

LC3

PB15111679 庄涛

平台

EDA 工具及版本号：ISE 14.7

开发板信号：Spartan6 XC6LX16-CS324

外设

Nexys3 上的 5 个按键（上下左右中），8 个开关，4 个 7 段数码管

技术点

之前所有的实验的技术点都有用到，其中 ip 核的功能是自己实现的，因为需要一些额外的信号。以下列举本次实验在之前几个实验中没有涉及到的技术点：

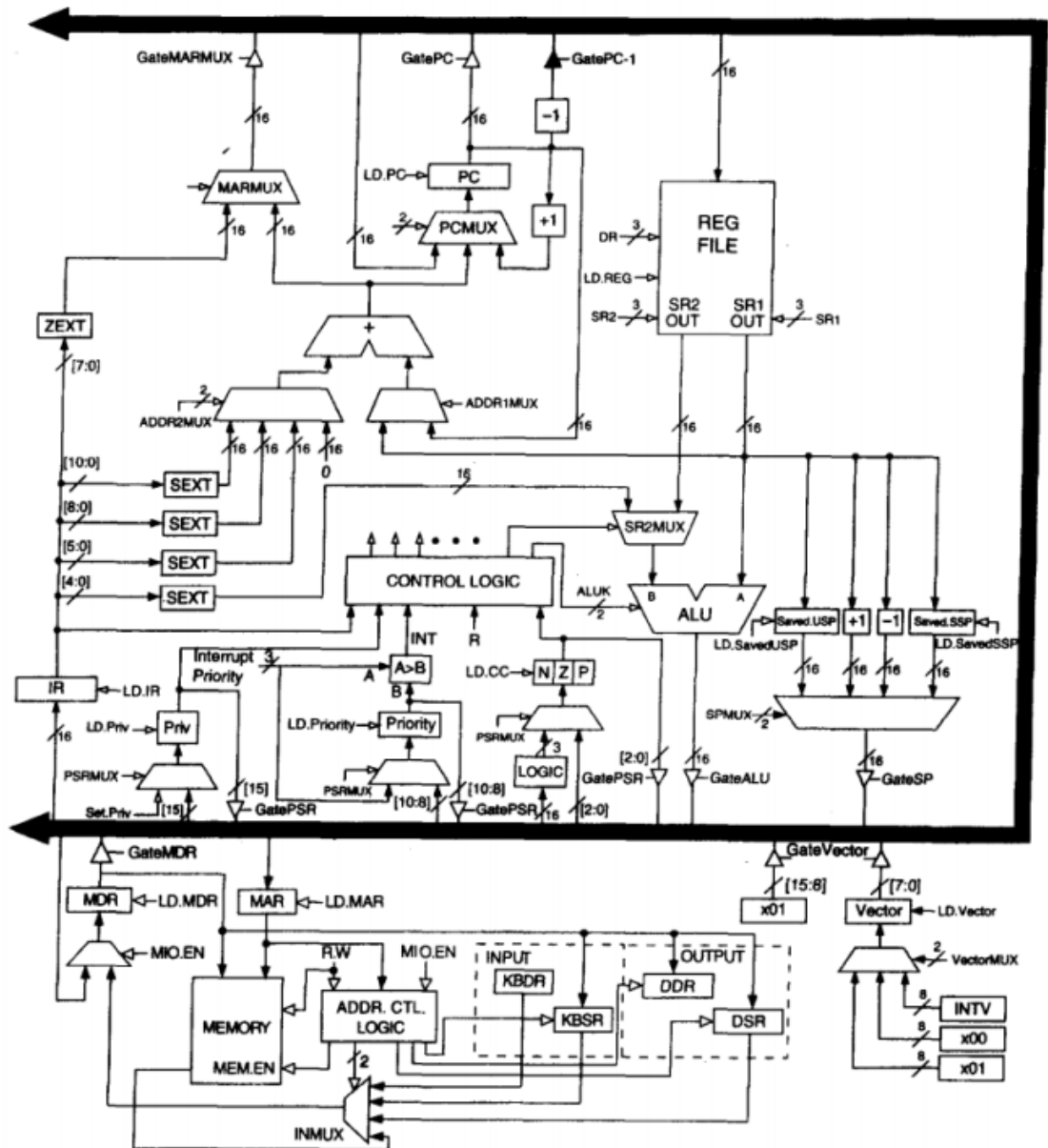
- 冯诺依曼模型
- LC3 结构
- 汇编指令
- 内存映射
- 中断机制
- 栈
- 微指令结构的状态机

