



第2讲—典型部件的设计与测试

计算机科学与技术学院

张瑞华

课程设计的三个步骤

课程设计
初始阶段

明确设计要求

方案设计

逻辑图设计

编程设计阶段

↓

用库元件或自定义元件
生成逻辑图

↓

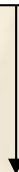
编译与优化

↓

生成编程文件

编程与测试阶段

编程下载



功能测试

❖ 课程设计步骤:

❖ 第一阶段为计算机的典型部件设计

✎ 学生通过对部件的设计, 以期达到对部件的构成、设计方法、工作原理及在计算机硬件中的功能作一系统的了解。

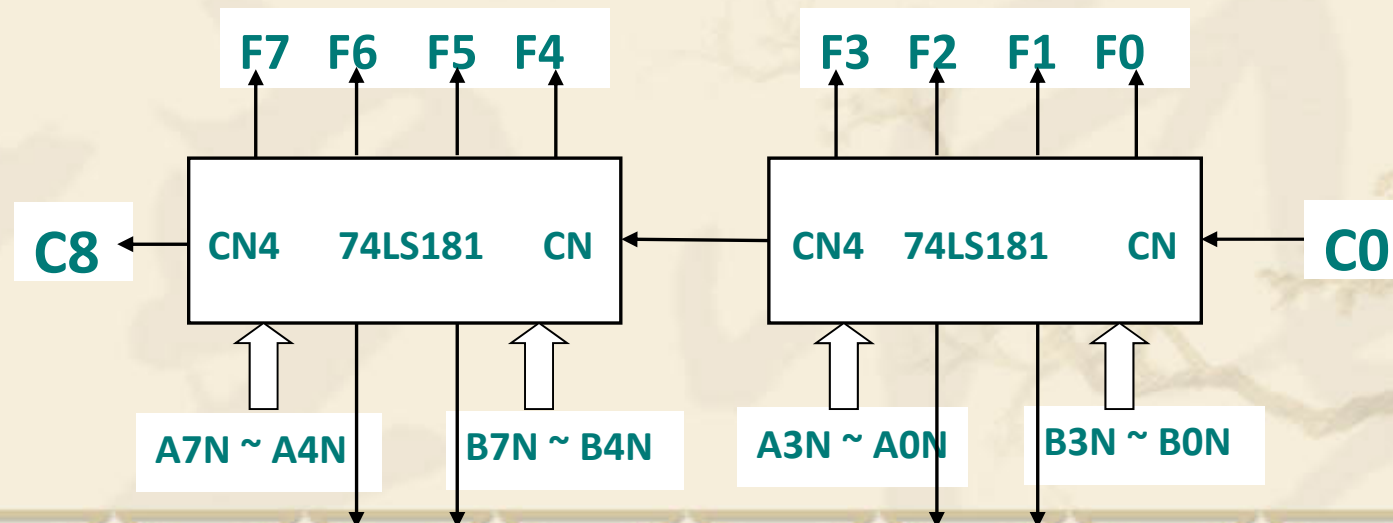
❖ 第二个阶段为计算机综合设计(微程序、硬布线2种实现方法)

✎ 学生将用多个部件构造一台较为复杂的计算机硬件系统

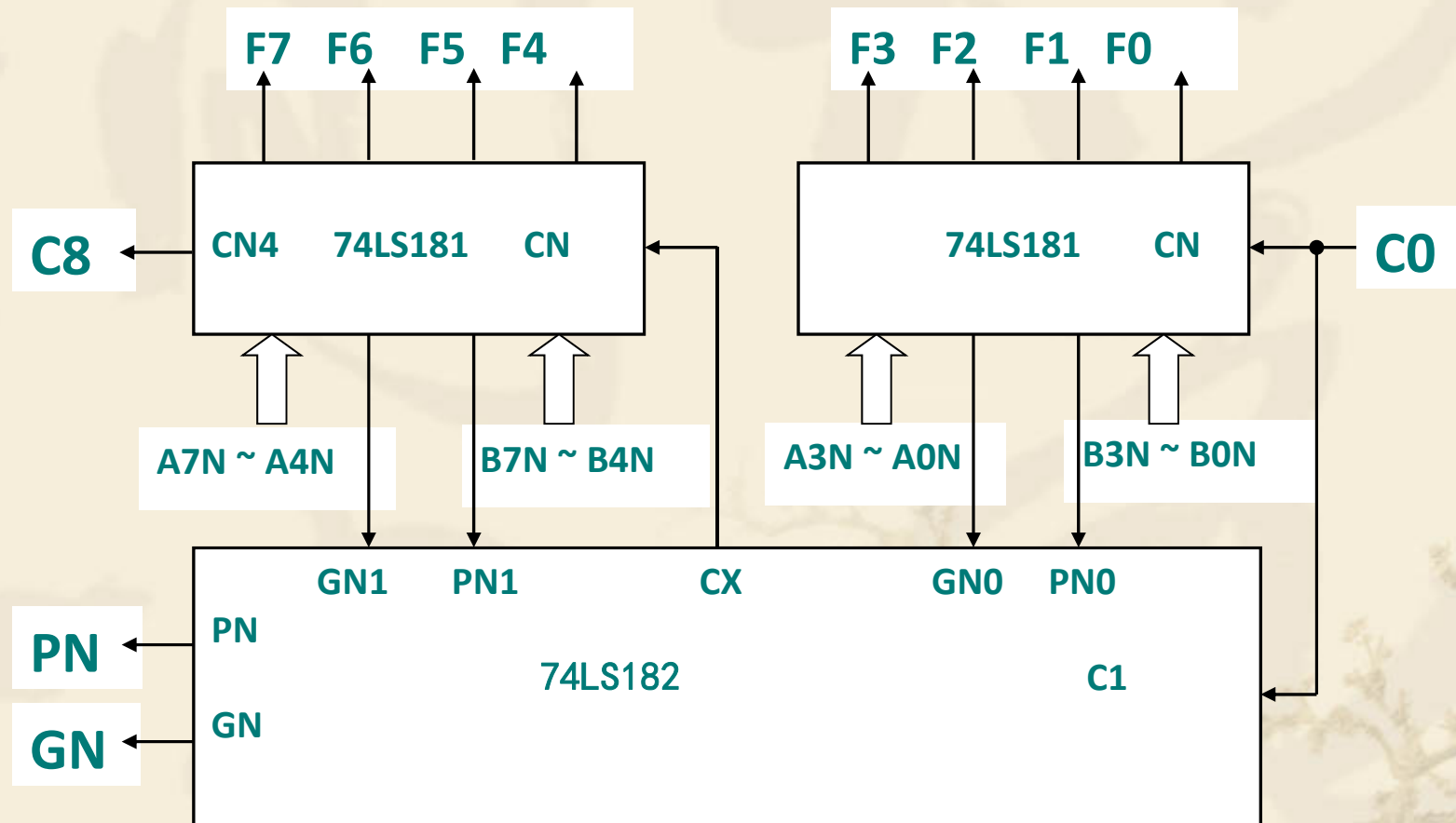
✎ 以期达到对计算机的总体设计、基本构成、基本原理有一个清楚的认识并能建立一个清晰的整机概念, 从而扎实地掌握一种数字系统的设计方法

2.1 典型部件课程设计

- ❖ 1、算术逻辑运算单元ALU的设计
- ❖ 在新的实验平台上,QuartusII环境下,可以利用集成电路二片74LS181运算器构成八位组间串行进位运算器,也可以借助先行进位发生器74LS182构成组间并行进位八位运算器.



