NANYANG TECHNOLOGICAL UNIVERSITY

SEMESTER 2 EXAMINATION 2020-2021

CE3001/CZ3001 – ADVANCED COMPUTER ARCHITECTURE

Apr/May 2021 Time Allowed: 2 hours

INSTRUCTIONS

1. This paper contains 4 questions and comprises 7 pages.
2. Answer **ALL** questions.
3. This is a closed-book examination.
4. All questions carry equal marks.
5. Appendix in Page 7 comprises of the details of the instruction formats.

---

1. (a) The speedup of machine M1 over M2 for a given program is 2. Let machines P1 and P2 be the enhanced versions of M1 and M2, respectively. It is found that one-third of the program runs 2 times faster in P1 than in M1 and half of the program runs 3 times faster in P2 than in M2. Calculate the speedup of P1 over P2.

(6 marks)

(b) State the addressing mode used by the conditional branch instruction "CBNZ X1, address" in the LEGv8 architecture. If the content of the 64-bit program counter (PC) is 0xCC, find the maximum possible address of the instruction memory to which the current conditional branch instruction "CBNZ X1, address" could branch forward.

(7 marks)

(c) Briefly explain the working of the instructions "LDUR X0, [X1, #8]" and "XOR X2, X1, X3". Use a neat diagram to show the datapath of a single-cycle architecture that supports the execution of both given instructions with minimal number of multiplexers (control signals can be simplified).

(12 marks)

2. (a) Listing Q2 shows a code segment that is intended to be executed in a 5-stage pipelined LEGv8 processor which can perform write-back and register-read operations of different instructions in the same clock cycle. The program counter is updated with the branch target address at the Execute stage. Let the initial values be `X5=0x0000000010000000` and `X6=0x0000000000001000` (`CBNZ`: *branch on not equal to 0*).

Listing Q2

```
I1   loop:  LDUR  X0, [X5,#0]
I2          LDUR  X1, [X6,#0]
I3          ADD   X2, X1, X0
I4          XORI  X2, X2, #15
I5          STUR  X2, [X6,#0]
I6          SUBI  X5, X5, #8
I7          SUBI  X6, X6, #8
I8          CBNZ  X6, loop
    finish
```

(i) Calculate the steady state CPI of the code segment in Listing Q2 with the help of a reservation table for the execution of the code if no data forwarding is allowed. Also find the total number of loop iterations.

(6 marks)

(ii) Perform loop unrolling by a factor of 2 for the code segment in Listing Q2 and do the necessary reordering of instructions to reduce the number of stall cycles to the minimum. Find the steady state CPI achieved by such loop unrolling and instruction reordering when no data forwarding is allowed. You are allowed to use new temporary registers to get rid of hazards.

(6 marks)

(iii) The code segment shown in Listing Q2 is now intended to be executed in a two-way superscalar processor. In the superscalar processor, one way is exclusively for load and store instructions whereas another way can execute all instructions except load and store. Find the CPI achieved for the code segment shown in Listing Q2 using superscalar architecture. Note that, no data forwarding is allowed but write-back and register-read operations of different instructions can be performed in the same clock cycle.

(8 marks)

Note: Question No. 2 continues on Page 3

(b) Consider the following sequence of actual outcomes for a branch (N N N N N N N, T N T N T N), where T means that the branch is taken and N means that the branch is not taken.

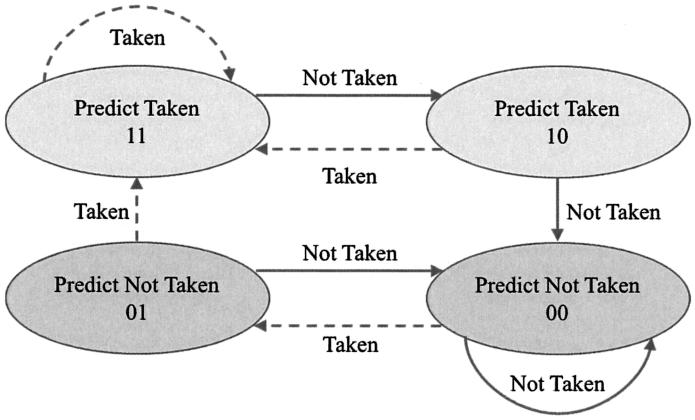


Figure Q2b

Assume that there is only one branch instruction in the program. What is the prediction accuracy for the last 6 occurrences of this branch if the 2-bit branch predictor as shown in Figure Q2b is applied?