原理

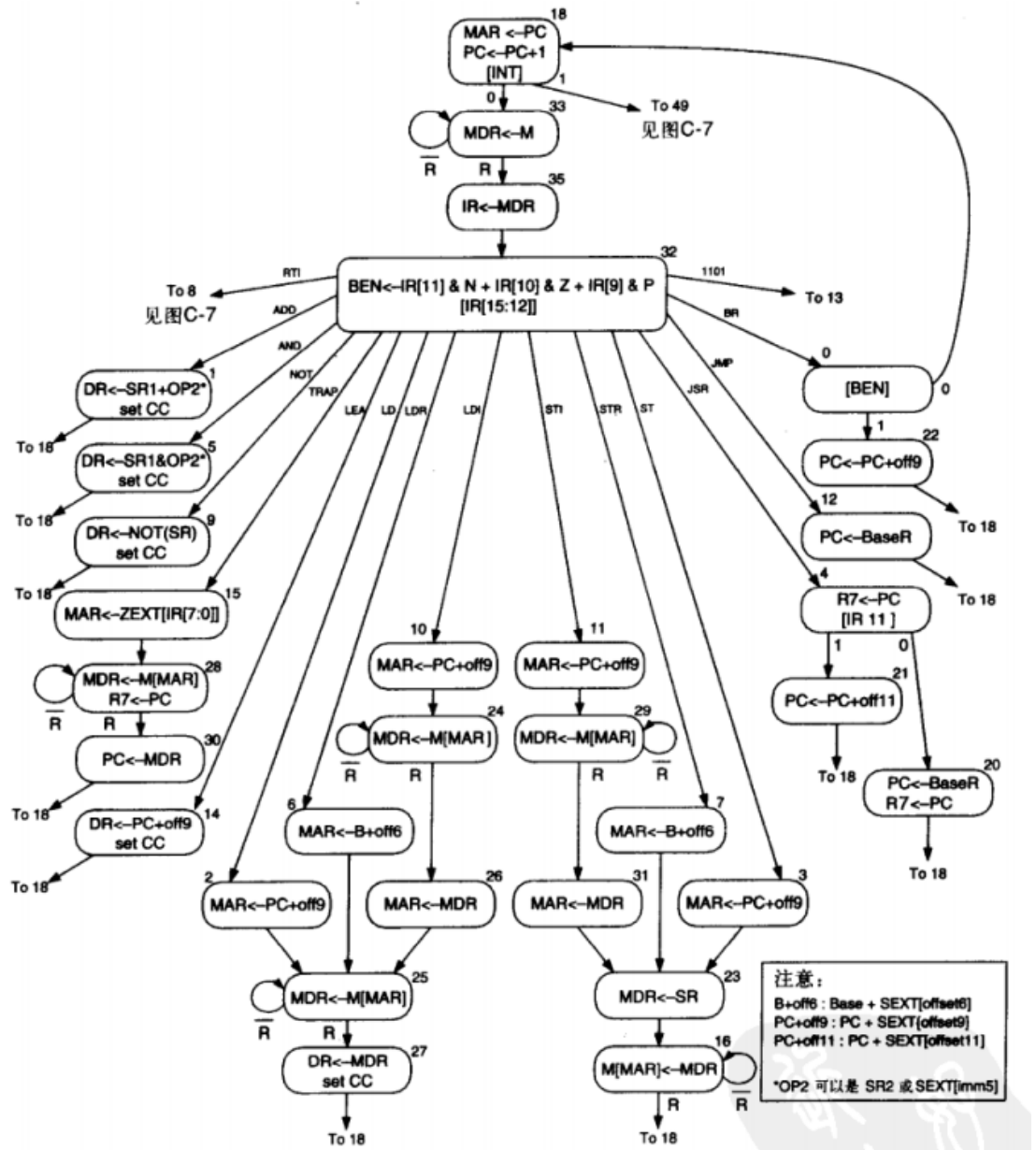
0. 基本原理

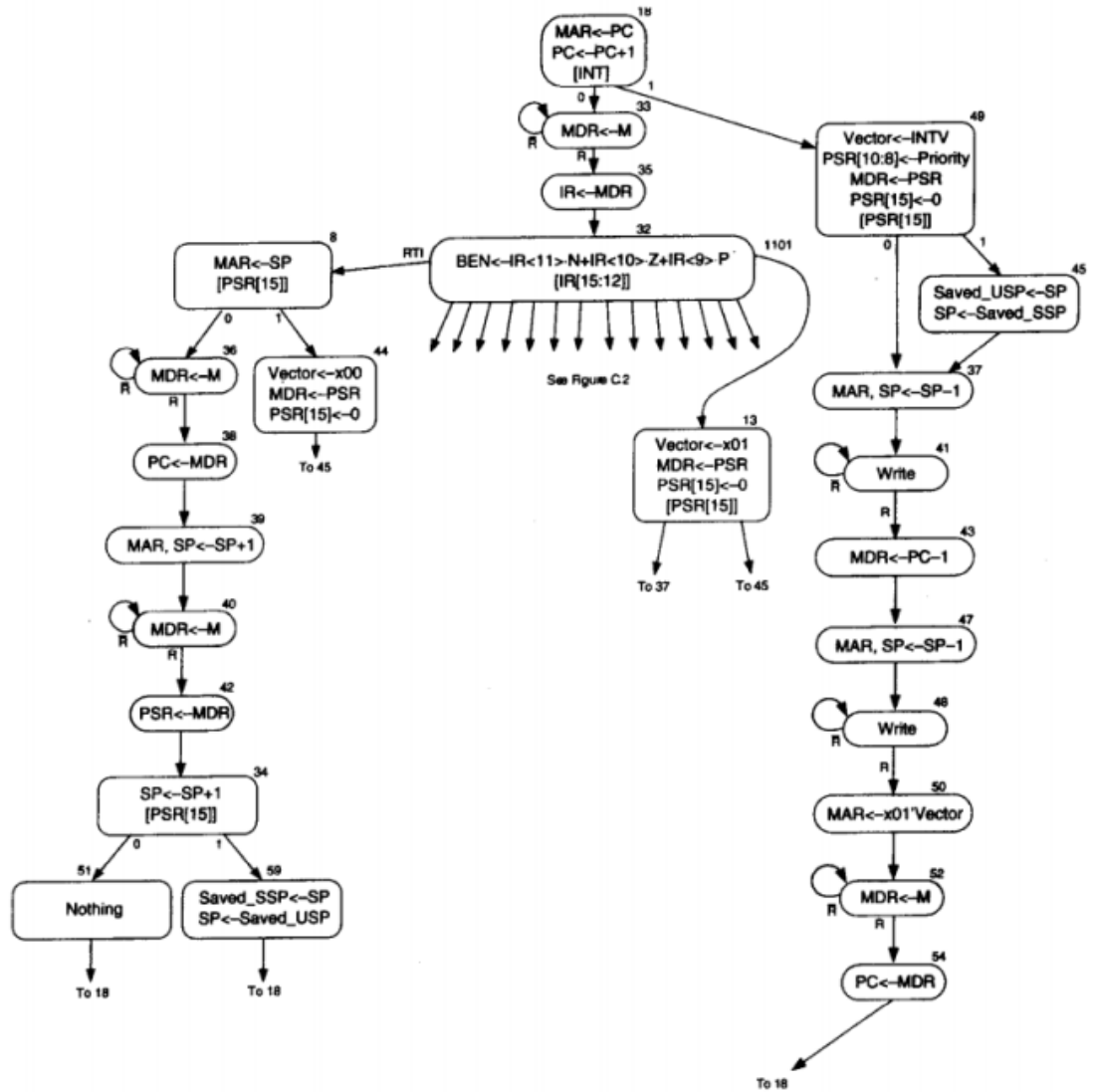
由于内容过多（内存映射，指令集结构，汇编指令集，中断机制，异常处理，状态机结构，数据通路），因此这里不再赘述，想了解基本原理的话请参考《Introduction to computing systems》的附录。

1. 数据通路



2. 状态图









































3. 键盘

键盘实现采用的是 nexys3 上的 5 个按键，使用了整流和优先编码方式，保证一次只有一个按键写入。KBDR = { 11'b0, up, down, left, right, middle } 映射到 x7E00，KBSR 映射到 x7E02。另外由于读写内存需要给 CPU 提供反馈信号 R(ready)，当需要读写键盘时，需要提供一个键盘的相应的 R 信号。

