Dynamic Thermal Management for MPEG-2 Decoding

Wonbok Lee, Kimish Patel, Massoud Pedram

University of Southern California
Los Angeles CA

October 5th 2006

Outline

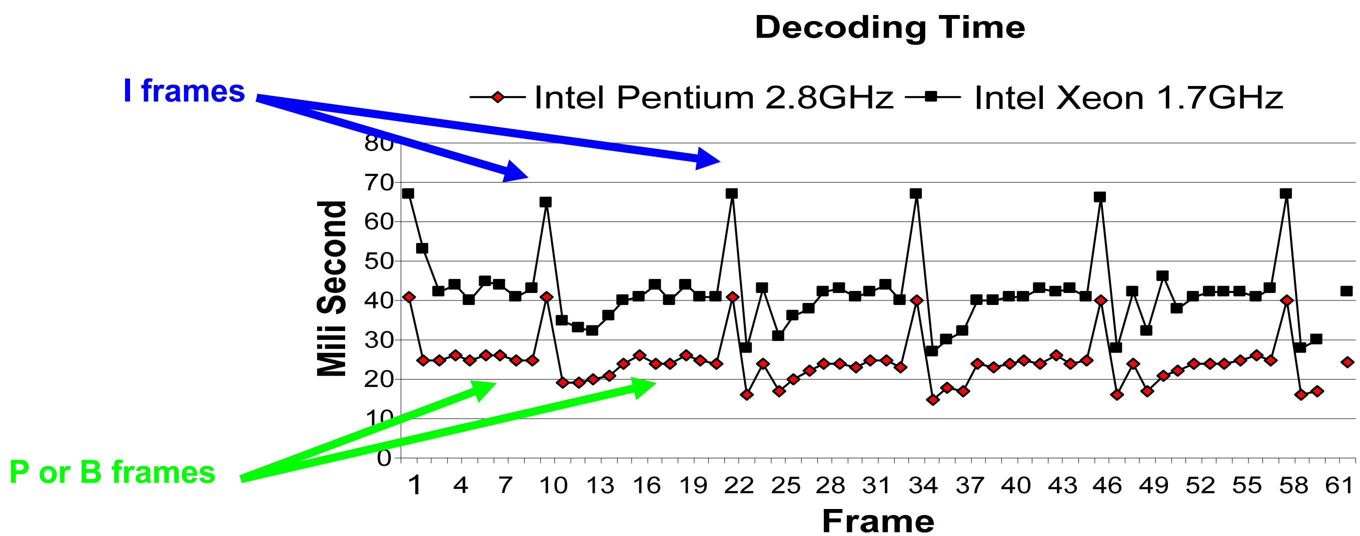
- Background
- Proposed Dynamic Thermal Management (DTM) Technique
- Spatial/Temporal Quality Degradation in MPEG-2
- Simulation Environment and Implementation
- Experimental Results
- Conclusion

Background

- Dynamic Thermal Management (DTM)
  - Do not design for the worst-case chip temperature; manage worst-case conditions by employing DTM
  - DTM aims to achieve a thermally safe state of a microprocessor at the expense of minimal performance degradation
- Two Thermal Thresholds :
  - **Trigger temperature:** Temperature above which DTM initiates
  - **Emergency temperature:** Temperature above which microprocessor starts to experience logical/timing errors
- Examples of previous DTM techniques
  - Fetch Toggling
  - Instruction Cache Throttling
  - Dynamic Instruction Window Resizing
  - Switching Off Active Functional Units
  - Deactivating Appropriate Register Ports
  - Activity Migration
  - Dynamic Voltage & Frequency Scaling

