实验二：DLX/MIPS/RISC-V 指令格式

19281171　　王雨潇

实验报告要求：实验报告用 Word 排版，文件内一定要有"学号姓名_学号"。实验报告应提交到课程平台，按时提交。实验报告的内容应尽量全面，避免简单化。简述实验内容，实验过程，实验结果，实验分析，心得体会等。答每道题前应将题目拷贝到实验报告上。

一、 实验内容

本次实验的主要目的是熟悉 DLX-MIPS 的指令格式，并初步了解 RISC-V 的指令格式。

二、 实验过程

实验分为两个部分：

1. 熟悉 winDLX 模拟器，并完成以下实验及分析：

A. 参考提供的资料，学习 winDLX 模拟器的各窗口及各项功能。

B. 运行 FACT.S, GCM.S, PRIM.S 程序，了解各程序的功能。要求详细分析 PRIM.S 程序的指令及功能（Obersve the memory areas in bytes for prime numbers）。

C. 对于 FACT.S 程序，请从中选出 3-5 条不同的指令，并分别指出它是哪种指令（R-type，I-type，还是 J-type），并参照教科书 2-28 页中的图 2.19（如下所示），填写指令格式中各个域的二进制值。为了清楚起见，最好用填表的形式。注：分析指令各域应该填写数值的原因。

Format	**Bits**					
	0　　5	**6　　10**	**11　　15**	**16　　20**	**21　　25**	**26　　31**
R-type	**0x0**	**Rs1**	**Rs2**	**Rd**	***unused***	***opcode***
I-type	***opcode***	**Rs1**	**Rd**	***immediate***		
J-type	***opcode***	***value***				

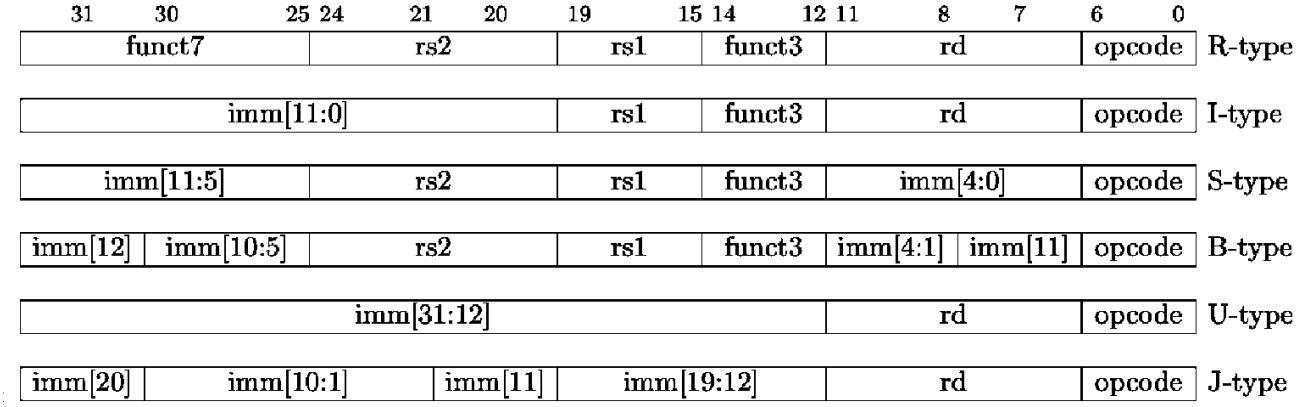
D. 请通过 CODE 窗口的各种选项查看数据，进而判断出 DLX 是 big-endian 还是 little-endian，为什么（结合实验中的观察分析）？是否是对齐的（aligned）？

2. 了解 RISC-V 的在线模拟器 Venus 的使用方法。包括如下内容：

A. 参考提供的资料，学习 RISC-V 的在线模拟器 Venus 的用法。

B. 程序 ex1.s 的功能是什么？程序运行完后，得出的结果是什么？

C. 对于 ex1.s 程序，请从中选出 3-5 条不同的指令，并对于其中每条指令，指出它是哪种指令。请注意：RISC-V 的指令格式分 6 种（除了 R-type，I-type，J-type，还有 U-type, S-type, B-type），并填写指令格式中各个域的二进制值。

