

## 第五章 数字系统

### 5.1 数字系统概述

#### 5.1.1 数字系统的基本模型

#### 5.1.2 数字系统设计方法

#### 5.1.3 数字系统设计描述工具

### 5.4 A/D转换和D/A转换

#### 5.4.1 模数A/D转换

#### 5.4.2 数模D/A转换

# 第五章 数字系统设计

## *Digital System Design*

组合逻辑电路和时序逻辑电路只能完成某些特定的逻辑功能，属功能部件级。电路分析和设计是建立在真值表、卡诺图、逻辑方程式、状态表和状态图的工具基础上，主要依赖于设计者的熟练技巧和经验，称“凑试法”。

若由功能部件级组成一个功能复杂、规模较大的数字系统时，虽然在理论上仍可以把它看成是一个大型时序逻辑电路，仍可以采用凑试法，但实际实现上很难、甚至无法达到完整地描述其逻辑功能。因为这种设计方法：原始、受限制最多、效率与效果均欠佳、局限性大。要用方框图、定时图、逻辑流程图、ASM图、MDS图等系统描述工具。

## 5.1 数字系统概述

对数字系统进行分析 and 设计时，通常把系统从逻辑上划分成**控制单元CU**和**信息处理单元**两大部分。其中：

**信息处理单元**对信息进行不同的处理和传递，

**控制单元**保证信息处理单元按规定的微操作序列处理数据。

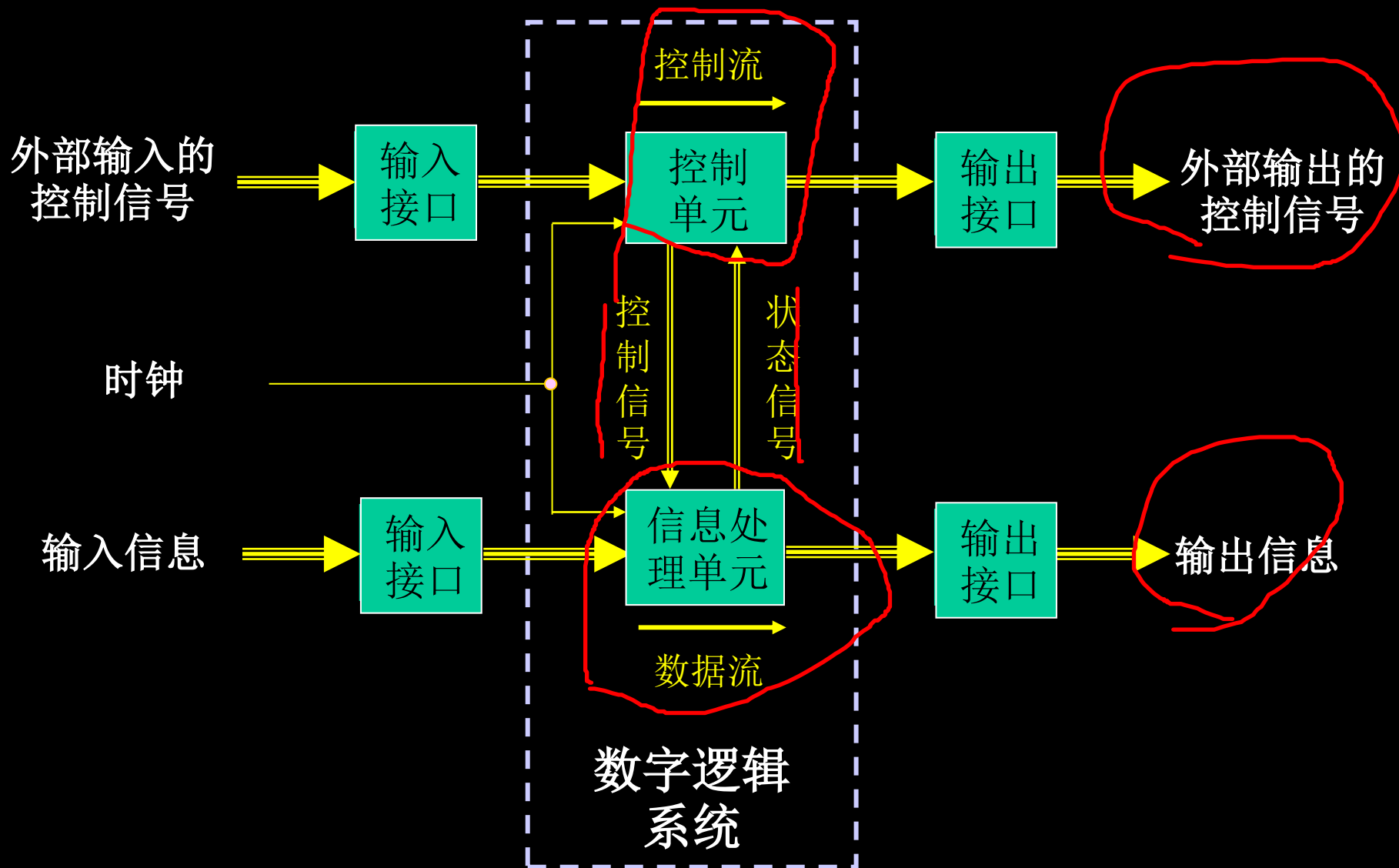
控制单元——**不断生成和发送控制信号序列**，控制信息处理单元不断地执行特定的操作；

——接收来自信息处理单元的**状态信息**，用以选择下一个需执行的操作。

——接收外来的控制信息，用以改变正在执行的操作序列。

**控制单元是区别数字系统与功能部件的标志。**

## 5.1.1 数字系统的基本模型



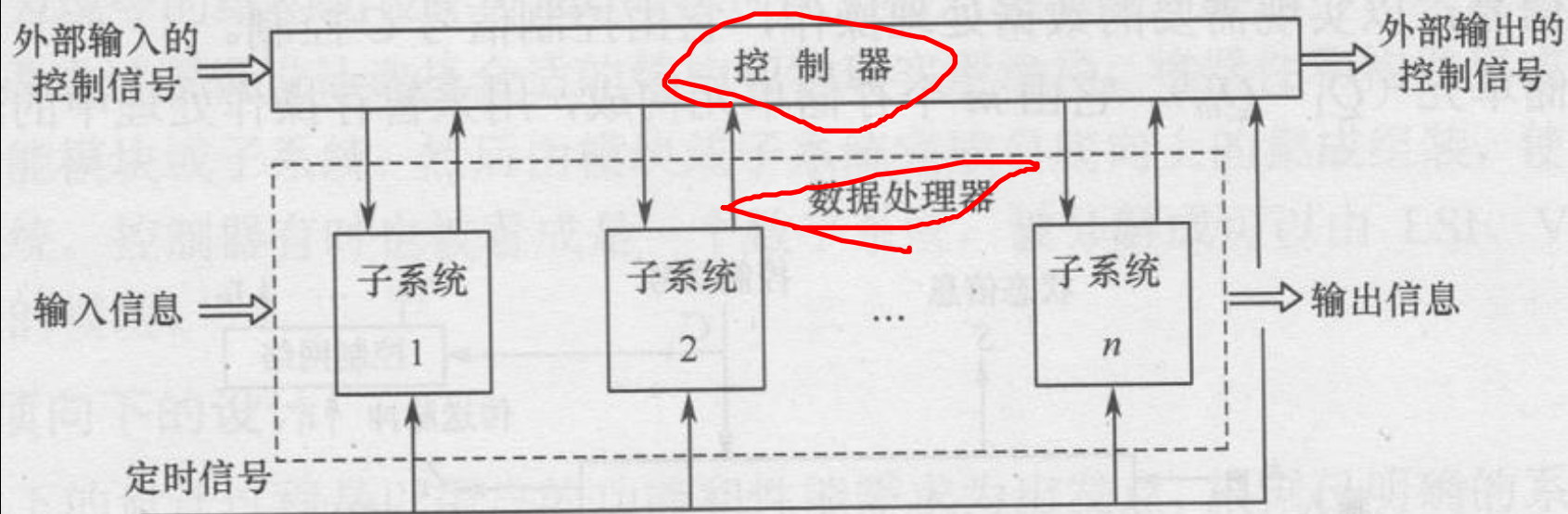
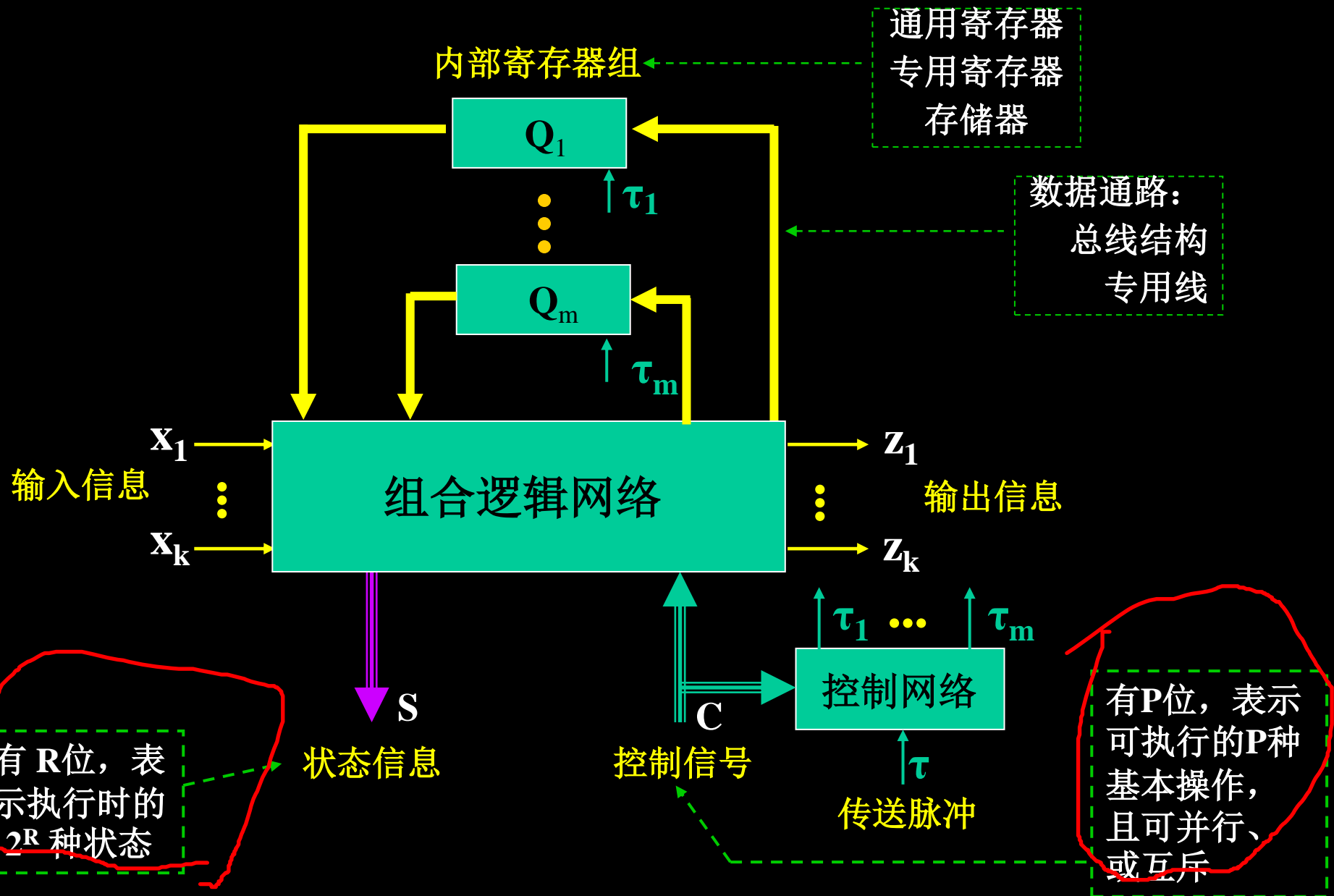


图 5.1 数字系统的一般模型

# 1. 信息处理单元的构成



## 状态信息表及操作表举例

S的编码位	该编码位定义的状态标志
$S_1$	$S_1 := (x > 0)$
$S_2$	$S_2 := (x = 0)$
$S_3$	$S_3 := (x < 0)$
$\vdots$	$\vdots$
$S_r$	$S_r := (Q_1 = x) \vee (Q_2 = x)$

(a) 状态信息表

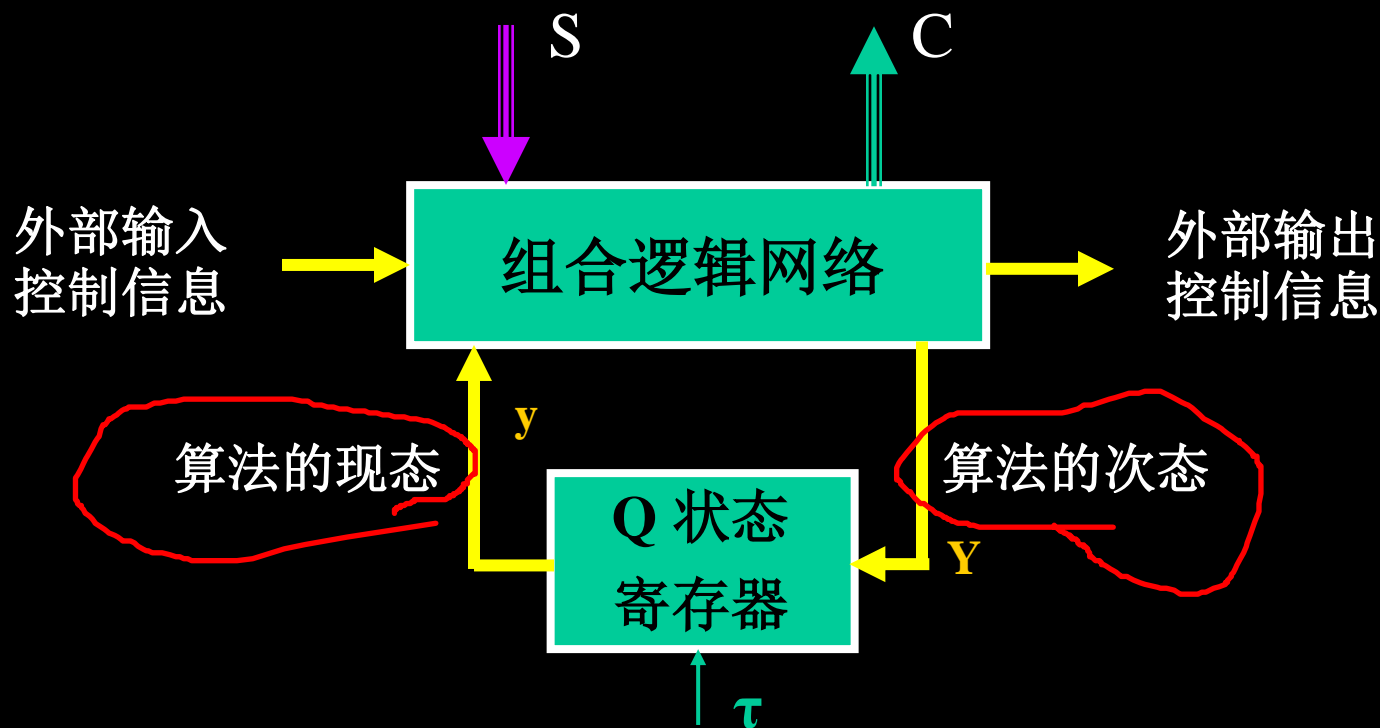
控制信号C	执行的操作
$C_1 := \text{CLR}$	$Q_1 \leftarrow 0$
$C_2 := \text{ADD}$	$Q_1 \leftarrow Q_1 + x$
$C_3 := \text{SUB}$	$Q_1 \leftarrow Q_1 - x$
$\vdots$	$\vdots$
$C_p := \text{INC}$	$Q_1 \leftarrow Q_1 + 1$

(b) 操作表

## 2. 控制单元CU的构成

将数字系统执行的复杂任务<sub>任务</sub>转化为一个操作和测试序列，称为“算法”。

用控制单元产生与操作序列相对应的控制信号序列，每一个控制信号控制信息处理单元执行与算法相关的一个操作。所以，控制单元的基本功能具体上是对指令流和数据流实施时间上和空间上的正确的控制。





控制单元的**核心**是时序电路，本质上是一个**状态寄存器**。状态寄存器主要有两个功能——寄存控制单元的**现态**，生成**次态**。采用触发器作为状态寄存器的元件。

存在着两种不同的控制单元实现方法：

**硬件逻辑方法**——用**逻辑电路**生成每一个微操作的控制信号；特点：速度快、动一发而动全身。

**微程序方法**——计算机的每一条指令的功能通过**执行一个微指令序列**(微程序)来实现的。设计好的微程序被固化在只读存储器中，这个存储器称为**控制存储器**。控制器按微程序的执行步骤逐个发出控制信号。特点：速度低、但设计、修改及扩充容易。

显然，采用不同的实现方法，将影响控制单元的组成和结构。在现代计算机控制器中，常常两者混用。

## 5.1.2 数字系统逻辑设计的基本步骤

第一步：确定系统的逻辑功能。

设计者应对系统仔细分析、消化和理解，逐步明确其逻辑功能，输入、输出信号等内容。

第二步：确定系统方案。

这是设计工作中最困难、最有创造性的工作，设计者根据设计要求分析推演出信息处理的基本原理和可供选择的结构形式，因为可以采用不同的原理和方法实现某一逻辑功能。为此设计者要进行认真的比较和权衡，从中选取较为满意的方案。

第三步：对系统进行逻辑划分。

将系统按信息处理单元和控制单元划分为两大部分，列出信息处理单元的说明，并用流程图等方法描述信息处理的算法（即控制单元的逻辑要求）。每个部分应具备基本独立的逻辑功能。逻辑划分和确定系统方案的过程要同时进行、相辅相成。

# 数字系统设计的基本步骤

第四步：设计信息处理单元和控制单元。

- 定义要求信息处理单元必须执行的处理和操作，列出操作表；
- 提出实现的算法，确定控制单元必须保存的、或产生的状态及状态之间的转换关系；
- 由系统的控制状态、信息处理单元产生的状态信息和要求生成的控制信号建立控制单元的状态转换表；
- 根据信息处理单元的操作表和状态信息、控制单元的状态转换表, 进行逻辑设计。
- 选择逻辑功能部件，如SSI、MSI、LSI等，进行合理连接，构成严格能协调工作的系统。

## 5.1.3 数字系统设计的描述工具

### ➤ 方框图

设计系统首先应当**建立模型**，方框图是描述模型最常用、最重要的工具。

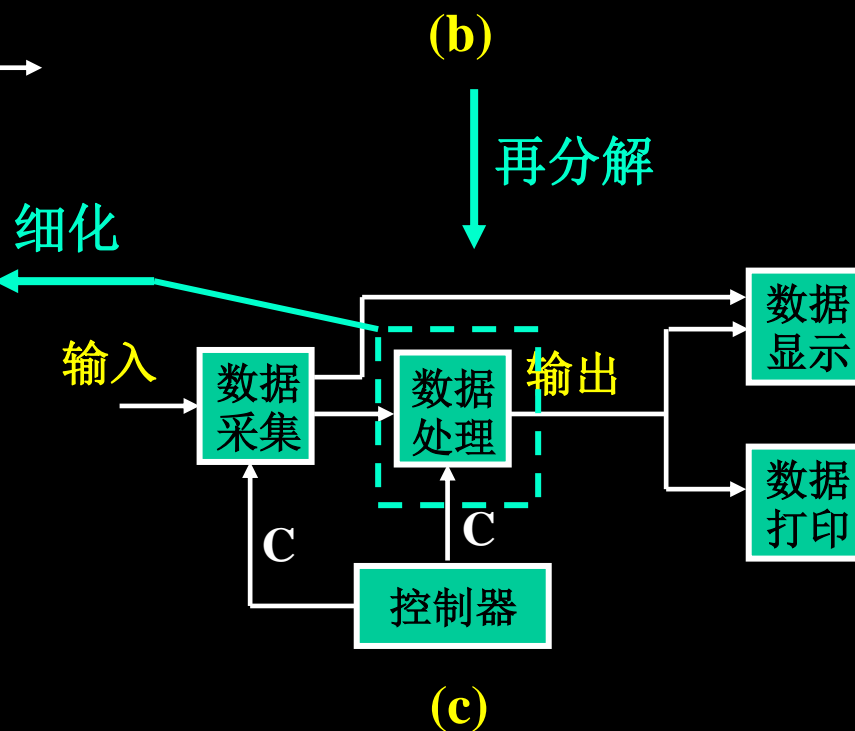
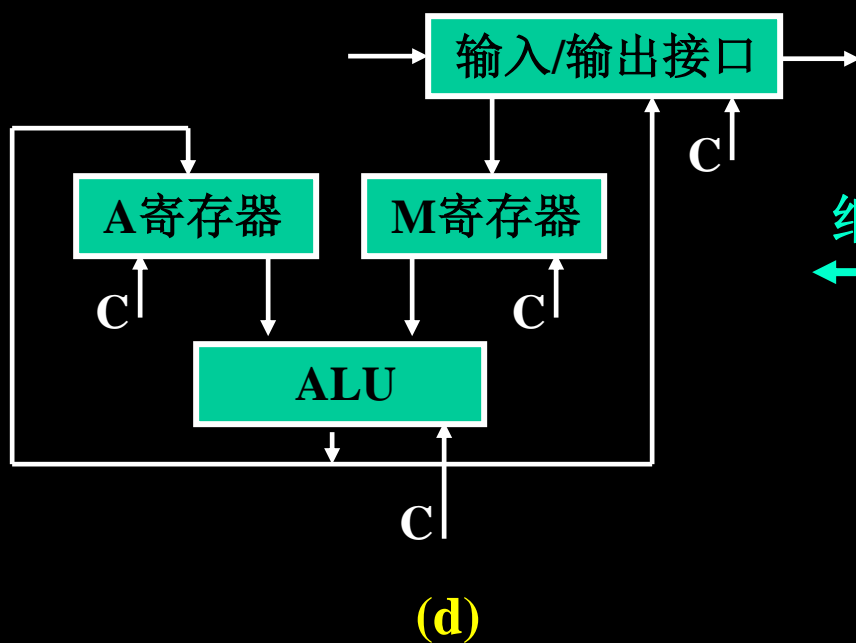
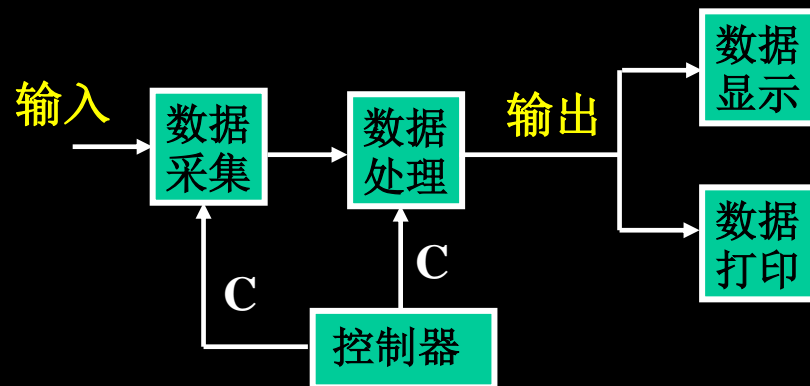
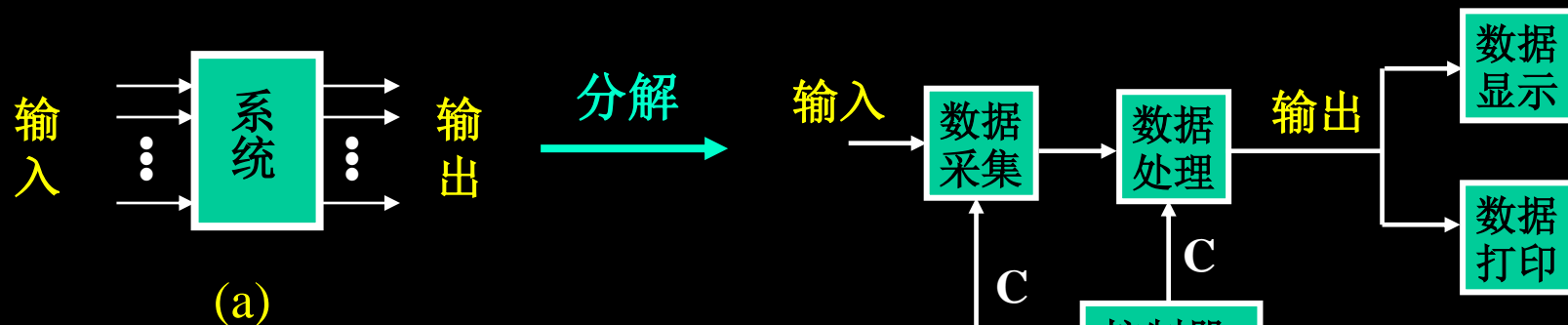
方框图不涉及过多的技术细节，直观易懂，具有：

