

# 第八章 数字系统设计基础

## 8.1 数字系统概述

## 8.2 ASM图表

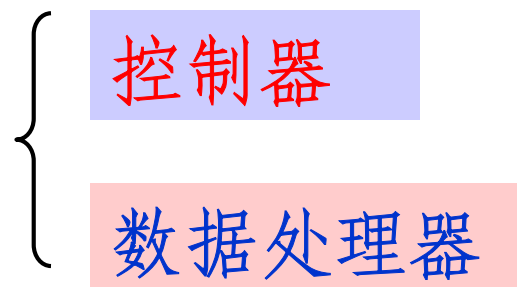
## 8.3 数字系统设计

# 第八章 数字系统设计基础

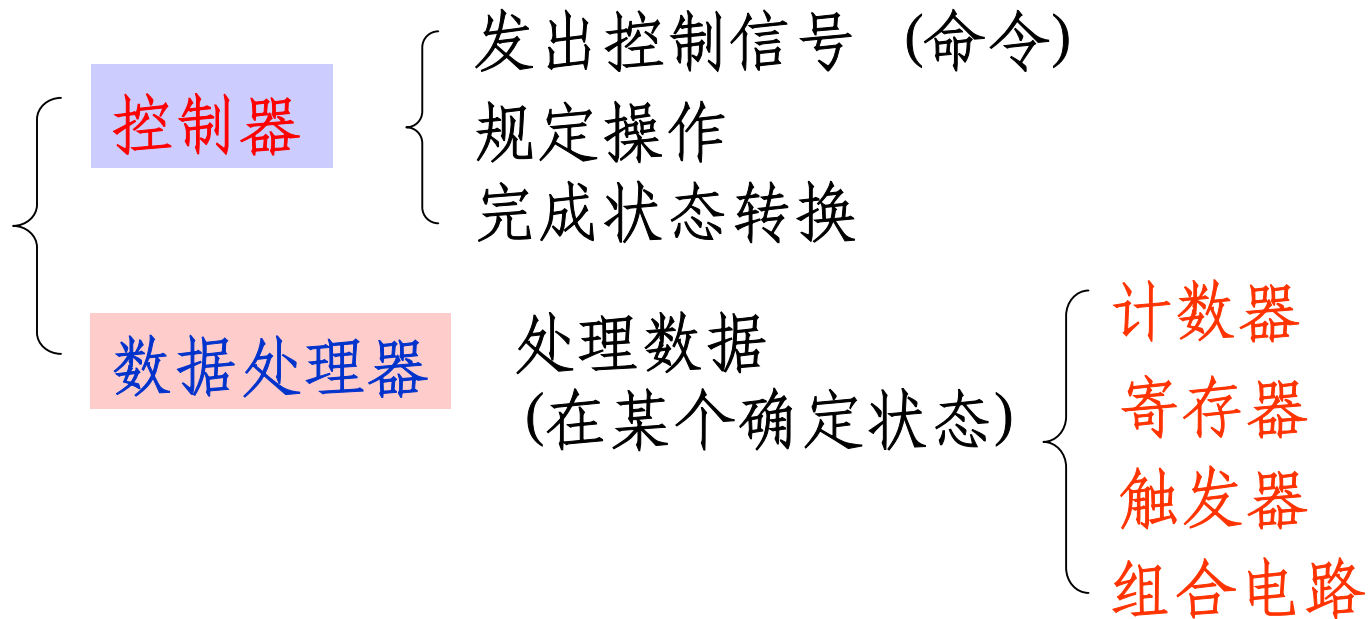
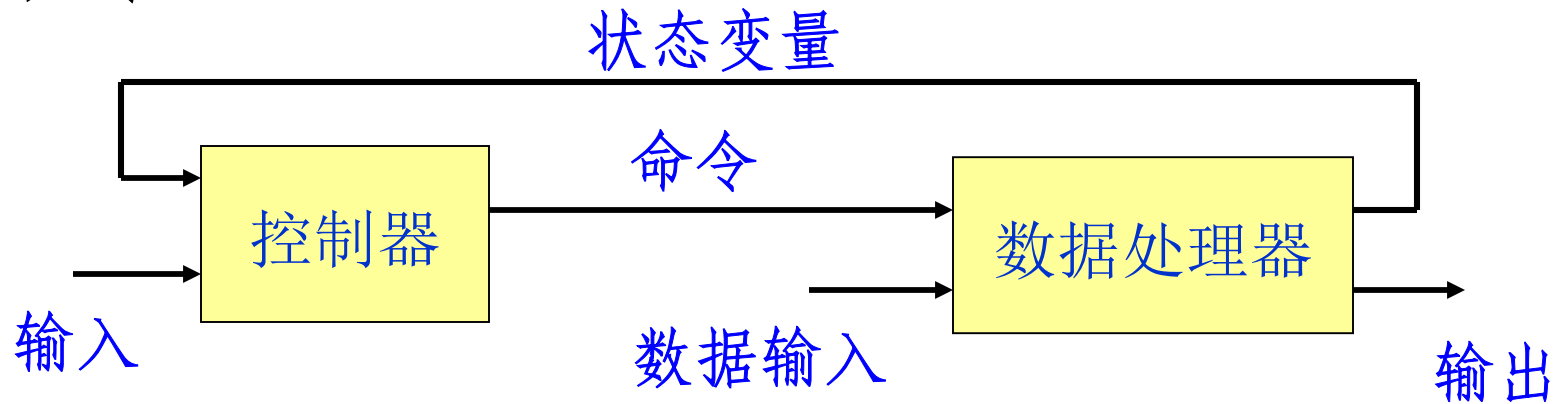
## § 8.1 数字系统概述

一个数字系统  $\Longrightarrow$  模块 子系统

数字系统



关系：



结果作为状态变量反馈给控制器

## § 8.2 算法状态机 — ASM 图表

**ASM :**

数字系统控制过程的算法流程图

与通常算法流程图不同, ASM 图表示了准确的时间序列

特性:

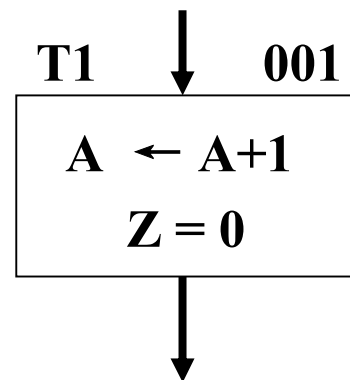
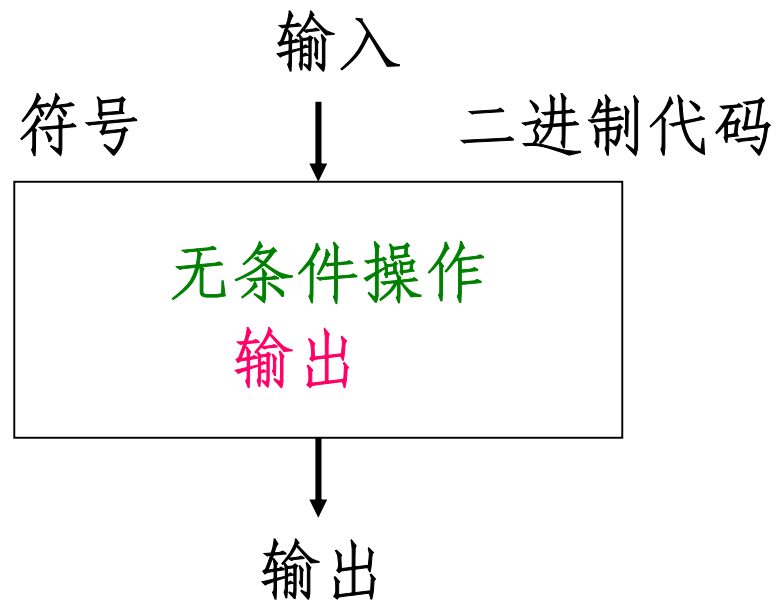
1. 操作按时间序列进行
2. 操作取决于某一判断 (外输入及反馈信号)

## 8.2.1 ASM图表符号

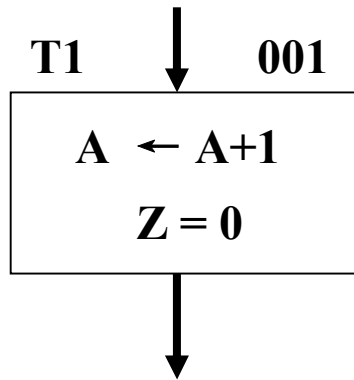
三种基本符号

{ 状态框  
判断框  
条件框

### 1. 状态框 (矩形)



在 T1 状态,  
代码 001,  
输出  $Z = 0$ ,  
下一个 *CLK* 到来,  
数据处理器实现无  
条件操作  $A + 1$ .



$A \leftarrow A+1$

在状态框中的操作是无条件操作，是此状态下将要实现的操作，将会在下一个***CLK***来临时完成。

寄存器传输语言

寄存器  
广义

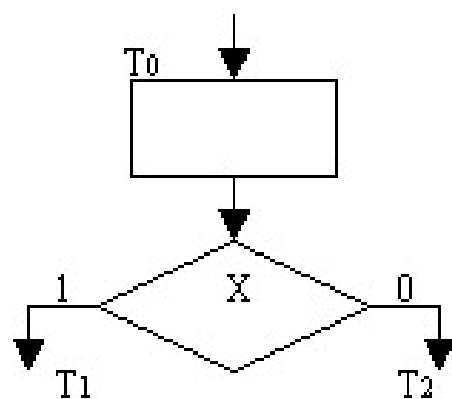
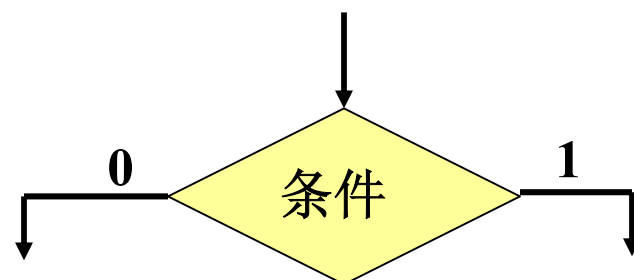
{ 寄存器  
计数器  
触发器  
存储器

$R \leftarrow SR$  (**R 向右转换**)

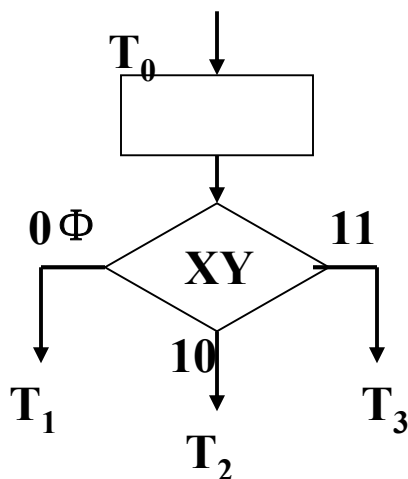
$A \leftarrow 0$  (**A 清零**)

$F \leftarrow 1$  (**F 置1**)

