國立中山大學109學年度
碩士暨碩士專班招生考試試題

科目名稱：計算機結構【資工系碩士班甲組、乙組】

—作答注意事項—

考試時間：100分鐘

- **考試開始鈴響前不得翻閱試題，並不得書寫、劃記、作答。**請先檢查答案卷（卡）之應考證號碼、桌角號碼、應試科目是否正確，如有不同立即請監試人員處理。
- **答案卷限用藍、黑色筆(含鉛筆)書寫、繪圖或標示**，可攜帶橡皮擦、無色透明無文字墊板、尺規、修正液（帶）、手錶(未附計算器者)。每人每節限使用一份答案卷，不得另攜帶紙張，請衡酌作答。
- 答案卡請以2B鉛筆劃記，不可使用修正液（帶）塗改，未使用2B鉛筆、劃記太輕或污損致光學閱讀機無法辨識答案者，其後果由考生自行負擔。
- 答案卷（卡）應保持清潔完整，不得折疊、破壞或塗改應考證號碼及條碼，亦不得書寫考生姓名、應考證號碼或與答案無關之任何文字或符號。
- 可否使用計算機請依試題資訊內標註為準，如「可以」使用，廠牌、功能不拘，唯不得攜帶具有通訊、記憶或收發等功能或其他有礙試場安寧、考試公平之各類器材、物品（如鬧鈴、行動電話、電子字典等）入場。
- **試題及答案卷（卡）請務必繳回，未繳回者該科成績以零分計算。**
- **試題採雙面列印，考生應注意試題頁數確實作答。**
- **違規者依本校招生考試試場規則及違規處理辦法處理。**

國立中山大學109學年度碩士暨碩士專班招生考試試題

科目名稱：計算機結構【資工系碩士班甲組、乙組】　　題號：434001

※本科目依簡章規定「不可以」使用計算機(問答申論題)　　

1. (10% 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.

1.1 (2%) What is the maximum amount of **physical memory** this system can address?

1.2 (2%) What is the maximum amount of **virtual memory** any single process on this system can address?

1.3 (2%) How many **virtual pages** are available to each process, assuming 4KB pages?

1.4 (2%) Assume each page table entry is 4 bytes, how much memory would a single-level page table require?

1.5 (2%) Assuming a two-level page table where half the bits of the virtual page number are used to index the first level, and the other half are used to index the second level, how many second-level page tables can a process use if its total page table size is limited to 400KB?

2. (15% 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

2.1 (3%) Calculate the **CPI** of the given CPU?

2.2 (4%) **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**?

2.3 (4%) **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)?

2.4 (4%) **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**?

3. (10% 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%**.

3.1 (5%) What would be the speed up for the fastest possible execution of the program?

3.2 (5%) 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. (15% 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.

4.1 (4%) What is the hit rate of the TLB for this sequence of references?
4.2 (3%) At the end of this sequence, what three entries are contained in the TLB?
4.3 (8%) What are the contents of the 8 physical frames?

5. (15% 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.

5.1 (5%) How many instructions of wasted work are there per branch misprediction on this machine?
5.2 (5%) 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.
5.3 (5%) 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.

6. (10% total) A byte-addressable system with 16-bit address ships with a three-way set associative, write-back cache. The cache implements a true LRU replacement policy using the minimum number of replacement policy bits necessary to implement it. The tag store requires a total of 264 bits of storage. What is the block size of the cache?

7. (10% total) A byte-addressable processor is connected to a single memory channel that has a single rank of DRAM. The physical address space is 32 bits, and the processor uses the following mapping shown in Table 3, to index the DRAM. Each DRAM row has a certain number of columns, where a column has the same size of a cache line. The processor uses 64-byte cache lines. The Columns is 6 bits and the Cache Line Offset is also 6 bits. In addition, the row size is 4 KB.

Table 3 Mapping from the physical address to DRAM

MSB			LSB
Rows	Banks	Columns	Cache Line Offset

Table 4 shows the memory request queue that has 4 pending memory request at time 0.
Assuming that

- A row buffer hit takes **50 cycles**
- A row buffer conflict takes **250 cycles**.
- Requests going to different banks can be processed by the banks in parallel.
- All the row buffers are closed at times 0.
- The controller cannot issue two requests at the same time. Each request takes **10 cycles** to process, so it takes **10 cycles** between issuing two separate requests to the memory.
- The controller employs First Ready-First Come First Serve (FR-FCFS) scheduling policy.

Table 4. the state of the memory request queue at time 0.

Request	Physical Address
A (oldest)	0x0000-4000
B	0x0000-1040
C	0x0000-3040
D (youngest)	0x0000-4a00

If it takes **320 cycles** to finish processing all four requests in the memory, **at least** how many banks does this rank of DRAM have?

8. (15% total) Please answer the following questions.

8.1 (4%) Give two disadvantages of static code scheduling.

8.2 (4%) Give two disadvantages of dynamic code scheduling.

8.3 (4%) What are two mechanisms that dynamic code scheduling uses to mitigate the shortcomings of static code scheduling, and why does each mechanism help?

8.4 (3%) What types of dependencies can occur during out-of-order execution?