

一. 绪论:

1. BCD 码

- ① 二—十进制编码
- ② 用四位二进制数表示一位十进制数
- ③ 每位二进制数都有权值

2. 余 3 码

- ① 二—十进制编码
- ② 无权码
- ③ 可正确产生进位信号
有进位结果+3, 无进位结果-3
余 3 码是对 9 的自补码
0 与 9, 1 与 8 的余 3 码互为反码
方便十进制运算, 可将减法变加法

3. 格雷码

- ① 无权码
- ② 相邻两码之间只有一个 bit 不一样

格雷码的用途:

- (1) 卡诺图
- (2) 循环计数

格雷码的写法:

(1) 算法: $\text{Gray}[i] = \text{Bin}[i] \oplus \text{Bin}[i + 1]$

(2) 反射法:

0	0	0	0	0	0
1	0	1	0	0	1
	1	1	0	1	1
	1	0	0	1	0
			1	1	0
			1	1	1
			1	0	1
			1	0	0
	1bit	2bit	3bit		

(3) 图形法:

	0	1
0	0	1
1	3	2

	00	01	11	10
0	0	1	2	3
1	7	6	5	4

	00	01	11	10
00	0	1	2	3
01	7	6	5	4
11	8	9	A	B
10	F	E	D	C

二. 逻辑代数基础

1. 逻辑代数公式、定理和规则

(1) Distributive Law (分配律)

$$A \cdot (B+C) = AB+AC \quad A+BC=(A+B) \cdot (A+C)$$

(2) DeMorgan's Laws (德摩根定理)

$$(1) \overline{X_1 + X_2 + \dots + X_n} = \bar{X}_1 \bar{X}_2 \dots \bar{X}_n$$

$$(2) \overline{\bar{X}_1 \bar{X}_2 \dots \bar{X}_n} = X_1 + X_2 + \dots + X_n$$

懒得排版懒得写详细信息, 结合 PPT 食用风味更佳, 最后祝大家 98.5 (滑稽)

(3) 对偶规则:

或变与, 与变或 异或变同或, 同或变异或
优先级不改变

推论: ①两个逻辑表达式相等, 它们的对偶也相等

$$\textcircled{2} F = (F^D)^D$$

(4) 部分定律:

合并律 $AB + A\bar{B} = A$

吸收律 $A + AB = A$

消除律 $A + \bar{A}B = A + B$

蕴含律 $AB + \bar{A}C + BC = AB + \bar{A}C$

冗余律 $AB + \bar{A}C + BCD = AB + \bar{A}C$

包含律 $(A+B)(\bar{A}+C)(B+C) = (A+B)(\bar{A}+C)$

(5) 化简原则:

① 与项 (和项) 的个数最少

② 每个与项 (和项) 中变量的个数最少

(可以使用对偶规则推论②)

2. 逻辑函数表示方法

(1) 最小 (大) 项表达式:

Row No.	A B C	Minterms	Maxterms
0	0 0 0	$A'B'C' = m_0$	$A + B + C = M_0$
1	0 0 1	$A'B'C = m_1$	$A + B + C' = M_1$
2	0 1 0	$A'BC' = m_2$	$A + B' + C = M_2$
3	0 1 1	$A'BC = m_3$	$A + B' + C' = M_3$
4	1 0 0	$AB'C' = m_4$	$A' + B + C = M_4$
5	1 0 1	$AB'C = m_5$	$A' + B + C' = M_5$
6	1 1 0	$ABC' = m_6$	$A' + B' + C = M_6$
7	1 1 1	$ABC = m_7$	$A' + B' + C' = M_7$

性质:

$$\textcircled{1} m_i = \overline{M_i}$$

$$\textcircled{2} F(X_1, X_2, \dots, X_n) = \sum(m_a, m_b, \dots, m_i) = \prod(M_{2^n-m_a}, M_{2^n-m_b}, \dots, M_{2^n-m_i})$$

$$\bar{F}(X_1, X_2, \dots, X_n) = \sum(m_{2^n-m_a}, m_{2^n-m_b}, \dots, m_{2^n-m_i}) = \prod(M_a, M_b, \dots, M_i)$$