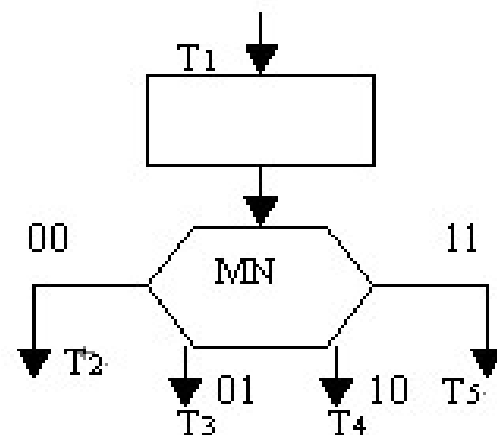
## 2. 判断框 (菱形)



2 输出



3 输出



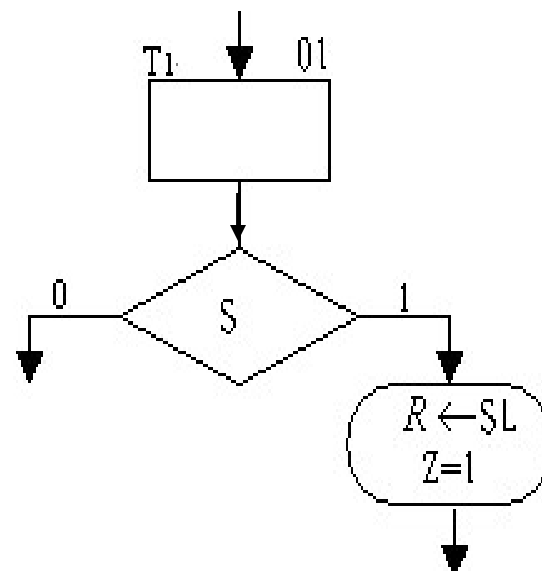
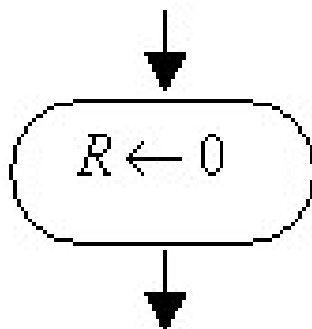
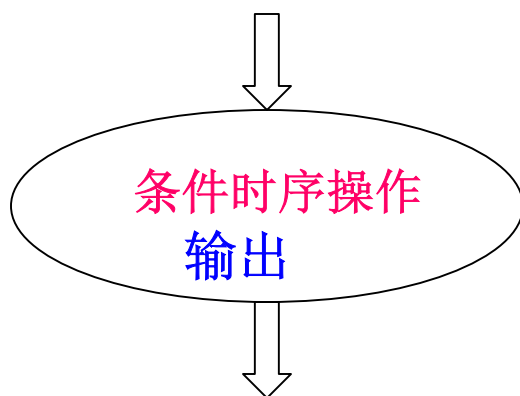
4 输出

控制器根据判断框内容(条件)决定下一个 **CLK** 到时  
状态转换

### 3. 条件框 (椭圆)

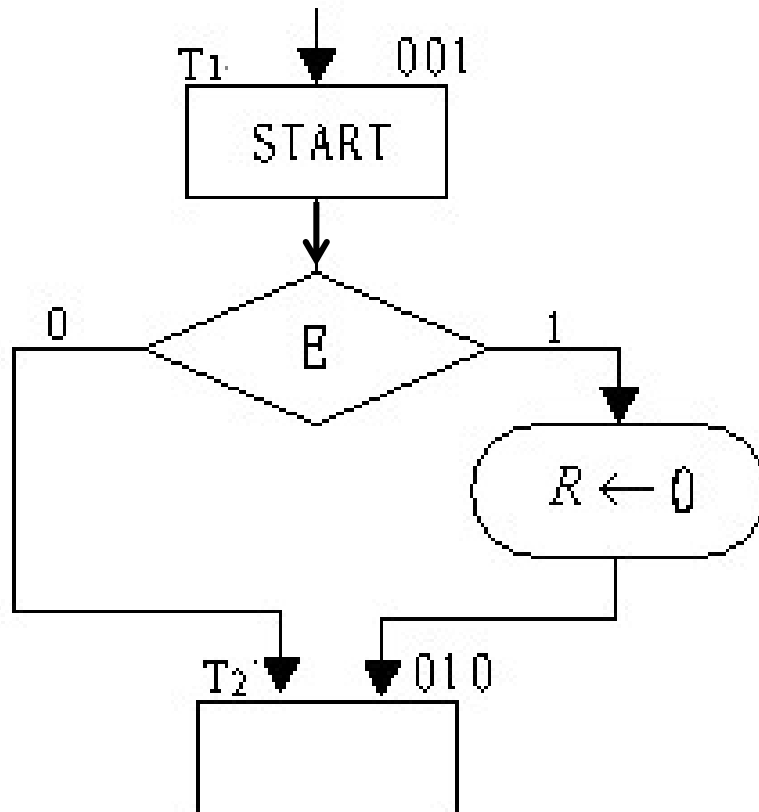
条件框内的操作为条件操作

入口只能接判断框的分支





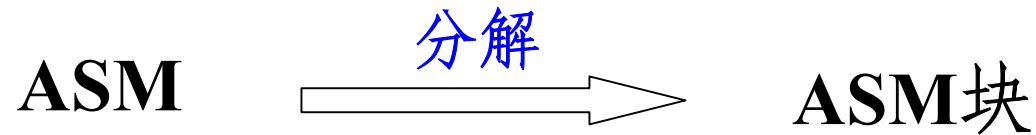
例：分析以下ASM图



系统：状态  $T_1$ ，  
输出：START

如果输入  $E = 1$ ，  
下一个  $CLK$  来临  
 $R$  复位(清零)，  
否则  $R$  不改变，  
新状态  $T_2$

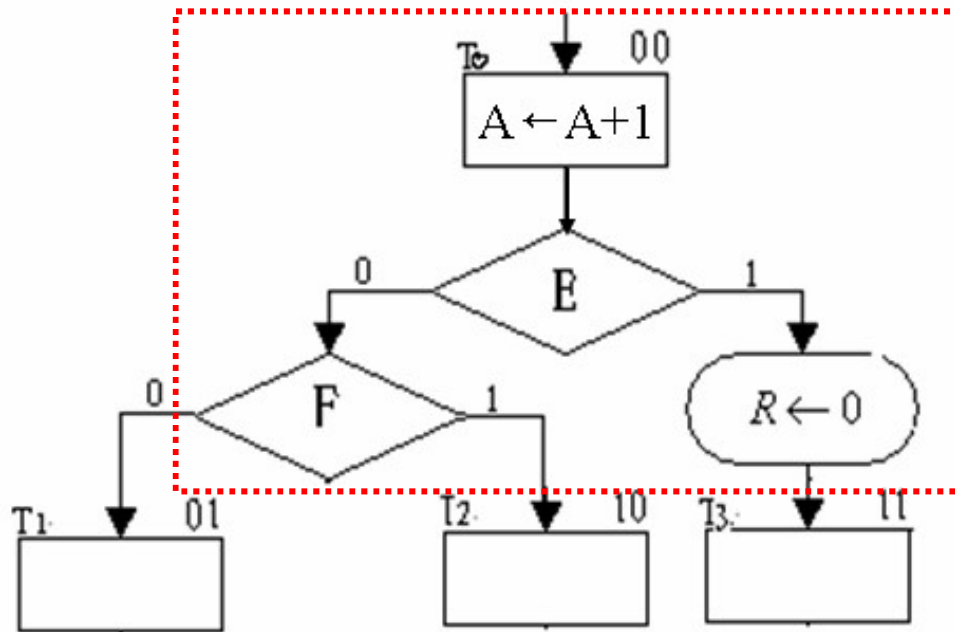
## 8.2.2 ASM块



规则：

一个ASM块包含且只包含一个状态框，并且和一些判断框或者条件框相连接。

练习

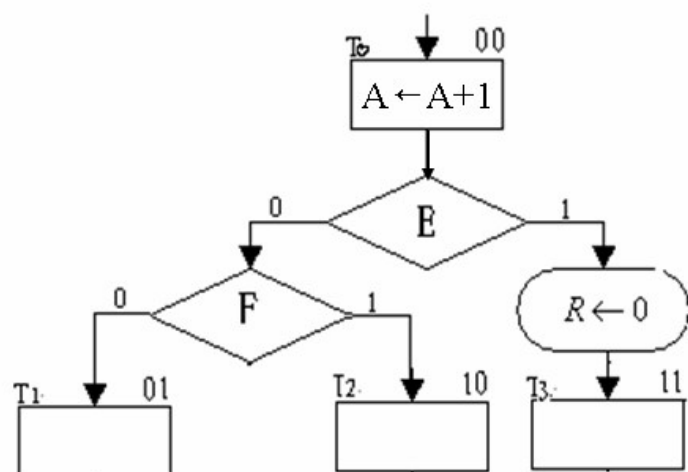


ASM块 { 一个状态框  
两个判断框  
一个条件框

划分 ASM 块的意义：

一个ASM块定义数字系统的一个时序，

即一个ASM块内的操作在一个CLK周期完成。



$T_0$ 状态,  
下一个 $CLK$ 到来:

数据处理器

控制器

$A \leftarrow A+1$   
(无条件操作)

$A \leftarrow A+1$

与下面三个操作中的一个同时完成

$\left\{ \begin{array}{ll} \text{If } E = 1, & R \leftarrow 0 \\ & \text{(条件操作)} \\ \text{If } E = 0, F = 0, & \\ & E = 0, F = 1, \end{array} \right.$

状态  $T_0 \rightarrow T_3$

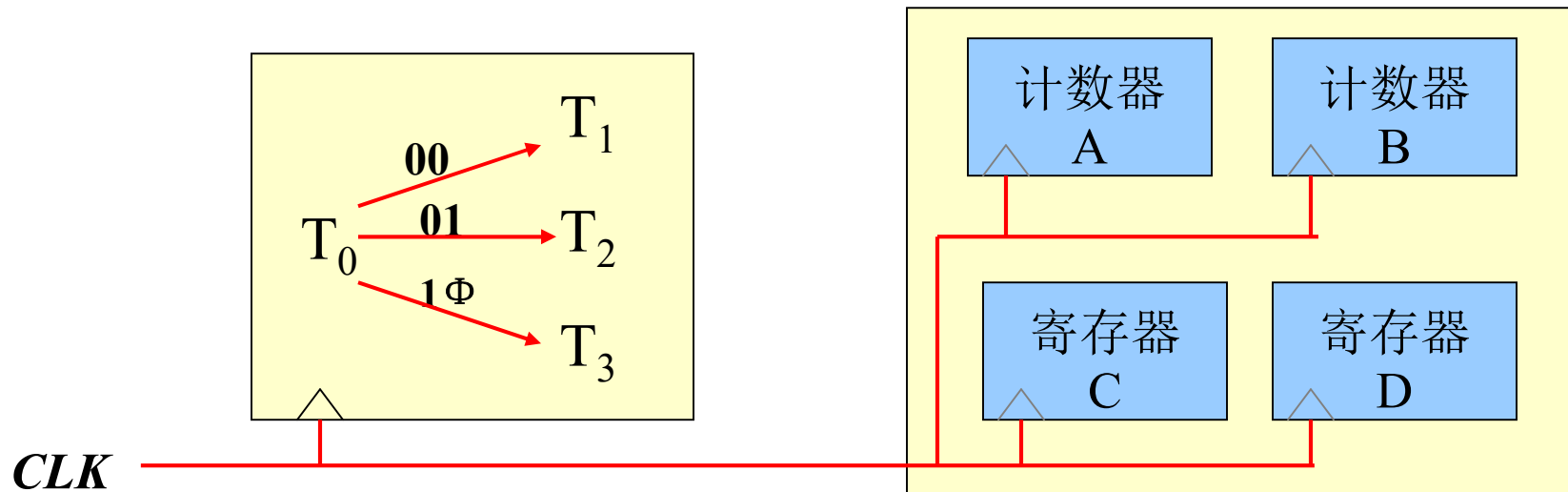
$T_0 \rightarrow T_1$

$T_0 \rightarrow T_2$

一个ASM块中的不同器件在一个 $CLK$ 内各自完成各自的操作 (同一器件不能同时做两件事)

