Name ______________________________

Computer Architecture
EE 4720
Final Examination

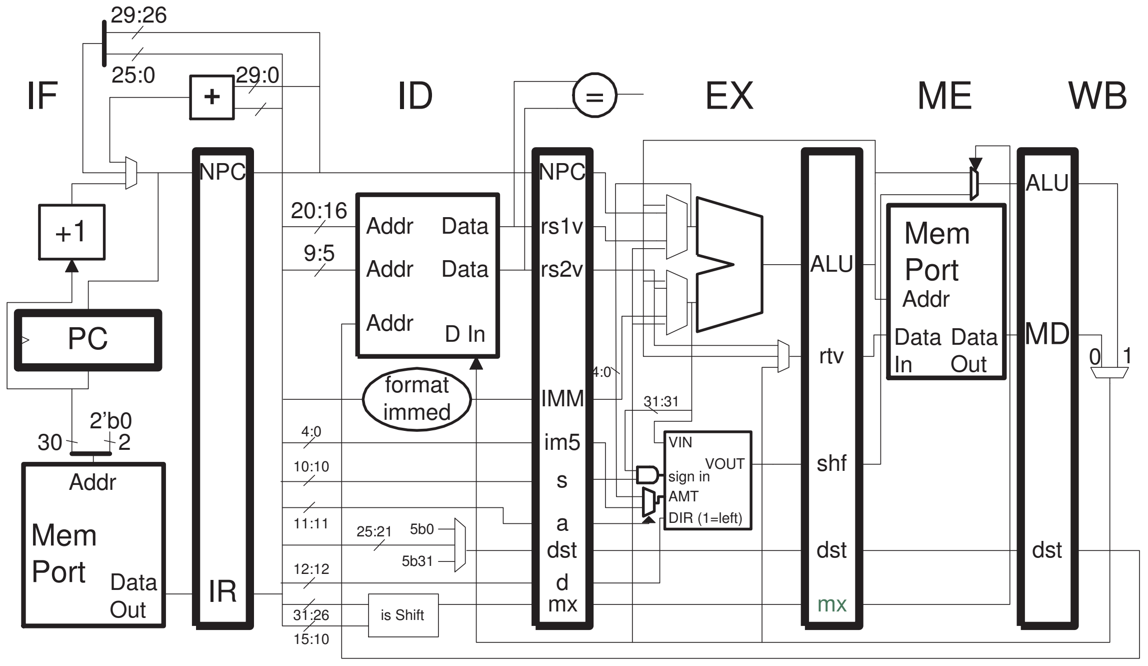
6 December 2010, 7:30–9:30 CST

Problem 1	________	(10 pts)
Problem 2	________	(14 pts)
Problem 3	________	(14 pts)
Problem 4	________	(14 pts)
Problem 5	________	(15 pts)
Problem 6	________	(15 pts)
Problem 7	________	(18 pts)
Exam Total	________	(100 pts)

Alias ______________________________

Good Luck!

Problem 1: (10 pts) The diagram below shows a 5-stage pipeline that looks alot like our familiar MIPS implementation but it's actually an implementation of ISA *X*. (The diagram is based on the solution to Homework 3, in which a shift unit was added to MIPS.)



(*a*) ISA *X* instruction format T encodes the shift instructions and others, it is the equivalent of format R in MIPS. Based on the diagram above show the encoding for ISA *X* format T.

☐ Format T encoding, including bit positions and field names.

Encoding:

31 0

(*b*) Consider the shift instructions `sll`, `sllv`, `srl`, `srlv`, `sra`, and `srav`. Suppose that the encoding of one of these instructions is zero (meaning that every field value is zero). Show the opcode field value(s) for each of these instructions based on the diagram above. *Hint: The control signal for each top mux input is 0, etc.*

☐ Opcode field value(s) for: `sll`, `sllv`, `srl`, `srlv`, `sra`, and `srav`.

(*c*) Explain why the implementation of instructions such as `sw r1,2(r3)` and `beq r1, r2 TARG` would be less elegant for ISA *X* than for MIPS. *Hint: It has something to do with registers.*

☐ `sw` and `beq` less elegant because...

Problem 2: (14 pts) Answer the questions below.

(*a*) Complete the execution diagram for the MIPS code below on a two-way superscalar statically scheduled implementation of the type described in class.

```
add r1, r2, r3

add r4, r5, r6

add r7, r4, r8

lw  r9, 0(r7)

addi r1, r9, 3

xor r10, r11, r12
```

☐ Complete the diagram above.

