



南京理工大学

NANJING UNIVERSITY OF SCIENCE & TECHNOLOGY

# 第4章 指令系统与 控制单元

主讲：张功萱

©第1版 2023.08 张功萱



# 控制单元设计-逻辑控制器

## 第4.6节

1. 控制器的设计步骤有哪些？
2. 模型机采用什么方法设计？
3. 如何设计组合逻辑控制器？

## 4.6.1 设计的基本步骤

- 用硬联逻辑方法设计控制器的微操作控制信号形成部件，需要根据每条指令的要求，让节拍电位和脉冲有步骤地去控制机器的有关部件，一步一步地依次执行指令所规定的微操作序列，从而在一个指令周期内完成一条指令所规定的全部操作功能。



- 在设计控制器之前，需由计算机系统结构设计人员设计好**机器的指令系统、数据表示、时序系统**，在此基础上，才能进行控制器的硬件线路设计。
- 硬联逻辑控制器设计的设计步骤：
  - **1. 绘制指令操作流程**
  - 首先要分析指令操作流程，目的是确定指令执行的具体步骤，决定各步所需的控制命令。

- **指令操作流程圖：**根据机器指令的结构格式、数据表示方式及各种运算的算法，把每条指令的执行过程分解成若干功能部件能实现的基本微操作，并以图的形式排列成有先后次序、相互衔接配合的流程。
- 指令流程图可以比较形象、直观地表明一条指令的执行步骤和基本过程。

- 绘制指令流程图的**两种思路**。
- ① **以指令为线索**，按指令类型分别绘制各条指令的流程。
- 这种方法对一条指令的全过程有清晰的线索，易于理解。
- ② **以周期为线索**，按机器周期拟定各类指令在本周期内的操作流程，再以操作时间表形式列出各个节拍内所需的控制信号及它们的条件。
- 这种方法便于微操作控制信号的综合、化简，容易取得优化结果。
- 为理解控制器设计方法，模型机设计基本采用了后一种方法。

## 2. 编排指令操作时间表

- **指令操作时间表**：把指令流程图中的各个微操作具体落实到各个机器周期的相应节拍和脉冲中去，并以微操作控制信号的形式编排成一张表。
- 指令操作时间表用于形象地表明控制器应该在什么时间，根据什么条件发出哪些微操作控制信号。指令操作时间表是指令流程图的进一步具体化。



### 3. 进行微操作综合

- 对操作时间表中各个微操作控制信号分别按其条件进行归纳、综合，列出其综合的逻辑表达式，并进行适当的调整、化简，得到比较合理的逻辑表达式。



## 4. 设计微操作控制信号形成部件

- 根据各个微操作控制信号的逻辑表达式，用硬件逻辑电路加以实现。
- 实现方法：
  - ① 根据逻辑表达式画出逻辑电路图，用组合逻辑网络实现
  - ② 直接根据逻辑表达式用PLD器件如PLA、PAL、GAL等实现。

## 4.6.2 模型机的设计方法

### • 1. 指令流程图

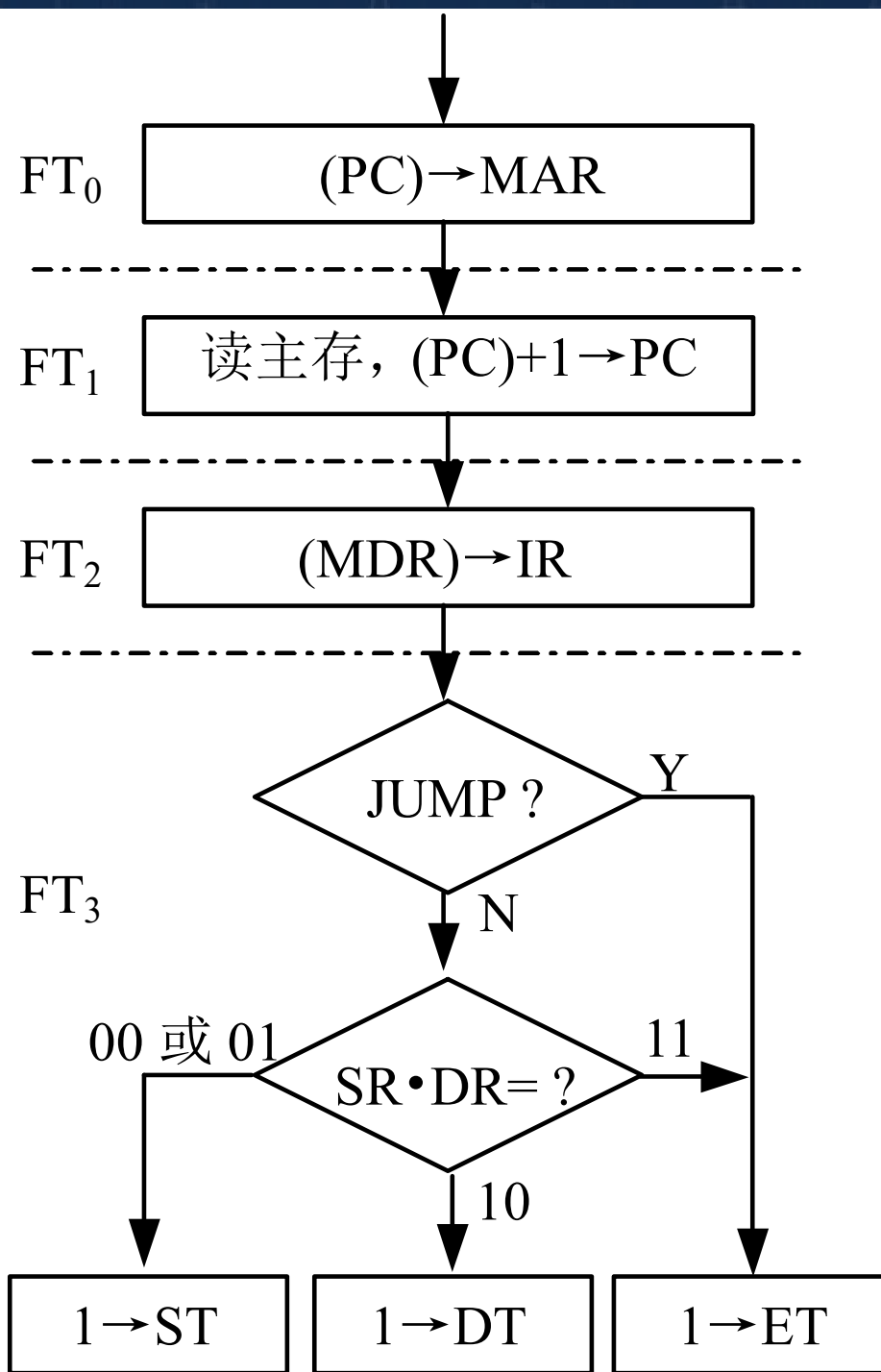
- 涉及访存的周期通常需要有4个节拍，其中3个节拍用于访存取数，1个节拍用于操作和判断。各节拍的操作为：
- $T_0$ ：将访存地址送入MAR。
- $T_1$ ：发访存请求信号和读信号，从主存读出信息送入MDR。
- $T_2$ ：将读出信息从MDR传送到有关寄存器。
- $T_3$ ：进行相关操作和判断。



# 1) 取指周期 (FT) 操作流程图

- 因为任何指令都必须进入取指周期，所以取指操作也称为**取指公操作**。
- 在取指周期内，各节拍的操作为：
- **FT<sub>0</sub>**: 指令地址→MAR
- **FT<sub>1</sub>**: 读指令→MDR, PC+1→PC
- **FT<sub>2</sub>**: 将MDR中的现行指令传送到IR，并由ID进行译码。
- **FT<sub>3</sub>**: 根据指令译码结果进行判断，以确定下面应进入哪个机器周期。

# 取指令操作流程





- 在 $FT_3$ 节拍中，首先判断所取指令是否为转移类指令。模型机中共定义了4条转移指令：

指令	操作码 $IR_{15}IR_{14}IR_{13}IR_{12}$
JP	1010
JC	1011
JZ	1100
JSR	1101

直接进入执行周期ET

$$\begin{aligned} JUMP &= JP\_op + JC\_op \\ &+ JZ\_op + JSR\_op \\ &= IR_{15}(IR_{14} \oplus IR_{13}) \end{aligned}$$


- 如果源（SR）和目的（DR）操作数均采用寄存器寻址，也可以直接进入**执行周期ET**；否则，如果为双操作数指令需进入**取源周期ST**；如果为单操作数指令需进入**取目的周期DT**。
- 指令有源操作数SR时：IR<sub>11</sub>IR<sub>10</sub>IR<sub>9</sub>表示寻址方式
- 指令有目的操作数DR时：IR<sub>5</sub>IR<sub>4</sub>IR<sub>3</sub>表示寻址方式
- 令  $SR = \overline{IR_{11}IR_{10}IR_9}$      $DR = \overline{IR_5IR_4IR_3}$