**结构化设计**——系统结构清晰和易理解性，易构思设计等在方框图中：

- 每一个**方框定义**了一个信息处理、存储或传递的子系统（或模块）；
- 方框内用文字、表达式、例行符号、图形**表示**该模块的名称或主要功能；
- 方框之间用**指向线**相连，表示模块之间的数据流或信息流的**信息通道及方向**，连线旁的文字或符号是通道的名称、功能或信息类型。

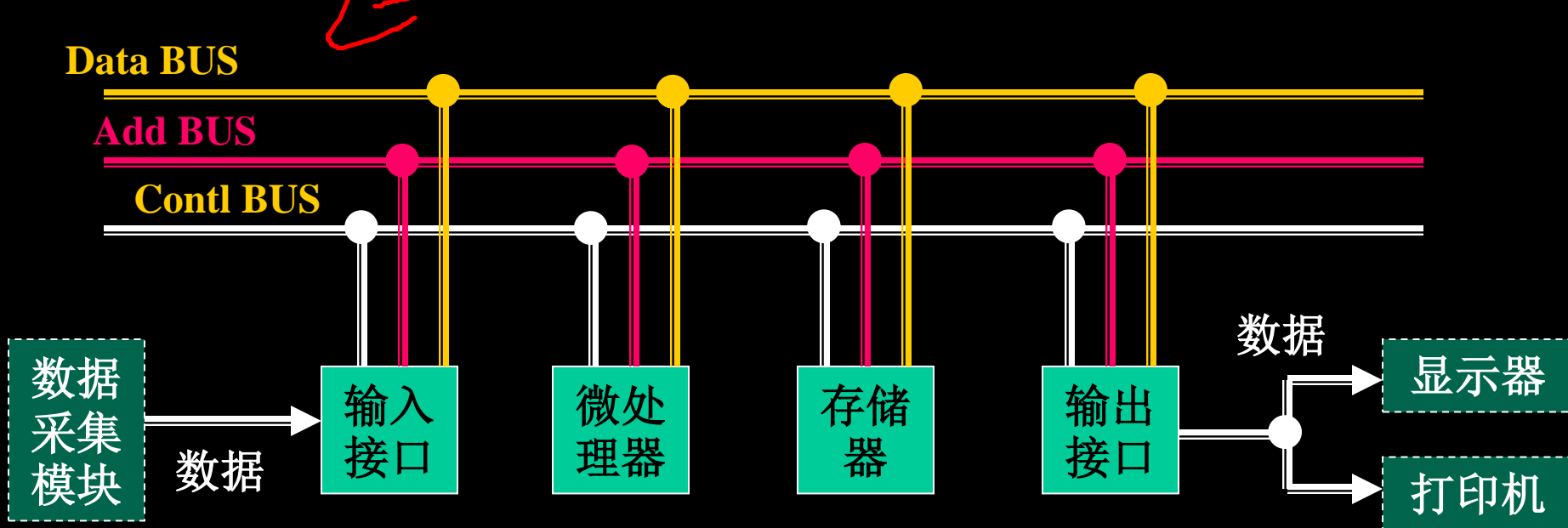
方框图的设计过程：**自顶而下、逐步细化。**

# 例 一个智能仪表的方框图。



上例设计的这种结构框图，其任何一处的功能修改和扩充均涉及全局，而且总控制器的设计将十分复杂。

若采用总线结构，则该智能仪表的方框图如下：



同一种功能的数字系统，可以设计出不同的结构，在总体结构设计中，任何设计优化的考虑都要比物理实现阶段中的技术优化产生大得多的效益，这是系统设计过程中**最具创造性**的工作。

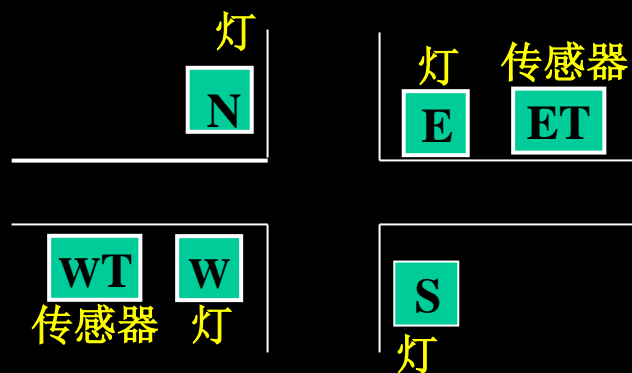
### 5.1.3 数字系统设计的描述工具

#### ➤ 定时图 (时序图、时间关系图)

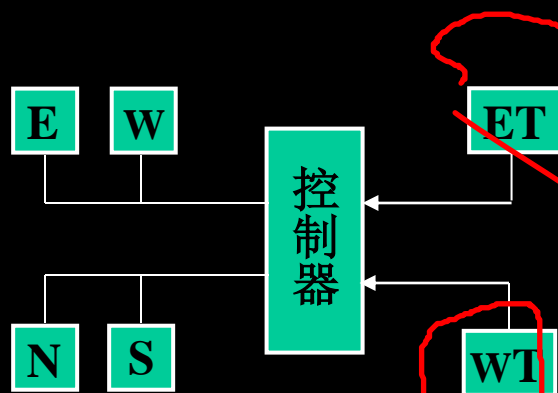
在数字系统中，信息的传送、处理或存储都是在特定时间意义上的操作，是按照严格的时序进行协调和同步的。系统各模块之间，模块内部各功能部件之间、各功能部件内部的各逻辑门电路或触发器之间，**输入信号、输出信号和控制信号的对应关系及特征**，通常用时序图来描述。

时序图的描述也是一个逐步细化的过程。从描述系统输入、输出之间的定时关系的**简单时序图**开始，随著系统设计的不断深入，时序图将不断地反映新出现的系统内部信号的时序关系，直到最终一个**完整的时序图**。

# 例 交通灯控制系统：要求N、S绿灯常亮，E、W红灯常亮

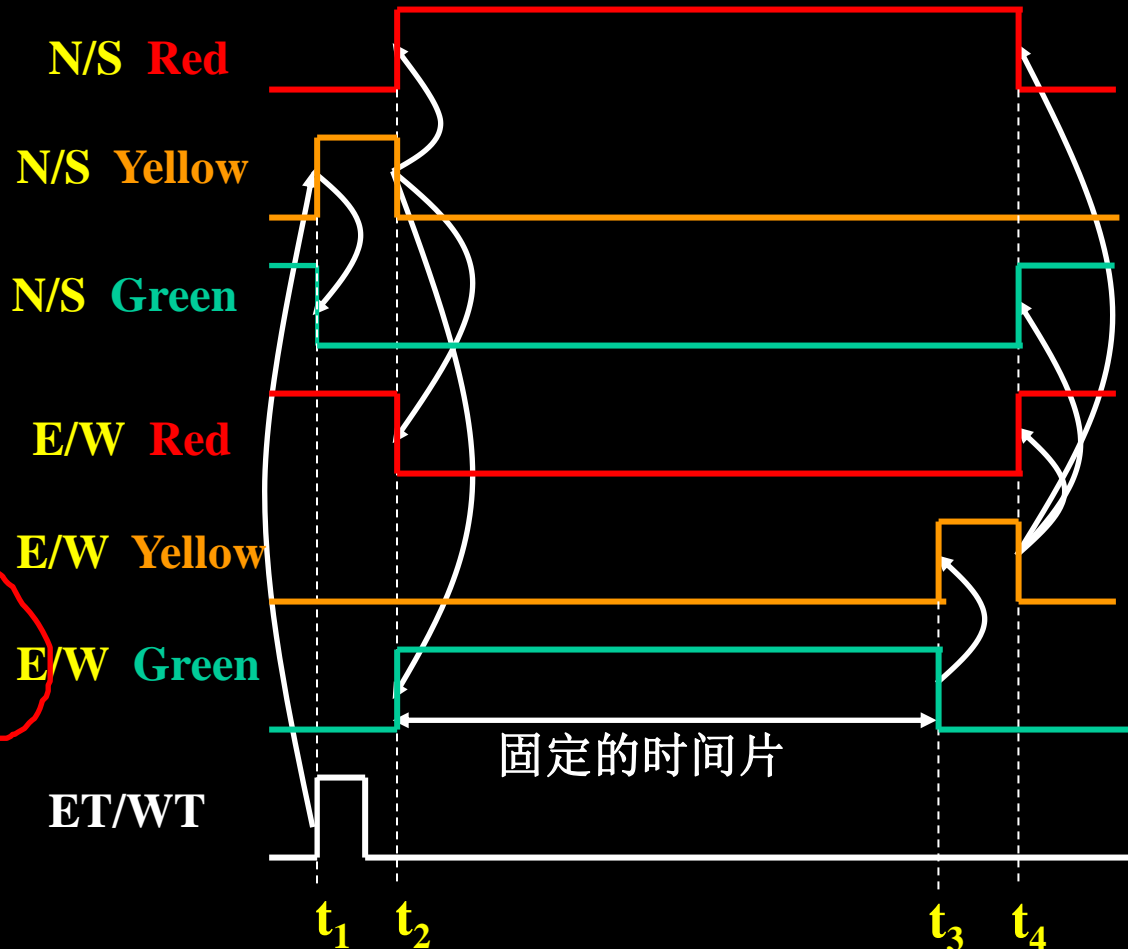


(a) 示意图



(b) 框图

关注各信号：电平/脉冲、同步/异步



(c) 定时图



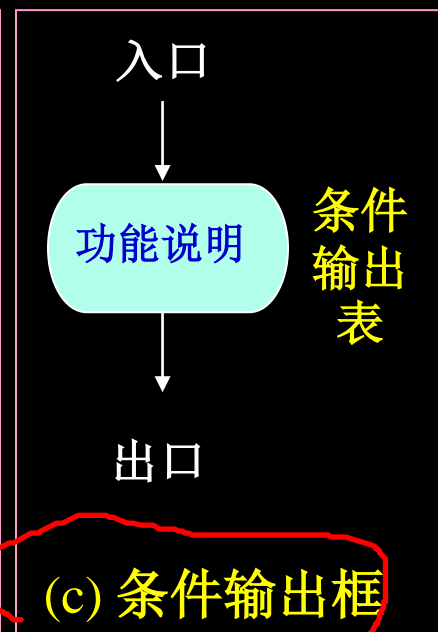
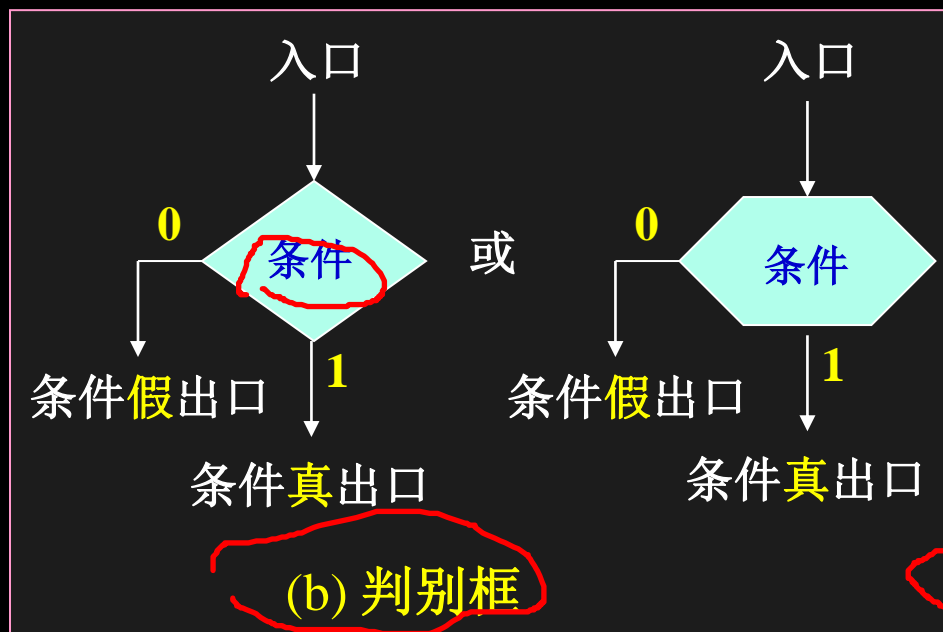
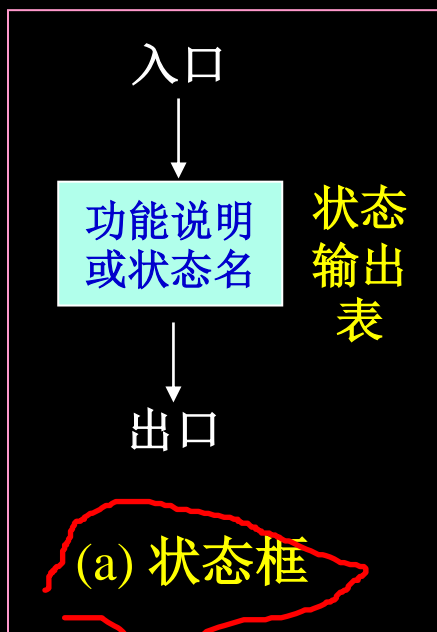
# 1. 逻辑流程图

也称流程图，它用约定的几何图形(矩形、菱形、椭圆形等)、指向线和简练的文字说明，描述系统的基本工作过程。

逻辑流程图的描述对象是**控制单元**，并且以**系统时钟**来驱动整个流程。这一点与由事件驱动的软件流程图不同。

## 1)、基本符号

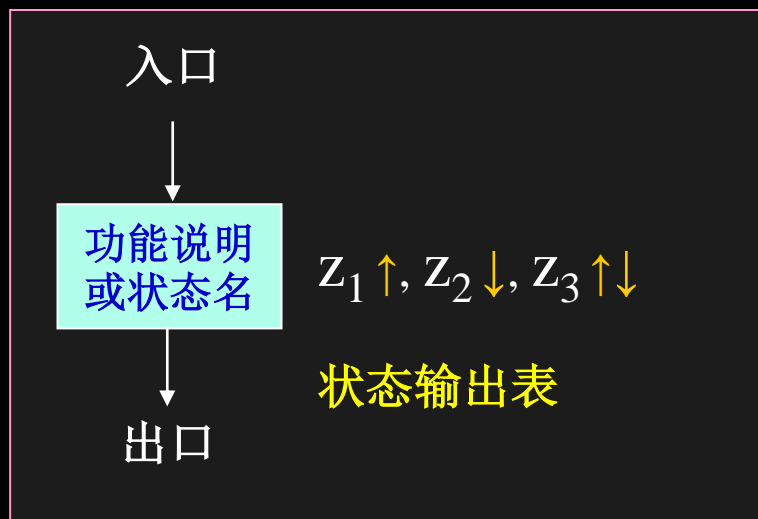
用三种符号：矩形状态框、菱形判别框、椭圆形条件框。



## 1)、基本符号

状态框表示系统必须具备的状态，判别框及条件输出框不表示状态，只表示某状态框在不同的输入条件下的分支出口及条件输出。用一个状态框及若干个判别框或条件输出框，组成一个状态单元。

如果在某状态下的输出与输入无关，即 Moore 型输出，则该状态输出可标注在状态框旁的状态输出表中，且这个状态单元必定不包括条件输出框。



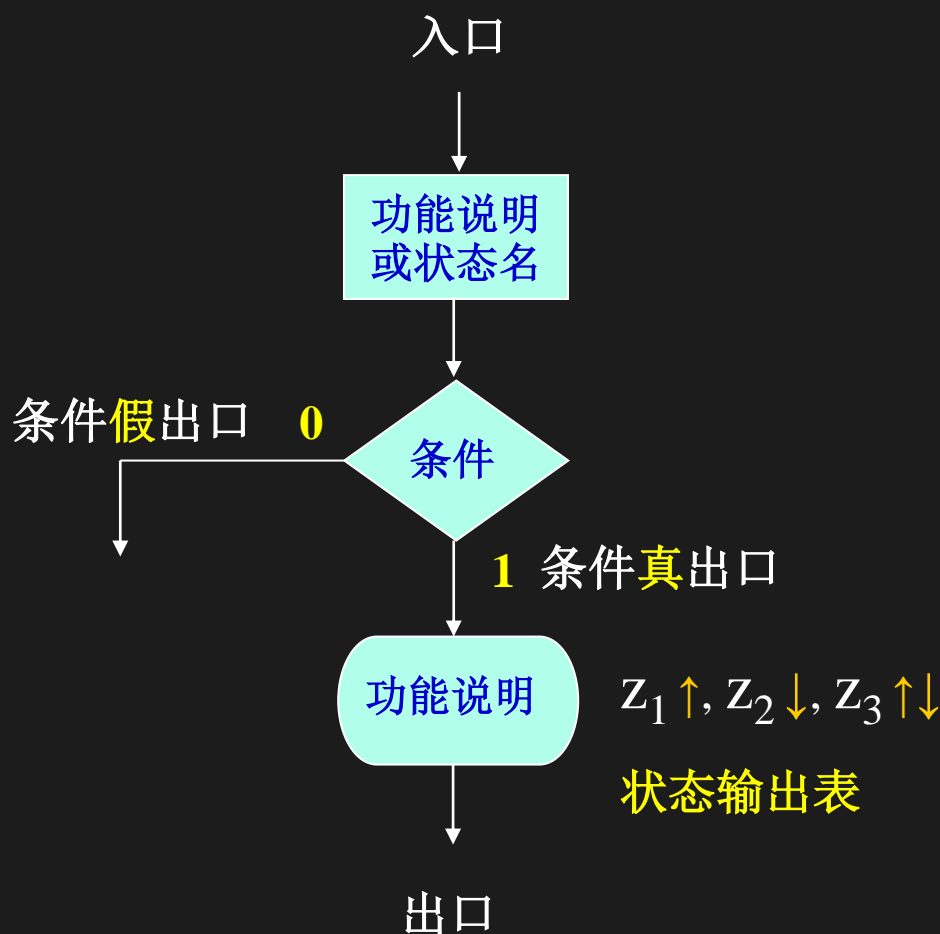
$z_1 \uparrow$  表示进入状态state，输出 $z_1$ 有效。

$z_2 \downarrow$  表示进入状态state，输出 $z_2$ 无效。

$z_3 \updownarrow$  表示进入状态state，输出 $z_3$ 有效，  
并在退出状态state后，输出 $z_3$ 无效。

# 1)、基本符号

如果在某状态下的输出与输入有关，即 **Mealy 型** 输出，则该状态输出要标注在菱形判别框下的椭圆形条件框旁的状态输出表中。



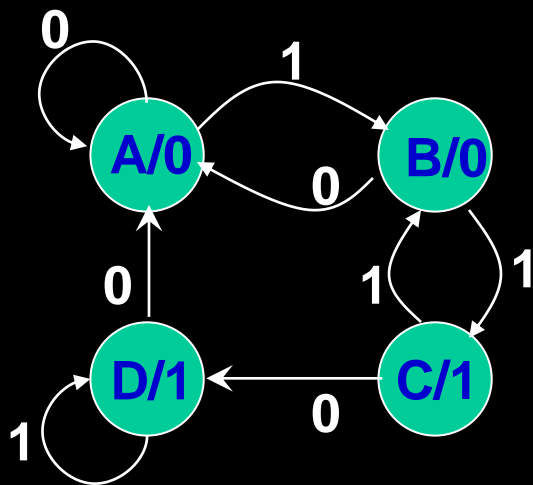
$z_1 \uparrow$  表示进入状态,  $z_1$  有效。  
 $z_2 \downarrow$  表示进入状态,  $z_2$  无效。  
 $z_3 \uparrow \downarrow$  表示进入状态,  $z_3$  有效,  
并在退出状态后,  $z_3$  无效。

## 2) 从状态图得到逻辑流程图

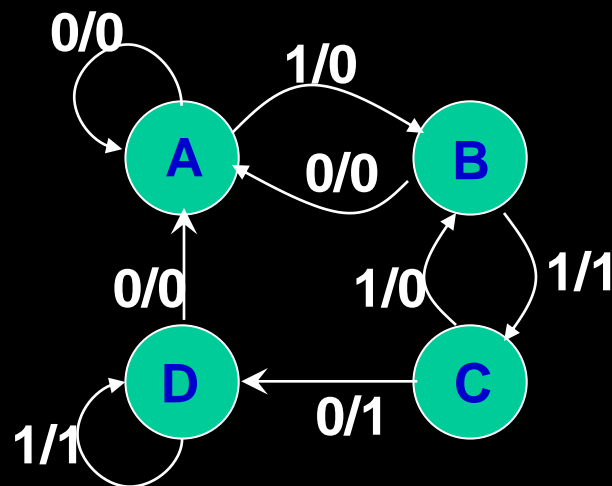
逻辑流程图上的一个状态框及若干个判别框或条件输出框所组成一个状态单元对应了状态图上的一个状态和它的输入输出。