三. 组合逻辑电路的分析与设计

1. 三态门与 OC 门的应用 (略)

2. 常用组合逻辑芯片及其功能 (见附录)

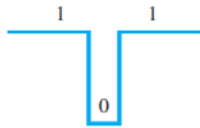
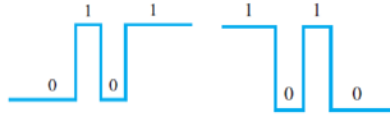
3. 竞争与险象

(1) 概念:

竞争: 多个信号经不同路径到达某一点有时间差, 称为竞争。☒

险象 (hazard): 由于竞争使得电路产生了暂时的错误输出, 称为险象。

(2) 险象的分类:

Static-1 Hazard Static-0 Hazard**Dynamic Hazards****(3) 险象的检查与消除:**

检查表达式中是否存在某个变量 x ,

- ①. 它同时以原变量和反变量的形式出现;
- ②. 在特定条件下简化成下面形式之一: $x+x'xx'$

步骤: 代数法 (略)

卡诺图法:

- ①. 检查是否有边相邻的卡诺圈
- ②. 在相邻边处添加卡诺圈, 逻辑表达式中添加冗余项

四. 触发器**1. 常用触发器的功能表与次态方程****(1) S-R 触发器**

S	R	Q_{n+1}
0	0	Q_n
1	0	1
0	1	0
1	1	X

$Q_n \rightarrow Q_{n+1}$	S	R
$0 \rightarrow 0$	0	X
$0 \rightarrow 1$	1	0
$1 \rightarrow 0$	0	1
$1 \rightarrow 1$	X	0

$$Q_{n+1} = S + \bar{R}Q_n$$

(2) J-K 触发器

J	K	Q_{n+1}
0	0	Q_n
1	0	1
0	1	0
1	1	Q_n'

$Q_n \rightarrow Q_{n+1}$	J	K
$0 \rightarrow 0$	0	X
$0 \rightarrow 1$	1	X
$1 \rightarrow 0$	X	1
$1 \rightarrow 1$	X	0

$$Q_{n+1} = J\bar{Q}_n + \bar{K}Q_n$$

(3) D 触发器

D	Q_{n+1}
0	0
1	1

$Q_n \rightarrow Q_{n+1}$	D
$0 \rightarrow 0$	0
$0 \rightarrow 1$	1
$1 \rightarrow 0$	0
$1 \rightarrow 1$	1

$$Q_{n+1} = D$$

(4) T 触发器

T	Q_{n+1}
0	Q_n
1	Q_n'

$Q_n \rightarrow Q_{n+1}$	T
$0 \rightarrow 0$	0
$0 \rightarrow 1$	1
$1 \rightarrow 0$	1
$1 \rightarrow 1$	0

$$Q_{n+1} = T\overline{Q_n} + \overline{T}Q_n$$

(5) T' 触发器

$$Q_{n+1} = \overline{Q_n}$$

2. 触发器功能转换

(1) 代数法：直接联立次态方程，注意 $RS=0$ 。

(2) 卡诺图法：列出现有触发器参数关于 Q_n 和目标触发器参数的卡诺图并化简，注意 $RS=0$ 。

3. 触发器转换公式：

(1) RS:

$$\textcircled{1} RS \rightarrow JK: S = J\overline{Q_n}, R = KQ_n$$

$$\textcircled{2} RS \rightarrow D: S = D, R = \overline{D}$$

$\textcircled{3} RS \rightarrow T:$

$$\textcircled{4} RS \rightarrow T': S = \overline{Q_n}, R = Q_n$$

(2) JK:

$$\textcircled{1} JK \rightarrow RS: J = S, K = \overline{S}R$$

$$\textcircled{2} JK \rightarrow D: J = D, K = \overline{D}$$

$$\textcircled{3} JK \rightarrow T: J = T, K = T$$

$$\textcircled{4} JK \rightarrow T': J = 1, K = 1$$

(3) D:

$$\textcircled{1} D \rightarrow RS: D = J\overline{Q_n} + \overline{K}Q_n$$

$$\textcircled{2} D \rightarrow JK: D = T \oplus Q_n$$

$$\textcircled{3} D \rightarrow T: D = \overline{Q_n}$$

$$\textcircled{4} D \rightarrow T': D = S + \overline{R}Q_n$$

(4) T:

$$\textcircled{1} T \rightarrow RS: T = S\overline{Q_n} + RQ_n$$

$$\textcircled{2} T \rightarrow JK: T = J\overline{Q_n} + KQ_n$$

$$\textcircled{3} T \rightarrow D: T = D \oplus Q_n$$

$$\textcircled{4} T \rightarrow T': T = 1$$

五. 时序逻辑电路

1. 常用时序逻辑芯片及其功能（见附录）

2. 时序逻辑分析的步骤

确定系统变量（输入变量、输出变量、状态变量）

① 列输入方程（驱动方程，控制函数）（触发器的输入方程）

懒得排版懒得写详细信息，结合 PPT 食用风味更佳，最后祝大家 98.5（滑稽）

- ② 列输出方程（输出函数）
- ③ 列次态方程（状态方程）
- ④ 列状态转换表
- ⑤ 画状态图
- ⑥ 画时序图（波形图）

3.时序逻辑设计的步骤（**UNIT12PPT P32-40, P48-51** 信息密度高，实用性强，建议打印！（本人此处偷懒了一下下，其余文字不再赘述））

1. 需求 → 原始状态图、状态表
2. 最小化原始状态表
3. 状态分配（编码）→ 状态转移表
4. 状态转移表 → 触发器驱动表（激励表）
5. 卡诺图化简 → 激励函数表达式(输入方程) 输出函数表达式(输出方程)
6. 电路实现
7. 检查无关状态

六．可编程逻辑器件（建议打印 ppt p12-18,28-30,47-55,78-80）

ROM/PROM—— n 输入 m 输出

- 存放包含 2^n 个 words（字）的 array（数组），每个字长为 m bits.
- 存放 n 输入、 m 输出的组合逻辑功能的真值表
- 每个输入组合（地址）选中一个存储单元，给定输入，输出为相应的 word.
- PROM 的或阵列可编程

