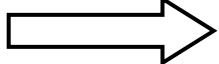


# 第 8 章 数字系统设计基础

## Design of Digital Systems

### §8.1 概述

**数字系统，是指交互式的以离散形式表示的具有存储、传输、处理信息能力的逻辑子系统的集合物。**

**数字系统**  **模块**      **子系统**

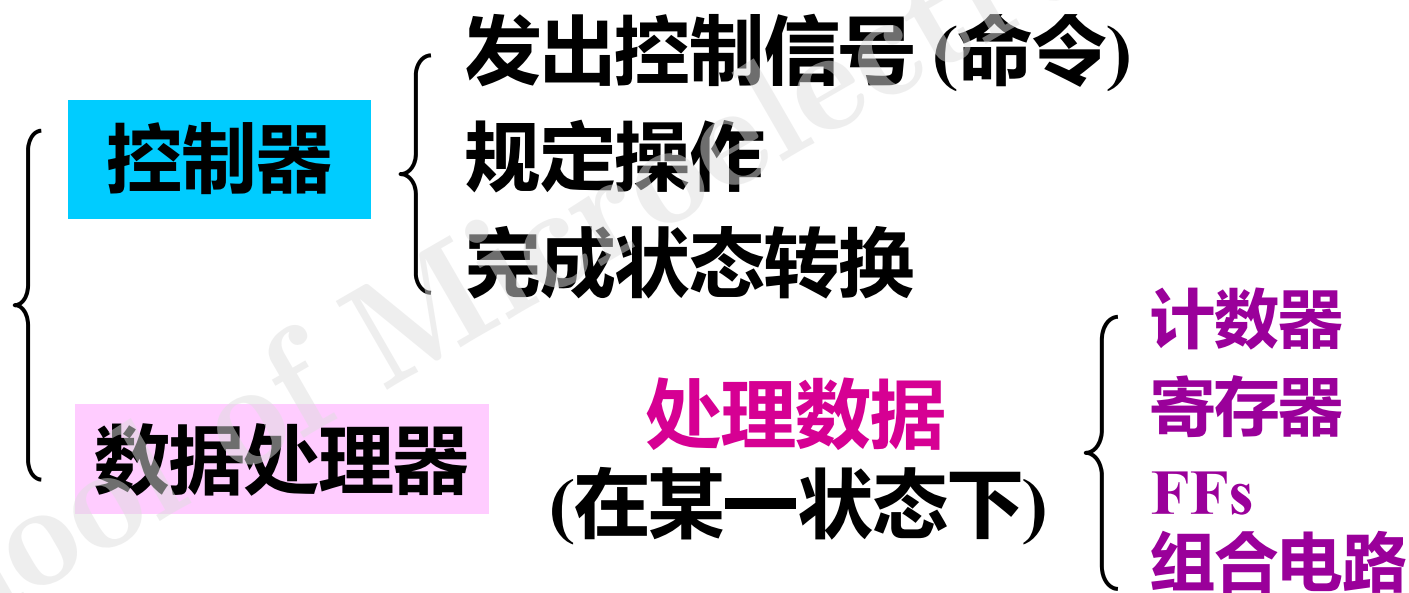
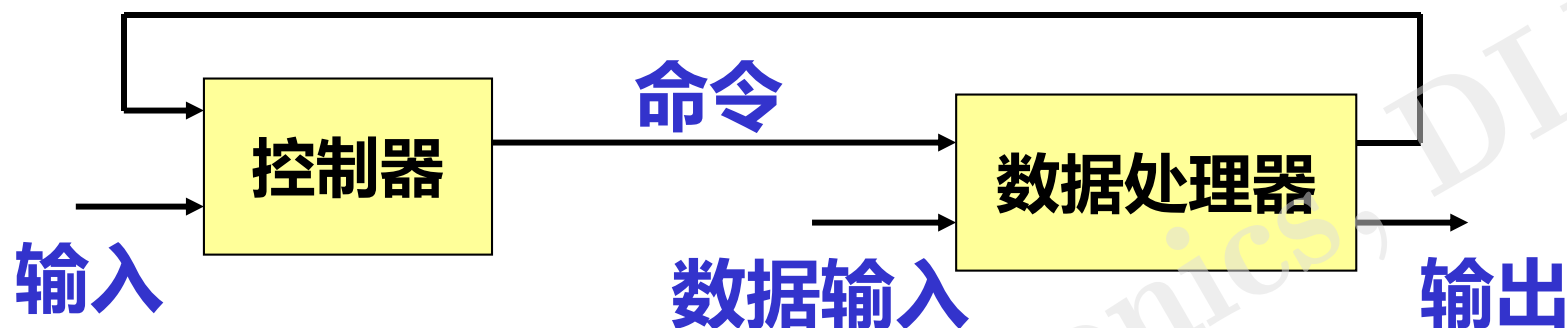
**每个子系统**

**控制器**

**数据处理器**

关系:

状态变量



结果作为状态变量反馈给控制器

## §8.2 算法状态机 — ASM 图表

### Algorithmic State Machine

**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**

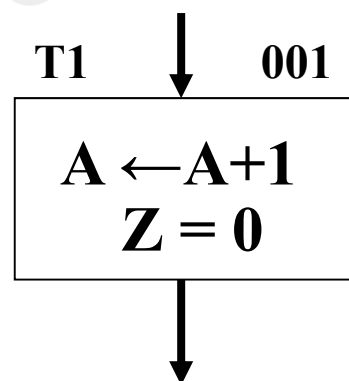
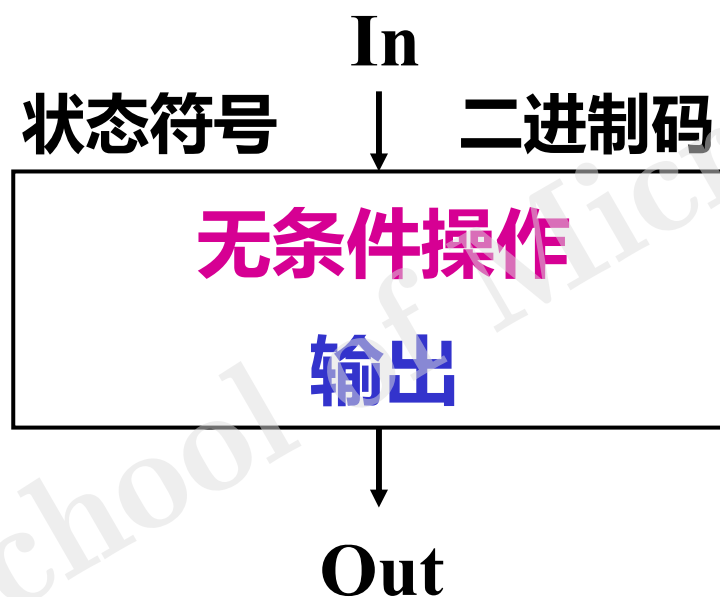
**特点:**

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

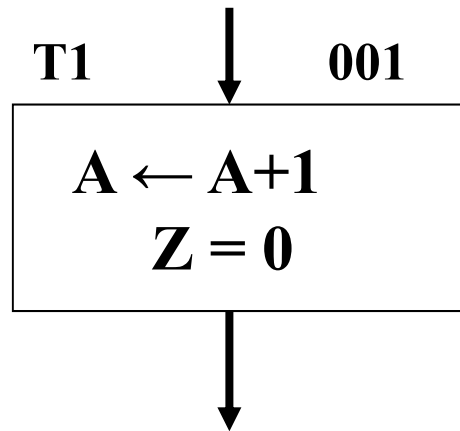
## § 8.2.1 ASM 符号

三种基本符号 { 状态框  
判断框  
条件框

### 1. 状态框 (rectangle)



在T1 (代码001) 状态下, 输出  $Z = 0$ , 下一个 *CLK* 到来, 数据处理器进行操作  $A+1$ 。



$A \leftarrow A+1$

状态框内的操作为**无条件**操作。是此状态下**将要实现**的操作, 将在下一个**CLK**到来时执行。

## 寄存器传输语言

Register Transfer Languages ( RTL )

广义

Register

Counter

FF

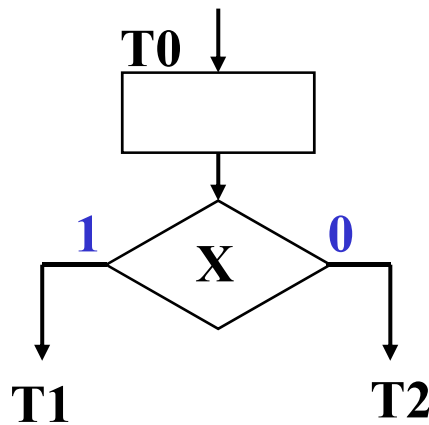
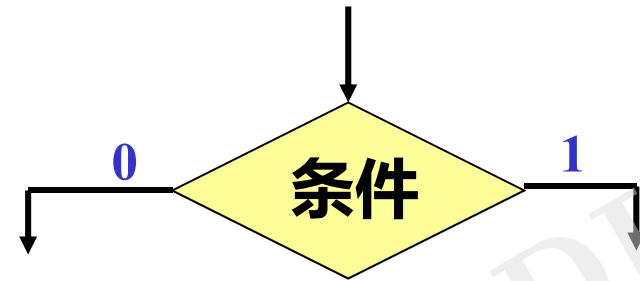
Memory

$R \leftarrow SR$     ( $R$  shift right)

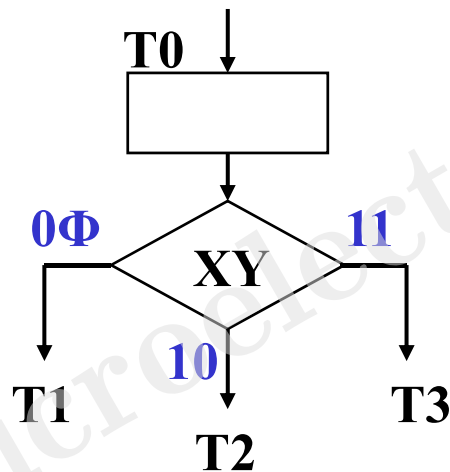
$A \leftarrow 0$     ( $A$  clear)

$F \leftarrow 1$     ( $F$  set 1)

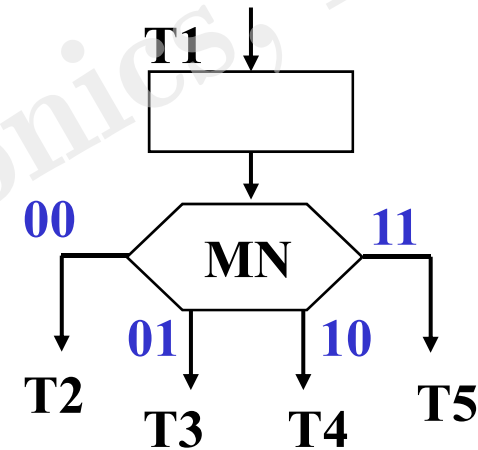
## 2. 判断框 (prism 菱形)



2 输出



3 输出



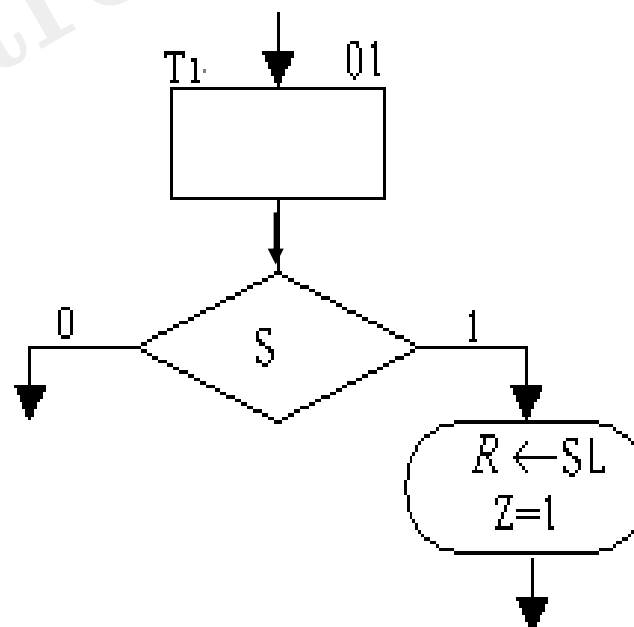
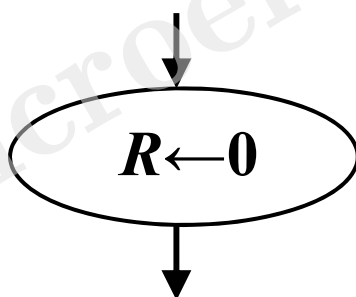
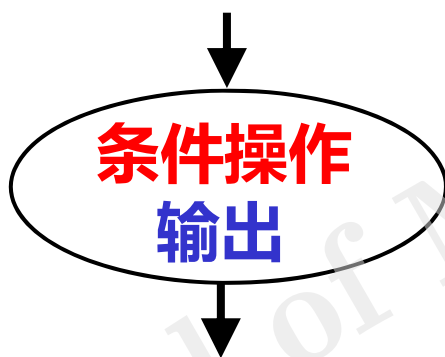
4 输出

