



Architecture and Synthesis for Power-Efficient FPGAs

Jason Cong
VLSI CAD LAB
University of California, Los Angeles

Joint work with Deming Chen, Lei He, Fei Li, Yan Lin

Partially supported by NSF Grants CCR-0096383, and CCR-0306682,
and Altera under the California MICRO program

Outline

- Introduction
- Understanding Power Consumption in FPGAs
- Architecture Evaluation and Power Optimization
- Low Power Synthesis
- Conclusions

FPGA Advantages



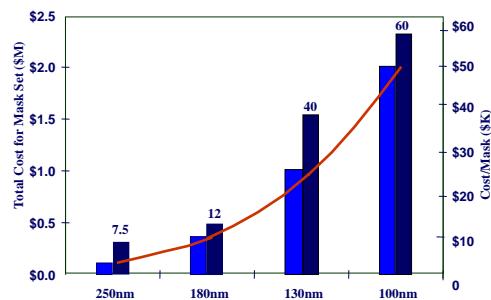
- Short TAT (total turnaround time)
- No or very low NRE

ASICs Increasingly Expensive

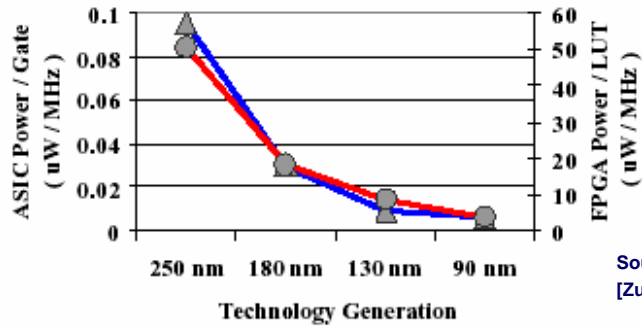
- Traditional ASIC designs are facing rapid increase of NRE and mask-set costs at 90nm and below

Process (um)	2.0	...	0.8	0.6	0.35	0.25	0.18	0.13	0.10
Single Mask cost (\$K)	1.5		1.5	2.5	4.5	7.5	12	40	60
# of Masks	12		12	12	16	20	26	30	34
Mask Set cost (\$K)	18		18	30	72	150	312	1,000	2,000

Source: EETimes



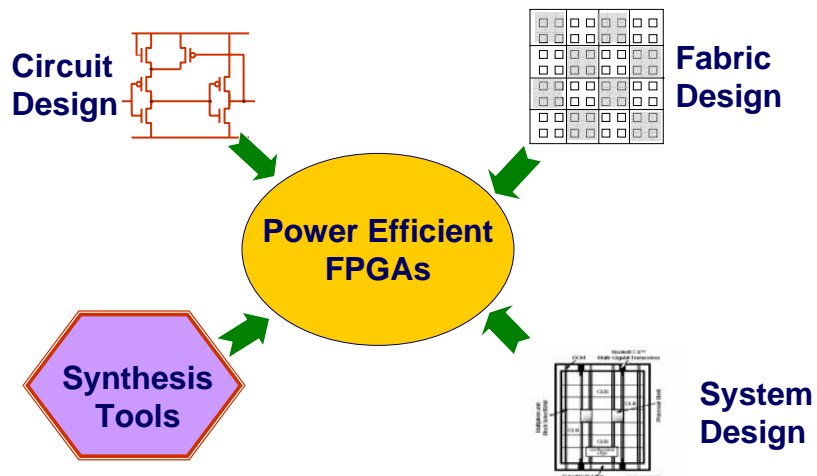
But ... FPGA is Known to be Power Inefficient



Source:
[Zuchowski, et al, ICCAD02]

