

第五次理论作业

—

4.1

4.1.1

RegWrite MemRead Branch 都为 0。

ALUop 为 10 -> ADD。

MemWrite 为 1。

4.1.2

除处理分支指令的加法器之外的其它所有单元。

4.1.3

处理分支指令的加法器。

无。

4.2

4.2.1

指令存储器，寄存器堆，加法器 ALU，数据存储器。

4.2.2

无需增加。

4.2.3

无需增加。

4.3

4.3.1

按照用时最长指令 lw 来算，改进前： $400+200+30+120+350+30=1130ps$ 。
改进后为 $1430ps$ 。

4.3.2

$$\frac{1130}{1430} \times \frac{1}{0.95} = 0.8318$$

4.3.3

原价 $1000 + 2 \times 30 + 3 \times 10 + 100 + 200 + 2000 + 500 = 3890$
改进后为 4490 。

$$\text{因 } \frac{3890}{1130} = 3.4425 > \frac{4490}{1430} = 3.1399$$

虽然每 $1ps$ 成本降低，但是用更多的钱换来更低性能。

4.4

4.4.1

$200ps$

4.4.2

$200+15+10+70+20=315ps$

4.4.3

$200+90+20+90+20=420ps$

4.4.4

相对跳转指令。

4.4.5

无条件相对跳转指令。

4.4.6

分支发生时用时 $15+10+70=95\text{ps}$

分支不发生时用时 $90+20+90=200\text{ps}$

在该单元延迟低于 $10+105=115\text{ps}$ 时对时钟周期无影响。高于时影响。

三

2.23

`slt $t2, $0, $t0` 将 `$t2` 设为 1。因此 `bne` 后跳入 ELSE，`$t2` 增加 2。
因此 `$t2` 值为 3。

2.25

2.25.1

I 型指令。

2.25.2

```
addi $t2, $t2, 1
beq $t2, $0, loop
```

四

OF：即 Overflow Flag，判断最高位是否产生溢出。是将两个源操作数当成有符号数看时：
若两个正数相加产生负数、或两个负数相加产生正数，则此标志位为 1。

SF：即 Sign Flag，判断结果的正负。若结果的最高位为 1 则置 1。只对符号数有意义。

CF：即 Carry Flag，判断最高位是否产生进位或借位。相当于无符号数的溢出判断。只对无符号数有意义。

ZF：即 Zero Flag，判断结果是否为 0。若是则置 1。

下面用 `src1` `src2` 表示两个源操作数。以 `res = src1 - src2` 表示结果。

对于有符号数：(1)若 $ZF = 1$ ，那么两数相等；
(2)若 $SF = 1$ 且 $OF = 0$ ，也即结果为负数且不溢出，则 $src1 < src2$ ；
(3)若 $SF = 0$ 且 $OF = 0$ ，也即结果为正数且不溢出，则 $src1 > src2$ ；
(4)若 $SF = 1$ 但 $OF = 1$ ，也即结果为负数但向上溢出，则 $src1 > src2$ ；
(5)若 $SF = 0$ 但 $OF = 1$ ，也即结果为正数但向下溢出，则 $src1 < src2$ 。

对于无符号数：(1)若 $ZF = 1$ ，那么两数相等；
(2)若 $CF = 1$ ，也即结果溢出(最高位借位)，则 $src1 < src2$ ；
(3)若 $CF = 0$ ，也即结果不溢出(最高位不借位)，则 $src1 > src2$ ；