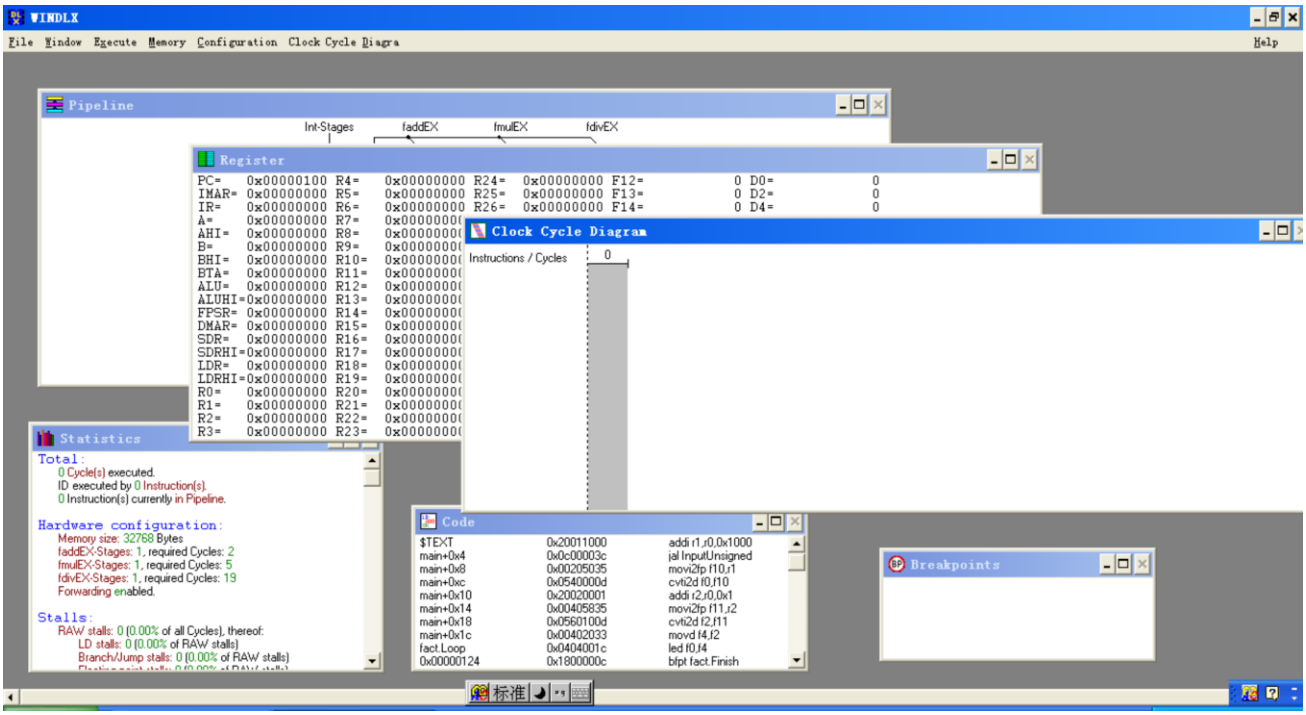


三、 实验结果与分析

1. winDLX 模拟器部分

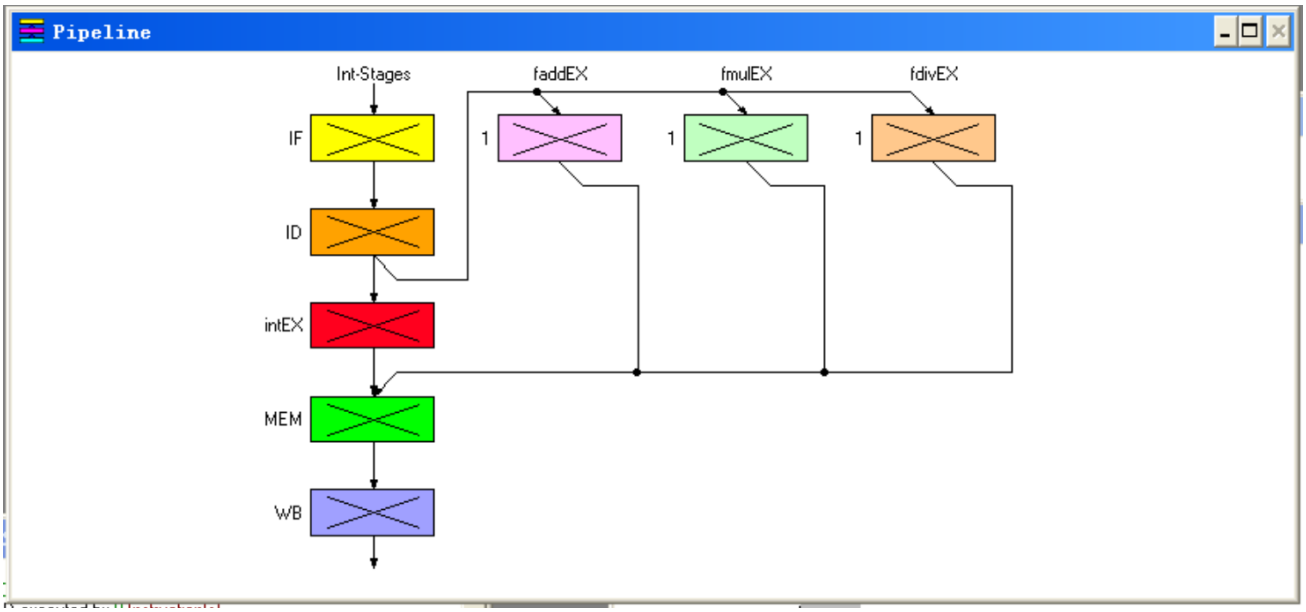
A. 参考提供的资料，学习 winDLX 模拟器的各窗口及各项功能。

答：如下图所示，winDLX 模拟器的初始窗口共有 6 个：

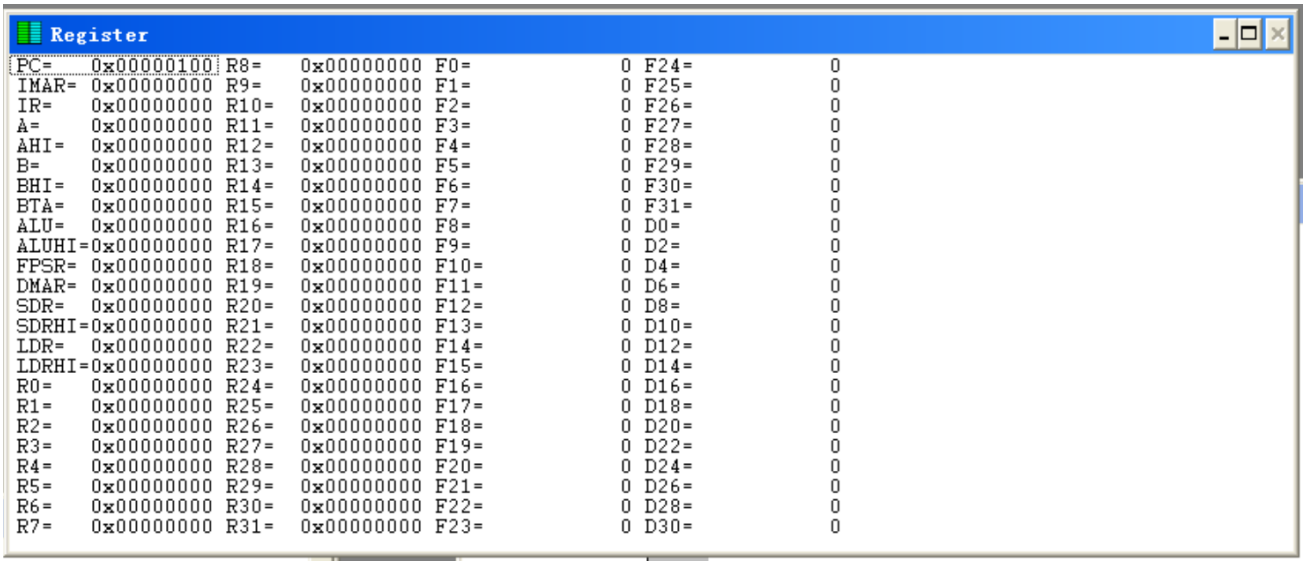


其中：

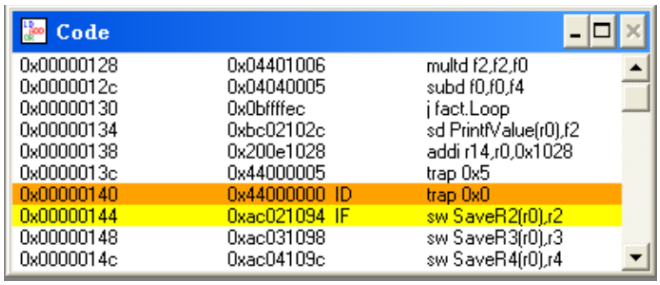
（1） Pipeline 窗口显示 DLX 的流水段和操作单元情况；



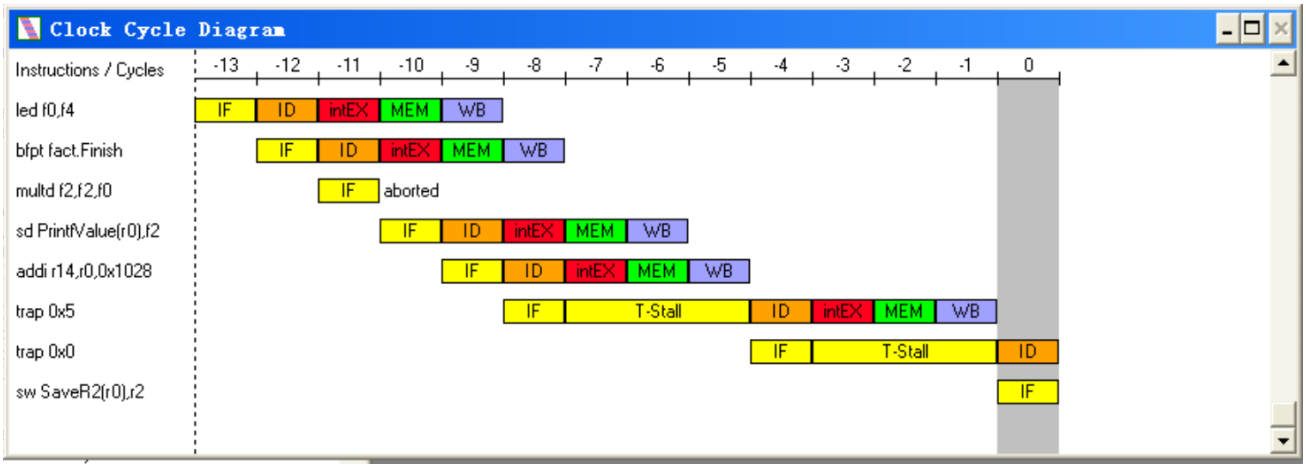
（2） Register 窗口显示各寄存器内容；



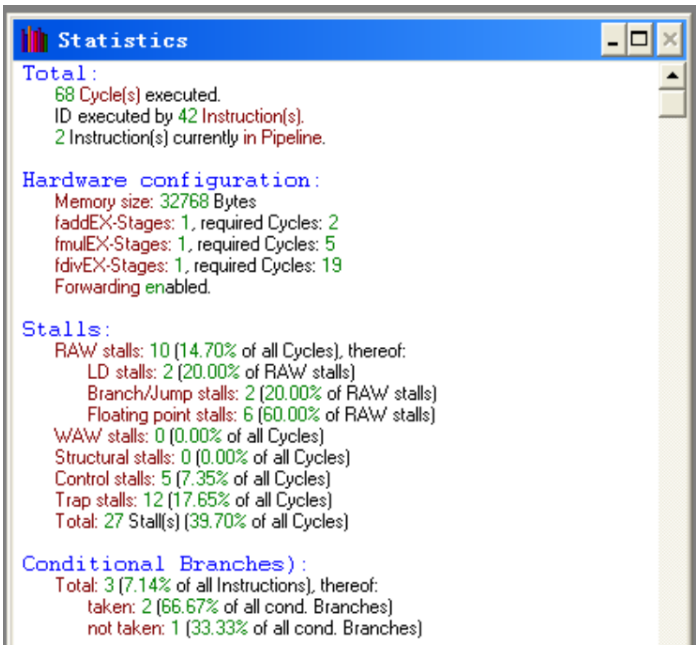
（3） Code 窗口每行显示地址，机器码，汇编代码，可以按 F7 单步调试



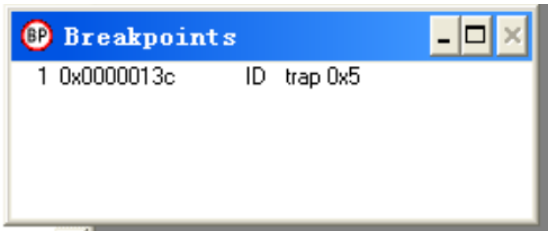
（4） Clock Cycle Diagram 窗口显示时空图



（5） Statistics 窗口是统计信息

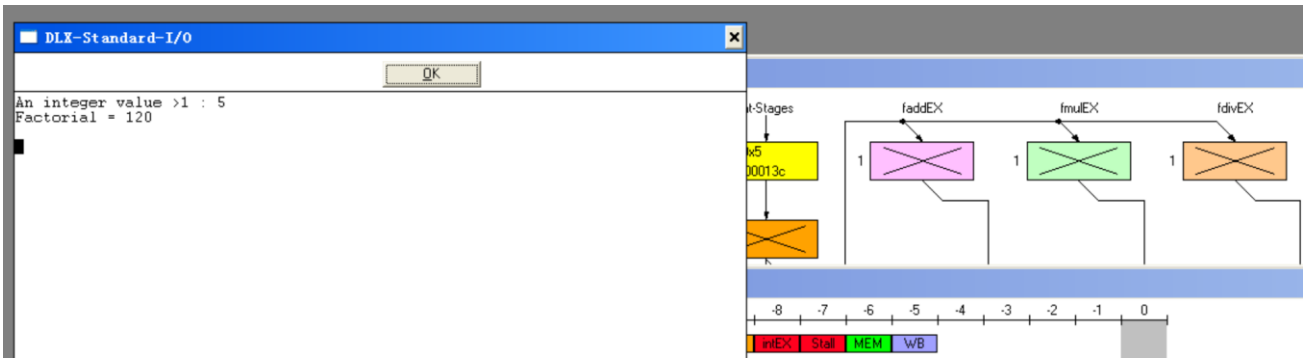


（6） Breakpoints 用于显示设置的断点

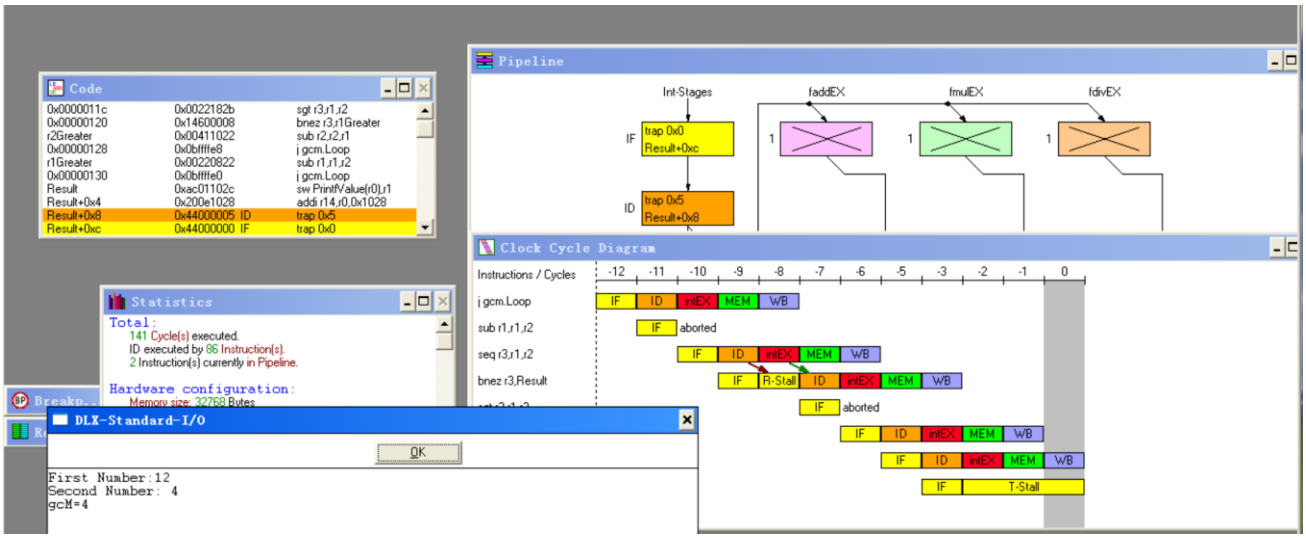


B. 运行 FACT.S, GCM.S, PRIM.S 程序，了解各程序的功能。要求详细分析 PRIM.S 程序的指令及功能（Obersve the memory areas in bytes for prime numbers）。

