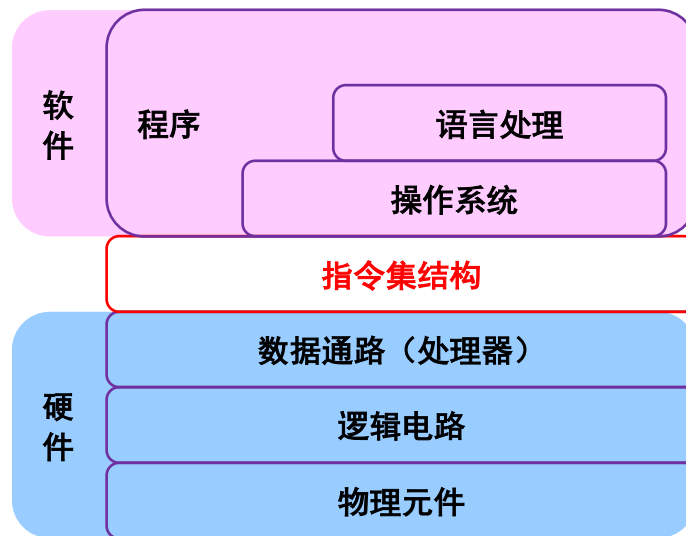


第十章 机器语言程序设计

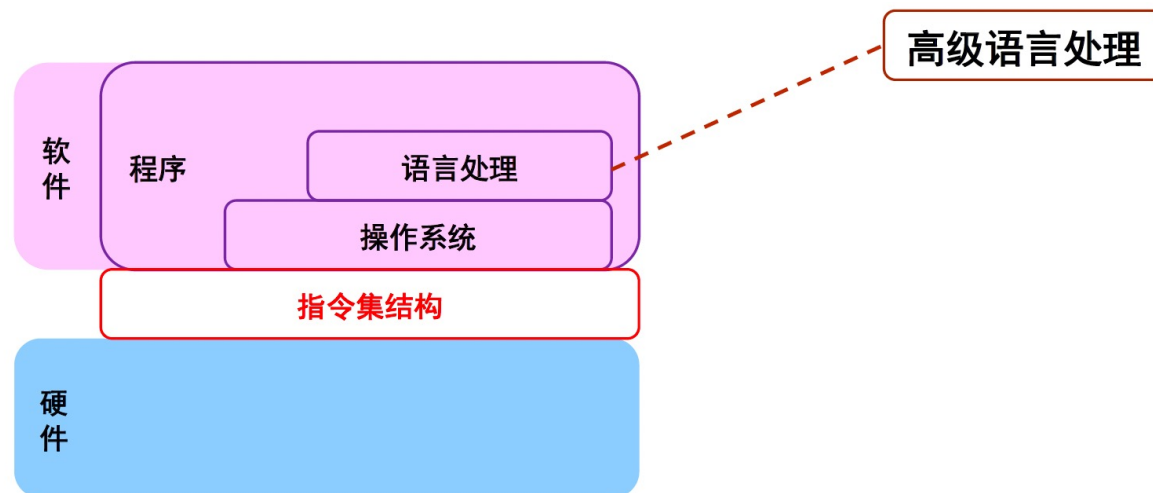
程序设计语言与ISA

- 指令集结构（ISA）是计算机硬件和软件之间的接口
 - 处理器设计的依据
 - 编写程序时所要注意的全部信息



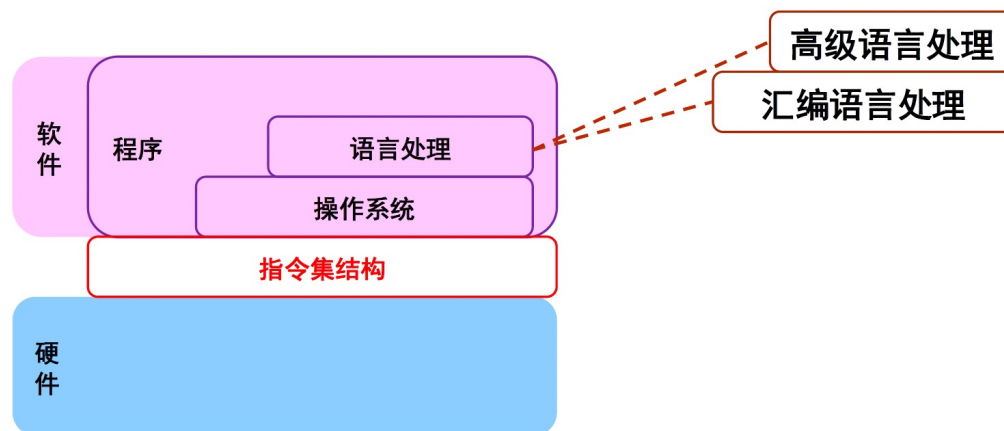
高级语言

- 与底层计算机指令集无关
- “独立于机器”
- 不能直接被计算机执行
- 被翻译为目标机器ISA的二进制指令序列



低级语言

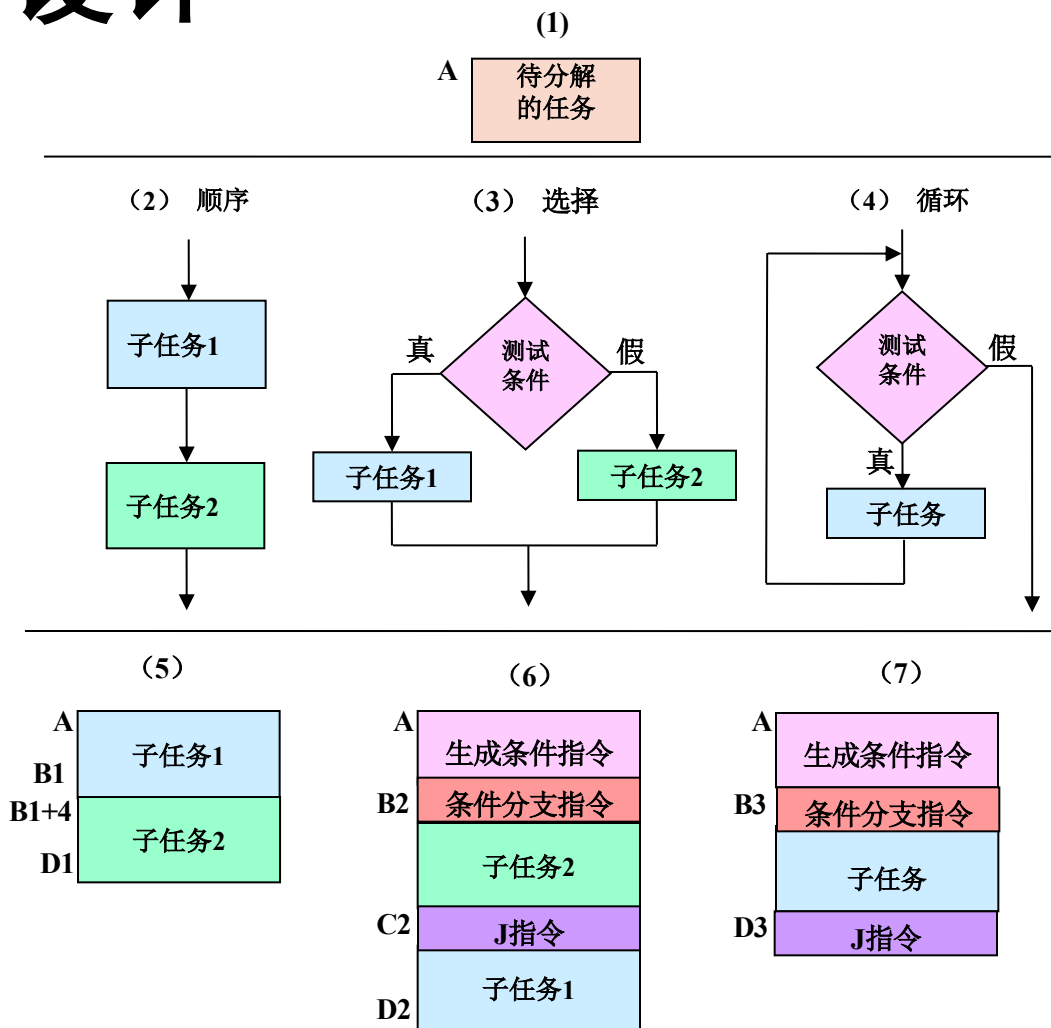
- 与执行程序的计算机指令集紧密相关
- **机器语言**
 - 依据指令集使用二进制编码，直接在计算机上执行，不需要经过语言处理
- **汇编语言**
 - 依据指令集的汇编语言格式编写，需经过语言处理，翻译为机器语言才能在计算机上执行



结构化程序设计

- 三种基本结构

- 顺序
- 选择
- 循环



顺序

(1)

A

待分解
的任务

(2) 顺序



子任务1



子任务2



(5)

A

子任务1