如果某状态的输出与输入有关(**Mealy型**)，则逻辑流程图中对应的状态单元必定包括有**条件输出框**；

如果某状态的输出与输入无关(**Moore型**)，则逻辑流程图中对应的状态单元必定**没有**条件输出框。

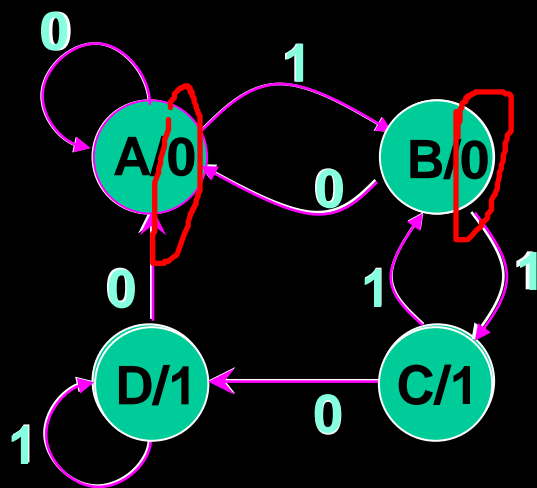


Moore 型状态图



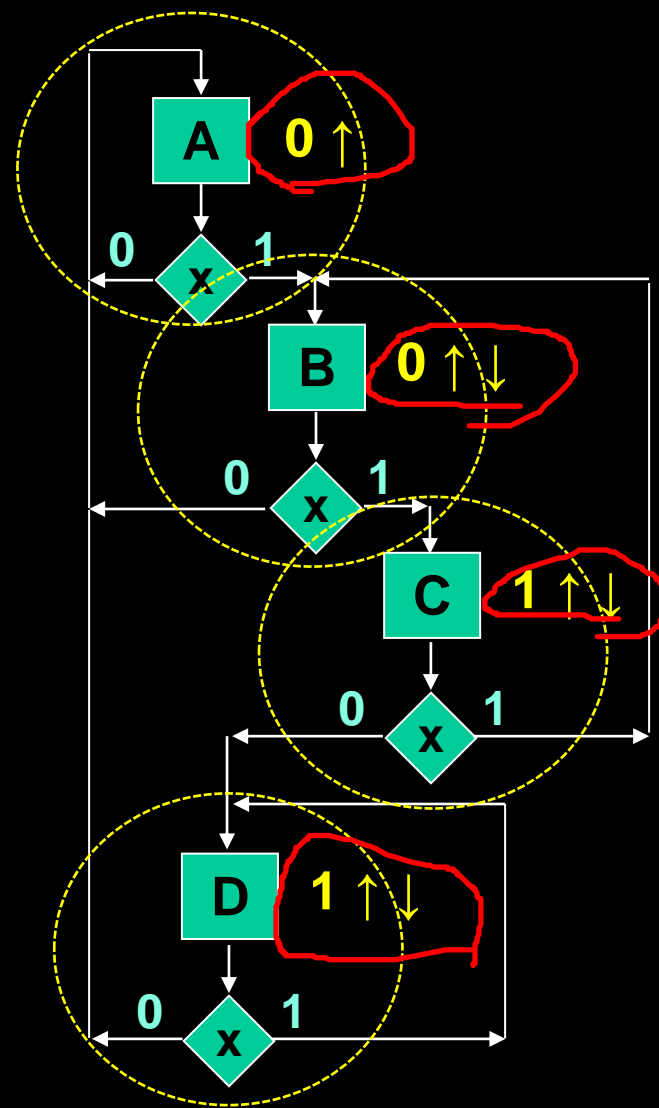
Mealy 型状态图

# Moore型



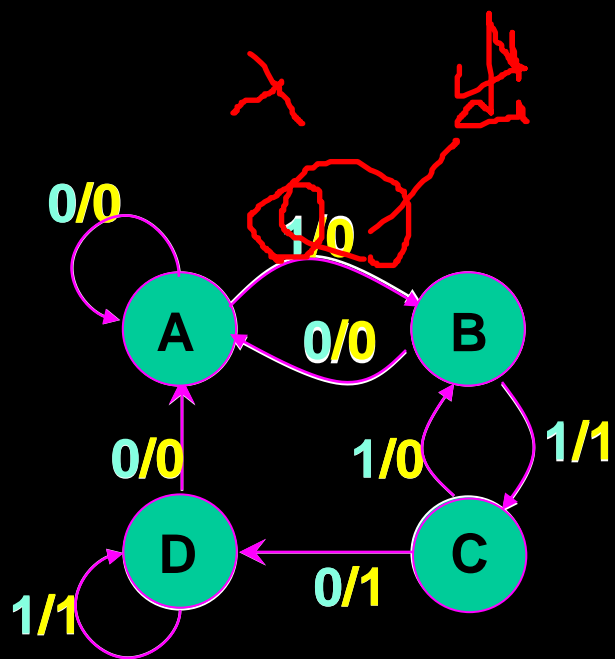
(a) 状态图

状态机

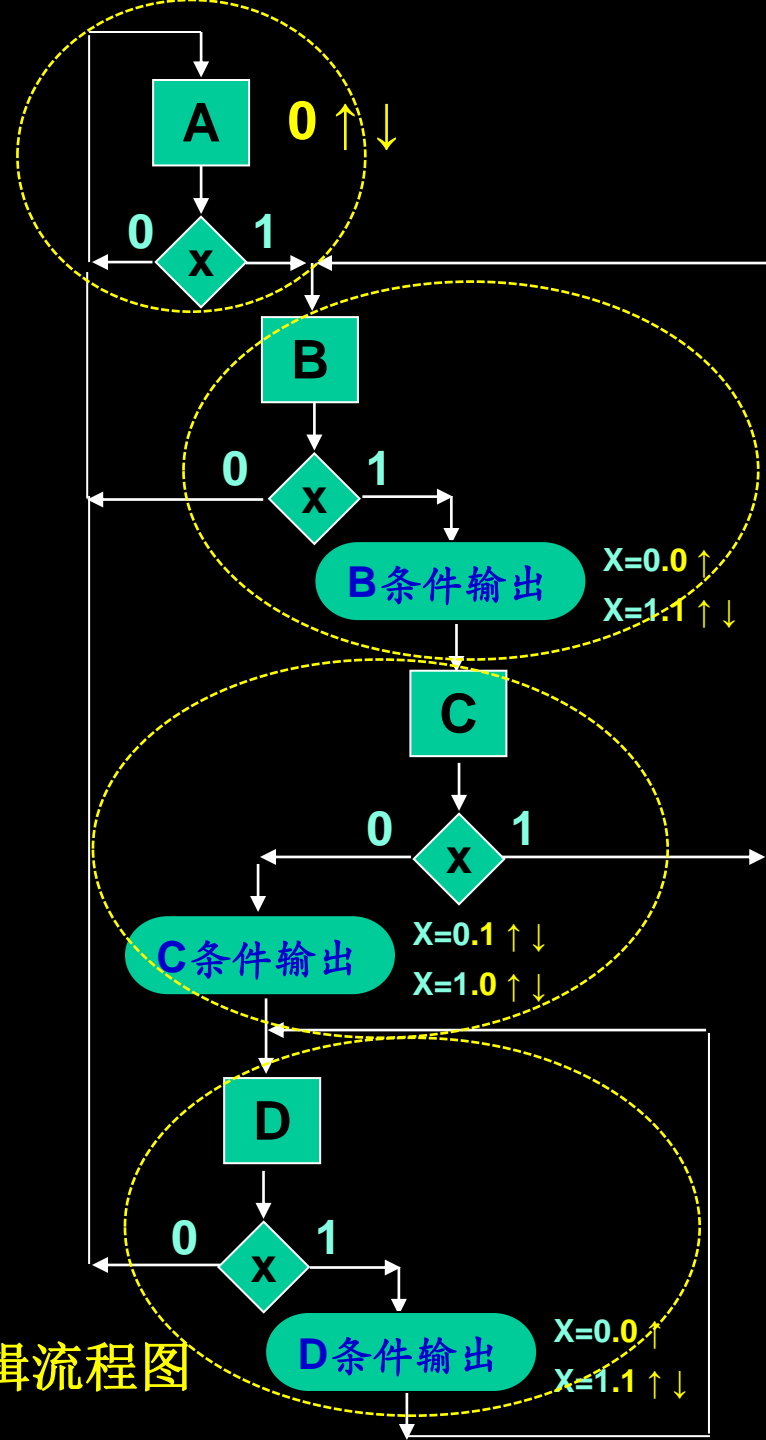


(b) 逻辑流程图

# Mealy型



(a) 状态图



(b) 逻辑流程图

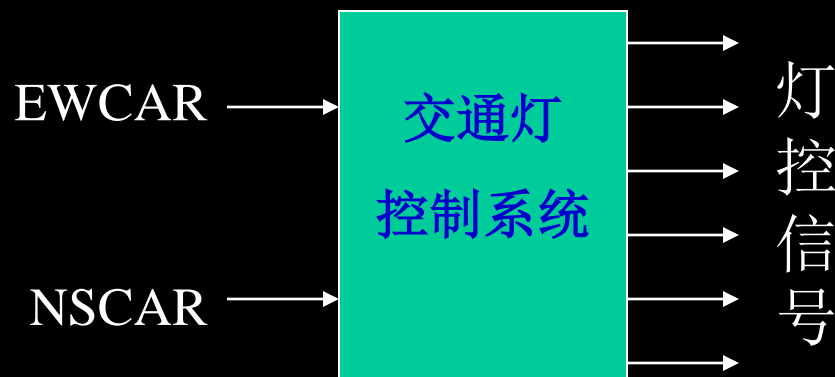
### 3、逻辑流程图的应用

逻辑流程图的描述过程是一个逐步细化（深化）的过程。它既便于设计者发现和改进信息处理过程中的错误，又是后续电路设计的依据。

## 例 设计一个十字路口交通灯控制系统。

设：东西道(**EW**)为主道，南北道(**NS**)为副道。

- 若 **EW**及**NS**均有车，  
则 **EW**每次通行 **60秒(绿灯)**， **NS**每次通行**40秒(绿灯)**，  
**EW**、**NS**轮流放行；
- 若 仅有一个通道有车，  
则禁止无车通道(**红灯**)；
- 若 两通道均无车，  
则**NS**禁止， **EW** 放行；
- 若 通道转换时，  
两通道均需停车**3秒(黄灯)**。



框图

变量定义

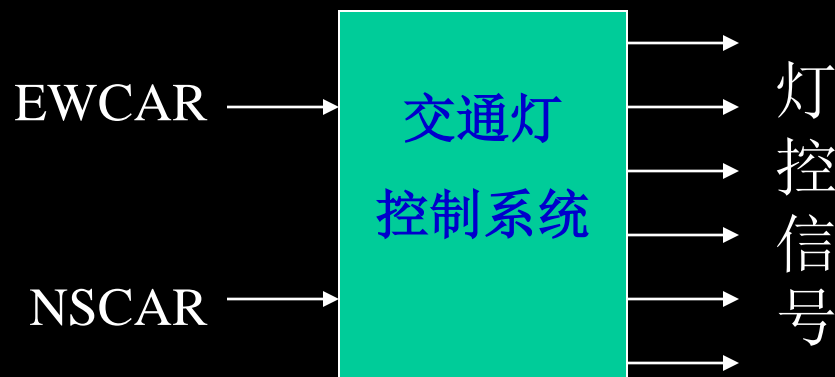
监测器输出			输出灯光信号
NSCAR EWCAR (由 ET、ST WT、NT 生成)			NS <b>Red</b> NS <b>Green</b> NS <b>Yellow</b> EW <b>Red</b> EW <b>Green</b> EW <b>Yellow</b>



## 例 设计一个十字路口交通灯控制系统。

设：东西道(**EW**)为主道，南北道(**NS**)为副道。

- 若 **EW**及**NS**均有车，  
则 **EW**每次通行 **60秒(绿灯)**， **NS**每次通行**40秒(绿灯)**，  
**EW**、**NS**轮流放行；
- 若 仅有一个通道有车，  
则禁止无车通道(**红灯**)；
- 若 两通道均无车，  
则**NS**禁止， **EW** 放行；
- 若 通道转换时，  
两通道均需停车**3秒(黄灯)**。



框图

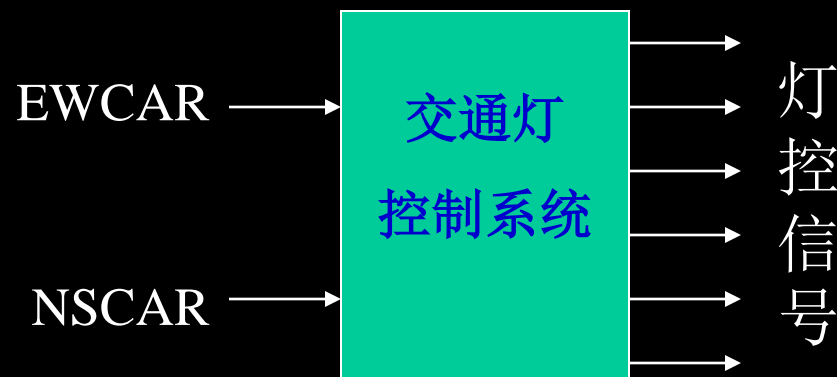
变量定义

监测器输出	定时器输出		输出灯光信号
NSCAR	TM60		NS <b>Red</b>
EWCAR	TM40		NS <b>Green</b>
(由	TM3		NS <b>Yellow</b>
ET、ST	(计数器的		EW <b>Red</b>
WT、NT	进位输出)		EW <b>Green</b>
生成)			EW <b>Yellow</b>

## 例 设计一个十字路口交通灯控制系统。

设：东西道(**EW**)为主道，南北道(**NS**)为副道。

- 若 **EW**及**NS**均有车，  
则 **EW**每次通行 **60秒(绿灯)**， **NS**每次通行**40秒(绿灯)**，  
**EW**、**NS**轮流放行；
- 若 仅有一个通道有车，  
则禁止无车通道(**红灯**)；
- 若 两通道均无车，  
则**NS**禁止， **EW** 放行；
- 若 通道转换时，  
两通道均需停车**3秒(黄灯)**。

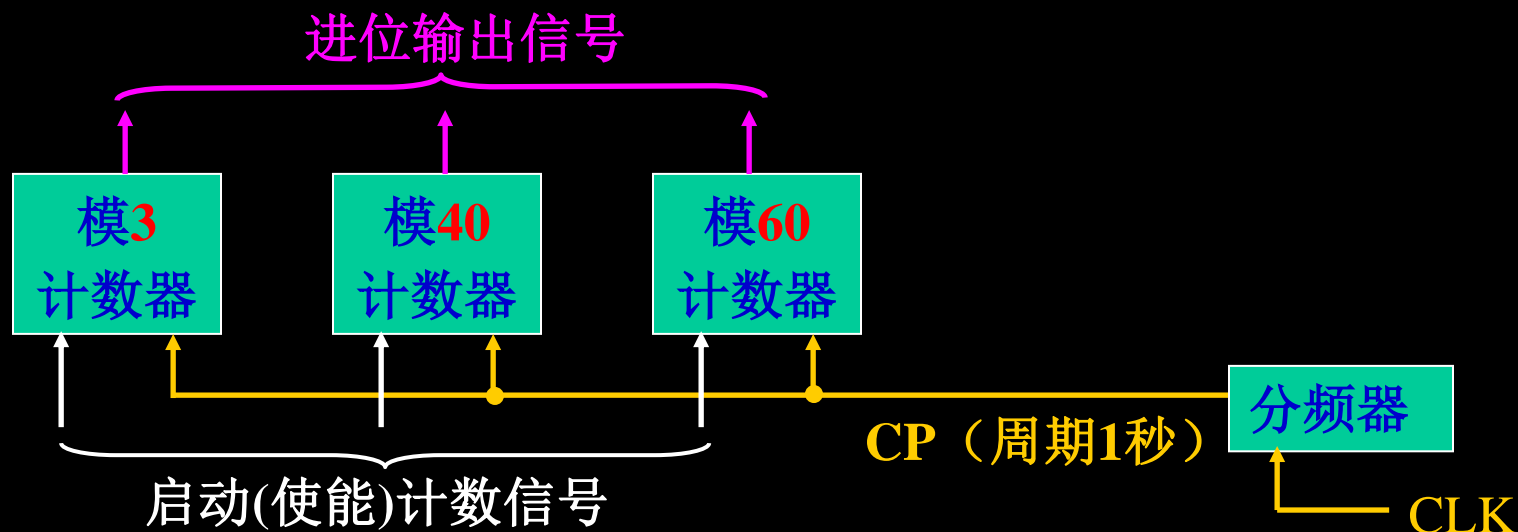


框图

变量定义

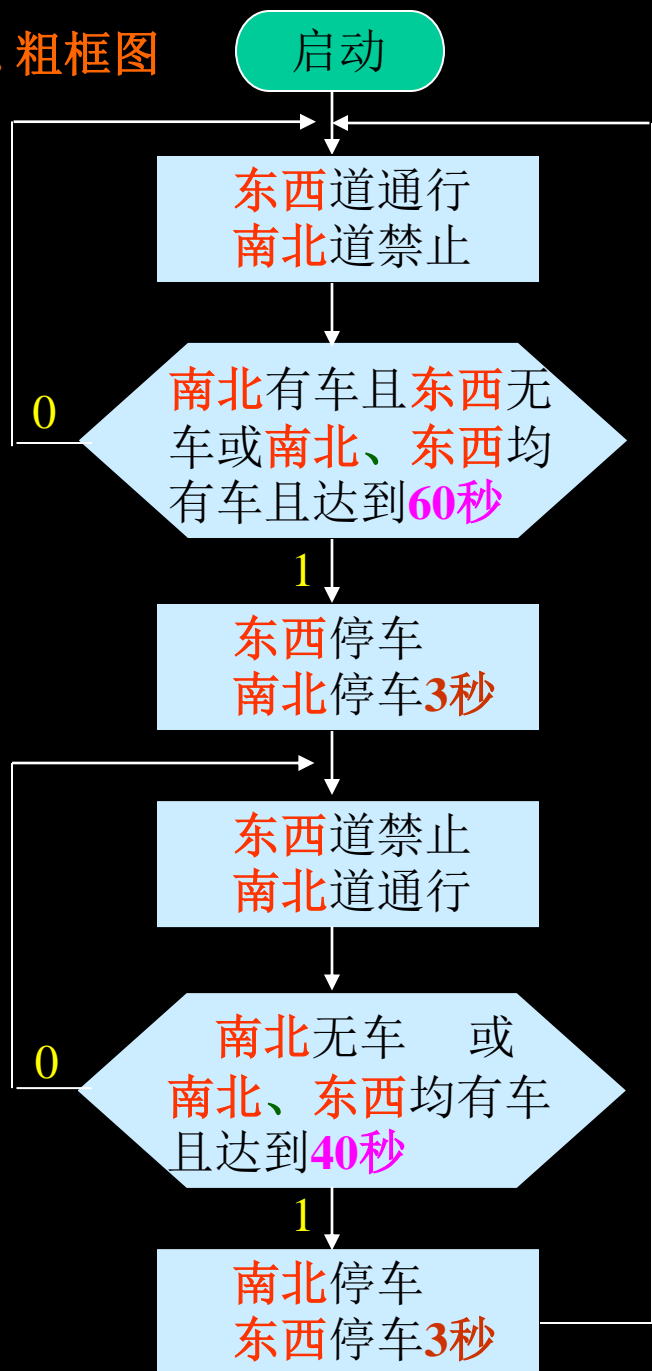
监测器输出	定时器输出	定时器使能输入	输出灯光信号
NSCAR	TM60	ENTM60	NS <b>Red</b>
EWCAR	TM40	ENTM40	NS <b>Green</b>
(由 ET、ST WT、NT 生成)	TM3 (计数器的 进位输出)	ENTM3	NS <b>Yellow</b>
			EW <b>Red</b>
			EW <b>Green</b>
			EW <b>Yellow</b>

系统配有三个**3秒**、**40秒**和**60秒**的定时器，如下：



**定时器**在系统中起到时间**节拍**指挥的作用，各部件均按照统一的时间节拍协调地工作。

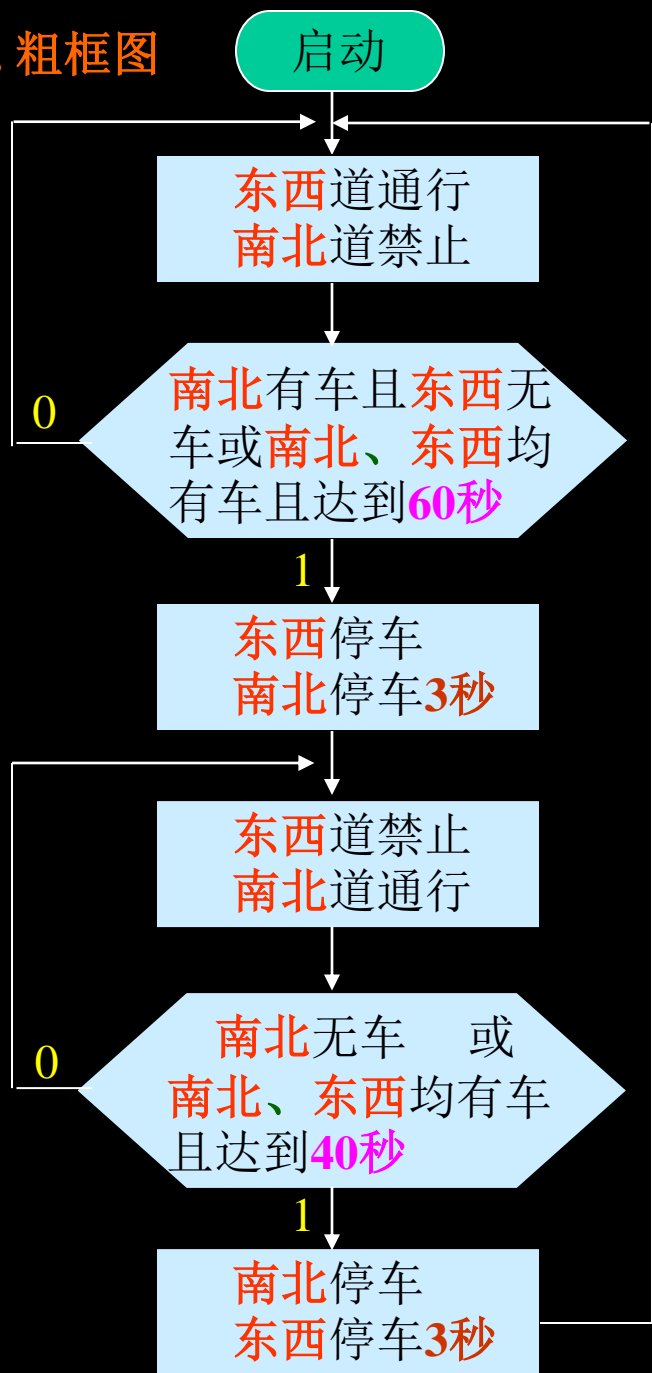
## a. 粗框图



## 十字路口交通灯控制器逻辑框图

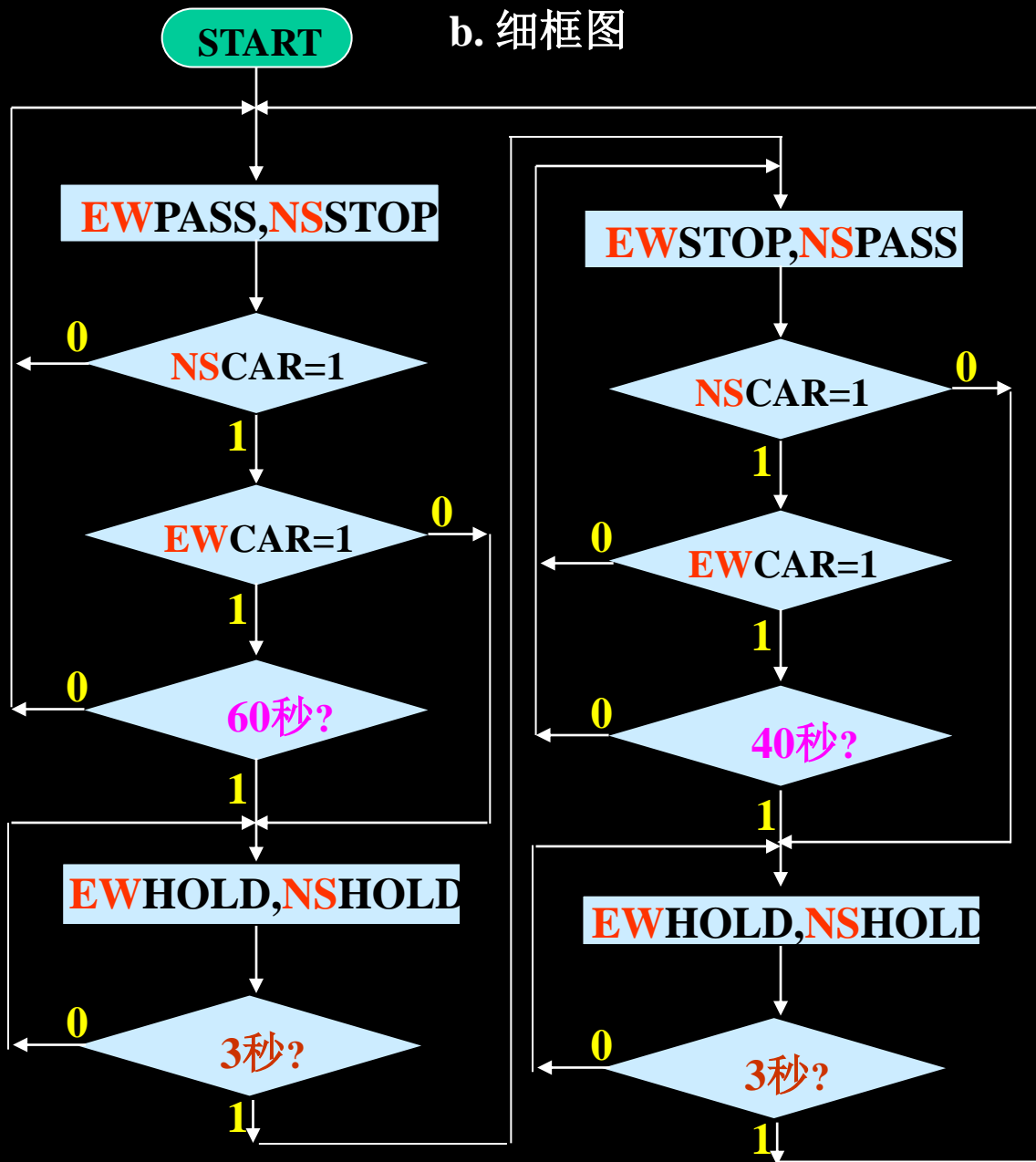
- 若 **EW**及**NS**均有车，则**EW**每次通行**60秒(绿灯)**，**NS**每次通行**40秒(绿灯)**，**EW**、**NS**轮流放行；
- 若 仅有一个通道有车，则禁止无车通道(**红灯**)；
- 若 两通道均无车，则**NS**禁止，**EW**放行；
- 若 通道转换时，两通道均需停车**3秒(黄灯)**。

