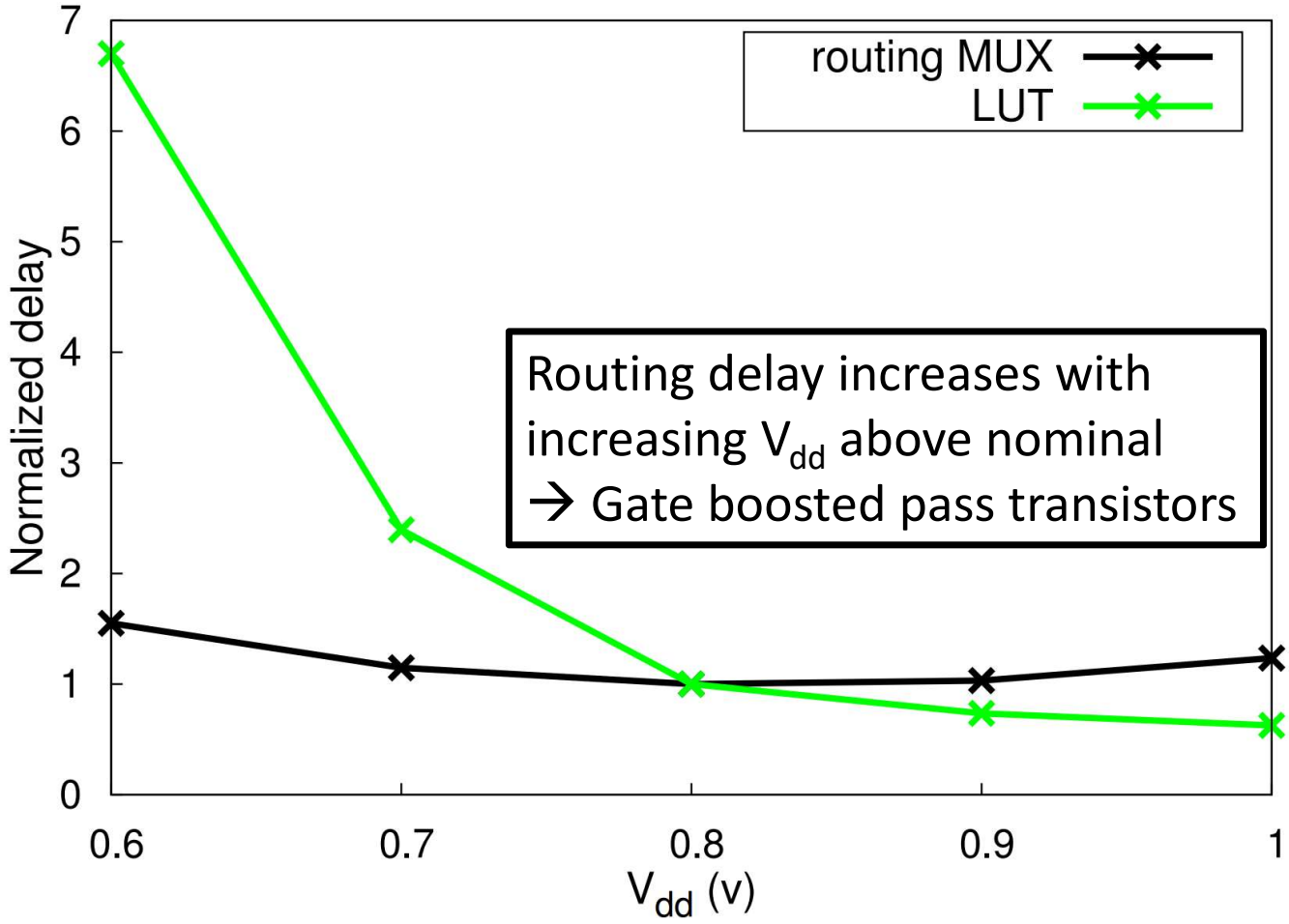


Measuring different types of paths on Stratix V

Analyzing Existing FPGAs: Block-level (Spice Simulations)

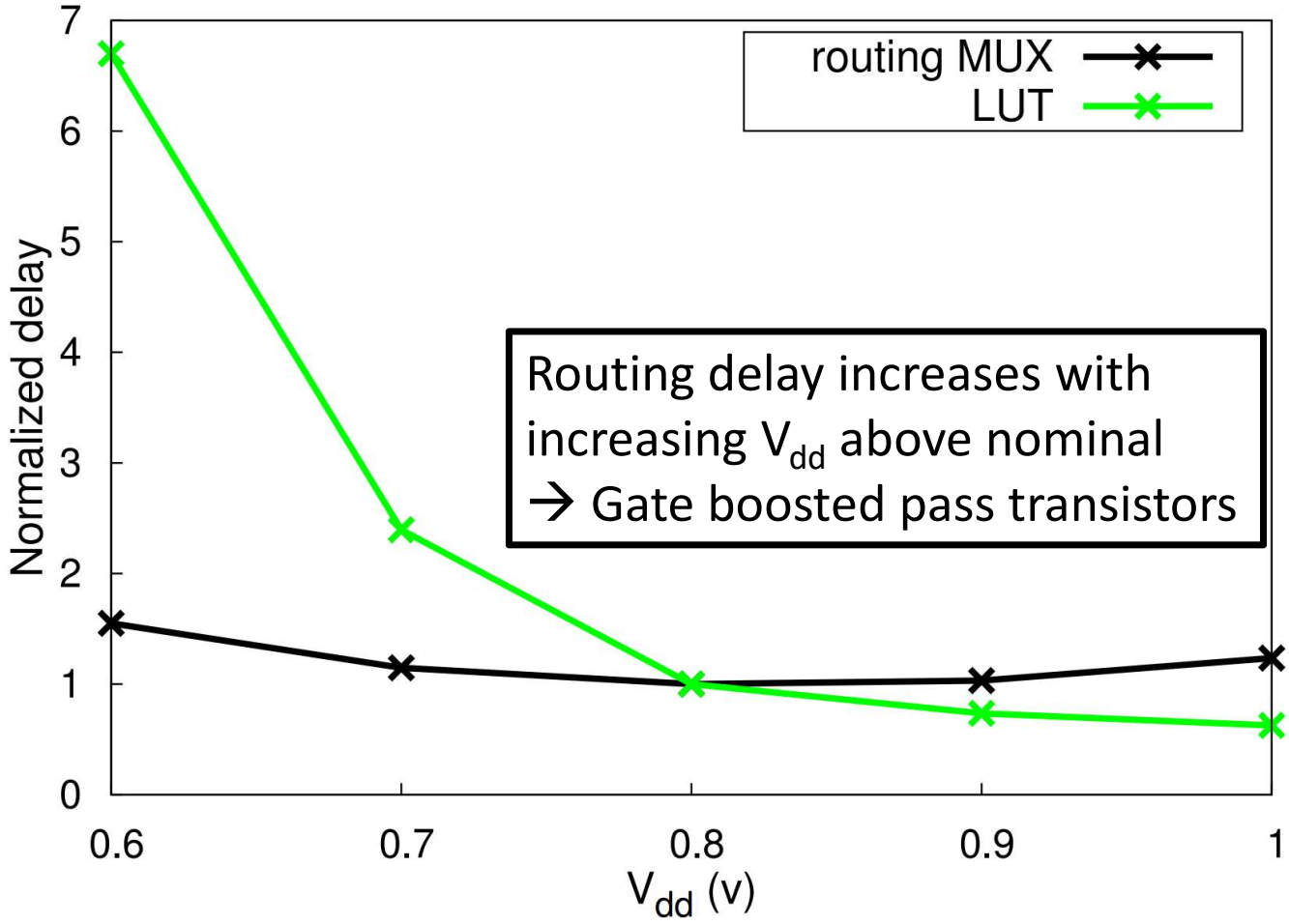


Analyzing Existing FPGAs: Block-level (Spice Simulations)



Analyzing Existing FPGAs: Block-level (Spice Simulations)

