NANYANG TECHNOLOGICAL UNIVERSITY

SEMESTER 1 EXAMINATION 2023-2024

CE3001/CZ3001/SC3050 – ADVANCED COMPUTER ARCHITECTURE

Nov/Dec 2023 Time Allowed: 2 hours

INSTRUCTIONS

1. This paper contains 4 questions and comprises 6 pages.
2. Answer **ALL** questions.
3. This is a closed-book examination.
4. All questions carry equal marks.
5. The appendix provides the LEGv8 instruction formats.

---

1. (a) A benchmark program P is executed to analyze two computer systems S1 and S2. On system S1, each register type (R-type) instruction takes 4 clock cycles, a load instruction takes 5 clock cycles, and a branch instruction takes 3 clock cycles, respectively. On system S2, each R-type instruction takes 3 clock cycles, a load instruction takes 5 clock cycles, and a branch instruction takes 2 clock cycles, respectively. In both systems, the program P executed 80 million R-type instructions, 40 million load instructions, and 25 million branch instructions.

(i) Find the CPI for system S1 and S2.

(3 marks)

(ii) Assuming that system S2's clock speed is 15% slower than system S1's clock speed, which system is faster in running program P, and what is the speedup?

(6 marks)

Note: Question No. 1 continues on Page 2

(b) If a fraction P of a program is enhanced by a factor of F when executed on an enhanced machine, derive the expression of the speedup over the original machine. Based on that expression, find the speedup value when P=0.4 and F=1000.

(4 marks)

(c) Use a neat diagram to show the datapath of a single-cycle architecture that supports the execution of both "LDUR X2, [X1, #8]" and "ADDI X4, X3, #5". The datapath needs to have minimal number of multiplexers (control signals can be simplified). Briefly explain the working of the instructions "LDUR X2, [X1, #8]" and "ADDI X4, X3, #5".

(12 marks)

2. (a) Listing Q2 shows a code segment that is intended to be executed in a 5-stage pipelined LEGv8 processor. The program counter is updated with the branch target address at the Execute stage. Let the initial values be `X4=0x0000000000010000` and `X5=0x0000000000001000` (`CBNZ`: *branch on not equal to 0*).

Listing Q2

```
I1  loop:  LDUR  X0, [X4,#0]
I2         LDUR  X1, [X5,#0]
I3         SUB   X2, X1, X0
I4         STUR  X2, [X4,#0]
I5         SUBI  X4, X4, #16
I6         SUBI  X5, X5, #8
I7         CBNZ  X5, loop
```

(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 full data forwarding is allowed. Show the forwarded paths and the dependencies. Find the total number of loop iterations.

(9 marks)

(ii) 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 the other way can execute all instructions except load and store. Find the CPI achieved by the superscalar architecture. Note that full data forwarding is allowed.

(8 marks)

Note: Question No. 2 continues on Page 3

(iii) Briefly comment on the methods that can reduce the CPI of the superscalar architecture depicted in Q2(a)(ii).

(3 marks)

(b) Briefly explain the concept of delayed branching with an example. How is this technique used to improve the execution time of a processor?

(5 marks)

3. (a) Consider a Byte-addressable memory with the address space of 32 bits. A 64 KB (1 KB = 1024 Bytes) eight-way set-associative cache is used for this memory system with the cache block size of 256 Bytes. Determine the number of bits for the tag, set index and block offset fields of the address, respectively.

(8 marks)

(b) Tables Q3a and Q3b show the miss rates and average memory access times of a cache system configured in different cache and block sizes.

Table Q3a: Actual Miss Rate v.s. Block Size

Block Size (Bytes)	Cache Size			
	4KB	16KB	64KB	256KB
64	7.00%	2.64%	X%	0.51%
256	9.51%	3.29%	1.15%	0.49%

Table Q3b: Average Memory Access Time v.s. Block Size

Block Size (Bytes)	Cache Size			
	4KB	16KB	64KB	256KB
64	7.16	Y	1.93	Z
256	11.65	4.69	2.29	1.55

(i) Assume that the hit time takes 1 clock cycle, and the memory system takes 80 clock cycles of overhead and then delivers 16 bytes every 2 clock cycles. For example, it can supply 16 bytes in 82 clock cycles, 32 bytes in 84 clock cycles, 64 bytes in 88 clock cycles, and so on. Compute the missing entries X, Y and Z in Tables Q3a and Q3b.

(8 marks)

Note: Question No. 3 continues on Page 4

(ii) Between miss rate and average memory access time, which one is more important to the cache performance? Briefly justify your answer.

(3 marks)

(c) Name two cache replacement algorithms and two cache write policies for cache system management. For the two cache write policies, which one has better performance considering the huge difference in speed between the cache and the main memory in a computer?

(6 marks)

4. (a) Briefly explain why the flow control of a conditional program is managed differently on GPUs and CPUs.

(4 marks)

(b) Answer the following questions based on the CUDA code in Figure Q4a.

```
Line
1   __global__ compute (float *a, float *b, int BLOCKSIZE) {
2     __shared__ s_a[128], s_b[128];
3     /* copy portion of input data into shared memory */
4     s_a[threadIdx.x] = a[blockIdx.x*BLOCKSIZE+threadIdx.x];
5
6     /* Time step loop */
7     for (int t = 0; t<MAX_TIME; t++) {
8       /*alternate inputs and outputs on even/odd time steps*/
9       if (t % 2 == 0) {
10        int boundary = min((blockIdx.x+1)*BLOCKSIZE-1,
11                           blockDim.x*BLOCKSIZE-1,threadIdx+2);
12        s_b[threadIdx.x] = s_a[threadIdx.x] + s_a[boundary];
13      }
14
15      else /* (t%2 == 1) */ {
16        int boundary = min((blockIdx.x+1)*BLOCKSIZE-1,
17                           blockDim.x*BLOCKSIZE-1,threadIdx+2);
18        s_a[threadIdx.x] = s_b[threadIdx.x] + s_b[boundary];
19      }
20    }
21
22    /* Result is in s_b, and must be copied to b */
23    b[blockIdx.x*BLOCKSIZE+threadIdx.x] = s_b[threadIdx.x];
24  }
```

Figure Q4a

Note: Question No. 4 continues on Page 5

(i) Add synchronization to the code in Figure Q4a to derive a correct implementation that has no race conditions. (Hint: You should be able to simply insert __synchthreads() calls without modifying the code.)

(6 marks)

(ii) What is the number of threads in a warp in a typical Nvidia GPU design? 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 as `compute<<8,512>>`?

(5 marks)

(iii) Briefly explain the functions of the `__global__` and `__shared__` specifiers in CUDA programming.

(4 marks)

(c) Consider the following sequence of actual outcomes of a branch instruction (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. Assume that there is only one branch instruction in the program. What is the prediction accuracy for the last 6 occurrences of this branch, i.e., (T, N, T, N, T, N), if the 2-bit branch predictor as shown in Figure Q4b is applied?

(6 marks)

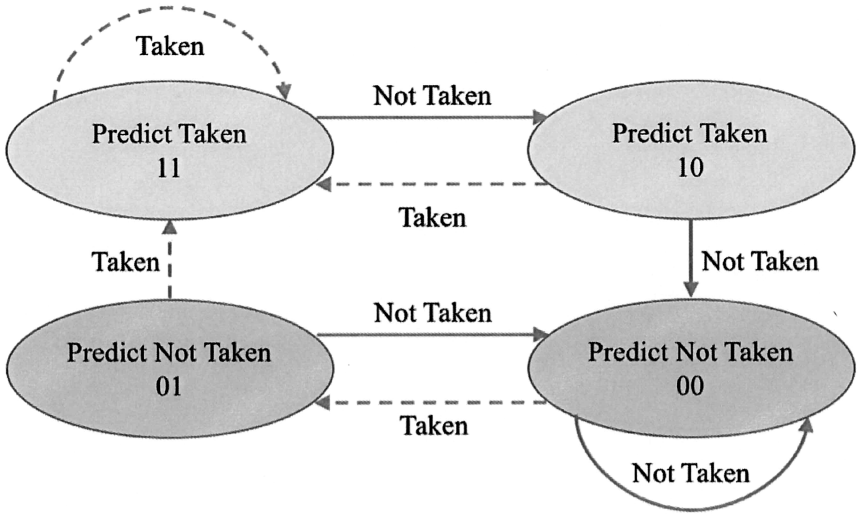
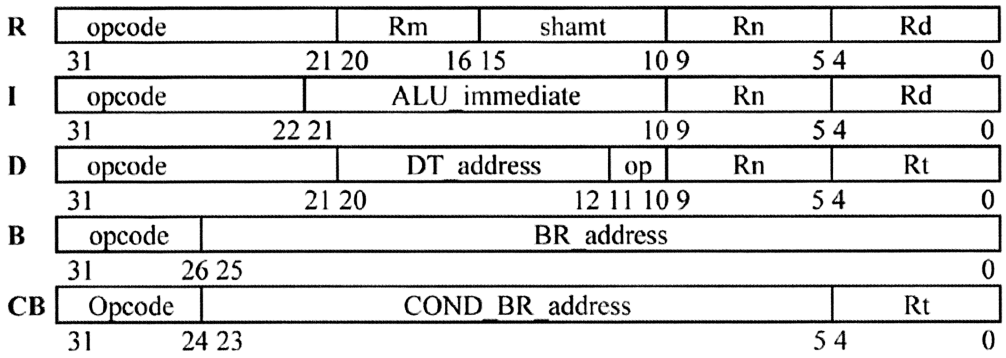


Figure Q4b

Appendix - Instruction Formats



END OF PAPER

CE3001 ADVANCED COMPUTER ARCHITECTURE
CZ3001 ADVANCED COMPUTER ARCHITECTURE
SC3050 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.