答：FACT.S 的功能是计算一个整数的阶乘；



GCM.S 的功能是求两个整数的最大公约数；



PRIM.S 程序的功能是求素数，详细分析如下

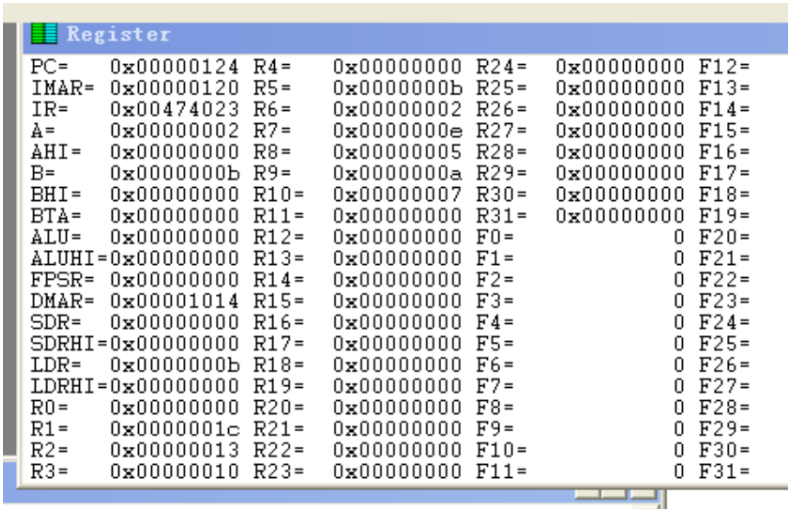
（1） 程序源码、以及我添加的注释如下所示：

```
;*********** WINDLX Exp.2: Generate prime number table *************
;*********** (c) 1991 Günther Raidl            *************
;*********** Modified 1992 Maziar Khosravipour         *************

;-----------------------------------------------------------------------
; Program begins at symbol main
; generates a table with the first 'Count' prime numbers from 'Table'
;-----------------------------------------------------------------------

        .data
        ;*** size of table
        .global     Count
Count:
        .word       10
        .global     Table
Table:
        .space      Count*4