SR·DR	操作数寻址情况	
00	源和目的均不是寄存器寻址	} 进入ST
01	源不是寄存器寻址，目的是寄存器寻址	
10	源是寄存器寻址，目的不是寄存器寻址	} 进入DT
11	源和目的均是寄存器寻址	} 进入ET



# 取指操作时间表

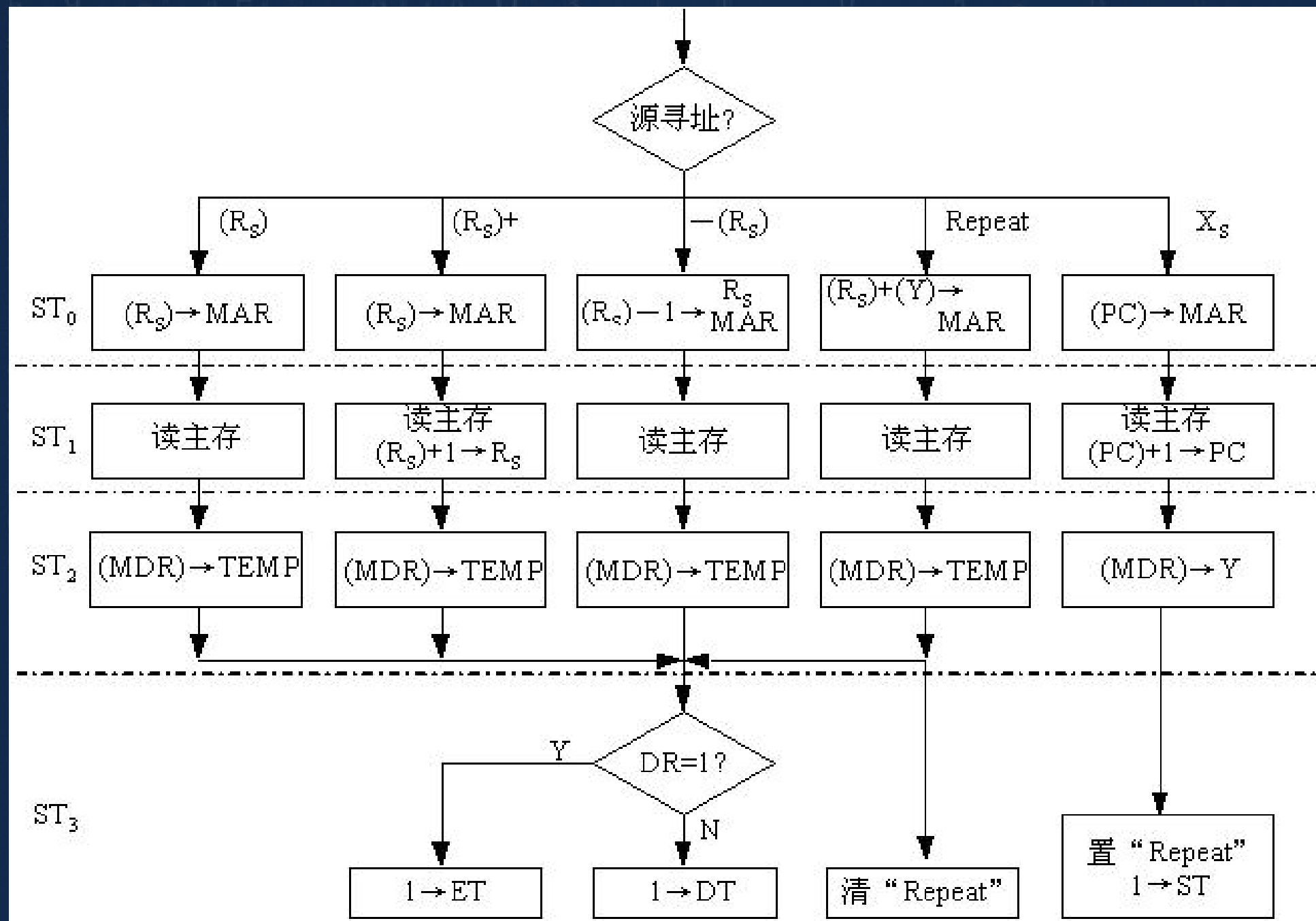
节拍	控制电位信号	控制脉冲信号
FT <sub>0</sub>	PC→BUS <sub>1</sub> , S <sub>3</sub> S <sub>2</sub> S <sub>1</sub> S <sub>0</sub> M, DM	CPMAR[P]
FT <sub>1</sub>	MREQ, R/ $\overline{W}$ =1, PC→BUS <sub>1</sub> , S <sub>3</sub> S <sub>2</sub> S <sub>1</sub> S <sub>0</sub> $\overline{M}$ , C <sub>0</sub> =1, DM	CPPC[P]
FT <sub>2</sub>	MDR→BUS <sub>1</sub> , S <sub>3</sub> S <sub>2</sub> S <sub>1</sub> S <sub>0</sub> M, DM	CPIR[P]
FT <sub>3</sub>	1→ST[ $\overline{JUMP} \cdot \overline{SR}$ ] 1→DT[ $\overline{JUMP} \cdot \overline{SR} \cdot \overline{DR}$ ] 1→ET[JUMP+SR·DR]	<div> CPFT[<math>\overline{P}</math>]  CPST[<math>\overline{P}</math>]  CPDT[<math>\overline{P}</math>]  CPET[<math>\overline{P}</math>] </div> <div> } T<sub>END</sub>·<math>\overline{P}</math> </div>

- P表示时钟脉冲的前沿（上升沿）起作用。
- $\overline{P}$ 表示时钟脉冲的后沿（下降沿）起作用。



## 2) 取源周期 (ST) 的操作流程图

- 双操作数指令的源寻址方式为非寄存器寻址时，需进入取源周期，完成源操作数寻址并取源操作数。取出的源操作数暂存在暂存器**TEMP**中。
- 不同寻址方式有效地址形成方法不同，因此操作流程也不同。
- 由于变址寻址是到主存取出变址值，形成有效地址后再取数，需访问两次主存。因此，当**源寻址为变址寻址时，取源周期需重复一次**。
- 在取源周期ST的 $T_3$ 建立Repeat信号和 $1 \rightarrow ST$ 信号，可重复进入ST。
- 在重复的ST的 $T_3$ ，将Repeat信号清除，即可进入下一个周期。





- (1) 源操作数采用寄存器间接寻址:
- $ST_0$ : 寄存器中的地址信息 $\rightarrow$ MAR
- $ST_1$ : 读主存数据 $\rightarrow$ MDR
- $ST_2$ : (MDR) $\rightarrow$ TEMP
- $ST_3$ : 根据是否需要读目的操作数, 判断应进入执行周期还是取目的周期。



- (2) 源操作数采用变址寻址:
- 由于变址寻址需到主存取出变址值, 再形成有效地址, 所以需两次访问主存, 即需重复执行一次取源周期。
- ① 第一取源周期:
- 读变址值 $\rightarrow Y$ , 并在 $T_3$ 时, 建立Repeat信号和 $1 \rightarrow ST$ 信号, 使之重复进入ST。
- ② 重复取源周期:
- 计算有效地址, 读入数据 $\rightarrow TEMP$ , 并在重复ST的 $T_3$ 时, 将Repeat信号清除, 以便进入下一个周期。



- 源操作数寻址方式的判断条件：
- 寄存器间接寻址： $(R_S) = \overline{IR_{11}} \overline{IR_{10}} IR_9$
- 自增型寄存器间接寻址： $(R_S) + = \overline{IR_{11}} IR_{10} \overline{IR_9}$
- 自减型寄存器间接寻址： $-(R_S) = \overline{IR_{11}} IR_{10} IR_9$
- 变址寻址： $X_S = IR_{11} \overline{IR_{10}} \overline{IR_9}$
- 表**4-16**是取源操作时间表  $ST_0$ - $ST_3$

