System-wide Energy Optimization for Multiple DVS Components and Real-time Tasks

Heechul Yun, Po-Liang Wu, Anshu Arya, Tarek Abdelzaher,
Cheolgi Kim, and Lui Sha

DVS in Real-time Systems

- The Goal
  - To minimize energy consumption by adjusting freq. and voltage but still meet the deadline

- Most consider CPU only
  - Assume execution time depends on CPU freq.

- **But memory and bus are also important**
  - Affect execution time (e.g., memory intensive app will be slowed if memory or bus is slow.)
  - Consume considerable energy (similar order of energy compared to CPU)
  - *Are DVS capable in many recent embedded processors*

Motivation

Memxfer5b : memory benchmark program

CPU(Mhz)	Mem(Mhz)	Time(s)	Energy(mJ)
200	100	3.46	1690
100	100	3.55	1182

Half of CPU clock

Exec. time increased only 3%

Energy saved 30%

Motivation

Dhrystone: CPU benchmark program

CPU(Mhz)	Mem(Mhz)	Time(s)	Energy(mJ)
200	100	4.26	2364
200	50	4.28	2106

Half of Mem clock

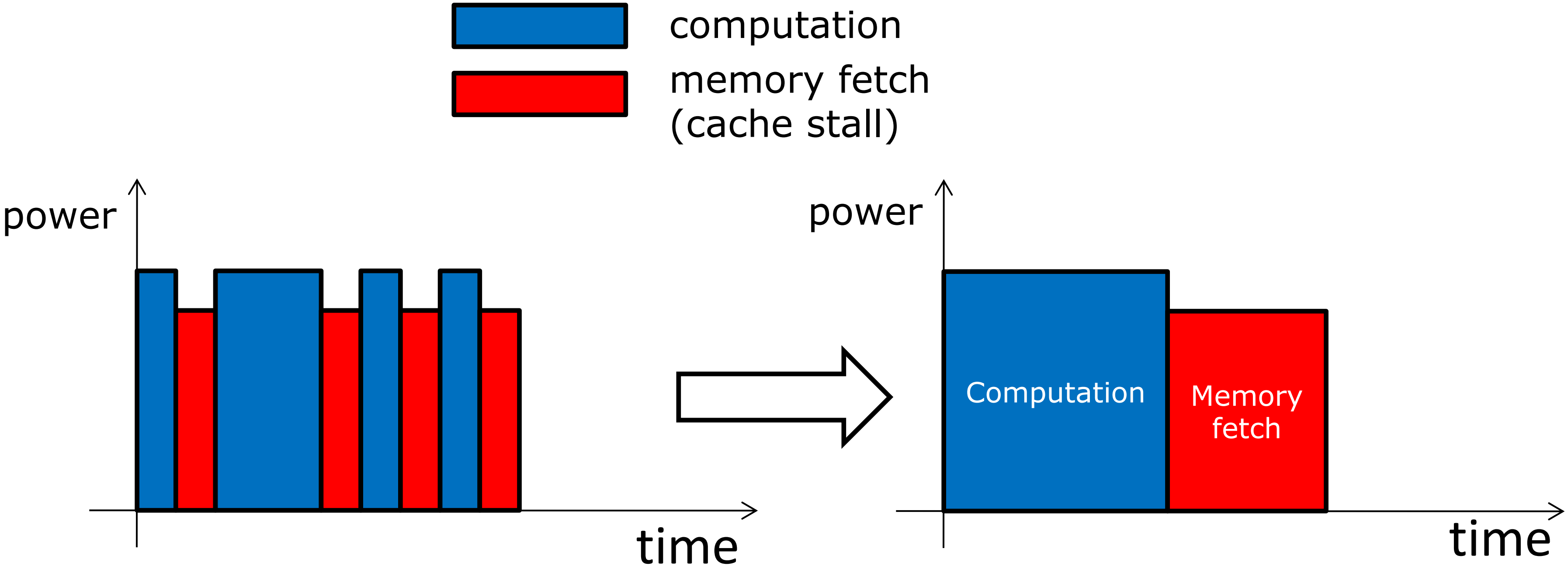
Exec time increased only 0.05%

Energy saved 10%

Contents

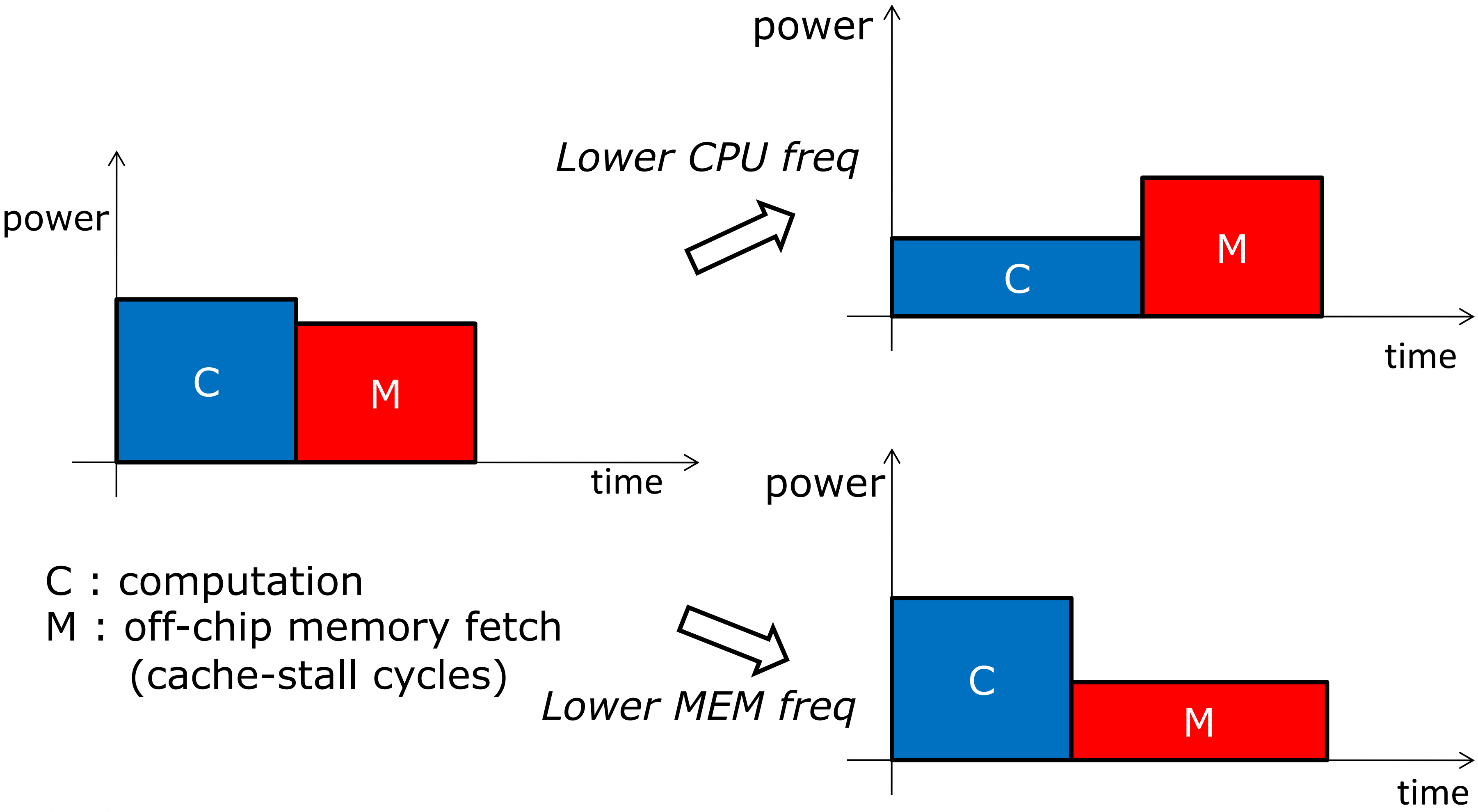
- Motivation
- **Energy Model**
  - **Considers CPU, BUS and Memory and task characteristics**
  - **Evaluation (Model validation)**
- Energy Optimization of Real-time Tasks
  - Static multi-DVS problem and solution
  - Evaluation
- Conclusion