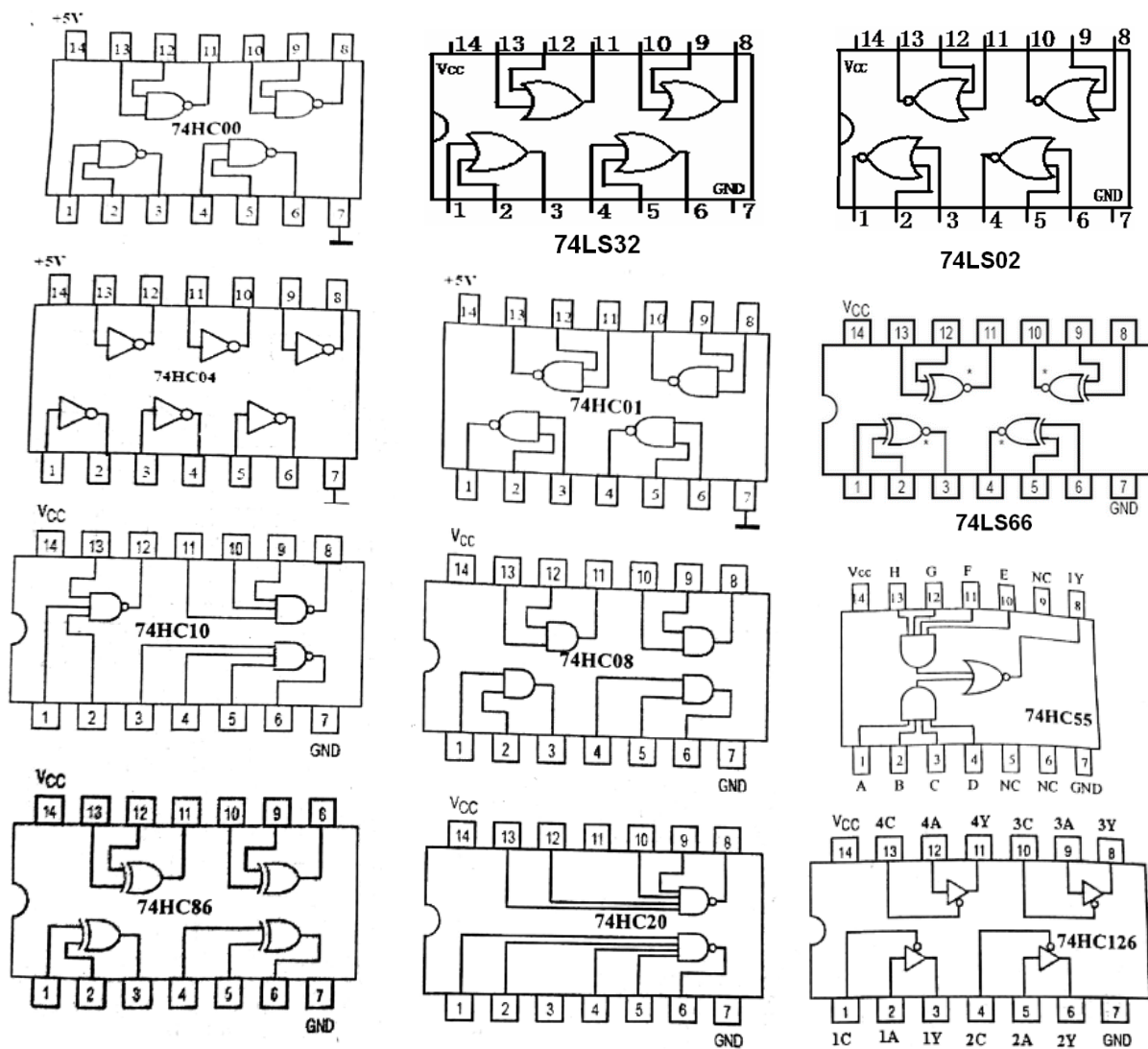
PLA ——将内容相同的存储单元用一个存储单元来代替（令几个地址码读出同一存储单元的内容）

- 与阵和或阵都可编程，每条“字”线不一定是最小项，且字数少于或等于 2^n
- 地址和字之间没有一一对应关系，一个地址可能同时访问两个或两个以上的字
- 必须对表达式化简，即矩阵中存储的是化简后的内容，与真值表不再一一对应
- 有的 PLA 包含记忆元件（触发器网络），既能实现组合逻辑又能实现时序逻辑。
- 在使用 PLA 时需考虑变量复用

