Questions

1. Questions on simple MIPS coding

   (a) Translate the following fragment of code into *MIPS* assembly language ((**points 6**)):

   ```
   for(int i=0; i != 100; ; i++) {
     a[i] = b[i+3] - c;
   }
   ```

   assuming the base address of the array `a` is stored in register `s0`, the base address of `b` is stored in register `s1` and `c` is stored in register `s2`. You may use any other normal register in your code.

   (b) With your code, how many instructions will be executed when your code is run. (**points 5**)

   (c) How many times will the main memory be accessed for reading or writing during the execution of your code (**points 5**)

2. Pipelines

   (a) What is an instruction pipeline and how does it attempt to increase the performance of a system (**points 5**)

   (b) What is a pipeline stall, a pipeline stall effect the performance of a processor.

3. Caches and Virtual memory

   (a) What is a memory cache and how does it attempt to increase the performance of a system? (**points 3**)

   (b) Describe the following types of caches: Direct Mapped, n-way associative and set-associative caches. (**points 6**).

   (c) What is the role of the block size in a cache and how does it improve the performance of the cache memory system. (**points 3**).

   (d) Assume you have a direct mapped cache of size 8K with a block size of 16 bytes. Assuming that no elements of the integer array `A`, in the following piece of code, are inside the cache, how many cache misses will the be when the code is executed:

   ```
   for(int i=0; i<100; i=i+3) {
    k += A[i]
   }
   ```

assuming that the variables `k` and `i` are in the Cache. (**points 5**)

(e) Describe what a virtual memory system is, in particular talk about the TLB. (**points 5**)

(f) When you have a virtual memory system and a program request a page which is not in the main memory, if the main memory is full then some page in main memory has to be swapped out back to disk. Describe some bad choices for swapping pages out of main memory. (**points 3**)

4. Input and output.

(a) What is memory mapped I/O? (**points 2**)

(b) Describe with examples the difference between polled I/O and interrupt driven I/O. In particular you should describe the role of control registers in both polling and interrupt driven I/O.(**points 5**)

(c) What tradeoffs are involved in deciding between using polling and interrupt driven I/O? Give some examples.(**points 3**)

5. MIPS coding and pipelines.

(a) Briefly describe the principle of pipelining. (**4 points**)

(b) What causes pipeline stalls?

(c) Why are interrupts and exceptions a problem for the designer of a pipeline? (**2 points**)

(d) Consider the following fragment of C++ code:

```
int dp = 0;
for(int i=0; i<n ; i++ ) {
  dp += A[i] * B[i] ;
}
```

Assume that `dp` is stored in register `$s0`, the base address of the integer array `A` is stored in `$s1` , the base address of `B` is in `$s2` and `n` is stored in `$s3`. Write out the MIPS code corresponding to the fragment of code, you may use any temporary registers you want and the registers `$s0` and `$s0` need not have the original values. Remember with C++ loops if `n` is zero or smaller than the loop body will not execute. How many pipelines stalls will there be with your code. If there are any pipeline stalls rewrite your code with as few pipeline stalls as possible. (**6 points**)

(e) A common compiler optimization is loop unrolling. Suppose in this case it was always known that n was a multiple of two, In this case the compiler might unroll the C++ code to:

```
int dp = 0;
for(int i=0; i<n ; i++) {
 dp += A[i] * B[i];
 i++;
 dp += A[i] * B[i];
}
```

Produce the corresponding MIPS code.

6. Caches and Virtual Memory.

   (a) What it a memory Cache? What principles does it rely on to increase the performance? In particular you should discuss the type of memory access patterns that are conducive to good Cache performance. (**4 points**)

   (b) What are multi-level Caches? (**2 points**)

   (c) 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, n-way associative and fully associative. Describe the pros and cons of each different scheme. (**6 points**)

   (d) What is virtual memory? Describe how it operates and how it interacts with the operating system. (**4 points**)

   (e) Although conceptually virtual memory is very similar to Cache memory there are some very practical differences. Describe why a virtual memory system uses write-back not write-through strategies, describe both strategies. (**2 points**)

   (f) Explain what a *translation-look-aside buffer* is and what it attempts to achieve in virtual memory system. (**4 points**)

7. Given a truth table, for example:

A	B	
0	0	0
0	1	1
1	0	1
1	1	0

By considering the rows that give true:

A	B	
0	1	1
1	0	1

A boolean formula can be obtained by observing that the the truth table is true if and only if the following condition holds:

((A = 0) and (B = 1)) or ((A =1) and (B = 0))

In terms of boolean algebra this translates as:

$$(\overline{A}.B) + (A.\overline{B})$$

where $\overline{A}$ represents the logical negation of A, . represents logical conjunction (and) , and + represents logical disjunction (or). Such a construction of a truth table is referred as the sum of products construction.