Task Model



- Task = Computation + Memory fetch

Task Model (2)



C : computation
M : off-chip memory fetch
(cache-stall cycles)

Task Model (3)

- Execution time of a task

$$e = \frac{C}{f_c} + \frac{M}{f_m}$$

  - C : CPU cycles of a given task
  - M : memory cycles of a given task
  - f_c : CPU clock frequency
  - f_m : Memory clock frequency

Power Model

- Power of a component (i.e., CPU)

$$W = kfV^2 + R$$

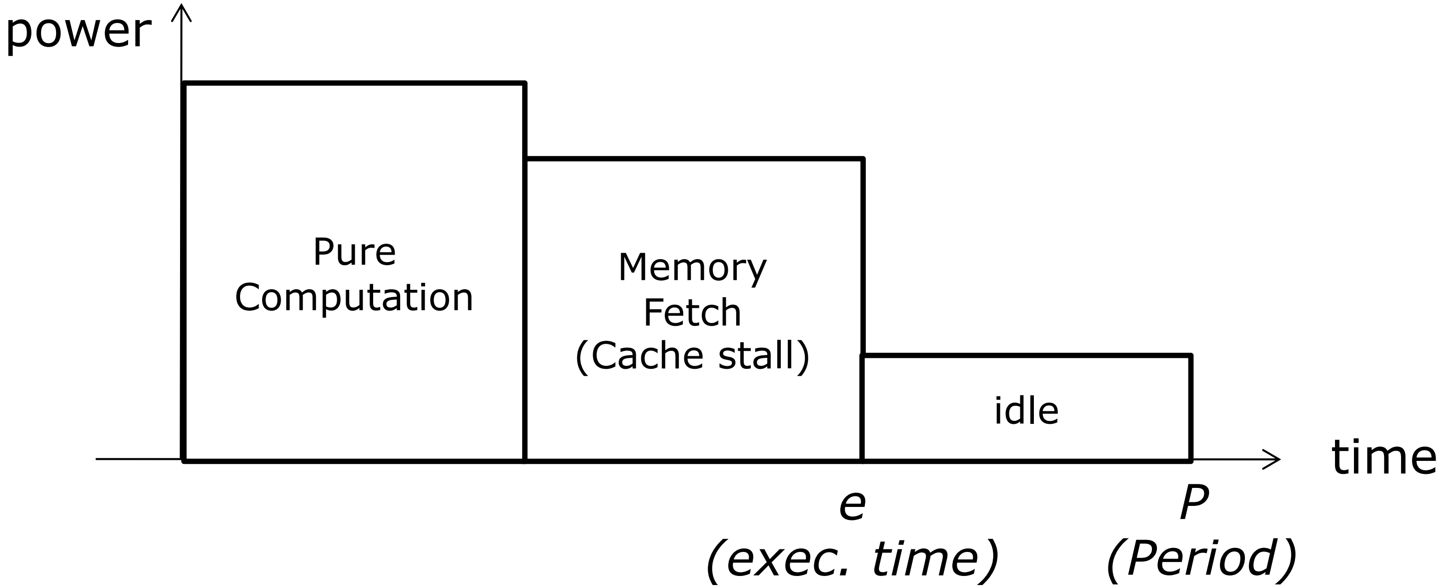
  – *k* : capacitance constant

Different *k* for different modes:

k_{active} - active mode capacitance

$k_{standby}$- standby mode capacitance

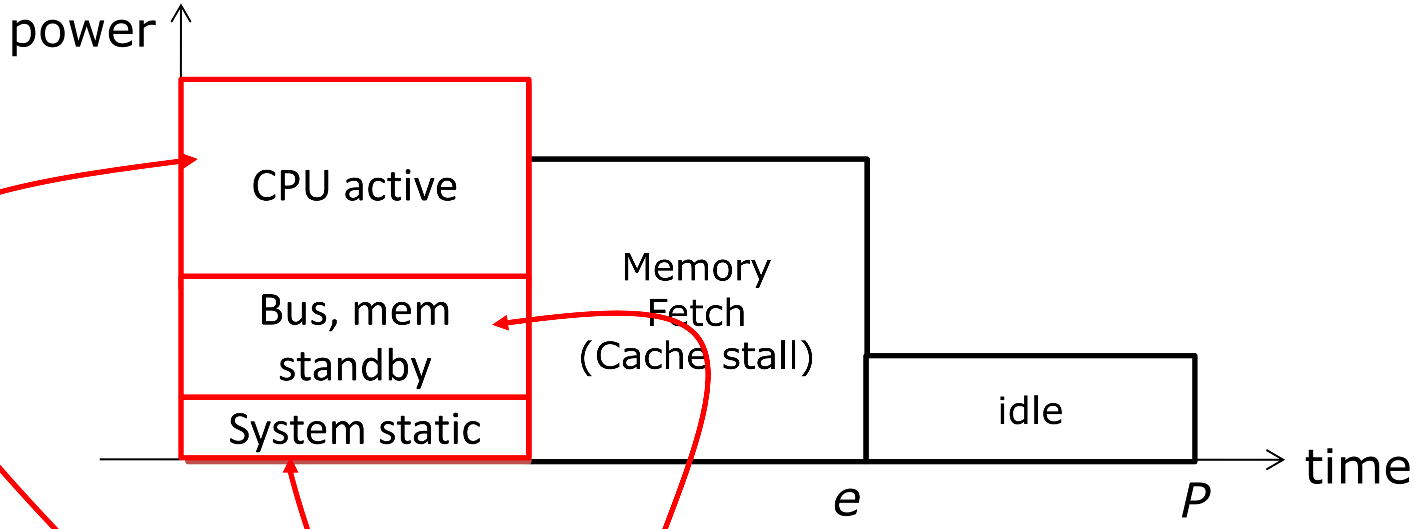
Energy Model



- Total system energy is

$$E = E_{comp} + E_{mem} + E_{idle}$$

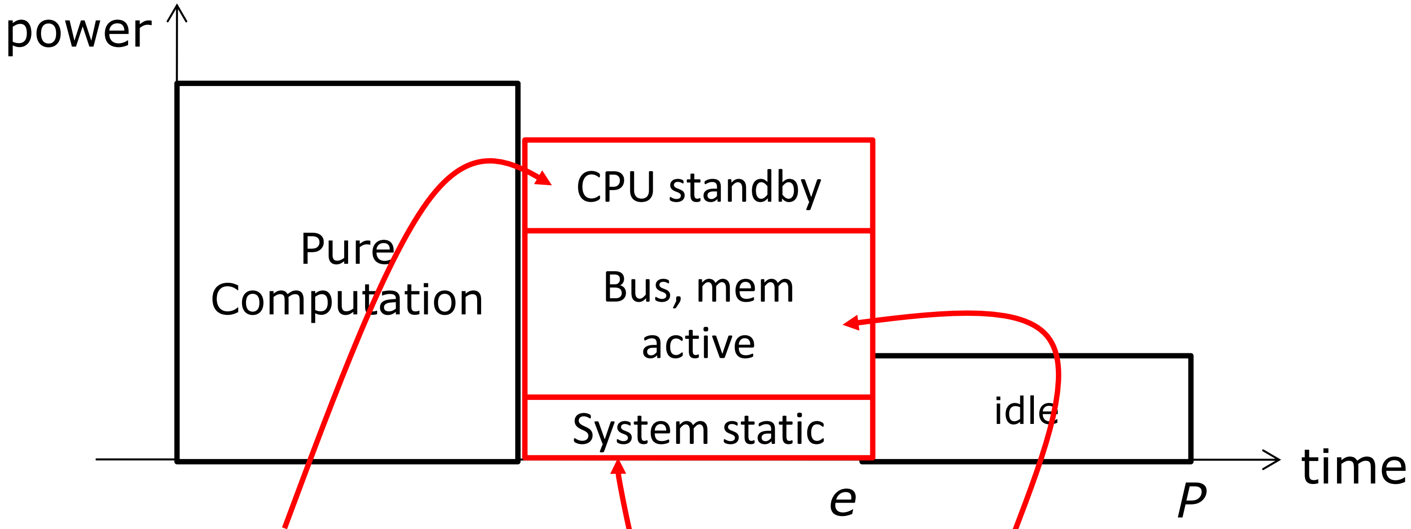
Pure Computation Block



$$E_{comp} = (\boxed{k_{ca}V_{cpu}^{2} f_{c}} + \boxed{k_{bs}V_{bus}^{2} f_{b} + k_{ms}V_{mem}^{2} f_{m}} + \boxed{R}) \times \frac{C}{f_{c}}$$

