ECE 485/585 Midterm Exam Solution

Name: ________________________________

Time allowed: 100 minutes
Total Points: 70
Points Scored: _____

Problem No. 1 (12 points)

For each of the following statements, indicate whether the statement is TRUE or FALSE:

(a) In single handshake protocol, the sender does not need to seek permission from the receiver before transmitting the data. **TRUE**
(b) Using DMA can speed up the data transfer between a network interface card and main memory. **TRUE**
(c) Adding multiple ports to an SRAM does not incur any area overhead. **FALSE**
(d) The use of “additive latency” commands in DDR2 reduced bus bandwidth utilization. **FALSE**
(e) A write-through cache does not need any “dirty” bits in the tag array. **TRUE**
(f) DRAM refresh overhead usually decreases with an increase in DRAM device density. **FALSE**

Problem No. 2 (8 points)

For each of the following questions, encircle **ALL** the correct answers:

(a) (2 points) A divide-by-zero exception is an example of:
   i. Asynchronous interrupt
   ii. Non-maskable interrupt
   iii. **Synchronous interrupt**
   iv. None of the above

(b) (4 points) A DDR3-1600 DRAM has the following timing parameters: $t_{\text{RAS}} = 14$ cycles, $t_{\text{RP}} = 6$ cycles, $t_{\text{RRD}} = 2$ cycles, $t_{\text{FAW}} = 10$ cycles. What is the maximum number of ACTIVATE commands that can be sent to the DRAM within a time window of 25 nanoseconds:
   i. 10
   ii. 8
   iii. 2
   iv. 1

(c) (2 points) In the SRAM vs. DRAM comparison, which of the following statements are true?
   i. SRAM has a lower cost per bit compared to DRAM
   ii. **Both SRAM and DRAM are volatile memories**
   iii. SRAM cell contents need to be restored after every read operation
   iv. **SRAM technology is more suitable for use in CPU caches**
   v. All of the above
Problem No. 3 (12 points)

(a) (5 points) A finite state machine (FSM) is being used to recognize a certain bit sequence. The input to the FSM is a sequence of binary bits in series. The following state transition diagram shows the FSM state transitions and output for each state ("x" represents the serial input):

![State Transition Diagram](image)

Assume that the following input sequence is applied to the FSM:
Input: 0001100101001
Given the above input sequence, what will be the output sequence generated by the FSM?

This FSM detects the “001” sequence in its last 3 input bits. Therefore, the output for the provided input sequence is as follows: **0001000100001**

(b) (3 points) Why did the Intel 8086 CPU use multiplexed address/data pins?

Using the same pins for both address and data reduced the total pin count, which in turn reduced the package cost.

(c) (4 points) State ONE advantage and ONE disadvantage of using memory-mapped I/O as compared to using direct I/O.

**Advantage:** No special I/O instructions (simpler instruction set)

**Disadvantage:** Must have a solution to avoid caching device registers
Problem No. 4 (15 points)

(a) (6 points) A computer system uses 44-bit memory addresses. It has a 32Kbyte direct-mapped cache with 64B blocks. Calculate the number of bits in each of the “Tag”, “Index” and “Byte Select” fields of the memory address.

Block size = 64 bytes = $2^6$ bytes
Therefore, Number of bits in the Byte Select field = 6
Cache size = 32K-byte = $2^{15}$ bytes
Total number of cache blocks = Cache size / Block size = $2^{15} / 2^6 = 2^9 = 512$
Since this is a direct-mapped cache, number of sets = $2^9 / 1 = 2^9$
Therefore, Number of bits in the Index field = 9
Total number of address bits = 44
Therefore, Number of bits in the Tag field = 44 - 6 - 9 = 29

(b) (9 points) The following figure shows the organization of a memory system built by using 64M x 16 chips as building blocks:

Answer the following questions:

I. (3 points) How many words does this memory system have? What is the word size?

There are 4 rows of memory chips with 64M words in each chip.
Therefore, total number of words = 64M * 4 = 256M
There are 4 chips in each row. All these chips operate in parallel on a single request and each chip provides 16 bits. Therefore, size of each word = 16 * 4 = 64 bits

II. (2 points) How many address lines are needed to access this memory system?

Number of address lines needed = $\log_2$(Number of words) = $\log_2$(256M) = 28

III. (4 points) Assume that the memory system is using low order address interleaving. How are the address lines distributed amongst different chips? Show your answer by replacing the “?”s in the figure with address line numbers.

