An Analytical Dynamic Scaling of Supply Voltage and Body Bias Exploiting Memory Stall Time Variation

[†]**Jungsoo Kim**, [†]Younghoon Lee, [‡]Sungjoo Yoo, and [†]Chong-Min Kyung

[†] Dept. of EE at KAIST, Korea

[‡] Dept. of EE at POSTECH, Korea

Contents

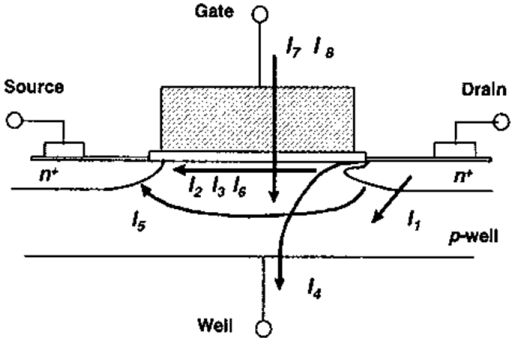
- Introduction
- Motivation
- Related Works
- Preliminary
  - Processor Energy Model
  - Workload Profiling
- Memory Stall Time-Aware DVFS
- Experimental Results
- Conclusion

Introduction

- Dynamic voltage and frequency scaling (DVFS)
  - **Supply voltage scaling**: effective to reduce switching energy consumption

$$P_{sw} = \alpha C_{eff} V_{dd}^2 f$$

- Adaptive body biasing (ABB)
  - **Body bias voltage scaling**: effective to reduce leakage energy consumption



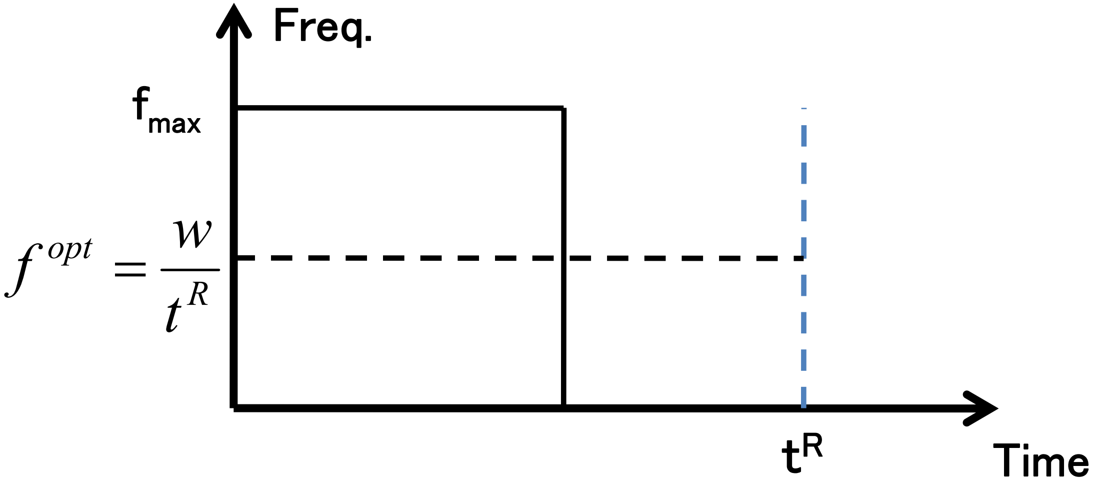
– : reverse body bias (RBB)
+ : forward body bias (FBB)

Introduction

- Dynamic voltage and frequency scaling (DVFS)
  - Processor frequency is set as follows

$$\text{Frequency (f)} = \frac{\text{Remaining workload (w)}}{\text{Remaining time - to - deadline } (\text{t}^{\text{R}})}$$

➔ The **accuracy of remaining workload prediction** plays a crucial role in DVFS.

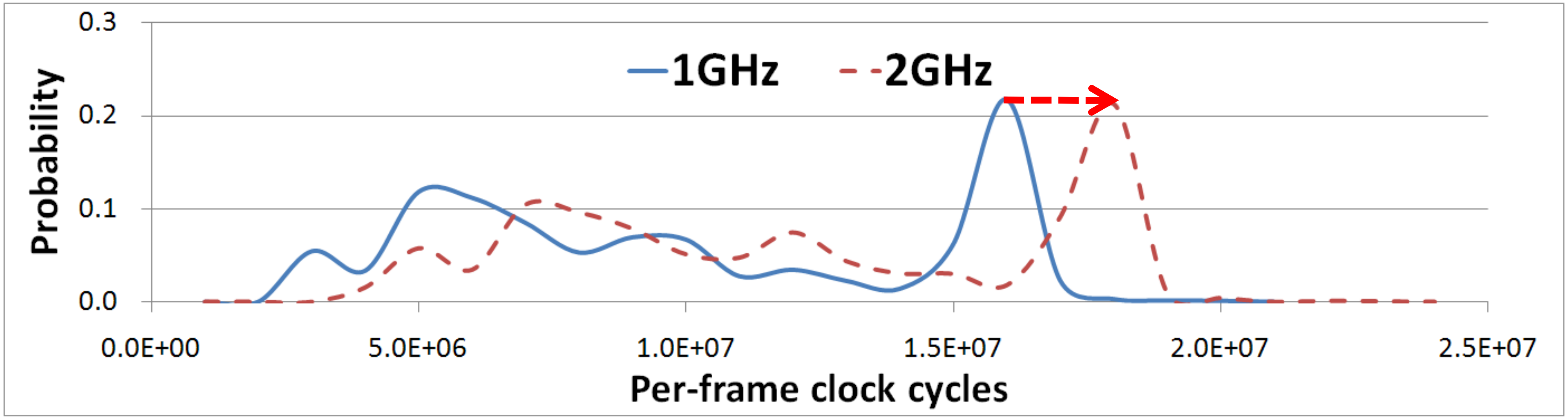


〈 Voltage/frequency scaling in DVFS 〉

Motivation

- Software workload varies due to
  - Data dependency (data dependent loop iterations)
  - Control flow dependency (if/else, switch-case)
  - Architectural dependency (cache/TLB hit/miss)

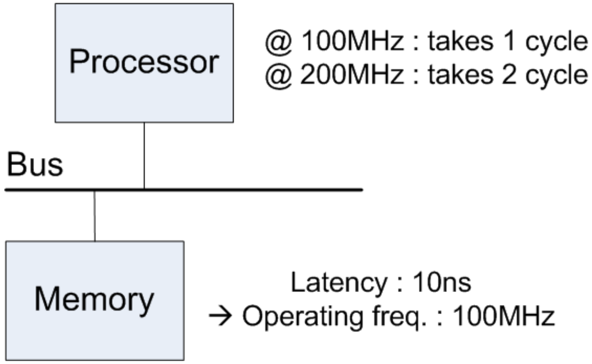
@ *LG XNOTE LW25* laptop



< Per-frame profiling results of MPEG4 decoder when decoding 2000 frames of 1920x800 'Harry Potter'>

Motivation

- Memory stall cycle varies as processor frequency changes.
  - Processor frequency ↑ ➔ number of execution cycles ↑
- Memory stall time has runtime distribution.



