

# 第 8 章 数字系统设计基础

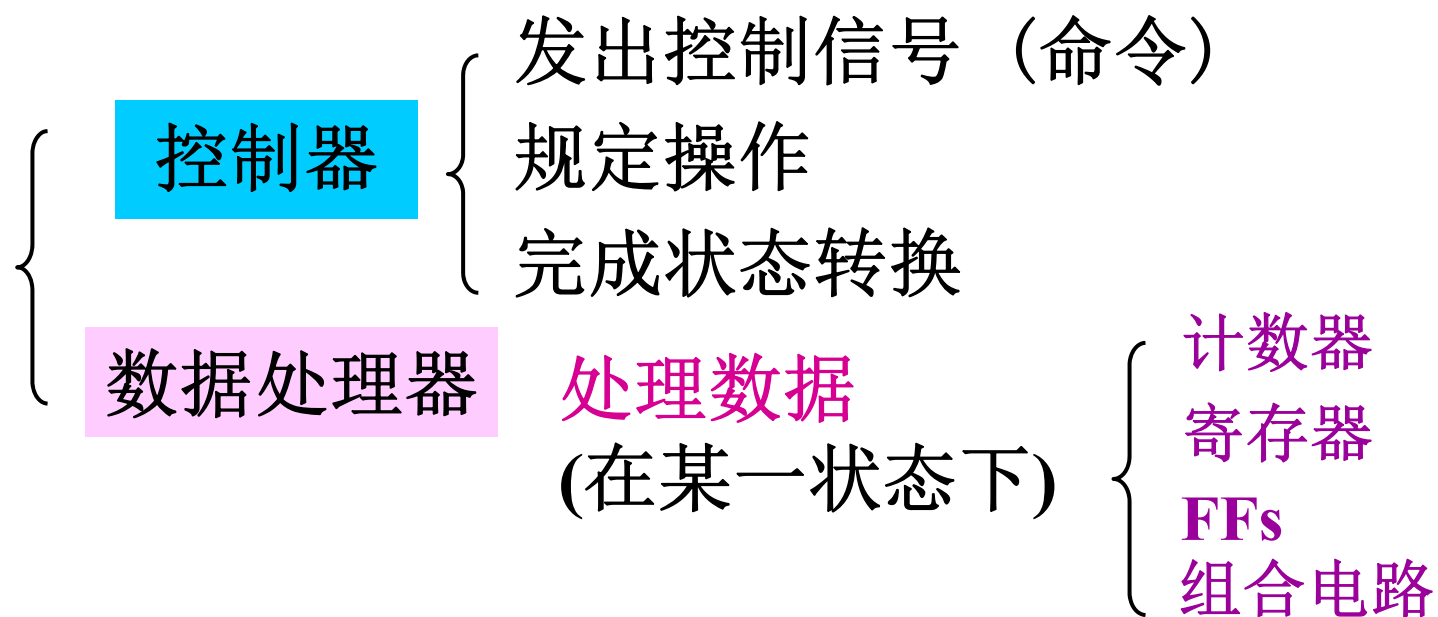
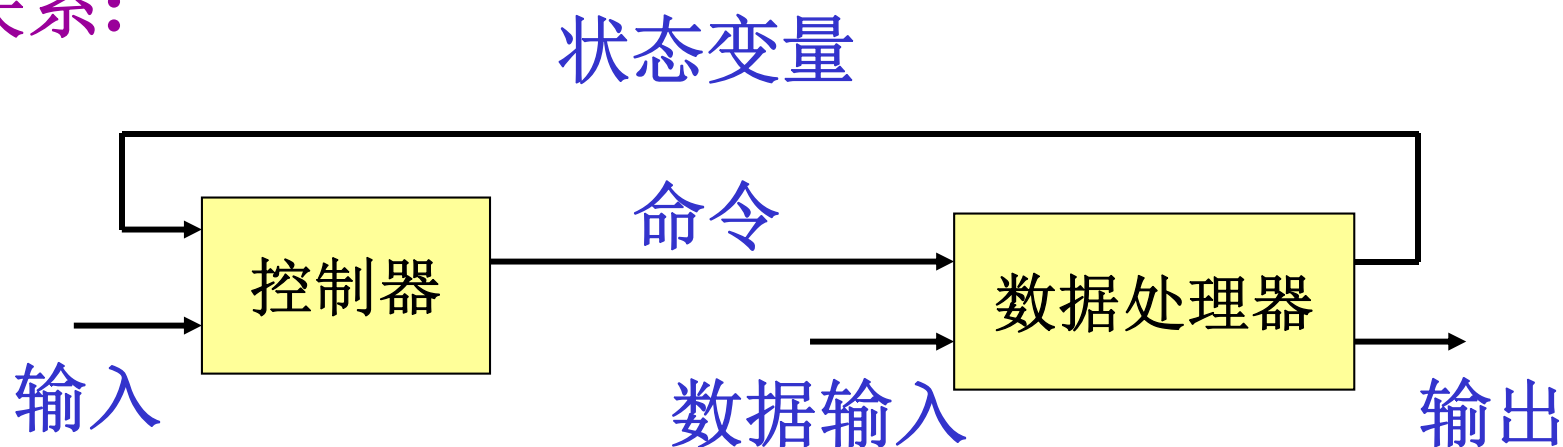
## Design of Digital Systems

### § 8.1 概述

数字系统  $\longrightarrow$  模块      子系统

每个子系统 {  
    控制器  
    数据处理器

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

### 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

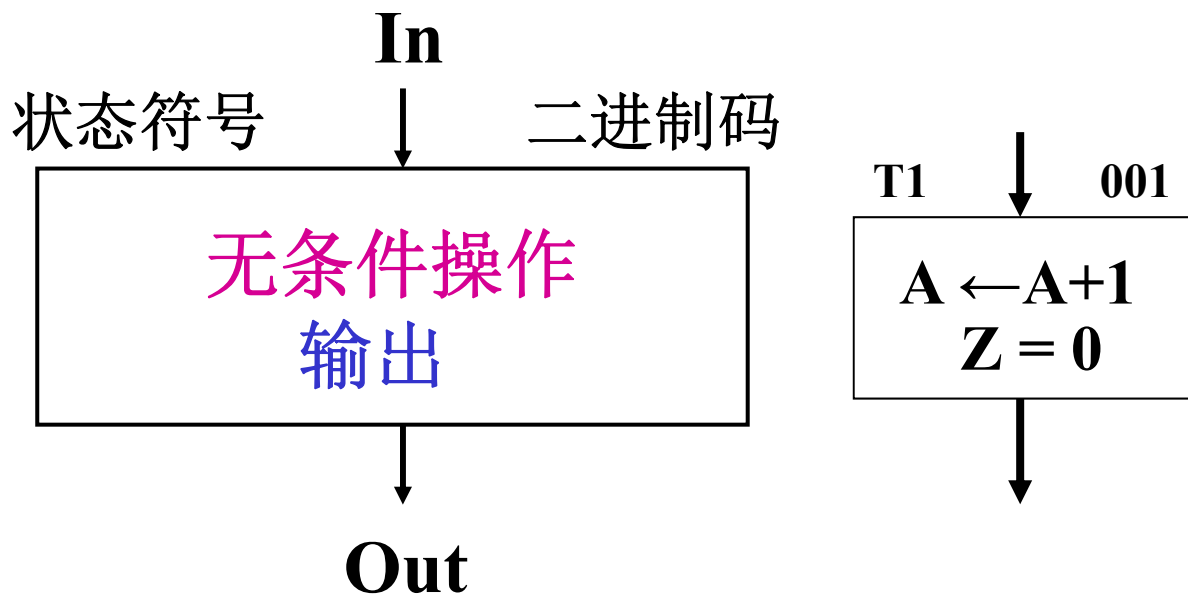
特点:

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）

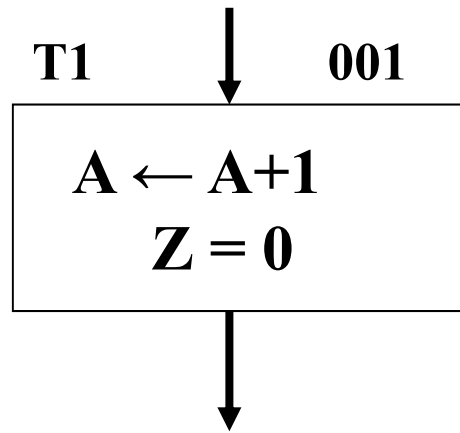
## 8.2.1 ASM 符号

三种基本符号 { 状态框  
判断框  
条件框

### 1. 状态框 (rectangle)



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



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 {

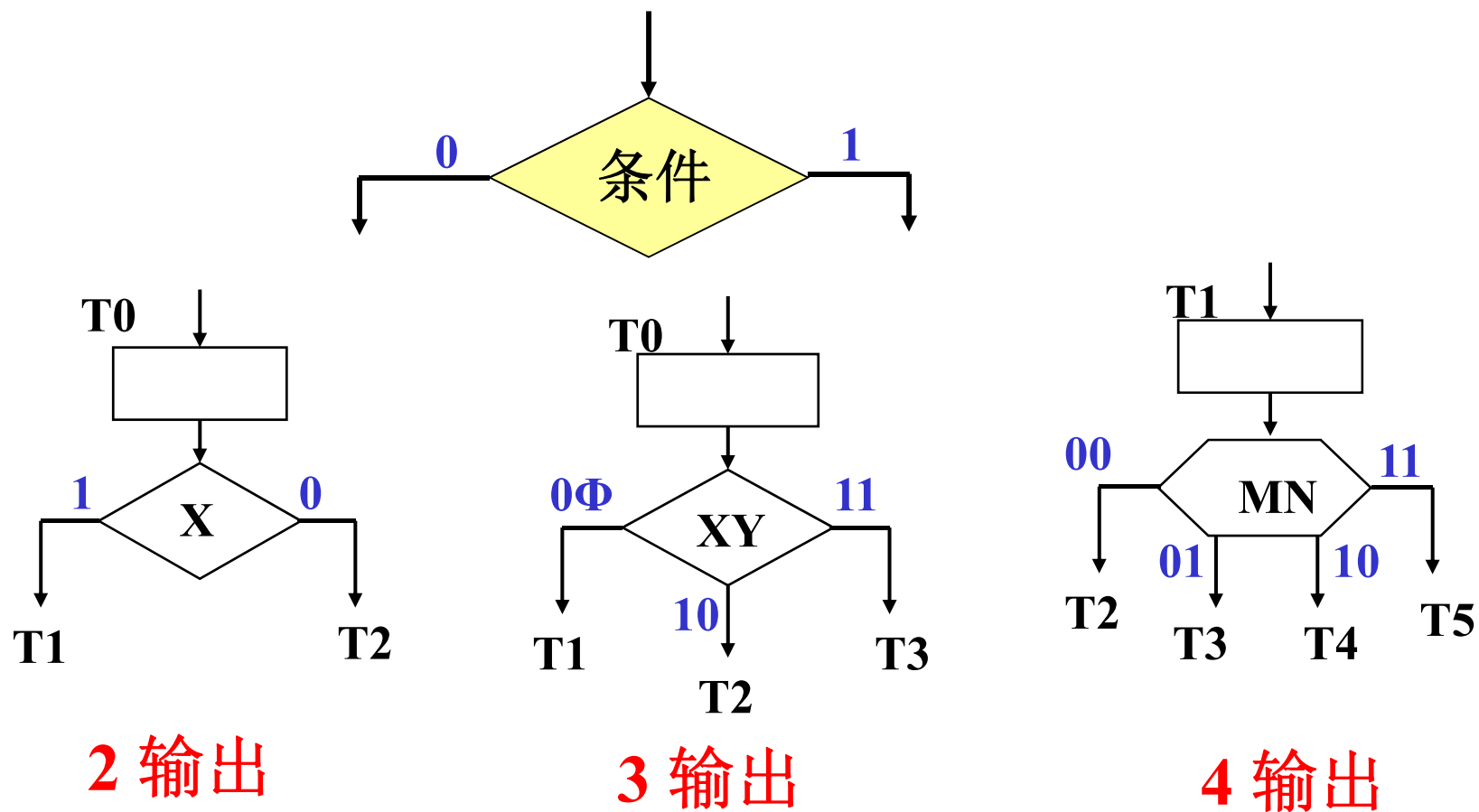
- Register
- Counter
- FF
- Memory

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

$A \leftarrow 0$  ( $A$  clear)

$F \leftarrow 1$  ( $F$  set 1)

## 2. 判断框 (prism 菱形)

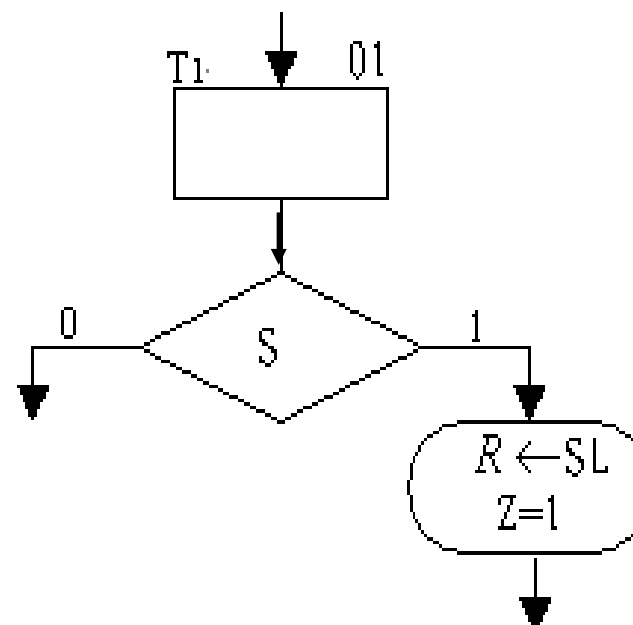
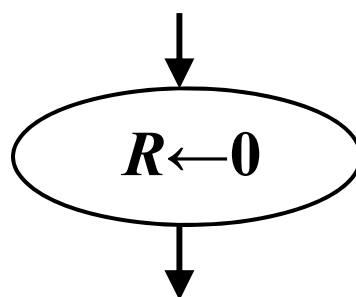
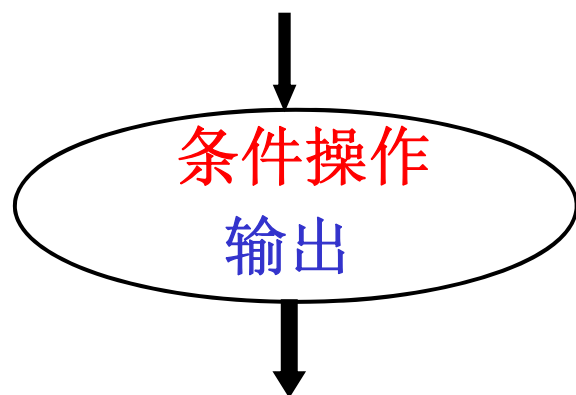


控制器根据判断框内容(条件)决定下一个 *CLK* 到时状态转换

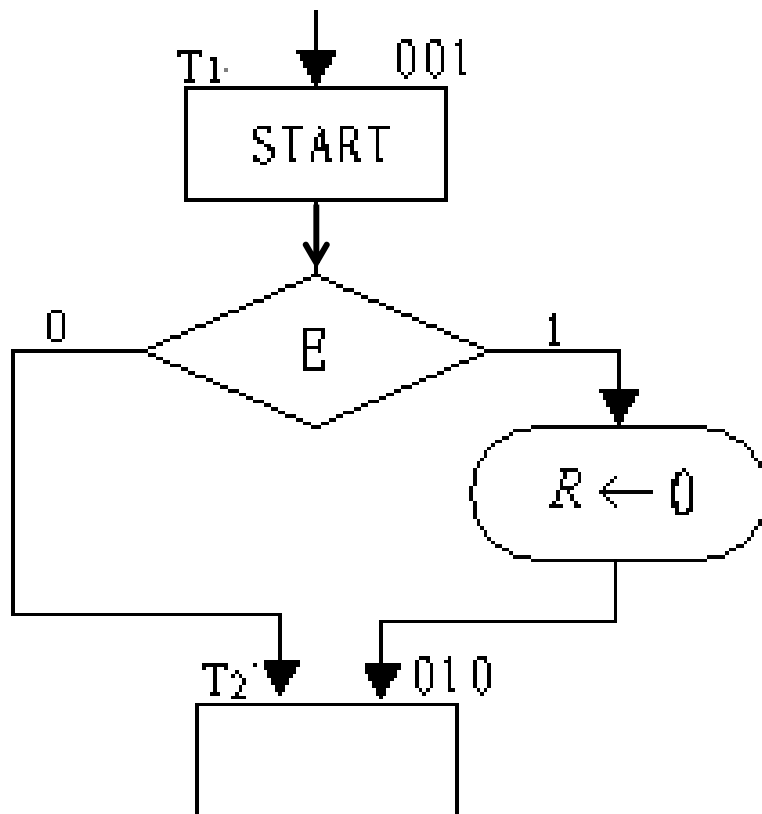
### 3. 条件框 (ellipse 椭圆)

