



计算机组成原理

第三章 中央处理器

2025-8-22



信息与软件工程学院
School of Information and Software Engineering



主要内容

- 1 模型机的总体设计
- 2 算术逻辑运算部件
- 3 运算方法
- 4 模型机的组合逻辑控制器
- 5 模型机的微程序控制器
- 6 MIPS32架构CPU设计实例



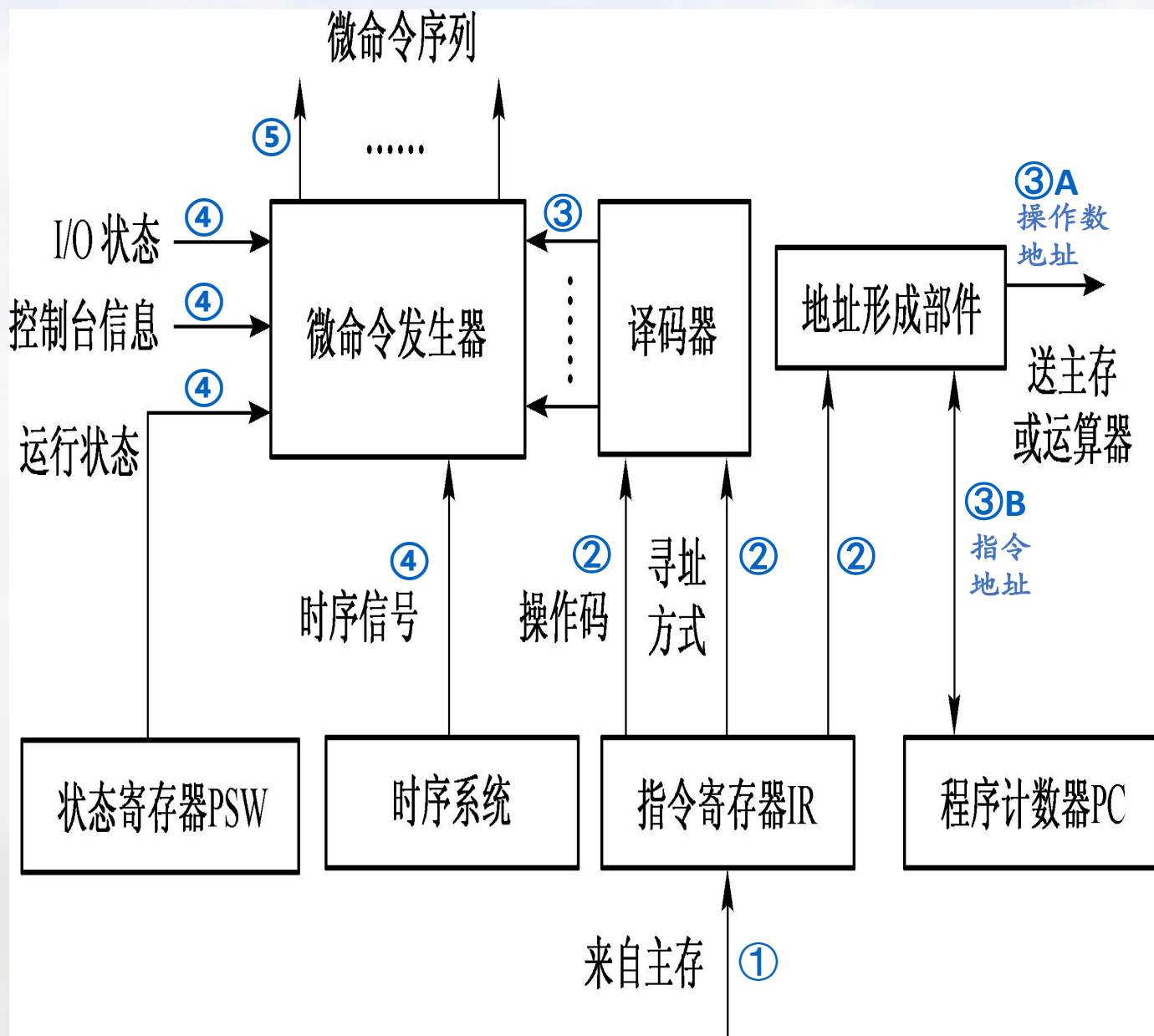
3.4 模型机的组合逻辑控制器

- 01. 组合逻辑控制器概述
- 02. 组合逻辑控制器时序系统
- 03. 指令流程与操作时间表
- 04. 组合逻辑控制方式的优缺点

1、定义：组合逻辑控制器的微命令是由组合逻辑电路来实现。
每种微命令都需要一组逻辑电路产生，全机所有微命令需要的逻辑电路就构成了微命令发生器。

2、硬件组成：

一、组合逻辑控制器概述



程序计数器PC
 指令寄存器IR
 状态寄存器PSW
 时序系统
 微命令发生器
 译码器
 地址形成部件

3、工作原理：