a. 粗框图

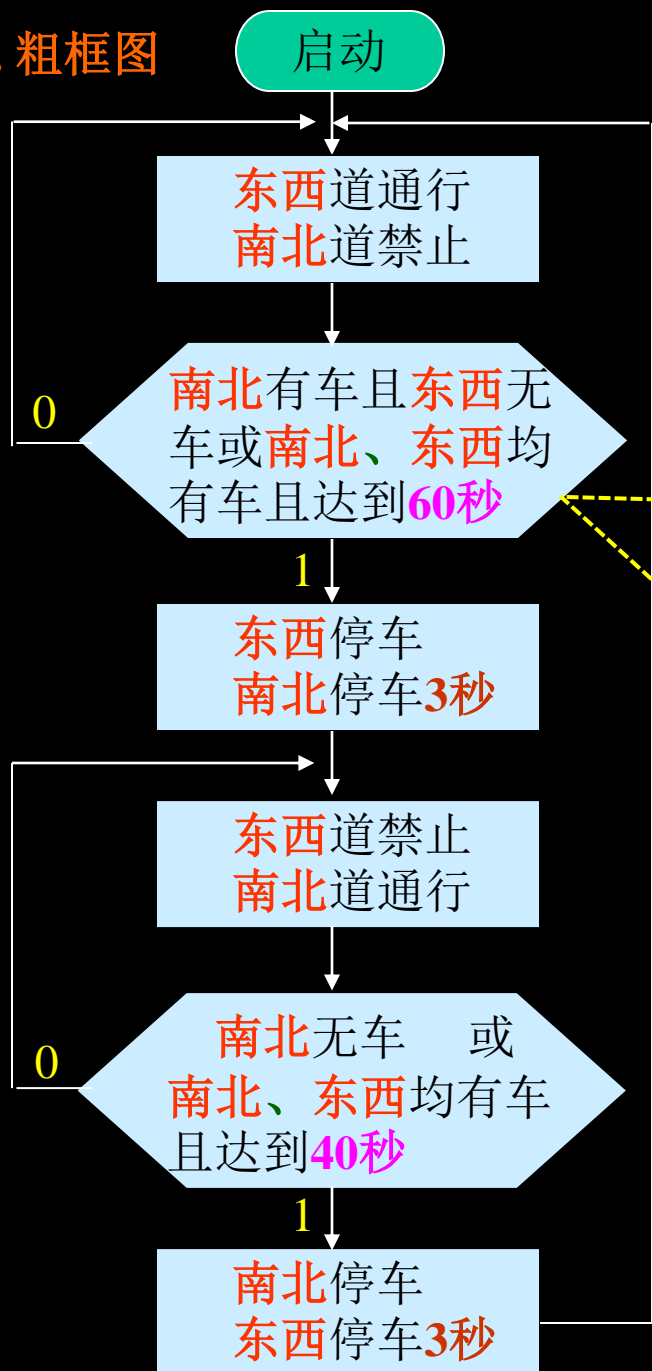


# 十字路口交通灯控制器逻辑框图

b. 细框图

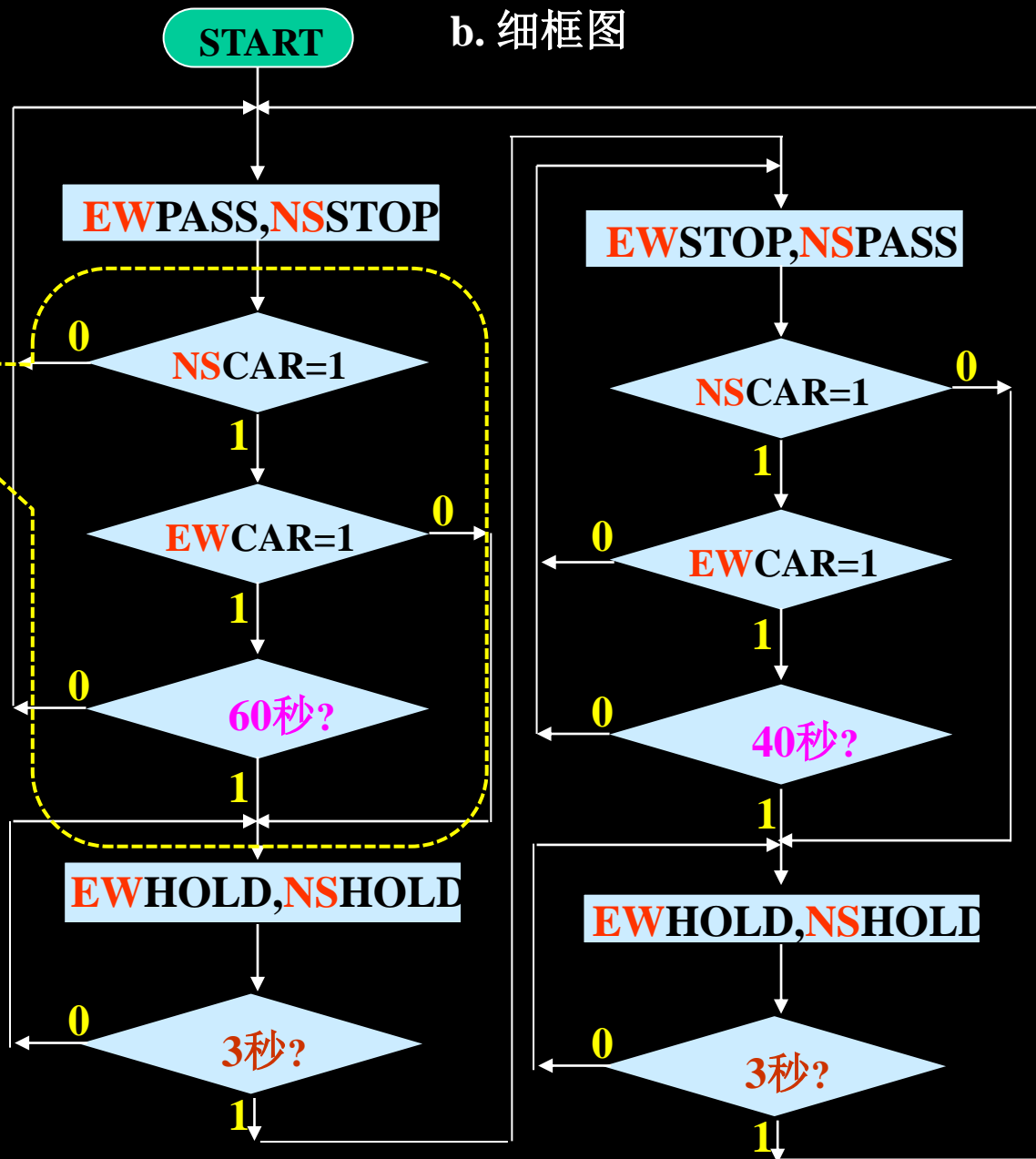


a. 粗框图

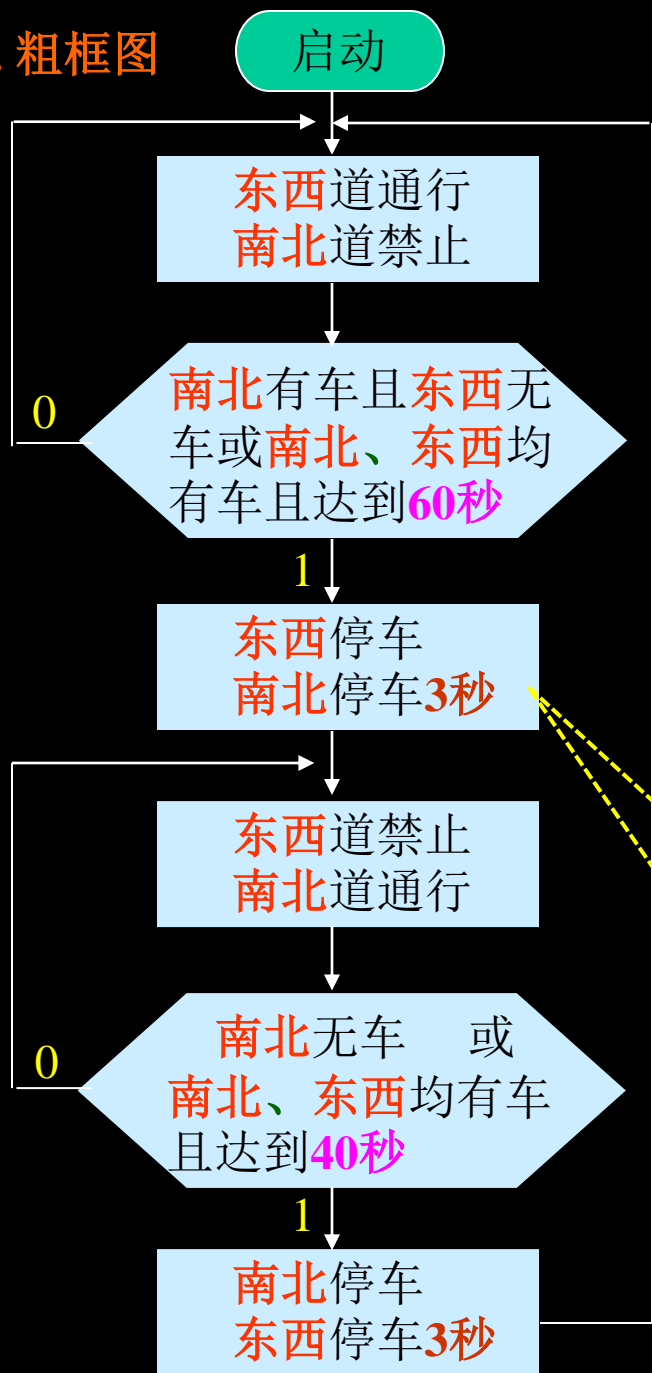


十字路口交通灯控制器逻辑框图

b. 细框图

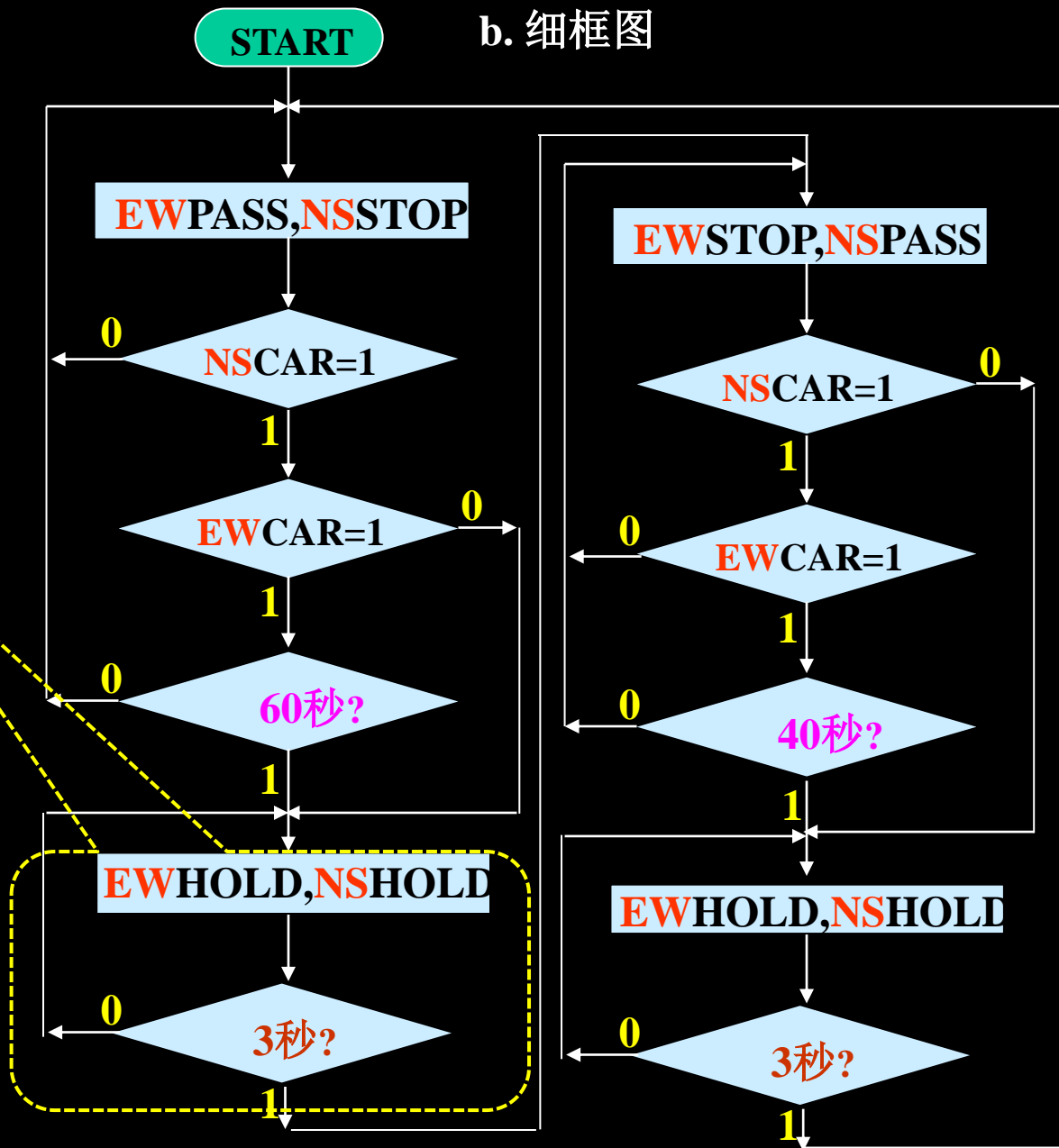


a. 粗框图

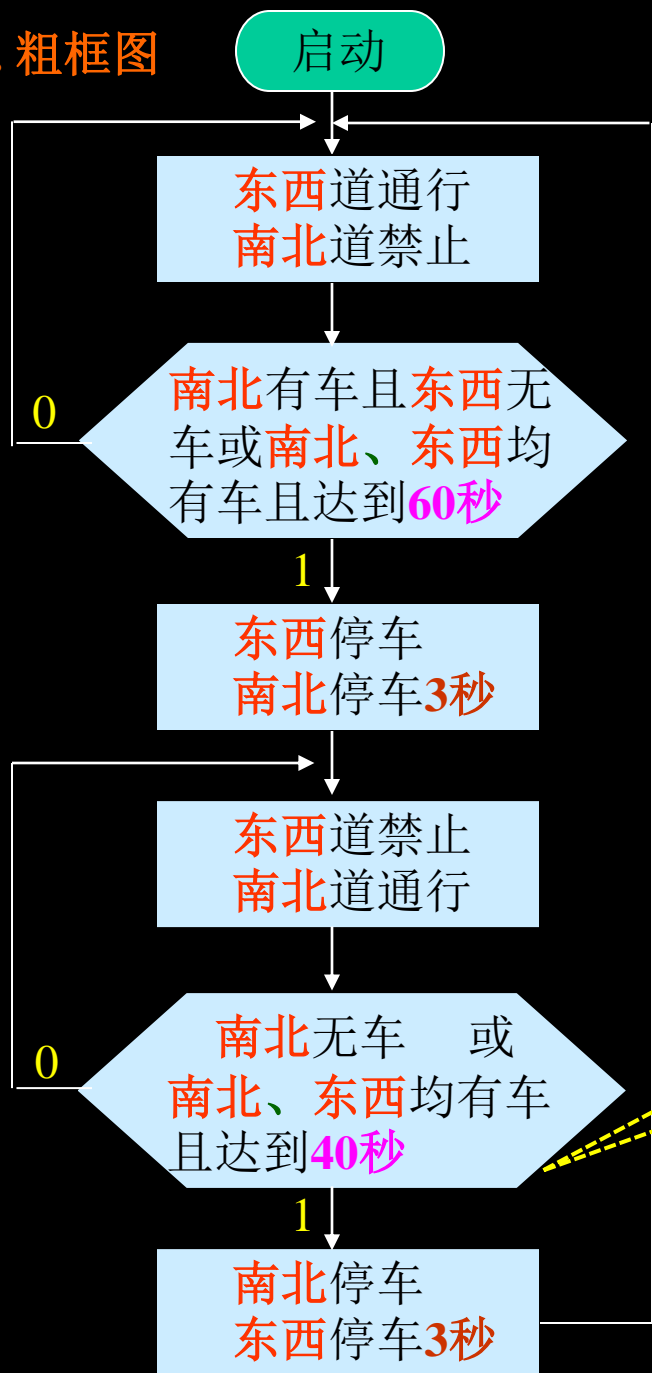


# 十字路口交通灯控制器逻辑框图

b. 细框图

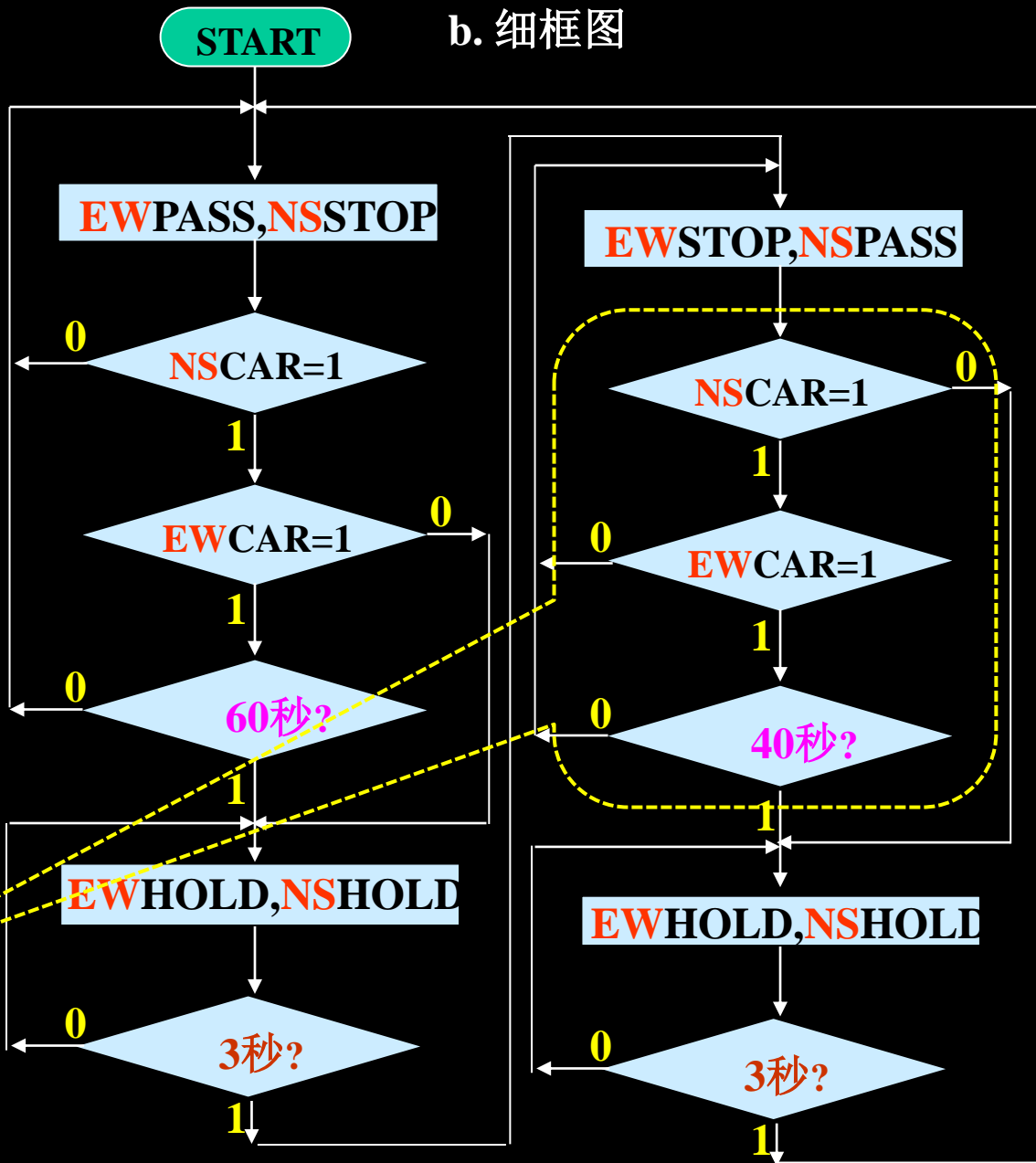


a. 粗框图



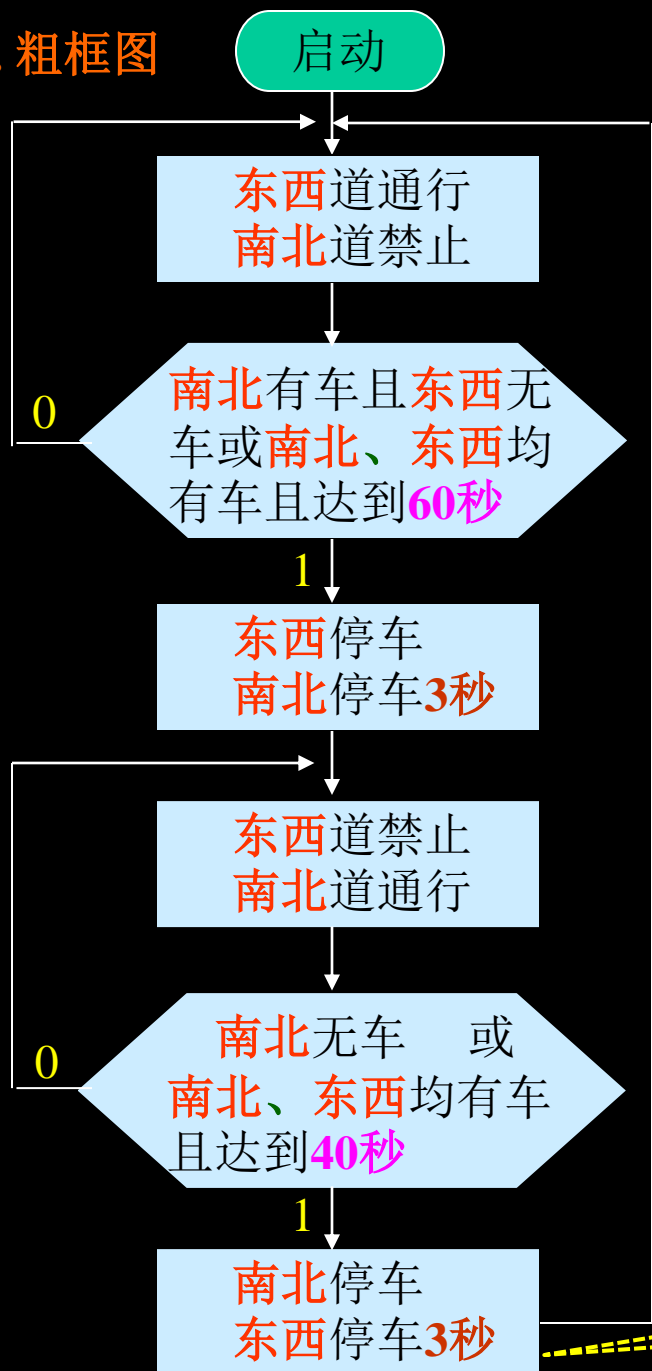
# 十字路口交通灯控制器逻辑框图

b. 细框图



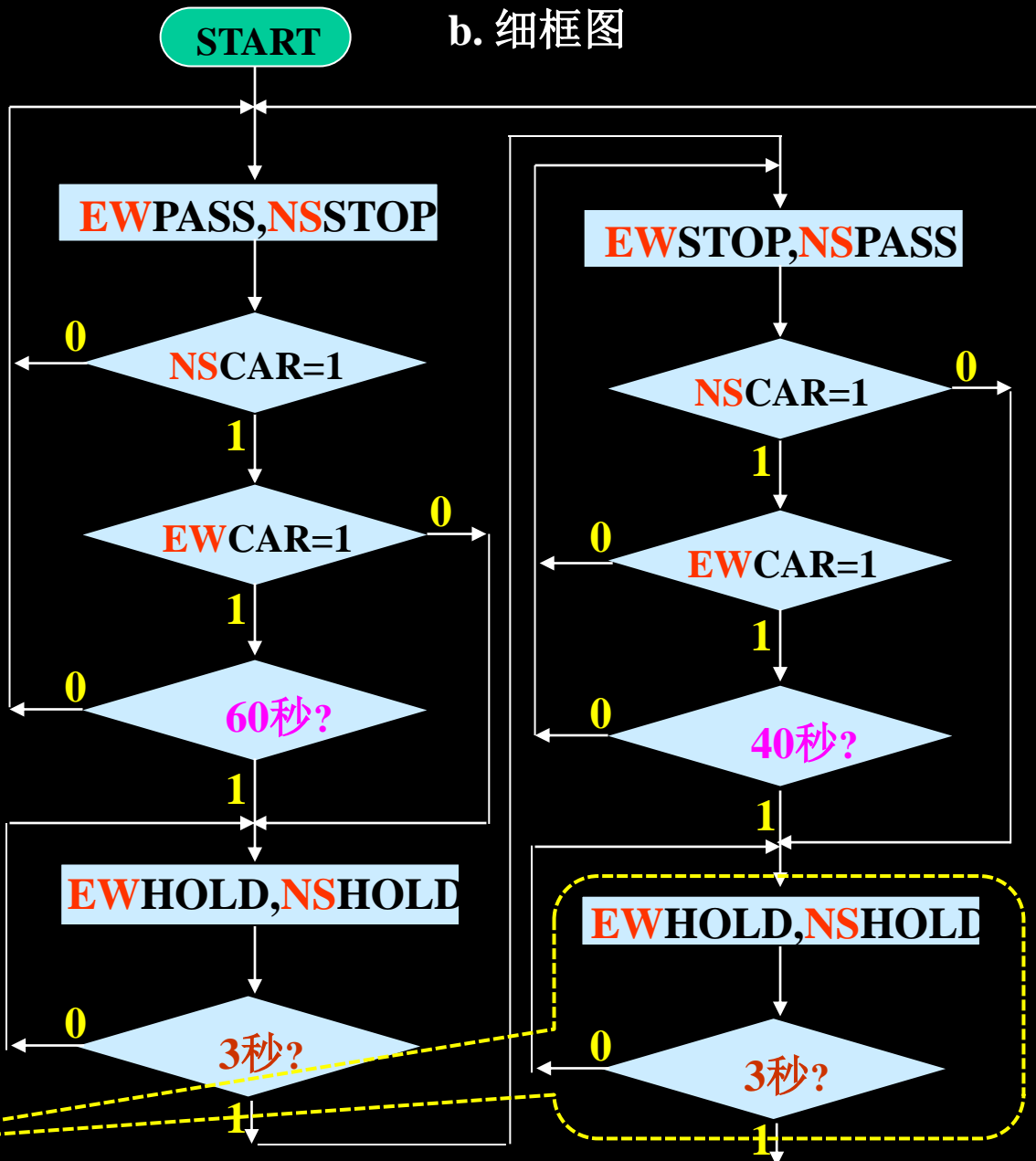


a. 粗框图



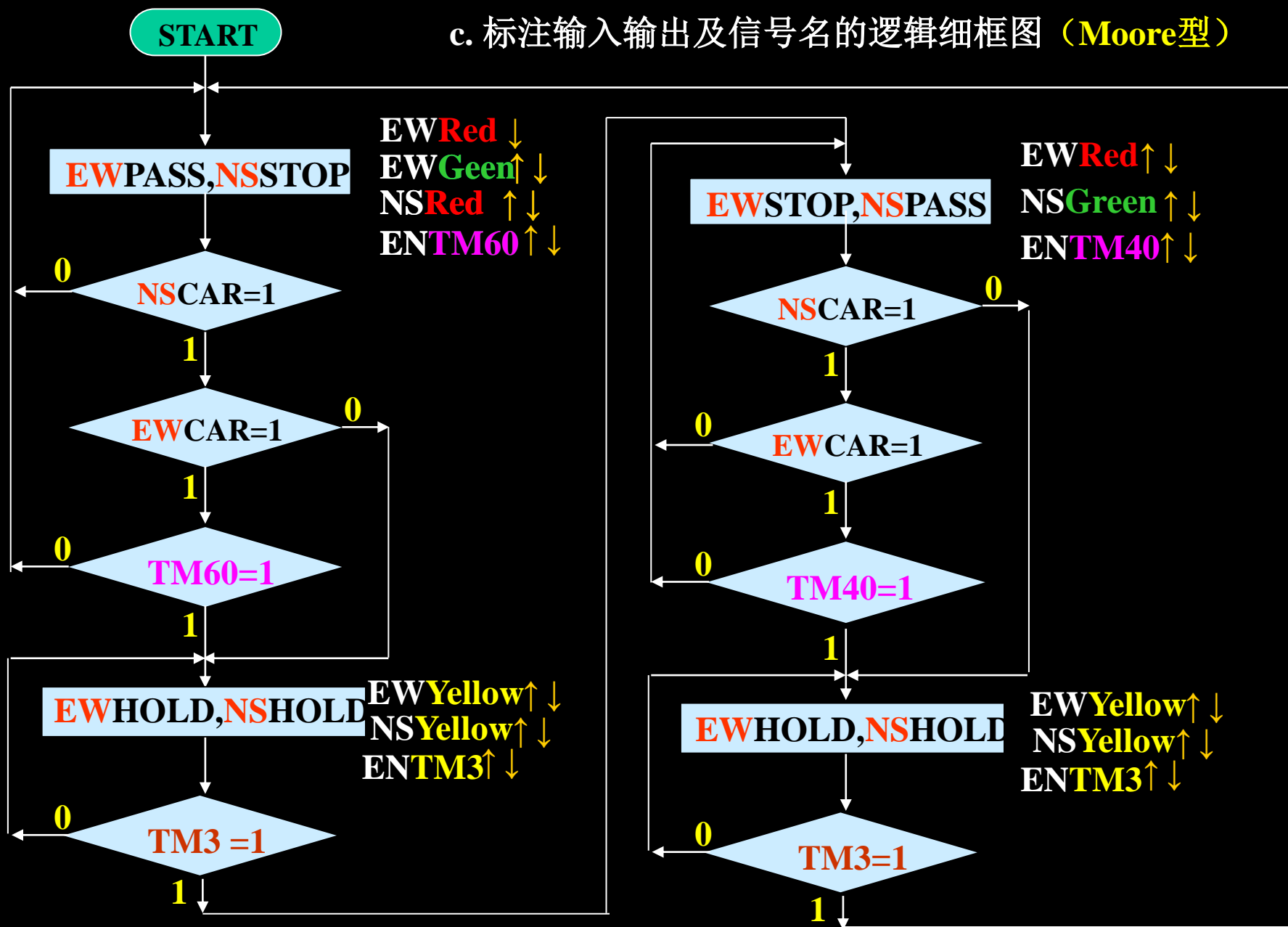
# 十字路口交通灯控制器逻辑框图

b. 细框图



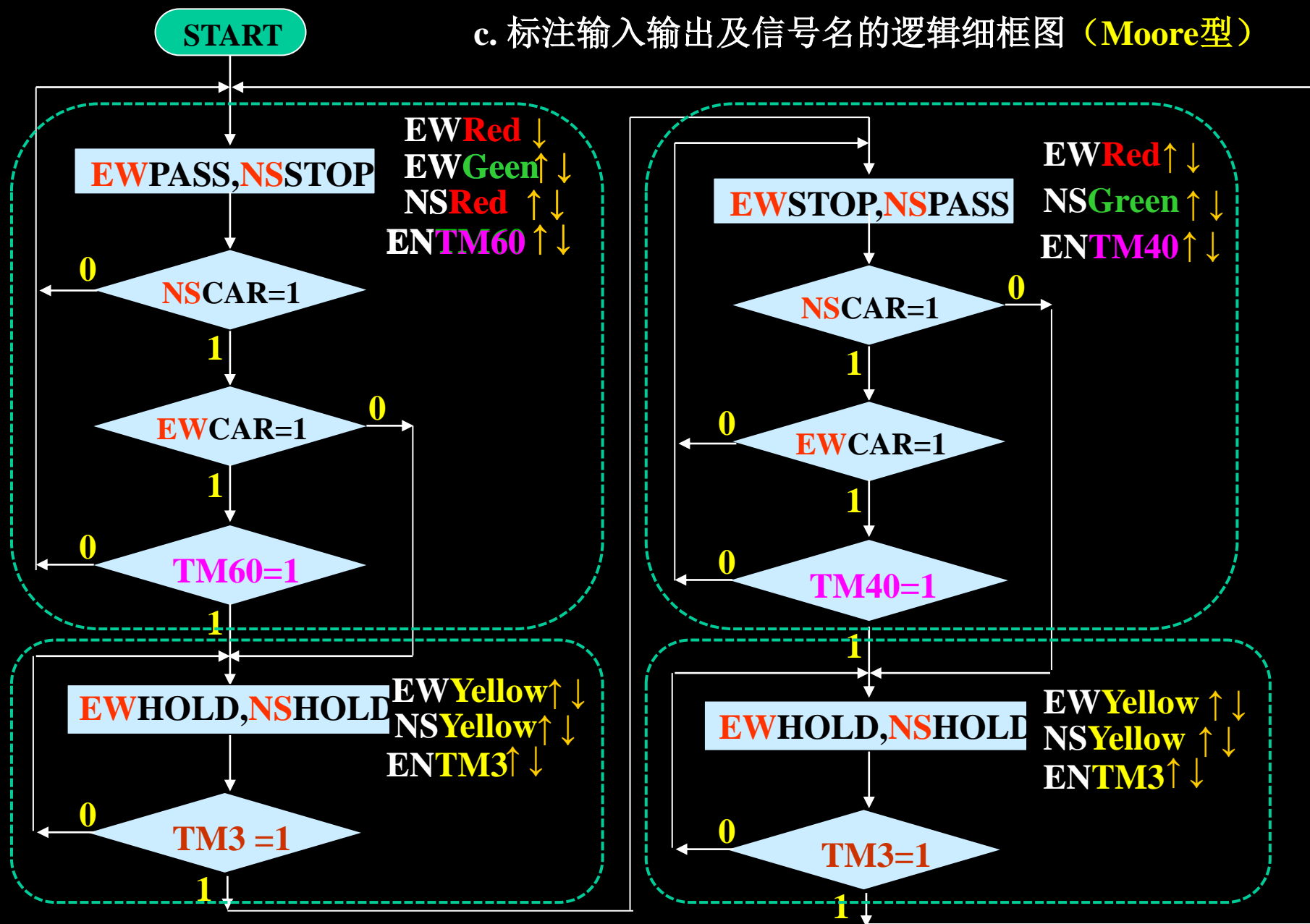
# 十字路口交通灯控制器逻辑流程图

c. 标注输入输出及信号名的逻辑细框图 (Moore型)



# 十字路口交通灯控制器逻辑流程图

c. 标注输入输出及信号名的逻辑细框图 (Moore型)

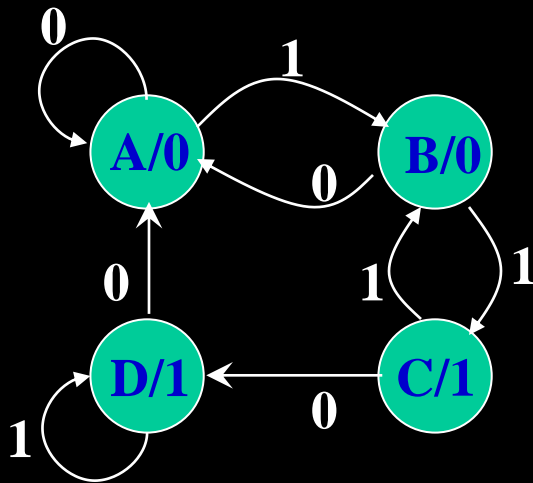


### 3、从状态图得到逻辑流程图

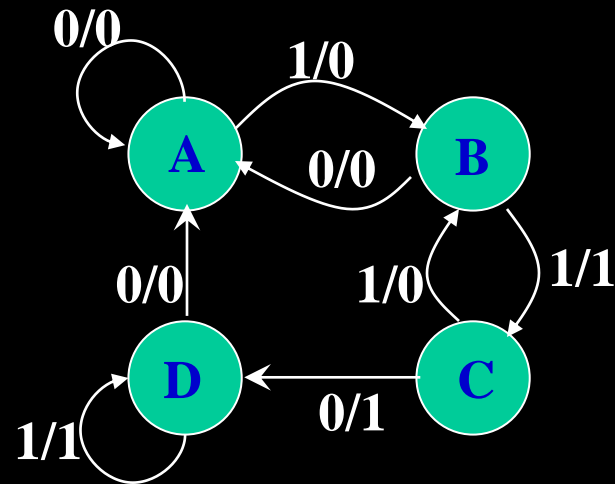
逻辑流程图上的一个状态框及若干个判别框或条件输出框所组成一个状态单元对应了状态图上的一个状态和它的输入输出。

如果某状态的输出与输入有关(**Mealy型**)，则逻辑流程图中对应的状态单元必定包括**有条件输出框**；

如果某状态的输出与输入无关(**Moore型**)，则逻辑流程图中对应的状态单元必定**没有条件输出框**。



Moore 型状态图



Mealy 型状态图

