

## 中山大学本科生期中考试

考试科目:《计算机组成原理》

学年学期: 2023 学年第一学期

姓 名: \_\_\_\_\_

学 院/系: 计算机学院

学 号: \_\_\_\_\_

考试方式: 闭卷

年级专业: \_\_\_\_\_

考试时长: 120 分钟

班 别: \_\_\_\_\_ / \_\_\_\_\_

**警示** 《中山大学授予学士学位工作细则》第八条: “考试作弊者, 不授予学士学位。”

-----以下为试题区域, 总分 100 分, 考生请在答题纸上作答-----

### 一、填空 (10 分)

1.  $10101_2$  除以  $11_2$ , 商 1010 余数是 1.
2. 计算机实现有符号数除法运算, 要求余数与被除数符号一致       , 否则会产生商的绝对值得到不同结果。
3. 在计算机浮点数运算中  $x + (y + z) = (x + y) + z$  等式        (成立, 不成立)。
4. 运算器的两个主要功能是: 算术运算, 逻辑运算。
5. 两个二进制表示的无符号数进行乘法运算  $10011010 \times 01110111$ , 用移位加(“shift and add”)和布斯算法(Booth's Algorithm), 那么储存这两个数相乘的积需要 16 位寄存器, 标准移位加乘法器需要进行 8 次循环, 实际加法 6 次, 如果采用布斯算法, 实际进行加法运算 2 次, 进行了 2 次减法运算。

### 二、数据表示 (20 points)

1. 用6位二进制补码算法进行两个数的加法:  $(-9) + (-10)$ . 填下表空白处: (i) 每个数的二进制补码表示 (ii) 它们的和, 以及 (iii) 结果转化为10进制.

<i>Decima I</i>	<i>Action</i>	<i>6-bit Binary</i>
-9	Convert to binary	
-10	Convert to binary	
	Sum in binary	
	Convert to decimal	

<i>Decima</i>	<i>Actio</i>	<i>6-bit Binary</i>
-9	Convert to binary	110111



- 1) 测量处理器性能，数据传送、立即数、分支指令需要更多的周期数(lw, addi, lui, sw, beq, etc.)，如下表所示:

Instruction Type	Clock Cycles
Data Transfer / Immediate / Branch	2
All Other	1

当运行benchmark program P，它执行10 Million 条指令显示这些指令30% 是表格中第一种传送类等的指令, 70% 是其它类型指令.

- (a) 运行P 程序需要多少时钟周期数? (5 Points)

Clock Cycles =  $0.7 \times 10^6 \text{ instr.} \times 1 \text{ cycle/instr} + 0.3 \times 10^6 \text{ instr.} \times 2 \text{ cycle/instr} = 13 \times 10^6$   
clock cycles

- (b) 时钟频率500MHz，程序P执行时间多少? (5 Points)

Exec. Time =  $13 \times 10^6 \text{ cycles} / 500 \times 10^6 \text{ cycles/sec} = 0.026 \text{ sec}$

- 2). 假设编译器有两个选择，5 条或 6 条指令，如下表所示:

Class	A	B	C
CPI for class	1	2	3
IC in sequence 1	2	1	2
IC in sequence 2	4	1	1

每个选择的 CPI 是多少？ 哪个更好? ( Same computer) (5 Points)

Sequence 1: IC = 5

- Clock Cycles=  $2 \times 1 + 1 \times 2 + 2 \times 3 = 10$
- Avg. CPI =  $10/5 = 2.0$

Sequence 2: IC = 6

- Clock Cycles=  $4 \times 1 + 1 \times 2 + 1 \times 3 = 9$
- Avg. CPI =  $9/6 = 1.5$

Sequence 2 的周期数较小，同一计算机，时钟周期相同，它所化时间较少，故更好.

