

EXPLOITING TEMPORAL IDLENESS TO REDUCE LEAKAGE POWER IN PROGRAMMABLE ARCHITECTURES

R. P. Bharadwaj, R. Konar, P. T. Balsara, D. Bhatia

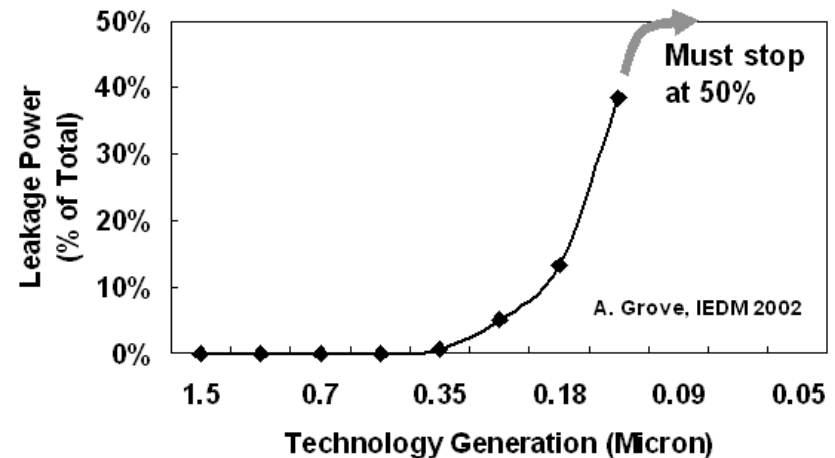
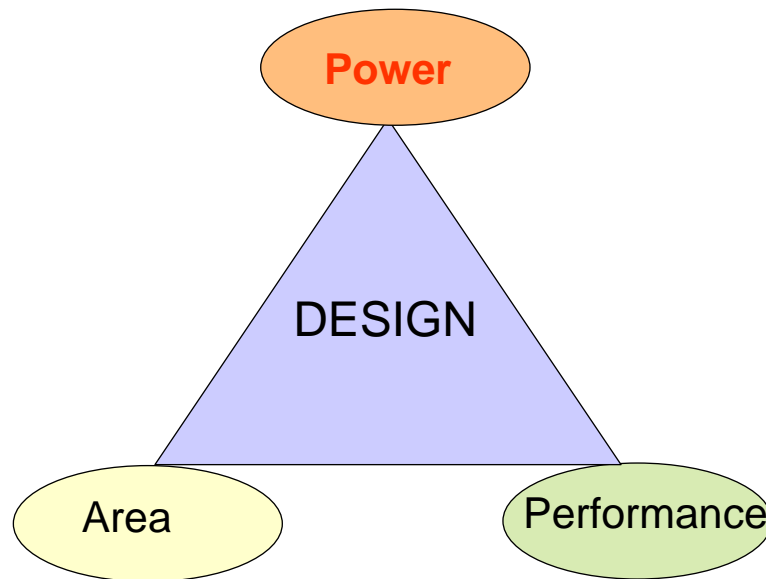
Center for Integrated Circuits and Systems
Erik Jonsson School of Engineering and Computer Science
University of Texas at Dallas

OUTLINE OF TALK

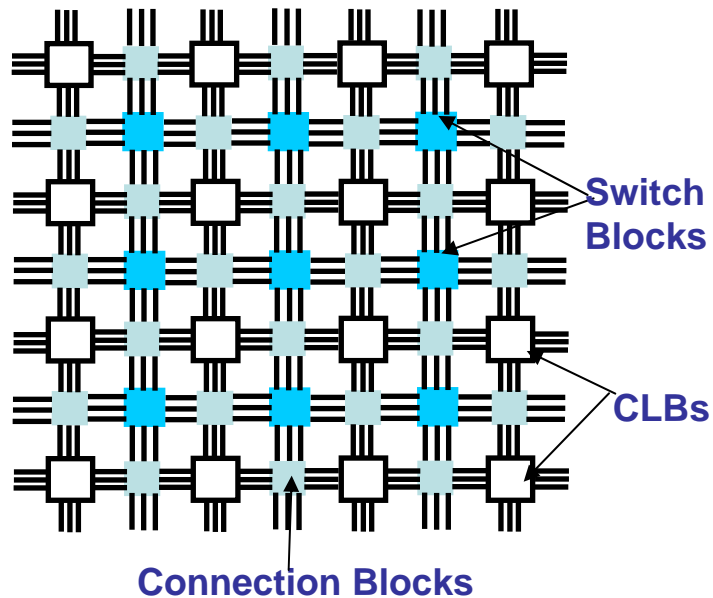
- ◆ Introduction.
- ◆ Motivation and Prior Work.
- ◆ Definitions and Preliminaries.
- ◆ Leakage Control based on Temporal Activity.
- ◆ Experimental Framework.
- ◆ Result and Analysis.
- ◆ Conclusion.

