

学号：202000130143	姓名： 郑凯饶	班级： 计科 1 班
实验题目：用 WinDLX 模拟器完成求素数程序		
实验学时：2	实验日期：4.19	
<p>实验目的：</p> <p>通过本次实验，熟练掌握 WinDLX 模拟器的操作和使用，特别注意在单步执行 WinDLX 程序中，流水线中指令的节拍数。</p>		
<p>硬件环境：</p> <p>Dell Latitude 5411 Intel (R) Core (TM) i5-10400H CPU @ 2.60GHz (8GPUs), ~2.6GHz</p>		
<p>软件环境：</p> <p>VMware Workstation 16 Player Windows 7</p>		
<p>实验步骤与内容：</p> <p>1. 阅读汇编代码，程序利用朴素求法输出大小为 Count 的素数表：</p> <pre> .data ;*** size of table .global Count Count: .word 10 .global Table Table: .space Count*4 .text .global main main: ;*** Initialization addi r1,r0,0 ;Index in Table addi r2,r0,2 ;Current value ; r1 是维护素数表的长度 ;*** Determine, if R2 can be divided by a value in table NextValue: addi r3,r0,0 ;Helpindex in Table ; r3 为素数表的索引 Loop: seq r4,r1,r3 ;End of Table? ; r3 和 r1 比较，判断已经将素数表遍历完 bnez r4,IsPrim ;R2 is a prime number </pre>		

```

lw      r5,Table(R3)    ; 将素数读取至 r5
divu    r6,r2,r5
multu   r7,r6,r5
subu    r8,r2,r7        ; 通过(r2/r5)*r5?=r2 判断 r2 是否被当前素数整除
beqz    r8,IsNoPrim
addi    r3,r3,4 ; 下一个素数
j       Loop

```

IsPrim: ;*** Write value into Table and increment index

```

sw      Table(r1),r2
addi    r1,r1,4 ; 将 r2 加入素数表中

```

;*** 'Count' reached?

```

lw      r9,Count
srli    r10,r1,2        ; r1/4=素数的个数
sge     r11,r10,r9
bnez    r11,Finish      ; 判断是否达到 Count 上限

```

IsNoPrim: ;*** Check next value

```

addi    r2,r2,1        ;increment R2
j       NextValue

```

Finish: ;*** end

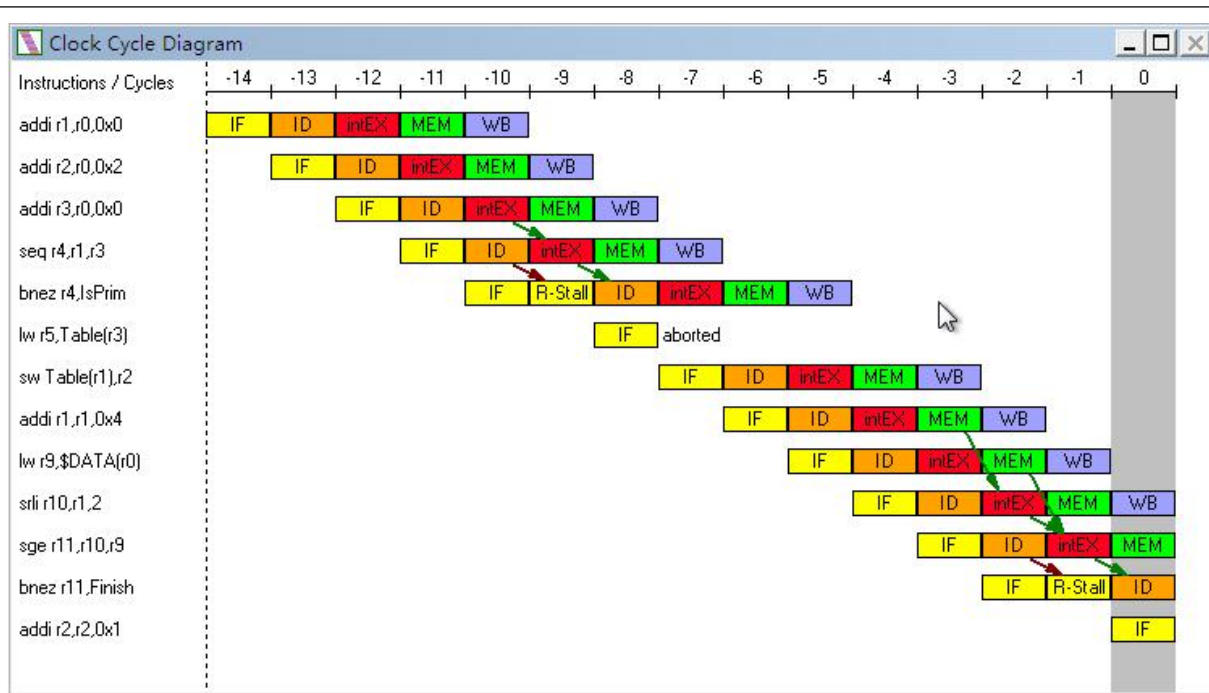
```

trap    0

```

2. 装载程序，F5 单步执行

求取第一个素数 2:



逐条指令分析：

bnez r4,IsPrim: 成功跳转，因为素数表为空，r1=0

在 WinDLX 模拟器的 5 级流水线中跳转指令在 ID 阶段确定是否成功跳转，因此下一条读入的指令会被 aborted。

Sw Table[r1],r2: 将 r2=2 写入素数表中

Addi r1,r1,0x4: 将 r1 加 4，因为 Int 类型占 4 个字节

Lw r9, \$DATA[r0]: 获取数据区的 0 号元素也就是 Count=10

Srli r10,r1,2: 将 r1 右移 2 位，相当于除以 4，获取当前素数表的长度

Sge r11,r10,r9: 判断 r10 是否不小于 r9

Bnez r11, Finish: 若是则已经找到 Count 个素数，执行 Finish 结束程序，没有成功跳转

求取素数 3:

Lw r5, Table[r3]: 将素数表中的第 r3 个素数读取至 r5

Divu r6,r2,r5

Mulu r7,r6,r5

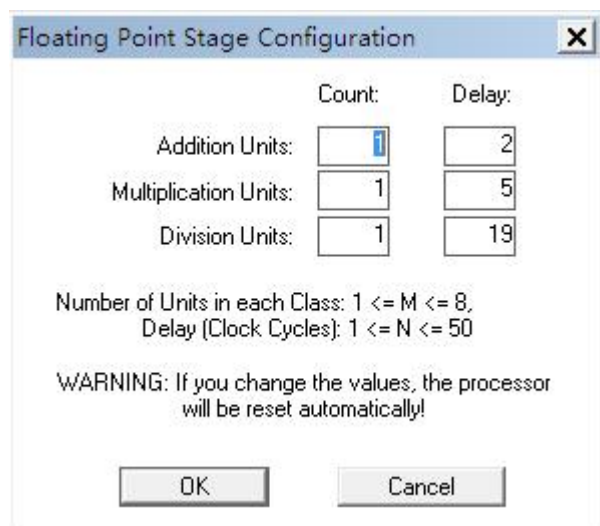
Subu r8,r2,r7

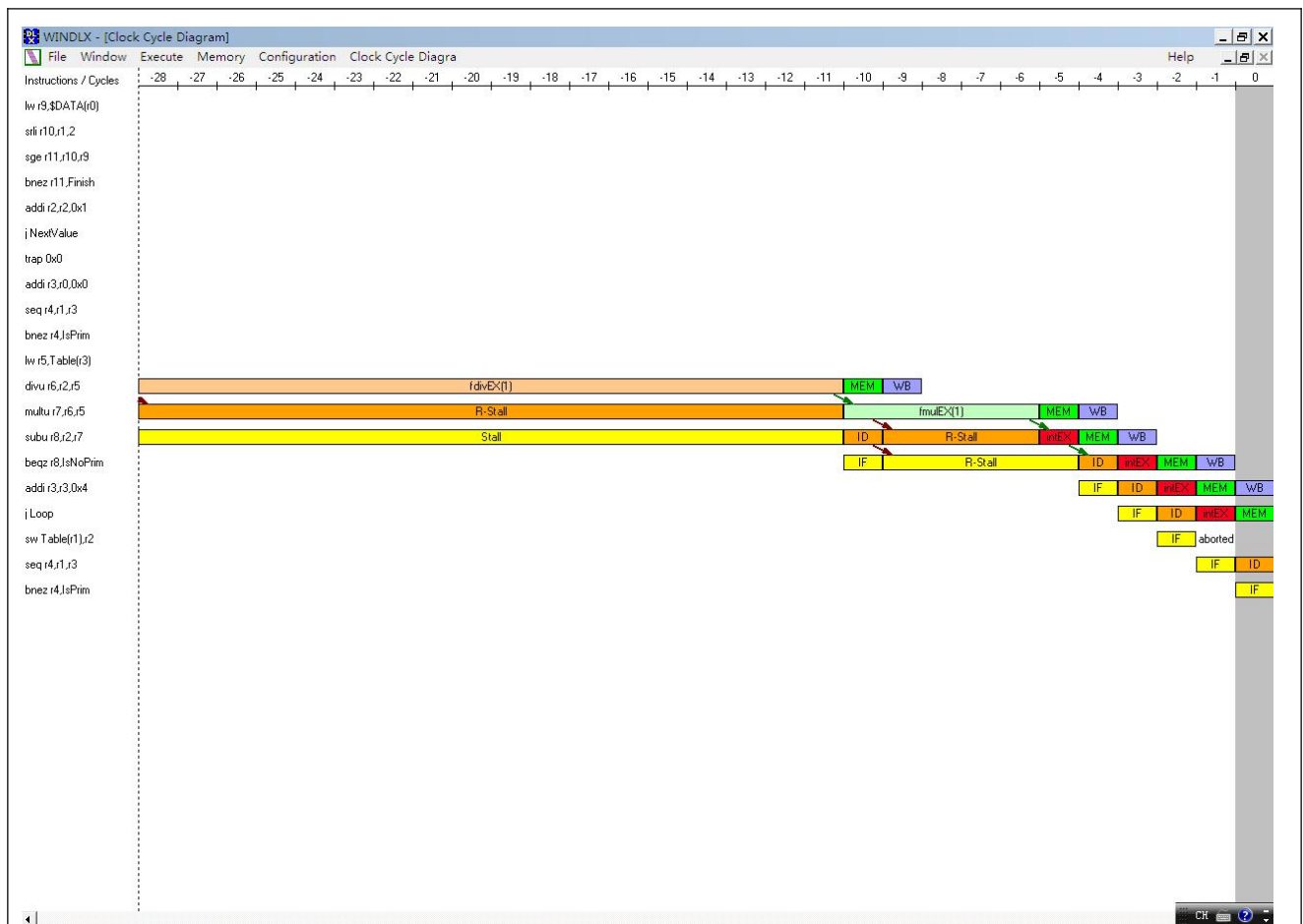