4. 显示器

显示器的实现采用的是 nexy3 上的 7 段数码管，下面是 26 个字母和 0-9 数字在 7 段数码管上的解决方案。

0	1	2	3	4	5	6	7	8	9
									
a/A	b/B	c/C	d/D	e/E	f/F	g/G	h/H	i/I	j/J
									
k/K	l/L	m/M	n/N	o/O	p/P	q/Q	r/R	s/S	t/T
									
u/U	v/V	w/W	x/X	y/Y	z/Z				
									

由于 7 段数码管的表达能力有限，因此有些字母并不太像其原貌。如 “k, m, s, t, v, w, x, z”，但以尽量做到相似或含义明显。比如 “M”，在 n 上加一横杠说明是两个 n，相应的还有 “w”。“S” 与 “5” 相似，因此为了对称，把 “s” 的中间去掉了，相应的还有 “Z”；“x” 难以表达，因此只显示了一个斜杠来代表。

DDR = { position, 6'b0, ASCII } 映射到 x7E04。DSR 映射到 x7E06。ASCII 大于或等于 128 的那些字符，与 7 段数码管的 128 种显示情况相对应，以方便任意显示。

同键盘一样，显示器也要提供相应的反馈信号 R。

5. 开关

开关实现采用的是 nexys3 上的 8 个开关。{ 8'b, sw } 映射到 x7E08，只可读不可写，一般用于字符或 8 位二进制数字输入，搭配按键一起使用。

6. 内存

由于 nexys3 上的存储资源有限，因此只用了大小为 $x8000 \times 16 \text{ bit}$ 的内存。

由地址控制逻辑来控制内存与外设内存映射的读写

7. 有限状态机的控制存储器

这里的实现比较繁琐。主要是《Introduction to computing systems》并没有提供微指令的具体内容，只能自己根据数据通路和状态图来自己一个一个地填写，然后在仿真中不断找出错误并修改，花费时间极大。下面给出微指令的详细内容。

[illegible]

11.系统函数的改写

由于 nexys3 的特殊性，需要专门改写适用于 nexys3 上 LC3 的系统函数

以下是相应内容

陷入矢量	功能	详细描述
X20	读键盘	循环检测 KBSR[15]，当它为 1 时，读取 KBDR 到 R0
X21	读开关	循环检测 KBSR[15]，当它为 1 时，读取 KBDR，若 KBDR==1（按了中键），那么读取 SWDR 到 R0；否则重新开始。
X22	显示字符	将 R0 中的 ASCII 对应的字符显示出来，位置由 R1 决定
X23	输出四个字符	将 mem[R0], mem[R0+1], mem[R0+2], mem[R0+3]依次显示出来
X24	提示输入开关	首先在屏幕上提示 “in=” 并将输入的开关作为 ASCII，将相应的字符回显出来
X25	停机	在屏幕上显示 “HALT”，程序终止运行，按下中键后，程序继续运行
X26	输出字符串	在屏幕上显示首地址为 R0 的字符串，到 16'h0000 为止。其中，每显示 4 个字符，就等待用户按下中键，然后再显示接下来的字符串。
X27	等待	程序将等待 R0ms

具体的汇编代码实现请查看 program/sys.asm

12.PSR

图 C-8 中把 PSR 分成了三个部分：Priv, Priority, CC，实现时我把他们放在了一个 16 位寄存器里。特别要注意对 PSR 的更新，容易出错。

13.勘误

- 微指令总共有 50 个控制信号，而书中说 49 个。
- 图 C-8 中 CONTROL LOGIC 并没有输出到 SP2MUX 的控制信号，实际上这个控制信号是 IR[5]
- 图 C-8 中控制信号 SR2 实际上是 IR[2:0]
- 系统函数中不建议使用固定位置来存储寄存器的值以空出位置给临时变量。因为当发生中断时，中断时调用的程序有可能去调用这些系统函数，这样就会导致该固定位置的内容被覆盖，使得中断返还时内容丢失，因此应该使用栈来存储那些寄存器的值。

电路功能

电路功能简单的说就是一个支持了一套汇编指令因而图灵完备的计算机。

以下是四个演示程序的功能描述，相应的代码在文件夹 program 中，相应的视频在文件夹 vedio 中，相应的烧写文件在文件夹 bit 中

1. 猜数字 (guess)

猜一个事先确定好的一个 0-255 的数字

把 8 个开关作为输入的二进制数字，设定好后按中键。

若输入小了，就显示 MORE，并接受下一个输入

若输入大了，就显示 LESS，并接受下一个输入

若输入正确，就显示 DONE

2. 排序 (sort)

依次输入四个 0-9 的数字，把 8 个开关作为输入（注意输入范围是 0-9），设定好后按中键。输入四个数字后按中键，数码管上将显示从大到小的数字序列

3. 交互 (hi)