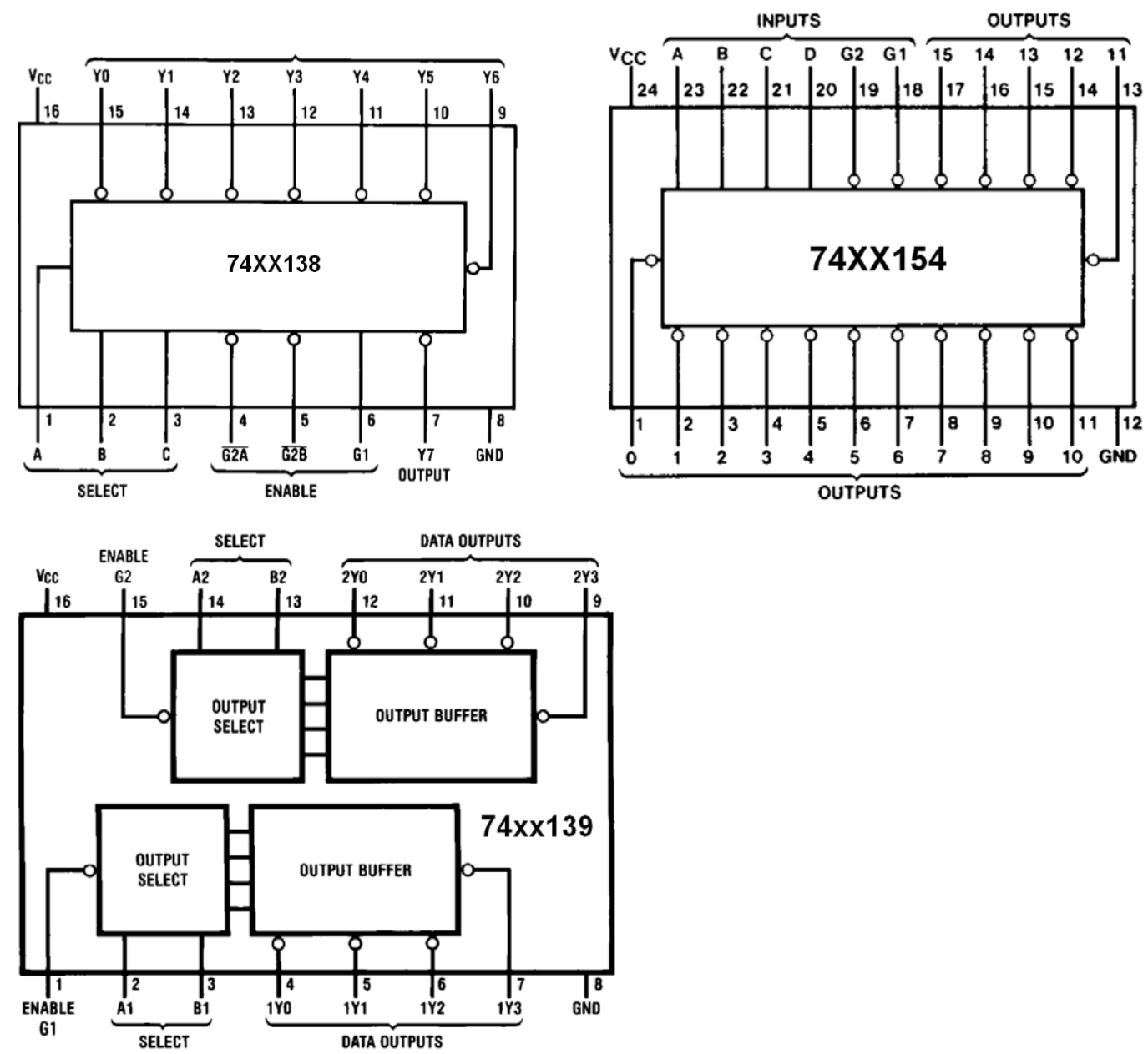
PAL——

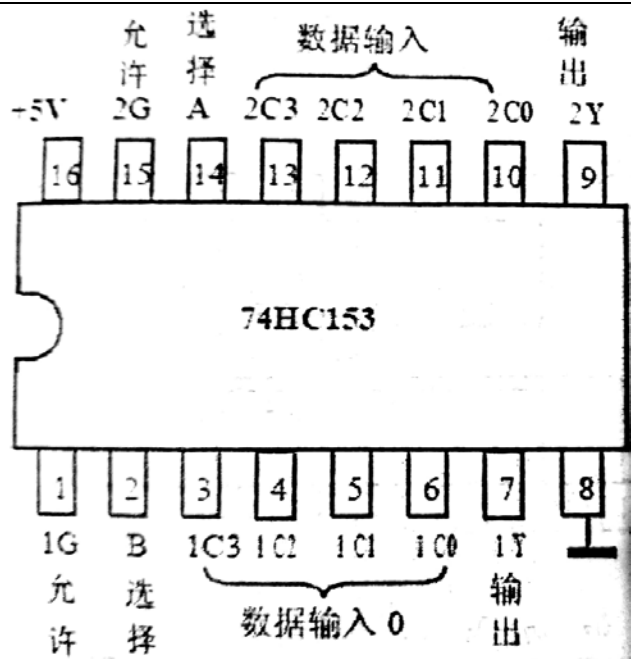
- 与阵列可编程
- 需根据输出端所连变量个数决定与阵列
- 需要化简表达式

附录：
逻辑门：



