

**Concordia**  
UNIVERSITY

course: <b>COMP 326/526</b>	section: <b>U and UU</b>
exam: <b>Final examination</b>	# of pages: <b>6 (including cover page)</b>
instructor: <b>H.F. Li and T. Li</b>	
materials allowed: <b>No</b>	
calculators allowed: <b>Nonprogrammable calculator</b>	
special instructions: <ol style="list-style-type: none"><li><b>1. Answer any all questions.</b></li><li><b>2. Maximum mark is 110</b></li><li><b>3. Concise answers will be appreciated.</b></li><li><b>4. Closed book examination.</b></li></ol>	

1. [15] Answer each of the following by indicating if the assertion stated is true or false.

- a. In a uniprocessor system, the CPU time is linearly proportional to the number of memory stall cycles per instruction of the program executed.
- b. Set-associative mapping always outperforms direct mapping in a cache.
- c. Given a choice, it is better to use on-chip area of a processor to hold more registers than to hold more buffers in the form of an on-chip cache.
- d. Pipelining normally reduces the execution time of an instruction.
- e. The longer a pipeline, the more difficult it is to exploit delay branching effectively.
- f. Increasing the number of pipeline stages always increases performance.
- g. Load-store architecture facilitates optimization of pipeline design: pipeline stalls are deterministic except in load/store instructions.
- h. Register windows are used in some RISC processors to reduce procedure call/return overhead.
- i. Fully associative mapping always outperforms set-associative mapping in a cache.
- j. Multi-level cache aims at reducing cache miss penalty.
- k. WAR hazards never introduces a stall in any pipelined processor.
- l. The GCD test is a necessary and sufficient test to detect if a given loop has loop-carried dependency.
- m. The speedup of a pipelined processor [compared with a non-pipelined counterpart] is limited by the number of stages present in the pipeline.
- n. It is possible that a DLXV processor executes a scalar correspondence of a given vector program in the same amount of time.
- o. A multiple-issue processor is better than a corresponding VLIW processor in the compactness of its program code but usually has an inferior performance.

2. Consider the following DLX program loop:

```

loop:  LD      F0, 0(R1)
       LD      F4, 0(R2)
       MULTD   F0, F0, F4
       ADDD    F2, F0, F2
       SUBI    R1, R1, #8
       SUBI    R2, R2, #8
       BNEQZ   R1, loop
    
```

a. [9] Assume the program is executed on a DLX processor with one-cycle delayed branch and whose pipeline hazard stalls are shown below.

Instruction producing result	Instruction using result	Stalls
FP ALU op	Another FP ALU op	3
FP ALU op	Store double	2
Load Double	FP ALU op	1
Load Double	Store Double	0

- (i) Identify all potential hazards among the instructions of each iteration.
  - (ii) Unroll the loop a minimum number of times so that it could be scheduled without any stalls. Determine the execution time assuming R1 initially contains 512.
- b. [8] Rewrite the scalar code into vector code and determine its execution time on a DLXV processor with chaining. DLXV vector instructions are shown in **Figure 1**. Assume the following latency parameters for DLXV. State other assumptions you made.