## 5.2.4 ASM图

逻辑流程图是数字系统中使用得最广泛的一种非形式化的描述工具，但它的规范性不够。经过不断改进，将流程图改造成描述数字系统硬件的形式化工具

——算法状态机图ASM

*(Algorithmic State Machine Chart)*。

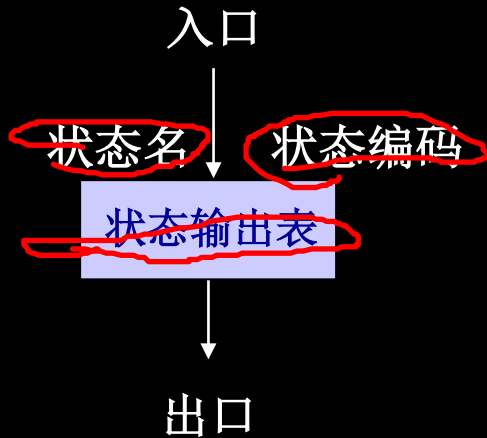
**ASM图**是形式化描述数字系统的流程图，是按照一组简单和明确的规则绘制的。在时序电路的ASM图中，**每一个状态由一个ASM块来表示**。一个ASM块是一种小型的符号化结构图形，由它指出了**当前状态**、它的输出和次态的**条件**。一个时序数字系统的所有状态的ASM块，按照一定的顺序联结成一个网状结构，由此产生的ASM图可用来精确地描述系统所具有的功能。

# 1、基本符号

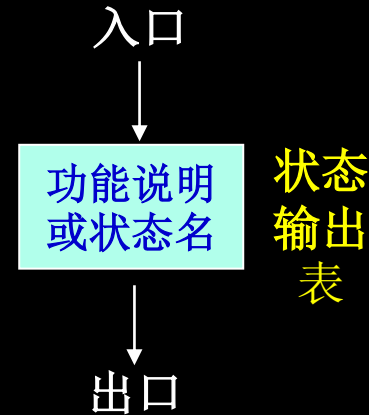
一个ASM块至多包含三种不同类型的符号：

**矩形状态框、菱形判别框和条件输出框。**

注意ASM图的标注（左边）与流程图的（右边）不同。



(a) ASM的状态框



(a) 流程图的状态框

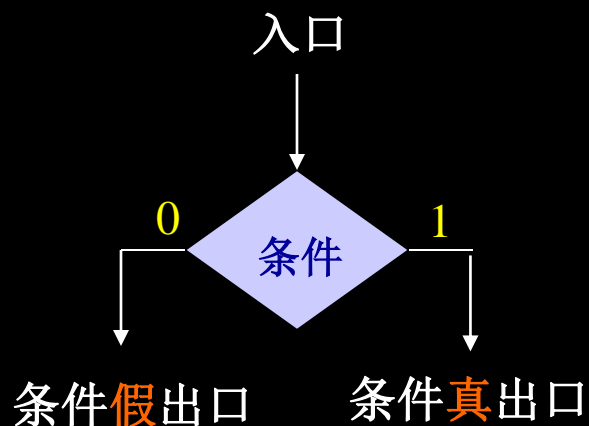
其中：状态编码为该状态下的触发器值。

# 1、基本符号

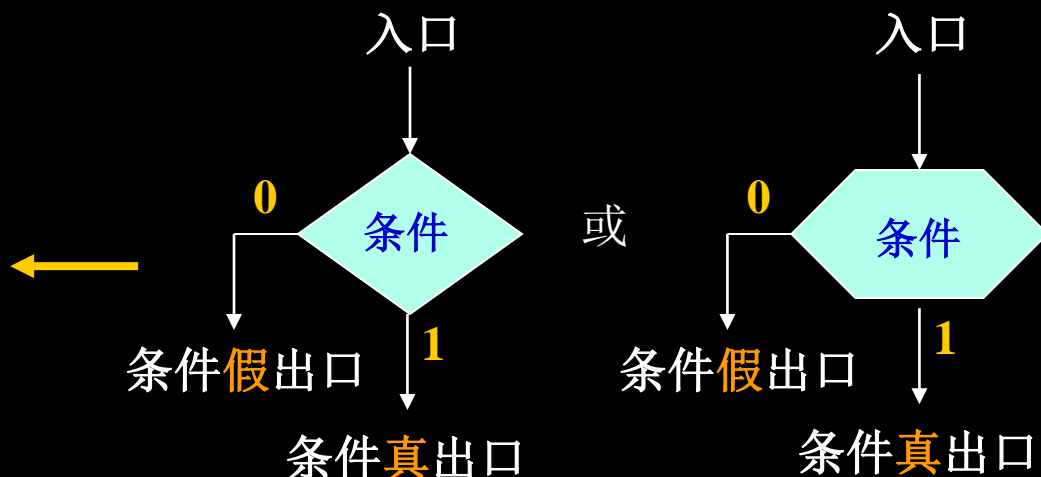
一个ASM块至多包含三种不同类型的符号：

矩形状态框、菱形判别框和条件输出框。

注意图的标注与流程图的不同。



(b) ASM的判别框



(b) 流程图的条件判别框

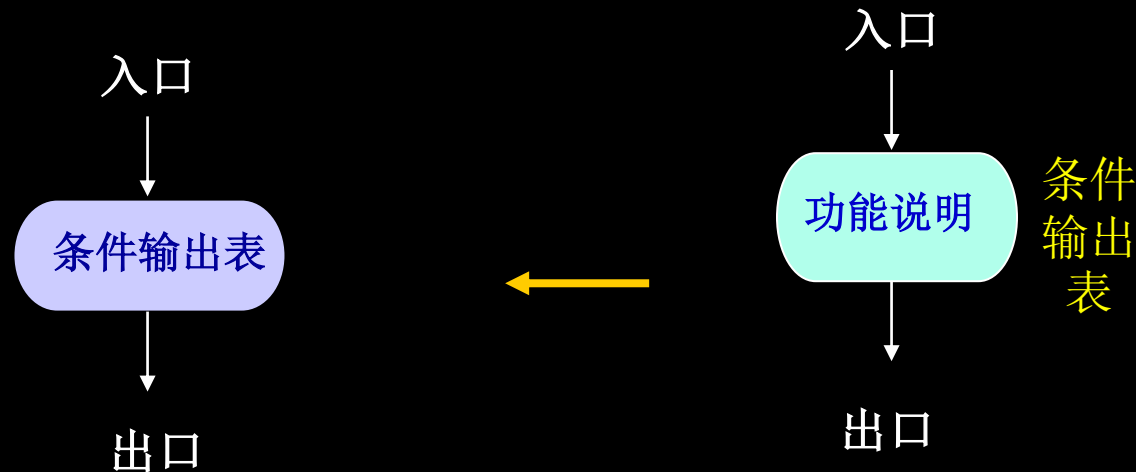
其中：框内的输入条件为布尔表达式。

# 1、基本符号

一个ASM块至多包含三种不同类型的符号：

**矩形状态框、菱形判别框和条件输出框。**

注意图的标注与流程图的不同。



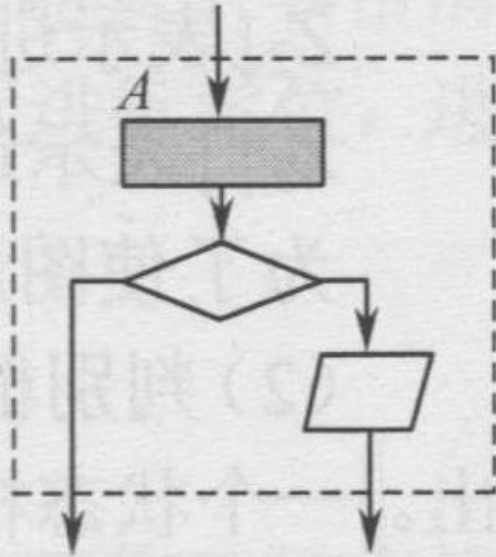
(c) 条件输出框

(c) 条件输出框

条件输出框的输入总是来自判别框，由这个判别框给出了输出所需要的条件，即在某一状态下，某个输出变量是输入变量的函数，就在条件输出框中填入**条件满足时产生的输出**。