- ❖ 也可以借助先行进位发生器74LS182构成组间超前进位并行八位运算器.



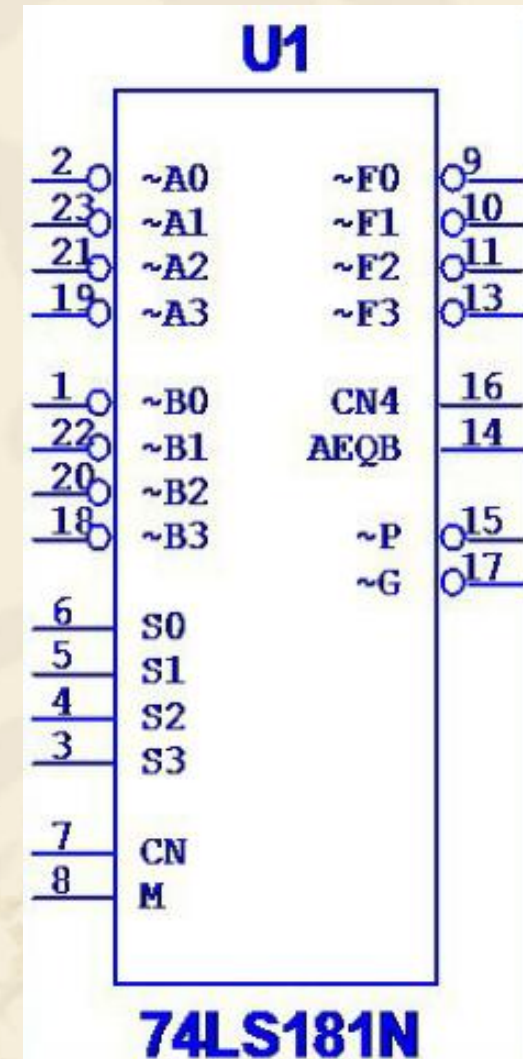
74LS181芯片的介绍

❖ 引脚图与引脚说明

ALU—74LS181 引脚说明：M=1 逻辑运算，M=0 算术运算。

引 脚	说 明
M 状态控制端	M=1 逻辑运算；M=0 算术运算
S3 S2 S1 S0 运算选择控制	S3 S2 S1 S0 决定电路执行哪一种算术
A3 A2 A1 A0	运算数 1，引脚 3 为最高位
B3 B2 B1 B0	运算数 2，引脚 3 为最高位
Cn 最低位进入输入	Cn=0 有进位，Cn=1 无进位
Cn+4 本片产生的进位信号	Cn+4=0 有进位，Cn+4=1 无进位
F3 F2 F1 F0	F3 F2 F1 F0 运算结果，F3 为最高位

- ❖ 引脚P、N是2个级联输出端，可同时产生快速进位，配合74ls182可产生全速并行进位链。（参考计组课本6.5节）



❖ 74ls181功能图:

❖ 算术运算关注:

- ❧ A加B(1001CN=1)
- ❧ A减B(0110CN=0)
- ❧ A加1(0000CN=0)
- ❧ A减1(1111CN=1)
- ❧ A加B加1 (1001CN=0)

❖ 逻辑运算关注:

逻辑异或(0110)

逻辑与(1011)

逻辑或(1110)

A逻辑非(0000)

B逻辑非(0101)

A直传(1111)

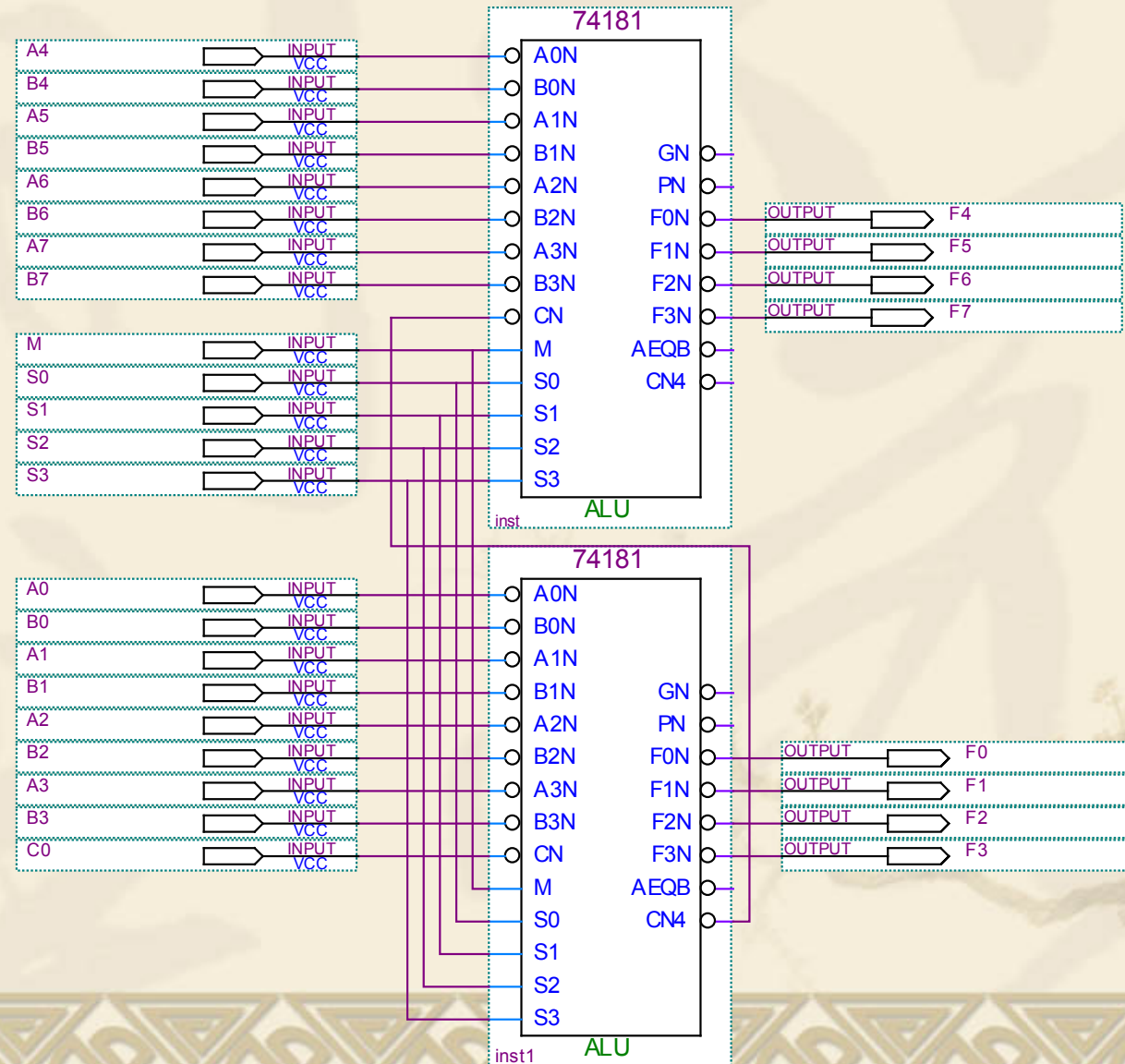
B直传(1010)

