Computer Science Comprehensive Examination
Computer Architecture
[60 points]

This examination is open book. Please do all of your work on these sheets. Do not do your work in a blue book.

Number: ___________________________________

Problem	Max Score	Your Score
1	30	
2	24	
3	24	
4	22	
TOTAL	100	

The following is a statement of the Stanford University Honor Code:

A. The Honor Code is an undertaking of the students, individually and collectively:

(1) that they will not give or receive aid in examinations; that they will not give or receive unpermitted aid in class work, in the preparation of reports, or in any other work that is to be used by the instructor as the basis of grading;

(2) that they will do their share and take an active part in seeing to it that others as well as themselves uphold the spirit and letter of the Honor Code.

B. The faculty on its part manifests its confidence in the honor of its students by refraining from proctoring examinations and from taking unusual and unreasonable precautions to prevent the forms of dishonesty mentioned above. The faculty will also avoid, as far as practicable, academic procedures that create temptations to violate the Honor Code.

C. While the faculty alone has the right and obligation to set academic requirements, the students and faculty will work together to establish optimal conditions for honorable academic work.

By writing my "magic number" below, I certify that I acknowledge and accept the Honor Code.

Problem 1 : Short Answer [3 points each, 30 points total]

A. Compared to an 8K direct-mapped cache, what type of misses will a 16K direct-mapped cache have fewer of? Circle all that apply.

(a) compulsory
(b) conflict
(c) capacity

B. Which instruction set is better able to express instruction-level parallelism?

(a) An accumulator instruction set
(b) A three-address general-register instruction set

C. Which mean should be used to combine execution speeds expressed in instructions/sec? Circle all that apply.

(a) Arithmetic mean
(b) Geometric mean
(c) Harmonic mean

D. A machine with register renaming is able to reorder instructions without regard to what type of dependencies? Circle all that apply.

(a) Data dependencies
(b) Output dependencies
(c) Anti-dependencies
(d) Control dependencies

E. Adding accurate branch prediction to a processor reduces the impact on performance of which pipeline latency? Circle all that apply. Assume that branch target and branch condition are computed during the execution stage of the pipeline.

(a) From the completion of execution to where the results are available from a register.
(b) From the register stage to where the results of execution are available
(c) From the register stage to where the results of a memory load are available
(d) From the fetch stage to the register stage.

F. In the steady state, what will be the prediction accuracy of a one-bit branch predictor on the repeating sequence TTTTNTTTN (T=taken, N=not taken)?

G. An instruction for protected subsystem entry (sometimes called system call) must change which subset of the following five things atomically? (circle all that apply)

(a) The contents of a data register
(b) The program counter (sometimes called instruction pointer)
(c) The contents of the page table
(d) The contents of the cache
(e) The privilege level

H. Suppose that cache A, a 16K-byte fully-associative cache, cache B, a 16K-byte direct-mapped cache, and cache C, a 4K-byte fully-associative cache are all referenced with an identical address sequence. All caches use a true least-recently used (LRU) replacement policy. Which of the following statements are true: (circle all that apply)

(a) A will contain a superset of the data in B
(b) A will contain a superset of the data in C
(c) B will contain a superset of the data in A
(d) B will contain a superset of the data in C

I. In a 64K-byte four-way set-associative cache with 128-byte blocks, how large is the index field used to address the cache array? (write down the number of bits)

____

J. If the cache of question I is physically tagged and physical addresses are 40-bits long, what is the minimum possible length the tag may be for correct operation? (write down the number of bits)

____

Problem 2 : Pipeline Architecture [24 points total]

Consider a memory-to-memory machine with the following pipeline stages:

IF	Instruciton Fetch	Fetch instructions
SA	Source Address	Compute address of source operands
DF	Data Fetch	Fetch source operands from memory and/or registers
X	Execute	Execute arithmetic operations and compute destination address
DS	Data Store	Store results to memory or register

The destination operand for each instruction may be either a memory location or a register. Also, at most one source operand may be a memory location. The second source operand is always a register. Thus, for each operation type, op, this machine provides the following four types of instructions:

```
RR     R <- R op R          Register to register
MR     R <- R op M          Memory to register
RM     M <- R op R          Register to memory
MM     M <- R op M          Memory to memory
```