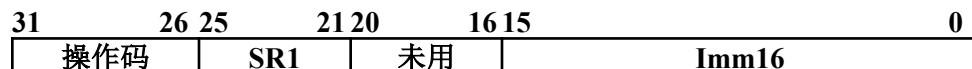
B1

B1+4

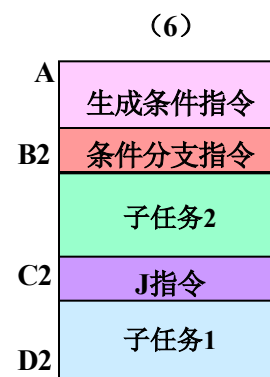
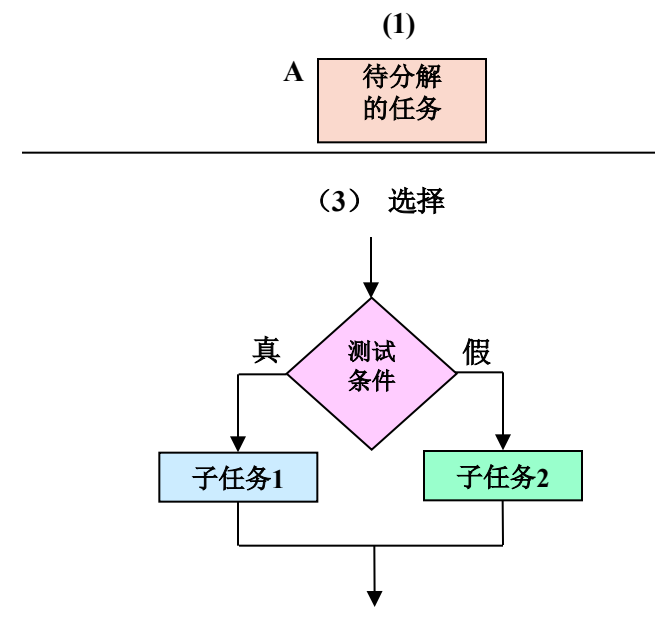
D1

子任务2

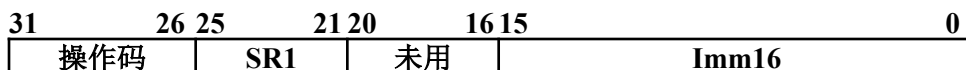
选择



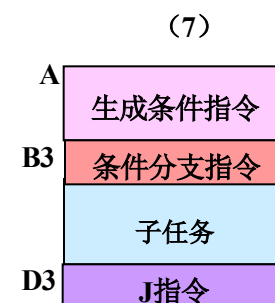
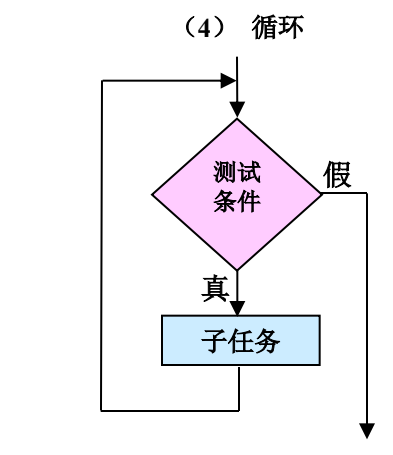
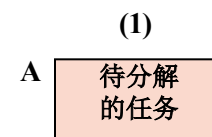
- 一组指令序列生成条件
 - 将**某个**寄存器Rx设置为零（假）/非零（真）
- 地址B2 “条件分支指令” 测试该寄存器
 - 条件为真（BNEZ Rx, Y）
 - $PC \leftarrow C2 + 4$
 - 立即数Y: 子任务2的指令数目加1后再乘以4
 - 条件为假
 - $PC \leftarrow B2 + 4$
 - 子任务2
 - 终止于C2中的无条件跳转指令
 - $PC \leftarrow D2 + 4$
 - J指令中的立即数: 子任务1的指令数目乘以4