INTRODUCTION

- ◆ Power forms one of the vertices in the design optimization triangle.
- ◆ A significant portion of total power consumption in UDSM is Leakage power.
- ◆ Interaction between Leakage and Temperature may lead to a ***thermal breakdown***



POWER AND PROGRAMMABLE ARCHITECTURES

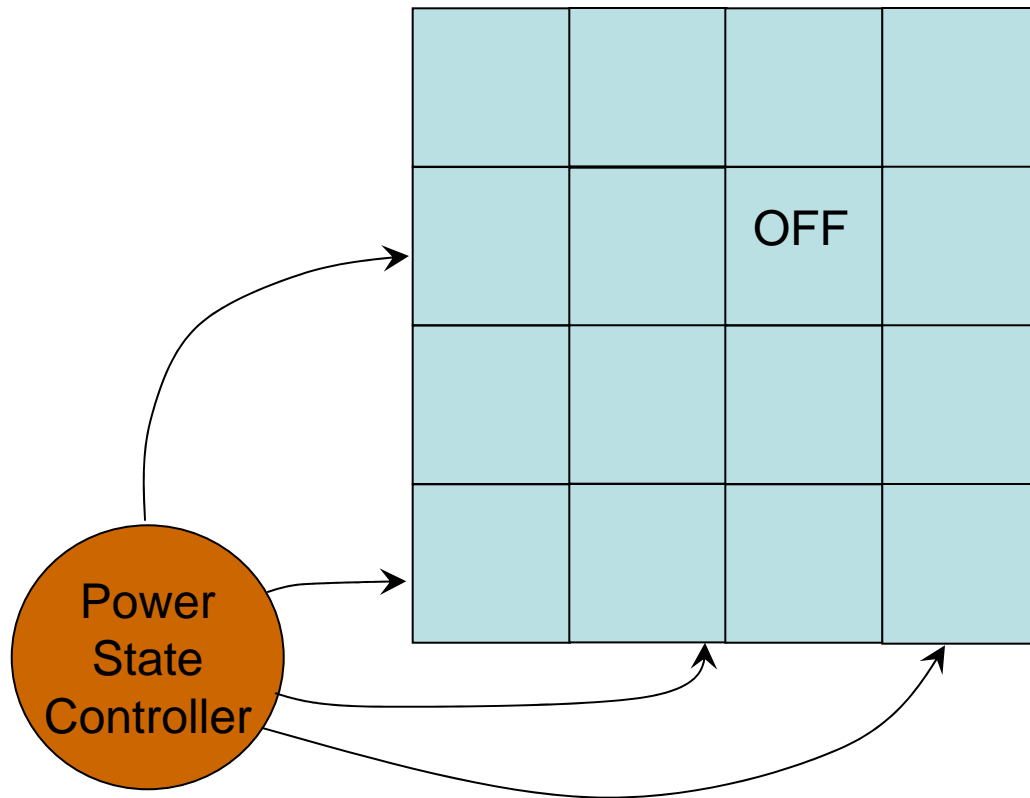


An island style FPGA architecture

Leakage has to be a key design variable in the Programmable Fabric

- ◆ FPGAs tradeoff area, power and performance, for configurability.
- ◆ Flexibility comes at the cost of extra transistors in CLBs, Switch Blocks, Connection Blocks etc.
- ◆ In UDSM, extra transistors lead to excessive power, out of which considerable portion is from leakage.
- ◆ Sources of leakage in FPGAs.
 - ◆ Redundant transistors
 - ◆ Spatial Underutilization
 - ◆ Temporal Underutilization