Low order address bits (A0 and A1) go to the 2-to-4-decoder. Higher order address bits (A2 – A27) go to each chip.
Problem No. 5 (23 points)

A processor uses a dual-rank DDR4 memory system. Table-1 shows the relevant memory system parameters. Some of the parameters have been intentionally left out.

The second table (Table-2) shows a list of DRAM requests initiated by the processor. For each request, the table shows the DRAM rank, bank and row numbers corresponding to the address being accessed by that request. Also, for each request, the table provides the start time and the completion time of each request in terms of number of cycles. The completion time refers to the time at which the entire data burst corresponding to that request has been received.

Assume that all the banks were in the precharged state (no open rows) at $t = 0$. Also assume that the memory controller is using an “open page” policy, such that once a row in a bank has been activated, it is kept open as long as there is no conflicting request to a different row in the same bank.

<table>
<thead>
<tr>
<th>Table-1</th>
<th>Table-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRAM Parameter</td>
<td>Value</td>
</tr>
<tr>
<td>Number of ranks</td>
<td>2</td>
</tr>
<tr>
<td>DRAM channel width</td>
<td>64 bits</td>
</tr>
<tr>
<td>DRAM chip output width</td>
<td>8 bits</td>
</tr>
<tr>
<td>DRAM chip capacity</td>
<td>8 Gbits</td>
</tr>
<tr>
<td>Number of banks</td>
<td>16</td>
</tr>
<tr>
<td>Row size</td>
<td>2KBytes</td>
</tr>
<tr>
<td>Burst length</td>
<td>8</td>
</tr>
<tr>
<td>Memory controller policy</td>
<td>Open page</td>
</tr>
</tbody>
</table>

Based on the information provided in the two tables, answer the following questions:

(a) (3 points) Calculate the total DRAM capacity available in the system.

Number of ranks = 2
DRAM chip capacity = 8 Gbits
Number of DRAM chips per rank = DRAM channel width / DRAM chip output width = 64 / 8 = 8
Capacity of each DRAM rank = 8 Gbits * 8 = 64 Gbit
Total DRAM capacity = Capacity of a rank * Number of ranks = 64 Gbit*2 = 128 Gbits or 16 Gbytes

(b) (4 points) When the memory controller sends an ACTIVATE command to this memory system, how many address lines does it need to specify the desired rank/row/bank combination?

Address lines needed to specify the desired “rank” = $\log_2(2) = 1$
Address lines needed to specify the desired “bank” = $\log_2(16) = 4$
Number of DRAM rows per bank = Capacity of each bank / Capacity of each row = (Rank capacity / Number of banks per rank) / Row capacity = (64Gbit/16)/2Kbytes = 4Gbit/16Kbit = $2^{32} / 2^4 = 2^{18}$
Therefore, Address lines needed to specify the desired “row” = $\log_2(2^{18}) = 18$
Total number of address lines needed to specify the rank/row/bank combination = 1+18+4 = 23
(c) **(5 points)** Based on the start time and completion time of requests shown in Table 2, calculate the following DRAM parameters (in terms of number of clock cycles): (i) $t_{\text{CL}}$, (ii) $t_{\text{RCD}}$, (iii) $t_{\text{RP}}$.