循环



- 一组指令序列生成条件
 - 将**某个寄存器Rx**设置为零（**假**）/非零（**真**）
- 地址B3 “条件分支指令” 测试该寄存器
 - 条件为**假**（BEQZ Rx, Y）
 - $PC \leftarrow D3+4$
 - 立即数Y：子任务的指令数目加1后再乘以4
 - 条件为**真**
 - $PC \leftarrow B3+4$
 - 子任务
 - 结束于D3中的无条件跳转指令
 - $PC \leftarrow A$
 - 问题：J指令中的立即数应为多少？

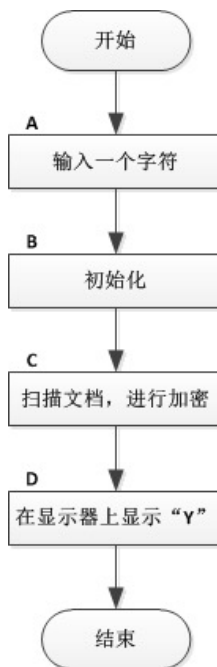


示例：文档加密

- 根据键盘输入的数值 n （0到9之间的整数），对文档进行加密
- 加密算法：如果文档中的字符ASCII码值大于“ $126-n$ ”，那么，将该字符减去“ $94-n$ ”，并替换原来的字符；而其他字符则加上 n ，进行替换；最后在显示器上显示字符“Y”，表示加密结束。假设文档中的字符ASCII码值在33~126范围内。

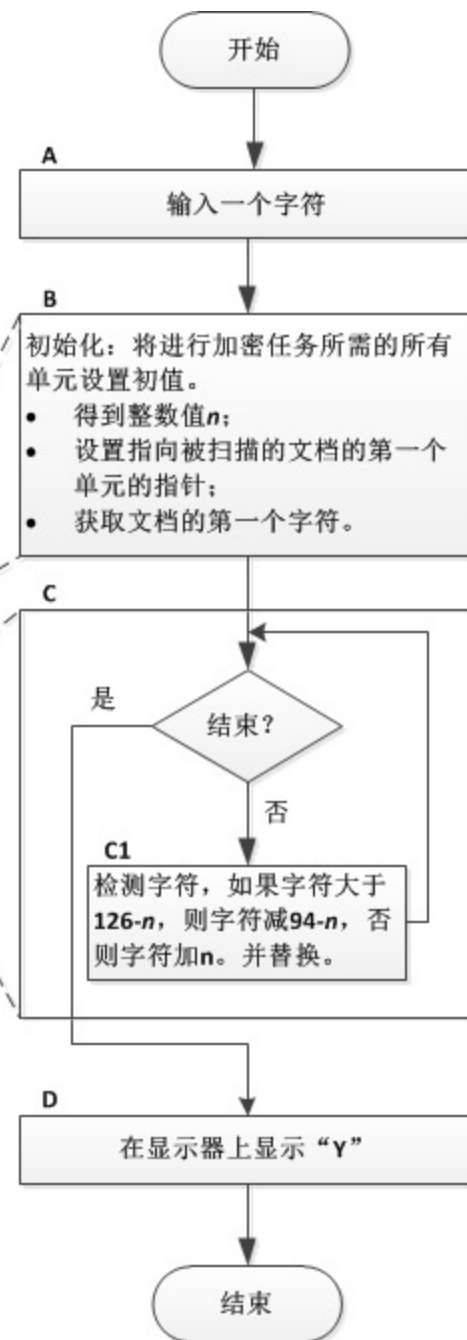
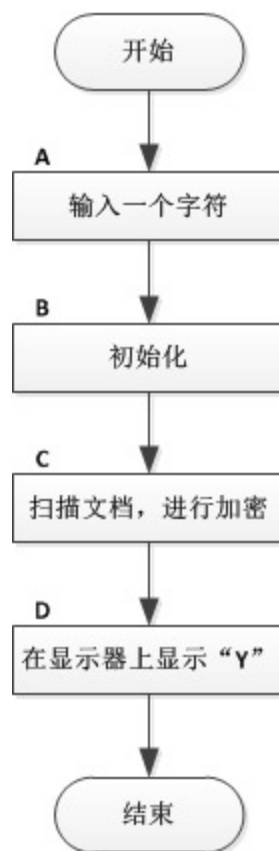
系统分解过程