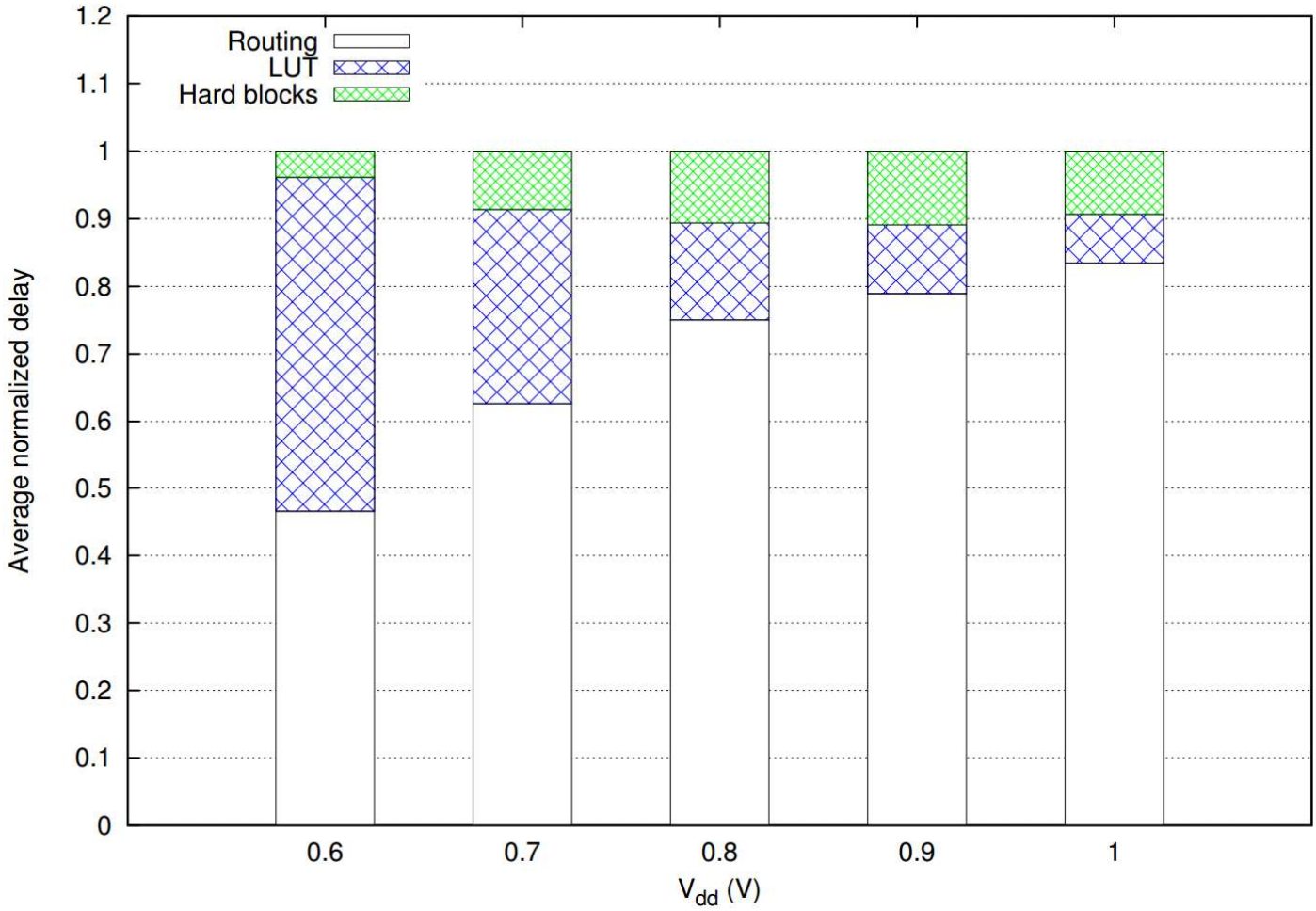
LUTs get much slower at lower V_{dd}



Outline

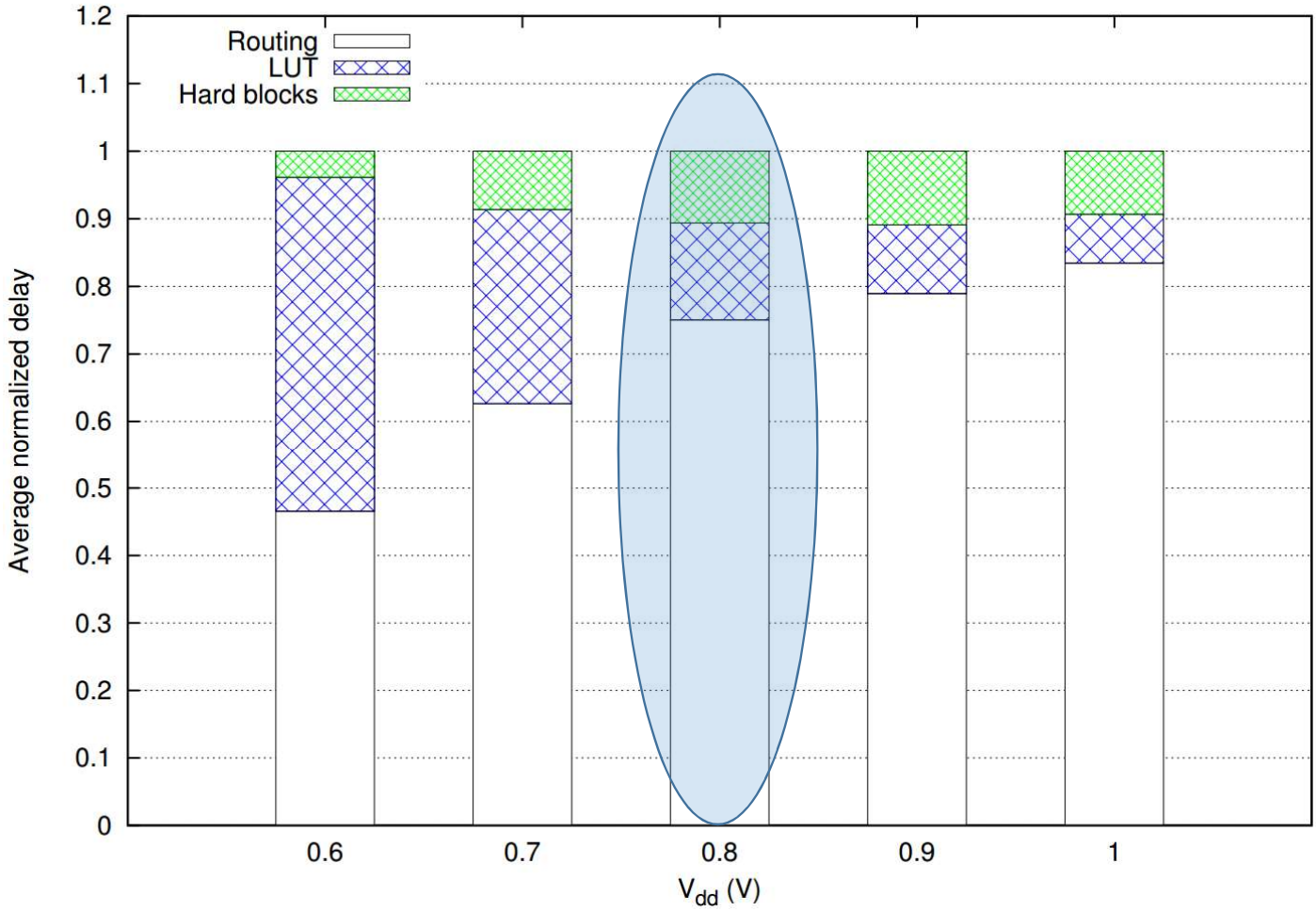
- Background
- Analyzing Existing FPGA building blocks (logic and routing)
- VPR analysis over benchmarks
- Designing new LUTs
- Summary and Future Work

VTR benchmarks' CP Delay Breakdown



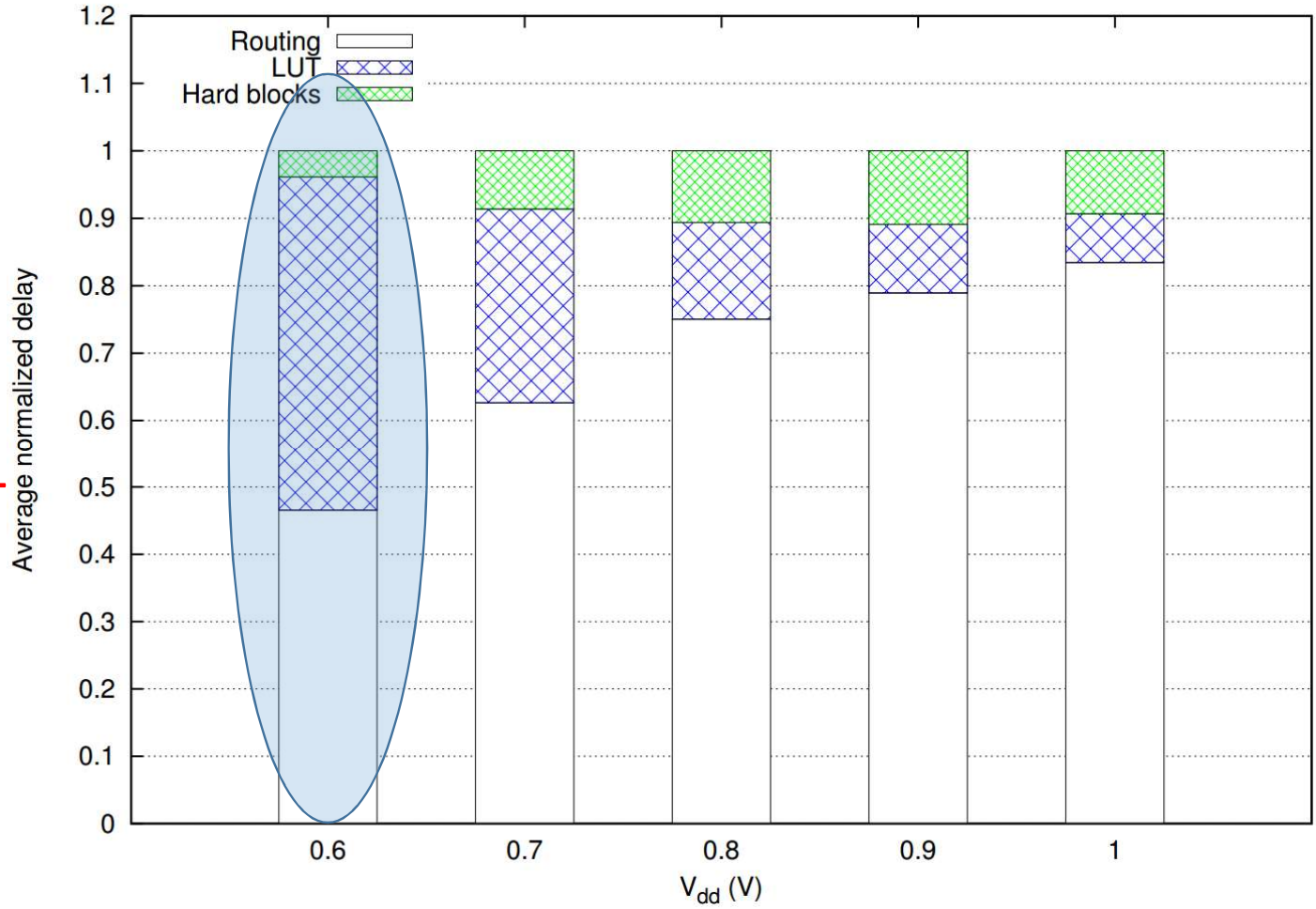
VTR benchmarks' CP Delay Breakdown

- Nominal: ~75% routing, ~15% LUT



VTR benchmarks' CP Delay Breakdown

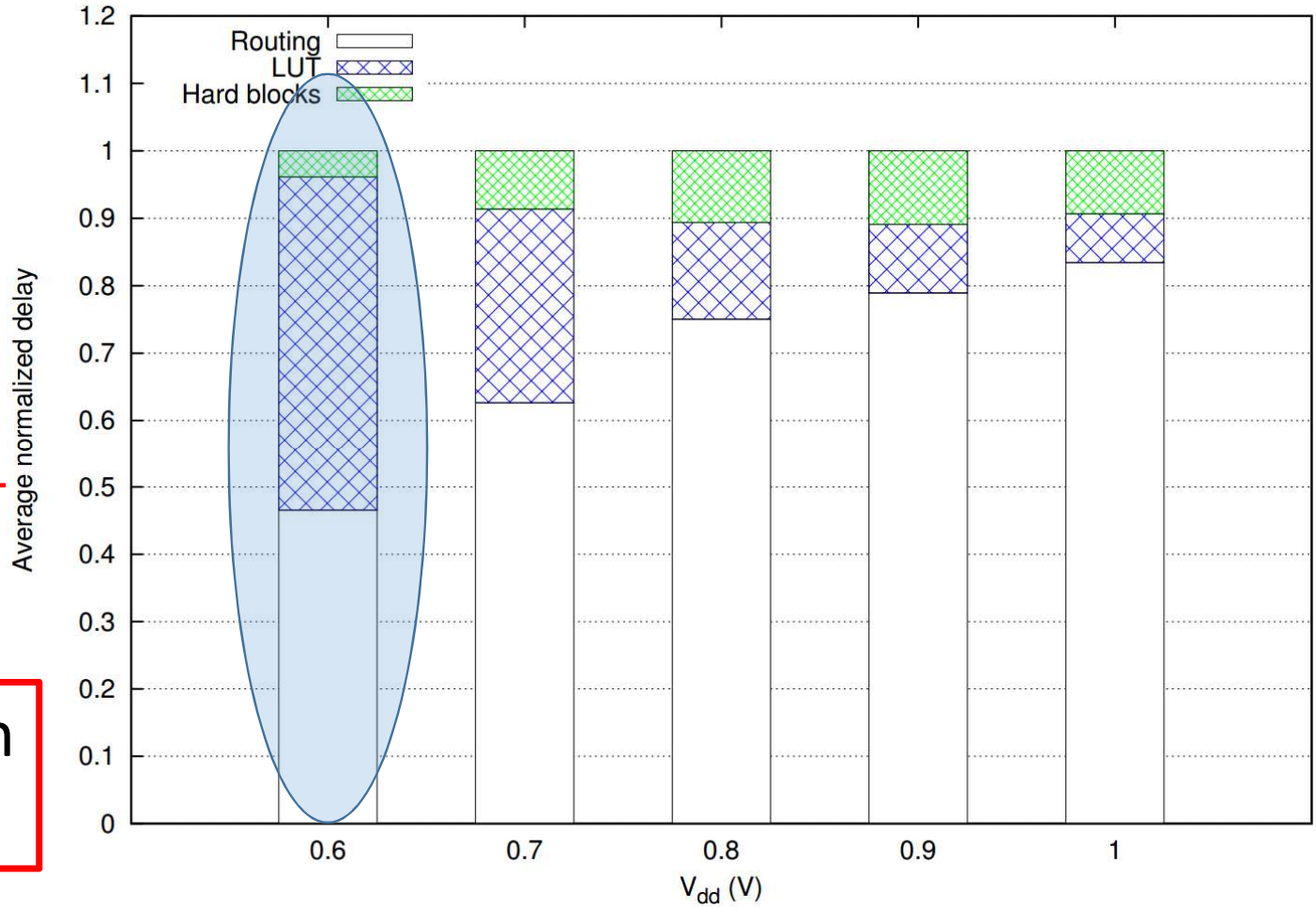
- Nominal: ~75% routing, ~15% LUT
- 0.6 V: ~45% routing, ~50% LUT



VTR benchmarks' CP Delay Breakdown

- Nominal: ~75% routing, ~15% LUT
- 0.6 V: ~45% routing, ~50% LUT

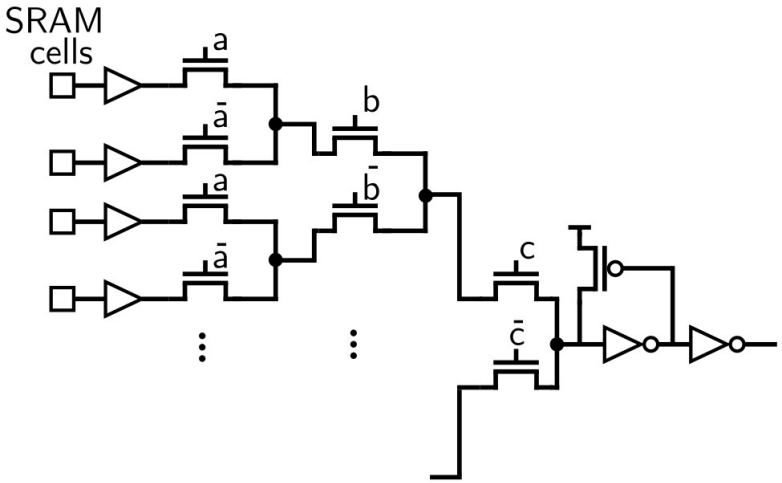
Redesign LUTs



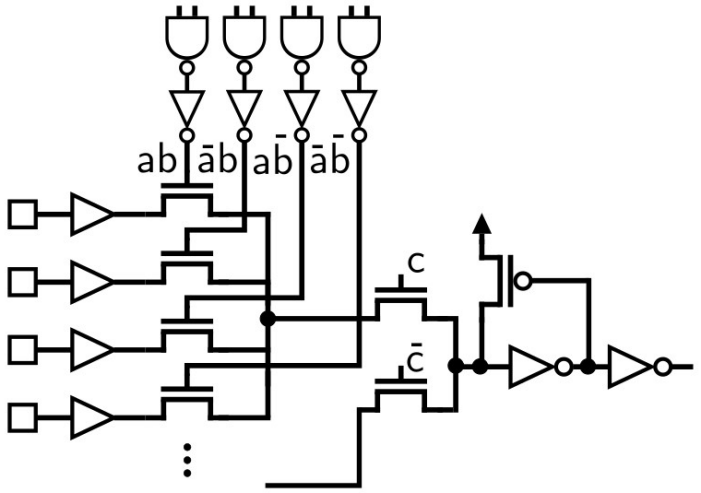
Outline

- Background
- Analyzing Existing FPGA building blocks (logic and routing)
- VPR analysis over benchmarks
- Designing new LUTs
- Summary and Future Work