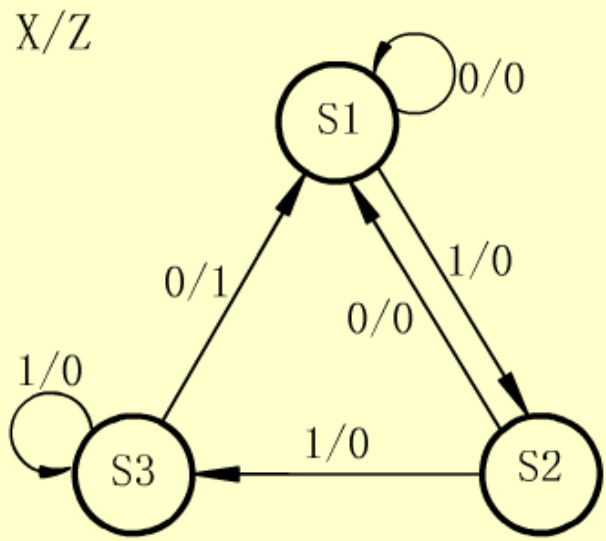
# 1、基本符号



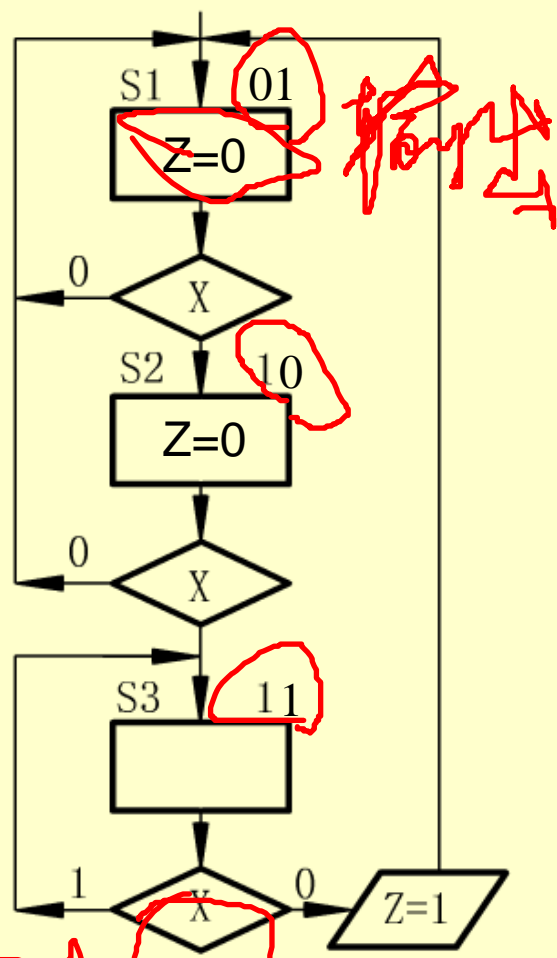
(d) 状态单元

一个**状态框**和若干个**判别框**或**条件输出框**组成一个**状态单元**。状态单元的入口必须是状态框的入口，出口可以有几个，但必须指向状态框。

【例3】 将图 5.10(a) 所示的米里机状态图转换成ASM流程图。

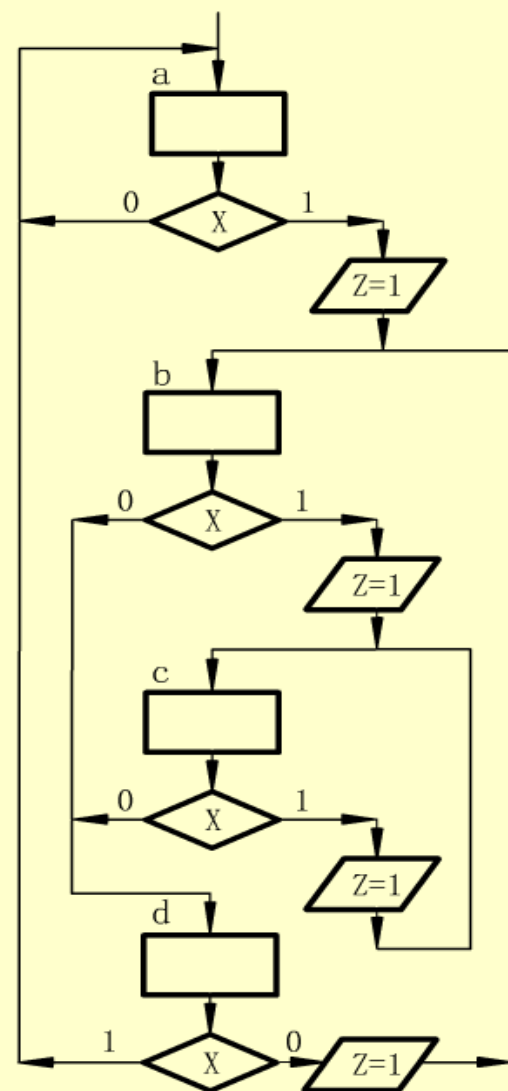
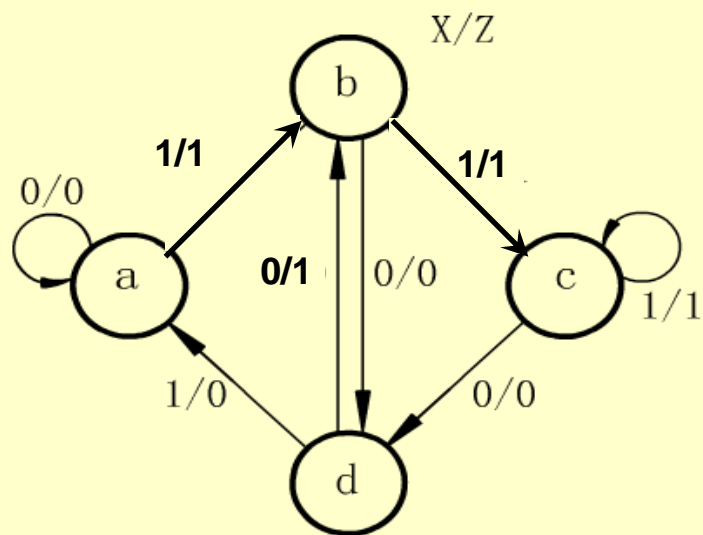


状态图



ASM流程图

【例4】 将图6.11 (a) 所示的四状态机转换成ASM流程图。



## 2、由逻辑流程图转换成ASM图

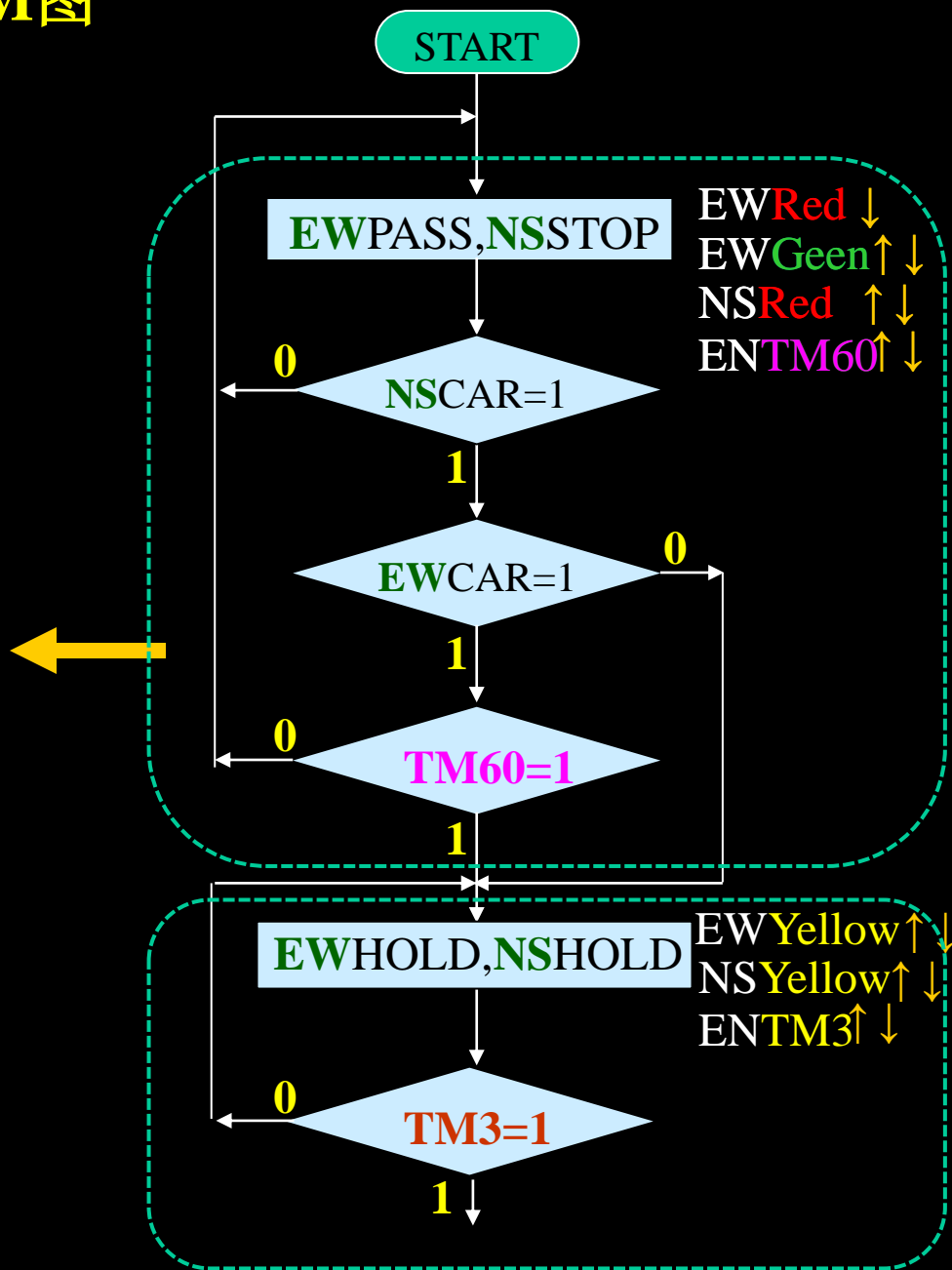
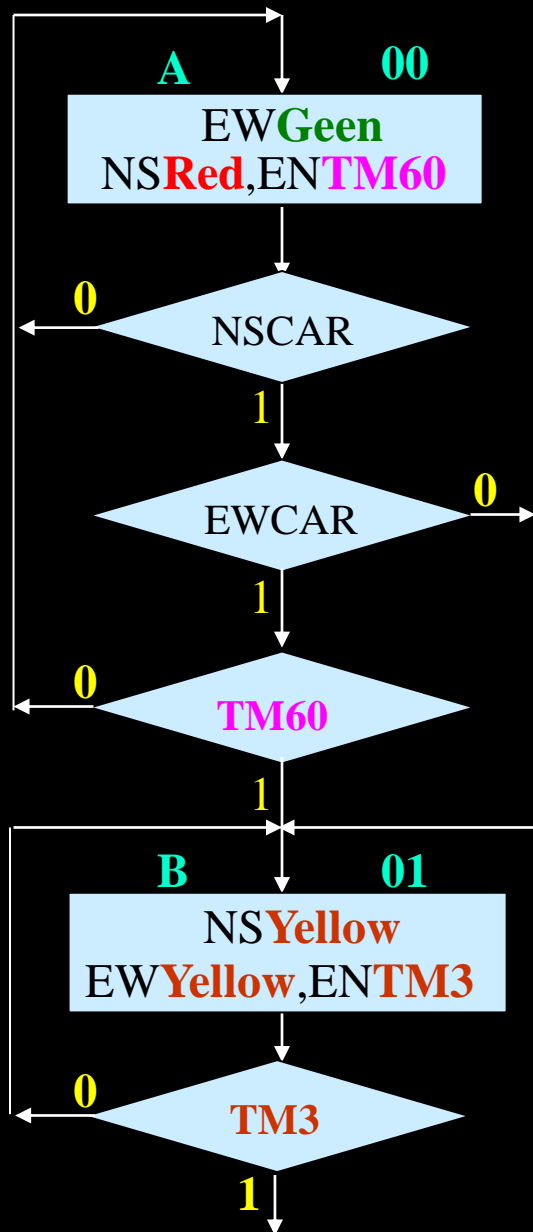
ASM图可以直接送入计算机辅助逻辑设计系统，由该系统自动完成控制单元的设计。

逻辑流程图可以很容易地转换成ASM图。

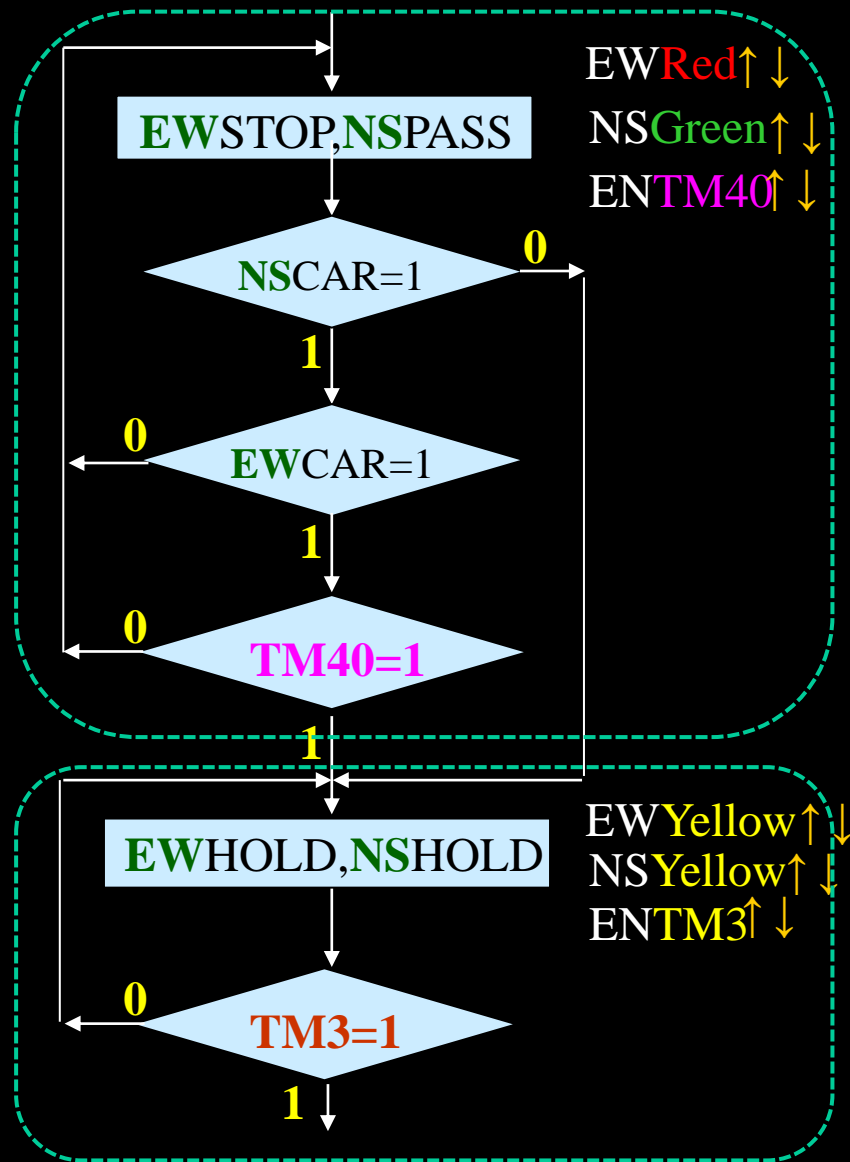
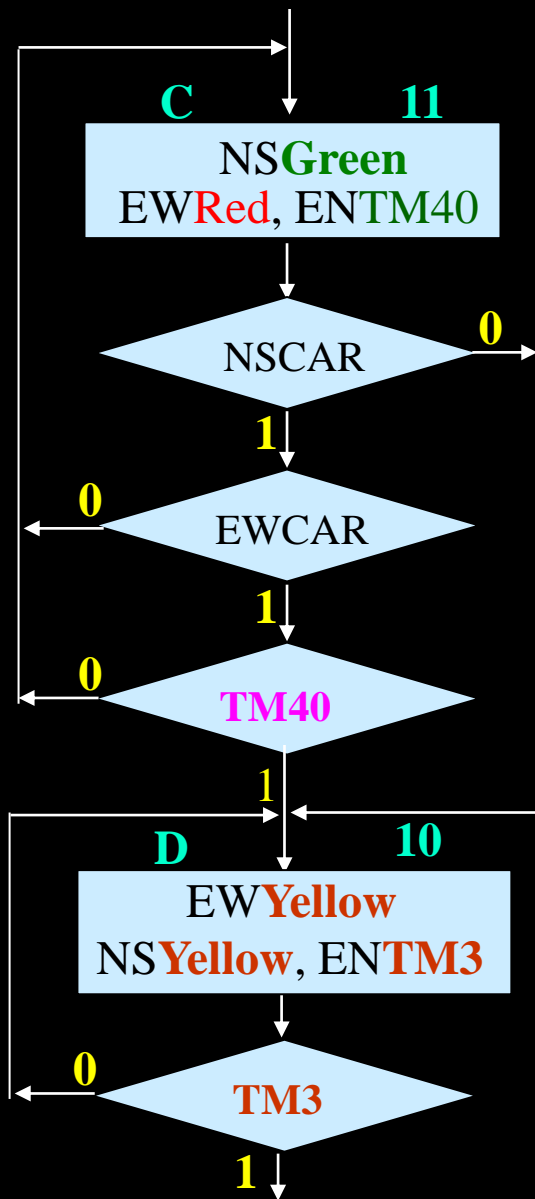
在ASM图中，所有的当前状态下的**Moore型**有效输出都应列在状态框内；

所有的当前状态下的满足输入条件的**Mealy型**有效输出都应列在条件输出框内。

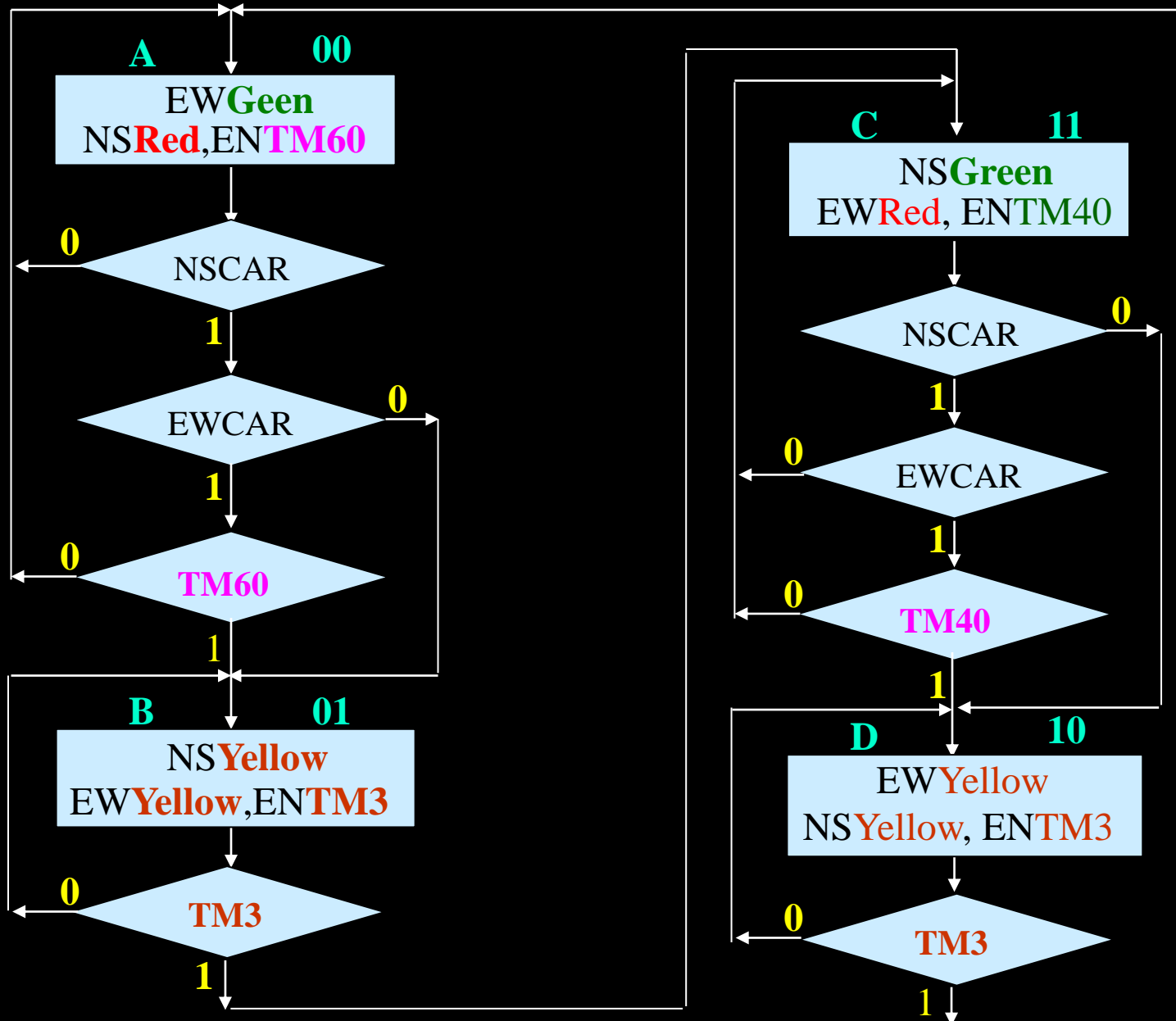
# 十字路口交通灯控制器ASM图



# 十字路口交通灯控制器ASM图



# 十字路口交通灯控制器ASM图



### 3、应用ASM图进行数字系统中控制器的设计

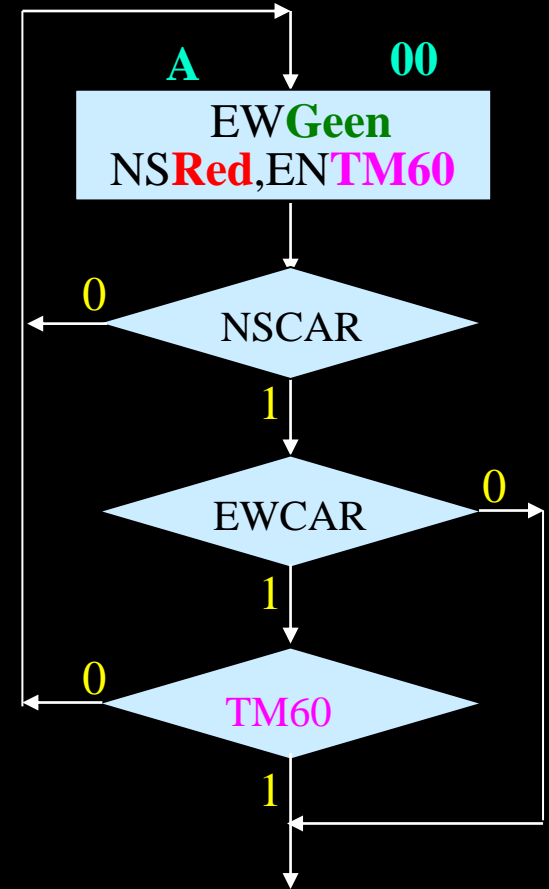
主要步骤：

- (1) 按设计要求写出问题说明。
- (2) 定义输入、输出信号并用助记符表示。
- (3) 将问题说明转换成详细逻辑流程图。
- (4) 将详细逻辑流程图转换成ASM图。
- (5) 从ASM图得到状态转换表。
- (6) 由状态转换表得到次态方程式，从ASM图列出输出函数表达式。
- (7) 按照次态方程式及输出函数表达式，画出控制器逻辑电路图。