从主存读取的**现行指令**存放在**IR**中，其中，操作码与寻址方式代码分别经译码电路形成一些中间逻辑信号，送入**微命令发生器**，作为产生微命令的**基本逻辑依据**。

微命令的形成还需考虑各种**状态信息**，如PSW所反映的CPU内部运行状态、由控制台（如键盘）产生的操作员控制命令、I/O设备与接口的有关状态、外部请求等等。

一、组合逻辑控制器概述

微命令是**分时产生**的，所以还需引入时序系统提供的**周期、节拍、脉冲**等时序信号。

IR中的**地址段信息**送往地址形成部件，按照寻址方式码形成实际地址，或送主存以访问主存单元；或送往运算器，按指定的寄存器号选取相应的寄存器。

当程序**顺序执行**时，PC增量计数，形成后续指令的地址；
当程序需要转移时，IR中的地址段信息经地址形成部件产生转移地址，送入PC，使程序发生转移。

组合逻辑控制器依靠不同的**时间标志**，使CPU分步工作，**模型机**按常规采用**工作周期、时钟周期、工作脉冲**三级时序。

1、工作周期

模型机设置了六种**工作周期状态**，用六个周期状态触发器作为它们的标志。其中，四个工作周期（**取指、源、目的、执行**）用于**指令的正常执行**，两个工作周期（**中断、DMA**）用于**I/O传送控制**。