Let us consider the different latency scenarios:

**Requests # 1 and 2**
Latency = 22 cycles (from the table)
Each of these requests is a row buffer miss. In addition, for each of these requests, the bank was precharged before the request. Therefore, the latency is equal to $t_{\text{RCD}} + t_{\text{CL}} + t_{\text{BURST}}$. Hence:

$$t_{\text{RCD}} + t_{\text{CL}} + t_{\text{BURST}} = 22$$

**Requests # 3 and 6**
Latency = 30 cycles (from the table)
Compared to the first two requests, these request require an additional overhead of closing the already open row. Therefore, the latency is equal to $t_{\text{RP}} + t_{\text{RCD}} + t_{\text{CL}} + t_{\text{BURST}}$. Hence:

$$t_{\text{RP}} + t_{\text{RCD}} + t_{\text{CL}} + t_{\text{BURST}} = 30$$

**Requests # 4, 5 and 7**
Latency = 10 cycles (from the table).
All these requests are row hits. Therefore, their latency should be $t_{\text{CL}} + t_{\text{BURST}}$. Hence:

$$t_{\text{CL}} + t_{\text{BURST}} = 10$$

Since the burst length is 8, $t_{\text{BURST}} = 8/2 = 4$ (due to DDR).
Solving the above equations yields: $t_{\text{CL}} = 6$, $t_{\text{RCD}} = 12$, $t_{\text{RP}} = 8$

(d) **(4 points)** Given the request sequence shown in Table 2, would it be better for the memory controller to use a closed page policy instead of an open page policy? Explain why or why not?

For a closed page policy, all the requests will have the same latency: $t_{\text{RCD}} + t_{\text{CL}} + t_{\text{BURST}} = 22$ cycles
Therefore, total latency for closed page policy = $22 \times 7 = 154$ cycles
For the open page policy, requests # 1 and 2 take 22 cycles each, requests # 3 and 6 take 30 cycles each, whereas request # 4, 5 and 7 are row hits, which take 10 cycles each.
Therefore, total latency for open page policy = $(2 \times 22) + (2 \times 30) + (3 \times 10) = 134$ cycles
Comparing the total latencies, we conclude that it is profitable to use an open page policy.

(e) **(7 points)** To retain data at high temperature, each DRAM row must be refreshed once in every 32 milliseconds. The DRAM datasheet specifies the following two refresh commands that could be used by the memory controller:

(i) REF1: refreshes 32 rows in a bank and takes 500 nanoseconds
(ii) REF2: refreshes 64 rows in a bank and takes 700 nanoseconds

Each refresh command triggers a parallel refresh in every bank in both the ranks.
Memory controller designers have been given a target that the fraction of time for which the memory system remains unavailable due to refreshes must be less than 10%. Which one of the
above two refresh commands (if any) should the memory controller use to satisfy this target? Show your calculations.

Fraction of time for which the memory system is unavailable = $t_{RFC}/t_{REFI}$

In order to meet the target, $t_{RFC}/t_{REFI}$ should be less than 10%.

$t_{RFC}$ is the latency of a single refresh command and $t_{REFI}$ is the interval between two refresh commands.

**For REF1:**
- $t_{RFC} = 500$ nanoseconds = 0.5 microseconds
- Number of rows refreshed in a bank in one refresh command = 32
- Number of rows per bank = $2^{18}$ (from part (b))
- Therefore number of refresh commands needed in a 32millisecond interval = $2^{18}/32 = 8192$
- Hence, $t_{REFI} = 32ms/8192 = 3.9$ microseconds
- $t_{RFC}/t_{REFI} = 0.5/3.9 = 12.8\%$ does not meet the target.

**For REF2:**
- $t_{RFC} = 700$ nanoseconds = 0.7 microseconds
- Number of rows refreshed in a bank in one refresh command = 64
- Therefore number of refresh commands needed in a 32millisecond interval = $2^{18}/64 = 4096$
- Hence, $t_{REFI} = 32ms/4096 = 7.8$ microseconds
- $t_{RFC}/t_{REFI} = 0.7/7.8 = 9\%$ meets the target. Therefore, the memory controller should use **REF2**.