译码器:

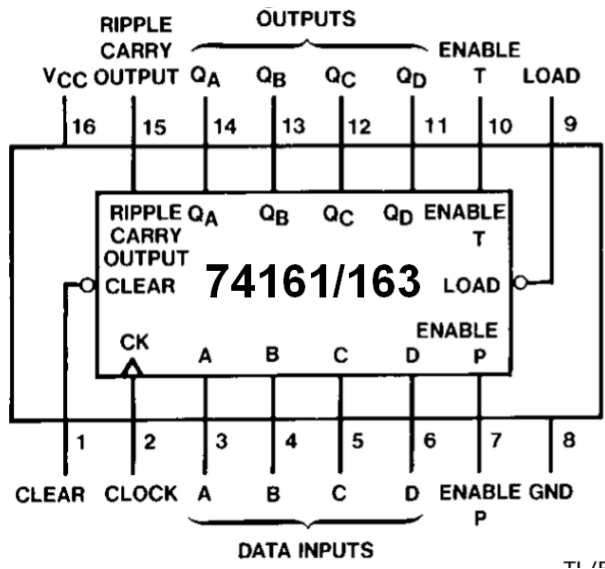
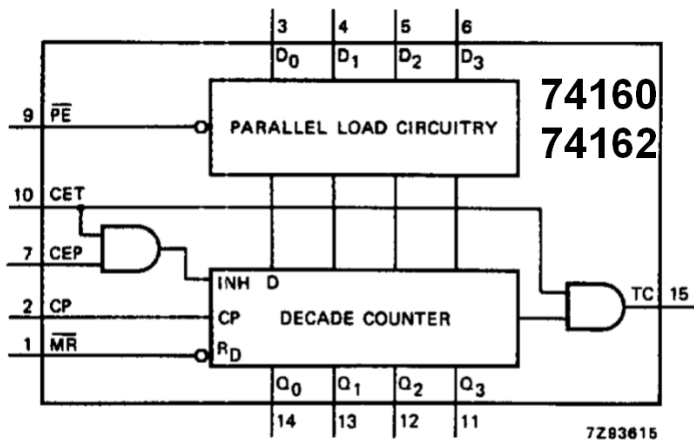




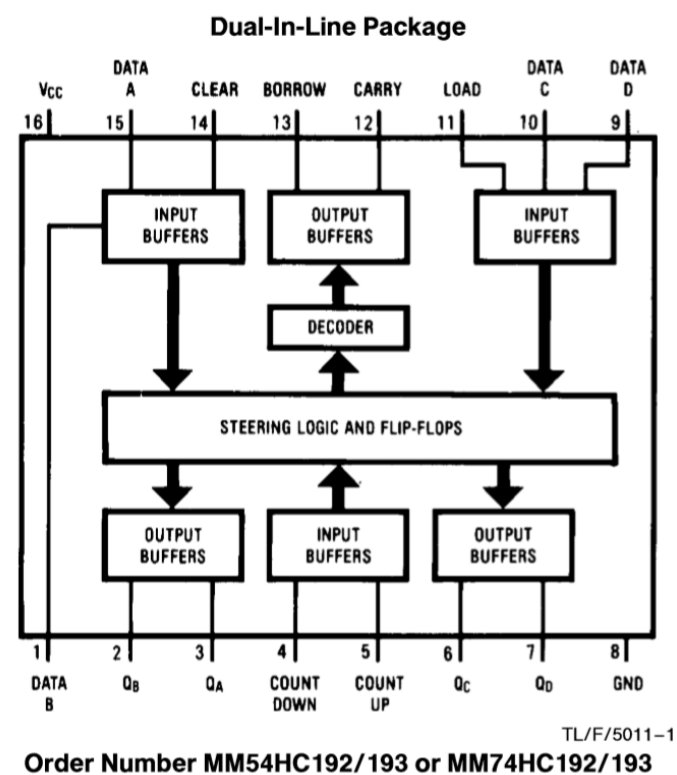
数据选择器:

计数器:

芯片型号	功能描述	置数方式	清零方式
74160	模十加法计数器	同步	异步
74161	模十六加法计数器	同步	异步
74162	模十加法计数器	同步	同步
74163	模十六加法计数器	同步	同步
74193	模十六可逆计数器	异步	异步
7490	模二/五/十加法计数器	异步	异步



Connection Diagram



Order Number MM54HC192/193 or MM74HC192/193

Truth Table

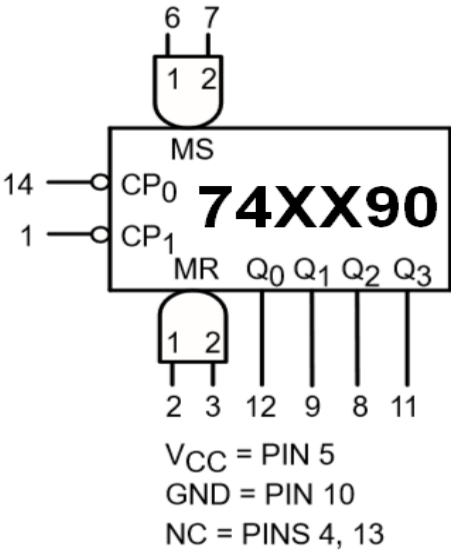
Count		Clear	Load	Function
Up	Down			
↑	H	L	H	Count Up
H	↑	L	H	Count Down
X	X	H	X	Clear
X	X	L	L	Load

H = high level
L = low level
↑ = transition from low-to-high
X = don't care

74193

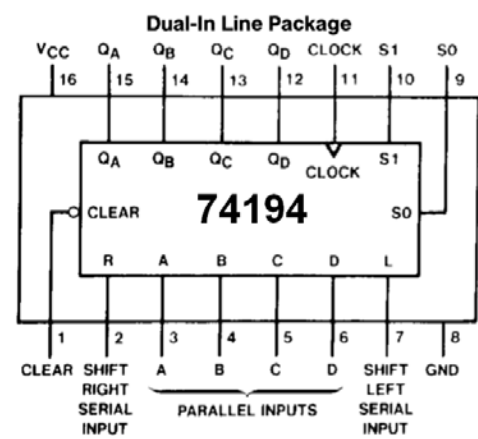
LS90

MODE SELECTION



RESET/SET INPUTS				OUTPUTS			
MR ₁	MR ₂	MS ₁	MS ₂	Q ₀	Q ₁	Q ₂	Q ₃
H	H	L	X	L	L	L	L
H	H	X	L	L	L	L	L
X	X	H	H	H	L	L	H
L	X	L	X	Count			
X	L	X	L				
L	X	X	L				
X	L	L	X				

移位寄存器:



Function Table

Inputs						Outputs			
Clear	Mode		Clock	Serial		Parallel			
	S1	S2		Left	Right	A	B	C	D
L	X	X	X	X	X	X	X	X	X
H	X	X	L	X	X	X	X	X	X
H	H	H	↑	X	X	a	b	c	d
H	L	H	↑	X	H	X	X	X	X
H	L	H	↑	X	L	X	X	X	X
H	H	L	↑	H	X	X	X	X	X
H	H	L	↑	L	X	X	X	X	X
H	L	L	X	X	X	X	X	X	X

锁存器与触发器:

