



数字逻辑电路实验

实验七

(可编程小系统设计)

guyujun@seu.edu.cn

实验目的



- ◆综合前面所学的各项内容
- ◆了解掌握数字系统设计的流程和方法
- ◆掌握原理图输入方式设计数字系统的方法和流程
- ◆掌握复杂电路连接和调试技能





自顶向下法步骤

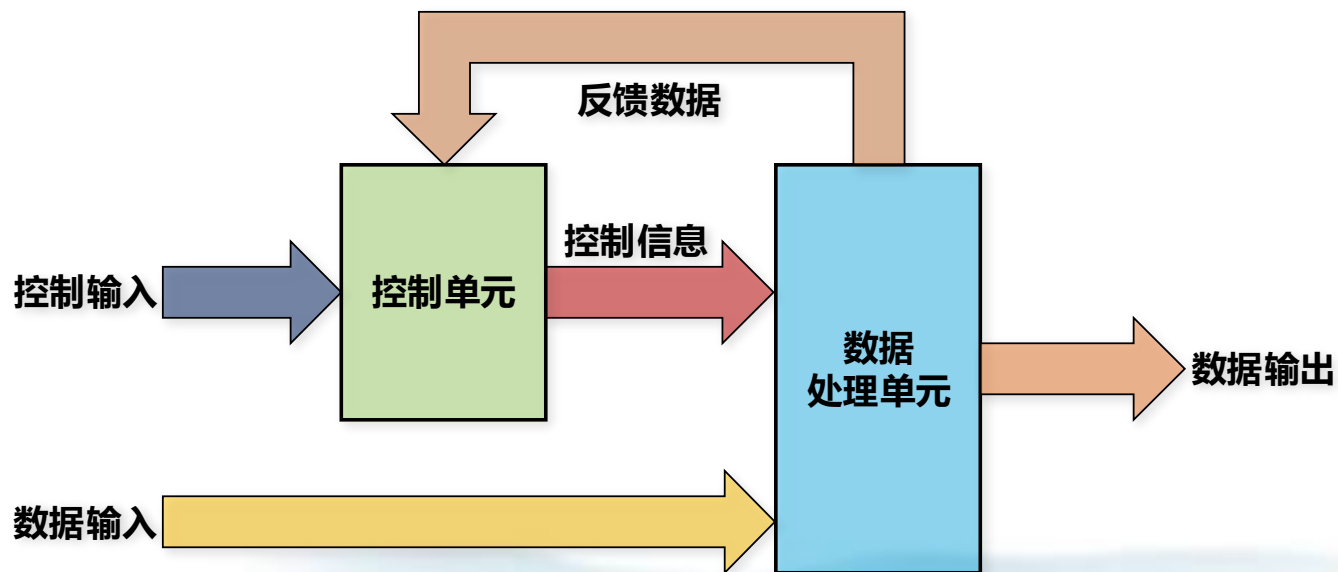
- 设计数据处理单元
 - 根据硬件算法确定数据处理单元的基本运算和操作
 - 简单、明晰、易控制
- 设计控制单元
 - 根据数据处理单元的操作和操作顺序，确定控制单元的逻辑功能
- 单元电路验证
- 系统总体调试验证
 - 将各单元电路连接起来，完成整个系统的总体设计验证