控制器

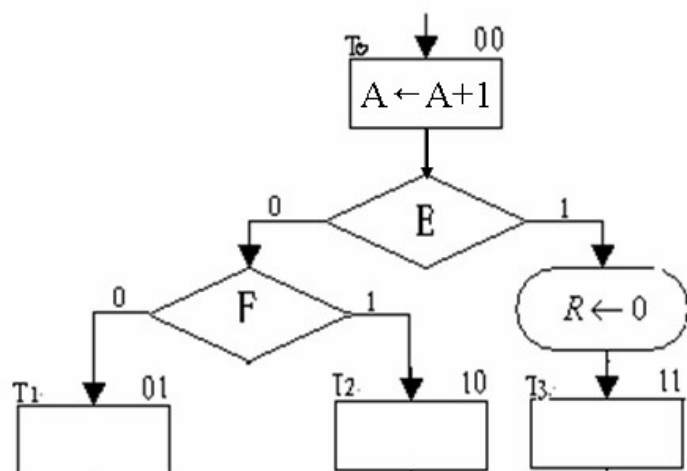
数据处理器



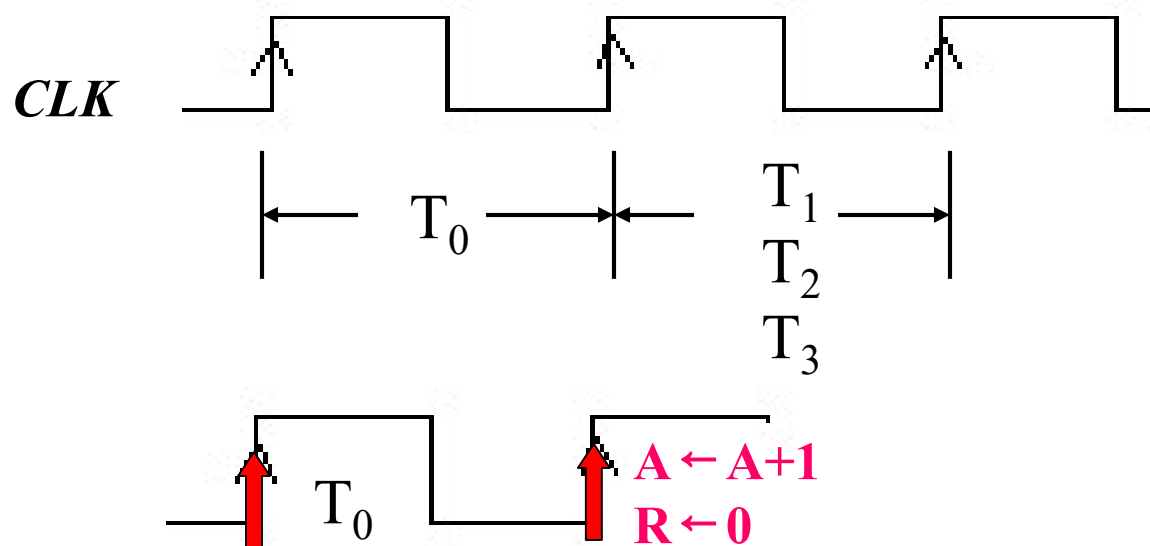
$E, F$  : 外部输入, 已知,

不是  $T_0$  完成后的第二步, 同时判断并操作。

$T_0$  在前一个 $CLK$ 形成。

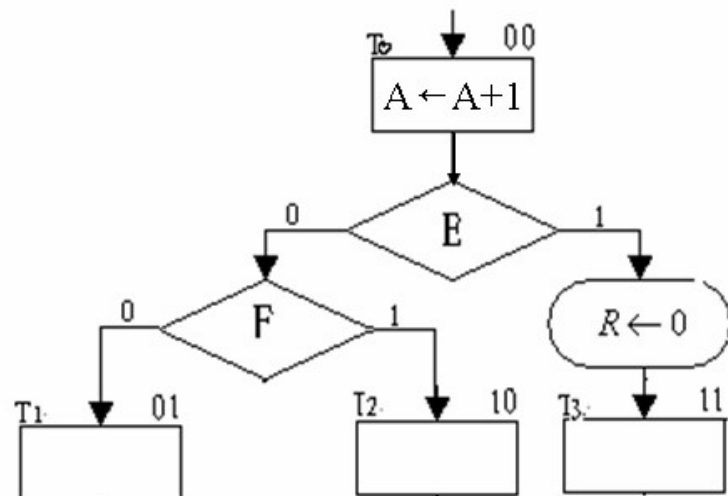


从波形图(时序图)加深理解

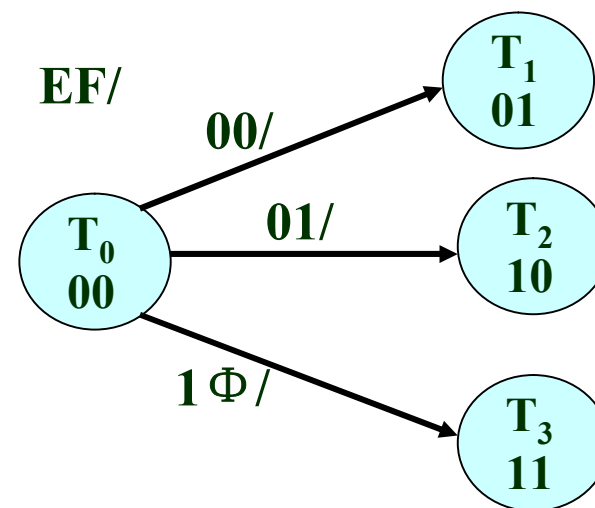


现态  $T_0$  与状态框内的操作不是在同一个  $CLK$  内

## 关系: ASM ~ 状态图



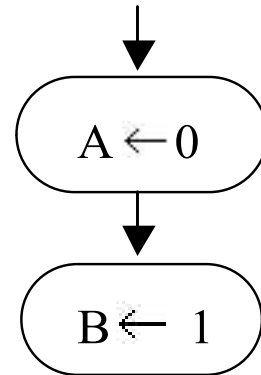
## ASM → 状态图



	ASM	状态图	}	相同
状态转换	✓	✓		
转换条件	✓	✓	}	不同
数据处理器操作	✓	✗		
描述	系统	控制器		

## 例1. 纠错

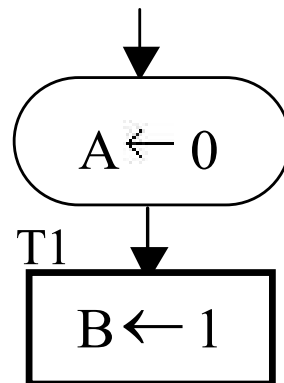
(1)



错误:

条件框的入口  
只能接判断框

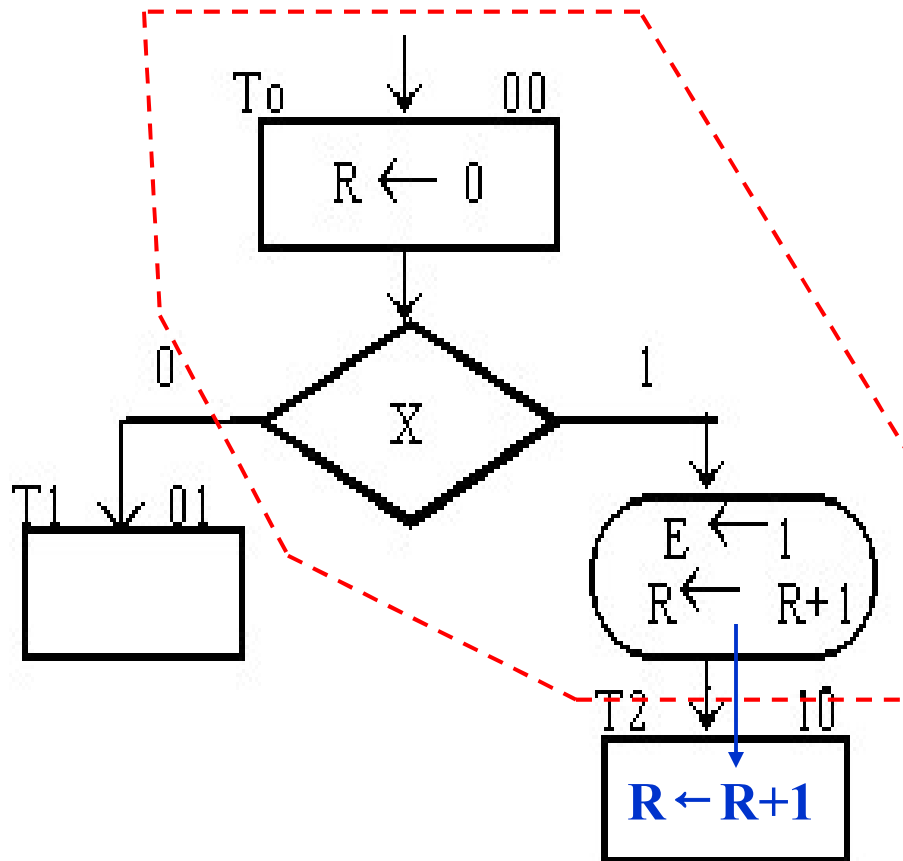
改变为:



( 在两个 *CLK* 的操作 )



(2)

