Homework 3

PB22111711 陈昕琪

T1

64位操作数需要6bits，56个寄存器也需要6bits，而32bits的指令由OPCODE（操作数），SR（源寄存器），DR（目标寄存器）组成，因此IMM可表示的位数为32-6-6-6=14bits，又因为IMM是二进制补数，因此可以表示的数值范围是

$$-2^{13}\ 2^{13}-1$$

T2

T3

x3000 0001 000 001 0 00 010
ADD,将R1+R2的内容储存到R0中
x3001 0000 001 000000111
BR,判断最近装入寄存器的内容是否>=0，如果成立，则offset9=7，因此只需R1+R2的值>=0即可

T4

1. 对ADD（0001）和AND（0101）操作是有好处的，寄存器位数变小可以使IMM位数变大，可操作空间更大。
   对于NOT（1001）则无影响，因为NOT指令除了操作数，目的寄存器，源寄存器其余位全为1，所以无影响。
2. 对LD（0010）和ST（0011）操作是有好处的，寄存器位数变小可以使偏移量更大，有更大的操作空间。
3. 对BR（0000）无影响，BR指令本身不带寄存器

T5

观察电路图得知，只有当WE=1且A[1:0]经过逻辑门后显示为1时，才能输入内存，因此只需考虑WE=1的情况即可。
根据电路图得知，最终锁存的数据为

A[1:0]	Cycle No.	$D_{in}[2:0]$
00	6	101
01	4	011
10	3	010
11	7	100

对于第八个cycle结束时，

$$D_{in}[2:0] = 101$$

，并分析此时的电路图得出的结果如下：

T6

MDR用来保存要被写入地址单元或者从地址单元读入的数据。MAR用来保存数据被传输到的位置的地址或者数据来源位置的地址。

注意到从 Access1 到 Access3， x4000 的值和 x4001 的值都变化了，并且 x4001 末态的值和 x4000 不同，证明 Access1-3 应该读入一次写入两次（实际上可以假定存在 Access0 是对于 x4000 的数据的读入？），那么 Operation No.2 应为 R ；观察 x4001 的数据来源，符合 _0__0 特点的只有 x4003 。观察从 Access3 到 Access5， x4003 的值也变化，且 01101 和 Operation No.2 读入的数据不同，
那么 Operation No.4-5 应该是先 R 后 W ；考虑从哪里读入 01101，只能是 x4002 ，那么第一张表格补全如下：

Operation No.	R/W	MAR	MDR
1	W	x4000	11110
2	R	x4003	10110
3	W	x4001	10110
4	R	x4002	01101
5	W	x4003	01101

第二张表补全如下：

Address	Before Access 1	After Access 3	After Access 5
x4000	01101	11110	11110
x4001	11010	10110	10110
x4002	01101	01101	01101
x4003	10110	10110	01101
x4004	11110	11110	11110

T7

T8

Address	Instruction
x3000	1001 111 001 111111
x3001	1001 100 010 111111
x3002	0101 101 111 000 010
x3003	0101 100 110 000 001
x3004	1001 001 101 111111
x3005	1001 010 100 111111
x3006	0101 000 001 000 010
x3007	1001 011 000 111111

T9

T10

使用 BR 指令时，如果上一个操作过的寄存器的值符合指令中的 n z p 的条件，则对现有的 PC 进行 offset 跳转，否则继续执行下一个指令。这要求我们必须预知或者规定 BR 前面的一条语句，以及 PC 此
时的地址。 JMP 指令则不需要知道这些，使用直接寻址跳转，非常方便。