Proposed LUTs: Decode LUT Inputs (decode LUT)



Conventional LUT (baseline)

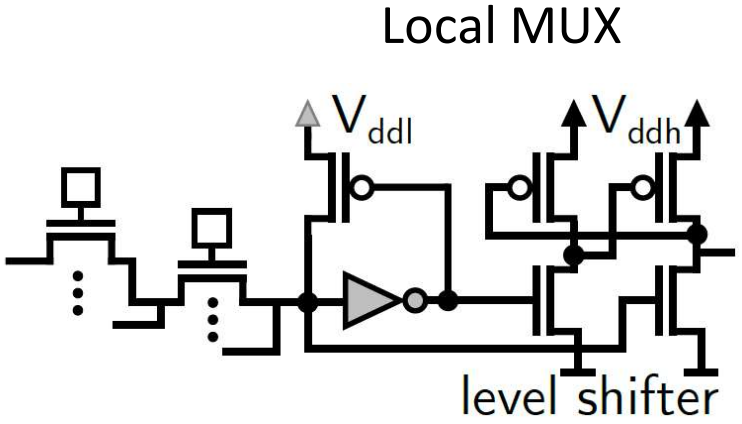


decode LUT

- Decrease number of pass transistors in series
- Reduce number of transistors in a 6-input LUT

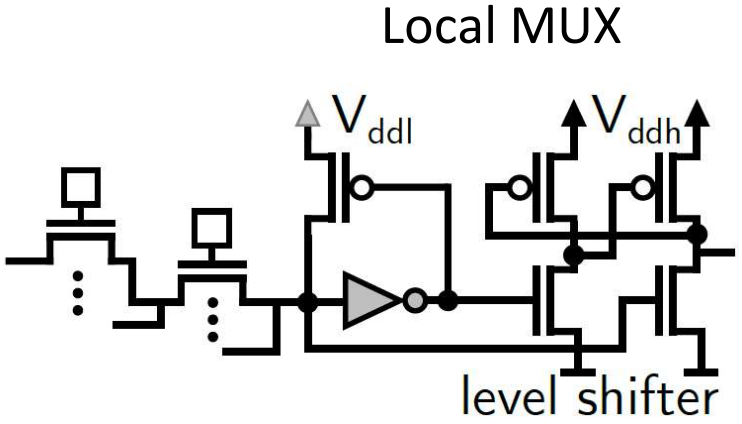
Proposed LUTs: Gate Boosting LUTs (GB LUT)

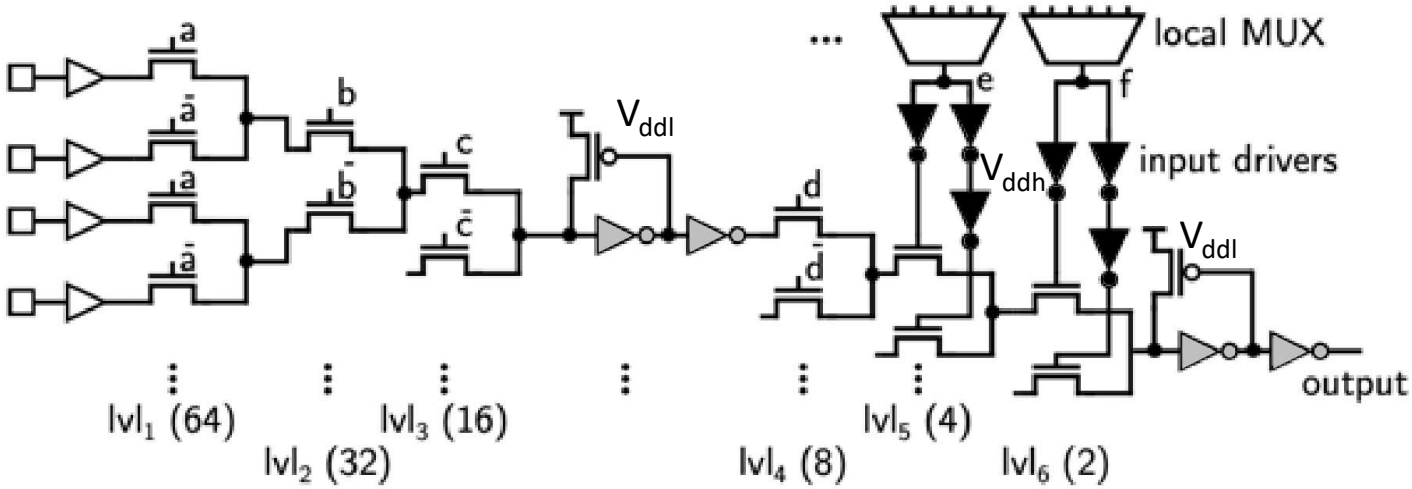
- Add level shifter to local MUX
  - Shifts from low supply voltage to the fixed SRAM 1 V



Proposed LUTs: Gate Boosting LUTs (GB LUT)

- Add level shifter to local MUX
  - Shifts from low supply voltage to the fixed SRAM 1 V
- LUT input drivers supplied by the SRAM 1 V

