

# System-level Exploration of Dynamical Clustering for Adaptive Power Management in Network-on-chip

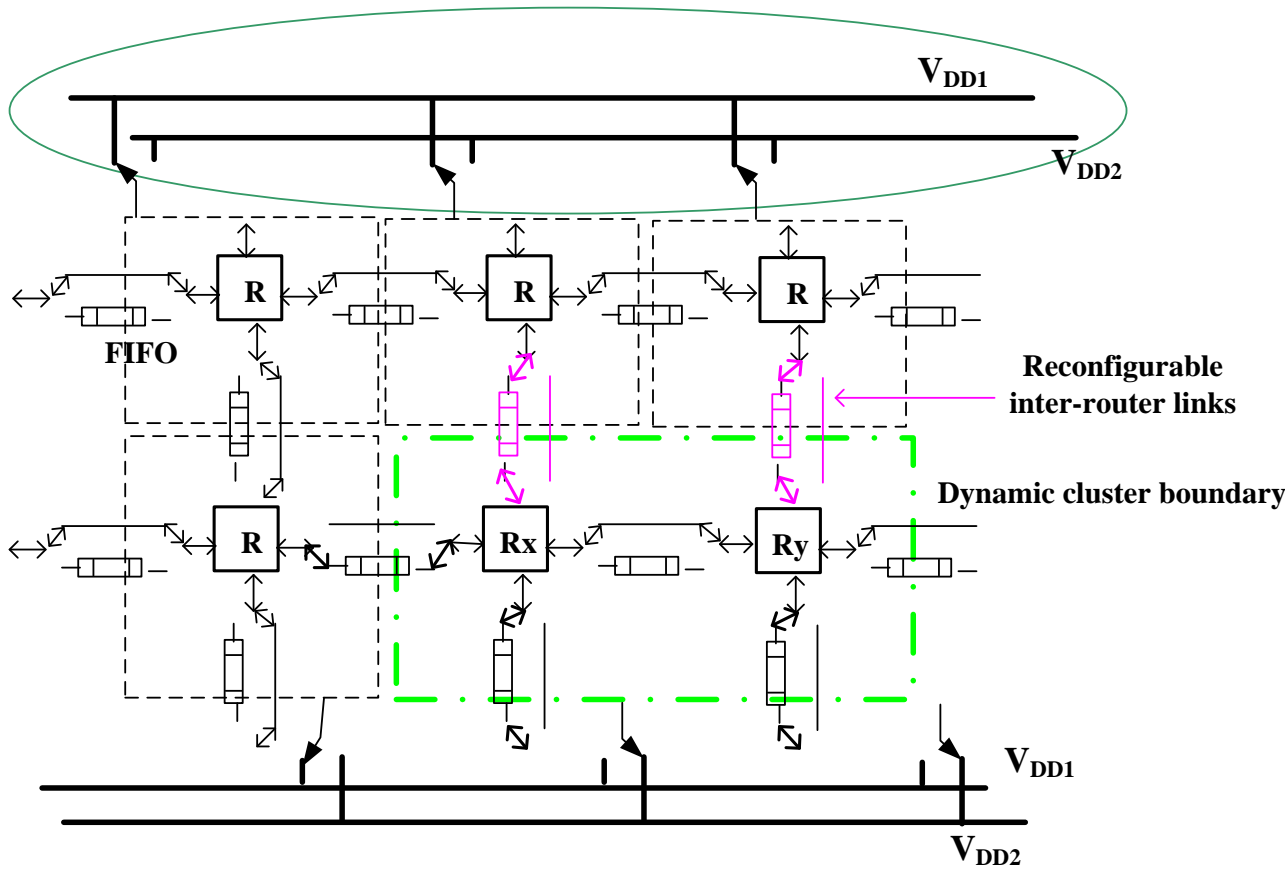
Liang Guang, Ethiopia Nigussie, Hannu Tenhunen,  
Dep. of Information Technology,  
University of Turku, Finland

# Introduction

- **Many-core platform with NoC** as the communication structure is steadily growing. More cores are being integrated with simpler each core being simpler. Examples: Teraflop 80-core, Tileria 64-core, ASAP 167-core.
- Realizing **multiple voltage and frequency islands** is an effective method to provide high power efficiency, as the workload in massively parallel platform has temporal and spatial variations.
- **Global communication** between cores is a major power consumer. Its contribution will constantly increase with the platform further parallelized into smaller units connected by a larger communication network.
- This work is an innovative yet initial exploration of realizing **dynamically clustered power management in many-core systems**. Integrating supporting power delivery and clocking techniques, clusters can be reconfigured at the real-time to tradeoff power and performance with minimized latency and power overhead.

