Department of Electrical Engineering, IIT Bombay

EE309 Computer Organization, Architecture and Microprocessors: Tutorial Sheet VIII

Pipelining, RISC and Parallel Computations

1. **Tables reserved for A function**
Consider function A, with the following Reservation Table:

	0	1	2	3	4	5	6
Stage 1			×		×		
Stage 2		×		×		×	
Stage 3	×		×				×

   (a) What is the MAL bound, as given by the MAL Lemma (Lemma 3-1) ?

   (b) Compute the initial Collision Vector, $CV_{initial}$

   (c) Now, construct the State Diagram, considering collision vectors. How many states does this have ?

   (d) Construct the Modified State Diagram. How many states, now ?

   (e) Enumerate all simple cycles

   (f) What is the upper bound on the average latency of any greedy simple cycle for Reservation Table A, as given by Lemma 3-3 ?

2. **Crass Pro-Procrastination**
Consider the following Reservation Table:

	0	1	2	3	4	5
Stage 1	×		×			×
Stage 2		×	×		×	
Stage 3			×	×		

Find out the Compute time of the above Reservation Table, the MAL bound (from the MAL Lemma (Lemma 3-1)), the actual MAL (from the Modified State Diagram), and the/an optimal cycle. Now, consider the following situation. Suppose it is possible to change the reservation table through the insertion of delays, in such a manner that the overall functionality of the Reservation Table is not affected (*i. e.*, it still computes the same function):

	00	01	02	03	04	05	06	07	08	09	10
Stage 1	×		×								×
Stage 2		×		×		×					
Stage 3			×		×						

Find out the Compute time of the above Reservation Table, the MAL bound (from the MAL Lemma (Lemma 3-1)), the actual MAL (from the Modified State Diagram), and the/an optimal cycle. Do you find anything surprising ?

3. **Chinese Whisper ?**
Consider the problem of one-to-all communication between p processors, arranged in a certain topology. The parameters determining the communication time for a message are:

   - Startup time t_s: time to handle a message at the sending processor - to prepare the data, encode it, packetize it, etc.
   - Per-word transfer time t_w: this is the inverse of the channel bandwidth - the time taken to traverse a link between two processors linked directly, for a word.

   Assume m words to be transmitted, and Store-and-Forward routing. Compute the time for one-to-all communication for p processors organized in a ring, a wraparound mesh, and a hypercube. Illustrate the methodology using $p = 8$ for a ring, $p = 16$ for a wraparound mesh, and $p = 8$ for a hypercube.

4. **All for one, and one for all**
Consider the problem of adding p numbers (each number is m words long), where p is a power of 2. Initially, each of the p numbers is stored in a separate processor, and the routing mechanism is Store-and-Forward routing. Assume the startup time to be t_s, and per-word transfer time to be t_w. At the end of the computations, processor 0 stores the sum. Further, assume that the time taken by a processor to add two numbers is a constant t_a, independent of the word length m. Also, the result of addition(s) can be assumed not to exceed the word length m in any case. Note that the algorithms used have to be *efficient and optimal* - memory and time.

   (a) Calculate the total time required, if the processors are arranged in a (i) linear ring, (ii) a wraparound mesh (a torus), and (iii) a hypercube. Briefly explain the logic used to arrive at the expressions.

   (b) Illustrate the steps in the computations, for p = 16.

   (c) Write the pseudo-code of a program, that runs on processor i. for each of the above three connection architectures.

5. **Taking a RISC**
Consider a RISC pipeline, where register-to-register instructions have two phases:

   - **(I)** Instruction Fetch
   - **(E)** Execute

   For load and store operations, three phases are needed:

   - **(I)** Instruction Fetch
   - **(E)** Execute. Calculates Memory address
   - **(D)** Memory. Register-to-memory or memory-to-register operation

   Suppose all branches (jumps) are unconditional ones *only*. Explain the purpose of the delayed branch (using the `NOP` - 'Perform no operation' statement), and the significance of the optimized delayed branch below:

Address	*Normal Branch*	*Delayed Branch*	*Optimized Delayed Branch*
100	`LOAD X, A`	`LOAD X, A`	`LOAD X, A`
101	`ADD 1, A`	`ADD 1, A`	`JUMP 105`
102	`JUMP 105`	`JUMP 106`	`ADD 1, A`
103	`ADD A, B`	`NOP`	`ADD A, B`
104	`SUB C, B`	`ADD A, B`	`SUB C, B`
105	`STORE A, Z`	`SUB C, B`	`STORE A, Z`
106		`STORE A, Z`	