(*b*) Show the execution of the code below on an 8-way superscalar statically scheduled processor of the type described in class. Branches are not predicted. Find the CPI for a large number of iterations.

```
LOOP: # Address of first insn is 0x1000

and r1, r1, r5

add r3, r3, r1

lw r1, 0(r2)

bne r2, r4 LOOP

addi r2, r2, 4
```

☐ Complete diagram above for enough iterations to determine CPI.

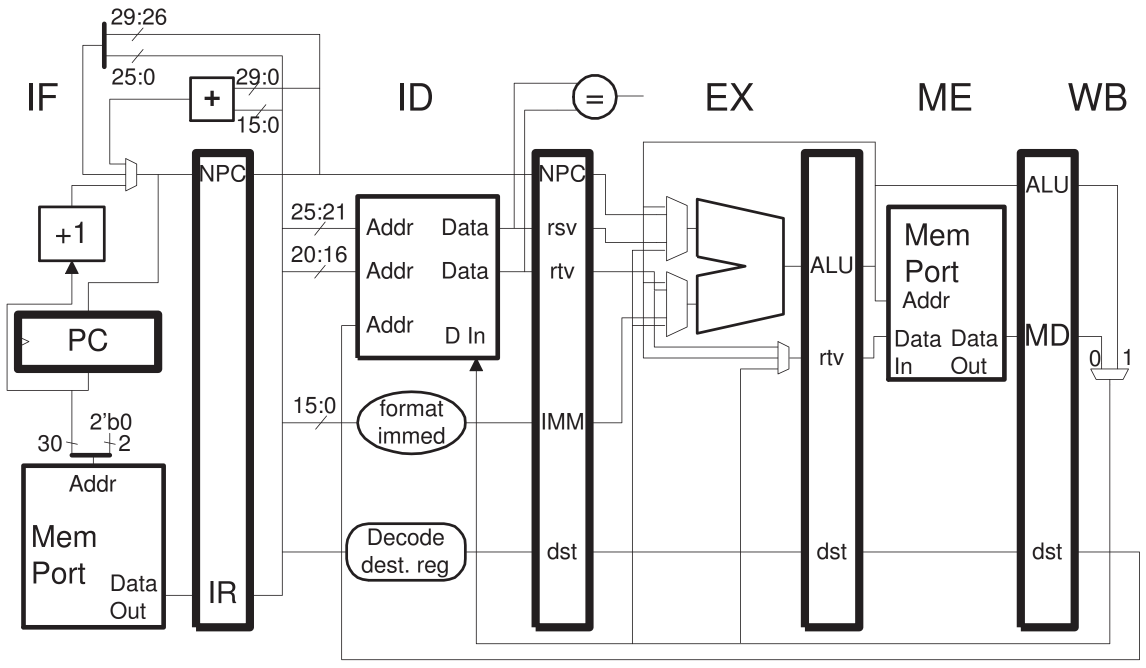
☐ Find the CPI for a large number of iterations.

(*c*) Complete the execution diagram for the MIPS code below on a two-way superscalar dynamically scheduled implementation of the type described in class. The execution of the first instruction is shown. The `lw` instruction uses stages `EA` and `ME` in place of `EX`.

```
add r1, r2, r3          IF   ID   Q    RR   EX   WB   C

add r4, r5, r6

add r7, r4, r8

lw  r9, 0(r7)

addi r1, r9, 3

xor r10, r11, r12
```

☐ Complete diagram above.

Problem 3: (14 pts) A deeply pipelined MIPS implementation can be constructed by dividing some stages of our familiar 5-stage statically scheduled scalar implementation (shown below) into two or more parts. In this problem the technique is applied to construct several 8-stage implementations. All have just one `ID` and `WB` stage, and in all implementations **it takes** 4.4 ns **for an instruction to pass through all 8 stages**, from the beginning of `IF1` to end of `WB1`. The stages are divided without changing what they do. For example, if an "original" MIPS stage, say `EX`, is divided into multiple stages, say `EX1 EX2 ... EXn`, then all values from `ID` and bypass paths are needed when `EX1` starts, and values reach `ME` in the cycle after `EXn`. Our familiar 5-stage implementation is shown below for reference:



(*a*) Consider the 8-stage *baseline* implementation below (indicated by the stage labels). What is the execution rate of a friendly program (one written to maximize performance) on the implementation, in units of instructions per second? The answer can be given as a formula of constants (as opposed to putting down just x as an answer).

`Baseline Implementation: IF1 IF2 ID1 EX1 EX2 ME1 ME2 WB1`

☐ Execution rate in instructions per **second**.