# System Architecture

Multiple On-chip Power Networks



**Network regions  
dynamically configured  
into power domains**

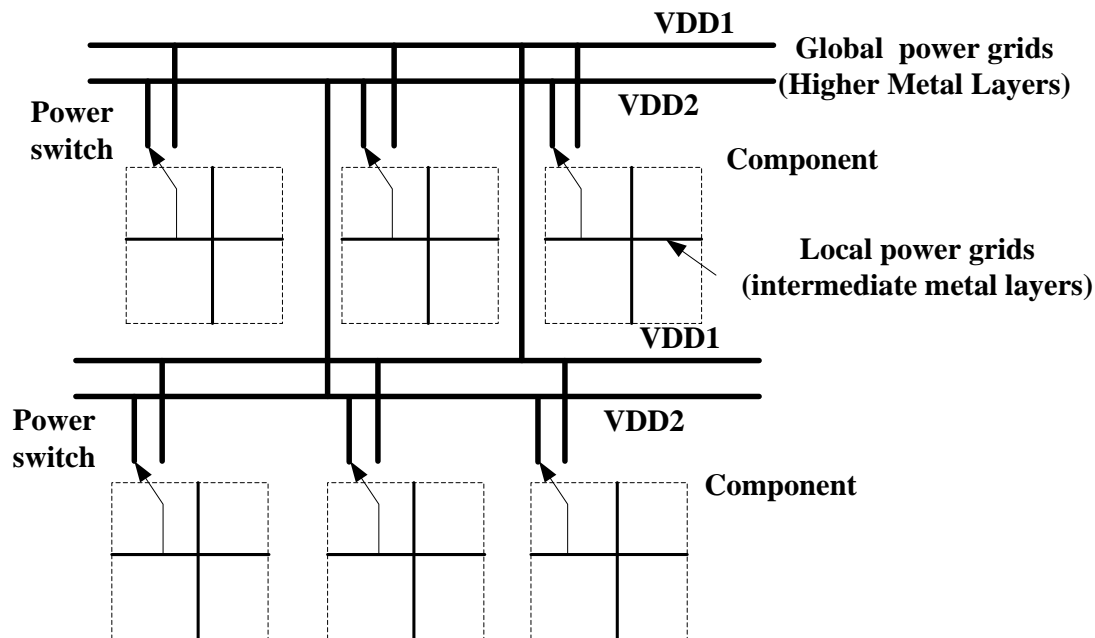
supported by

✓ *Multiple on-chip power  
delivery networks*

✓ *Reconfigurable  
inter-router links*

# Multiple On-chip PDN(Power Delivery Networks)

- A scalable approach to provide adaptive power domain configuration
- Used in ASAP 167-core NoC (Truong et al. 2009)