#### 四. MIPS 指令 (20 Points)

1. 假设数组 `int* arr = {1,2,3,4,5,6,0}`.让数组首地址 `arr` 是 4 的倍数存在寄存器 `$s0` 里.下列程序做什么?(5 pts)

a) lw \$t0, 12(\$s0) add \$t1, \$t0, \$s0 sw \$t0, 4(\$t1) <b>arr[2] &lt;- 4;</b>	d) addiu \$t0, \$0, 12 sw \$t0, 6(\$s0) <b>alignment error;</b>
b) addiu \$s1, \$s0, 27 lh \$t0, -3(\$s1) <b>\$t0 &lt;- 0;</b>	e) addiu \$t0, \$0, 8 sw \$t0, -4(\$s0) <b>out of bounds;</b>
	f) addiu \$s1, \$s0, 10

c) addiu \$s1, \$s0, 24 lh \$t0\$, -3(\$s1)      alignment error;	addiu \$t0, \$0, 6 sw \$t0, 2(\$s1)      arr[3] <- 6;
--	--

2. 下列汇编程序读无符号32位数从数组，计算结果放到 \$v0 返回. 这个程序至少包含一条伪指令，它可能转化成多条真正指令.

```

main: la $t0, 0x40000          # t0 = base address of array
      lw $v0, 0($t0)          # initialize v0
loop: lw $t1, 0($t0)
      beq $t1, $zero, done
      slt $t2, $t1, $v0
      beq $t2, $zero, endl
      add $v0, $t1, $zero
endl: addi $t0, $t0, 4
      j loop
done:      # end of program

```

数组的首地址是 . 下图是数组（及它附近）在内存存储内容:

### Memory Address (hexadecimal)

这个程序在汇编后将占多少内存空间，以字为单位（ 32-bit memory words ）？ (Be careful!)（5 分）

地址	数据
0x0003FFFC	3
0x00040000	10
0x00040004	4
0x00040008	-1
0x0004000C	0
....	

2 instructions for la + 8 instructions = 10 words

1. 这个程序 LOOP 执行多少次？（5 分）

The program enters the loop 4 times, but exits the last iteration immediately after the “lw” at loop:

2. 填空 每次循环后 \$t0-t2 和 \$v0 的值，以十进制或十六进制表示。（10 分）

loop1	\$t0=	\$t2=
	\$t1=	\$v0=
loop2	\$t0=	\$t2=
	\$t1=	\$v0=



2) Conditional branches they have a longlatency path that goes through I-memory,Registers, Mux, and ALU to compute the PCSrc condition, Mux. The critical path is the longer of the two, and the path through PCSrc is longer for these latencies:

