Control and Power Management in Presence of Workload Variations

Radu Marculescu

To cite this version:

Outline

- Part I: Multi-Domain Processors Design Overview (2:00-2:45PM)
  - Multi-domain server, cell phone, and media processors
  - Power management techniques
- Part II: Router Design and Synchronization Issues (2:45-3:30PM)
  - Asynchronous router design
  - Quality of Service and virtual channels in QNoC
- Part III: Control and Power Management in Presence of Workload Variations (4:00-4:45PM)
  - VFI partitioning and voltage assignment
  - Workload modeling and dynamic control of multi-VFI designs
- Part IV: DVFS in Presence of Process Variations (4:45-5:30PM)
  - Impact of process variations on DVFS controller performance
  - Technology-driven limits on DVFS controllability

ISCA-2010 Tutorial #2

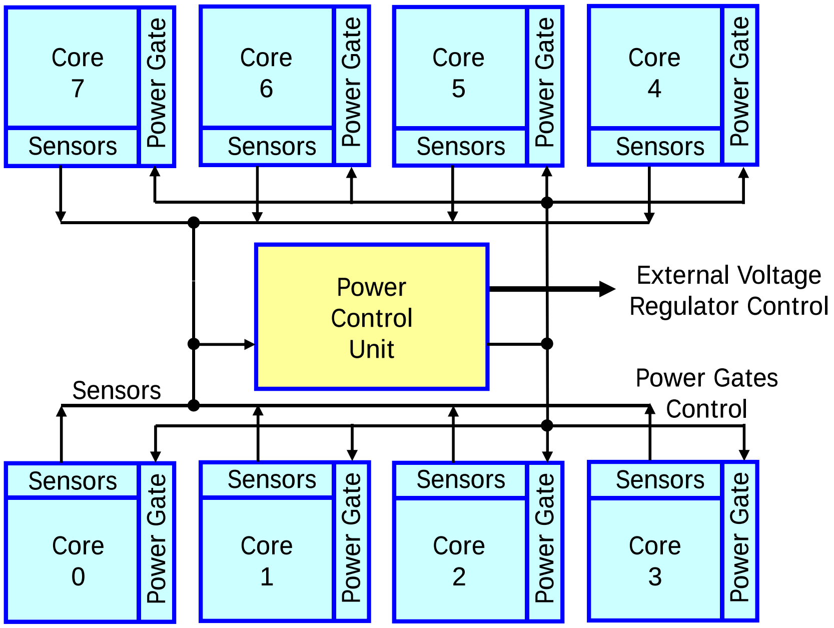
Control and Power Management in Presence of Workload Variations

Radu Marculescu

Carnegie Mellon University

radum@cmu.edu

Power Management Unit



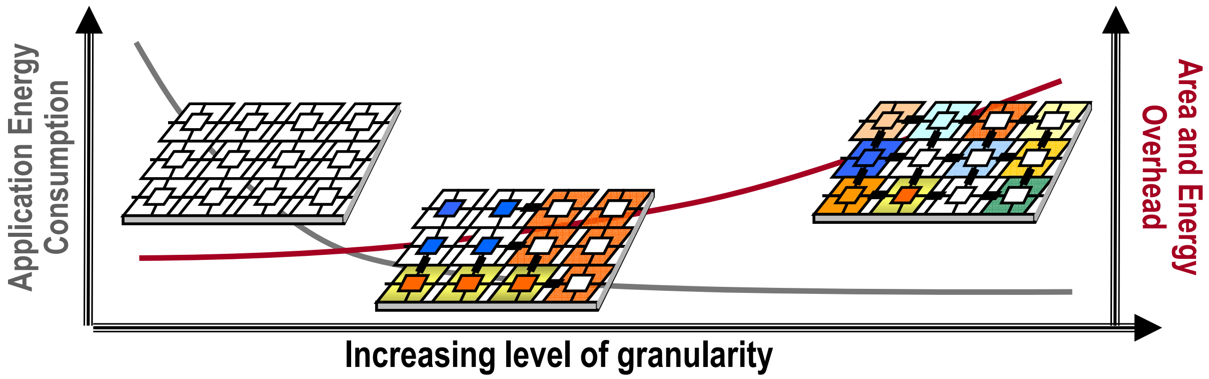
[Rusu, A-SSCC 2009]

Outline

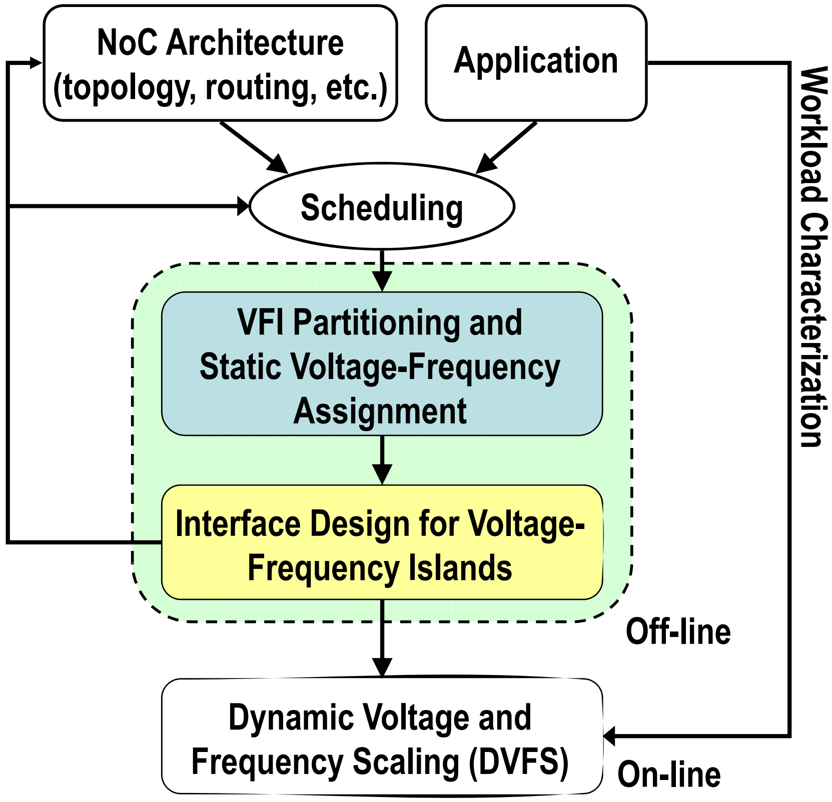
- VFI partitioning
  - Multi-VFI NoC designs
  - Partitioning and voltage assignment
  - Examples
- On-line control
  - State-based model construction
  - Feedback control architecture
  - Stability issues
- Summary

SoC VFI Partitioning

- What is the most efficient partitioning and what are the corresponding voltage/frequency assignments?



Design Methodology for Multi-VFI NoCs



NoC Architecture (topology, routing, etc.)

Application

Scheduling

VFI Partitioning and Static Voltage-Frequency Assignment

Interface Design for Voltage-Frequency Islands

Off-line

Dynamic Voltage and Frequency Scaling (DVFS)

On-line

Workload Characterization

VFI Partitioning Problem

- **Given**
  - **NoC architecture and a schedule for the driver application**
  - **Maximum number of allowed VFIs and physical constraints**
- **Find**
  - **VFI partitioning (i.e., optimum number of VFIs, *n* ≤ *N*)**
  - **Assignment of the supply and threshold voltages to each island**
- **Such that** the *total energy consumption* is minimized

Number of VFIs

$$E_{Total} = \underbrace{E_{App}}_{\text{Application (useful) energy consumption (comp+comm)}} + \sum_{i=1}^{n} \underbrace{E_{VFI}(i)}_{\text{Overhead of } i^{th} \text{ VFI}}$$

$$E_{VFI} = E_{ClkGen} + E_{Vconv} + E_{MixClkFifo}$$

Voltage/Frequency Assignment Problem

- **Given a *VFI partitioning***
- **Find supply (V_i) and threshold (V_{ti}) voltage assignments**
- **Such that application energy consumption is minimized**

$$\min E_{App} = \underbrace{\sum_{\forall i \in T} E_i(V_i, V_{ti})}_{\text{Energy consumed when the task is executed at } (V_i, V_{ti})} + \underbrace{\sum_{\forall i \in T} \sum_{\forall i \in T} vol(i,j) E_{bit}(i,j)}_{\text{Communication energy}}$$

