ICCD 2006

USC

Stochastic Dynamic Thermal Management: *A Markovian Decision-based Approach*

Hwisung Jung, Massoud Pedram

University of Southern California

Outline

- Introduction
- Background
- Thermal Management Framework
  - Accuracy of Modeling
  - Policy Representation
- Stochastic Dynamic Thermal Management
  - SDTM Algorithm
  - Multi-objective Design Optimization
- Experimental Result
- Conclusions

Introduction

- "Smaller and faster" translate into high power densities, higher operating temperature, and lower circuit reliability.
- Local Hot Spots are becoming more prevalent in VLSI circuits.
- It is no longer sufficient to merely add a bigger fan as a downstream fix for thermal problems.
- Thermal managements need to be best accomplished when it is incorporated starting at the beginning of the design cycle.
- Any applications that generate heat should engage in runtime thermal management technique, i.e., *dynamic thermal management*.

Prior Work

- K. Skadron, et al. : architectural-level thermal model, *HotSpot*, (*ISCA 2003*)
- D. Brooks, et al. : trigger mechanism (*HPCA 2001*).
- J. Srinivasan, et al. : predictive DTM (*Int'l Conf. on Supercomputing 2003*)
- S. Gurumurthi, et al. : performance optimization problem for disk drives (*SIGARCH 2005*)

- For a good survey, see P. Dadvar, et al. (*Semi-Therm Symp. 2005*)

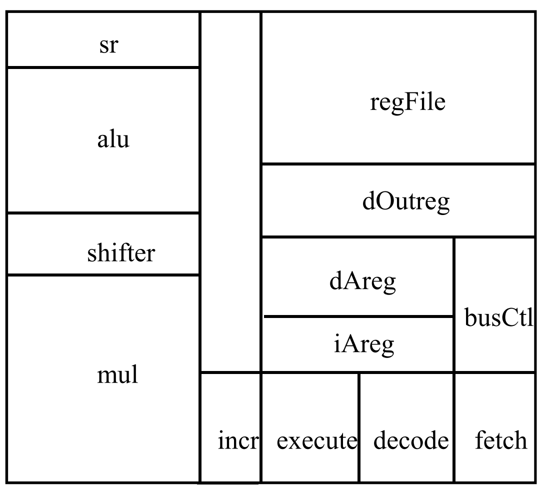
Motivation

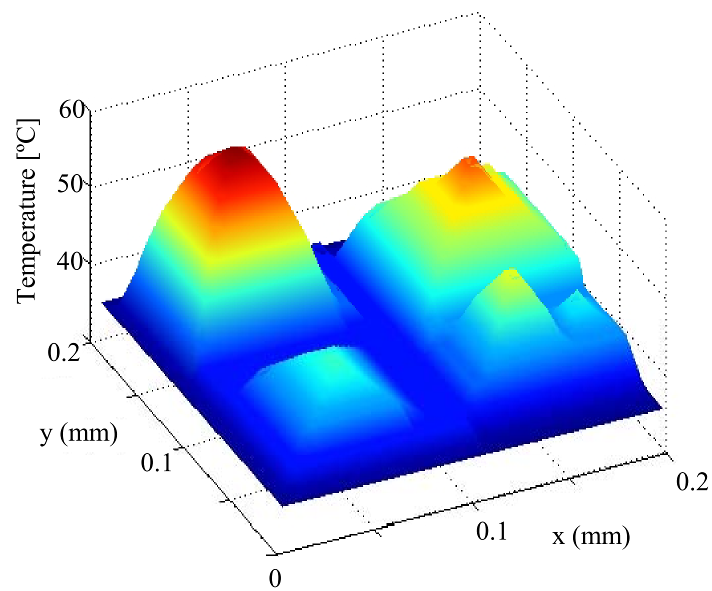
- The above-mentioned DPM techniques have difficulty *observing* the peak power dissipation on the chip because they rely on a single temperature sensor, whereas the peak temperature may appear in a number of different places on the chip. This gives rise to *uncertainty* about the true temperature state of a chip.
- Improving the accuracy of decision making in thermal management by modeling and assessing the *uncertainty* in temperature observation is an important step to guarantee the quality of electronics.

- Develop a stochastic thermal management framework, based on
  - Partially observable Markov decision process (POMDP)
  - Semi-Markov decision process (SMDP)
- Combine DTM and DVFS to control temperature of system and its power dissipation.