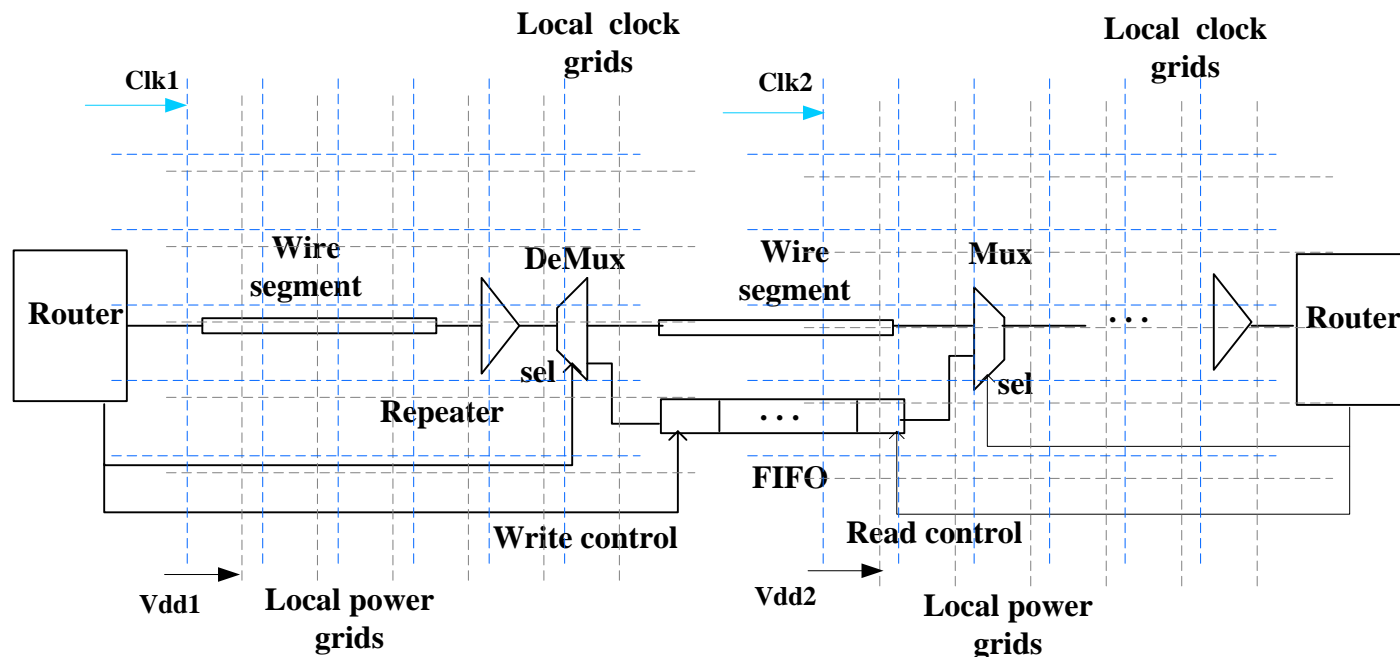
- ASAP prototype results: 7 power grids are fabricated on M6/7 metal layers. The power switch only accounts for 4% in each tile's area.

*(Truong et al. 2009) A 167-processor computational platform in 65nm CMOS. JSSC 44(4):1130-1144, 2009*

# Reconfigurable Inter-Router Links (1)

Adaptive inter-router link structure reconfigurable for different power domain settings:

- ✓ *In case both ends are configured into the same power domain, normal wire channels are enabled to minimize*
- ✓ *In case the ends are configured into different power domains, bi-synchronous FIFOs are needed for synchronization.*

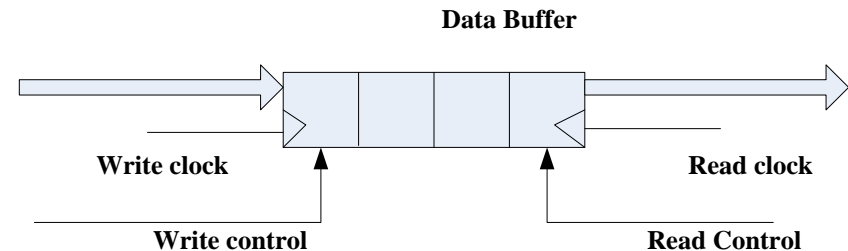


# Reconfigurable Inter-Router Links (2)

- **Bi-synchronous FIFO**
  - ✓ The synchronization manner most convenient for CAD flow integration (for example DSPIN NoC)
  - ✓ The more different clockings at the two ends are, the deeper FIFO is required to minimize metastability while ensuring certain throughput (Panades et al. 2007)
- **Pseudochronous /Quasi-synchronous clocking**
  - ✓ A special mesochronous timing with predictable and controllable constant phase shift between two adjacent nodes on regular layout NoC (öberg 2003)
  - ✓ Used when two adjacent network regions configured with the same frequency
  - ✓ Controllable skew without metastability issues .

Panades et al. 2007, *Bi-synchronous FIFO for Synchronous Circuit communication Well Suited for NoC in GALS structures*. In *Proc. of NOCS2007*.

Öberg 2003, *Clocking Strategies for Networks-on-Chip, Networks on Chip*, 153-172, *Kluwer Academics Publishers*