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：分析下面 ASM 图。



在  $T_1$  状态下，  
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  $T_2$



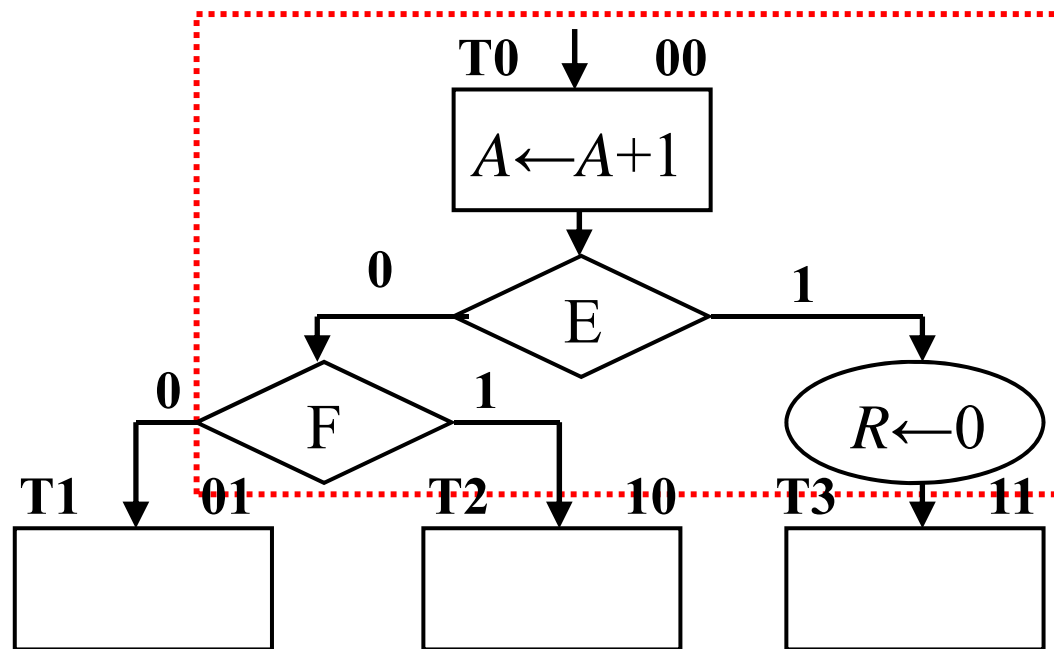
## 8.2.2 ASM块



**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。

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。



例：ASM块

一个状态框  
两个判定框  
一个条件框

$T_0$  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)

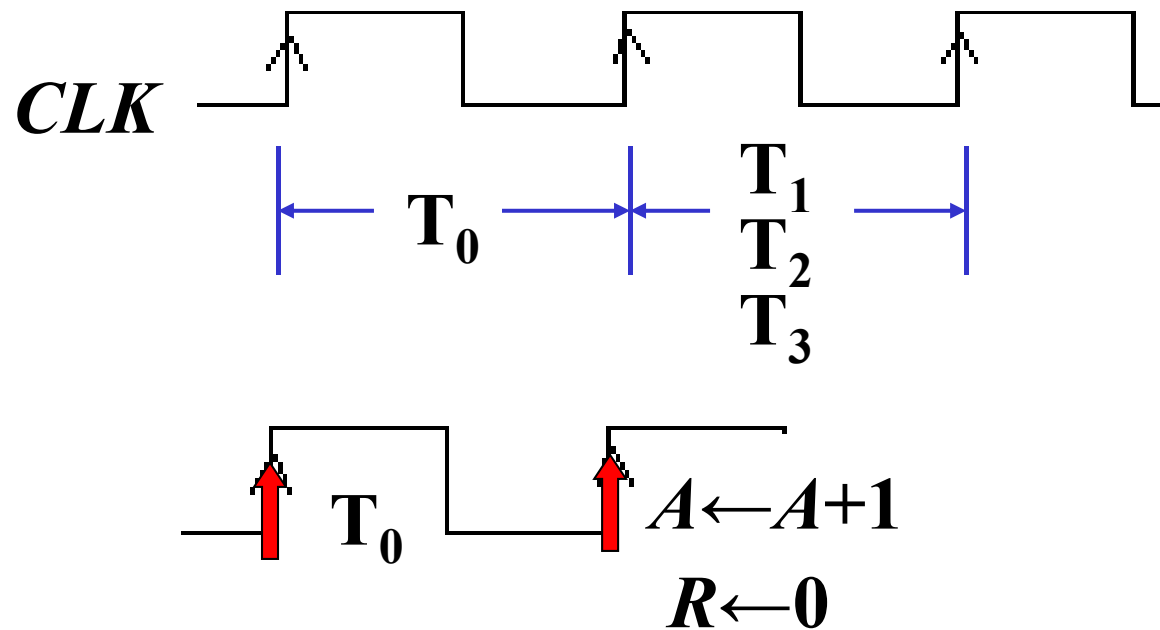
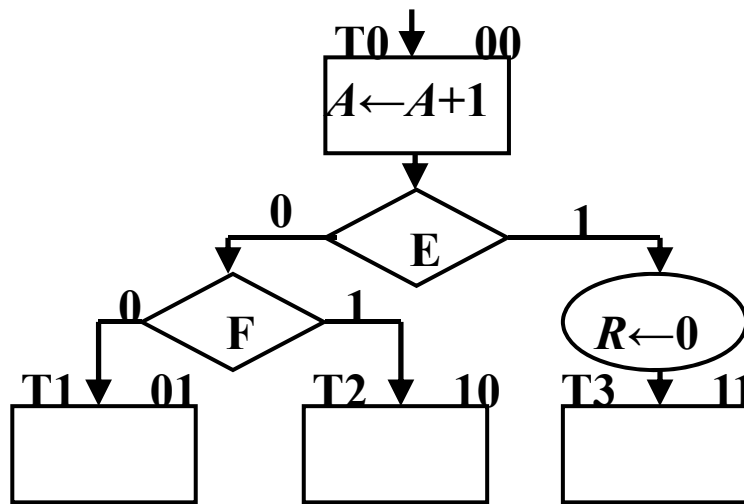
If  $E = 1$ ,  $R \leftarrow 0$   
(条件操作)