OPPORTUNITIES OF LEAKAGE POWER SAVINGS



Analyze, Partition, and Map Temporal dependencies

PRIOR WORK

- ◆ Tuan et. al.
 - ◆ Analyzed in detail, leakage of various circuit blocks in 90nm FPGAs. [CICC 2002]
 - ◆ Suggested CAD support for clustering unused logic from used ones.
- ◆ Gayasen et al. [FPGA 2004]
 - ◆ FPGA fabric divided into regions of slices.
 - ◆ Constrain placement to occupy specific regions.
 - ◆ Cutoff power to the unused regions.
- ◆ Rahman et al. [FPGA 2004]
 - ◆ Explored circuit level techniques to reduce leakage in multiplexers of switch boxes.
 - ◆ Suggested switchboxes with different V_{th} transistors to control leakage.
- ◆ Anderson et al. [FPGA 2004]
 - ◆ FPGA hardware structures leak less for a '1' than '0'.
 - ◆ Change the polarity of signals such that maximum time is spent in low leakage state.

CONTINUED...

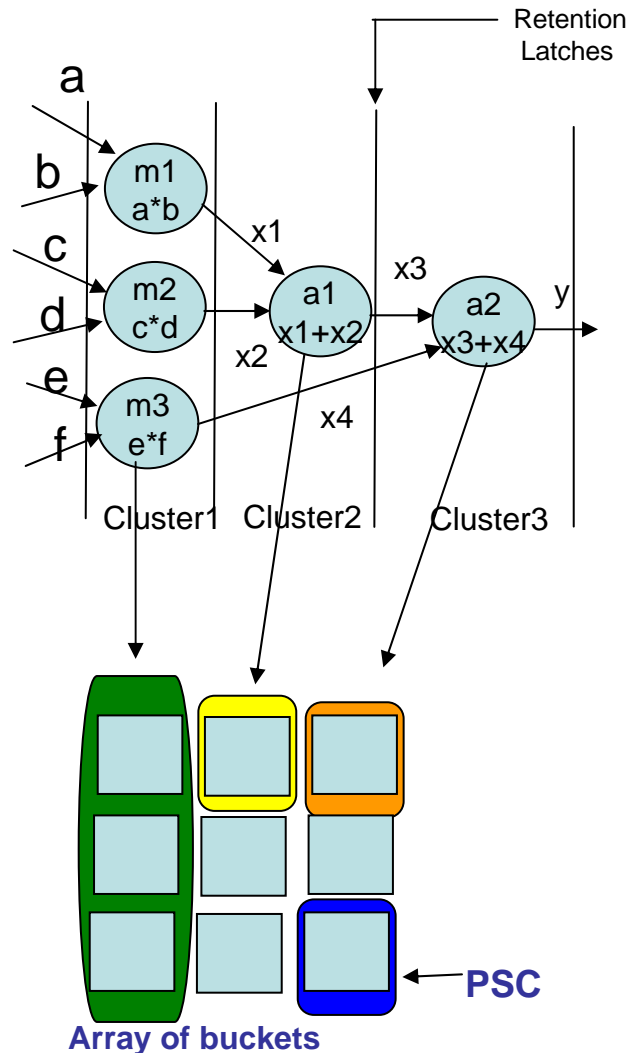
- ◆ Calhoun et al. [ISPLED 2003]
 - ◆ A low power FPGA core was designed using fine grained leakage control.
- ◆ Our primary focus is on
 - ◆ Exploiting temporal underutilization of a design to group them into clusters.
 - ◆ Converting these temporal clusters into spatial clusters by physically mapping them into specific regions in the FPGA layout.
 - ◆ Automatically synthesizing a Power State Controller, based upon cluster characteristics for controlling switching on/off the regions.
 - ◆ We assume a MTCMOS based architecture, in which the regions can be selectively turned on/off.

