1.99
3 ⟌ 40

1.21
75 ⟌ 100

$\frac{100}{75} = \frac{4}{3}$

1.17
85 ⟌ 150

$\frac{1}{\frac{0.2}{4} + 0.8} = \frac{1}{0.85} = 1.17$

Department of Computer Science
National Tsing Hua University
CS4100 Computer Architecture
Midterm, Nov.15, 2021
13:20-15:10 PM

0.05 0.25+0.5
0.75

1. (15%) 15
   (a) (5%) Let a processor *A* take 0.45 sec to complete a program. There are 6 × 10^9 instructions in this program and the average CPI is 1.5. What is the clock rate of processor *A* in MHz?
   (b) (5%) You are going to enhance the computer, and there are two possible improvements: either make multiply instructions run four times faster or make memory access instructions run two times faster. You repeatedly run a program 100 seconds. Of this time, 20% is used for multiplication, 50% for memory access instructions, and 30% for other tasks. What will the speedup be if you improve only multiplication?
   (c) (5%) Is it possible to have 1.5 speedup of processor *A* if you improve only memory access? Why?

10 8
30 15
30 ~~12~~
30 15

48
12

2. (20%) Consider the following I-type instruction. Let \$17 = 0x000003A4. \$18 = 0x000004F6 (in base 16) and A= -10 (in base 10). 10

31 opcode 26	25 rs 21	20 rt 16	15 immediate 0
opcode	17	18	A

meanless

   (a) Let the above instruction is located at address 0x1000010C. If it is a *bne* instruction, what is the address of target instruction? 0x000003A4
   (b) If the above instruction is an *addi* instruction, what value is stored in \$18 after the instruction is executed?
   (c) For R-type instruction, why is 5-bit used for the field of *shamt*?
   (d) Let the program counter (PC) be 0x40000200 now. Is it possible to use *jump* instruction in MIPS to jump to the address of 0x60003000 and Why?

3. (10%) You wish to write a subroutine named **FOO**. \$a0 is already used in the caller routine and the caller needs to pass its one argument by this register. Moreover, caller will use registers \$t2 when the subroutine returns and callee will use \$s1. Write MIPS code segments for caller and callee, respectively, to save register values in activation record. 3 \$10

4. (8%) MIPS does NOT provide branch instructions such as blt (branch if less than), bgt (branch if greater than), ble (branch if less and equal than) and bge (branch if greater and equal than). But we can use instructions such as *beq*, *bne* and *slt* in MIPS to execute the above four instructions. Please match (A) to (D) with the following instruction sequences (1) to (4). 8

s1<s2 (A) blt \$s1, \$s2, L (B) bgt \$s1, \$s2, L (C) ble \$s1, \$s2, L (D) bge \$s1, \$s2, L
s1>s2 s1≤s2 s1≥s2

```
(1) slt  $t0, $s1, $s2   s1≥s2       (2) slt  $t0, $s1, $s2   s1<s2
    beq  $t0, $zero, L   D               bne  $t0, $zero, L   A

(3) slt  $t0, $s2, $s1   s2≥s1       (4) slt  $t0, $s2, $s1   s2<s1
    beq  $t0, $zero, L   C               bne  $t0, $zero, L   B
```

5. (15%) Adder design. Let p_i, g_i be computed in one gate delay.

(a) (3%) Design a carry look-ahead adder. Please give the logic equation of pi (propagate term) and gi (generate term). Given cin0, p0, g0, p1, g1, p2, g2, p3, g3. Give the equation to produce $cout_3$.

(b) (9%) Compute the gate delay of $cout_{30}$ for the following 32 bits adder:

i. Carry ripple adder

ii. Cascaded carry look-ahead adder with a block of 8 bits

iii. Multilevel carry look-ahead adder with a block of 8-bit and 2 level

6 (c) (3%) We know that Pi and Gi to each block are produced from the bottom to the top of the tree while carry bits are produced from the top to the bottom. Given cin_0, cin_k must be produced before $cout_{30}$ be produced. What are the values of k? 0 8 16 32

6. (10%) Let *a, b, c* be three 8-bit operands to a carry save adder and their values be *a* = 01000110, b = 11001110 and c = 01101011. (5%) What are the two outputs of this carry save adder? (5%) Given the above carry save adder as basic building block, please show the block diagram of 16 × 16 Wallace-tree multiplier with minimum delay.

7. (10%) Let a single precision floating point number be represented as 32 bits, where bit 31 is a sign-bit, bits 30 to 23 represent the exponent, bit 22 to 0 the significant, bias 127 for exponent and hidden 1 are used (IEEE 754). 0.3-1875

(a) By the above IEEE 754, how to represent a decimal number -36.1875.

(b) What is the largest number represented (IEEE 754) in binary? 8

8 8. (12%) There are four design principles:1. Simplicity favors regularity, 2. Smaller is faster, 3. Make the common case fast, 4. Good design demands good compromise. Please give one design example of MIPS instruction for each of the principle. 64 32 16 8 2

8 0
4 0
2
132

-1.361875×2

0

36 0 - ~~1875~~ 0

~~.55~~
~~.05~~
~~.55~~

47

0 0 ~~.15~~

-100100.0011

0.062
16 / 100
96
40
32

0.065
16 / 100
96
40

$\frac{1}{2}$ $\frac{1}{4}$ $\frac{1}{8}$ $\frac{1}{16}$

0.125
0.0065
0.1315

.1875
.125
0.0625

2 | 36 0
2 | 18 0
2 | 9 1
2 | 4 0
2 | 2 0
1

0.1875 0
0.3750 0
0.7500 0
1.5000 1
1.000 1

64 32 16 8 4 2 1

$-1.00100\,0011 \times 2^5$

5

127
5
132