(5 marks)

3. (a) Name three cache replacement algorithms and two write policies for cache.

(5 marks)

(b) Figure Q3b depicts the memory access workflow for a byte-addressing machine A. Make use of the information in Figure Q3b to answer the following questions.

(i) Given that the L1 cache is a direct-mapped cache which contains 512 cache blocks with the block size of 128 bytes, calculate **Y** and **Z** in Figure Q3b.

(4 marks)

(ii) What is the size of the main memory of the machine A? Suppose a program accesses the physical memory in a random fashion, that is, random physical addresses are uniformly accessed for a sufficiently long time. What is the miss rate on the L1 cache?

(6 marks)

Note: Question No. 3 continues on Page 4

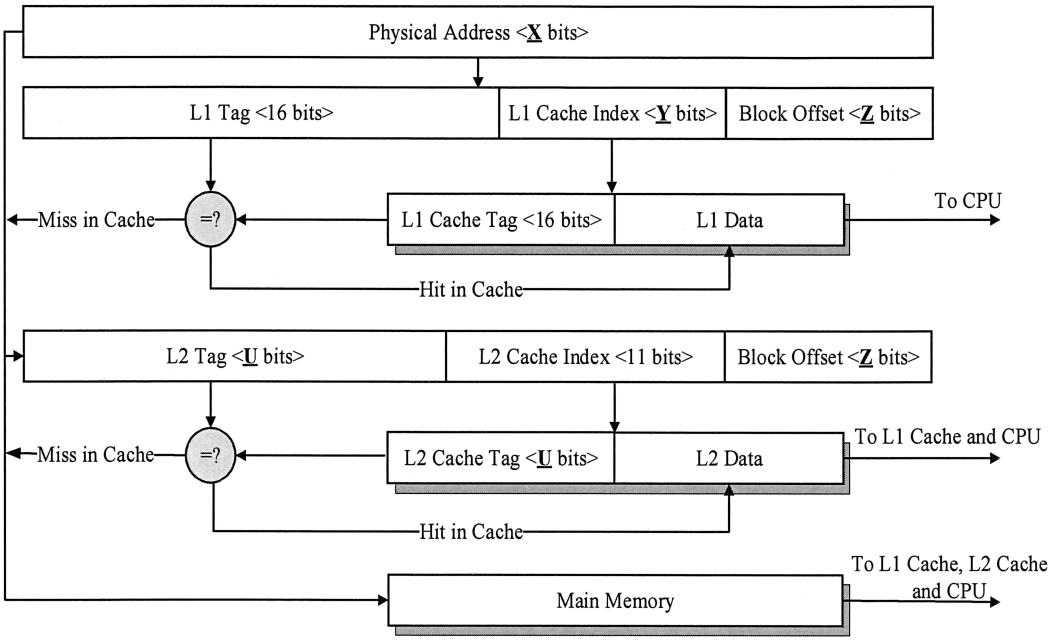


Figure Q3b

(iii) Given that the L2 cache is a 4-way set associative cache in which a cache entry includes a valid bit, a dirty bit, a use bit and two LRU (Least Recently Used) bits in addition to a tag field and a data field, what is the size of the L2 cache?

(6 marks)

(iv) Suppose the miss rates of the L1 and L2 caches are 5% and 1%, respectively. The access times for the L1 cache, the L2 cache and the main memory are 4, 20 and 100 cycles, respectively. What is the average memory access time (AMAT) of the machine A?

(4 marks)

4. (a) Briefly describe how a typical GPU architecture is designed for SIMD/SIMT based parallel programming paradigm.