- k_{ca} : capacitance constant for **active** cpu
- k_{bs} : capacitance constant for **standby** bus
- k_{ms} : capacitance constant for **standby** memory
- R : system wide static power consumption

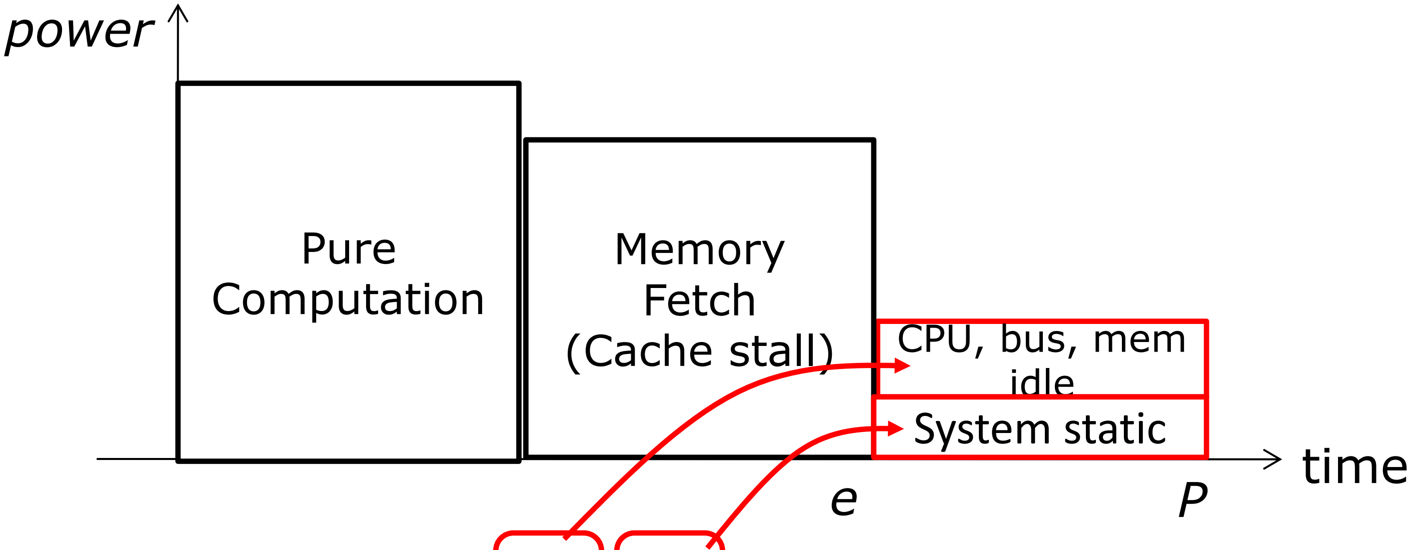
Memory Fetch Block



$$E_{mem} = (\boxed{k_{cs}V_{cpu}^{2} f_{c}} + \boxed{k_{ba}V_{bus}^{2} f_{b} + k_{ma}V_{mem}^{2} f_{m}} + \boxed{R}) \times \frac{M}{f_{m}}$$

- k_{cs} : capacitance constant for **standby** cpu
- k_{ba} : capacitance constant for **active** bus
- k_{ma} : capacitance constant for **active** memory