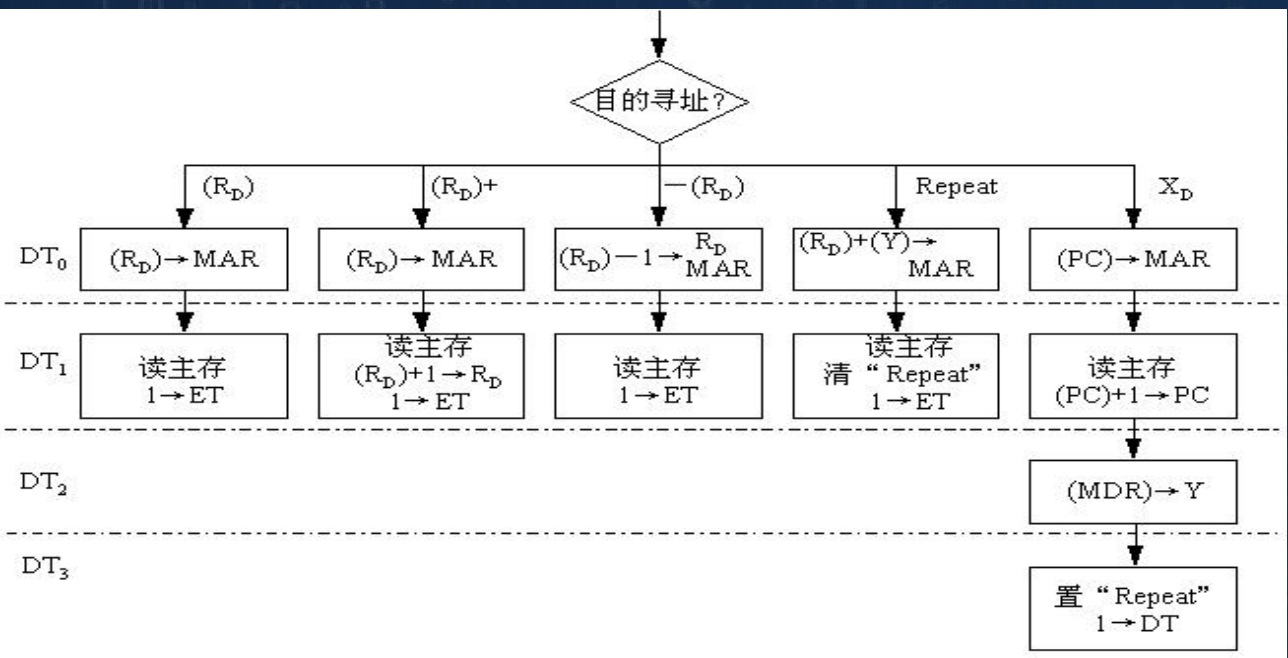




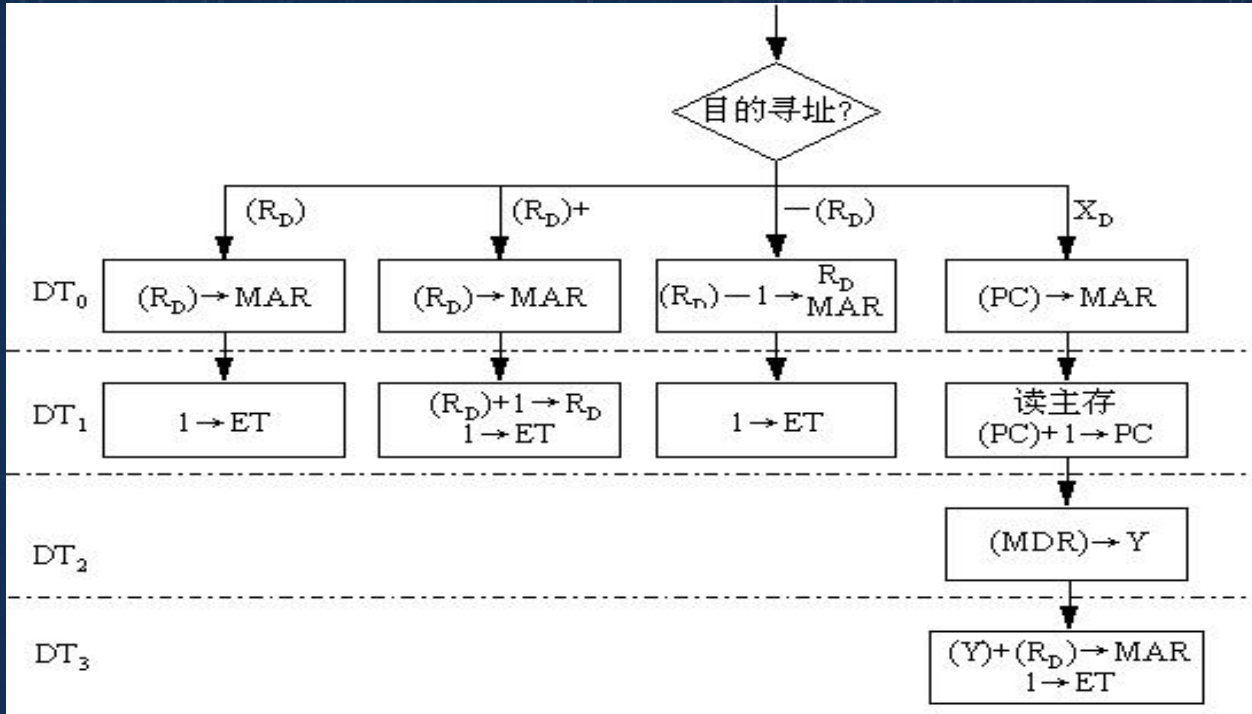
### 3) 取目的周期(DT)的操作流程图

- 当双操作数指令和单操作数指令的目的寻址方式为非寄存器寻址时，进入取目的周期完成目的操作数地址的寻址并读取目的操作数。
- 由于目的操作数在执行周期才送入ALU的A输入端运算，所以取出的目的操作数暂不传送，而保留在MDR中，因此对于非变址寻址的寻址方式只安排两个节拍即可完成操作。
- **MOV**指令的目的寻址为非寄存器寻址时，也需进入目的周期，但因**MOV**指令只需传送目的地址而不需取目标操作数，因而操作比较简单。

# 非MOV指令的DT周期



# MOV指令的DT周期



● 表4-17是取目的的操作时间表DT<sub>0</sub>-DT<sub>3</sub>

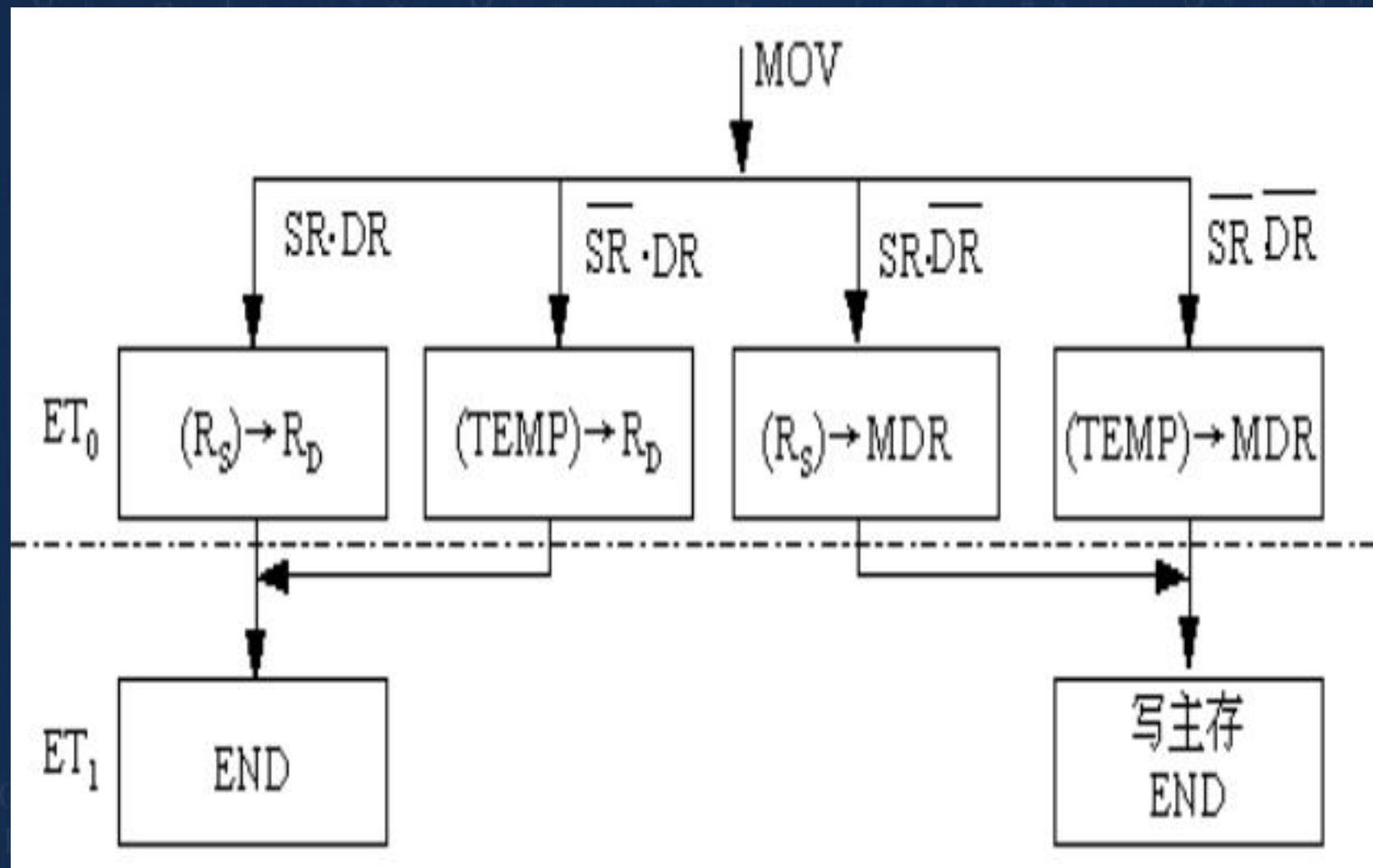
## 4) 执行周期(ET)的操作流程图

- 由于不同指令具有不同的操作功能，不同的寻址方式，操作数有不同的来源，因此不同指令在执行周期的操作流程也各不相同，需要按不同的指令绘制执行周期的操作流程。
- 对于**多数指令**，**执行周期的操作均可在两拍内完成**，所以多数指令的执行周期内只设两个节拍。
- 但对于JSR、RTS以及双操作数指令的目的寻址为非寄存器寻址（即 $DR=0$ ）时，因为需要访存，所以在执行周期内设四个节拍来完成。
- 在模型机的节拍发生器中，通过控制触发器 $C_2$ 的T输入端，可以选择执行周期的节拍数。