        ; wyx: 此处定义两个全局变量 Count 和 Table, 初始值分别为 10 和 40

        .text
        .global main
main:
        ;*** Initialization
        addi        r1,r0,0     ;Index in Table
        addi        r2,r0,2     ;Current value
```

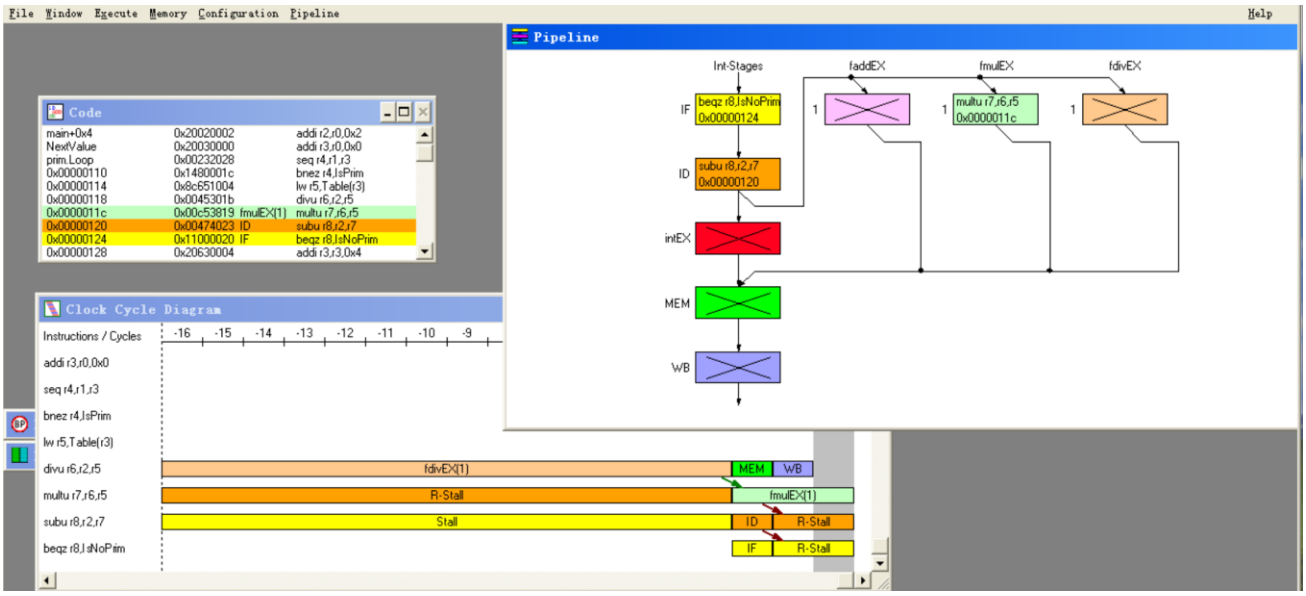
```
        ; wyx: 进入主程序，首先令 R1 = 0, R2 = 2
        ; 下方自带注释说明目标是判断 R2 能不能被 Table 中的数值整除

        ;*** Determine, if R2 can be divided by a value in table
NextValue:
        addi        r3,r0,0     ;Helpindex in Table
Loop:
        seq     r4,r1,r3    ;End of Table?
        bnez        r4,IsPrim   ;R2 is a prime number
        lw      r5,Table(R3)
        divu        r6,r2,r5
        multu       r7,r6,r5
        subu        r8,r2,r7
        beqz        r8,IsNoPrim
        addi        r3,r3,4
        j       Loop
        ; wyx: R3 是数组 Table 的下标，此处循环 Table(R3) 就是从 0 开始遍历整个数组
        ; 每次循环: loop {R5 = Table[R3], R6 = R2/R5, R7 = R6*R5, R8 = R2-R7}
        ; 可以看出 R8 是求 R2 % Table[R3] 的余数,
        ; 若 R8 == 0, 则 R2 被整除（可下结论 R2 不是质数）;
        ; 反之, 则 R2 没被整除;
        ; 程序如果能运行到 R1 == R3, 说明整个数组遍历了都不能整除 R2, 即 R2 是质数

IsPrim:     ;*** Write value into Table and increment index
        sw      Table(r1),r2
        addi        r1,r1,4

        ;*** 'Count' reached?
        lw      r9,Count
        srli        r10,r1,2
        sge     r11,r10,r9
        bnez        r11,Finish
        ; wyx: 找到质数就记录在 Table[R1] 中，找够 Count 个即停止

IsNoPrim:   ;*** Check next value
        addi        r2,r2,1     ;increment R2
        j       NextValue
        ; wyx: 没找到就 R2 += 1 继续找下一个

Finish:     ;*** end
           trap         0
```