LEAKAGE CONTROL BASED ON TEMPORAL ACTIVITY

TEMPORAL PARTITIONING:

- ◆ We consider applications represented by Data Flow Graph $G(V,E)$.
 - ◆ V is vertex set representing the set of operations in the DAG
 - ◆ edge set E are the dependencies among operations represented by V .
- ◆ Create different partitions of $G(V,E)$, and map their temporal proximities to spatial proximities.
 - ◆ Level $I(v_i)$ for each node v_i is identified through topological sort.
 - ◆ A user defined parameter S is provided as the number of temporal partition in which the design must be divided.
 - ◆ Graph $G(V,E)$ is divided into S temporal partitions such that for any two nodes v_I and v_J whose $I(v_I) < I(v_J)$, temporal slots T_I, T_J must satisfy $T_i \leq T_j$

EXAMPLE



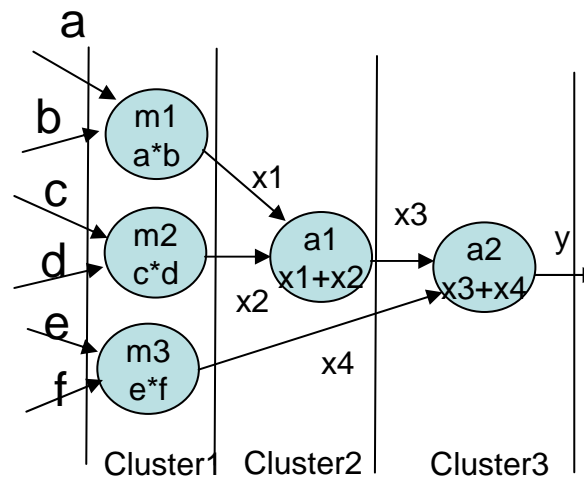
- In the DAG for dot product of two vectors, the output y is given by $y = a \times b + c \times d + e \times f$
- On topological sort, m1, m2, m3 have level 1, a1 has level 2, a2 has level 3.
- If $S=3$, m1, m2, m3 will be scheduled for first timeslot, a1 for second, a2 for third. As multipliers occupy more slices, partition will have imbalances.

POWER STATE CONTROLLER

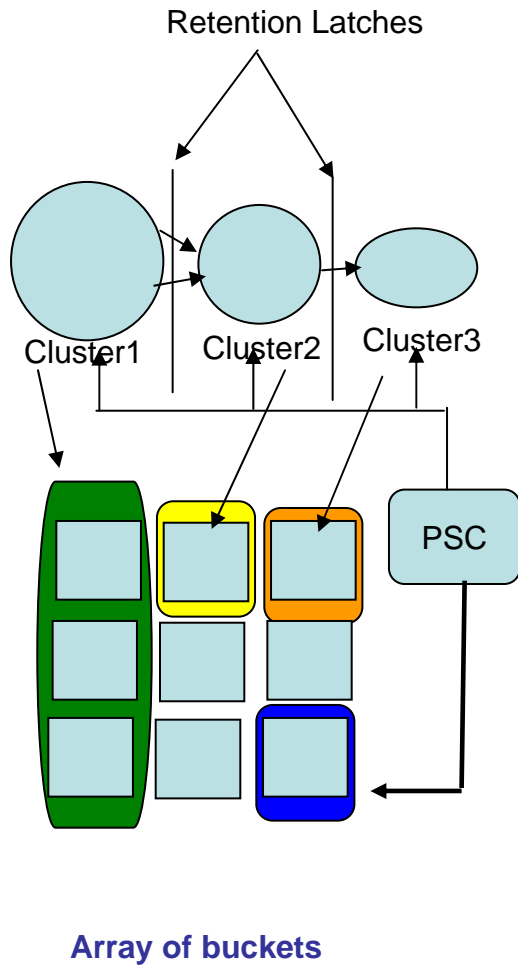
- ◆ Once temporal schedules for various clusters are identified, Power State Controller (PSC) is synthesized to control the sequence of turning on/off the clusters.
- ◆ PSC is a synchronous Mealy machine that cycles through the sequence of states, turning on one cluster and turning off rest off the clusters.
- ◆ PSC is synthesized along with the design and integrated into the netlist.
- ◆ No Dynamic Reconfiguration overhead to control the switching of the clusters. Saves a considerable amount of performance overhead.

RETENTION OF VALUES

- ◆ Whenever there are data dependencies among the clusters, we need to save the output of the active cluster before turning it off.
 - ◆ use retention latches.
 - ◆ low leakage latches are designed to hold values with very little performance overhead.
 - ◆ add registers at the cluster boundaries for retaining the data.