- FPGA consumes 50-100X more power
- Need to explore power efficient FPGAs

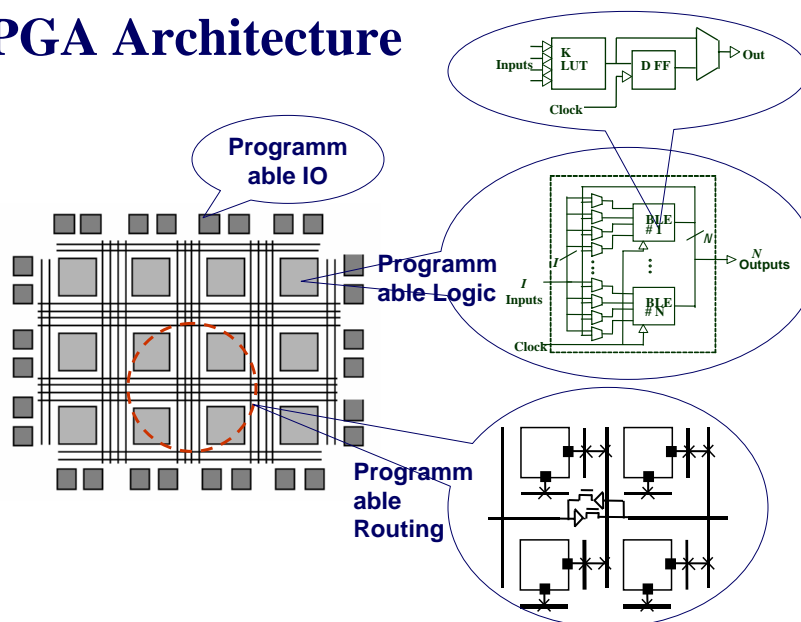
Our Research



Outline

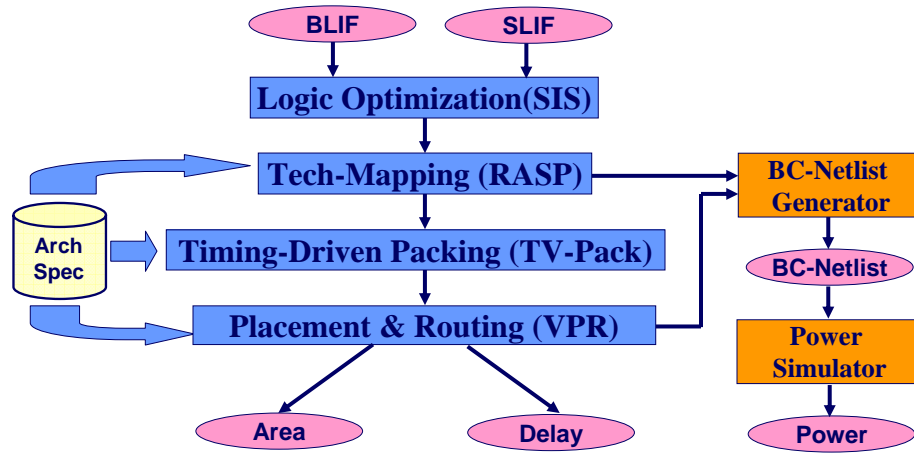
- Introduction
- Understanding Power Consumption in FPGAs
- Architecture Evaluation and Power Optimization
- Low Power Synthesis
- Conclusions

FPGA Architecture

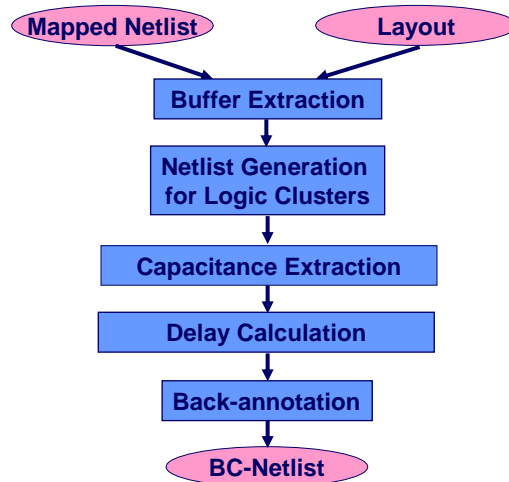


Evaluation Framework – *fpgaEva-LP*

fpgaEva-LP [Cong, et al, ICCAD'00]