If  $E = 0, F = 0$ ,  
 $E = 0, F = 1$ ,

State  $T_0 \rightarrow T_3$

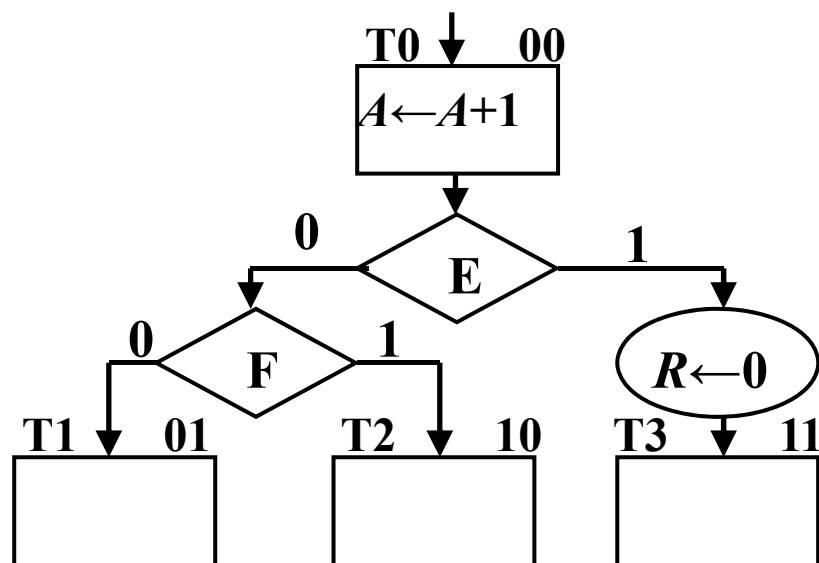
$T_0 \rightarrow T_1$

$T_0 \rightarrow T_2$

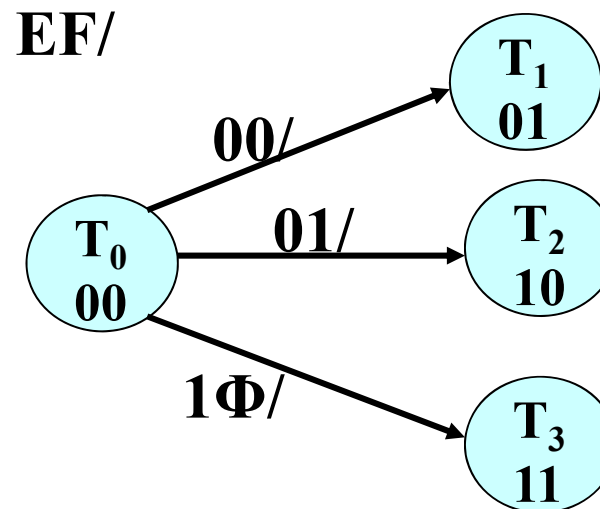


现态  $T_0$  与状态框内的操作不是在同一个 *CLK* 内

## 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

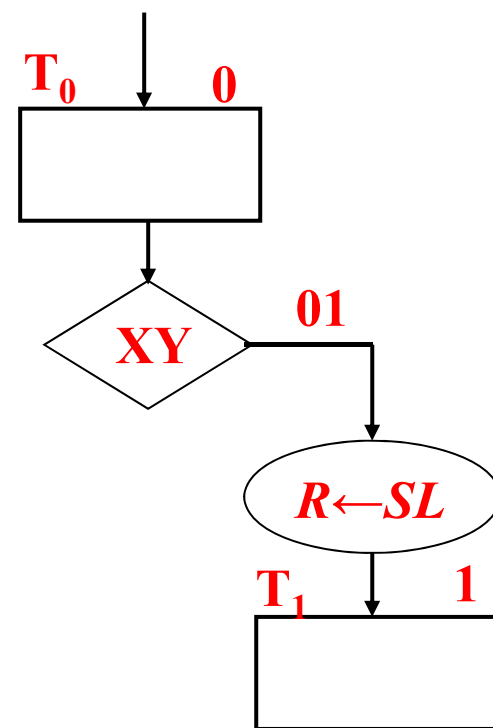
} 相异

### 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**：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是一可逆计数器，画出该数字系统的ASM图表。

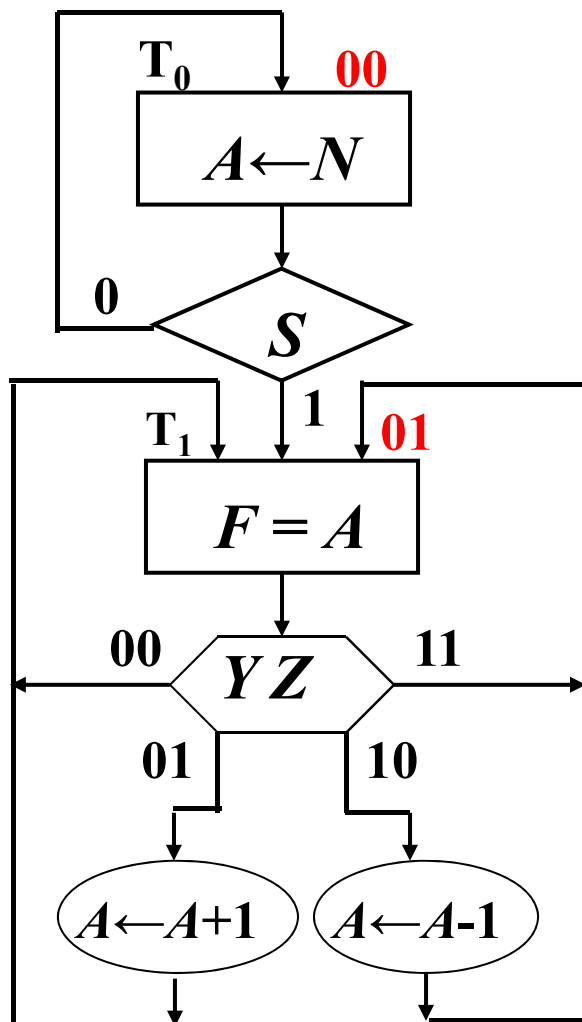
光电  
传感器

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

设  $N$ ：车场内目前的车辆数

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

# ASM



A: 计数器

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

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

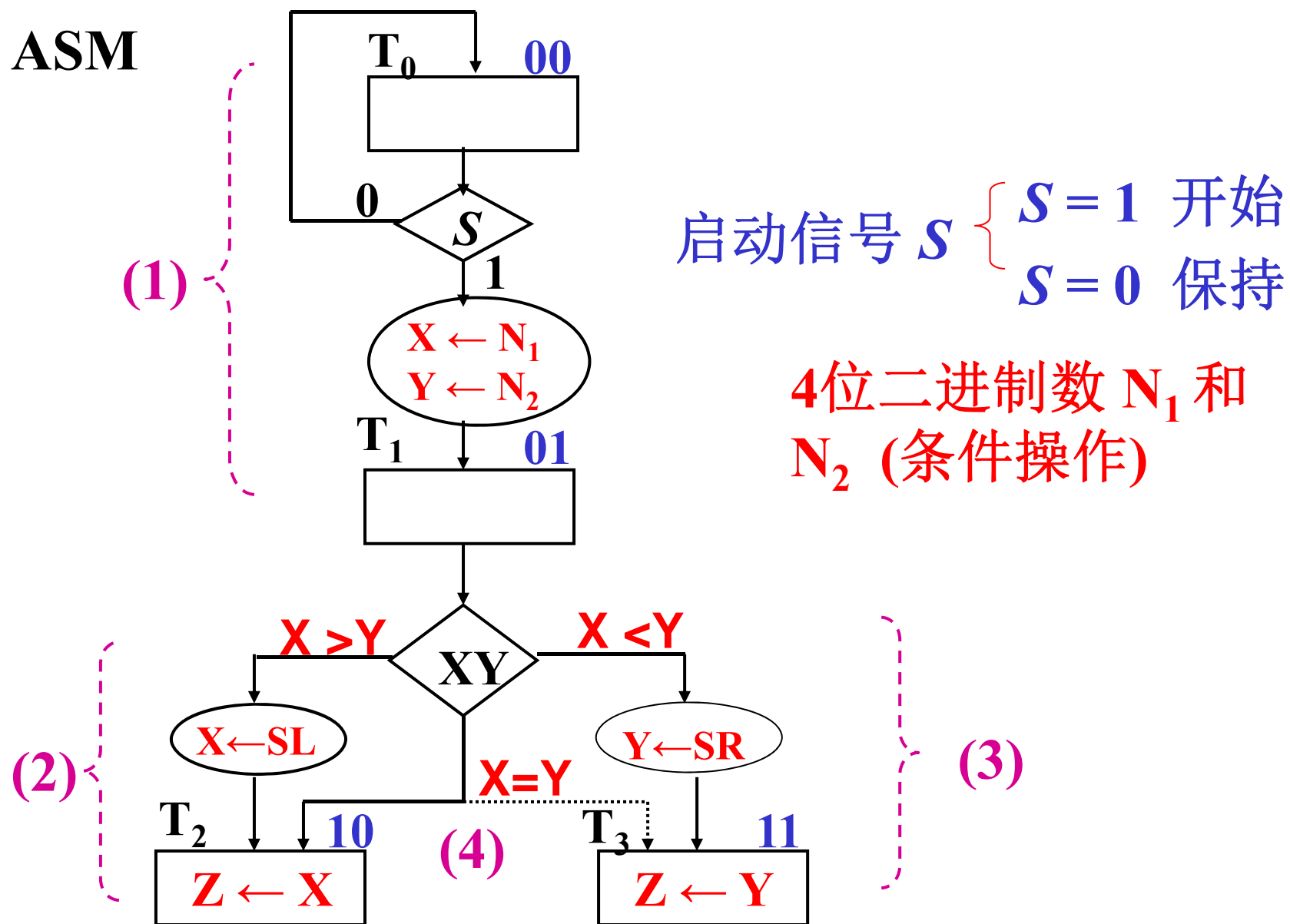
出,  $Z$   $\begin{cases} Z = 1 & \text{无车出} \\ Z = 0 & \text{车出} \end{cases}$