Simplified view of bi-synchronous FIFO, highlighting most power-hungry datapath

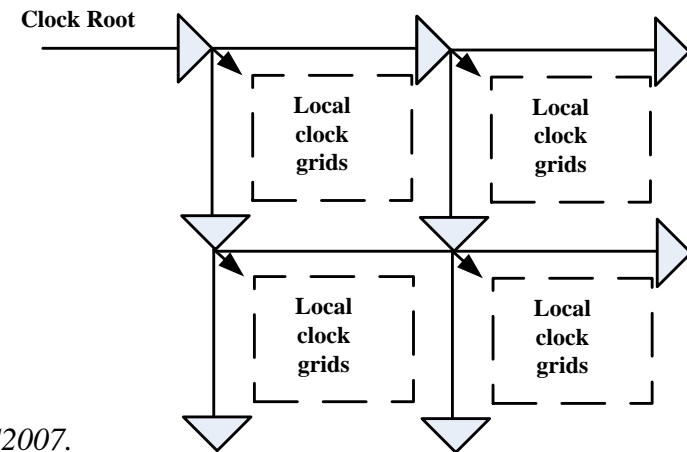
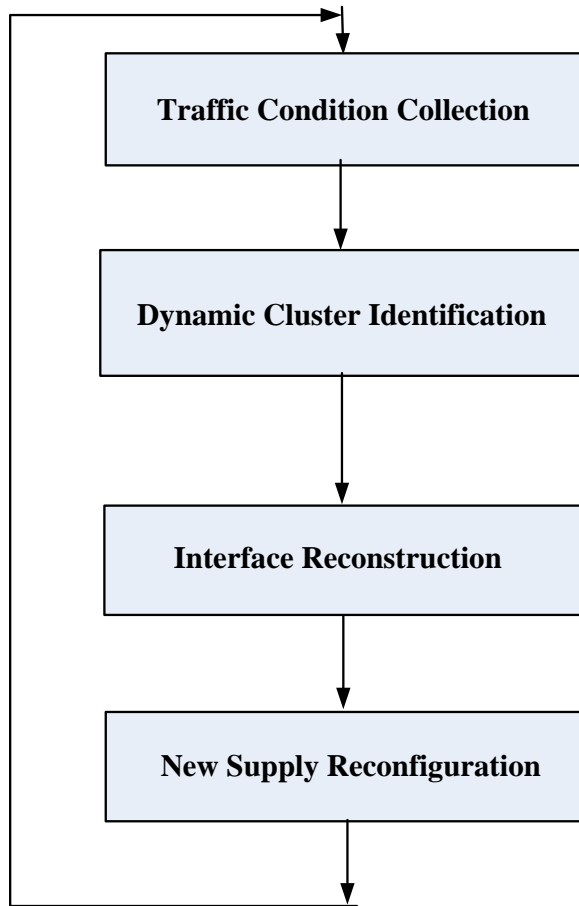


Illustration of Pseudochronous clocking (öberg 2003)

# Dynamic Clusterization Steps (1)



- 1) The traffic condition of each region needs to be collected**
- 2) Dynamic clusters are identified**
- 3) The boundary links of the clusters are configured with FIFO-based channels**
- 4) Switching to the proper Vdd and clock**

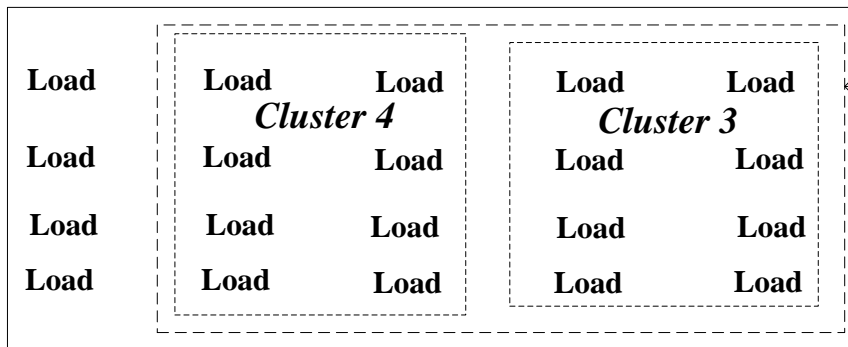
# Dynamic Clusterization Steps (2)

## 1) Run-time traffic condition collection

- ✓ The traffic load of each region, averaged in a history window needs to be collected by a central monitor
- ✓ Such traffic load reporting will be generalized into monitoring flow. With relatively long reporting interval, the overhead is minimal. The detailed implementation is initially explored in (Guang et al. 2008)

## 2) Dynamic cluster identification

### *Cluster 1*



### *Cluster 2*

- ✓ Search for the largest cluster (minimizing the interface overhead)
- ✓ Managed by the central monitor with the traffic information collected

*Guang et al. 2008, Low-latency and Energy-efficient Monitoring Interconnect for Hierarchical-agent-monitored NoCs. In Proc. Norchip 2008.*



# Dynamic Clusterization Steps (3)

## 3) **Interface reconstruction**

- ✓ The links on the boundaries of the identified clusters need to enable FIFO-based connection.
- ✓ The reconstruction has to be done before switching to new Vdd and clocking.