逻辑运算与CN无关

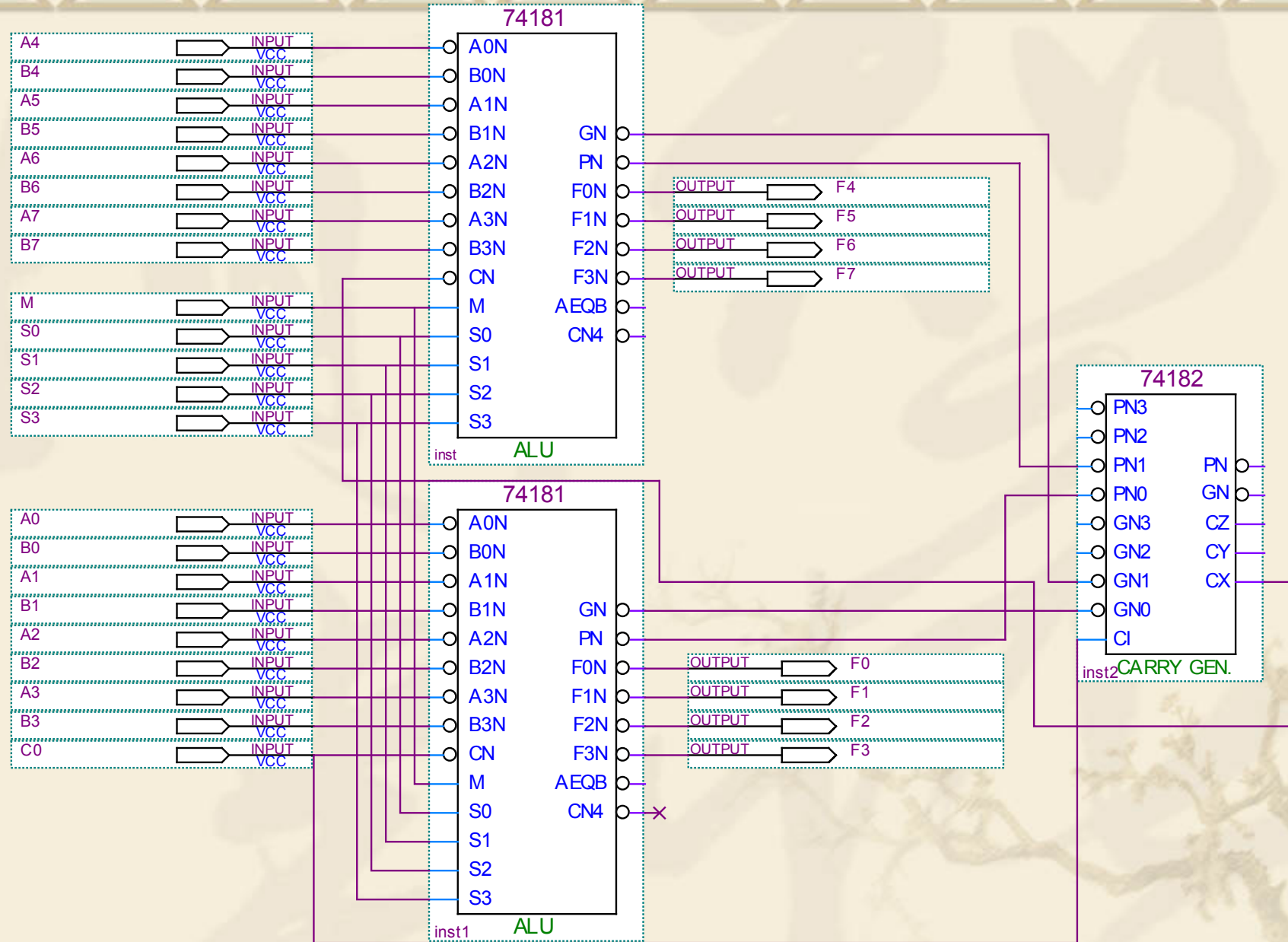
方式	M=1 逻辑运算	M=0 算术运算	
S3 S2 S1 S0	逻辑运算	CN=1 (无进位)	CN=0 (有进位)
0 0 0 0	$F = /A$	$F = A$	$F = A \text{ 加 } 1$
0 0 0 1	$F = / (A+B)$	$F = A+B$	$F = (A+B) \text{ 加 } 1$
0 0 1 0	$F = (/A)B$	$F = A+/B$	$F = (A+/B) \text{ 加 } 1$
0 0 1 1	$F = 0$	$F = \text{负 } 1$	$F = 0$
0 1 0 0	$F = / (AB)$	$F = A \text{ 加 } A (/B)$	$F = A \text{ 加 } A/B \text{ 加 } 1$
0 1 0 1	$F = /B$	$F = (A+B) \text{ 加 } A/B$	$F = (A+B) \text{ 加 } A/B \text{ 加 } 1$
0 1 1 0	$F = A \oplus B$	$F = A \text{ 减 } B \text{ 减 } 1$	$F = A \text{ 减 } B$
0 1 1 1	$F = A/B$	$F = A (/B) \text{ 减 } 1$	$F = A (/B)$
1 0 0 0	$F = /A+B$	$F = A \text{ 加 } AB$	$F = A \text{ 加 } AB \text{ 加 } 1$
1 0 0 1	$F = / (A \oplus B)$	$F = A \text{ 加 } B$	$F = A \text{ 加 } B \text{ 加 } 1$
1 0 1 0	$F = B$	$F = (A+/B) \text{ 加 } AB$	$F = (A+/B) \text{ 加 } AB \text{ 加 } 1$
1 0 1 1	$F = AB$	$F = AB \text{ 减 } 1$	$F = AB$
1 1 0 0	$F = 1$	$F = A \text{ 加 } A$	$F = A \text{ 加 } A \text{ 加 } 1$
1 1 0 1	$F = A+/B$	$F = (A+B) \text{ 加 } A$	$F = (A+B) \text{ 加 } A \text{ 加 } 1$
1 1 1 0	$F = A+B$	$F = (A+/B) \text{ 加 } A$	$F = (A+/B) \text{ 加 } A \text{ 加 } 1$
1 1 1 1	$F = A$	$F = A \text{ 减 } 1$	$F = A$