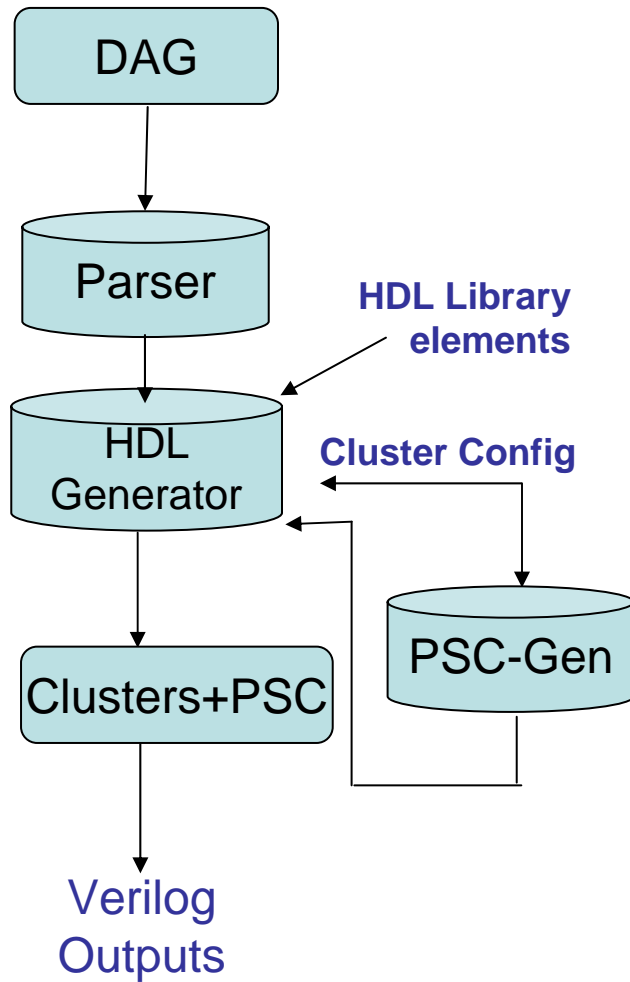
PLACEMENT AND ROUTING



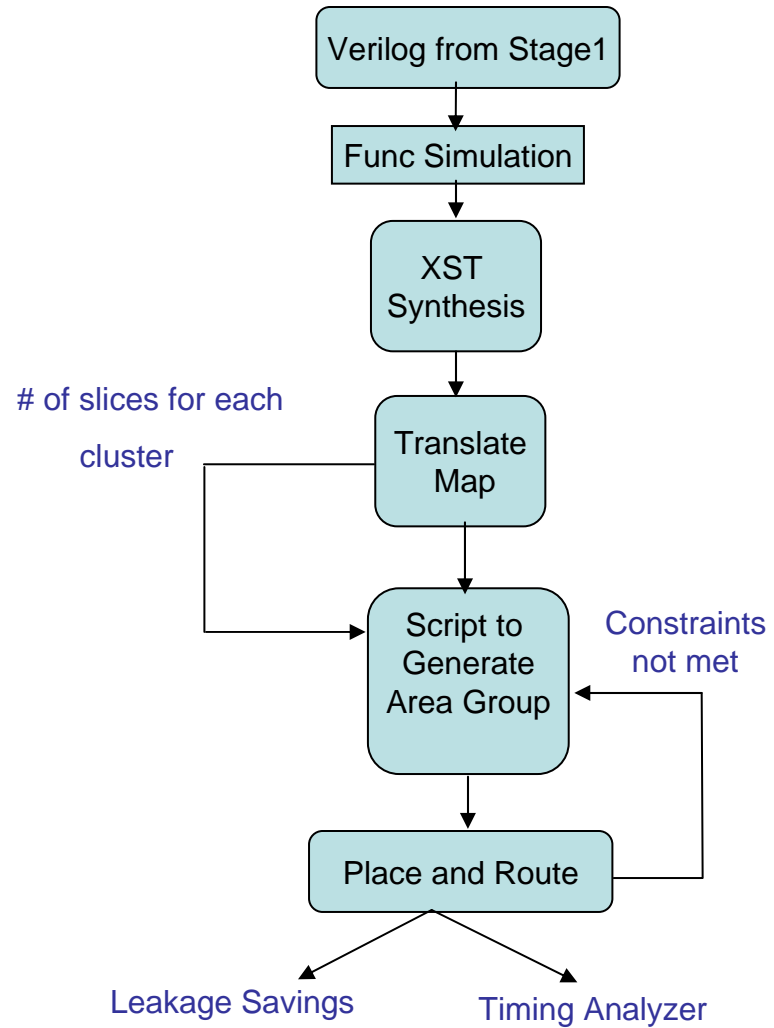
- ◆ Temporal proximities of the clusters have to be converted into spatial proximities.
- ◆ Mapping clusters into contiguous buckets in the programmable fabric restricts spreading of the design all over the layout.
- ◆ This approach maximizes the buckets that can be turned off.
- ◆ This regular arrangement compared to randomly spread buckets reduces the routing stress from the PSC.
- ◆ The PSC can be placed into any desired location.