- 分解为由4个子任务组成的顺序结构
 - 初始化：得到数值 n ，将指针指向被检查文档中第一个字符的地址，然后从被检查文档中提取第一个字符。



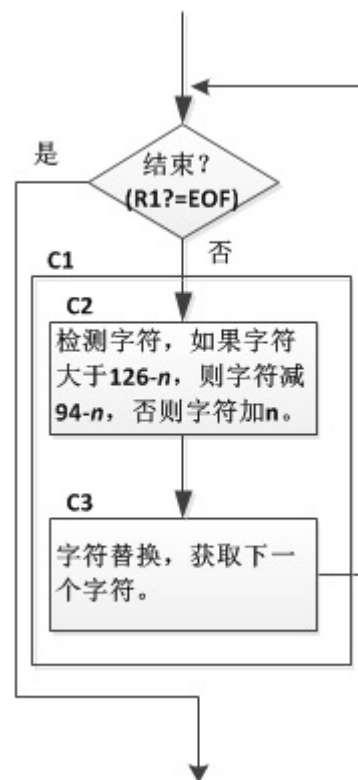
分解C

- 循环结构：只要该文档还有字符需要加密
- 文档结束，标志为 EOT（传输结束，ASCII码为00000100）

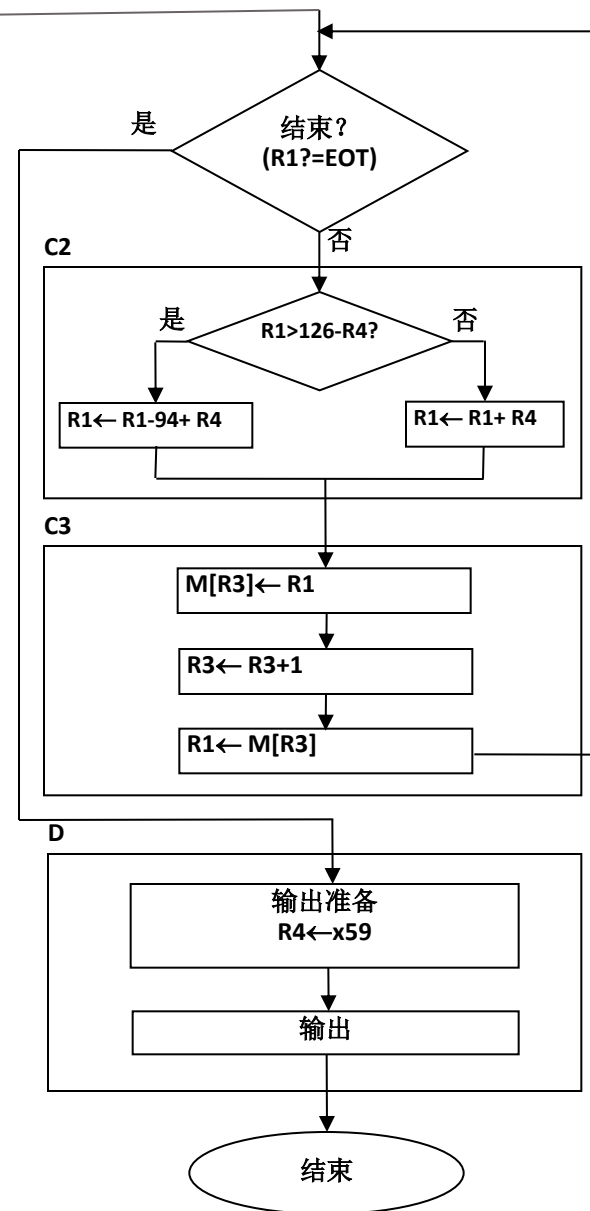
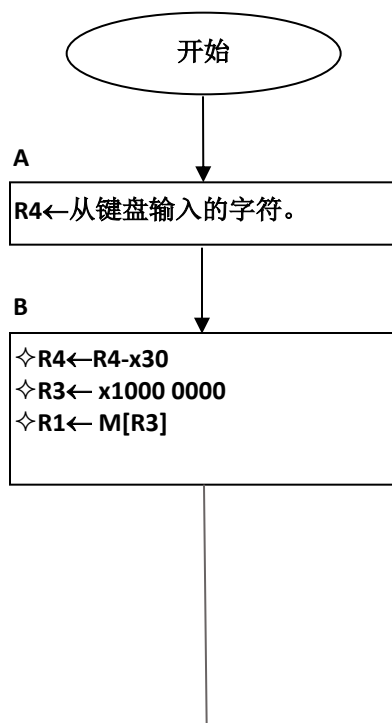


