第一次习题课

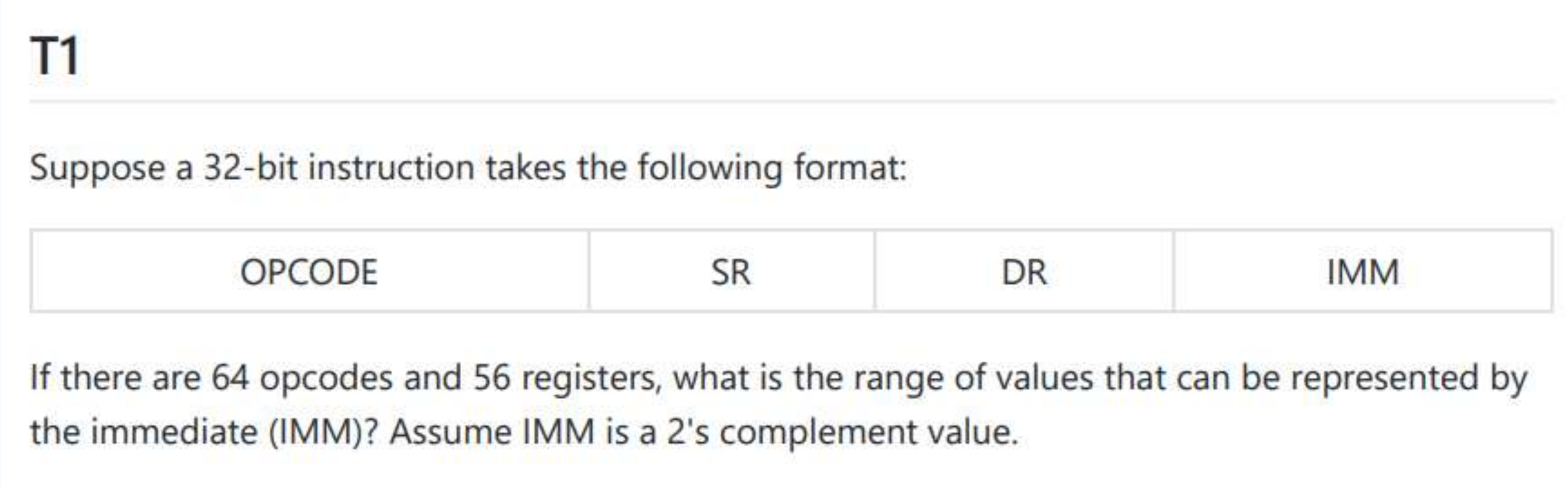
Hw3 & Midterm

中国科学技术大学
University of Science and Technology of China

Hw3

作业讲解

T1



T1

Suppose a 32-bit instruction takes the following format:

OPCODE	SR	DR	IMM

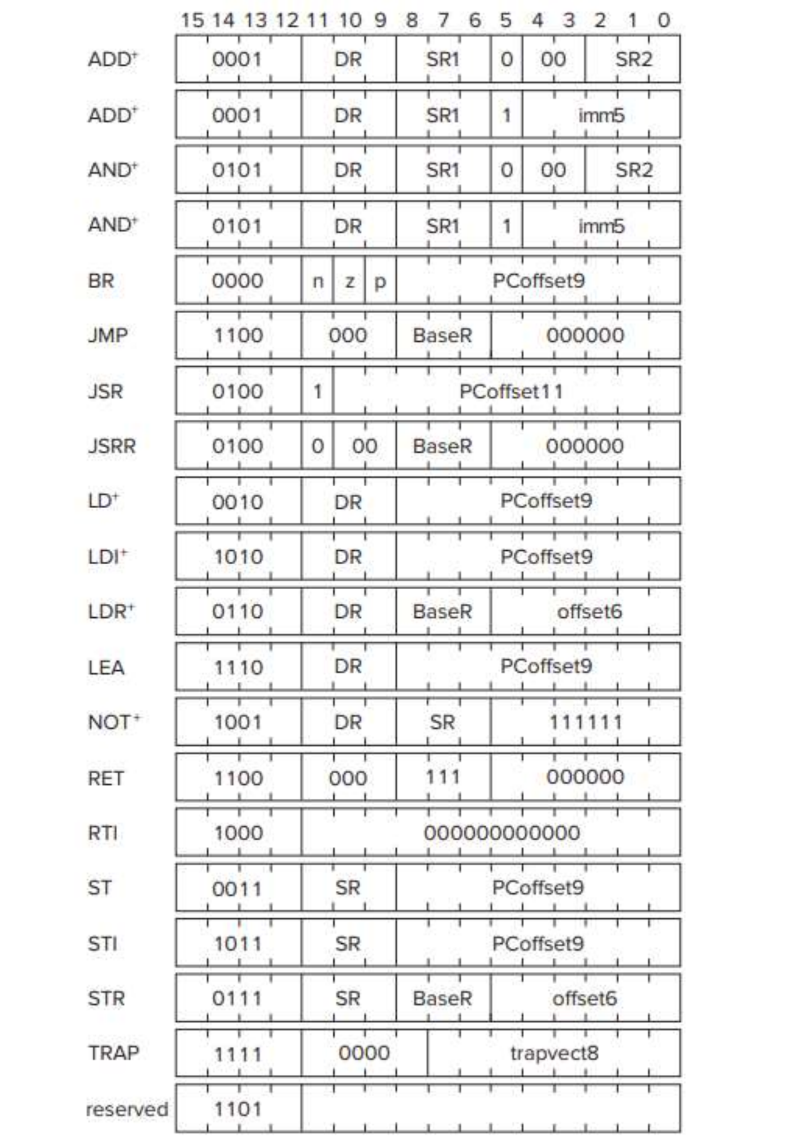
If there are 64 opcodes and 56 registers, what is the range of values that can be represented by the immediate (IMM)? Assume IMM is a 2's complement value.