(上表中的“/”表示求反)

- ❖ 加法器的实现方法：利用2片74ls181组成8位的加法器，微控制信号相应增多：M、S3、S2、S1、S0。



8位超前进位并行加法器,也可用74181、74182设计实现



实验一：微程序控制的运算器设计

❖ 1、设计目的

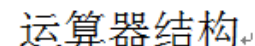
- ❧ ①熟悉简单运算器的结构。
- ❧ ②熟悉微命令的产生和时序。
- ❧ ③熟悉运算器功能测试。

❖ 2、设计简述

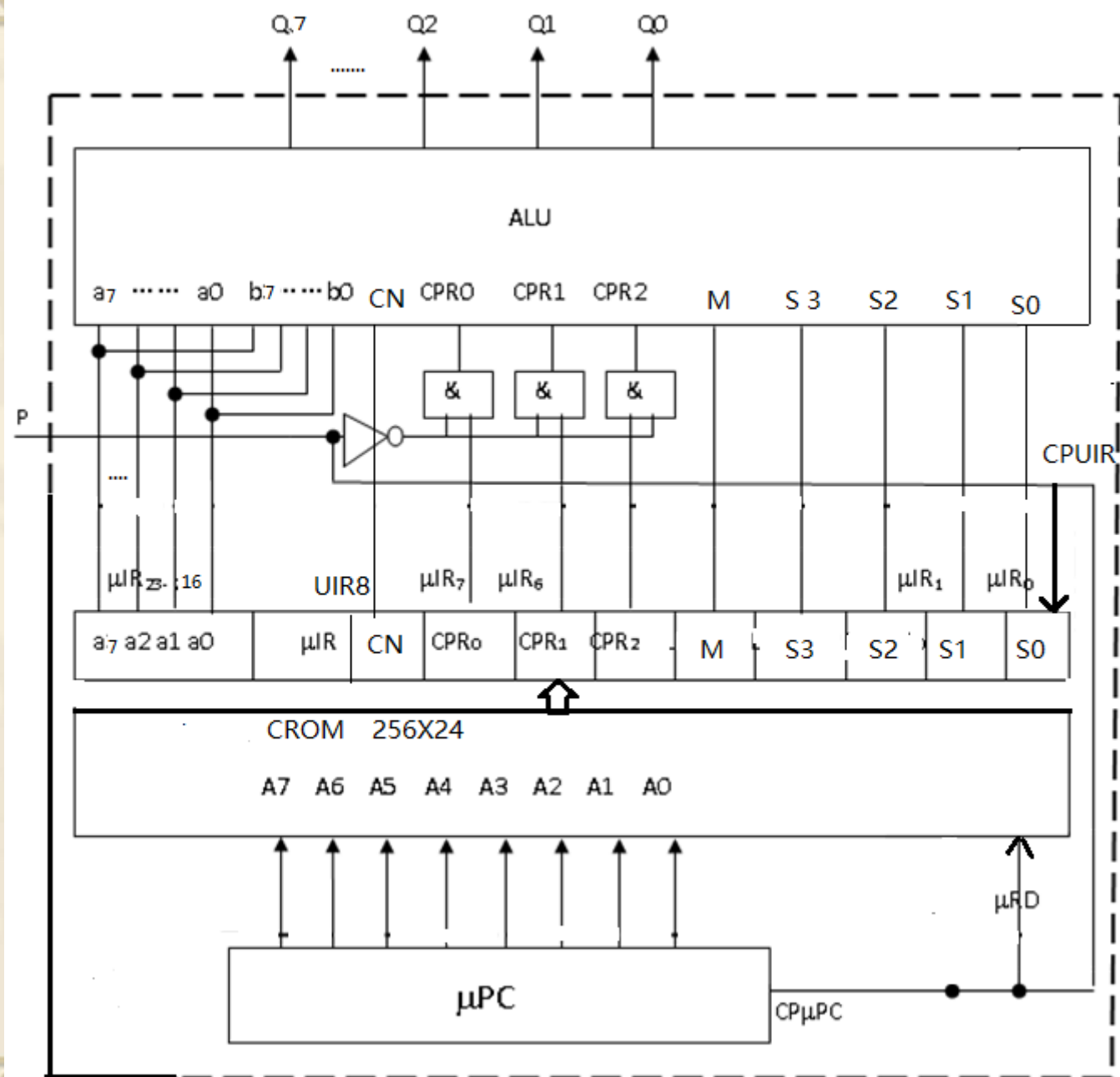
- ❧ 设计一个八位算法逻辑运算单元ALU
- ❧ 两操作数由八位寄存器 R_0 、 R_1 提供，其结果放入 R_2 中。具体何种操作可由微命令任意设定。

- ❖ R_0 、 R_1 、 R_2 均为D触发器组成的八位寄存器，在打入脉冲 CPR_i 的作用下，接收数据输入端提供的信息送入 R_i 中。

❖ 进位信号 C_0 、打入脉冲 CPR_0 、 CPR_1 、 CPR_2 、 M 、 S_0 、 S_1 、 S_2 、 S_3 均由微指令寄存器的 μIR_8 和 μIR_7 -- μIR_0 产生。



- ❖ 实现框图如下：
- ❖ 要求自己设计，并下载到FPGA中。



❖ (2) 微程序控制器的结构

- ❖ 设计控制存储器CROM, 256×24 位, 由 $\overline{\mu RD}$ 控制读出, 锁存于24位寄存器 μIR , 由 $CP_{\mu IR}$ 打入寄存器。
- ❖ 设计的微程序计数 μPC 向控制存储器提供8位微地址, 在控存读信号 的作用下, 读出一条长24位的微指令代码, 并在打入命令 $CP_{\mu IR}$ 的作用下, 送入 μIR 寄存器。