## 4) **New supply reconfiguration**

- ✓ Reconfigure the power switches to the proper Vdd, and the PLLs with proper clocking output.

# Experiment Setup (1)

- **Network Configuration**

- ✓ 8\*8 mesh NoC, STF switching, X-Y routing
- ✓ 64-bit wires, 1mm long
- ✓ FIFO depth 6 (to ensure 100% throughput in asynchronous timing; Panades et al. 2007)

- **Power Estimation**

- ✓ Two voltage/frequency pairs (0.6G, 0.6V), (1.2G, 1.5)
- ✓ Router and normal wiring energy estimated by Orion 2.0
- ✓ FIFO access energy estimated by the buffer energy in a router, latency modelled by Panades et al. 2007.

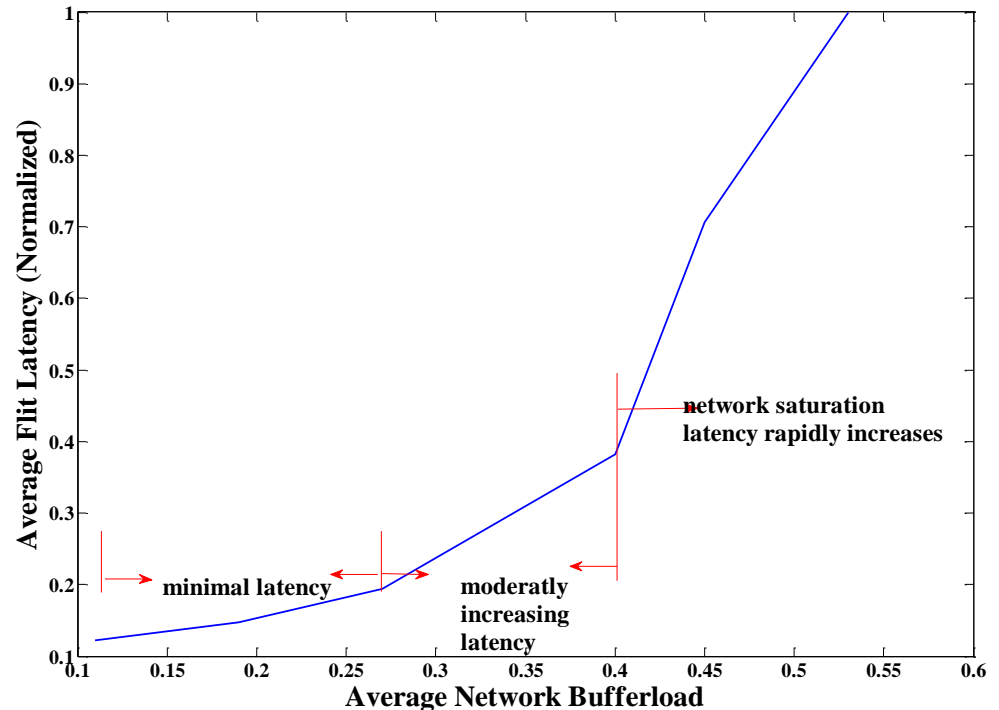
- **DVFS algorithm setting**

- ✓ The traffic load is averaged and reported every 50 cycles
- ✓ By default, the low voltage/frequency pair is used. When the average buffer load is above a threshold, the high voltage/frequency pair is used.

# Experiment Setup (2)

## Energy/performance tradeoff monitoring buffer load (Guang&Jantsch2006)

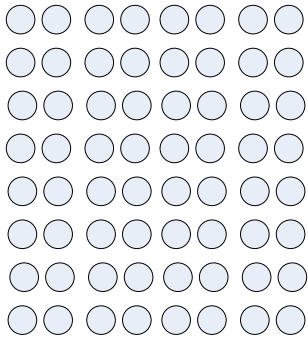
- ✓ Buffer load is a simple and direct indicator of the network performance.
- ✓ Lower frequency leads to higher buffer load (given same input traffic), with lower energy consumption.
- ✓ The exact curve of buffer load vs. latency varies based on the network configuration
- ✓ The tradeoff is dependent by the latency tolerance of the processing elements.



