

## § 6.1 数字系统的基本概念

### 1. 数字系统

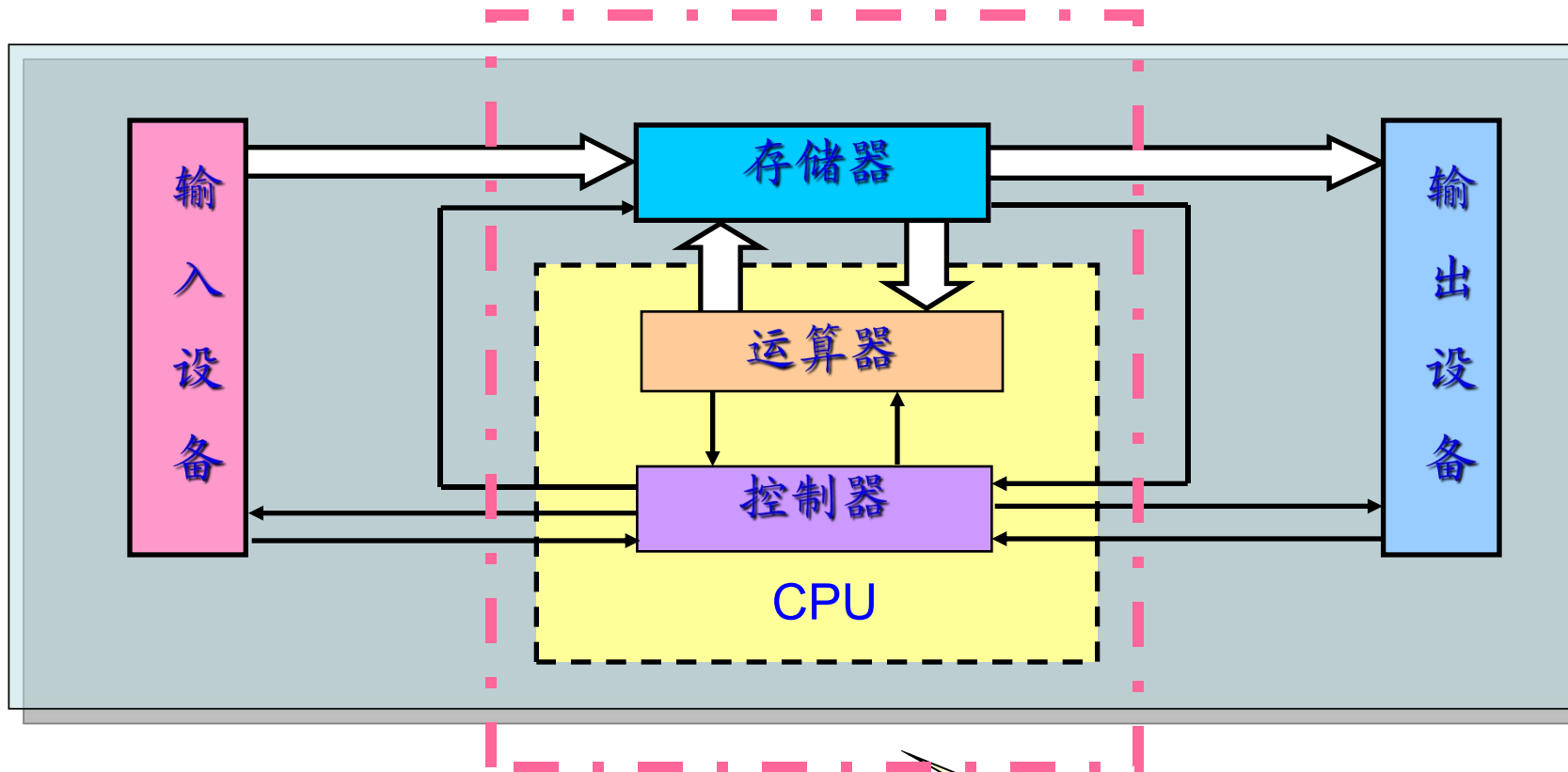
#### 1) 定义:

具有存储、传输、处理信息能力的逻辑系统集合。

由“数据通路+控制逻辑”构成，其工作具有周期性。

#### 2) 数字系统的基本构成

由五大部件组成。



五大部件通过总线和接口连接。

# 逻辑系统

# 数字系统

控制部件

执行部件

数据通路结构

运算器

存储器

算术逻辑单元 ALU

寄存器

累加寄存器

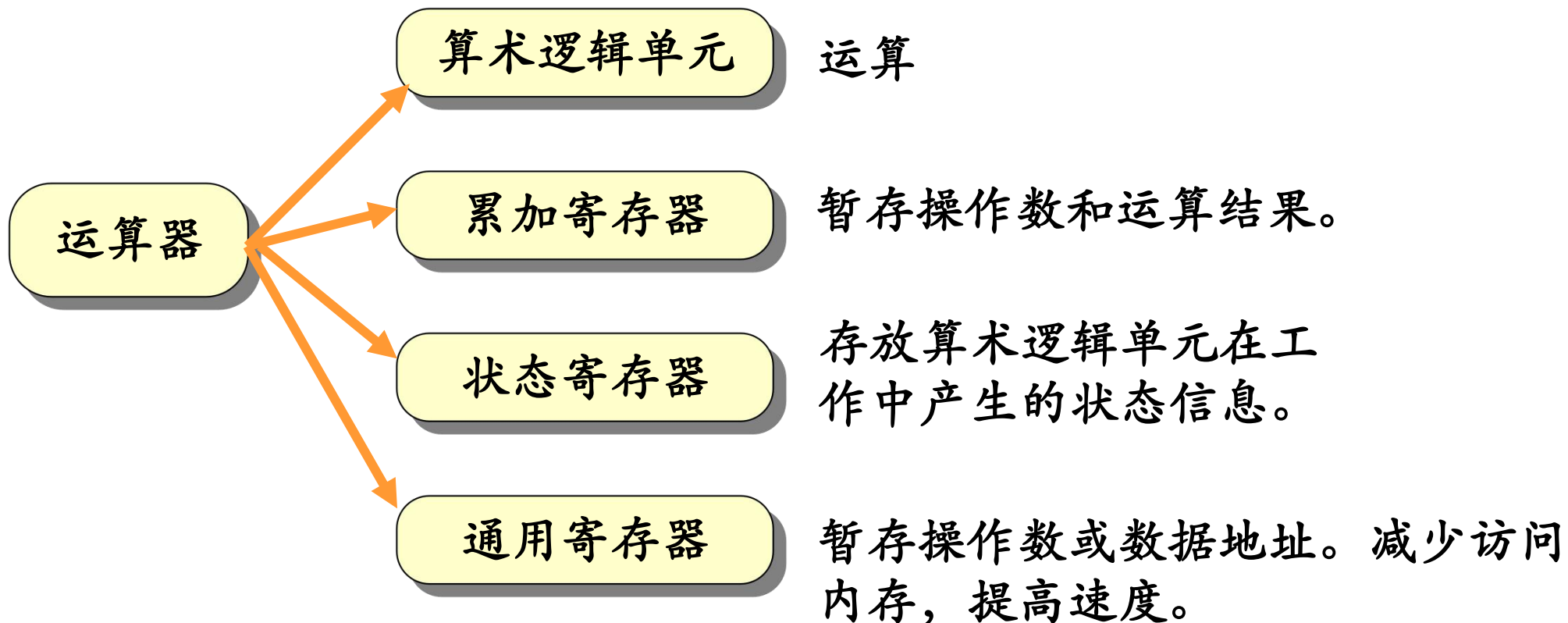
状态寄存器

通用寄存器

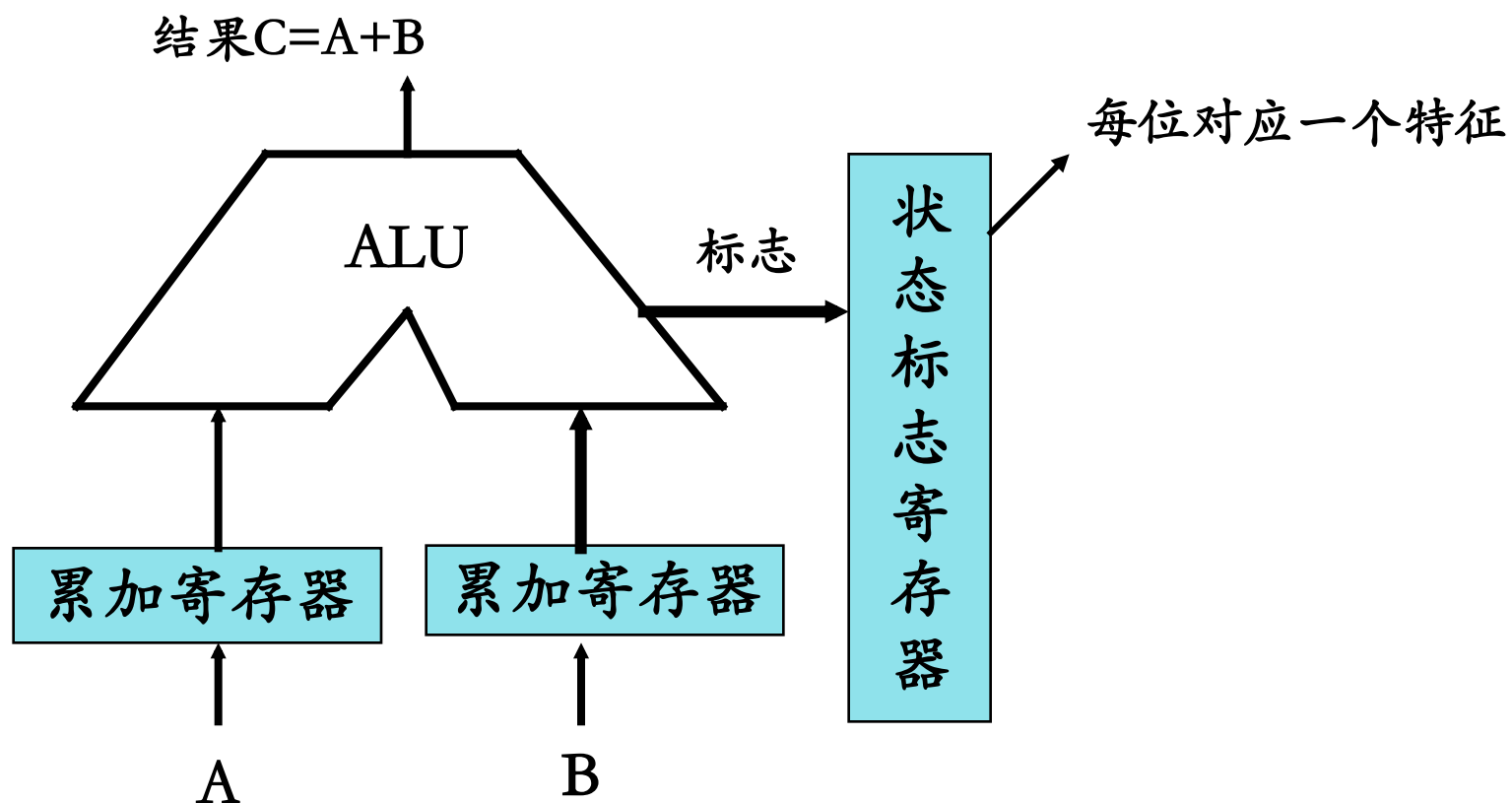
## 2. 基本子系统

### 1) 运算器

对信息进行加工处理（算术、逻辑、传输）。全部操作都是由控制器发出的控制信号来指挥的。

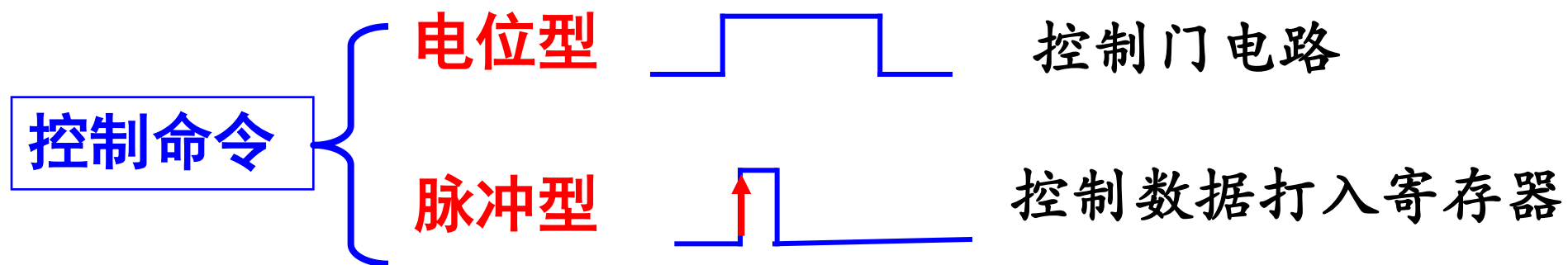


# 算术逻辑运算部件ALU (Arithmetical Logic Unit)



## 2) 控制器

发出控制命令，指挥、控制各部件协调工作的中心部件。



实现方法:

全硬件控制器、微程序控制器。

数字系统与逻辑功能部件的区别：有无控制器。

## § 6.2 数据通路

在各部件之间传送信息的公共通路，称为数据通路。

由总线连接的运算单元（组合电路）和存储单元构成。

数据通路：数据信号、控制信号。

由控制器控制数据流通过BUS总线在各子系统之间流动。

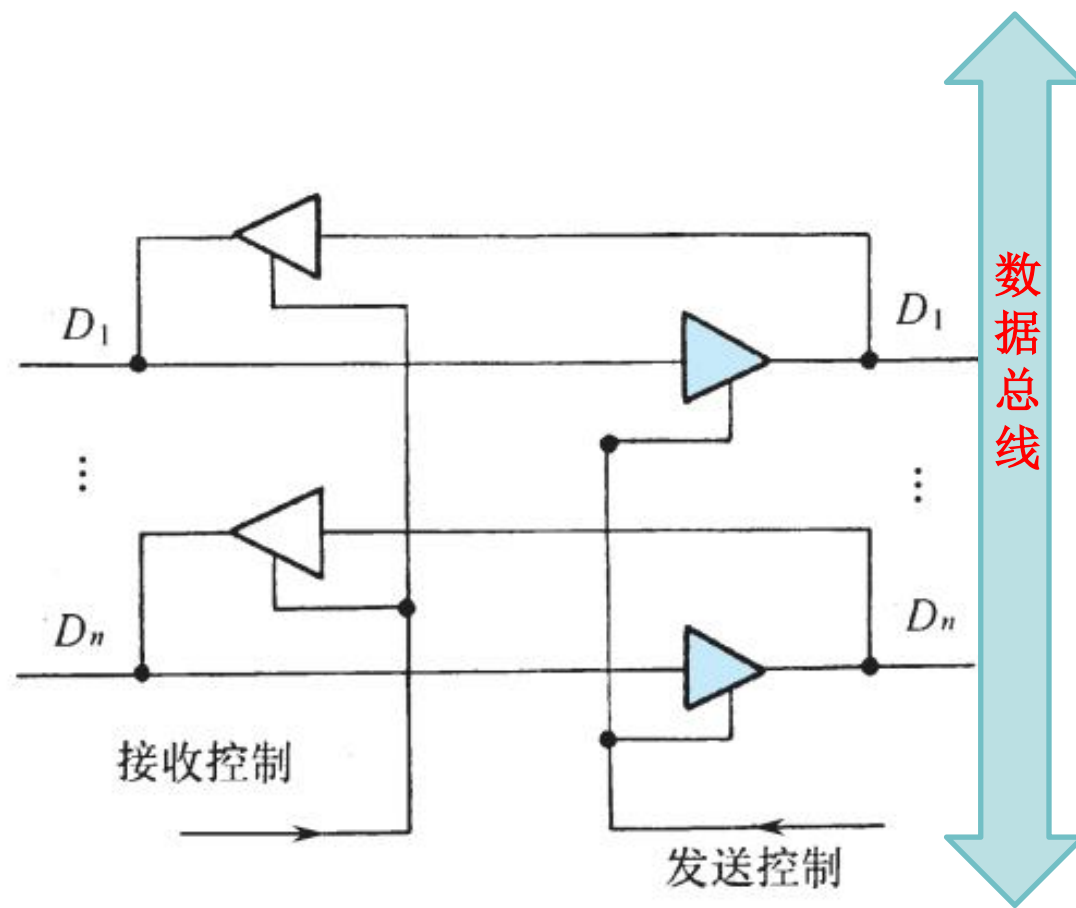
# 1. 总线

用来分时地发送与接收各部件信息的公共传送线路。

特点：多源、分时

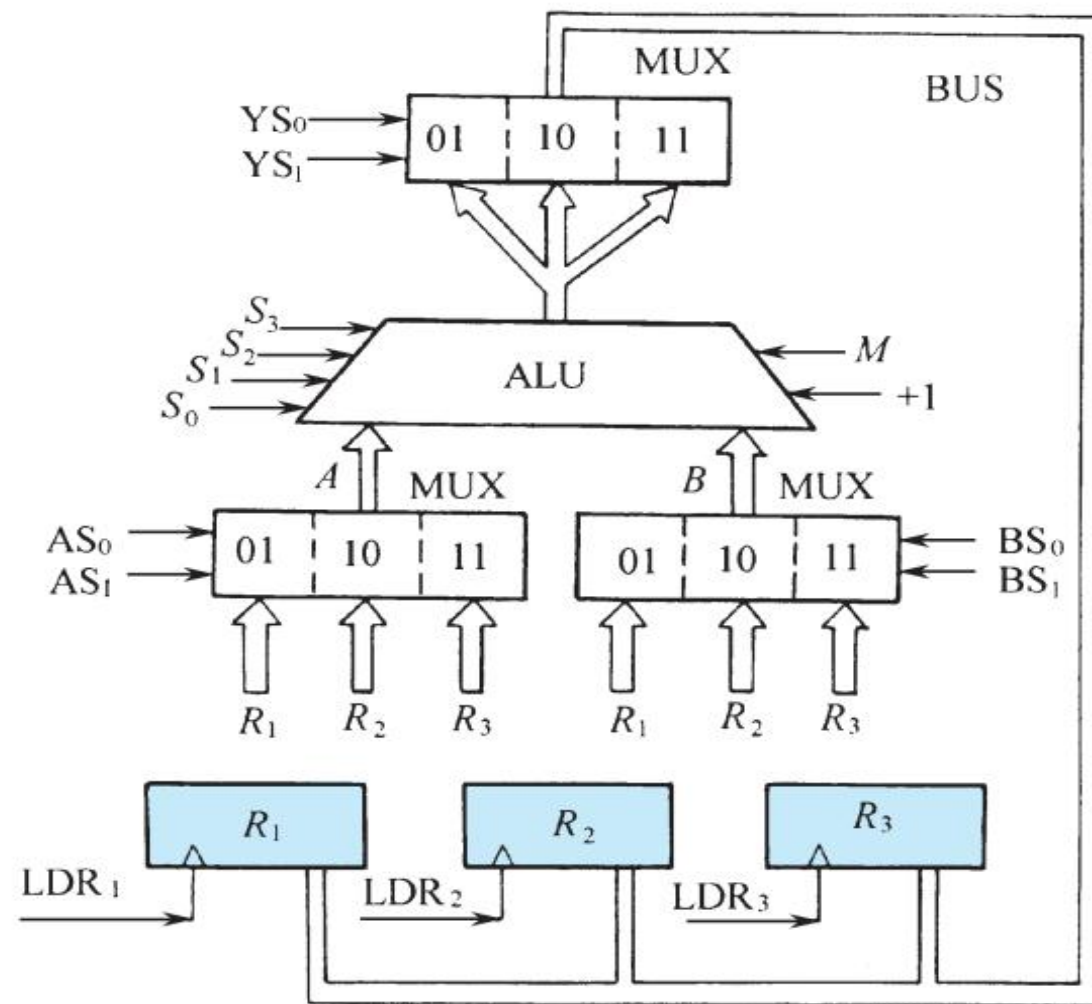
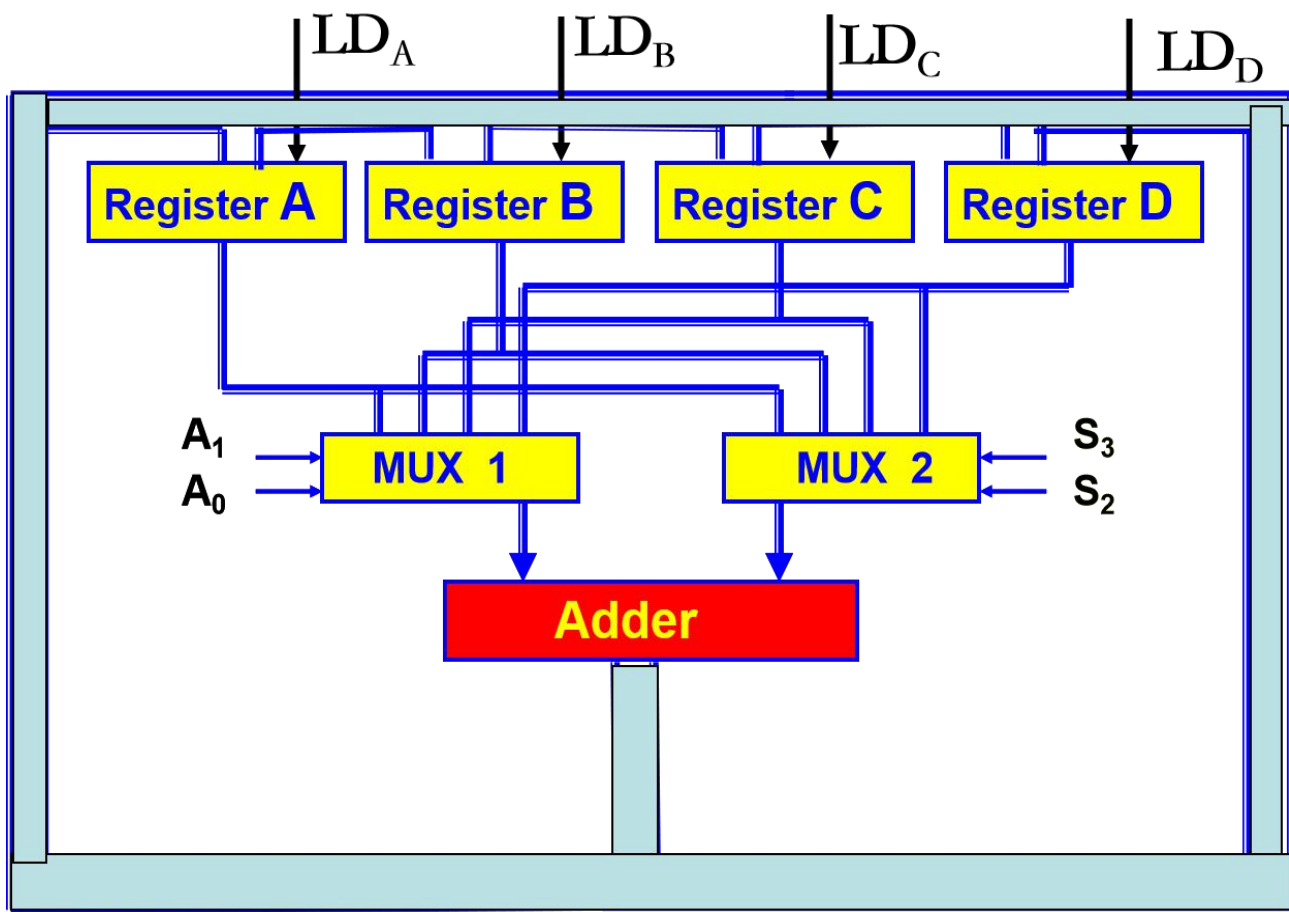
## • 总线的逻辑结构

总线 { 单向总线：选择器方式  
双向总线：三态门方式

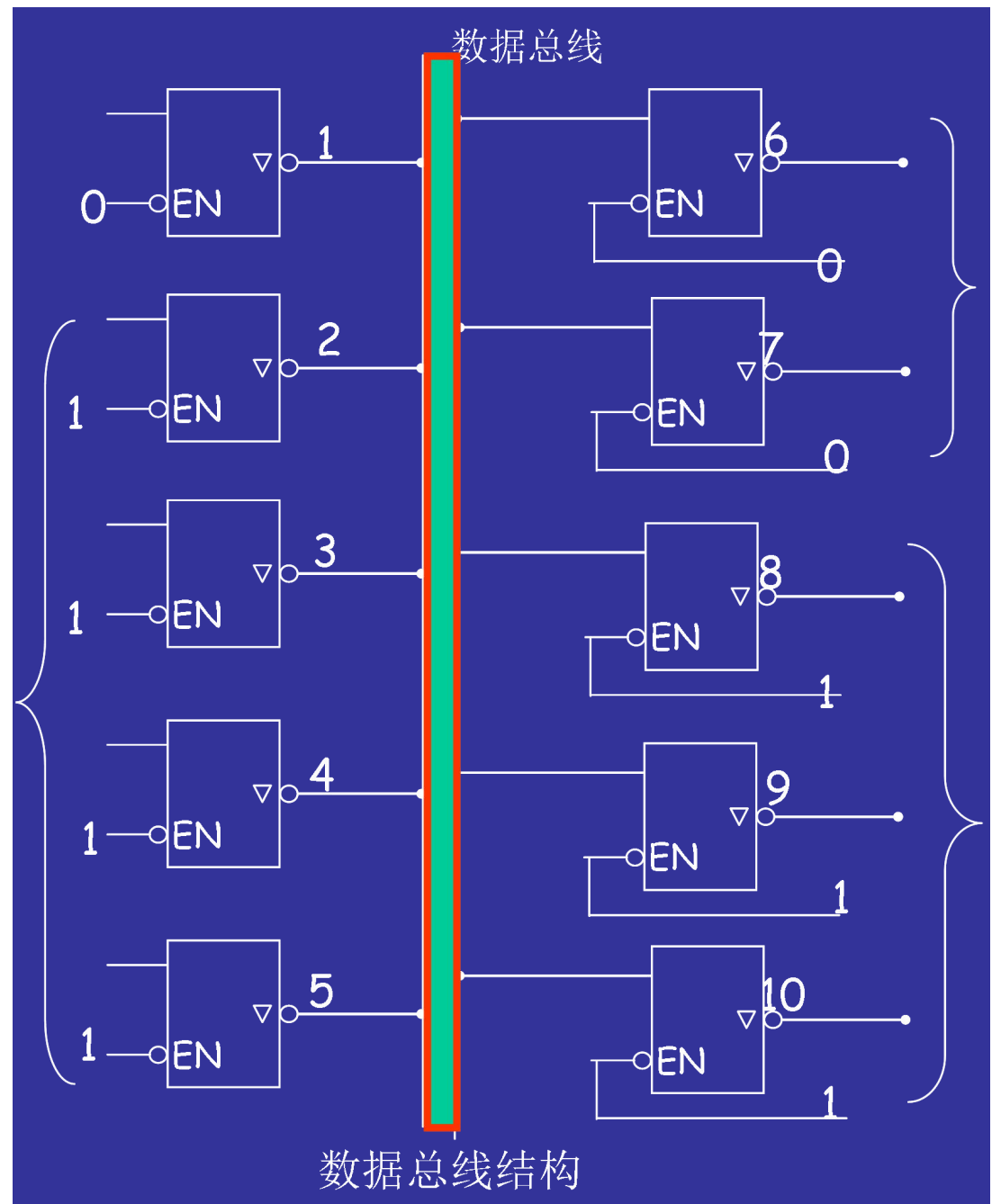
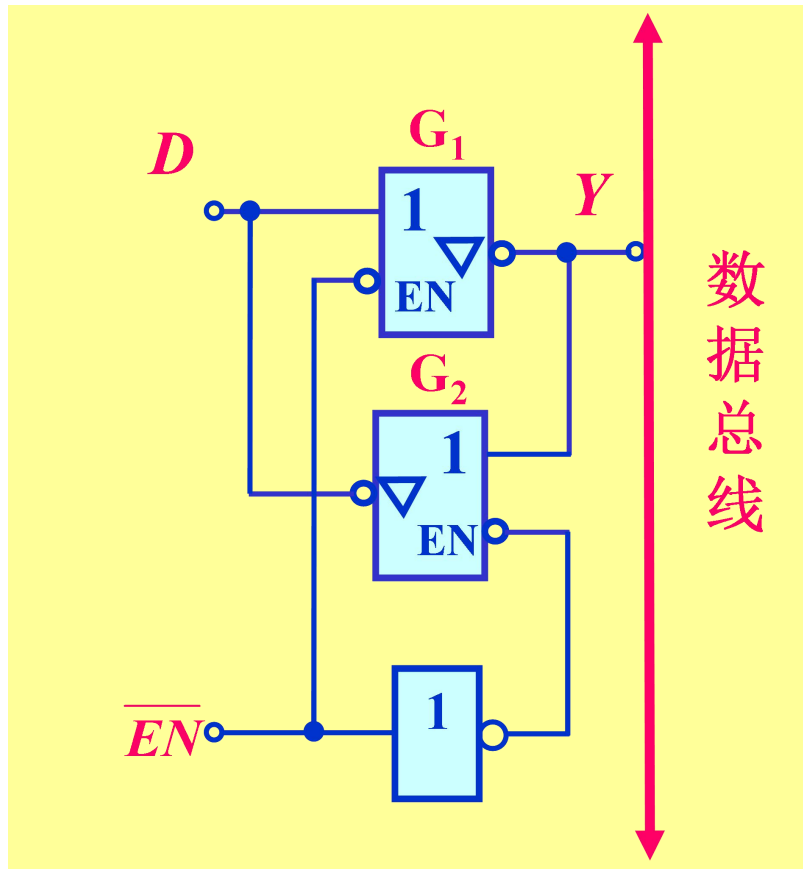