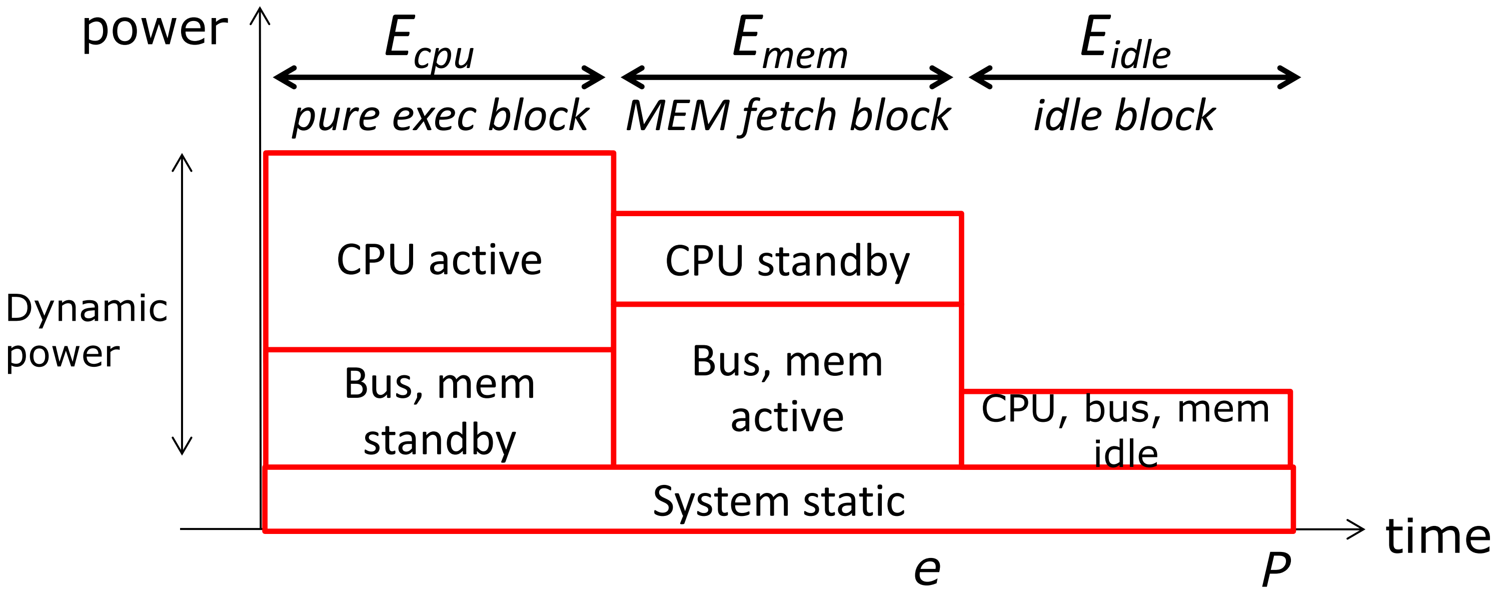
Idle Block



$$E_{idle} = (\boxed{I} + \boxed{R}) \times (P - e)$$

- I : idle mode power consumption.
- e: execution time $(C/f_c + M/f_m)$

Energy Model Summary



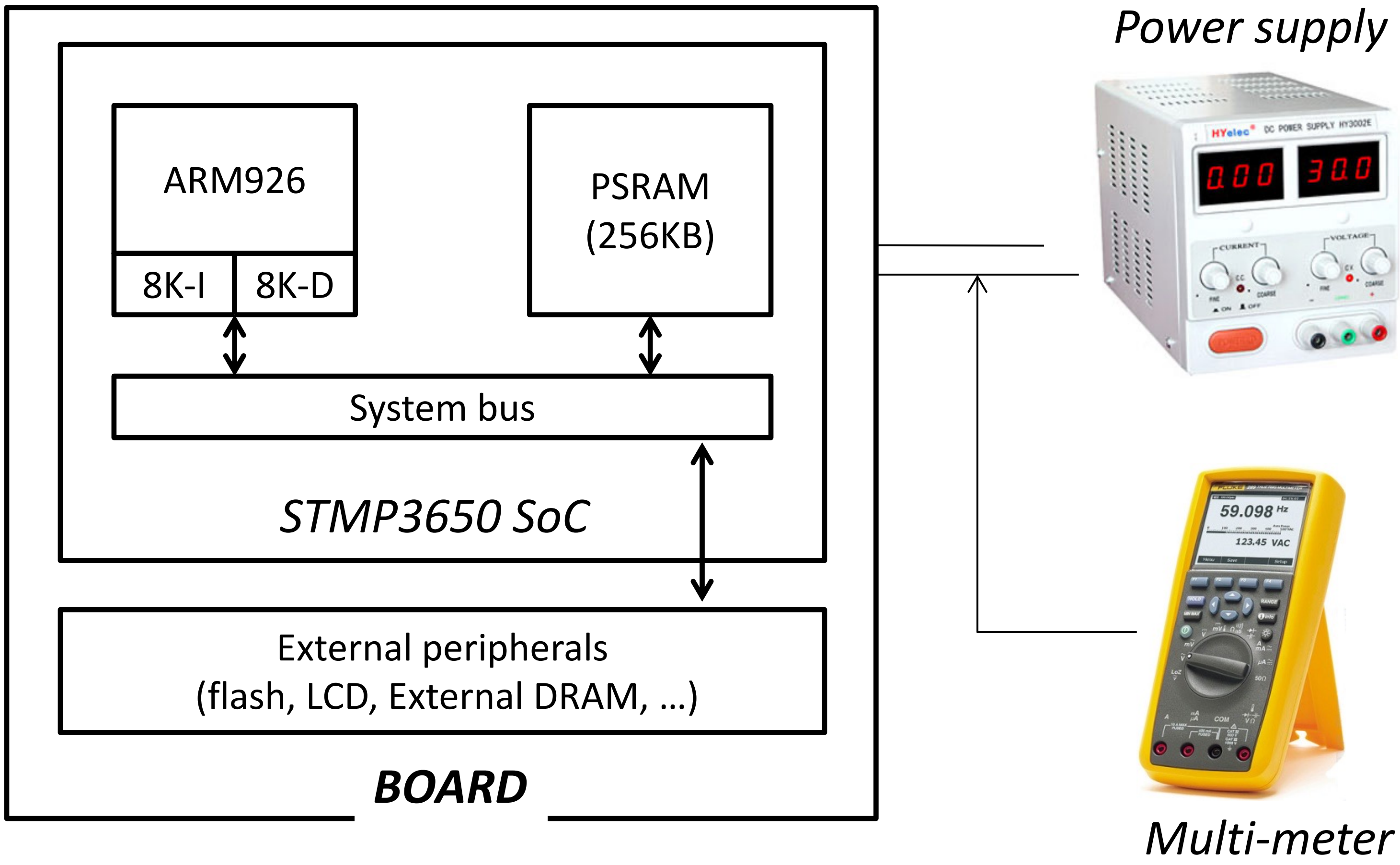
- System wide energy model
  - Considers CPU, bus, and memory power consumption
  - Considers active, standby and idle modes
  - Other components are assumed to be static (included in *R*)

Energy Equation

$$
\begin{aligned}
E &= E_{comp} + E_{mem} + E_{idle} \\
&= (k_{ca}V_{cpu}^{2}f_{c} + k_{bs}V_{bus}^{2}f_{b} + k_{ms}V_{mem}^{2}f_{m} + R) \times \frac{C}{f_{c}} && \text{-------- CPU block} \\
&+ (k_{cs}V_{cpu}^{2}f_{c} + k_{ba}V_{bus}^{2}f_{b} + k_{ma}V_{mem}^{2}f_{m} + R) \times \frac{M}{f_{m}} && \text{-------- Memory block} \\
&+ (I + R) \times (P - e) && \text{-------- Idle block}
\end{aligned}
$$

- System-wide energy consumption of a task during period *P*

Evaluation Platform



ARM926

8K-I

8K-D

PSRAM
(256KB)

System bus

STMP3650 SoC

External peripherals
(flash, LCD, External DRAM, …)

BOARD

Power supply

Multi-meter

Evaluation Platform (2)

- ARM9 based SoC
  - CPU : up to 200Mhz, BUS : up to 100Mhz
  - CPU and BUS are synchronous (BUS = CPU/N)
  - **Memory (PSRAM) freq is equal to system bus frequency (f_b=f_m)**
  - **CPU, BUS, and memory all share the common voltage**
  - Vdd : 1.504V ~ 1.804V (0.32V step)

- Energy equation

$$E = (k_{ca}V^2 f_c + \boxed{k^*_{ms}V^2 f_m} + R)\times\frac{C}{f_c} + (k_{cs}V^2 f_c + \boxed{k^*_{ma}V^2 f_m} + R)\times\frac{M}{f_m}$$