控制器根据判断框内容(条件)决定下一个 *CLK* 到时  
状态转换

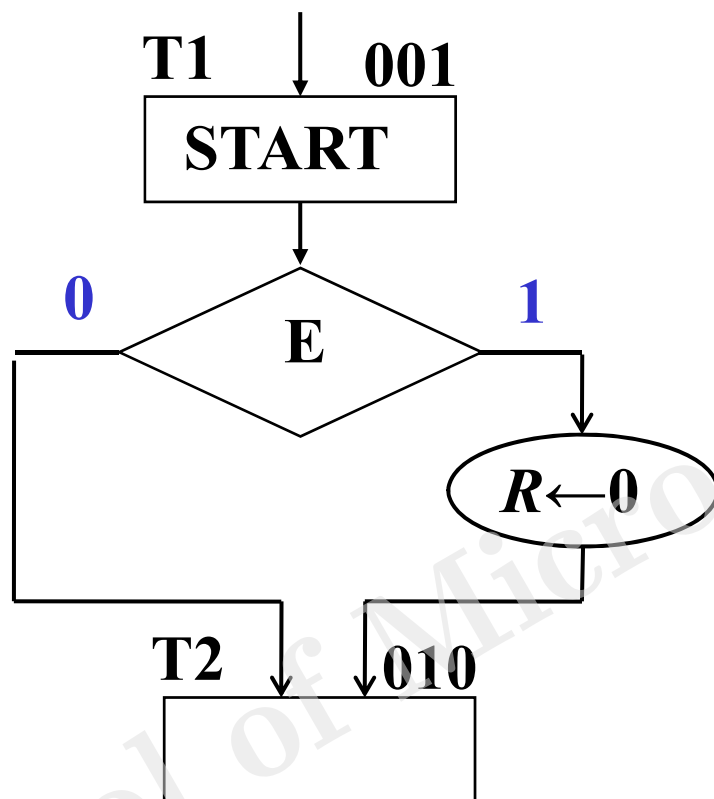
### 3. 条件框 (ellipse 椭圆)

条件框内的操作为**条件操作**

它的入口只能**接判断框**的分支



## 例1 分析下面 ASM图



在**T<sub>1</sub>**状态下(001),

输出 : **START**

若输入 **E = 1**,

下一个 **CLK** 到来

**R** 复位 (清0),

否则 **R** 保持,

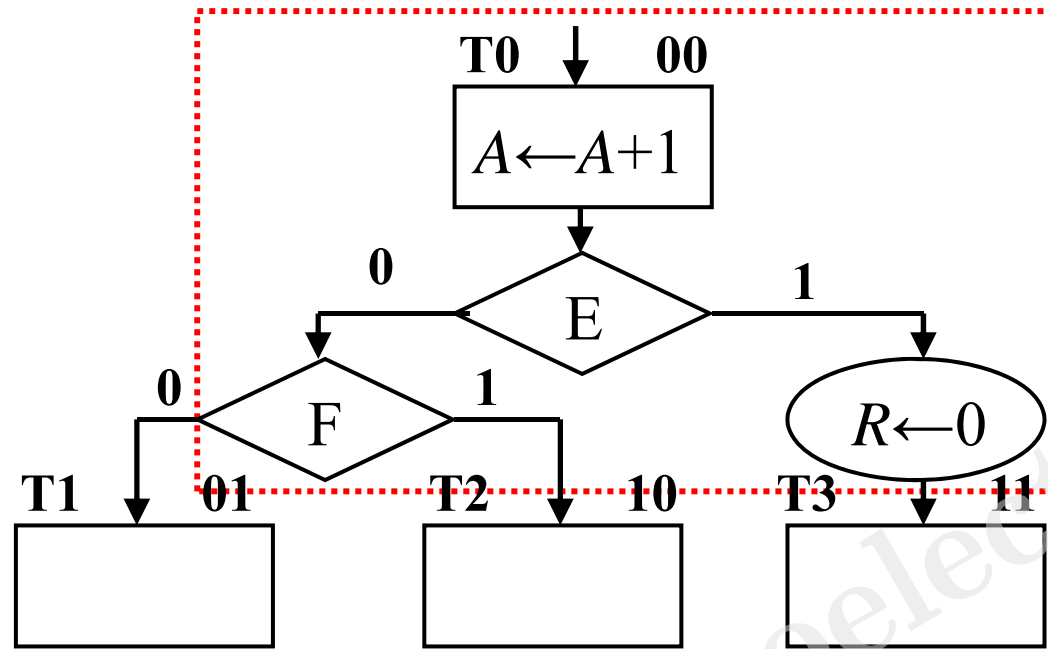
新状态为 **T<sub>2</sub>** (010)



## § 8.2.2 ASM块



**规则：** 每个ASM块必须包含只能包含一个状态框，以及与之相连的判定框和条件框。



## 例2 ASM块

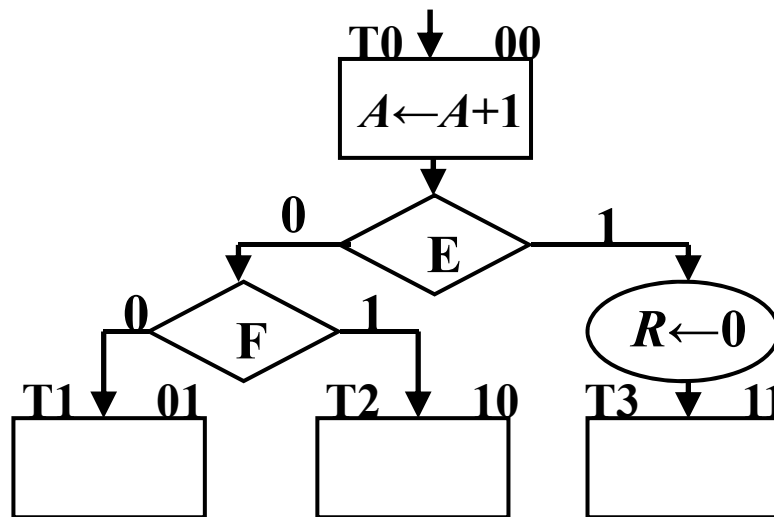
一个状态框

两个判定框

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, 下一个  $CLK$  到来:**

**数据处理器**

**控制器**

$A \leftarrow A + 1$

与下面  
三个操  
作中的  
一个同  
时完成

$A \leftarrow A + 1$   
(无条件操作)

若  $E = 1$ ,  $R \leftarrow 0$

(条件操作)

若  $E = 0, F = 0$ ,

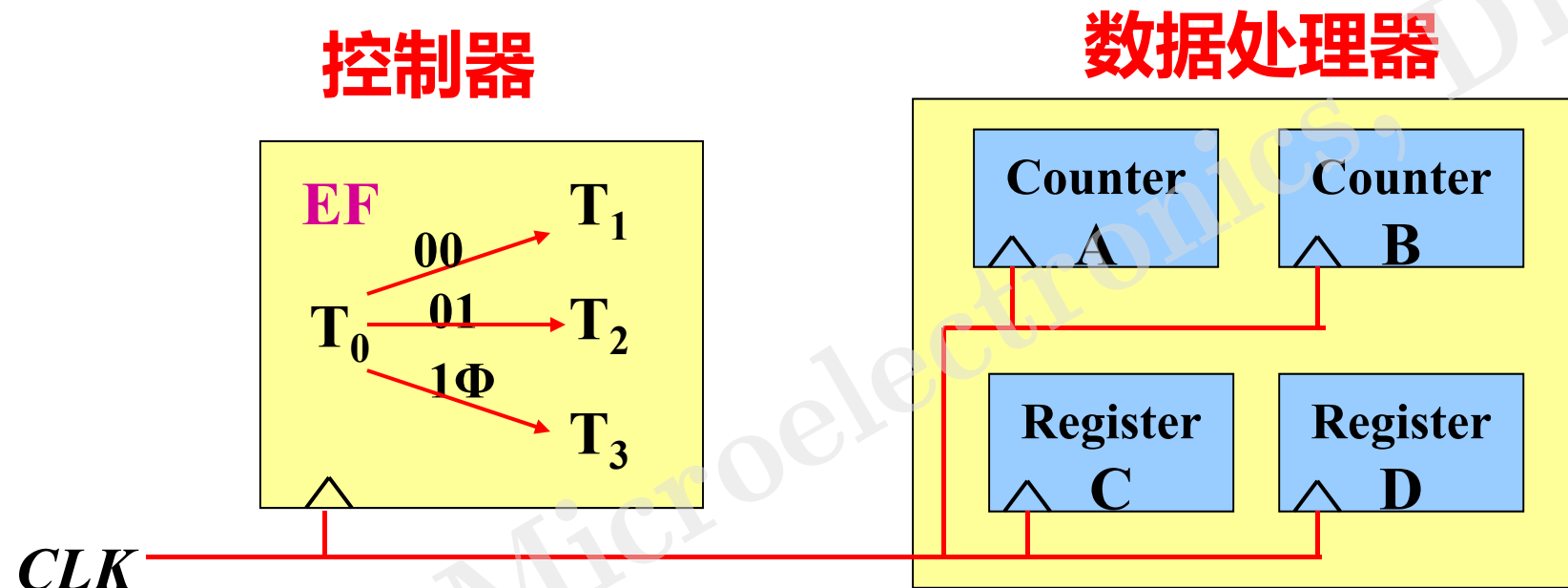
$E = 0, F = 1$ ,

状态  $T_0 \rightarrow T_3$

$T_0 \rightarrow T_1$

$T_0 \rightarrow T_2$

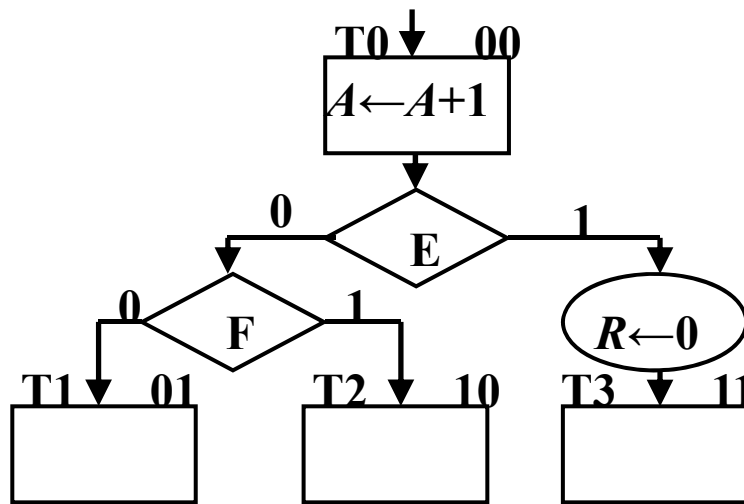
一个ASM块中的不同器件在一个 $CLK$ 内各自完成各自的操作 (同一器件不能同时做两件事)