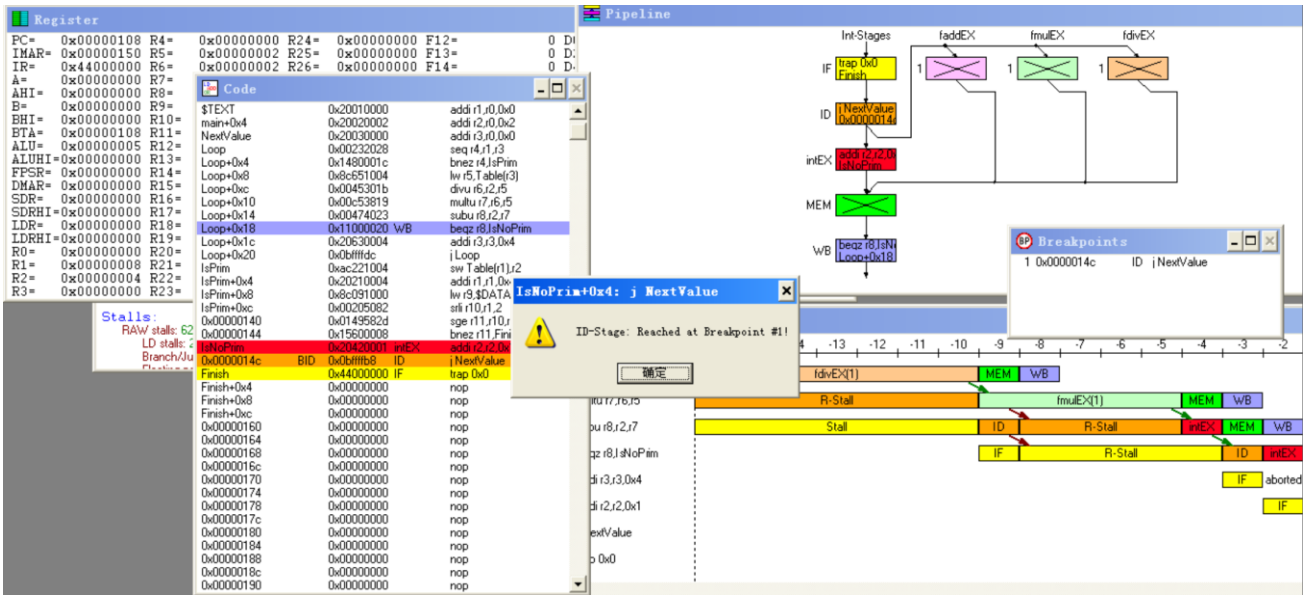
（2） 程序运行过程

从各 R 寄存器变化规律，可以验证之前阅读源码的结果；



单步执行过程中发现，每个循环里最耗时的操作是计算除法和乘法，周期非常长；





C. 对于 FACT.S 程序，请从中选出 3-5 条不同的指令，并分别指出它是哪种指令（R-type，I-type，还是 J-type），并参照教科书 2-28 页中的图 2.19（如下所示），填写指令格式中各个域的二进制值。为了清楚起见，最好用填表的形式。

答：指令各域应该填写数值的原因：数值机器码才是计算机能理解的信息；

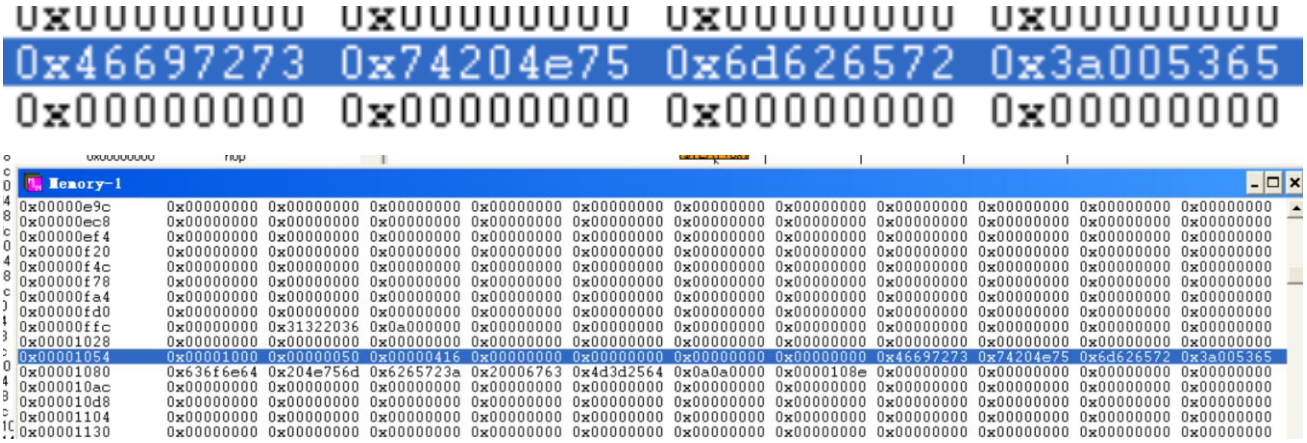
指令截图	比特位				结论
lw r2,SaveR2(r0) Adr.: input.Finish Code: 0x8c021060	1000 11 (opcode)	00 000 (Rs1)	0 0010 (Rd)	0001 0000 0110 0000 (imm)	应为 I-type 指令
addi r1,r0,0x1070 Adr.: main Code: 0x20011070	0010 00 (opcode)	00 000 (Rs1)	0 0001 (Rd)	0001 0000 0111 0000 (imm)	应为 I-type 指令
jr r31 Adr.: input.Finish+0x10 Code: 0x4be00000	0100 10 (opcode)	11 1110 0000 0000 0000 0000 0000 (value)			应为 J-type 指令