分解C1

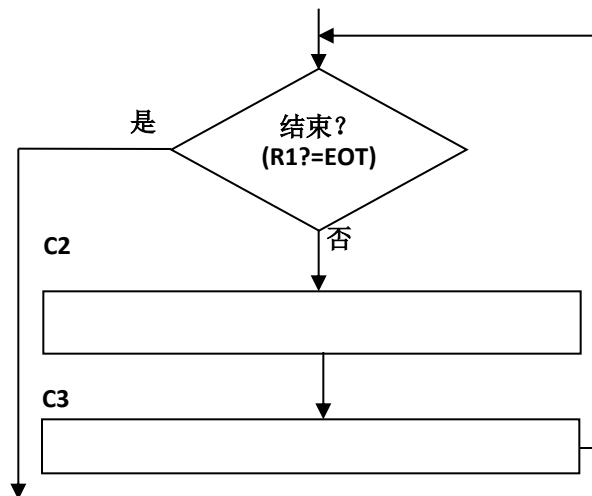
- 两个顺序的子任务C2和C3



- 使用选择结构代替C2

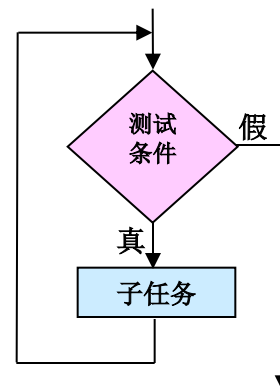


循环结构

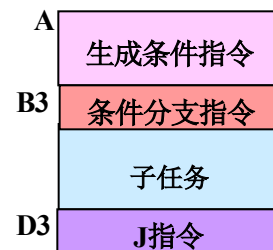


地址	31	26	25	21	20	16	15	11	10	6	5	0	
x0400 0000	110000				000000	0000	0000	0000	0000	0110			TRAP x06/IN
x0400 0004	000011		00100		00100			0000	0000	0011	0000		SUBI R4, R4, x30
x0400 0008	001100		00000		00011			0001	0000	0000	0000		LHI R3, x1000
x0400 000C	010110		00011		00001			0000	0000	0000	0000		LB R1, 0(R3)
x0400 0010	010100		00001		00010			0000	0000	0000	0100		SEQI R2, R1, #4
x0400 0014	101001		00010					00000	0000	0011	0000		BNEZ R2, x30
x0400 0018	000001		00000		00101			0000	0000	0111	1111		ADDI R5, R0, x7F
x0400 001C	000000		00101		00100		00101		000000		000011		SUB R5, R5, R4
x0400 0020	000000		00001		00101		00010		000000		010000		SLT R2, R1, R5
x0400 0024	101001		00010					00000	0000	0000	1100		BNEZ R2, x0C
x0400 0028	000011		00101		00101			0000	0000	0010	0001		SUBI R5, R5, x21
x0400 002C	000000		00001		00101		00001		000000		000011		SUB R1, R1, R5
x0400 0030	101100				000000	0000	0000	0000	0000	0100			J x04
x0400 0034	000000		00001		00100		00001		000000		000001		ADD R1, R1, R4
x0400 0038	010111		00011		00001			0000	0000	0000	0000		SB 0(R3), R1
x0400 003C	000001		00011		00011			0000	0000	0000	0001		ADDI R3, R3, #1
x0400 0040	010110		00011		00001			0000	0000	0000	0000		LB R1, 0(R3)
x0400 0044	101100				111111	1111	1111	1111	1100	1000			J #-56
x0400 0048	000001		00000		00100			0000	0000	0101	1001		ADDI R4, R0, x59
x0400 004C	110000				000000	0000	0000	0000	0000	0111			TRAP x07/OUT
x0400 0050	110000				000000	0000	0000	0000	0000	0000	0000		TRAP x00/HALT

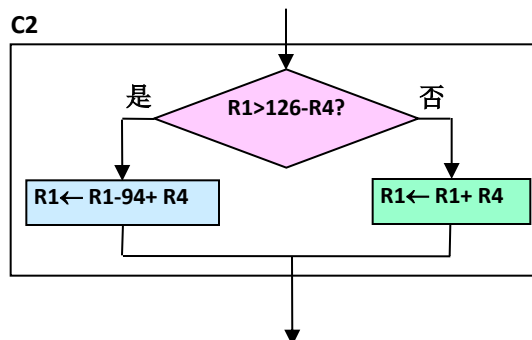
(4) 循环



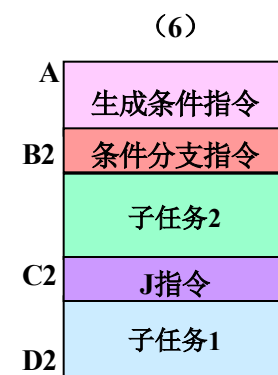
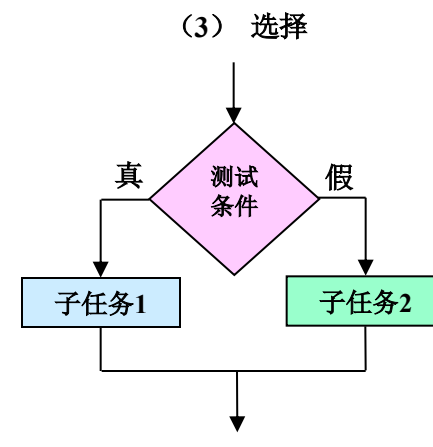
(7)



C2选择结构



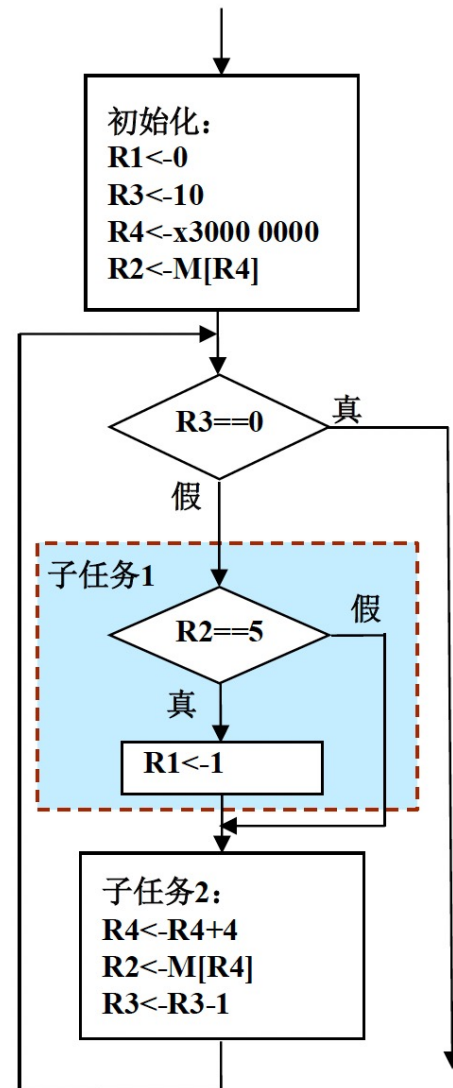
地址	31	26	25	21	20	16	15	11	10	6	5	0		
x0400 0000	110000		000000 0000 0000 0000 0110										TRAP x06/IN	
x0400 0004	000011		00100		00100		0000 0000 0011 0000							SUBI R4, R4, x30
x0400 0008	001100		00000		00011		0001 0000 0000 0000							LHI R3, x1000
x0400 000C	010110		00011		00001		0000 0000 0000 0000							LB R1, 0(R3)
x0400 0010	010100		00001		00010		0000 0000 0000 0100							SEQI R2, R1, #4
x0400 0014	101001		00010		00000 0000 0011 0000								BNEZ R2, x30	
x0400 0018	000001		00000		00101		0000 0000 0111 1111							ADDI R5, R0, x7F
x0400 001C	000000		00101		00100		00101		000000		000011		SUB R5, R5, R4	
x0400 0020	000000		00001		00101		00010		000000		010000		SLT R2, R1, R5	
x0400 0024	101001		00010		00000 0000 0000 1100								BNEZ R2, x0C	
x0400 0028	000011		00101		00101		0000 0000 0010 0001							SUBI R5, R5, x21
x0400 002C	000000		00001		00101		00001		000000		000011		SUB R1, R1, R5	
x0400 0030	101100		000000 0000 0000 0000 0000 0100										J x04	
x0400 0034	000000		00001		00100		00001		000000		000001		ADD R1, R1, R4	
x0400 0038	010111		00011		00001		0000 0000 0000 0000							SB 0(R3), R1
x0400 003C	000001		00011		00011		0000 0000 0000 0001							ADDI R3, R3, #1
x0400 0040	010110		00011		00001		0000 0000 0000 0000							LB R1, 0(R3)
x0400 0044	101100		111111 1111 1111 1111 1100 1000										J #-56	
x0400 0048	000001		00000		00100		0000 0000 0101 1001							ADDI R4, R0, x59
x0400 004C	110000		000000 0000 0000 0000 0000 0111										TRAP x07/OUT	
x0400 0050	110000		000000 0000 0000 0000 0000 0000										TRAP x00/HALT	



