

The University of Alabama in Huntsville
Electrical & Computer Engineering Department
CPE 431 01
Test 1 Solution
Fall 2007

1. (1 point) List one of the design principles given in the book ___Good design demands good compromises or Make the common case fast or Simplicity favors regularity___
2. (1 point) The fastest growing segment of the processor market is ___embedded___
3. (1 point) ___Abstraction___ is a model that renders low-level details of computer systems temporarily invisible to facilitate design of sophisticated systems.
4. (1 point) A ___return address___ is a link to the calling site that allows a procedure to return to the proper address; in MIPS it is stored in register \$ra.
5. (1 point) An ___executable file___ is a functional program in the format of an object file that contains no unresolved references, relocation information, symbol table, or debugging information.
6. (10 points) MIPS chooses to simplify the structure of its instructions. The way we implement complex instructions is to decompose such complex instructions into multiple simpler MIPS ones. The complex instruction `swap $rs, $rt` can be decomposed into three instructions. If the implementation of the `swap` instruction in hardware will increase the clock period of a single-cycle implementation by 10%, what percentage of `swap` operations in the instruction mix would recommend implementing it in hardware?

Let x represent the proportion of instructions that are `swap` instructions.
Consider two cases: before change in which `swap` instructions are pseudo-instructions which are replaced by three instructions, after change in which `swap` instructions are implemented directly in hardware.

$$CPI_{\text{before}} = CPI_{\text{after}} = 1, CT_{\text{after}} = 1.1CT_{\text{before}}, IC_{\text{before}} = (x*3 + (1-x)*1)IC_{\text{after}}$$

$$ET_{\text{before}} = ET_{\text{after}}$$

Substituting for IC_{before} and CT_{after} ,

$$\begin{aligned} CPI_{\text{before}} * IC_{\text{before}} * CT_{\text{before}} &= CPI_{\text{after}} * IC_{\text{after}} * CT_{\text{after}} \\ 1 * (x*3 + (1-x)*1)IC_{\text{after}} * CT_{\text{before}} &= 1 * IC_{\text{after}} * 1.1CT_{\text{before}} \\ 3x + 1 - x &= 1.1 \\ 2x &= 0.1 \\ x &= 0.05 \text{ or } 5\% \end{aligned}$$

7. (10 points) When designing memory systems, it becomes useful to know the frequency of memory reads versus writes as well as the frequency of accesses for instructions versus data. Using the average instruction mix information for MIPS for the program SPEC2000int in the table below, find the following:

- a. The percentage of all memory accesses (both data and instruction) that are for data.
- b. The percentage of all memory accesses (both data and instruction) that are for reads.

Assume that two-thirds of data transfers are loads.

Instruction Class	Frequency
Arithmetic	24 %
Data Transfer	36 %
Logical	18 %
Conditional Branch	18 %
Jump	3 %

a.

$$\frac{\text{DataAccesses}}{\text{InstructionAccesses} + \text{DataAccesses}} = \frac{\text{DataTransferInstructions}}{\text{AllInstructions} + \text{DataTransferInstructions}} = \frac{0.36IC}{IC + 0.36IC} = 26.5\%$$

b.

$$\frac{\text{Reads}}{\text{InstructionAccesses} + \text{DataAccesses}} = \frac{\text{InstructionFetches} + \text{lwInstructions}}{\text{AllInstructions} + \text{DataTransferInstructions}} = \frac{IC + 1/3(0.36)IC}{IC + 0.36IC} = \frac{1.12IC}{1.36IC} = 82.3\%$$

8. (15 points) We wish to compare the performance of two different computers: M1 and M2. The following measurements have been made on these computers:

Program	Time on M1	Time on M2
1	2.0 seconds	1.5 seconds
2	5.0 seconds	10.0 seconds

A user has the following requirements for the two computers: P1 must be executed 1600 times each hour. Any remaining time is used to run P2. If the computer has enough performance to execute program 1 the required number of times per hour, then performance is measured by the throughput for program 2. Which computer is faster for this workload and by how much?

M1: 1600 executions * 2.0 seconds/execution = 3200 seconds, leaving 400 seconds for executions of Program 2 or 400/5 = 80 executions of Program 2

M2: 1600 executions * 1.5 seconds/execution = 2400 seconds, leaving 1200 seconds for executions of Program 2, or 1200/10 = 120 executions of Program 2

M2 does more executions of Program 2 and is 120/80 = 1.5 times as fast for this workload

9. (10 points) Show the IEEE 754 binary representation for the floating-point number 579.25₁₀ in single and double precision.

$$579_{10} = 512_{10} + 64_{10} + 2_{10} + 1_{10} = 2^9 + 2^6 + 2^1 + 2^0 = 1001000011_2$$

$$0.25_{10} = 0.01_2 \text{ So, } 579.25_{10} = 1001000011.01_2$$

Normalizing gives 1.00100001101 × 2⁹

For single precision, we have the sign bit, an 8-bit exponent, and a 23 bit fraction. The bias is 127. 9 = Exponent-Bias, Exponent = 9 + 127 = 136.

So, the sign bit is 0, the fraction is 0010 0001 1010 0000 0000 000, and the exponent is 10001000

For double precision, we have the sign bit, an 11-bit exponent, and a 52 bit fraction. The bias is 1023. Exponent = 9 + 1023 = 1032.