控制单元和数据处理单元之间的关系



- 数据处理单元在控制信号作用下对输入数据进行处理，并将相关信息反馈给控制单元
- 控制单元根据外部控制输入和数据处理单元反馈信息，决定数据处理单元数据操作顺序

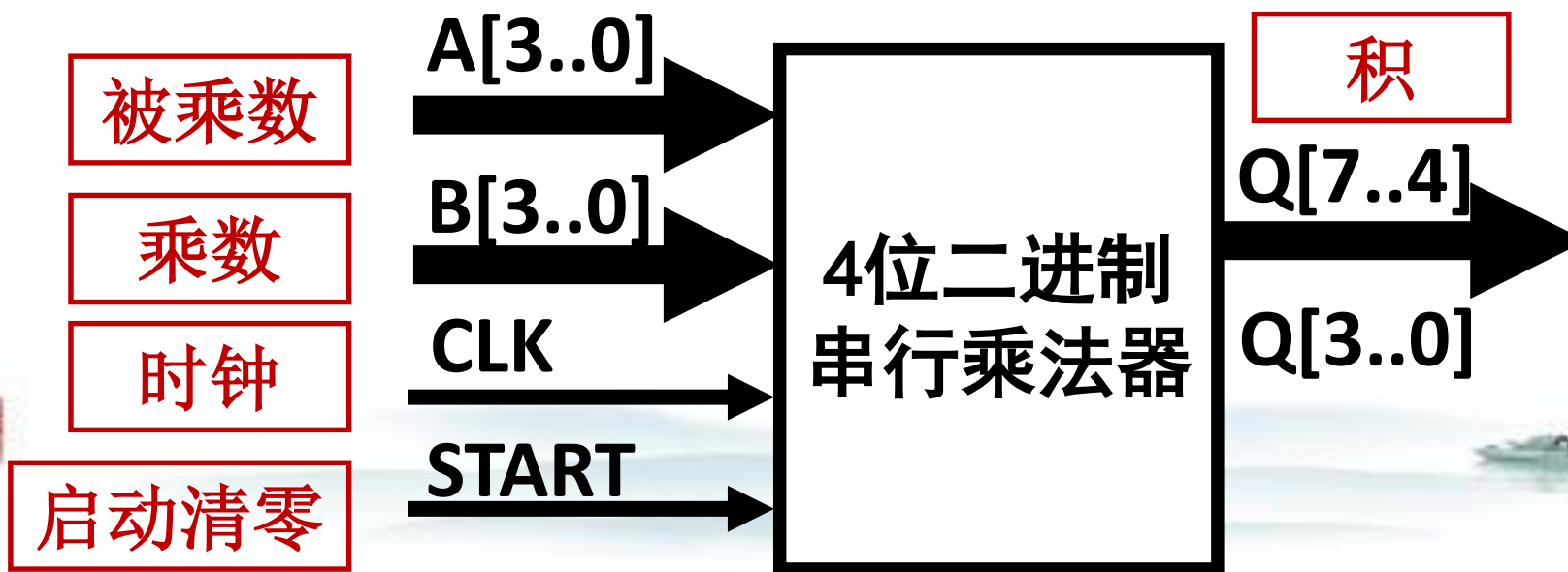


实验内容

二进制串行乘法器



利用实验六设计的全加器(或者74283)，设计一个4位二进制串行乘法器。其中“被乘数”及“乘数”均用4个逻辑电平开关输入，“积”用2个数码管显示（十六进制）。



设计要求



1. 使用4个逻辑电平开关输入“被乘数”；
2. 使用4个逻辑电平开关输入“乘数”；
3. 运行乘法器电路，2个数码管显示“积”（十六进制）；
4. 输入新的“被乘数”与“乘数”，重置电路得到新的“积”。



实现算法



二进制乘法可以采用移位相加的方法。即用乘数的各位数码，从低位开始依次与被乘数相乘，每相乘一次得到的积称为部分积，将第一次（由乘数最低位与被乘数相乘）得到的部分积右移一位并与第二次得到的部分积相加，将加得的和右移一位再与第三次得到的部分积相加，再将相加的结果右移一位与第四次得到的部分积相加……，直到所有的部分积都被加过一次。