Unit	Start-up Overhead
Load and Store	12
Multiply	7
ADD	6

- c. [6] Compare the cost and performance tradeoff between the scoreboard and CDB [Tomasulo's algorithm]. Which of these two methods is more scalable, i.e., could accommodate more efficiently increases in the number of functional units?
  - d. [5] Explain concisely (i) restartable computer, and (ii) precise interrupt.
- 3.
- a. [5] Compare and contrast the key ideas behind (i) VLIW, (ii) loop unrolling, (iii) software pipelining, and (iv) vector processor. Identify any common ground between any of them as well as their differences in realization.
  - b. [4] What is the relationship between data hazards and data dependency? Identify their similarities and differences, if any.
  - c. [5] Describe a major problem confronting multiprocessor system design. Discuss how architecture and programming paradigm have to be developed together to address this problem.
- 4.
- a. [7] For each of the following network topologies (i) 4 x 4 mesh, (ii) 4 x 4 torus and (iii) a hypercube with 16 nodes, decide on the number of time units required to do a multinode broadcast: each node broadcast a same message to all other nodes. Describe concisely how the broadcast is scheduled. Assume that every link has the same speed. State other assumptions made.
  - b. [7] A 128-node multicomputer connected by a hypercube interconnection network is given. The interconnection network uses wormhole routing, with each switch buffer being 8 bits per port. A fixed size packet carrying 24 byte payload is used. (i) what is the minimum header size? (ii) what is the minimum transfer delay for a packet? (iii) what is the minimum transfer delay for a message containing 10 packets? Assume each switch takes 0.5 microsecond and the transfer rate of a link is 20 Mbyte/sec.

5.

a. [9] Consider the following program sequence:

```
A = ...
B = ...
.. = A
.. = C
acquire (S)
D = ...
C = ...
.. = A
E = ...
.. = C
release (S)
E = ...
.. = C
A = ...
```

Identify clearly the program orders that should be enforced under each of the following shared memory models:

- (i) sequential consistency
- (ii) weak consistency
- (iii) release consistency

[Hint: do not ignore data dependency]

b. [6] Consider the following concurrent processes:

```
Process 1: SD 0(R1), F4
           SD 0(R2), F6
Process 2: LD 0(R1), F2
           LD 0(R2), F8
```

Can the two instructions in each process be executed in arbitrary program order while guaranteeing sequential consistency? Why? Assume R1 contains the same value in both processes, and so does R2.

6. Shown in Figure 2 is the state transition diagram of a write-invalidate coherence protocol of a snooping bus.

A two-processor system executing the following program is analyzed.

```
Process 1: loop: LW   R2, 1024(R1)
                 ACQ  108(R2)
                 LW   R3, 112(R2)
                 ADDI R3, R3, #1
                 SW   112(R2), R3
                 REL  108(R2)
                 SUBI R4, R1, #4
                 BNEZ R4, loop
```

```

Process 2: loop: LW   R2, 2048(R1)
                  ACQ  108(R2)
                  LW   R3, 112(R2)
                  ADDI R3, R3, #1
                  SW   112(R2), R3
                  REL  108(R2)
                  SUBI R4, R1, #4
                  BNEZ R4, loop

```

Assume initially R1 in both processors contain 64, ACQ is an acquire (lock) instruction executed atomically at the memory location identified by the register specified while REL (unlock) is the corresponding release instruction.

- a. [7] Each processor has a 4K byte write-back cache managed by direct mapping. Each block of the cache contains 32 bytes. A cache hit takes one cycle. An invalidate broadcast on the snooping bus takes one cycle. A read transfer from another cache takes another cycle. The main memory is interleaved 8 ways and each memory access takes 8 cycles. The memory bus is time staggered one cycle apart among the 8 memory banks. Derive the effective memory access time of this multiple cache memory system. Assume (i) miss rate is 1%, (ii) 20% accesses are writes, (iii) each processor executes independent programs, and (iv) 50% the cache blocks are dirty.
- b. [4] Repeat a. by assuming instead (i) each cache block is in the invalid state 5% time, shared state 30% time, exclusive read/write state 65% time, (ii) 20% accesses are writes, and (iii) 50% of the cache blocks are dirty.
- c. [4] Assuming an execution CPI of 1.5, derive the speedup of the multiple cache system in b. when compared with one without the caches. You may also ignore memory conflicts in the latter case.
- d. [6] Now consider the concurrent execution of the given program above. Suppose the processors are the standard 5-phase DLX processors with forwarding and they start execution at the same time. With the same architecture assumptions given in a., trace the execution of the first two iterations of the program loop, identifying clearly all the activities related to the cache coherence protocol. Assume initially memory locations starting from 1088 contain 4, 100, ... and those starting from 2112 contain 8, 100, ... and so on. State other assumptions made.
- e. [3] Estimate the execution time of the program by assuming instead initially memory locations starting from 1088 contain 4,4,4,4,... and those starting from 2112 contain 8,8,8,8,... and so on.