**例 3:** 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：

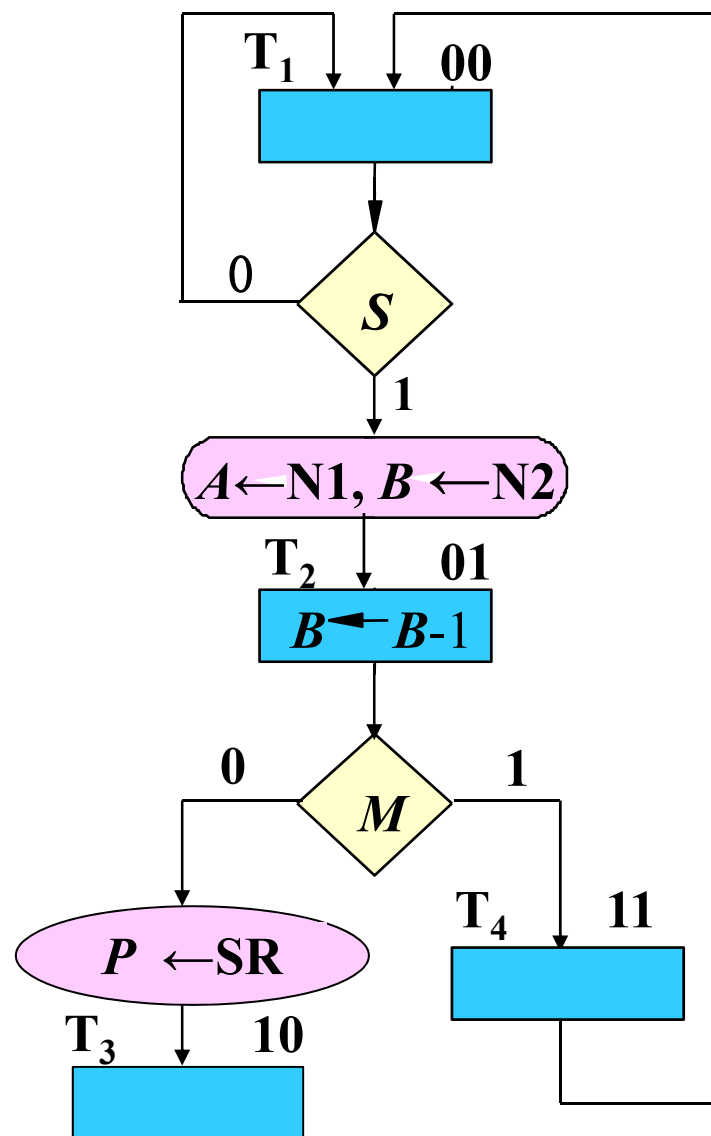
- (1) 启动信号 $S$  出现, 传送两个4 位二进制数 $N_1$ 、 $N_2$  给寄存器  $X$ 、 $Y$ ;
- (2) 如果  $X > Y$ , 左移 $X$ , 结果送给 $Z$ ;
- (3) 如果  $X < Y$ , 右移 $X$ , 结果送给 $Z$ ;
- (4) 如果  $X = Y$ , 将  $X$  或  $Y$  送给  $Z$ .



ASM



**练习：** 一个数字系统在 $T_1$ 状态下，若启动信号 $S=0$ ，则保持 $T_1$ 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。



## § 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  $\left\{ \begin{array}{l} 16\text{ s} : X=1 \\ 12\text{ s} : Y=1 \\ 9\text{ s} : Z=1 \end{array} \right.$

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

输出：

灯亮

苹果形： $A=1$

香蕉形： $B=1$

葡萄形： $G=1$

高有效

## 2. 建立 ASM 图

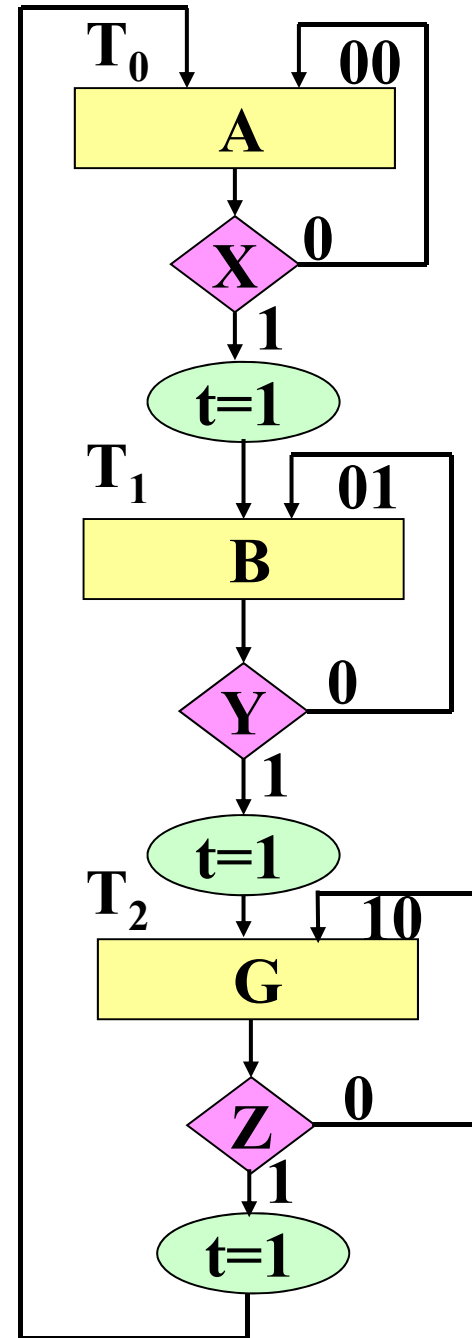
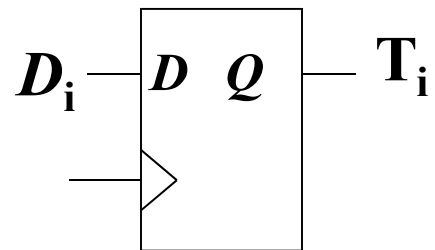
## 3. 设计控制器