D. 请通过 CODE 窗口的各种选项查看数据，进而判断出 DLX 是 big-endian 还是 little-endian，为什么（结合实验中的观察分析）？是否是对齐的（aligned）？

答：用两个程序段来验证这个问题：

（1） 首先，以 GCM.S 程序为例，发现 Prompt1 和 Prompt2 变量在内存中的存储方式是

```
13
14          ;*** Prompts for input
15      Prompt1:    .asciiz     "First Number:"
16      Prompt2:    .asciiz     "Second Number: "
17
```



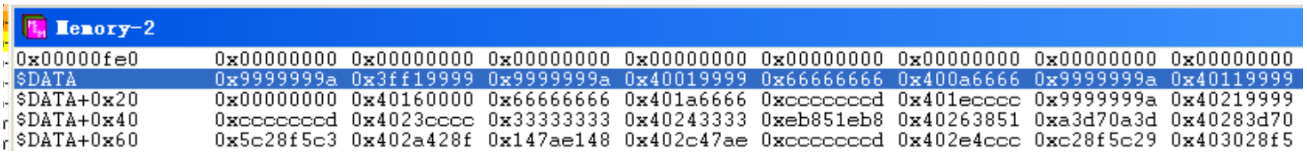
对比 ascii 码参考表：

对应字符	F	i	r	s	t	空格	N	u	……	:	NUL	S	e
十六进制表示	46	69	72	73	74	20	4e	75	……	3a	00	53	65

（2） 再编写一句数据段定义代码：

.double 1.1,2.2,3.3,4.4,5.5,6.6,7.7,8.8,9.9,10.10,11.11,12.12,13.13,14.14,15.15,16.16