(*b*) Call a program *favorable* for an implementation if it runs faster on the implementation than on the baseline (repeated below). If it runs slower, call it *unfavorable*. For each implementation below write a two- or three-instruction favorable program and a two- or three-instruction unfavorable program. Also provide a concise but clear explanation of what it is about the program and implementation that makes it favorable or unfavorable. If a program is favorable on one implementation and unfavorable on another write it once, but provide an explanation for each. *Hint: The three implementations differ in how they are affected by certain dependencies.*

```
Baseline Impl.:    IF1 IF2 ID1 EX1 EX2 ME1 ME2 WB1
Implementation 1:  IF1 IF2 IF3 ID1 EX1 ME1 ME2 WB1
```

☐ Favorable program. ☐ Explanation.

☐ Unfavorable program. ☐ Explanation.

```
Baseline Impl.:    IF1 IF2 ID1 EX1 EX2 ME1 ME2 WB1
Implementation 2:  IF1 ID1 EX1 EX2 EX3 ME1 ME2 WB1
```

☐ Favorable program. ☐ Explanation.

☐ Unfavorable program. ☐ Explanation.

```
Baseline Impl.:    IF1 IF2 ID1 EX1 EX2 ME1 ME2 WB1
Implementation 3:  IF1 ID1 EX1 EX2 ME1 ME2 ME3 WB1
```

☐ Favorable program. ☐ Explanation.

☐ Unfavorable program. ☐ Explanation.

Problem 4: (14 pts) Answer the following branch predictor questions.

(*a*) Code producing the branch patterns shown below is to run on three systems, each with a different branch predictor. All systems use a 2^{14}-entry BHT. One system uses a bimodal predictor, one system uses a local predictor with a 16-outcome local history, and one system uses a global predictor with a 16-outcome global history.

```
0x1000: B1:  T N  T T N  T T T N   T N  T T N  T T T N     ...
0x1010: B2:     T      T        N     T      T        N    ...
0x1020: B3:                      T                     T   ...
```

For the questions below accuracy is after warmup.

☐ What is the accuracy of the bimodal predictor on `B1`?

☐ What is the accuracy of the local predictor on branch `B1`?

☐ What is the warmup time of the local predictor on branch `B1`?

☐ What is the minimum local history size needed for a local predictor to predict `B1` with 100% accuracy?

☐ What is the accuracy of a global predictor with a three-outcome global history on branch `B2` (not `B1`)?

☐ What is the minimum global history size needed for a global predictor to predict `B2` (not `B1`) with 100% accuracy?

Problem 4, continued:

(*b*) Consider code producing the patterns below running on two systems, one using a global predictor and the other using a gshare predictor. Both systems use a 2^{16}-entry BHT and a 12-outcome global history. The hexadecimal numbers indicate the address of the branch instruction. For example, `B5: 0x1100` indicates that the instruction we call `B5` is at address `0x1100`.

```
Pattern L:  T T T ... N    (A hundred iteration loop.)

B5: 0x1100:  L       L       L       L
B6: 0x1110:    N       N       N       N
B7: 0x1200:      L       L       L       L
B8: 0x1210:        T       T       T       T
```

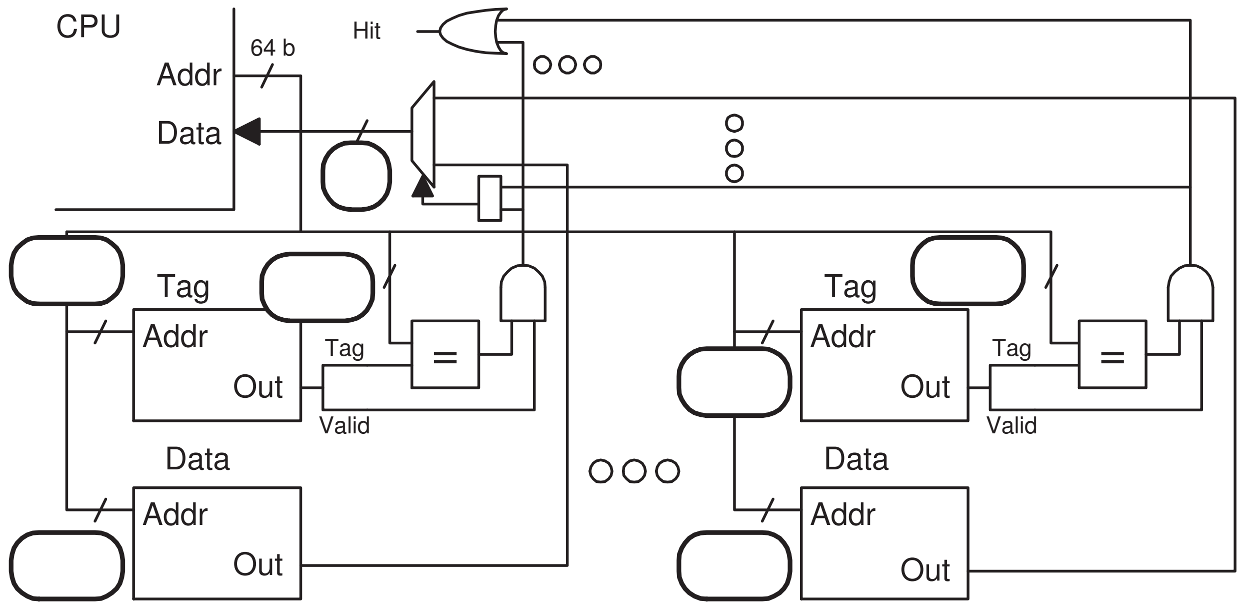
☐ Why is the accuracy of the gshare predictor so much better than the global predictor on this example?