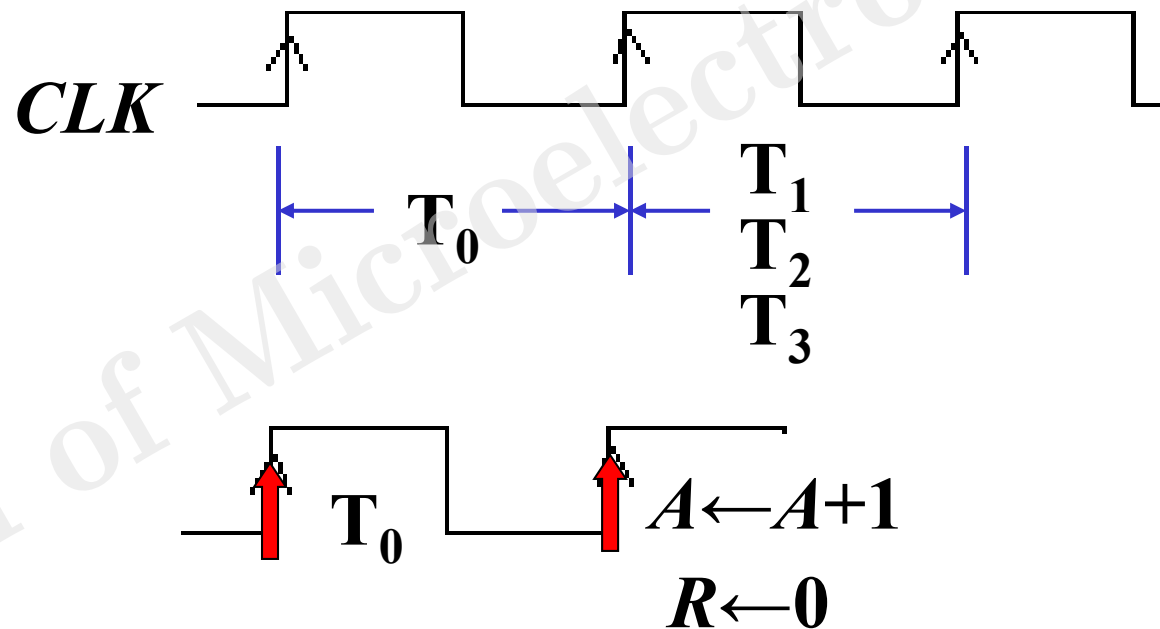
**E, F: 已知外输入**

**不是  $T_0$  完成后的第二步, 是同时判断并操作。**

**$T_0$  是在前一个  $CLK$  时形成的。**

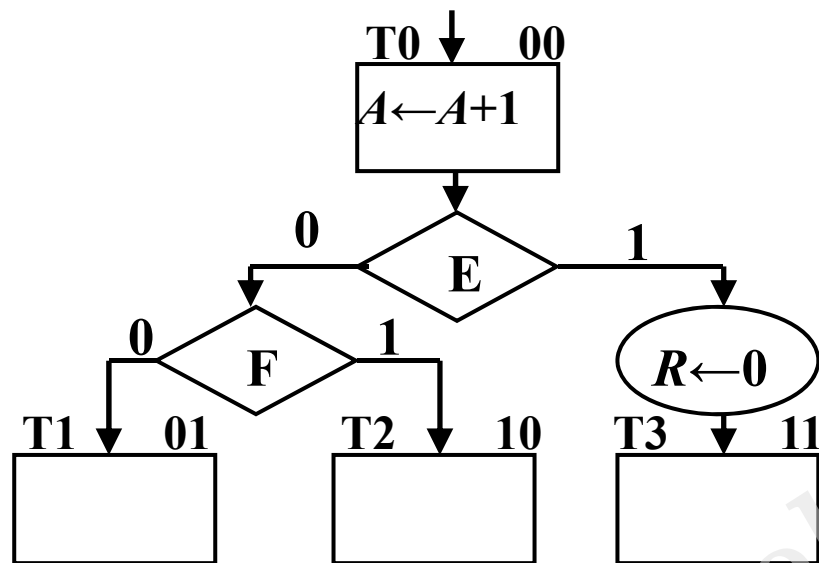


从波形图(时序图)  
加深理解

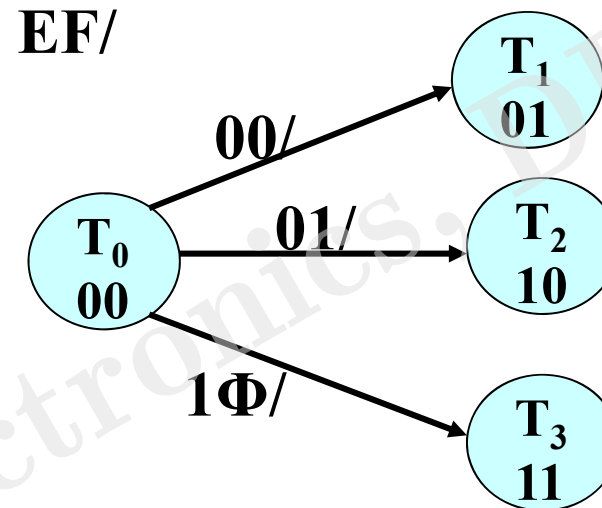


现态  $T_0$  与状态框内的操作不是在同一个 *CLK* 内

## ASM ~ 状态图的关系



## ASM → 状态图



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

相同

相异

## 例 1

需要几个 *CLK* ?

**3 *CLK*s**

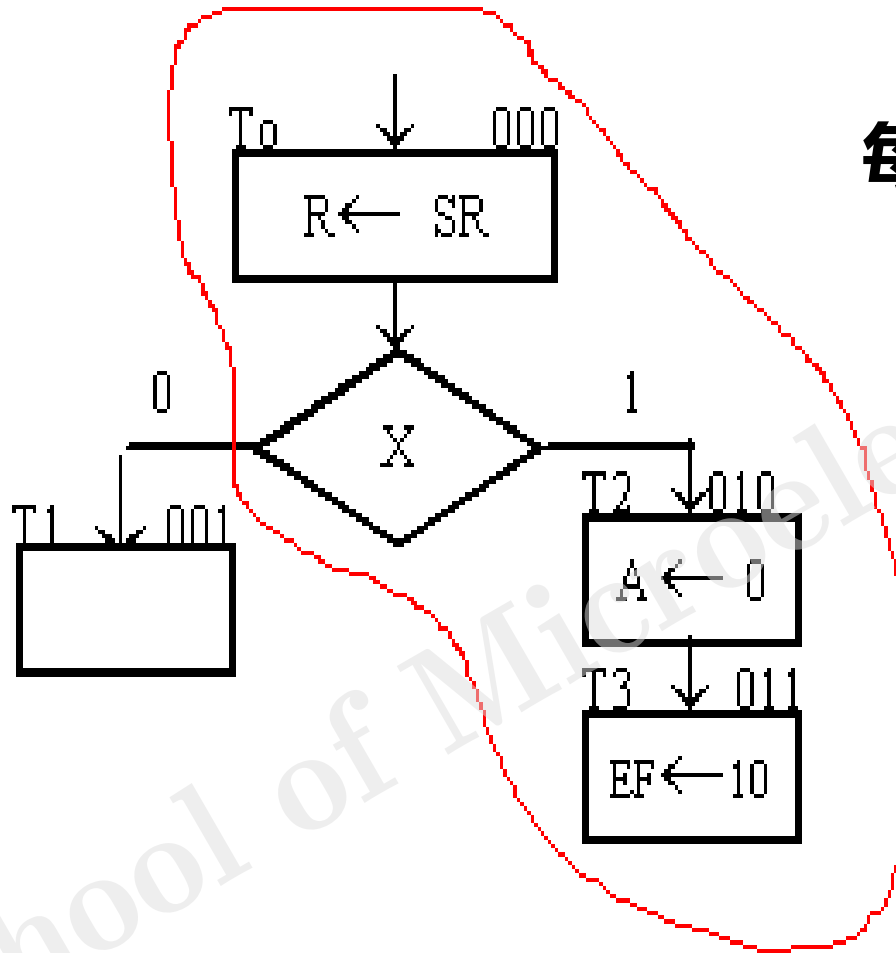
每个ASM块一个*CLK*实现

**1<sup>st</sup> *CLK* : 实现  $T_0$**

**2<sup>nd</sup> *CLK* :  $R \leftarrow SR$ ,  
若  $X = 1$ ,  $T_0 \rightarrow T_2$**

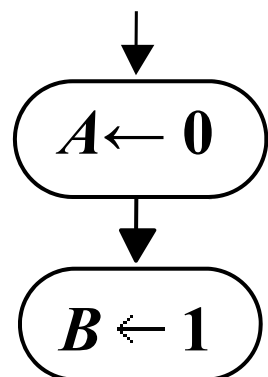
**3<sup>rd</sup> *CLK* :  $A \leftarrow 0$**

**$T_2 \rightarrow T_3$**



## 例 2 纠错

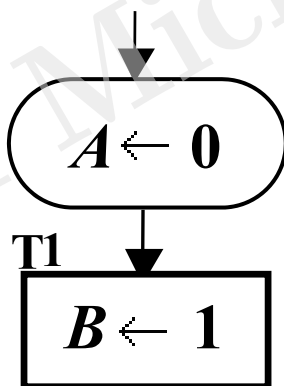
(1)



错:

条件框的入口只能接判断框

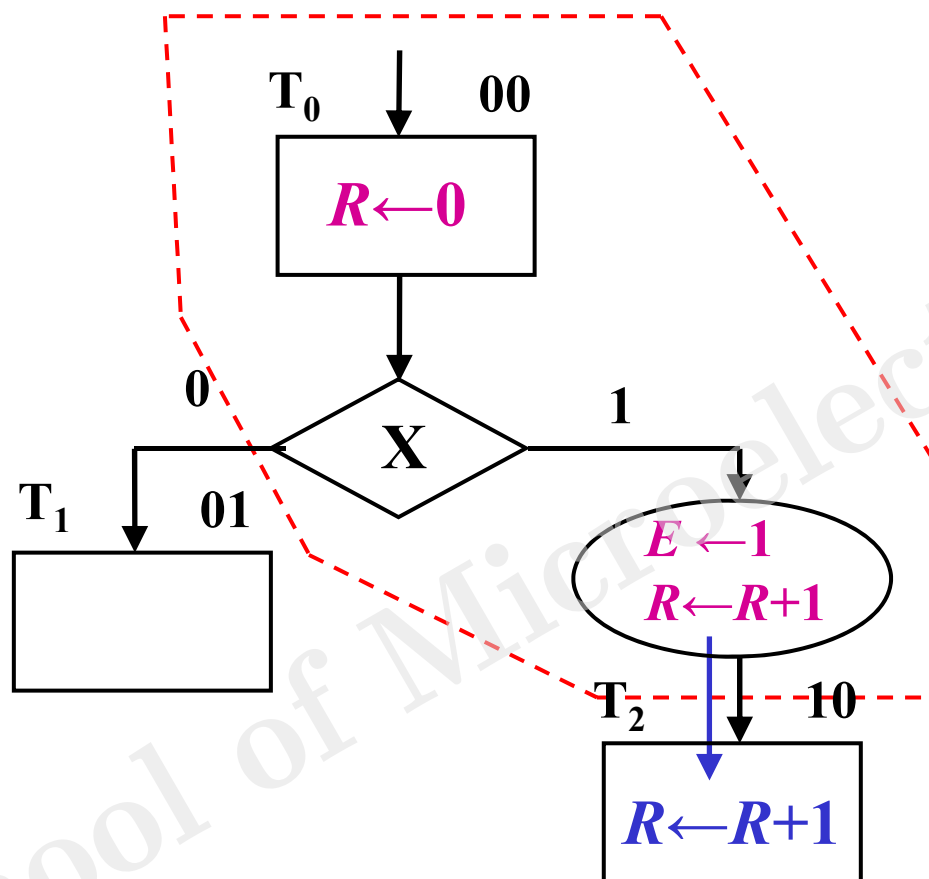
换成:



在两个  $CLK$  操作



(2)

