

CSSE7013 - Advanced Computer Architecture Tutorial 3

Learning Objectives

This tutorial aims to help you to understand approaches to pipeline design.

QUESTION 1—Pipeline Scheduling

For the following MIPS-like code fragment, under the assumptions

- a 5-stage pipeline:
 - instruction fetch (IF): next instruction fetched from [PC]
 - instruction decode (ID): instruction type determined, register input operands fetched or forwarding logic invoked, branch outcome and target computed
 - execute (EX): ALU operations completed, memory read/write address calculation, memory write operand from register; successor instruction may use result next cycle through forwarding
 - memory (MEM): ALU operations to registers, memory read complete, memory write start
 - writeback (WB): memory read into register
- data reads and writes do not interfere with IF
- forwarding logic allows a result to be used in the CPU even if it has not yet reached its destination register in the cycle after it is computed
- R3 is initially R2 + 32

```

loop:  LW    R1, 0(R2)      ; fetch and increment A[i]
      ADDI  R1, R1, #1
      SW   0(R2), R1
      LW   R1, 0(R5)      ; fetch and double B[i]
      ADD  R1, R1, R1
      SW   0(R5), R1
      ADDI R5, R5, #4     ; increment indexes by 4
      ADDI R2, R2, #4     ; (since word size is 4 bytes)
      SUB  R4, R3, R2
      BNZ  R4, loop
    
```

- a. Mark all dependences and anti-dependences, using arrows with solid lines for dependences, and dashed lines for anti-dependences.
- b. Use a timing diagram in the following format (the example is different) to determine timing to the point where the next iteration starts:

LW R1, 0(R2)	IF	ID	EX	MEM	WB			
ADDI R2, R2, #1		IF	ID	stall	EX	MEM	WB	
ADDI R4, R4, #1			IF	stall	EX	MEM	WB	
SW 0(R2), R1					IF	ID	MEM	WB

- c. Now consider how dependences wrap around to another iteration of the loop.
- d. If we now introduce a superscalar pipeline in which up to 2 instructions can be issued simultaneously without out of order completion, what fraction of the available instruction bandwidth can be used (assuming no stalls caused by the memory hierarchy)?
- e. Reschedule to code to separate dependences as far as possible. How much would your previous answers change?
- f. Would branch prediction or register renaming help in any of the examples?

QUESTION 2—Branch Prediction

Taking the code sequence of Question 1, work out the values in the branch prediction tables for the following cases, after the first, 4th and last iteration, assuming all predictors are initialised to NT:

- a. a 1-bit 1-level branch predictor.
- b. 2-bit 1-level branch predictor.
- c. What difference would a 2-level predictor make in this case? When would it make a bigger difference?

QUESTION 3—More Aggressive Parallelism

- a. Consider the effects of a 4-way superscalar pipeline.
 - i. Redo the timing chart without any code changes, and no branch predictor.
 - ii. Work out the effects of adding in branch prediction, using whichever strategy looked best in Question 2.
 - iii. Now imagine there is another thread which happens to contain the same sequence of code which may be interleaved with the original, filling any stalls. Assume the original runs whenever it can, and the second sequence has its own registers (i.e., similar names do not refer to the same registers as the original):
 - a. How many stalls can you fill now?
 - b. What does this example tell you about SMT?
 - iv. If instead of SMT, we had a chip multiprocessor where each CPU could only schedule two instructions per clock, but with the same clock speed as the 4-way superscalar design, which would be faster, only taking into account factors in this tutorial?
- b. If you had to design an architecture for low power but capable of scaling up to high-speed designs with minimal change, which would you choose (explain trade-offs in each case)?
 - i. as aggressive a superscalar design as would fit the power requirements?
 - ii. as aggressive an SMT design as would fit the power requirements?
 - iii. as aggressive a chip multiprocessor (CMP) design as would fit the power requirements?

QUESTION 4—More Exercise

- a. Work through the examples in Chapter 3.
- b. Work through questions at the end of Chapter 3. Warning: the exercises in the book can sometimes be hard e.g. because details are left out, requiring you to make assumptions.

Last update: Monday, 29 August 2005 at 5:02 PM