

# COD310 Notes

## 1 Stuff to Read

1. 3D memory systems - 3D DRAM
2. Memory leakage power
3. SPEC CPU2006 benchmarks
4. Hybrid Memory Cube (HMC), High Bandwidth Memory (HBM) and Wide IO (WIO)
5. CACTI-3DD

## 2 Doubts

1. What mechanism controls the memory management in memory devices
2. ~~Why is 3D memory better?~~ (resolved)
  - compactness
  - Through Silicon Via - TSV (like?) technology is used

## 3 Leakage-Aware Dynamic Thermal Management of 3D Memories

### 3.1 Overview (Abstract)

1. Controlling leakage by monitoring temperature
2. Turn off specific memory channels to control temperature (before turning off, migrate data to 2D memory) - **FastCool**
3. **Energy-Efficient FastCool (EEFC)** - decides which channels to be closed

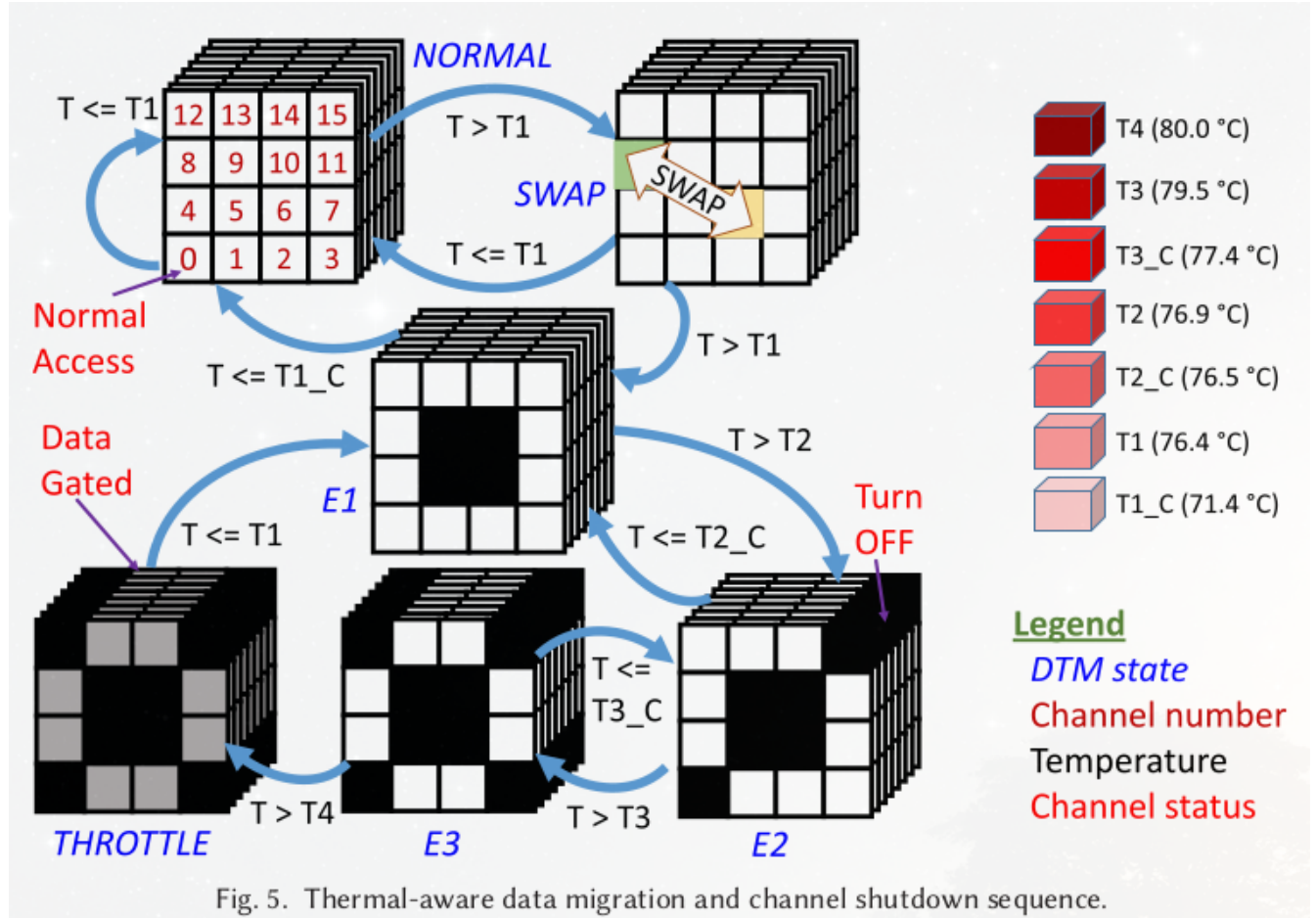
### 3.2 Introduction

1. 3D memory is stacked 2D DRAM, thus has higher power density
2. Power consumption involves dynamic (48%) and static/leakage power (52%)
3. Static power increases exponentially with temperature, thus a positive feedback between temperature and leakage

### 3.3 Proposed DTM Strategies

#### 3.3.1 TAM (Thermal-Aware Migration) States

1. NORMAL
2. SWAP
3. E1 (thermal Emergency 1)
4. E2
5. E3
6. THROTTLE



#### 3.3.2 Memory Delay Models

2D memory request time = Data Migration Delay (DMD) + Data Access Delay (DAD)