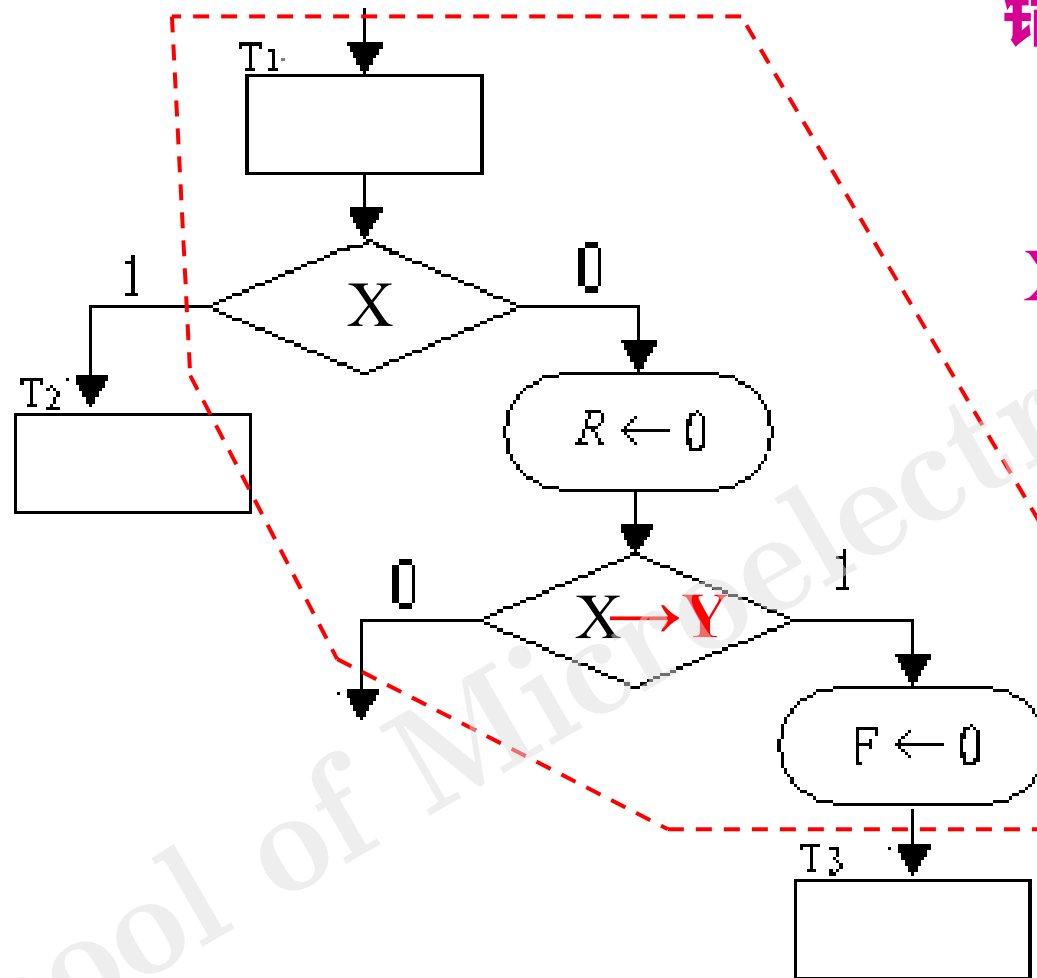


错:

在一个  $CLK$   
周期内  $R$  操作两  
次 (一个 ASM)

将一个操作移  
到另一个框内

(3)



错:

一个CLK周期内  
X 使用两次

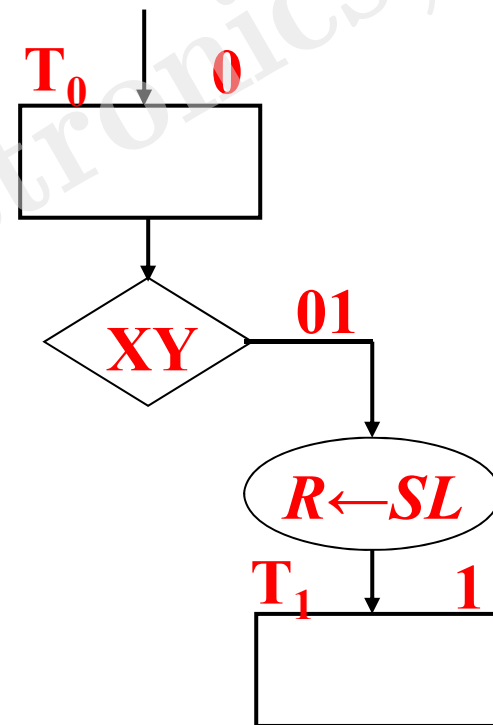
将一个 X 换成 Y

### § 8.2.3 ASM图表的建立

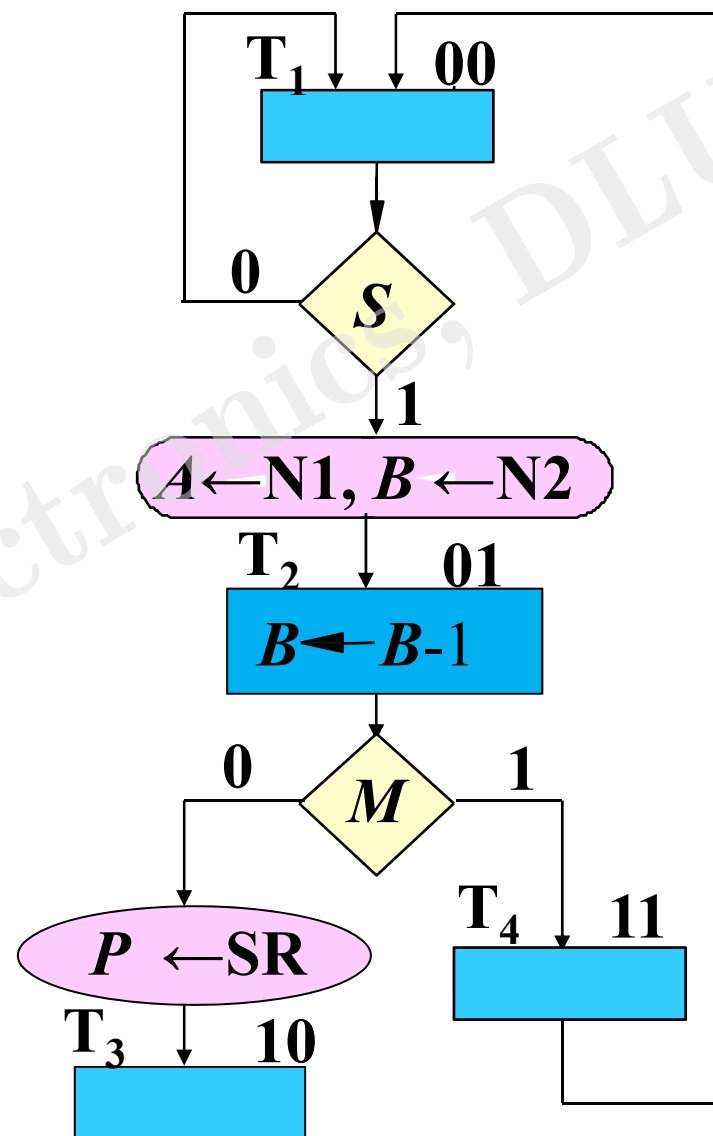
#### 例 1

在 $T_0$ 状态下, 若控制输入  $X$  和  $Y$  分别等于0和1, 系统实现条件操作: 寄存器  $R$  左移, 并转移到状态 $T_1$ , 试画出其ASM图。

ASM:



**例 2** 一个数字系统在 $T_1$ 状态下，若启动信号 $S=0$ ，保持 $T_1$ 状态不变；若 $S=1$ ，则完成条件操作： $A \leftarrow N1$ ， $B \leftarrow N2$ ，状态由 $T_1 \rightarrow T_2$ 。在 $T_2$ 状态下，下一个 $CLK$ 到，完成无条件操作 $B \leftarrow B - 1$ ，若 $M=0$ ，则完成条件操作： $P$ 右移，状态由 $T_2 \rightarrow T_3$ ；若 $M=1$ ，状态由 $T_2 \rightarrow T_4 \rightarrow T_1$ 。画出该数字系统的ASM图。



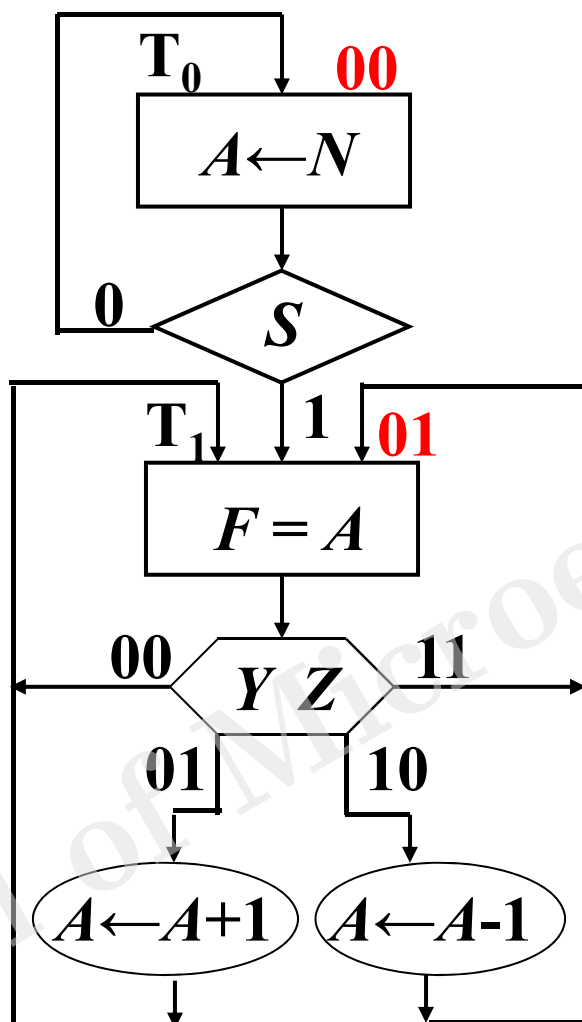
**例3** 用数字系统记录并显示车场内的存车数目, 入口出口都有光电元件, 每当有汽车进入车场时, 光线有变化, 信号 $Y$  由 $1 \rightarrow 0$ ; 汽车离开车场时, 出口信号 $Z$  由 $1 \rightarrow 0$ ; 信号 $Y, Z$  与时钟同步, 记录车场车辆数目的数据处理器是一可逆计数器, 画出该数字系统的ASM图表。

**光电传感器**  $\left\{ \begin{array}{l} \text{入, } Y \left\{ \begin{array}{l} Y = 1 \text{ 无车进入} \\ Y = 0 \text{ 有车进入} \end{array} \right. \\ \text{出, } Z \left\{ \begin{array}{l} Z = 1 \text{ 无车出} \\ Z = 0 \text{ 有车出} \end{array} \right. \end{array} \right.$