A. (8 Points) Assume for now that each pipeline stage operates in one clock cycle, that there are no cache misses, that there is bypassing of registers but not of memory, that there is no memory disambiguation, that a store must complete before a load to the same location can take place, and that there are no resource conflicts. Consider the following instruction sequence:

```
1.     8(R3) <- R4 + R5
2.     R6     <- R2 + 16(R7)
3.     12(R1)<- R6 + 8(R3)
4.     4(R1) <- R6 + 4(R3)
```

Note that because there is no disambiguation, the hardware cannot tell whether 8(R3) (the address formed by adding 8 to the contents of R3) is the same as 16(R7). Draw a pipeline diagram showing this code executing and show all data dependencies. (Hint: Your pipeline diagram should have a row for each instruction and a column for each clock cycle – you may use the table below as a guide or draw freehand. Indicate each dependency with an arrow between the source and destination).

B. (8 Points) Now assume that the hardware does do disambiguation and as soon as the addresses are calculated is able to show that 8(R3) is a different address than 16(R7). Draw the pipeline diagram for this case, again showing dependencies. (If you need to you may also assume that 12(R1), 4(R1), and 4(R3) are also different addresses).

C. (8 Points) Now, starting with the machine of part B, assume that there is only a single memory port that must be shared between the DF and DS stages. Again draw the pipeline diagram for this case showing dependencies and identifying resource conflicts.

3. Virtual memory [24 points total]

Consider a hypothetical machine with one-byte virtual addresses consisting of a 4-bit page field and a 4-bit offset field. The page field of a virtual address references a page table stored in page 0 of physical memory. Each entry of the page table is either the index of the page frame in physical memory that contains the page in question or the constant FF if the page is not in physical memory. At a given point in time the page 0 of physical memory has the following entries (all numbers are in hexadecimal):

```
0: 07                  8: 06
1: 01                  9: 02
2: 03                  A: FF
3: FF                  B: 05
4: 00                  C: FF
5: FF                  D: FF
6: FF                  E: FF
7: FF                  F: 04
```

A. [4 points] What physical address, if any, corresponds to virtual address 17 (hex)?

B. [4 points] What virtual address, if any, corresponds to physical address 47 (hex)?

C. [4 points] What physical address, if any, corresponds to virtual address 77 (hex)?

D. [4 points] Is the mapping from virtual addresses to physical addresses one-to-one? Explain your answer?

E. [8 points] To what virtual address would one write to, and what value should be written to that location to map virtual addresses 30-3F to physical addresses 60-6F?

4. Instruction Issue [22 points total]

Consider the following instruction sequence:

```
1      LD     X,R1
2      LD     Y,R2
3      ADD    R1,R2,R3          ; R3 <- R1 + R2
4      LD     Z,R4
5      ADD    R3,R4,R5          ; R5 <- R3 + R4
6      MUL    R5,3,R6           ; R6 <- R5 * 3
7      ADD    R2,R4,R7          ; R7 <- R2 + R4
```

You may assume that all arithmetic operations have two-cycle latency and all loads have three-cycle latency. That is, the result of an arithmetic (memory) operation is available two (three) cycles after that operation enters the first execution stage of the machine. Also assume that there is full bypassing, that all loads hit in the cache, and that an arbitrary number of loads (and arithmetic operations) can be in flight at a single time. Hint: you need only consider the execution and memory stages of the pipeline to answer this question.

A. [6 points] How many cycles does this sequence take to execute on a single-issue in-order machine? Measure time from the cycle that the first instruction issues (enters the execution stage) to the cycle in which the result of the last instruction is available for use. Show your work.

B. [5 points] If issue order and resources are not limited, what is the shortest time this sequence of instructions could take? Show your work.

C. [5 points] How many cycles does this sequence take to execute on an in-order machine with multiple issue (assume an issue width as wide as you need)? Show your work

D. [6 points] Can you statically reorder the code to give the wide in-order machine of part C the performance bound of part B? If so, show the new ordering.