### INSTRUCTIONS

1. Answer any FIVE of the SEVEN Questions ( $5 \times 22 = 110$  marks).
2. Indicate on top-right corner of each page the question you are answering in that page. *Failure to do so may result in losing some marks.*
3. Concise Answers will be appreciated.

#### 1. Cache Memory

- (a) [4 marks] Explain clearly the differences in the use of registers and cache. Why one could not replace the other?
- (b) Consider a memory system with a 2-way set associative cache. The cache miss penalty for  $k$ -word blocks is  $(6+k)$  cycles and the following are the cache miss rates for various block sizes.

Block Size	Miss Rate per reference
1 words	4%
2 words	2.2%
4 words	1.5%
8 words	1%
16 words	0.8%

- (i) [3 marks] Find the average memory access time when the block size is 4 words assuming the cost of a reference that hits 0.
- (ii) [3 marks] Find the optimal block size.
- (iii) [3 marks] For a machine whose base CPI for execution is 1.4 and an 8-word block, how much faster would be the machine if all cache references are hits?
- (iv) [3 marks] Suppose we make our 2-way set associative cache into a direct-mapped cache with 8-word blocks. The new miss rate is 1.5%. How much faster must the clock be to equal the performance of the original machine with 8-word blocks?
- (c) [6 marks] A shared-memory multiprocessor uses the following cache coherence protocol:

"When a processor wishes to read a variable, the read request is broadcast to all other caches. The processor then waits until it has received acknowledges from everyone and picks up the value returned last among the acknowledges. When a processor wishes to write a variable, it simply updates its local copy, if there is one; if not, it writes to the home copy."

Is the above protocol correct? If it is correct, explain why it is not a very useful one in general. If it is incorrect, provide a counter example to show its fallacy.

#### 2. Microprogramming

- (a) [4 marks] Microinstructions are said to be *atomic*, meaning they cannot be interrupted. Briefly explain why microinstructions should be atomic.

- (b) [4 marks] What is a restartable computer? Does the microprogram for DLX make it restartable? Why or why not?
- (c) [6 marks] A complex machine instruction can be implemented either in microcode (as a microroutine) or in macrocode (as a procedure). What is the major advantage of a microcode implementation of a complex instruction? What is the major disadvantage of the same?
- (d) [8 marks] One advantage of microcode is that it can handle rare cases without having the overhead of invoking the operating system before executing the trap routine. Suppose a machine with a CPI of 1.5 has an operating system that takes 100 clock cycles on a trap before it can execute the appropriate code. Suppose the trap code takes 10 clock cycles whether it is microcode or macrocode. For an instruction occurring 5% of the time, what percentage of the time must it trap before a microcode implementation is 1% faster overall than a macrocode implementation?

### 3. Pipelined Processing

Consider the following pipeline for a high speed version of DLX.

IF1	Instruction fetch starts
IF2	Instruction fetch completes
ID	Instruction decode and register fetch; begin computing branch target
EX1	Execution starts; branch condition tested this cycle; branch target computation completed
EX2	Execution completes — effective address or ALU result available
MEM1/ALUWB	First part of memory cycle plus WB of ALU operation
MEM2	Memory access completes
LWB	Write back for a load instruction

The ALU used during EX and the memory system are both pipelined. As in the standard DLX pipeline, assume register writes are in the first half of a cycle and reads are in the second half.