- IMM 位宽14不难计算
- 但应该注意题设IMM为二进制补码，因此应该包含负数部分，答案为【2^13 ， 2^13 -1】

T2

- 遇到机器码，肯定需要根据LC3-ISA翻译成汇编指令如下：

addr	asm	effect
x3000	AND R0,R0,#0	R0 = 0
x3001	ADD R0,R0,#2	R0 += 2
x3002	BRzp PC+X	PC = x3003 + X
x3003	ADD R0,R0,#3	R0 += 3
x3004	ADD R0,R0,#1	R0 += 1
x3005	HALT	



15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

ADD+	0001	DR	SR1	0	00	SR2
ADD+	0001	DR	SR1	1	imm5	
AND+	0101	DR	SR1	0	00	SR2
AND+	0101	DR	SR1	1	imm5	
BR	0000	n	z	p	PCoffset9	
JMP	1100	000	BaseR	000000		
JSR	0100	1	PCoffset11			
JSRR	0100	0	00	BaseR	000000	
LD+	0010	DR	PCoffset9			
LDI+	1010	DR	PCoffset9			
LDR+	0110	DR	BaseR	offset6		
LEA	1110	DR	PCoffset9			
NOT+	1001	DR	SR	111111		
RET	1100	000	111	000000		
RTI	1000	000000000000				
ST	0011	SR	PCoffset9			
STI	1011	SR	PCoffset9			
STR	0111	SR	BaseR	offset6		
TRAP	1111	0000	trapvect8			
reserved	1101					

Figure A.2 Format of the entire LC-3 instruction set. *Note:* + indicates instructions that modify condition codes

T2

- 根据这段代码可以轻松得到前四个的答案，但是注意第五个会循环加二直到溢出变成x8000(-2^15)<0，循环终止，再加上后面的3和1得到最终结果（很多同学都没有考虑到负溢出，认为一直在循环）

addr	asm	effect
x3000	AND R0,R0,#0	R0 = 0
x3001	ADD R0,R0,#2	R0 += 2
x3002	BRzp PC+X	PC = x3003 + X
x3003	ADD R0,R0,#3	R0 += 3
x3004	ADD R0,R0,#1	R0 += 1
x3005	HALT	

X	Dec	Does the program halt?	Value stored in R0
000000010	2	Yes	2
000000001	1	Yes	3
000000000	0	Yes	6
111111111	-1	No	--
111111110	-2	Yes	x8004 (-2^{15}+4)

T2 – 其他问题

- 部分同学不认识一些英文符号：
  - Dash ：破折号 “—”
  - Slash ：斜杠，有两种，正斜杠(forward slash) “/” 和 反斜杠(back slash) “\”
- 部分同学不理解HALT是什么意思
  - 一般来讲程序中止都需要halt，否则会程序会不停地运行下去

T3

- 翻译两个指令

addr	inst
x3000	ADD R0,R1,R2
x3001	BRzp PC+7

- x3001跳转后将会执行x3002 + 7 = x3009
- 因此需要跳转条件成立：R1+R0 >= 0
- 再次强调BR跳转时的PC已经增量过了（具体原因可以参考LC3各指令状态图）

