Frame-Based Dynamic Voltage and Frequency Scaling for an MPEG Decoder

Kihwan Choi, Karthik Dantu,
Wei-Chung Cheng, Massoud Pedram
Dept. of EE-Systems
University of Southern California

Outline

- Dynamic Voltage and Frequency Scaling (DVFS)
- Challenges of DVFS for MPEG decoding
- The Proposed DVFS Policy
- Experimental Results
- Conclusions

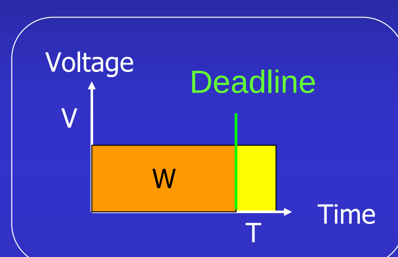
Dynamic Voltage & Frequency Scaling

- DVFS is an effective way of reducing the CPU energy consumption by providing “just-enough” computation power $E = \alpha * C_{eff} * V^2 * f * T$
- Example of a real-time application:

If $V_2 = V_1/2$ and $f_2 = f_1/2$, then $E_1 / E_2 = 4$

Prerequisites for DVFS

- A fixed deadline
  - Usually given in real-time operation
- A method for accurate workload prediction
- A method for voltage and frequency setting
- An error compensation method for workload prediction



Interval-based Workload Prediction

- (Simple) Moving Average

$$\text{WindowSize} = n$$

$$\text{Workload}(t+1) = \frac{\sum_{\tau=0}^{n-1} \text{Workload}(t-\tau)}{n}, \quad t \geq n-1$$

$$= \frac{\sum_{\tau=0}^{t} \text{Workload}(t-\tau)}{n}, \quad \text{otherwise}$$

- (Exponentially) Weighted Average

$$\text{Workload}_{avg}(0) \equiv \text{Workload}(0)$$

$$\text{Workload}(t+1) = \alpha \cdot \text{Workload}(t) + (1-\alpha) \cdot \text{Workload}_{avg}(t)$$

$$= \alpha \cdot \sum_{\tau=0}^{t} (1-\alpha)^{\tau} \cdot \text{Workload}(t-\tau)$$

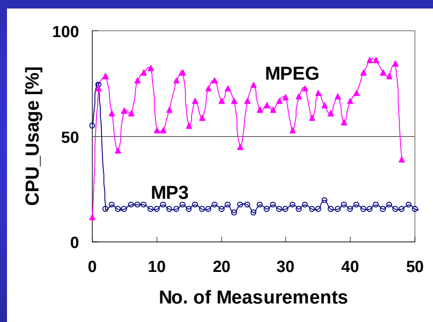
Simple Attempt at DVFS for MP3 & MPEG

- Use “Moving Average” as the prediction method
- MP3
  - Sizeable energy saving with little sound quality degradation
- MPEG
  - Some energy saving but at the cost of a significant video quality degradation

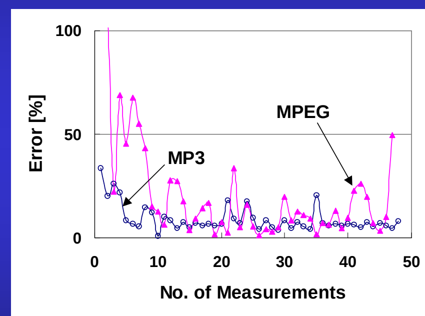
Difference between MP3 & MPEG

- Workload distributions of MP3 and MPEG

workload distribution



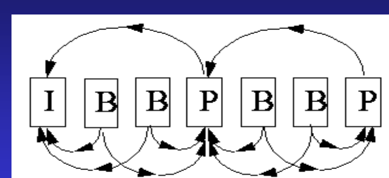
prediction error



- MPEG workload is non-uniform and time-varying, hence its prediction is much more error-prone

MPEG Coding: 101

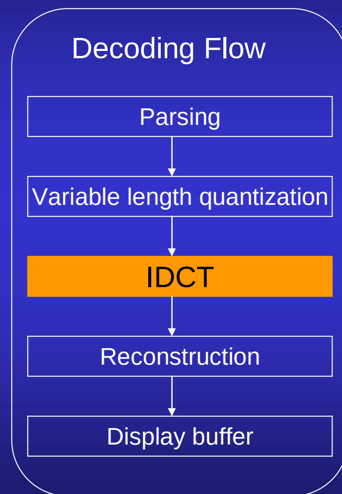
- There are three types of frames: I, P, and B frame



- I (intra) frame has no reference frame
  - all pixels of the frame should be encoded
- P (predictive) frame has a reference frame
  - only parts that are different from the reference frame must be encoded
- B (bi-directional) frame has two reference frames
  - results in the least number of encoded bits