- (a) [6 marks] For the following code sequence show all possible forwarding requirements (not stalls). Indicate the forwarding requirements by an arrow from the source to the destination.

```

LW    R1, 0(R2)
ADD   R2, R3, R4
SW    0(R3), R1
SW    0(R4), R1
SW    0(R2), R1

```

- (b) [5 marks] For the above code sequence, show all required stalls (rather than forwards).
- (c) [5 marks] Assuming a predict not taken strategy, find the branch penalty for a taken and untaken branch. Assume that a predicted instruction can be executed up to, but not including, a pipestage that does a write back.
- (d) [6 marks] Assume the frequency of various exceptions per instruction is as follows:

Instruction TLB miss	5%
Data TLB miss	10%

Assume these are recognized during the IF1 and MEM1 stages and each take 10 clock cycles to handle in operating system software. If the interrupts are precise and the CPI without these exceptions is 2, how much faster is the machine when these exceptions do not occur?

#### 4. Vector Processing

(a) The execution time of a vectorized program loop is modeled by the following formula:

$$T_n = T_{\text{base}} + \left\lceil \frac{n}{\text{MVL}} \right\rceil * (T_{\text{loop}} + T_{\text{start}}) + n * T_{\text{element}}$$

where  $n$  is the length of the vector.

- (i) [3 marks] Define  $T_{\text{base}}$ ,  $T_{\text{loop}}$ ,  $T_{\text{start}}$ , and  $T_{\text{element}}$  concisely.
  - (ii) [3 marks] Which of the parameters in (i) are more or less fixed for a given compiler running on DLX? Why?
  - (iii) [3 marks] What is the effect of vector chaining on any of these parameters?
  - (iv) [3 marks] What is the effect of vector strides on any of these parameters?
  - (v) [3 marks] What is the effect of the number of memory pipes on any of these parameters?
- (b) [7 marks] Consider the following program loop:

```
FOR i:=1 TO 64 DO
  A[i] := A[2*i] * B[i-1]
```

Can the above program be converted into vector code? If so, write the DLXV code for it. If not, justify why not.

#### 5. Parallel Processing

A recent shared memory multi-threaded multiprocessor has been designed with the following characteristics.

Number of processors:	256
Number of memory nodes:	256
Interconnection network:	3-D mesh with 4K nodes
Processor design:	superpipeline, runs superscalar instructions and consists of 70 segments.
Register File:	4K registers.
Process Contexts:	128.
Clock Rate:	400 MHz.

- (a) [5 marks] State the fundamental problems facing shared memory multiprocessing.
- (b) [6 marks] The architect of this machine has decided against the use of cache. What could have prompted him to make such a decision? Which features mentioned in the above represent an alternate solution to the use of cache?
- (c) [6 marks] What is the purpose of using a 3D-mesh containing more nodes than the total number of processors and memories? What could have been the functions performed at the rest of the nodes, assuming that a small subset of these nodes are used to attach the 256 processors and memories?
- (d) [5 marks] Describe a potential major problem that can arise in this design which is so heavily pipelined.

## 6. Parallel Programming

Consider the following parallel program (written in multi-Pascal) and assume the following runtime parameters:

Time to create a thread: 1 millisecond  
 Time to perform VectorProduct(i,j):  $10n$  microseconds

```
PROGRAM ParallelMatrixMultiply;
  CONST n=5;
  VAR  A,B,C: ARRAY [1..n, 1..n] OF REAL;
        i,j : integer;

  PROCEDURE VectorProduct(i,j: INTEGER);
  VAR  sum: REAL;
        k: INTEGER;
  BEGIN
    sum:=0;
    FOR k:=1 TO n DO
      sum:=sum+A[i,k]*B[k,j];
    C[i,j]:=sum;
  END;

  BEGIN
    FORALL i:=1 TO n DO
      FORALL j:=1 TO n DO
        VectorProduct(i,j);
      END.
  END.
```

- [5 marks] Estimate the running time of the given program, ignoring memory access conflicts and matrix initiation time.
- [5 marks] Estimate the speedup caused by parallel processing ( $n = 5$ ).
- [6 marks] For arbitrary values of  $n$ , determine the speedup caused by parallel processing. Hence or otherwise, determine the value of  $n$  which maximizes the speedup.
- [6 marks] If memory conflicts arise because different processors try to access common variables, what effect is it likely to produce to the time to perform VectorProduct? (how would the formula  $10n$  be changed?)