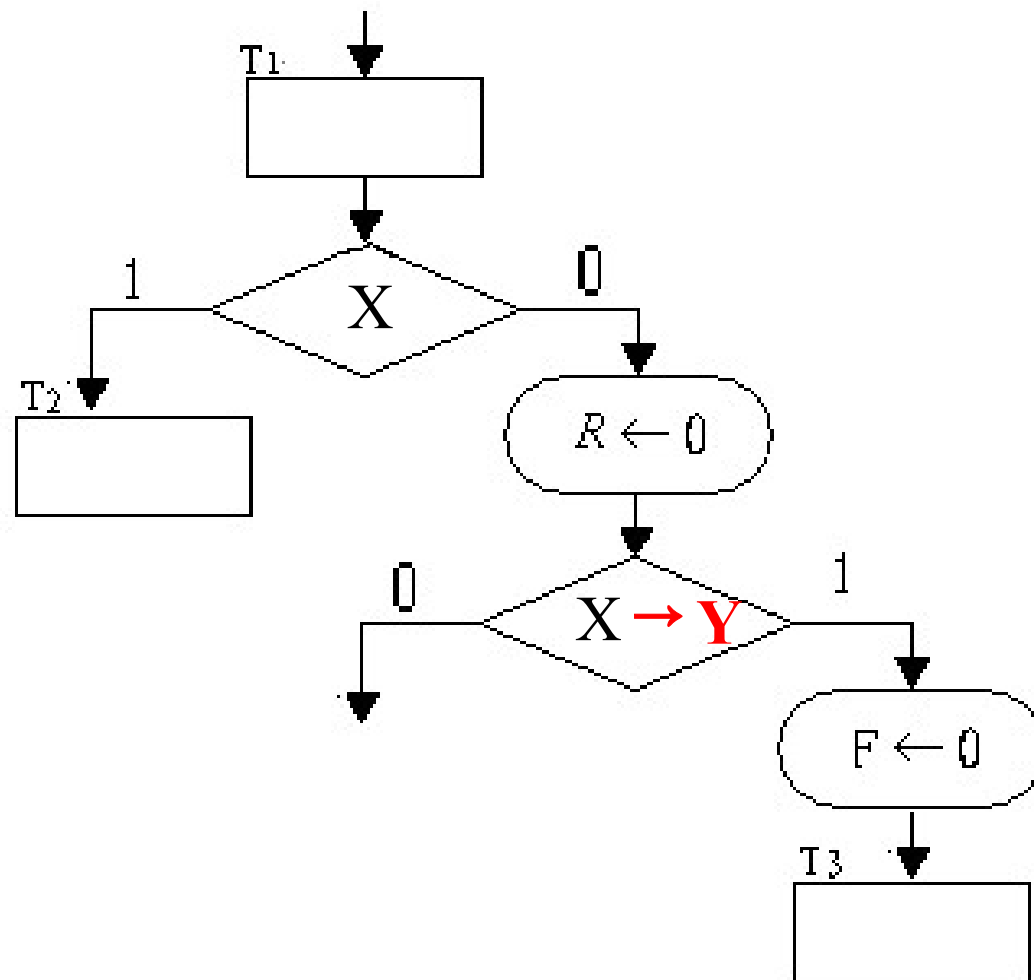


错误:

$R$  在一个  $CLK$  中运行两次(一个 ASM 块)

把一个操作  
移向另外一个  
框中

(3)



错误:

**X** 在一个 *CLK* 中  
使用两次

纠错:

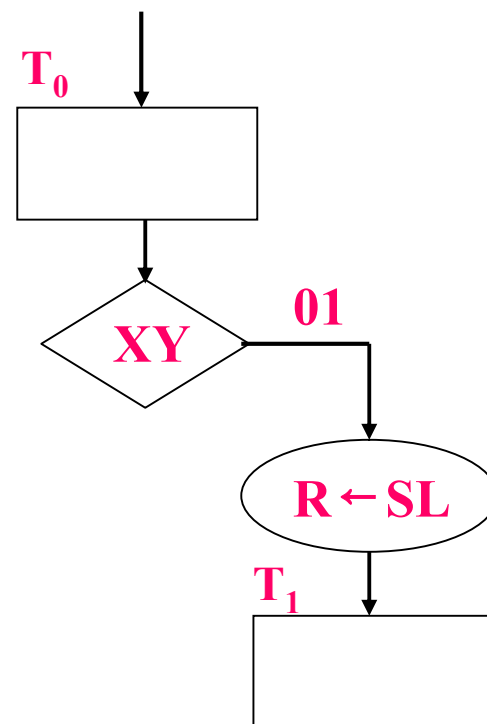
把 **X** 变成 **Y**

## 8.2.3 ASM图表的建立

### 例1

状态 $T_0$ ，如果控制输入 $X=0, Y=1$ ，寄存器 $R$ 左移，状态 $T_0$ 变为 $T_1$ 。画出ASM图。

ASM:



## 例 2

用数字系统记录并显示车场内的存车数目,入口出口都有光电元件,每当有汽车进入车场时,光线有变化,信号Y由1→0;汽车离开车场时,出口信号Z由1→0;信号Y,Z与时钟同步,记录车场车辆数目的数据处理器是一可逆计数器,画出该数字系统的ASM图表.

光电传感器

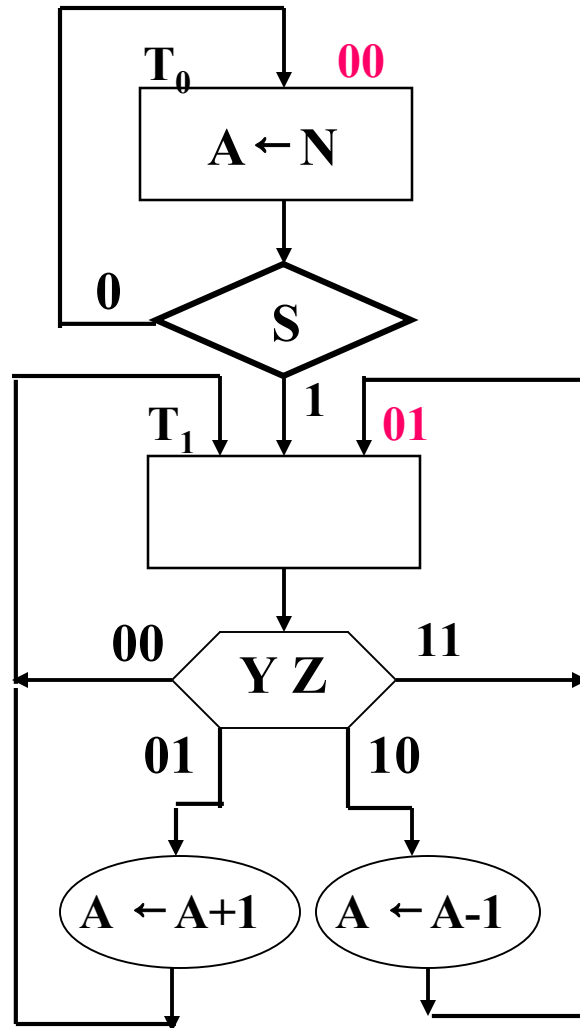
输入, Y  $\begin{cases} Y = 1 & \text{没有车进入} \\ Y = 0 & \text{有车进入} \end{cases}$

输入, Z  $\begin{cases} Z = 1 & \text{没有车离开} \\ Z = 0 & \text{有车离开} \end{cases}$

假设N:当前在公园中车的数量

S:启动信号  $\begin{cases} S = 1 & \text{开始} \\ S = 0 & \text{不变} \end{cases}$

# ASM



A: 计数器

操作  $A \leftarrow N$  需要一个 **CLK**,  
一个状态框

输入, Y  $\begin{cases} Y = 1 \text{ 没有车进入} \\ Y = 0 \text{ 有车进入} \end{cases}$

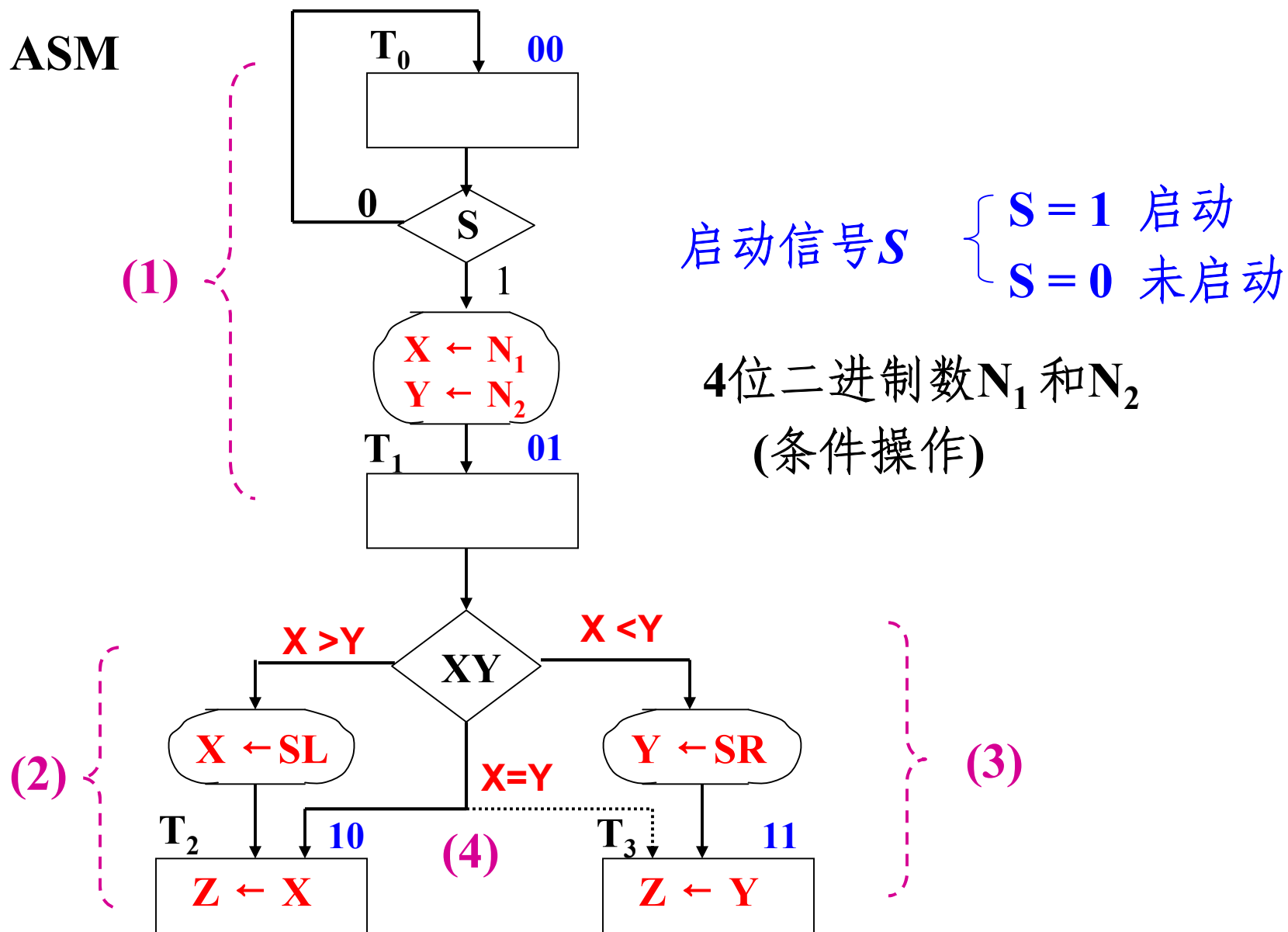
输入, Z  $\begin{cases} Z = 1 \text{ 没有车离开} \\ Z = 0 \text{ 有车离开} \end{cases}$

### 例3

设计一个有三个4位寄存器  $X, Y, Z$  的数字系统，实现功能如下：

- (1). 启动信号  $S$  出现, 传送两个4 位二进制数  $N_1, N_2$ , 给寄存器  $X, Y$ ;
- (2). 若  $X > Y$ , 左移  $X$ , 结果送给  $Z$ ;
- (3). 若  $X < Y$ , 右移  $X$ , 结果送给  $Z$ ;
- (4). 若  $X = Y$ , 传送  $X$  或  $Y$  给  $Z$ .