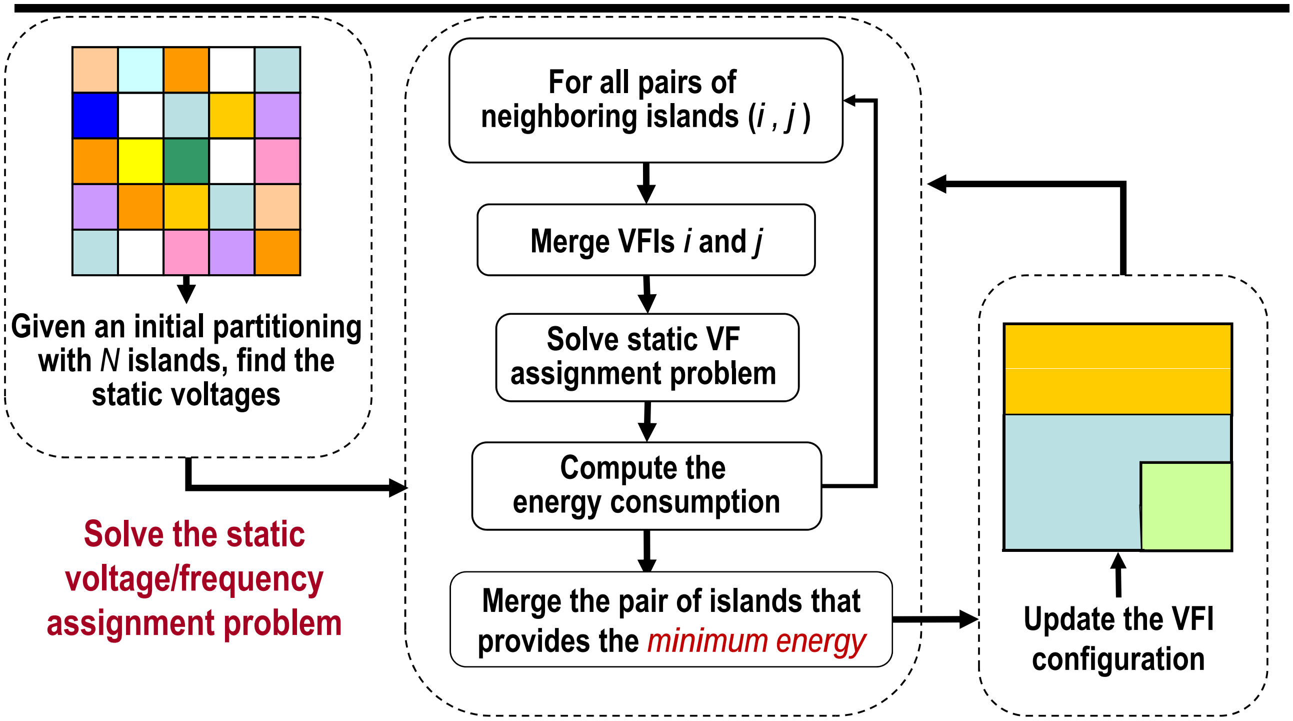
$$E_i(V_i, V_{ti}) = R_i C_i V_i^2 + T_i k_i V_i e^{\left(-\frac{V_t}{S_t}\right)}$$

- Subject to the following deadline constraints per task *t*:

$$\underbrace{\frac{x_t}{f_t}}_{\text{Execution time}} + \underbrace{t^t_{Comm}}_{\text{Communication delay}} \le deadline_t - start\_time_t$$

$$t_{comm}(src, dst) = \sum_{i \in P} \frac{\mu_s}{f_i} + t_{fifo} \left\lceil \frac{vol(src, dst)}{W} \right\rceil$$

VFI Partitioning and Voltage Assignment Algorithm



Given an initial partitioning with *N* islands, find the static voltages

Solve the static voltage/frequency assignment problem

For all pairs of neighboring islands (*i* , *j*)

Merge VFIs *i* and *j*

Solve static VF assignment problem

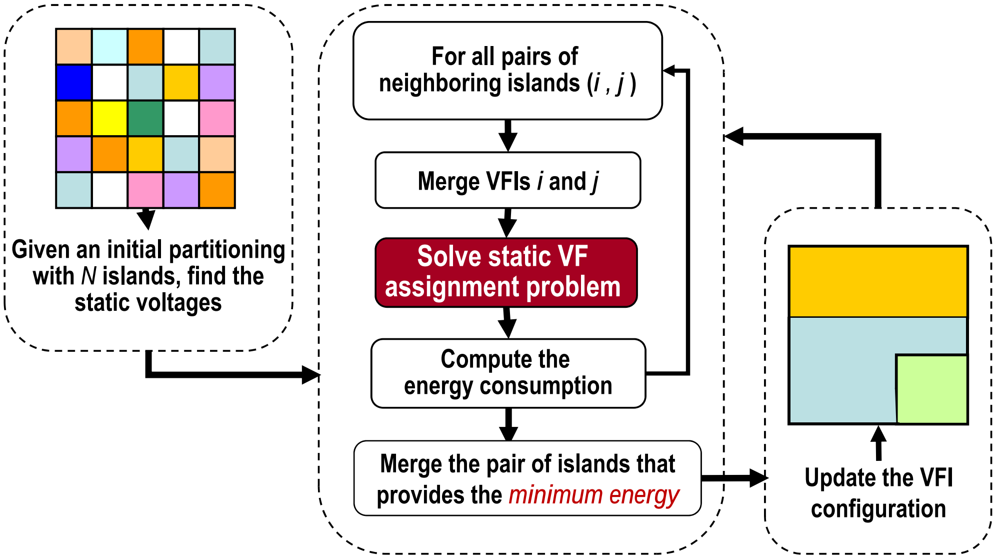
Compute the energy consumption

Merge the pair of islands that provides the *minimum energy*

Update the VFI configuration

This can be also implemented as a branch & bound algorithm.
We can obtain *exact* results for small examples.

Voltage Assignment Algorithm



Given an initial partitioning with *N* islands, find the static voltages

For all pairs of neighboring islands (*i* , *j*)

Merge VFIs *i* and *j*

Solve static VF assignment problem

Compute the energy consumption

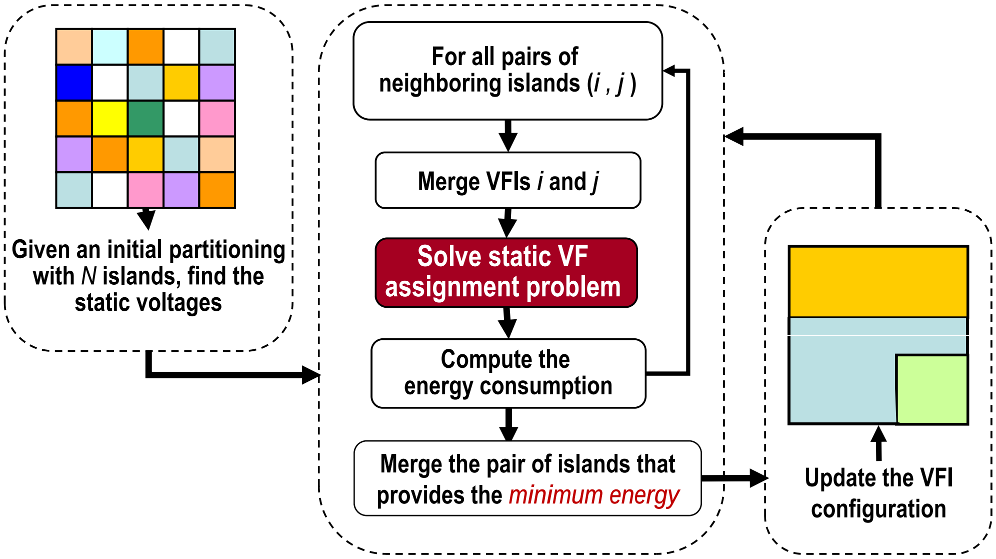
Merge the pair of islands that provides the *minimum energy*

Update the VFI configuration

$$min\ E_{App} = \sum_{\forall i \in T} E_i(V_i, V_{ti}) + \sum_{\forall i \in T} \sum_{\forall i \in T} vol(i,j) E_{bit}(i,j)$$

subject to $$\frac{x_t}{f_t} + t_{Comm}^{t} \le deadline_t - start\_time_t$$

Voltage Assignment Algorithm



Given an initial partitioning with *N* islands, find the static voltages

For all pairs of neighboring islands (*i* , *j*)