**设  $N$ : 车场内目前的车辆数**

**$S$ : 开始信号**  $\left\{ \begin{array}{l} S = 1 \text{ 开始} \\ S = 0 \text{ 保持} \end{array} \right.$

## ASM



$A$ : 计数器

$N$ : 车场内现有车辆数

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

$F$ : 输出

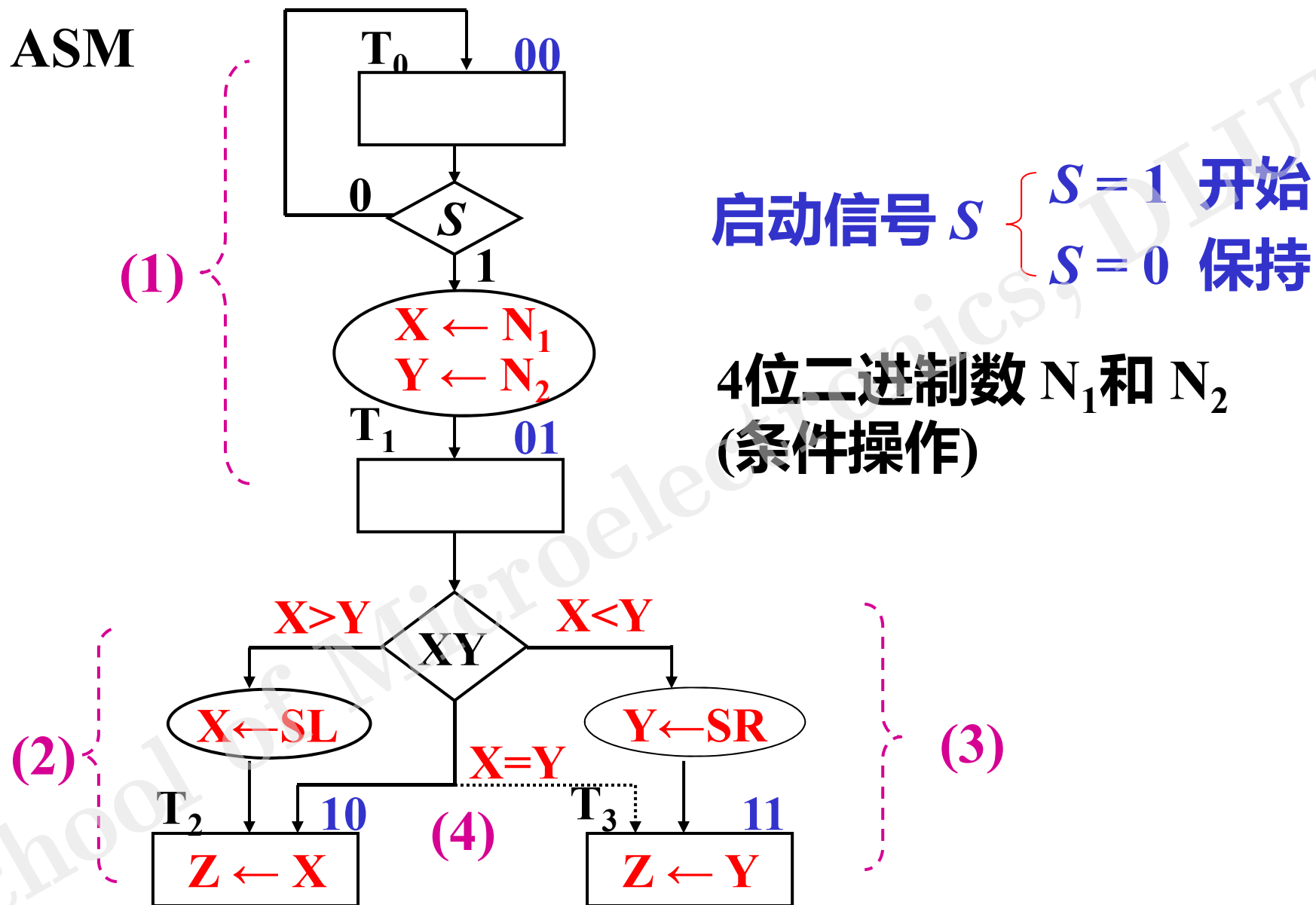
入,  $Y$   $\begin{cases} Y = 1 & \text{无车入} \\ Y = 0 & \text{有车入} \end{cases}$

出,  $Z$   $\begin{cases} Z = 1 & \text{无车出} \\ Z = 0 & \text{有车出} \end{cases}$

**例4** 设计一个数字系统，它有三个4位的寄存器 $X$ 、 $Y$ 、 $Z$ ，并实现下列操作：

- (1) 启动信号 $S$  出现, 将两个4 位二进制数 $N_1$ 、 $N_2$  分别传送给寄存器  $X$ 、 $Y$ ;
- (2) 如果  $X > Y$ , 左移 $X$ , 结果送给 $Z$ ;
- (3) 如果  $X < Y$ , 右移 $X$ , 结果送给 $Z$ ;
- (4) 如果  $X = Y$ , 将  $X$  或  $Y$  送给  $Z$ 。

ASM





## § 8.3 数字系统设计

### § 8.3.1 数字系统设计步骤

1. 分析

2. ASM

3. 设计控制器

状态转换

4. 设计数据处理器

条件操作和无条件操作

5. 电路

## § 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 16 s，香蕉形图案灯亮 12 s，葡萄形图案灯亮 9 s。

### 1. 分析

输入

计时信号  $\begin{cases} 16\text{ s} : X=1 \\ 12\text{ s} : Y=1 \\ 9\text{ s} : Z=1 \end{cases}$

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

输出

灯亮

苹果形 :  $A = 1$

香蕉形 :  $B = 1$

葡萄形 :  $G = 1$

逻辑高有效

## 2. 建立 ASM图

## 3. 设计控制器

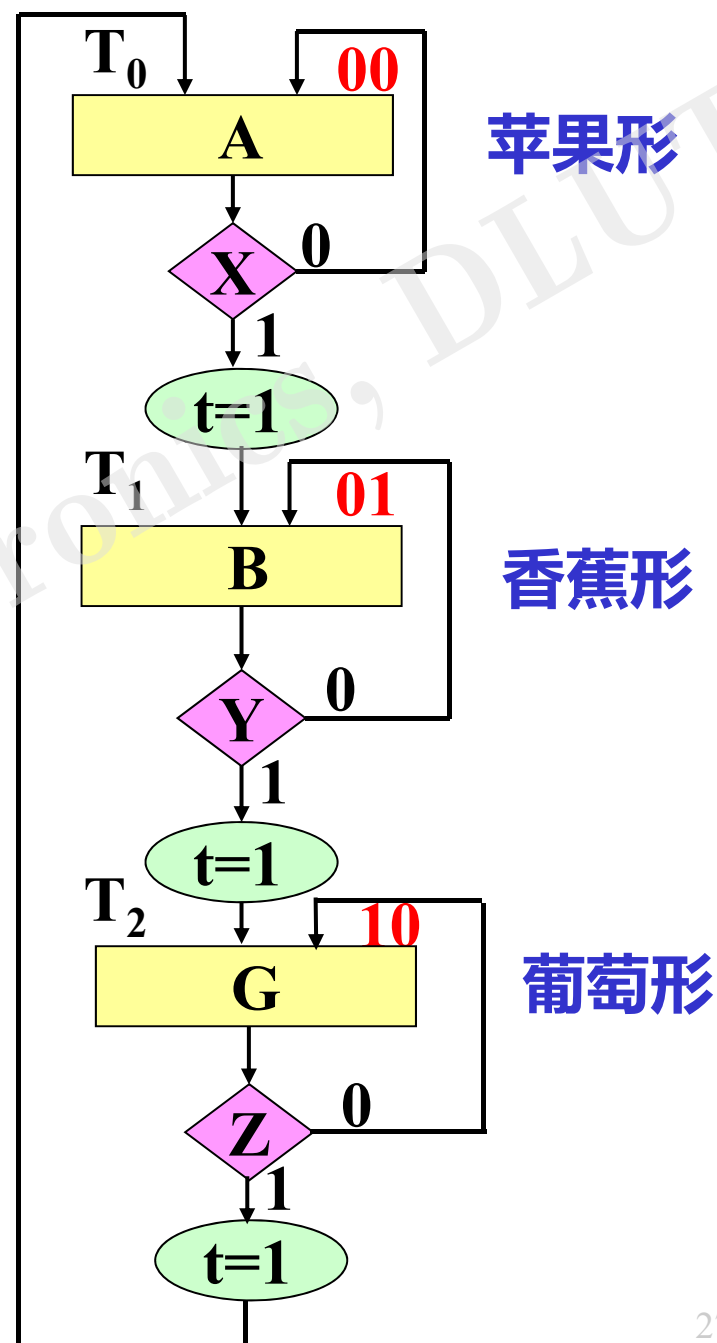
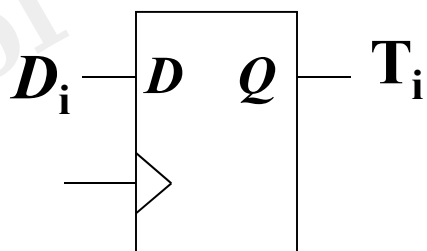
$$(X, Y, Z \rightarrow T_0, T_1, T_2)$$

方法 1:

每个状态一个触发器

状态数 = FFs

选择 D-FF,  $Q^{n+1} = D$



根据ASM图，各个状态的输入条件作为D-FF的控制输入方程。

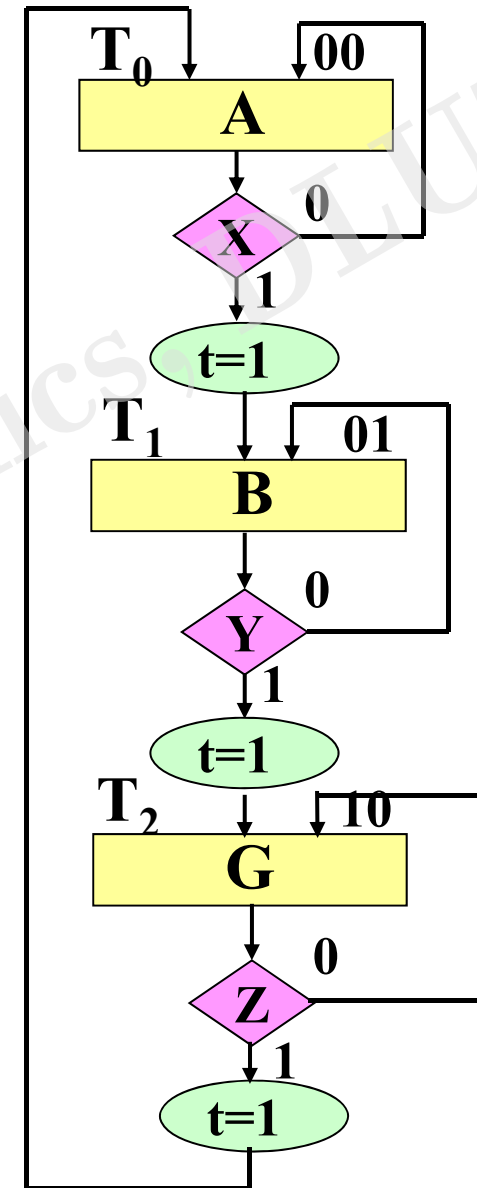
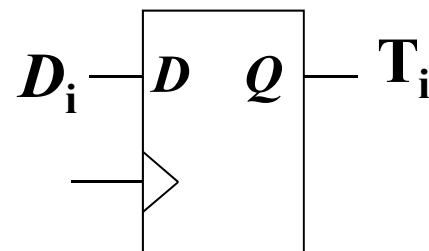
任何时刻，只能存在一个状态  
(= 1)，其它状态= 0

3 状态, 3 D-FF, 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

选择 3个 D-FF,

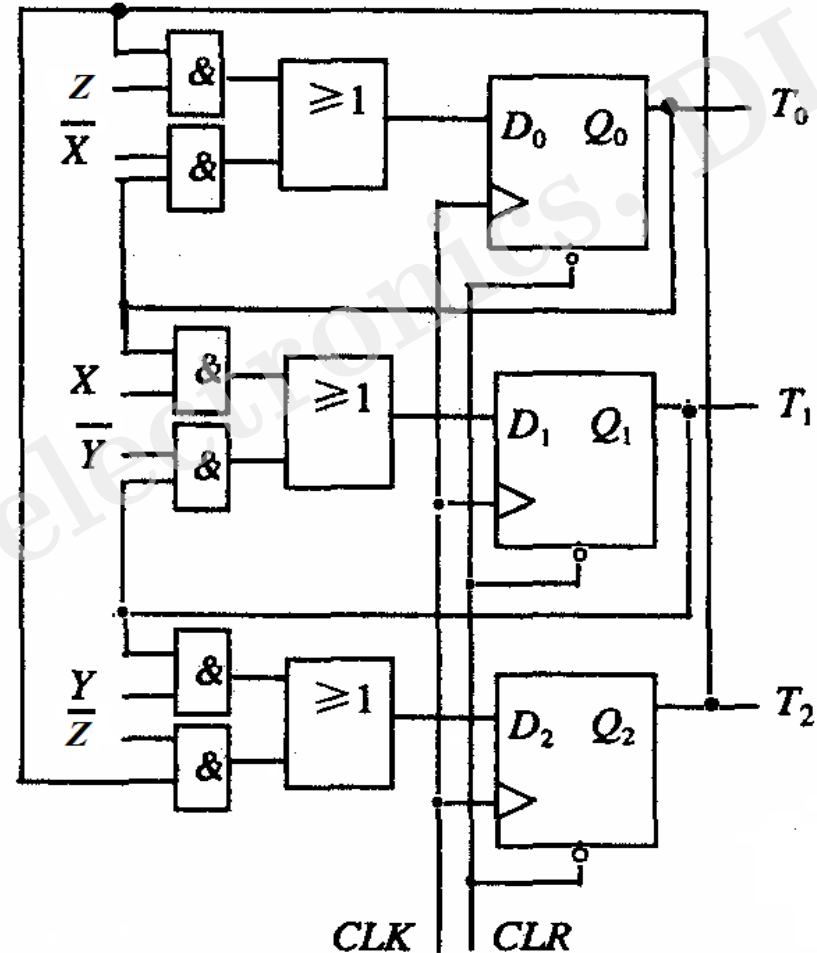
$$Q^{n+1} = D$$

输入  $D$ , 输出  $T_i$

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

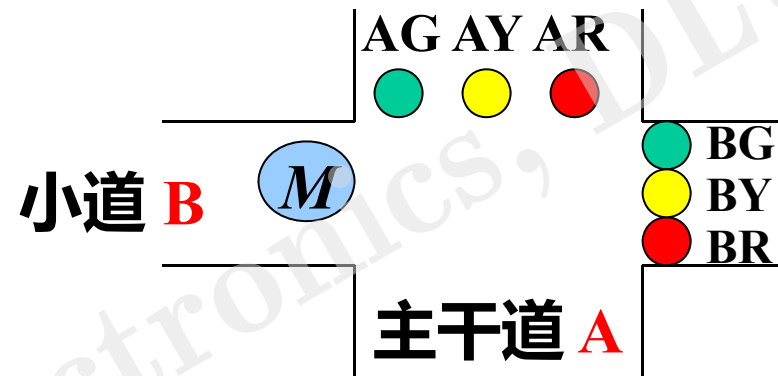
$$T_1=1, \quad D_1 = T_0 X + T_1 \bar{Y}$$