T4

- 8 regs -> 4 regs
- reg bitlength : 3 -> 2
- IMM bitlength ++

1. ADD(0001) 和 AND(0101) 有了更大的立即数范围。但 NOT(1001) 没有获益。

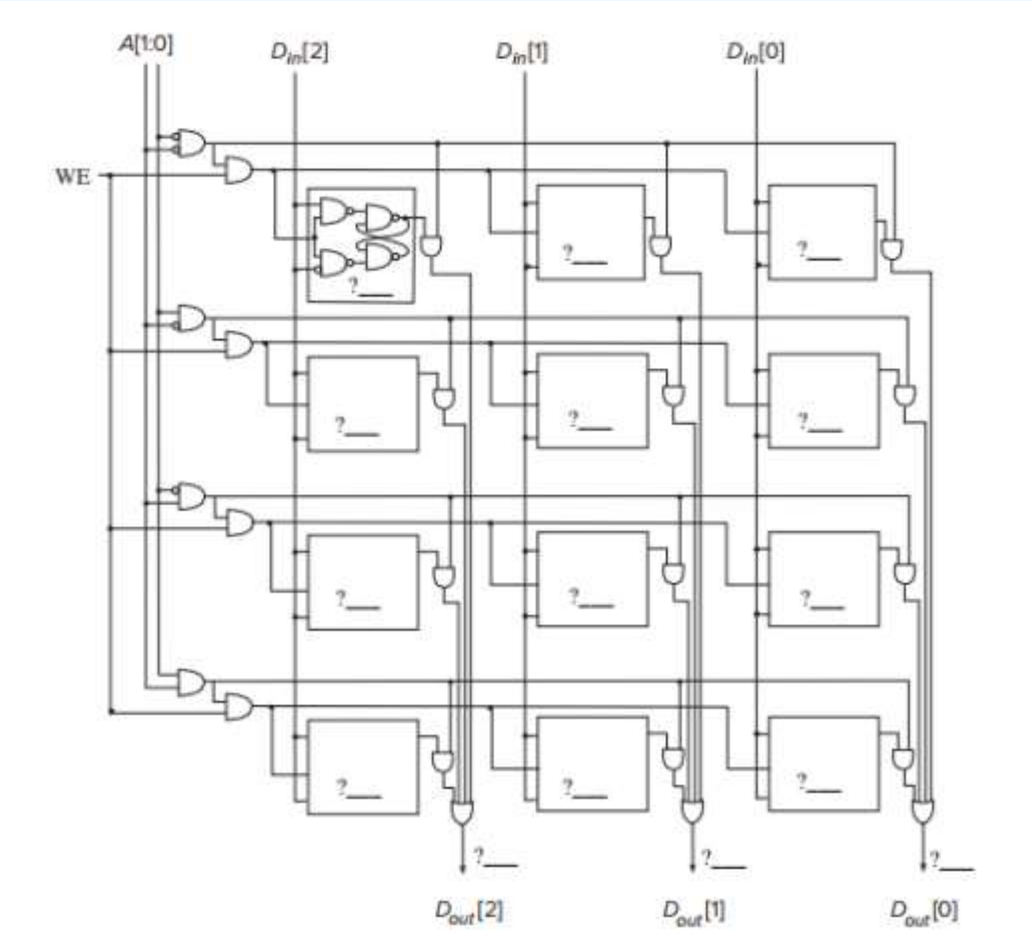
2. LD(0010) 和 ST(0011) 能有更多的一位去寻址。

3. BR(0000) 中没有寄存器，因此没有获益。

T5

- 部分同学没能注意到：只有 WE 为 1 时才写入

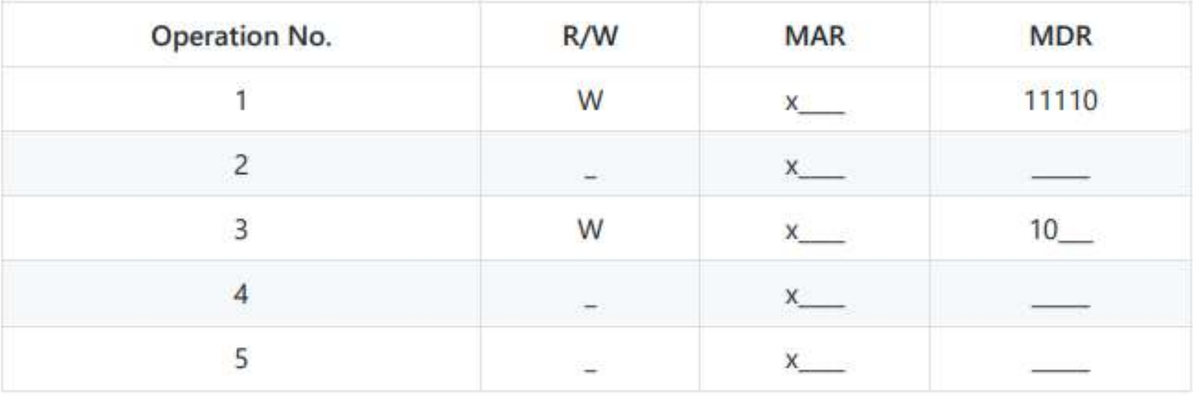
addr	Data
00	1->101
01	1->101->011
10	1->010
11	1->110



- D_out只需要看最后一次输入的地址，是11，所以是110

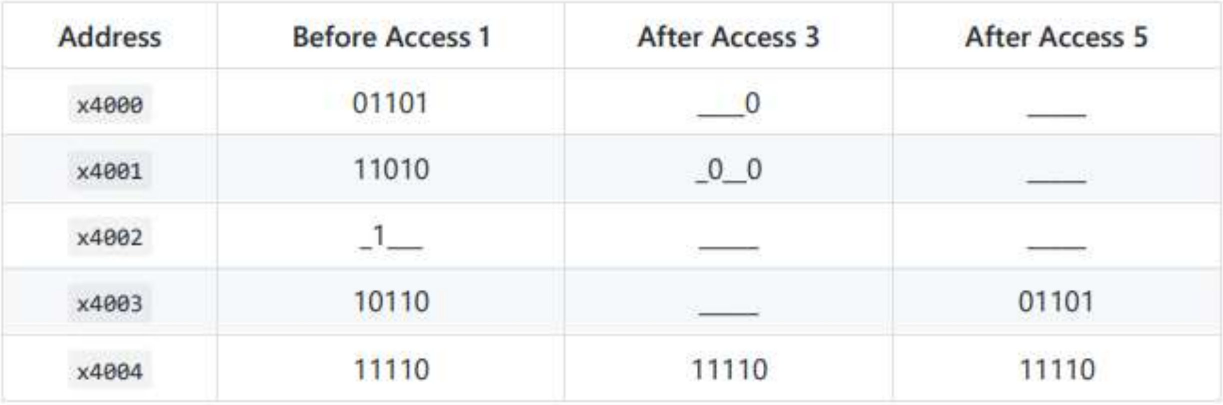
T6

- 这题需要进行严密的推理：
  - 根据题意限制，MDR的值只可能来自于前一轮的读结果，access3中MDR的值相较于 access1发生了改变，因此access2是读操作．access1之前只有x4003是10开头的，access1写 入的11110也不是10开头的，因此access2是读x4003，MDR为10110
  - x4000和x4001在前三个access中都发生变化，所以access1和access3一定是写入他们，具体来讲就是在access1中写入x4000，在access3中写入x4001
  - x4003在access3到access5发生了变化，被写入了01101，因此access4一定是读取了01101，并在aceess5写入x4003
  - 只有x4002可能为01101，因此x4002从始至终都是01101