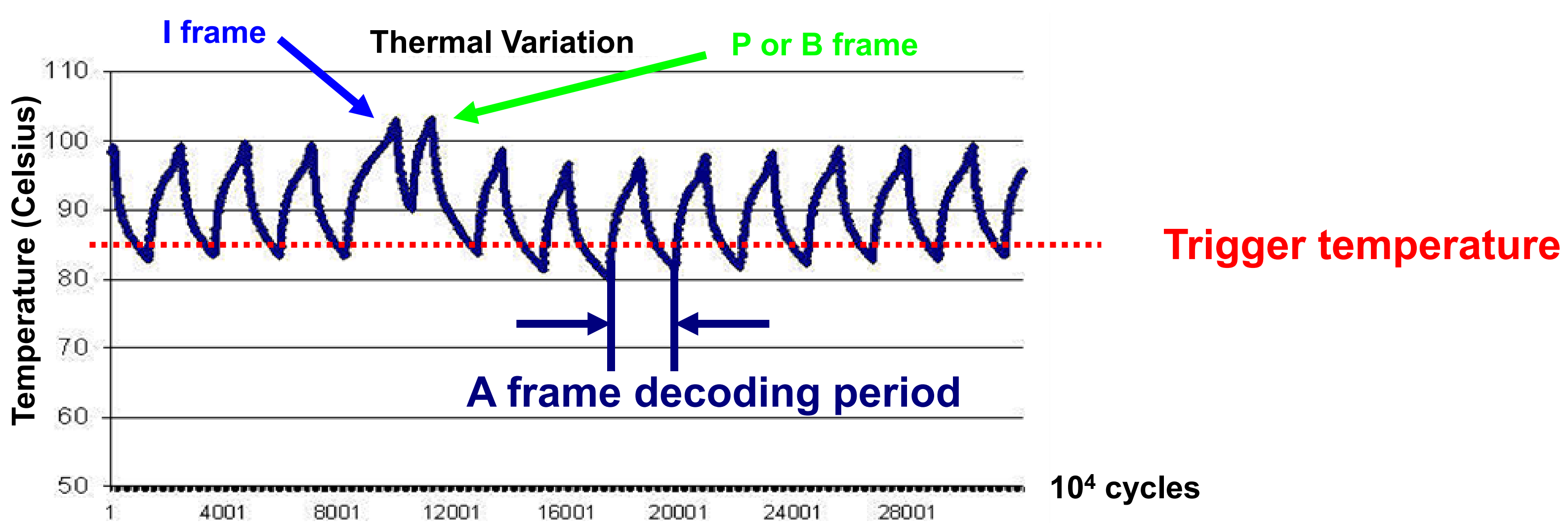
Decoding Time

- As microprocessors become faster, the absolute time needed to decode each MPEG frame becomes smaller
  - The frame rate is fixed: 29.97fr/sec(NTSC), i.e., 33msec per frame
    - Total frame count = 60, image resolution=704X480
  - We compare MPEG decoding times (w/o dithering) for two cases:
    - Decoding Speed: 42.01msec/frame vs. 24.01msec/frame
- Can we utilize the residual time (frame decoding deadline – actual frame decoding time) to make the system thermally safe?