## 4、状态转换表达式表

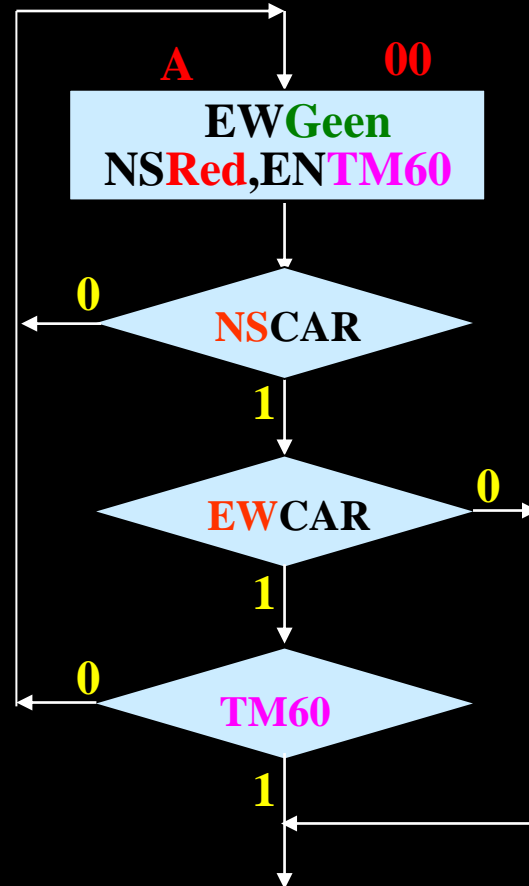
按照ASM图中的每一个ASM块中判别框和条件输出框内标注的**有效输入**，列出**状态转换表达式表**，如下：



现态	输入条件			状态转换表达式	次态
S(A)	NSCAR	EWCAR	TM60		$S^{n+1}$
$\overline{Q_1}\overline{Q_0}$					
$\overline{Q_1}Q_0$					
$Q_1\overline{Q_0}$					
$Q_1Q_0$					

## 4、状态转换表达式表

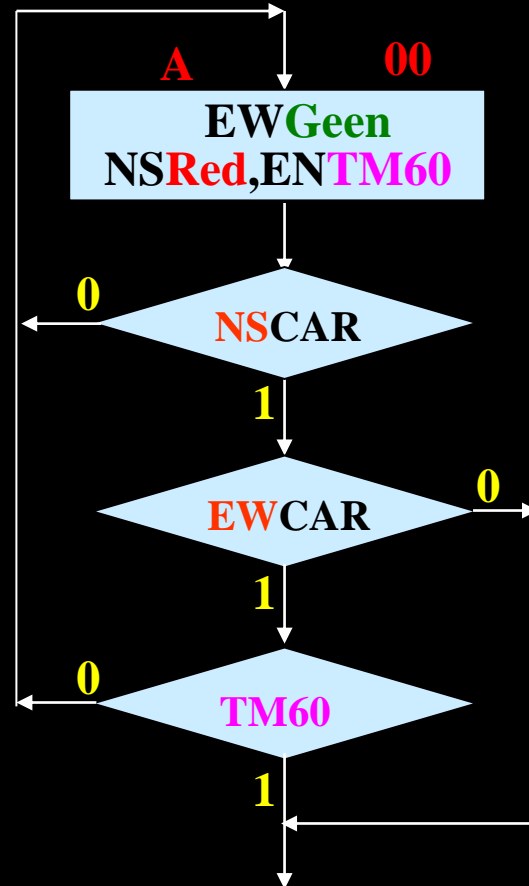
按照ASM图中的每一个ASM块中判别框和条件输出框内标注的有效输入，列出状态转换表达式表，如下：



现态	输入条件			状态转换表达式	次态
S(A)	NSCAR	EWCAR	TM60		S <sup>n+1</sup>
$\overline{Q_1}\overline{Q_0}$	0				A
$\overline{Q_1}Q_0$					
$Q_1\overline{Q_0}$					
$Q_1Q_0$					

## 4、状态转换表达式表

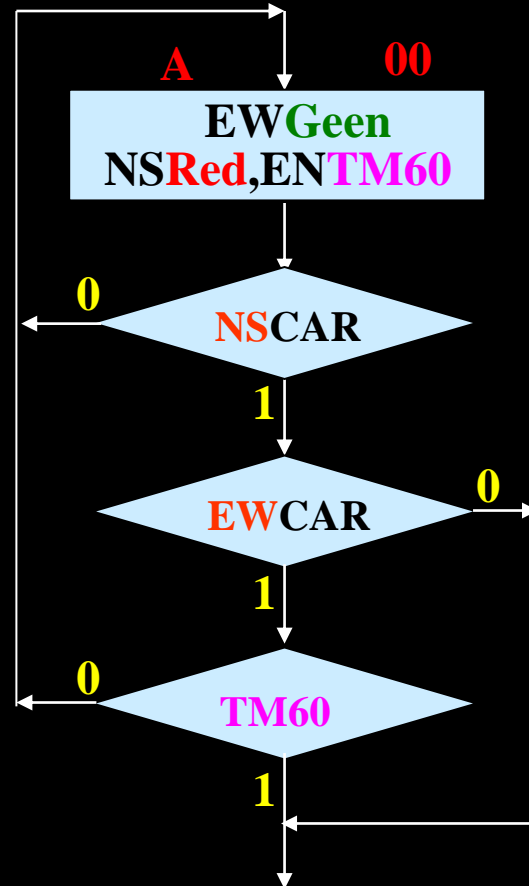
按照ASM图中的每一个ASM块中判别框和条件输出框内标注的有效输入，列出状态转换表达式表，如下：



现态	输入条件			状态转换表达式	次态
S(A)	NSCAR	EWCAR	TM60		$S^{n+1}$
$\overline{Q_1}\overline{Q_0}$	0	—	—		A
$\overline{Q_1}Q_0$					
$Q_1\overline{Q_0}$					
$Q_1Q_0$					

## 4、状态转换表达式表

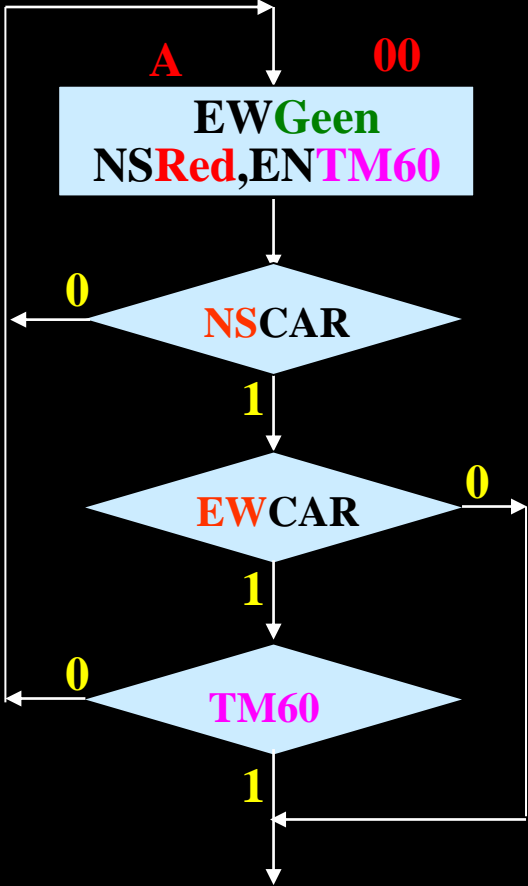
按照ASM图中的每一个ASM块中判别框和条件输出框内标注的有效输入，列出状态转换表达式表，如下：



现态	输入条件			状态转换表达式	次态
S(A)	NSCAR	EWCAR	TM60		$S^{n+1}$
$\overline{Q_1}\overline{Q_0}$	0	—	—	$\overline{NSCAR}$	A
$\overline{Q_1}Q_0$					
$Q_1\overline{Q_0}$					
$Q_1Q_0$					

# 4、状态转换表达式表

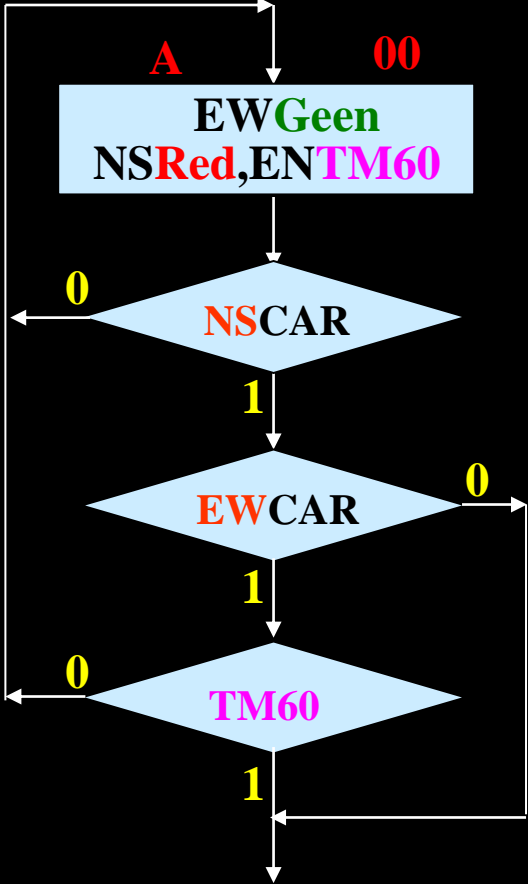
按照ASM图中的每一个ASM块中判别框和条件输出框内标注的有效输入，列出状态转换表达式表，如下：



现态	输入条件			状态转换表达式	次态
S(A)	NSCAR	EWCAR	TM60		$S^{n+1}$
$\overline{Q_1}\overline{Q_0}$	0	—	—	$\overline{NSCAR}$	A
$\overline{Q_1}Q_0$	1	0			B
$Q_1\overline{Q_0}$					
$Q_1Q_0$					

# 4、状态转换表达式表

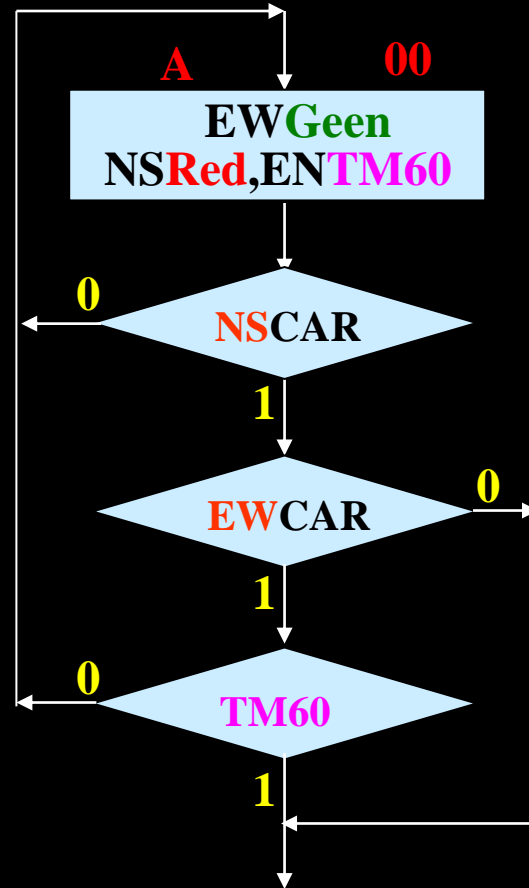
按照ASM图中的每一个ASM块中判别框和条件输出框内标注的有效输入，列出状态转换表达式表，如下：



现态	输入条件			状态转换表达式	次态
S(A)	NSCAR	EWCAR	TM60		$S^{n+1}$
$\overline{Q_1}\overline{Q_0}$	0	—	—	$\overline{NSCAR}$	A
$\overline{Q_1}Q_0$	1	0	—		B
$Q_1\overline{Q_0}$					
$Q_1Q_0$					

## 4、状态转换表达式表

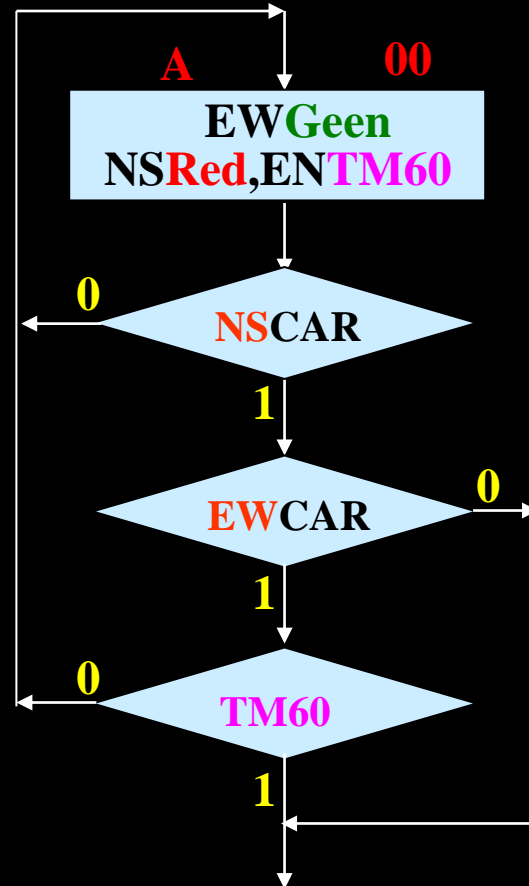
按照ASM图中的每一个ASM块中判别框和条件输出框内标注的有效输入，列出状态转换表达式表，如下：



现态	输入条件			状态转换表达式	次态
S(A)	NSCAR	EWCAR	TM60		$S^{n+1}$
$\overline{Q_1}\overline{Q_0}$	0	—	—	$\overline{NSCAR}$	A
$\overline{Q_1}Q_0$	1	0	—	$NSCAR \cdot \overline{EWCAR}$	B
$Q_1\overline{Q_0}$					
$Q_1Q_0$					

## 4、状态转换表达式表

按照ASM图中的每一个ASM块中判别框和条件输出框内标注的有效输入，列出状态转换表达式表，如下：

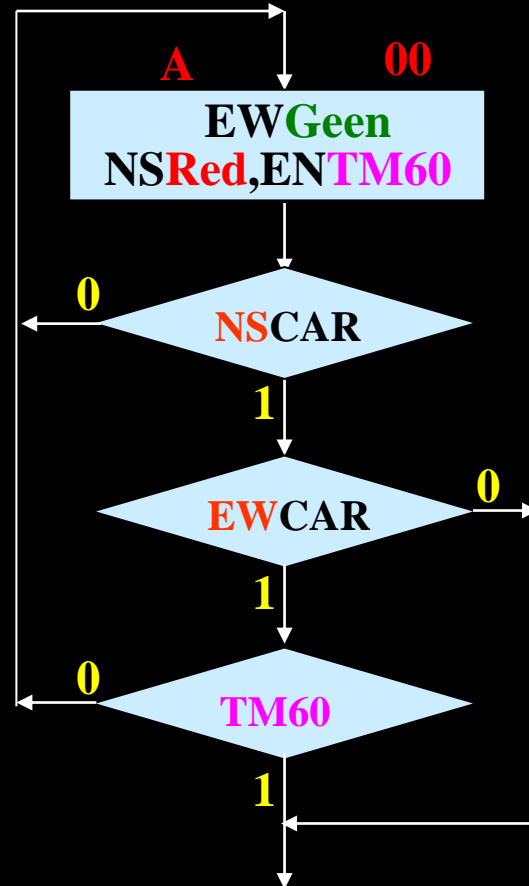


现态	输入条件			状态转换表达式	次态
S(A)	NSCAR	EWCAR	TM60		$S^{n+1}$
$\overline{Q_1}\overline{Q_0}$	0	—	—	$\overline{\text{NSCAR}}$	A
$\overline{Q_1}Q_0$	1	0	—	$\text{NSCAR} \cdot \overline{\text{EWCAR}}$	B
$Q_1\overline{Q_0}$	1	1	0		A
$Q_1Q_0$					



## 4、状态转换表达式表

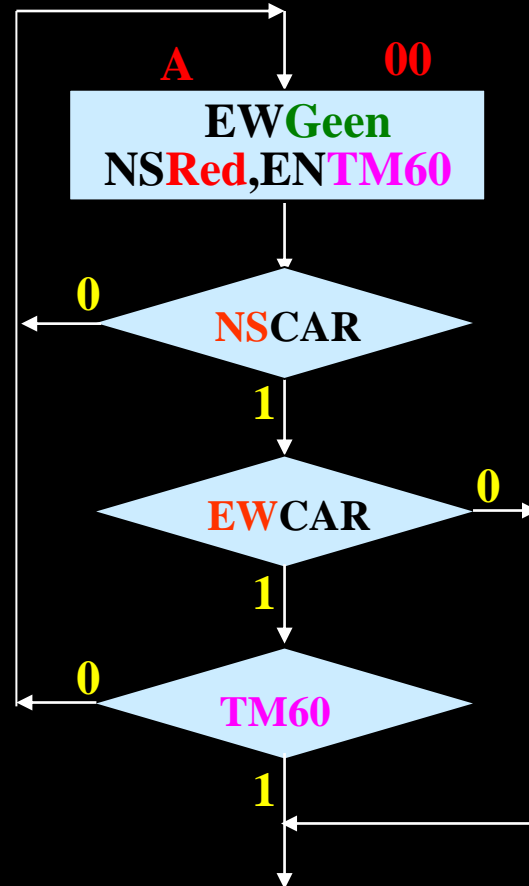
按照ASM图中的每一个ASM块中判别框和条件输出框内标注的有效输入，列出状态转换表达式表，如下：



现态	输入条件			状态转换表达式	次态
S(A)	NSCAR	EWCAR	TM60		$S^{n+1}$
$\overline{Q_1}\overline{Q_0}$	0	—	—	$\overline{NSCAR}$	A
$\overline{Q_1}Q_0$	1	0	—	$NSCAR \cdot \overline{EWCAR}$	B
$Q_1\overline{Q_0}$	1	1	0	$NSCAR \cdot EWCAR \cdot \overline{TM60}$	A
$Q_1Q_0$					

## 4、状态转换表达式表

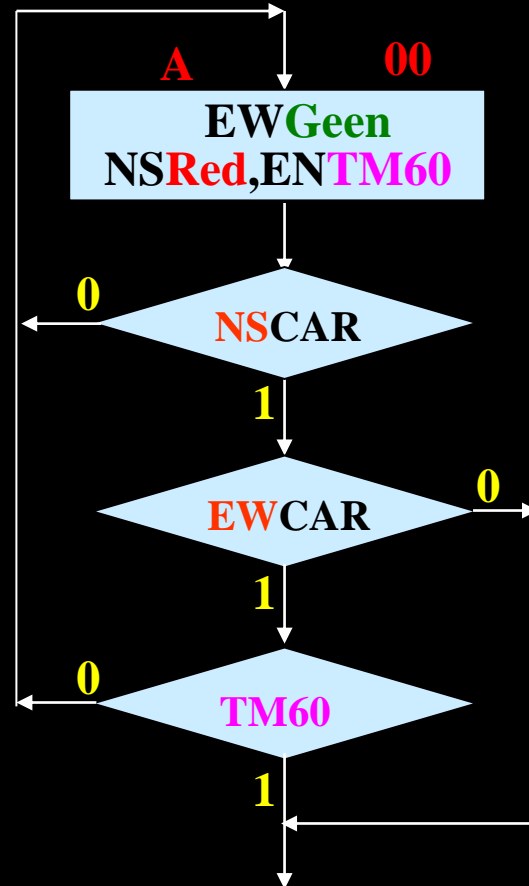
按照ASM图中的每一个ASM块中判别框和条件输出框内标注的有效输入，列出状态转换表达式表，如下：



现态	输入条件			状态转换表达式	次态
S(A)	NSCAR	EWCAR	TM60		S <sup>n+1</sup>
$\overline{Q_1}\overline{Q_0}$	0	—	—	$\overline{\text{NSCAR}}$	A
$\overline{Q_1}Q_0$	1	0	—	$\text{NSCAR} \cdot \overline{\text{EWCAR}}$	B
$Q_1\overline{Q_0}$	1	1	0	$\text{NSCAR} \cdot \text{EWCAR} \cdot \overline{\text{TM60}}$	A
$Q_1Q_0$	1	1	1		B

## 4、状态转换表达式表

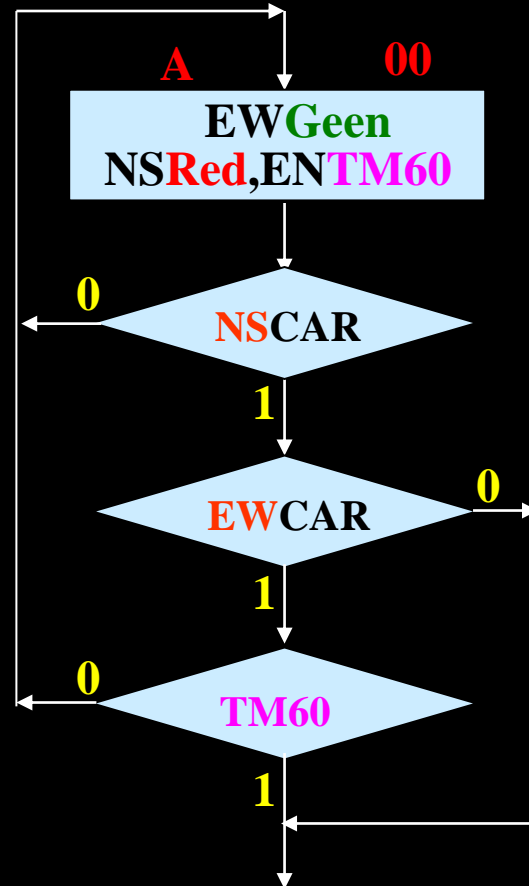
按照ASM图中的每一个ASM块中判别框和条件输出框内标注的有效输入，列出状态转换表达式表，如下：



现态	输入条件			状态转换表达式	次态
S(A)	NSCAR	EWCAR	TM60		S <sup>n+1</sup>
$\overline{Q_1}\overline{Q_0}$	0	—	—	$\overline{\text{NSCAR}}$	A
$\overline{Q_1}Q_0$	1	0	—	$\text{NSCAR} \cdot \overline{\text{EWCAR}}$	B
$Q_1\overline{Q_0}$	1	1	0	$\text{NSCAR} \cdot \text{EWCAR} \cdot \overline{\text{TM60}}$	A
$Q_1Q_0$	1	1	1	$\text{NSCAR} \cdot \text{EWCAR} \cdot \text{TM60}$	B

## 4、状态转换表达式表

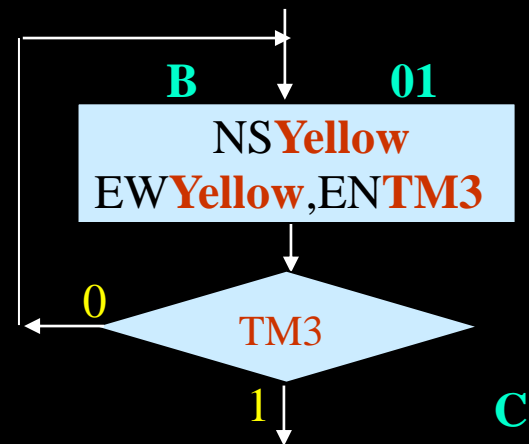
按照ASM图中的每一个ASM块中判别框和条件输出框内标注的有效输入，列出状态转换表达式表，如下：