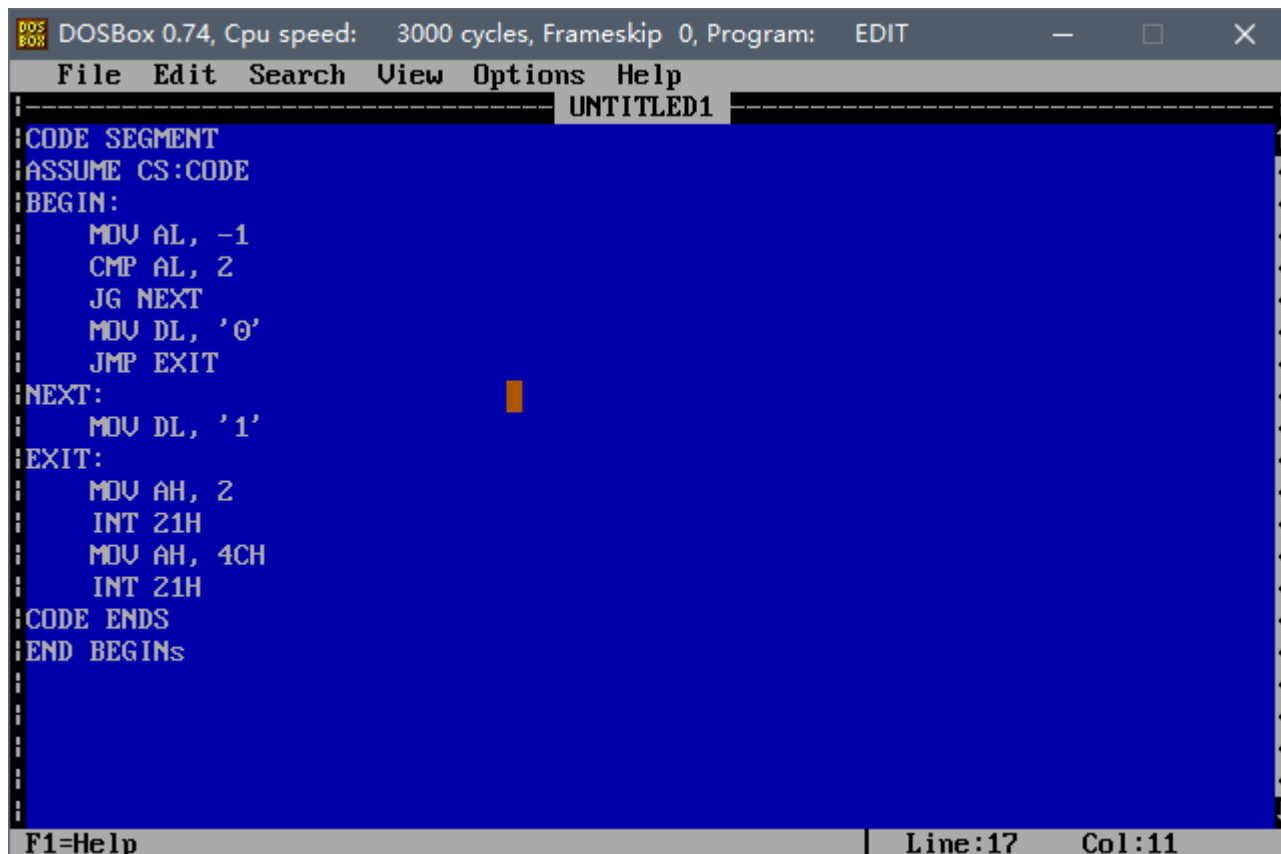
五

区别仅在于 `JG NEXT` 和 `JA NEXT` 一句，分别是有符号大于则跳转及无符号大于则跳转。

第一个程序判断有符号数的 $-1 > 2$ ，为小于，故 `JG NEXT` 无效，`DL` 设置为 `'0'`。之后跳转到 `EXIT`，输出为 `'0'`。

寄存器：`AL = -1` `ZF = 0` `SF = 1` `OF = 0` `DL = '0'` `AH = 4CH`

下面三张图是代码截图、运行截图、寄存器截图：



```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: EDIT
File Edit Search View Options Help
UNTITLED1
CODE SEGMENT
ASSUME CS:CODE
BEGIN:
    MOV AL, -1
    CMP AL, 2
    JG NEXT
    MOV DL, '0'
    JMP EXIT
NEXT:
    MOV DL, '1'
EXIT:
    MOV AH, 2
    INT 21H
    MOV AH, 4CH
    INT 21H
CODE ENDS
END BEGINs
F1=Help | Line:17 Col:11
```

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX
0 Severe Errors

C:\ASM>LINK JG.OBJ

Microsoft (R) Overlay Linker Version 3.69
Copyright (C) Microsoft Corp 1983-1988. All rights reserved.

Run File [JG.EXE]:
List File [NUL.MAP]:
Libraries [LIB]:
LINK : fatal error L1093: JG.OBJ : object not found

C:\ASM>LINK JG1.OBJ

Microsoft (R) Overlay Linker Version 3.69
Copyright (C) Microsoft Corp 1983-1988. All rights reserved.

Run File [JG1.EXE]:
List File [NUL.MAP]:
Libraries [LIB]:
LINK : warning L4021: no stack segment

C:\ASM>JG1
0
C:\ASM>s

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
C:\ASM>DEBUG JG1.EXE
-U
076A:0000 B0FF      MOV     AL,FF
076A:0002 3C02      CMP     AL,02
076A:0004 7F05      JG      000B
076A:0006 B230      MOV     DL,30
076A:0008 EB03      JMP     000D
076A:000A 90          NOP
076A:000B B231      MOV     DL,31
076A:000D B402      MOV     AH,02
076A:000F CD21      INT     21
076A:0011 B44C      MOV     AH,4C
076A:0013 CD21      INT     21
076A:0015 76EC      JBE     0003
076A:0017 6A          DB      6A
076A:0018 00E8      ADD     AL,CH
076A:001A 7AEF      JPE     000B
076A:001C FF76EC     PUSH    [BP-14]
076A:001F FF76EC     PUSH    [BP-14]
-G0013
0
AX=4C30 BX=0000 CX=0015 DX=0030 SP=0000 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=076A IP=0013  NV UP EI NG NZ NA PO NC
076A:0013 CD21      INT     21
-s
```

由寄存器截图，ZF SF OF 标志位分别是 NZ NG NV，对应上述。

第二个程序判断无符号数的 $-1 \geq 2$ ，为大于，故 JG NEXT 有效，DL 设置为 '1'。之后跳转到 EXIT，输出为 '1'。

寄存器： AL = -1 ZF = 0 CF = 0 DL = '0' AH = 4CH

下面三张图是代码截图、运行截图、寄存器截图：

The screenshot shows a DOSBox 0.74 window with the CPU speed at 3000 cycles and Frameskip at 0. The program being run is EDIT, editing the file C:\ASM\JA1.ASM. The assembly code is as follows:

```
CODE SEGMENT
ASSUME CS:CODE
BEGIN:
    MOV AL, -1
    CMP AL, 2
    JA NEXT
    MOV DL, '0'
    JMP EXIT
NEXT:
    MOV DL, '1'
EXIT:
    MOV AH, 2
    INT 21H
    MOV AH, 4CH
    INT 21H
CODE ENDS
END BEGINs
```

The status bar at the bottom of the editor window shows "F1=Help", "Line:17", and "Col:11". Below the editor window, the DOSBox window shows the output of the Microsoft Macro Assembler (MASM) Version 5.00. The output includes the object filename [JA1.OBJ], source listing [NUL.LST], and cross-reference [NUL.CRF]. It also shows the symbol space free (51694 + 464850 Bytes) and the number of warning and severe errors (0 each). The linker output shows the Microsoft Overlay Linker Version 3.69, which generated the executable file [JA1.EXE]. The linker output also includes the list file [NUL.MAP], libraries [LIB], and a warning L4021: no stack segment.

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: EDIT
File Edit Search View Options Help
C:\ASM\JA1.ASM
CODE SEGMENT
ASSUME CS:CODE
BEGIN:
    MOV AL, -1
    CMP AL, 2
    JA NEXT
    MOV DL, '0'
    JMP EXIT
NEXT:
    MOV DL, '1'
EXIT:
    MOV AH, 2
    INT 21H
    MOV AH, 4CH
    INT 21H
CODE ENDS
END BEGINs
F1=Help Line:17 Col:11
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987. All rights reserved.
Object filename [JA1.OBJ]:
Source listing [NUL.LST]:
Cross-reference [NUL.CRF]:
51694 + 464850 Bytes symbol space free
0 Warning Errors
0 Severe Errors
C:\ASM>LINK JA1.OBJ
Microsoft (R) Overlay Linker Version 3.69
Copyright (C) Microsoft Corp 1983-1988. All rights reserved.
Run File [JA1.EXE]:
List File [NUL.MAP]:
Libraries [LIB]:
LINK : warning L4021: no stack segment
C:\ASM>JA1
1
C:\ASM>s
```

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
C:\ASM>DEBUG JA1.EXE
-U
076A:0000 B0FF      MOV     AL,FF
076A:0002 3C02      CMP     AL,02
076A:0004 7705      JA      000B
076A:0006 B230      MOV     DL,30
076A:0008 EB03      JMP     000D
076A:000A 90          NOP
076A:000B B231      MOV     DL,31
076A:000D B402      MOV     AH,02
076A:000F CD21      INT     21
076A:0011 B44C      MOV     AH,4C
076A:0013 CD21      INT     21
076A:0015 B104      MOV     CL,04
076A:0017 D3E0      SHL     AX,CL
076A:0019 0346F6     ADD     AX,[BP-0A]
076A:001C 80E401      AND     AH,01
076A:001F 8946EC      MOV     [BP-14],AX
-G0013
1
AX=4C31 BX=0000 CX=0015 DX=0031 SP=0000 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=076A IP=0013  NU UP EI NG NZ NA PO NC
076A:0013 CD21      INT     21
-s
```

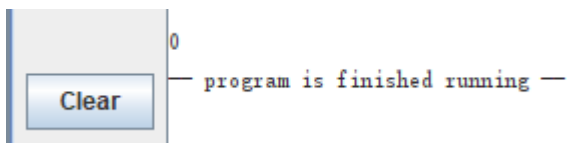
由寄存器截图，ZF CF 标志位分别是 NZ NC，对应上述。

六

对第一个程序，MIPS 汇编改写如下：

```
JG1.asm*
1  .data
2  zero: .asciiz "0"
3  one: .asciiz "1"
4
5  .text
6  .globl main
7
8  main:
9  li $t0, -1
10 li $t1, 2
11 bgt $t0, $t1, NEXT
12 la $a0, zero
13 j EXIT
14
15 NEXT:
16 la $a0, one
17
18 EXIT:
19 li $v0, 4
20 syscall
21 li $v0, 10
22 syscall
```

运行结果为 0，如下：



对第二个程序，MIPS 汇编改写如下：


```
JA1.asm
1  .data
2  zero: .asciiz "0"
3  one: .asciiz "1"
4
5  .text
6  .globl main
7
8  main:
9  li $t0, -1
10 li $t1, 2
11 bgtu $t0, $t1, NEXT
12 la $a0, zero
13 j EXIT
14
15 NEXT:
16 la $a0, one
17
18 EXIT:
19 li $v0, 4
20 syscall
21 li $v0, 10
22 syscall
```

运行结果为 1，如下：

```
Clear
1
— program is finished running —
```