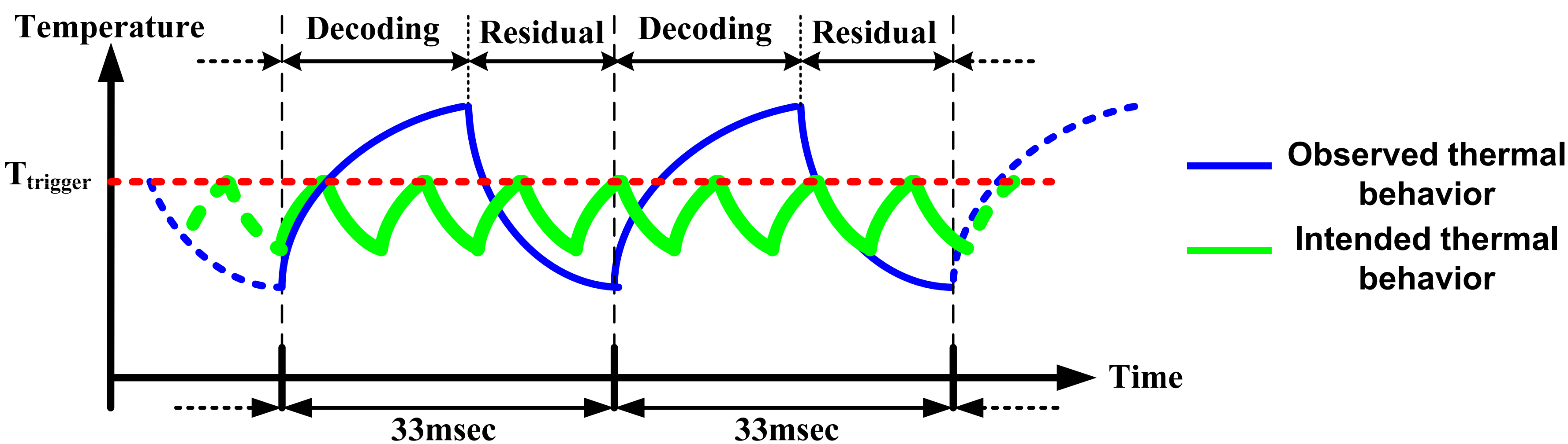
Temperature Violation w/o DTM

- Simulation Setup
  - Simplescalar + Wattch + Hotspot
  - Assume Alpha 21364 processor floor-plan
  - Set the trigger temperature = 82℃
- Once a program behavior settles down, temperature variance is captured in 10K cycle granularity



How DTM Works

- How to cope with this thermal crisis?
  - Each time we reach the trigger threshold, we stall the processor to cool off
    - Ideally, a frame decoding will finish within its target deadline
    - If not, we may end up with some spatial/temporal quality degradation
  - Bottom line: Distribute decoding workload such that chip temperatures never exceed the threshold temperature



Thermal Model and Gradients

- We adopt the thermal model used in Skadron, et al. (HPCA 2002)

$$\Delta T = \left(\frac{P}{C_{th}} - \frac{T_{old}}{R_{th} \cdot C_{th}}\right) \cdot \Delta t$$

ΔT : Temperature variation
P: Average power in an interval
R_{th}: Thermal resistance
C_{th}: Thermal capacitance
T_{old}: Initial temperature
Δt : A time interval

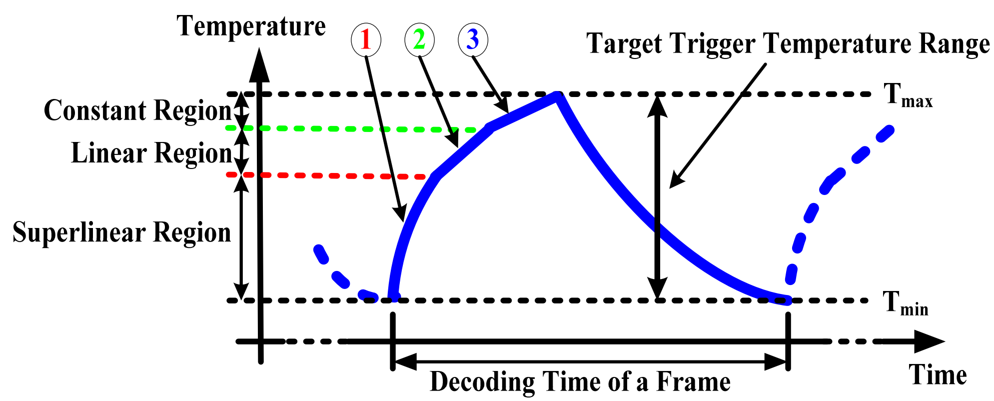
- Important observations:
  - During the period of decoding
    - Rising thermal gradient is calculated as:
  - During the period of resting (stall)
    - Falling thermal gradient is calculated as:
  - Leakage power is not considered in our simulations

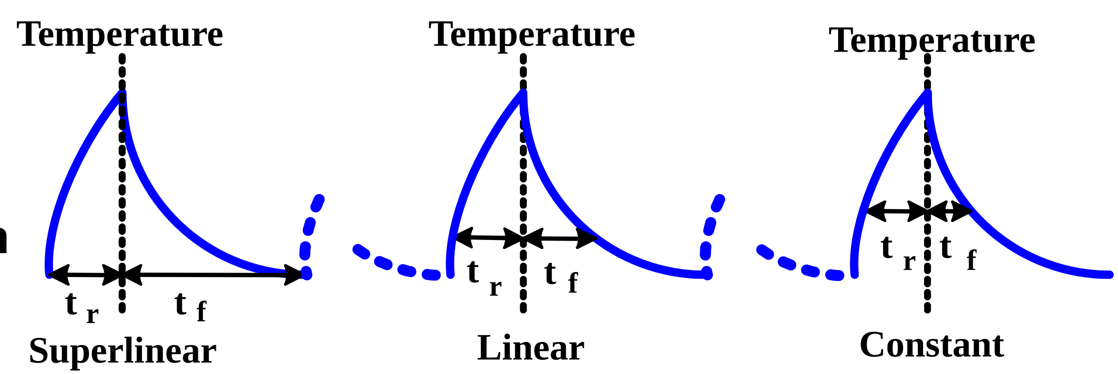
$$\text{Rising: } \frac{\Delta T_r}{\Delta t} = \left(\frac{P}{C_{th}} - \frac{T_{old}}{R_{th} C_{th}}\right)$$

$$\text{Falling: } \frac{\Delta T_f}{\Delta t} = \left(-\frac{T_{old}}{R_{th} C_{th}}\right)$$

