

ECE/CS 552: Introduction to Computer Architecture
ASSIGNMENT #5

Due Date: In class December 14th, 2005

This homework is to be done individually.

Total 4 Questions, 80 points

1. (10 points) Virtual Memory

Consider a virtual memory system with the following properties:

- 40-bit virtual byte address
- 16-KB pages
- 36-bit physical byte address

What is the total size of the page table for each process on this machine, assuming that the valid, protection, dirty and use bits take a total of 4 bits and that all the virtual pages are in use? (Assume that disk addresses are not stored in the page table.)

2. (20 points) Branch Prediction

Executing a piece of code leads to the following sequence of branch outcomes (a **repeated** pattern of 4 taken branches followed by a not taken branch):

T T T T N T T T T N T T T T N T T T T N T T T T N...

This sequence of branches is to be predicted with a predictor that uses only two bits of global history to access a table of two bit counters.

- In the steady state**, what is the branch misprediction rate? (The initial counter state is unknown.)
- What is the minimum length global history register that will result in 100% prediction accuracy in the steady state? (Regardless of the initial state of the predictor.)

3. (20 points) Register Renaming

The following set of instructions is to be renamed onto a set of physical registers. The initial register map and free list are given. Show the final renamed instruction sequence.

R2 <- mem (R3 + R1)
R1 <- R1 + 4
R7 <- R1 + R2
R4 <- R4 - 1

Logical	Physical
R1	P9
R2	P3
R3	P2
R4	P7
R5	P1
R6	P10
R7	P12

Free: P4, P5, P6, P8, P11

4. **(30 points)** *SimpleScalar Simulation*

(For this problem, use the SimpleScalar simulator, which you created from the previous)

"sim-bpred" is a branch predictor simulator in SimpleScalar. Report the branch condition prediction rate, branch target prediction rate, and return address stack prediction rate for the following simulation with the default parameters. Also describe the configurations being simulated by examining the simulator output:

```
./RUNgo ~ece552/simplesim-3.0/sim-bpred ./go.ss >& outfile  
(Note: Your command line may look slightly different)
```

This will run the go benchmark and simulate a branch predictor. The output of the simulation will be placed in "outfile".