Uncertainty

- Critical problems in temperature profile characterization are:
  - Placement restriction for the on-chip temperature sensor
  - Non-uniform temperature distribution across the chip.
- This causes uncertainty in the temperature observation.





Example of non-uniform temperature distribution

Stochastic Process Model

- The uncertainty problem, where a thermal manager cannot reliably identify the thermal state of the chip, can be solved by modeling decision making by a stochastic process.
- A thermal manager observes the overall thermal state and issues commands (i.e., actions) to control the evolution of the thermal state of the system.
- These actions and thermal states determine the next-state probability distribution over possible next steps.

- The sequence of thermal states of the system can be modeled as a stochastic process.

Thermal Management Framework

- SMDP to model event-driven decision making.
- POMDP to consider the uncertainty in temperature observation.

Note: The time spent in a particular state in the SMDP follows an arbitrary distribution, realistic assumption than an exponential distribution.

- Definition 1: Partially Observable Semi-Markov Decision Process.

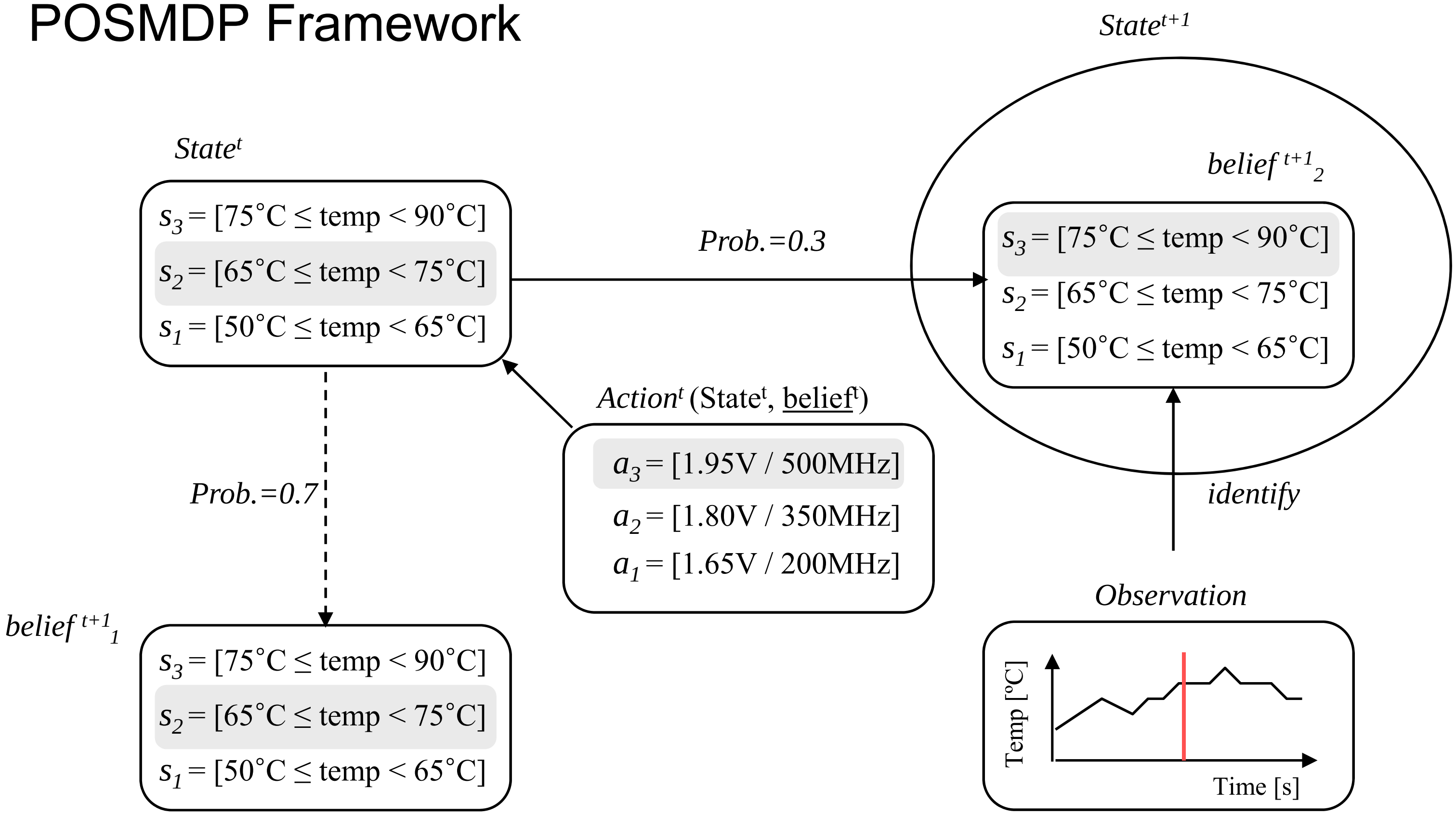
A POSMDP is a tuple (S, A, O, T, R, Z) such that