❖ 每当按一次脉冲键便产生一个负脉冲，该脉冲的作用是：

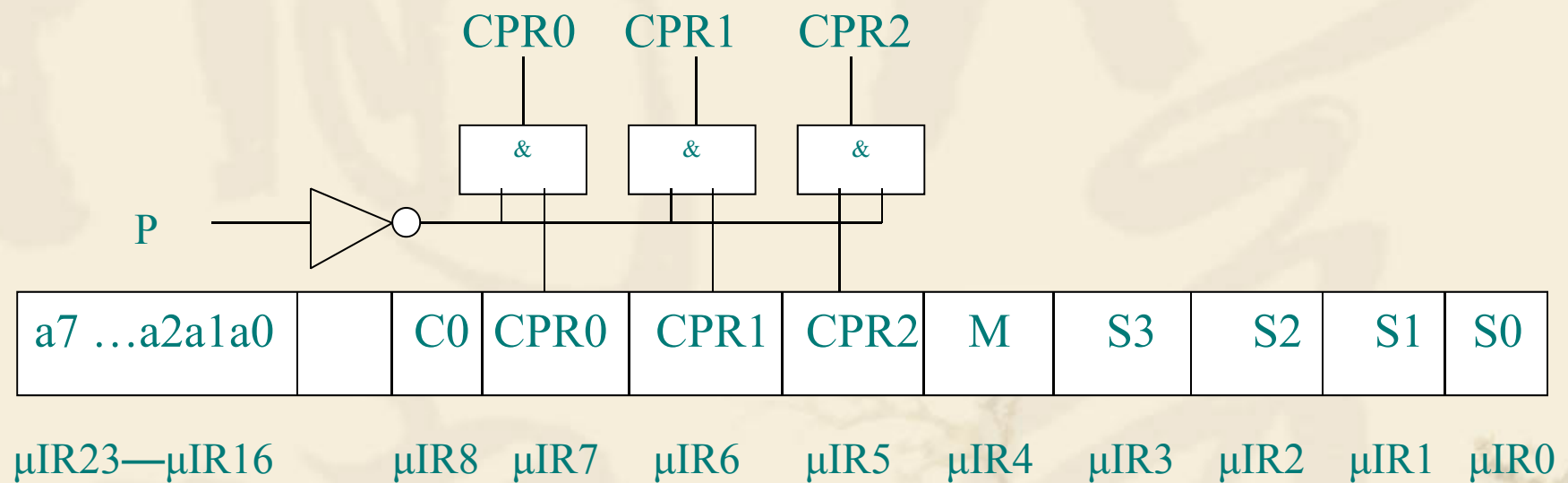
∞ 作为读控存的命令 $\overline{\mu RD}$ 。

∞ 负脉冲当作 $CP_{\mu IR}$ ，将读出的微指令代码打入 μIR

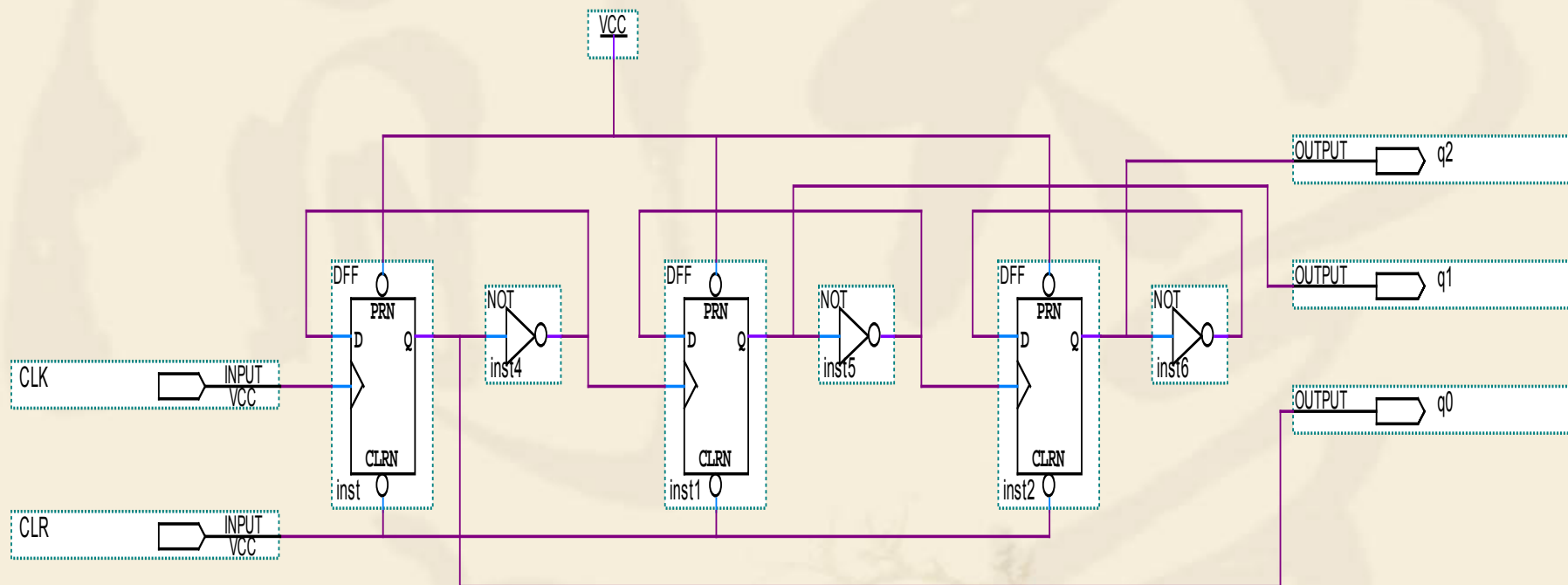
∞ 负脉冲的上升沿使 $\mu PC + 1$ 形成下一条微指令的地址。

∞ 负脉冲反相后的上升沿作为寄存器打入脉冲。

❖ 微指令可确定如下格式:



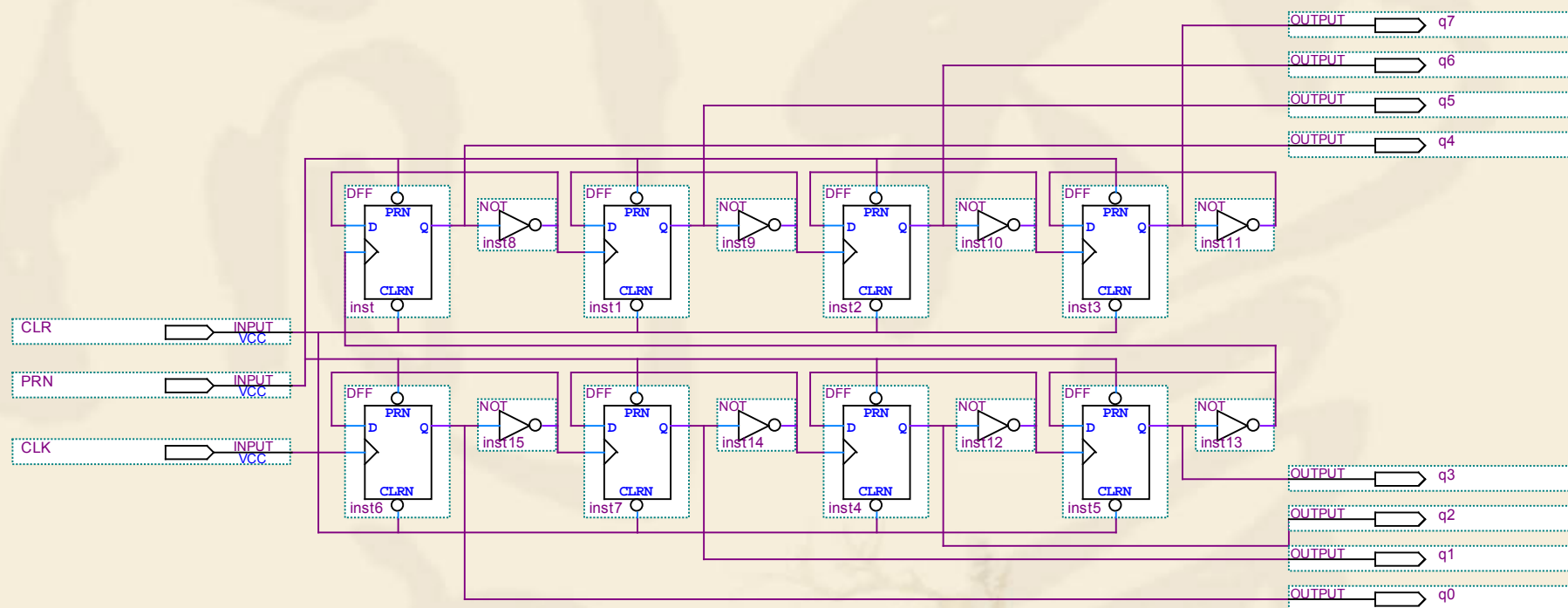
❖ 异步模8加1计数器的原理图(3位计数器)



❖ 功能:

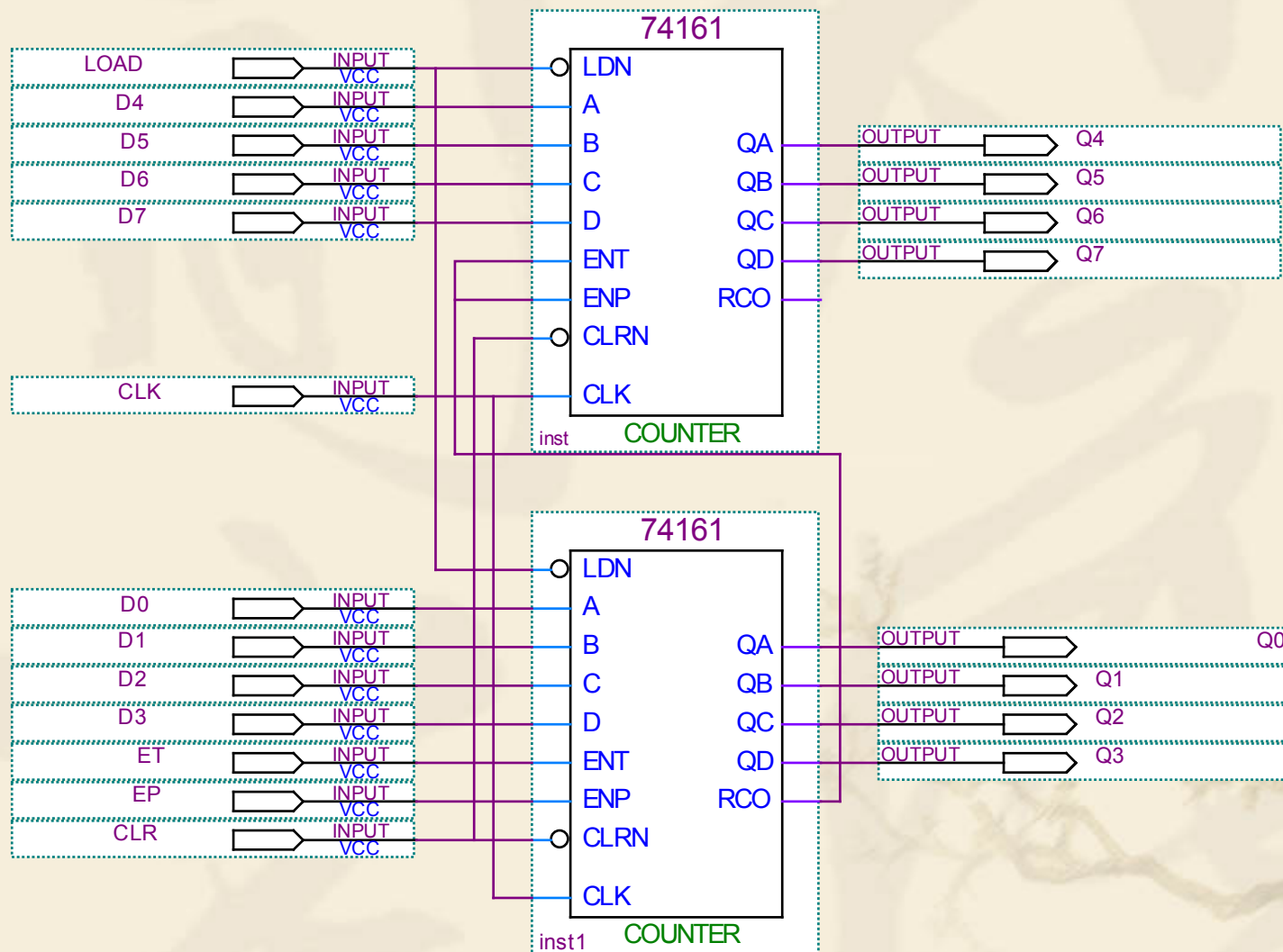
- ∞ CLR: 清零端,低电平有效,即若CLR接低电平时, $q_2q_1q_0=000$;
CLR接高电平时,对CLK增1计数,从000、001。。。111、000等。

❖ 上图扩展为异步模256加1计数器的原理图(8位计数器)



- ❖ 8位的增1计数器：对CLK增1计数。
- ❖ CLR：清零端，低电平有效；CLR=0时， $q7q6q5q4q3q2q1q0=00000000$ ；
- ❖ PRN：置位端，低电平有效；PRN=0时， $q7q6q5q4q3q2q1q0=11111111$ ；

