CS 3853: Computer Architecture
Final
Spring 2014
May 7, 2014
Total Marks: 100
Time: 2 hour 30 min

Name:
Banner Id:
Alias:

1. [15 pt] Indicate **True** or **False** for the following statements:

   - Throughput can be a measure of performance.
   - To summarize ratios, weighted average is better that geometric mean.
   - If system A is n% faster than system B, $ET_A/ET_B = 1 + n/100$.
   - All instructions take equal number of cycles for a MIPS 5 stage pipeline.
   - There are 3 types of hazards in a pipeline.
   - Delayed branch is not a static branch handling technique.
   - If branch target calculation and condition evaluation are done in the same stage, then *pipeline freeze* and *predict taken* strategies cause the same number of stalls.
   - Output dependence is a type of name dependence.
   - A dynamically scheduled processor can execute instructions out of order.
   - Tomasulo's algorithm eliminates all 3 dependences (WAW, WAR, and RAW).
   - The key idea behind speculation is to allow instructions to execute out of order but to force them to commit in order.
   - A fully associative cache does not index calculation.
   - Bigger cache blocks increase compulsory misses.
   - Cache miss is handled by hardware but page fault is handled by OS.
   - Locality that occurs in time is called temporal locality.

2. [15 pt] CPU Performance and Amdahl's Law:

   (a) [2+3 pt] Suppose you can apply two enhancements - enh_1 and enh_2. enh_1 can be applied to 30% of the program whereas enh_2 can be applied to 20% of the first enhancement. Suppose the speed ups of the enhancements are 3.0 and 4.0 respectively.

      i. How much will be the overall speed up when only enh_1 is applied?

      ii. How much will be the overall speed up if you apply both enhancements together?

(b) [2+3+3+2 pt] Suppose we have made the following measurements:

- Frequency of FP operations = 25%
- Average CPI of FP operations = 4.0
- Average CPI of other instructions = 1.33
- Frequency of FPSQR= 2%
- CPI of FPSQR = 20

Assume that the two design alternatives are (a) to decrease the CPI of FPSQR to 2, or (b) to decrease the average CPI of all FP operations to 2.5.

i. What is the CPI of the original design?

ii. What is the CPI of design alternative (a)?

iii. What is the CPI of design alternative (b)?

iv. Which design alternative is better? What is its speed up?

3. [10 pt] Pipeline:
   Let us consider a multicycle MIPS pipeline shown in Figure 1.

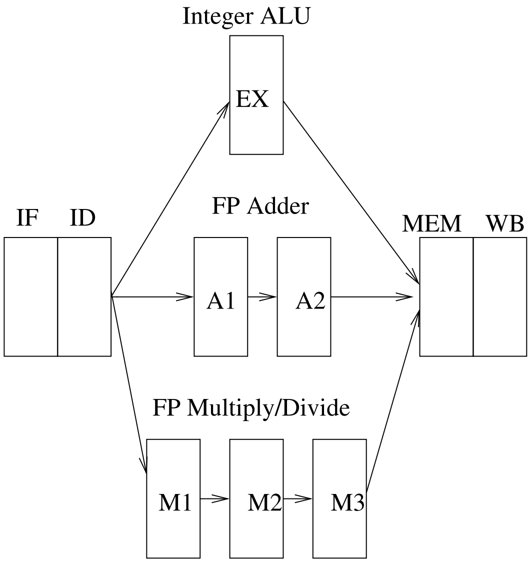

Figure 1: Pipeline stages.

- A branch target address is calculated in ID stage and condition is evaluated in EX stage. Branches are handled by using static predict taken policy.
- The pipeline forwards data from MEM or EX or M3 or A2 stage.
- Registers are written in the first half of a clock cycle and read in the second half.
- We have separate instruction and data memory.
- Branches are handled using **predict-taken** strategy.

(a) [6 pt] Fill up the following pipeline timing diagram.

Ins	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
LD R1, 0(R4)																		
BEQZ R1, L1																		
L1: ADD.D F3, F1, F2																		
MUL.D F2, F3, F4																		
SD F3, 8(R2)																		
ADD.D F1, F3, F4																		

(b) [4 pt] Is it possible to have WAW and WAR hazards in this pipeline? If so, give an example of the corresponding hazards?