1) S is a finite set of states.
2) A is a finite set of actions.
3) O is a finite set of observations.
4) T is a transition probability function. $T: S \times A \rightarrow \Delta(S)$
5) R is a reward function. $R: S \times A \rightarrow \Re$
6) Z is an observation function. $Z: S \times A \rightarrow \Delta(Z)$

where $\Delta(\cdot)$ denotes the set of probability distributions.

POSMDP By Way of an Example

- POSMDP Framework



POSMDP framework for dynamic thermal management

Full History

- In an observable environment, observations are probabilistically dependent on the underlying chip temperature.
- Transition probability function determines the probability of a transition from thermal state *s* to *s'* after executing action *a*

$$T(s',a,s) = Prob(s^{t+1} = s' \mid a^t = a, s^t = s)$$

- Observation function captures the relationship between the actual state and the observation.

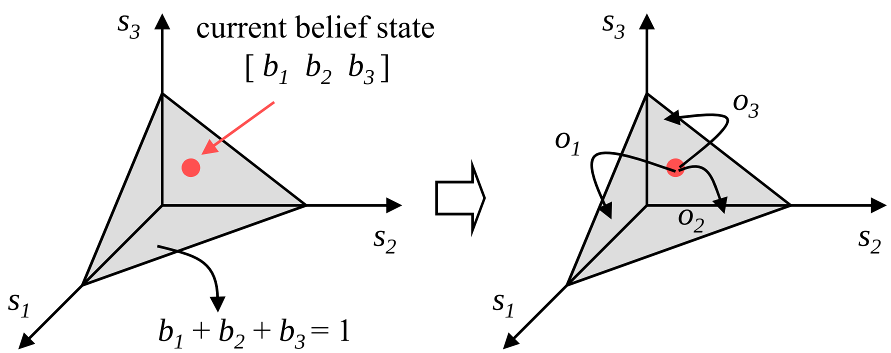
$$Z(o',s',a) = Prob(o^{t+1} = o' \mid a^t = a, s^{t+1} = s')$$

- Since thermal manager cannot fully observe the thermal state of the system, it makes decisions based on the observable system history.

➡ Relying on the full history $<s^0, a^0>, <s^1, a^1>, \ldots, <s^t, a^t>$ makes the decision Process *non-Markovian,* which is not desirable.

How to Avoid Reliance on Full History

- Although the observation gives the thermal manager some evidence about the current state *s*, *s* is not exactly known.
  - We maintain a distribution over states, called a belief state *b*.
- The belief state (vector) for state *s* is denoted as *b(s); Given any actual state,* the sum of belief state probabilities over all belief states is equal to 1.



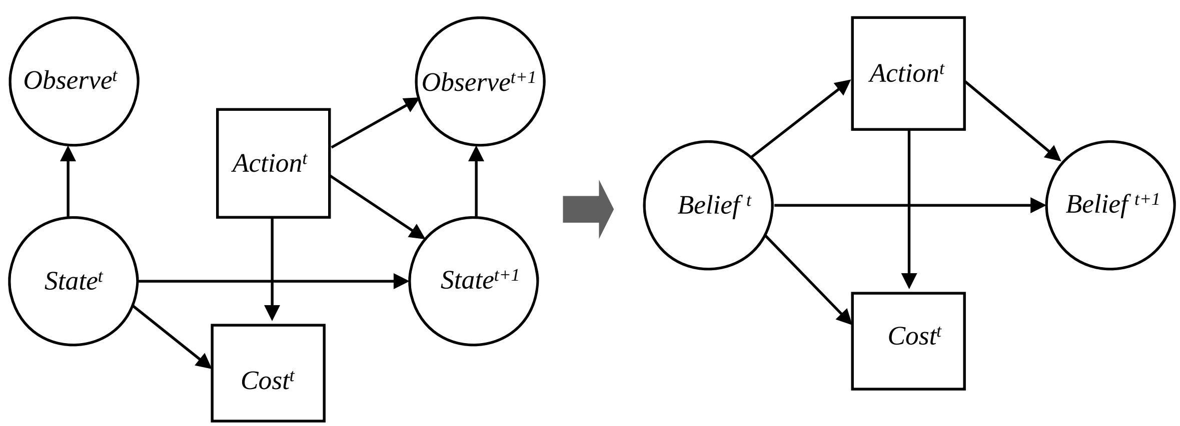
Note: Even though we know the action, the observation is not known in advance since the observations are probabilistic.