输入四位字符（最好由数字和字母组成）作为你的名字，然后不断按中键可查看输出的字符串，直到结束。输出的字符串为 “Hi, xxxx. My name is LC3.” 其中 xxxx 为输入的字符。

4. 动态波形图像+中断机制 (wave)

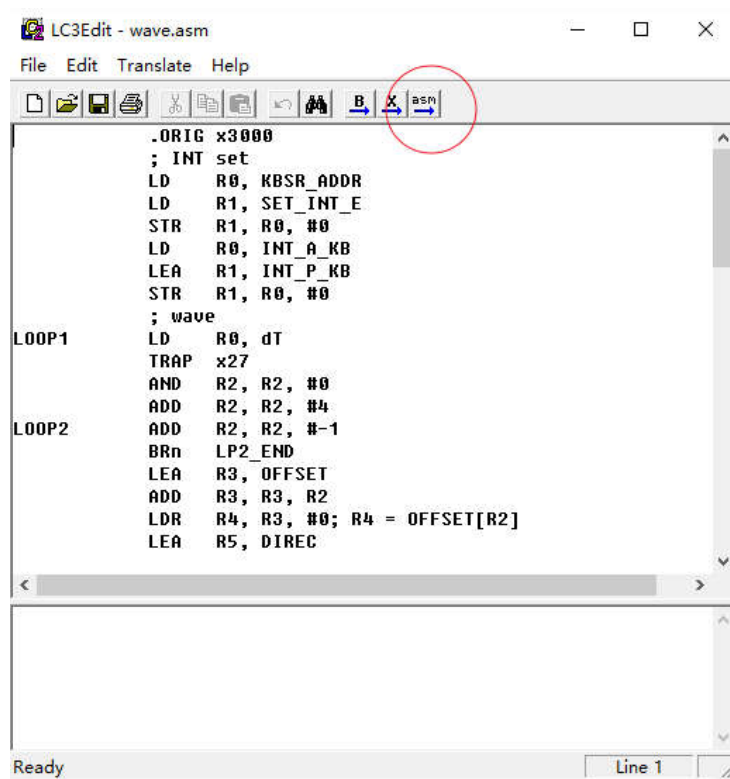
显示器将显示一个动态的波形图像，当按下中键后发生中断，中断时的执行大致就是显示 HALT 并暂停，当中键再次按下时，程序返回之前的程序即继续显示动态波形图像。这里要注意的是实现方式是中断而非轮询。

使用方法

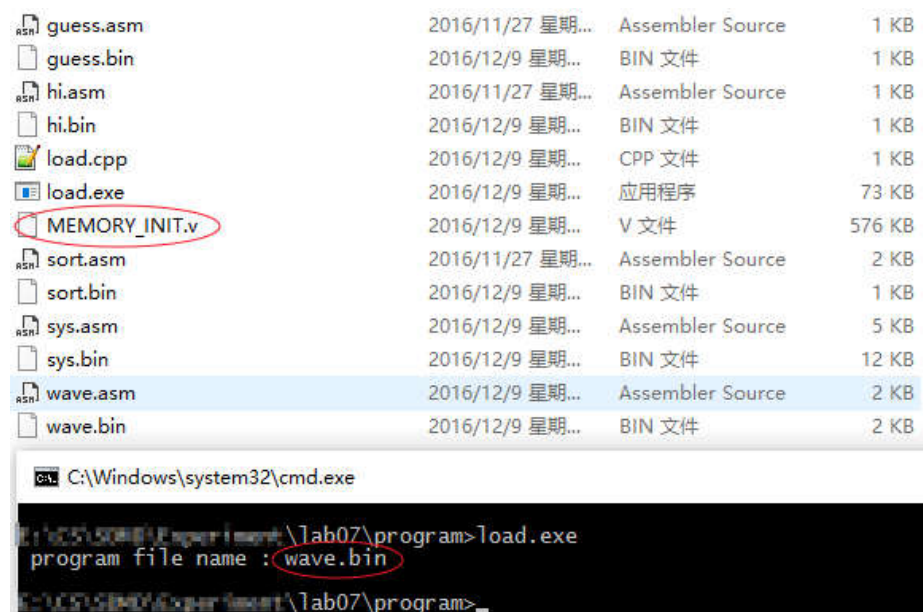
1. 编写一个汇编程序

```
1      .ORIG x3000
2      ; INT set
3      LD    R0, KBSR_ADDR
4      LD    R1, SET_INT_E
5      STR    R1, R0, #0
6      LD    R0, INT_A_KB
7      LEA    R1, INT_P_KB
8      STR    R1, R0, #0
9      ; wave
10     LOOP1  LD    R0, dT
11            TRAP  x27
12            AND   R2, R2, #0
13            ADD   R2, R2, #4
14     LOOP2  ADD   R2, R2, #-1
15            BRn   LP2_END
16            LEA    R3, OFFSET
17            ADD   R3, R3, R2
18            LDR    R4, R3, #0; R4 = OFFSET[R2]
19            LEA    R5, DIREC
20            ADD   R5, R5, R2
21            LDR    R6, R5, #0; R6 = DIREC[R2]
22            ADD   R4, R4, R6; R4 = OFFSET[R2] + DIREC[R2]
23            BRz   DIRCed
24            BRn   ToUP
25     ToDOWN ADD   R6, R6, #-2
26            STR    R6, R5, #0; DIREC[R2] = #-1
27            BR     DIRCed
28     ToUP   ADD   R6, R6, #2
29            STR    R6, R5, #0; DIREC[R2] = #1
30     DIRCed STR    R4, R3, #0; OFFSET[R2] = OFFSET[R2] + DIREC[R2]
31            LEA    R3, MID
32            ADD   R3, R3, R4
33            LDR    R0, R3, #0; R0 = MID+OFFSET[R2]
34            ADD   R1, R2, #0
```