Processor

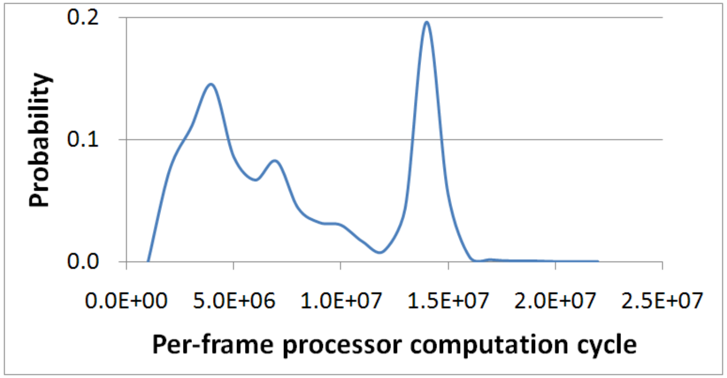
@ 100MHz : takes 1 cycle
@ 200MHz : takes 2 cycle

Bus

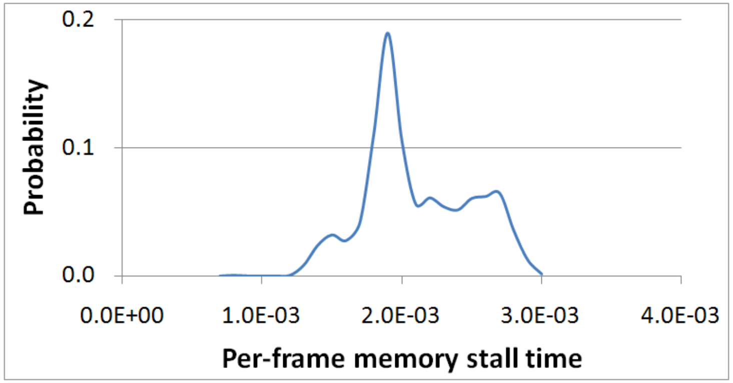
Memory

Latency : 10ns
→ Operating freq. : 100MHz

< An example of memory wall >



< Processor computation Workload>



< Memory stall time>

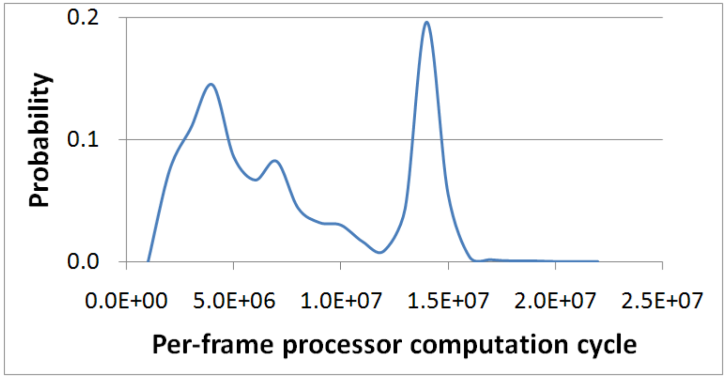
Related Works

- Runtime distribution-aware DVFS [1][2][3][4][5]
  - Only exploit runtime distribution of processor computation workload

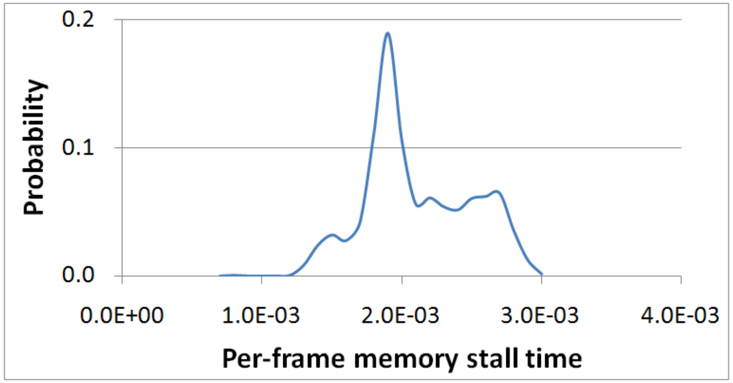
- Memory stall-aware DVFS [6][7]
  - Not exploit workload distribution

Our Contribution

- First approach which tackles the distributions of both processor computation and memory stall workloads and their correlation
  - **9.6% ~ 30.0%** further energy savings compared to the previous approaches



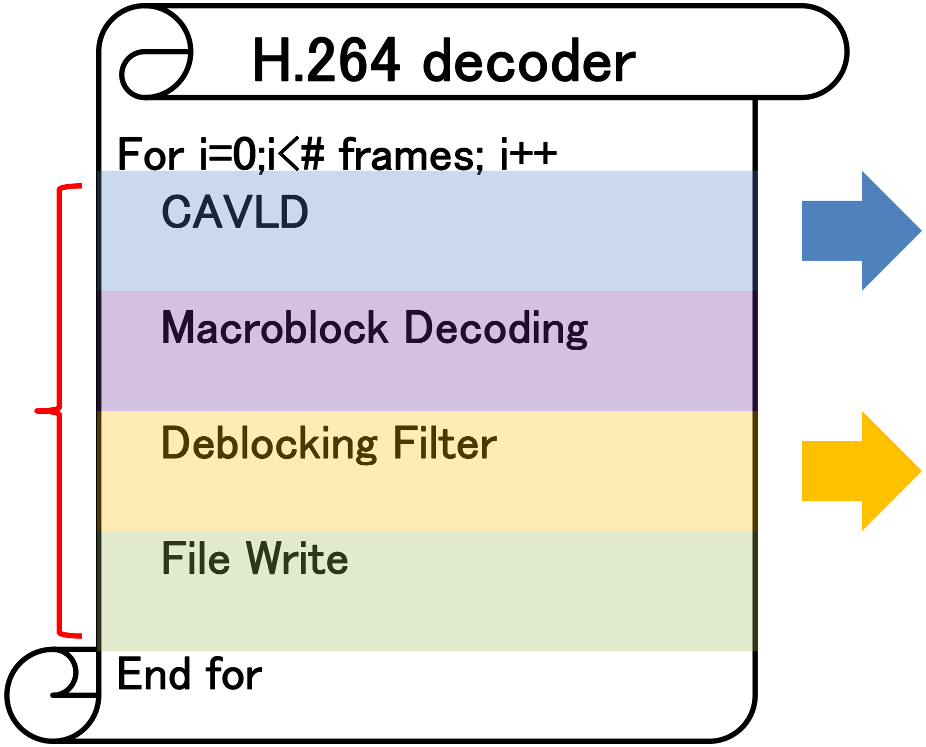
< Processor computation Workload>



< Memory stall time>

Terminology

Target application:
periodic task with real-time constraint



H.264 decoder

For i=0;i<# frames; i++

CAVLD

Macroblock Decoding

Deblocking Filter

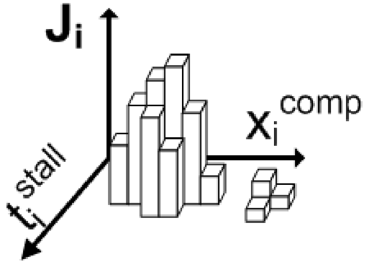
File Write

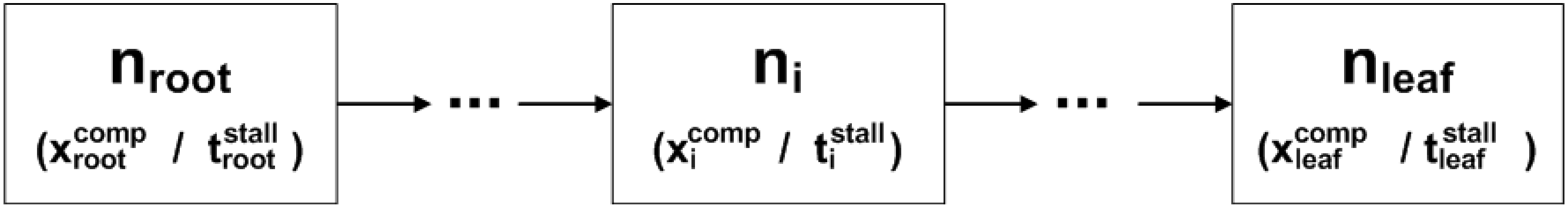
End for

Period
(E.g., 33*ms*
for H.264 dec.)