A Program's Thermal Behavior and the Trigger Temperature

- Classify a program's thermal behavior into three regions:
  - Superlinear (cool off much slower than heat up): $\Delta T_r/\Delta T_f$ is much larger than 1
  - Linear: $\Delta T_r/\Delta T_f$ is nearly one
  - Constant (cool off much faster than heat up): $\Delta T_r/\Delta T_f$ is much less than 1
- T_{min} and T_{max} are circuit, floorplan and input file-dependent
- Trigger temperature (which is package, heat sink and architecture dependent) can end up lying in any of these regions





For same ΔT, $t_r << t_f$ $t_r = t_f$ $t_r >> t_f$

Key Concepts Behind the Proposed DTM Policy

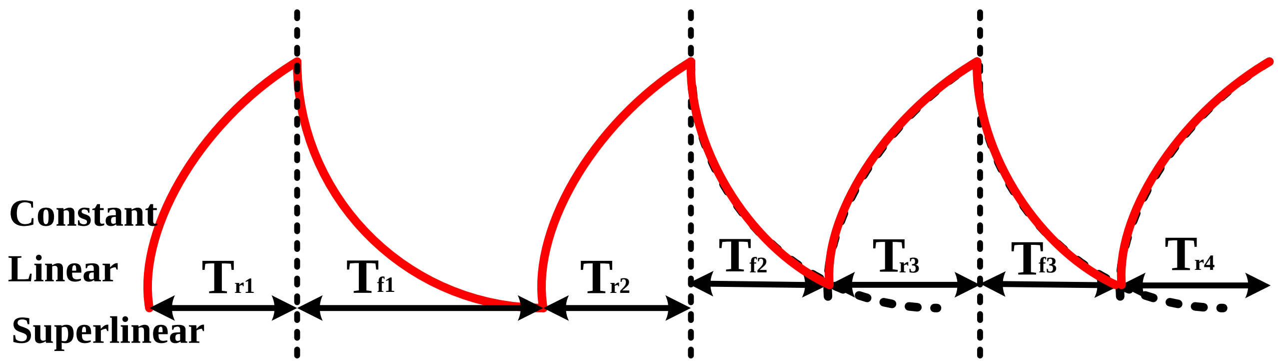
- **Run MPEG stream without any DTM policy to obtain T_{max} and T_{min}**
- **If $T_{trigger}$ > T_{max}, the chip is thermally safe w/o any effort**
- **If $T_{trigger}$ < T_{min}, significant quality degradation should be accrued to achieve thermal safety**
- **If T_{min} < $T_{trigger}$ < T_{max}, check the level of $T_{trigger}$. If it lies in**
  - **Constant region: thermally safe w/ little or no quality degradation**
  - **Linear region: thermally safe at the cost of some quality degradation**
  - **Super-linear region: thermally safe at the cost of sizeable quality degradation**