BC-Netlist Generator

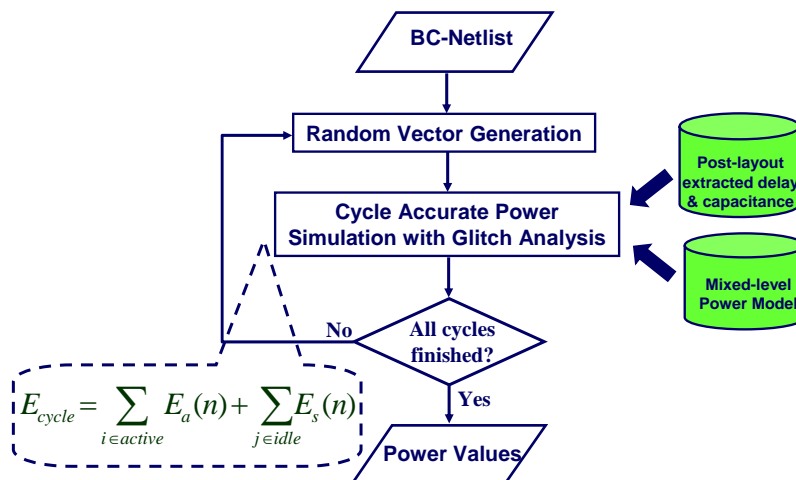


Mixed-level Power Model – Overview

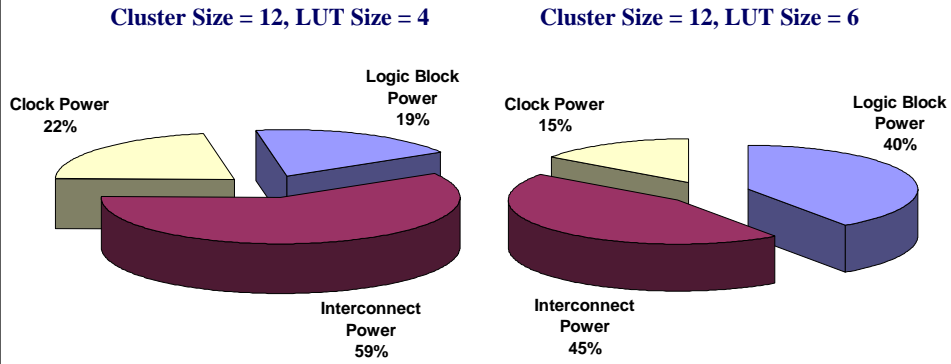
- **Dynamic power**
 - ◆ Switching power
 - ◆ Short-circuit power
- **Static Power**
 - ◆ Sub-threshold leakage
 - ◆ Gate leakage
 - ◆ Reverse biased leakage
- **Related to signal transitions**
 - Functional switch
 - Glitch
- **Depending on the input vector**

components power sources	Logic Block	Interconnect & clock
Dynamic	Macro-model	Switch-level model
Static	Macro-model	Macro-model

Cycle-Accurate Power Simulator

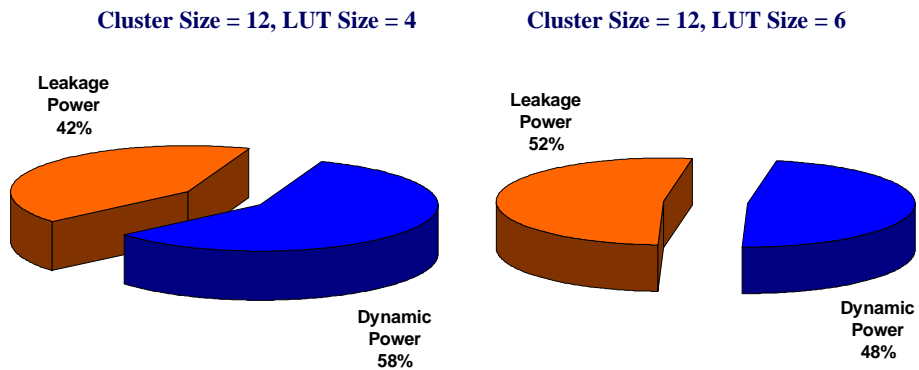


Power Breakdown



■ Interconnect power is dominant

Power Breakdown (cont'd)

