SysScale: Exploiting Multi-domain Dynamic Voltage and Frequency Scaling for Energy Efficient Mobile Processors

Jawad Haj-Yahya¹

Mohammed Alser¹ Jeremie Kim¹ A. Giray Yaglıkçı¹ Nandita Vijaykumar^{1,2,3} Efraim Rotem² Onur Mutlu¹



Executive Summary

Problem: A modern thermally-constrained mobile SoC has three domains: compute, IO, and memory. The **SoC allocates a fixed power budget to these domains unfairly** based on their worst-case performance demands even if they are underutilized.

<u>Goal</u>: Increase the energy efficiency and performance of mobile SoCs by dynamically orchestrating the distribution of the SoC power budget across the *three* domains based on their actual performance demands.

Mechanism: SysScale, a new multi-domain power management technique that introduces

- A new DVFS (dynamic voltage and frequency scaling) mechanism to distribute the SoC power to each domain based on its predicted performance demand
- An accurate algorithm to predict each domain's performance demand
- Domain-specialized techniques to optimize the energy efficiency of each domain at different operating points

Evaluation: We implement SysScale on the Intel Skylake SoC for mobile devices. **SysScale:**

- Improves the performance of SPEC CPU2006 and 3DMark workloads by up to 16% and 8.9% (9.2% and 7.9% on average), providing larger benefits at lower power budgets
- Reduces the average power consumption of battery life workloads by up to 10.7% (8.5% on average)

Presentation Outline

- 1. Overview of a Modern SoC
- 2. Motivation and Goal
- 3. SysScale
 - I. Power Management Flow
 - II. Demand Prediction Mechanism
 - III. Holistic Power Management Algorithm
- 4. Evaluation

5. Conclusion

Presentation Outline

1. Overview of a Modern SoC

2. Motivation and Goal

3. SysScale

- I. Power Management Flow
- II. Demand Prediction Mechanism
- III. Holistic Power Management Algorithm

4. Evaluation

5. Conclusion

Overview of a Modern SoC Architecture

- 3 domains in modern thermally-constrained mobile SoC: Compute, Memory, IO
- Several voltage sources exist, and some of them are shared between domains
- IO controllers and engines, IO interconnect, memory controller, and DDRIO typically each has an independent clock



Overview of a Modern SoC Architecture



Presentation Outline

1. Overview of a Modern SoC

2. Motivation and Goal

3. SysScale

- I. Power Management Flow
- II. Demand Prediction Mechanism
- III. Holistic Power Management Algorithm

4. Evaluation

5. Conclusion

Motivational Experiment

- We evaluate the potential benefits of employing DVFS across the three SoC domains
- We carry out an experiment on the Intel Broadwell processor under two setups
 - Baseline
 - Multi-Domain DVFS (MD-DVFS)
- We use multiple workloads from SPEC CPU2006 and 3DMark, and a workload that exercises the peak memory bandwidth of DRAM

Component	Baseline	MD-DVFS
DRAM frequency	1.6GHz	1.06GHz
IO Interconnect	0.8GHz	0.4GHz
Shared Voltage	V_SA	0.8·V_SA
DDRIO Digital	V_IO	0.85·V_IO
2 Cores (4 threads)	1.2GHz	1.2GHz

- The SoC power budget management algorithm (PBM) assigns a *fixed* power budget to the IO and memory domains
- The power budget corresponds to the *worst-case* performance demands (bandwidth/latency)
- Running three SPEC CPU2006 workloads using baseline and MD-DVFS



- The SoC power budget management algorithm (PBM) assigns a *fixed* power budget to the IO and memory domains
- The power budget corresponds to the *worst-case* performance demands (bandwidth/latency)
- Running three SPEC CPU2006 workloads using baseline and MD-DVFS



- The SoC power budget management algorithm (PBM) assigns a *fixed* power budget to the IO and memory domains
- The power budget corresponds to the worst-case performance demands (bandwidth/latency)
- Running three SPEC CPU2006 workloads using baseline and MD-DVFS



- The SoC power budget management algorithm (PBM) assigns a *fixed* power budget to the IO and memory domains
- The power budget corresponds to the worst-case performance demands (bandwidth/latency)
- Running three SPEC CPU2006 workloads using baseline and MD-DVFS



- The SoC power budget management algorithm (PBM) assigns a *fixed* power budget to the IO and memory domains
- The power budget corresponds to the worst-case performance demands (bandwidth/latency)
- Running three SPEC CPU2006 workloads using baseline and MD-DVFS



Observation 2: Power Budget Redistribution

- PBM (Power Budget Management Algorithm) employs a power budget redistribution mechanism between components within a domain
 - E.g., between cores and graphics engines
- Current PBMs do not support dynamic power redistribution across different domains
- We evaluate the impact of increasing CPU cores' frequency of the MD-DVFS setup from 1.2GHz to 1.3GHz (redistribute power budget)



Observation 2: Power Budget Redistribution

- PBM (Power Budget Management Algorithm) employs a power budget redistribution mechanism between components within a domain
 - E.g., between cores and graphics engines
- Current PBMs do not support dynamic power redistribution across different domains

Applying DVFS in IO and memory domains and redistributing the saved power budget between domains can improve performance in compute-bound workloads.



