Canvas Group Name: group\_58 Student 1: Matthew Wark, Mtwark@ucsc.edu Student 2: Kevin Le, kevle@ucsc.edu

CMPE110 HA6: DRAM & I/O Detective

Due Date: Friday 06/01/18 - **We are using our 2 day grace period**

# 1) DRAM Controller

Given below is a sequence provided to a DRAM chip using the chip interface consisting of the signals: bank select, address, row address strobe, column address strobe, write enable

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Time | BS | Address | RAS | CAS | WE |
| 1 | 1 | 5 | 1 | 0 | 0 |
| 2 | 1 | 3 | 0 | 1 | 0 |
| 3 | 2 | 4 | 1 | 0 | 0 |
| 4 | 1 | 6 | 1 | 0 | 0 |
| 5 | 1 | 2 | 0 | 1 | 1 |
| 6 | 2 | 2 | 0 | 1 | 0 |

1. Show the sequential sequence of commands (activate, precharge, read, write, refresh) as seen by the bit-cell arrays. (3 Points)

|  |  |  |
| --- | --- | --- |
| Command | Bank | Row/Col ID |
| ACTIVATE | 1 | Row 5 |
| READ | 1 | Col 3 |
| PRECHARGE | 1 | Row 5 |
| ACTIVATE | 2 | Row 4 |
| ACTIVATE | 1 | Row 6 |
| WRITE | 1 | Col 2 |
| READ | 2 | Col 2 |

b) **Assume a latency of 2 for all commands** (the first command occupies the cells marked with \*). Assume a simple DRAM controller that can have a single outstanding command. What is the overall latency of the access sequence? (3 Points)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Cycle | CMD 1 | CMD 2 | CMD 3 | CMD 4 | CMD 5 | CMD 6 | CMD 7 |
| 1 | ACT B1,R5 |  |  |  |  |  |  |
| 2 | ACT B1,R5 |  |  |  |  |  |  |
| 3 |  | READ B1,C3 |  |  |  |  |  |
| 4 |  | READ B1,C3 |  |  |  |  |  |
| 5 |  |  | PRE B1, R5 |  |  |  |  |
| 6 |  |  | PRE B1, R5 |  |  |  |  |
| 7 |  |  |  | ACT B2,R4 |  |  |  |
| 8 |  |  |  | ACT B2,R4 |  |  |  |
| 9 |  |  |  |  | ACT B1,R6 |  |  |
| 10 |  |  |  |  | ACT B1,R6 |  |  |
| 11 |  |  |  |  |  | WRITE B1,C2 |  |
| 12 |  |  |  |  |  | WRITE B1,C2 |  |
| 13 |  |  |  |  |  |  | READ B2,C2 |
| 14 |  |  |  |  |  |  | READ B2,C2 |
| 15 |  |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |  |

**Overall latency of 14**

c) Assume a controller that can have one outstanding command per bank. The controller processes commands in-order. The DRAM chip can receive one command per cycle but can have multiple outstanding commands. What is the overall latency of the access sequence (assuming a latency of 2 for each command)? (3 Points)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Cycle | CMD 1 | CMD 2 | CMD 3 | CMD 4 | CMD 5 | CMD 6 | CMD 7 |
| 1 | ACT B1,R5 |  |  |  |  |  |  |
| 2 | ACT B1,R5 |  |  |  |  |  |  |
| 3 |  | READ B1,C3 |  |  |  |  |  |
| 4 |  | READ B1,C3 |  |  |  |  |  |
| 5 |  |  | PRE B1, R5 |  |  |  |  |
| 6 |  |  | PRE B1, R5 | ACT B2,R4 |  |  |  |
| 7 |  |  |  | ACT B2,R4 | ACT B1,R6 |  |  |
| 8 |  |  |  |  | ACT B1,R6 |  |  |
| 9 |  |  |  |  |  | WRITE B1,C2 |  |
| 10 |  |  |  |  |  | WRITE B1,C2 | READ B2,C2 |
| 11 |  |  |  |  |  |  | READ B2,C2 |
| 12 |  |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |  |