Markovian Property

- By using the state space *B*, a properly updated probability distribution over the thermal state *S*, we can convert the original history-based decision making into into a fully observable SMDP.

$$b'(s') = \frac{\sum_{s\in S} Prob(s', o \mid s, a) b(s)}{\sum_{s\in S} \sum_{s'\in S} Prob(s', o \mid s, a) b(s)}$$

- The process of maintaining the belief state is Markovian, where the belief state SMDP problem can be solved by adapting the value iteration algorithms.



Thermal Management Framework

- Thermal manager's goal is to choose a policy that minimizes a cost.
- Let $\pi: \mathrm{B} \rightarrow \mathrm{A}$ represent a stationary policy that maps probability distribution over states to actions.

- By incorporating expectation over actions, the set of stationary policies can be determined by using Bellman equation

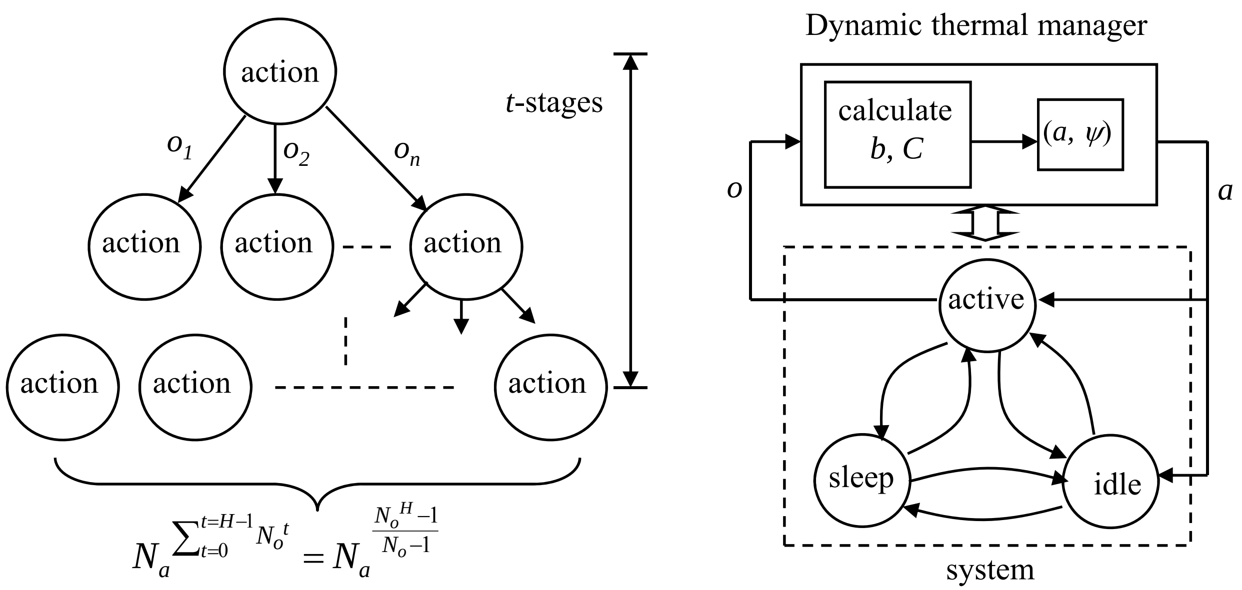
$$C^{\pi}(b) = \sum_{s \in S} b(s) k(s, a) + \gamma \sum_{o \in O} \sum_{s' \in S} Z(o, s', a) \sum_{s \in S} T(s', a, s) C^{\pi}(b')$$

- The optimal action to take in *b* is obtained by

$$\pi^{*}(b) = \arg\min_{a \in A} C^{\pi}(b)$$

Stochastic DTM

- A stochastic dynamic thermal management based on POSMDP.