• 选择器方式:



• 三态门方式:



## § 6.3 数字系统的设计方法

### 一. 设计步骤

自顶向下方法 (from top to down) :

基本方法: 是将规模较大的系统从逻辑上划分为控制电路 + 受控电路。

步骤:

#### 1. 划分子系统、模块

数据处理器、控制器;

2. 数据通路（不唯一）

3. 用数学语言(工具)，描述控制器的控制过程

- ASM图 (Arithmetic State Machine Flowchart ) ;
- MDS图 (Mnemonic Documented State ) 。

4. 设计控制器

硬布线控制器、微程序控制器。

5. 设计数据处理器

## 二. 算法状态机

### 1. ASM 流程图

**ASM图：**用状态框、判断框、条件输出框和文字来描述数字系统控制器，在不同时间内完成的操作。

**特点：**

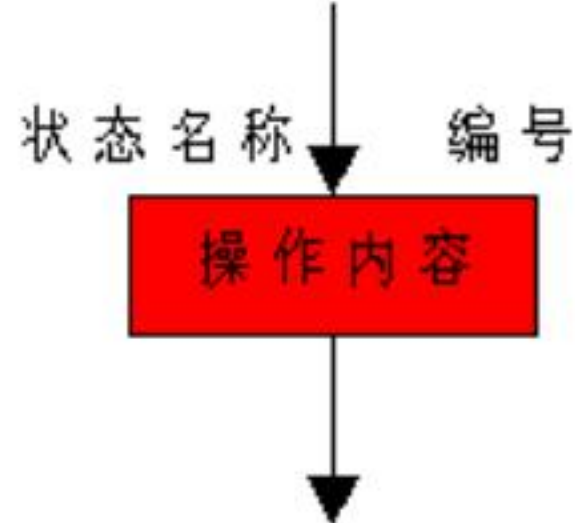
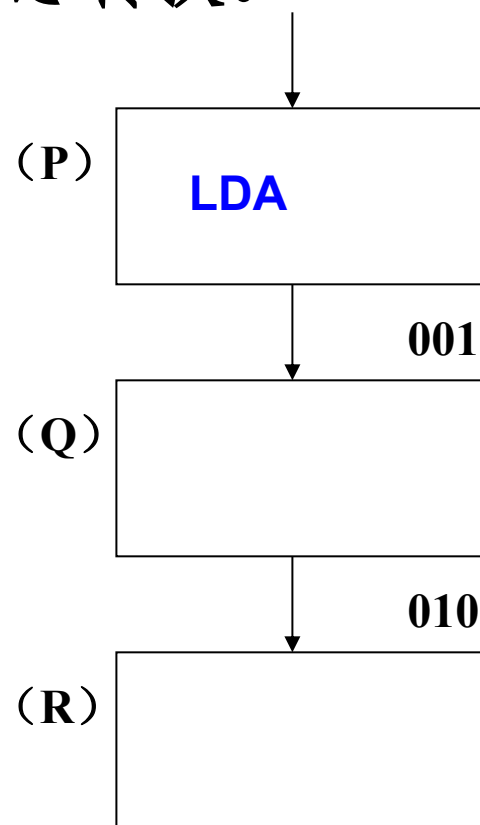
- ① 描述控制器的控制状态及其转换关系；
- ② 可精确地表示状态转换的时间关系。

# 1) 状态框

代表系统一个状态的矩形框。

在规定数量脉冲作用下实现状态转换。

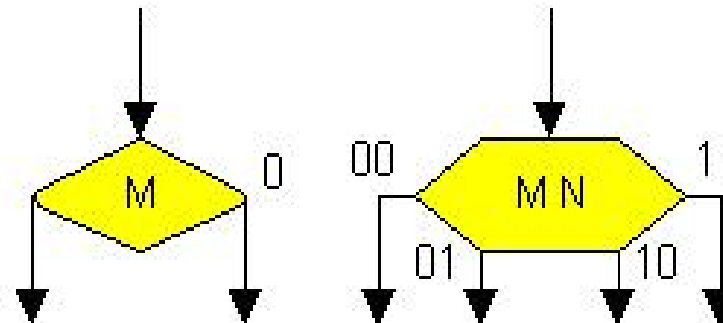
框内：控制命令



## 2) 条件判断框

对控制器输入信号的判断。

框内：检测条件



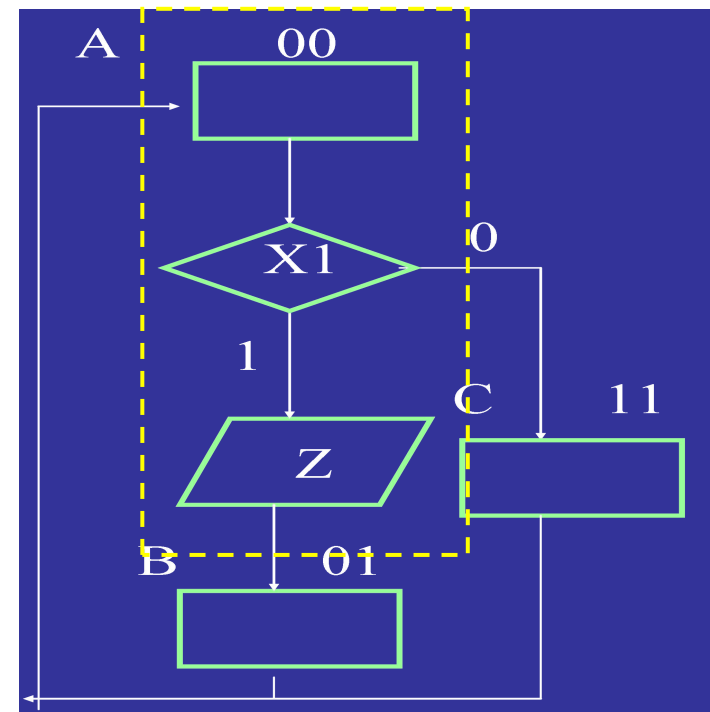
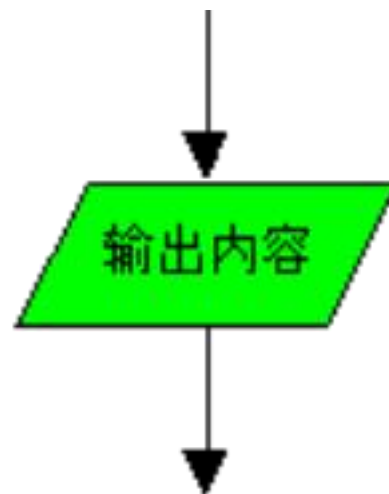
## 3) 条件输出框

在给定的状态下，判断条件满足时才发生的输出。

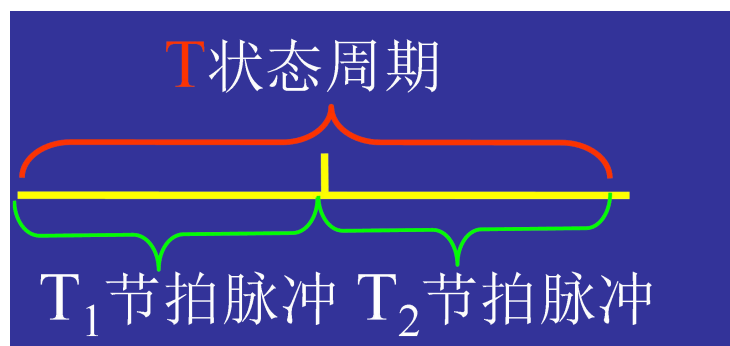
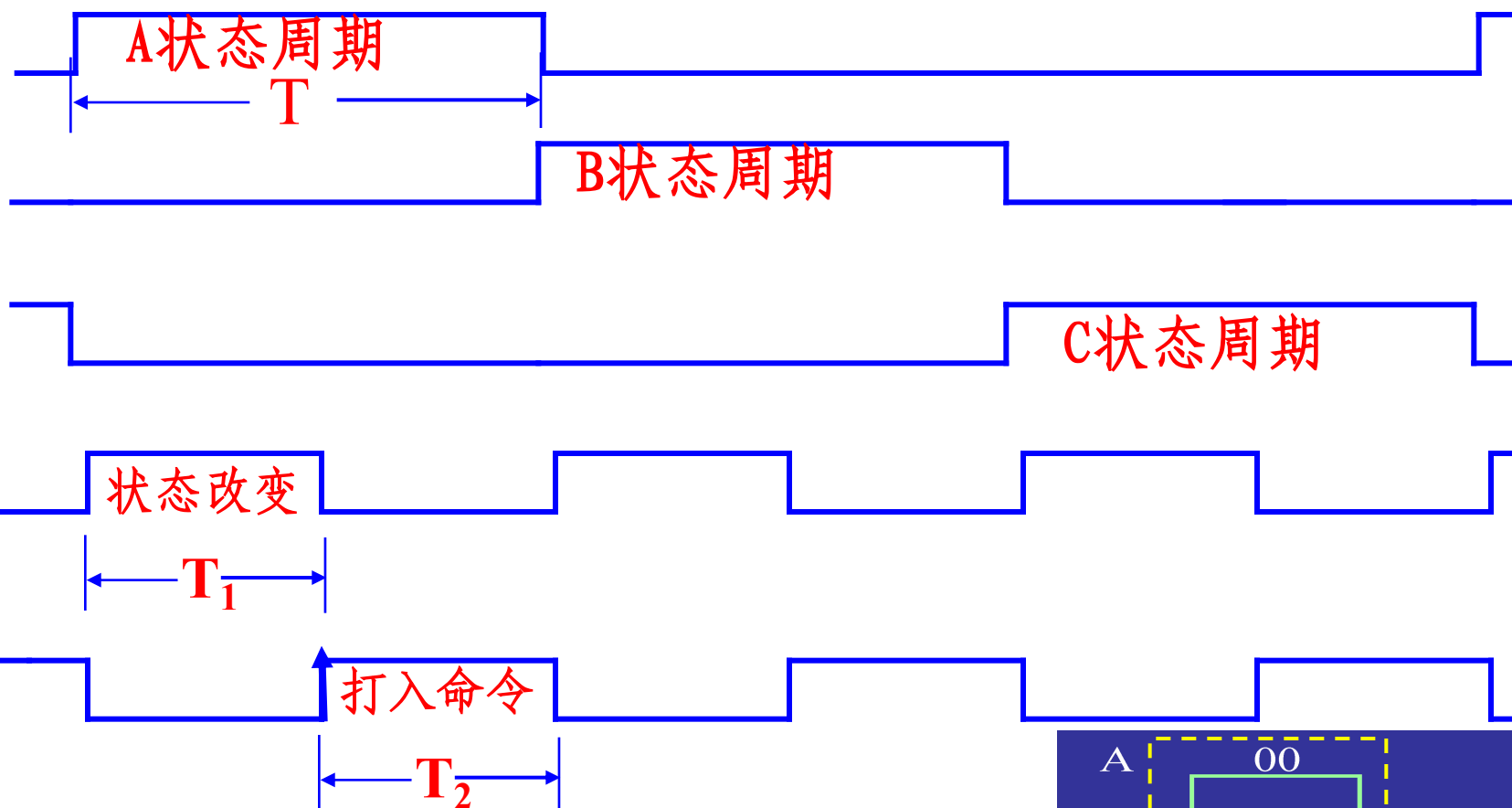
框内：控制命令