So, the sign bit is 0, the fraction is 0010 0001 1010 0000 0000 0000 0000 0000 0000 0000 0000 0000, and the exponent is 10000001000

10. (15 points) Consider two different implementations, P1 and P2, of the same instruction set. There are six classes of instructions (A, B, C, D, E, and F) in the instruction set. P1 has a clock rate of 4 GHz and P2 has a clock rate of 6 GHz. The average number of cycles for each instruction class for P1 and P2 is as follows:

Class	CPI on P1	CPI on P2
A	1	2
B	2	3
C	3	2
D	4	5
E	3	4
F	2	1

If the number of instructions executed in a certain program is divided equally among the classes of instructions except for class D, which occurs twice as often as each of the others, how much faster is P2 than P1?

$$w_A = w_B = w_C = w_E = w_F = 1/7, w_D = 2/7$$

$$CPI_1 = 1/7*1 + 1/7*2 + 1/7*3 + 2/7*4 + 1/7*3 + 1/7*2 = (1 + 2 + 3 + 8 + 3 + 2)/7 = 19/7$$

$$CPI_2 = 1/7*2 + 1/7*3 + 1/7*2 + 2/7*5 + 1/7*4 + 1/7*1 = (2 + 3 + 2 + 10 + 4 + 1)/7 = 21/7$$

$$CR_1 = 4 \text{ GHz}, CR_2 = 6 \text{ GHz}$$

$$\frac{P_{P2}}{P_{P1}} = \frac{ET_{P1}}{ET_{P2}} = \frac{IC_{P1} * CPI_{P1} * CR_{P2}}{IC_{P2} * CPI_{P2} * CR_{P1}} = \frac{CPI_{P1} * CR_{P2}}{CPI_{P2} * CR_{P1}} = \frac{19/7 * 6}{21/7 * 4} = 1.36$$

11. (20 points) Given the bit pattern:

0000 0000 1001 0100 0011 0000 0001 1011₂

what does it represent, assuming that it is

- a two's complement integer?
- an unsigned integer?
- a single precision floating-point number?
- a MIPS instruction?

- a. b. The answer is the same since the most significant bit is 0, the number is

$$1*2^{23} + 1*2^{20} + 1*2^{18} + 1*2^{13} + 1*2^{12} + 1*2^4 + 1*2^3 + 1*2^1 + 1*2^0 =$$

$$8388608 + 1048576 + 262144 + 8192 + 4096 + 16 + 2 + 1 = 9711635$$

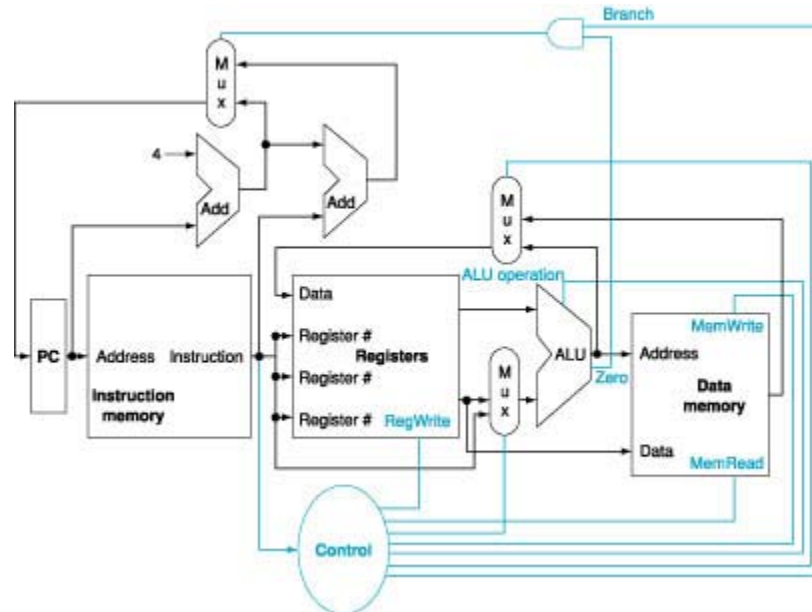
- c. The MSB, 0, is the sign bit. The next 8 bits, 0000 0001, 1 is equal to exponent + bias (127) so the exponent is -126 and the fraction is 001 0100 0011 0000 0001 1011, representing $1*2^{-3} + 1*2^{-5} + 1*2^{-10} + 1*2^{-11} + 1*2^{-19} + 1*2^{-20} + 1*2^{-22} + 1*2^{-23} = 0.1577$. The complete number is $1.1577*2^{-126} = 1.36*10^{-38}$

- d. opcode = 000000, so R-type instruction, rs = 00100, \$4, rt = 10100, \$20, rd = 00110, \$6, shamt = 00000, funct = 11011, 27. The instruction is `divu $6, $4, $20`.

12. (15 points) We wish to add a variant of the `lw` (load word) instruction, which increments the index register after loading word from memory. This instruction (`lw_inc`) corresponds to the following two instructions:

```
lw      $rs, L($rt)
addi    $rt, $rt, 1
```

Explain whether it is possible to implement `l_inc` without making any changes to the register file of the single-cycle datapath. If it is not possible, describe the changes that must be made to the register file.



It is not possible to implement this instruction without modifying the register file. This instruction requires that two registers be written at the same time, the one that gets data from the memory and the one that gets incremented. The current register file has two read ports and one write port, an additional write port is needed, the signals needed would be a register number, a 32 input that comes from the output of an incrementer and a control signal to control the writing on that port.

Extra Credit

13. (1 point) An acronym is a word constructed by taking the initial letters of a string of words.
14. (1 point) A basic block is a sequence of instructions without branches and without branch targets or branch labels.