In this question you are to design and analysis a simple multiplier.

(a) First construct the truth table for a 2 bit multiplier. That is given two two-bit numbers a_1a_0 and b_1b_0 (where a_0 and b_0 are the least significant bits) you want to produce the binary number $c_3c_2c_1c_0$ which represents the product of a and b. First construct the truth table which gives the number $c_3c_2c_1c_0$ in terms of a_1a_0 and b_1b_0. The truth table for c_3 is as follows (only the lines in the truth table where c_3 is true are shown):

a_1	a_0	b_1	b_0	c_2
1	0	1	0	1
1	0	1	1	1
1	1	1	0	1

The three entries for which c_2 represent the facts that $10_2 \times 10_2 = 2_{10} \times 2_{10} = 4 = 0100_2$, $10_2 \times 11_2 = 0110_2$ and $11_2 \times 10_2 = 3_{10} \times 2_{10} = 6 = 0110_2$. (**6 points**)

(b) Next by using the sum-of-products construction produce a boolean formula for each of c_3, c_2, c_1, c_0 in terms of a_1, a_0, b_1, b_0. If you are able, reduce the size of the expressions. Thus the boolean expression for c_2 would be:

$$c_3 = ((a_1.\overline{a_0}).(b_1.\overline{b_0})) + ((a_1.\overline{a_0}).(b_1.b_0)) + ((a_1.a_0).(b_1.\overline{b_0}))$$

(**6 points**)

(c) Draw four logic diagrams, using only two input gates, for each of the four expressions $c_3, \ldots, c_0$ you derived in the previous part. (**4 points**)

(d) Assuming each gate takes 1ns to compute its answer given a new input, work out the time of the longest path for each of the four circuits. (**4 points**)

(e) Using your construction, estimate the number of gates required to build a n-bit multiplier? (**4 point**)

8. Code the following fragment of code in MIPS assembly language:

```
for(i = 0 ; i<100 ; i++) {
 A[i] = 0; }
for(i=0; i<100 ; i++ ) {
 A[i] = B[100 - i] + 1;
 }
```

Assume that register `$s0` holds i, register `$s1` holds the address of `A[0]` and `$s2` holds the address of `B`. (**points 5)**

9. Caches and Virtual Memory.

(a) What it a memory Cache? What principles does it rely on to increase the performance? In particular you should discuss the type of memory access patterns that are conducive to good Cache performance. (**4 points**)

(b) What are multi-level Caches? (**2 points**)

(c) 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, n-way associative and fully associative. Describe the pros and cons of each different scheme. (**6 points**)

(d) What is virtual memory? Describe how it operates and how it interacts with the operating system. (**4 points**)

(e) Although conceptually virtual memory is very similar to Cache memory there are some very practical differences. Describe why a virtual memory system uses write-back not write-through strategies, describe both strategies. (**2 points**)

(f) Explain what a *translation-look-aside buffer* is and what it attempts to achieve in virtual memory system. (**4 points**)

10. What is the longest path digital circuit and how does it effect the clock speed of a processor? (**2 points**). What is microcode? ((**2 points**) What are the advantages and disadvantages of microcode over directly coding a state-machine in hardware? (**2 points**).

11. Code the following fragment of code in MIPS assembly language:

```
for(i = 0 ; i<100 ; i++) {
 A[i] = A[i]+1; }
for(i=0; i<100 ; i++ ) {
 B[i] = B[100 - i] + 1;
 }
```

Assume that register `$s0` holds i, register `$s1` holds the address of `A[0]` and `$s2` holds the address of `B`. (**points 5**)

12. Caches and Virtual Memory.

(a) What it a memory Cache? What principles does it rely on to increase the performance? performance. (**4 points**)

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

(c) What are multi-level Caches? (**2 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, n-way associative and fully associative. Describe the pros and cons of each different scheme. (**6 points**) interacts with the operating system. (**4 points**).

(e) Why would it be inefficient to have a small page size (say 256 bytes) in a virtual memory system (**2 points**).

(f) Explain what a *translation-look-aside buffer* is and what it attempts to achieve in virtual memory system. (**4 points**)

13. What is the longest path digital circuit and how does it effect the clock speed of a processor? (**2 points**). What is microcode? ((**2 points**) What are the advantages and disadvantages of microcode over directly coding a state-machine in hardware? (**2 points**).

Additional Material from the book

Refers to Tanenbaum.

1. Problems on pages 641-642
2. Page 650-651. You should be able to do Questions 5 and 9 if you had a description of the floating point standard (you don't have to memorise the floating point standard, but be able to understand what it is saying).
3. There might be other types of questions not represented here. There is no substitute for reading the book.