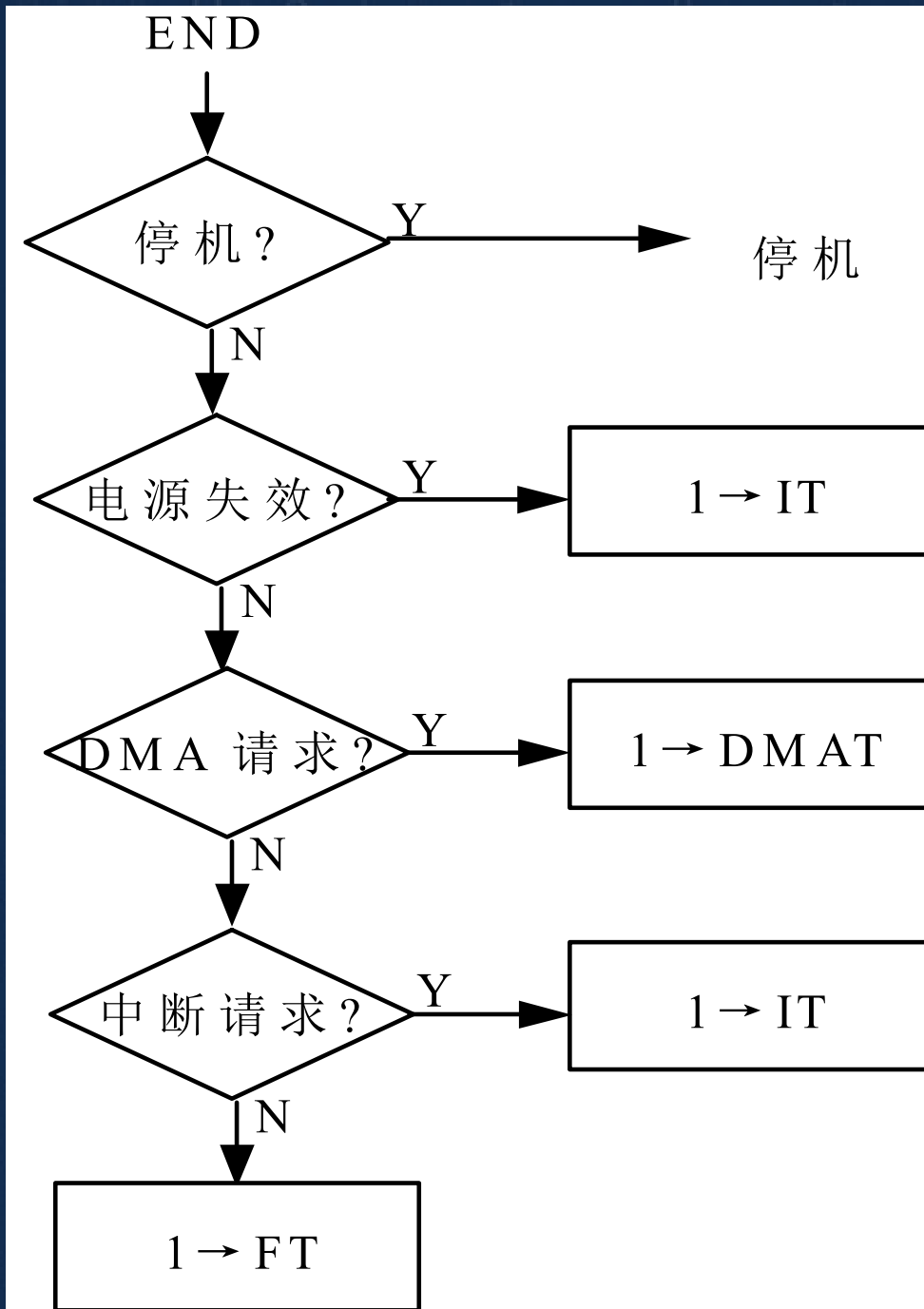
# (1) 传送指令的执行周期流程图

- 传送指令在执行周期中的功能是：将源操作数送入目的寄存器或存储单元。传送操作均可在两个节拍内完成。
- ①  $SR、DR=11$ 或 $01$ ，目的操作数为寄存器，不用访存。
- 执行： $(R_S) \rightarrow R_D$ ，或 $(TEMP) \rightarrow R_D$ 。
- ②  $SR、DR=10$ 或 $00$ ，目的操作数为存储器单元，需要访存。
- 执行： $(R_S) \rightarrow MDR$ ，或 $(TEMP) \rightarrow MDR$ ，写主存。



- 执行周期的结束，意味着一条指令的执行完毕。
- 在一条指令执行结束时，机器需自动检测有没有意外情况（如电源失效）和特殊请求（如中断请求）发生。若有，需进入相应周期进行处理；若没有，则机器又进入取指周期读取下条指令。
- 流程图中的**END**框即表示一条指令执行结束，进行结束后的检测。检测操作均在执行周期的最后一个节拍内完成。



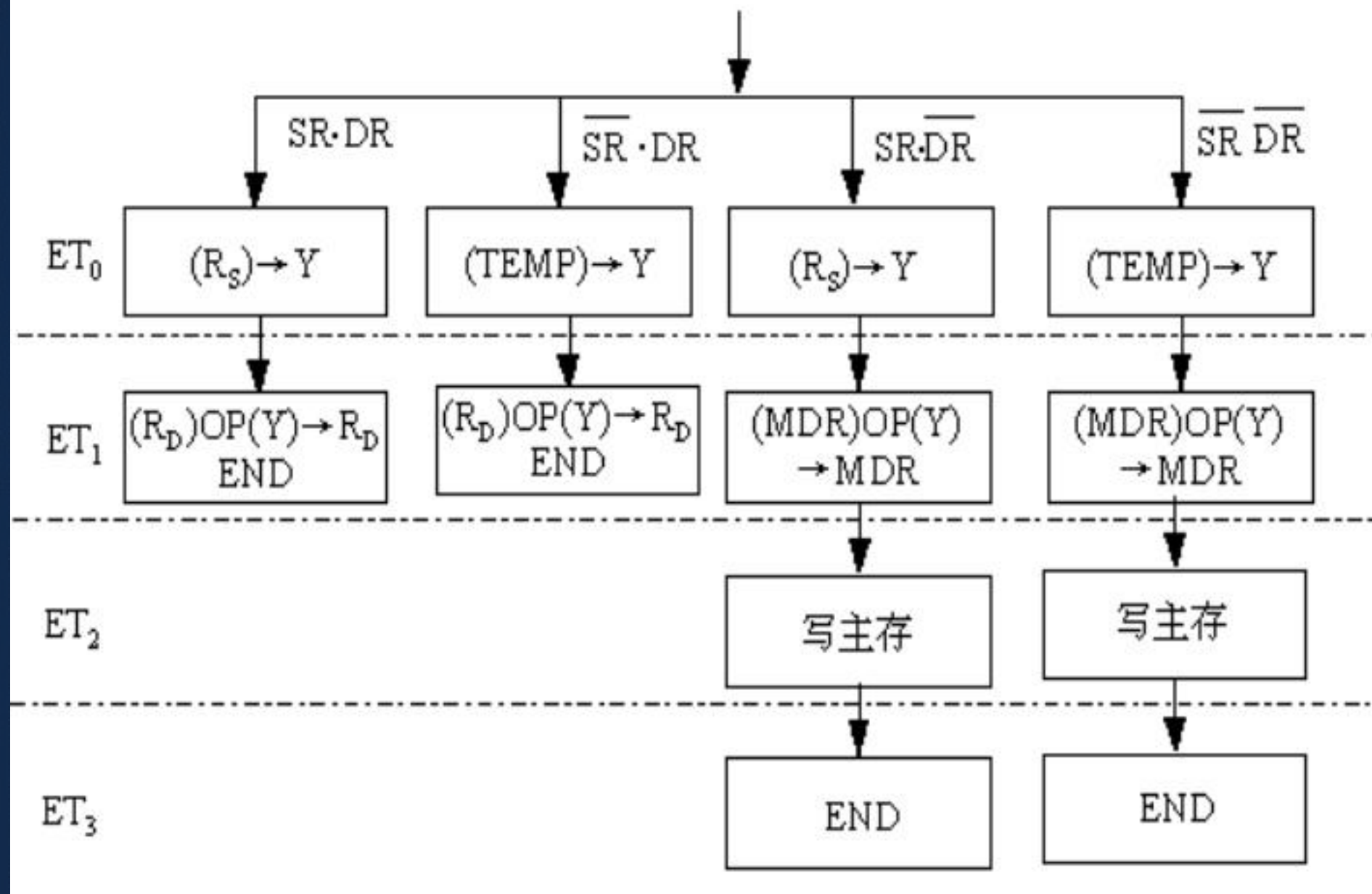


## (2) 双操作数指令的执行周期流程图

- 双操作数指令的执行周期的功能是：将源操作数与目的操作数进行指定的操作后送入目的寄存器或存储单元。
- ①  $SR、DR=11$ 或 $01$ ，目的操作数为寄存器，不用访存。
- 执行： $(R_S)$ 或 $(TEMP) \rightarrow Y$ ， $(R_D) OP (Y) \rightarrow R_D$ 。可在两个节拍内完成。
- ②  $SR、DR=10$ 或 $00$ ，目的操作数为存储器单元，需要访存。
- 执行： $(R_S)$ 或 $(TEMP) \rightarrow Y$ ， $(MDR) OP (Y) \rightarrow MDR$ ，写主存。需要在四个节拍内完成。