(5 marks)

Note: Question No. 4 continues on Page 5

(b) The code snippet in Figure Q4a shows a program that uses a CUDA kernel `saxpy()` to compute the SAXPY (Single precision A.X plus Y) operation:

$$\mathbf{Y} = A\mathbf{X} + \mathbf{Y}$$

where `A` is a scalar, **`X`** and **`Y`** are vectors, each consisting of `N` floating-point numbers.

```
Line
1    __global__
2    void saxpy(int n, float a, float *x, float *y) {
3      int i = blockIdx.x * blockDim.x + threadIdx.x;
4      if (i < n)
5         y[i] = a*x[i] + y[i];
6    }
7
8    int main(void) {
9      int N = 4096;          // size of vectors X and Y
10     float A = 3.0;
11     float X[N] = {.....};// initialize vector X
12     float Y[N] = {.....};// initialize vector Y
13     int *d_X, *d_Y;
14     cudaMalloc((void**)&d_X, sizeof(float)*N);
15     cudaMalloc((void**)&d_Y, sizeof(float)*N);
16     cudaMemcpy(d_X, X, sizeof(float)*N, cudaMemcpyHostToDevice);
17     cudaMemcpy(d_Y, Y, sizeof(float)*N, cudaMemcpyHostToDevice);
18     saxpy<<<.....>>>(.....);
.....}
```

Figure Q4a

(i) Complete the code shown in Line ***18*** if the number of threads per block is set as 512.

(5 marks)

(ii) Assume a Stream Multiprocessor (SM) in a GPU has sufficient register and shared memory resources to reside all the blocks. What is the total number of warps that will be created by launching the kernel? Please elaborate with working details.

(4 marks)

Note: Question No. 4 continues on Page 6

(c) Figure Q4b shows a CUDA kernel that runs on a GPU to compute the dot product of two vectors **A** and **B** and outputs a scalar value C:

$$C = \mathbf{A} \cdot \mathbf{B} = \sum_{i=1}^{N} a_i b_i = a_1 b_1 + a_2 b_2 + \cdots + a_N b_N$$

```
Line
1  void dot_prod(int N, int *a, int *b, int *c) {
2    int temp[N];
3    int i = threadIdx.x;
4    temp[i] = a[i]*b[i];
5
6    // Thread 0 sums the pairwise products
7    if (i == 0) {
8      int sum = 0;
9      for (int j = 0; j < N; j++)
10       sum += temp[j];
11     *c = sum;
12   }
13 }
```

Figure Q4b

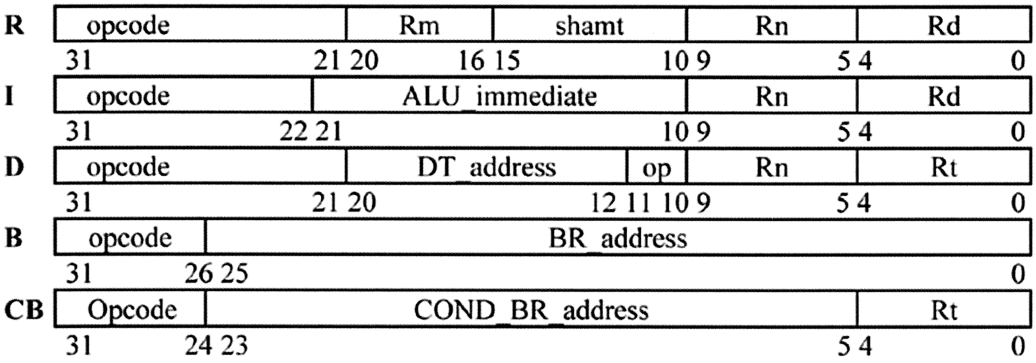
(i) Identify three GPU programming-related mistakes in the CUDA C code in Figure Q4b. Explain and show clearly how they could be fixed.

(6 marks)

(ii) If N = 256, explain how the kernel should be launched in terms of the number of block(s) and thread(s). Give brief justification for your answer.

(5 marks)

Appendix - Instruction Formats



END OF PAPER

CE3001 ADVANCED COMPUTER ARCHITECTURE
CZ3001 ADVANCED COMPUTER ARCHITECTURE

Please read the following instructions carefully:

1. **Please do not turn over the question paper until you are told to do so. Disciplinary action may be taken against you if you do so.**

2. You are not allowed to leave the examination hall unless accompanied by an invigilator. You may raise your hand if you need to communicate with the invigilator.

3. Please write your Matriculation Number on the front of the answer book.

4. Please indicate clearly in the answer book (at the appropriate place) if you are continuing the answer to a question elsewhere in the book.