某一时期内**只有其中一个周期状态**触发器为**1**，指明CPU现在所处的工作周期状态，为该阶段的工作提供时间标志与依据。

二、组合逻辑控制器时序系统

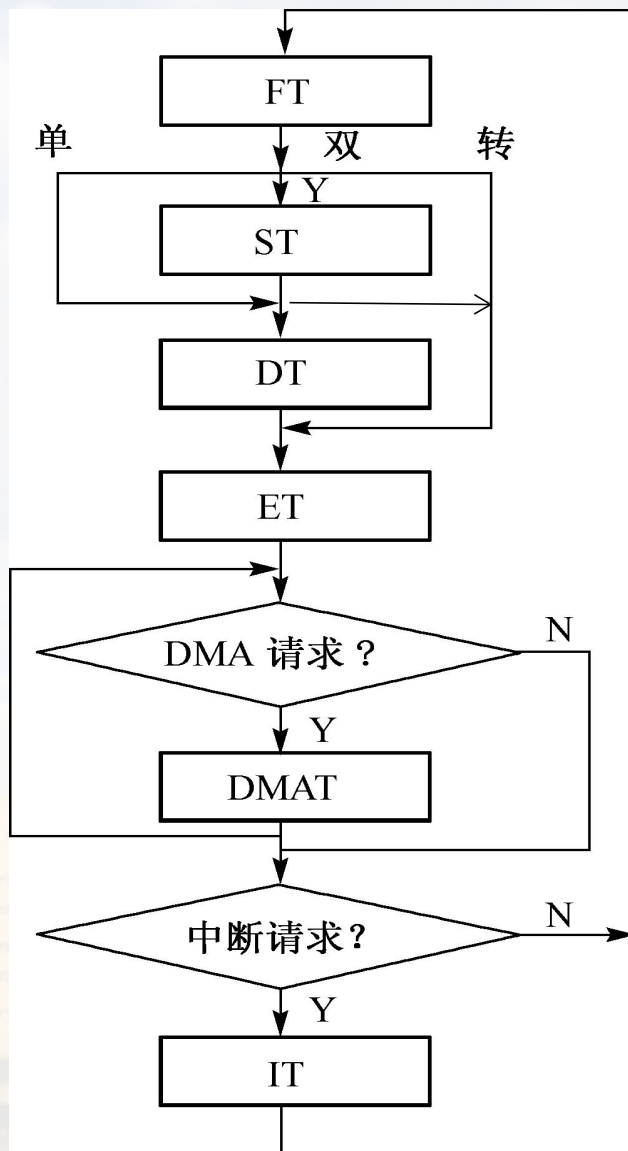


- ① 取指周期FT：在FT中完成的操作是公共性操作；
- ② 源周期ST——如果需要从主存中读取源操作数（非寄存器寻址），则进入ST。
- ③ 目的周期DT——如果需要从主存中读取目的地址或目的操作数（非寄存器寻址），则进入DT。
- ④ 执行周期ET——取得操作数后，CPU进入ET，这也是各类指令都需经历的最后一个工作阶段（公共性操作）。
- ⑤ 中断周期IT——除了考虑指令的正常执行，还需考虑外部请求带来的变化。在响应中断请求之后，到执行中断服务程序之前，需要一个过渡期，称为中断周期IT。

二、组合逻辑控制器时序系统

⑥ **DMA周期DMAT**——响应DMA请求之后，CPU进入DMAT。在DMAT中，**CPU交出系统总线的控制权**，即MAR、MDR与系统总线断开（呈高阻态），改由DMA控制器控制系统总线，实现主存与外围设备间的数据直传。

各工作周期状态之间的转换示意图：



二、组合逻辑控制器时序系统

2、时钟周期（节拍）

指令的读取与执行既有CPU内部数据通路操作，也包含访问主存的操作。为简化时序控制，模型机将两类操作周期统一起来，即以主存访问周期所需时间为时钟周期的宽度，这里设为1微秒。

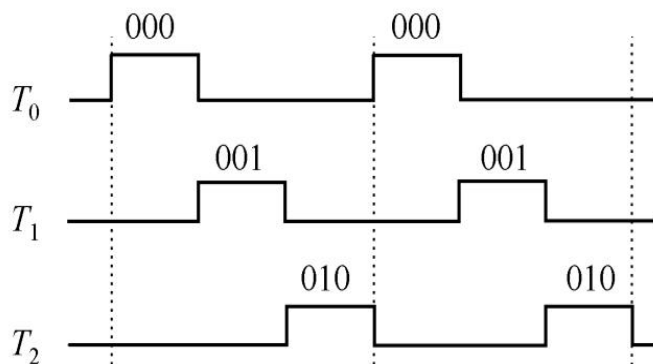
3、工作脉冲

时钟周期表示一个时间段，在这段时间内可以进行某种数据通路操作，如两数相加。但有些操作需要同步定时脉冲进行控制，如将稳定的运算结果打入寄存器，又如进行周期状态切换。模型机在每个时钟周期的末尾发一个工作脉冲P，作为各种同步脉冲的来源，如下页图所示。