ASM

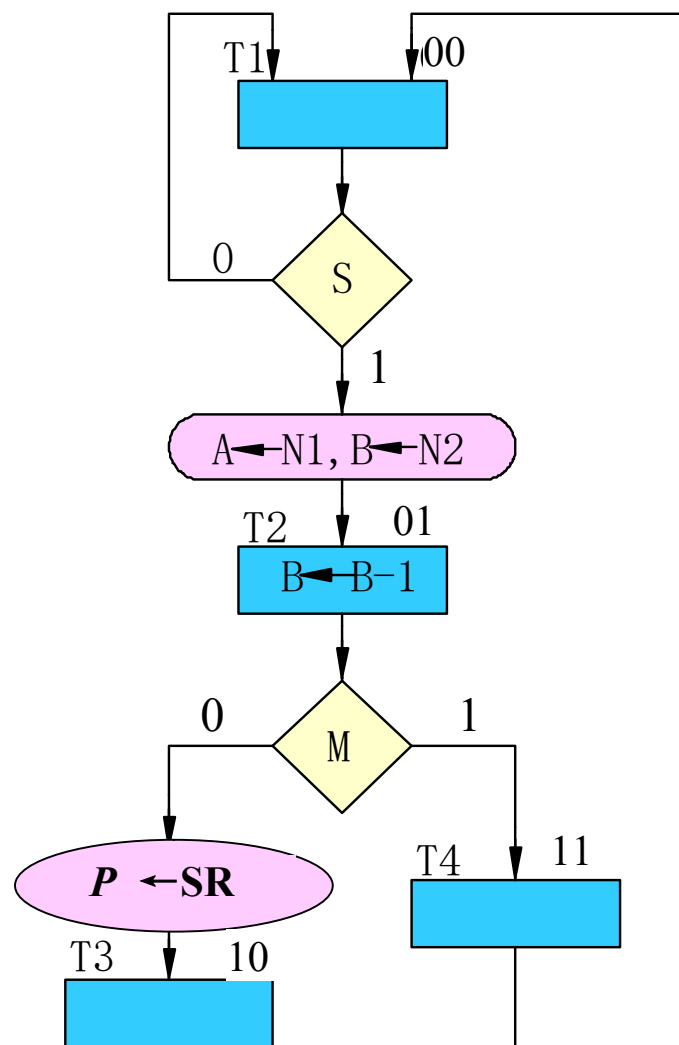


例：

一个数字系统, 在状态  $T_1$ , 如果启动信号  $S=0$ , 状态没有改变; 若  $S=1$ , 操作  $A \leftarrow N_1$  和  $B \leftarrow N_2$ , 状态改变为  $T_2$ .

在状态  $T_2$ , 下一个  $CLK$  到来, 操作  $B \leftarrow B-1$ , 如果  $M=0$ , 到下一个操作  $P$  右移, 状态  $T_2$  变为  $T_3$ ; 若  $M=1$ , 状态  $T_2 \rightarrow T_4 \rightarrow T_1$ .

画出此系统的ASM图





# § 8.3 数字系统设计

## 8.3.1 设计步骤

1. 分析

2. 画出ASM

3. 设计控制器

状态转换

4. 设计数据处理器

有条件或无条件操作

5. 电路图

## 8.3.2 数字系统设计举例

### 例1

设计三种图案彩灯控制系统的控制器，三种图案彩灯依次循环亮，其中苹果形图案灯亮**16s**，香蕉形图案亮**12s**，葡萄形图案亮**9s**。

### 1. 分析

输入：

计时信号  $\left\{ \begin{array}{l} 16s : X=1 \\ 12s : Y=1 \\ 9s : Z=1 \end{array} \right.$

定时启动  $\left\{ \begin{array}{l} = 1 \text{ 计时开始} \\ = 0 \text{ 否则} \end{array} \right.$

输出：

灯亮

苹果形：  $A = 1$

香蕉形：  $B = 1$

葡萄形：  $G = 1$

高电平有效

## 2. 画出ASM

## 3. 控制器设计

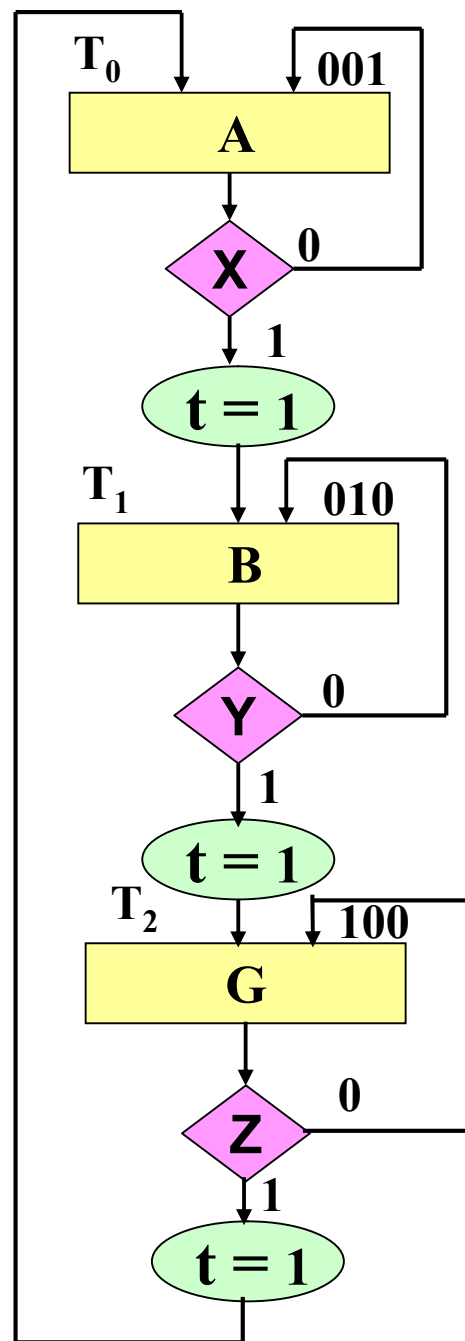
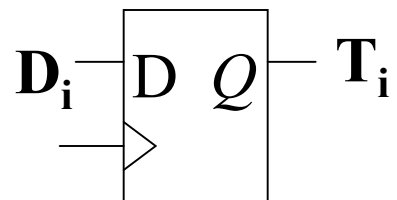
$(X, Y, Z \rightarrow T_0, T_1, T_2)$

方法1:

每个状态一个触发器

状态数= 触发器个数

选择D触发器,  $Q^{n+1} = D$



根据ASM图，各个状态的输入条件作为D触发器的控制输入方程

在任何时候, 只有一种状态 = 1,  
其他状态 = 0

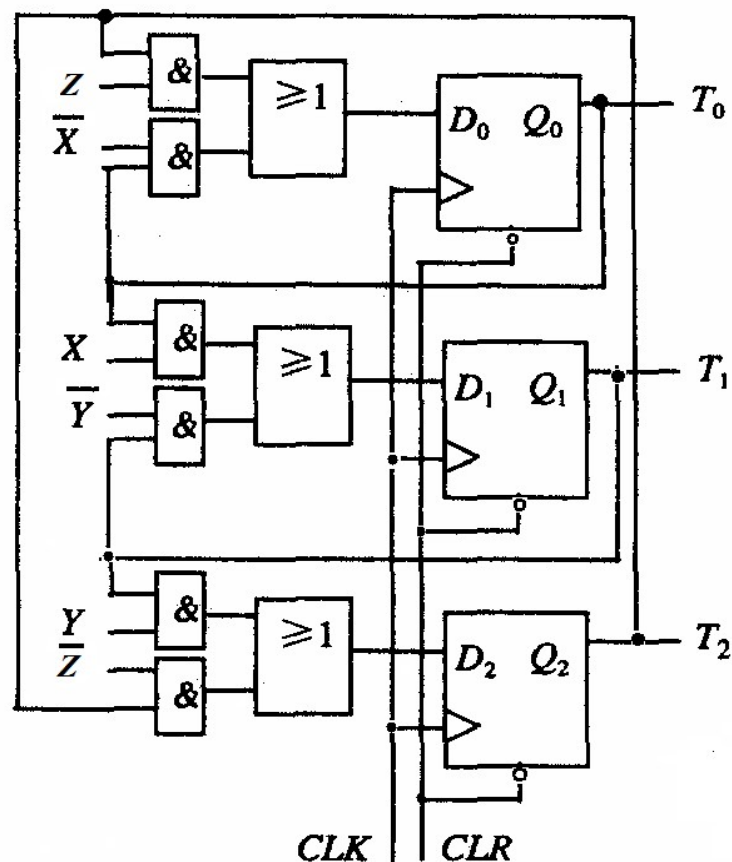
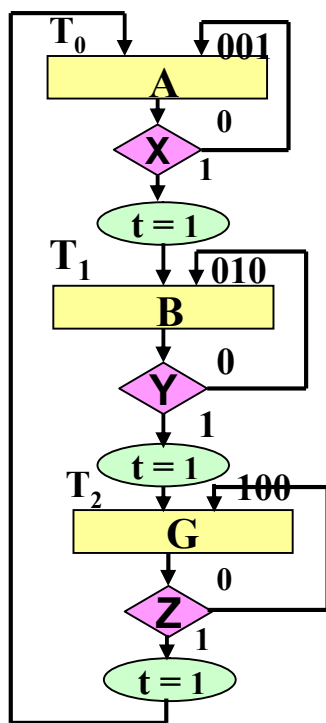
3个状态, 3个D触发器,  
输入 D, 输出  $T_i$

$$T_0=1, \quad D_0 = T_0 \bar{X} + T_2 Z$$

$$T_1=1, \quad D_1 = T_0 X + T_1 \bar{Y}$$

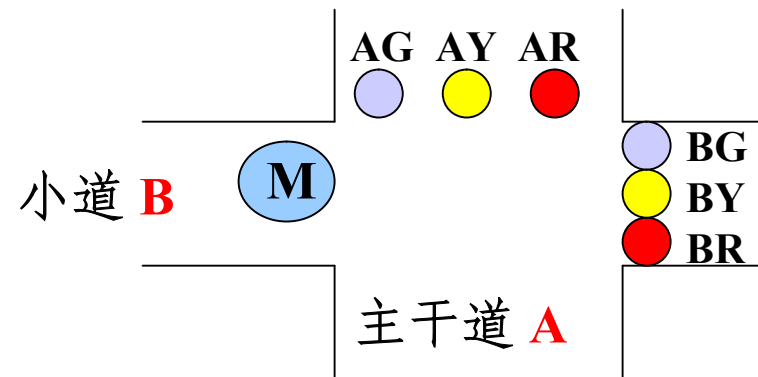
$$T_2=1, \quad D_2 = T_1 Y + T_2 \bar{Z}$$

