Department of Computer Science and Engineering
National Sun Yat-sen University
First Semester of 2025 PhD Qualifying Exam

Subject： Computer Architecture

1. (25% total) The instruction mix and latencies of a given CPU are shown in Table 1.

Table 1.

Instructions	Breakdown (instruction mix, relative instruction count)	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 (10%) Calculate the **CPI** of this CPU.

1.2 (15%) Calculate the **CPI** of this revised CPU**:** Reduce the latency of the `divide` instruction by a factor of four, but increase the latency of the `branch` instruction by 50%.

2. (25% 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.

2.1 (8%) How many instructions are wasted per branch misprediction on this machine?

2.2 (8%) 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***.

2.3 (9%) If the machine is modified to use 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***.

3. (25% total) An old program needs to be parallelized so it can run faster on modern multicore processors. In order to execute the program, which has both 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 a positive integer. Area not used by the large core will be filled with smaller cores.
   - The **serial portion** is only **10%** of the program, and the **parallel portion** is the remaining **90%**.

3.1 (10%) What would be the speed up for the fastest possible execution of the program?

3.2 (15%) 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?

4. (25% total) Assume a 32-bit, byte-addressed machine with virtual addressing. The two high-order bits are **11** is treated as **unmapped**. These addresses are only accessible by the operating system and bypass virtual address translation. The answers need to express as a multiple of a power of 2, or in terms of KB, MB, GB, or TB as appropriate.

4.1 (5%) What is the maximum amount of **physical memory** this system can address?

4.2 (5%) What is the maximum amount of **virtual memory** any single process on this system can address?

4.3 (5%) How many **virtual pages** are available to each process, assuming 4KB per page?

4.4 (10%) Assume each page table entry is 4 bytes, how much memory would have a single-level page table require?