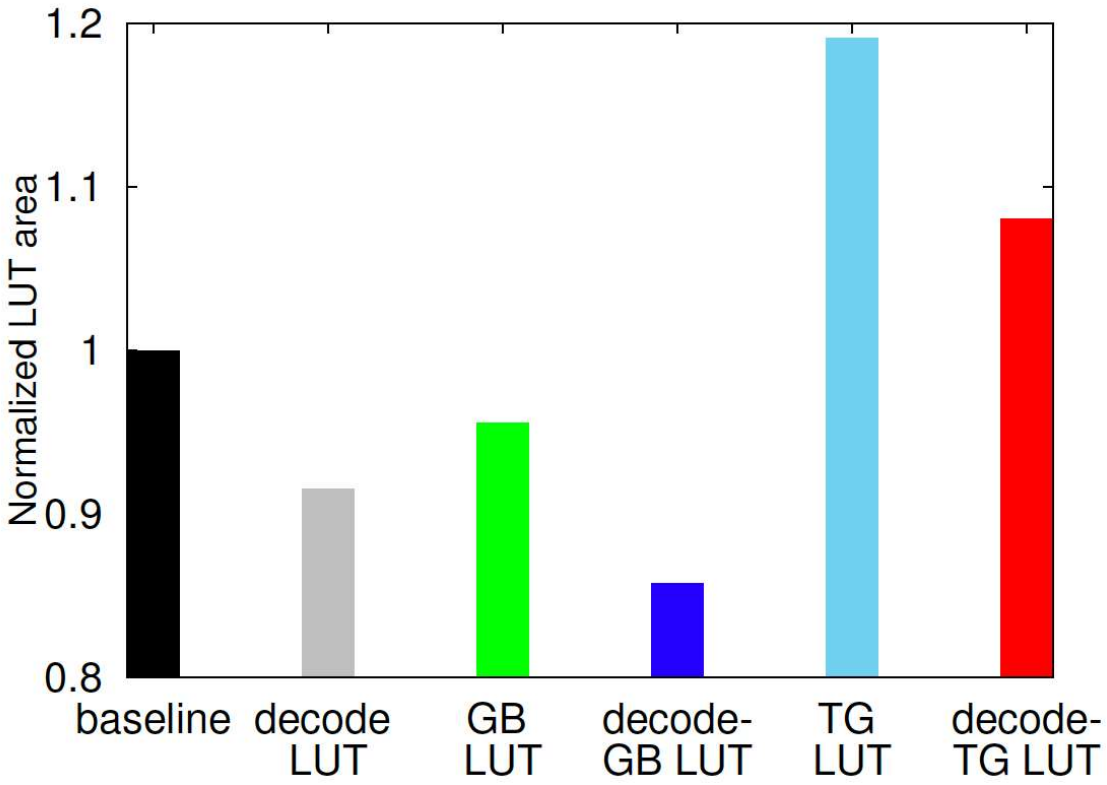


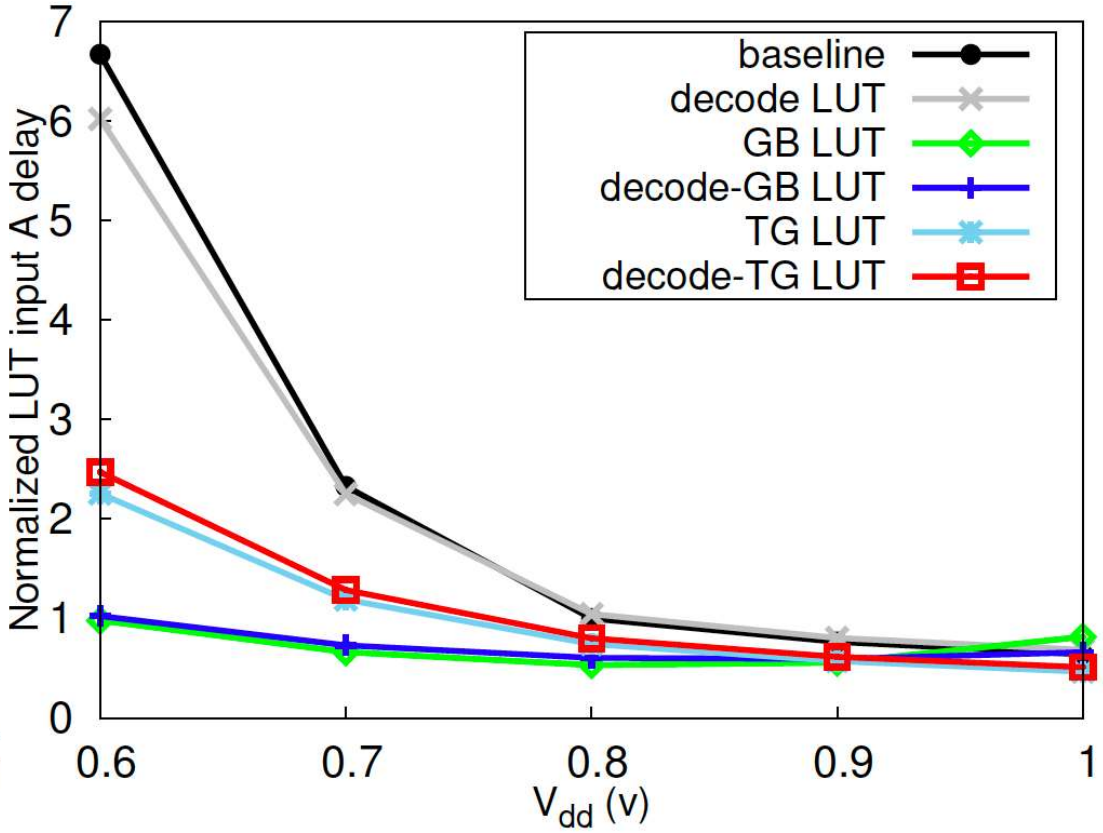


Proposed LUTs: TG LUTs and Hybrid LUTs

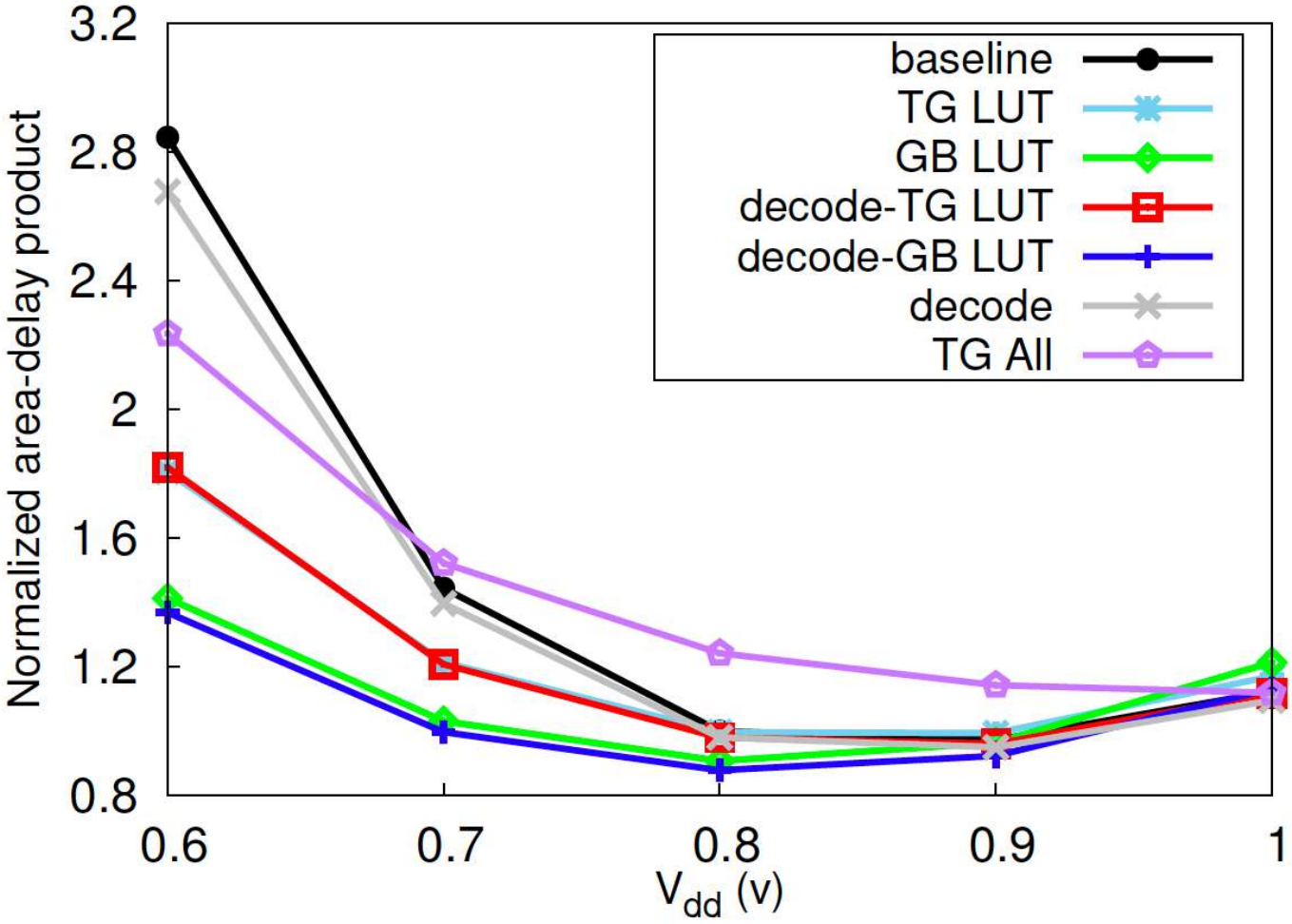
- Using TG in LUTs, while using pass transistors in routing MUXes
- Hybrid LUTs:
  - Gate boosting LUTs + decoding slowest two inputs (decode-GB LUT)
  - TG LUTs + decoding slowest two inputs (decode-TG LUT)

LUT Area and Delay Analysis



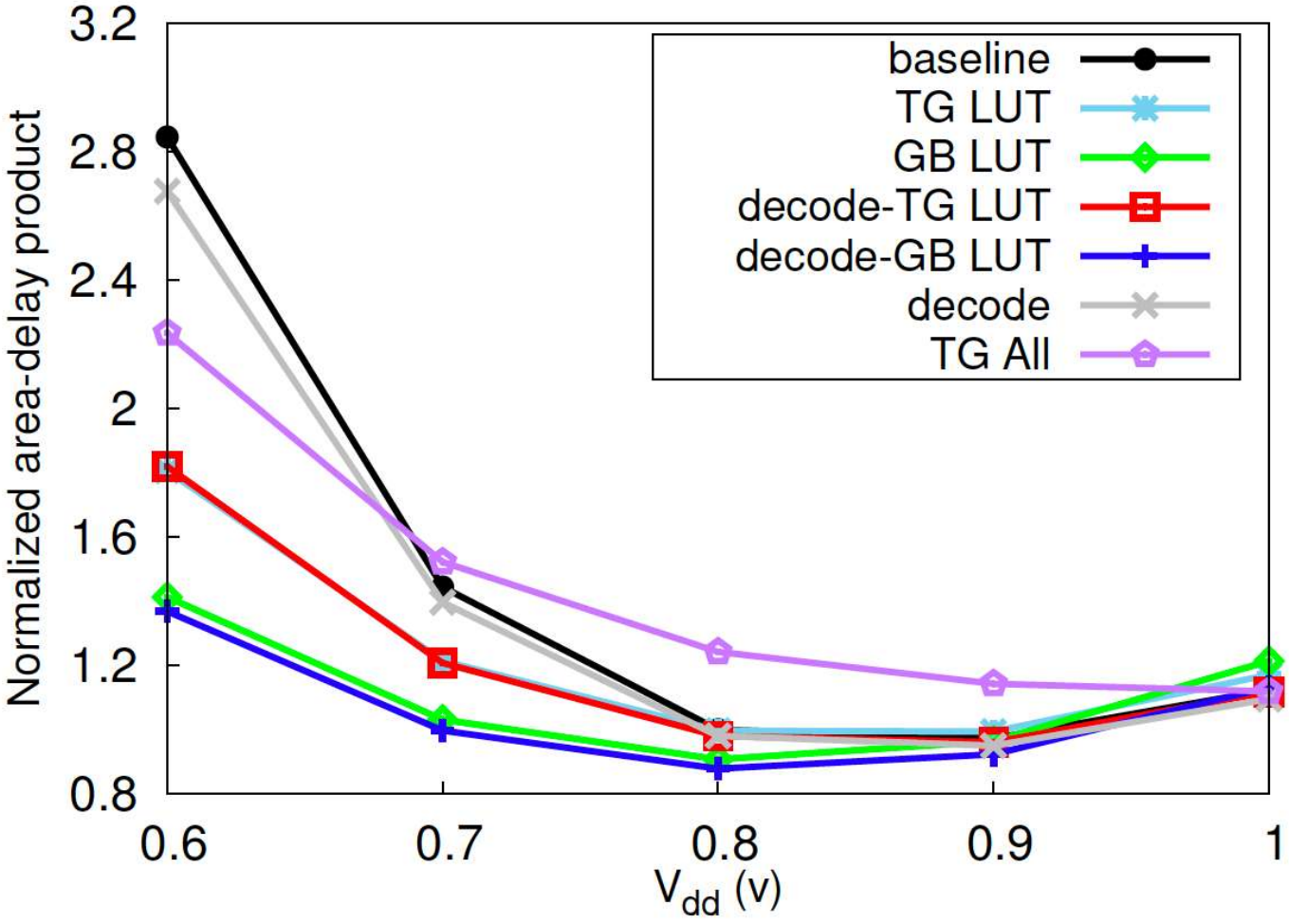


FPGA Tile (Logic + Routing) Area-Delay Product



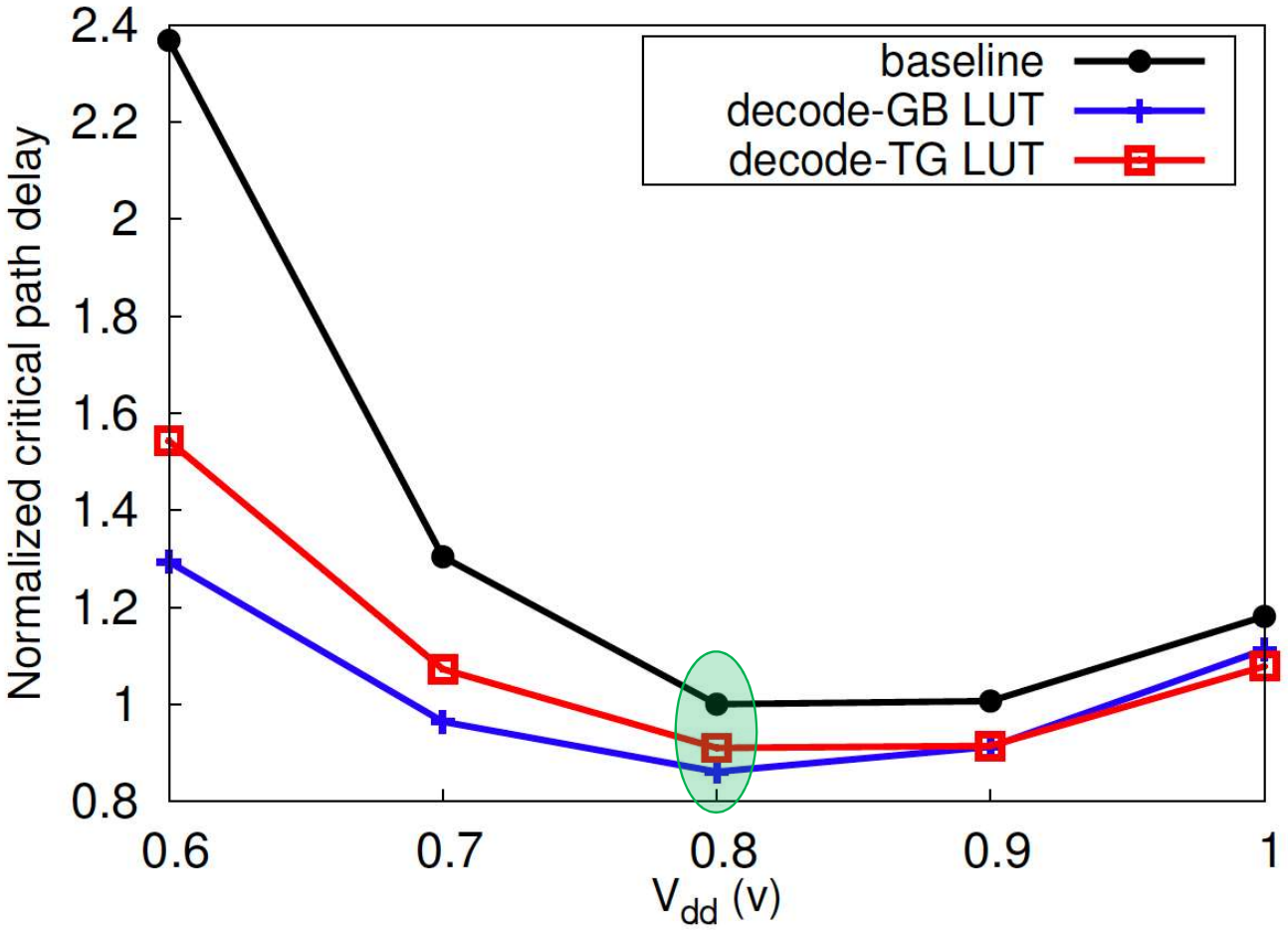
FPGA Tile (Logic + Routing) Area-Delay Product

- Proposed LUTs → better FPGAs at nominal and below
- Decode-GB LUT → 12% lower area-delay than baseline at nominal