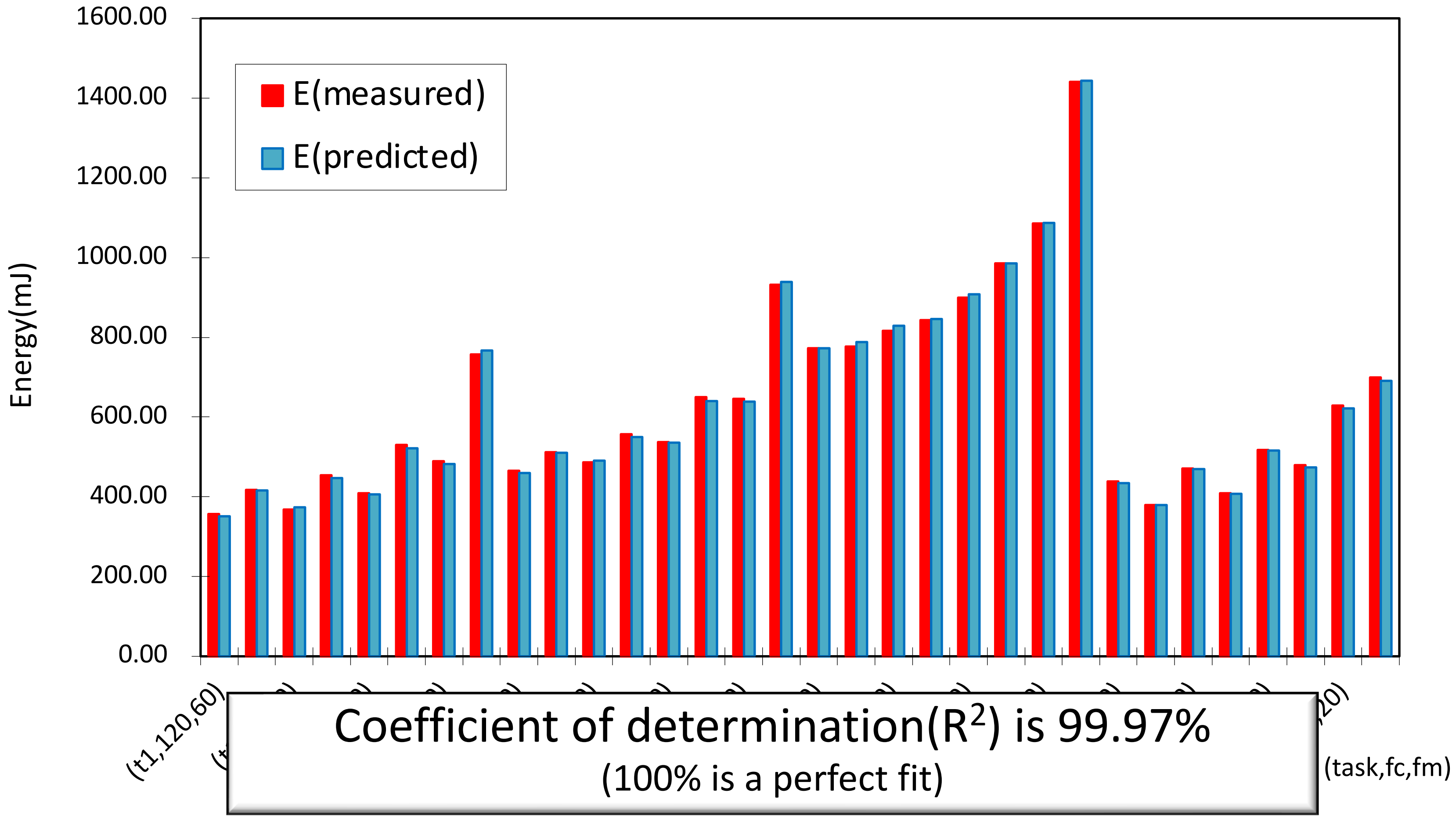
$$+(I+R)\times(P-e)$$

  - *V* : *shared voltage* for CPU, bus, and memory
  - k^*_{ma} : active bus and memory constant
  - k^*_{ms} : standby bus and memory constant

Validation

- Methodology
  - 4 synthetic programs with different cache stall ratio (0%, 10%, 25%, 55%)
  - 8 clock configurations (f_c, f_m) for each program
  - Performed nonlinear least square analysis for total 32 data points against the energy equation

Energy Model Fitting



Coefficient of determination(R^2) is 99.97%

(100% is a perfect fit)

Energy Equation for Our Platform

$$E \ = (k_{ca} V^2 f_c + k_{ms}^{*} V^2 f_m + R) \times \frac{C}{f_c} + (k_{cs} V_{cpu}^2 f_c + k_{ma}^{*} V^2 f_m + R) \times \frac{M}{f_m}$$

$$+ (I + R) \times (P - e)$$

Capacitance (nF)				Power (mW)	
K_{ca}	K_{cs}	K_{ma}*	K_{ms}*	I	R
0.505	0.224	0.540	0.210	6.570	67.434

Obtained coefficients in the energy equation

Contents

- Motivation

- Energy Model
  - Considers CPU, BUS and Memory and task characteristics
  - Evaluation

- **Energy Optimization of Real-time Tasks**
  - **Static Multi-DVS Problem and optimal solution**
  - **Evaluation**

- Conclusion

Static Multi-DVS Problem

- Given a set of periodic real-time tasks (T_1, …, T_n), where each task invocation requires up to C_i CPU cycles and up to M_i memory cycles at worst.

- Find the energy optimal **static frequencies** for **multiple DVS capable components** (CPU, bus, and memory)

Problem Formulation

Minimize $$\sum_{i=1}^{n} \frac{H}{P_i}(E_{comp,i} + E_{mem,i}) + E_{idle}$$

Subjects to $$\sum_{i=1}^{n} \frac{e_i}{P_i} \leq 1.$$

where

H : hyper period

e_i : execution time of task i

$E_{comp,i}$: computation block energy of task i

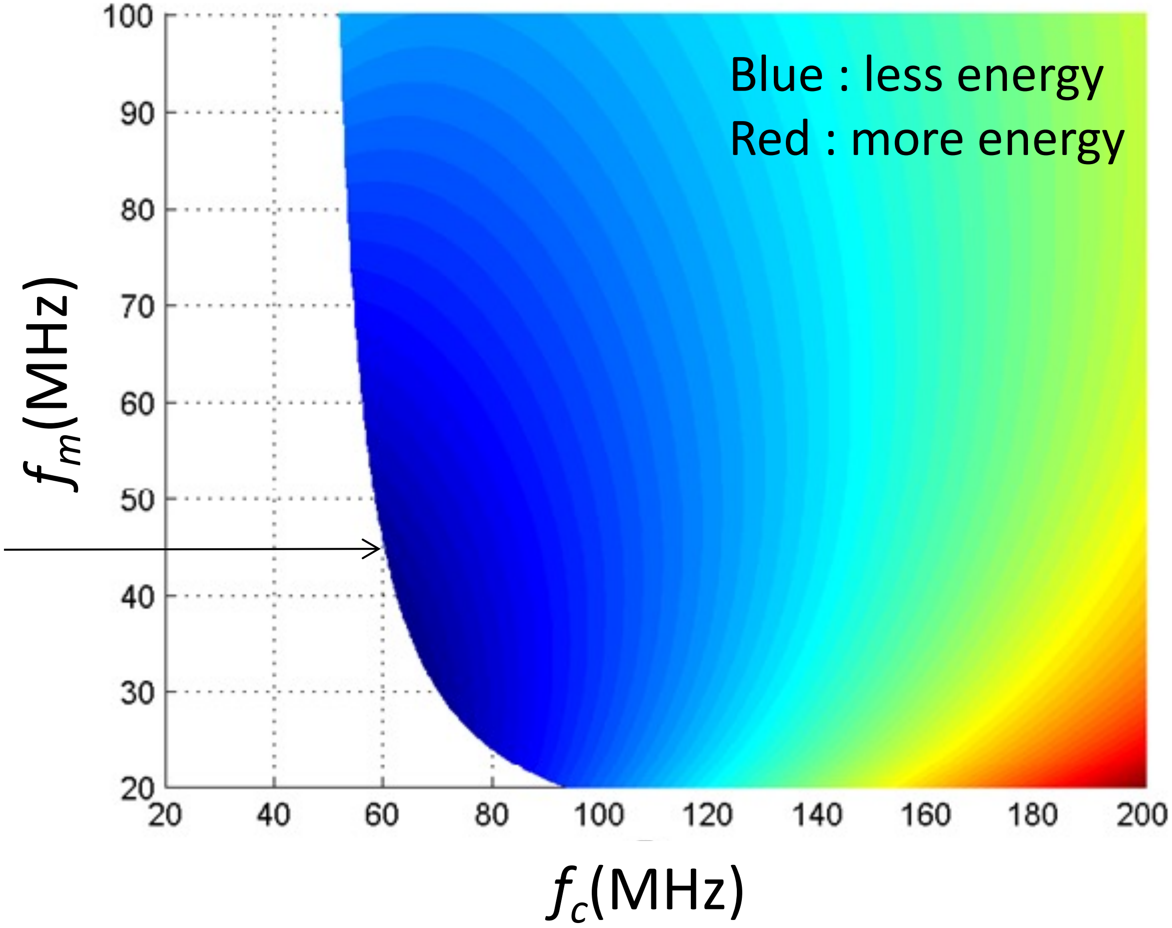
$E_{mem,i}$: cache stall block energy of task i

E_{idle} : idle block energy

Optimal Solution

- Intuitive procedure
  - Find an unconstrained minimal over f_c and f_m ($f_b = f_m$)
  - Check boundary conditions due to system specific constraints. (e.g., minimum and maximum clock range)
  - Details are in the paper