The Proposed DTM Policy

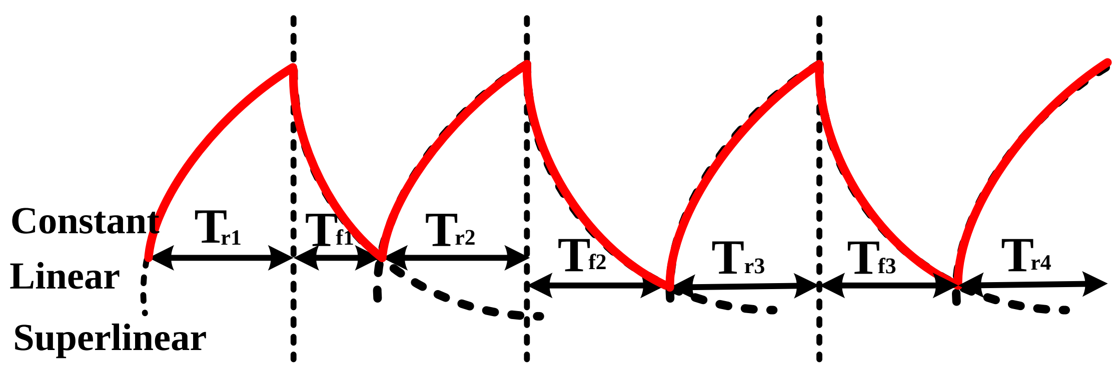
- Stall the processor for the length of time for as long as the falling temperature is comparable to the rising temperature
  - Every time we reach $T_{trigger}$, we initially stall the processor for 1M cycles
- We may miss a frame decoding deadline (which means that either some level of spatial or temporal quality degradation will be necessary)
- We predict the frame decoding time by online linear regression
- If a deadline miss is predicted, we do spatial quality degradation during the frame decoding
  - If the deadline is in fact missed, we do temporal quality degradation (drop the next P or B frame)
  - Otherwise, we accrue the positive slack time for future use
- From our experimental setup, we have found that $T_{trigger}$ mostly lies in the linear region

Adaptive Stall Periods

- Dynamically determine the stall period that creates equal rising & falling thermal changes
  - We start with some stall period (T_{f1}) and adapt the stall period on the next DTM cycles



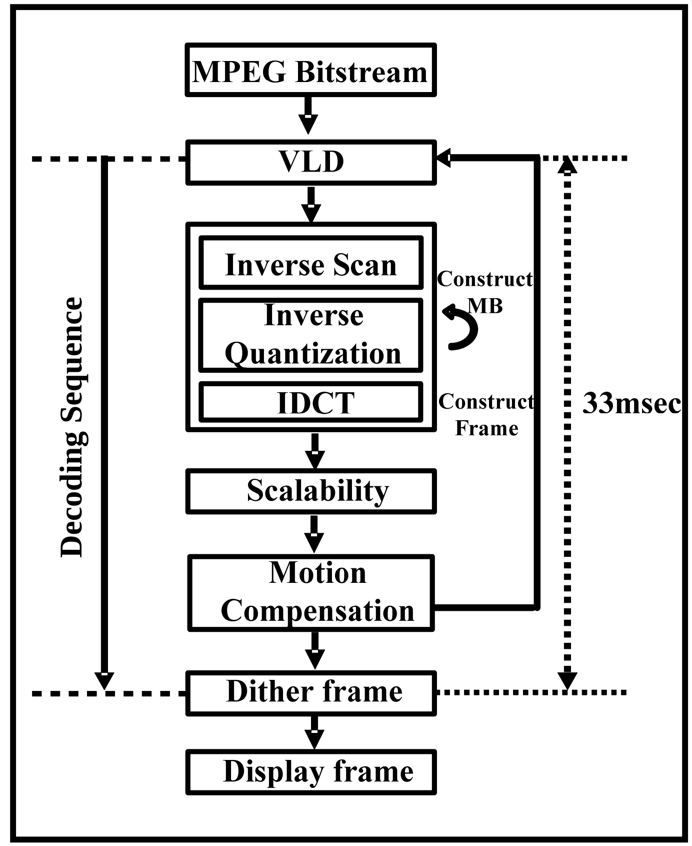
First cycle states that we are in the super-linear region
- Stall period is decreased over time
  - $T_{f1} > T_{f2} = T_{r3}$
  - $T_{r3} = T_{f3} = T_{r4} = \ldots$



First cycle states that we are in the constant region
- Stall period is increased over time
  - $T_{f1} < T_{f2}$
  - $T_{f2} = T_{r3} = T_{f3} = T_{r4} = \ldots$