入口：判断框的输出

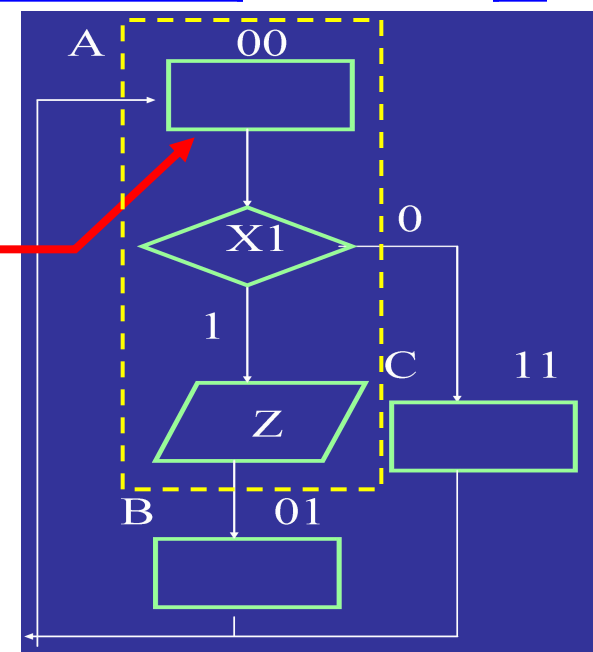
出口：指向状态框



和状态框相连的判断框、条件输出框都属于该状态框，它们的操作是在同一状态周期下完成的。



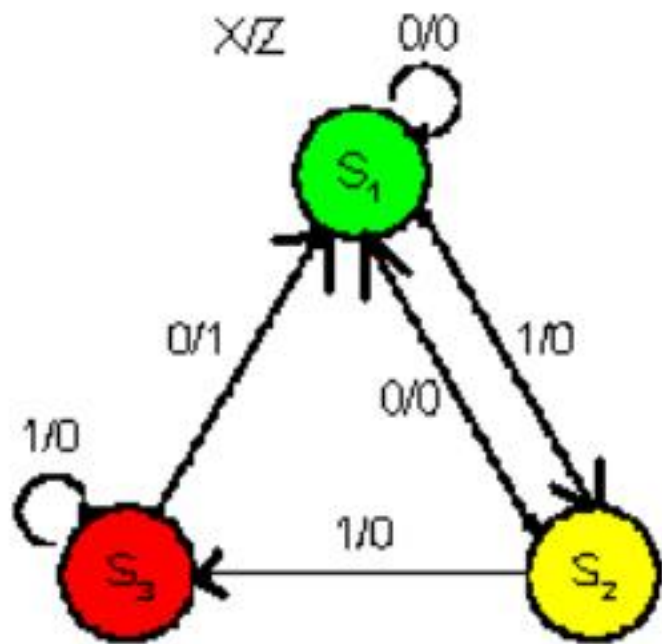
状态单元



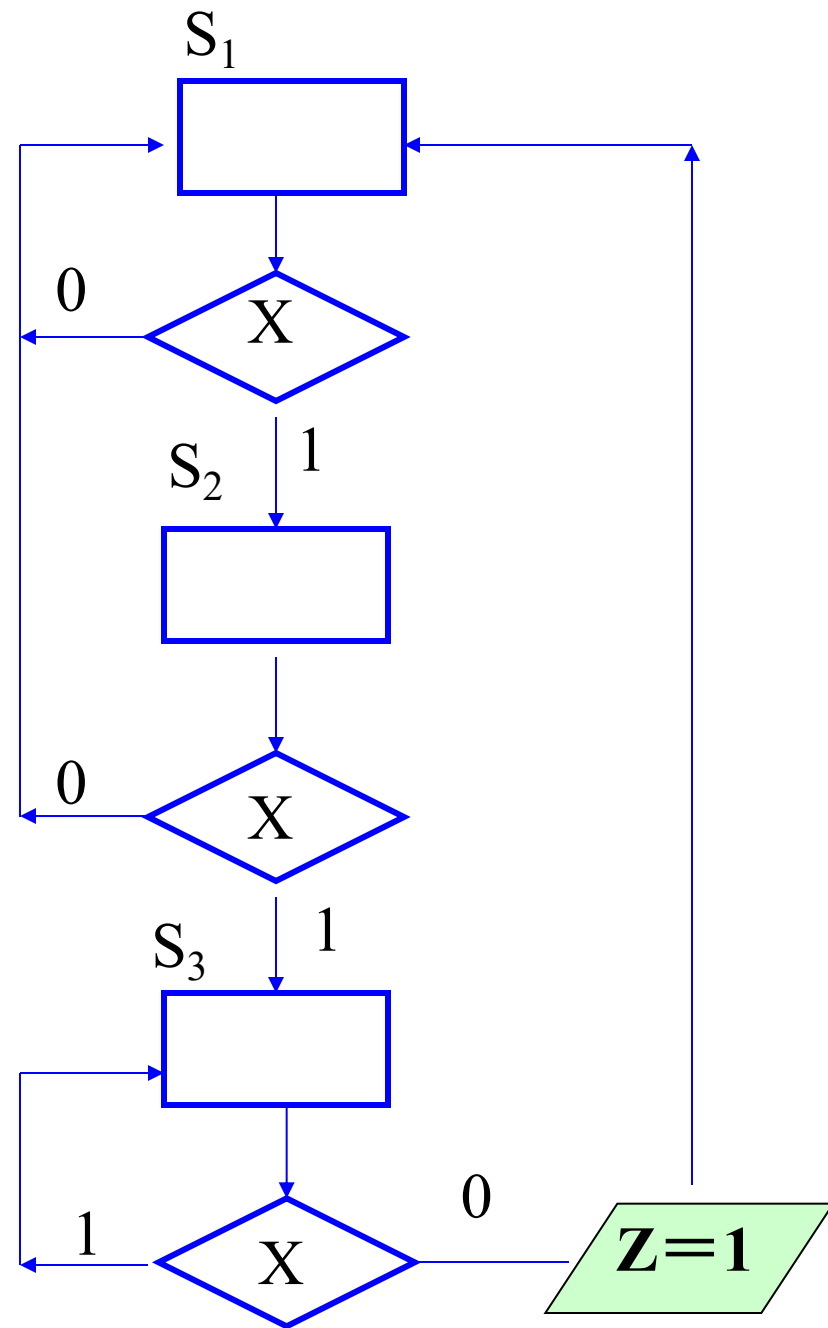
状态周期：每一个状态所经历的时间。



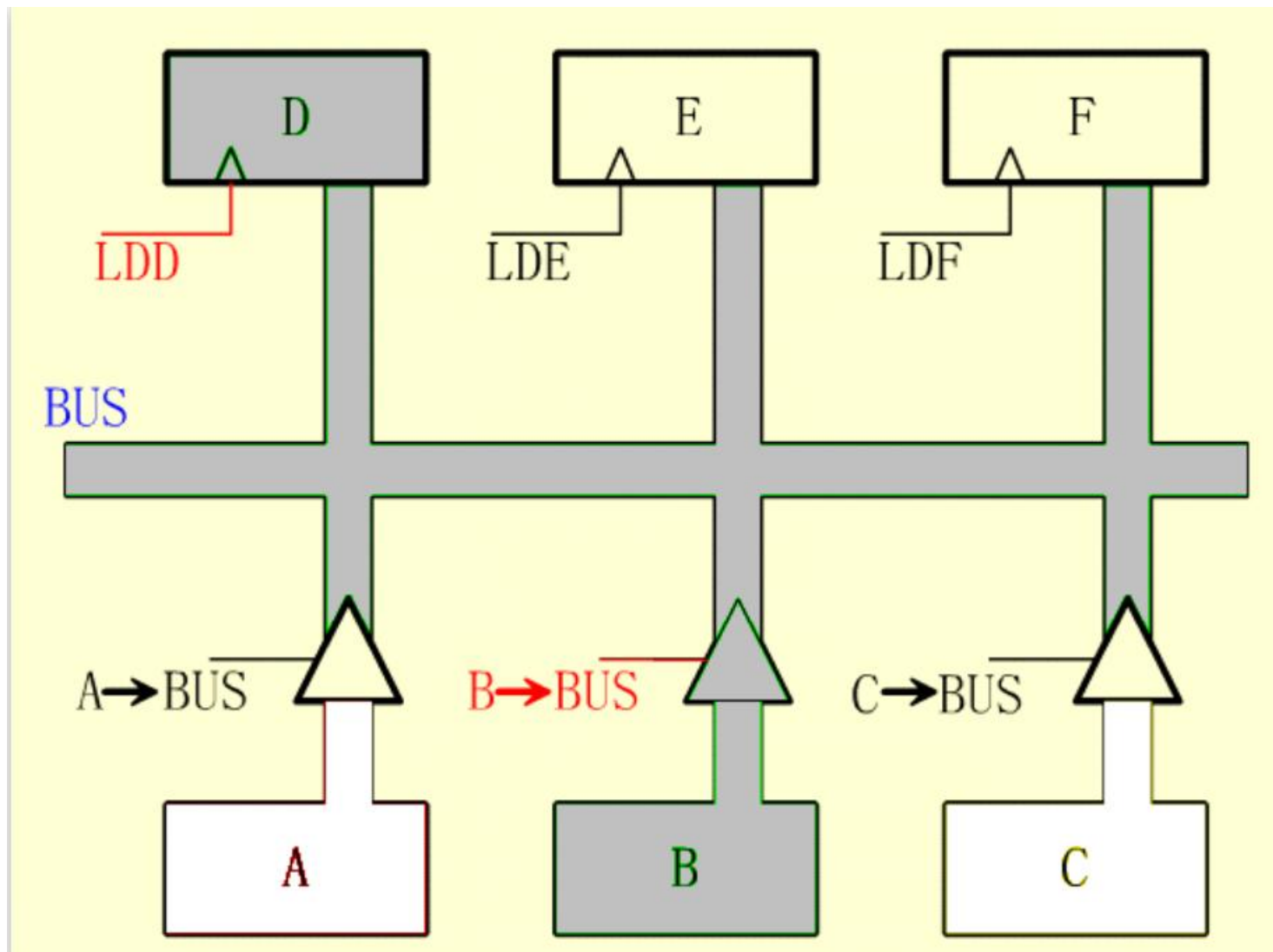
## 2. MDS图和ASM图的转换:



(a) 三状态机



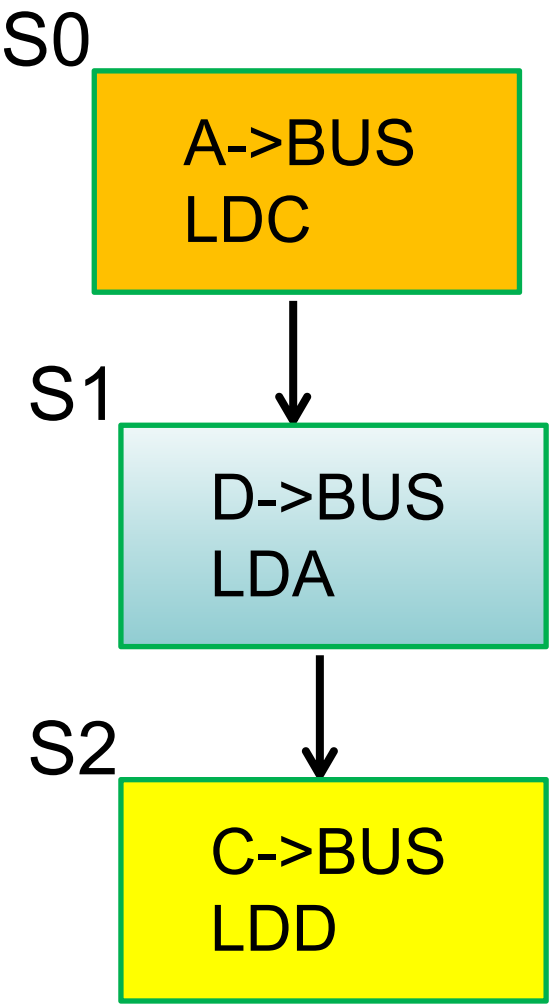
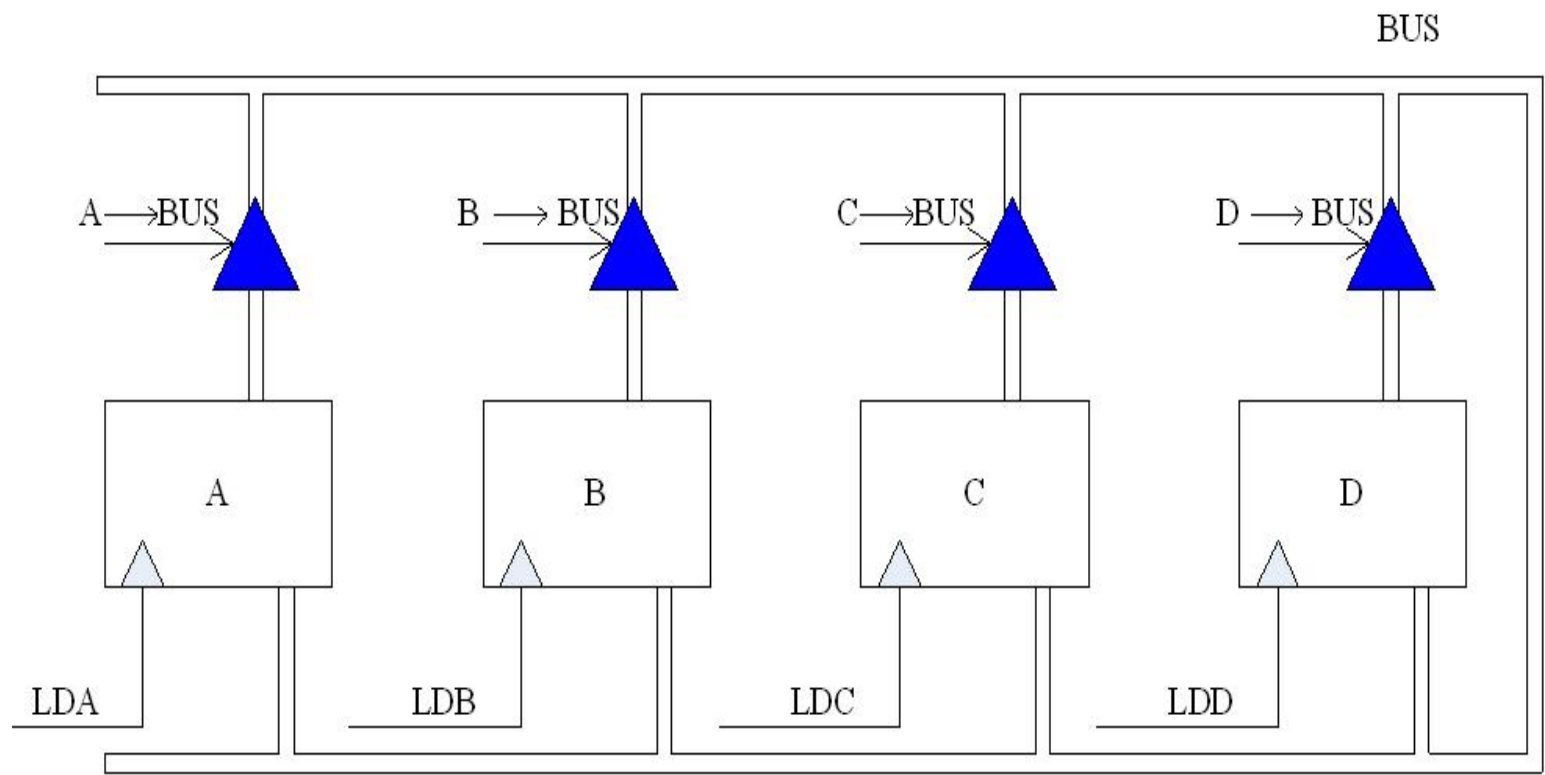
### 3. ASM图:



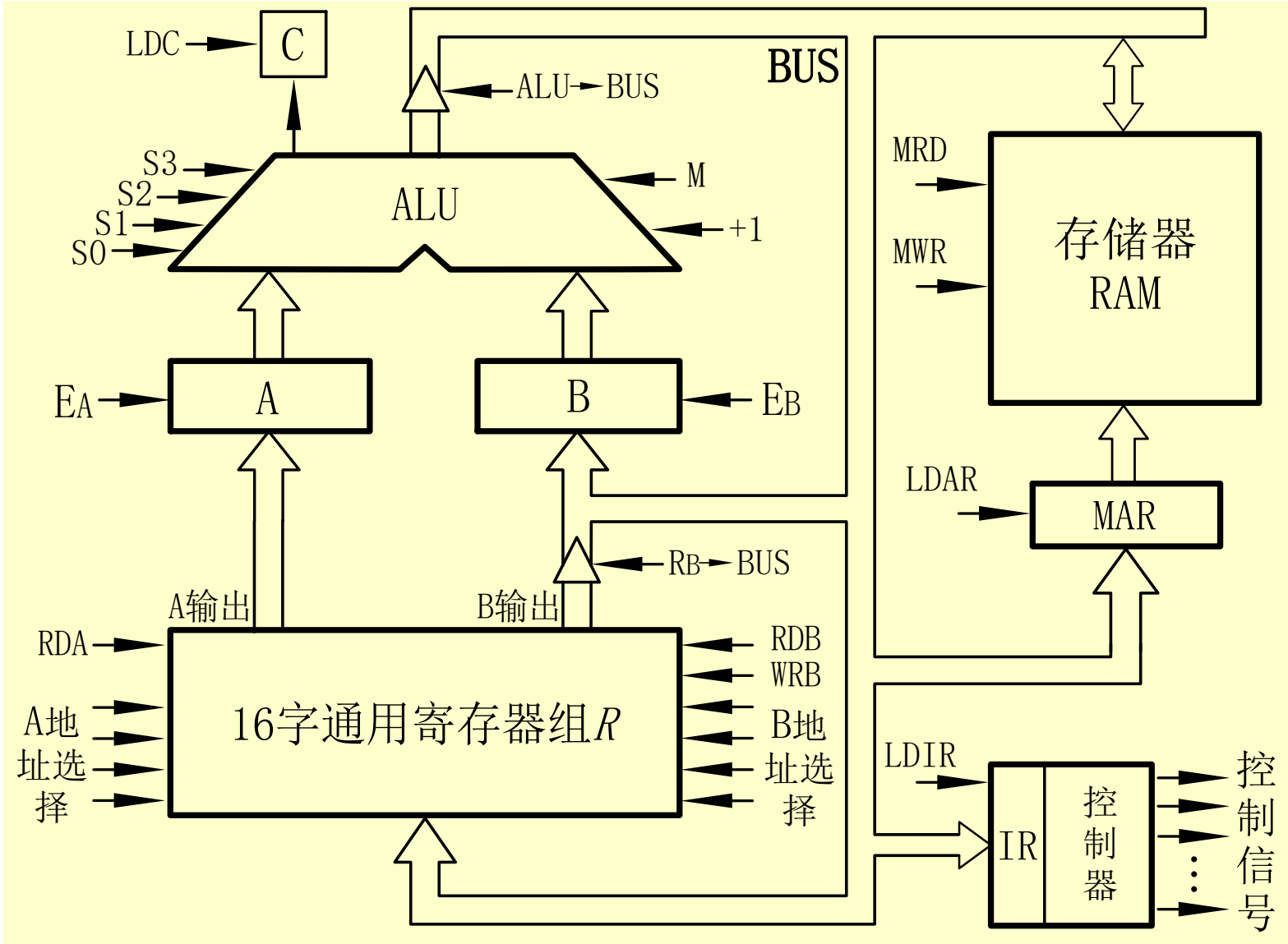
S

B- $\rightarrow$ BUS  
LDD

给出将A、D两个寄存器中已有的操作数进行交换的ASM图。



4. 数据通路实例



- 从寄存器到寄存器的操作

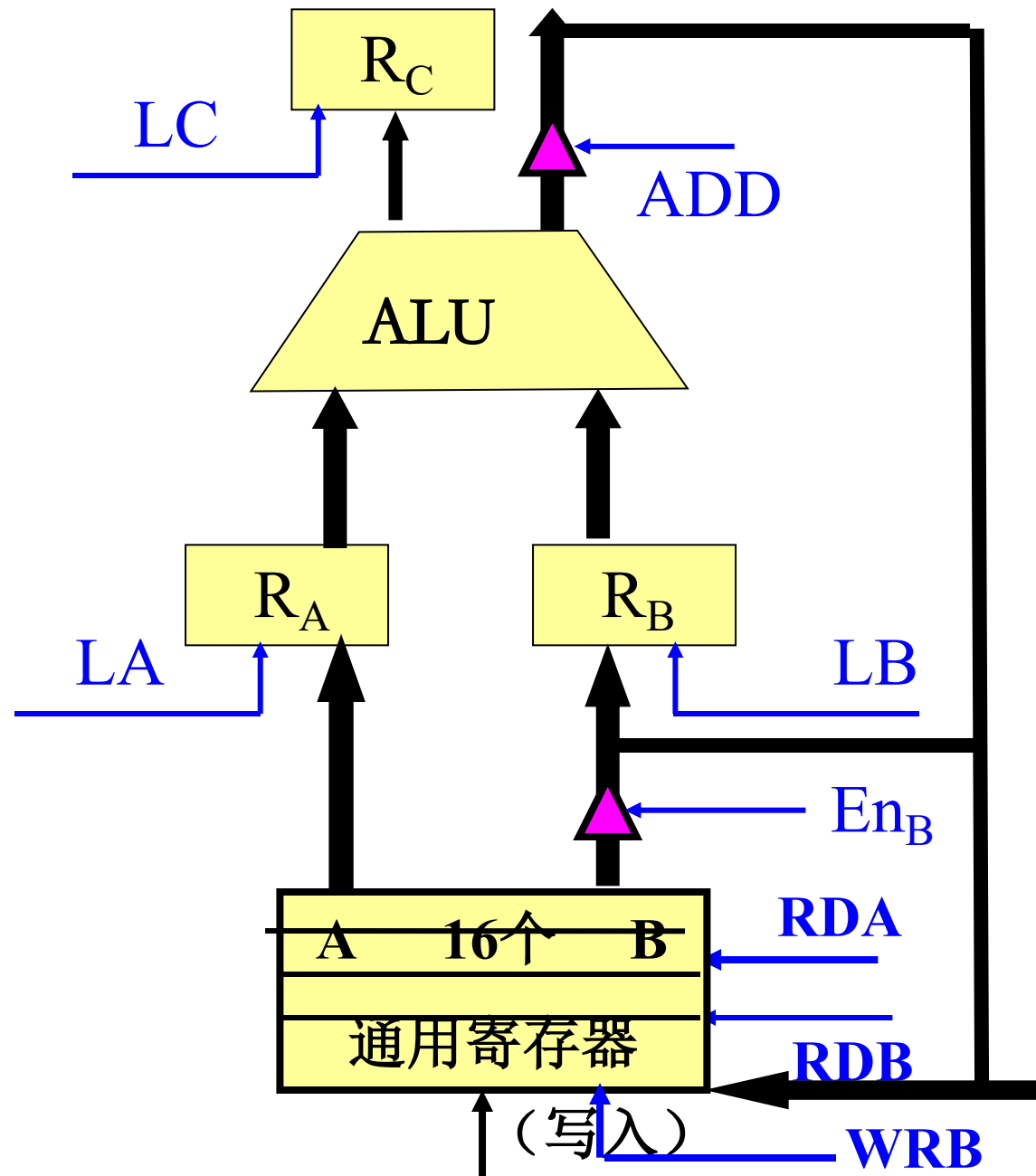
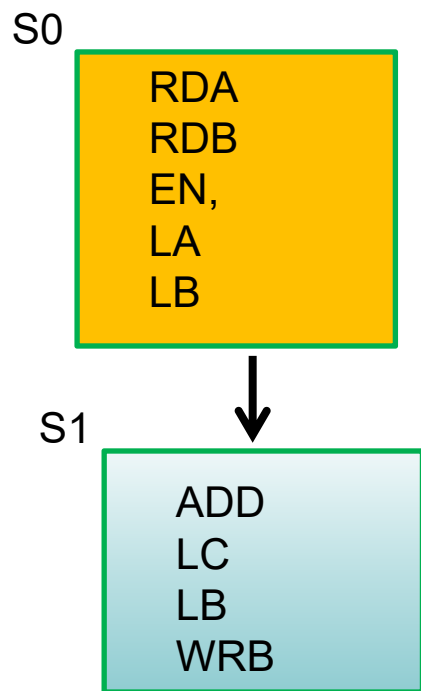
$$R_A + R_B \rightarrow R_B$$

寄存器控制信号：

接受方：打入脉冲；

发送方：三态门。

### 双总线通路

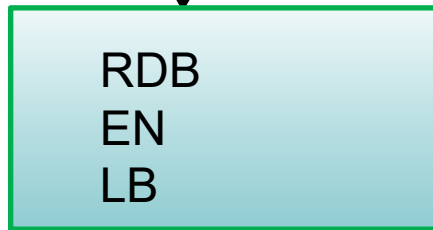


# 单总线通路

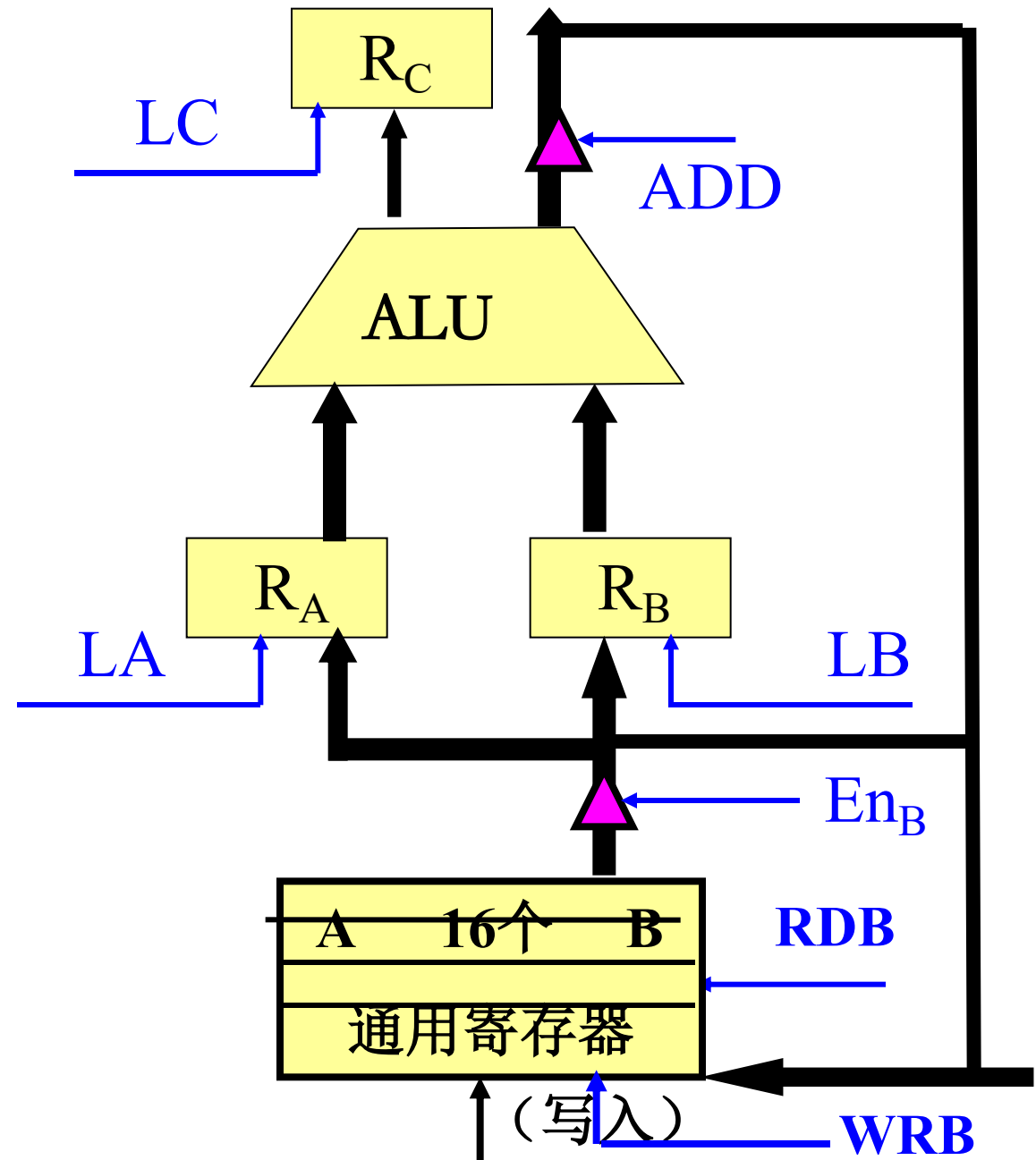
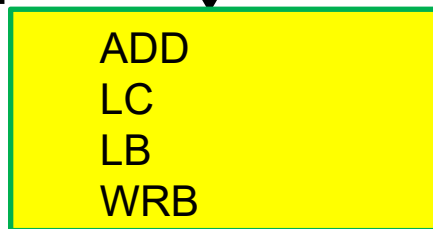
S0



S1

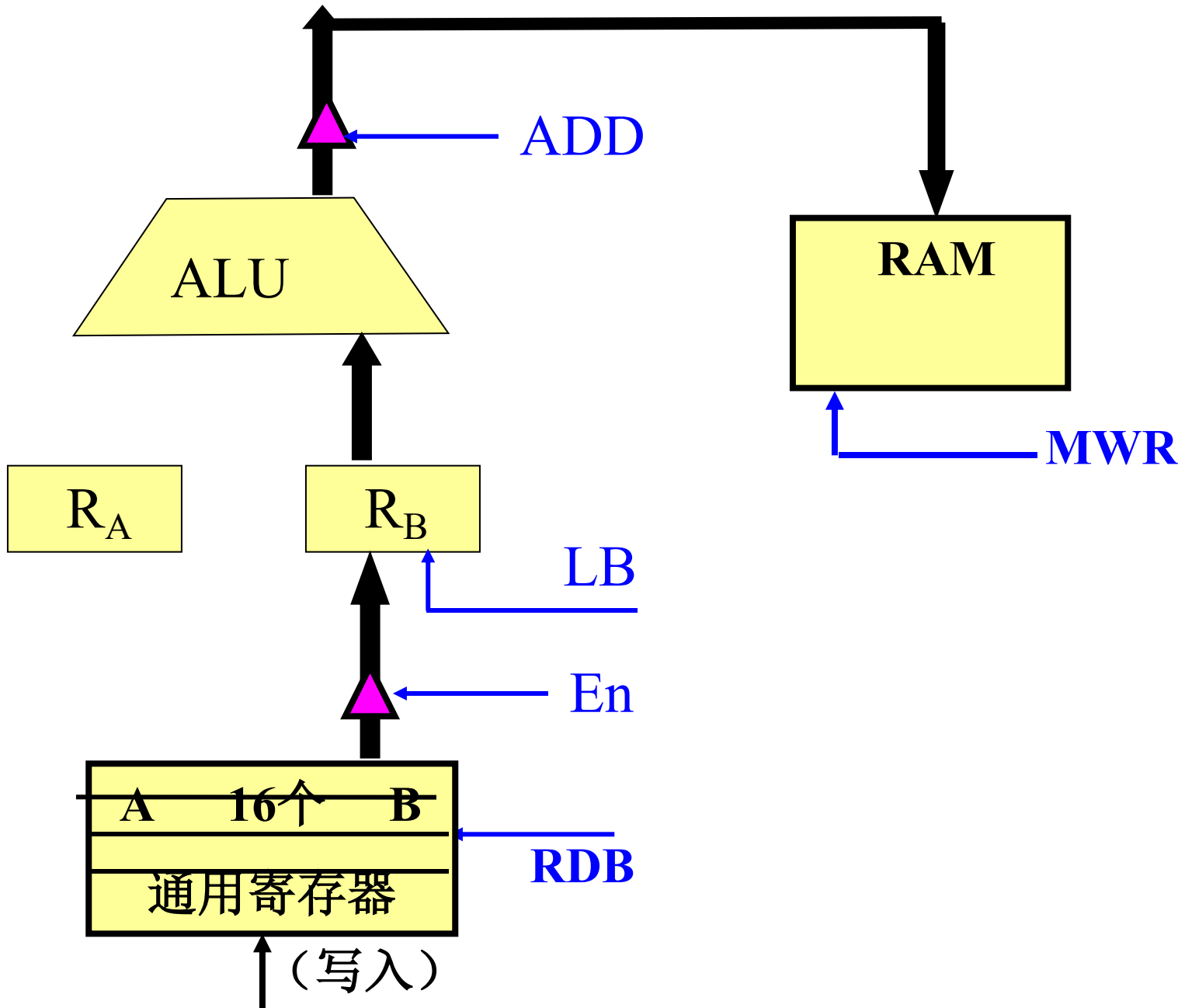
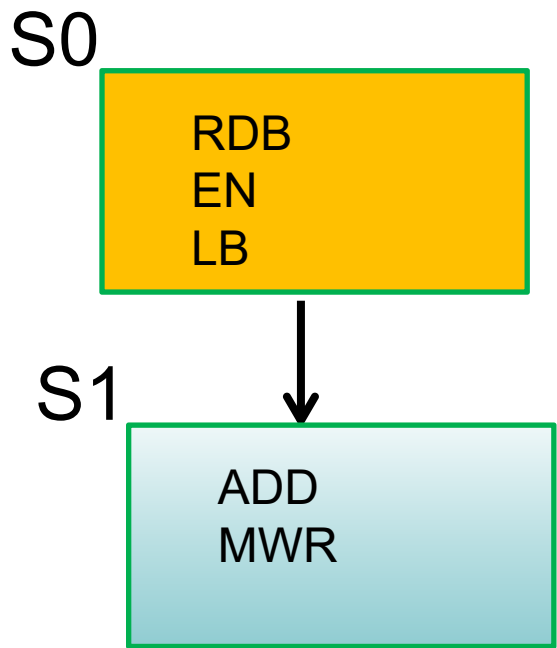