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

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。

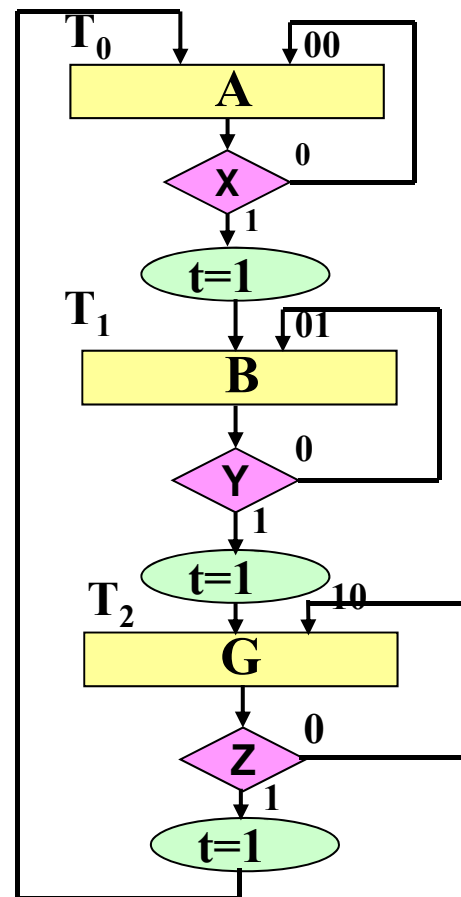
任何时刻，只能存在一个状态  
(=1)，其它状态=0

3 状态, 3 D-FF, 输入  $D$ , 输出  $T_i$

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

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

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

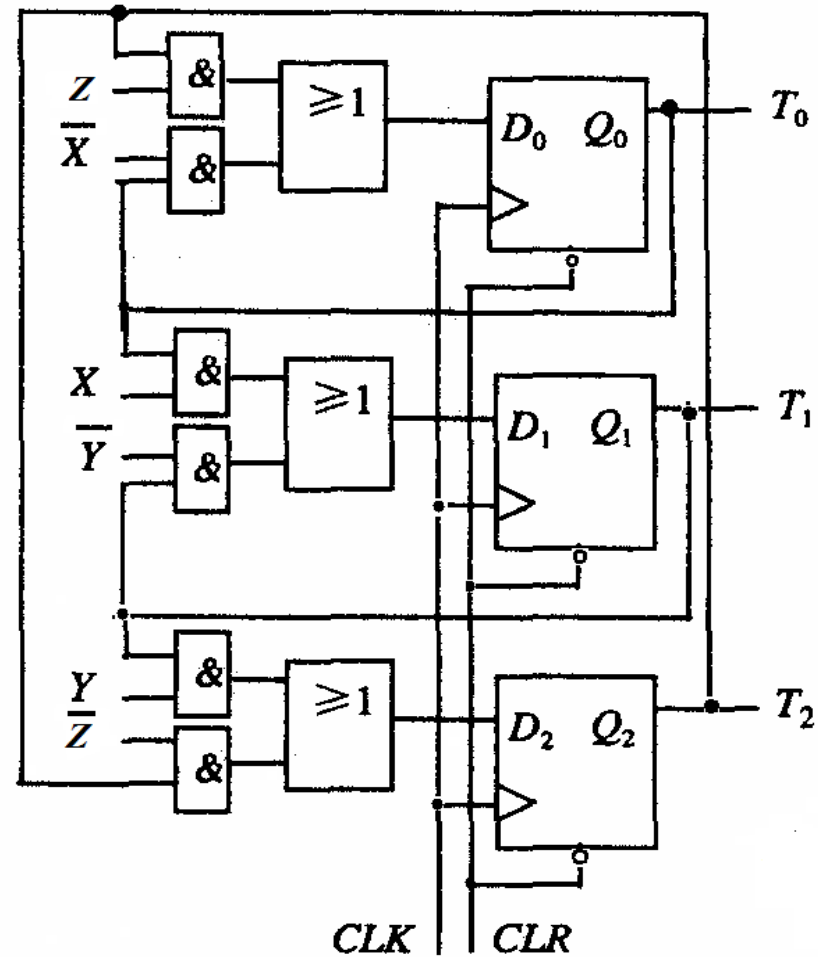


## 4. 电路

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

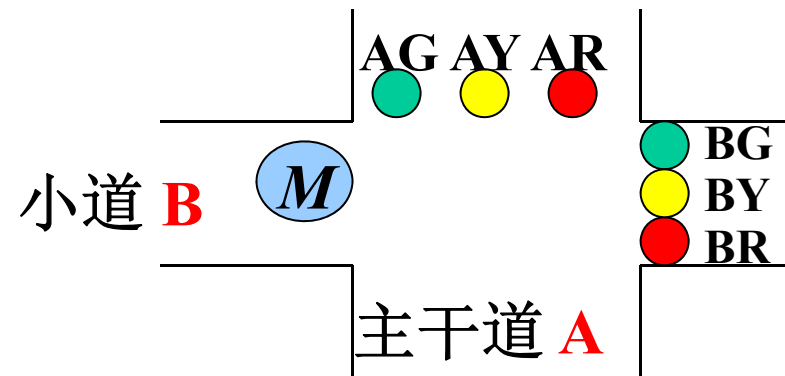
$$T_1=1, D_1 = T_0 X + T_1 \overline{Y}$$

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



## 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 **16 s**，超过 **16 s**，若小道有车 (**M = 1**)，主干道绿灯灭黄灯亮 **3 s**，然后红灯亮。小道绿灯 (通车) 最长时间 **16 s**，在 **16 s** 内，只要小道无车 (**M = 0**)，小道由绿灯变黄灯 (**3 s**) 后变红灯，主干道红灯变绿灯。 **16 s** 和 **3 s** 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  $M$   $\begin{cases} = 1 & \text{有车} \\ = 0 & \text{无车} \end{cases}$

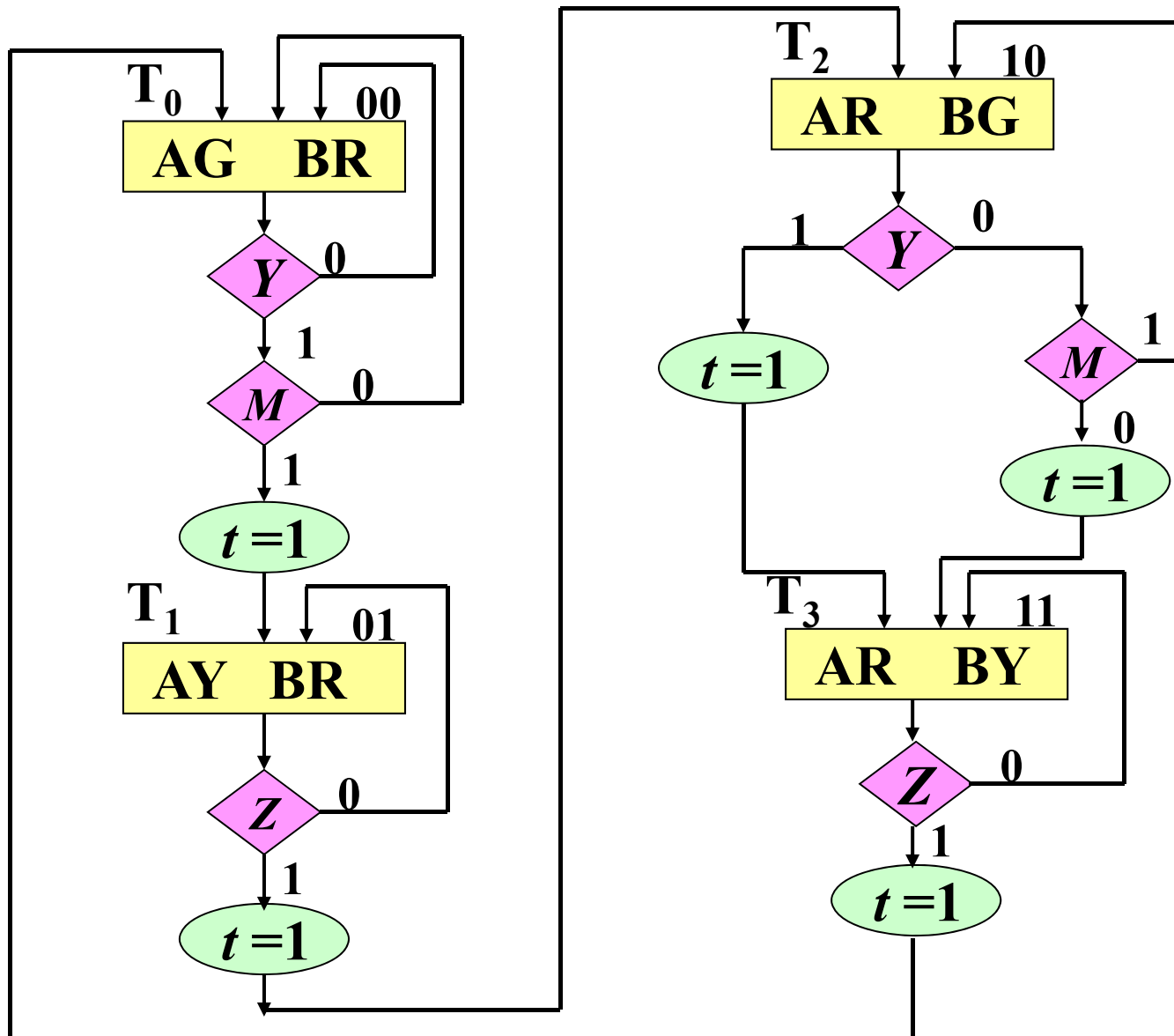
定时启动  $t$   $\begin{cases} = 1 & \text{计时开始} \\ = 0 & \text{否则} \end{cases}$