**Overall latency of 11**

d) Assume a smart controller that has one outstanding command per bank and can **re-order accesses** (as long as functionality is not affected). The chip can still receive one command per cycle but have multiple outstanding commands. What is the overall latency of the access sequence (assuming latency of 2 for each command)? (3 Points)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Cycle | CMD 1 | CMD 2 | CMD 3 | CMD 4 | CMD 5 | CMD 6 | CMD 7 |
| 1 | ACT B1,R5 |  |  |  |  |  |  |
| 2 | ACT B1,R5 |  |  |  |  |  |  |
| 3 |  | READ B1,C3 |  |  |  |  |  |
| 4 |  | READ B1,C3 | ACT B2,R4 |  |  |  |  |
| 5 |  |  | ACT B2,R4 | PRE B1, R5 |  |  |  |
| 6 |  |  |  | PRE B1, R5 | READ B2,C2 |  |  |
| 7 |  |  |  |  | READ B2,C2 | ACT B1,R6 |  |
| 8 |  |  |  |  |  | ACT B1,R6 | WRITE B1,C2 |
| 9 |  |  |  |  |  |  | WRITE B1,C2 |
| 10 |  |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |  |

**Overall latency of 9**

# 2) Database Transaction Processing System

In this assignment you need to analyze a multi-core server for transaction processing. You will design a system that maximizes the metric: transactions processed per second (TPS). Assume the following characteristics and properties:

Transaction processing:  
**A transaction consists of two random 128-byte disk accesses and 3.2 Million instructions.**

Database size:

Consider an unlimited data base size, however, the TPS is limited to 1 TPS per 10 GB. I.e you will be able to fill every disk that you connect to the system but TPS is limited by the filesize placed on a disk.

Hardware cost:

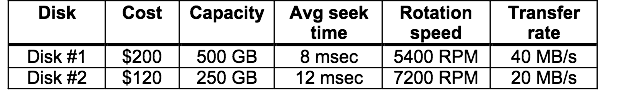
The baseline system (mainboard, rack, power) is $2,000.

Each CPU chip is $1,000, with each CPU having 4 cores per chip. Each processor can be connected to any number of disks

Per core performance:

**100 million instructions per second**

**The disk controller delay is 2 msec**

You can choose between two disk types, but you cannot mix them  
  


a) What is the access time for the two disk options? (2 Points)

**Disk 1:**

8ms seek time + ½ / (5400/60) = 0.00556 sec \* 1000 = 5.56ms rotational latency

* 128B / 40 MB/s = 128B / 40,000,000B/s = 0.0000032s \* 1000 = 0.0032ms
* 2ms controller delay

= **15.5632ms**

**Disk 2:**

12ms seek time + ½ / (7200/60) = 0.00417 sec \* 1000 = 4.17ms rotational latency

* 128B / 20 MB/s = 128B / 20,000,000B/s = 0.0000064s \* 1000 = 0.0064ms
* 2ms controller delay

= **18.1764ms**

b) What TPS do the 2 disk options provide, considering disk performance and file size? (1 Point)

Disk 1 max theoretical possible TPS = 500 GB / 10GB (per 1 TPS) = 50 TPS

Disk 2 max theoretical possible TPS = 250 GB / 10GB (per 1 TPS) = 25 TPS

Disk realistic max access time:

**Disk 1** = 15.5632ms\* 2 = 31.1264ms for one transaction (two accesses)

**Disk 2**  = 18.1764ms \* 2 = 36.3528ms for one transaction (two accesses)

**Disk 1** TPS considering our theoretical max disk performance and file size:

1 / 0.0311264 Seconds = **32.12706 TPS**

**Disk 2**  TPS considering our theoretical max disk performance and file size:

1 / .0363528 Seconds = 27.50819 TPS ---> capped at 25 so final answer is **25 TPS**

c) What cost/performance is delivered by the 2 options? (2 Points)

**Disk 1:** $200/32.12706 TPS = **$6.22528 per TPS**

**Disk 2:** $120/25 TPS = **$4.8 per TPS**

d) How many CPUs do you need for the best disk option? (2 Points)

32.12706 TPS \* 3.2 million instructions = 102,806,592

102,806,592 / 400,000,000 -> We need 0.257 CPU for disk 1.

e) What is the optimal system you can build for $10,000? (2 Points)

1. **Base = $2,000**
2. **Hard Drives = $3,000**
3. **CPUs = $5,000**

**Total = $10,000**

We used disk 2 since it was more **cost efficient at $4.8 per TPS** compared to disk 1’s $6.22 per TPS.

We started off putting all of our budget after our base build and one CPU into our biggest bottleneck which was our disk size. Then we reduced the amount of disks we got until we got to a number of CPUs we can afford.

f) How many TPS does the system of e) deliver? (1 Points)

$3000 (HDD budget) / $120 per HDD (Disk 2) = 25 Hard drives \* 25 TPS (per disk 2 HD) = **625 TPS (final answer)**

Reference:

$5000 (CPU budget) / $1000 per CPU = 5 CPUs

5 CPUs \* 400,000,000 million instructions per second (per CPU) = 2,000,000,000 instruction per transaction = 625 TPS \* 3.2 million instructions (per transaction)