Written Exam / Tentamen

Computer Organization and Components / Datorteknik och komponenter (IS1500), 9 hp
Computer Hardware Engineering / Datorteknik, grundkurs (IS1200), 7.5 hp

KTH Royal Institute of Technology

2020-01-13

14.00-19.00

Suggested Solutions

Part I: Fundamentals

In part I, on the real exam, only short answers are expected. The elaborated answers given here are just for your information, and are not needed on the real exam.

1. **Module 1: C and Assembly Programming**

   (a) Short answer:

   ```
   s += *(p1 + *p2);
   p2++;
   i++;
   ```

   Max 5 points. One point for summing to variable `s`. One point for correctly retrieving the index value using `*p2`. One point for correctly retrieving the indexed value (dereferencing the sum of pointer `p1` and the retrieved index value). One point for incrementing pointer `p2` and one point for incrementing `i`.

   (b) Short answer: `0x824bfff5`

   Max 3 points. Three points if completely correct. A correct value, but not written in hexadecimal form, gives 2 points. Two points if at least 3 bytes out of the 4 bytes are correct. One point if at least 2 bytes are correct.

2. **Module 2: I/O Systems**

   (a) Short answer: The complete solution is shown below.

```
loop:
        lui     $t0,0xbf88
        lw      $t1,0x60D0($t0)
        srl     $t1,$t1,8               # PART 1 starts
        andi    $t2,$t1,0x7
        andi    $t3,$t1,0x8
        beq     $t3,$zero,do_add
        mul     $t1,$t2,$t2
        j       next                    # PART 1 ends
do_add:
        add     $t1,$t2,$t2
next:
        lw      $t2,0x6110($t0)         # PART 2 starts
        andi    $t2,$t2,0xff00
        or      $t1,$t2,$t1
        sw      $t1,0x6110($t0)         # PART 2 ends
        j       loop
```

i. Max 3 for points. One point for correctly right shifting, and for correctly masking the 3-bit value. One point for the correct `and` operation, to mask for switch 4. One point for correct jumping and performing the multiplication correctly.

ii. Max 3 points. One point for storing the value correctly using `sw`. One point for correctly handling of `andi` and `ori`. One point for `lw`.

(b) Short answer: the prescaling should be `1:8`.

Elaborated answer:

Let p be the prescaling factor. Then,

$$\frac{4\,000\,000}{p} = 500\,000 \tag{1}$$

Hence, $p = 8$ and the prescaling is `1:8`

3. **Module 3: Logic Design (for IS1500 only)**

(a) Short answers:

i. $C = 0$ and $D = 1$.

ii. $C =$ `unknown` and $D =$ `unknown`.

iii. SR latch.

Max 5 points. For item (i), max 2 points. 1 point for each correct signal. For item (ii), max 2 points. 1 point for each correct signal. One point for the explicit answer *SR latch*.

(b) Short answer: 128 bytes.

Elaborated answer: A 6-bit address signal means that there are $2^6 = 64$ registers. Since each register takes 2 bytes (16 bits), the capacity of the register file is $64 \cdot 2 = 128$ bytes.

Max 2 points. Two points for a correct answer in bytes. 1 point if the correct answer is given in bits.

(c) Short answers: $Y_0 = 0$, $Y_1 = 0$, $Y_2 = 1$, and $Y_3 = 0$.

Max 1 point. One point for a correct answer.

4. **Module 4: Processor Design**

(a) Short answer: $A =$ `0x9`, $B =$ `unknown`, $C =$ `0x0`, $D =$ `0x40004010`, and $E =$ `0x0`.

Max 5 points. One point for each correct answer. A note on the correction: signals D and E are known signals, but in this specific execution, the exact signal values do not actually matter. Hence, for signals D and E, an answer `unknown` is also OK.

(b) Short answer:

i. Data hazards: 2-4, 3-4, 5-7

ii. Needs stalling: 3-4

Max 3 points. For (i), two points if all three hazards are found, and at most one incorrect hazards is given. For (i), if at least two correct hazards are found, and at most two incorrect hazards are given, then give 1 point. For (ii), give one point if the correct hazard is given.

5. **Module 5: Memory Hierarchy**

   (a) Short answer: i) 3 bits ii) 6 bits, iii) it is a 2-way associative cache, and iv) 128 valid bits.

   Elaborated answers:

   i. The byte offset field is 3 bits ($2^3 = 8$).
   ii. The set field size is $32 - 3 - 23 = 6$ bits.
   iii. There are $2^6 = 64$ sets and there are $1024/8 = 128$ blocks. Hence, the associativity is $128/64 = 2$.
   iv. There are as many valid bits as there are blocks. Hence, 128 valid bits.

   Max 4 points. One point for each correct answer.

   (b) Short answer: i) $\frac{2}{3}$ ii) 100%, iii) spatial locality.

   Elaborated answers:

   i. There are in total 6 memory accesses. The instruction cache sequence is: miss, miss, hit, miss, hit, miss. Hence, the cache miss rate is $\frac{4}{6} = \frac{2}{3}$.
   ii. Both accesses result in cache misses (different blocks). Hence, the cache miss rate is 100%.
   iii. They show only spatial locality. The same memory address is never accessed twice.

   Max 4 points. Correct answers give full points, that is, two points for (i), one point for (ii), and one point for (iii).

6. **Module 6: Parallel Processors and Programs**

   (a) Short answer: The shortest execution time is 2 seconds.