#### 4. 电路



## 例2.十字路口交通灯管理系统

### 十字路口交通灯管理系统 (P187. 例 8.8)



在主干道 **A** 和小道 **B** 的十字交叉路口，设置交通灯管理系统。小道 **B** 路口设有传感器 **M**，小道有车 **M=1**，否则 **M=0**。主干道通车最短**16s**，超过**16s**，若小道有车 (**M=1**)，主干道绿灯灭黄灯亮**3s**，然后红灯亮。小道绿灯 (通车) 最长时间**16s**，在**16s**内，只要小道无车 (**M=0**)，小道由绿灯变黄灯 (**3s**) 后变红灯，主干道红灯变绿灯。**16s** 和**3s** 定时信号由加法计数器完成，时间到，**t=1**，计数器清**0**，重新计时下一个定时时间。

## 1. 分析:

输入: 传感器 **M**  $\begin{cases} = 1 & \text{有车} \\ = 0 & \text{没有车} \end{cases}$

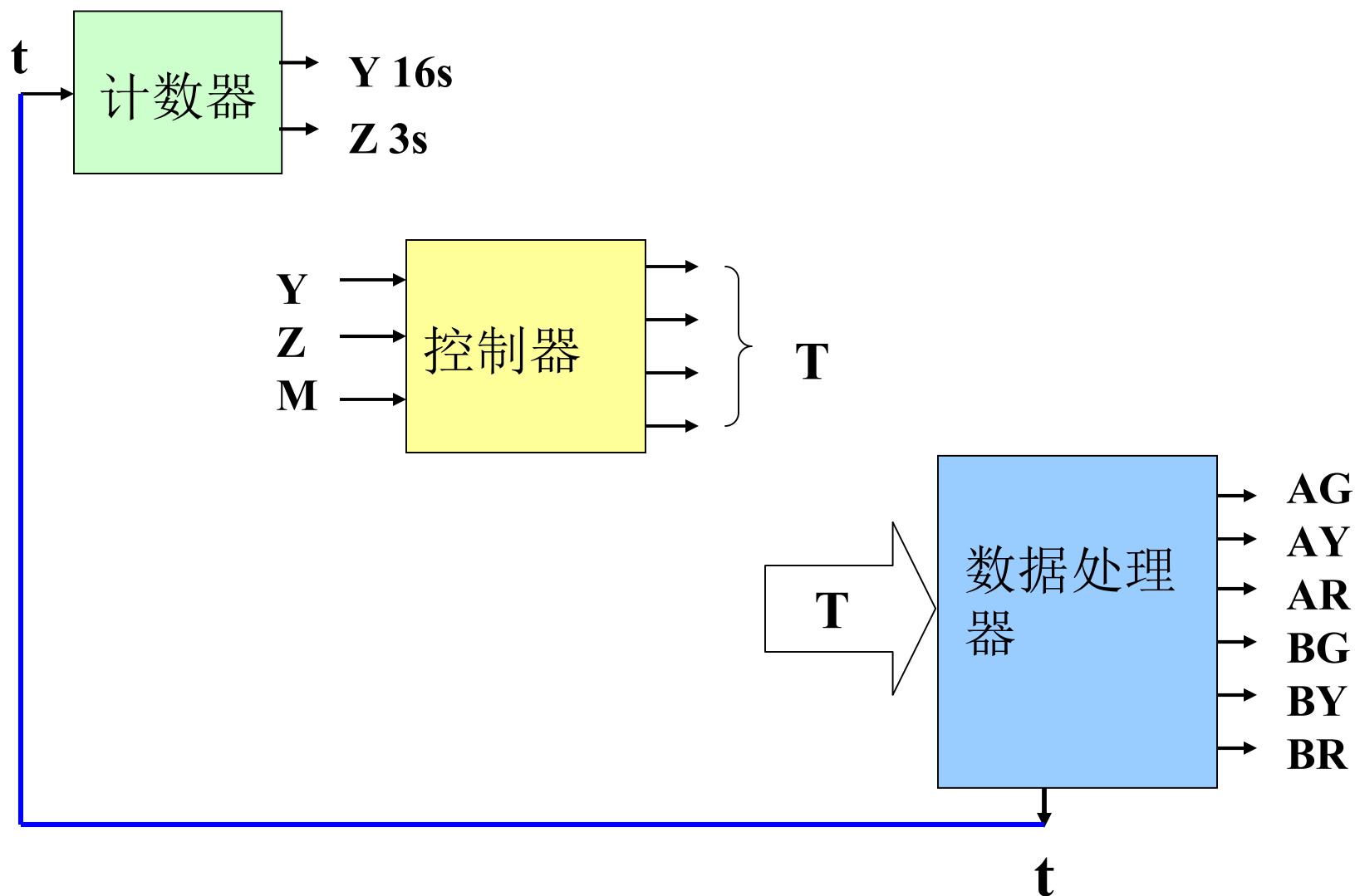
定时启动 **t**  $\begin{cases} = 1 & \text{计时开始} \\ = 0 & \text{其他} \end{cases}$

计时信号  $\begin{cases} \mathbf{Y} \begin{cases} = 1 & 16\text{s 以上} \\ = 0 & \text{其他} \end{cases} \\ \mathbf{Z} \begin{cases} = 1 & 3\text{s 以上} \\ = 0 & \text{其他} \end{cases} \end{cases}$

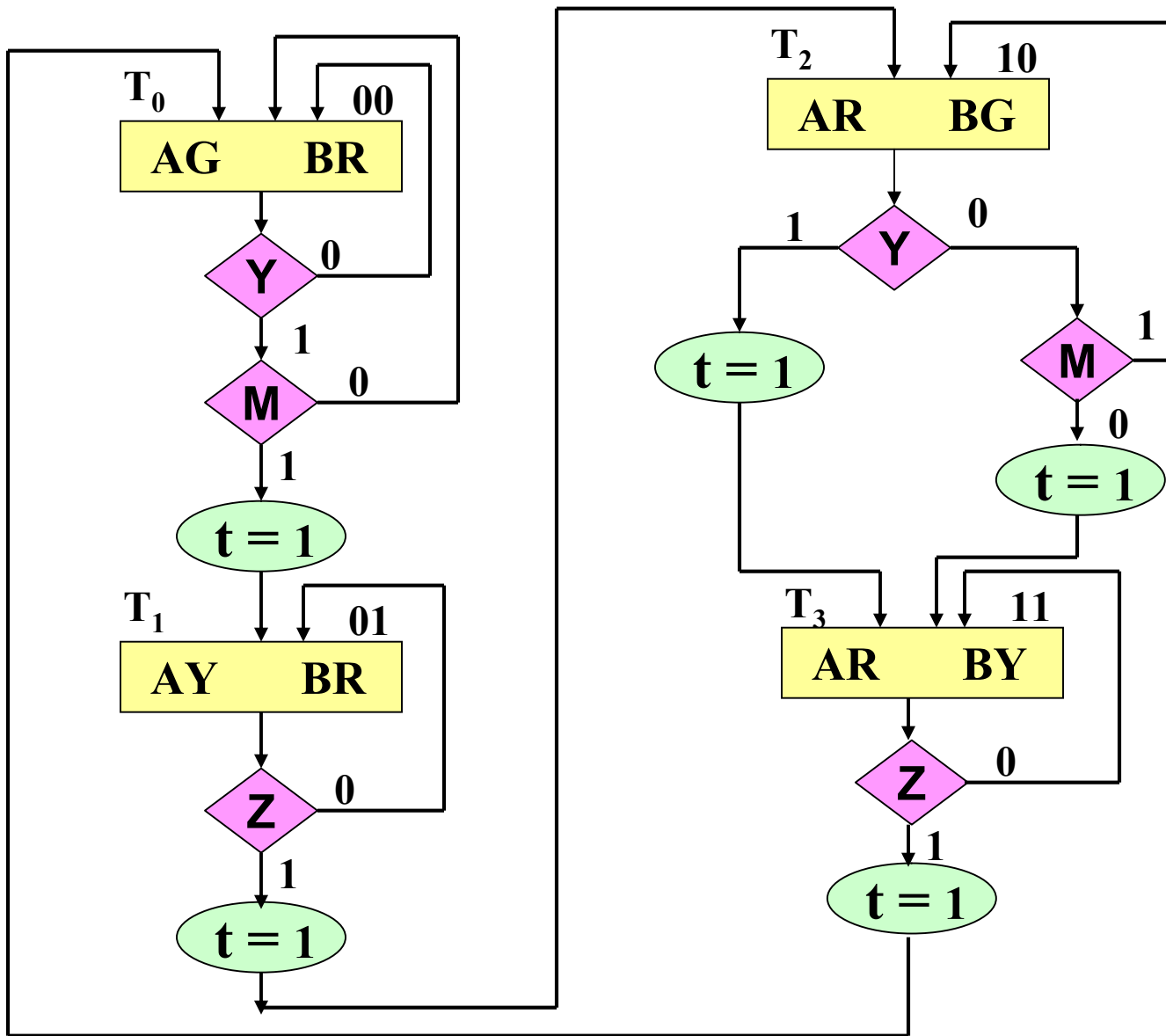
输出: **AG, AY, AR, BG, BY, BR = 1 亮**

输入和输出都是高电平有效 (=1)

系统需要几部分:



## 2. 画出ASM





### 3. 控制器设计

(用  $M, Y, Z$  得到:  $T_0, T_1, T_2, T_3$ )

方法2: 使用数据选择器, D触发器, 译码器设计控制器

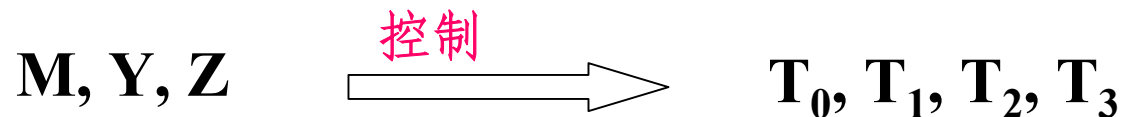
根据ASM: 4个状态  $T_0, T_1, T_2, T_3$

输出: 高电平有效2-4译码器

其入口接两个 D触发器 的出口  $Q_0, Q_1$

D触发器的入口各接一个4-1 数据选择器

4-1 数据选择器入口接  $M, Y, Z$ , 实现



状态图，找到  $Q_1^{n+1}$ ,  $Q_0^{n+1}$  (即  $D_1$ ,  $D_0$ ) 与总输入  $M, Y, Z$  的关系  
从ASM图：

状态符号	现 态 $Q_1^n$ $Q_0^n$	输 入 Y Z M	次 态 $Q_1^{n+1}$ $Q_0^{n+1}$	输 出 $T_0$ $T_1$ $T_2$ $T_3$
$T_0$	0 0	0 $\Phi$ $\Phi$ 1 $\Phi$ 0 1 $\Phi$ 1	$\mathbf{0} \left\{ \begin{array}{l} 0 \\ 0 \\ 0 \end{array} \right. \begin{array}{l} 0 \\ 0 \\ 1 \end{array} \right\} \mathbf{YM}$	1 0 0 0
$T_1$	0 1	$\Phi$ 0 $\Phi$ $\Phi$ 1 $\Phi$	$\mathbf{Z} \left\{ \begin{array}{l} 0 \\ 1 \end{array} \right. \begin{array}{l} 1 \\ 0 \end{array} \right\} \mathbf{\bar{Z}}$	0 1 0 0
$T_2$	1 0	0 $\Phi$ 1 0 $\Phi$ 0 1 $\Phi$ $\Phi$	$\mathbf{1} \left\{ \begin{array}{l} 1 \\ 1 \\ 1 \end{array} \right. \begin{array}{l} 0 \\ 1 \\ 1 \end{array} \right\} \begin{array}{l} Y + \bar{Y}\bar{M} \\ Y + \bar{M} \end{array}$	0 0 1 0
$T_3$	1 1	$\Phi$ 0 $\Phi$ $\Phi$ 1 $\Phi$	$\mathbf{\bar{Z}} \left\{ \begin{array}{l} 1 \\ 0 \end{array} \right. \begin{array}{l} 1 \\ 0 \end{array} \right\} \mathbf{\bar{Z}}$	0 0 0 1

4-1MXU的输入即转换条件, 也就是 **D-FF** 的输入变量方程,  
用引入变量 卡诺图 (VEM), 把 **M, Y, Z** 作引入变量.