Symbol	Description	Typical value
$D$	Time duration for an epoch	1 ms
$C$	Cache Line Size	64 Bytes
<b>3D Memory Parameters</b>		
$b_{3D}$	Per channel 3D memory bandwidth	8 GBps
$L_{3D}$	3D memory access latency	29 ns
$s_{3D}$	Size of data migrated, 3D $\rightarrow$ 2D	<i>Runtime</i> <sup>1</sup>
$A_{3D}$	Total accesses made to $s_{3D}$ data (in last epoch)	<i>Runtime</i> <sup>1</sup>
$n_{3D}$	Number of 3D channels (migrating 3D $\rightarrow$ 2D)	<i>Runtime</i> <sup>1</sup>
<b>2D DRAM Parameters</b>		
$b_{2D}$	Per channel bandwidth	12.8 GBps
$L_{2D}$	2D memory access Latency	45 ns
$s_{2D}$	Size of data migrated, 2D $\rightarrow$ 3D	<i>Runtime</i> <sup>1</sup>
$A_{2D}$	Accesses made to 2D memory (in last epoch)	<i>Runtime</i> <sup>1</sup>
$N_{2D}$	Number of 2D memory channels	4
$R_{2D}$	Number of ranks per channel	2
$BA_{2D}$	Number of banks per rank	8
	Page Policy	Closed

<sup>1</sup> *Runtime* – Values are determined by DTM policy with the help of hardware counters.

### 3.3.2.1 DMD

$$DMD = (s_{3D} + s_{2D}) \times \max\left(\frac{1}{B_{3D}}, \frac{1}{B_{2D}}\right)$$

$$B_{3D} = b_{3D} \times n_{3D} \times (1 - \text{3D Memory Refresh Overhead})$$

$$B_{2D} = b_{2D} \times N_{2D} \times (1 - \text{2D Memory Refresh Overhead})$$

$$\text{Refresh Overhead} = \frac{\text{Time required to refresh a row} \times \text{Number of rows}}{\text{Refresh interval}}$$

### 3.3.2.2 DAD

$$DAD = DAD_B + DAD_L$$

$$A = A_{2D} + A_{3D}, \text{ total 2D memory accesses}$$

$$DAD_B = \frac{A \times C}{B_{2D}}$$

$$DAD_L = QD + LD, \text{ queuing delay} + \text{latency delay}$$

### 3.3.2.2.1 Latency Delay

$$LD = \frac{A \times L_{2D}}{BA_{2D} \times R_{2D} \times N_{2D}}$$

### 3.3.2.2.2 Queuing Delay

Uses Queuing Theory to model the waiting time.  $M/M/1$  model is used, having a single queue for each server, and arrival and service rates are Poisson and exponential respectively.

$$QD = \frac{A \times C}{T \times B_{2D}} \times \frac{N_{2D}}{B_{2D} - (A \times C/T)}$$

$$\left( \lambda = \frac{A \times C}{T \times N_{2D}}, \mu = \frac{B_{2D}}{N_{2D}}, \text{expected time} = \frac{\lambda}{\mu \times (\mu - \lambda)} \right)$$

### 3.3.3 FastCool

1. Transition to E1 happens only if total access count of channels  $\{5, 6, 9, 10\}$  exceeds  $A_{MIN}$  ( $A_{MIN} = 160,000$ )

$$A_{3D} > A_{MIN}$$

2. Queuing stability needs to be ensured before migrating ( $\lambda < \mu$ )

$$A < \frac{B_{2D} \times T}{C}$$

3. Ensure transfer to 2D memory happens only if 2D delay is below a certain threshold to prevent slow down of operations

$$Delay < D_{MAX}(= 8.415ms)$$

### 3.3.4 FC Policy Improvements

*later*

### 3.3.5 EEFC

*later*

### 3.3.6 Leakage Current Estimation

*later*

### 3.3.7 DTM Policy Implementation

*later*

## 4 PredictNcool: Leakage Aware Thermal Management for 3D Memories Using a Lightweight Temperature Predictor

### 4.1 Overview

1. Instead of reacting to temperature changes, this model attempts to utilise predicted temperature changes to reduce application runtime and memory energy
2. Symmetries in floor-plan and other design insights are used to reduce the predictor model parameters

## 5 CoreMemDTM: Integrated Processor Core and 3D Memory Dynamic Thermal Management for Improved Performance

### 5.1 Overview

Independent thermal management of core and memory leads to inefficient management since both cores and memories slow down

## 6 Project Problem Statement

Given  $n$  cores and  $k$  3D memory ranks, with each core accessing memory across all ranks (in some manner), maximise the total instructions per second (IPS) under a memory power budget and a thermal constraint:

$$\max \sum_i IPS_i,$$

$$\sum_r P_r \leq P_M$$

$$\max_r T_r \leq T_M$$

### 6.1 Points to Consider

1. Leakage power and temperature have a positive feedback loop
2. The problem formulation is similar to a knapsack problem but more constrained
3. Need to model a formal relation between  $P_r$  and  $IPS_i$
4. Model needs to be robust enough to be able to work efficiently for different corner cases such as:
  - low power budget
  - no memory accesses
  - high memory accesses

- specific memory is being accessed more

## 6.2 Basic Background

1. Memories have multiple standard power states
  - i. Accessing
  - ii. Active
  - iii. Standby
  - iv. Nap
  - v. Powerdown
  - vi. (a few more?)