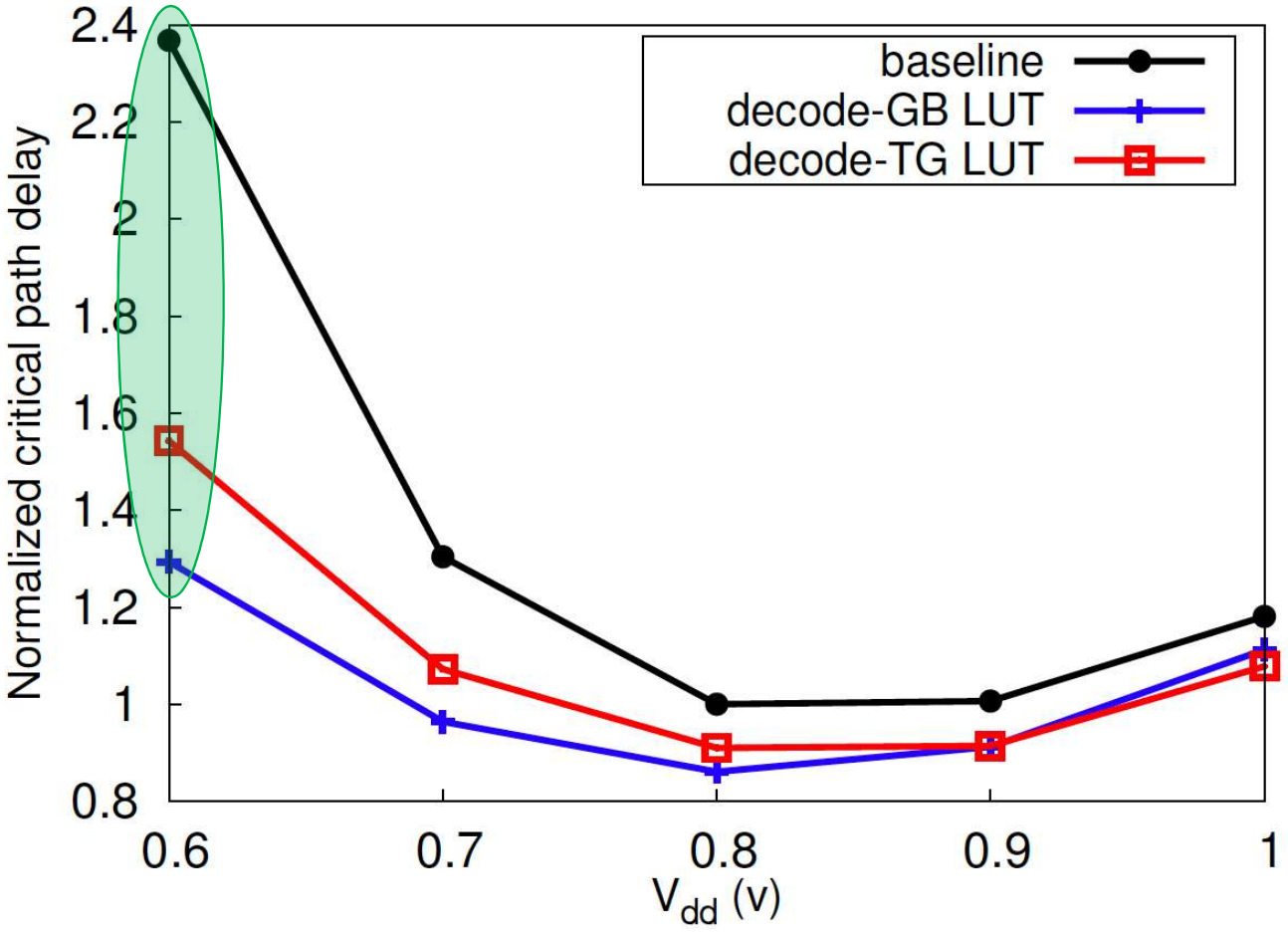
VTR Benchmarks' CP delay (Geomean)

- **14% faster** at 0.8 V

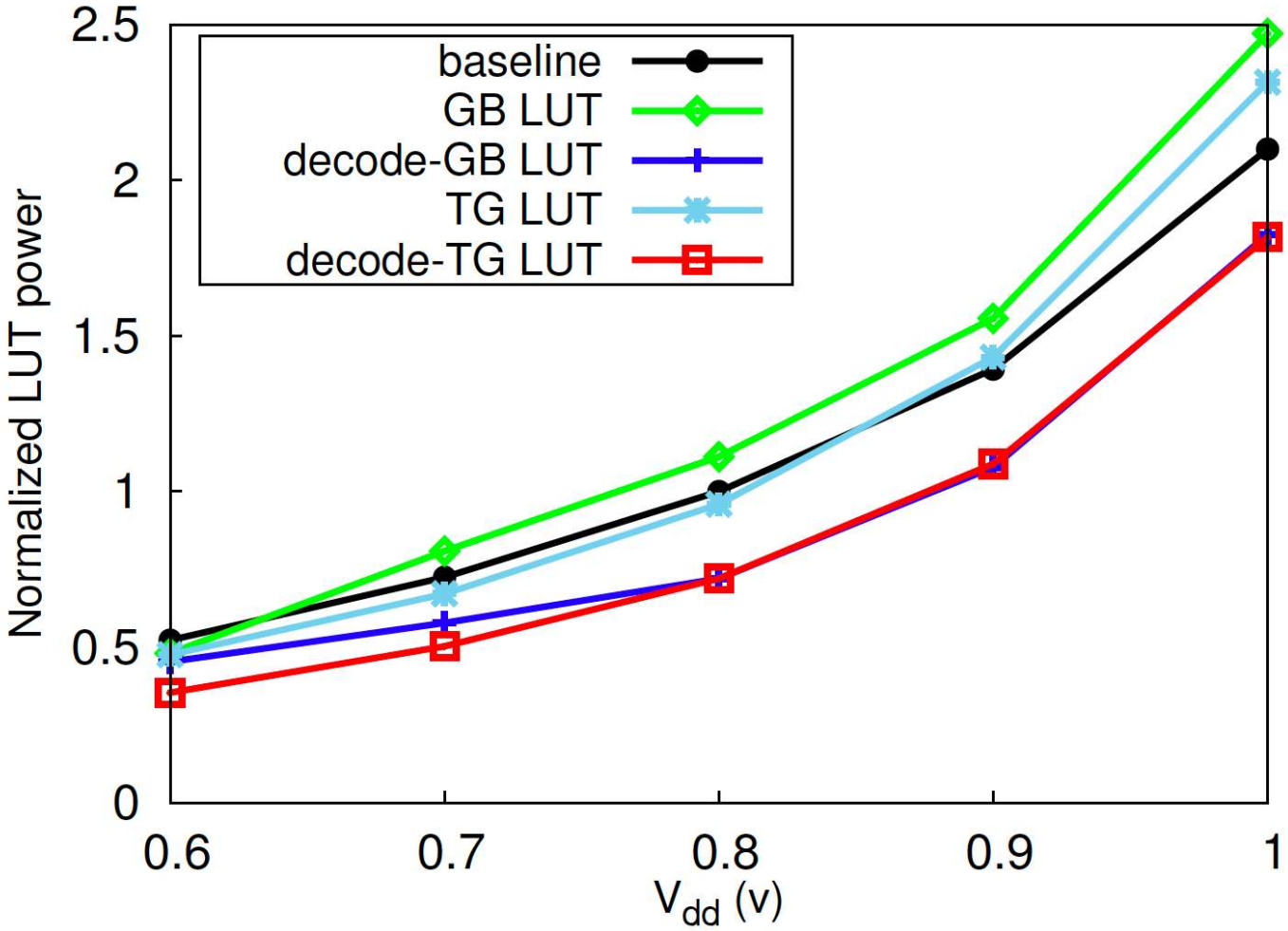


VTR Benchmarks' CP delay (Geomean)