❖ μ PC的另外一种设计方法（功能更强大）：



- ❖ 上图的功能说明：
- ❖ CLR：清零端，，低电平有效；CLR=0时， $Q_7Q_6Q_5Q_4Q_3Q_2Q_1Q_0=00000000$ ；
- ❖ LOAD：置数端，低电平有效；LOAD=0时，在CLK的上升沿， $Q_7Q_6Q_5Q_4Q_3Q_2Q_1Q_0=D_7D_6D_5D_4D_3D_2D_1D_0$ ；
- ❖ 当CLR=1，LOAD=1，ET=1，EP=1时，对CLK进行增1计数。
- ❖ 在整机设计时，使用该图作为 μ PC设计。

❖ (3) 调试步骤:

❖ 设计原理图并下载至FPGA中

❖ 编制微程序:

 ❧ 例如 $55 + AA \rightarrow R_2$

 ❧ 执行步骤:

 ❧ $55 \rightarrow R_0$: 01010101 00000000 10000000 即550080

 ❧ $AA \rightarrow R_1$: 10101010 00000000 01000000 即AA0040

 ❧ $R_0 + R_1 \rightarrow R_2$: 00000000 00000001 00101001 即000129

 ❧ $R_0 - R_1 \rightarrow R_2$: 00000000 00000000 00100110 即000026

❖ 按下表共写出13条微指令（每一步骤需一条微指令），把这13条微指令写入ROM中。

❖ 实验需求：2片74181(1片74182用或不用都可) 组成电路，需要测试的74181功能表：

M	$S_3S_2S_1S_0$	CN	操作
算术运算：0	1001	1	A加B
	0110	0	A减B
	0000	0	A加1
	1111	1	A减1
	1001	0	A加B加1
逻辑运算：1	1011	×	AB
	1110	×	A+B
	0110	×	A异或B
	1111	×	A直传
	1010	×	B直传
	0000	×	A逻辑非
	0101	×	B逻辑非

- ❖ 利用安装的计算机组成原理与系统结构软件把13条微指令以十六进制代码写入到控存ROM3#、ROM2#、ROM1#中，微指令从零号单元开始存放。
- ❖ 执行操作：
 - ∞ 按一次RET键将 μ PC置零。
 - ∞ 每按一次脉冲键读出并执行一条微指令。
 - ∞ 当微程序执行完时检查并记录结果

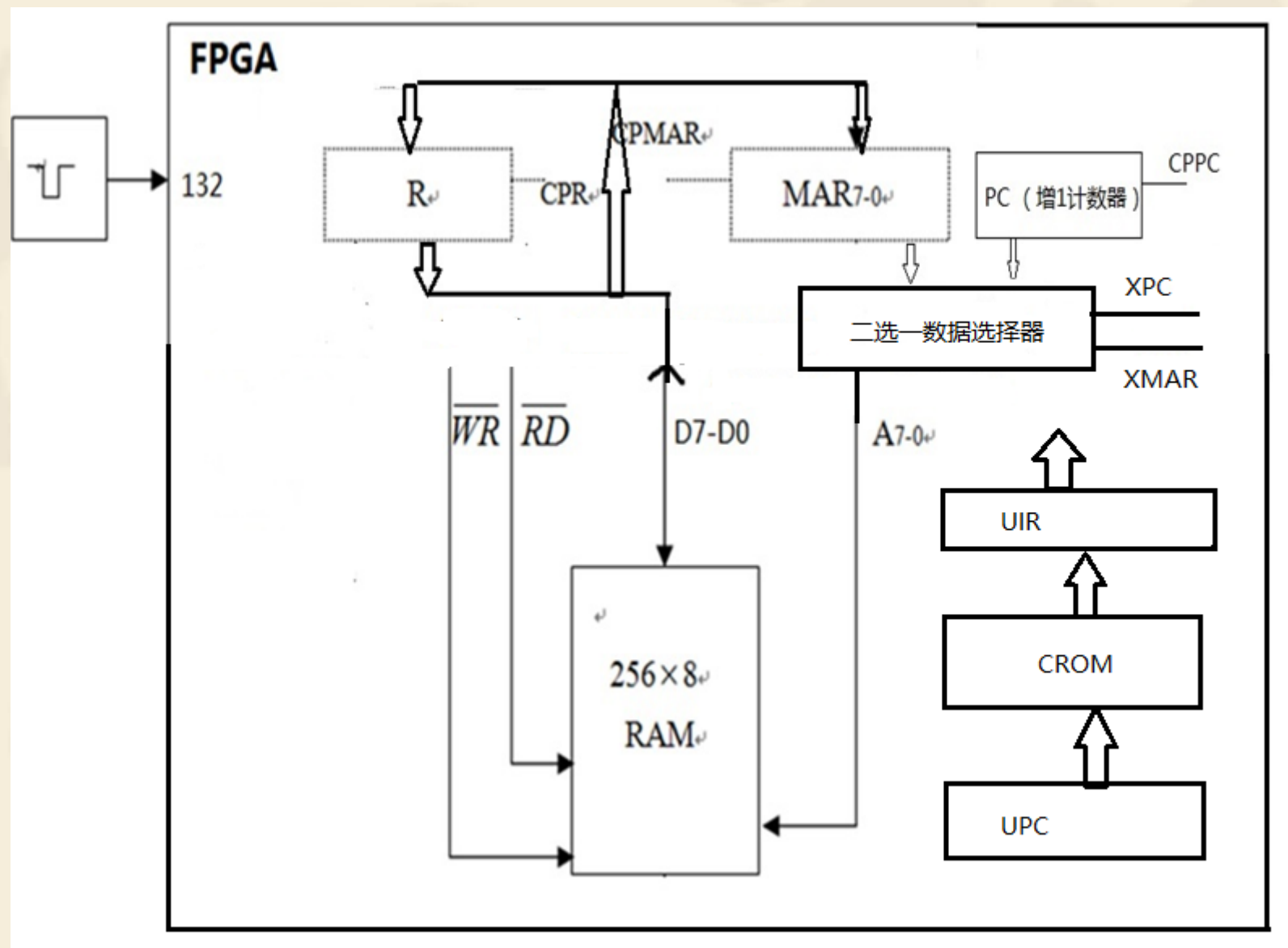
实验二 微程序控制的存储器读写系统设计

❖ 1、设计目的

- ❧ (1) 熟悉随机存储器读写系统结构设计。
- ❧ (2) 熟悉随机存储器的读写时序。
- ❧ (3) 熟悉随机存储器的读写操作的微程序实现。
- ❧ (4) 熟悉随机存储器的功能测试。

❖ 2、设计简述

- ❧ 要设计容量为 256×8 的随机存储器RAM和容量为 256×24 的控制存储器CROM。
- ❧ 在此基础上，学生只要设计相应的外围电路和时序就可以对随机存储器进行读写操作。
- ❧ 其结构框图如图所示。



