

Name _____

Computer Architecture
EE 4720
Practice Final Examination
10 May 1997, 02:00–04:00 CDT :-)

Problem 1 _____ (25 pts)

Problem 2 _____ (25 pts)

Problem 3 _____ (25 pts)

Problem 4 _____ (25 pts)

Alias _____

Exam Total _____ (100 pts)

Good Luck!

Problem 1: Systems A and B are DLX implementations using a branch history table (as described in class) for branch prediction. System A uses a 1024-entry, 2-bit BHT ($h = 10, n = 2$) and system B uses a 512-entry, 4-bit BHT ($h = 9, n = 4$).

(a) Write a program (in DLX assembler) which has more correctly predicted branches when run on A than when run on B . (*Hint: the answer would be the same if $n = 2$ in system B .*) (12 pts)

(b) Write a program (in DLX assembler) which has more correctly predicted branches when run on B than when run on A . (*Hint: use a random number generator or read a prepared table of random numbers. The answer would be the same if $h = 10$ in system B .*) (13 pts)

Problem 2: An ISA has variable-length instructions. The length of each instruction in bytes is given by an integer in a three-bit field, called the *length field*, in the instruction. A 1 in the length field indicates a 1-byte instruction, a 2 indicates a 2-byte instruction, etc. To encourage careful programming, a zero in the length field initiates the processor's warantee-voiding, self-destruct mechanism¹.

(a) Where should the length field be placed? (10 pts)

(b) The first two stages of an implementation of this ISA are called IF and ID, and as one would expect instructions are fetched in IF and decoded in ID. The pipeline is wide enough to pass the largest instruction. (That is, whether 1 byte or 7 bytes or anything in between, an instruction passes through each stage in one cycle just as in the DLX implementations.) Design instruction fetch hardware which can fetch one instruction per cycle. The design should include the PC, memory port used for instructions, IF/ID latch, etc. The memory system supports unaligned access and handles 64-bit wide data. Do not include hardware for branches. The address must be presented to the memory port in the beginning of the cycle and an instruction can be decoded no sooner than the cycle after it is fetched. (*Hint: It's no coincidence that memory is fetched 64-bits at a time even though the largest instruction is 56 bits.*) (For reduced credit, the design can present an address to the memory in the middle of a cycle or can decode an instruction in IF.) (15 pts)

¹ Spacecraft in many science fiction movies have this feature, why not a computer? If you prefer, "The behavior of an instruction with a zero in the length field is not defined."

Problem 3: A system has a 32-bit address space (32-bit addresses), is limited to word-aligned accesses, and uses a 4-way set-associative cache having 1024 sets, a block size of 64 bytes, and LRU replacement.

(a) What is the cache size in bytes? Indicate the bit positions in a memory address that specify the tag, index, and offset. (5 pts)

(b) The code fragments below run on the system. Assume that the only memory accesses performed are to load the array values as shown. The size of an element of `a` is one word (four bytes).

Assuming the cache were cold (no data was cached) when the fragments started, find the hit ratio for each program. Also explain how the hit ratio would be different if random replacement were used, but do not compute a number for random replacement. (20 pts)

```
for( i=0; i<400; i++) total += a[i];
```

```
for( j=0; j<2; j++)
  for( i=0; i<400; i++) total += a[i];
```

```
for( j=0; j<2; j++)
  for( i=0; i<400; i++) total += a[i] + a[ i + 0x08000000 ];
```

```
for( j=0; j<2 j++)
  for( i=0; i<16; i++) total += a[ i * 0x08000000 ]; /* It's a big array. */
```

Problem 4: Answer each question below and on the next page. Be brief and to the point.

(a) Find the CPI of the machine executing the familiar loop shown below. Assume a large number of iterations. The time scale is repeated for readability.(7 pts)

```

muld f10, f20, f22
addd f22, f10, f10
muld f24, f10, f26
subd f0, f10, f30
loop:
ld f2, 0(r2)
subd f0, f0, f2
ld f2, 8(r2)
subd f0, f0, f2
ld f2, 16(r2)
subd f0, f0, f2
subi r3, r3, #1
addi r2, r2, #24
bnez r3, loop

```

```

Time 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28
muld IF ID 5:A1 5:A2 5:A5 5:A4 5:WB
addd IF ID 3:RS 3:RS 3:RS 3:RS 3:A1 3:A2 3:A3 3:A4 3:WB
muld IF ID 6:RS 6:RS 6:RS 6:M1 6:M2 6:M3 6:M4 6:WB 6:WB
subd IF ID 4:RS 4:RS 4:RS 4:A1 4:A2 4:A3 4:A4 4:WB 4:WB
<b>loop:</b>
Time 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28
ld IF ID 2:EX 2:MI 2:WI IF ID
subd IF ID 3:RS 3:RS 3:A1 3:A2 3:A3 3:A4 3:WB
ld ID 1:EX 1:MI 1:WB
subd IF ID 4:RS 4:RS 4:A1 4:A2 4:A3 4:A4 4:WB
ld IF ID 2:EX 2:MI 2:WB
subd IF ID 3:RS 3:A1 3:A2 3:A3 3:A4 3:WB
subi ID 1:EX 1:MI 1:WB
addi IF ID 2:EX 2:MI 2:WB
bnez IF ID 1:EX 1:MI 1:WB
Time 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28

```

(b) Explain the advantages and disadvantages of small page sizes in virtual memory systems. (6 pts)

(c) Explain what might go wrong on a system without precise exceptions running code that expects precise exceptions. Illustrate with a code example. (6 pts)

(d) In some ISAs CTIs have a delay slot. How does this feature help? Considering that ISA lifetimes are measured in decades, why are delay slots a bad idea? (6 pts)