Merge VFIs *i* and *j*

Solve static VF assignment problem

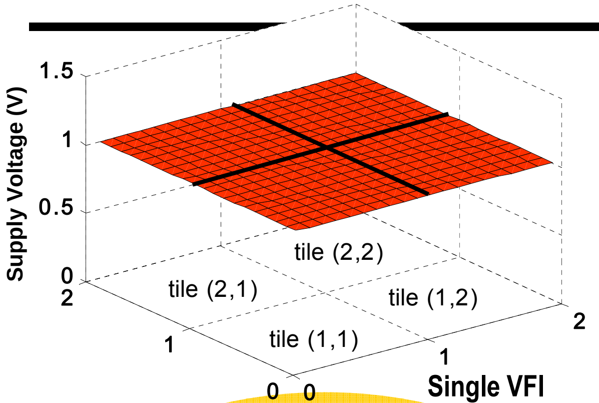
Compute the energy consumption

Merge the pair of islands that provides the *minimum energy*

Update the VFI configuration

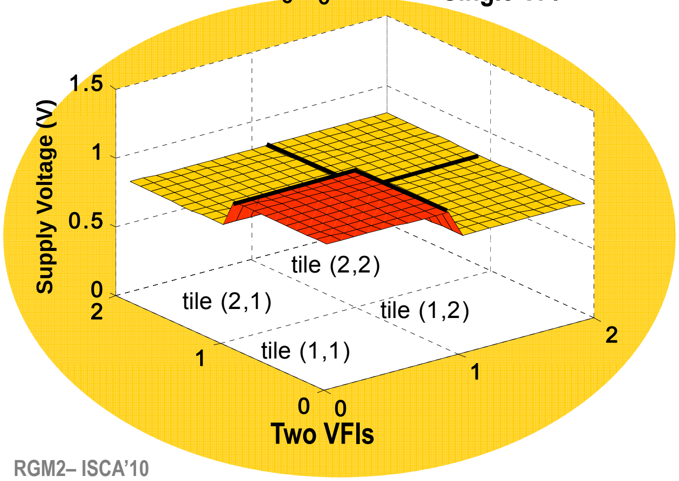
- *Constrained nonlinear optimization* or *nonlinear programming*
  - Finds a constrained minimum of a scalar function of several variables
  - Use Matlab nonlinear solver (*fmincon*)

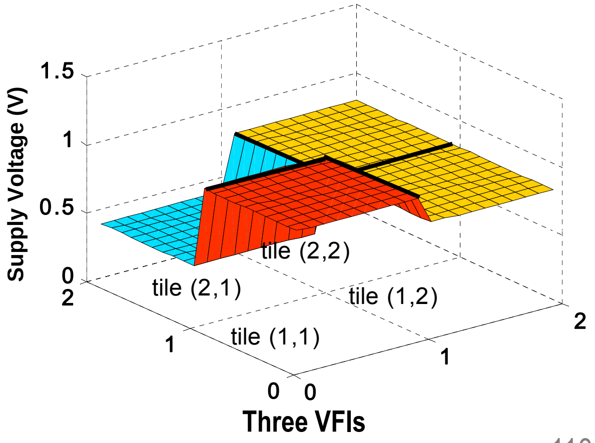
Why Does VFI Partitioning Matters?



- Small benchmark scheduled on a 2×2 network using EDF
- BPM 70 nm used for the technology parameters
- Energy consumption
  - 1-VFI: 10.5mJ
  - 2-VFI: 7.5mJ
  - 3-VFI: 7.6mJ

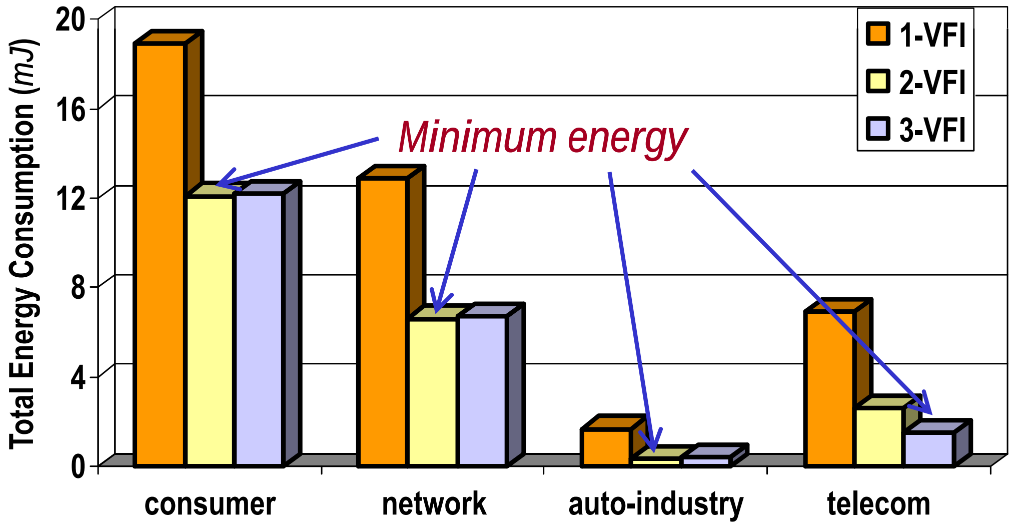
29%





Experiments with Realistic Benchmarks

- **Several E3S benchmarks** *(consumer, network, auto-industry, telecom)*
- **Applications scheduled to NoCs ranging from 3×3 to 5×5**



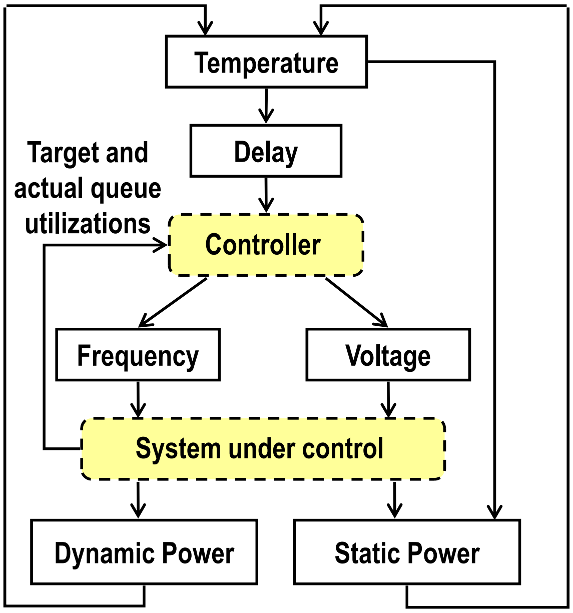
Improvement over 1-VFI solution	consumer	network	auto-industry	telecom
	36%	49%	80%	78%

Outline

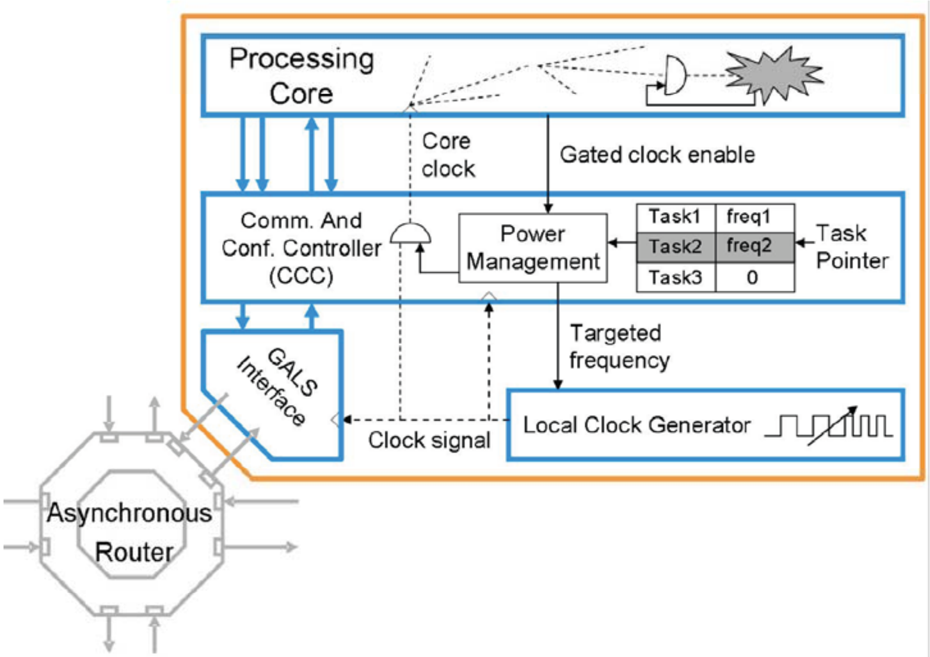
- VFI partitioning
  - Multi-VFI NoC designs
  - Partitioning and voltage assignment
  - Examples
- On-line control
  - State-based model construction
  - Feedback control architecture
  - Stability issues
- Summary

Why On-line Control?

- Cannot rely on nominal values because they vary
  - Sources of concern are workload, process, voltage, temperature variations
  - Cope with the parameter variations which cannot be predicted or accurately modeled at design time
- Heuristic techniques and manual tuning won't work!