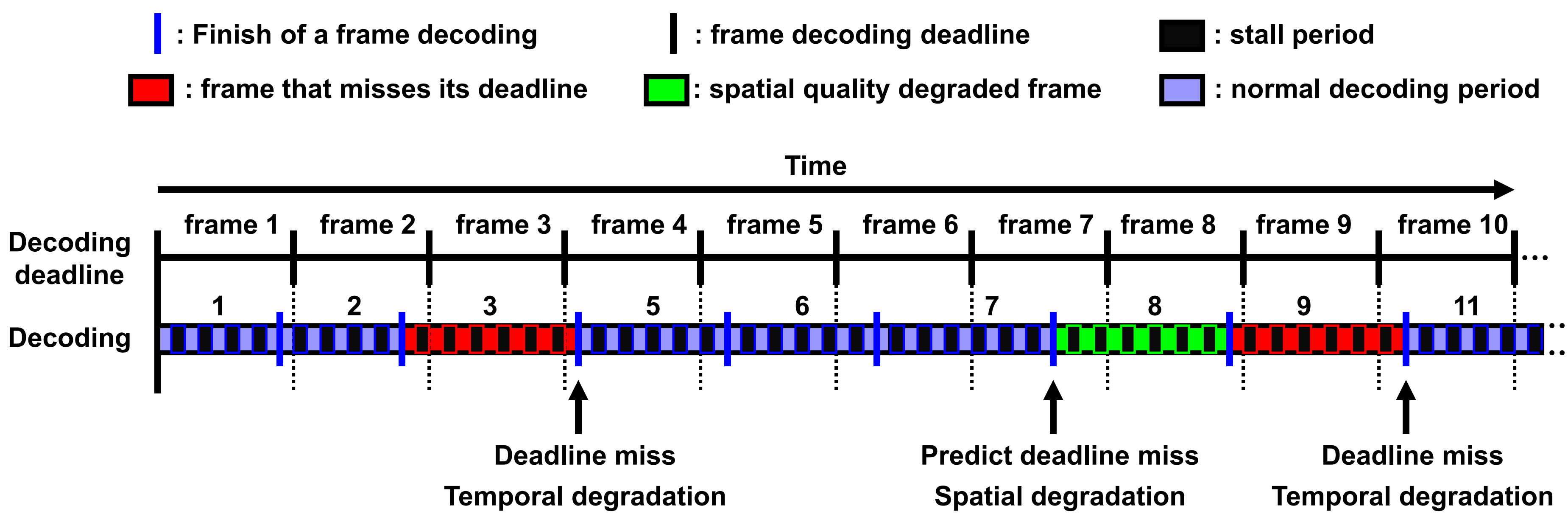
Spatial/Temporal Quality Degradation

- Spatial quality degradation (soft)
  - Two Fine Granularity Scalability (FGS) methods are chosen
    - SNR scalability
    - Saturation Control
  - Together, they consume about 10% of frame decoding time
  - Their quality degradations are negligible (as shown by RMSE values)
- Temporal quality degradation (hard)
  - Simply drop either P or B frames
  - This is similar to frame discarding scheme in MPEG when the decoding time becomes too long



Quality Degradation (Cont'd)

- Example to show how we apply spatial & temporal degradation
  - Based on the previous non I-frame, predict the frame decoding time
  - We cannot say which form of quality degradation will prevail:
    - If prediction is accurate and decoding workload is medium,
      - No. of spatially degraded frames > No. of dropped frame
    - If many frames have heavy decoding workload,
      - No. of spatially degraded frames < No. of dropped frame