$Q_1^{n+1}$		$Q_1^n$		0	1
		$Q_0^n$		0	1
0				0	1
1				Z	$\overline{Z}$

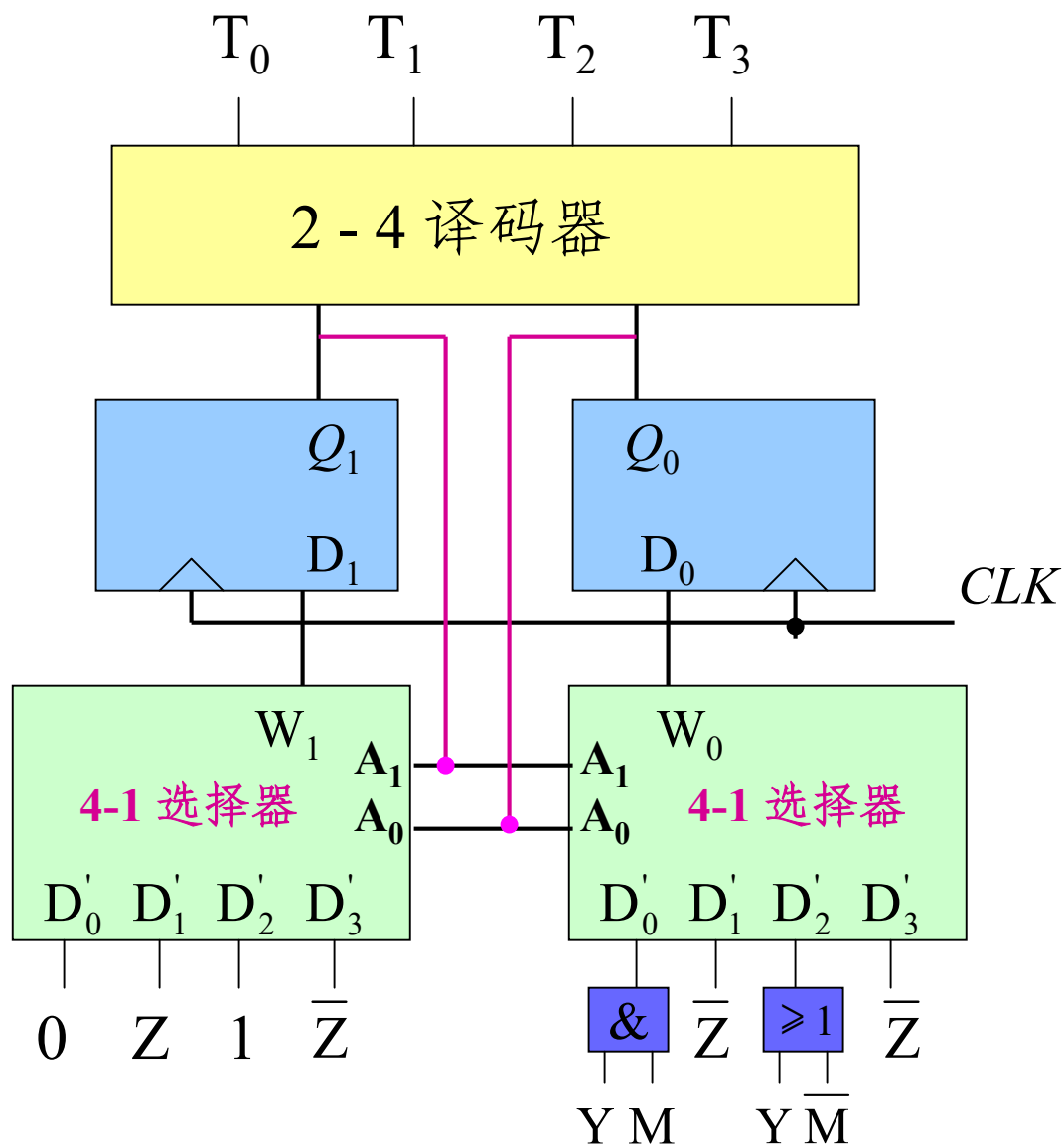
**$D_1$**

$Q_0^{n+1}$		$Q_1^n$		0	1
		$Q_0^n$		0	1
0				YM	$Y+\overline{M}$
1				$\overline{Z}$	$\overline{Z}$

**$D_0$**

四个小格分别为4-1MUX的输入端变量

# 电路



$Q_1^{n+1} \backslash Q_1^n$	0	1
0	0	1
1	Z	$\bar{Z}$

$Q_0^{n+1} \backslash Q_1^n$	0	1
0	YM	$Y+\bar{M}$
1	$\bar{Z}$	$\bar{Z}$

**T<sub>0</sub> 状态** (  $T_0T_1T_2T_3 = 1000$  ),  $Q_1Q_0 = 00$ , 使  $A_1A_0 = 00$ , (4-1选  $D_0'$  )

$$D_1 = W_1 = 0, \quad D_0 = W_0 = MY,$$

下一个 *CLK* 到来,  $Q_1^{n+1} = D_1 = 0$ ,  $Q_0^{n+1} = D_0 = MY$ .

(若  $MY = 0$ , 即  $\left\{ \begin{array}{l} \text{或支路无车} \\ \text{或16s未到} \end{array} \right\}$   $Q_1Q_0 = 00$ , 保持状态  $T_0$ )

当  $MY = 1$  (支路有车, 16s 到),  $Q_1Q_0 = 01$ , 输出新状态 **T<sub>1</sub> (0100)** ;

**T<sub>1</sub> 状态**,  $Q_1Q_0 = 01$ ,  $A_1A_0 = 01$ ,  $D_1 = W_1 = Z$ ,  $D_0 = W_0 = \bar{Z}$ ,

下一个 *CLK* 到来,  $Q_1^{n+1} = D_1 = Z$ ,  $Q_0^{n+1} = D_0 = \bar{Z}$

当  $Z = 1$  (3s 到),  $\bar{Z} = 0$ ,  $Q_1Q_0 = 10$ , 输出新状态 **T<sub>2</sub> (0010)** .

## 4. 数据处理器设计

### (1) 灯的电路

$$AG = T_0$$

$$AY = T_1$$

$$AR = T_2 + T_3$$

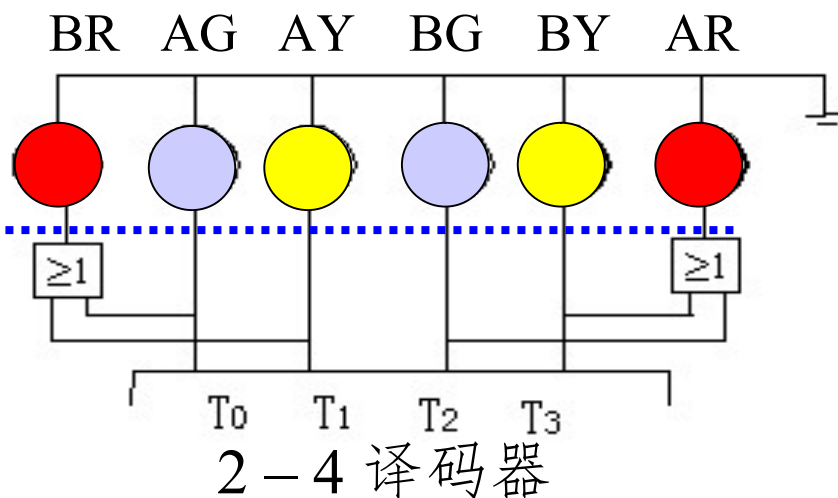
$$BG = T_2$$

$$BY = T_3$$

$$BR = T_0 + T_1$$

状态	AG	AY	AR	BG	BY	BR
$T_0$	1	0	0	0	0	1
$T_1$	0	1	0	0	0	1
$T_2$	0	0	1	1	0	0
$T_3$	0	0	1	0	1	0

驱动电路



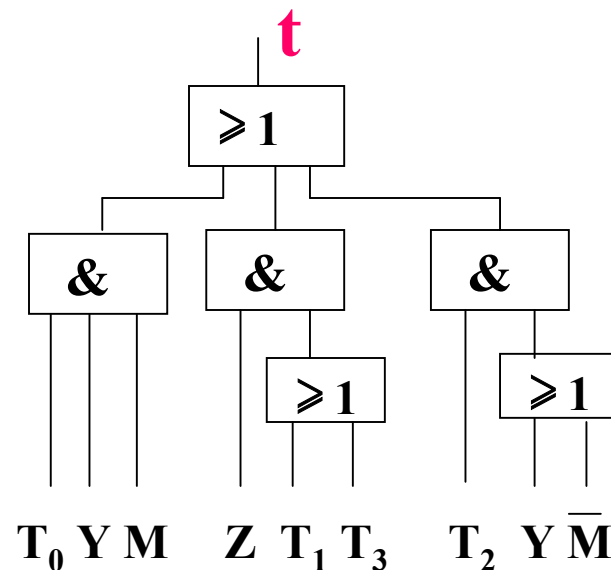
## (2) 定时启动电路

(产生  $t = 1$ )

定时启动:  $t = 1$

ASM:  $t = 1$  条件

$$\begin{aligned} t &= T_0 Y M + T_1 Z + T_2 Y + T_2 \bar{Y} \bar{M} + T_3 Z \\ &= T_0 Y M + (T_1 + T_3) Z + T_2 (Y + \bar{M}) \end{aligned}$$



### (3) 计时电路

(产生Y, Z)

两个计数器 Y : 16s, Z : 3s. 用74161实现: M-16 (Y) 和M-3 (Z)

#### 74161驱动要求

控制信号	操作	驱动条件							
		$\overline{CLR}$	$\overline{LD}$	CNT	CNP	$D_3$	$D_2$	$D_1$	$D_0$
下降沿   <b>t</b>	清0	<b>0</b>	$\Phi$	$\Phi$	$\Phi$	$\Phi$	$\Phi$	$\Phi$	$\Phi$
	启动 (预置)	<b>1</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>

$$\left\{ \begin{array}{l} \overline{CLR} \downarrow \text{ 负边沿} \\ \overline{LD} = \overline{\mathbf{t}} \\ D_3 D_2 D_1 D_0 = 0000 \\ ENT = ENP = 1 \end{array} \right.$$

$$\mathbf{74161} \left\{ \begin{array}{l} Q_3 Q_2 Q_1 Q_0 = 1111 \text{ (CO = 1) 时,} \\ \quad \mathbf{Y = 1 (M-16)} \\ Q_3 Q_2 Q_1 Q_0 = 0010 \text{ 时,} \\ \quad \mathbf{Z = 1 (M-3)} \end{array} \right.$$



