

Name \_\_\_\_\_

Computer Architecture  
EE 4720  
Practice Midterm Examination  
Before 14 March 1997, 12:40 CST

*Modified 11 March 1998*

Problem 1 \_\_\_\_\_ (25 pts)

Problem 2 \_\_\_\_\_ (25 pts)

Problem 3 \_\_\_\_\_ (25 pts)

Problem 4 \_\_\_\_\_ (25 pts)

Alias \_\_\_\_\_

Exam Total \_\_\_\_\_ (100 pts)

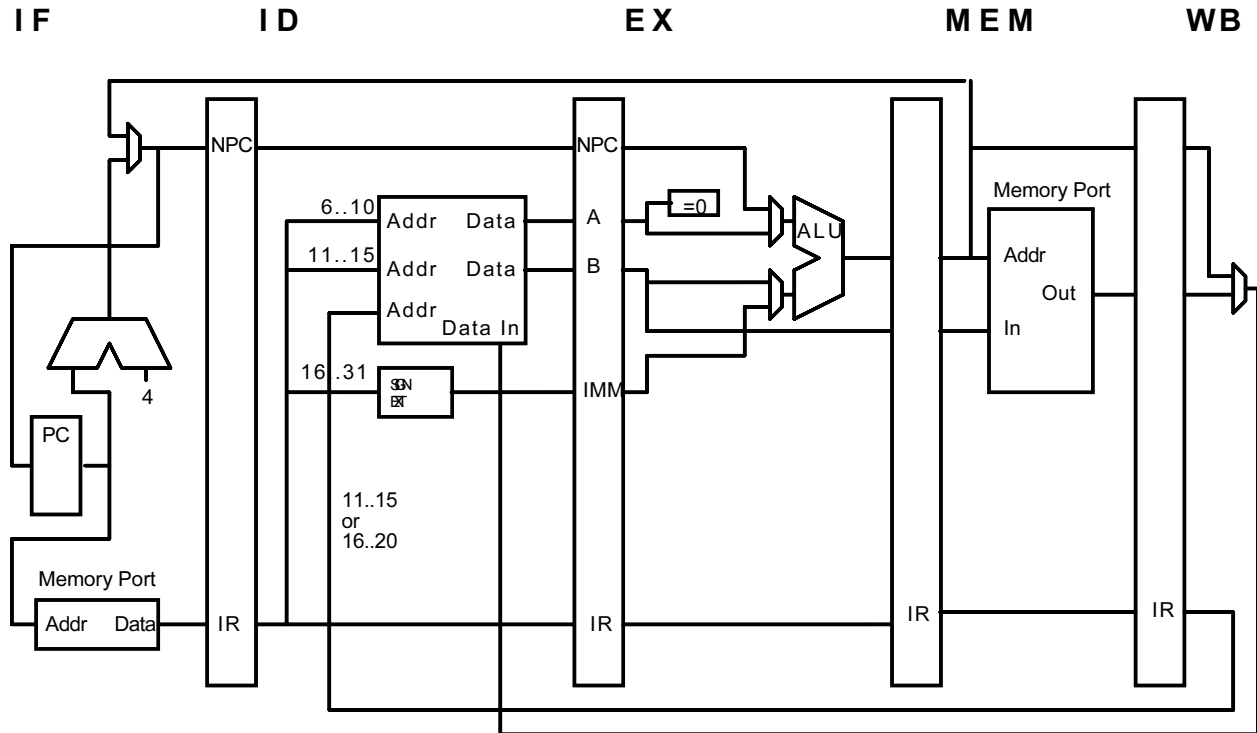
*Good Luck!*

Problem 1: Consider a new instruction, `addbi` (add big immediate). This instruction adds the contents of two registers plus a 32-bit immediate. For example,

```
addbi r1, r2, r3, #123 ! r1 = r2 + r3 + 123 (a 32-bit constant).
```

The instruction is coded using two words, the first word is coded the same way as an `add` instruction, except the opcode is different. The second word contains the 32-bit immediate.

(a) Show how the DLX pipeline could be modified to implement `addbi`.



(b) Show the pipeline execution diagram for the following fragment (which includes `addbi`), running on a pipeline with register bypassing.

```
lw r2, 10(r8)
addbi r1, r2, r3, #123456 ! r1 = r2 + r3 + 123456 (a 32-bit constant).
sub r1, r5, #456
```

Problem 2: Branches in a processor implementation generate stalls as indicated below:

Predicted taken, not taken, stall 1 cycle.

Predicted taken, taken, stall 2 cycles.

Predicted not taken, not taken, stall 0 cycles.

Predicted not taken, taken, stall 4 cycles.

Other control-transfer instructions (*e.g.*, jumps) stall 2 cycles. No other instructions generate stall cycles. The system has a pipeline depth a five.

Consider a benchmark consisting of 1,000,000 instructions, in which 10% are branches and 5% are other control-transfer instructions. Seventy percent of the branches are taken.

(a) Suppose the hardware is designed based on the predicted outcome of all branches being the same. (That is, either all branches are predicted taken, or all branches are predicted not taken.) Compute the CPI and execution time (in cycles) for each choice, and indicate which prediction choice is better.

(b) Compute the CPI and execution time for the system assuming perfect branch prediction.

(c) Compute the CPI and execution time for a system in which 80% of branches are correctly predicted. The prediction accuracy is the same for taken and non-taken branches.

**Problem 3:** Consider an ISA which features variable-size integers in which integer operands can vary from 1 bit to 256 bits. Thus an add instruction might specify that a 30-bit quantity is to be added to a two-bit quantity and stored as a 32-bit result. The system has 32 registers.

(a) Assuming all registers can hold the largest size integer, devise an instruction format for three-register arithmetic instructions using such variable size integers.

(b) Instead of assuming large registers, suppose that there is a fixed amount of register storage which can be divided into a small number of large registers, large number of small registers, or any mixture of sizes no larger than 256 bits. For example, 2048 bits could be addressed as 32 64-bit registers, 2048 1-bit registers, 8 256-bit registers, 4 256-bit registers plus 4 128-bit registers plus 512 1-bit registers, etc. For such a register configuration, devise an instruction format for three-register arithmetic instructions using such variable-size integers.

Problem 4: Answer each question below.

(a) Most ISAs specify 32 or 64 registers. What would be the drawback of an ISA which includes 1,048,576 registers, assuming that the cost of the register storage itself is small?

(b) The `lw` instruction below increments the address in addition to performing a load. Explain how it might generate a WAW hazard if `r2` is written in the EX stage.

```
lw r1,(r2)+ ! r2 incremented after load.
```