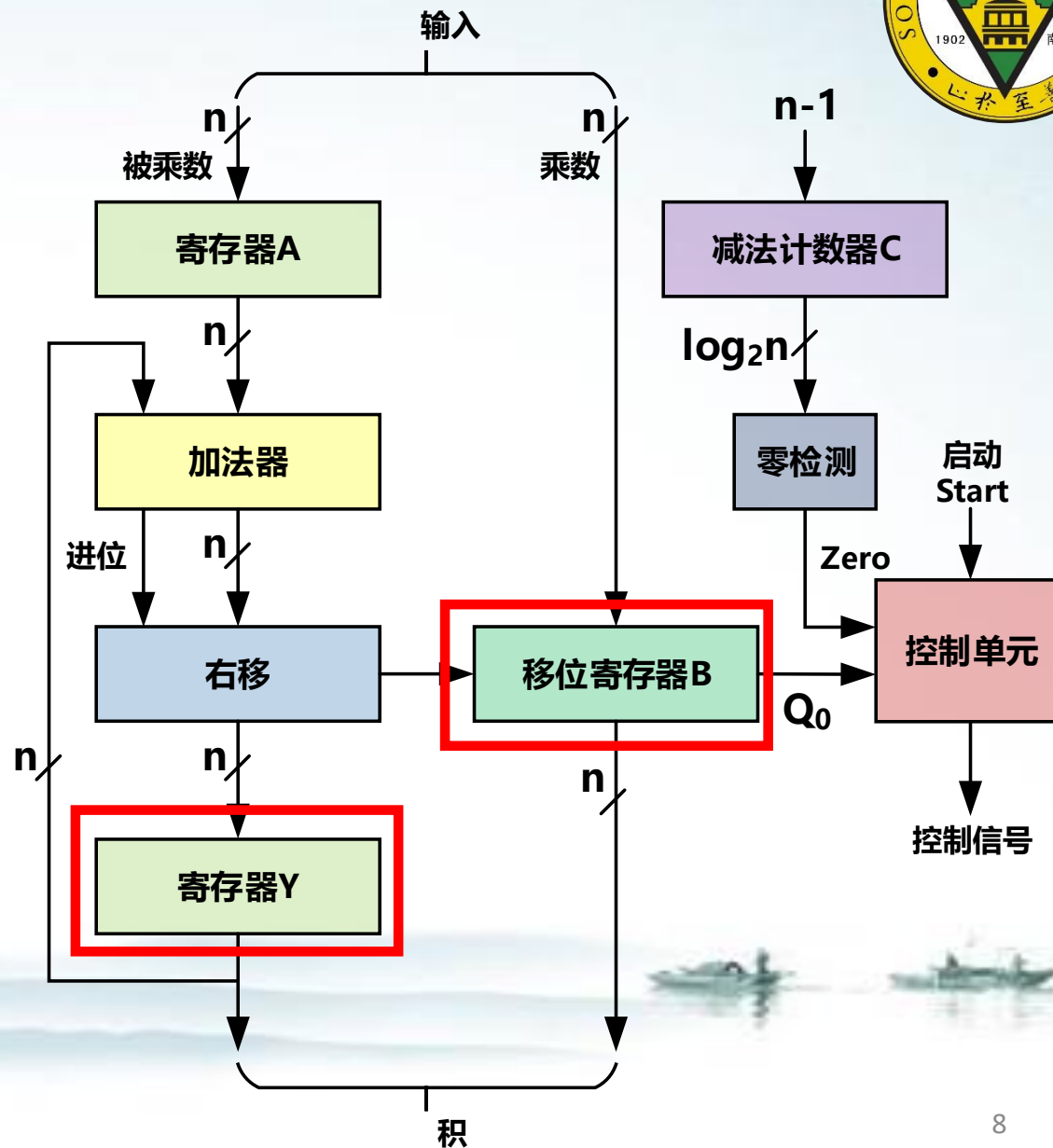
系统设计框图



1、 n 位被乘数从输入加载到寄存器A中， n 位乘数从输入加载到移位寄存器B中，减法计数器C初始化为 $n-1$ ，部分积寄存器Y初始化为0;

2、判断寄存器B最低位 Q_0 的值，如果值为“1”则部分积与被乘数A相加，加法器输出右移一位，如果值为“0”则加法器输出直接右移一位。其中移出的最低位移入寄存器B的最高位，其余保存在寄存器Y中。同时计数器C减1。

3、对乘数的每一位都执行2中的操作，直到计数器 $C=0$ ，结束全部操作，输出运算结果 $Y[n-1..0]B[n-1..0]$ 。



算法流程示例



以被乘数 $M_3M_2M_1M_0$ 和乘数 $N_3N_2N_1N_0$ 分别为1101和1001为例，其计算过程为：

1. 使用寄存器A、移位寄存器B分别存放被乘数M(1101)、乘数N(1001)；
2. 寄存器Y（初始0000）存放部分积，联合移位寄存器B输出最终积；
3. 依据 $B_0(1)$ ，将A(1101)与Y(0000)的值累加，将累加溢出(0)+累加结果(1101)集体右移一位更新给Y，被移出的最低位(1)送给 B_3 ，当前Y-B更新为01101100；
4. 依据新的 $B_0(0)$ ，确定此次A不参与累加，直接将当前的部分积Y(0110)右移一位更新给Y， Y_3 由溢出位补充，被溢出的最低位(0)送给 B_3 ，当前Y-B更新为00110110；
5. 重复步骤3、4，移位相加共计四次后，完成一次乘法运算；
6. 输出Y-B的最终值作为积结果。