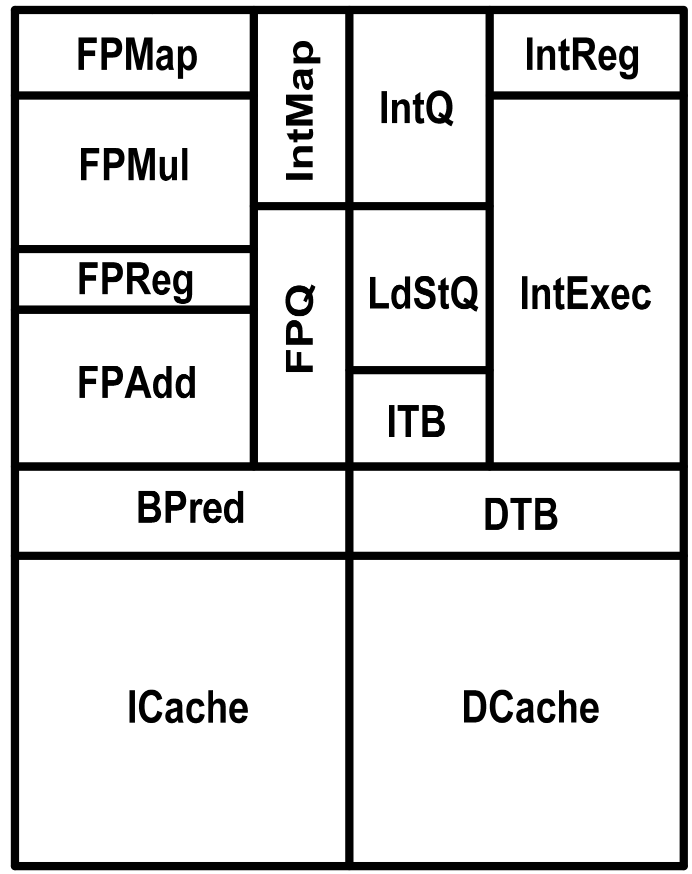


Simulation Setup

- Our thermal simulator
  - Combine Simplescalar 3.0, Wattch, and HotSpot
    - Generate per-structure temperature data for every 10K cycles
    - Based on the Alpha 21364 Chip floor-plan at 0.18μ, 1.8V, 1.2GHz
    - Emergency / Trigger temperatures: 85.0 / 81.8℃
    - Ambient / Initial temperatures: 40.0 / 60.0℃
- Application program
  - MPEG-2 decoder program in Media-bench
  - DTM policies are implemented in the MPEG-2 decoder program and interact with the thermal simulator
- Test input files
  - MPEG-2 video file (.m2v) from http://www.mpeg2.de/video/stream

Architecture Parameters and Floorplan

Memory Latency	100 cycles/10 cycles
L1 I/D Cache	64KB 2-way 32Byte block, 1 cycle hit latency
I/D-TLB	Fully associate, 128 entries, 30 cycles miss latency
Branch Predictor	4K Bimodal
Functional Units	4 INT ALU, 1 INT MULT/DIV, 2 FP ALU, 1 FP MULT/DIV
RUU/LSQ size	64/32
Instruction Fetch Queue	8
In order Issue	False
Wrong Path Execution	True
Issue Width	6 instruction per cycles



ALPHA 21364 Floor-plan in 0.13um

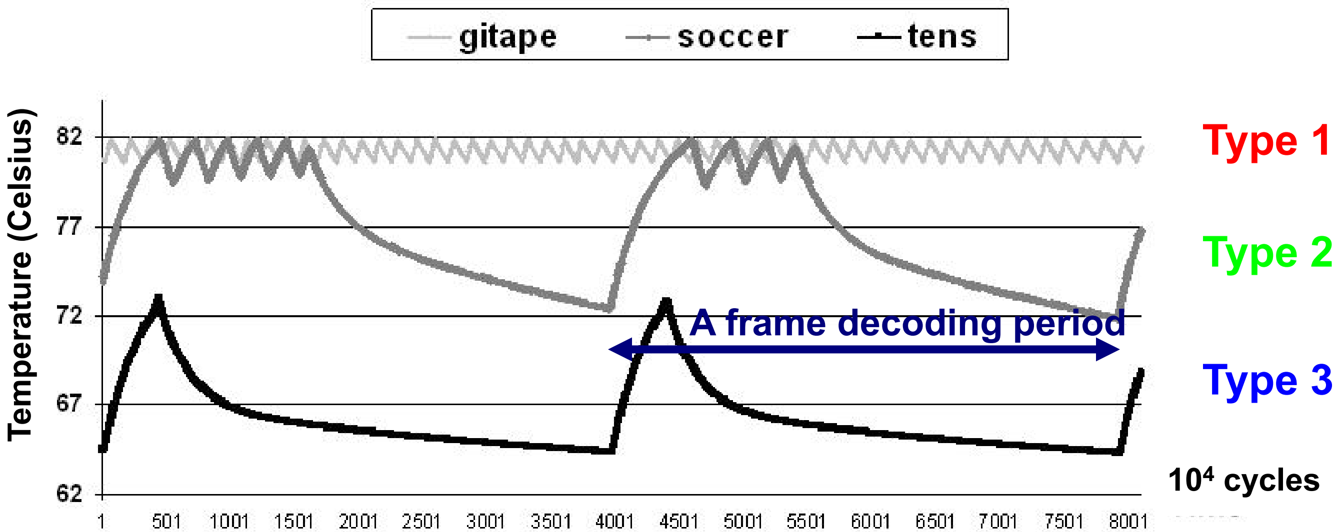
Experimental Results

- **Thermal results between no DTM vs. DTM-aware systems**
  - **When per-frame decoding time exceeds a certain value, DTM is needed**
  - **Our experimental results show that DTM support is clearly needed**