Distributed Power Management in Magali



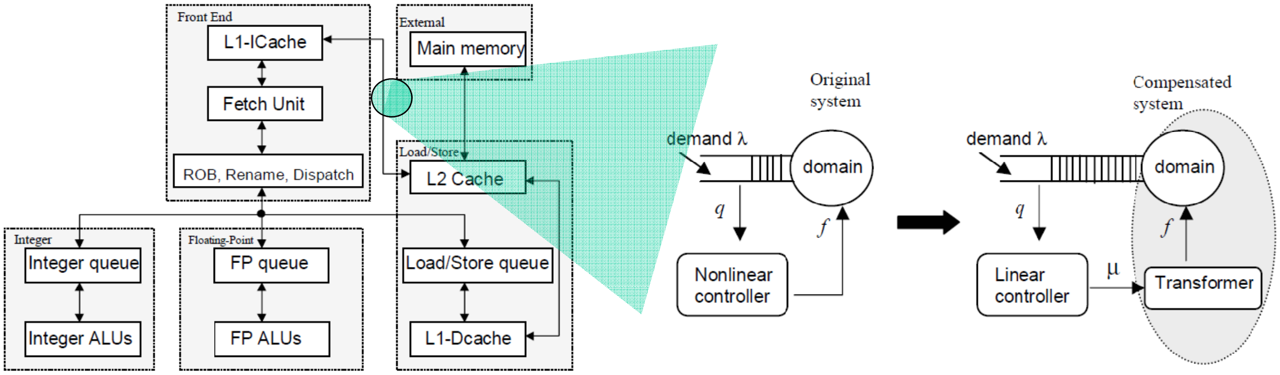
A 477mW NoC-based digital baseband for MIMO 4G SDR chip organized around a 15-router asynchronous NoC that connects 22 processing units.

[Clermidy et al, ISSCC'10]

Local Control in Multi Clock Domain Processors

The clock domain partitions in an MCD processor

Interface model between the domains


Front End
L1-ICache
Fetch Unit
ROB, Rename, Dispatch
External
Main memory
Load/Store
L2 Cache
Integer
Integer queue
Integer ALUs
Floating-Point
FP queue
FP ALUs
Load/Store queue
L1-Dcache

Original system
demand λ
domain
q
f
Nonlinear controller

Compensated system
demand λ
domain
q
f
Linear controller
μ
Transformer

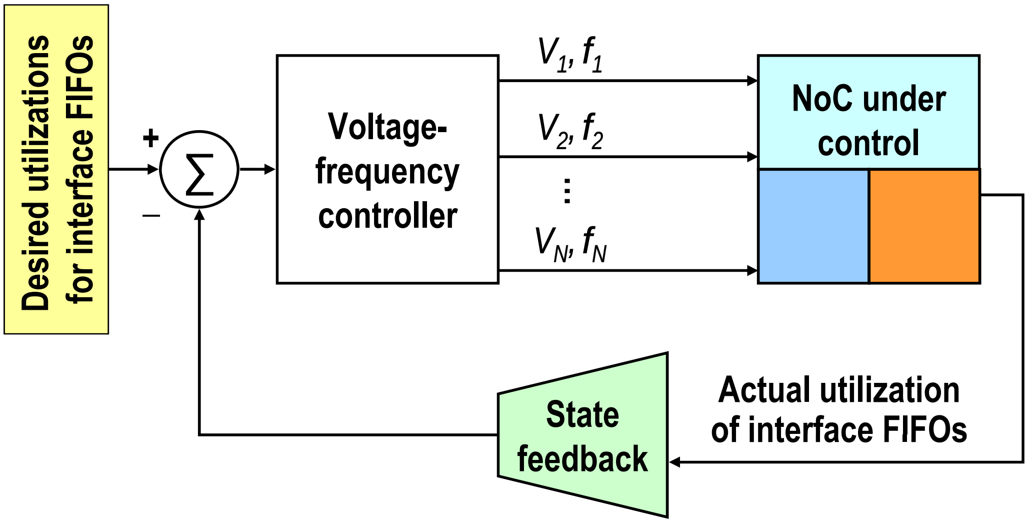
[Semeraro, et al, HPCA'02]

[Wu, et al, ASPLOS'04]

- **PID controller for voltage/frequency control proposed previously using only local queue information**
  - **Ignores interactions among multiple queues**
  - **Works fine if frequency change in one clock domain has negligible impact on other domains**
- **For an MCD processor with arbitrary partitions and strong interactions among multiple queues, a centralized online DVFS scheme may be needed**

Design Methodology for Multi-VFI NoCs

- **Traditionally, PID controllers are used due to simplicity. However, state-space modeling brings new opportunities**
  - Precise controllability and stability analysis
  - Pole placement, linear quadratic regulator, robust controller


Desired utilizations for interface FIFOs

+

–

$\sum$

Voltage-frequency controller

V_1, f_1

V_2, f_2

⋮

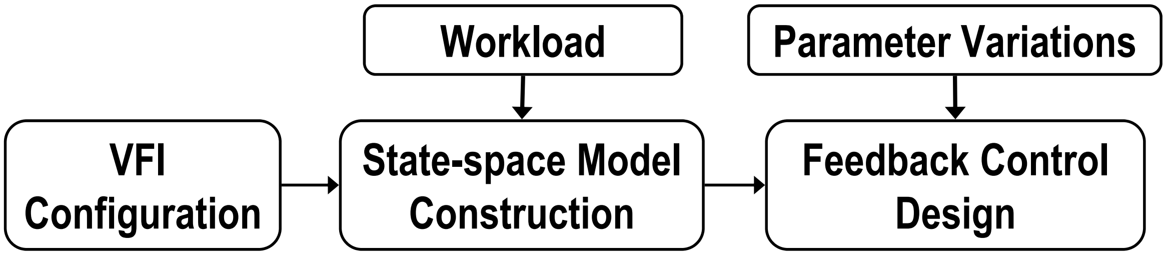
V_N, f_N

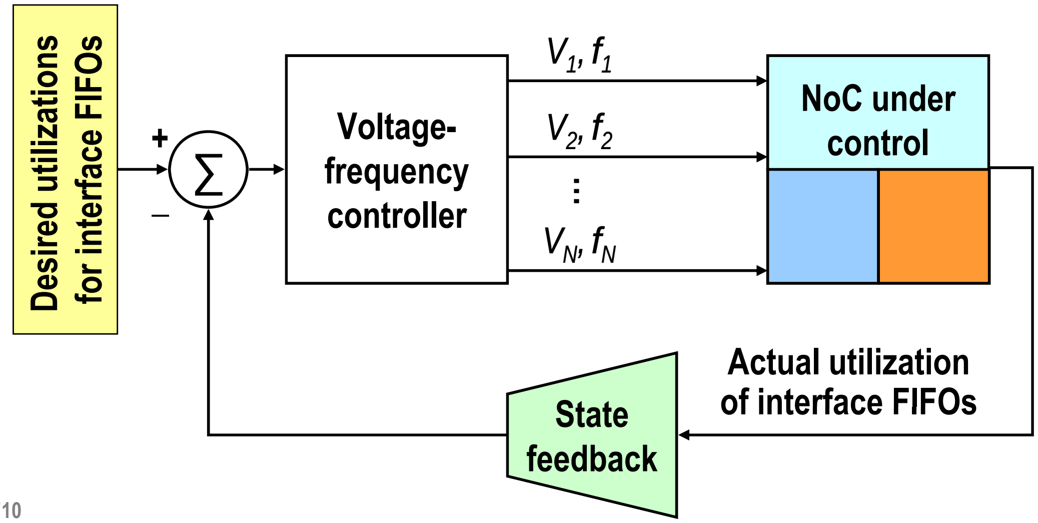
NoC under control

Actual utilization of interface FIFOs

State feedback

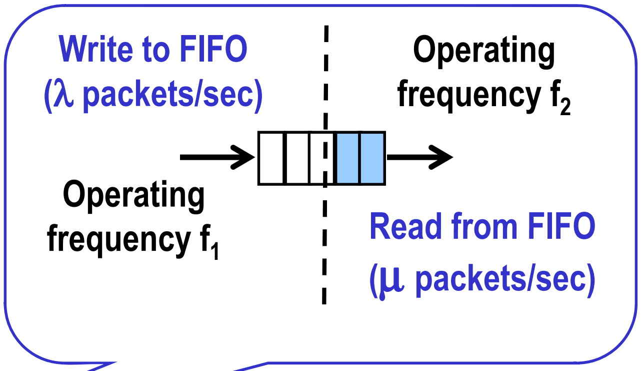
Design Methodology for Multi-VFI NoCs

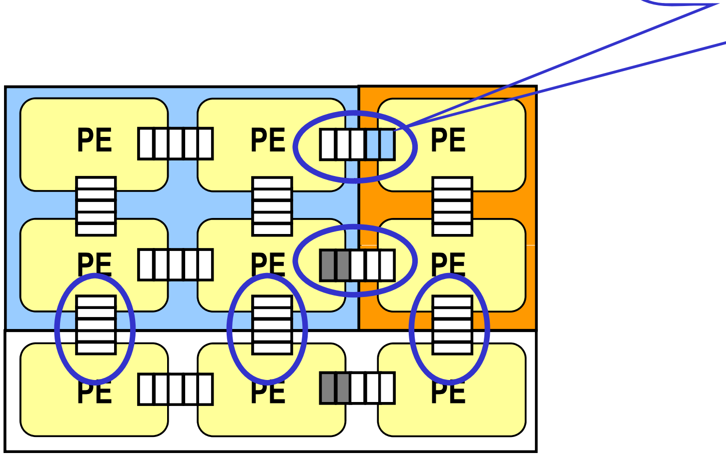



Desired utilizations for interface FIFOs
+
–
∑
Voltage-frequency controller
V_1, f_1
V_2, f_2
⋮
V_N, f_N
NoC under control
Actual utilization of interface FIFOs
State feedback