MPEG Decoding Flow

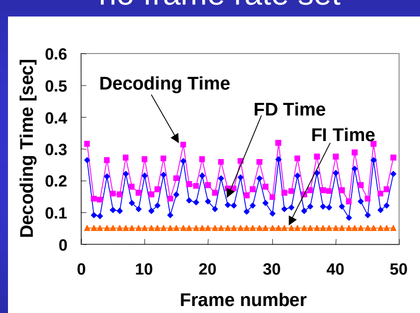
- IDCT (Inverse Discrete Cosine Transform) is the most CPU intensive step
- IDCT time of each frame
  I-frame > P-frame > B-frame
- Wide variations in workload are due to different IDCT times for different frames
- A MPEG customized workload prediction method is needed



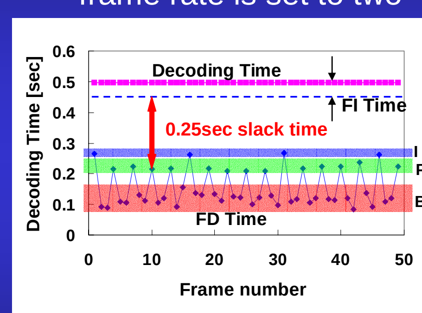
Characteristics of the MPEG Decoding

- Two experiments:

no frame rate set



frame rate is set to two



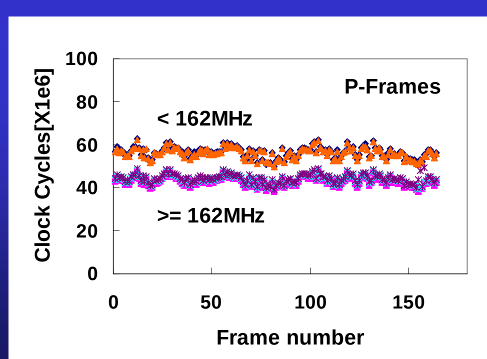
- Decoding time consists of two phases:
  a frame-dependent (FD) part and a frame-independent (FI) part
- Frames of the same type impose the same workload

roposed DVFS Policies for MPEG Decoding

- Frame-based workload prediction method
  - divide the decoding sequence into two parts: FD and FI
  - separately maintain workload statistics for each frame type
  - Use MA or WA to estimate workload of each frame type
- Voltage and frequency setting method
  - set frequency (and subsequently the voltage level) to finish the frame in the allocated time
- Error compensation method
  - use the FI part as a buffer to correct errors that may occur in the FD workload prediction

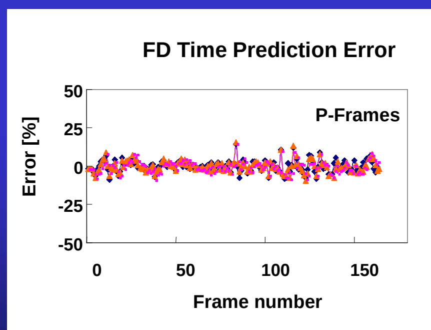
Non-linearity in Memory Performance

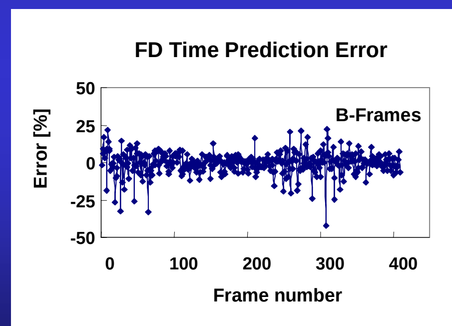
- Workload = Execution_Time * CPU_Freq.
- There is a discontinuity in the calculated workloads due to the non-linear memory performance as the CPU frequency is changed across the 162 MHz



Frame-based Workload Prediction Errors

- ❑ Average prediction errors
  - I : 5% error ; P : 3% error ; B = 10% error

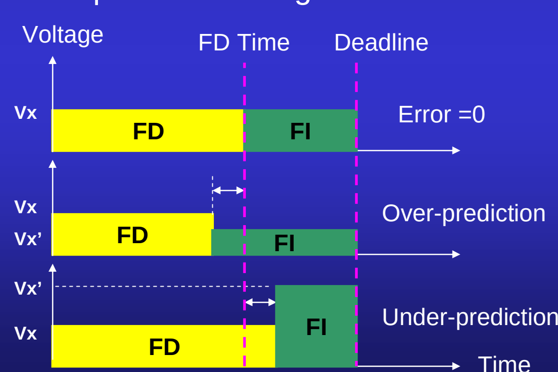




ICCAD 2002

Error Compensation Method in Action

- ❑ If a prediction error occurs for the FD part, then it can be compensated for by changing the CPU frequency during the FI part
  - we use the FI part as a timing window buffer



