IIT Guwahati - Department of Computer Science & Engineering

CS 223- Computer Organization & Architecture – Assignment 3 (25.05.2020)

- **Read the following general instructions carefully before attempting your question.**
- Each student has to answer 2 questions; one from Part A (A1/A2/A3/A4) and one from Part B (B1/B2/B3/B4). Identify your question carefully as per details given below. If you are answering a question not assigned to you, marks will not be awarded even if the answer is correct.
- Maximum mark is 15; (5 for Part A and 10 for Part B).
- Write your answer with pen on a paper. You have to use two pages for your response, with one page to each question. Write your full name (in BLOCK LETTERS) and IITG Roll number on the top side of each page of the answer sheet. Put your signature also there. Scan the answer sheet and merge it into one pdf file with maximum 2 pages. Make sure your mobile phone has the right scanner apps so that you can do scanning and merging quickly. Kindly do practice this before the assignment time so that you are familiar with these processes. Save the pdf file as <RollNo_A3>.pdf where Roll No is your 9-digit unique roll number. However, provisions for two file uploads is also given for those who could not merge the file where each question can be separately uploaded. Single file submission is preferred. Max submission size is 1MB.
- Students can see the questions from 15:30 hours onwards in Moodle. Submissions link in the Moodle will be open from 16:00 hours onwards. Early submissions carry more weightage. Submissions after 17:00 hours, but before 18:00 hours, will have late penalty. Any submissions after that will get zero marks. Moodle will still accept submissions up to 20:00 hours, today. Preferred mode of submission is through Moodle. If Moodle upload is taking time/facing issues, you can send it via email after 16:45 hours. Alternatively, you can send the scanned copy of the sheets via WhatsApp to respective TAs. In all cases, the file should be clearly visible, properly named, and your Roll number should be given in the mail/WhatsApp messages clearly. It is your responsibility to ensure quality of the scan and naming conventions before upload. Multiple emails/WhatsApp messages and uploads on multiple platforms are strictly discouraged.
- Consider your IITG Roll Number. It has 9 digits. Take the last two digits (N). Perform N%16. Based on your answer, identify the questions allotted to you from the table.
  Eg: Let your Roll Number be 180101033. Here N=33. 33%16 is 1. So, you have to do A4 and B2.

N%16	Questions to be answered	N%16	Questions to be answered
0	A2, B4	8	A3, B2
1	A4, B2	9	A4, B1
2	A1, B3	10	A1, B4
3	A3, B1	11	A2, B2
4	A2, B3	12	A3, B3
5	A3, B4	13	A4, B4
6	A4, B3	14	A1, B2
7	A1, B1	15	A2, B1

Submission Mode: Even if you are late in completing the assignment, try Moodle submission. But if there are upload issues, email Instructor/Whatsapp TAs only after 16:45 hours. [Use only one of the three modes; Moodle/Email/WhatsApp. Multiple submissions will attract penalty.]

From 16:45 onwards CSE/M&C students	From 16:45 onwards CSE students, N<50	From 16:45 onwards CSE students, N>=50	From 16:45 onwards M&C students
Email: John Jose johnjose@iitg.ac.in	WhatsApp: Sivakumar S. +91-8089421619	WhatsApp: Manju R. +91-9961330220	WhatsApp: Dipika D. +91-9435875127

Question A1: Consider a 500 MHz processor with a 2-level, inclusive, look through cache system that has the following specifications. L1 hit time = 2 ns, L1 hit rate = 80%, L2 hit time = 10 ns, L2 miss penalty = 100 cycles. If average memory access time is 10 ns, what is the hit rate of L2 cache?

Question A2: A 32-bit word processor that uses a 32 KB, 4-way, L1 cache and 256 KB, 16-way, L2 cache has a physical address space of 16 MB. The block size of L1 and L2 caches are 16 and 64 words, respectively. Upon an L1 cache miss, it takes 50 ns to return the first word from L2 cache and additional 5 ns each for bringing subsequent words in the same block. An optimization is done on L1 cache such that the missed word can be forwarded to the processor to resume execution as and when it reaches L1 cache controller than waiting for the entire cache block to be filled.

(a) What is the physical address split of tag, set-index and byte-offset of L1 and L2 caches?

(b) How long does it take for the processor to resume if an L1 cache miss occurred for a word whose first byte is at address 0x8645A4?

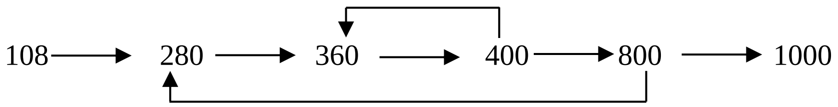
Question A3: Consider a cache that has a miss penalty of 200 clock cycles, and all instructions normally take 1 clock cycles for a perfect cache that is 100% hit. It was found that the processor with the perfect cache has 7 times speed up over the realistic cache while running a program that has an average of 1.5 memory references per instruction.