Formal Feedback Control

- **Multi-VFI Network-on-Chip**
  - Interface queue utilizations are the *states* of the system
  - State feedback for voltage-frequency control
  - Control interval is *T* µsec





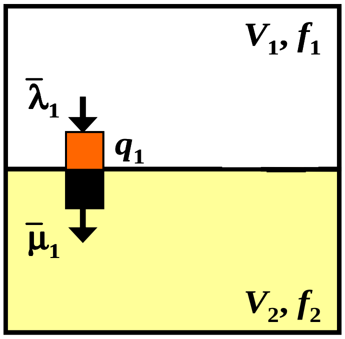
State (queues utilization)

$$Q = [q_1, q_2, \ldots, q_N]$$

Input (clock speeds)

$$F = [f_1, f_2, \ldots, f_M]$$

Step-by-step Model Construction *(one queue)*



$$\overbrace{q(k)}^{\text{Average utilization in the } k^{th} \text{ control interval}} = q(k-1) + \underbrace{T\lambda_1(k-1)}_{\text{Amount of data (packets) written to the queue}} - \overbrace{T\mu_1(k-1)}^{\text{Amount of data (packets) read from the queue}}$$

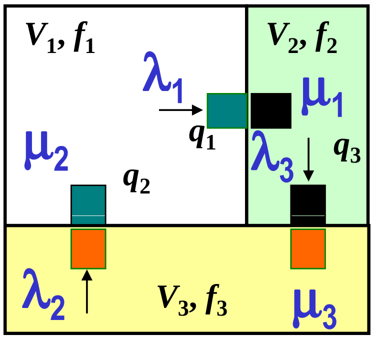
- **If data read/write rates are proportional to the frequency of the VFI**

$$\lambda_1(k-1) = \bar{\lambda}_1 f_1(k-1), \;\; \mu_1(k-1) = \bar{\mu}_1 f_2(k-1)$$

- **The state-space equation can be written as**

$$q(k) = q(k-1) + T\underbrace{\begin{bmatrix}\bar{\lambda}_1 & -\bar{\mu}_1\end{bmatrix}}_{B}\begin{bmatrix} f_1(k-1) \\ f_2(k-1)\end{bmatrix}$$

Step-by-step Model Construction *(three queues)*



$$B = \begin{array}{l} \left[\begin{array}{ccc} \quad & \quad & \quad \\ & & \\ & & \end{array}\right] \\ \underbrace{\quad}_{f_1} \; \underbrace{\quad}_{f_2} \; \underbrace{\quad}_{f_3} \end{array}$$

First row → q_1

Second row → q_2

Third row → q_3

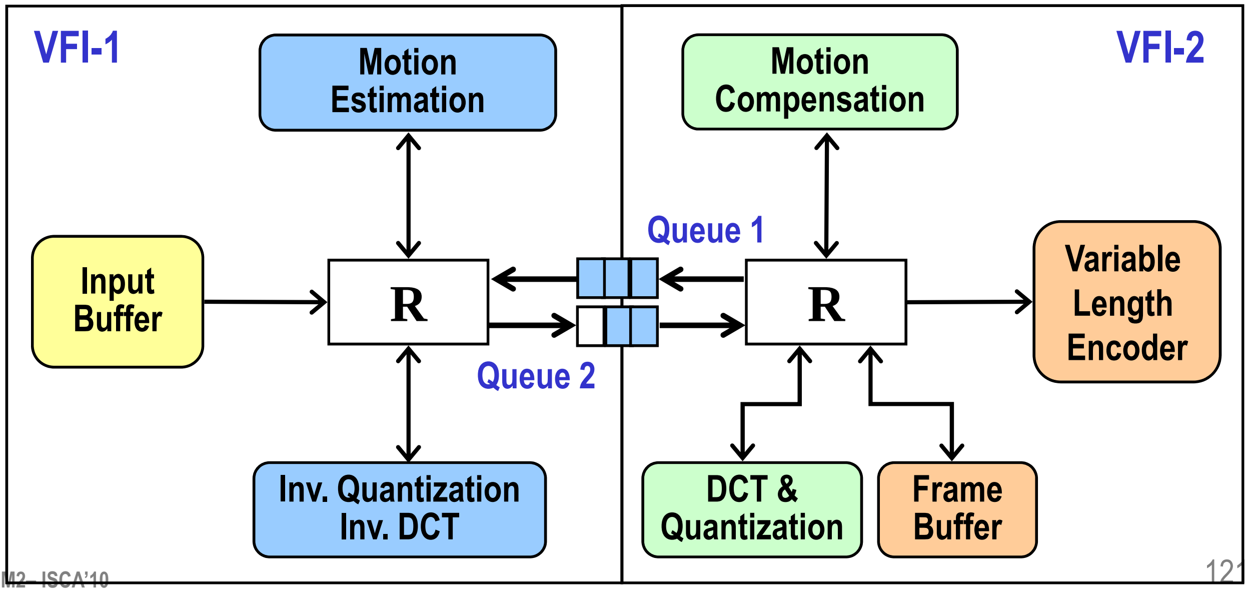
- **The topology of the VFIs determines the matrix** *B*
- **An algorithm automatically constructs** *B*
- **The structure of the model is the same regardless of** *B*

$$\boxed{Q(k)_{N\times 1} = Q(k\text{-}1)_{N\times 1} + TB_{N\times M}F(k\text{-}1)_{M\times 1}}$$

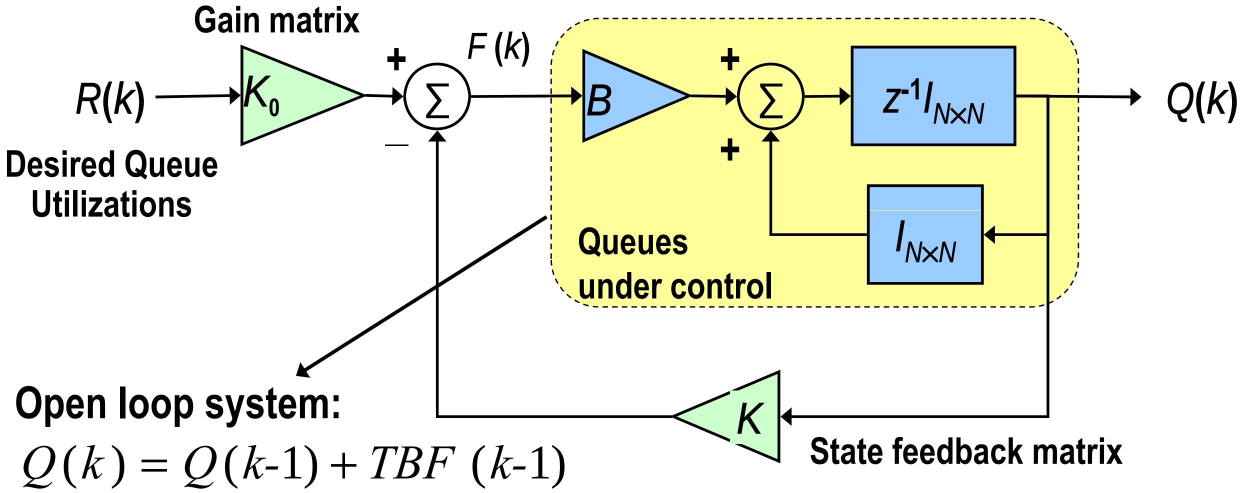
System Controllability

In the multiple voltage-frequency island system with *M* islands, utilization of at most *M* queues can be controlled.

