

Question 1: Short Answers (35 points)

a) (5 points) In what circumstances is throughput the desired performance metric and in what circumstances is latency the desired metric?

b) (5 points) Which of the three factors of CPU time can a compiler influence? Provide examples.

c) (5 points) How much faster than a 1GHz single-cycle MIPS processor would a 3.0GHz Pentium4 x86 processor be if it achieved a CPI of 0.5 on a workload?

d) (10 points) Using only XOR, AND, OR and SLL instructions (and not necessarily all of those), write the shortest possible MIPS program to add two unsigned each of which is either zero or one (i.e. the sum will be either zero, one or two). Assume that the source values are in \$r2 and \$r3 and the result is stored in \$r4.

e) (10 points) The MIPS instruction set provides instructions which support conditional branching (beq, bne, etc.). It is also possible to implement instructions which perform other operations conditionally. One such instruction is a conditional move.

Suppose a new instruction is added to the instruction set:

cmove \$r1, \$r2, \$r3

This instruction works as follows:

```
if($r3 == 0)
    $r1 = $r2;
else
    do nothing
```

In other words, the instruction only performs a move from \$r2 to \$r3 if \$r3 is zero.

Assume that the instruction is used to replace the following two instruction MIPS sequence whenever they are seen.

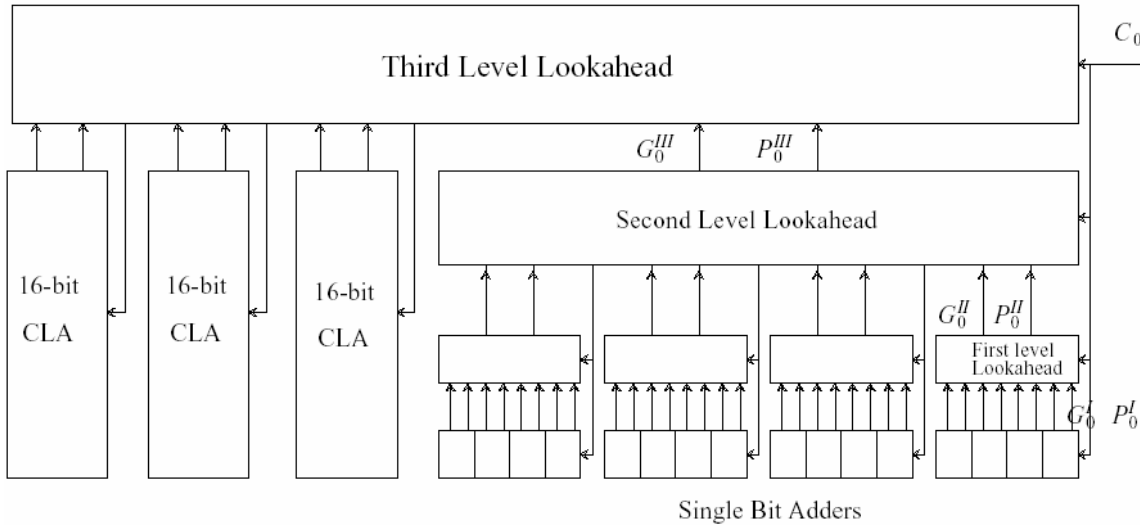
```
    bne $r3, $r0, skip
    move $r1, $r2
skip: ...
```

Suppose that 20% of the conditional branches in a program are used in this manner, so that the two instruction sequence can be replaced with a conditional move. Assume that the CPI of the cmove instruction is the same as the CPI of the move instruction. Thus, the effect of this change is to eliminate the time required by the branch instruction. What is the overall program speedup from this change, given the data in the following table?

Instruction Category	Frequency	CPI
Arithmetic	43%	1.0
Data Transfer	40%	1.4
Conditional Branch	15%	1.8
Other	2%	1.3

Question 2: Arithmetic (30 points)

The figure below shows a 64-bit carry lookahead adder with three levels of lookahead. Some signals are shown in the figure to give you an idea of the structure of the adder; however not all relevant signals are shown (for example the carries from the first level lookahead to the single-bit adders).



a) (15 points) Let X_i , Y_i , C_i and S_i , with $i = 0, 1, \dots, 63$, represent the individual bits of the two operands, the carries, and the sums, respectively. Let G_i^I and P_i^I , $i = 0, \dots, 63$ represent the first level generates and propagates. Let G_j^{II} and P_j^{II} , $j = 0, \dots, 15$ represent the second level generates and propagates, and let G_k^{III} and P_k^{III} , $k = 0, \dots, 3$ represent the third level generates and propagates.

Write the logic equations for the following signals, in the space provided. (The equations should be in terms of the inputs to the logic block for which they represent the outputs.)

$G_3^I =$
$G_2^{III} =$
$P_1^{III} =$
$C_8 =$
$C_{32} =$

b) (6 points) Assume that each gate delay is τ time units, and all gates are available with up to 5 inputs. Further assume that all X_i , all Y_i , and C_0 are ready at time 0. In the table below, show at what time the chosen signal value is ready. Use the comment column for any comments (comments will be used to determine partial credit in case of an incorrect answer).

Signal	Time Ready	Comments
S_2		
S_8		
C_{32}		

c) (9 points) Calculate the number of gates of each type in the lookahead logic (include all levels and do not count gates in the 64 single-bit adders). Show your work.

Name _____
Student ID _____

Page 6

c) (5 points) Assume that memories and the ALU have 2ns delays, and the registers have a 1ns delay. Find the minimum clock cycle times for *both* the original single-cycle datapath and your modified one.

d) (10 points) What general conclusions can you draw, if any, about the performance of the original processor as compared to your modified one?