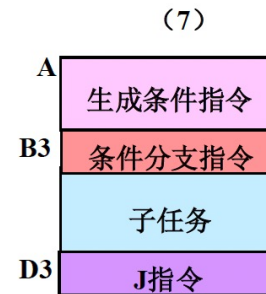
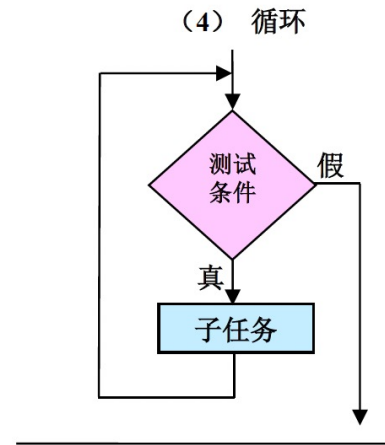
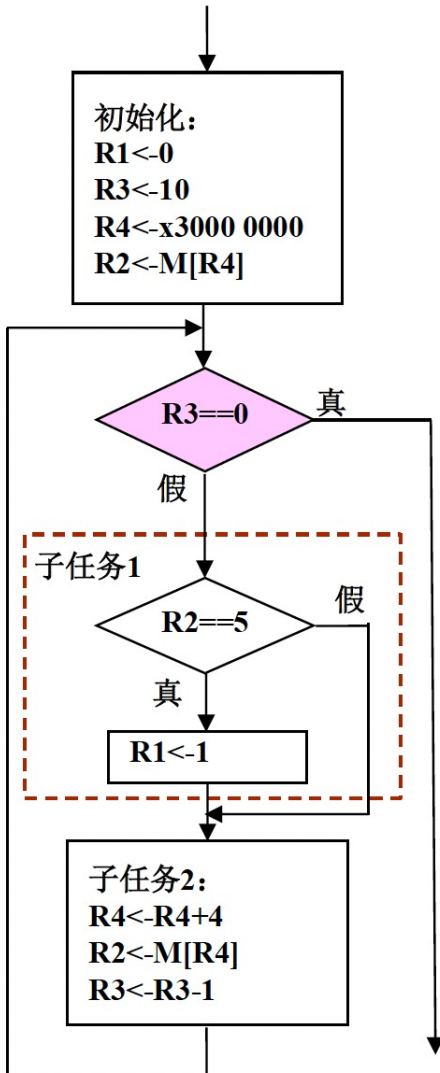
示例：判断连续存储单元内是否包含5

- 检查：
- 从地址x3000 0000开始存储的10个整数
 - 有5，R1设置为1
 - 没有5，R1为0

- 计数器控制的循环
 - R3, 计数器
- 子任务1
 - 选择结构

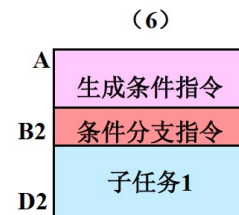
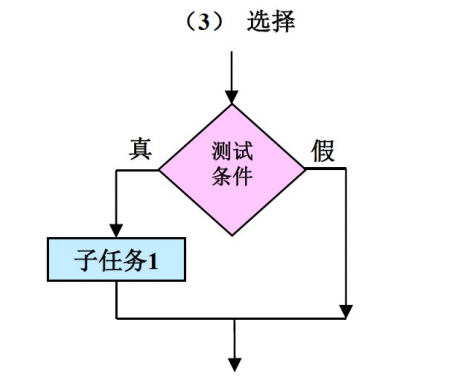
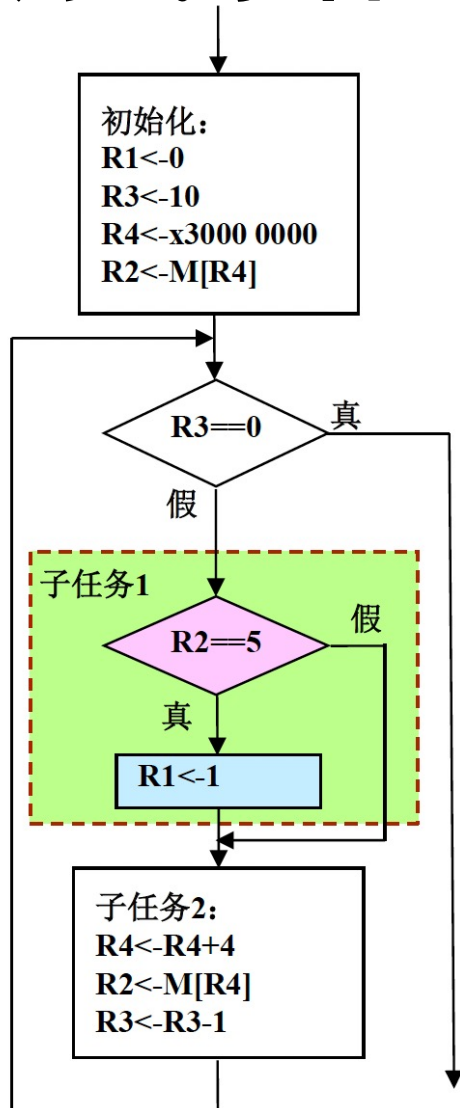