   Elaborated answer:

   Assume that x is the time of the program that is sequential, and cannot be parallelized. Hence, the part that can be parallelized is $6 - x$. We then get the equation

   $$\frac{6 - x}{4} + x = 3 \tag{2}$$

   If we solve for x, we get $x = 2$. Hence, if we have infinite many cores, the total execution time would be the sequential part, i.e., 2s.

   (b) Short answers:

   i. True.
   ii. False. SIMD corresponds to data-level parallelism.
   iii. False. AVX is a standard for handling SIMD. It is not used for ILP.
   iv. True.
   v. False. MapReduce is a programming model, typically used in distributed computing. VLIW is used as way to achive instruction level parallelism (ILP).

   Max 5 points. One point for each correct answer. The false statements need to have clear motivations.

Part II: Advanced

7. The complete solution is omitted. Please see the lecture slides and the course book.

   For IS1500 students: The question is graded in three levels S, G, and VG, with the following criteria:

   - Satisfactory (S): Some of the concepts in the question are clearly explained.
   - Good (G): Basically all concepts in the question are clearly explained, and some of the concepts are related to each other, by discussing similarities and differences.
   - Very Good (VG): Basically all concepts in the question are clearly explained, and all of the concepts are related to each other, by discussing similarities and differences.

   If none of the three levels is achieved, the exercise is considered as failed (F).

   For IS1200 students (retake exam), max 15 points. For each of the three items, max three points for the explanations of the two concepts, max one point for at least one difference, and max one point for at least one similarity. That is, max five points for each of the three items.

8. The following C code shows an example of a C code solution.

```
void bar(const char* str, int* x){
  if(*str == 0)
    return;

  if(*str == ' ')
      *x = *x + 1;
  else
    return;

  bar(str+1, x);
}

void foo(char* str){
  int len=0;
  bar(str,&len);

  while(1) {
    *str = *(str+len);
    if(*(str+len) == 0)
      break;
    str++;
  }
}
```

   Grading for IS1500 students: The question is graded in three levels S, G, and VG, with the following criteria:

   - Satisfactory (S): Some minor parts of a C or Assembly program are constructed correctly, but there are major errors or missing parts of the program.

- Good (G): The majority of the program is constructed correctly, including the main flow and structure of the program, but there are a number of minor errors within the program.
- Very Good (VG): The program is correct, with basically no errors.

If none of the three levels is achieved, the exercise is considered as failed (F).

For IS1200 students (retake exam), max 20 points.

9. Potential solutions for the three different sub-problems L1, L2, and L3 are given below.

   - L1: The `foo` function removes space characters to the left of the text in the string. This is often called a *left trim* function. The function performs a destructive change, and updates the data directly in memory.
   - L2: Function `bar` starts on an address that points to the last word in one of the cache blocks. Hence, the next instruction (`sw`) is located as the first word in the next cache block. We start by dividing the whole code into 9 blocks, each consisting of 4 instructions. To solve the problem, it is enough to see how many of these 9 cache blocks that are accessed, since one access results in a miss, and any further accesses to that cache block results in a hit. We can now see that when `foo` is called, block 5 and block 6 are accessed. When bar is called, block 1 and 2 are accessed. However, since the first character in the string is a `K` and not a space character, the code jumps over block 3. Then block 4 is accessed, and the `bar` function returns, and continues in block 7. The while loop will iterate in blocks 7 and 8, and then finally return in block 9. Hence, all blocks are accessed, except one (block 3). Hence, there are 8 instruction cache misses.
   - L3: First, we need to check if there can be a conflict between the stack pointer and the string address. The data cache capacity was 4096 bytes, and the block size 4 bytes. Hence, there are 1024 sets, which means that the set field is 10 bits. By just inspecting the addresses, we can see that the set fields will always be different for the string and the stack.

     Starting in function `foo`, there are two memory accesses using the stack pointer (`sw`). Both will miss, because the block size is 4 bytes. In block 6 (using the counting from the previous exercise), we have another `sw` that results in another miss. Accessing the stack in bar gives a miss (block 2). The load byte in block 2 gives the first miss in the character array. The `bar` function then returns, which gives a hit (we have already accessed the stack frame). In block 7, we get two hit, loading back the stack values. In the while loop, we have 10 accesses to the string (the string length 4 + the terminating 0 character). There are 2 cache blocks, but we only get a cache miss in the second block (when accessing the 0 byte) because the load byte in function `bar` has already accessed the first block. Hence, we get 1 miss and 9 hits. Finally, in block 9, we get one cache hit when accessing the stack. Hence, in total, we have 19 data memory accesses and 6 data cache misses. Hence, the data cache miss rate is $\frac{6}{19}$.

For IS1500 students, the question is divided into three analysis tasks, each corresponding to one of three levels of difficulty: L1, L2, and L3. The question is graded in three levels S, G, and VG, with the following criteria:

- Satisfactory (S): The task at level L1 is solved correctly.
- Good (G): Either the task at level L2 or the task at level L3 is solved correctly.
- Very Good (VG): Both the tasks at level L2 and L3 are solved correctly.

If none of the three levels is achieved, the exercise is considered as failed (F).

For IS1200 students (retake exam), max 15 points. Sub-task L1 gives max 4 points, L2 max 5 points, and L3 max 6 points.

Correction of the Exam

The examiner David Broman authored the exam questions, the correction guidelines, and the suggested solutions. The following persons took part in the correction: Saranya Natarajan, Viktor Palmkvist, Daniel Lundén, and Fredrik Lundevall.