The system is controllable *iff rank*(*B*) = *N* (i.e., number of controlled queues)



Feedback Control Architecture



Open loop system:

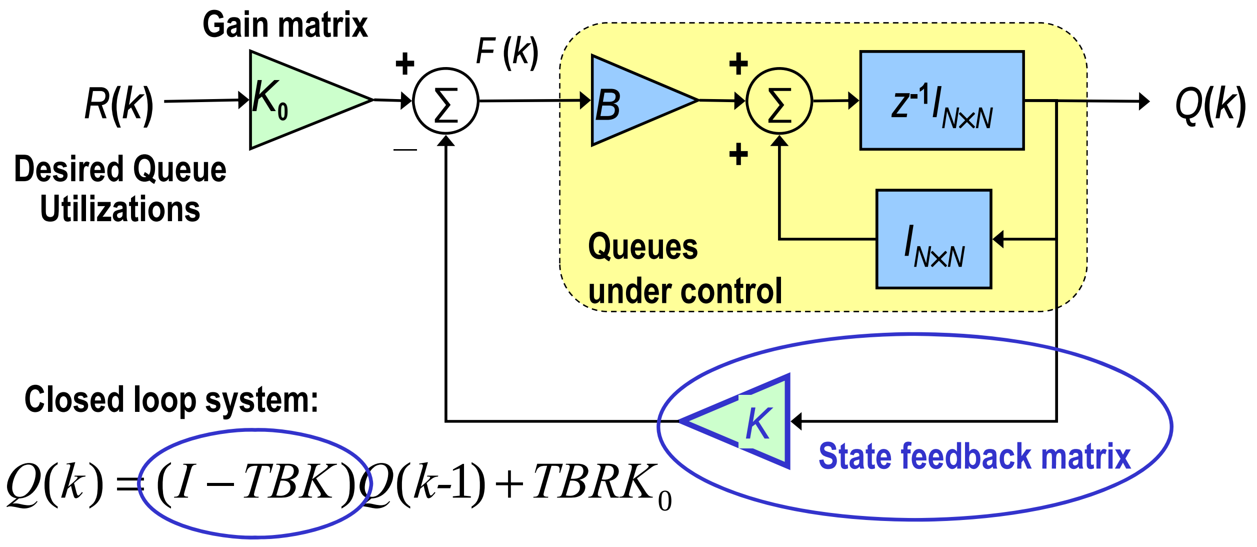
$$Q(k) = Q(k\text{-}1) + TBF\ (k\text{-}1)$$

Closed loop system:

$$Q(k) = (I - TBK)Q(k\text{-}1) + TBRK_0$$

$$B = \begin{bmatrix} \bar{\lambda}_1 & -\bar{\mu}_1 & 0 \\ -\bar{\mu}_2 & 0 & \bar{\lambda}_2 \\ 0 & \bar{\lambda}_3 & -\bar{\mu}_3 \end{bmatrix}$$

Feedback Control Architecture

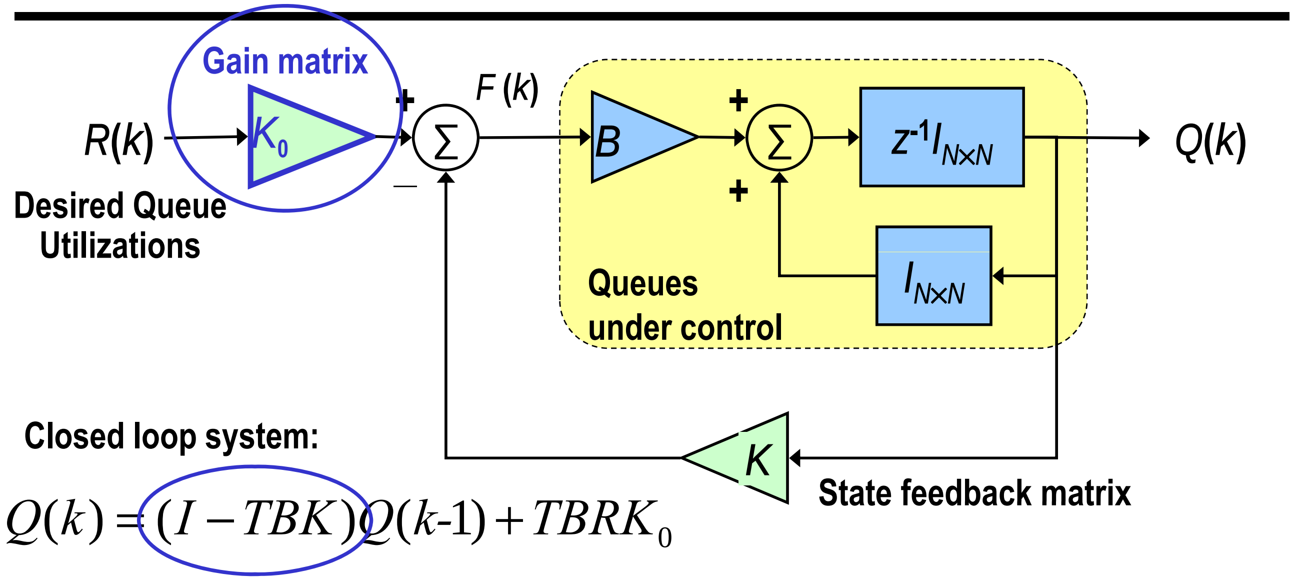


Closed loop system:

$$Q(k) = (I - TBK)Q(k\text{-}1) + TBRK_0$$

- **Design of the state feedback matrix K**
  - Find K such that the eigenvalues of the closed loop system are inside the unit circle despite the workload variations
  - Eigenvalue placement, linear quadratic regulator (LQR) design

Feedback Control Architecture



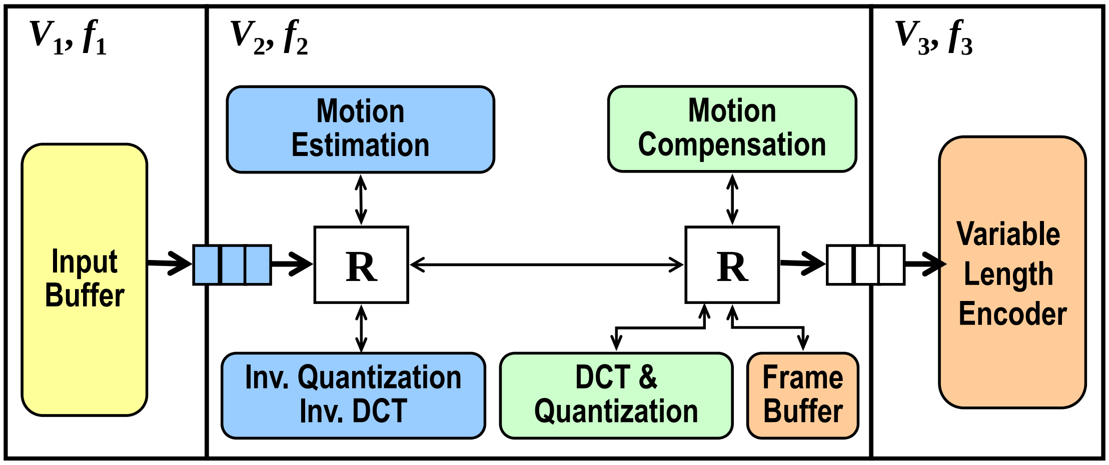
Closed loop system:

$$Q(k) = (I - TBK)Q(k\text{-}1) + TBRK_0$$

- **By finite value theorem, gain matrix K0 = K**
- **Possible extensions**
  - Adaptive techniques, such as gain scheduling
  - Monitor the workload and compute *K* or use values computed off-line

Experiments with MPEG-2 Encoder

- The encoder is divided into three VFI islands and mixed clock FIFOs are used at the interfaces
- The frequency of Variable Length Encoder is set to achieve the desired encoding rate