- **14% faster** at 0.8 V
- **45% faster** at 0.6 V

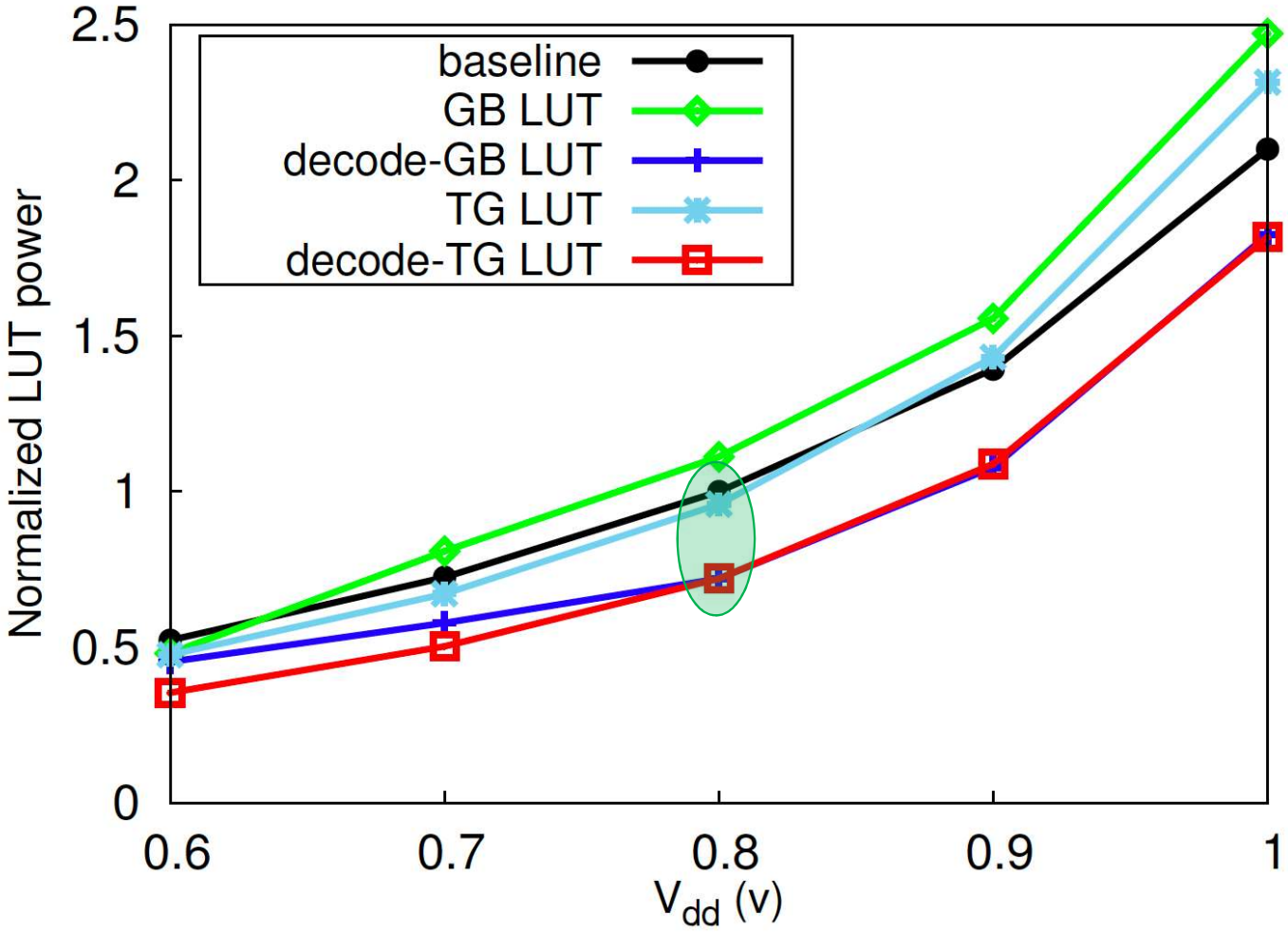


LUT Power Consumption



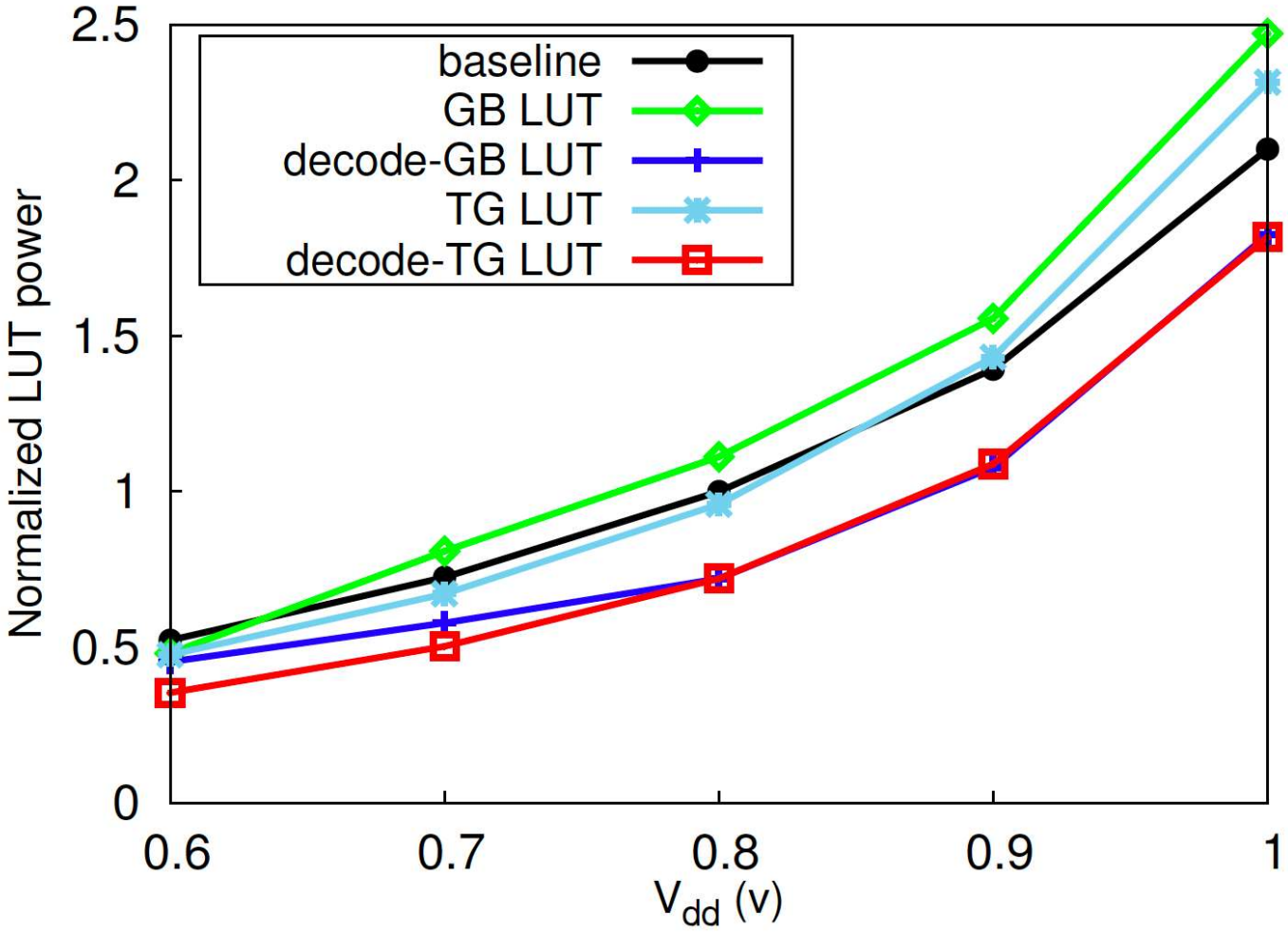
LUT Power Consumption

- Decode-* LUTs have 28% lower power than baseline

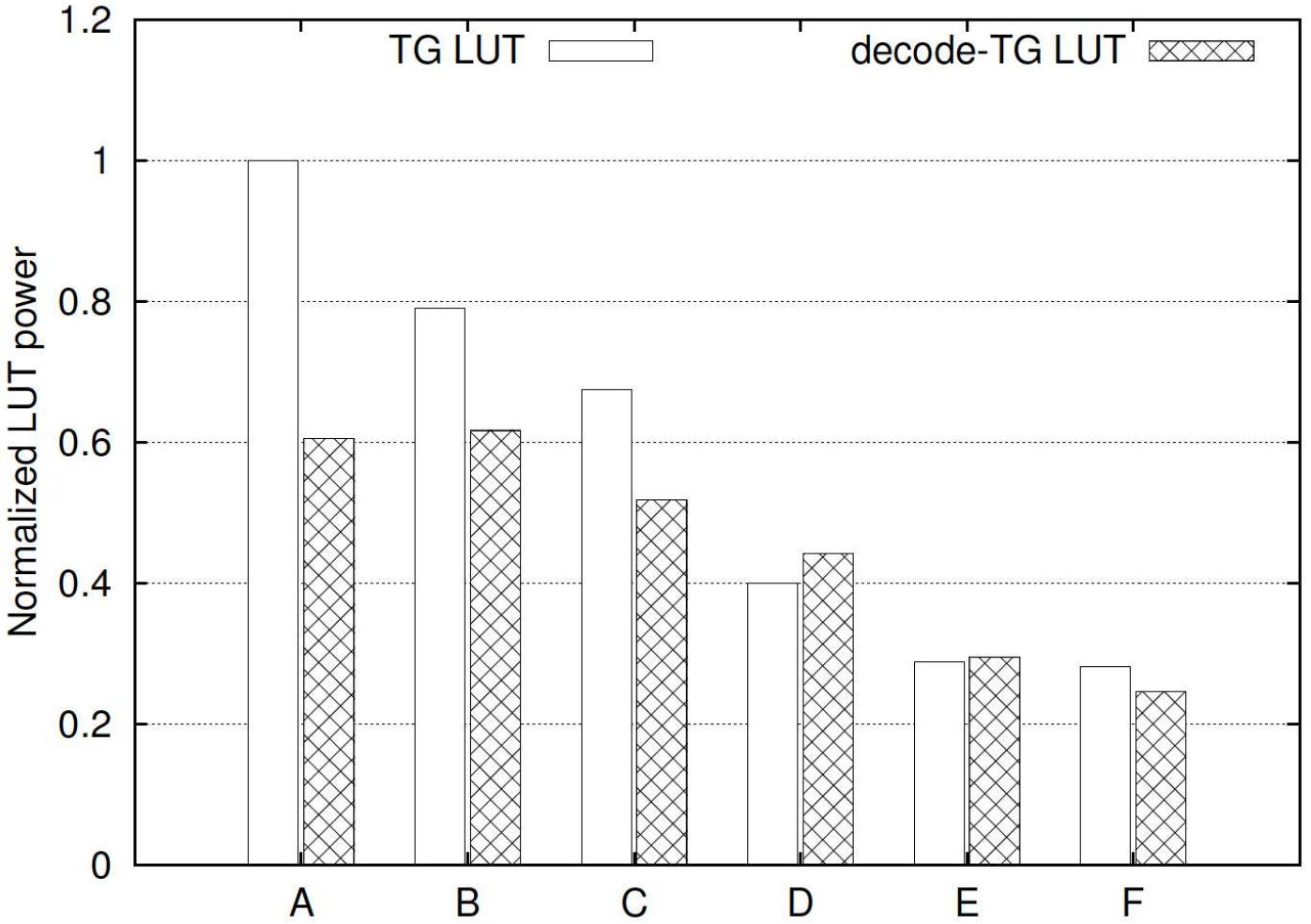


LUT Power Consumption

- Decode-* LUTs have 28% lower power than baseline
- At 0.8 V, decoding reduces the GB LUT and TG LUT power by 35% and 25%, respectively

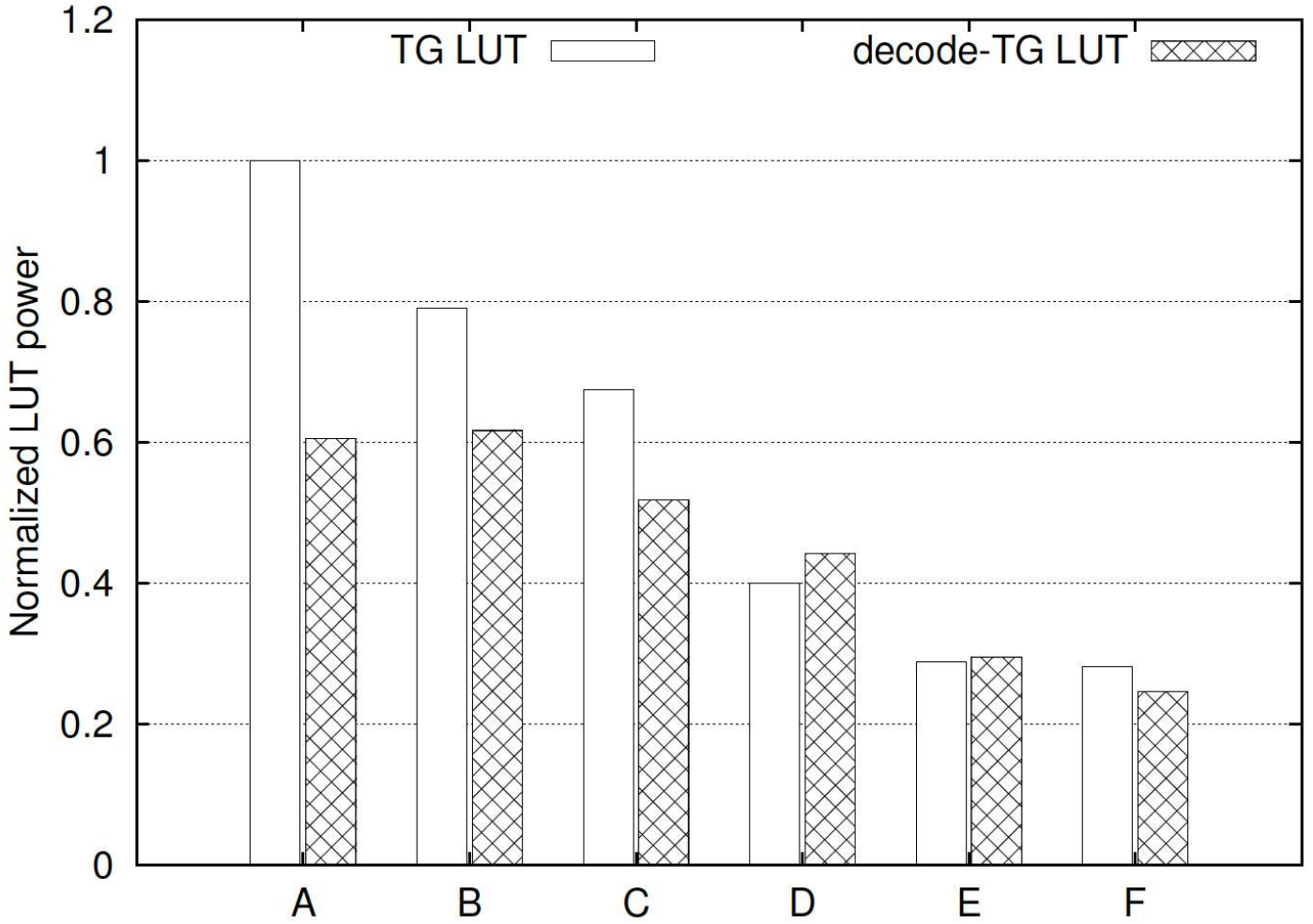


LUT Power Consumption: Decoding Effects



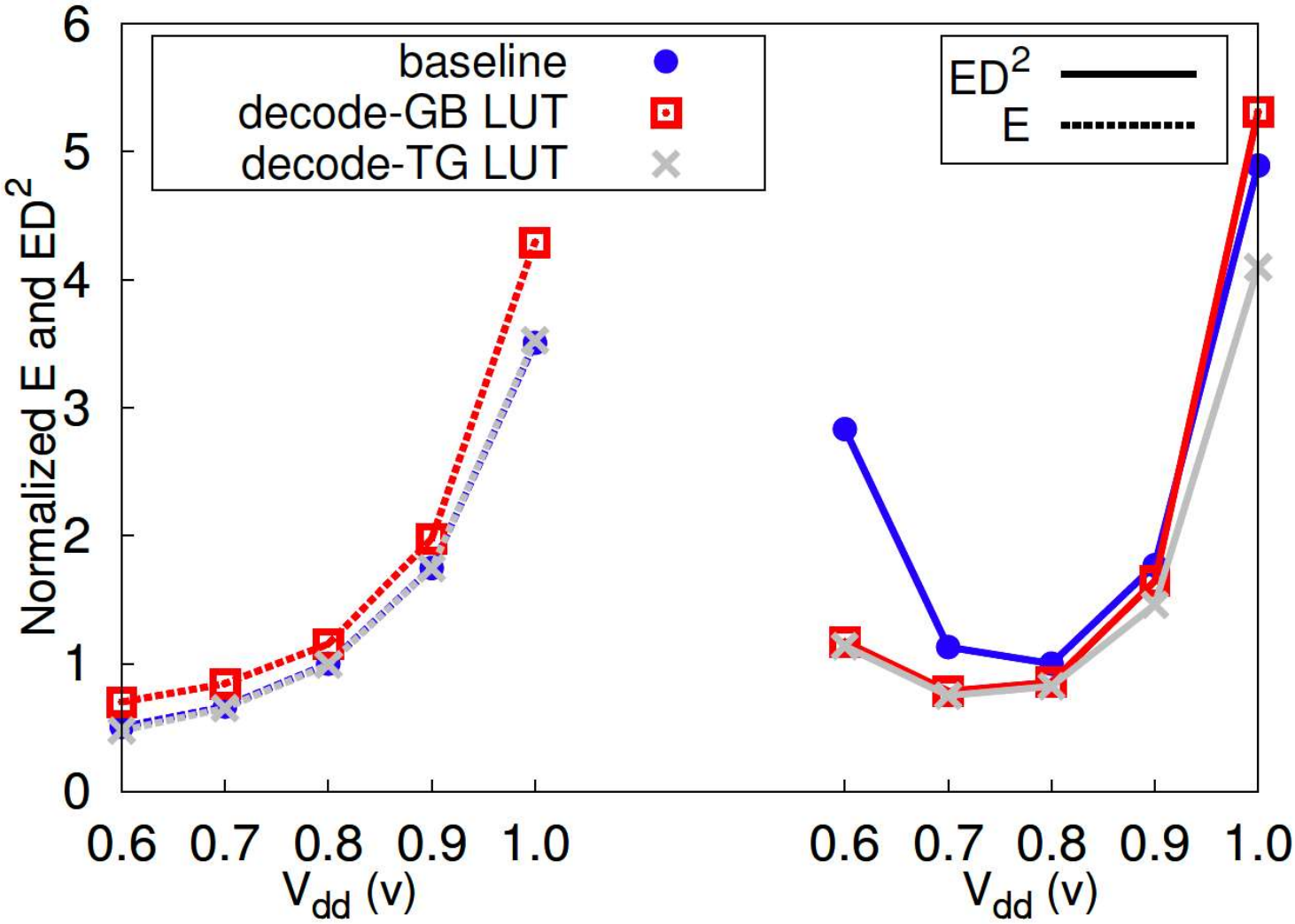
LUT Power Consumption: Decoding Effects

- **40%** power reduction when input A toggles
- Power reductions when B or C toggles