Program region:
granularity of v/f setting
(few *ms*)

Computation cycle (x_i^{comp})
Memory stall time (t_i^{stall})

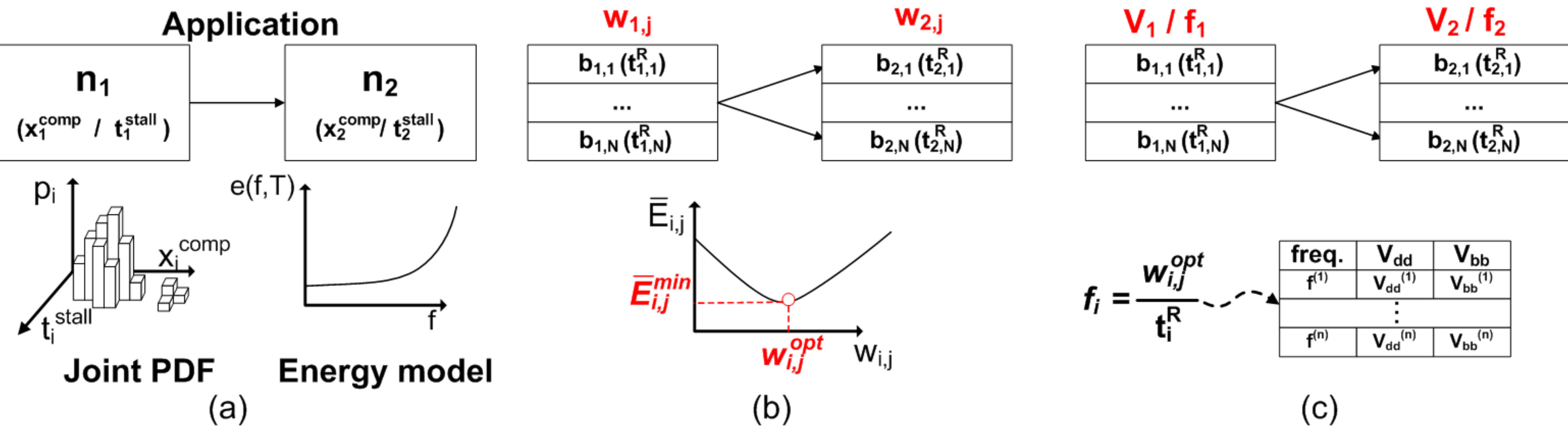




n_{root}
(x_{root}^{comp} / t_{root}^{stall}) → … → n_i
(x_i^{comp} / t_i^{stall}) → … → n_{leaf}
(x_{leaf}^{comp} / t_{leaf}^{stall})

Solution Overview

- Solution consists of design-time and runtime jobs.
  - **Design time job**: finding remaining workload prediction which minimizes average energy consumption
  - **Runtime job**: scaling voltage and frequency with slack reclamation

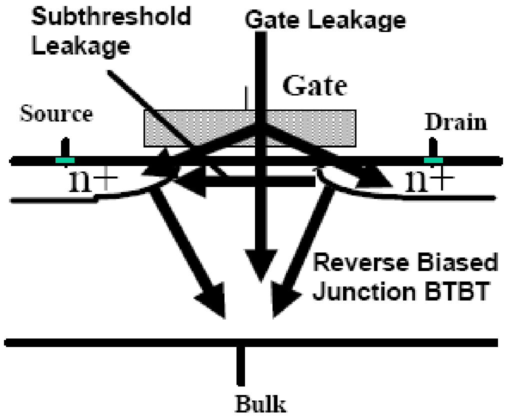


<Inputs> <Design-time solution> <Runtime solution>

Contents

- Introduction
- Motivation
- Related Works
- Preliminary
  - Processor Energy Model
  - Workload Decomposition
- Memory Stall Time-Aware DVFS
- Experimental Results
- Conclusion

Energy Model



< Fig. 21. Leakage source >

- Golden energy model [EM1]
  - Energy consumption per cycle

$$e(f,V_{dd},V_{bb}) = \frac{P}{f} = \underbrace{C_{eff}V_{dd}^{2}}_{\text{Switching energy}} + N_g f^{-1}\left(\underbrace{V_{dd}K_1 e^{K_2V_{dd}} e^{K_3V_{bb}}}_{\text{Subthreshold leakage energy}} + \underbrace{V_{dd}K_4 e^{K_5V_{dd}}}_{\text{Gate leakage energy}} + \underbrace{\left|V_{bb}\right| I_j}_{\text{Junction leakage energy}}\right)$$

where

K_1~K_6 & I_j: **technology process-specific** parameters

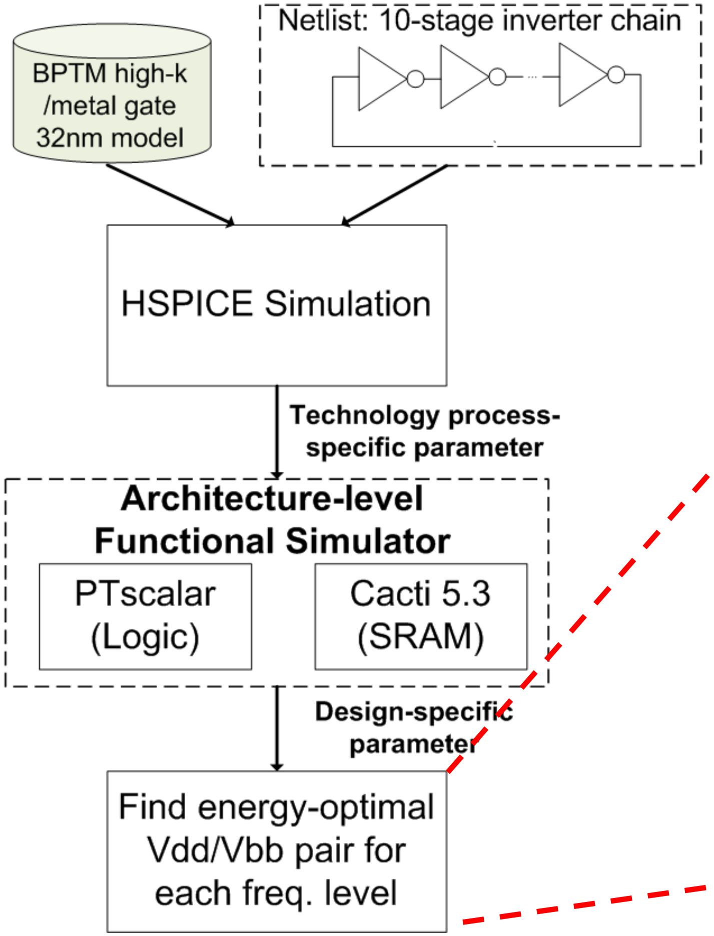
C_{eff} : **design-specific** parameters which models effective capacitance (including average switching activity)

N_g : **design-specific** parameters which models effective gate count

[EM1] S. M. Martin, et al., "Combined dynamic voltage scaling and adaptive body biasing for lower power microprocessors under dynamic workloads," in Proc. ICCAD 2002.