4. [20 pt] Instruction Level Parallelism:

(a) [12 pt] Tomasulo's algorithm:
We have the following assumptions.

- Assume that you have 2 Integer, 1 FP Add/Sub, and 1 FP Divider/Multiplier Functional Unit. Integer FU takes 1 cycle, FP Add/Sub takes 2 cycles and FP Divider/Multiplier takes 10 cycles.
- Assume **no hardware speculation** and **single issue**.
- There is just **one CDB**.
- Whenever there is a conflict for a functional unit or CDB, assume that the oldest (by program order) of the conflicting instruction gets access, while others are stalled.
- Branches, loads, and stores use integer FU.
- A branch instruction does not use CDB or memory.
- Assume that the branch is not taken.
- Loads and stores use the integer functional unit to perform effective address calculation during the EX stage. Memory access takes **1 cycle**. Data memory can serve only one access at a time.
- If an instruction moves to CDB write stage in cycle **x**, then an instruction that is waiting on the same functional unit (due to a structural hazard) can start executing in cycle **x** unless the waiting instruction is dependent on the first one. In that case, the waiting instruction will start at cycle **x+1**.
- Assume that the number of reservation stations is 2 for Integer, 1 for FP Divider/Multiplier and 1 for FP Add/Sub. Load, store and branch instructions use integer reservation stations.

Complete Table 1 using Tomasulo's algorithm with the above specifications.

Instruction	Res. Station	IS	EX	Mem. Access	CDB Write
LD F2, 0(R1)					
MUL.D F4, F2, F0					
SD F4, 8(R1)					
MUL.D F4, F0, F1					
DSUBUI R1, R1, #32					
BEQZ R1, **L1**					
LD F1, 0(R1)					
ADD.D F3, F4, F2					

Table 1: Instruction sequence

(b) [3 pt] How does a brach target buffer work?

(c) [5 pt] 90% of all branch instructions are found in a branch target buffer. Among the branches that are found there, 90% are correctly predicted. For the branches that are not found in the target buffer or found in the target buffer with incorrect prediction, the pipeline incurs a 2 cycle stall. What is the branch penalty (i.e. average number of stalls) for the branch target buffer?

5. [40 pt] Memory hierarchy:

   (a) [32 pt] Let us assume a 2-way set associative 128 KB L1 cache with LRU replacement policy. The cache implements write back and no write allocate policy. Cache block size is 16 Byte. Page size is 64 KB. The system has a direct mapped TLB with16 entries. TLB also implements LRU policy. Virtual address is 24 bit and physical address is 20 bit long. The cache is virtually indexed and physically tagged.

      i. [2 pt] How many entries are there in the page table?

      ii. [2 pt] How many physical pages are there?

      iii. [2 pt] How many bits are used for TLB index and tag?

      iv. [2 pt] How many bits are used for cache index and tag?

v. [24 pt] Please fill up the following the table. All numbers are in hexadecimal.

Addr R/W	Binary Addr	TLB Tag	TLB Index	TLB Hit	Physical Addr	Cache Index	Cache Tag	Found Way
16103A W	000101100001000000111010							
17113C R	000101110001000100111100							
26112B R	001001100001000100101011							
16103B R	000101100001000000111011							
16103C W	000101100001000000111100							
23113D R	001000110001000100111101							
26103A R	001001100001000000111011							
17113D R	000101110001000100111101							
18113A R	000101110001000100111101							
23113C W	001000110001000100111100							
18113E R	000101110001000100111110							
23103A R	001000110001000000111010							

Table 2: Memory access sequence

Virtual Page	Physical Page
16	4
17	8
18	9
23	6
26	7

Table 3: Partial Page Table

TLB Index	TLB Tag	Physical Page	V
3			
6			
7			
8			

Table 4: TLB

(b) [3 pt] What is the effect of increasing block size on three types of misses?

(c) [2 pt] Give an example of loop interchange?

(d) [3 pt] What is early restart policy?