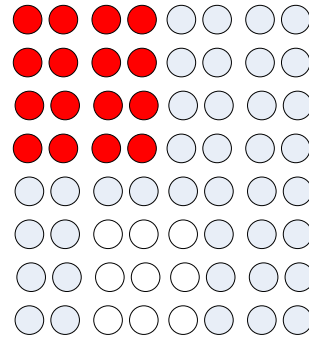
## Buffer Load vs. Latency (8\*8 NoC, STF switching, X-Y routing)

Guang&Jantsch 2006, Adaptive power management for the on-chip communication network, In Proc. of DSD2006.

# Traffic Patterns



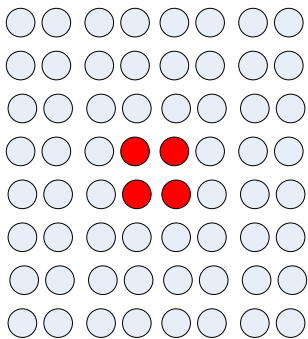
Type 1. Uniform Traffic



Type 2. Hotspot Traffic

Type 3. Hotspot Traffic

(as Type 2), but with locality destination pattern ( Lu et al. 2008)



Type 4. Hotspot traffic with a different hotspot location

Type 5: Same spatial variation as Type 4, but with a higher input traffic

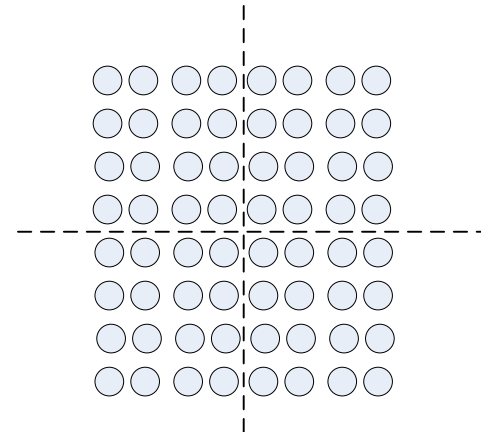
Type 6: Same spatial variation as Type 5, but with even higher input traffic

*Lu et al. 2008. Network-on-chip benchmarking specification part 2: Microbenchmark specification version 1.0. Technical report, OCP International Partnership Association, 2008.*

# Evaluation (1)

## Alternative Architectures

- **PNDVFS (Per-Network DVFS)**
  - ✓ The whole NoC is configured with lower power supply if the general traffic load is low
  - ✓ Most simple manner of DVFS with no synchronization overhead (Guang&Jantsch 2006)
- **SCDVFS (Static-clustered DVFS)**
  - ✓ Clusters are partitioned at design time. (Guang et al. 2008)
- **Per-core DVFS**
  - ✓ Conventional per-core DVFS with static synchronization interface is too "expensive".
  - ✓ Potential per-core DVFS with reconfigurable links requires further analysis in avoiding frequent scaling.