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

输出:  $AG, AY, AR, BG, BY, BR = 1$  亮

输入、输出均是高电平有效 (=1)

## 2. 建立 ASM 图



### 3. 控制器设计 (用 $M, Y, Z$ 得到: $T_0, T_1, T_2, T_3$ )

用 MUX, D-FF, 译码器设计控制器

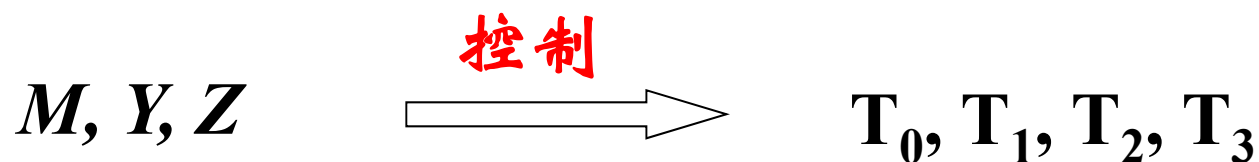
根据 ASM: 4 个状态  $T_0, T_1, T_2, T_3$

输出: 高电平有效 2-4 译码器

其入口接两个 D-FF 的出口  $Q_0, Q_1$

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	现状态 $Q_1^n \quad Q_0^n$	输入 $Y \quad Z \quad M$	新状态 $Q_1^{n+1} \quad Q_0^{n+1}$	输出 $T_0 \quad T_1 \quad T_2 \quad T_3$
$T_0$	0    0	$\begin{matrix} 0 & \Phi & \Phi \\ 1 & \Phi & 0 \\ 1 & \Phi & 1 \end{matrix}$	$\mathbf{0} \left\{ \begin{matrix} 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{matrix} \right\} YM$	1   0   0   0
$T_1$	0    1	$\begin{matrix} \Phi & 0 & \Phi \\ \Phi & 1 & \Phi \end{matrix}$	$\mathbf{Z} \left\{ \begin{matrix} 0 & 1 \\ 1 & 0 \end{matrix} \right\} \bar{Z}$	0   1   0   0
$T_2$	1    0	$\begin{matrix} 0 & \Phi & 1 \\ 0 & \Phi & 0 \\ 1 & \Phi & \Phi \end{matrix}$	$\mathbf{1} \left\{ \begin{matrix} 1 & 0 \\ 1 & 1 \\ 1 & 1 \end{matrix} \right\} \begin{matrix} Y + \bar{Y} \cdot \bar{M} \\ Y + \bar{M} \end{matrix}$	0   0   1   0
$T_3$	1    1	$\begin{matrix} \Phi & 0 & \Phi \\ \Phi & 1 & \Phi \end{matrix}$	$\mathbf{\bar{Z}} \left\{ \begin{matrix} 1 & 1 \\ 0 & 0 \end{matrix} \right\} \bar{Z}$	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 **K-map (VEM)**, 把  $M, Y, Z$  作引入变量.