- ❖ 要求:根据pc访问内存，取出地址Ad1，根据Ad1访问取出数据X，将X保存在Ad2地址单元。

地址	数据
0	Ad1
1	Ad2
...	
Ad1	X
...	
Ad2	

❖ (1) 结构及信号索引

∞ 图中MAR为地址寄存器（前面已设计REG）。



∞ R寄存器必须为具有三态输出的寄存器，直接调用74374即可。

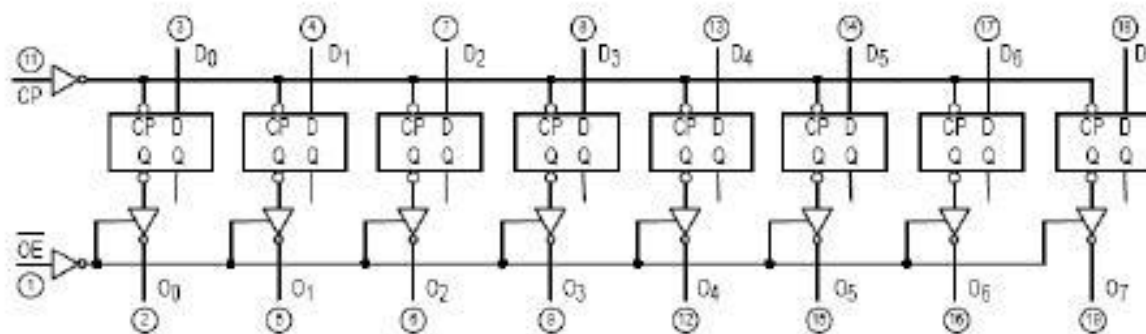
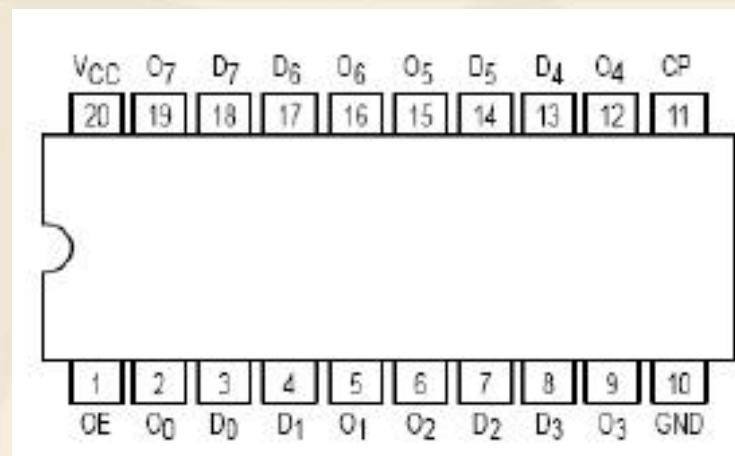
∞ PC为带清零端的8位增1计数器（前面已设计μPC）。

❖ 图中所需的信号如XPC、XMAR、CPR、CPMAR等由微程序提供。

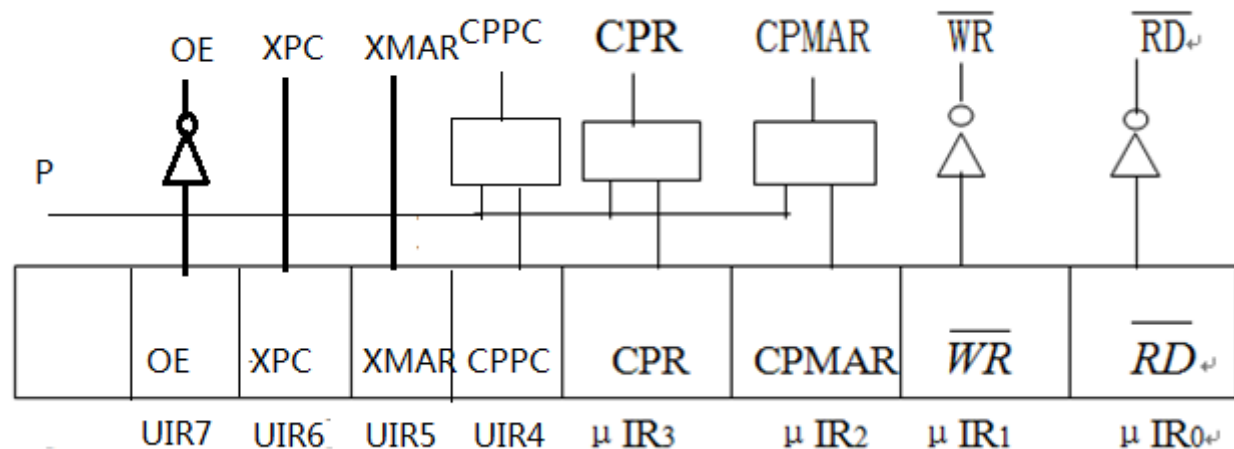
- ❖ **74ls374**引脚图及功能真值表
- ❖ 引出端符号:
- ❖ D0~D7 数据输入端
- ❖ OE 三态允许控制端（低电平有效）
- ❖ CP 时钟输入端
- ❖ O0~O7 输出端
- ❖ 74ls374引脚图:

LS374

D _n	LE	OE	O _n
H		L	H
L		L	L
X	X	H	Z ^a



(2) 微指令格式及微程序编制



- ❖ μIR_3 为1产生CPR； μIR_2 为1产生CPMAR； μIR_4 为1产生CPPC；
- ❖ μIR_1 为1产生 \overline{WR} ； μIR_0 为1产生 \overline{RD} ； μIR_5 为1产生XMAR选择信号； μIR_6 为1产生XPC选择信号； μIR_7 为1产生寄存器R的三态允许信号OE；
- ❖ 因为利用微程序，还必须设计微地址增1计数器 μPC 。
- ❖ 微程序的编制，由存储器的操作确定：
- ❖ 1) 读出地址ad1放入MAR寄存器：
- ❖ XPC、RD、CPMAR有效，即微指令：000045（16进制数）

- ❖ 2) M (ad1) 即 x 送入寄存器R:
- ❖ XMAR、RD、CPR、CPPC (PC增1) 即微指令: 000039H
- ❖ 3) 读出ad2送入MAR, 即微指令: 000045 (16进制数)
- ❖ 4) x 送入地址ad2中
- ❖ XMAR、WR、OE有效, 即微指令: 0000A2H
- ❖ 共4条微指令。
- ❖ (3) 调试步骤
- ❖ 设计存储器外围逻辑并下载至FPGA中。
- ❖ 设计微程序并将微程序以十六进制代码键入ROM3#、ROM2#、ROM1#中
- ❖ 按一次脉冲键读出并执行一条微指令, 并检查微指令执行情况, 直至结束。
- ❖ 在RAM中, 预先写入ad1、ad2、x具体数据。