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



## 例 2 十字路口交通灯管理系统 (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  $M$   $\begin{cases} = 1 & \text{有车} \\ = 0 & \text{无车} \end{cases}$

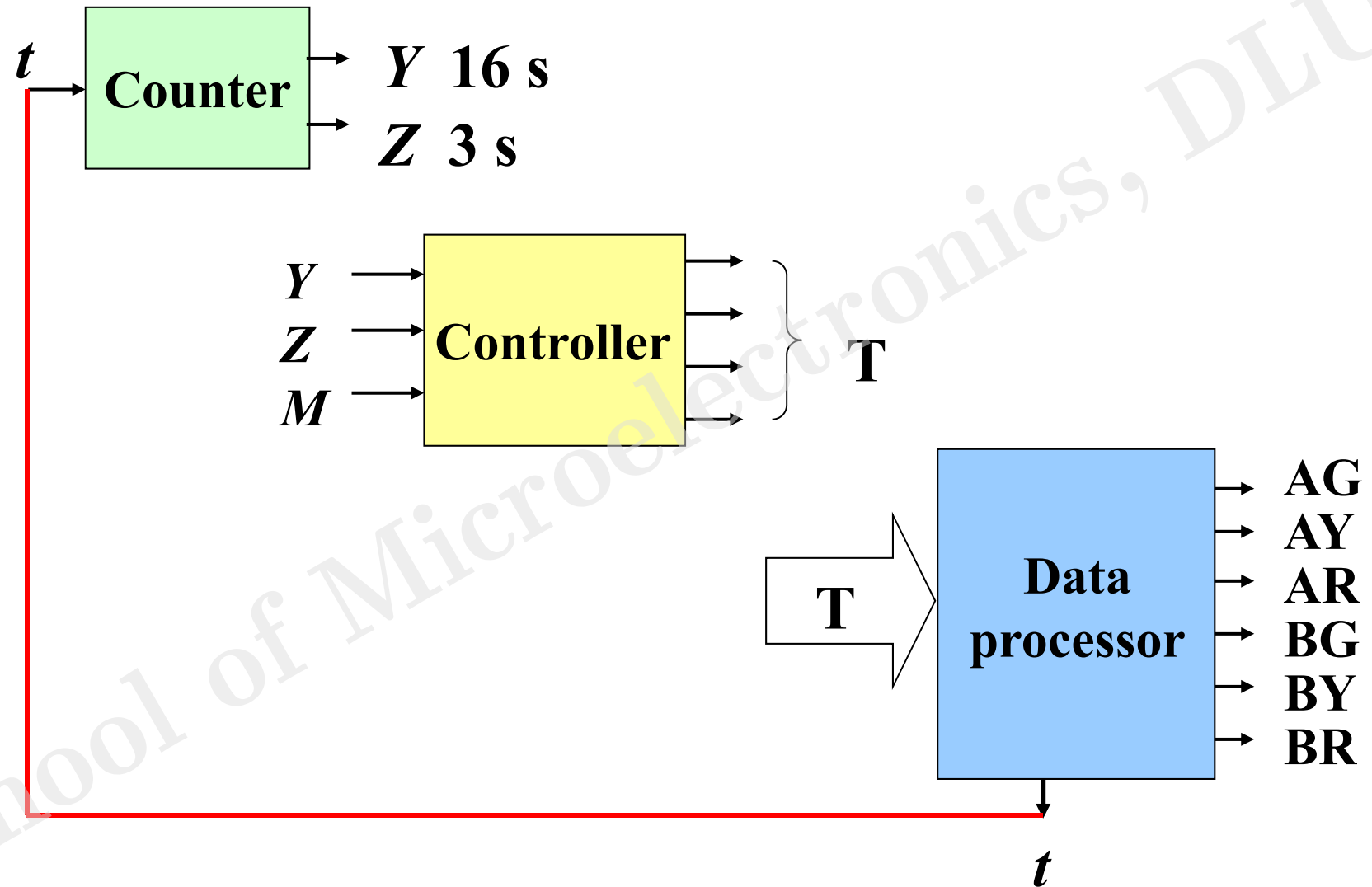
定时启动  $t$   $\begin{cases} = 1 & \text{计时开始} \\ = 0 & \text{否} \end{cases}$

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

输出:  $AG, AY, AR, BG, BY, BR = 1$  亮

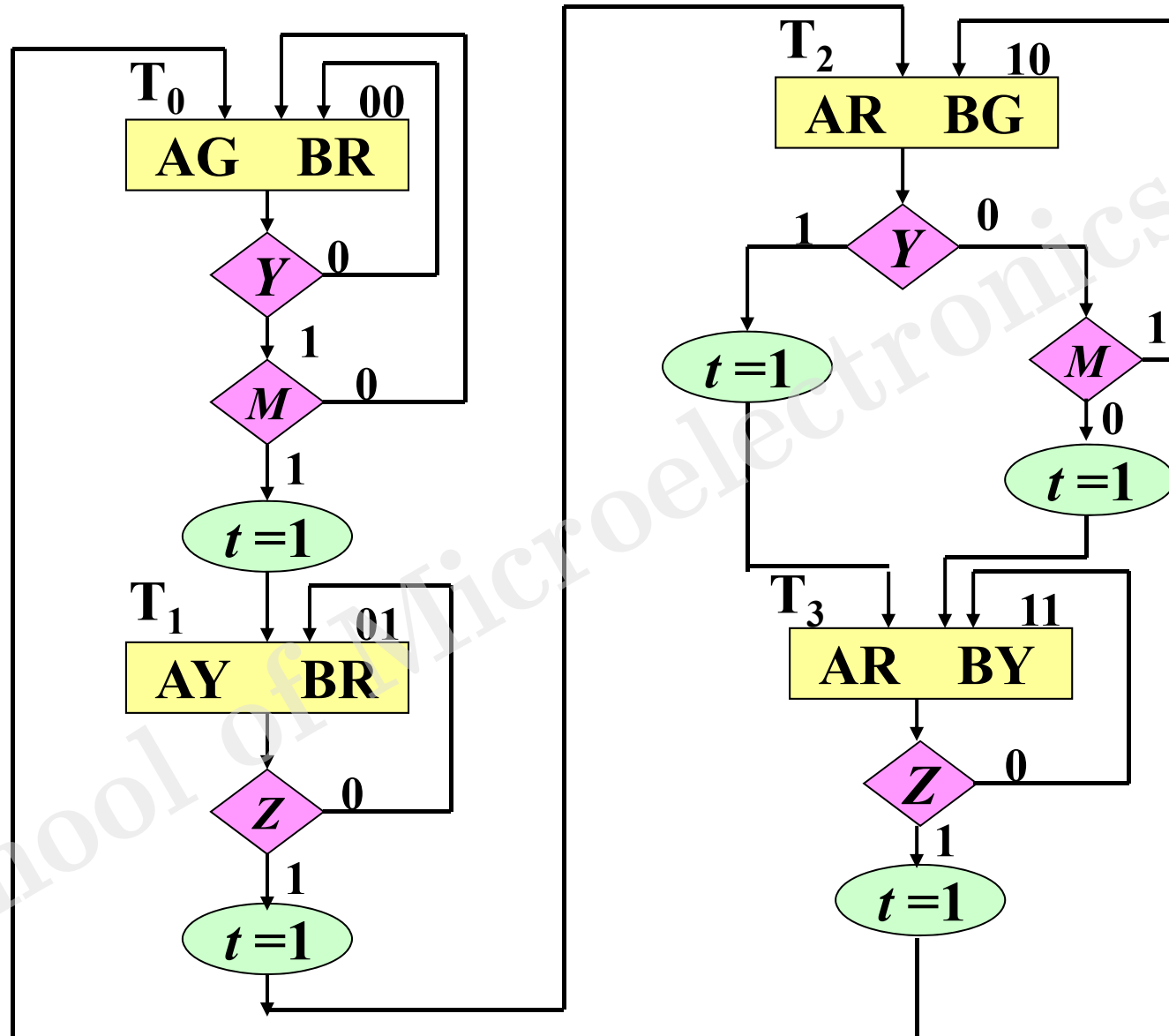
输入、输出均是高电平有效 (=1)

## 系统需要几部分:





## 2. 建立ASM图



四个状态  
两个FF

### 3. 控制器设计

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

方法2：用 MUX, D-FF, 译码器设计控制器

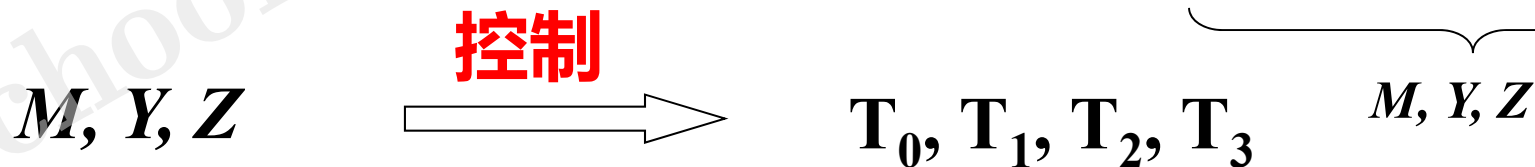
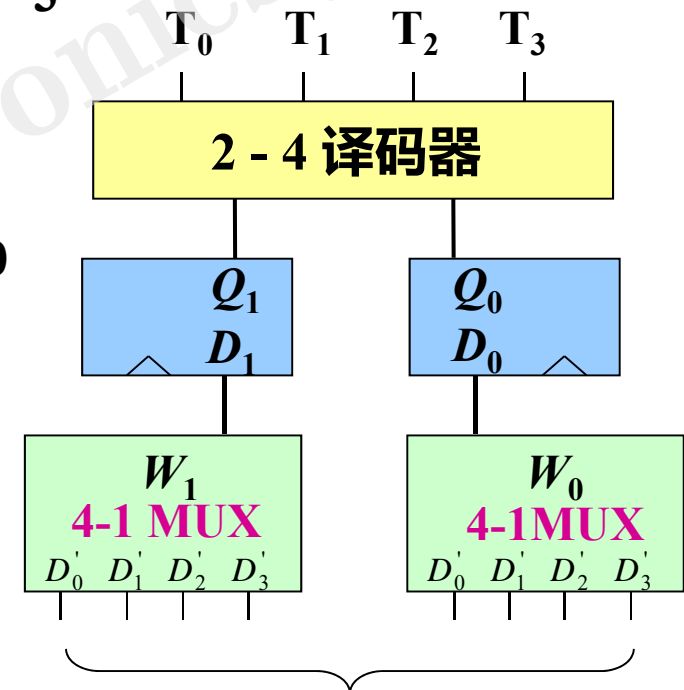
根据 ASM：4 个状态  $T_0, T_1, T_2, T_3$