S2



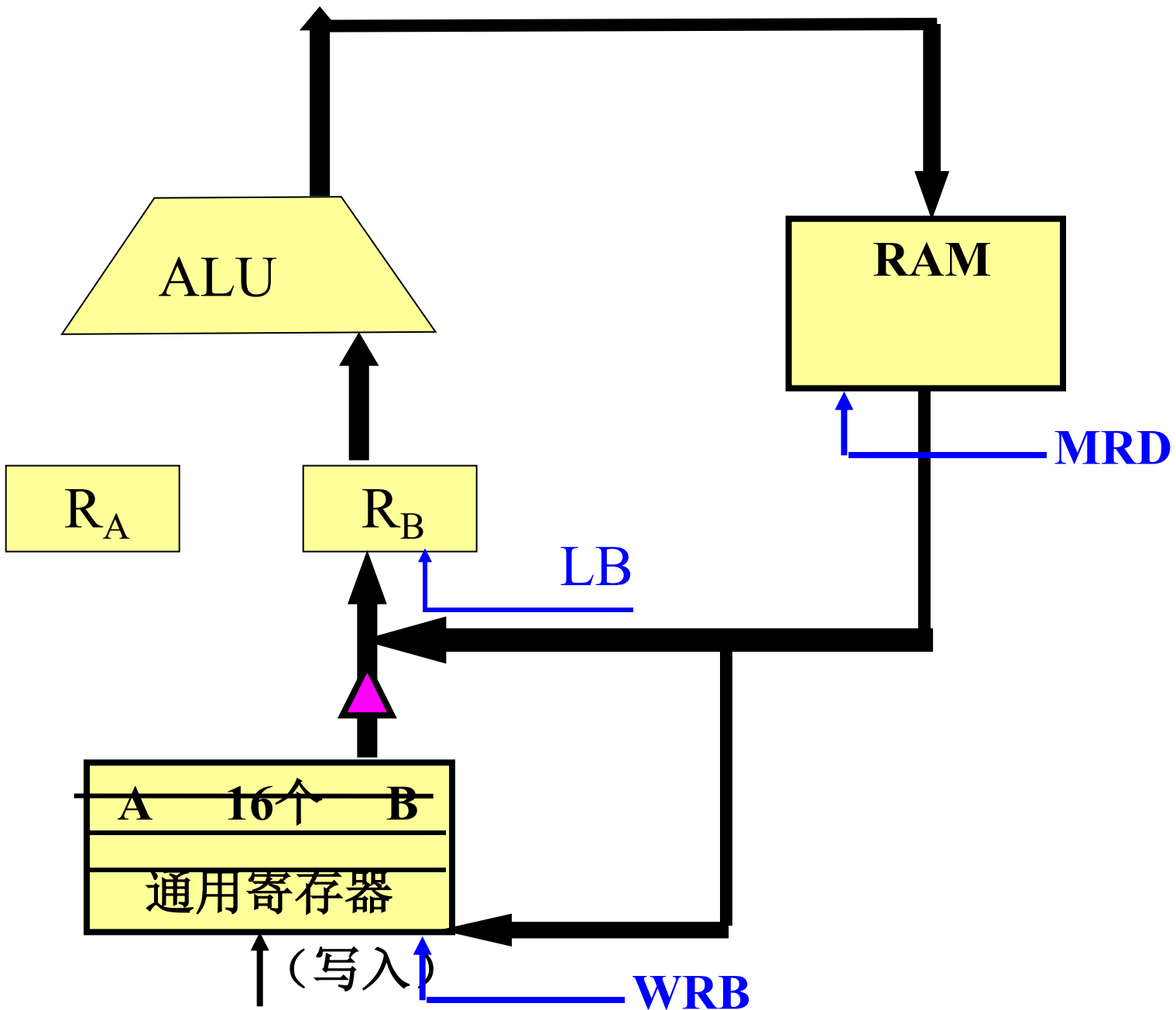
从寄存器到存储器的操作

$R_B \rightarrow \text{RAM}$



- 从存储器到寄存器的操作

RAM  $\rightarrow$   $R_B$





例：将四位二进制数X，Y分别存入寄存器A和B中，然后比较两数大小，使大数存入寄存器A，设计控制器的ASM图。

- 逻辑划分

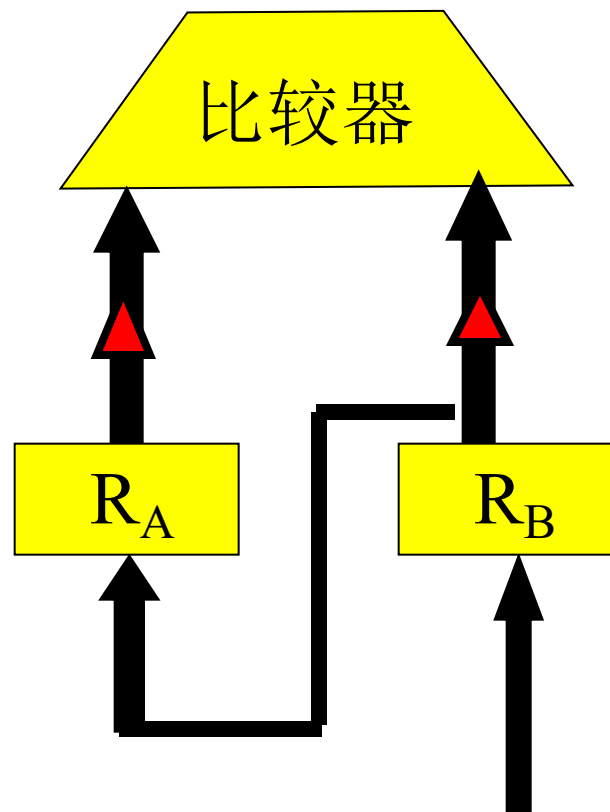
控制器、执行部件（寄存器、比较器）：

2个四位寄存器：A（X），B（Y）；

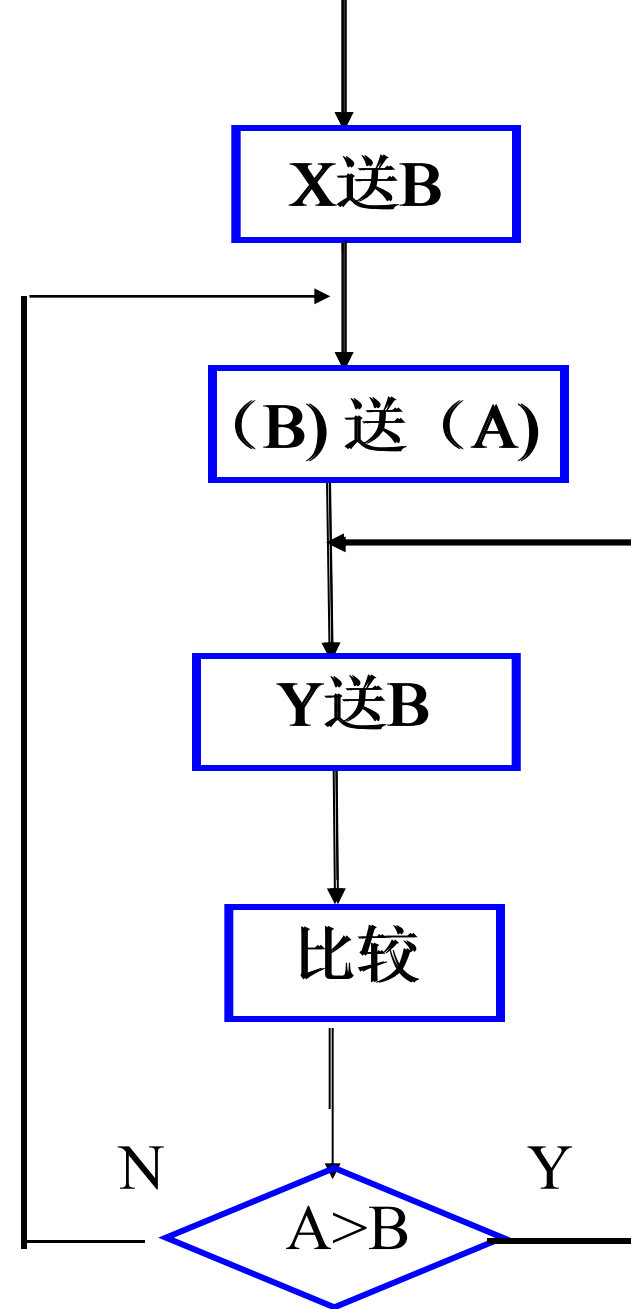
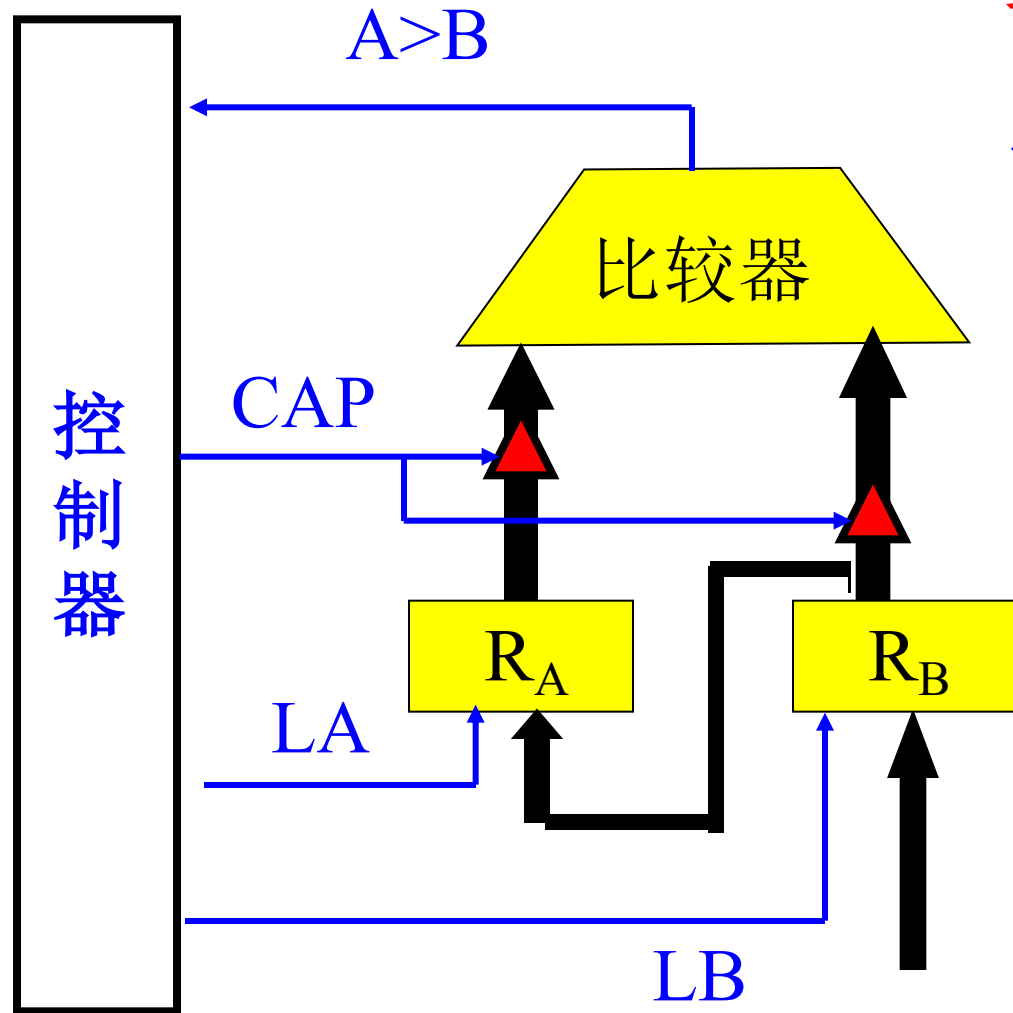
四位比较器： $C = 1 (A \geq B)$ ；