算法流程示例

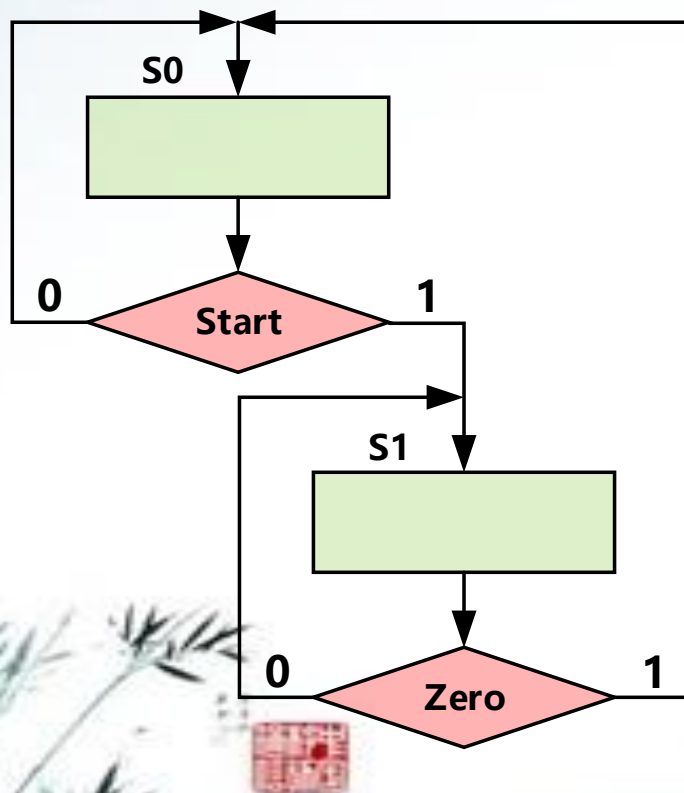
以1101*1001为例



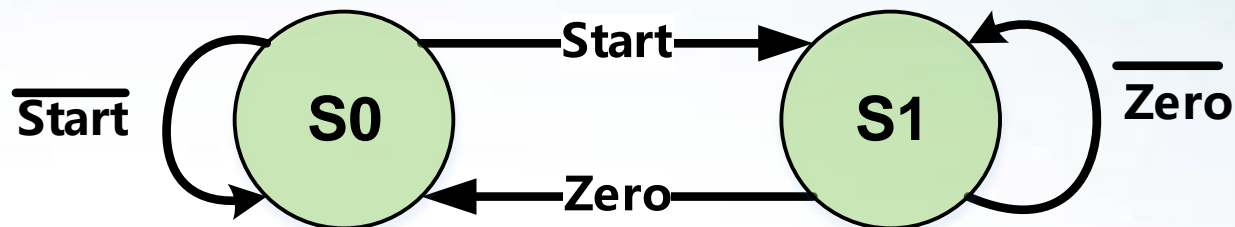
	累加器					4位寄存器Y				4位移位寄存器B			
序号	Cout	S3	S2	S1	S0	Y3	Y2	Y1	Y0	B2	B2	B1	B0
0	0	0	0	0	0	0	0	0	0	1	0	0	1
1	0	1	1	0	1	0	1	1	0	1	1	0	0
2	0	0	1	1	0	0	0	1	1	0	1	1	0
3	0	0	0	1	1	0	0	0	1	1	0	1	1
4	0	1	1	1	0	0	1	1	1	0	1	0	1