## 7. General Questions

For each of the following assertions, briefly explain whether it is correct or not giving precise justifications. (Do not write more than fifty words for each.)

- [3 marks] For a given program, Amdahl's law asserts that the efficiency of the processors will eventually drop to zero when the number of processors becomes very large.
- [3 marks] Cache memory will not improve the performance of a vector processor.
- [3 marks] Wormhole routing is much faster than store-and-forward message routing.

- 
- (d) [4 marks] Tomasulo's algorithm removes all stalls (bubbles) in processing instructions that contain data hazards.
  - (e) [3 marks] An advantage of memory-memory architectures (over load/store architectures) is simple program design and a disadvantage is optimizing processor is more difficult.
  - (f) [3 marks] If a loop fails the GCD test, then loop-carried dependence must exist.
  - (g) [3 marks] A fully pipelined processor must be used with separate instruction and data caches.

**\*\*\* END OF EXAMINATION \*\*\***

CONCORDIA UNIVERSITY  
DEPARTMENT OF COMPUTER SCIENCE

COMP 326/526

Winter 1994

FINAL

---

**Instruction:** Answer all the questions. Total points = 55.

1. (12 points) Using the DLXV vector instruction set, write quality programs for
  - (a) the dot product of two vectors, and
  - (b) the matrix transpose.

You may assume that the lengths of the vectors are less than 64, the maximum size of a vector register.

2. (10 points) Consider the following loop in FORTRAN-like notation as used in the text.

```
DO I=1,50
    MM(I) = NN(I) * PP(I)
    DO J=1,50
        QQ(J) = MM(I-1) - SS(J-1)
        SS(J) = QQ(J-1) * X
    DO K=1,50
        RR(K) = TT(I-1) + Y
        TT(I) = MM(I) ** 2
```

where  $X^{**2}$  denotes the square of  $X$ .

- (a) Find out all the data dependences, antidependences, and output dependences.
  - (b) Apply as many vectorizing transformations as you can on this program and show the program resulting from each transformation.
3. (10 points) Suppose a processor has a 16 MHz clock rate and a CPI of 8 (not counting the memory access time). Suppose memory access time is 100 ns (nanoseconds). There is no cache in the memory hierarchy. Furthermore, suppose on the average two operands are fetched from the memory for each instruction.
  - (a) What is the CPI when memory access behavior is taken into account?
  - (b) Suppose the memory access time is 150 ns. What is the CPI when memory access behavior is taken into account?
4. (4 points) Following are some general questions regarding vector processing.
  - (a) Explain why the displacement addressing mode is not normally used in vector instructions.

- (b) Explain what is the GCD test in vectorizing compilation.
5. (3.5 points) Which ones of the following are not *directly* related to the design of a data cache memory?
- (a) hit ratio
  - (b) placement policy
  - (c) least recently used policy
  - (d) translation lookaside buffer
  - (e) coherence
  - (f) page fault
  - (g) capacity miss
6. (8 points) Consider a cache (M1) and main memory (M2) hierarchy with the following characteristics:
- M1: 16K words, 50 ns access time.
  - M2: 1M words, 400ns access time.

Assume eight-word cache blocks and a set size of 256 words with set-associative mapping; the time to transfer a block of data between the cache and the main memory is 2000ns; and all of the memory accesses are memory reads.

- (a) Show the mapping scheme between M2 and M1.
  - (b) Calculate the effective memory-access time with a cache hit ratio of 0.95.
7. (3.5 points) Which ones of the following are *directly* employed in controller design?
- (a) floating point adder pipeline.
  - (b) hardwired control
  - (c) horizontal encoding
  - (d) microprogramming
  - (e) instruction register
  - (f) interlock mechanism
  - (g) delay slot
8. (4 points) Why the DLX microprogrammed controller used three decode tables (decode 1-3)? Explain the advantages and disadvantages of using three decode tables.