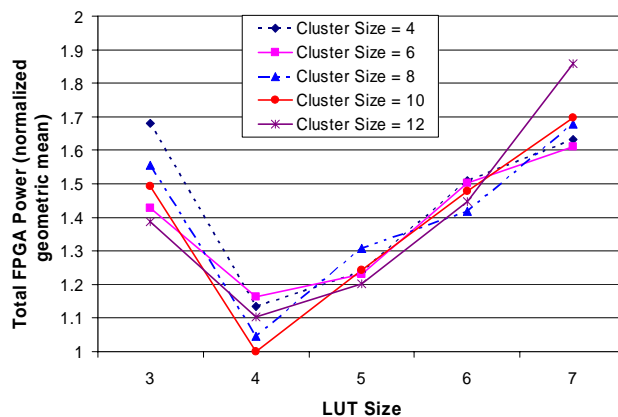


Leakage power becomes increasingly important

Outline

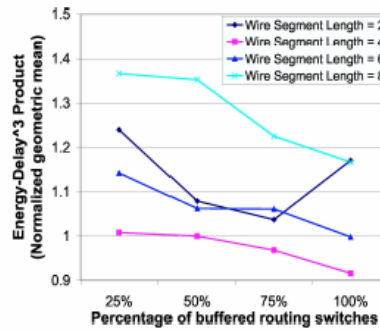
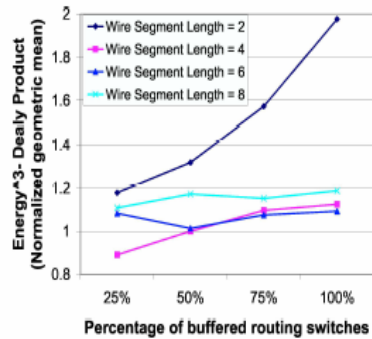
- Introduction
- Understanding Power Consumption in FPGAs
- Architecture Evaluation and Power Optimization
 - ◆ Architecture Parameter Selection
 - ◆ Dual-Vdd/Dual-Vt FPGA Architecture
- Low Power Synthesis with Dual-Vdd
- Conclusion

Total Power along LUT and Cluster Size Changes



Routing architecture: segmented wire with length of 4, and 50% tri-state buffers in routing switches

Routing Architecture Evaluation



Architecture of Low-power and High-performance

Applications	Best FPGA architecture	Energy (E)	Delay (t)	E^3t	Et^3
Low-power (E^3t)	Cluster size 10, LUT size 4, wire segment length 4, 25% buffered routing switches	0.9653	0.9904	0.8909	1.0080
High-performance (Et^3)	Cluster size 12, LUT size 4, Wire segment length 4, 100% buffered routing switches	1.0502	0.8865	1.0268	0.7865

- Arch. Parameter selection leads to 10% power/delay trade-off
- Uniform FPGA fabrics provide limited power-performance tradeoff
- Need to explore heterogeneous FPGA fabrics, e.g. *dual-Vt and dual-Vdd fabrics*

Outline

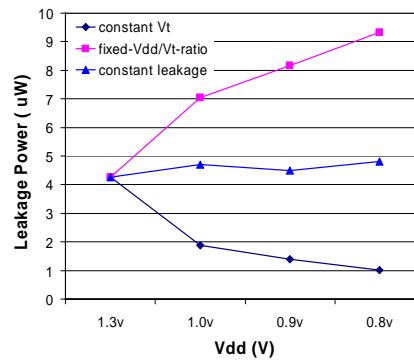
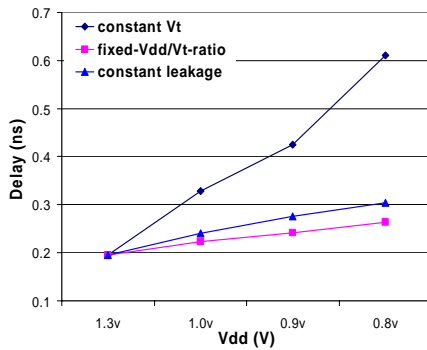
- **Introduction**
- **Understanding Power Consumption in FPGAs**
- **Architecture Evaluation and Power Optimization**
 - ◆ **Architecture Parameter Selection**
 - ◆ **Dual-Vdd/Dual-Vt FPGA Architecture [Li, et al, FPGA'04]**
- **Low Power Synthesis with Dual-Vdd**
- **Conclusion**

Dual-Vdd LUT Design

- **Dual-Vdd technique makes use of the timing slack to reduce power**
 - ◆ **VddH devices on critical path → performance**
 - ◆ **VddL devices on non-critical paths → power**
 - ◆ **Assume uniform Vdd for one LUT**
- **Threshold voltage V_t should be adjusted *carefully* for different Vdd levels**
 - ◆ **To compensate delay increase**
 - ◆ **To avoid excessive leakage power increase**

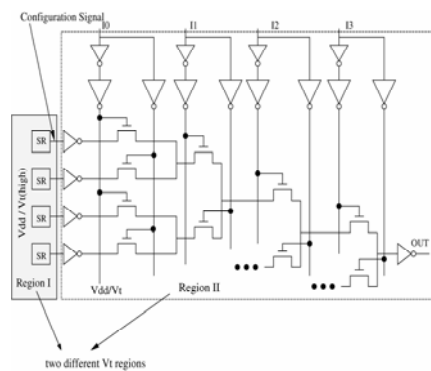
Vdd/Vt-Scaling for LUTs

- Three scaling schemes
 - ◆ Constant-Vt scaling
 - ◆ Fixed-Vdd/Vt-ratio scaling
 - ◆ Constant-leakage scaling
- Constant-leakage scaling obtains a good tradeoff
- useful for both single-Vdd scaling and dual-Vdd design