Operation No.	R/W	MAR	MDR
1	W	x___	11110
2	_	x___	____
3	W	x___	10__
4	_	x___	____
5	_	x___	____

Operations on Memory

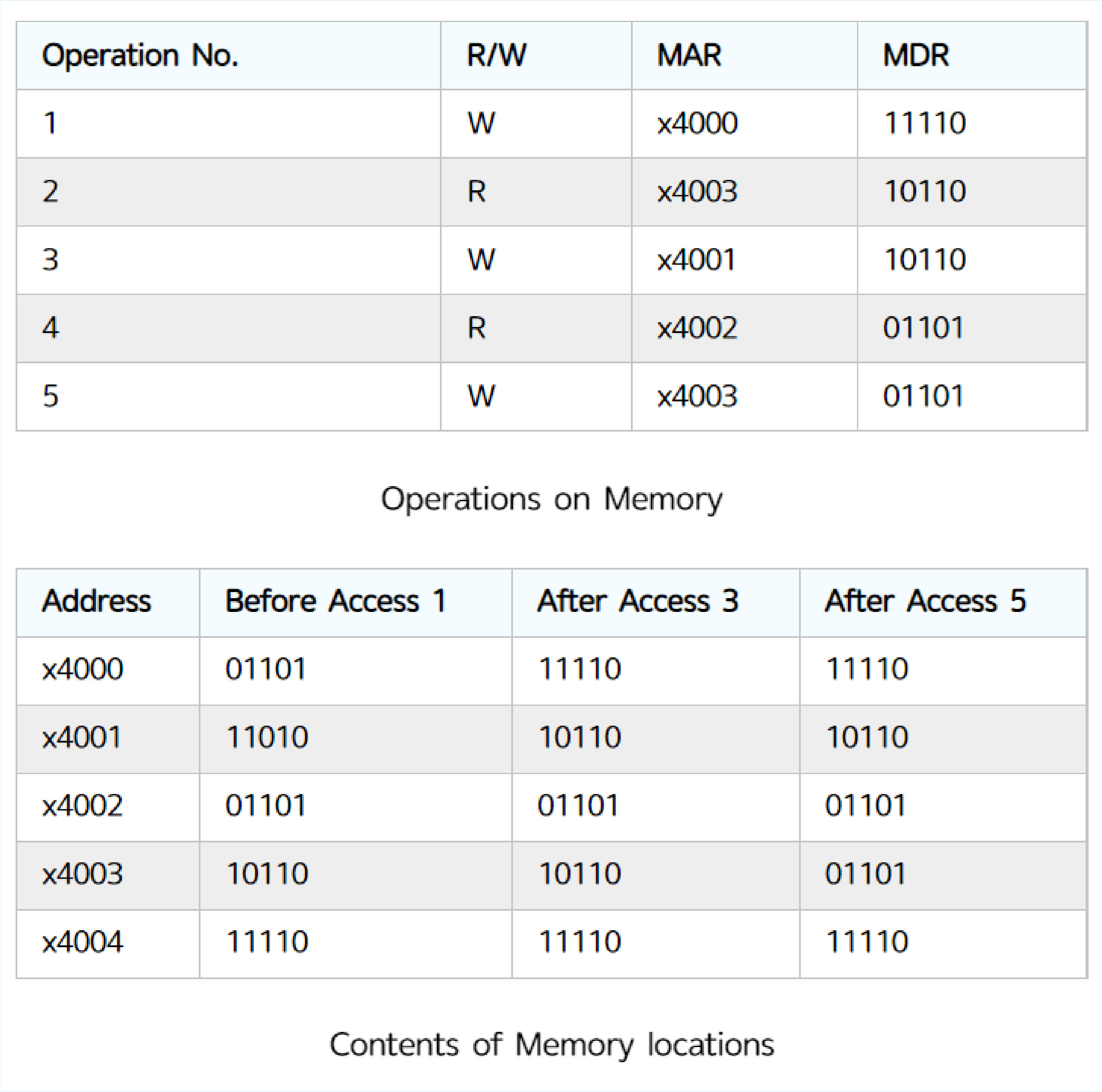


Address	Before Access 1	After Access 3	After Access 5
x4000	01101	___0	____
x4001	11010	_0_0	____
x4002	_1__	____	____
x4003	10110	____	01101
x4004	11110	11110	11110