Energy Model

- Characterization flow



BPTM high-k /metal gate 32nm model

Netlist: 10-stage inverter chain

HSPICE Simulation

Technology process-specific parameter

Architecture-level Functional Simulator

PTscalar (Logic)

Cacti 5.3 (SRAM)

Design-specific parameter

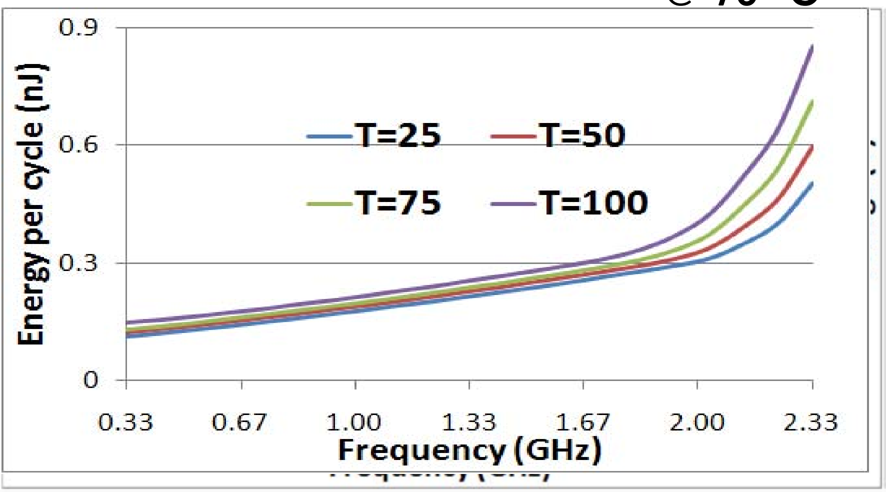
Find energy-optimal Vdd/Vbb pair for each freq. level

< Characterization flow of energy model >

$$0.6\mathrm{V} \le V_{dd} \le 1.0\mathrm{V} \qquad -0.6\mathrm{V} \le V_{bb} \le 0\mathrm{V}$$

$$333MHz \le f \le 2.333GHz$$

@ 75 ℃



< E_{cycle} vs. frequency >

Energy Model

- Approximated energy model [KIM][BYUN]

$$e^{comp} \approx \underbrace{a_s f^{b_s}}_{\text{Switching energy}} + \underbrace{a_l f^{b_l} + c}_{\text{Leakage energy}}$$

$$e^{stall} \approx \underbrace{\beta}_{\text{Clock gating factor}}(a_s f^{b_s} + a_l f^{b_l} + c)$$

where
a_s, b_s, a_l, b_l, c are curve fitting parameters.

Temp	Fitting parameters					Max (Avg.) error (%)
	a_s	b_s	a_l	b_l	c	
25	1.82e-1	1.28	3.83e-9	21.89	0.15	2.97 (1.07)
50	1.82e-1	1.28	1.25e-7	18.16	0.17	1.32 (0.52)
75	1.82e-1	1.28	1.39e-6	15.65	0.19	1.26 (0.42)
100	1.82e-1	1.28	8.72e-6	13.78	0.21	1.95 (0.65)

< Parameters of the approximated energy model >

[KIM] J. Kim, et al., "An analytic dynamic scaling of supply voltage and body bias based on parallelism-aware workload and runtime distribution," in IEEE TCAD, Vol. 28, No. 4, Apr. 2009.

[BYUN] W.-H. Byun, et al., "Processor energy estimation method using cycle-approximate simulator," in ISOCC 2008.

Workload Decomposition

- Total number of execution cycles consists of
  - Processor computation cycle (x^{comp})
  - Memory stall cycles (x^{stall})

$$x(f^{prof}) = x^{comp} + x^{stall} = x^{comp} + f^{prof} \cdot t^{stall}$$

where

x: total # of cycles

x^{comp}: # of computation cycles

x^{stall}: # of memory stall cycles

t^{stall}: amount of memory stall time

f^{prof}: profiling frequency

- Memory stall time

$$t^{stall} = \frac{x(f_{(j)}^{prof}) - x(f_{(k)}^{prof})}{f_{(j)}^{prof} - f_{(k)}^{prof}}$$

where

$f_{(j)}^{prof}$: profiling frequency level at the *j*–th frequency level

Contents

- Introduction
- Motivation
- Related Works
- Preliminary
  - Processor Energy Model
  - Workload Profiling
- Memory Stall Time-Aware DVFS
- Experimental Results
- Conclusion

Calculation of Total Energy Consumption

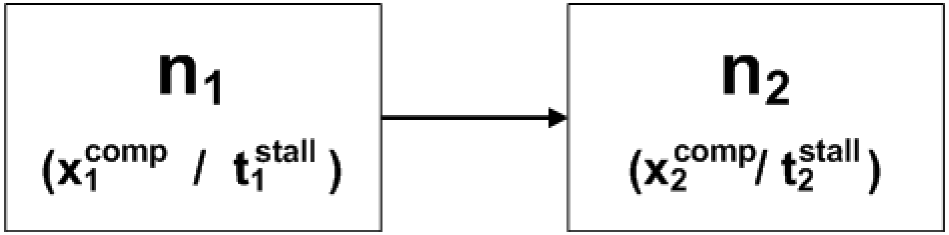
- Total energy consumption (E_i) consists of
  - Processor computation energy consumption (E_i^{comp})
  - Memory stall energy consumption (E_i^{stall})

$$E_1 = E_1^{comp} + E_2^{stall}$$

where

$$E_1^{comp} = \left(e^{comp}\left(f_1\right) \cdot x_1^{comp} + e^{comp}\left(f_2\right) \cdot x_2^{comp}\right)$$

$$E_1^{stall} = \left(e^{stall}\left(f_1\right) \cdot f_1 t_1^{stall} + e^{stall}\left(f_2\right) \cdot f_2 t_2^{stall}\right)$$


n_1
(x_1^{comp} / t_1^{stall})
n_2
(x_2^{comp}/ t_2^{stall})

< Basic case >

Frequency vs. Workload Prediction

- Frequency is set as follows.

Freq. setting of n_i

$$f_1 = \frac{w_1}{t_1^R}$$

where

w_i: computation workload prediction

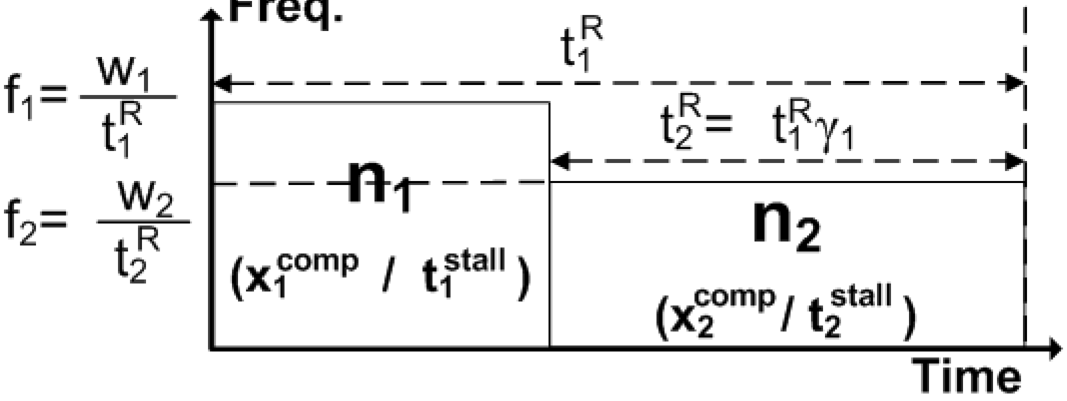
t_i^R: remaining time–to–deadline

Freq. setting of n_{i+1}

$$f_2 = \frac{w_2}{t_2^R} = \frac{w_2}{t_1^R \gamma_1}$$

where

$$\gamma_1 = 1 - \frac{x_1^{comp}}{w_1} - \frac{t_1^{stall}}{t_1^R}$$


Freq.
$f_1 = \frac{w_1}{t_1^R}$
$f_2 = \frac{w_2}{t_2^R}$
t_1^R
$t_2^R = t_1^R \gamma_1$
n_1
$(x_1^{comp} / t_1^{stall})$
n_2
$(x_2^{comp} / t_2^{stall})$
Time

