CHALMERS

2004-12-03

Exam in Computer Architecture (EDA111)

Time: December 13, 2004, XXXX in the M-Building

Person in charge of the exam: Charlotta Bååth, phone: 772 1711

Supporting material/tools: None

Exam Review: On appointment with Per Stenstrom (pers@ce.chalmers.se) before XXX

Grading intervals:

- **Fail**: Result < 20
- **Grade 3**: 20 <= Result < 29
- **Grade 4:** 30 <= Result < 39
- **Grade 5:** 40 <= Result

Important Note: Answers can be given in Swedish. The reason this exam is presented in English is that the course is also offered to the International Master's Program on Dependable Computer Systems**.**

GOOD LUCK!
Per Stenström

CHALMERS UNIVERSITY OF TECHNOLOGY
DEPARTMENT OF COMPUTER ENGINEERING
412 96 Göteborg
Visiting address: Hörsalsvägen 11
Phone: 031-772 1761 Fax: 031-772 3663
Org. Nr: 556479-5598
E-mail: pers@ce.chalmers.se

ASSIGNMENT 1

A) Assume that every fourth instruction is a branch and that CPI=1 for all instructions except for branches that execute in 4 cycles if the branch prediction is wrong and in two cycles if branch prediction is correct. **What branch prediction accuracy would make the average CPI=1.5? (*3 points*)**

B) Consider the code below:

```
A = B;
C = A+D;
B = E
   If (Cond==true) then F=G
```

Mark all data, name, and control dependencies in the code (*3 points*)

C) Assume that we access 32-bit words in a computer. **What word addresses are possible if alignment is required? (*2 points)***

D) What is a victim cache? (*2 points)*

ASSIGNMENT 2

A) Consider the following code

A = B + C

where A, B, and C are memory locations. Show the code for a stack architecture, an accumulator architecture and a load/store architecture by inventing assembly mnemonics for the three architectures. ***(3 points)***

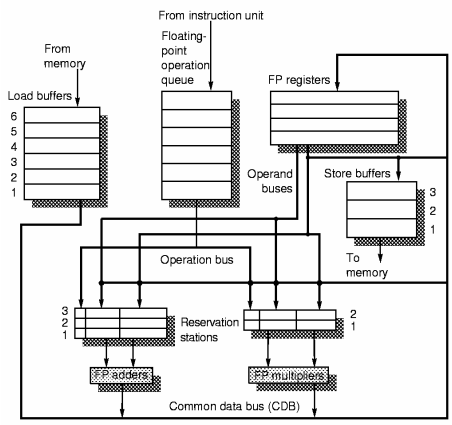
B) As an architect you need to make a decision whether to choose a memory/memory architecture or a load/store instruction. The load/store architecture implements delayed load, but all other instructions execute in one cycle. You are interested in comparing the execution time for the code in (A) for the two architectures.

(i) **Write the code in (A) for the memory/memory architecture**. ***(2 points)***

(ii) Assuming that the clock cycle time is the same in both architectures, **how many cycles must the add instruction in the memory/memory instruction take to not exceed the execution time for the load/store architecture?** ***(2 points)***

C) There are essentially two reasons why PC-relative addressing is used for branches. **Which two?** Moreover, **what is the intuitive reason why the branch displacement is often quite short?** ***(3 points)***

ASSIGNMENT 3



(A) Explain what happens when an instruction issue and when it writes back the result. You should especially explain what the purpose of the *reservation stations* and the common data bus is in these processes. ***(2 points)***

(B) You should find the organization with a pipeline implementing Tomasulo's algorithm above familiar. Consider the following code:

LD F6, A (6 cycles)
LD F2, B (6 cycles)
MUL F0, F2, F4 (10 cycles)
DIV F10, F0, F6 (40 cycles)
ADD F6, F8, F2 (1 cycle)

(i) **When can each instruction start execution given the instruction latencies? (*4 points)***

(ii) **How is the name dependency between the DIV and the ADD instruction resolved? (*2 points)***

(C) A reorder buffer is used in a dynamic speculative processor. **What is the purpose of the reorder buffer and what actions are taken when an instruction commits**? ***(2 points)***

ASSIGNMENT 4

A) Consider the following code:

```
LOOP:  LD F0, 0(R1)
       DADDUI R1,R1,#-8
       ADD F4,F2,F0
       stall
       SD  F4, 8(R1)
       BNE  R1,R2, LOOP
```

(i) **Do *loop unrolling* twice** and show which instructions you get rid of and how register renaming is needed to get rid of the stall cycles. ***(3 points)***

(ii) Do *software pipelining* of the same loop and show what code is needed in front of the loop and after the loop to make sure that the pipeline is filled and drained correctly. ***(3 points)***

(B) Consider the following loop:

```
for (i=1; i<=100; i=i+1)
    X[4*i+4]=X[2*i+1]*6.0;
```

Does this loop contain any loop-carried dependences? ***(2 points)***
Hint: Use the GCD test.

(C) Use the unrolled loop in (A) to schedule instructions on a VLIW architecture that contains two floating point units and two memory ports. **How many cycles does the code take?** ***(2 points)***

ASSIGNMENT 5

(A) Suppose we have a write-through cache with a write buffer. **What is the danger of letting a subsequent read miss bypass the writes that are posted in the write buffer?** ***(2 points)***

(B) How does the block size impact on capacity, cold, and conflict misses? ***(2 points)***

(C) It is possible to index the cache in parallel with the address translation if the first-level cache is sufficiently small. Suppose that the page size is 8 Kbytes and the first level cache is 2-way associative. **How big can it be to allow cache indexing to perform in parallel?** ***(2 points)***

(D) The existence of multiple processors with their own caches in a multiprocessor introduces the cache coherence problem. **Explain what it is and give an example of a snooping protocol that solves this problem. (*2 points)***

(E) Explain why snooping protocols are not appropriate in distributed shared memory multiprocessors and why directory protocols are often used. ***(2 points)***

GOOD LUCK!