☐ What is the minimum number of BHT entries for which the gshare predictor will outperform global on this code? *Hint: Look at the branch addresses.*

Problem 5: (15 pts) The diagram below is for a **4-way** set-associative cache with a capacity of 8 MiB (2^{23} bytes). The system has the usual 8-bit characters.

(*a*) Answer the following, formulæ are fine as long as they consist of grade-time constants.

☐ Fill in the blanks in the diagram.



☐ Show the address bit categorization. Label the sections appropriately. (Alignment, Index, Offset, Tag.)

Address:

7 **4**

☐ Memory Needed to Implement (Indicate Unit!!):

☐ Line Size (Indicate Unit!!):

☐ Show the bit categorization for a direct mapped cache with the same capacity and line size.

Address:

Problem 5, continued: The problems on this page are **not** based on the cache from the previous page. The code fragments below run on a 32 MiB (2^{25} byte) direct-mapped cache with a 64-byte line size. Each code fragment starts with the cache empty; consider only accesses to the array, `a`.

(*b*) Find the hit ratio executing the code below.

☐ What is the hit ratio running the code below? Explain

```
float sum = 0.0;
float *a = 0x2000000; // sizeof(float) == 4
int i;
int ILIMIT = 1 << 11;    // = 2^11

for (i=0; i<ILIMIT; i++) sum += a[ i ];
```

(*c*) Find the smallest value of `STRIDE` for which the cache hit ratio in the program below will be zero.

☐ Fill in smallest value for `STRIDE` for which hit ratio 0.

☐ Briefly explain.

```
float sum = 0.0;
float *a = 0x2000000; // sizeof(float) == 4
int i;
int ILIMIT = 1 << 11;    // = 2^11

int STRIDE =                                          ;    //   <---- FILL IN

for (i=0; i<ILIMIT; i++) sum += a[ i ] + a[ i + STRIDE ];
```

Problem 6: (15 pts) Answer each question below.

(*a*) What is the difference between instruction-level parallelism (ILP) and explicit parallelism?

☐ ILP and explicit parallelism difference.

(*b*) A company has a large customer base for its ISA Y products, the most advanced of which is a 4-way superscalar dynamically scheduled implementation. The company is considering three possible next-generation systems: Develop an 8-way superscalar dynamically implementation of ISA Y, develop a chip with 16 scalar implementations of ISA Y, or develop a VLIW ISA and implementation. For each strategy indicate how much effort it will take for customers to use the new system.

☐ Customer effort to use 8-way superscalar dynamically scheduled system. Explain.

☐ Customer effort to use chip with 16 scalar implementations. Explain.

☐ Customer effort to use VLIW implementation.

(*c*) Consider the three systems from the previous part. For each system indicate an advantage over the others. The advantage should specify an assumed workload, an answer might start "The advantage, those customers that run programs that are __________, is . . ..

☐ Advantage of 8-way superscalar dynamically scheduled system.

☐ Advantage of 16 scalar system chip.

☐ Advantage of VLIW implementation.

Problem 7: (18 pts) Answer each question below.

(*a*) Why might a 1% change in branch prediction accuracy have a larger impact on the performance of a 4-way superscalar processor than on a 2-way superscalar system?

☐ Larger impact on 4-way over 2-way because...

(*b*) Dynamically scheduled systems use a technique called register renaming in which an instruction's architected registers are renamed into physical ones. Provide a brief example illustrating why register renaming is necessary.

☐ Example illustrating need for register renaming and explanation.

(*c*) The SPECcpu rules require that the compilers used to prepare a run of the suite be real products that the company (or some other company) makes an honest effort to sell (or give away). Explain how the SPECcpu scores might be inflated if the compilers could be products in a technical sense, but not something the company is trying to put in customer hands.

☐ Scores inflated by unmarketed compilers because...

☐ If the compilers produce faster programs, why not promote them?

(*d*) What is the difference between a trap and an exception?

☐ Difference between trap and exception.