Contents of Memory locations

T6

- 将推理结果进行填表：



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

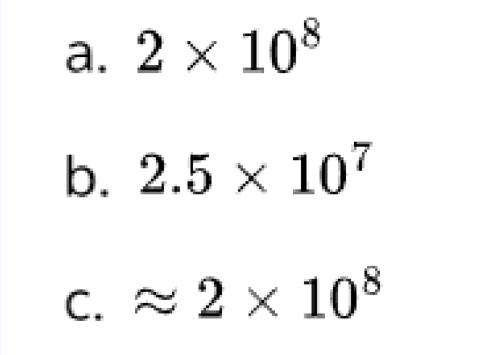
Operations on Memory

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

Contents of Memory locations

T7

- 概念分析题，认真读题计算，比较简单。



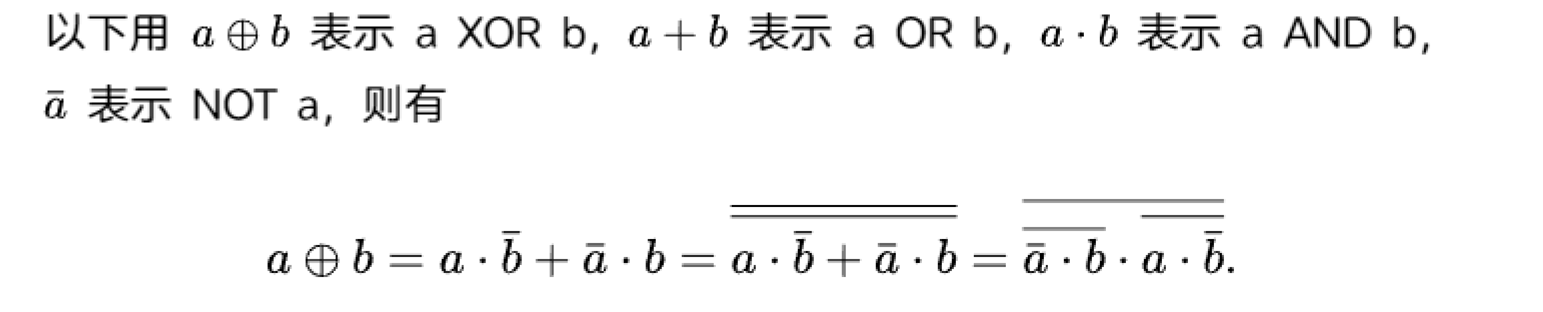
a. 2×10^8

b. 2.5×10^7

c. $\approx 2 \times 10^8$

- 第三题涉及到流水线cpu，属于超纲，考试不会考。但还是简单介绍一下。第四章介绍了指令执行时需要的 6 个 phase，把指令分成一个个阶段的其中一个目的就是为了能够实现流水线 CPU，可以简单理解为流水线cpu在每个phase都处理着不同的指令。因此本题目原本一条指令需要8个cycle，但是经过流水化后平均值需要一个cycle

T8



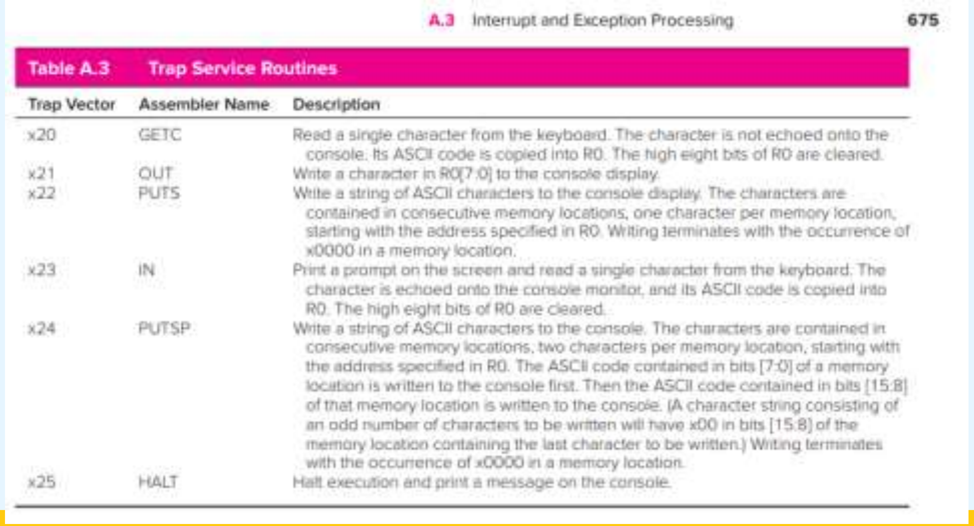
以下用 $a\oplus b$ 表示 a XOR b，$a+b$ 表示 a OR b，$a\cdot b$ 表示 a AND b，$\bar{a}$ 表示 NOT a，则有

$$a\oplus b=a\cdot\bar{b}+\bar{a}\cdot b=\overline{\overline{a\cdot\bar{b}+\bar{a}\cdot b}}=\overline{\overline{\bar{a}\cdot b}\cdot\overline{a\cdot\bar{b}}}.$$

Address	Value	Comments
x3000	**1001 111 001 111111**	NOT R7, R1
x3001	1001 110 010 111111	NOT R6, R2
x3002	0101 101 111 000 010	AND R5, R7, R2
x3003	**0101 100 110 000 001**	AND R4, R6, R1
x3004	1001 001 101 111111	NOT R1, R5
x3005	**1001 010 100 111111**	NOT R2, R4
x3006	**0101 000 001 000 010**	AND R0, R1, R2
x3007	1001 011 000 111111	NOT R3, R0

T9

- 注意：NOP指令就是执行结果对目前机器的状态毫无影响，像是ADD R0, R0, #0这种就不行，因为可能会改变状态码
- 题目中五段代码翻译成汇编如下：
  - 1. ADD R2,R1,#2 ; R2=R1+2
  - 2. BRnzp PC+0 ; PC=PC
  - 3. BRnp PC+4 ; PC=PC+4 if np
  - 4. NOT R2,R7 ; R2=~R7
  - 5. TRAP x23 ; print a prompt on the screen and read a single character fromthe keyboard
- Trap指令具体服务例程可以看附录