Dual-Vt LUT Design

- LUT is divided into two parts
 - ◆ Part I: configuration cells → high Vt
 - ◆ Part II: MUX tree and input buffers → normal Vt (decided by constant-leakage Vdd-scaling)



- Configuration SRAM cells
 - ◆ Content remains unchanged after configuration
 - ◆ Read/write delay is not related to FPGA performance
- Use high Vt ~40% of Vdd
 - ◆ Maintain signal integrity
 - ◆ Reduce SRAM leakage by 15X and LUT leakage by 2.4X
 - ◆ Increase configuration time by 13%

Pre-Defined Dual-Vt Fabric

- Power saving
- FPGA fabric arch-SVDT

◆ 14.6% for sequential circuits

◆ 11.6% for combinational circuits

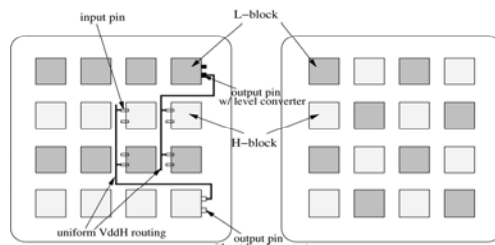
◆ Homogeneous fabric at logic block level with much reduced leakage power

- Traditional design flow in VPR can be applied

	arch-SVST (Single Vt) power (watt)	arch-SVDT (Dual Vt) power saving		arch-SVST (Single Vt) power (watt)	arch-SVDT (Dual Vt) power saving
am4	0.0798	8.5%	bigkey	0.148	12.3%
apex2	0.108	9.1%	clma	0.632	14.8%
apex4	0.0536	12.3%	diffeq	0.0391	19.7%
des	0.234	10.7%	dsip	0.134	14.5%
ex1010	0.179	17.3%	elliptic	0.140	16.3%
ex5p	0.059	11.6%	frisc	0.190	19.2%
misex3	0.0753	9.4%	s298	0.0736	13.4%
pdc	0.256	14.7%	s38417	0.307	11.7%
seq	0.0927	9.4%	s38484	0.261	10.2%
spla	0.180	12.4%	tseng	0.0351	14.0%
Avg.		11.6%	Avg.		14.6%

Dual-Vdd FPGA Fabric

- Granularity: logic block (i.e., cluster of LUTs)
 - Smaller granularity => intuitively more power saving
 - But a larger implementation overhead
- Layout pattern: pre-defined dual-Vdd pattern
 - Row-based or interleaved pattern
 - Ratio of VddL/VddH blocks is 2:1 (benchmark profiling)
- Interconnect uses uniform VddH



L-block:
VddL

H-block:
VddH

(a) Row-based dual-Vdd layout pattern
(Ratio VddL/Row/VddH/Row = 1:1)

(b) Interleaved dual Vdd layout pattern
(Ratio VddL/Block/VddH/Block = 1:1)

Simple Design Flow for Dual-Vdd Fabric

- Based on traditional design flow, but with new steps

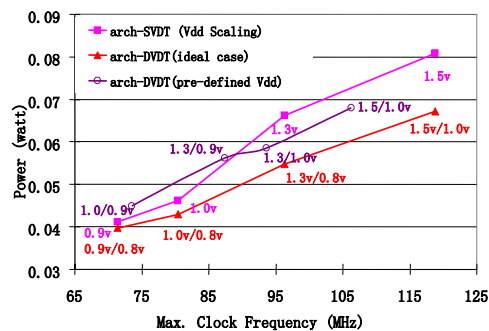
Step I: LUT mapping (FlowMap) + P & R assuming uniform VddH (using VPR)

Step II: *Dual-Vdd assignment* based on sensitivity

Step III: Timing driven P & R considering pre-defined dual-Vdd pattern (modified VPR)