测试条件 R3==0



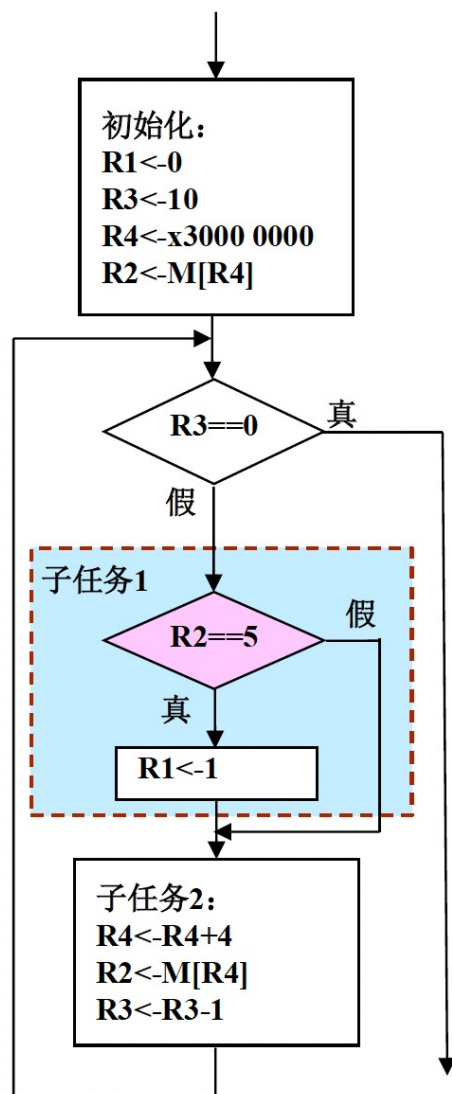
- 不需要生成条件指令
- 条件分支指令
 - BEQZ R3, D3+4

测试条件 R2==5



- 生成条件指令
 - SEQI Rx, R2, #5
- 条件分支指令
 - BEQZ Rx, D2+4

机器语言程序



31	26	25	21	20	16	15	11	10	6	5	0	解释
001001		00001		00001		0000 0000 0000 0000						ANDI R1,R1, #0
000001		00000		00011		0000 0000 0000 1010						ADDI R3,R0, #10
001100		00000		00100		0011 0000 0000 0000						LHI R4, x3000
011100		00100		00010		0000 0000 0000 0000						LW R2, 0(R4)
101000		00011		00000		0000 0000 0010 0000						BEQZ R3, #32
010100		00010		00101		0000 0000 0000 0101						SEQI R5, R2, #5
101000		00101		00000		0000 0000 0000 1000						BEQZ R5, #8
000001		00000		00001		0000 0000 0000 0001						ADDI R1,R0, #1
101100	00 0000 0000 0000 0000 0001 0000											J #16
000001		00100		00100		0000 0000 0000 0100						ADDI R4,R4, #4
011100		00100		00010		0000 0000 0000 0000						LW R2, 0(R4)
000011		00011		00011		0000 0000 0000 0001						SUBI R3,R3, #1
101100	11 1111 1111 1111 1111 1101 1100											J #-36
.....												

选择结构

- 当 $R2$ 为 5 时，设置 $R1$ 为 1
- 使用 J 指令跳出循环

示例：找到字中的第一个“1”