Energy and Energy-Delay^2 Product

- Decode-GB slightly higher energy
- Decode-* 14% lower ED^2 at 0.8 V
- Decode-* 60% lower ED^2 at 0.6 V



Outline

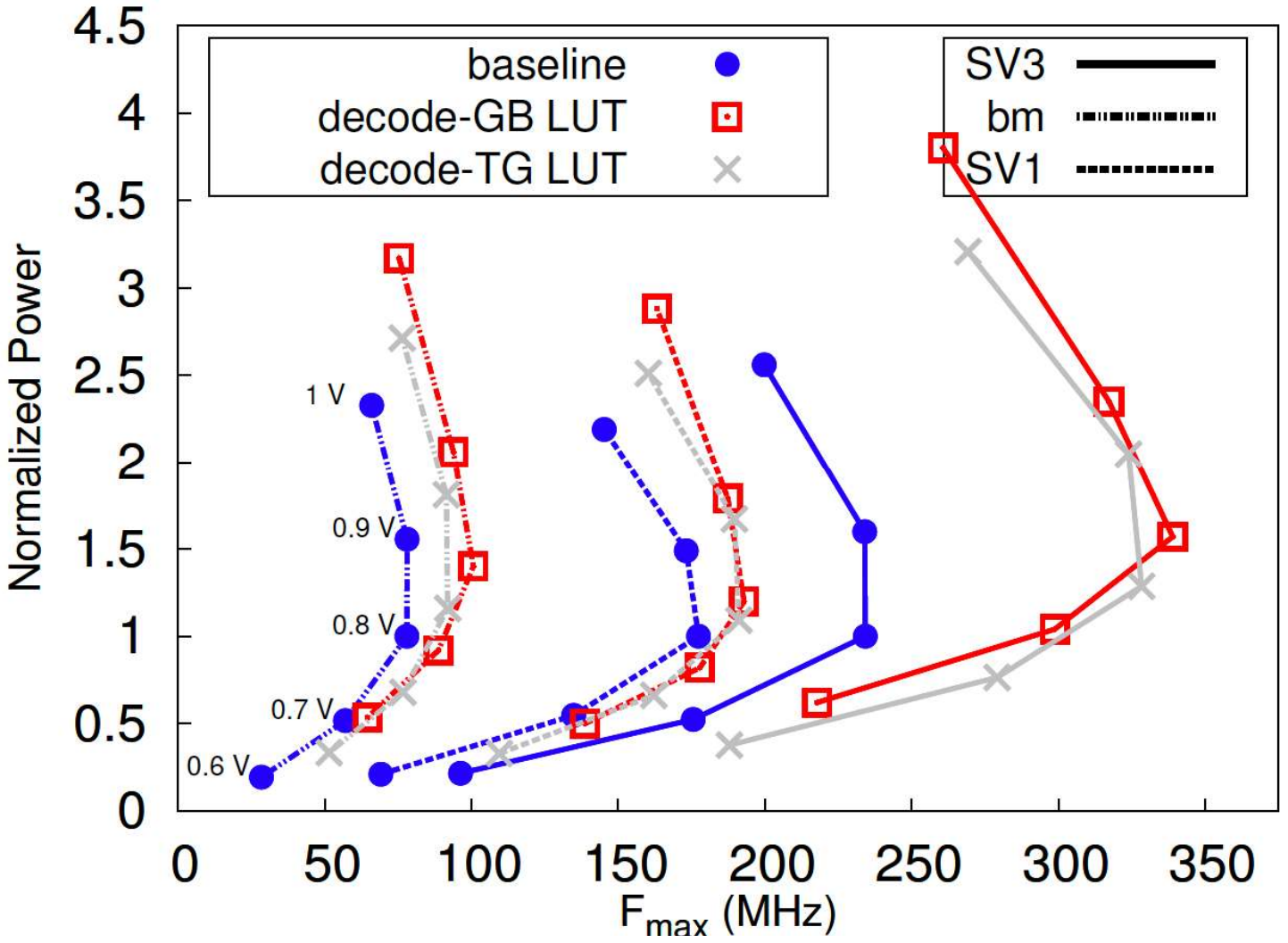
- Background
- Analyzing Existing FPGA building blocks (logic and routing)
- VPR analysis over benchmarks
- Designing new LUTs
- Summary and Future Work

Summary & Future Work

- Delay of a conventional FPGA LUT increases by 7X when V_{dd} reduces from 0.8 V to 0.6 V
- Novel LUTs with input decoding and gate boosting
  - Reduce LUT power by 28%
  - VTR benchmarks geomean CP delay decrease by 14% and 45% at 0.8 V and 0.6 V
  - Reduce ED^2 by 14% and 60% at 0.8 V and 0.6 V
- Future work
  - Using separate voltage islands for LUTs and routing

Power and F_{max} at different supply voltages

- Decode-* outperform baseline
- Decode-GB achieves largest F_{max}

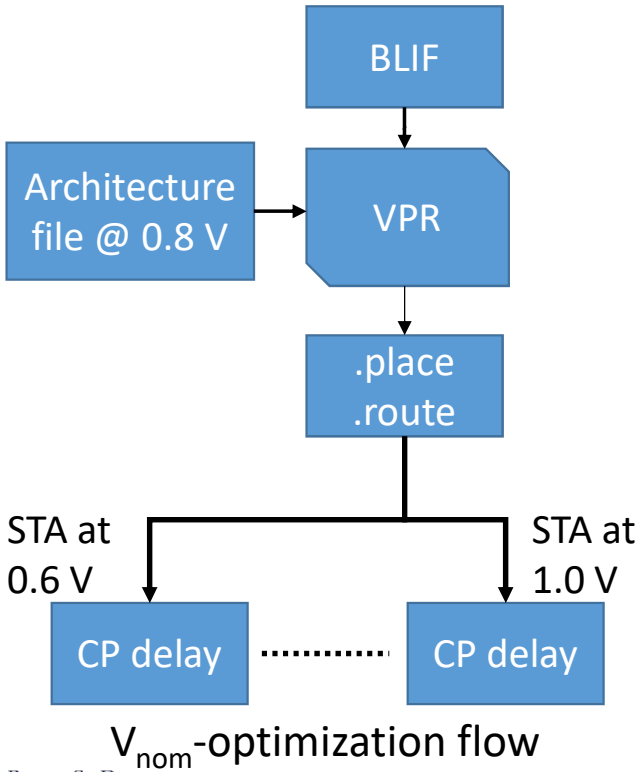


Backup: Area-Delay Product

(V_{ddl}, V_{ddh})	baseline	driver-and-LUT-island	driver-island	decode-driver-island
(0.6, 0.6)	2.85	2.90	2.90	4.03
(0.6, 0.7)	-	1.79	1.85	2.08
(0.6, 0.8)	-	1.44	1.61	1.59
(0.6, 0.9)	-	1.37	1.59	1.40
(0.6, 1.0)	-	1.34	1.58	1.37
(0.7, 0.7)	1.44	1.47	1.47	1.84
(0.7, 0.8)	-	1.12	1.16	1.21
(0.7, 0.9)	-	1.05	1.14	1.05
(0.7, 1.0)	-	1.03	1.13	0.99
(0.8, 0.8)	1.00	1.01	1.01	1.16
(0.8, 0.9)	-	0.95	0.98	0.94
(0.8, 1.0)	-	0.92	0.96	0.88
(0.9, 0.9)	0.97	0.99	0.99	1.01
(0.9, 1.0)	-	0.96	0.97	0.92
(1.0, 1.0)	1.12	1.15	1.15	1.13

Should We Rethink CAD Tools for Variable V_{dd}?

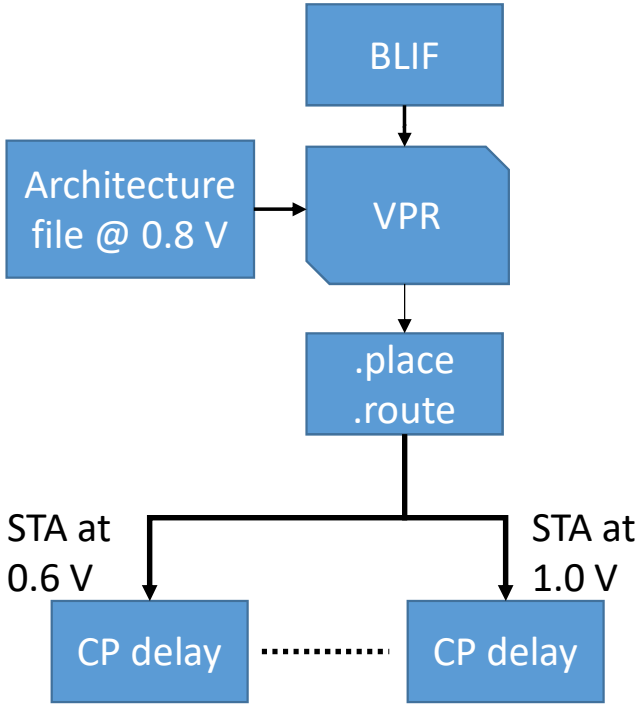
- VPR limit study → V_{nom}- vs V_{used}-optimization flows



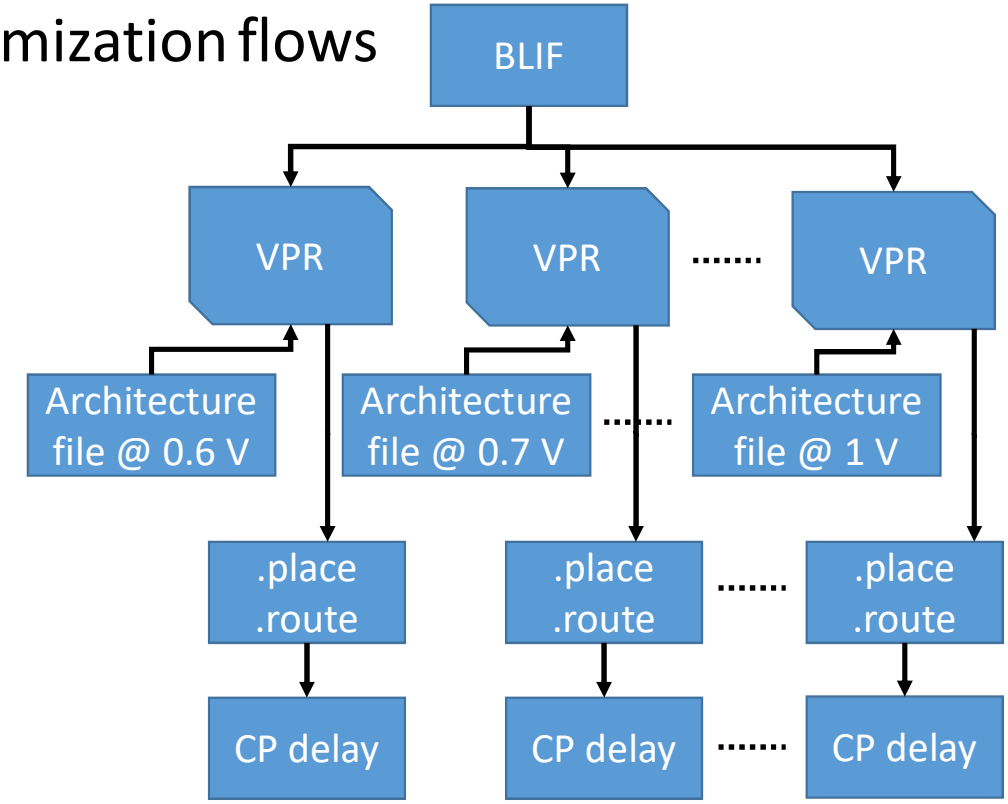
V_{nom}-optimization flow

Should We Rethink CAD Tools for Variable V_{dd}?