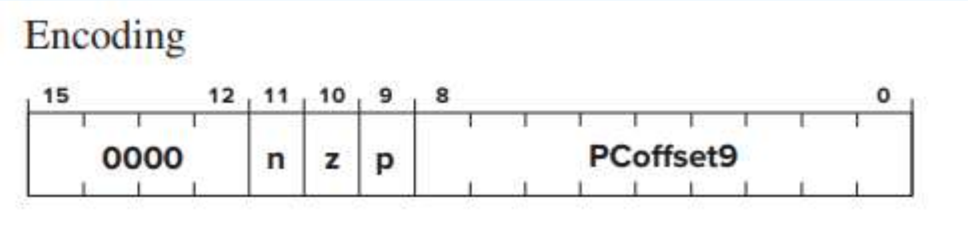
A.3 Interrupt and Exception Processing 675

Table A.3 Trap Service Routines

Trap Vector	Assembler Name	Description
x20	GETC	Read a single character from the keyboard. The character is not echoed onto the console. Its ASCII code is copied into R0. The high eight bits of R0 are cleared.
x21	OUT	Write a character in R0[7:0] to the console display.
x22	PUTS	Write a string of ASCII characters to the console display. The characters are contained in consecutive memory locations, one character per memory location, starting with the address specified in R0. Writing terminates with the occurrence of x0000 in a memory location.
x23	IN	Print a prompt on the screen and read a single character from the keyboard. The character is echoed onto the console monitor, and its ASCII code is copied into R0. The high eight bits of R0 are cleared.
x24	PUTSP	Write a string of ASCII characters to the console. The characters are contained in consecutive memory locations, two characters per memory location, starting with the address specified in R0. The ASCII code contained in bits [7:0] of a memory location is written to the console first. Then the ASCII code contained in bits [15:8] of that memory location is written to the console. (A character string consisting of an odd number of characters to be written will have x00 in bits [15:8] of the memory location containing the last character to be written.) Writing terminates with the occurrence of x0000 in a memory location.
x25	HALT	Halt execution and print a message on the console.

T10

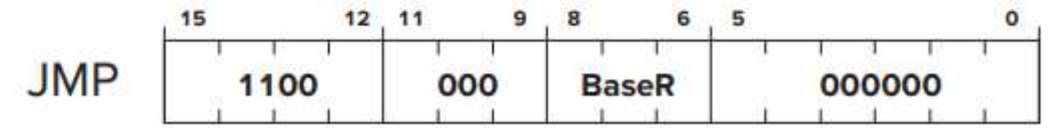
- BR:



Encoding

15–12	11	10	9	8–0
0000	n	z	p	PCoffset9

- JMP:



	15–12	11–9	8–6	5–0
JMP	1100	000	BaseR	000000

- BR指令无法跳转到PC寄存器和偏移量之和范围以外的指令，即其最大范围为距离当前指令+255到-256的地址空间

- 对于JMP指令，其能够将对应寄存器的内容装入PC寄存器，使得程序执行流可以跳转至内存空间的任意位置。

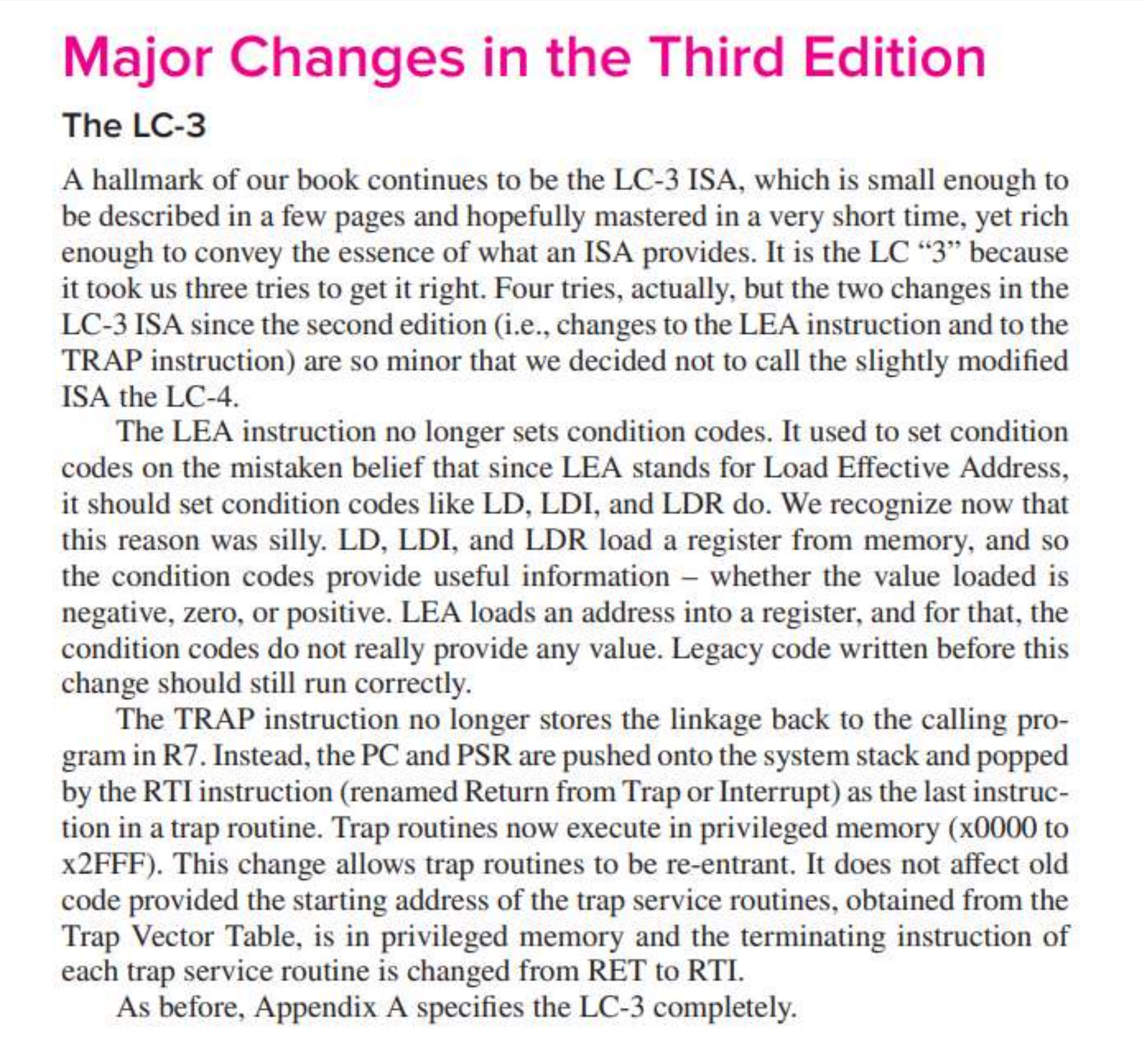
02

Midterm

备考讲解

考前须知：

- 开卷考试！
- 考试内容一切都以第三版英文原版书为主，注意第二版和第三版的差异（可以在第三版的Preface-Major Changes in the Third Edition中得到大概有那些差异）：



Major Changes in the Third Edition

The LC-3

A hallmark of our book continues to be the LC-3 ISA, which is small enough to be described in a few pages and hopefully mastered in a very short time, yet rich enough to convey the essence of what an ISA provides. It is the LC "3" because it took us three tries to get it right. Four tries, actually, but the two changes in the LC-3 ISA since the second edition (i.e., changes to the LEA instruction and to the TRAP instruction) are so minor that we decided not to call the slightly modified ISA the LC-4.

The LEA instruction no longer sets condition codes. It used to set condition codes on the mistaken belief that since LEA stands for Load Effective Address, it should set condition codes like LD, LDI, and LDR do. We recognize now that this reason was silly. LD, LDI, and LDR load a register from memory, and so the condition codes provide useful information – whether the value loaded is negative, zero, or positive. LEA loads an address into a register, and for that, the condition codes do not really provide any value. Legacy code written before this change should still run correctly.

The TRAP instruction no longer stores the linkage back to the calling program in R7. Instead, the PC and PSR are pushed onto the system stack and popped by the RTI instruction (renamed Return from Trap or Interrupt) as the last instruction in a trap routine. Trap routines now execute in privileged memory (x0000 to x2FFF). This change allows trap routines to be re-entrant. It does not affect old code provided the starting address of the trap service routines, obtained from the Trap Vector Table, is in privileged memory and the terminating instruction of each trap service routine is changed from RET to RTI.

As before, Appendix A specifies the LC-3 completely.

考前须知：

- 考试范围： 前五章内容
  - 第二章： 20%
  - 第三章： 35%（数字逻辑电路、状态机）
  - 第四章： 20%
  - 第五章： 20%

- 题型：

- 简答题

- 电路分析题

- 代码分析题

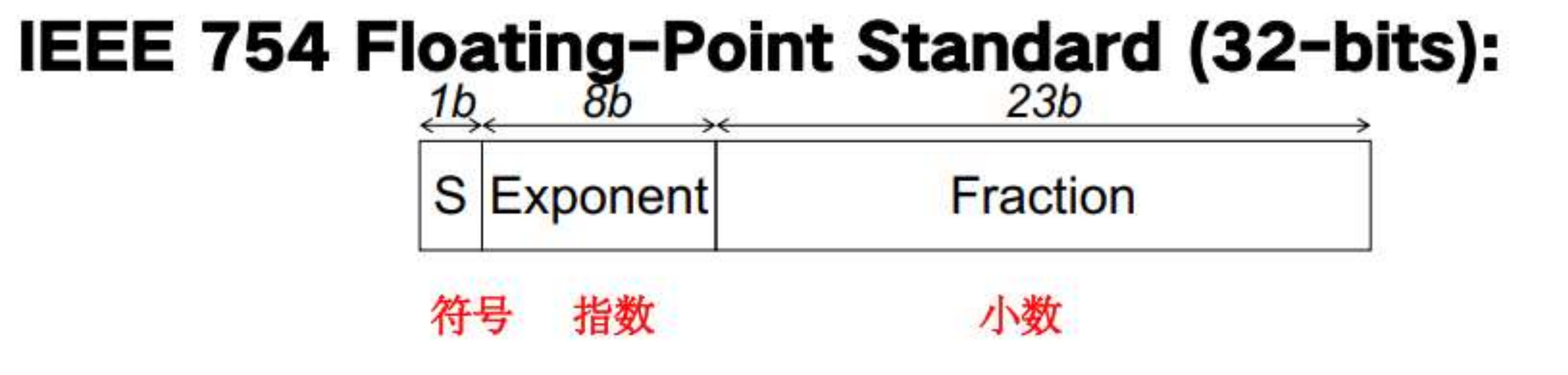
常用知识总结：

计算机常用进制及其英文必须牢记

序号	进制	缩写	全拼	读音
1	十六进制	HEX	Hexadecimal	[ˌheksə'desɪml]
2	十进制	DEC	Decimal	['desɪml]
3	八进制	OCT	Octal	['ɒktəl]
4	二进制	BIN	Binary	['baɪnərɪ]