状态控制模块

ASM图



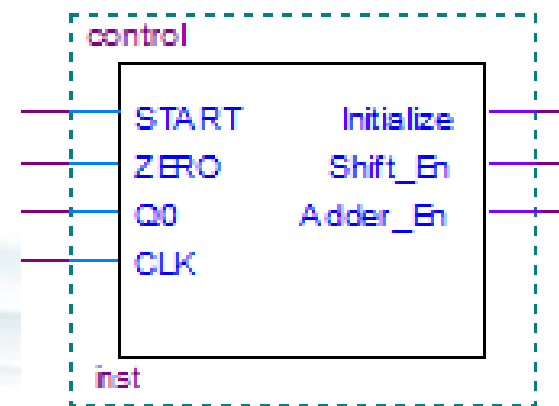
状态转移图



将算法流程图中的数据通路去除，即去掉所有条件输出，只保留控制通路，则Q0的两个出口路径指向了同一状态，多路判断框变成了单路判断框。

设计3个控制信号：

- 初始化信号 “Initialize”
- 计数和移位信号 “Shift_En”
- 被加数选择信号 “Adder_En”



状态控制模块

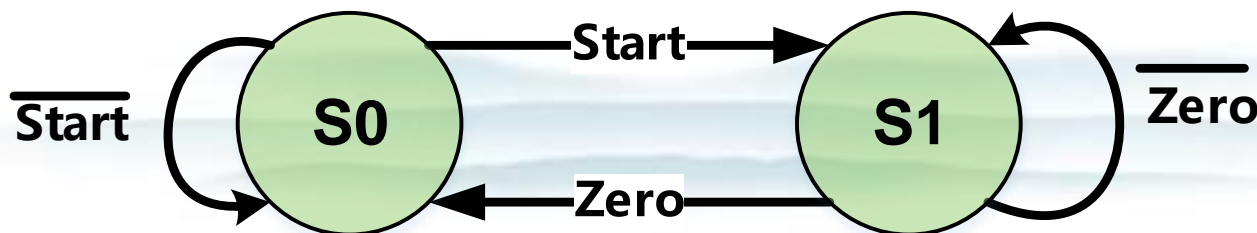
➤ 二进制编码实现

状态转移表:

现态	次态	输入	
		Start	Zero
S0	S0	0	x
	S1	1	x
S1	S0	x	1
	S1	x	0

用触发器实现，其状态方程：

$$D = \text{Start} \cdot \overline{Q} + \overline{\text{Zero}} \cdot Q$$





数据操作模块

- **寄存器A**: 具有置数功能4位寄存器, 可选4位锁存器、具有使能端的4位D触发器、计数器74161或移位寄存器74194实现
- **加法器**: 可选择前面课程设计的4位串行全加器, 也可以用中规模加法器74283
- **寄存器Y**: 具有置数功能4位寄存器, 可选4位锁存器、具有使能端的4位D触发器、计数器74161或移位寄存器74194实现
- **移位寄存器B**: 具有置数功能的4位移位寄存器, 可以选择4位双向移位寄存器74194实现
- **减法计数器C**: 可选择4位二进制加减计数器74191, 计数长度4



控制信号与数据执行元件输入信号的关系

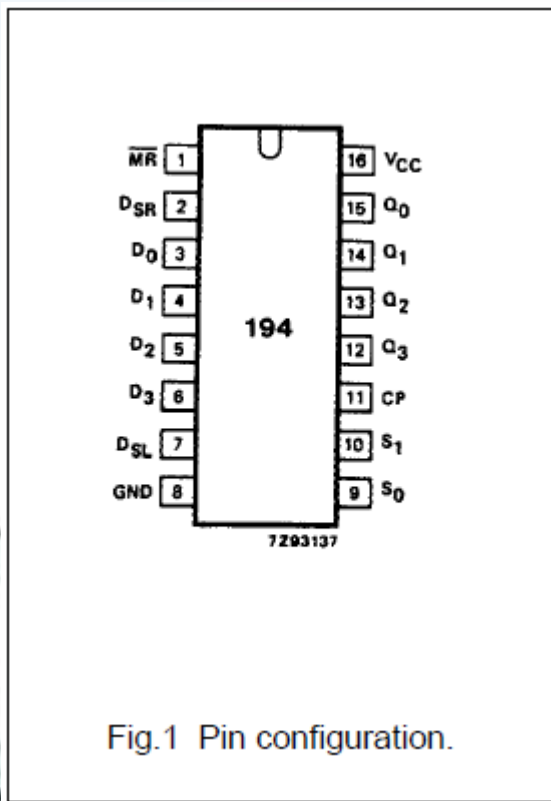
模块名	操作	控制信号	控制表达式	元件	元件输入信号
寄存器A	$A \leftarrow IN_A$	Initialize	$S0 \bullet Start$	74161	Load = $\overline{Initialize}$
寄存器Y	$Y \leftarrow 0$	Initialize	-	74161	Clear = $\overline{Initialize}$
	$Y \leftarrow (Cout, sum(3..1))$	Shift_En	S1		load = $\overline{Shift_En}$
寄存器B	$B \leftarrow IN_B$	Initialize	-	74194	$S_1 = Initialize \oplus Shift_En$ $S_0 = Initialize \bullet \overline{Shift_En}$
	$B \leftarrow sr\ B$	Shift_En	-		
计数器C	$C \leftarrow "11"$	Initialize	-	74191	Load = $\overline{Initialize}$
	$C \leftarrow C-1$	Shift_En	-		G = $\overline{Shift_En}$
全加器	$S \leftarrow A + Y$	Adder_En	$S1 \bullet Q_0$	74283	$A = Adder_En \bullet A_{in}$