V_1, f_1

V_2, f_2

V_3, f_3

Input Buffer

Motion Estimation

Motion Compensation

R

R

Inv. Quantization Inv. DCT

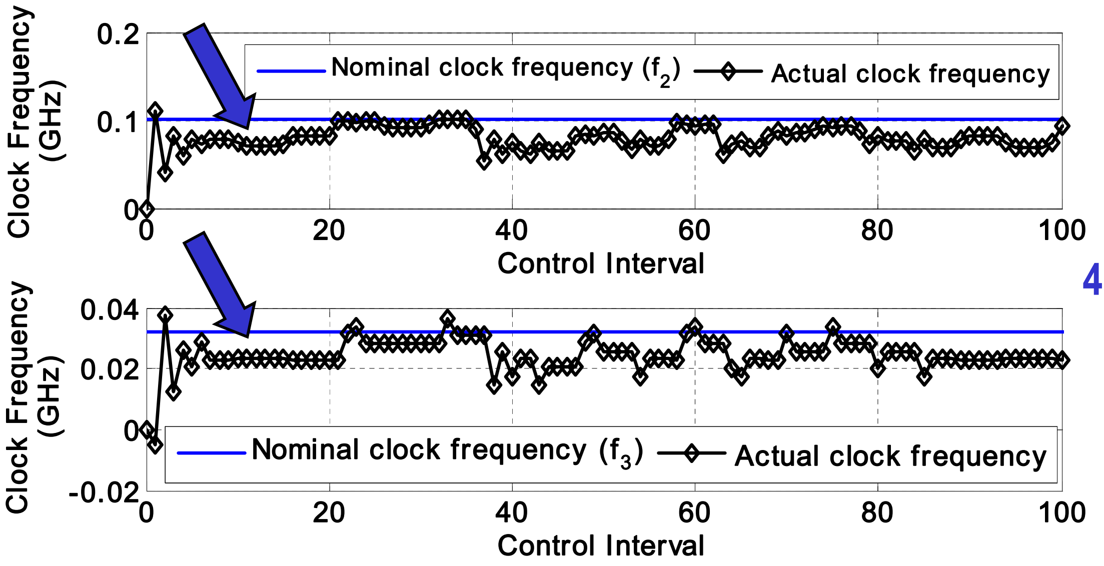
DCT & Quantization

Frame Buffer

Variable Length Encoder

Frequency Tracking Capabilities

- 50 Frames/sec for 352×288 CIF frames
- f_1 is set to meet the target, f_2 and f_3 follow f_1



46% power savings

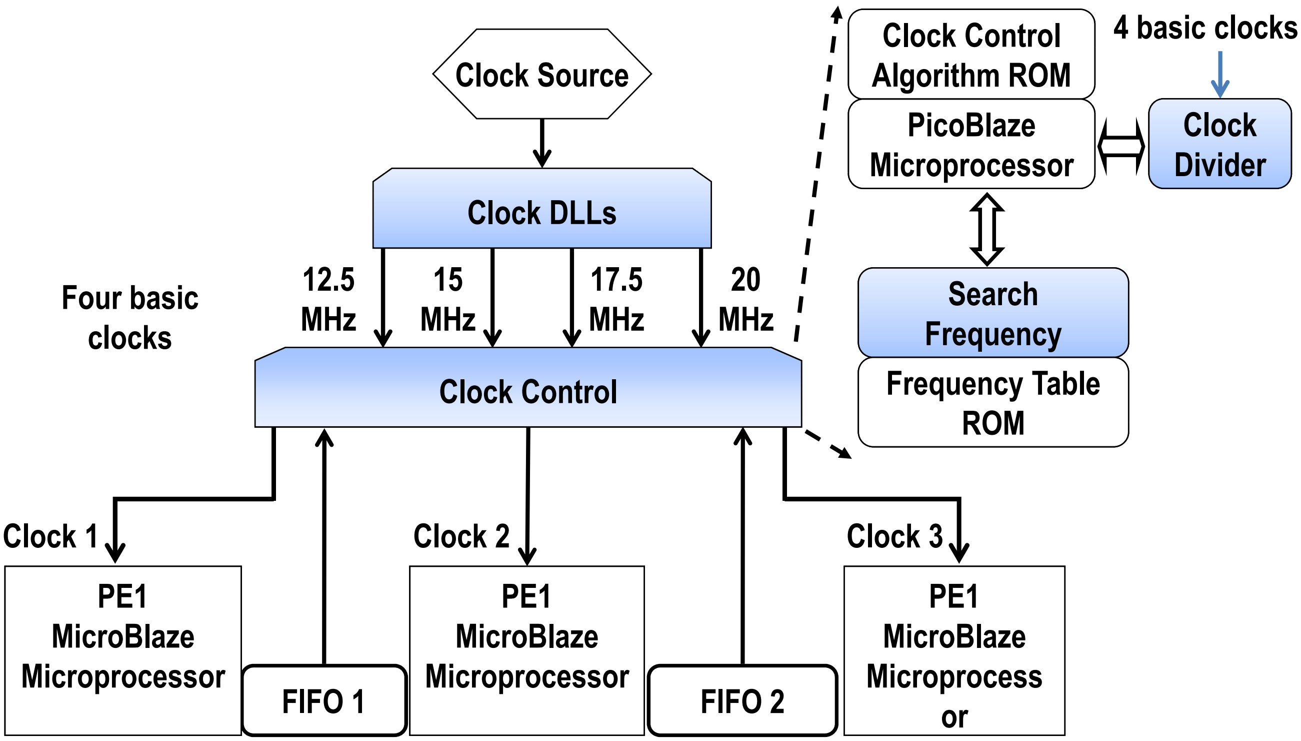
Results on FPGA prototyping

- Work on FPGA prototype using Virtex-II Pro FPGA from Xilinx
- Inter-domain communication
  - Delay Locked Loops (DLLs) used to generate individual clock signals
  - Block-RAM based mixed-clock FIFOs
  - Voltage conversion not supported yet by Xilinx boards

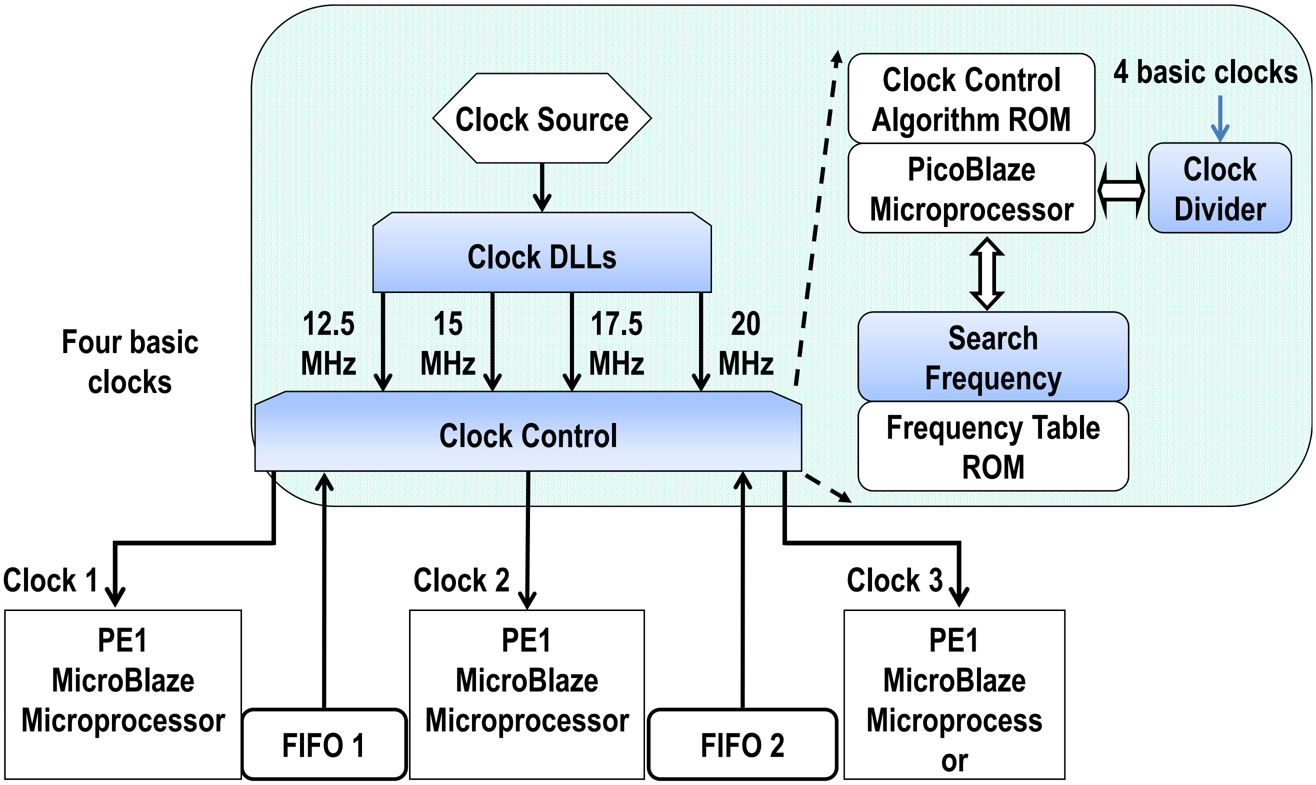
- MPEG-2 encoder design divided into three VFIs
  - Synchronous design utilizes 16966 LUTs
  - Design with three VFIs utilizes 19161 LUTs
  - Power consumption obtained using XPower
    - Without voltage scaling, power drops from 277W to 259W
    - Consistent with simulations