Observation 3: Memory Bandwidth Demands

- Multiple components in the IO and compute domains have widely-varying main memory bandwidth demands across different workloads.
- For example, memory bandwidth demands of some SPEC CPU2006 and 3DMark workloads over time look like:



Observation 3: Memory Bandwidth Demands

- Multiple components in the IO and compute domains have widely-varying main memory bandwidth demands across different workloads.
- Memory bandwidth demand of different SoC engines:



Observation 3: Memory Bandwidth Demands

- Multiple components in the IO and compute domains have widely-varying main memory bandwidth demands across different workloads.
- Memory bandwidth demand of different SoC engines:

Typical workloads have **modest memory demands**.

Yet, the SoC IO and memory domains are provisioned high.

This makes existing mobile SoCs inefficient for typical workloads.



Observation 4: Optimizing DRAM Configuration

- Memory reference code (MRC) training is part of the BIOS code that manages system memory initialization
- The purpose of MRC training is to:
 - Detect the DIMMs and their capabilities
 - Configure the configuration registers (CRs) of the memory controller (MC), DDRIO, and DIMMs
- Compared to using optimized MRC values for a given frequency, unoptimized MRC values can greatly degrade:
 - Average power (by 22%)
 - Performance (by 10%)



Observation 4: Optimizing DRAM Configuration

• Memory reference code (MRC) training is part of the BIOS code that manages system memory initialization

Average Power
Performance
220/
22%

Optimizing the DRAM configuration for each frequency is very important for multi-domain DVFS energy efficiency

(%)

- Compared to using optimized MRC values, unoptimized MRC values can greatly degrade:
 - Average power (by 22%)
 - Performance (by 10%).



Our Goal: Holistic SoC Power Management

- We conclude that a *holistic power management approach* is **needed** to mitigate the power management inefficiencies in current mobile SoCs
- Our goal is to provide such an efficient multi-domain power management approach
 - by dynamically orchestrating the distribution of the SoC power budget across the *three* domains based on their actual performance demands

Presentation Outline

1. Overview of a Modern SoC

2. Motivation and Goal

3. SysScale

- I. Power Management Flow
- II. Demand Prediction Mechanism
- III. Holistic Power Management Algorithm

4. Evaluation

5. Conclusion

SysScale

- **SysScale** is a new multi-domain power management technique to improve the energy efficiency of mobile SoCs
- SysScale is based on three key ideas:
 - 1. A new DVFS (dynamic voltage and frequency scaling) mechanism to distribute the SoC power to each domain based on its predicted performance demand
 - 2. An accurate algorithm to predict each domain's performance demand
 - 3. Domain-specialized techniques to optimize the energy efficiency of each domain at different operating points

SysScale Architecture

SysScale has three key components:

- A power management flow that is responsible for the DVFS process and reconfiguring DRAM with optimized MRC values
- 2. A demand prediction mechanism that predicts the performance demands from each SoC domain
- 3. A holistic power management algorithm that is responsible for DVFS of the SoC domains to meet the system's dynamic performance demand and redistributing the power budget across domains

Presentation Outline

- 1. Overview on Mobile SoC
- 2. Motivation and Goal

3. SysScale

- I. Power Management Flow
- **II. Demand Prediction Mechanism**
- **III. Holistic Power Management Algorithm**

4. Evaluation

5. Conclusion

Power Management Flow

- The SysScale power management flow is responsible for adjusting the frequencies and voltages of the IO interconnect and memory subsystem. The flow steps are:
 - 1. A demand prediction mechanism decides on target operating point
 - Evaluated every 30ms (configurable, tuned post-silicon)
 - 2. Adjust voltages depending on the DVFS direction
 - 3. Block and drain IO and memory domains
 - 4. Enter DRAM into self refresh
 - 5. Load optimized MRC (Memory reference code) values for the new operating point
 - 6. Adjust clock frequencies
 - 7. Resume SoC operation



Demand Prediction Mechanism

- Demand prediction mechanism predicts the performance demands using:
 - Peripheral configuration registers
 - Performance counters
- Peripheral configuration registers indicate the active devices and their configuration:
 - For example, number of active displays or cameras and frame rates
- We use 4 performance counters with thresholds corresponding to each counter:
 - The performance counter indicates the bandwidth/latency demand of CPU cores, graphics engines and IO devices

Holistic Power Management Algorithm

- SysScale implements a power distribution algorithm in the power management unit (PMU) firmware
- The system moves the IO and memory domains to a high- or low-performance operating point based on the decision of the demand prediction mechanism
 - The system redistributes the power budget across SoC domains when changing the operating point
- For example, when the SoC moves to a low-performance operating point, the PMU:
 - Reduces the power budgets of the IO and memory domains and
 - Increases the power budget of the compute domain