ICCAD 2002

Maximum Tolerable Error

- The maximum error that can be tolerated

$$CurrFreq = \frac{FD_{workload} \cdot (1 - Error) + FI_{workload}}{Deadline}$$

$$Deadline \geq \frac{FD_{workload}}{CurrFreq} + \frac{FI_{workload}}{MaxFreq}$$

- Assume that any CPU frequency can be used

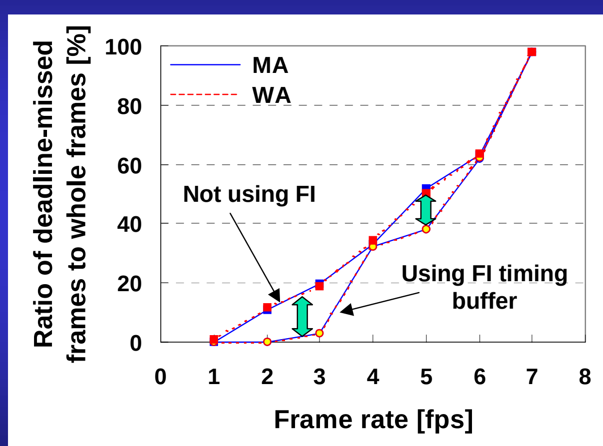
Workload data

	I	P	B
$FD_{workload}$	65e6	55e6	30e6
$FI_{workload}$	13e6	13e6	13e6
Avg.Error	5%	3%	10%

Max Tolerable Error

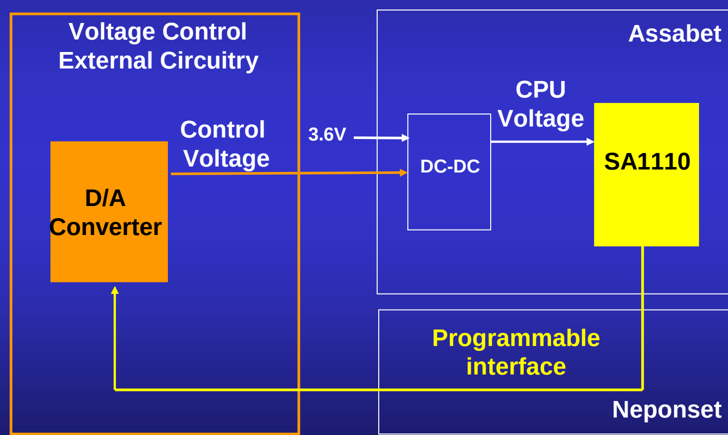
fps	I	P	B
1	14%	18%	37%
2	8%	12%	30%
3	1%	5%	24%
4	-	-	16%
5	-	-	8%
6	-	-	-

Effectiveness of the Proposed Policy



Design Environment for DVFS

- ❑ Block diagram of the DVFS circuitry

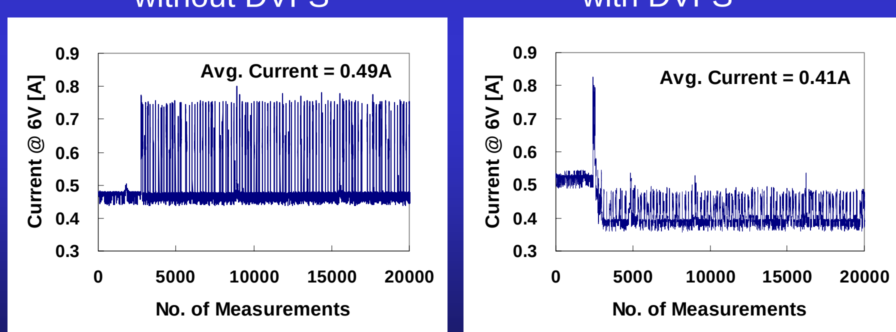


Apollo Testbed I

Experimental results

- ❑ Actual power consumption measurements
- ❑ Frame rate is set to one
- ❑ 16% total system energy reduction

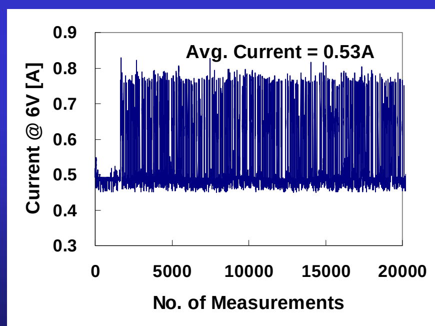
without DVFS | with DVFS



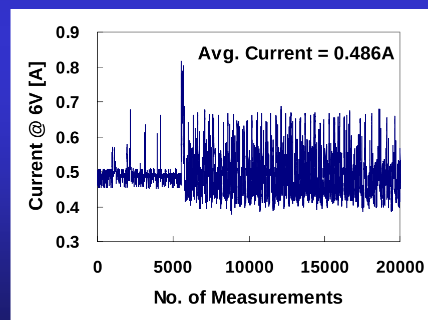
Experimental results (con't)

- ❑ Frame rate is set to two
- ❑ 8% of the total system energy is saved

without DVFS



with DVFS



ICCAD 2002

Summary

- ❑ A frame-based DVFS policy for low power MPEG decoding was proposed
  - workload prediction is performed on each frame type
  - compensation for the prediction error can be done with no or little performance degradation
- ❑ By applying this DVFS scheme on our test bed, 16% and 8% of whole energy consumption can be saved with fps=1 and fps=2, respectively
- ❑ Future work will address voltage scaling for streaming video server-client system

ICCAD 2002