< Remaining time vs. workload prediction >

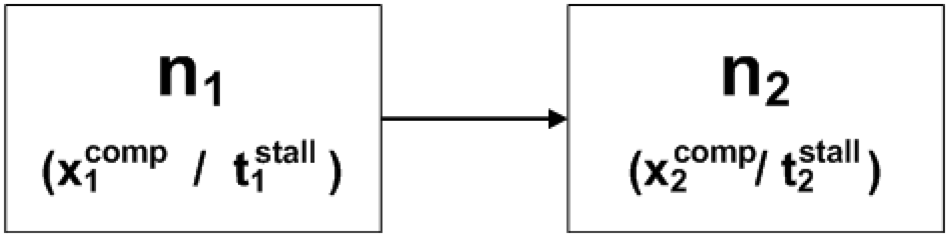
Energy Consumption vs. Workload Prediction

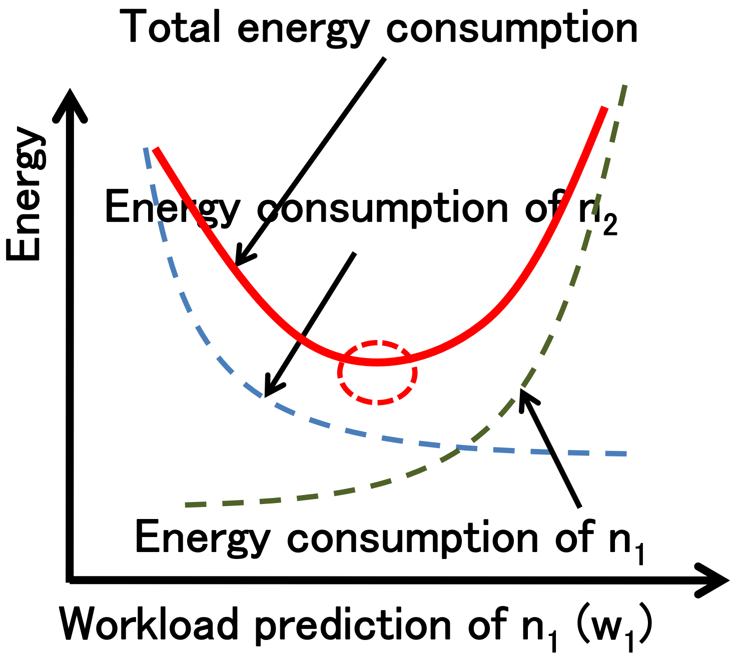
- Energy consumption varies according to workload prediction.
  - Workload prediction of n_1 (w_1) ↑ → frequency of n_1 (f_1) ↑ → remaining time of n_2 (t_2) ↑^{R} →frequency of n_2 (f_2) ↓

$$E_1 = E_1^{comp}(f_1, f_2, t_1^R) + E_1^{stall}(f_1, f_2, t_1^R)$$

$$f_1 = \frac{w_1}{t_1^R} \qquad f_2 = \frac{w_2}{t_2^R} = \frac{w_2}{t_1^R \gamma_1}$$

$$E_1 = E_1^{comp}\left(w_1, w_{2,}, t_1^R\right) + E_1^{stall}\left(w_1, w_2, t_1^R\right)$$

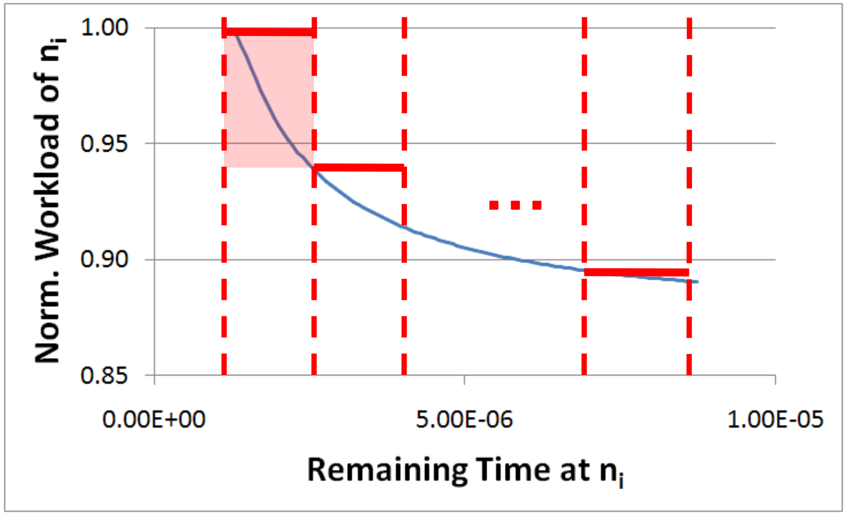

n_1
(x_1^{comp} / t_1^{stall})
n_2
(x_2^{comp}/ t_2^{stall})

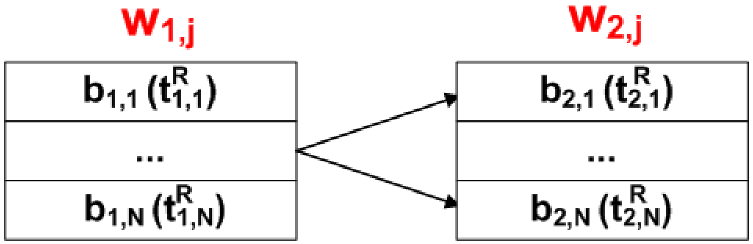


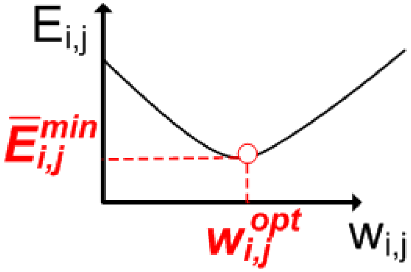
Workload Prediction

- Energy-optimal workload prediction varies according to remaining time.
  - Quantize probable remaining time into *N* levels (each level is called *bin*)
  - Workload prediction at each bin

➔ Quantization step size ↓ → Energy savings ↑ (due to more accurate workload prediction)



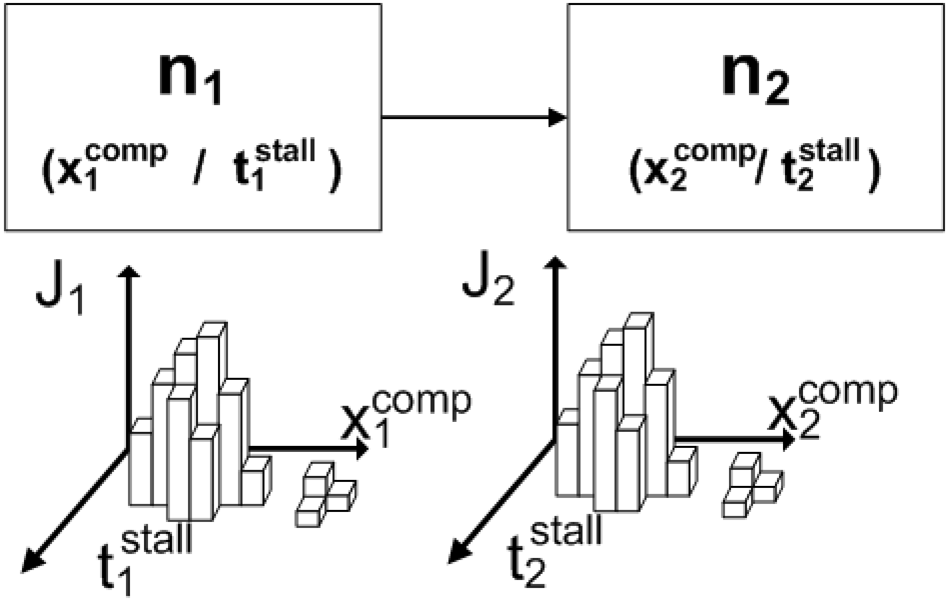




Calculation of Average Energy Consumption

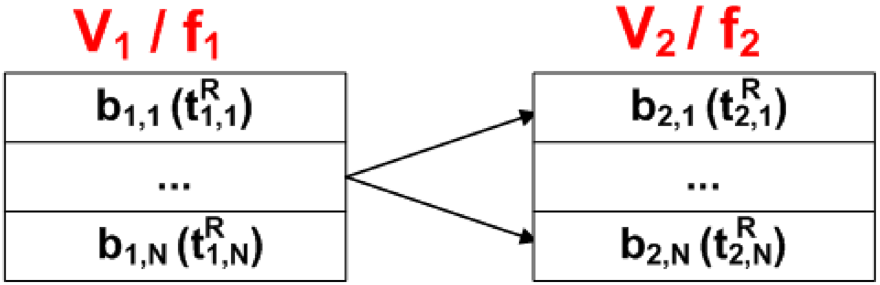
- Average energy consumption is calculated by integrating E_i w.r.t. joint PDF.