Energy Plot



Blue : less energy

Red : more energy

Deadline boundary

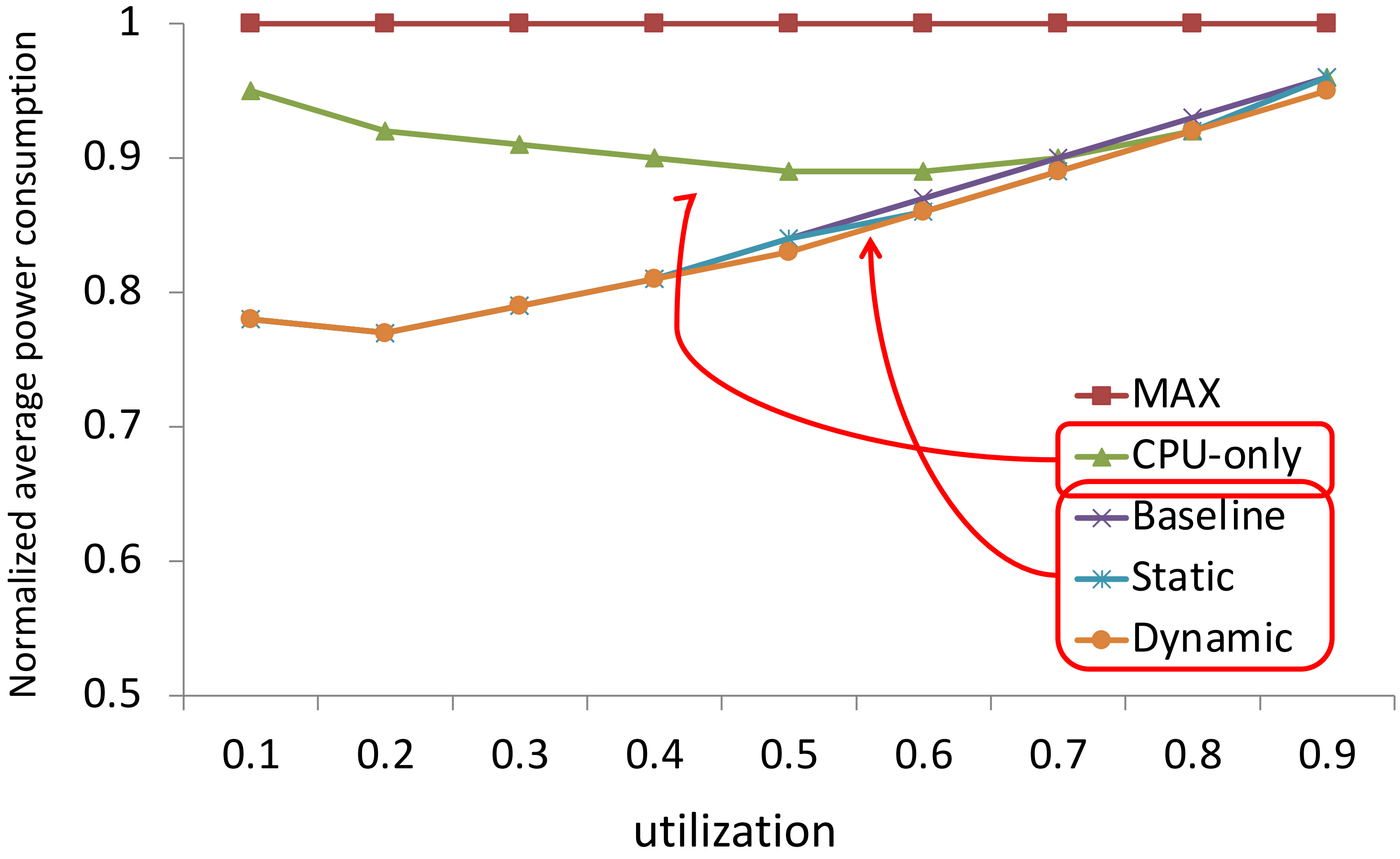
Task set : $C_H = 140*10^6$, $M_H = 30*10^6$,H = 3s

Evaluation

- Compare the following schemes:
  - MAX
    - CPU and memory are all set to maximum.
  - CPU-only static DVS
    - Memory frequency is set to maximum
  - Baseline static multi-DVS
    - CPU and memory frequencies change proportionally
  - **Optimal static multi-DVS**
    - Proposed scheme
  - Optimal dynamic multi-DVS
    - Can change frequencies at each task schedule
    - Brute force search among all the possible combination

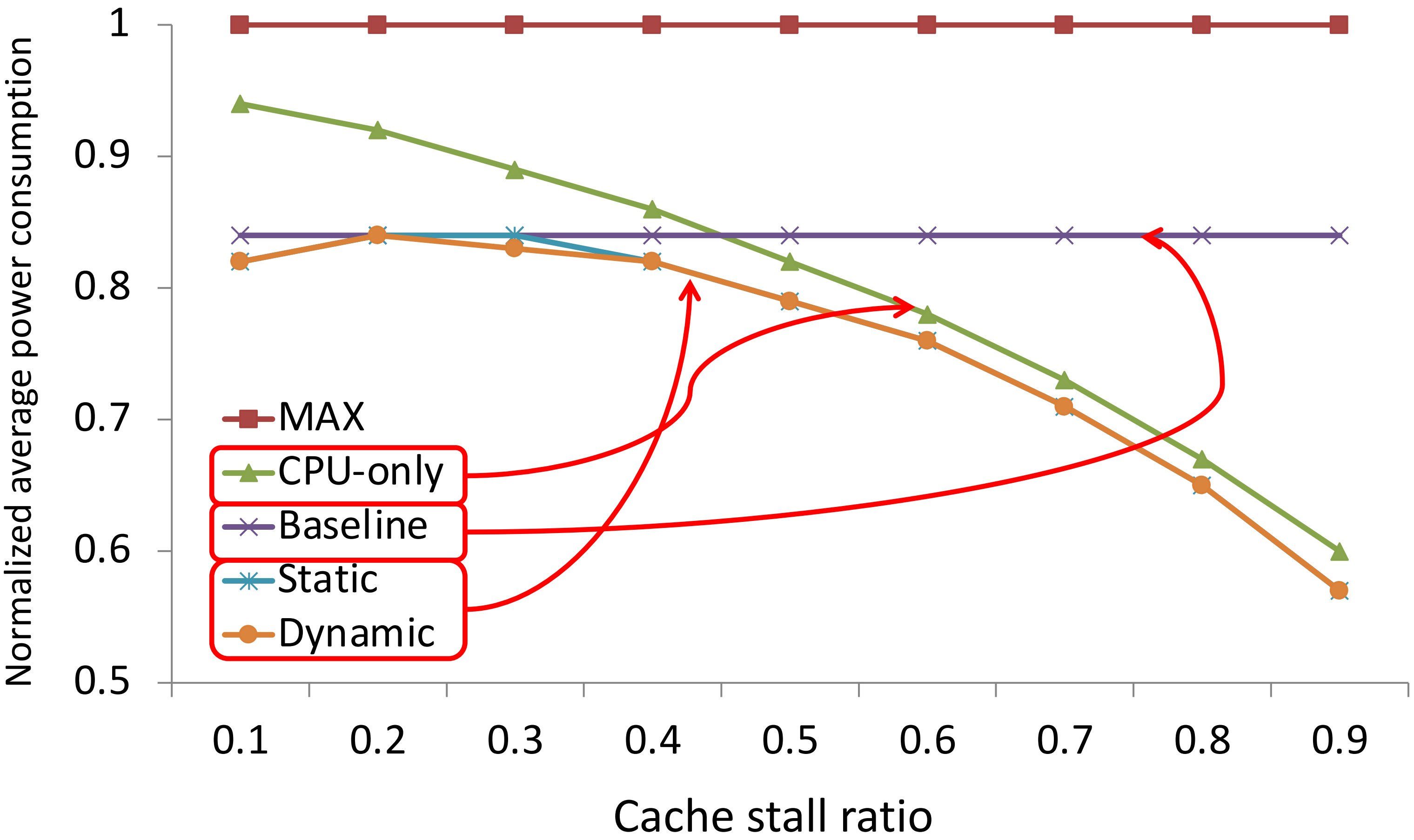
- Simulation setup
  - Use energy equation obtained from measurements on our real hardware platform