一个控制器。

- 数据通路

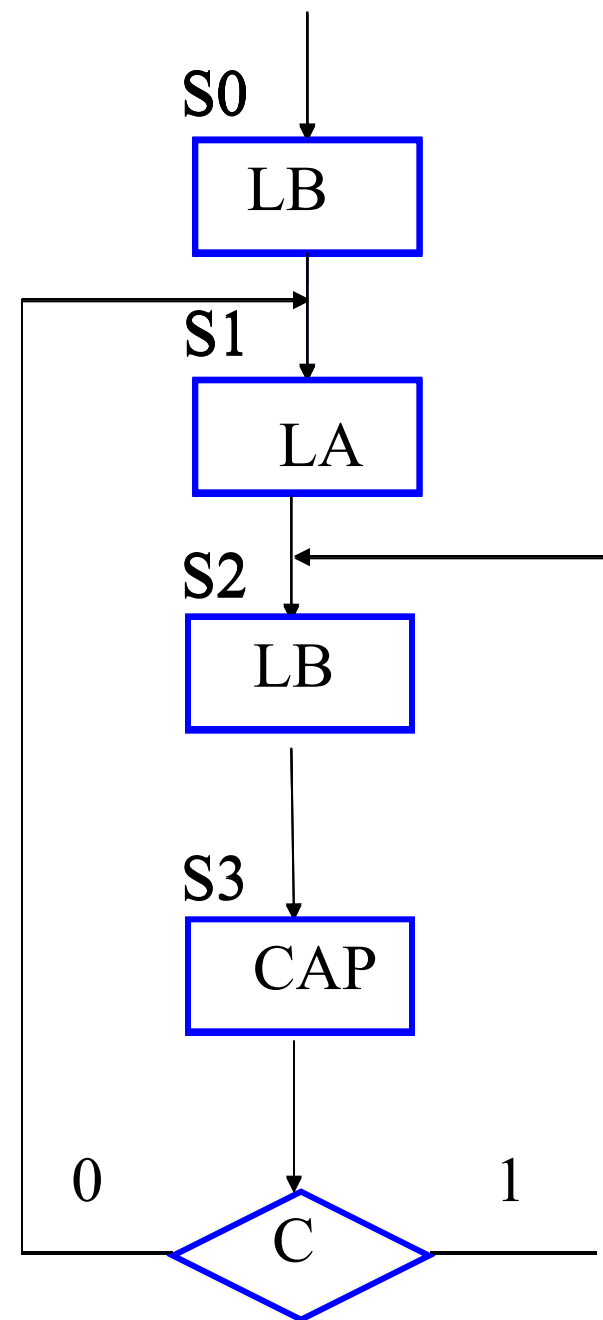
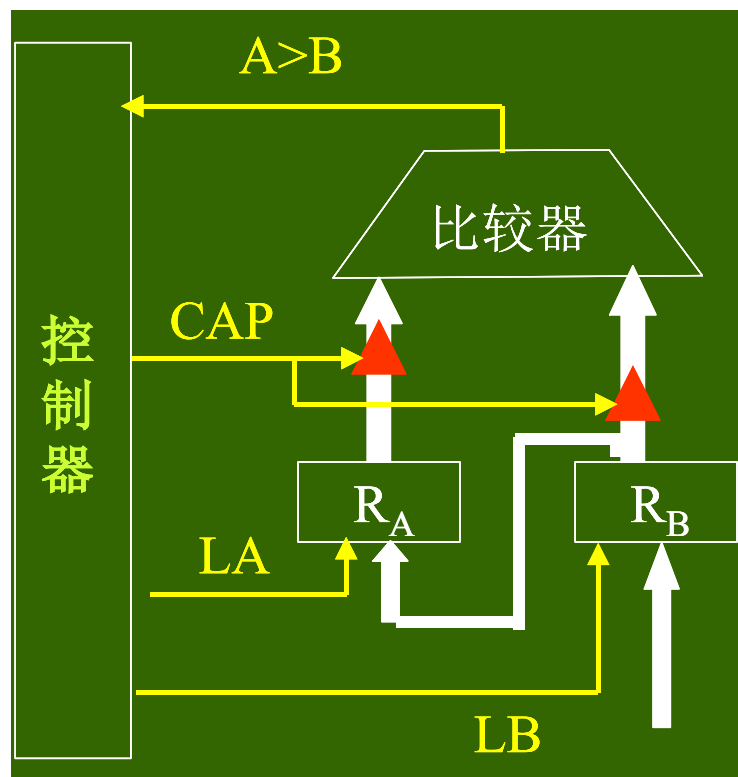


逻辑流程图:  
系统的工作过程

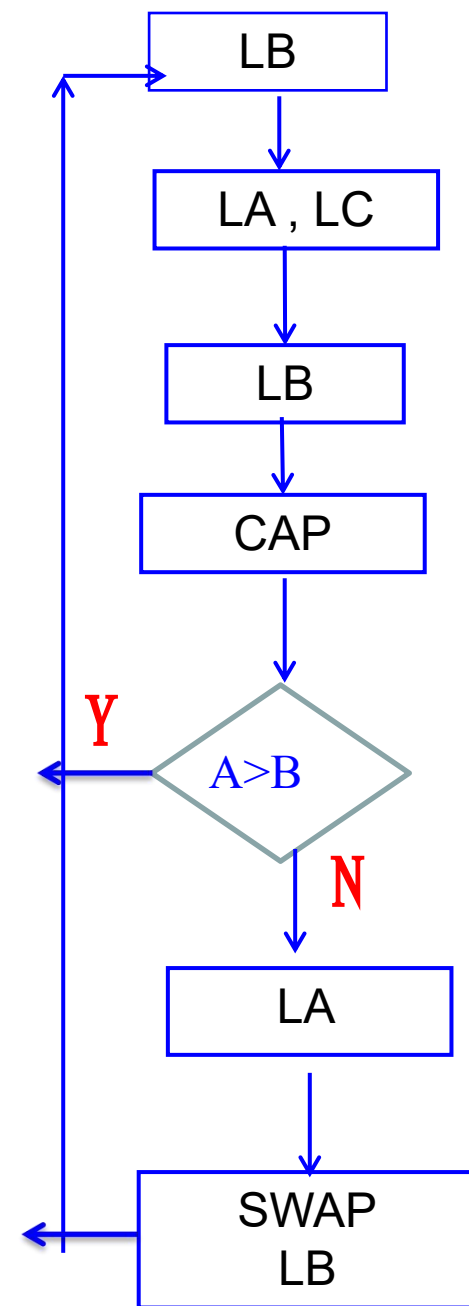
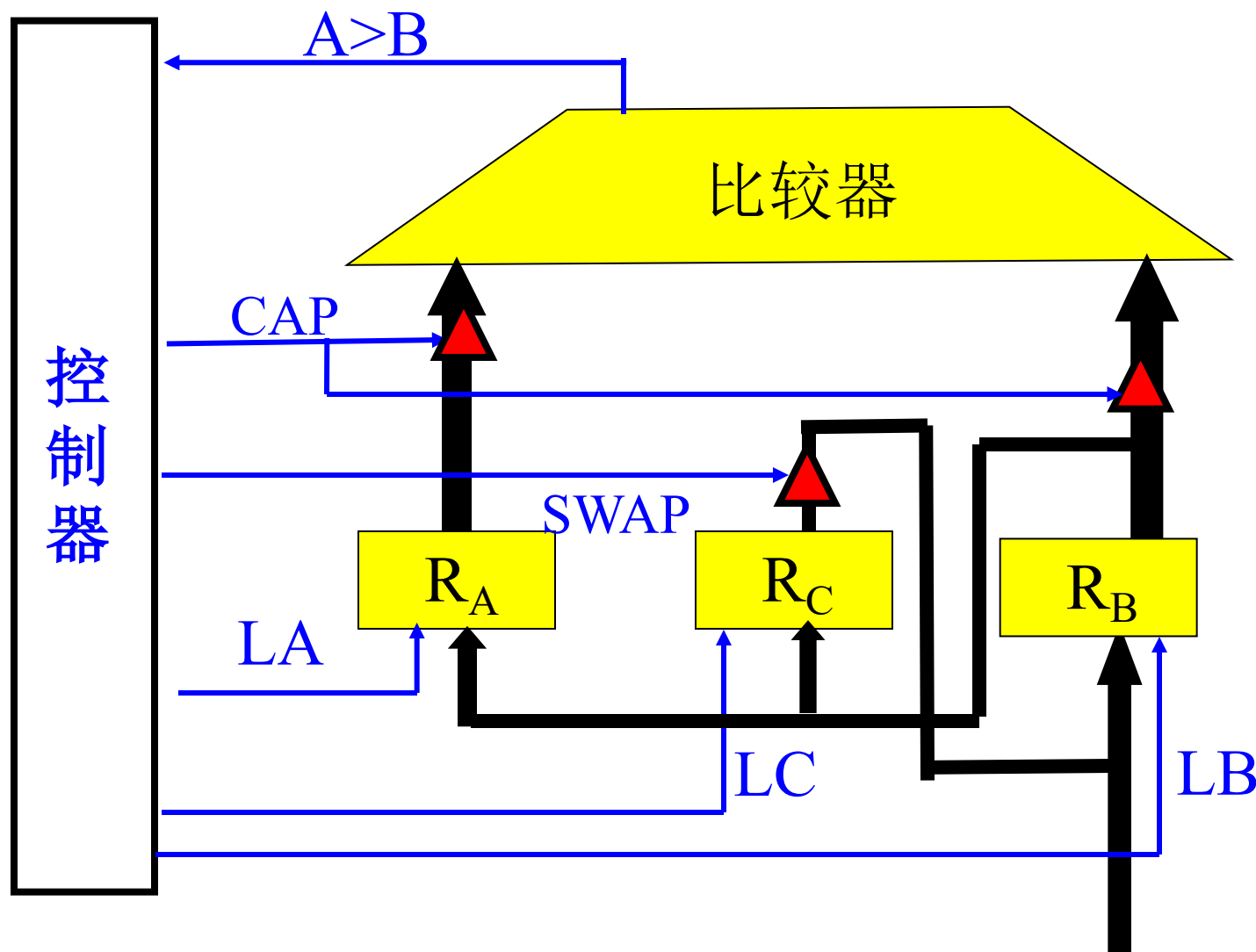


## • 控制器的ASM图

- $S_0$ : 打入命令 LB (脉冲),  $X \rightarrow B$  ;
- $S_1$ : 打入命令 LA (脉冲),  $(B) \rightarrow (A)$  ;
- $S_2$ : 打入命令 LB (脉冲),  $Y \rightarrow B$  ;
- $S_3$ : 比较命令 CAP (电位),  $(A) > (B)$   
比较器输出  $C = 1$ ;

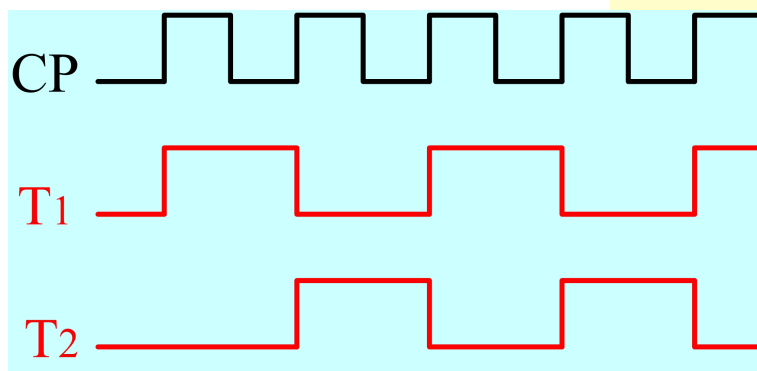
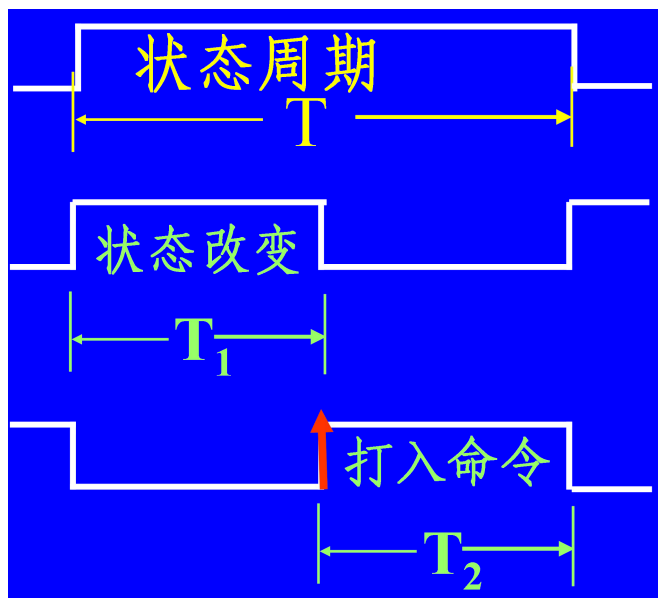
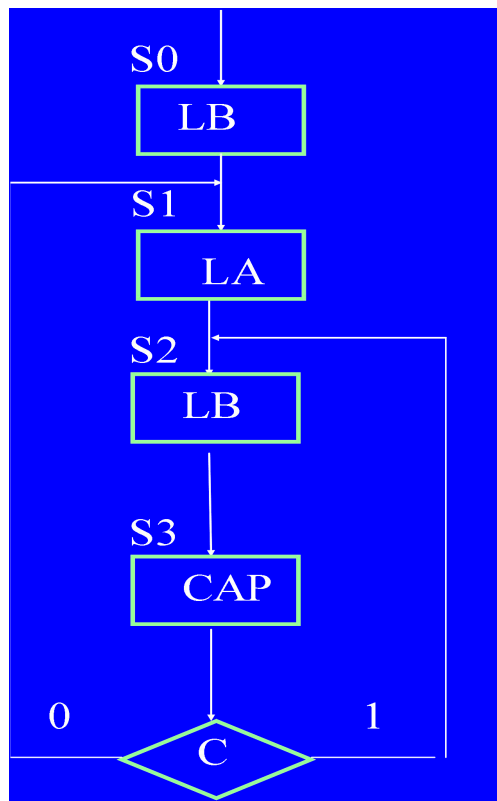