(a) What is the miss rate of the realistic cache and average MPKI for this program?

(b) How much slowdown (approximate) will the program experiences if there was no cache memory in the memory hierarchy?

Question A4: Assume that you are the cache design architect for a new processor that is going to be released by your company. Silicon designer engineers have given you the hit time values that is dependent on number of sets, associativity, and block size. Hit time (in nano seconds) of the cache is given by Ht = 0.2S + 2.5A + 0.3B, where S, A, and B are number of sets, associativity and block size (in bytes) of the cache. The total size of the cache is fixed to 1 KB. You are free enough to fix the associativity up to (including) 4 and to vary block size from 8 to 64 bytes. S, A, and B can only be power of 2. Which cache specification (S, A, and B combination) will be chosen by you, if your objective is to have the lowest hit time? Draw a graph of your observations with X axis as different cache configurations you analyzed and Y axis as Ht. What is your conclusion in terms of impact of S, A, and B in influencing the hit time?

Question B1: The memory of a computer is byte-addressable, and the word length is 8 bits. A program consists of two nested loops; a small inner loop (executed 20 times) and a larger outer loop (executed 10 times). The general structure of the program is given in the figure below. The decimal memory addresses shown delineate the location of the two loops and the beginning and end of the total program. All memory locations in the various sections of the program addresses, 108-279, 280-359, 360-400, 401-800, 801-1000 contain instructions to be executed in straight-line sequencing. Instructions at addresses 400 and 800 are conditional jumps. The program is to be run on a computer that has an instruction cache of 1 KB capacity, organized in the direct-mapped manner with a block size of 16 B. The miss penalty in the instruction cache is 80*x* clock cycles, where *x* is the hit time of the cache. Assume the cache is initially empty, Compute the miss-rate of the cache and the total time needed for instruction fetching of execution of the entire program.



Question B2: A system has a memory hierarchy that consists of one level of split cache and 1MB of main memory. Both I-cache and D-cache are 4KB direct mapped with 64 byte blocks. The system uses RISC architecture with 32-bit instructions and 32-bit data words. A program that does and array add operation C[ ]=A[ ]+B[ ] is stored in main memory beginning from address 0x42478. A, B and C are one dimensional arrays, each of size 64 data words. Elements are accessed in order from 0th element to 63rd element. The program consists of 10 instructions, out of which last 6 instructions are part of a loop. Each iteration of the loop access one unique element of A, B and C exactly once. The address allocation of A, B and C starts from physical addresses 0x27240, 0x4A2E0 and 0x27A20, respectively. Initially all the caches are empty. Assume all other temporary data variables used by the program, are in registers and are not affecting the cache memory mappings.

(a) Find the I-cache and D-cache miss rates.
(b) Find the average MPI (Misses per Instruction) of the program.
(c) Indicate non-empty sets in I-cache and D-cache at the end of the execution.
(d) Which all elements of A, B and C are residing in the cache at the end of execution of the program?

Question B3: Consider a 4-way set associative cache that uses pseudo LRU block replacement policy. Assume all the cache blocks are initially empty and filling up of empty blocks in a given cache set happens from way-0 to way-3. Consider the following 13 block numbers (excluding #, $, and @) all mapped to set n given in the order of arrival. A, R, S, R, B, A, D, C, S, A, S, B, E, #, $, @.

(a) Find the golden miss ratio of set n, if it is defined as the ratio of compulsory miss to conflict miss in the above sequence of 13 block requests.
(b) Consider the 14th, 15th and 16th requests #, $, and @, respectively given above mapped to set n. If access to # resulted in a conflict miss, and access to @ resulted in replacement of block ‘S’, give the set of all possible values that # and $ can have such that request to $ is a hit.

Question B4: An embedded system unit has a processor, a 64 KB cache memory and a 1 MB main memory. The word length is 64 bits. The cache memory is direct mapped with a block size of 256 words. Operating system kernel code is stored from word 0 to word 4095 (in decimal) in the main memory. The kernel code is in an infinite loop from word 2 to word 4092 (both inclusive). Execution of the code within this loop is fully sequential without any branch or loop instructions except two Interrupt Service Routine (ISR) calls. These ISR calls are at word 2000 (ISR_A) and 4000 (ISR_B). Both these ISRs are 1024 words each and are loaded from the word addresses 8192 and 10240, respectively. The last line of each of the ISR is a return statement back to the kernel. Each ISR code also is fully sequential without any branch or loop instructions. Assume caches are initially empty. All data of kernel and ISRs are in processor registers and will not affect cache mapping.

(a) Show the physical address split up of tag, set-index and byte offset.
(b) How many cache sets will be still empty after 50 iterations of the kernel loop? List them.
(c) If each context switch (control transfer from kernel to ISR/vice versa) takes ‘a’ milliseconds, execution of an instruction word takes ‘b’ milliseconds if it is a cache hit or ‘c’ milliseconds if it is a cache miss, then what is the execution time of 10 iterations of the kernel loop?