

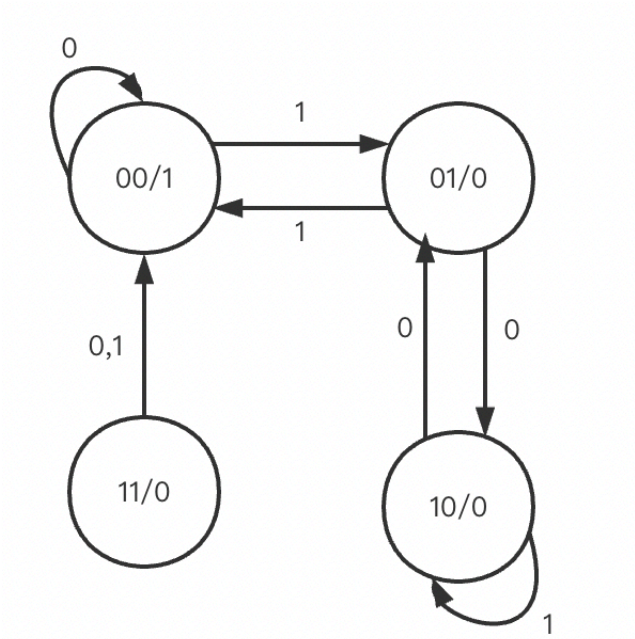
# HW3 solution

## T1

(a).

$S_1$	$S_0$	$X$	$Z$	$S'_1$	$S'_2$
0	0	0	1	0	0
0	0	1	1	0	1
0	1	0	0	1	0
0	1	1	0	0	0
1	0	0	0	0	1
1	0	1	0	1	0
1	1	0	0	0	0
1	1	1	0	0	0

(b).



## T2

R1 + R2 is zero or positive.

## T3

(a) opcode

(b) operand

T4

FETCH: 1 + 100 + 1 = 102

DECODE: 1 (memory access is not required)

EVALUATE ADDRESS: 0 (this phase is not required)

FETCH OPERANDS: 1 (memory access is not required)

EXECUTE: 1 (memory access is not required)

STORE RESULT: 0 (this phase is not required)

T5

opcode 有56条，则需要6bit才能表示完整。

register有64个，也需要6bit才能表示完整。

则IMM可用位数为：32 - 6 - 6 - 6= 14bit

则表示范围： $-2^{13} \leq IMM \leq 2^{13} - 1$

也可表示为：-8192≤IMM≤8191

T6

(a)

ADD(0001)和AND(0101)有了更大的立即数范围。

NOT(1001)没有获益。

(b)

load和store能有更多的一位去寻址。

(c)