$$\overline{E_1}\left(w_1, w_2, t_1^R\right) = \int \cdots \int \left(E_1^{comp} + E_1^{stall}\right) J_1 J_2 \, dx_1^{comp} \, dt_1^{stall} \, dx_2^{comp} \, dt_2^{stall}$$



Runtime Step

- Runtime step
  1. Find workload prediction by measured remaining time
  2. Adjust frequency while satisfying the real–time constraint
  3. Adjust (V_{dd}, V_{bb}) by accessing a table which stores energy–optimal pairs of (V_{dd}, V_{bb}) for frequency levels



$$f_i = \frac{w_{i,j}^{opt}}{t_i^R}$$

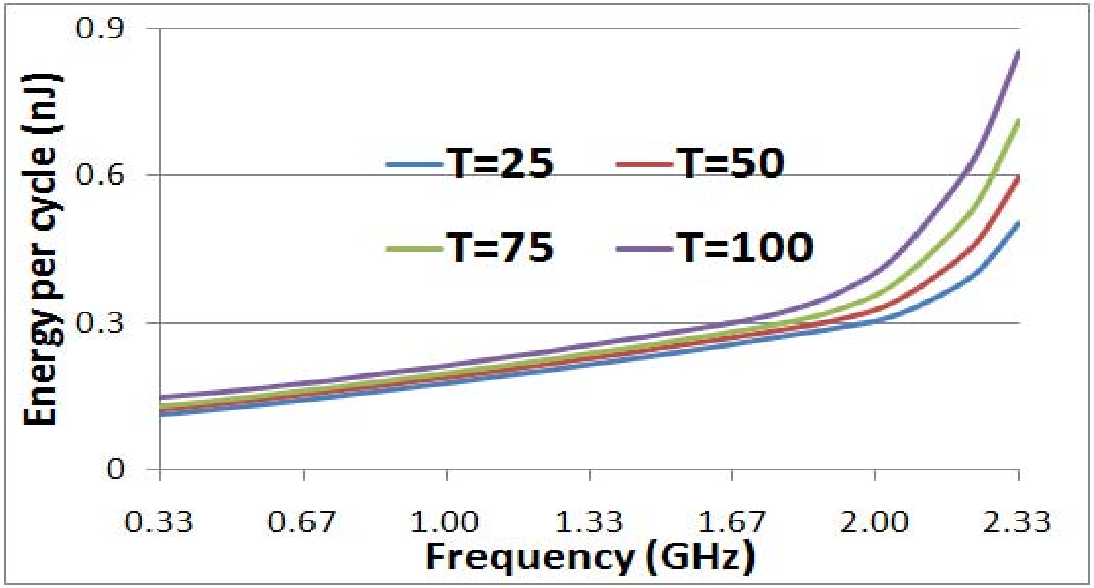
freq.	V_{dd}	V_{bb}
$f^{(1)}$	$V_{dd}^{(1)}$	$V_{bb}^{(1)}$
	⋮	
$f^{(n)}$	$V_{dd}^{(n)}$	$V_{bb}^{(n)}$

Contents

- Introduction
- Motivation
- Related Works
- Preliminary
  - Processor Energy Model
  - Workload Profiling
- Memory Stall Time-Aware DVFS
- Experimental Results
- Conclusion

Experimental Setup

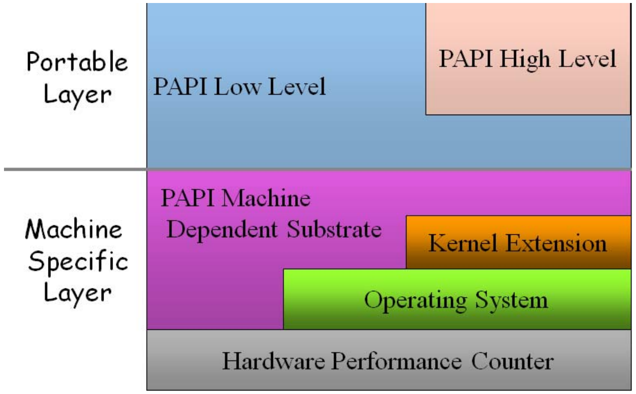
- Frequency
  - 333MHz ~ 2.333GHz, 7 levels (333MHz step)
  - Transition overhead: 40 μ s
- Voltage
  - Supply voltage (V_{dd}): 0.6V ~ 1.0V
  - Body bias voltage (V_{bb}): −0.6V ~ 0V
  - Transition overhead: 100ns (assuming that on-chip regulator)



< Energy model>

Experimental Setup

- Target application
  - MPEG4 and H.264 decoder in FFMPEG
  - 2000 frames of 1920x800 movie clip excerpted from *Harry Potter*
    - Training sample: first 1000 frames
    - Evaluation sample: the next 1000 frames
- Workload profiling environment
  - LG XNOTE LW25 laptop with Linux 2.6.3
  - PAPI: API for accessing hardware counter



–**Processor**: 2GHz INTEL Core2Duo T7200
–**Cache**: 128KB L1 I/D$ and 4MB L2$
–**Memory**: 667MHz 2GB DDR2

< Profiling environment>

Experimental Setup

- Comparison

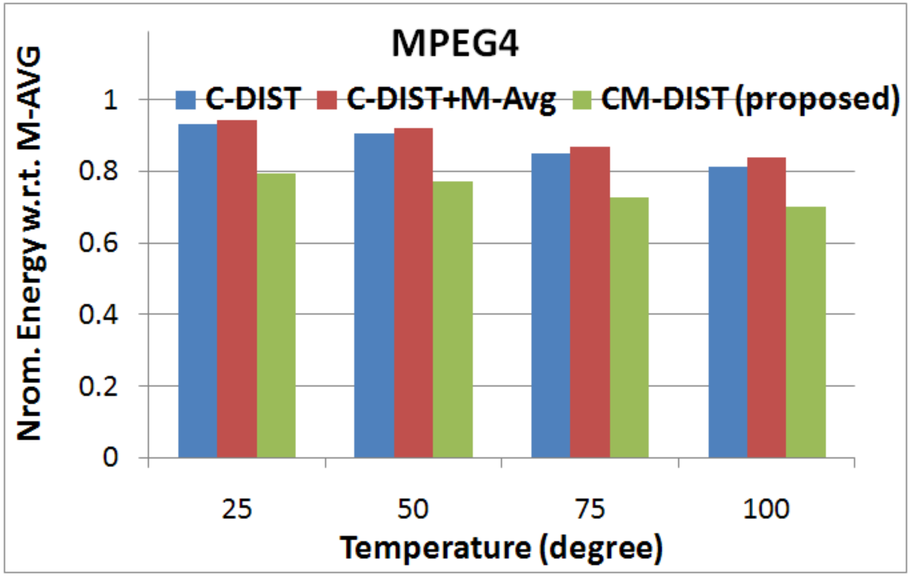
	M-AVG [6]	C-DIST [3]	C-DIST + M-AVG	CM-DIST (proposed)
Memory awareness	O	X	O	O
Workload distribution	X	Comp. only	Comp. only	Both