FRAME WORK

STAGE 1



STAGE 2



RESULTS AND ANALYSIS

| Benchmarks | #Clusters | (%)Area Imbalance | Average # Slices Shutdown | % Leakage Savings | | % Area Overhead | | PSC Time Period (nsecs.) | |
|------------------------------|-----------|--------------------|---------------------------|-------------------|-------|-----------------|-------|--------------------------|--------|
| | | | | V | H | V | H | V | H |
| 1. Binary Tree Comparator | 6 | 41.67 | 81 | 36.49 | 36.49 | 17.8 | 17 | 7.02 | 7.25 |
| | 5 | 40.1 | 77 | 35 | 35 | 15.75 | 15.7 | 11.82 | 12.95 |
| | 4 | 35.6 | 72.6 | 32.7 | 32.7 | 13.6 | 13.6 | 16.2 | 15.22 |
| | 3 | 25 | 64.7 | 29.13 | 29.13 | 11.8 | 11.58 | 22.06 | 21.25 |
| | 2 | 1 | 48.4 | 21.8 | 21.8 | 11.58 | 9.5 | 28.08 | 27.1 |
| 2. 5x5 Median Filter | 7 | 3.6 | 85 | 38.3 | 38.3 | 5.6 | 5.6 | 71.6 | 66.5 |
| | 5 | 13.14 | 79.5 | 35.8 | 35.7 | 5.2 | 5 | 60.81 | 52.8 |
| | 4 | 28.3 | 74.4 | 33.5 | 33 | 4.62 | 4.4 | 70.53 | 56.05 |
| | 3 | 40 | 66.2 | 29.8 | 29.8 | 1.7 | 1.7 | 58.53 | 61.5 |
| 3. 4x4 Matrix Multiplication | 3 | 62 | 66 | 29.8 | 23.3 | 17 | 13.4 | 8.03 | 14.34 |
| | 2 | 50 | 49.7 | 22.4 | 22.38 | 18 | 12.65 | 10.20 | 15.41 |
| 4. IIR filter | 3 | 71.5 | 63 | 28.6 | 28.9 | 15.8 | 30 | 10.26 | 10.816 |
| | 2 | 70 | 38 | 17.1 | 17.1 | 13.6 | 13.6 | 11.58 | 16.8 |

RESULTS AND ANALYSIS CONTINUED...