Uniform partition for SCDVFS

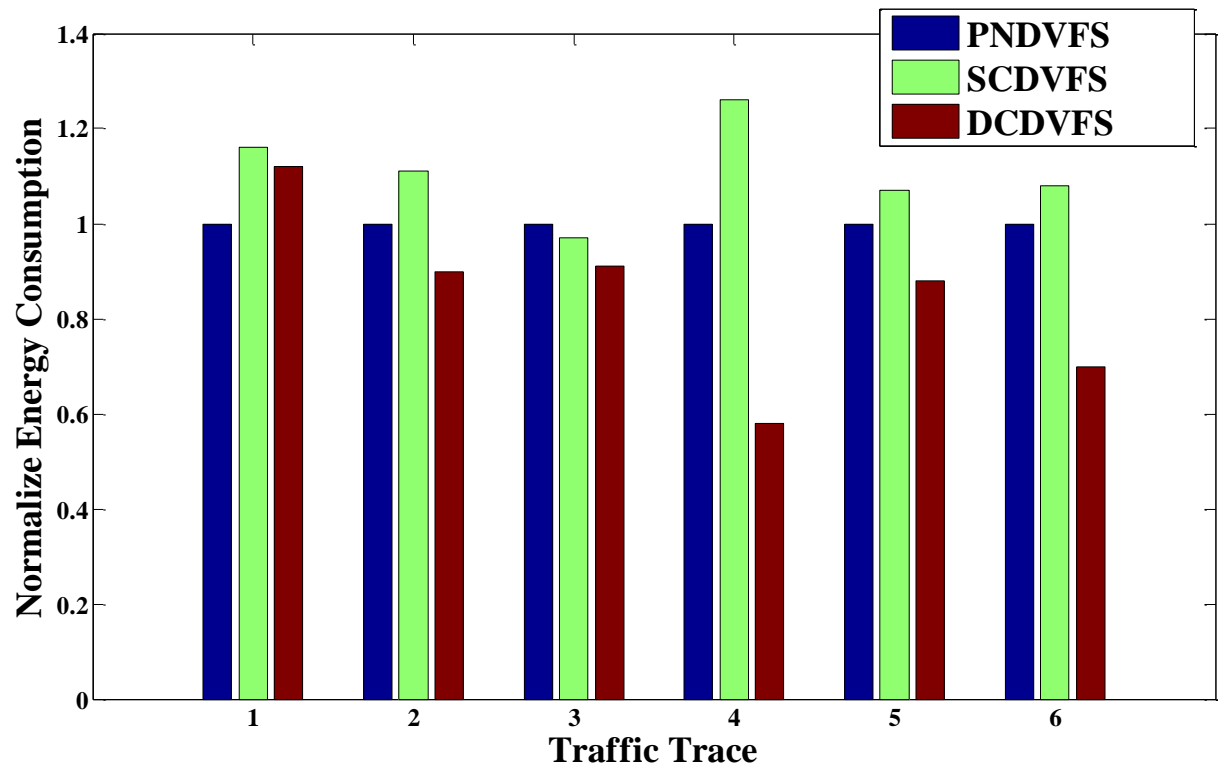
	Average Energy Per-flit (e-10J)	Average Latency Per-flit (Cycles)
Router + Link	6.24	16.83
FIFO	1.96	18.33
Increase	31%	112%

Initial Exploration of Overheads using Conventional Per-core DVFS

# Evaluation (2)

- **Energy comparison**

- ✓ In general, DCDVFS achieves lower average energy
- ✓ Except for uniform traffic with no spatial or temporal variation, FIFO overhead leads to more energy consumption
- ✓ More varying and unpredictably distributed the traffic, the higher energy benefit (T4-T6)
- ✓ The major overhead comes from the FIFO.

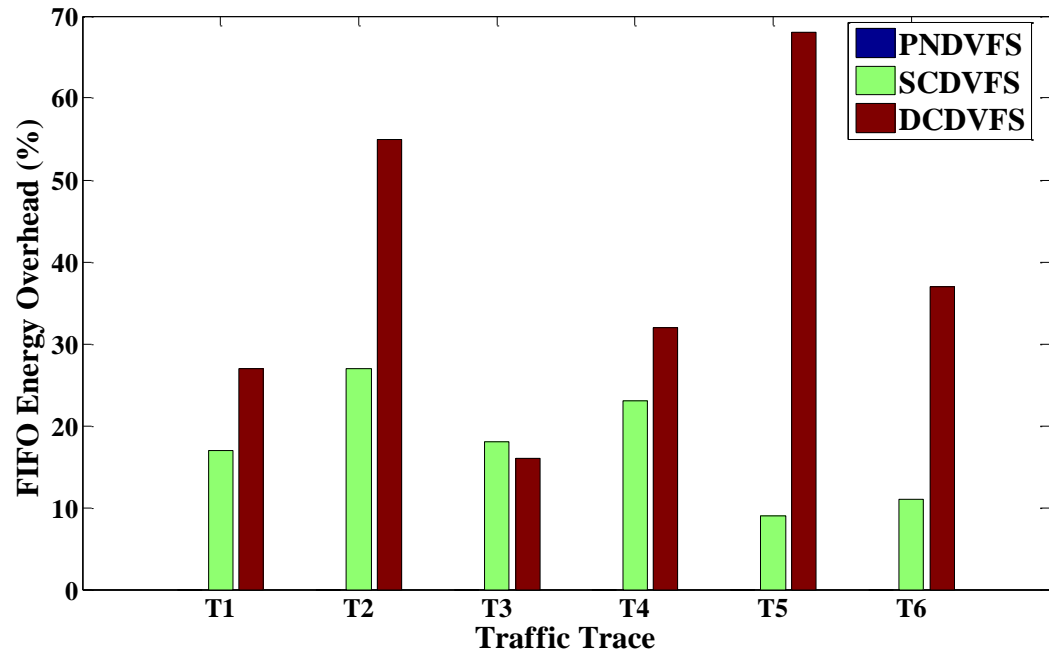


Comparison of Average Energy (Normalized) of Three DVFS Architectures

# Evaluation (3)

## FIFO energy overhead

- ✓ For DCDVFS, the FIFO contributes to significant energy overhead
- ✓ Despite such overhead, the energy is still lowered because of lowered running frequency
- ✓ For SCDVFS, the FIFO contributes smaller percentage of energy, due the larger cluster size
- ✓ No FIFO exists for PNDVFS