Comparison Between Vdd-Scaling and Dual-Vdd

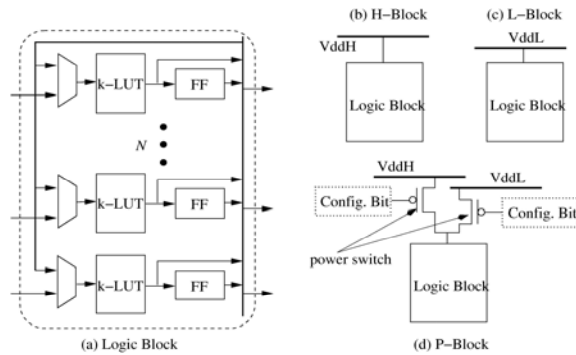
- For high clock frequency, dual Vdd achieves ~6% total power saving (~18% logic power saving)
- For low clock frequency, single-Vdd scaling is better
- Still a large gap between *ideal dual-Vdd* and *real case*
 - ◆ Ideal dual-Vdd is the result without layout pattern constraint



circuit: alu4

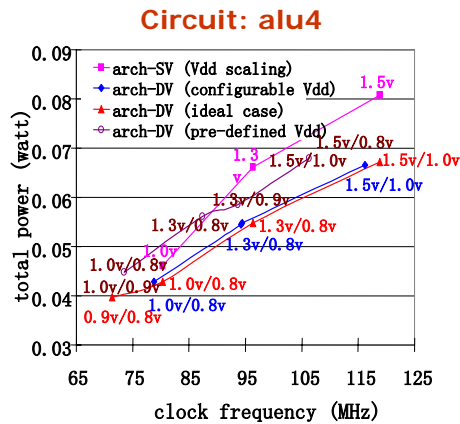
Vdd-Programmable Logic Block

- Power switches for Vdd selection and power gating
- One-bit control is needed for Vdd selection, but two-bit control power gating



Experimental Results with Vdd-Programmable Blocks

- Power v.s. performance



Outline

- Introduction
- Understanding Power Consumption in FPGAs
- Architecture Evaluation and Power Optimization
- Low Power Synthesis
- Conclusions

Low Power Synthesis for Dual Vdd FPGAs

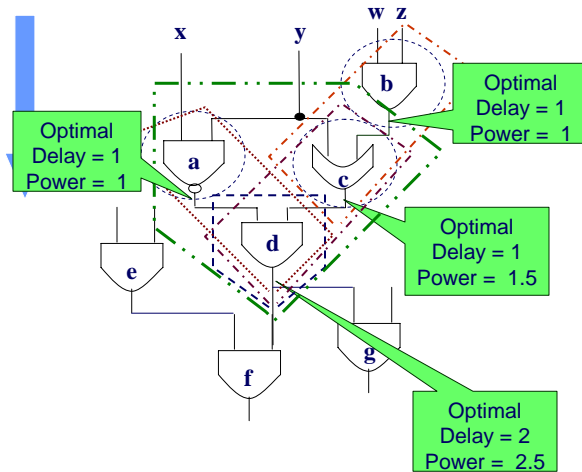
- FPGA architecture with dual-Vdds adds new layout constraints for synthesis tools
- Novel synthesis tools are required to support the architecture
 - ◆ Technology mapping [Chen, et al, FPGA'04]
 - ◆ Circuit clustering [Chen, et al, ISLPED'04]

Technology Mapping for Low-Power FPGAs with Dual Vdds

Cut Enumeration:

Topological Order from PIs to POs.

- Delay 1, Power 1
- Delay 2, Power 3.5
- Delay 2, Power 3.5
- Delay 2, Power 3.5
- Delay 2, Power 2.5

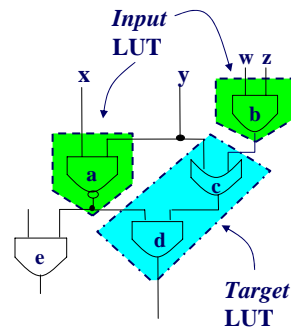


Represent 1 case: single high Vdd case

Dual-Vdd Cases

Four extra cases for dual-Vdd consideration

Cases	Input LUT	Target LUT	Converter
1	VddL	VddL	No
2	VddL	VddH	Yes
3	VddH	VddL	No
4	VddH	VddH	No



Consider:

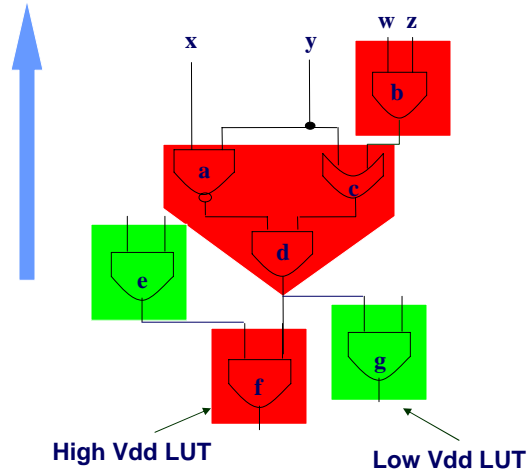
- Converter delay & power
- VddL LUT delay & power
- VddH LUT delay & power