在内存中表现为下图所示

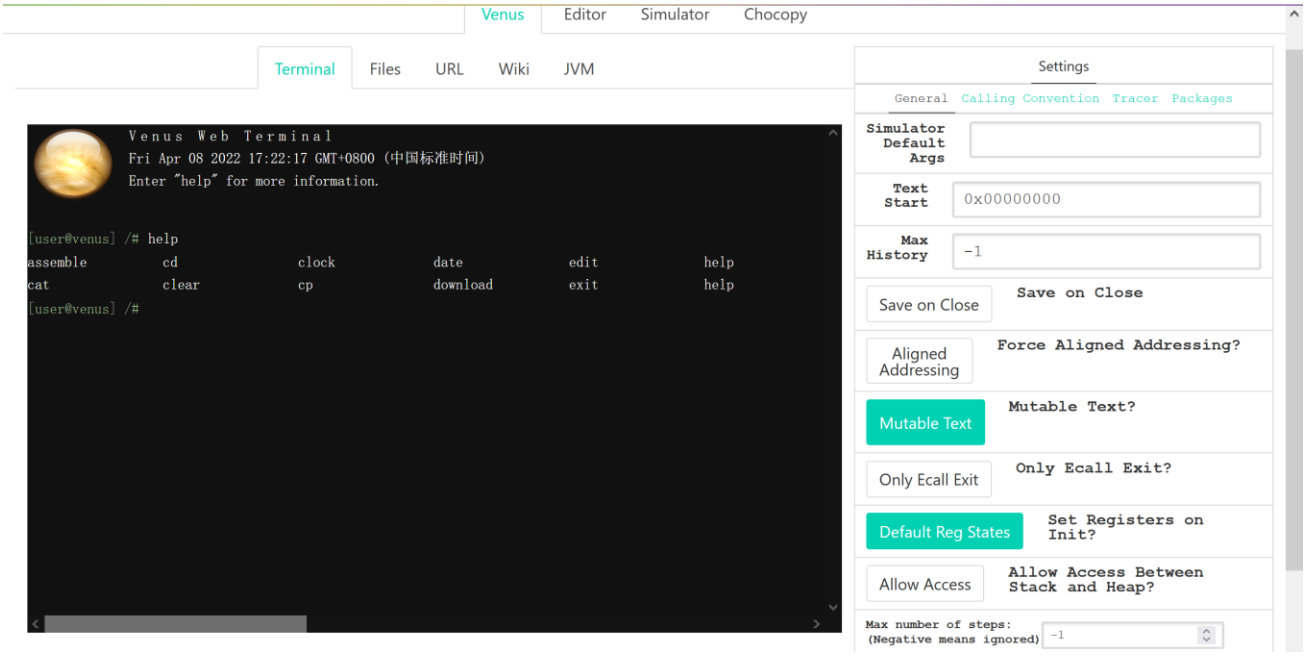


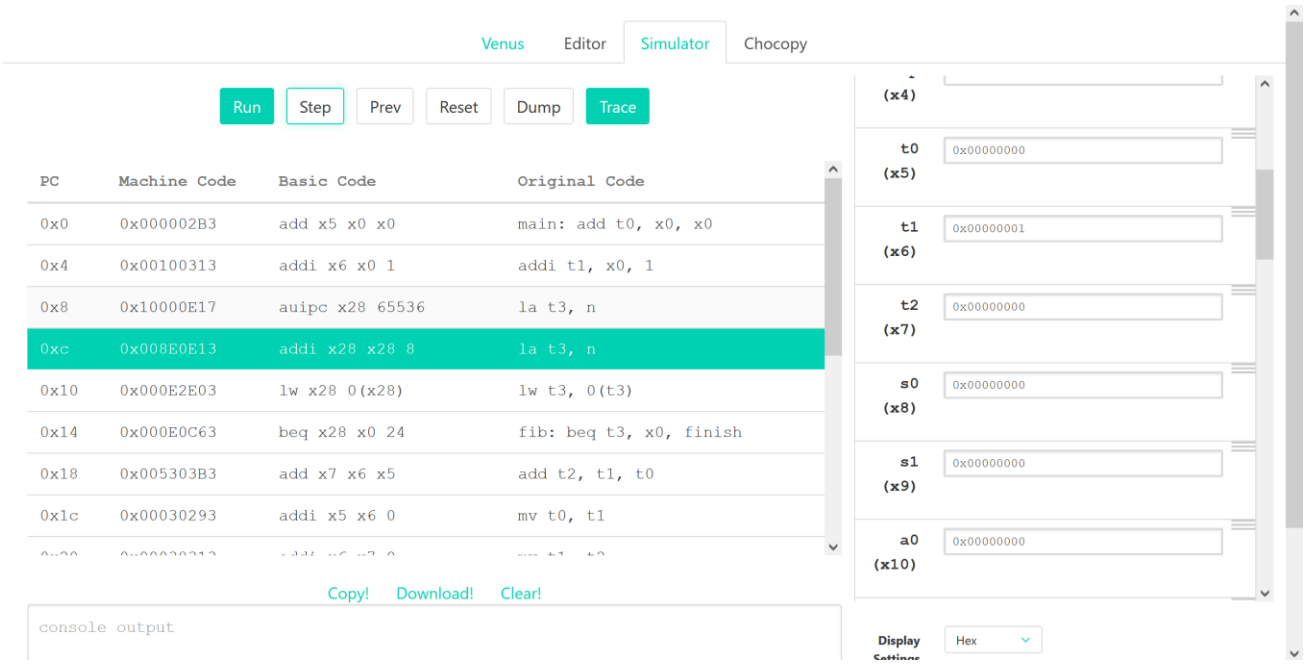
综合以上两例，发现 DLX 架构是大端序的、对齐的。

2. Venus 模拟器部分

了解 RISC-V 的在线模拟器 Venus 的使用方法。包括如下内容：

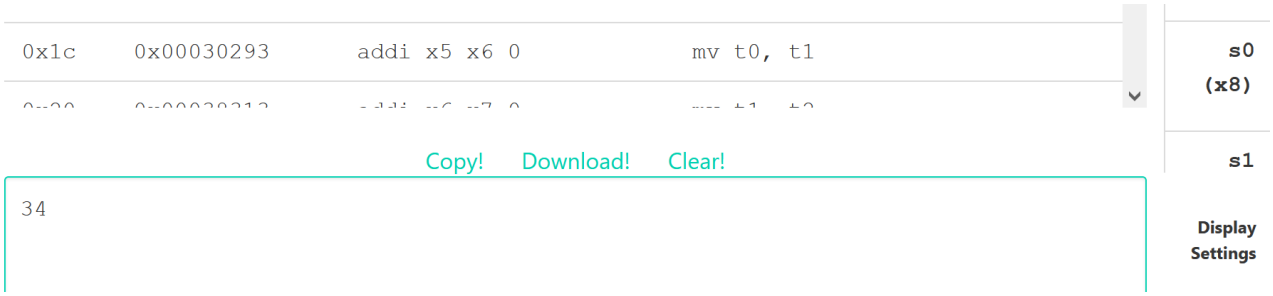
A. 参考提供的资料，学习 RISC-V 的在线模拟器 Venus 的用法。





B. 程序 ex1.s 的功能是什么？程序运行完后，得出的结果是什么？

答： 源程序得出的结果是斐波那契数列（下标从 0 开始的）第 9 个数，34



对代码段的分析注释如下：

```
.data
.word 2, 4, 6, 8
n: .word 9

.text
main:   add     t0, x0, x0
        addi    t1, x0, 1
        la      t3, n
        lw      t3, 0(t3)
fib:    beq     t3, x0, finish  # 循环下列过程，运行到下标 n 为止
        add     t2, t1, t0      # t2 = t1 + t0
        mv      t0, t1          # t0 = t1
        mv      t1, t2          # t1 = t2
        addi    t3, t3, -1      # t3--
        j       fib
finish: addi    a0, x0, 1       # a0 = 1, a1 = t0
        addi    a1, t0, 0
        ecall # print integer ecall
        addi    a0, x0, 10
        ecall # terminate ecall
```

再通过分别更改变量 n 的值为 10, 11,12，验证程序的功能为计算斐波那契数列，并在控制台输出数列下标为 n 的数字；

C. 对于 ex1.s 程序，请从中选出 3-5 条不同的指令，并对于其中每条指令，指出它是哪种指令。

指令	比特位							结论
addi x6 x0 1	0000 0000 0001(imm)	0000 0(Rs1)	000(add)	0011 0(Rd)	001 0011(opcode)			I-type
auipc x28 65536	0001 0000 0000 0000 0000(imm)			1110 0(Rd)	001 0111(opcode)			U-type
lw x28 0(x28)	0000 0000 0000(imm)		1110 0(Rs1)		010(lw)	1110 0(Rd)	000 0011(opcode)	I-type

四、 心得体会

本次实验我通过 WinDLX 和 Venus 模拟器，初步了解了 DLX 和 RISC-V 指令的格式。通过借助在模拟器上的实际操作过程，观察代码的单步执行结果，我们可以更深入的理解指令集体系结构，掌握识别指令类型的技能。