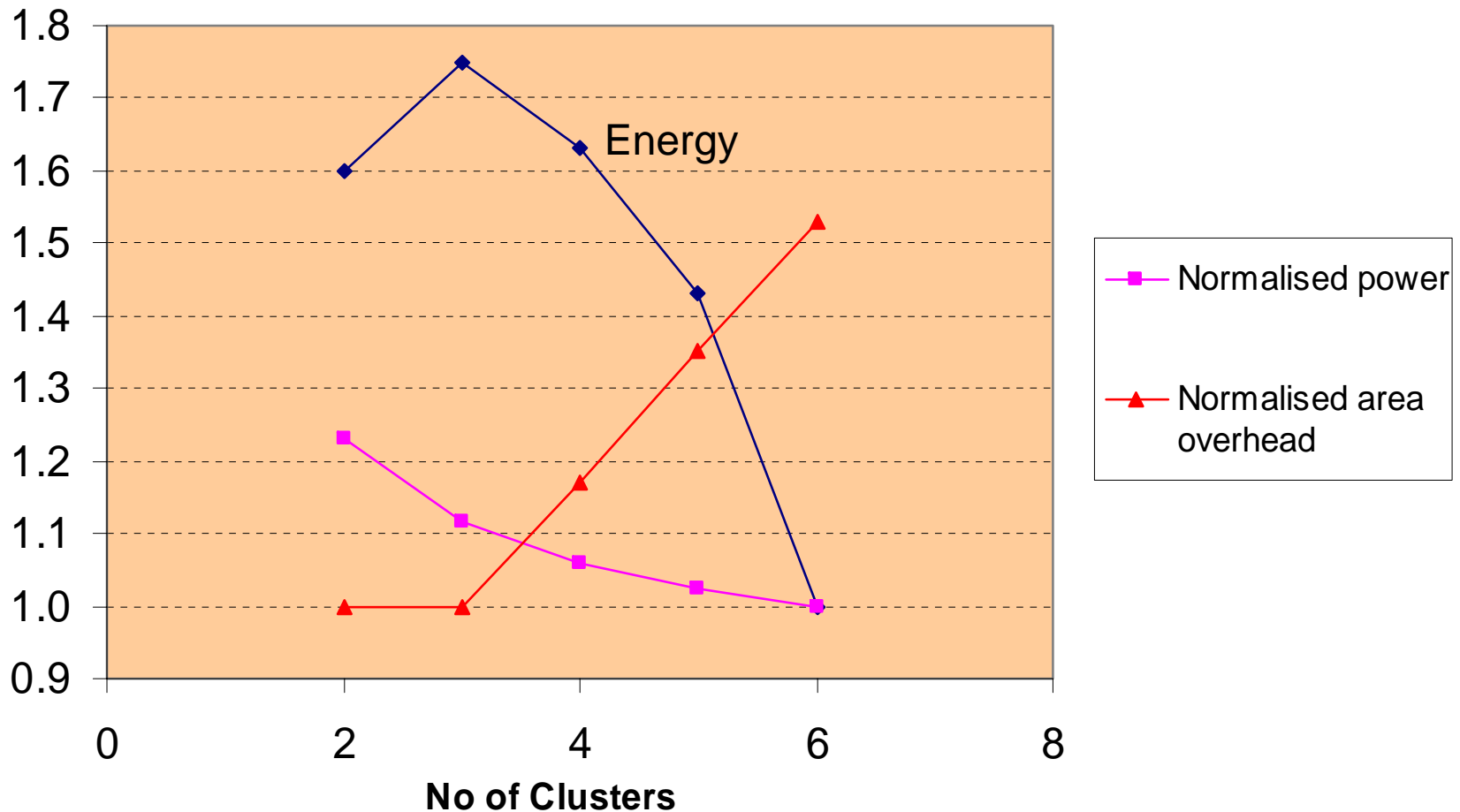
- ◆ Area imbalance is defined as the difference in percentage of slices occupied by the largest and the smallest cluster.
- ◆ Area overhead is expressed in terms of percentage of extra slices.
- ◆ Key Observations
 - ◆ Leakage Savings increases with fine grain clustering, due to finer control in shutting off the clusters.
 - ◆ Area Overhead increases with Leakage Savings, as there will be more buckets with unutilized slices.
 - ◆ Area Imbalance does not affect leakage savings that much, whereas with the increase of average number of slices that can be shut down, leakage savings increases.
 - ◆ Arrangement of buckets, cluster size and delay of each cluster affects PSC time period. This may lead to longer execution time.

RESULT AND ANALYSIS CONTINUED..

- ◆ Leakage Savings increases with fine grain clustering, due to finer control in shutting off the clusters.
- ◆ Area Overhead increases with Leakage Savings, as there will be more buckets with unutilized slices.
- ◆ Area Imbalance does not affect leakage savings that much, whereas with the increase of average number of slices that can be shut down, leakage savings increases.
- ◆ Arrangement of buckets, cluster size and delay of each cluster affects PSC time period. This may lead to longer execution time.
Hence there will be sweet spots in trade-off curve between energy, power and area.