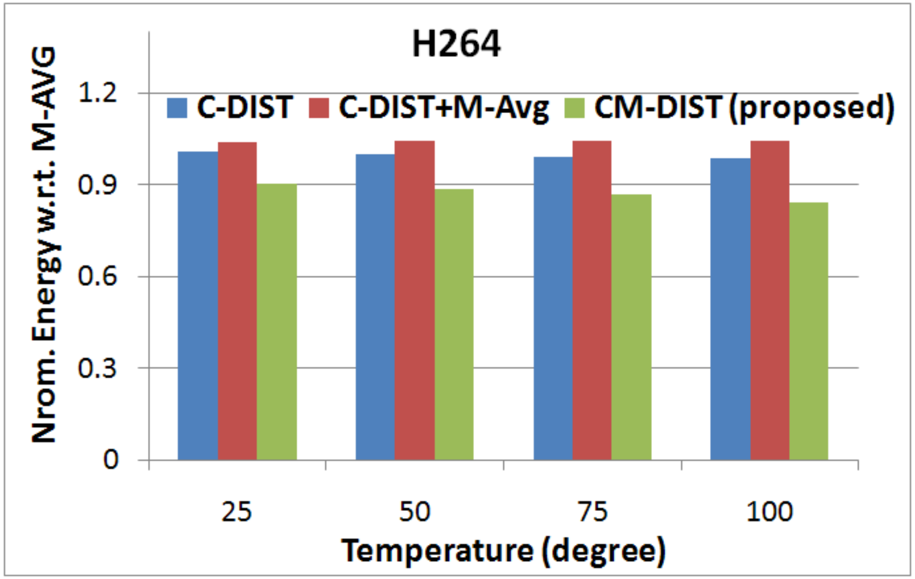
Experimental Results

- vs. M-AVG
  - MPEG4: 20.6% ~ 30.0% energy savings
  - H.264: 9.6% ~ 15.8% energy savings

→Due to considering workload distributions

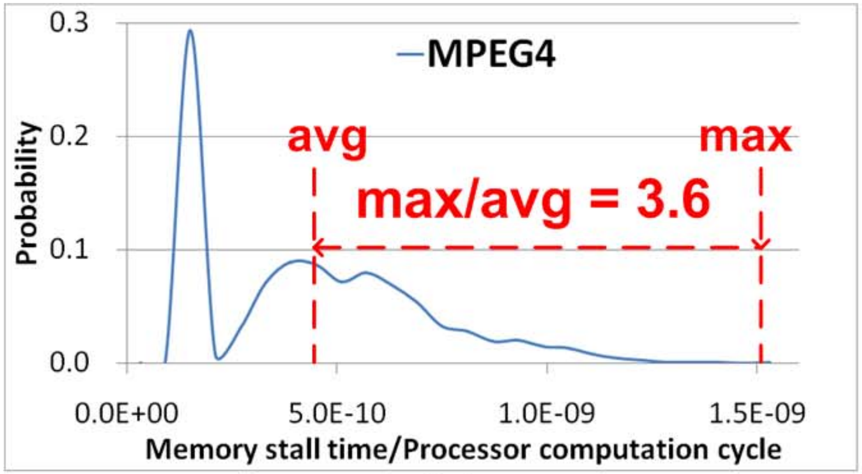
→More energy savings as temperature increases



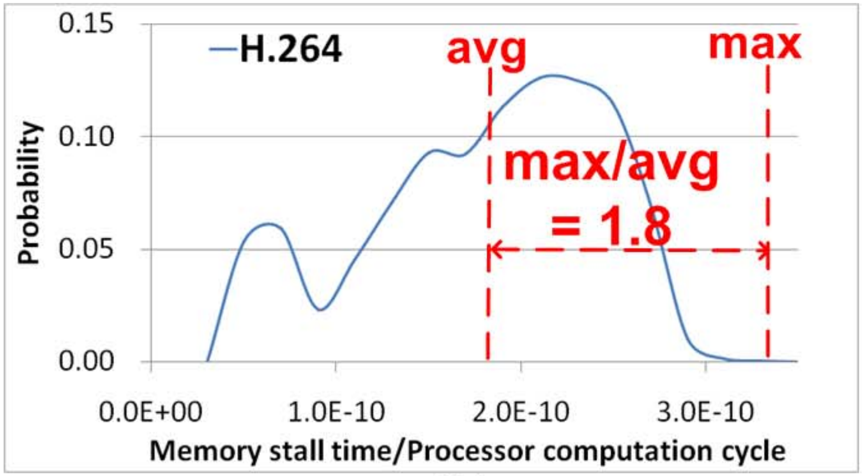


Experimental Results

- MPEG4 vs. H.264
  - More energy savings can be obtained in MPEG4 due to larger workload distribution.



(a)

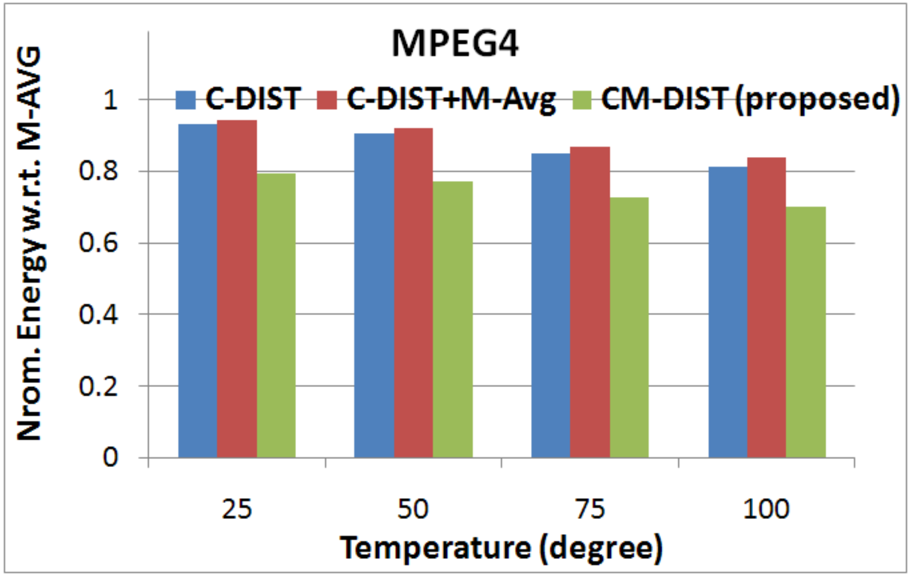


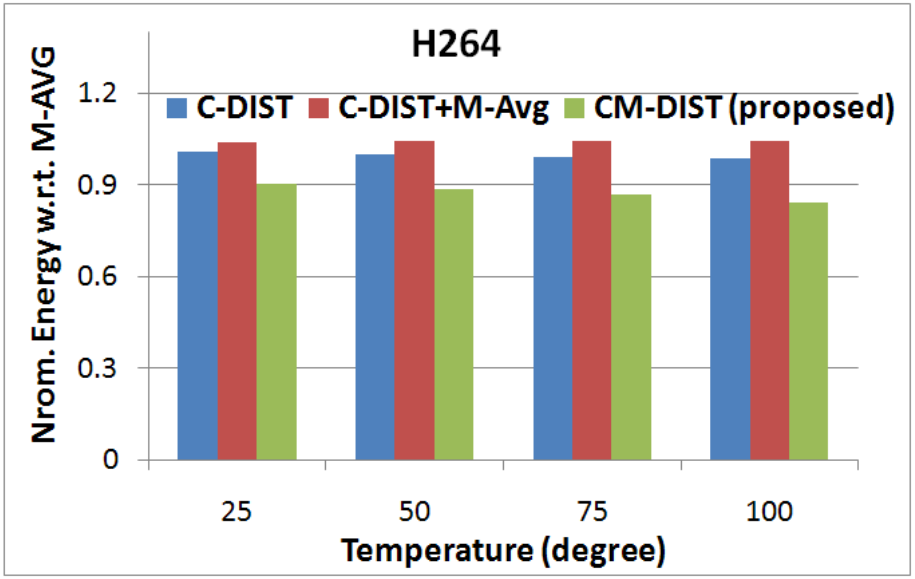
(b)