- Definition 2: Observation strategy. In policy tree, an observation strategy is defined as $\psi : O \rightarrow \Lambda$ such that

  1) O is a finite set of observations

  2) $\Lambda = \{(a, \psi) \mid a \in A, \psi \in \Lambda_o\}$

  where A is a finite set of actions, and Λ_o is a finite set of all policy trees.

Stochastic DTM

- A multi-objective design optimization method is used to optimize the performance metrics by formulating mathematical programming model.
- Let a sequence of belief states b^0, b^1,.., b^n denote a processing path δ from b^0 to b^n of length n.
- For a policy π, the discounted cost C of a processing path δ of length n is defined as

$$C^{\pi}(\delta) \triangleq \sum_{i=0}^{n} \gamma^{t_i} cost(b^i, a^i)$$

where t^i denotes the duration of time that the system spends in belief state b^i before action a^i causes a transition to state b^{i+1} , and

$$cost(b,a) = pow(b) + \frac{1}{\tau(b,a)} \sum_{b' \in B} Prob(b' \mid b,a) ene(b,b')$$

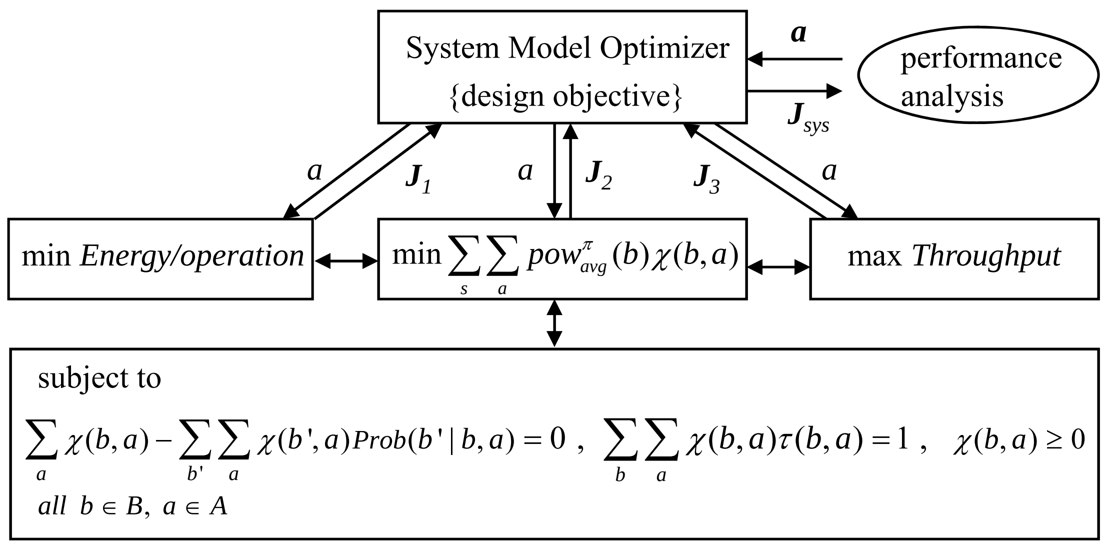
- *pow(b)* is the power consumption of the system in belief state *b*
- *Prob(b' | b, a)* is the probability of being in state *b'* after action *a* in state *b*
- *ene(b, b')* is the energy required by the system to transit from state *b* to *b'*
- τ*(b, a)* is the expected duration of time that system spent in *b* when action *a*

Stochastic DTM

- Considering the expectation with respect to the policy over the set of processing paths starting in state *b*, the expected cost of the system is

$$pow_{avg}^{\pi}(b) = EXP[C^{\pi}(\delta)]$$

- The design objective is a vector ***J*** of performance metrics we are trying to optimize, where the design vector ***a*** contains DVFS set.

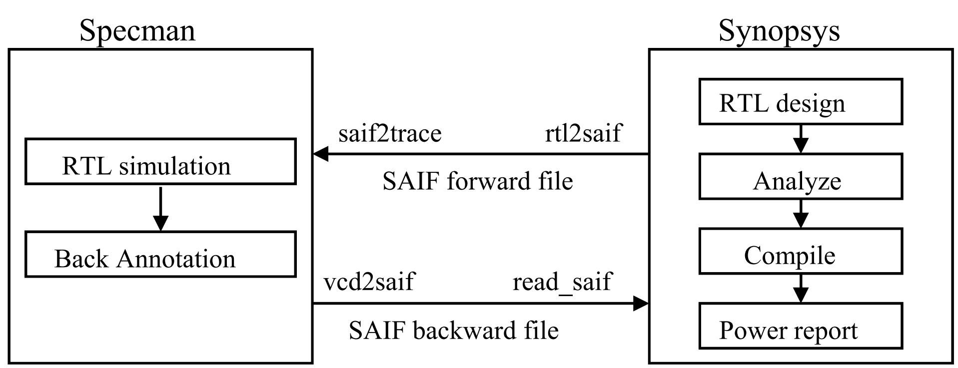


Note: $\chi(b, a)$ is the frequency that the system is in thermal state *b* and action *a* is issued.