## Y 和 Z 输出

ASM 分析,

什么状态下计时16s, 3s?

$$Y = (T_0 + T_2)CO = 1$$

即  $T_0$  或  $T_2$  状态,

$CO = 1$  ( $Q_3Q_2Q_1Q_0 = 1111$ )

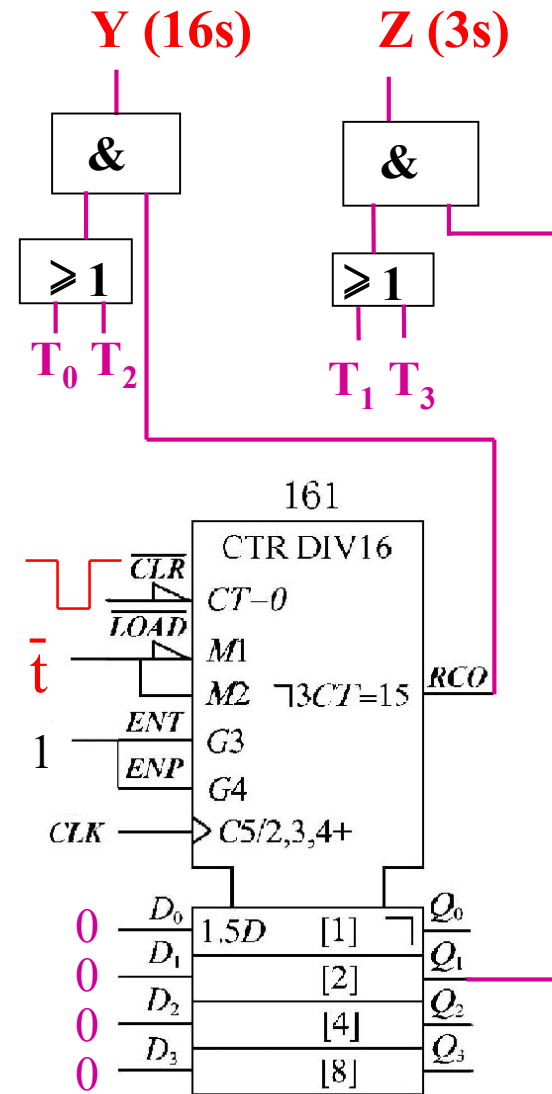
$Y = 1$ ;

$$Z = (T_1 + T_3)Q_1 = 1$$

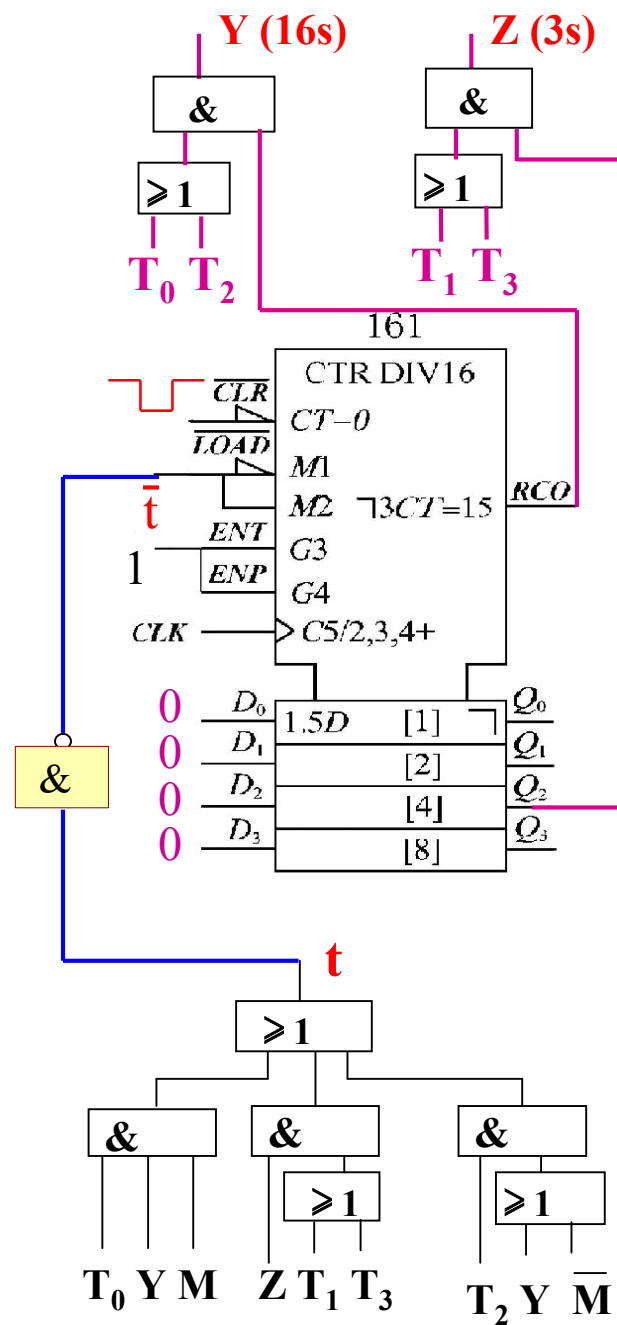
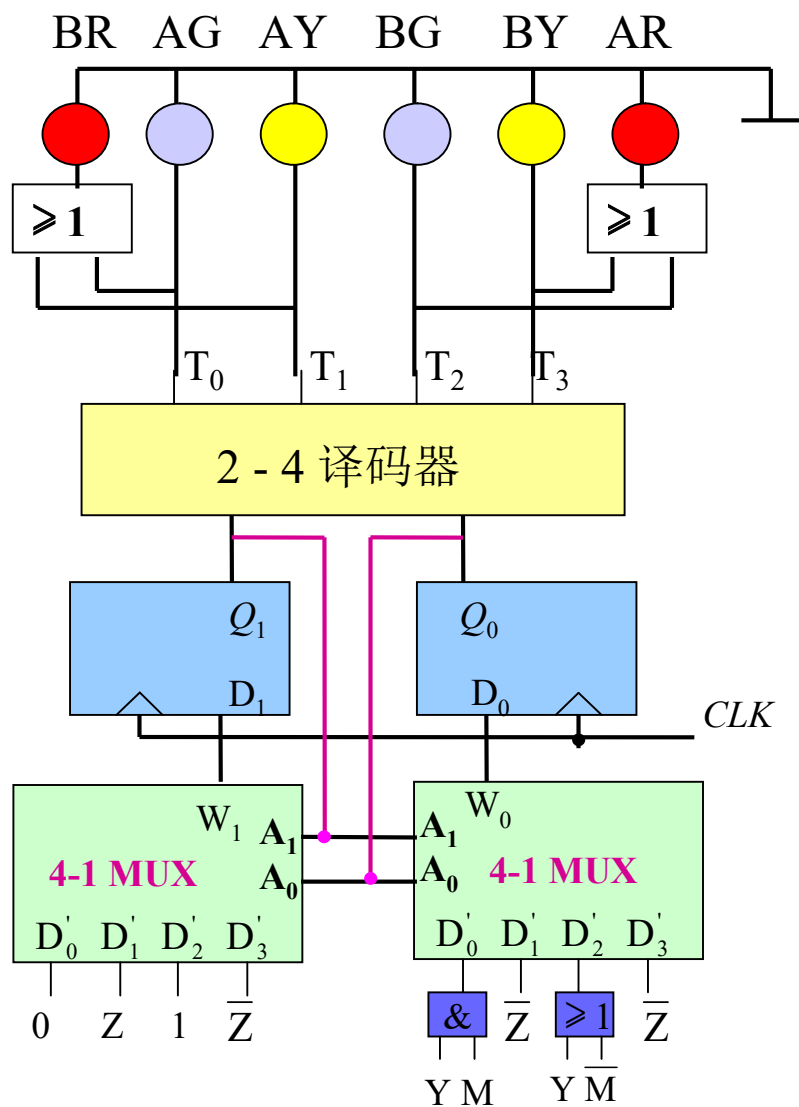
即  $T_1$  或  $T_3$  状态,

$Q_3Q_2Q_1Q_0 = 0010$

$Z = 1$ .

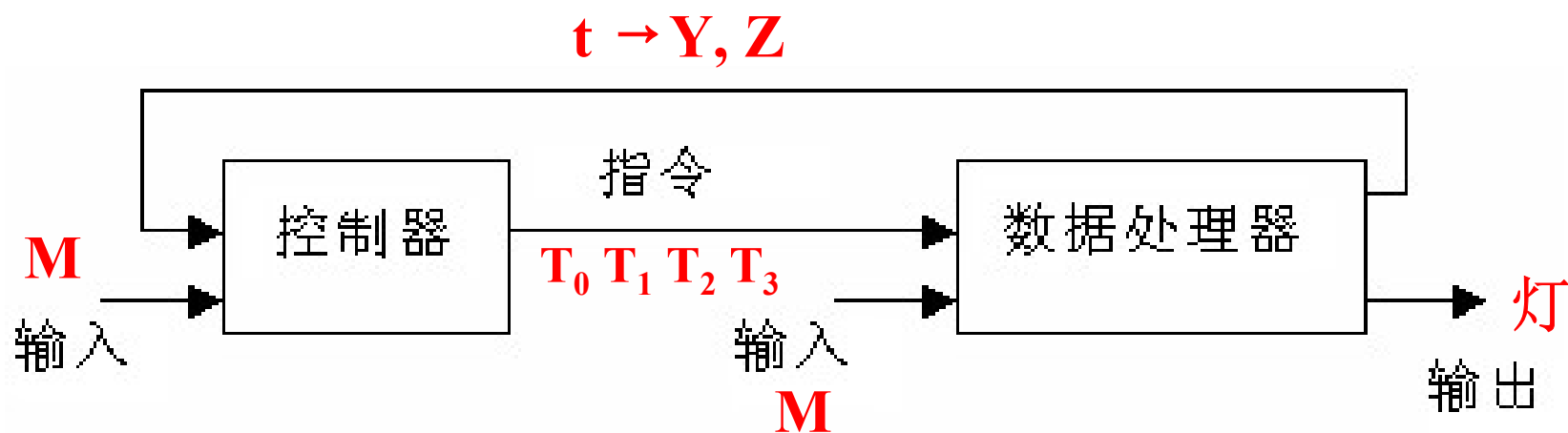


# 系统



注：① 整个系统用一个 $CLK$  脉冲 (控制器和计数器)

② 整个系统用一个 $\overline{CLR}$  (包括 $R_D$ )  $\sqcup$



# 小 结

## ■ ASM图表及生成

- ASM图与程序流程图、状态图之间的区别
- ASM图与状态图之间的转换

## ■ 数字系统的设计

- 理解控制器设计
- 了解数据处理器设计

课后作业：

**8.3**

**8.5**

**8.7**

**8.8**

**8.9**