Department of Computer Science and Engineering
National Sun Yat-sen University
First Semester of 2023 PhD Qualifying Exam

Subject： Computer Architecture

Name: Student ID:

1. (20% total) The instruction latency for a given CPU is shown in Table 1.

Table 1.

Instructions	Breakdown	Latency
load	5%	3 cycles
add	10%	5 cycles
divide	10%	8 cycles
branch	50%	2 cycles
shift left	15%	5 cycles
shift right	10%	1 cycles

1.1 (5%) Calculate the **CPI** of the given CPU?

1.2 (5%) **Variant 1:** All **add** instructions are replaced with corresponding **subtract** instructions that take the same amount of time. Calculate the **CPI** of the **Variant 1**?

1.3 (5%) **Variant 2:** Remove the highest-latency instruction, and replace all those instructions with additional latency of three cycles for the lowest latency instruction in the mix. Calculate the **CPI** of the **Variant 2** (relative to the original instruction count)?

1.4 (5%) **Variant 3:** Reduce the latency for **divides** by a factor of four, but increase the latencies of **branches** by 50%. Calculate the **CPI** of the **Variant 3**?

2. (20% total) An 8-bit byte-addressable virtual address space and the physical memory has 128 bytes. Each page contains 16 bytes. A simple, one level translation scheme is used and the page table resides in physical memory. The initial contents of the frames of the physical memory are shown in Table 2.

Table 2.

Frame Number	Frame Contents
0	Empty
1	Page 13
2	Page 5
3	Page 2
4	Empty
5	Page 0
6	Empty
7	Page Table

A three-entry translation lookaside buffer (TLB) that uses Least Recently-Used (LRU) replacement is added to this system. Initially, this TLB contains the entries for pages 0, 2, and 13. (Note: LRU is used to select pages for replacement in physical memory.)

References (to pages): 0, 13, 5, 2, 14, 14, 13, 6, 6, 13, 15, 14, 15, 13, 4, 3.

2.1 (6%) What is the hit rate of the TLB for this sequence of references?

2.2 (6%) At the end of this sequence, what three entries are contained in the TLB?

2.3 (8%) What are the contents of the 8 physical frames?

3. (20% total) Assume a machine with a 7-stage pipeline. Assume that branches are resolved in the sixth stage. Assume that 20% of instructions are branches. Answer the following questions.

3.1 (6%) How many instructions of wasted work are there per branch misprediction on this machine?

3.2 (6%) Assume N instructions are on the correct path of a program and assume a branch predictor accuracy of A. Write the equation for the number of instructions that are fetched on this machine in terms of N and A.

3.3 (8%) If the machine were modified so that it used the dual path execution (where an equal number of instructions are fetched from each of the two branch paths). Assume that branches are resolved before new branches are fetched. Write how many instructions would be fetched in this case as a function of N.

4. (20% total) A old program needs to be parallelized. Then, it can run faster on modern multicore processors. In order to execute the program with parallel and serial portions more efficiently, a custom heterogeneous processor needs to be designed.
   - The processor has one large core which executes code more quickly but takes greater die area on-chip, the multiple small cores which execute code more slowly but consume less area, all sharing one processor die.
   - When program in its parallel portion, all of its threads execute only on small cores.
   - When program in its serial portion, the one active thread executes on the large core.
   - Performance (execution speed) of a core is proportional to the square root of its area.
   - Assume 16 units of die area available. A small core takes 1 unit of die area. The large core can take any number of units of die area n^2, where n is the positive number. Area not used by the large core will be filled with smaller cores.
   - The **serial portion** is only **10%** of total work, and the **parallel portion** is the remaining **90%**.

4.1 (10%) What would be the speed up for the fastest possible execution of the program?

4.2 (10%) What would the same program's speedup be if all 16 units of die area were used to build a homogeneous system with 16 small cores, the serial portion ran on one of the small cores, and the parallel portion ran on all 16 small cores?

5. (20%) We define the SIMD utilization of a program that runs on a GPU as the fraction of SIMD lanes that are kept busy with active threads during the run of a program. The SIMD utilization of a program is computed across the complete run of the program. The following code segment is run on a GPU. A warp in the GPU consists of 32 threads, and there are 32 SIMD lanes in the GPU. Each thread executes a single iteration of the shown loop. Assume that the data values of the arrays A and B are already in vector registers so there are no loads and stores in this program. The value of k is constant across all iterations and $0 < k \leq 32$. (Hint: Notice that there are 2 instructions in each iteration. The two comparisons in the **if** statement are executed as a single instruction.)

```
for (i = 0; i < 3072; i++) {
    if   (i % k == 0 || A[i % k] > 0) { // Instruction 1
            B[i] = A[i] + 1;            // Instruction 2
    }
}
```

Please answer the following four questions.

5.1 (10%) How many warps does it take to execute this program?

5.2 (10%) What needs to be true about array A to achieve 100% utilization? Show your work. (Hint: The warp scheduler does not issue instructions where no threads are active).