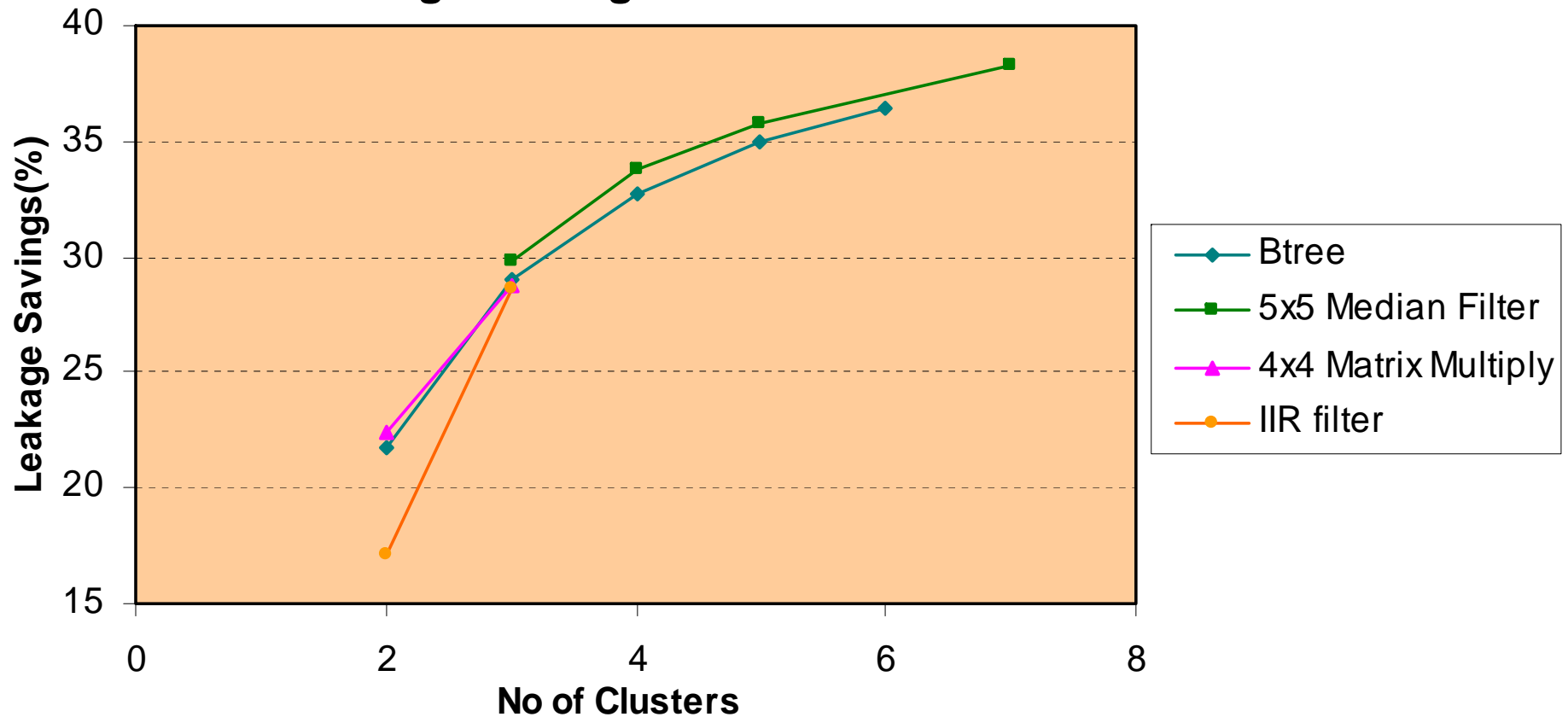
ENERGY-AREA-LEAKAGE POWER TRADEOFF FOR BENCHMARK #1

Energy Area Power Tradeoff



REDUCTION IN LEAKAGE POWER WITH FINE GRAIN CLUSTERING

Leakage Savings with Clusters



CONCLUSIONS

- ◆ We proposed a new design paradigm that exploit temporal idleness to break the design into temporal clusters such that each cluster is active in a particular temporal window.
- ◆ We showed that a Power State Controller can be synthesized to switch on/off these clusters, along with the design with minimal area overhead.
- ◆ Good amount of Leakage Power can be saved if temporal idleness is exploited efficiently.



**THANK
YOU!**