常用知识总结：

浮点数总结



IEEE 754 Floating-Point Standard (32-bits):

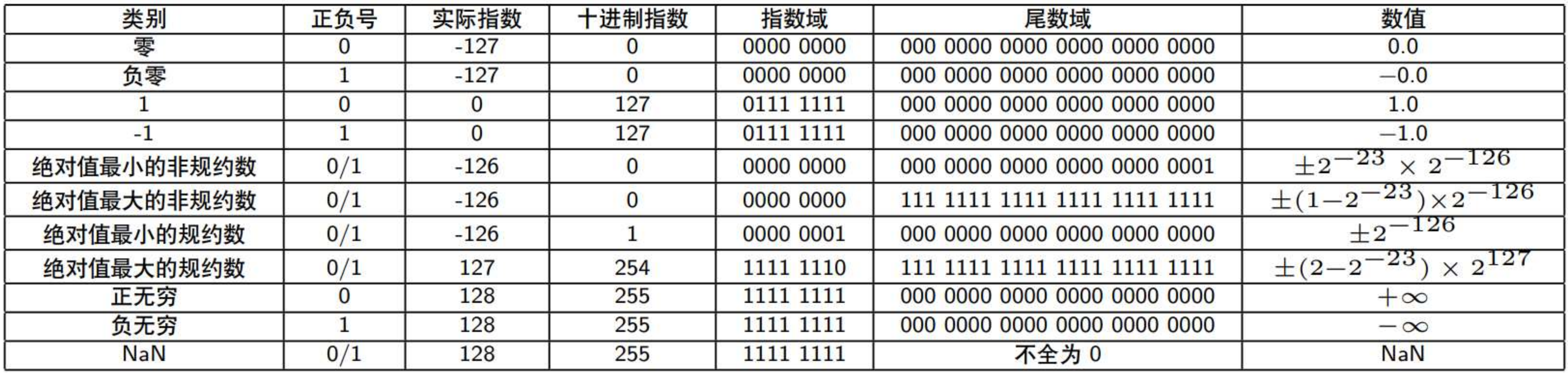
规范来讲指数部分大小从1~254（**全为0和全为1为特殊情况**）， 需要-127修正，故最小指数为-126，最大指数为127

记住以下特殊情况：

- **指数部分=全1，尾数部分=全0**，表示±无穷大（±Inf），正负号由符号位决定
- **指数部分=全1，尾数部分≠全0**，表示**NaN**（Not-a-Number）。NaN用于表示非数值结果，比如对负数开方（在实数域范围内是无解的）。NaN不区分正负。
- **指数部分=全0**，此时的数值被称为**subnormal number**。此时代表计数法为0.(小数部分)*2^-126

常用知识总结：

浮点数总结



类别	正负号	实际指数	十进制指数	指数域	尾数域	数值
零	0	-127	0	0000 0000	000 0000 0000 0000 0000 0000	0.0
负零	1	-127	0	0000 0000	000 0000 0000 0000 0000 0000	-0.0
1	0	0	127	0111 1111	000 0000 0000 0000 0000 0000	1.0
-1	1	0	127	0111 1111	000 0000 0000 0000 0000 0000	-1.0
绝对值最小的非规约数	0/1	-126	0	0000 0000	000 0000 0000 0000 0000 0001	$\pm 2^{-23} \times 2^{-126}$
绝对值最大的非规约数	0/1	-126	0	0000 0000	111 1111 1111 1111 1111 1111	$\pm(1-2^{-23})\times 2^{-126}$
绝对值最小的规约数	0/1	-126	1	0000 0001	000 0000 0000 0000 0000 0000	$\pm 2^{-126}$
绝对值最大的规约数	0/1	127	254	1111 1110	111 1111 1111 1111 1111 1111	$\pm(2-2^{-23}) \times 2^{127}$
正无穷	0	128	255	1111 1111	000 0000 0000 0000 0000 0000	$+\infty$
负无穷	1	128	255	1111 1111	000 0000 0000 0000 0000 0000	$-\infty$
NaN	0/1	128	255	1111 1111	不全为 0	NaN

如何备考?

- 考前可以多刷Patt往年卷，注意课程编号EE306，期中考内容大概对应Part1，我们的卷子排版、模式都是仿照EE306卷子出的
- 由于开卷 可以借本中文书翻得快（其实考试的时候不要寄希望于翻书，按照我去年的经验，翻附录查一下指令就已经够了）
- 打印英文书附录A，C（期末也用得到，主要是指令表和数据通路、状态图），也可以参考patt往年卷附录部分，打印那几页也行
- 也可以带往年卷，因为题型相似可以找找灵感（前提是你熟悉往年卷）
- 学有余力可以看课本附录奇数题，对着答案过一遍（反正我当年是没时间看）

课程资源 | ICS Fall 2023 (ics01-23.github.io)

EE 306 - Exams (utexas.edu)

谢谢！

中国科学技术大学
University of Science and Technology of China