FIFO energy overhead for three DVFS architectures

# Evaluation (4)

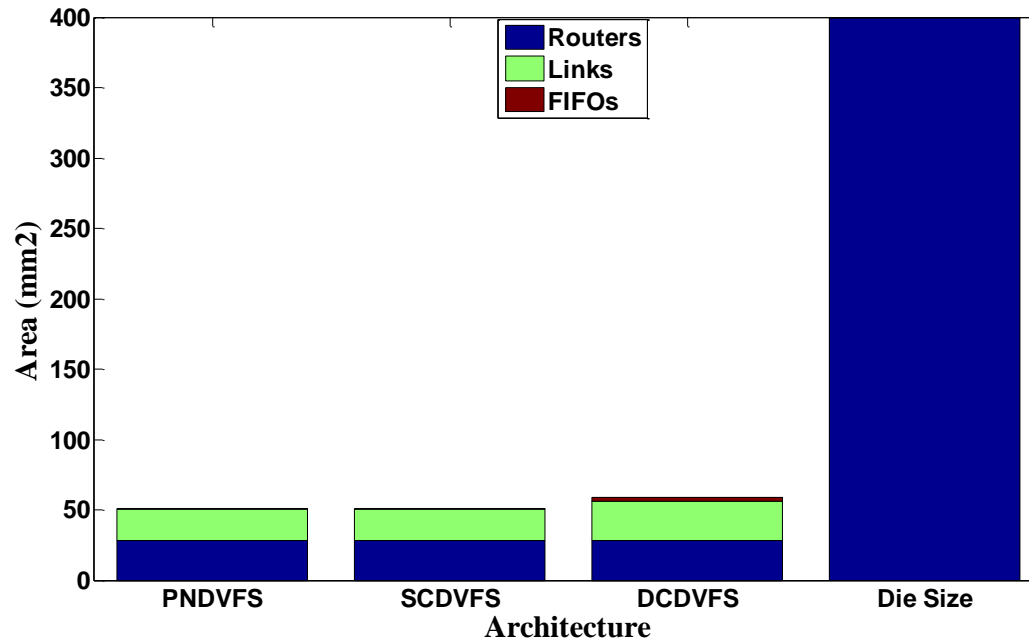
**Average Latency comparison of three DVFS architectures  
(Normalized with PNDVFS)**

<b>Traffic Trace</b>	<b>SCDVFS</b>	<b>DCDVFS</b>	<b>FIFO overhead</b>
<b>T1</b>	<b>1.09</b>	<b>1.49</b>	<b>24%</b>
<b>T2</b>	<b>1.04</b>	<b>1.68</b>	<b>26%</b>
<b>T3</b>	<b>1.09</b>	<b>1.17</b>	<b>10%</b>
<b>T4</b>	<b>0.80</b>	<b>1.45</b>	<b>19%</b>
<b>T5</b>	<b>1.03</b>	<b>1.40</b>	<b>18%</b>
<b>T6</b>	<b>0.93</b>	<b>1.32</b>	<b>11%</b>

- ✓ Natural consequence of lowered switching frequency
- ✓ Predictably bounded latency increase because of the congestion avoidance
- ✓ Significant FIFO latency overhead



# Evaluation (5)



## Area comparison of the DVFS architectures

- ✓ DCDVFS needs more area for the reconfigurable links.
- ✓ The increase is reasonable considering the whole die area
- ✓ Tradeoff of silicon area to gain power efficiency ( power budget > transistor and wiring limitation)

# Conclusion

- Run-time reconfiguration leads to better power efficiency
- For fast-growing massively parallel on-chip platform, run-time clusterization for applying adaptive power-management schemes is particularly useful to reduce the synchronization overhead
- System-level exploration is necessary before time-consuming low-level implementation
- Future study focuses on:
  - ✓ Further design choice exploration, for instance timing analysis of each configuration step
  - ✓ Circuit-level modeling of essential structures (reconfigurable link structure, pseudosynchronous clocking, etc..)