Beqz r8,IsNoPrim

以上代码通过判断 $(r2/r5)*r5=r2$ ，利用精度丢失确定当前素数是否整除当前数，此时 $r2=3, r5=2$ ，不整除因此不跳转。

可以看到乘法命令的执行周期为 19，除法的执行周期为 5，执行时间长，特别是在乘法执行的过程，流水线中几乎只有该指令在执行，复杂指令对流水线的工作表现有很重要的影响。

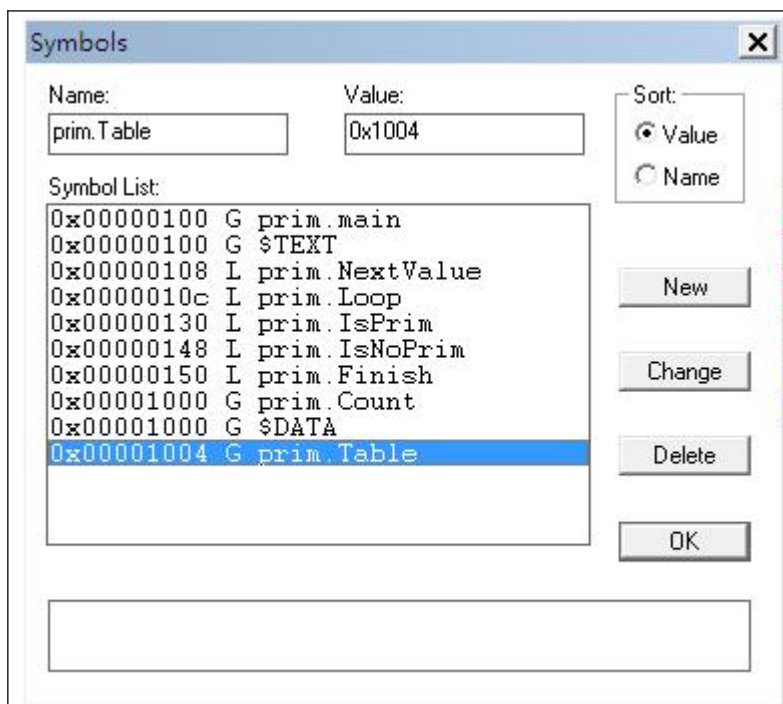
在 WinDLX 模拟器中，可以在 Floating Point Stage Configuration 设置乘除指令的时延。





之后执行 `nez r4,IsPrim` 成功跳转，将素数 3 加入素数表中，判断下一个数 4。

由于没有结果输出,想办法到内存中查看结果。查看 Symbols 找到 Table 的地址为 0x1004, 在内存中找到指定内存单元，可以看到素数表 2,3,5,7, b, d。



结论分析与体会：

这个实验主要观察复杂指令，如乘法，除法指令，这些指令极大地影响了流水线的并行执行，让我理解了为什么 RISC 指令集，需要通过精简指令，简化每个指令的执行功能。这样可以保证流水线的正常执行，保证各个部件的并发度。