- 检查：

- x3000 0000~x3000 0003中的字

- 找出第一个“1”（从左到右）

- 存储到R1中

- 如果没有1

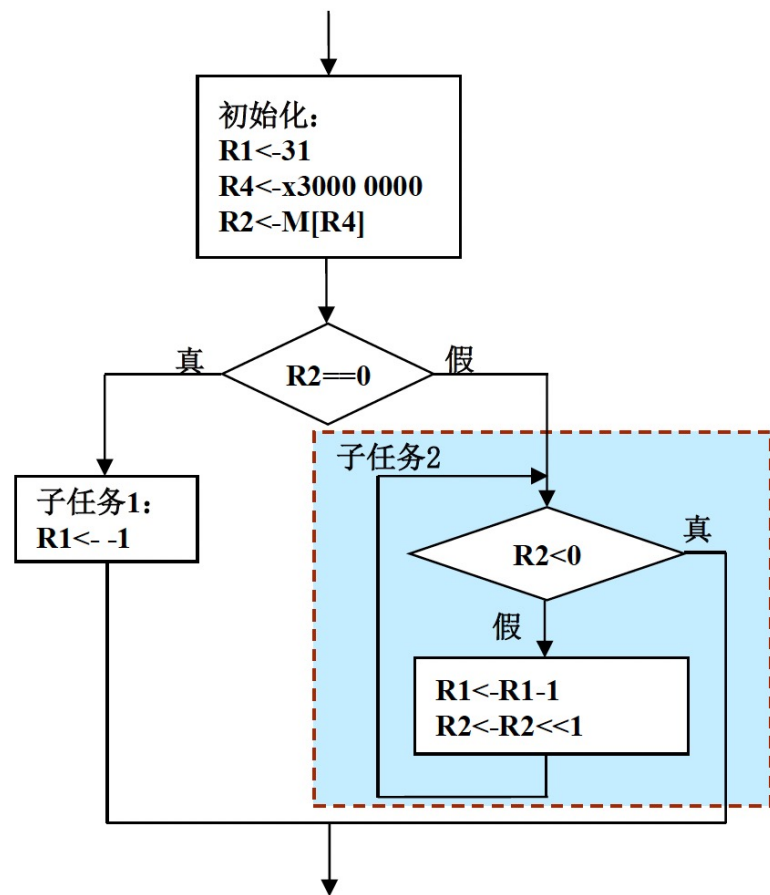
- $R1 \leftarrow -1$

- 例如

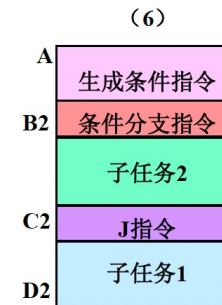
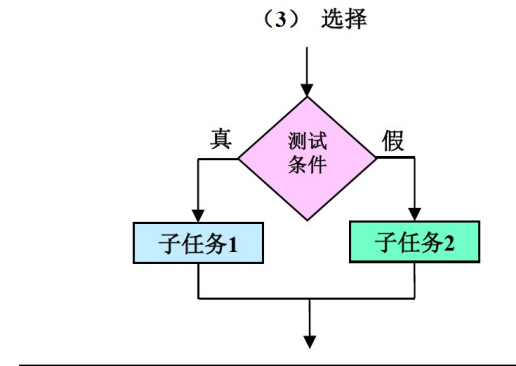
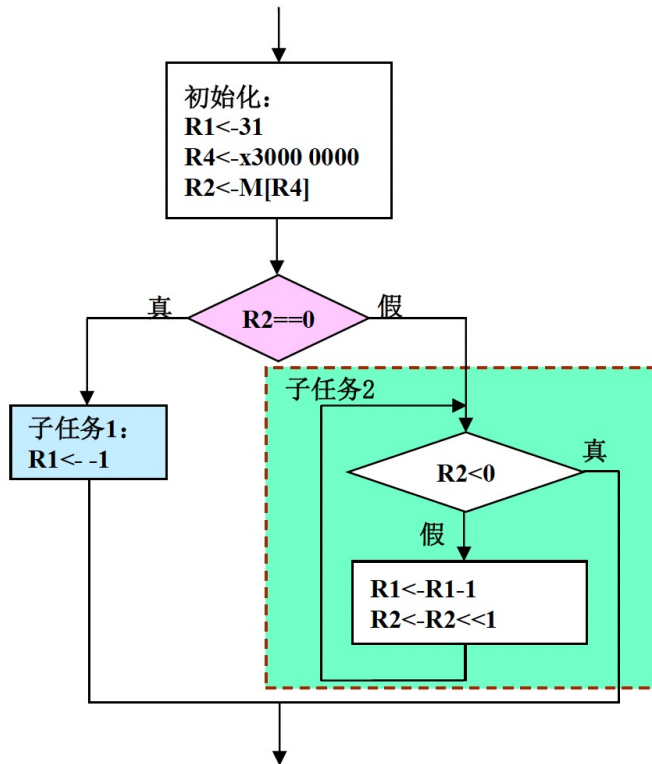
- 0010 0000 0000 0000 0000 0000 0000 0000, $R1=29$

- 0000 0000 0000 0000 0000 0000 0010 0000, $R1=5$

- 选择结构
- 子任务2
 - 标志控制的循环
 - 标志
 - $R2 < 0$: $R2[31] = 1$
 - 循环子任务
 - $R2 = R2 \ll 1$
 - $R2[30], R2[29] \dots == 1$?

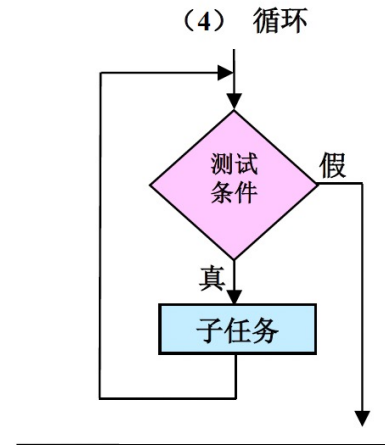
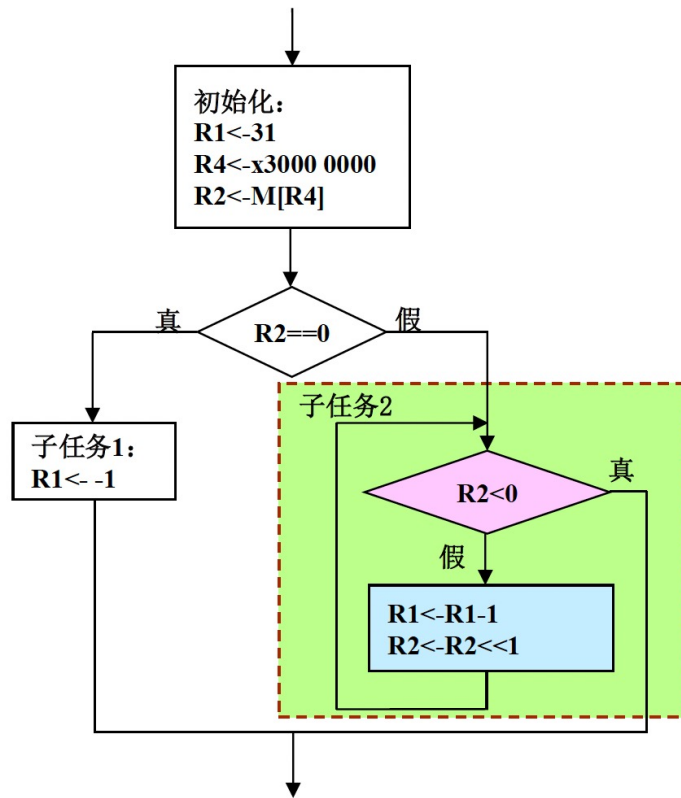


测试条件 R2==0

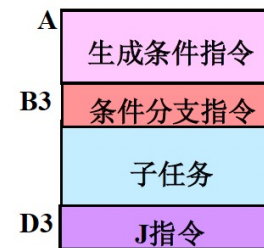


- 不需要生成条件指令
- 条件分支指令
 - BEQZ R2, C2+4

测试条件 $R2 < 0$



(7)



- 生成条件指令
 - `SLTI Rx, R2, #0`
- 条件分支指令
 - `BNEZ Rx, D3+4`

机器语言程序

31	26	25	21	20	16	15	11	10	6	5	0	解释
000001		00000		00001				0000	0000	0001	1111	ADDI R1,R0, #31
001100		00000		00100				0011	0000	0000	0000	LHI R4, x3000
011100		00100		00010				0000	0000	0000	0000	LW R2, 0(R4)
101000		00010		00000				0000	0000	0001	0100	BEQZ R2,#20
010000		00010		00011				0000	0000	0000	0000	SLTI R3, R2, #0
101001		00011		00000				0000	0000	0001	0000	BNEZ R3, #16
000011		00001		00001				0000	0000	0000	0001	SUBI R1,R1, #1
001101		00010		00010				0000	0000	0000	0001	SLLI R2,R2, #1
101100								111111	1111	1111	1110 1100	J #-20
001010		00000		00001				1111	1111	1111	1111	ORI R1,R0, #-1
.....												

测试和调试

- 在第十一章 DLX汇编语言编程后，再做介绍

书面作业

- 10. 1
- 10. 2