- VPR limit study → V_{nom}- vs V_{used}-optimization flows



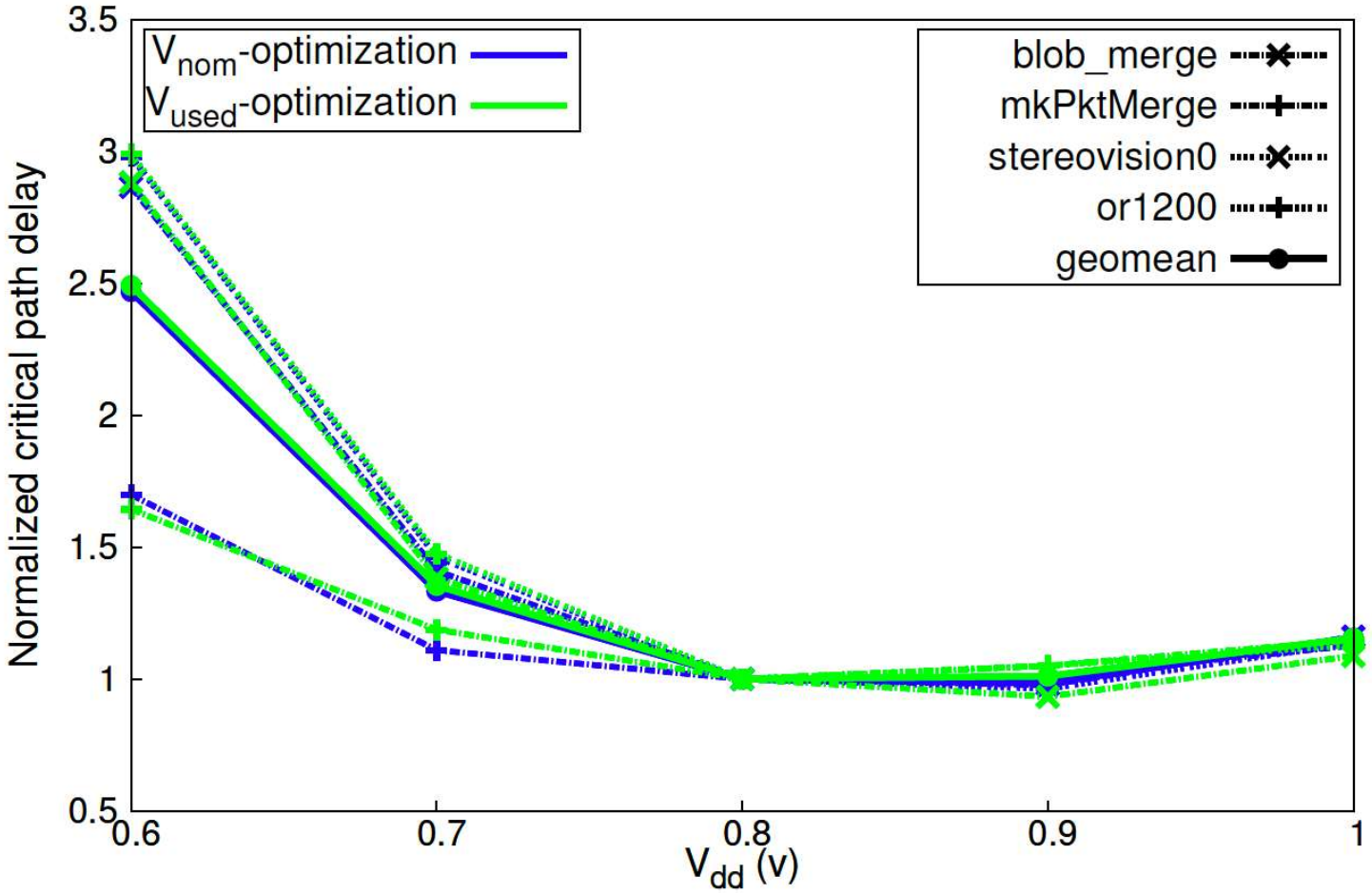
V_{nom}-optimization flow



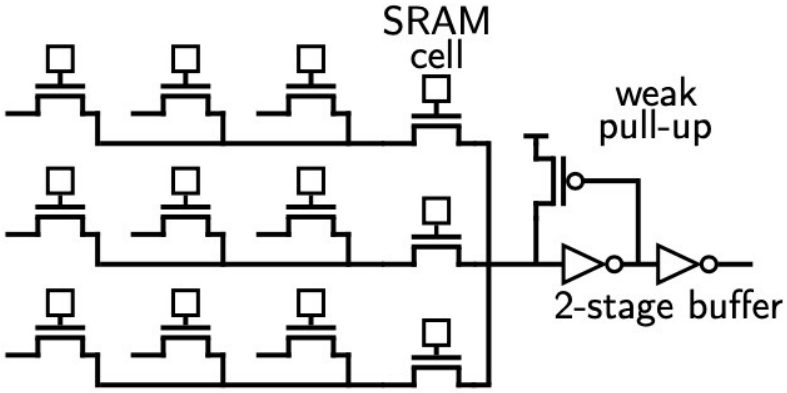
V_{used}-optimization flow

Geomean CP Delay of VTR Benchmarks

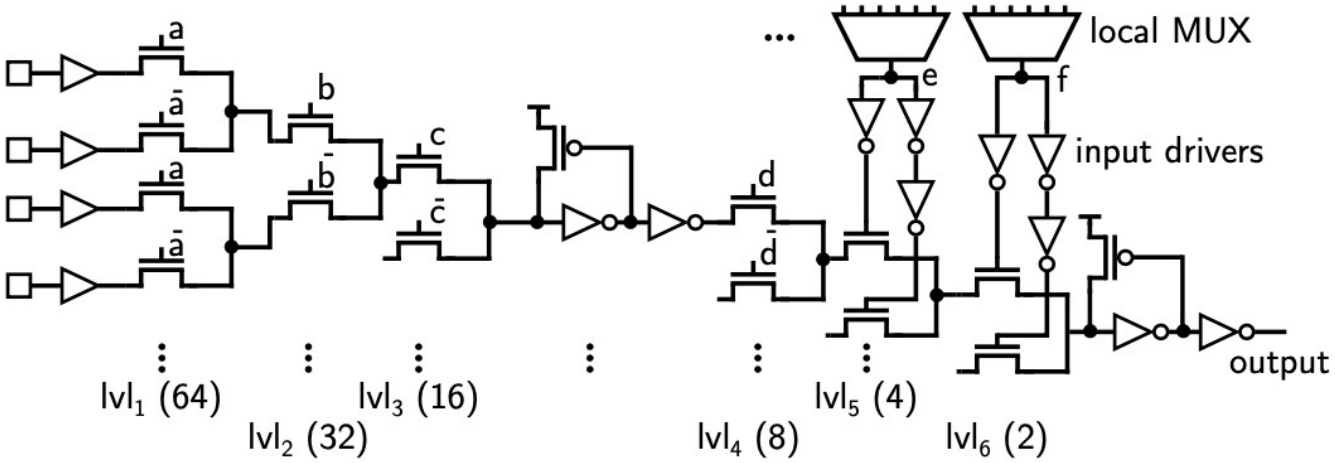
- No obvious gains from V_{used}-optimization
- Better to focus on circuit optimizations



Background: FPGA LUT and Routing Circuitry



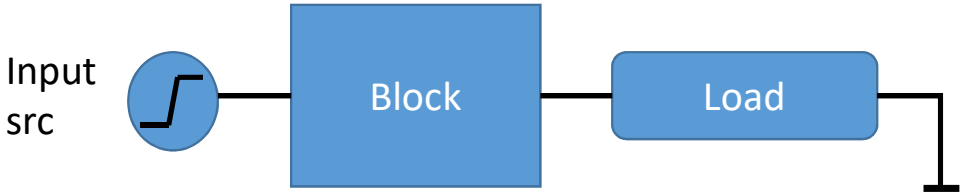
Two-stage routing multiplexer



Tree-based 6-input LUT multiplexer

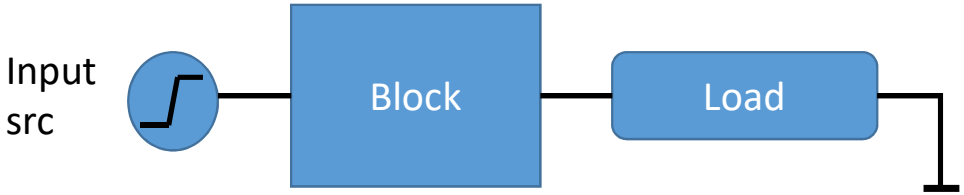
Power Modelling

- Single-input blocks: routing multiplexers, LUT input drivers, etc.
  - Hspice to monitor the current drawn by the block during an input transition



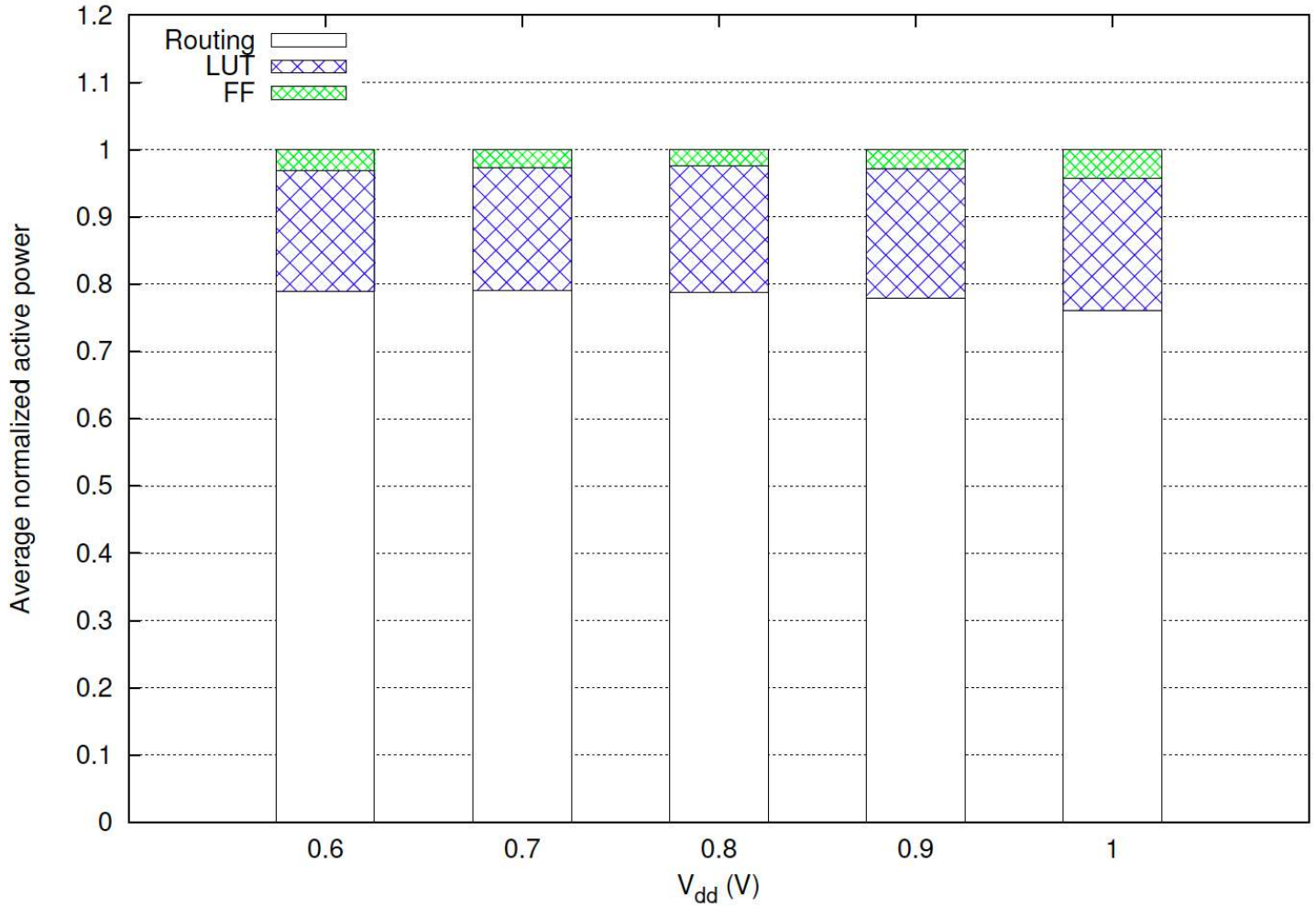
Power Modelling

- Single-input blocks: routing multiplexers, LUT input drivers, etc.
  - Hspice to monitor the current drawn by the block during an input transition



- LUTs have multiple inputs and the current drawn depends on LUT mask
  - Generate hundreds of random LUT masks, and for each mask:
    - Monitor the current drawn when each of the LUT inputs toggles

VTR benchmarks' Active Power Breakdown



VTR benchmarks' Active Power Breakdown

- Routing consistently contributes ~78% of the FPGA active power.