器件选择



双向移位寄存器 74194

注意左移是向低位移动，右移是向高位移动



FUNCTION TABLE

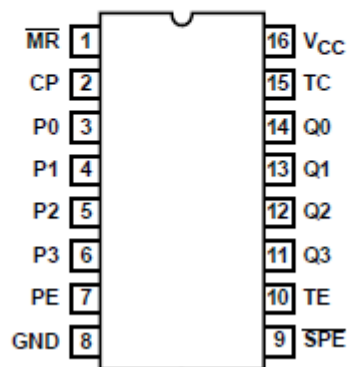
OPERATING MODES	INPUTS							OUTPUTS			
	CP	$\overline{\text{MR}}$	S_1	S_0	D_{SR}	D_{SL}	D_n	Q_0	Q_1	Q_2	Q_3
reset (clear)	X	L	X	X	X	X	X	L	L	L	L
hold ("do nothing")	X	H	L	L	X	X	X	q_0	q_1	q_2	q_3
shift left	\uparrow	H	h	L	X	L	X	q_1	q_2	q_3	L
	\uparrow	H	h	L	X	h	X	q_1	q_2	q_3	H
shift right	\uparrow	H	L	h	L	X	X	L	q_0	q_1	q_2
	\uparrow	H	L	h	h	X	X	H	q_0	q_1	q_2
parallel load	\uparrow	H	h	h	X	X	d_n	d_0	d_1	d_2	d_3

Notes

- H = HIGH voltage level
h = HIGH voltage level one set-up time prior to the LOW-to-HIGH CP transition
L = LOW voltage level
L = LOW voltage level one set-up time prior to the LOW-to-HIGH CP transition
 q, d = lower case letters indicate the state of the referenced input (or output) one set-up time prior to the LOW-to-HIGH CP transition
X = don't care
 \uparrow = LOW-to-HIGH CP transition

4位二进制计数器 74161

CD54HC161, CD54HCT161, CD54HC163, CD54HCT163
(CERDIP)
CD74HC161, CD74HCT161, CD74HC163, CD74HCT163
(PDIP, SOIC)
TOP VIEW



使用置数及计数功能

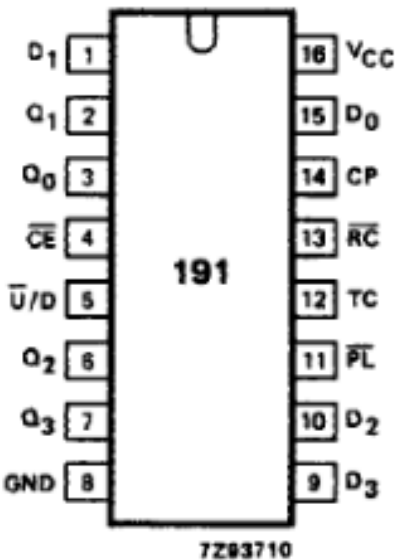
MODE SELECT - FUNCTION TABLE FOR 'HC161 AND 'HCT161