Experimental Results

- vs. C–DIST
  - MPEG4: 14.1% ~ 14.7% energy savings
  - H.264: 10.1% ~ 14.4% energy savings

➔ Due to considering memory stall workload with its distribution

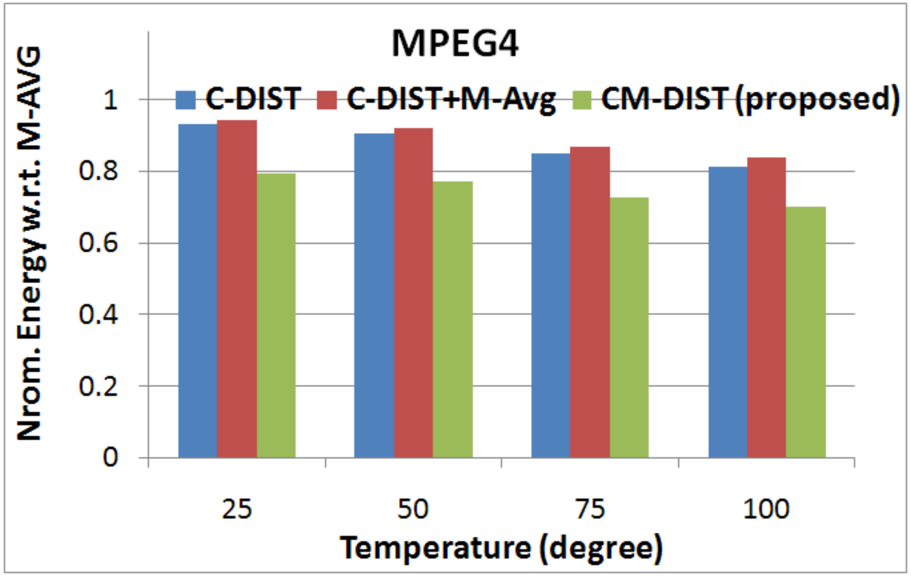


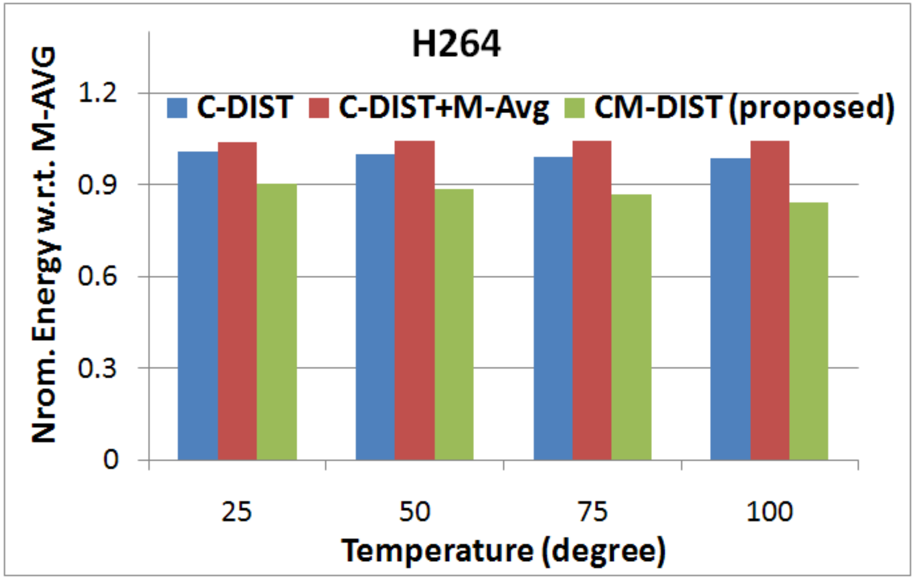


Experimental Results

- vs. C–DIST+M–AVG
  - MPEG4: 15.9% ~ 16.4% energy savings
  - H.264: 13.1% ~ 19.2% energy savings

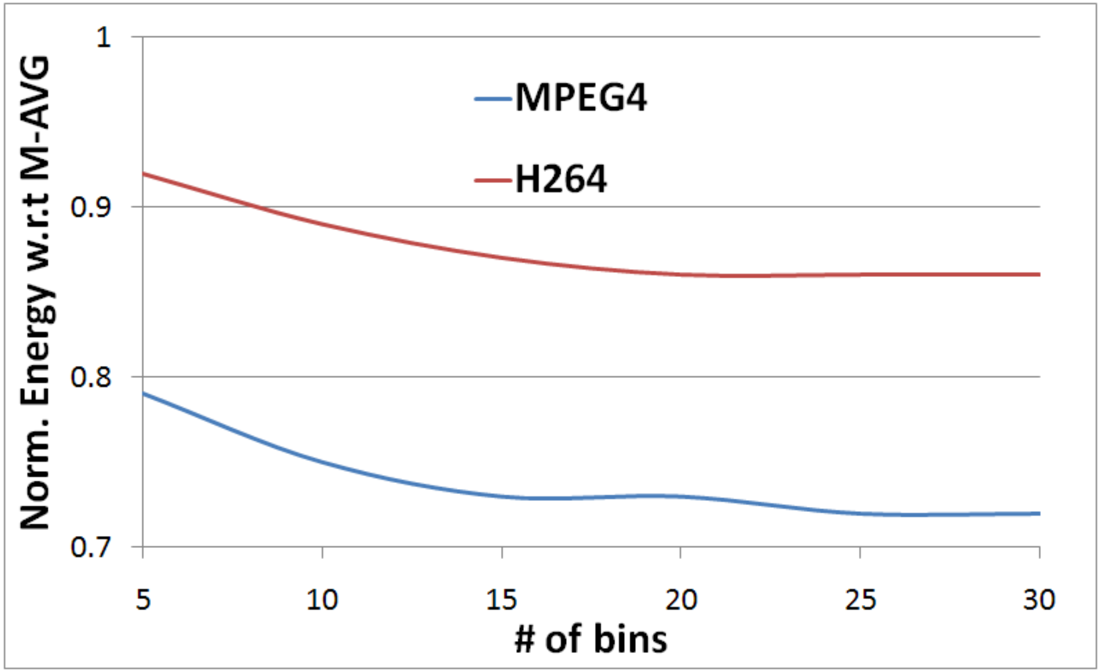
  ➔ Not considering the distribution of memory stall time and the correlation with processor computation workload





Experimental Results

- Energy savings w.r.t # of bins
  - The more the number of bins, the more energy savings can be obtained.
  - Improvement becomes saturated above 25 bins.



Conclusion

- We presented a novel DVFS method exploiting distributions of both
  - Processor computation workload
  - Memory stall time

- We presented a numerical solution which finds workload prediction which gives minimum average energy consumption considering the dependency of remaining time in energy-optimal workload prediction.

- Experimental results shows that the proposed method offers 9.6% ~ 30.0% further energy savings compared with existing methods.

Thank You!!!