Presentation Outline

- 1. Overview on Mobile SoC
- 2. Motivation and Goal
- 3. SysScale
 - I. Power Management Flow
 - II. Demand Prediction Mechanism
 - III. Holistic Power Management Algorithm

4. Evaluation

5. Conclusion

Methodology

- **<u>System</u>**: We implement SysScale on a real Intel Skylake system
 - For our baseline measurements, we disable SysScale
 - We use an SoC with 4.5W thermal design power (TDP)
- <u>Power Measurements</u>: We use National Instruments data acquisition device for power measurement
- **Workloads**: We evaluate SysScale with three classes of workloads
 - CPU: SPEC CPU2006 benchmarks
 - Graphics: 3DMARK benchmarks
 - Battery life: web browsing, light gaming, video conferencing, and video playback benchmarks
- <u>Comparison Points</u>: We compare SysScale to the two most relevant prior works, MemScale [Deng+, ASPLOS 2011] and CoScale [Deng+, MICRO 2012]:
 - MemScale applies DVFS only for memory subsystem
 - CoScale coordinates CPU cores and memory subsystem DVFS

Results – CPU Workloads



- SysScale improves real system performance by 9.2% on average
- SysScale provides 5.4×/2.4× the performance improvement of MemScale/CoScale, because:
 - SysScale is holistic, taking into account all SoC domains
 - Unlike the other mechanisms, SysScale optimizes the DRAM interface
- The performance benefit of SysScale correlates with the performance scalability of the running workload with CPU frequency

Results – CPU Workloads



SysScale significantly improves CPU core performance by holistically applying DVFS to SoC domains and redistributing power budget

- SysScale provides 5.4×/2.4× the performance improvement of MemScale/CoScale, because:
 - SysScale is holistic, taking into account all SoC domains
 - Unlike the other mechanisms, SysScale optimizes the DRAM interface
- The performance benefit of SysScale correlates with the performance scalability of the running workload with CPU frequency.

Results – Graphics Workloads



- SysScale improves real system performance (6.7%-8.9%) because it boosts the graphics engines' frequency by redistributing the power budget across the three domains
- SysScale provides approximately 5× the performance improvement of MemScale and CoScale
- MemScale and CoScale have similar performance improvements because their average power savings are identical.
 - In graphics workloads, CPU cores run at the lowest possible frequency
 - CoScale cannot further scale down the CPU frequency

Results – Graphics Workloads



SysScale significantly improves the graphics performance using the saved power budget from IO and memory domains

or memocale and Coocale

- MemScale and CoScale have similar performance improvements because their average power savings are identical.
 - In graphics workloads, CPU cores run at the lowest possible frequency
 - CoScale cannot further scale down the CPU frequency.

Results – Battery Life Workloads



• Battery life workloads have fixed performance requirements

- SysScale reduces average power (6.4%-10.7%) on a real system and workloads
- SysScale provides approximately 5× the power reduction of MemScale and CoScale
- MemScale and CoScale have similar average power reduction
 - In battery life workloads, CPU cores run at the lowest possible frequency
 - CoScale cannot further scale down the CPU frequency

Results – Battery Life Workloads



- SysScale provides approximately 5× the power reduction of MemScale and CoScale
- MemScale and CoScale have similar average power reduction
 - In battery life workloads CPU cores run at the lowest possible frequency
 - CoScale cannot further scale down the CPU frequency.

Other Results in the Paper

SysScale performance and average power consumption sensitivity to:

- System Thermal Design Power (TDP)
 - SysScale's performance benefit increases as TDP decreases
 - SysScale improves energy consumption across the entire TDP range (3.5W–91W)
- Different DRAM frequencies and types

Presentation Outline

- 1. Overview of a Modern SoC
- 2. Motivation and Goal
- 3. SysScale
 - I. Power Management Flow
 - II. Demand Prediction Mechanism
 - III. Holistic Power Management Algorithm

4. Evaluation

5. Conclusion

Conclusion

- SysScale is the first work to enable coordinated and highly-efficient DVFS across all SoC domains to increase energy efficiency
- SysScale optimizes and efficiently redistributes the total power budget across all SoC domains based on the performance demands of each domain
- We implemented SysScale on the Intel Skylake SoC for mobile devices
 - SysScale improves the performance of real CPU and graphics workloads (by up to 16% and 8.9%, respectively, for 4.5W TDP)
 - SysScale reduces the average power consumption of battery life workloads (by up to 10.7%) across all TDPs of the Intel Skylake system
- We **conclude** that SysScale is an effective approach to balance power consumption and performance demands across all SoC domains

SysScale: Exploiting Multi-domain Dynamic Voltage and Frequency Scaling for Energy Efficient Mobile Processors

Jawad Haj-Yahya¹

Mohammed Alser¹ Jeremie Kim¹ A. Giray Yaglıkçı¹ Nandita Vijaykumar^{1,2,3} Efraim Rotem² Onur Mutlu¹