例：将四位二进制数X, Y分别存入寄存器A、B中，然后比较，使大数存入寄存器A，小数存入寄存器B，设计控制器的ASM图。

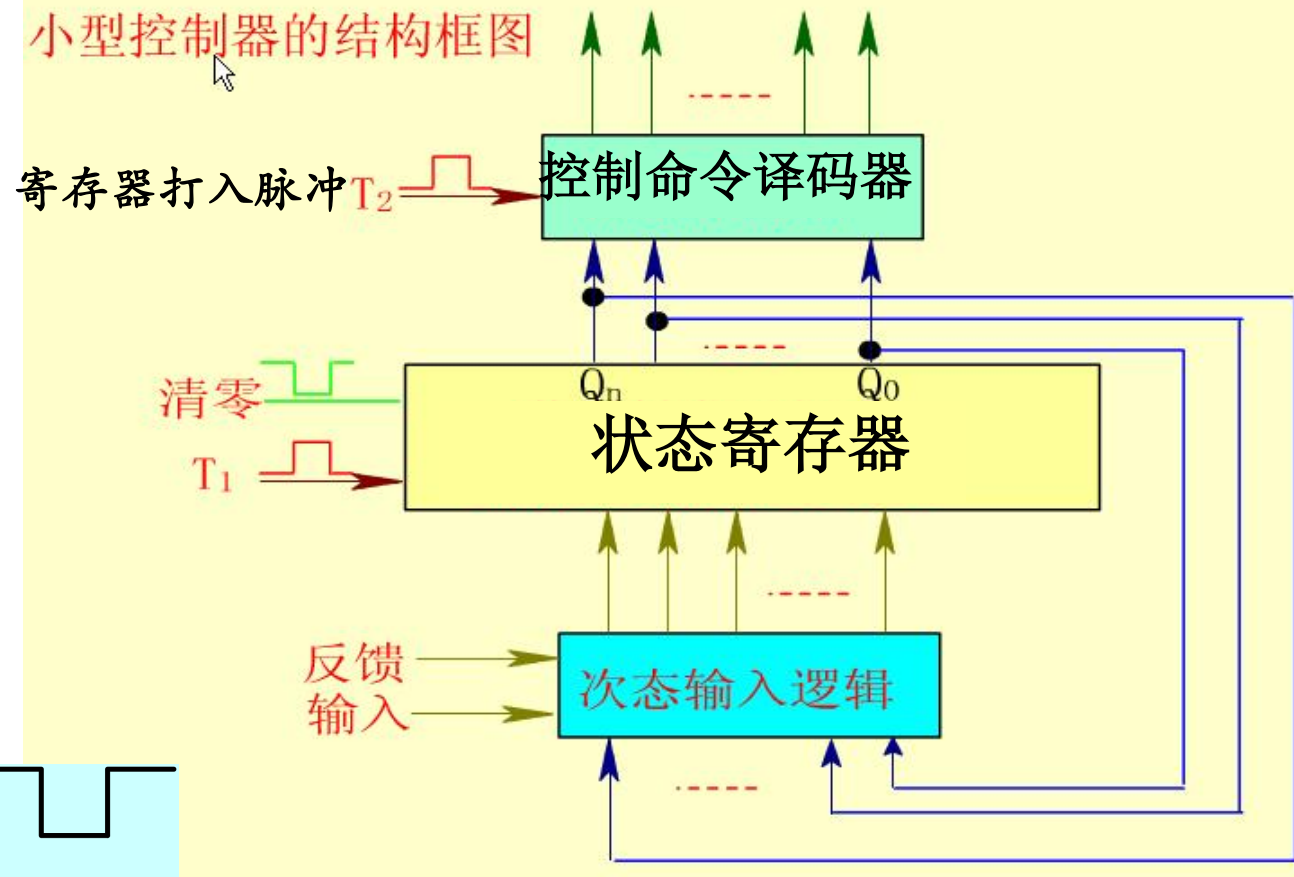


## § 6.4 小型控制器的设计

由硬件实现控制器：



小型控制器的结构框图



小型控制器

计数器型：

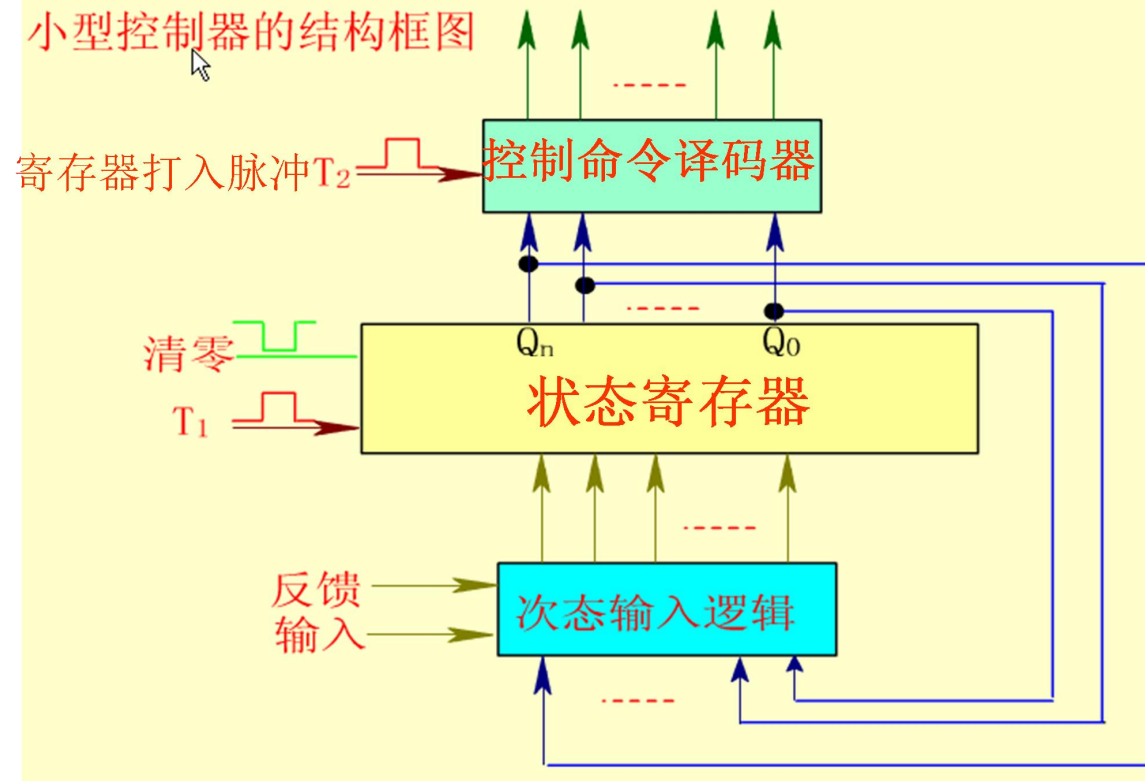
MUX型：

状态寄存器少，译码电路  
相同（较复杂）

定序型（一对一）：状态寄存器多，译码电路简单

无需化简

小型控制器的结构框图



# 1. 计数型控制器

适应于状态数较多， $N$  个触发器可构成  $2^n$  个控制状态。

设计方法同时序电路。

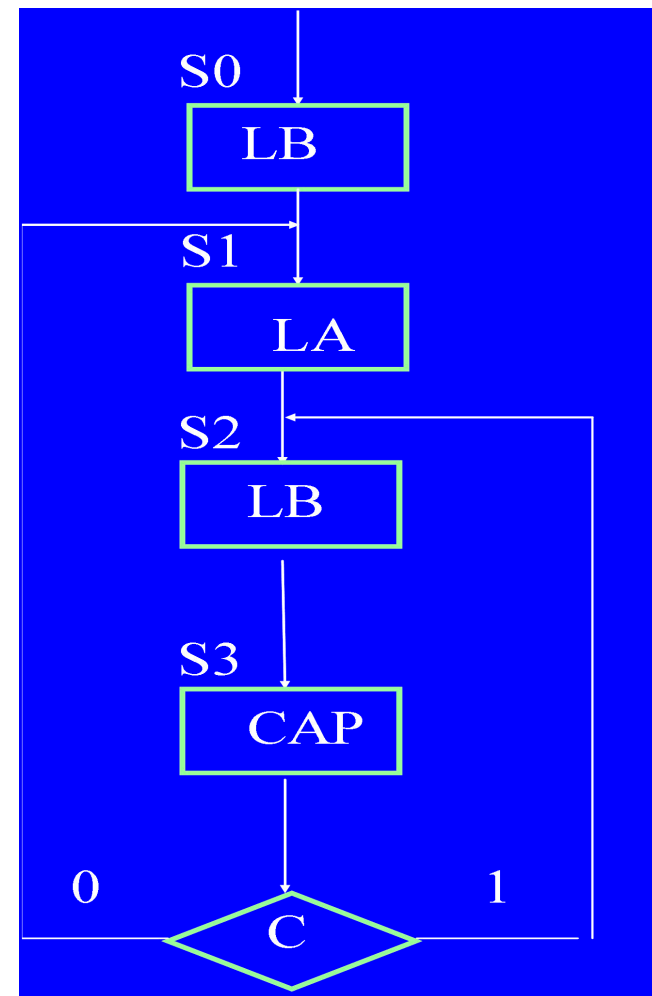
1) 给ASM图的状态框编码:

2) 由ASM 图得控制信号表达式 (译码):

3) 控制器的状态转移表:

4) 触发器驱动方程:

5) 电路实现:



1) 给ASM图的状态框编码(Q2Q1):

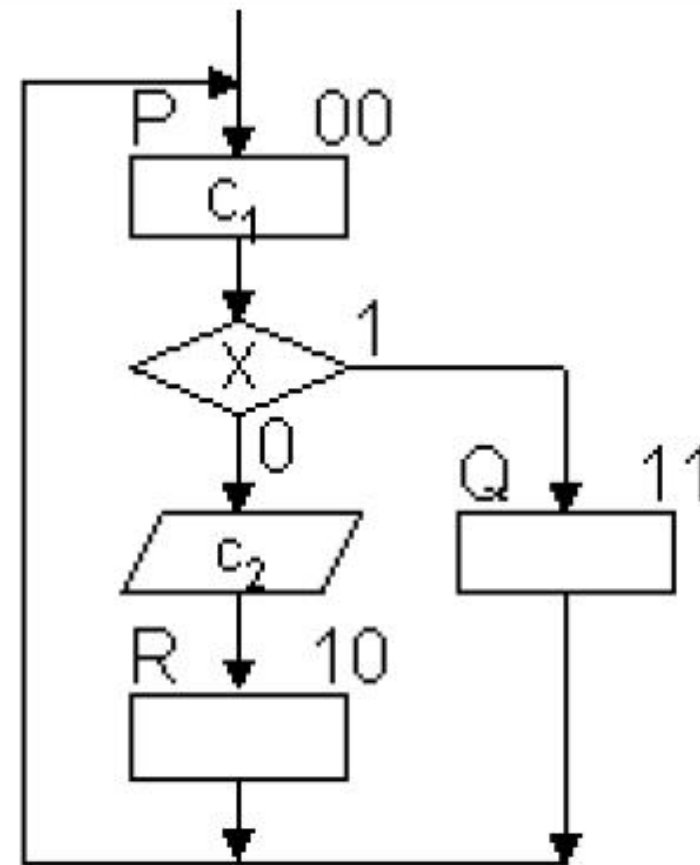
2) 由ASM 图得控制信号表达式:

$$C_1 = \overline{Q_2^n} \overline{Q_1^n}$$

$$C_2 = \overline{Q_2^n} \overline{Q_1^n} \overline{X}$$

3) 控制器的状态转移表:

$Q_2^n$	$Q_1^n$	$Q_2^{n+1}$	$Q_1^{n+1}$	转移条件
0	0	1	0	$\overline{X}$
0	0	1	1	X
1	0	0	0	
1	1	0	0	
0	1	0	0	





#### 4) 触发器驱动方程:

$Q_2^n$	$Q_1^n$	$Q_2^{n+1}$	$Q_1^{n+1}$	转移条件
0	0	1	0	$\overline{X}$
0	0	1	1	$X$
1	0	0	0	
1	1	0	0	
0	1	0	0	

总结公式:

NS: 次态

教材第三章最  
后一节

PS: 现态

C: 转移条件

$$Q_2^{n+1} = D_2 = \overline{Q_2^n} \overline{Q_1^n} (X + \overline{X}) = \overline{Q_2^n} \overline{Q_1^n}$$

$$J_2 = \overline{Q_1^n};$$

$$K_2 = 1$$

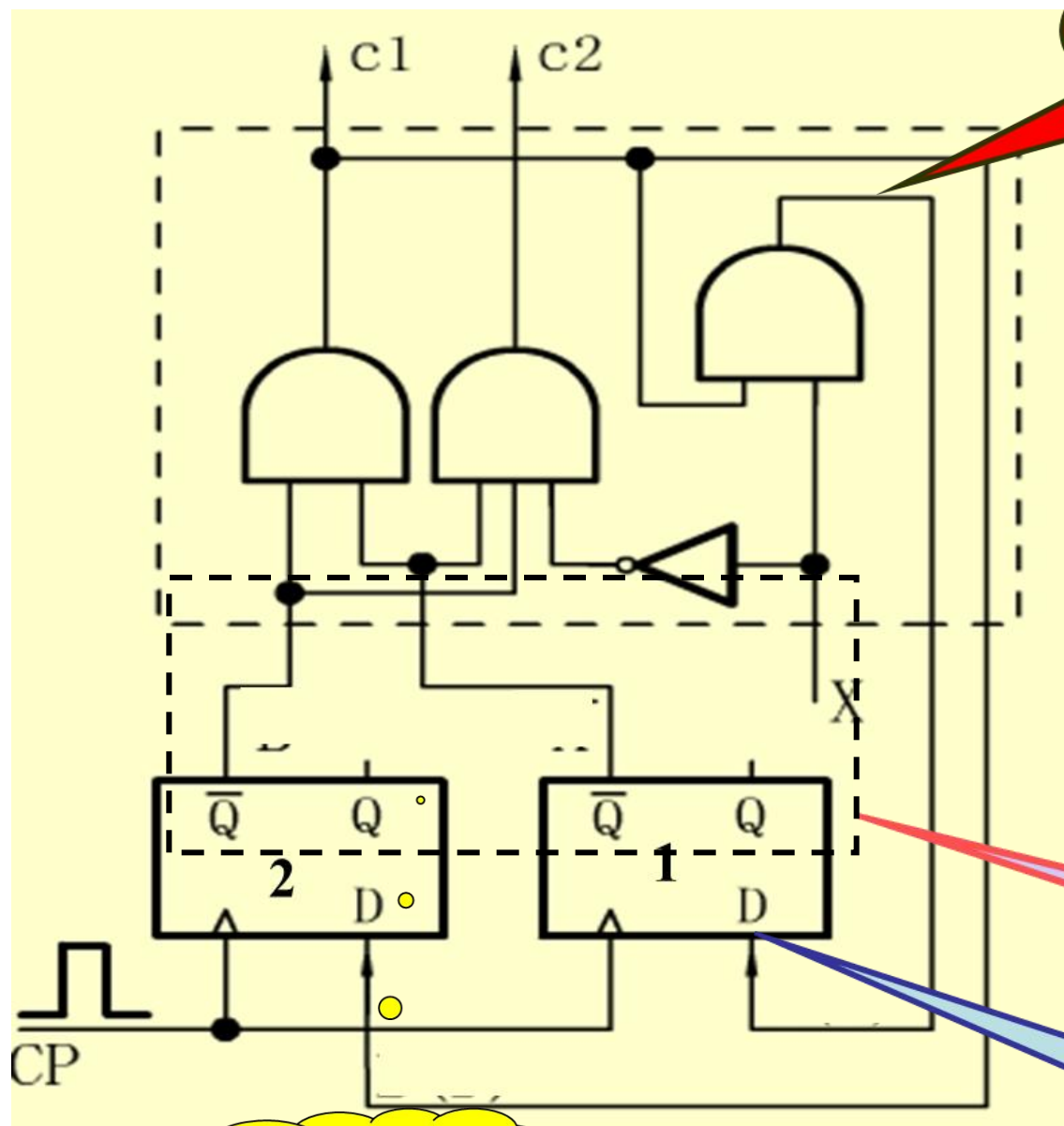
$$Q_1^{n+1} = D_1 = \overline{Q_2^n} \overline{Q_1^n} X$$

$$J_1 = \overline{Q_2^n} X;$$

$$K_1 = 1$$

$$NS = \sum PS \bullet C$$

## 5) 电路实现:



控制命令  
译码器

$$C_1 = \overline{Q_2^n} \overline{Q_1^n}$$

$$C_2 = \overline{Q_2^n} \overline{Q_1^n} \overline{X}$$

$$D_2 = \overline{Q_2^n} \overline{Q_1^n} (X + \overline{X}) = \overline{Q_2^n} \overline{Q_1^n}$$

$$D_1 = \overline{Q_2^n} \overline{Q_1^n} X$$

状态寄存器

次态输入逻辑

一拍完成