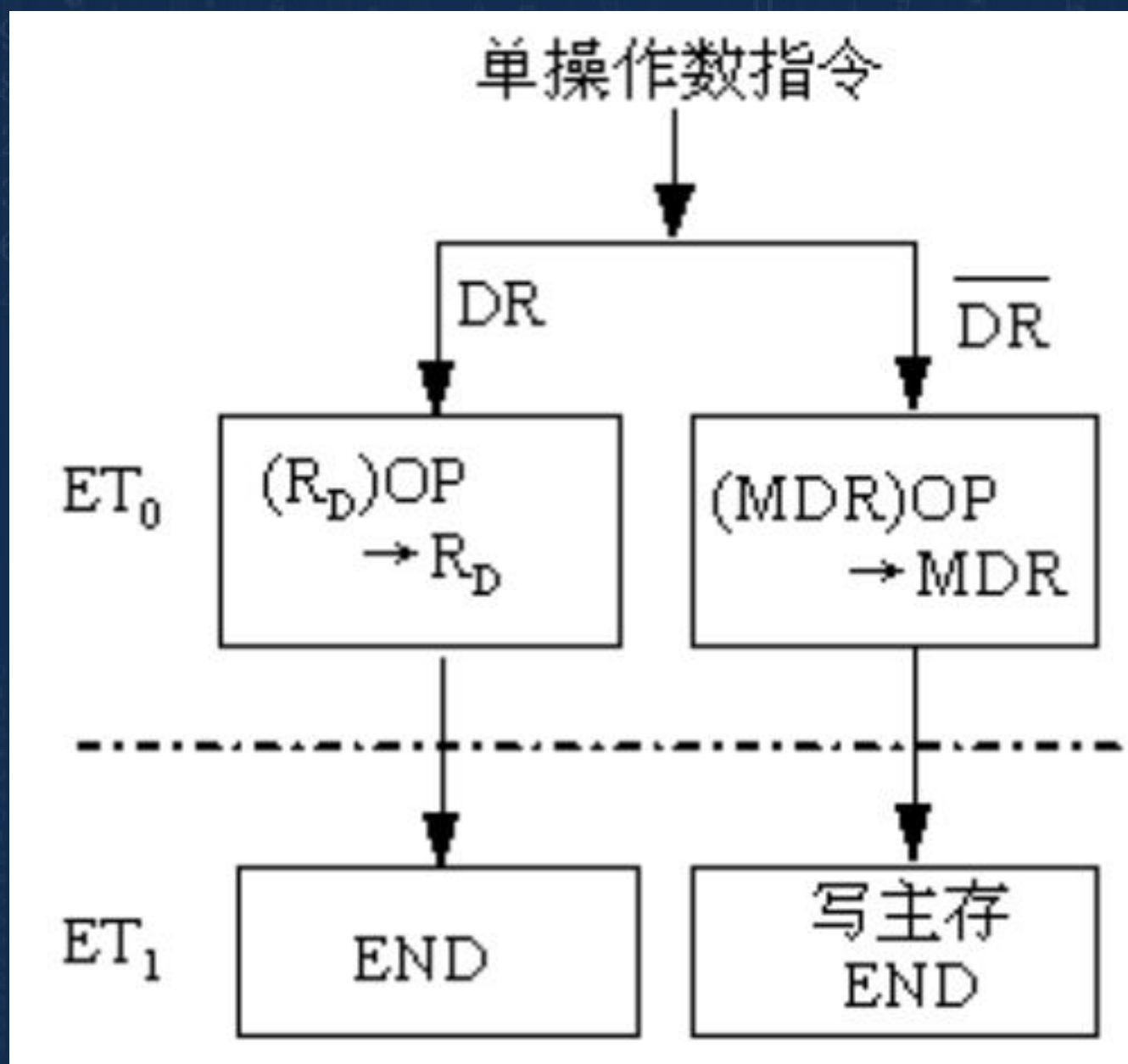
## 双操作数指令





### (3) 单操作数指令的执行周期流程图

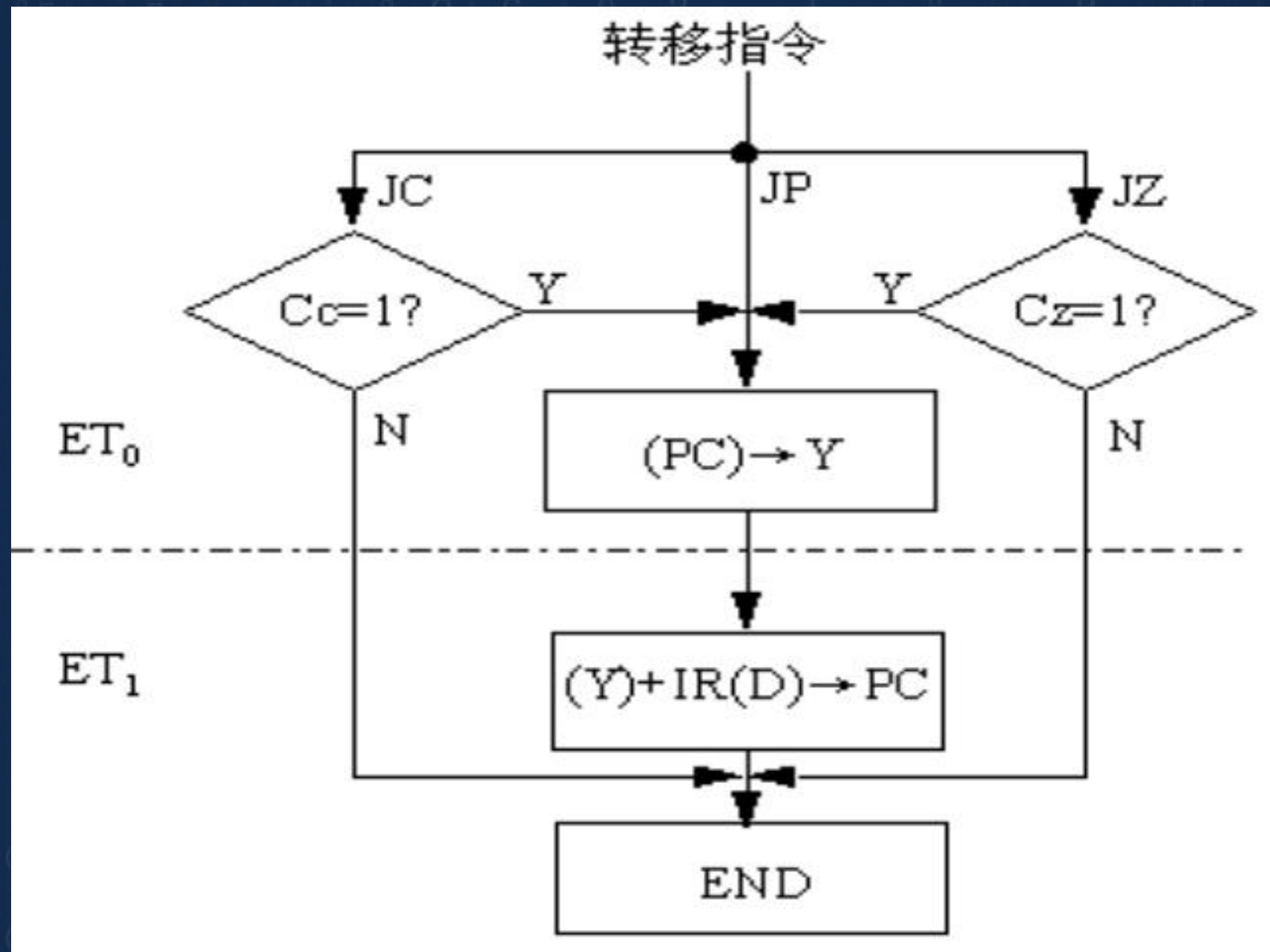
- 单操作数指令的执行周期的功能是：将目的操作数进行指定的操作后送入目的寄存器或存储单元。操作均可在两个节拍内完成。
- ①  $DR=1$ ，目的操作数为寄存器，不用访存。
- 执行： $(R_D)OP \rightarrow R_D$ 。
- ②  $DR=0$ ，目的操作数为存储器单元，需要访存。
- 执行： $(MDR)OP \rightarrow MDR$ ,写主存。