输出：高电平有效 2-4 译码器

其入口接两个 D-FF 的出口  $Q_1, Q_0$

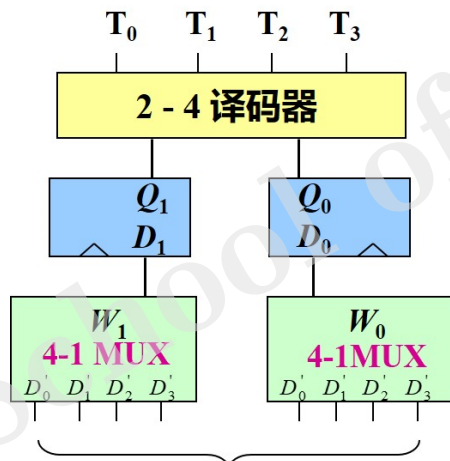
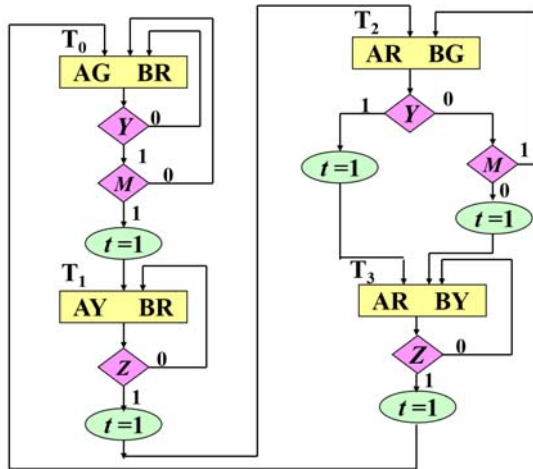
D-FF 的入口各接一个 4-1 MUX

4-1 MUX 入口接  $M, Y, Z$ , 实现



状态表, 找到  $Q_1^{n+1}, Q_0^{n+1}$  (即  $D_1, D_0$ ) 与输入  $M, Y, Z$  的关系

从ASM图:



$M, Y, Z$

状态符号	现状态 $Q_1^n Q_0^n$	输入 $Y Z M$	新状态 $Q_1^{n+1} Q_0^{n+1}$	输出 $T_0 T_1 T_2 T_3$
$T_0$	0 0	$\begin{matrix} 0 & \Phi & \Phi \\ 1 & \Phi & 0 \\ 1 & \Phi & 1 \end{matrix}$	$\left\{ \begin{matrix} 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{matrix} \right\} \begin{matrix} \\ YM \\ \end{matrix}$	1 0 0 0
$T_1$	0 1	$\begin{matrix} \Phi & 0 & \Phi \\ \Phi & 1 & \Phi \end{matrix}$	$\left\{ \begin{matrix} 0 & 1 \\ 1 & 0 \end{matrix} \right\} \begin{matrix} \\ \bar{Z} \\ \end{matrix}$	0 1 0 0
$T_2$	1 0	$\begin{matrix} 0 & \Phi & 1 \\ 0 & \Phi & 0 \\ 1 & \Phi & \Phi \end{matrix}$	$\left\{ \begin{matrix} 1 & 0 \\ 1 & 1 \\ 1 & 1 \end{matrix} \right\} \begin{matrix} \\ Y + \bar{Y} \cdot \bar{M} \\ Y + \bar{M} \end{matrix}$	0 0 1 0
$T_3$	1 1	$\begin{matrix} \Phi & 0 & \Phi \\ \Phi & 1 & \Phi \end{matrix}$	$\left\{ \begin{matrix} 1 & 1 \\ 0 & 0 \end{matrix} \right\} \begin{matrix} \\ \bar{Z} \\ \end{matrix}$	0 0 0 1

4-1MUX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 K-map (VEM), 把  $M, Y, Z$  作引入变量。

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

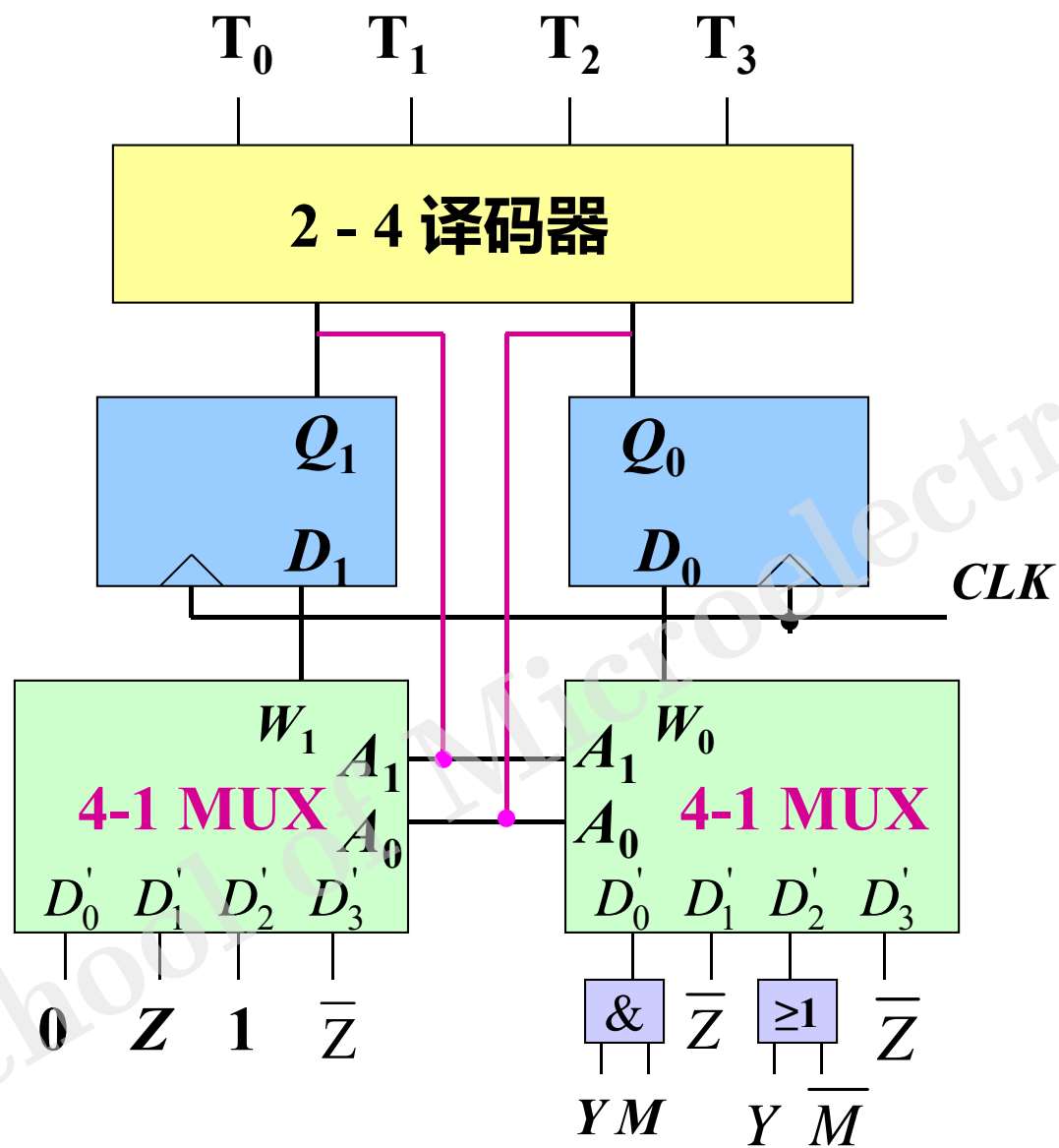
$D_1$

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

$D_0$

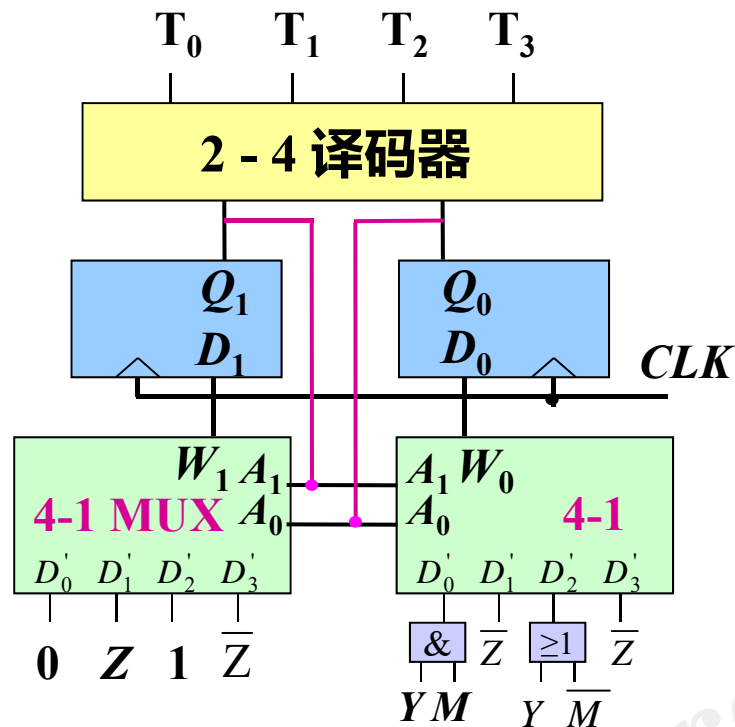
四个小格分别为4-1MUX的输入端变量

# 电路



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

$Q_0^{n+1} \backslash Q_0^n$	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  $D_0'$ )

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

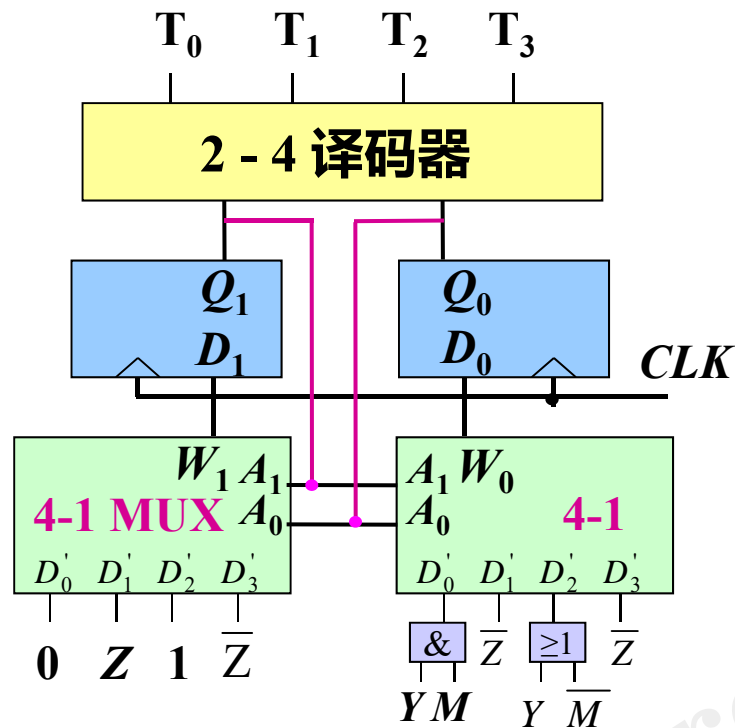
下一个  $CLK$  到来,

$Q_1^{n+1} = D_1 = 0$ ,  $Q_0^{n+1} = D_0 = MY$

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

当  $MY = 1$  (支路有车, 16 s 到),  $Q_1Q_0 = 01$ ,

输出新状态 **T<sub>1</sub> (0100)**



**$T_1$  状态** ( $T_0T_1T_2T_3 = 0100$ ),

$Q_1Q_0 = 01$ , 使  $A_1A_0 = 01$ ,  
(4-1选  $D_1'$ )

下一个  $CLK$  到来,

$Q_1^{n+1} = D_1 = Z$ ,  $Q_0^{n+1} = D_0 = \bar{Z}$

若  $Z = 0$ , 即黄灯3s未到,  $\bar{Z} = 1$ ,  $Q_1Q_0 = 01$ , 保持  $T_1$

若  $Z = 1$ , 即黄灯3s到,  $\bar{Z} = 0$ ,  $Q_1Q_0 = 10$ , 进入  $T_2$  状态  
(0010)