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

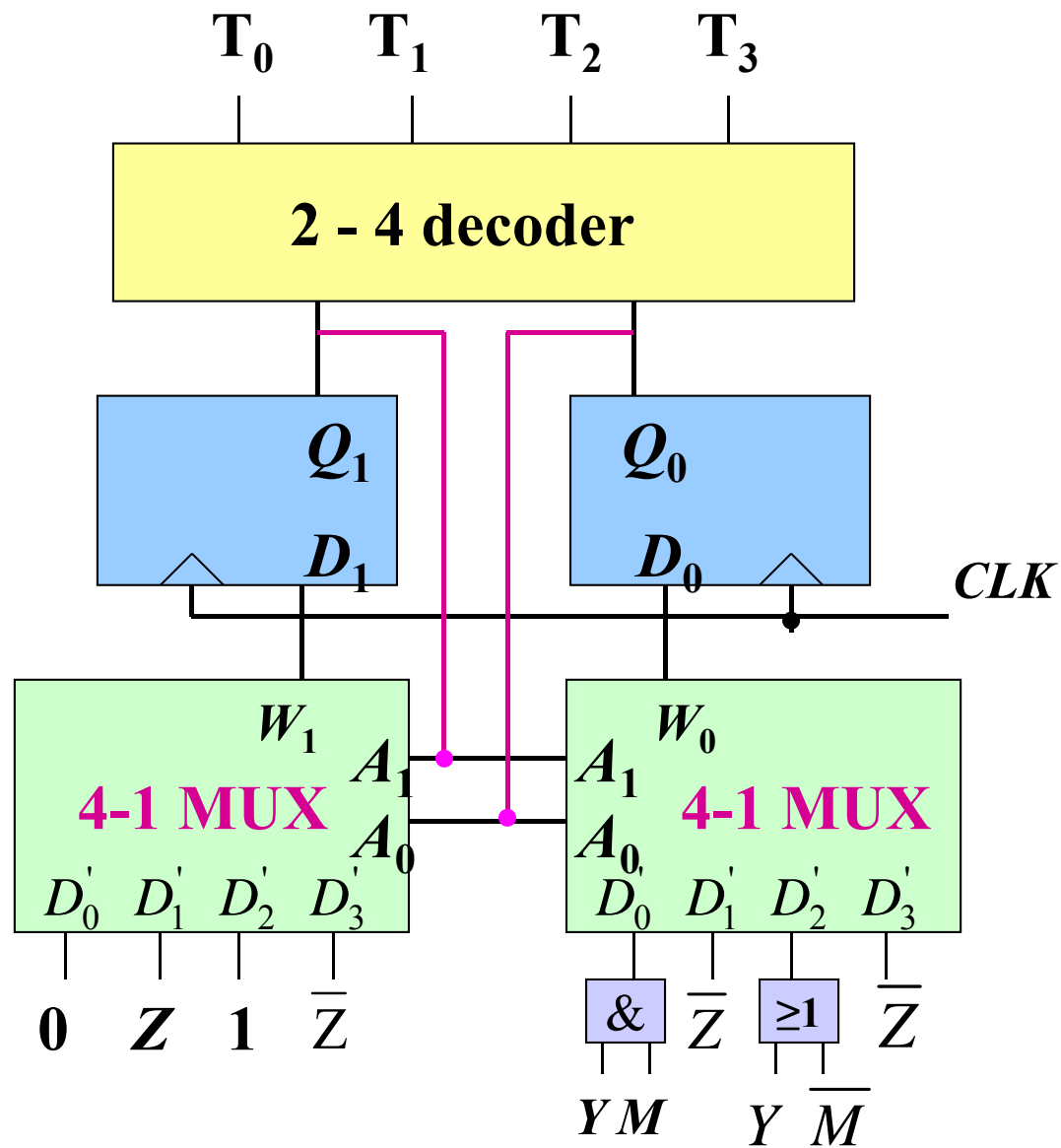
$D_1$

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

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

## 4. 数据处理器设计

### (1) 灯电路

$$AG = T_0$$

$$AY = T_1$$

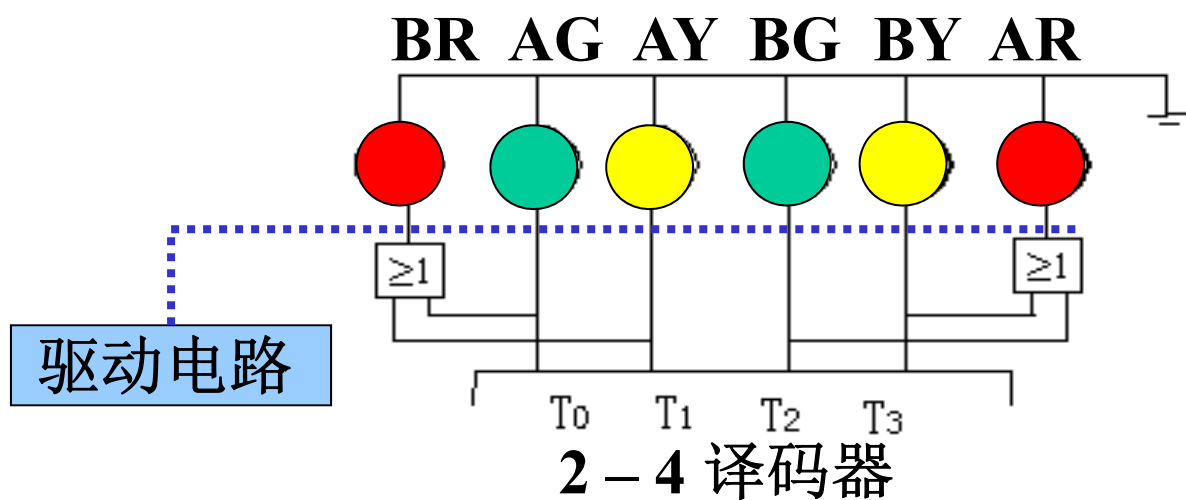
$$AR = T_2 + T_3$$

$$BG = T_2$$

$$BY = T_3$$

$$BR = T_0 + T_1$$

State	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

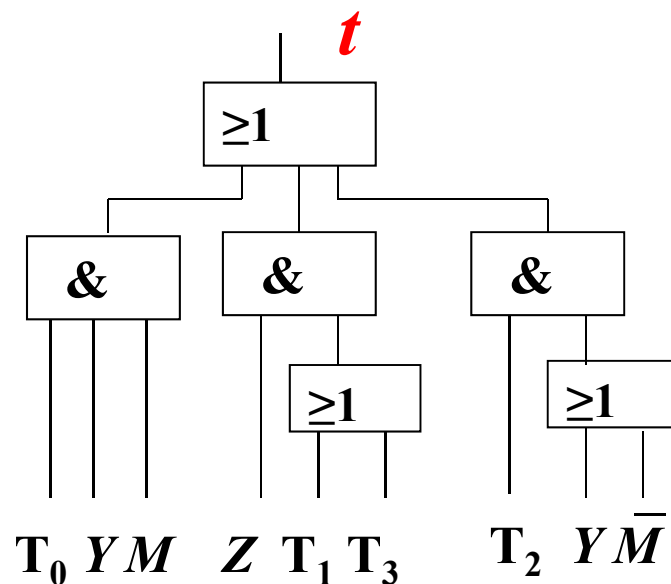


## (2) 定时启动电路 (产生 $t = 1$ )

定时启动 :  $t = 1$

ASM :  $t = 1$  条件

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





### (3) 计时电路 (产生 $Y, Z$ )


2个计数器  $Y: 16\text{ s}, Z: 3\text{ s}$ .

74161 驱动要求

用74161实现:

M-16 ( $Y$ ) 和 M-3 ( $Z$ )

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 	Clear (清0)	0	$\Phi$	$\Phi$	$\Phi$	$\Phi$	$\Phi$	$\Phi$	$\Phi$
$t$	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?

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

即  $T_0$  或  $T_2$  状态下,

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

$$Y = 1;$$

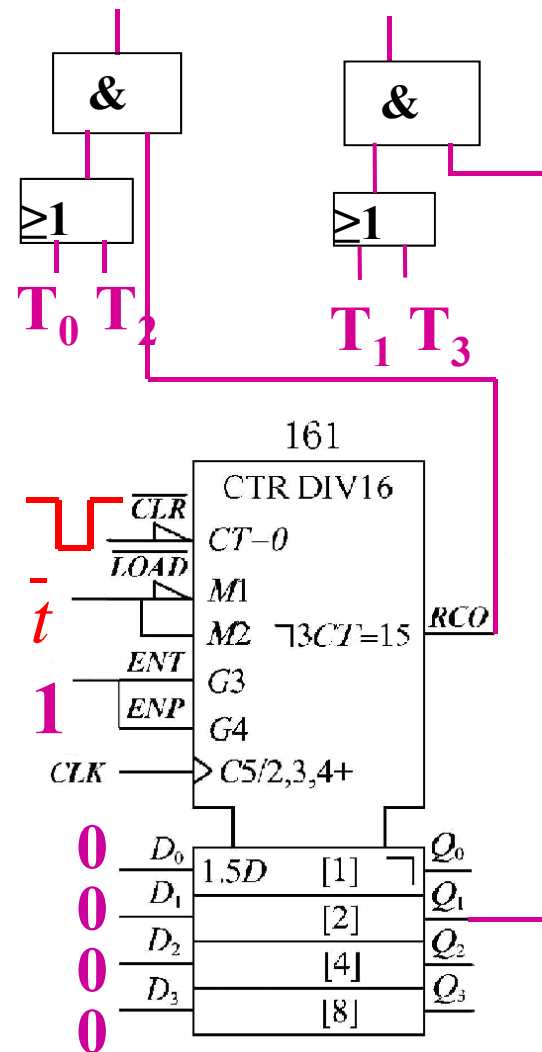
$$Z = (T_1 + T_3)Q_1 = 1$$

即  $T_1$  或  $T_3$  状态下,

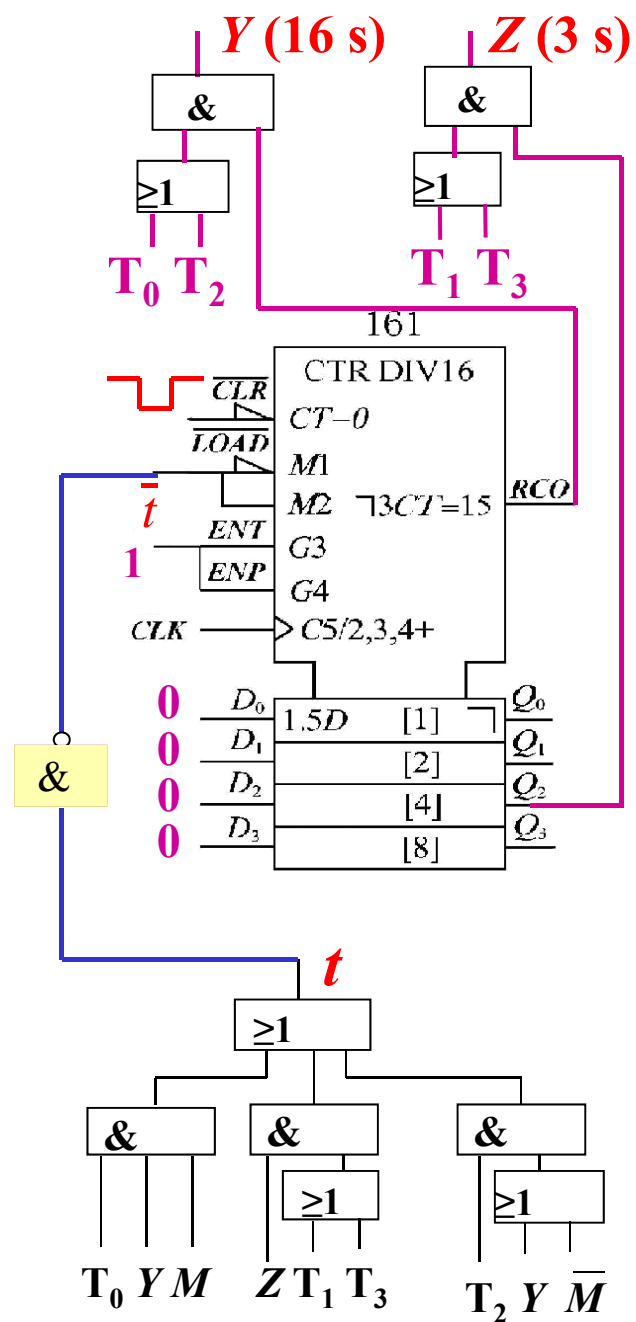
$$Q_3Q_2Q_1Q_0 = 0010$$

$$Z = 1.$$

$Y$  (16 s)     $Z$  (3 s)



The circuit diagram illustrates a 2-to-4 decoder implemented using two 4-to-1 multiplexers and two D flip-flops. The decoder's four outputs,  $T_0, T_1, T_2,$  and  $T_3$ , are connected to the  $D$  inputs of the flip-flops  $Q_1$  and  $Q_0$ . The flip-flops are clocked by a common  $CLK$  signal. The outputs of the flip-flops are  $Y$  and  $M$ , which are combined using an AND gate and an OR gate to produce the final outputs  $YM$  and  $Y\overline{M}$ .



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  $\overline{R_p}$ ) 