13% overhead

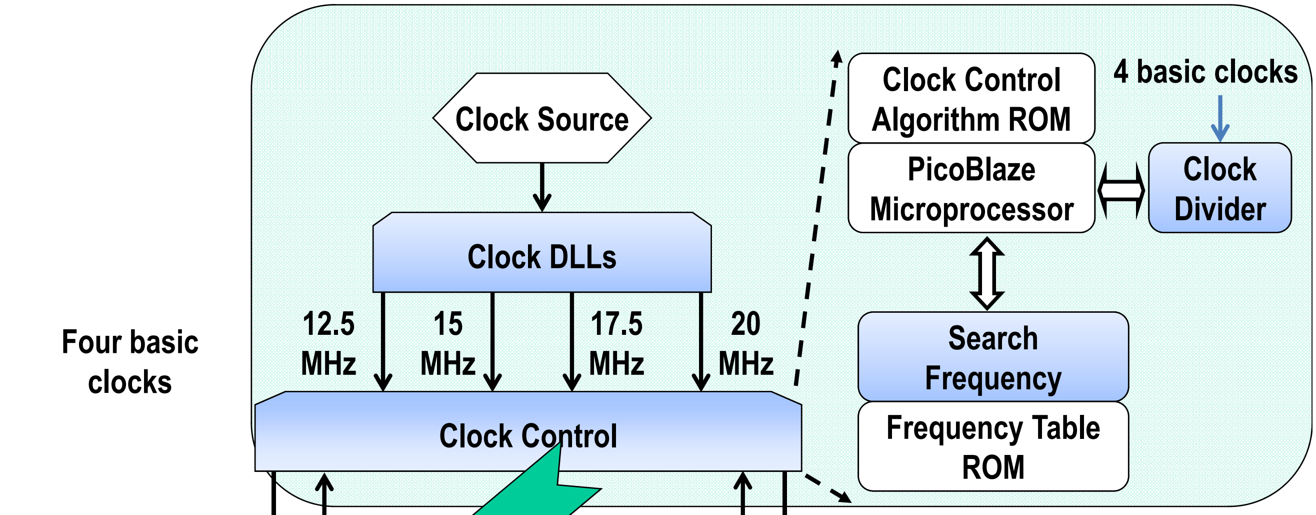
Clock Control Architecture

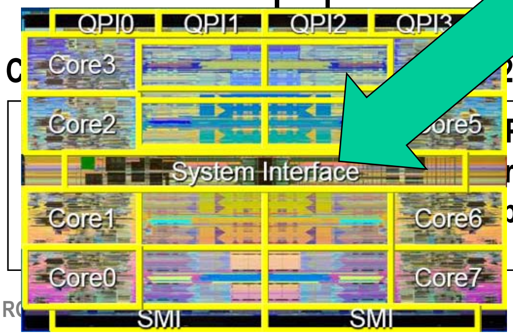


Clock Control Architecture



Clock Control Architecture





- The voltage/frequency control circuitry can be implemented in HW
- The control algorithm can be implemented in firmware / OS

Summary

- Energy issues in multi-VFI NoCs are crucial
  - VFI synthesis via partitioning and voltage allocation
  - Other formulations are possible
- Dynamic V/F control yields significant power savings over static approaches while being robust to workload variations
  - DVFS controller smoothes out variations in workload characteristics
  - Precise controllability and stability conditions can be defined
- More work needed to address
  - Adaptive techniques for VFI control
  - Run-time optimizations for multiple applications
  - Impact of dynamic traffic on overall DVFS-based power management

Outline

- Part I: Multi-Domain Processors Design Overview (2:00-2:45PM)
  - Multi-domain server, cell phone, and media processors
  - Power management techniques
- Part II: Router Design and Synchronization Issues (2:45-3:30PM)
  - Asynchronous router design
  - Quality of Service and virtual channels in QNoC
- Part III: Control and Power Management in Presence of Workload Variations (4:00-4:45PM)
  - VFI partitioning and voltage assignment
  - Workload modeling and dynamic control of multi-VFI designs
- **Part IV: DVFS in Presence of Process Variations (4:45-5:30PM)**
  - **Impact of process variations on DVFS controller performance**
  - **Technology-driven limits on DVFS controllability**

References (Part III)

- U. Y. Ogras, et al., ‘Design and Management of Voltage-Frequency Island Partitioned Networks-on-Chip,’ in IEEE Trans. VLSI, March 2009.
- P. Choudhary, D. Marculescu, ‘Power Management of Voltage/Frequency Island-Based Systems Using Hardware Based Methods,’ in IEEE Trans. on VLSI, March 2009
- T. Simunic, S. P. Boyd, P. Glynn, 'Managing power consumption in networks on chips,‘ in IEEE Trans. on VLSI, Jan. 2004.
- A. Alimonda, et al., ‘Feedback-Based Approach to DVFS in Data-Flow Applications,’ in IEEE Trans. on CAD of Integrated Circuits and Systems 28(11): 1691-1704 (2009)
- E. Beigne, et al., ‘Dynamic voltage and frequency scaling architecture for units integration within a GALS NoC,’ in Proc. Int. Symp. Netw. Chip, 2008, pp. 129–138.
- C.-L. Chou and R. Marculescu, ‘Energy- and performance-aware incremental mapping for networks on chip with multiple voltage levels,’ IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 27, no. 10, pp. 1866–1879, Oct. 2008
- Q. Wu, et al, ‘Formal Online Methods for Voltage/Frequency Control in Multiple Clock Domain Microprocessors’, in Proc. ASPLOS 2004
- G. Semeraro, et al, ‘Energy efficient processor design using multiple clock domains with dynamic voltage and frequency scaling,’ in Proc. HPCA 2002
- U. Y. Ogras, et. al, 'NoC Prototyping Using FPGAs: Challenges and Promising Results in NoC Prototyping Using FPGAs, ' in IEEE Micro, September/October 2007
- Clermidy et al, ‘A 477mW NoC-Based Digital Baseband for MIMO 4G SDR,’ IEEE ISSCC, February 2010
- P. Juang, et al. ‘Coordinated, distributed, formal energy management of chip multiprocessors,’ Proc. ISLPED 2005..

References (Part IV)

- Y. Abulafia and A. Kornfeld. Estimation of FMAX and ISB in microprocessors. IEEE Trans. on VLSI Systems, 13(10), Oct 2006.
- A. Bonnoit, S. Herbert, D. Marculescu and L. Pileggi. Integrating Dynamic Voltage/Frequency Scaling and Adaptive Body Biasing using Test-time Voltage Selection. In Proc. of IEEE/ACM ISLPED, Aug. 2009.
- K. Bowman, S. Duvall, and J. Meindl. Impact of die-to die and within-die parameter fluctuations on the maximum clock frequency distribution for gigascale integration. IEEE Journal of Solid-State Circuits, 37(2), Feb 2002.
- S. Garg, D. Marculescu. System-Level Mitigation of WID Leakage Variations using Body-Bias Islands. In Proc. ACM/IEEE CODES+ISSS, Atlanta, GA, October 2008.
- S. Garg, D. Marculescu, R. Marculescu and U. Ogras. Technology-driven Limits on DVFS Controllability of Multiple Voltage-Frequency Island Designs. In Proc. of IEEE/ACM Design Automation Conference (DAC), Jul. 2009.
- S. Herbert and D. Marculescu. Analysis of dynamic voltage/frequency scaling in chip-multiprocessors. In ISLPED ’07: Proc. of the 2007 ISLPED, 2007.
- S. Herbert and D. Marculescu. Variation-Aware Dynamic Voltage/Frequency Scaling. In Proc. of the 15th HPCA, Feb. 2009.
- C. Isci, A. Buyuktosunoglu, C.-Y. Cher, P. Bose, and M. Martonosi. An analysis of efficient multi-core global power management policies: Maximizing performance for a given power budget. In MICRO ’06, 2006.
- S.R. Sarangi, B. Greskamp, R. Teodorescu, J. Nakano, A. Tiwari and J. Torrellas. VARIUS: A Model of Process Variation and Resulting Timing Errors for Microarchitects. IEEE Transactions on Semiconductor Manufacturing (IEEE TSM), February 2008.
- R. Teodorescu and J. Torrellas. Variation-aware application scheduling and power management for chip multiprocessors. In ISCA’08: Proc. of the 35th ISCA, 2008.
- J.Tschanz, J.T. Cao, S.G. Narendra, R. Nair, D.A. Antoniadis, A.P. Chandrakasan, V. De. Adaptive Body Bias for Reducing Impacts of Die-to-Die and Within-Die Parameter Variations on Microprocessor Frequency and Leakage. IEEE Journal of Solid-State Circuits, Vol. 37, No. 11, Nov. 2002.
- W. Zhao and Y. Cao. New generation of predictive technology model for sub-45nm early design exploration. IEEE Trans. Electron Devices, vol. 53, no. 11, pp. 2816--2823, Nov. 2006.

- **This list of references is NOT exhaustive. There are many good contributions not mentioned here due to involuntary omissions or space limitations.**