Energy vs Utilization



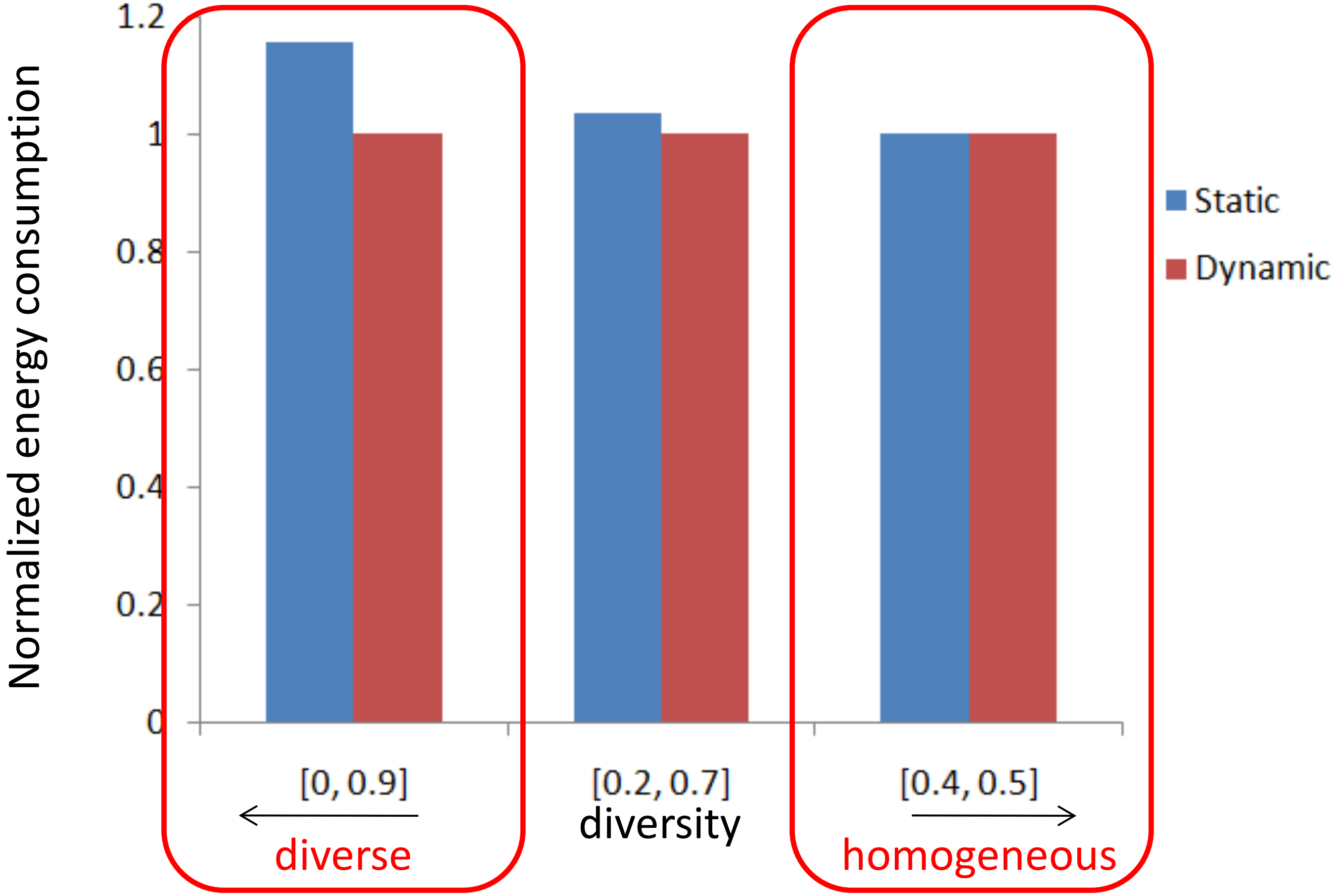
Task set cache stall ratio ($M_H/(C_H+M_H)$): 0.3

Energy vs Cache Stall Ratio



Task set utilization ratio(e_H/H): 0.5

Effect of Diversity of Cache Stall Ratio



Task set cache stall ratio = 0.45,
Task set utilization ratio(e_H/H): 0.5

Conclusion

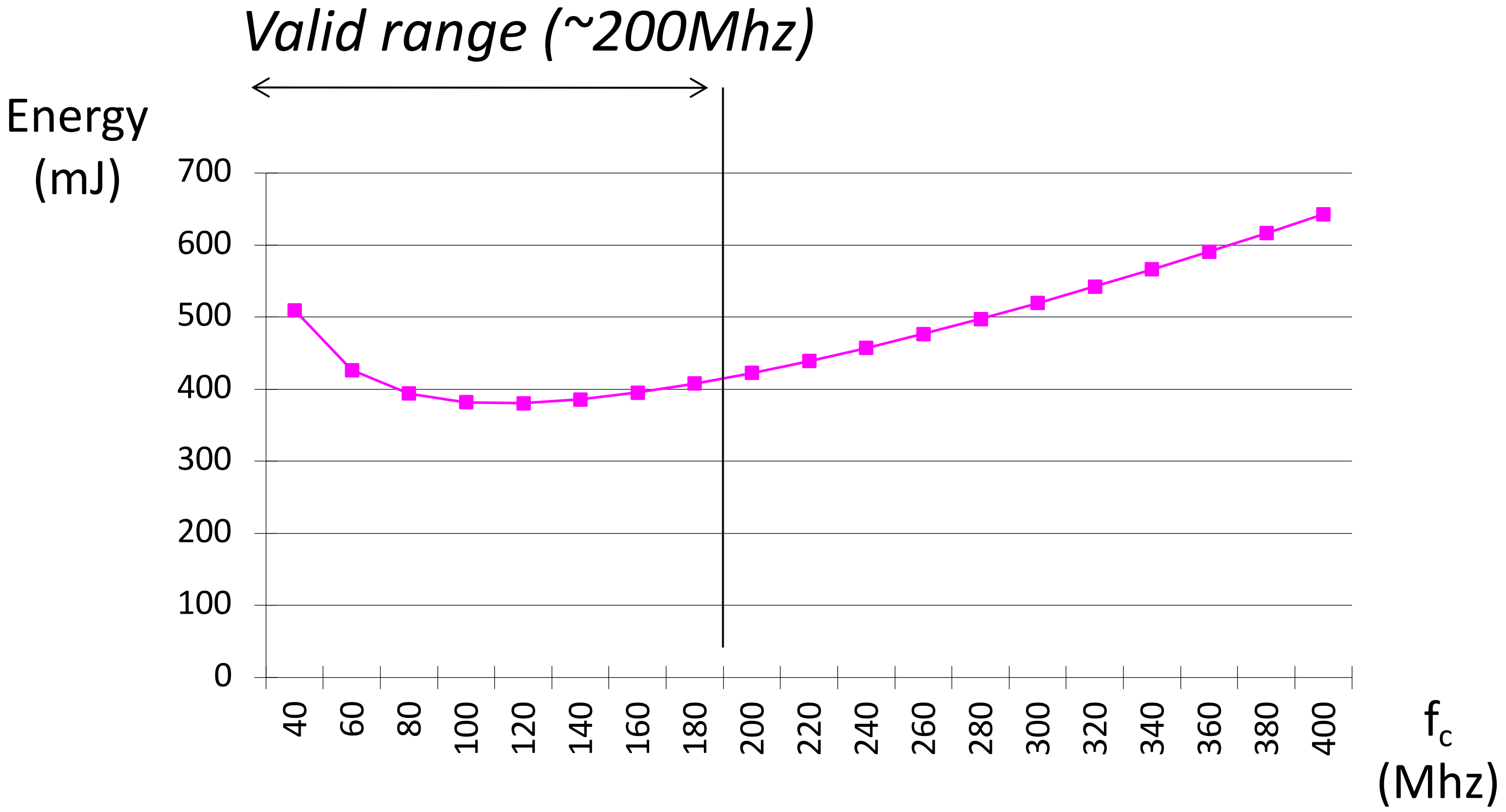
- Energy model
  - Considers multiple DVS capable components and task characteristic
  - Validated on a real hardware platform

- Static multi-DVS problem
  - Assigns energy optimal *static* frequencies of multiple DVS components for periodic real-time tasks
  - Optimal solution (static multi-DVS scheme) shows better energy saving compared to CPU-only DVS

Thank you.