## (4) 转移指令的执行周期流程图

- 模型机的转移指令均采用相对寻址。非转子指令的执行周期的功能是：将PC中的内容与指令中给出的位移量相加得到转移目的地址送入PC，实现转移。操作均可在两个节拍内完成。
- JP:  $PC \rightarrow Y, (Y) + IR(D) \rightarrow PC$
- JC、JZ: 先判断 $C_C$ 或 $C_Z$ ，若为1，满足转移条件，
- 则执行 $(Y) + IR(D) \rightarrow PC$ ，否则指令执行结束。

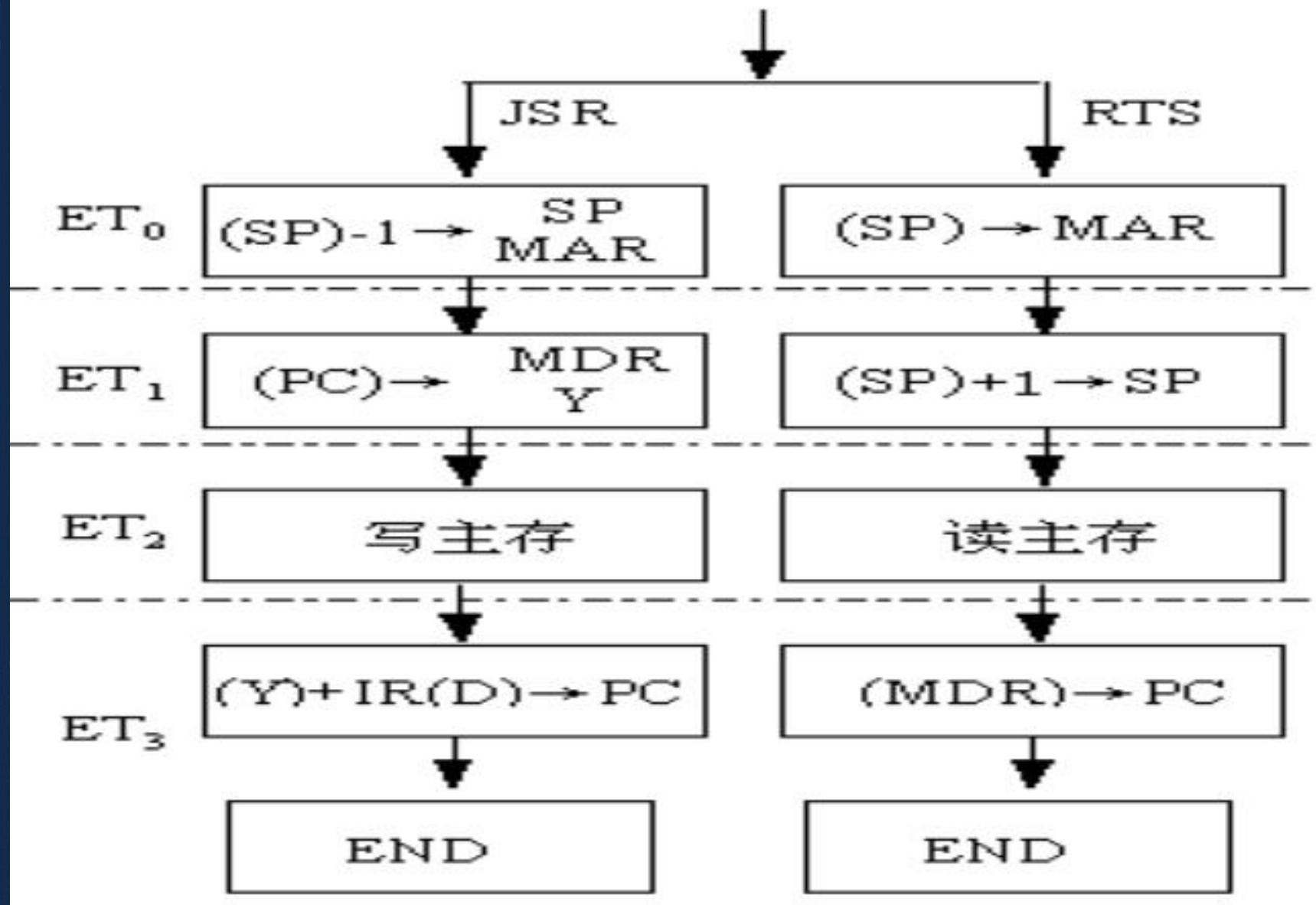




## (5) 转子和返回指令的执行周期流程图

- 转子(JSR)和返回(RTS)指令在执行周期中不仅要改变PC的内容，而且需要进行断点保护或恢复断点的工作。需要在四个节拍内完成。
- JSR:  $(SP) - 1 \rightarrow SP$  和  $MAR$ ,  $PC \rightarrow MDR$  和  $Y$ , 写主存,  $(Y) + IR(D) \rightarrow PC$
- RTS:  $(SP) \rightarrow MAR$ ,  $(SP) + 1 \rightarrow SP$ , 读主存,  $(MDR) \rightarrow PC$