Experimental Results

- For experimental setup, a 32bit RISC processor is designed in 0.18um technology, and stochastic framework is implemented in Matlab.
- In the first experiment, we analyzed power dissipation distribution of RISC.

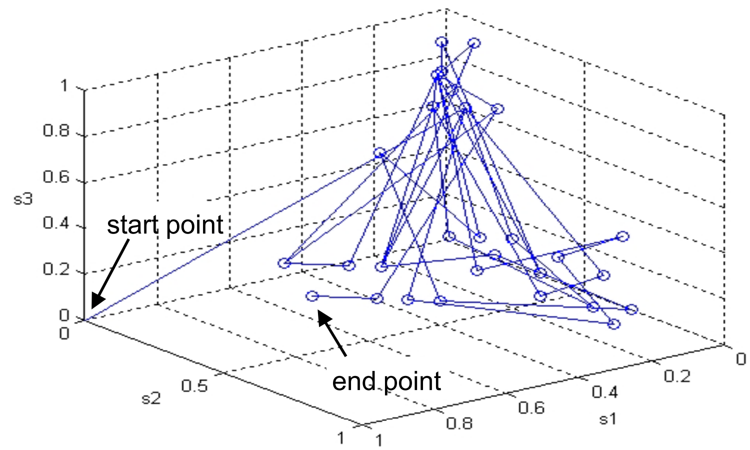
SPECint 2000	dOutreg	dAreg	iAreg	incr	mul	shifter	alu	sr	reg	decode	fetch	execute	busCtl
gcc	4.6%	14.4%	13.8%	4.1%	2.3%	2.7%	16.5%	2.2%	15.4%	4.1%	4.1%	4.1%	11.7%
gap	4.3%	13.1%	15.7%	4.0%	4.2%	3.1%	14.2%	1.7%	14.8%	4.2%	4.2%	4.2%	12.3%
gzip	4.6%	9.2%	15.4%	4.6%	4.5%	3.8%	18.6%	2.3%	15.1%	4.6%	4.6%	4.6%	8.1%



Power simulation flow

Experimental Results

- The second experiment is to demonstrate the effectiveness of our proposed SDTM algorithm.
  - Randomly choose a sequence of 40 program, e.g., gcc_1-$gzip_2$-gap_3-…-gcc_{39}-gcc_{40}.
  - The sequence of programs is executed on the RISC.
  - Calculate the belief state.