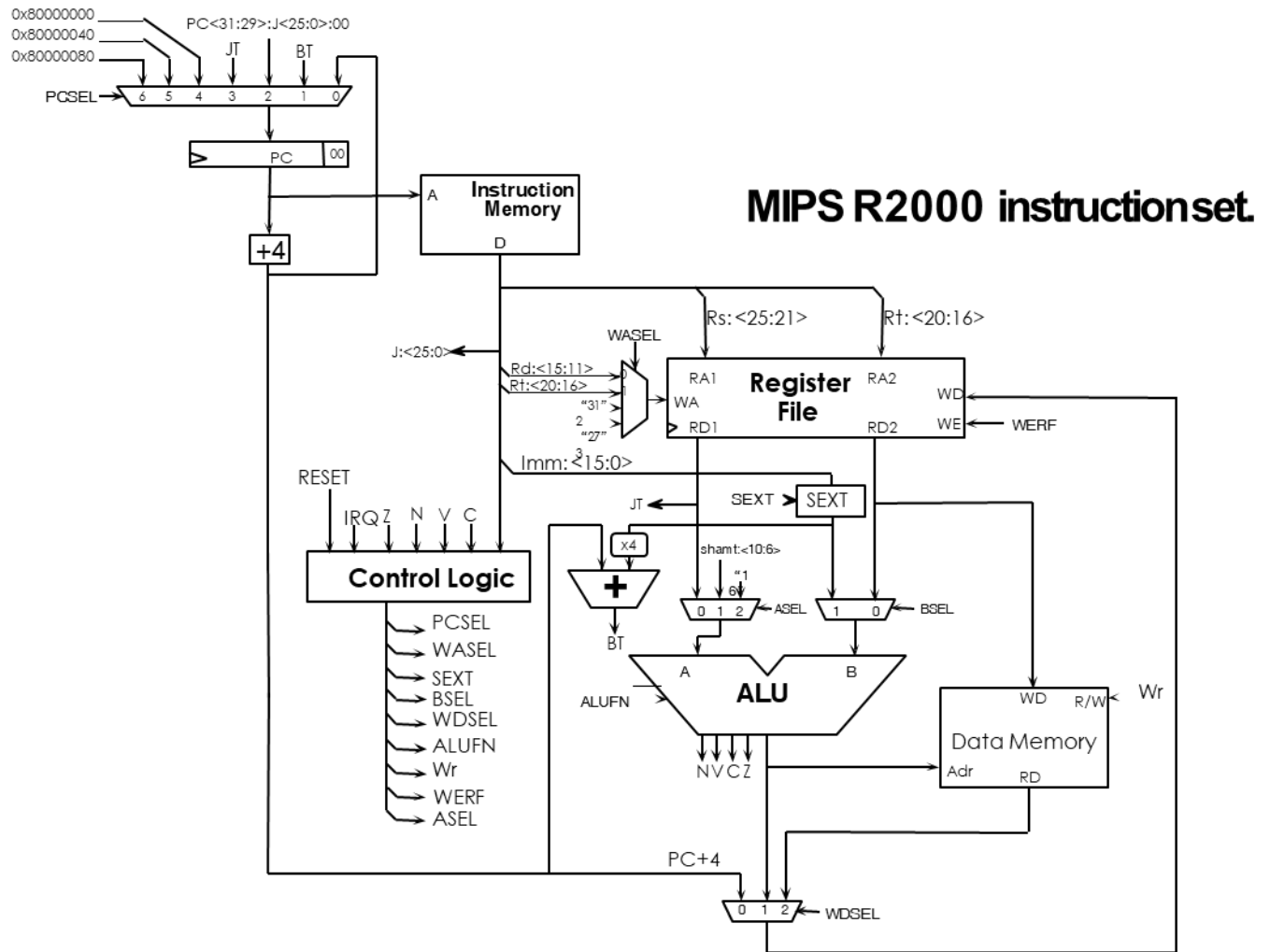
$$200 + 80 \text{ ps} + 20 \text{ ps} + 80 \text{ ps} + 20 \text{ ps} = 400 \text{ ps}$$

$$3) 200\text{ps} + 20\text{ps} + 80 \text{ ps} + 250\text{ps} + 80 \text{ ps} + 20 \text{ ps} = 650 \text{ ps}$$

4.单周期非流水线Mips 数据通路如下图所示，给出每条指令的控制信号取值 **(10points total)**

{0, 1, "don't care" ,"X" signifies "don't care."}.

ALUFN<2:0>	ALU Operation
000	Add
001	Subtract
100	And
101	Or
010	Subtract (set-on-less-than)



	WDSEL	WERF	SEXT	ASEL	BSEL	ALUFN	Wr	WASEL	PCSEL
<b>add rd, rs, rt</b>	1	1	<u>x</u>	0	0	000	0	0	0
<b>ori rt, rs, imm</b>			0					1	0
<b>beq rs, rt, lable</b>	x,				0		0		
<b>lw rt, imm(rs)</b>					1			1	0
<b>sw rt, imm(rs)</b>	x	x		0	1				
<b>Slt rd,rs,rt</b>					0			0	0

	WDSEL	WERF	SEXT	ASEL	BSEL	ALUFN	Wr	WASEL	PCSEL
<b>add rd, rs, rt</b>	1	1	<u>x</u>	0	0	000	0	0	0
<b>ori rt, rs, imm</b>	1	1	0	0	1	101	0	1	0



beq rs, rt, lable	x,	x,	1	0	0	001	0	x	1
lw rt, imm(rs)	2	1	1	0	1	000	0	1	0
sw rt, imm(rs)	x	x,	1	0	1	000	1	x	0
Slt rd,rs,rt	1	1	<u>x</u>	0	0	010	0	0	0

下表给出不同指令的在程序中占比

Table 1: Instruction mix

ALU	Load	Store	Branch	Jump
25%	22%	8%	25%	20%

1. 使用数据存储器的指令占比多少？
2. 如果是多周期CPU，它的平均CPI是多少？

Answer:

1. Only Load and Store use data memory.  
30%

$$2. 0.25 \times 4 + 0.22 \times 5 + 0.08 \times 4 + (0.25 + 0.2) \times 3 = 4.77$$