OPERATING MODE	INPUTS						OUTPUTS	
	\overline{MR}	CP	PE	TE	\overline{SPE}	P_n	Q_n	TC
Reset (Clear)	L	X	X	X	X	X	L	L
Parallel Load	H	\uparrow	X	X	L	L	L	L
	H	\uparrow	X	X	L	h	H	(Note 1)
Count	H	\uparrow	h	h	h (Note 3)	X	Count	(Note 1)
Inhibit	H	X	L (Note 2)	X	h (Note 3)	X	q_n	(Note 1)
	H	X	X	L (Note 2)	h (Note 3)	X	q_n	L



4位二进制正/倒数计数器 74191

可利用TC信号
实现倒计时结束的处理



FUNCTION TABLE

OPERATING MODE	INPUTS					OUTPUTS
	\overline{PL}	$\overline{U/D}$	\overline{CE}	CP	D_n	Q_n
parallel load	L	X	X	X	L	L
	L	X	X	X	H	H
count up	H	L	I	\uparrow	X	count up
count down	H	H	I	\uparrow	X	count down
hold (do nothing)	H	X	H	X	X	no change

TC AND RC FUNCTION TABLE

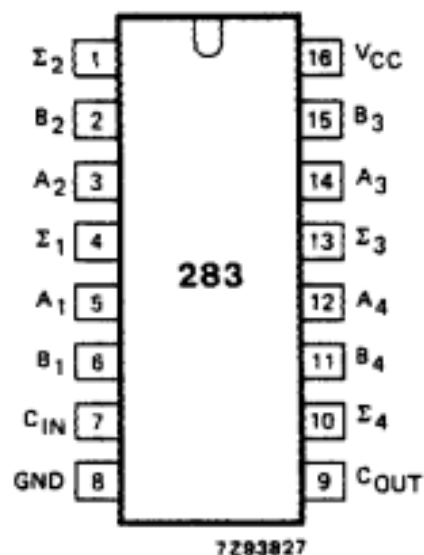
INPUTS			TERMINAL COUNT STATE				OUTPUTS	
$\overline{U/D}$	\overline{CE}	CP	Q_0	Q_1	Q_2	Q_3	TC	\overline{RC}
H	H	X	H	H	H	H	L	H
L	H	X	H	H	H	H	H	H
L	L		H	H	H	H		
L	H	X	L	L	L	L	L	H
H	H	X	L	L	L	L	H	H
H	L		L	L	L	L		

Notes

- 1. H = HIGH voltage level
L = LOW voltage level
I = LOW voltage level one set-up time prior to the LOW-to-HIGH CP transition
X = don't care
 \uparrow = LOW-to-HIGH CP transition
 = one LOW level pulse
 = TC goes LOW on a LOW-to-HIGH CP transition

4位二进制全加器 74283

亦可使用实验六的全加器



FUNCTION TABLE

PINS	C _{IN}	A ₁	A ₂	A ₃	A ₄	B ₁	B ₂	B ₃	B ₄	Σ ₁	Σ ₂	Σ ₃	Σ ₄	C _{OUT}	EXAMPLE ⁽²⁾
logic levels	L	L	H	L	H	H	L	L	H	H	H	L	L	H	
active HIGH	0	0	1	0	1	1	0	0	1	1	1	0	0	1	(3)
active LOW	1	1	0	1	0	0	1	1	0	0	0	1	1	0	(4)

Note

1. H = HIGH voltage level
L = LOW voltage level
2. **example**

```

1001
1010
-----
10011

```
3. for active HIGH, example = (9 + 10 = 19)
4. for active LOW, example = (carry + 6 + 5 = 12)

多样性方案

级联两个4bit移位寄存器，利用移位寄存器的置数/移位功能，完成N个状态的运算流程。



			8位寄存器Y-B								寄存器Y		寄存器B	
序号	RST	功能	Y3	Y2	Y1	Y0	B3	B2	B1	B0	S1	S0	S1	S0
0	0	清零	0	0	0	0	0	0	0	0	X	X	1	1
1	1	B置数	0	0	0	0	1	0	0	1	1	1	0	0
2	1	Y置数	1	1	0	1	1	0	0	1	1	0	1	0
3	1	移位	0	1	1	0	1	1	0	0	1	1	0	0
4	1	Y置数	0	1	1	0	1	1	0	0	1	0	1	0
5	1	移位	0	0	1	1	0	1	1	0	1	1	0	0
6	1	Y置数	0	0	1	1	0	1	1	0	1	0	1	0
7	1	移位	0	0	0	1	1	0	1	1	1	1	0	0
8	1	Y置数	1	1	1	0	1	0	1	1	1	0	1	0
9	1	移位	0	1	1	1	0	1	0	1	0	0	0	0