现态	输入条件			状态转换表达式	次态
S(A)	NSCAR	EWCAR	TM60		S <sup>n+1</sup>
$\overline{Q_1}\overline{Q_0}$	0	—	—	$\overline{\text{NSCAR}}$	A
$\overline{Q_1}Q_0$	1	0	—	$\text{NSCAR} \cdot \overline{\text{EWCAR}}$	B
$Q_1\overline{Q_0}$	1	1	0	$\text{NSCAR} \cdot \text{EWCAR} \cdot \overline{\text{TM60}}$	A
$Q_1Q_0$	1	1	1	$\text{NSCAR} \cdot \text{EWCAR} \cdot \text{TM60}$	B

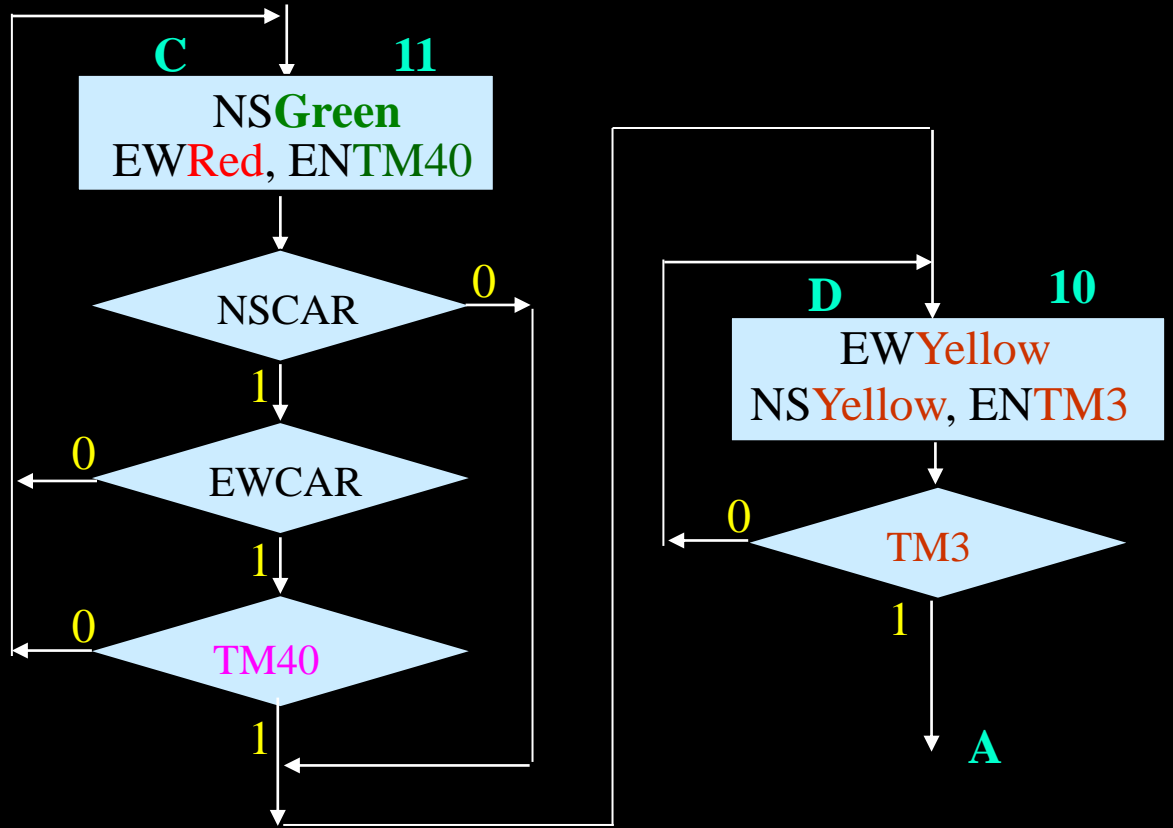
## 4、状态转换表达式表

按照ASM图中的每一个ASM块中判别框和条件输出框内标注的**有效输入**，列出**状态转换表达式表**，如下：



现态	输 入 条 件			状态转换表达式	次态
S(B)			TM 3		S <sup>n+1</sup>
$\overline{Q_1}Q_0$			0	$\overline{\text{TM 3}}$	B
$\overline{Q_1}Q_0$			1	TM 3	C

4、状态转换表达式表



现态	输 入 条 件				状态转换表达式	次态
S	NSCAR	EWCAR	TM40	TM3		S <sup>n+1</sup>
Q <sub>1</sub> Q <sub>0</sub>	0	—	—	—	$\overline{\text{NSCAR}}$	D
Q <sub>1</sub> Q <sub>0</sub>	1	0	—	—	$\text{NSCAR} \cdot \overline{\text{EWCAR}}$	C
Q <sub>1</sub> Q <sub>0</sub>	1	1	0	—	$\text{NSCAR} \cdot \text{EWCAR} \cdot \overline{\text{TM40}}$	C
Q <sub>1</sub> Q <sub>0</sub>	1	1	1	—	$\text{NSCAR} \cdot \text{EWCAR} \cdot \text{TM40}$	D
Q <sub>1</sub> $\overline{Q_0}$	—	—	—	0	$\overline{\text{TM3}}$	D
Q <sub>1</sub> $\overline{Q_0}$	—	—	—	1	TM3	A

## 5、状态转换表

由状态转换表达式表合并成状态转换表，如下：

现态	输 入 条 件			状态转换表达式	次态
S(A)	NSCAR	EWCAR	TM60		$S^{n+1}$
$\overline{Q_1}\overline{Q_0}$	0	—	—	$\overline{\text{NSCAR}}$	A
$\overline{Q_1}Q_0$	1	0	—	$\text{NSCAR} \cdot \overline{\text{EWCAR}}$	B
$Q_1\overline{Q_0}$	1	1	0	$\text{NSCAR} \cdot \text{EWCAR} \cdot \overline{\text{TM60}}$	A
$Q_1Q_0$	1	1	1	$\text{NSCAR} \cdot \text{EWCAR} \cdot \text{TM60}$	B



S	$Q_1$	$Q_0$	转换条件表达式	$S^{n+1}$	$Q_1^{n+1}$	$Q_0^{n+1}$
A	0	0	$\overline{\text{NSCAR}}$	A	0	0
A	0	0	$\text{NSCAR} \cdot \overline{\text{EWCAR}}$	B	0	1
A	0	0	$\text{NSCAR} \cdot \text{EWCAR} \cdot \overline{\text{TM60}}$	A	0	0
A	0	0	$\text{NSCAR} \cdot \text{EWCAR} \cdot \text{TM60}$	B	0	1

# 十字路口交通灯控制器状态转换表

S	Q <sub>1</sub>	Q <sub>0</sub>	转换条件表达式	S <sup>n+1</sup>	Q <sub>1</sub> <sup>n+1</sup>	Q <sub>0</sub> <sup>n+1</sup>
A	0	0	$\overline{\text{NSCAR}}$	A	0	0
A	0	0	$\text{NSCAR} \cdot \overline{\text{EWCAR}}$	B	0	1
A	0	0	$\text{NSCAR} \cdot \text{EWCAR} \cdot \overline{\text{TM60}}$	A	0	0
A	0	0	$\text{NSCAR} \cdot \text{EWCAR} \cdot \text{TM60}$	B	0	1
B	0	1	$\overline{\text{TM3}}$	B	0	1
B	0	1	$\text{TM3}$	C	1	1
C	1	1	$\text{NSCAR}$	D	1	0
C	1	1	$\text{NSCAR} \cdot \overline{\text{EWCAR}}$	C	1	1
C	1	1	$\text{NSCAR} \cdot \text{EWCAR} \cdot \overline{\text{TM40}}$	C	1	1
C	1	1	$\text{NSCAR} \cdot \text{EWCAR} \cdot \text{TM40}$	D	1	0
D	1	0	$\overline{\text{TM3}}$	D	1	0
D	1	0	$\text{TM3}$	A	0	0



## 十字路口交通灯控制器次态方程式

由于次态为  $S^{n+1} = S \cdot (\text{转换条件表达式})$ ，由前表可得到次态方程式并化简为：

$$\begin{aligned} Q_1^{n+1} &= \overline{Q_1} \cdot Q_0 \cdot \text{TM3} + Q_1 \cdot Q_0 \cdot \overline{\text{NSCAR}} + Q_1 \cdot Q_0 \cdot \text{NSCAR} \cdot \overline{\text{EWCAR}} \\ &\quad + Q_1 \cdot Q_0 \cdot \text{NSCAR} \cdot \text{EWCAR} \cdot \overline{\text{TM40}} \\ &\quad + Q_1 \cdot Q_0 \cdot \text{NSCAR} \cdot \text{EWCAR} \cdot \text{TM40} + Q_1 \cdot \overline{Q_0} \cdot \overline{\text{TM3}} \\ &= \overline{Q_1} \cdot Q_0 \cdot \text{TM3} + Q_1 \cdot Q_0 \cdot \overline{\text{NSCAR}} + Q_1 \cdot Q_0 \cdot \text{NSCAR} \cdot \overline{\text{EWCAR}} \\ &\quad + Q_1 \cdot Q_0 \cdot \text{NSCAR} \cdot \text{EWCAR} + Q_1 \cdot \overline{Q_0} \cdot \overline{\text{TM3}} \\ &= \overline{Q_1} \cdot Q_0 \cdot \text{TM3} + Q_1 \cdot Q_0 \cdot \overline{\text{NSCAR}} + Q_1 \cdot Q_0 \cdot \text{NSCAR} + Q_1 \cdot \overline{Q_0} \cdot \overline{\text{TM3}} \\ &= \overline{Q_1} \cdot Q_0 \cdot \text{TM3} + Q_1 \cdot Q_0 + Q_1 \cdot \overline{Q_0} \cdot \overline{\text{TM3}} \\ &= \overline{Q_1} \cdot Q_0 \cdot \text{TM3} + Q_1 \cdot Q_0 \cdot \text{TM3} + Q_1 \cdot Q_0 \cdot \overline{\text{TM3}} + Q_1 \cdot \overline{Q_0} \cdot \overline{\text{TM3}} \\ &= Q_0 \cdot \text{TM3} + Q_1 \cdot \overline{\text{TM3}} \end{aligned}$$

此方程也称为次态  $Q^{n+1}$  的转移方程，包含有转换条件表达式。

## 十字路口交通灯控制器次态方程式

由于次态为  $S^{n+1} = S \cdot (\text{转换条件表达式})$ ，由前表可得到次态方程式：

$$\begin{aligned} Q_1^{n+1} &= \overline{Q_1} \cdot Q_0 \cdot TM3 + Q_1 \cdot Q_0 \cdot \overline{NSCAR} \\ &\quad + Q_1 \cdot Q_0 \cdot NSCAR \cdot \overline{EWCAR} \\ &\quad + Q_1 \cdot Q_0 \cdot NSCAR \cdot EWCAR \cdot \overline{TM40} \\ &\quad + Q_1 \cdot Q_0 \cdot NSCAR \cdot EWCAR \cdot TM40 + Q_1 \cdot \overline{Q_0} \cdot \overline{TM3} \\ &= Q_0 \cdot TM3 + Q_1 \cdot \overline{TM3} \end{aligned}$$

$$\begin{aligned} Q_0^{n+1} &= \overline{Q_1} \cdot \overline{Q_0} \cdot \overline{NSCAR} \cdot \overline{EWCAR} + \overline{Q_1} \cdot \overline{Q_0} \cdot NSCAR \\ &\quad \cdot \overline{EWCAR} \cdot TM60 + \overline{Q_1} \cdot Q_0 \cdot \overline{TM3} + \overline{Q_1} \cdot Q_0 \cdot TM3 \\ &\quad + Q_1 \cdot Q_0 \cdot NSCAR \cdot \overline{EWCAR} \\ &\quad + Q_1 \cdot Q_0 \cdot NSCAR \cdot EWCAR \cdot \overline{TM40} \end{aligned}$$

选用**D触发器**构成的控制系统，次态  $Q^{n+1}=D$   
则可直接写出控制函数：

$$\begin{aligned} D_1 &= \overline{Q_1} \cdot Q_0 \cdot TM3 + Q_1 \cdot Q_0 \cdot \overline{NSCAR} \\ &\quad + Q_1 \cdot Q_0 \cdot \overline{NSCAR} \cdot \overline{EWCAR} \\ &\quad + Q_1 \cdot Q_0 \cdot \overline{NSCAR} \cdot \overline{EWCAR} \cdot \overline{TM40} \\ &\quad + Q_1 \cdot Q_0 \cdot \overline{NSCAR} \cdot \overline{EWCAR} \cdot TM40 + Q_1 \cdot \overline{Q_0} \cdot \overline{TM3} \\ &= Q_0 \cdot TM3 + Q_1 \cdot \overline{TM3} \end{aligned}$$

$$\begin{aligned} D_0 &= \overline{Q_1} \cdot \overline{Q_0} \cdot \overline{NSCAR} \cdot \overline{EWCAR} + \overline{Q_1} \cdot \overline{Q_0} \cdot \overline{NSCAR} \\ &\quad \cdot \overline{EWCAR} \cdot TM60 + \overline{Q_1} \cdot Q_0 \cdot \overline{TM3} + \overline{Q_1} \cdot Q_0 \cdot TM3 \\ &\quad + Q_1 \cdot Q_0 \cdot \overline{NSCAR} \cdot \overline{EWCAR} \\ &\quad + Q_1 \cdot Q_0 \cdot \overline{NSCAR} \cdot \overline{EWCAR} \cdot \overline{TM40} \end{aligned}$$

# 由ASM图可直接写出输出函数表达式

$$EW_{\text{Green}} = \overline{Q_1} \cdot \overline{Q_0}$$

$$NS_{\text{Red}} = \overline{Q_1} \cdot \overline{Q_0}$$

$$NS_{\text{Yellow}} = \overline{Q_1} \cdot Q_0$$

$$EW_{\text{Yellow}} = \overline{Q_1} \cdot Q_0$$

$$NS_{\text{Green}} =$$

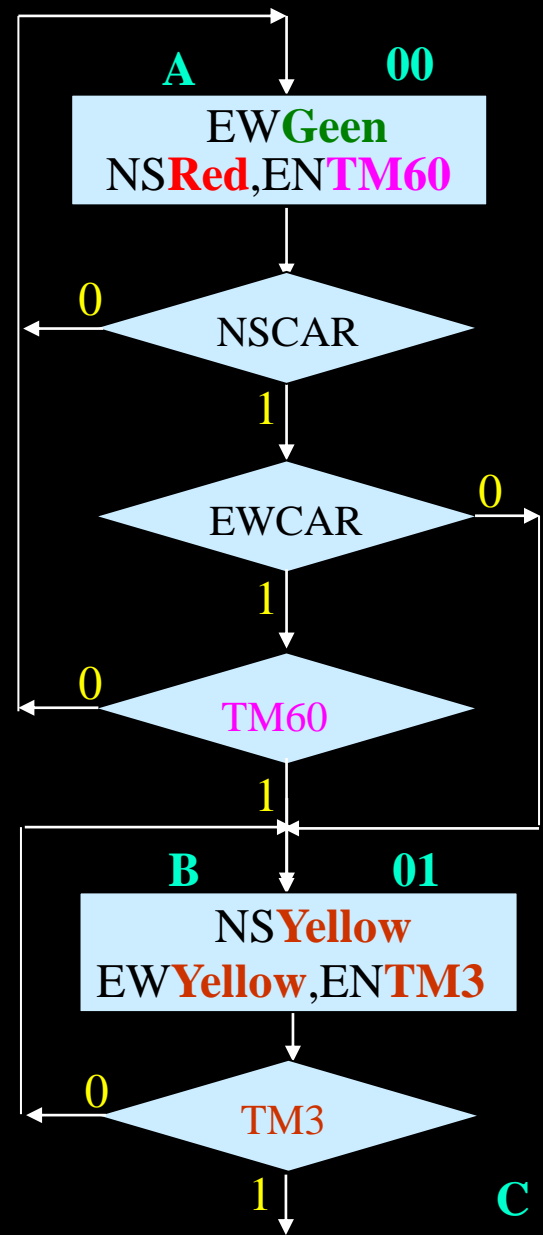
$$EW_{\text{Red}} =$$

$$ENTM60 = \overline{Q_1} \cdot \overline{Q_0}$$

$$ENTM40 =$$

$$ENTM3 = \overline{Q_1} \cdot Q_0$$

Moore型电路



## 由ASM图可直接写出输出函数表达式

$$EW_{\text{Green}} = \overline{Q_1} \cdot \overline{Q_0}$$

$$NS_{\text{Red}} = \overline{Q_1} \cdot \overline{Q_0}$$

$$NS_{\text{Yellow}} = \overline{Q_1} \cdot Q_0 + Q_1 \cdot \overline{Q_0} = Q_1 \oplus Q_0$$

$$EW_{\text{Yellow}} = \overline{Q_1} \cdot Q_0 + Q_1 \cdot \overline{Q_0} = Q_1 \oplus Q_0$$

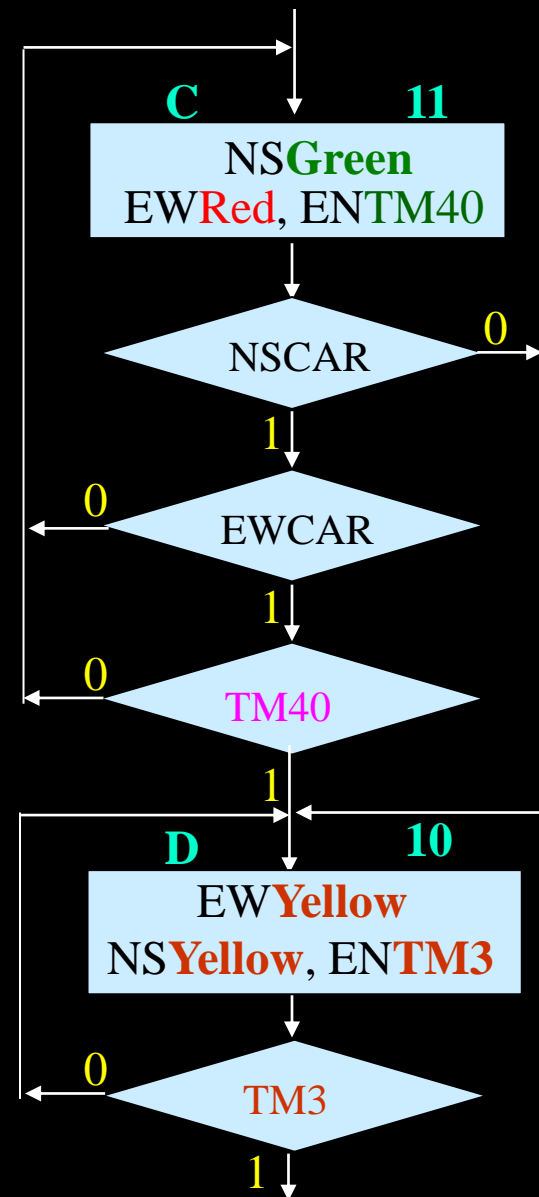
$$NS_{\text{Green}} = Q_1 \cdot Q_0$$

$$EW_{\text{Red}} = Q_1 \cdot Q_0$$

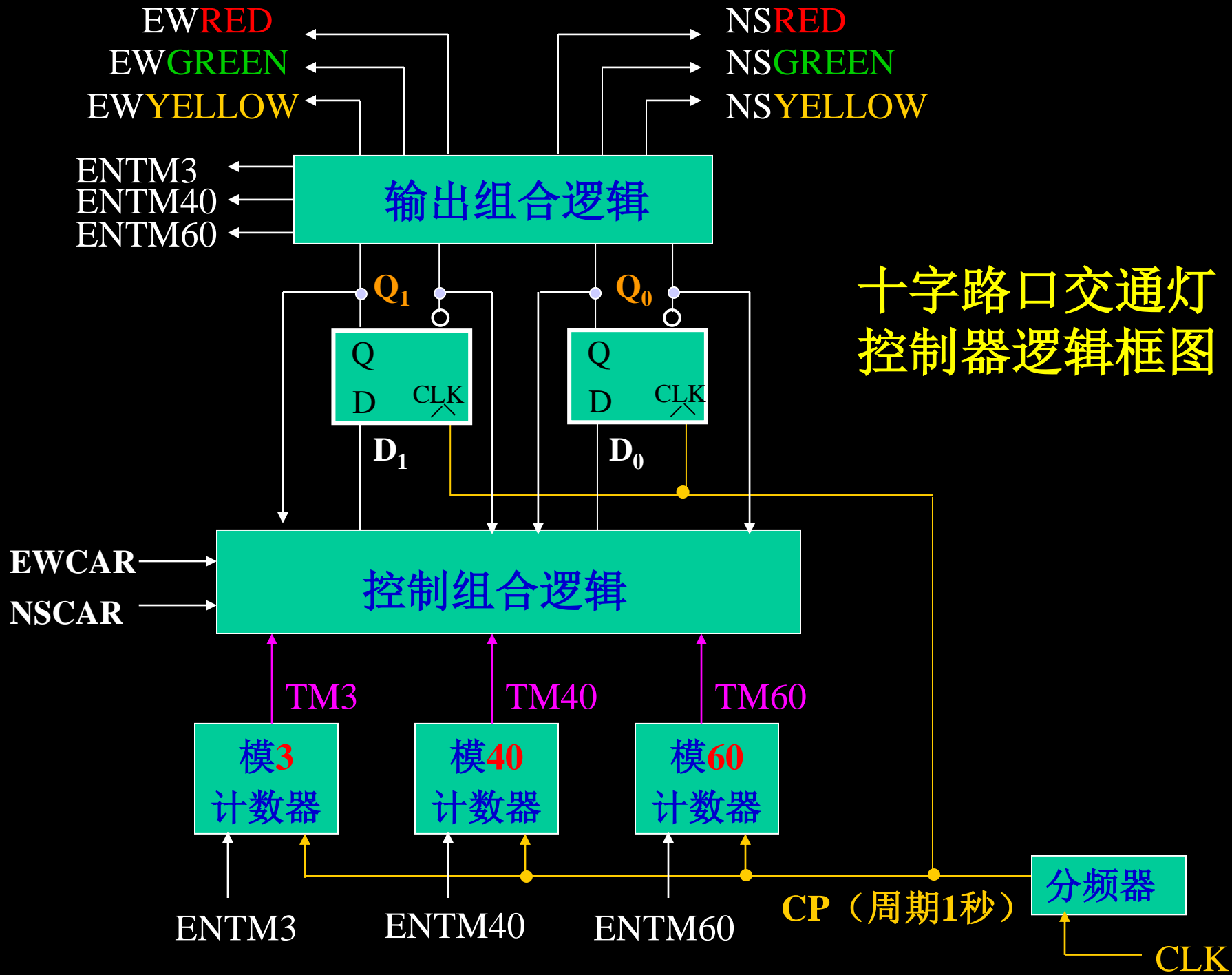
$$ENTM_{60} = \overline{Q_1} \cdot \overline{Q_0}$$

$$ENTM_{40} = Q_1 \cdot Q_0$$

$$ENTM_3 = \overline{Q_1} \cdot Q_0 + Q_1 \cdot \overline{Q_0} = Q_1 \oplus Q_0$$



可选用**D**触发器或**JK**触发器及门电路构成控制系统。



## ASM图的特点

ASM图为时序电路系统提供了形式化描述方法。

在ASM图中，每个状态框仅有一个出口（分支由判断框提供），保证了无二义性，且自动满足**闭合性和完整性**。

由于每个ASM块只能描述系统的一个状态，对于计算机和类似的复杂数字系统来说，仅仅采用ASM图作为硬件的描述工具是不够的，因此，需要一种功能更强的形式化工具来**描述和定义**数字系统中的操作和实现这些操作的硬件结构。

**寄存器传送语言（RTL）**就是这种形式语言之一。