Exam in Computer Architecture

Justin Pearson

2002-03-07

Cover Sheet

Problem no.	Solution provided	Max	Your points
1		1	
2		6	
3		14	
4		9	
5		13	
6		11	
7		10	
	Total:	64	

Name : .....

Pers.no. : .....

Exam Rubric

All answers should be written in English or Swedish (English is preferred). A mark of 50% is required for a 3.

Obs!!

If you are not registered on the TF course please indicate which Computer Architecture Course you are registered on.

Hjälpmedel:

Pen,pencil, ruler,rubber, dictionary. You are not allowed a calculator.

1. Mark the following statements true or false.

   (a) The processor can decide if the value in a memory location is a character or part of a floating point number.

   (b) If a processor is asked to execute a datafile, the processor will refuse and signal an error.

   (c) It is possible to decide if a bit pattern represents a 2's complement number or a normal number.

   (**points 1**)

2. Computer Arithmetic

   (a) Convert (where possible) the following numbers into 5 bit 2's complement numbers.

   - 3,−3, −10,16. (**points 2**)

   (b) Explain with examples what happens when numbers overflow with binary two's complement. (**points 2**)

   (c) When an overflow occurs on the MIPS processor an Exception occurs. Explain what an Expception is. (**points 2**)

3. Question on Processor Implementation

   (a) What is latch, give one example of a latch and describe how it works. (**points 2**).

   (b) In a multicycle processor why is it necessary to record intermediate results in a latch? (**points 2**).

   (c) In a multicycle approach to processor implementation a finite machine is needed to control the processor. What is a finite state machine and what does it do in a multicycle implementation? (**points 4**)

   (d) What is microcode, you should explain in some detail. ((**4 points**)

   (e) Discuss the difference (if any) between RISC and CISC style processors. (**points ???**)

4. Pipelines

   (a) What is an instruction pipeline and how does it enhance the performance of processor. (**points 2**)

(b) What are pipeline stalls, what causes them on the MIPS processor. You should provide examples where possible (**points 4**).

(c) The MIPS processor has data-forwarding between arithmetic instructions. What is data-forwarding, how does it work on the MIPS and how is it possible. (**points 3**)

5. Memory and Caches

(a) What it a memory Cache? What principles does it rely on to increase the performance? performance. (**2 points**)

(b) Describe the memory access characteristics of programs that perform badly on cache based systems. (**2 points**).

(c) What are multi-level Caches? (**1 points**)

(d) Given a memory reference a Cache has to determine if the memory is in the Cache. There are a number of schemes for determining where a global memory reference should be placed in the cache memory. Describe the following Schemes: Direct mapped and n-way associative caches. (**4 points**)

(e) What is a TLB? What function does it provide in a virtual memory system? Why is it required? What principle does it rely on to provide good performance? (**points 4**)

6. Simple questions on MIPS instructions.

(a) With examples describe how the `lw` instruction works on the MIPS. (**points 2**)

(b) How can constants be loaded into registers instructions without using pseudo-instructions (`li` and `la` you should explain what really happens with `li` and `la`). (**points 3**)

(c) Suppose you have two values stored in memory at locations `0x40000010` and `0x40000014` and you want to add the two numbers and write the result in location `0x40000020` , write the shortest sequence of MIPS instructions to do this. You should not use any pseudo-instructions (**points 3**).

(d) Explain with examples how the `slt` instruction works on the MIPS and how it is used. (**points 3**)

7. More MIPS programming

(a) Assuming that the base address of the integer array `A` is stored in register `$s0`; the base address of the integer array `B` is stored in `$s1` and the base address of the integer array `C` is stored in register `$s2` translate the following fragment of C code into assembly:

```
int i;
for(i=0; i<100; i++) {
 A[i] = 2*B[i] - 2*C[i]
}
```

You should provide comments in the code and make the code as efficient as possible. (**points 4**)

(b) Write a function that given the address of a string in `$a0` decides if the string is a palindrome[1]. The function should return the result `0` in register `$v0` if the string is a palindrome and `1` if the string is not a palindrome. You should provide comments for the code.(**points 6**).

Good Luck!!

[1] A palindrome is a sequence of characters that is the same when read forward and backwards e.g. `aBBa`