Computer Science Department
Stanford University
Comprehensive Examination in Computer Architecture
Autumn 1989

November 29, 1989

READ THIS FIRST!

1. You should write your answers for this part of the Comprehensive Examination in a BLUE BOOK. Be sure to write your MAGIC NUMBER on the cover of every blue book that you use.

2. Be sure you have all the pages of this exam. There are 3 pages.

3. The number of POINTS for each problem indicates how elaborate an answer is expected. For example, an essay-type question worth 6 points or less doesn't deserve an extremely detailed answer, even though a person can expound at length on just about any topic in computer science.

4. The total number of points is 60, and the exam takes 60 minutes. This "coincidence" can help you plan your time.

5. This exam is CLOSED BOOK. You may not use any notes, calculators, computers, or outside help.

6. Show your work, since PARTIAL CREDIT will be given for incomplete answers. For example, you can get credit for making a reasonable start on a problem even if the idea doesn't work out; you can also get credit for realizing that certain approaches are incorrect. On a true/false question, you might get partial credit for explaining why you think something is true when it is actually false. But no partial credit can be given if you write nothing.

Comprehensive: Computer Architecture (60 points) Autumn 1989

Problem 1 (17 points). Assume that we are given a 32-bit, pipelined processor. This processor is register-register (load/store) machine. The processor is configured to always stall the next instruction to execute if the current instruction takes more than a single cycle to execute. The following table gives the execution cycle counts and frequency distribution of different operations within a Floating-Point (FP) benchmark.

Operation	CPI	Frequency
Call/Return	2	5%
Branches	2	20%
ALU	1	45%
Load/Store	2	25%
FP Operation	5	5%

1a. (2 points). What is the CPI for this benchmark assuming a perfect memory system?

1b. (3 points). Now assume that we remove the hardware logic that automatically inserted the stall cycles for branches, calls/returns, and load/store operations. Now our compiler must insert NOP instructions in the delay slots if the delay slots cannot be filled with useful instructions. We find that the compiler can fill 70% of the branch slots, 100% of the call slots, and 40% of the load/store slots with useful instructions. What is the new CPI still assuming a perfect memory system?

1c. (4 points). The FP engineers say that they can increase the speed of the FP operations by 20%. What is the fraction of execution time spent doing FP operations in the original machine? Also, what speedup results from adding these FP improvements to our original machine?

1d. (3 points). Now the memory architects, not to be outdone, claim that they can also speedup the double-precision FP programs by implementing load double (LD) and store double (SD) instructions. These instructions replace two load word (LW) or store word (SW) instructions each, but they take 3 cycles to execute instead of 2. If we can replace 20% of the LW/SW instructions in our benchmark with LD/SD instructions, how much faster will the original machine run with just this enhancement?

1e. (5 points). Finally, a customer wants to know how much the CPI increases for the original machine if we use a split instruction and data caches. The caches are Write-Back (copy-back) and have a miss rate of 2% and 4% respectively. Assume that the miss penalty and the time to write a cache block to main memory are both 10 cycles. In addition, 40% of the cache blocks are dirty at any one time.

Problem 2 (13 points). You are given a computer with a 48-bit virtual address and a 32-bit physical address. The processor uses a 16KB unified cache with a 16B line.

2a. (8 points). What is the smallest page size you can use to allow overlapped TLB and cache access if the cache is direct-mapped? If the cache is 4-way set associative? Also give the number of bits of physical address used for cache tag, index, and offset in each case.

2b. (5 points). If you buy 32MB of main memory with this machine, and you are told the machine has the following characteristics:

CPI = 1.6 cycles
1.3 references/instruction are sent to TLB
bus can transfer 4B/cycle
page fault every 100 * (number of pages in memory) references

how much bus bandwidth is used for the page transfer from disks to main memory by each page size in part (2a)?

Problem 3 (4 points). What is the Boolean function, expressed as a sum of products, for the following Karnaugh map?

AB \ *CD*	00	01	11	10
00	1	0	0	1
01	1	0	1	1
11	1	1	1	1
10	1	1	1	1

Problem 4 (4 points). Synthesize the following equation using only two-input nand gates. (All signals are active high).

$$A = \overline{(B \cdot C) + (D \cdot E)}.$$

Problem 5 (22 points). Build a specialized processor that evaluates the following polynomial:

$$A_6 x^6 + A_4 x^4 + A_2 x^2 + A_0 .$$

The expression can be calculated using Horner's rule:

$$(((A_6)x^2 + A_4)x^2 + A_2)x^2 + A_0 .$$

You can use multipliers, adders, registers, muxes and any other logic gates. The specification of the computation is given by the following RTL (register transfer level) description.

Note on RTL notation: operations separated by commas (,) are executed in the same clock; operations in different clocks are separated by semi-colons (;).

Input registers	X (64-bit), RESET (1-bit)
Output registers	P (64-bit)
Internal registers	XSQ, A6, A4, A2, A0 (all 64-bit)

```
S0:  A6 <- X;
S1:  A4 <- X;
S2:  A2 <- X;
S3:  A0 <- X;
S4:  XSQ <- X*X, SUM <- A6;
S5:  SUM <- SUM*XSQ+A4;
S6:  SUM <- SUM*XSQ+A2;
S7:  P <- SUM*XSQ+A0, if RESET then goto S0 else goto S4;
```

5a. (4 points). What is the smallest number of buses needed?

5b. (7 points). Draw a logic diagram for the data processing unit (also known as the data path).

5c. (7 points). Design a hardwired control unit for the machine. Show only the logic to implement the state transitions. (You do not need to show how the control signals are generated.)

5d. (4 points). Describe how you would modify the data processing unit and the control unit of the machine if the polynomial is of degree 100:

$$A_{100} x^{100} + A_{98} x^{98} + \cdots + A_0 .$$