branch中没有寄存器，因此没有变化。

T7

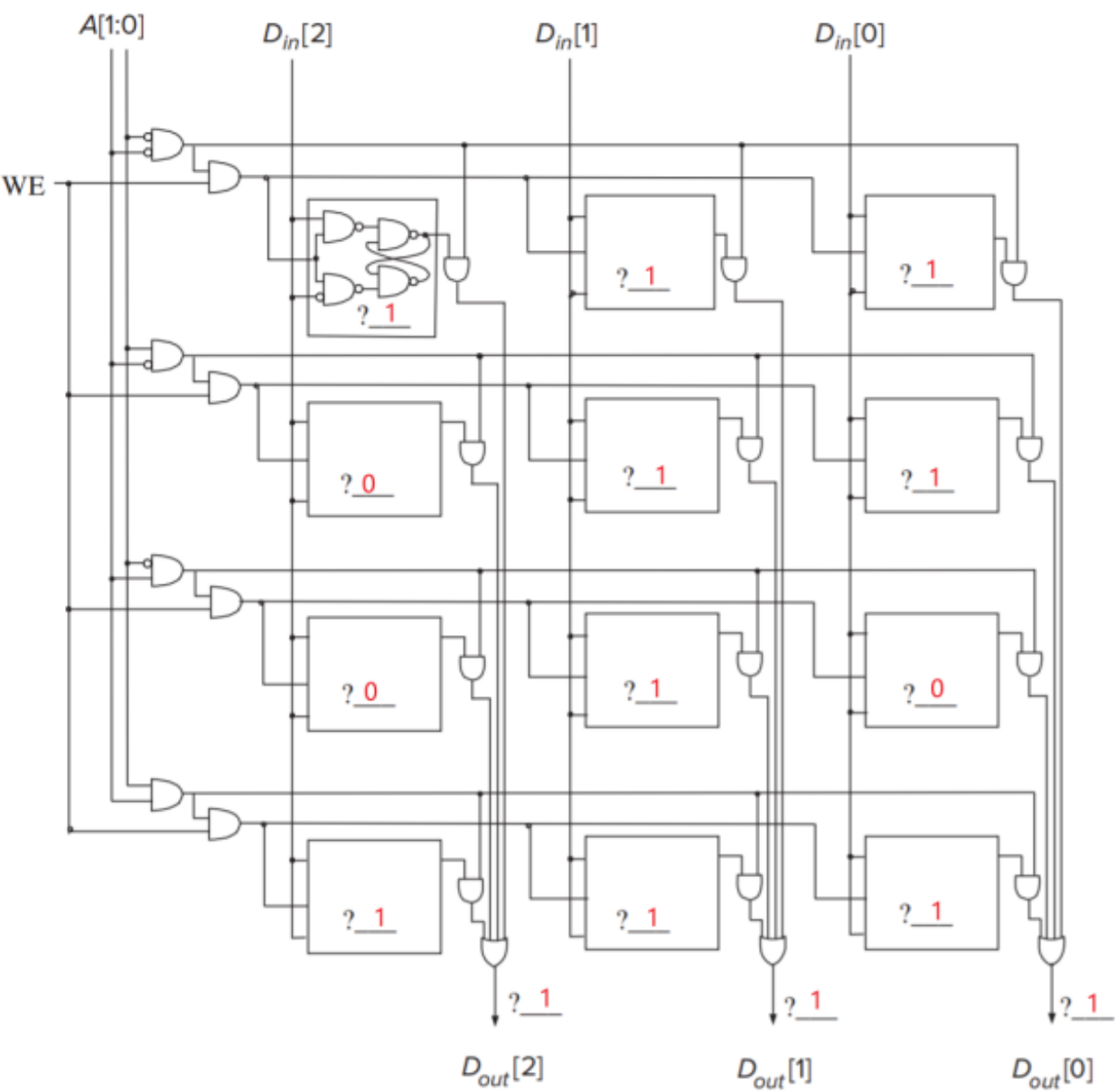
ADD的opcode:0001

STR的opcode:0111

JMP的opcode:1100

	fetch instruction	decode	evaluate address	fetch data	execute	store result
PC	0001, 0111, 1100				1100	
IR	0001, 0111, 1100					
MAR	0001, 0111, 1100					0111
MDR	0001, 0111, 1100					0111

**T8**



**T5**

- (a)  
MAR:001  
MDR:00011001
- (b)  
00010101

**T10**

为了叙述方便, 下面会简称 Operation i 为 i, 表 1 指代 Operations on Memory, 表 2, 3, 4 指代 Memory before Access 1, Memory after Access 3, Memory after Access 5. 同时建议大家在看的同时按照我的推导填表.

因为

The data in the MDR must come from a previous read (load).

而 Operation 3 的 MDR 与 1 有不同, 说明 Operation 2 是 Read. 而 Memory 在 Access 3 后与 Access 1 前 x4000 和 x4001 都发生了变化, 说明 Operation 1 和 3 写了这两处地址, 假设 1 写了 x4001, 与第二位是 0 矛盾, 因此 1 写了 x4000, 3 写了 x4001, 同时该步的 MDR 是 10xx0, 对比表 2, 知道应该来自 x4003. 到这里填出了表 1 前 3 行和表 2, 3 除了 x4002 外的行.

之后发现 x4003 在 Access 5 由 10110 变成了 01101, 发生了 Write, 说明 4 是 R, 5 是 W, 5 的 MDR 是 01101, 考虑到表 3 在上一步填完后没有 01101, 说明 x4002 处值为 01101, 得解, 最终答案如下:

## Operations on Memory

	R/W	MAR	MDR				
Operation 1	W	x4000	1	1	1	1	0
Operation 2	R	x4003	1	0	1	1	0
Operation 3	W	x4001	1	0	1	1	0
Operation 4	R	x4002	0	1	1	0	1
Operation 5	W	x4003	0	1	1	0	1

Memory before Access 1

x4000	0	1	1	0	1
x4001	1	1	0	1	0
x4002	0	1	1	0	1
x4003	1	0	1	1	0
x4004	1	1	1	1	0

Memory after Access 3

x4000	1	1	1	1	0
x4001	1	0	1	1	0
x4002	0	1	1	0	1
x4003	1	0	1	1	0
x4004	1	1	1	1	0

Memory after Access 5

x4000	1	1	1	1	0
x4001	1	0	1	1	0
x4002	0	1	1	0	1
x4003	0	1	1	0	1
x4004	1	1	1	1	0

## T11

- OPCODE有225条, 需要8位才能表示完整
- 寄存器有120个, 需要7位才能表示完整
- $32 - 7 * 3 - 8 = 3$ , 最多有3位UNUSED

## T12

- $5 \times 10^8$
- $5 \times 10^8 / 8 = 6.25 \times 10^7$
- 由于流水线, 每条指令执行时间相当于只需要1个cycle, 所以每秒可以执行约  $5 \times 10^8$  条指令 (流水线启动和退出的耗时导致不能完全达到最大值, 但是可以忽略)

## T13

Fetch: 将PC放入MAR, PC++, 将M[MAR]放入MDR, 将MDR的内容放入IR

Decode: 分析、检查指令类型

Evaluate Address: 地址计算 (比如地址值等于R3内容和立即数6之和)

Fetch Data: 读取指令处理所需要的源操作数 (来自寄存器或者内存)

Execute: 执行操作

Store Result: 将执行结果写入目的寄存器