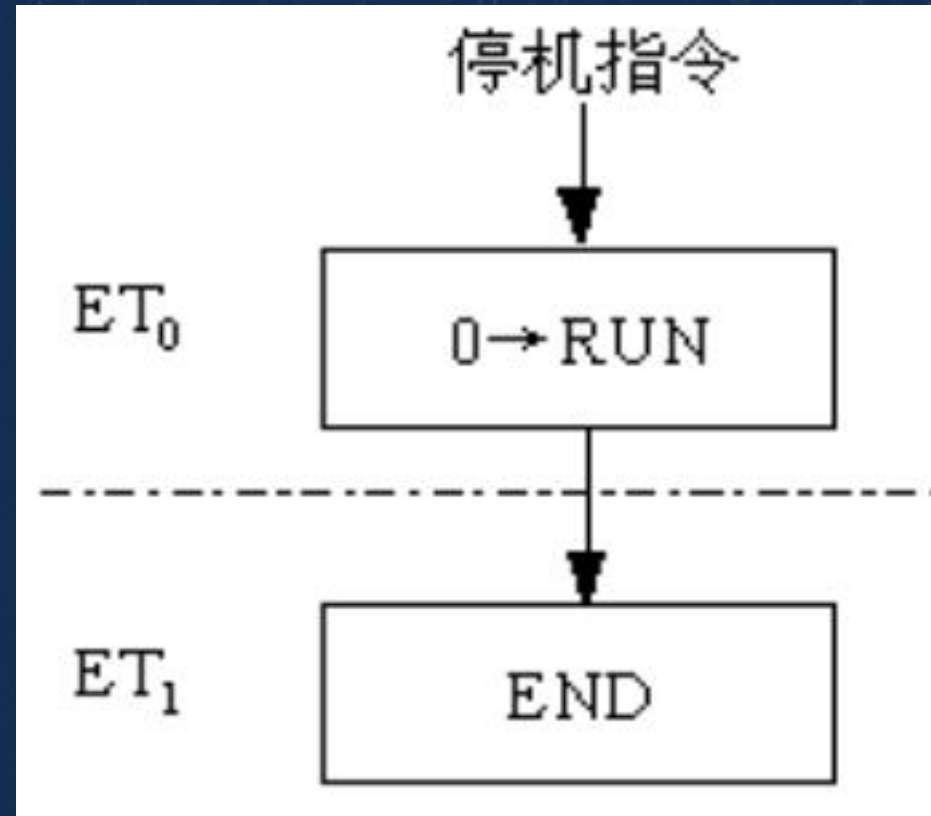
## 转子/返回指令





## (6) 停机指令的执行周期流程图

- 停机指令（HALT）的执行周期的功能是使机器暂停执行。
- 执行： $0 \rightarrow \text{RUN}$



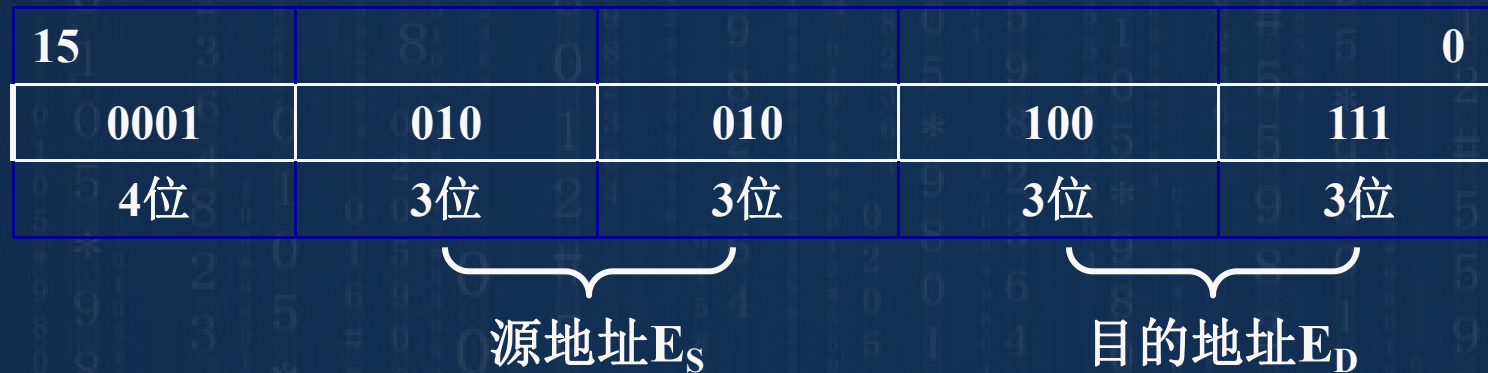
- 模型机严格按照周期、节拍工作，根据各指令所需的周期数和各周期所需的节拍数，可以计算每条指令所需的节拍数。
- 例：计算指令 **ADD R<sub>0</sub>, R<sub>1</sub>**所需的周期和节拍数。
- 解：∵ R<sub>0</sub>, R<sub>1</sub>采用寄存器寻址，∴需要取指和执行两个周期。
- ∵ FT周期需4个节拍，ET周期需2个节拍
- ∴ 共需6个节拍

- 例：计算指令  $\text{ADD } X_1(R_0), X_2(R_1)$  所需的周期和节拍数。

周期	取指	取源 (变址, 重复)	取目的 (变址, 重复)	执行 (双操作数, 需写主存)	共6个周期
节拍	4	4, 4	4, 2	4	共22个节拍

例：模型机指令ADD (R<sub>2</sub>)<sup>+</sup>，X(PC)的执行过程。

- 设(PC)=0081H，X=0004H，(R<sub>2</sub>)= 0086H，
- (0086H)=03FFH，(0087H)=FC00H
- 解：指令ADD (R<sub>2</sub>)<sup>+</sup>，X(PC)的机器码为14A7H。





- 执行过程







- 2. 指令操作时间表
- 见 表4-15到表4-23

### 3. 微操作控制信号综合

- 指令操作时间表根据指令流程、时序系统和数据通路结构，列出了实现整个指令系统的全部操作控制信号。将同一种控制信号在不同时间下产生的条件进行归纳、综合、化简后，就可写出该控制信号的综合逻辑表达式。
- 逻辑表达式中包括下列因素：
- 微操作控制信号 =  $F(\text{周期、节拍、脉冲、指令、状态条件})$



- 例：读信号（R/W）的逻辑表达式

$$R/\overline{W} = FT \cdot T_1 + ST \cdot T_1 + DT \cdot T_1 (\overline{MOV} + X_D) + ET \cdot T_2 \cdot RTS$$

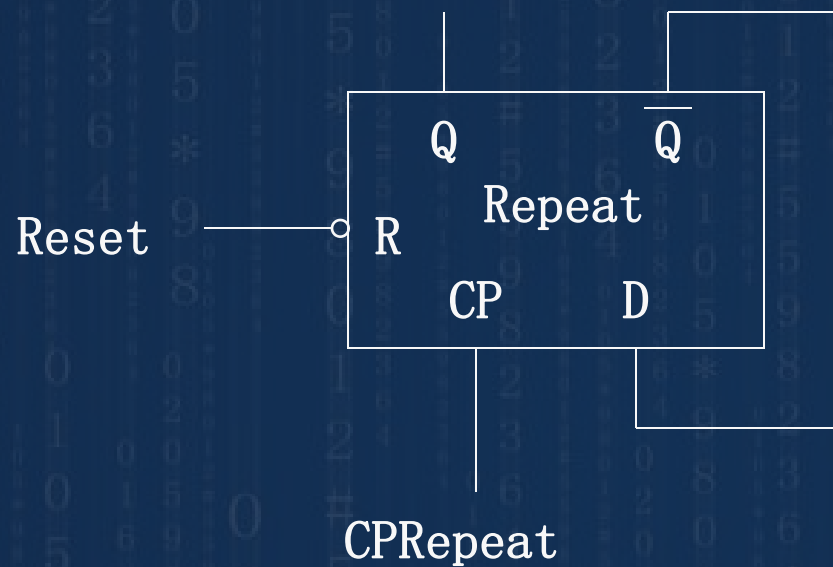
- 例：脉冲信号CPTEMP的逻辑表达式

$$CPTEMP = ST \cdot T_2 \cdot (\overline{X_S} + Repeat) \cdot P$$

- 例：脉冲信号PC→BUS<sub>1</sub>的逻辑表达式

$$\begin{aligned} PC \rightarrow BUS_1 = & FT \cdot (T_0 + T_1) + ST \cdot T_0 \cdot (\overline{IR_8 IR_7 IR_6} + X_S \cdot \overline{Repeat}) + ST \cdot T_1 \\ & \cdot (\overline{IR_8 IR_7 IR_6} \cdot (R_S) + X_S \cdot \overline{Repeat}) + DT \cdot T_0 \cdot (\overline{IR_2 IR_1 IR_0} + X_D \\ & \cdot \overline{Repeat}) + DT \cdot T_1 \cdot (\overline{IR_2 IR_1 IR_0} \cdot (R_D) + X_D \cdot \overline{Repeat}) + DT \cdot \\ & T_3 \cdot \overline{IR_2 IR_1 IR_0} \cdot MOV + ET \cdot T_0 \cdot \overline{IR_8 IR_7 IR_6} \cdot SR \cdot MOV + ET \cdot \\ & T_0 \cdot (JP + JC \cdot Cc + JZ \cdot Cz) + ET \cdot T_0 \cdot \overline{IR_8 IR_7 IR_6} \cdot SR \cdot (ADD + \\ & SUB + AND + OR + EOR) + ET \cdot T_1 \cdot JSR \end{aligned}$$