Produce these four cases for each cut and node

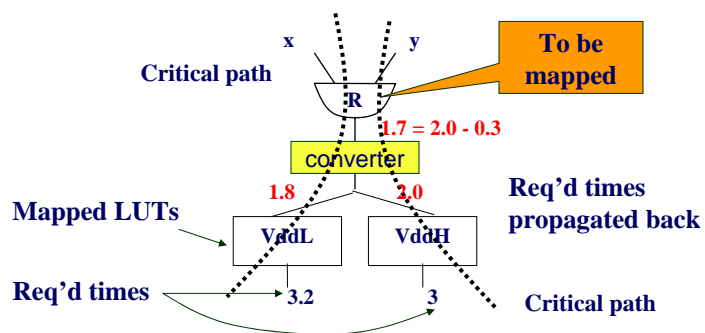
- More tradeoff solution points
- Smaller power requires larger delay
- Smaller delay requires larger power

Mapping Solution Generation

- From POs to PIs
- Critical path driven by VddH LUT
- Non-critical paths can be driven by VddL LUT, guided by low power



Two Types of Required Times



If R is using VddH:
Req'd time of R is 1.8

If R is using VddL:
Req'd time of R is 1.7

Each node maintains two req'd times:

- Propagated separately
- Interact with each other

Experimental Results

SVmap (Single high Vdd) compared to Emap [Lamoureux, ICCAD03]

Mapping area	Total edges	Est'ed power	Real power
- 4.04%	0.56%	- 1.29%	- 2.10%

- Mapping area considerably better
- Estimated power very close to the real power reported after P&R

DVmap (dual Vdd) compared to SVmap

SVmap	DVmap		
v1.3	v1.3 - v0.8	v1.3 - v0.9	v1.3 - v1.0
	- 11.63%	- 10.72%	- 9.44%

- v1.3 as VddH and v0.8 as VddL is the best combination

Circuit Clustering for Low-Power FPGAs with Dual Vdds

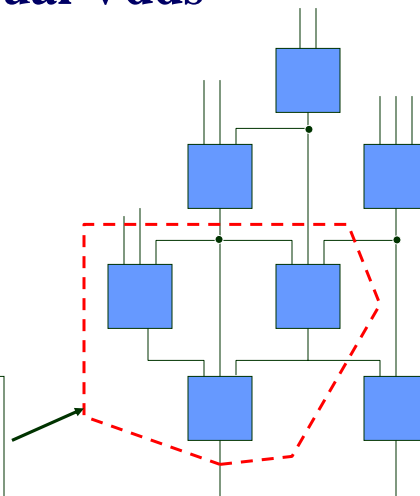
Given:

- Cluster Input $\leq K$
- Cluster Size $\leq M$
- Cluster Output $\leq M$
- LUT delay = d
- Edge delay = D

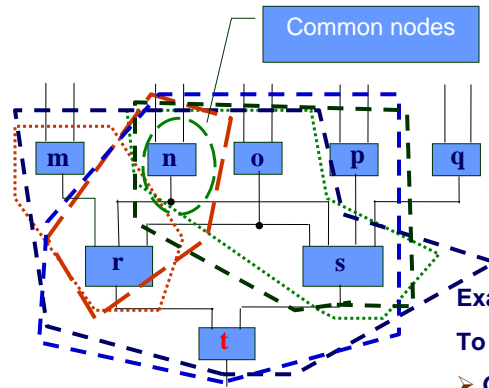
Goal:

- Optimal delay
- Minimum power

Example:
Input = 5
Size = 3
Output = 2



Cluster Enumeration



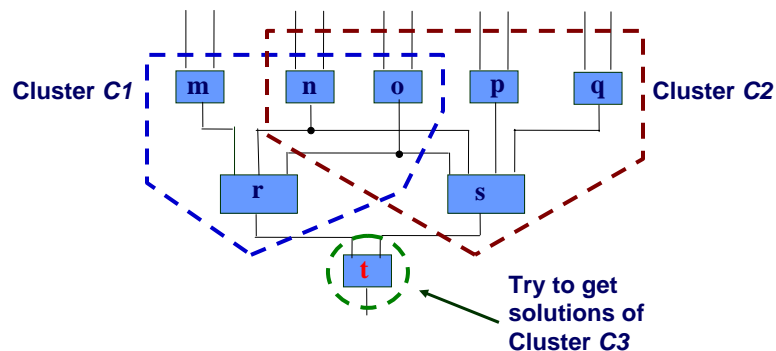
Example

To get a cluster of size 6 on LUT t

- Get 1 node on r , 4 on s , then merge with t ..., and
- Get 2 nodes on r , 3 on s ...
- Get 3 on r ...