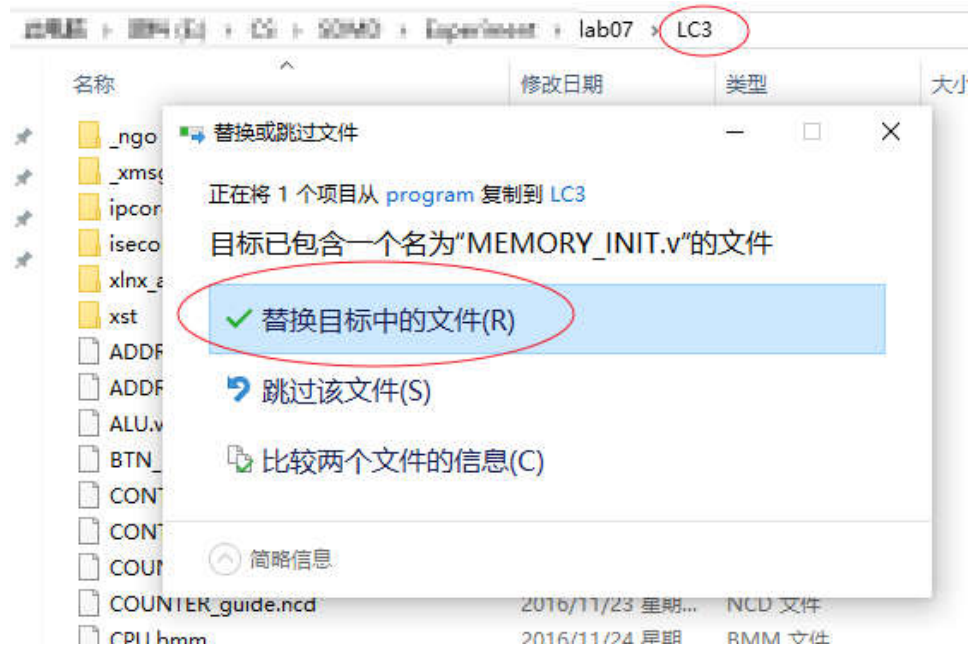
2. 使用 program/LC3/LC3Edit.exe 将汇编程序转换为 LC3 机器码 xx.bin



3. 运行 load.exe 输入 xx.bin, 生成相应的内存初始化文件 MEMORY_INIT.v



4. 将生成的 MEMORY_INIT.v 复制到文件夹 LC3 中，覆盖源文件



5. 在 ISE14.7 中编译得到 lc3.bit
6. 将 lc3.bit 烧写入 nexys3

总结

优点

图灵完备

缺点

没用使用标准外设（显示器，键盘、音响和鼠标）。

有大量的 warning，然后发现有时会出现随机烧写错误，重新烧写问题就没了，估计问题出在 warning，但只是烧写时随机出现，而在运行中不出现，而且时间有限，因此没有去修复这个不太关键的 bug。

附录

所有相关的文件可以在 home.ustc.edu.cn/~ustczt/LC3 中获取