- 可用一个触发器产生重复周期信号Repeat

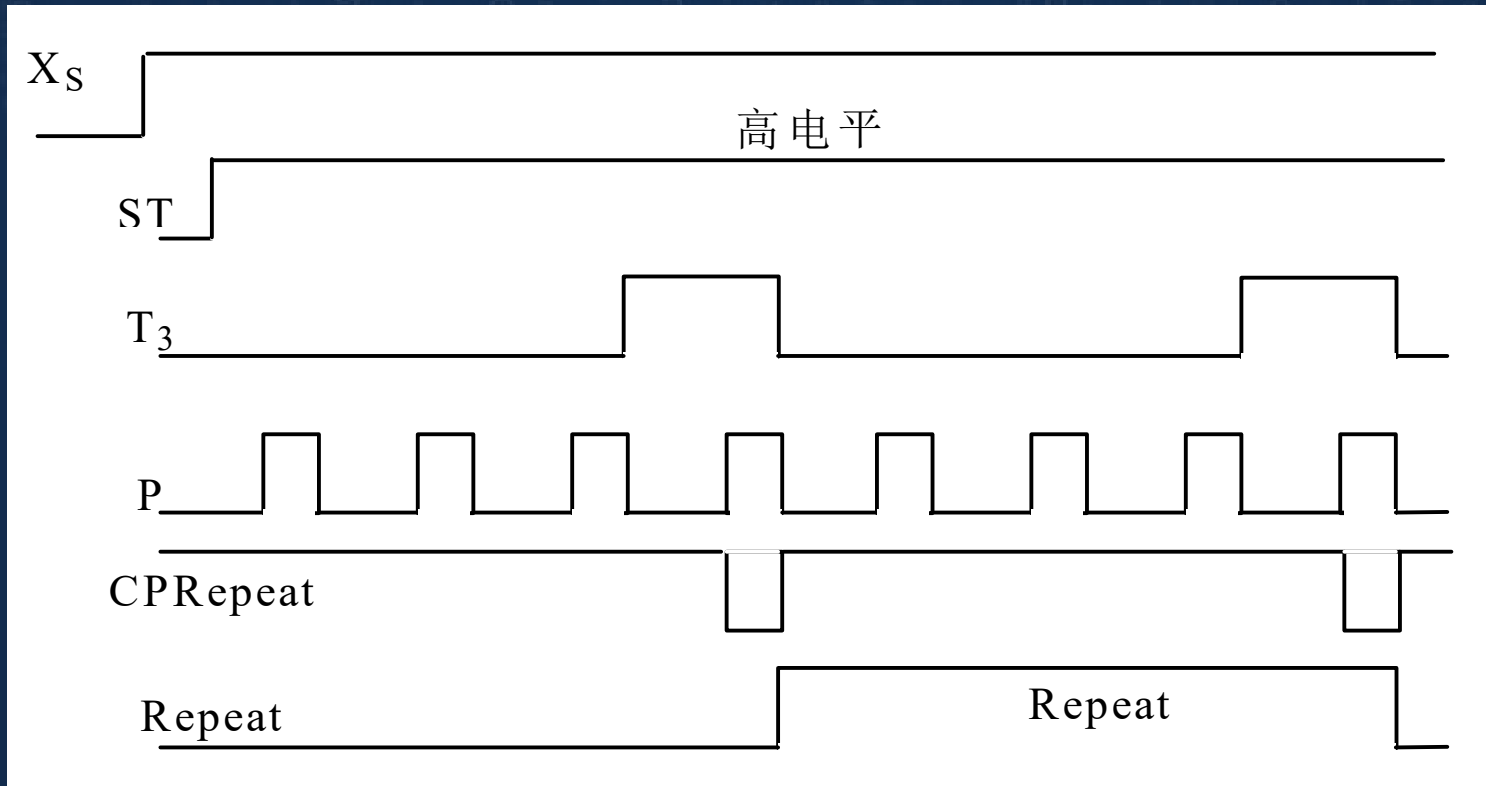


- **Reset**信号使Repeat=0。
- 进入ST或DT后，若需要重复周期，就使CPRepeat有效，**Repeat=1**。

- 在重复周期还需要CP脉冲，使Repeat翻转为0。
- CPRepeat的逻辑表达式

$$\text{CPRepeat} = (\text{ST} \cdot \text{T}_3 \cdot \text{X}_s + \text{DT} \cdot \text{T}_3 \cdot \overline{\text{MOV}} \cdot \text{X}_s + \text{DT} \cdot \text{T}_1 \cdot \text{Repeat})(\overline{\text{P}})$$

- Repeat信号与其他信号的时间关系



## 4. 控制信号形成电路

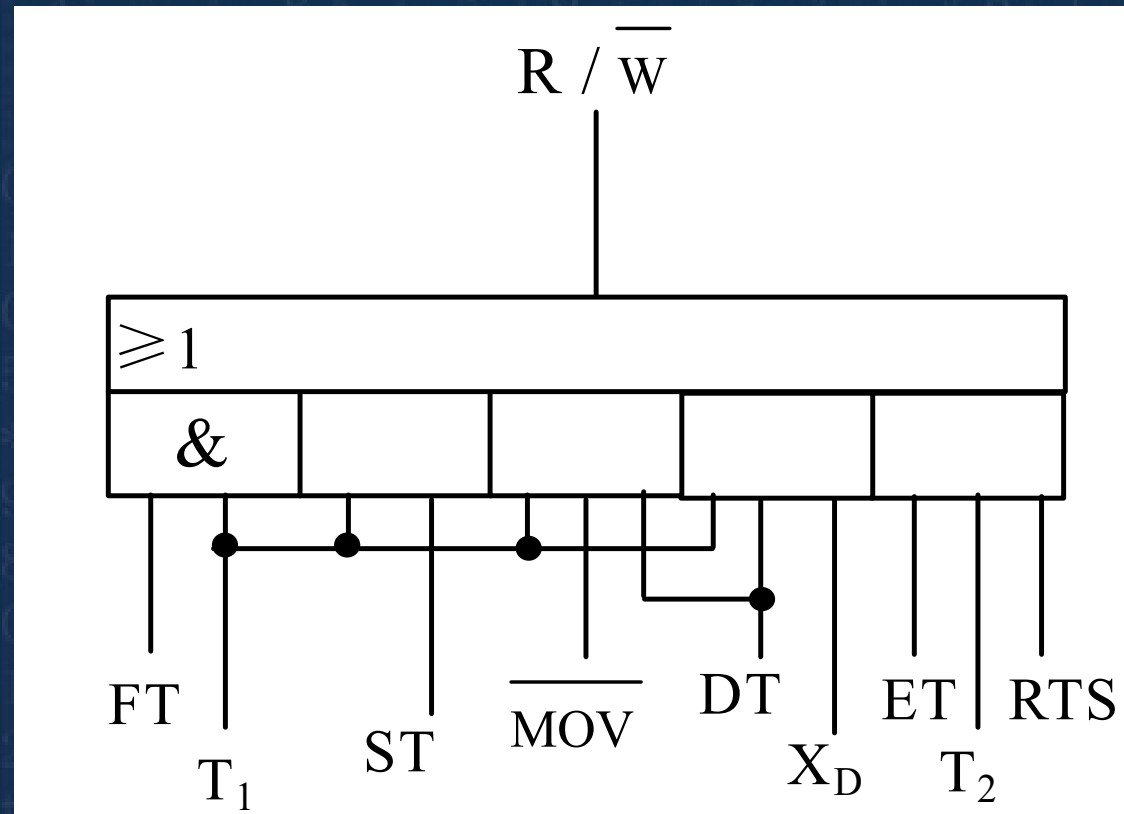
- 将逻辑表达式经逻辑综合、化简得到各个微操作控制信号的比较合理的表达式后，即可设计实现电路。
- 1) 用硬联逻辑电路实现
- 将各微操作控制信号用硬联辑电路实现，就构成了硬联逻辑控制器的微操作控制信号形成部件。
- 在实际设计过程中，用硬件实现逻辑电路图时，有时受逻辑门的扇入系数限制，需要修改逻辑表达式，此时就可能要增加逻辑电路。如果信号所经过的级数也增加的话，还将增加延迟时间。另外在实现时还有负载问题。



—

- 例：读信号（R/W）的逻辑电路图

$$R/\overline{W} = FT \cdot T_1 + ST \cdot T_1 + DT \cdot T_1 (\overline{MOV} + X_D) + ET \cdot T_2 \cdot RTS$$



## 2) 用PLA器件实现

- 可编程逻辑阵列PLA电路是由与阵列和或阵列组成，电路的输出为输入项的与或式。
- 经微操作综合而得到的微操作控制信号的逻辑表达式基本上都是与或表达式，因而可很方便地用PLA电路实现。

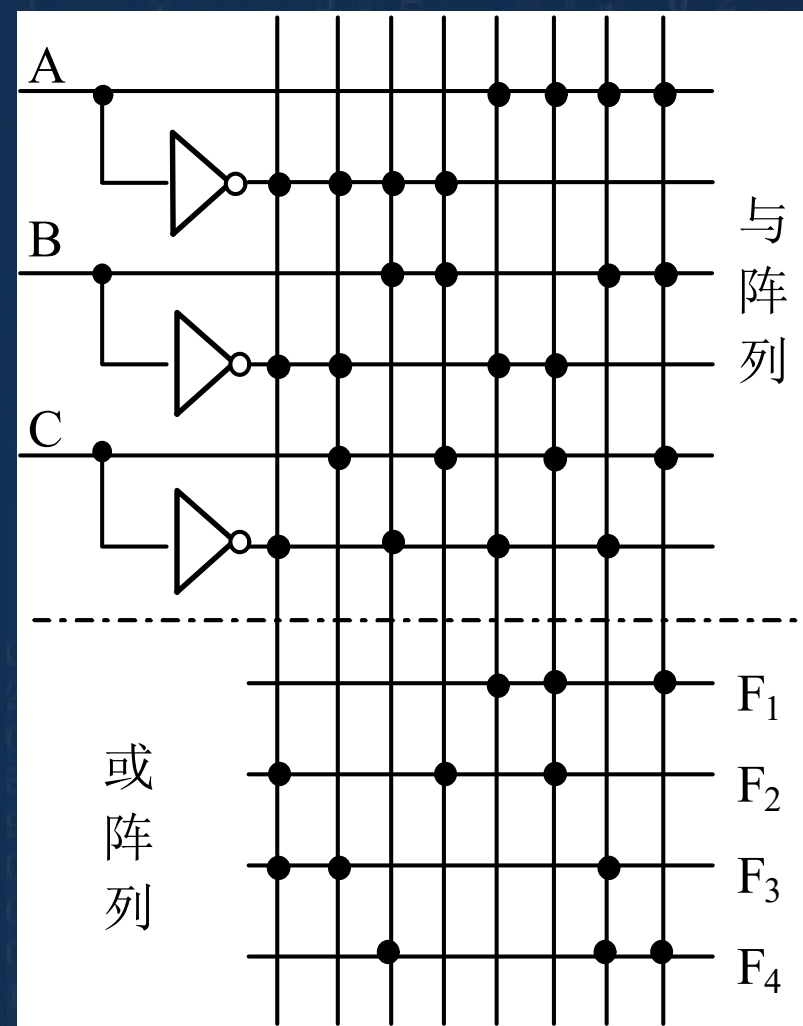
- 例：用PLA（可编程逻辑阵列）器件实现下列逻辑函数：

$$F_1 = A\bar{B} + AC$$

$$F_2 = \bar{A}BC + A\bar{B}C + \bar{A}\bar{B}\bar{C}$$

$$F_3 = \bar{A}\bar{B} + ABC$$

$$F_4 = \bar{A}B\bar{C} + AB$$



- 当用PLA器件实现模型机控制信号逻辑时，则将指令码、机器周期、节拍、脉冲及某状态条件作为PLA器件的与阵列输入信号，按微操作信号综合所得的逻辑表达式分别对与阵列、或阵列进行编程，即可由或阵列输出各个控制信号。