结果验证



测试用例1：运行 $1101*1001$ 移位相加过程中没有溢出；
测试用例2：运行 $1101*1101$ 移位相加过程中有溢出；
测试用例3：先运行 $1101*1001$ ，再运行 $1101*1101$ 查看结果切换是否正确。

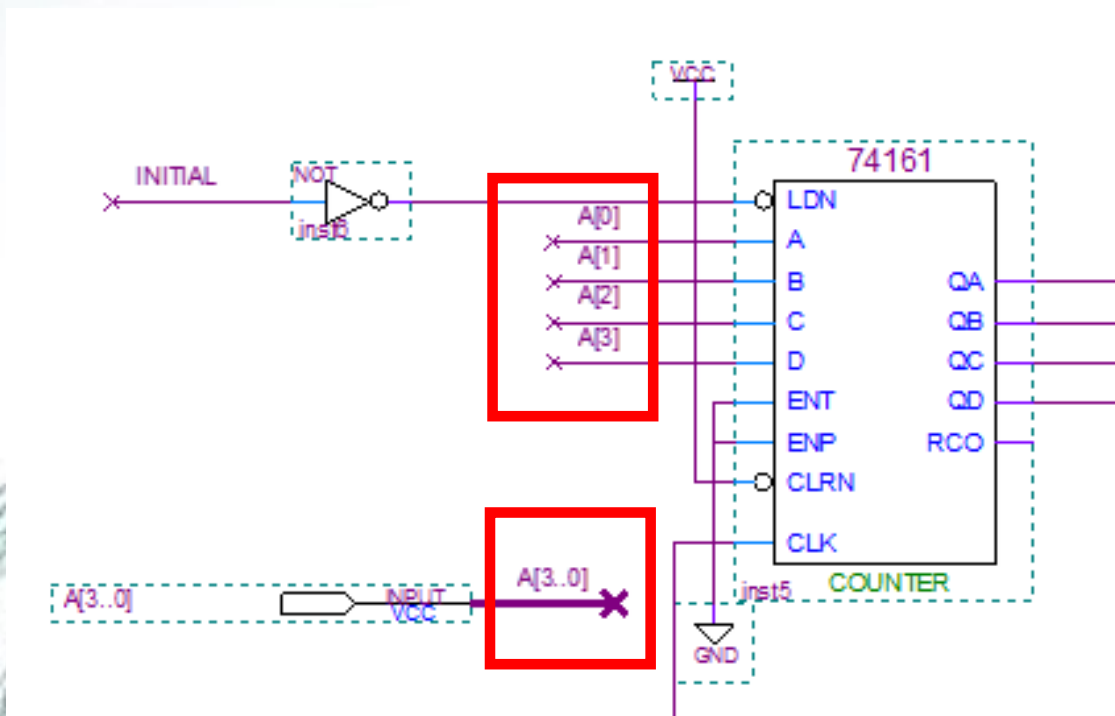
仿真波形使用总线形式展示（HEX），可增加若干个中间变量用于过程调试。



总线连接

为了便于调测：

- 多位数据采用总线连接的方式：Q[3..0]与Q[3], Q[2], Q[1], Q[0]电气相通；
- 定义名称相同的连接线，电气相通



几种提高设计效率的方法：
参考课本7.6小节



实验要求

- 实验预习：写出设计过程，画出逻辑电路图

- 验收要求：

- 1、各单元模块、整机联调仿真波形

- 2、四位二进制乘法结果仿真（HEX总线展示）

- ① $1101 \times 1001 = 75$ （累加过程无溢出）

- ② $1101 \times 1101 = A9$ （累加过程有溢出）

- 3、提高、自主发挥

- 实验时间：第12周

- 报告提交：第13周课内

用学号加下划线和题名作为
项目、文件夹名，例如：
JS319401_multiplier

发送

guyujun@seu.edu.cn