- Handles common nodes
- Handles non-monotone property on the input constraint

Solution Propagation – An Example



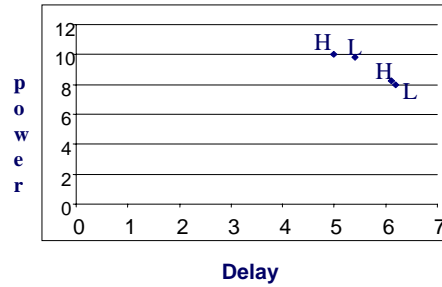
- Delay and power (form solution points) propagate through the clusters and nodes iteratively (similar to dual-Vdd mapping)
- All the good solutions are kept [Vaishnav, ICCAD'99]

Solution Curve of C1

Good solutions: Any two delay-power points ($D1, P1$) and ($D2, P2$)

- if $D1 > D2$, then $P1 < P2$
- if $D1 < D2$, then $P1 > P2$
- Each delay-power point has a Vdd setting

Delay	Power	Vdd
6.2	8	L
6.1	8.24	H
5.4	9.8	L
5	10	H

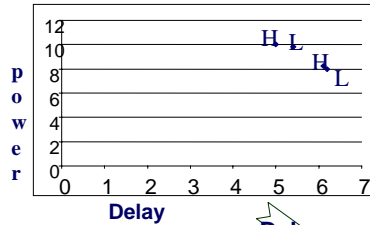


Good delay-power-vdd points

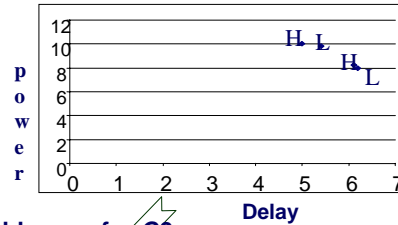
The corresponding solution curve

Curve Merging

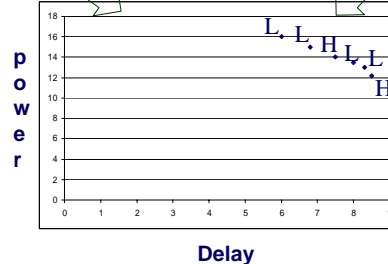
Delay-power-vdd curve for C1



Delay-power-vdd curve for C2



Delay-power-vdd curve for C3



- Consider:
- Converter
- VddL LUT
- VddH LUT
- Edge delay

- All the good solutions are generated
- All the inferior solutions are pruned away

Clustering Solution Generation

- Clustering solution generation follows the similar way as that in Dual-Vdd mapping procedure
- The amortized complexity of solution curve generation is quadratic on the order of the network depth
- Non-critical paths will be relaxed to accommodate low-Vdd clusters
- This algorithm is delay and power optimal for trees and delay optimal for directed acyclic graphs (DAGs) with dual-Vdd FPGAs

Experimental Results

Dual Vdd Clustering compared to Single Vdd Clustering

Single Vdd	Dual Vdd		
v1.3	v1.3 - v0.8	v1.3 - v0.9	v1.3 - v1.0
	- 21.8%	- 20.8%	- 19.6%

- v1.3 as VddH and v0.8 as VddL is the best combination

Outline

- **Introduction**
- **Understanding Power Consumption in FPGAs**
- **Architecture Evaluation and Power Optimization**
- **Low Power Synthesis**
- **Conclusions**

Conclusions

- **FPGA power consumption**
 - ◆ Majority on programmable interconnects
 - ◆ Leakage is significant
- **FPGA architecture optimization for power**
 - ◆ Architecture parameter tuning has a limited impact
 - ◆ Using high V_t for configuration SRAM cells is helpful
 - ◆ Using programmable dual Vdd for logic blocks is helpful
- **Power-efficient FPGA architectures introduce interesting CAD problems**
 - ◆ Dual-Vdd mapping
 - ◆ Dual-Vdd clustering

Up to 20% power saving reported using these algorithms