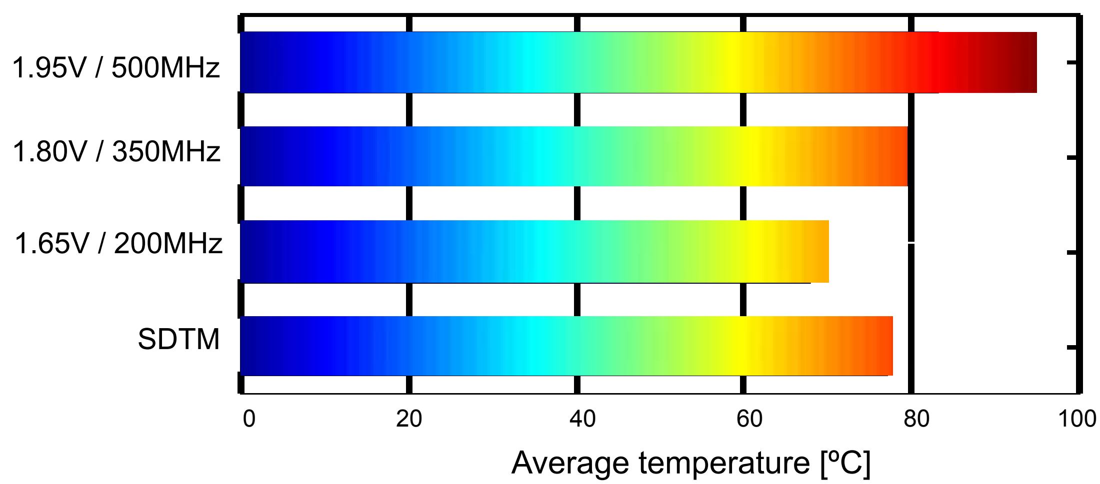


Action	Description	cost $[k(s_1,a)\ k(s_2,a)\ k(s_3,a)]$
a_1	[1.65V / 200MHz]	[1.5 1 0]
a_2	[1.80V / 350MHz]	[1 0 1]
a_3	[1.95V / 500MHz]	[0 1 1.5]

State , Observation	Description
s_1 , o_1	[50°C ≤ temp < 65°C]
s_2 , o_2	[65°C ≤ temp < 75°C]
s_3 , o_3	[75°C ≤ temp < 90°C]

Experimental Results

- Operating temperature of the RISC by running the sequence.



Average temperature for test scenario

Note: the average temperature is estimated based on $T_{chip} = T_a + \theta_{ja}(P_{s/w} + P_{s/c} + P_{static} + P_{int})$

Experimental Results

- A low value on the action axis means low supply voltage and low frequency Values of possible target actions, are obtained by multi-objective optimization

	1.95V/500MHz	1.80V/350MHz	1.65V/200MHz	SDTM
Average power	1.00	0.85	0.72	0.75
Throughput	1.00	0.69	0.40	0.69
Energy/operation	1.00	1.23	1.79	1.08

Achieve low power consumption and operating temperature with little performance Impact on throughput and energy/operation metrics.

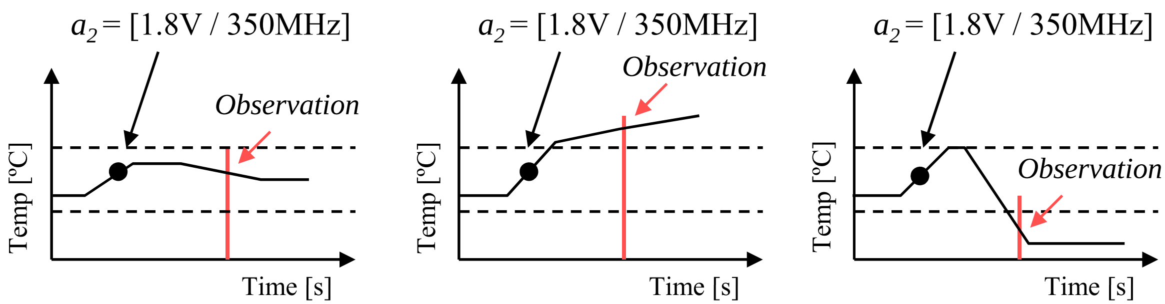
Conclusion

- Proposed a stochastic dynamic thermal management technique by providing stochastic management framework to improve the accuracy of decision making.
- POSMDP-based thermal management controls the thermal states of the system and makes a decision (DVFS set) to reduce operating temperature.
- Experimental results with design optimization formulations demonstrate the effectiveness of our algorithm (low temperature with little impact on performance metrics).

Thank You !!

Yet Another Example

- A scenario where starting with an action a_2 issued at time t, the next system state may be one of three possible ones as observed at time t+1.



- Case a: the system remains in the active mode with steady workload, resulting in the same chip temperature.
- Case b: the system remains in the active mode, but heavy workload, resulting in temperature increase.
- Case c: the system enters into the idle mode.

Backup Slide

Given CPU benchmark SPECint 2000 such as gcc, gap, and gzip with following characteristics,

program	# of ints. (in Million)
gcc	6765
gap	12726
gzip	75942

Calculate the CPI and total execution time.

For example, in gcc,

Operation	Freq.	Cycle	CPI	%
ALU	40%	1	40x1/100 = 0.4	0.4/1.4 = 25%
Load	35%	2	35x2/100 = 0.7	0.7/1.6 = 43%
Store	10%	2	10x2/100 = 0.2	0.2/1.6 = 13%
Branch	15%	2	15x2/100 = 0.3	0.3/1.6 = 19%

Total CPI = 1.6

CPU time = Inst/program x cycles/Inst x second/cycle
= I x CPI x C
= 6765 x 1.6 x 2ns (500MHz)
= 21648