Input files	Average decoding time (msec)	Resolution (pixel)	No. of Frame	I: P: B frame	Max/Min Temp (℃)	
					w/o DTM	w/ DTM
gitape	21.5	720 x 480	14	1: 4: 9	101.5 / 85.5	81.8 / 80.5
mei60f	19.6	704 x 480	50	5: 13: 32	99.6 / 83.8	81.8 / 80.5
hhilong	17.2	720 x 576	45	3: 8: 34	97.2 / 81.9	81.8 / 80.5
time	11.8	704 x 480	50	5: 12: 33	91.5 / 76.2	81.8 / 80.5
soccer	8.5	640 x 480	51	4: 14: 33	82.5 / 70.5	81.8 / 72.4
tens	4.0	352 x 192	47	5: 12: 30	73.4 / 63.2	73.4 / 63.2
cact	4.0	352 x 192	50	5: 12: 33	73.4 / 64.1	73.4 / 64.1

Experimental Results (Cont'd)

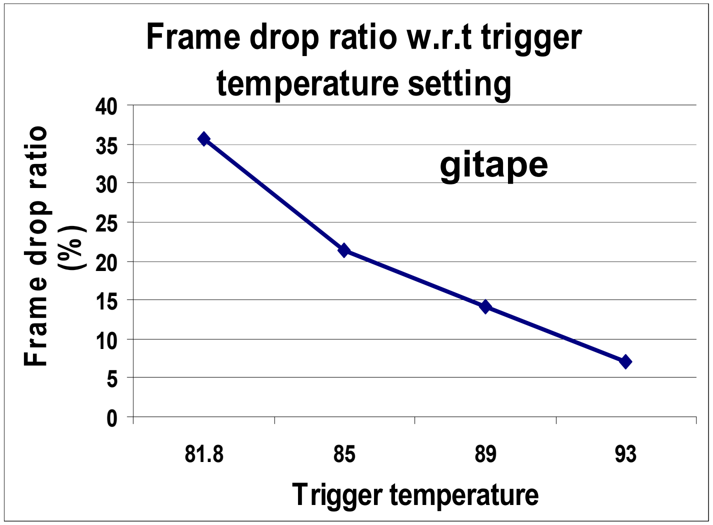
- Categorize simulated input files into three types and show thermal variations of each type
  - Type1: Large resolution (≥704X480) files: Need aggressive DTM most of time
  - Type2: Medium resolution (≈640X480) files: Some level of DTM is needed
  - Type3: Small resolution (≤ 352X192) files: No DTM is needed
- In the middle curve, stall time is adjusted to make thermal rising and falling gradient equal



Experimental Results (Cont'd)

- Spatial & Temporal Quality Degradation
  - As a frame resolution becomes large, DTM becomes aggressive, i.e., experience higher spatial-temporal quality degradation
  - If the trigger temperature is set to a higher value, the frame drop ratio becomes less

Input file	Resolution (pixel)	Image/Video Quality Degradation			
		Spatial		Temporal	
		Scaled frames	RMSE	Drop/Total frames	Drop ratio (%)
gitape	720 x 480	5	0.119	5/14	35.7
mei60f	704 x 480	8	0.125	15/50	30.0
hhilong	720 x 576	0	0	8/45	8.8
time	704 x 480	0	0	0/50	0
soccer	640 x 480	0	0	0/51	0
tens	352 x 192	0	0	0/47	0
cact	352 x 192	0	0	0/50	0



Conclusion

- Presented a DTM approach for MPEG-2 Decoding:
  - Utilizes residual time in a given decoding deadline for the thermal safety
  - Defines three thermal zones: super-linear, linear, and constant
  - Compared to the conventional DTM schemes
    - Does not pay the penalty of performance (speed) penalty but pays the penalty of quality degrardation instead

- Future Research:
  - Is FGS the best choice in terms of efficiency, i.e., maximize the time saving & minimize the image distortion?
  - Will DTM for the MPEG-4 be similar?
  - What if DVFS is applied instead?