Additional Slides

CPU-only DVS

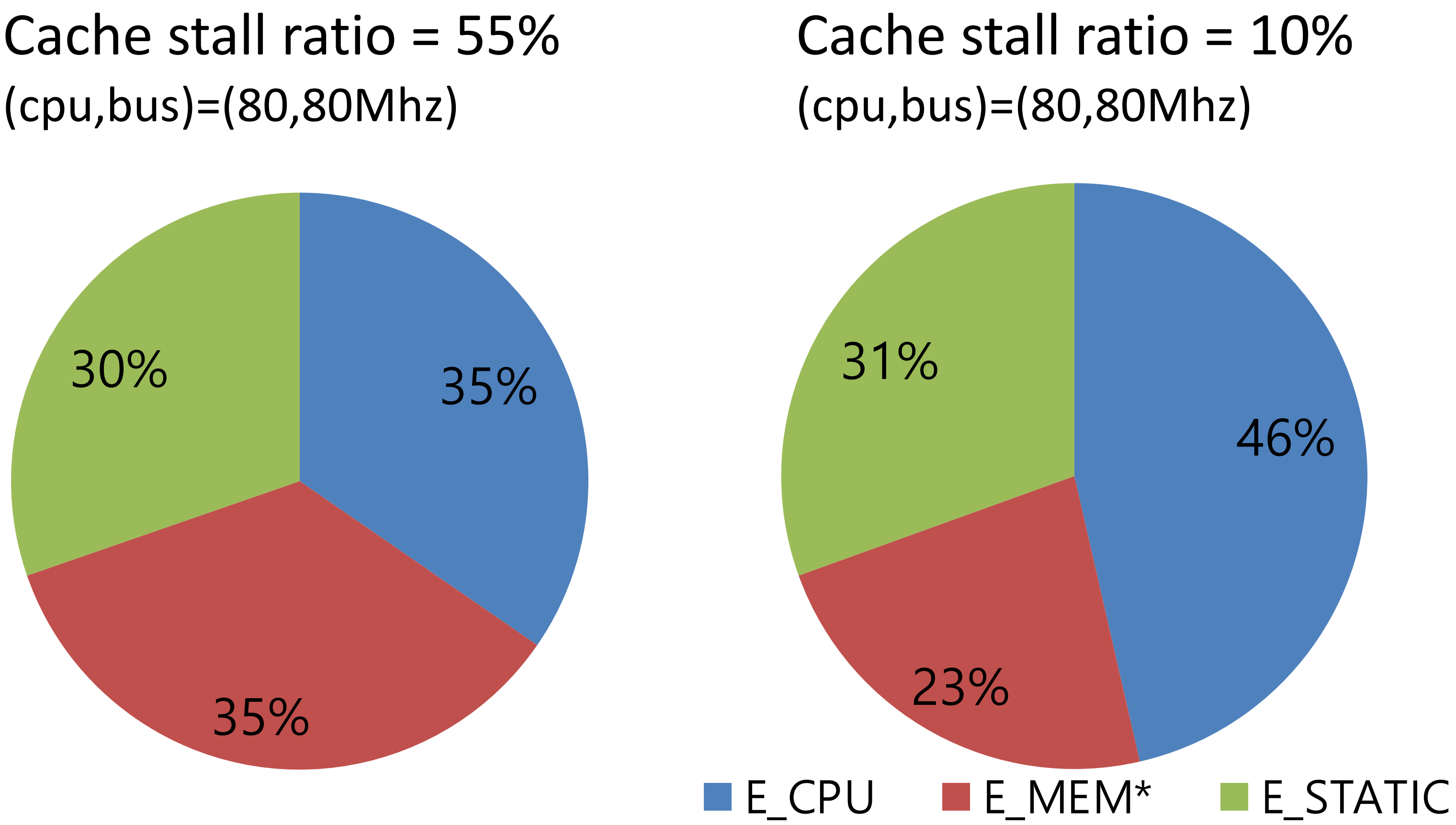


Valid range (~200Mhz)

Not effective in allowed range

(*) based on energy equation for out h/w platform. Memory clock was set to max

Power Distribution



Cache stall ratio = 55%

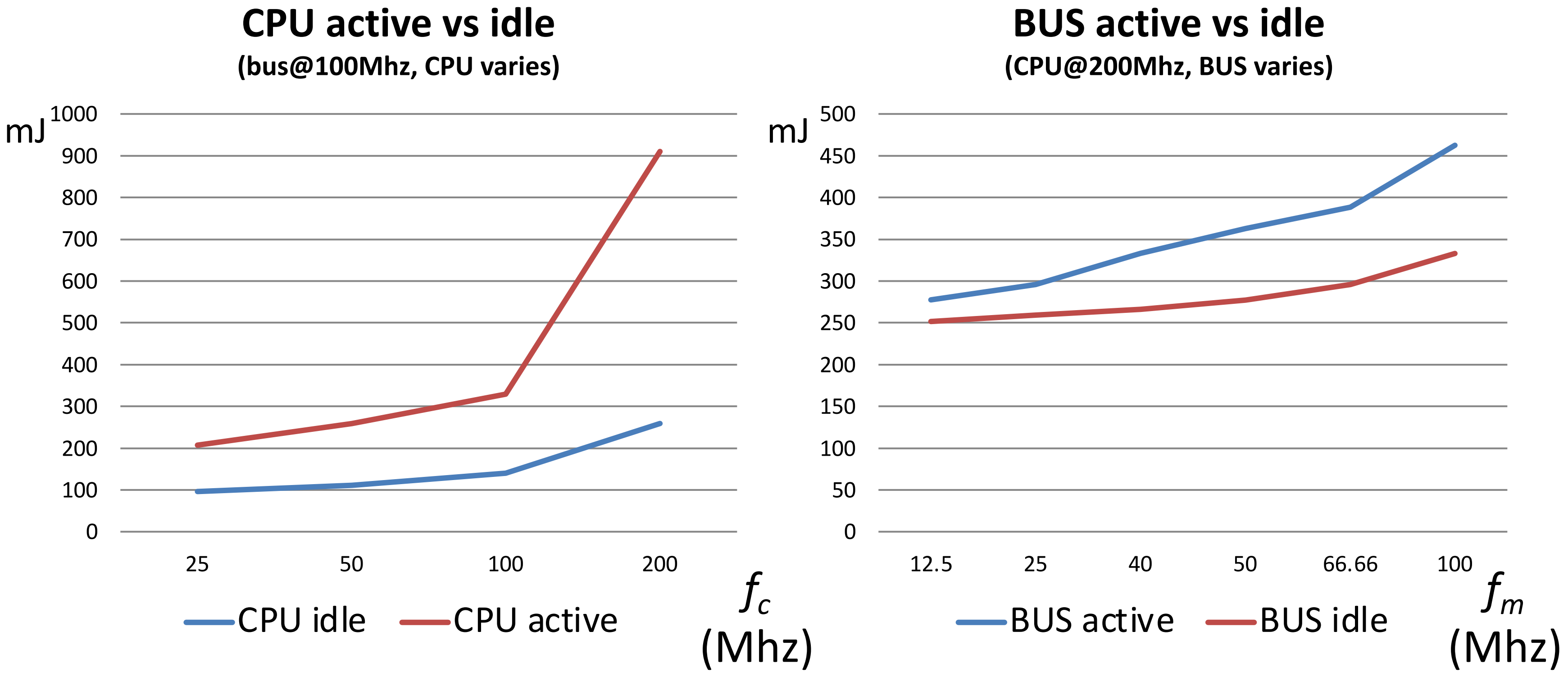
(cpu,bus)=(80,80Mhz)

Cache stall ratio = 10%

(cpu,bus)=(80,80Mhz)

(*) based on energy equation for our h/w platform E = Ecpu + Emem + Estatic

Active and Idle



(*) actual measurement result