## 4. 数据处理器设计

### (1) 灯电路

$$AG = T_0$$

$$AY = T_1$$

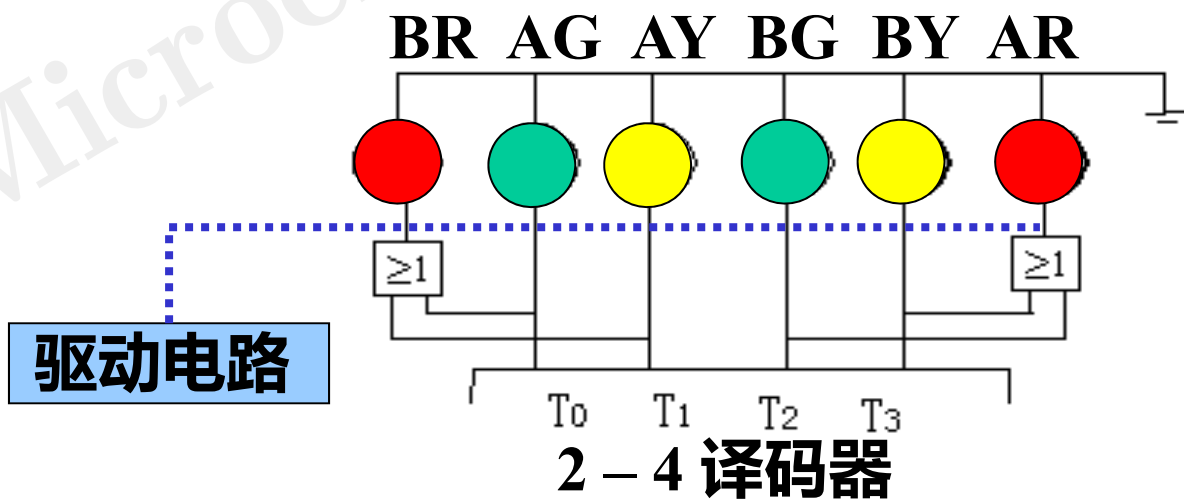
$$AR = T_2 + T_3$$

$$BG = T_2$$

$$BY = T_3$$

$$BR = T_0 + T_1$$

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





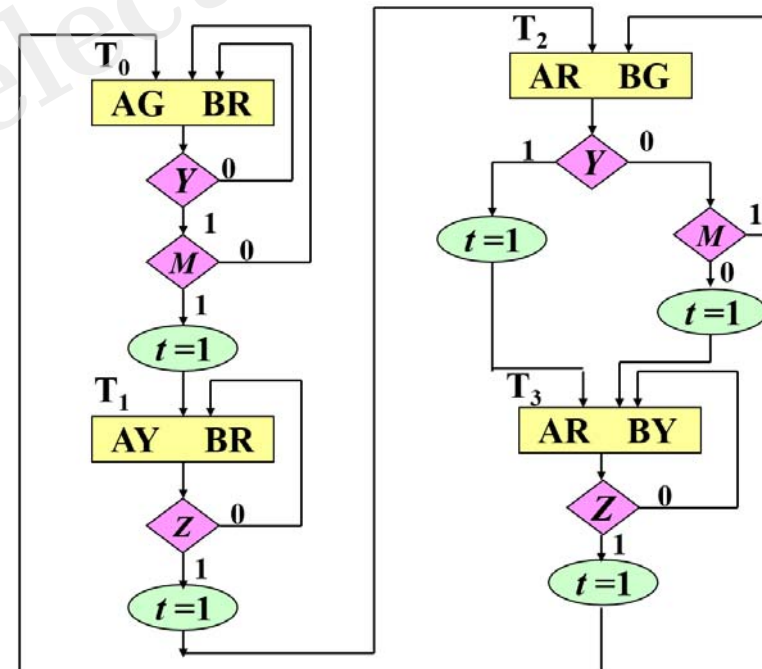
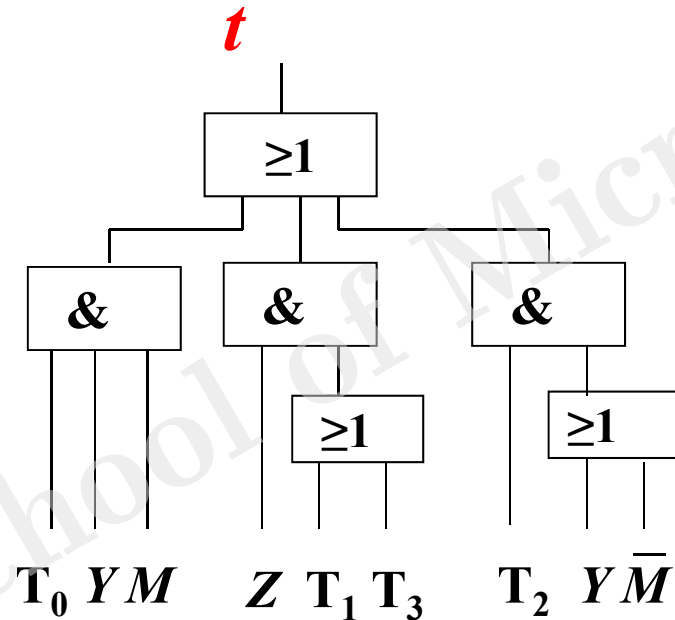
## (2) 定时启动电路 (产生 $t = 1$ )

定时启动 :  $t = 1$

ASM :  $t = 1$  条件

$$t = T_0 Y M + T_1 Z + T_2 Y + T_2 \bar{Y} \bar{M} + T_3 Z$$

$$= T_0 Y M + (T_1 + T_3) Z + T_2 (Y + \bar{M})$$




### (3) 计时电路 (产生Y, Z)

2个计数器 Y : 16 s, Z : 3 s

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  $t$	Clear (清0)	0	$\Phi$	$\Phi$	$\Phi$	$\Phi$	$\Phi$	$\Phi$	$\Phi$
	启动 (预置)	1	0	1	1	0	0	0	0

$\overline{CLR}$   下降沿  
 $\overline{LD} = \overline{t}$   
 $D_3 D_2 D_1 D_0 = 0000$   
 $ENT = ENP = 1$

74161

$Q_3 Q_2 Q_1 Q_0 = 1111$   
 (CO = 1) 时, Y = 1 (M-16)  
 $Q_3 Q_2 Q_1 Q_0 = 0010$  时,  
 Z = 1 (M-3)

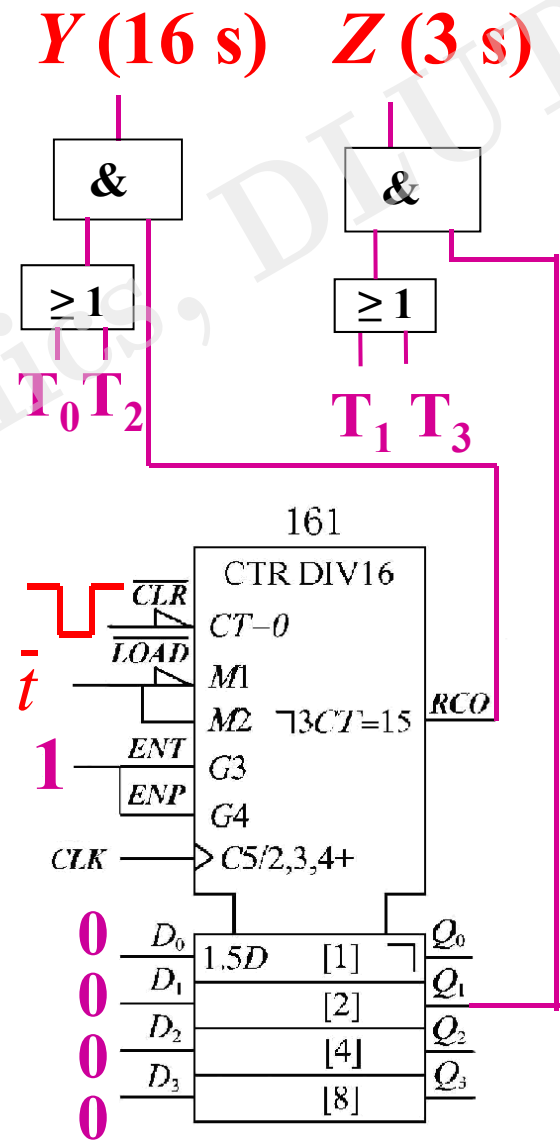
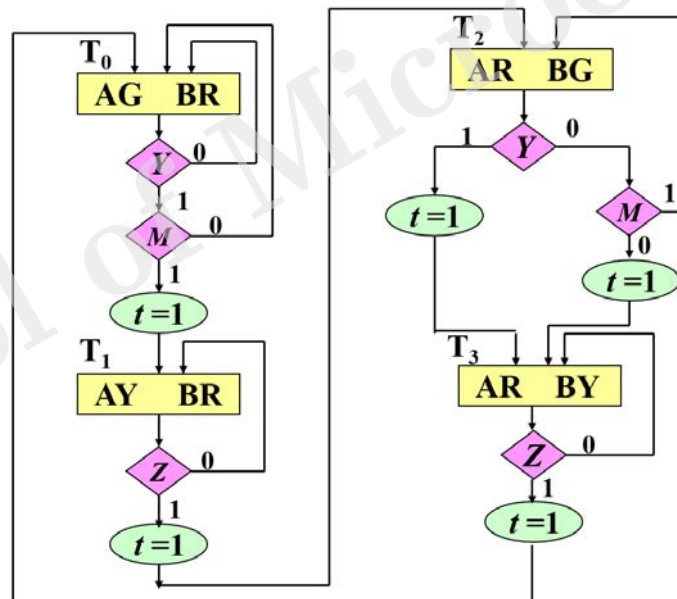
# Y 和 Z 输出 ASM 分析, 什么状态下 计时16 s, 3 s ( $t=1$ )?

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

即  $T_0$  或  $T_2$  状态下,  $CO = 1$  ( $Q_3Q_2Q_1Q_0 = 1111$ ),  $Y = 1$

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

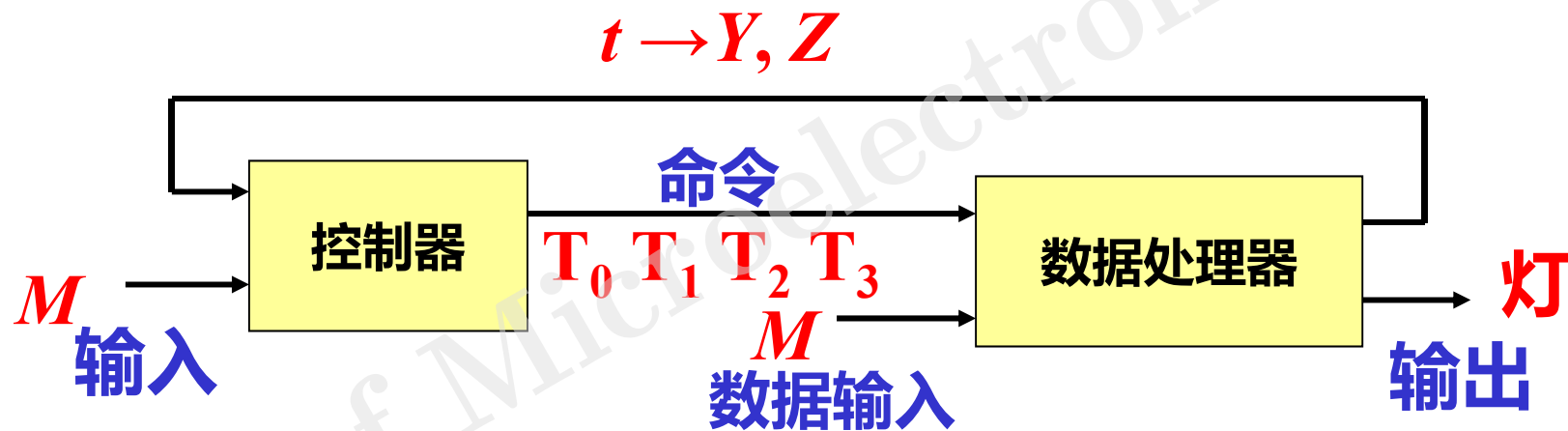
即  $T_1$  或  $T_3$  状态下,  $Q_3Q_2Q_1Q_0 = 0010$ ,  $Z = 1$





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 $\overline{R_D}$ ) 



# 作业

8 . 3

8 . 9

8 . 10