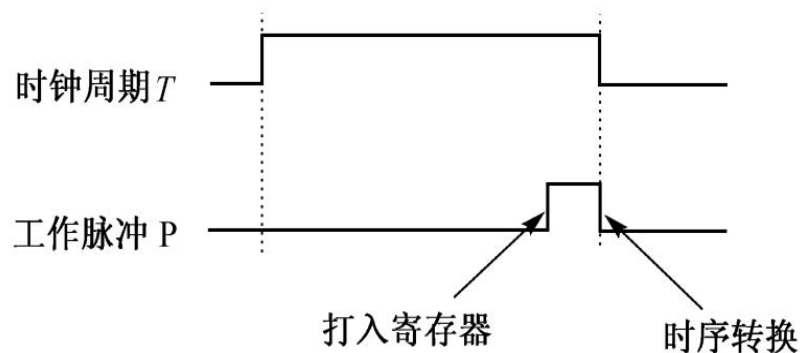
二、组合逻辑控制器时序系统



工作脉冲P的**前沿**作为**打入寄存器的定时**，它标志着一**次数据通路操作的完成**。P的**后沿**作为**时序转换的定时**，在此刻如果本工作周期未结束，则对时钟周期计数器T计数，进入新的节拍；如果本工作周期结束，则将T清零，并清除本工作周期状态标志，设置新的工作周期状态标志。



T 计数值译码产生时钟周期



时钟周期与工作脉冲的关系

通过有关讨论应能较深入地了解**CPU执行指令的工作机制**。

分**两个层次**进行讨论：

- ① 在**寄存器传送级**拟定**各类指令的执行流程**，也就是确定指令执行的具体步骤，即各类信息如何分步地按要求流动。
- ② 拟定**操作时间表**，即给出实现上述流程所需的微操作命令序列。其中包含维持一个时钟周期的电位型微命令，以及短暂的脉冲型微命令。操作时间表还将表明出现各种微命令的逻辑条件与时间条件。

通过这两种层次的工程化描述，我们就清晰地确定了CPU在什么时间、根据什么条件、发出什么命令、做什么事。

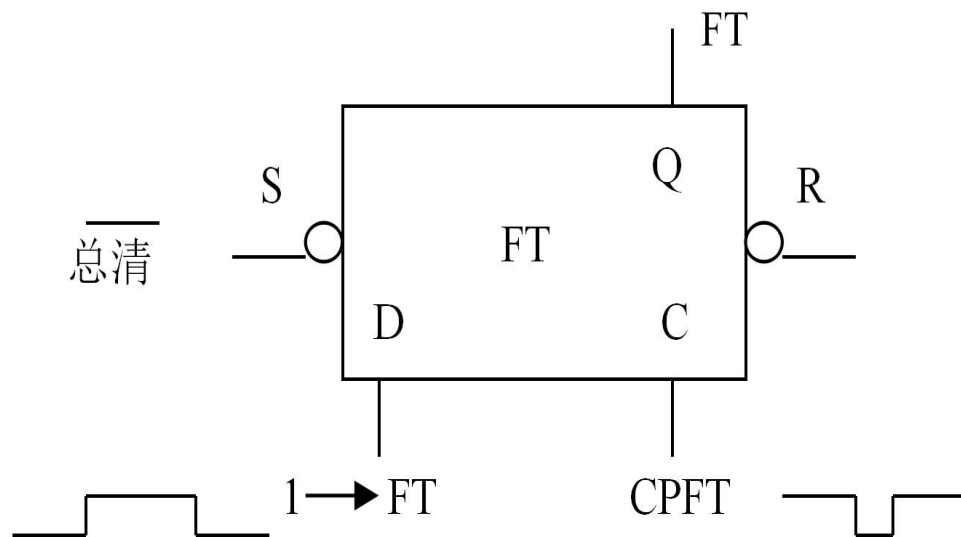
这里是**以指令为线索**，按指令类型分别拟定操作流程。这种方法的优点是对指令的执行过程拟出了清晰的线索，便于理解CPU的工作过程。

1、取指周期FT: Fetch Time

(1) 进入FT的方式和条件

① 由R/S端异步置入:

- a. 上电初始化
- b. 复位初始化



② 运算过程中同步打入FT(由D端打入) :

- a. 程序正常执行, 应进入FT;
- b. DMA周期执行后, 应进入FT;
- c. IT周期执行后, 应进入FT.

三、指令流程与操作时间表

(2) 取指流程: $M \rightarrow IR$, $PC+1 \rightarrow PC$ (参看第三章CPU (一) PPT34)

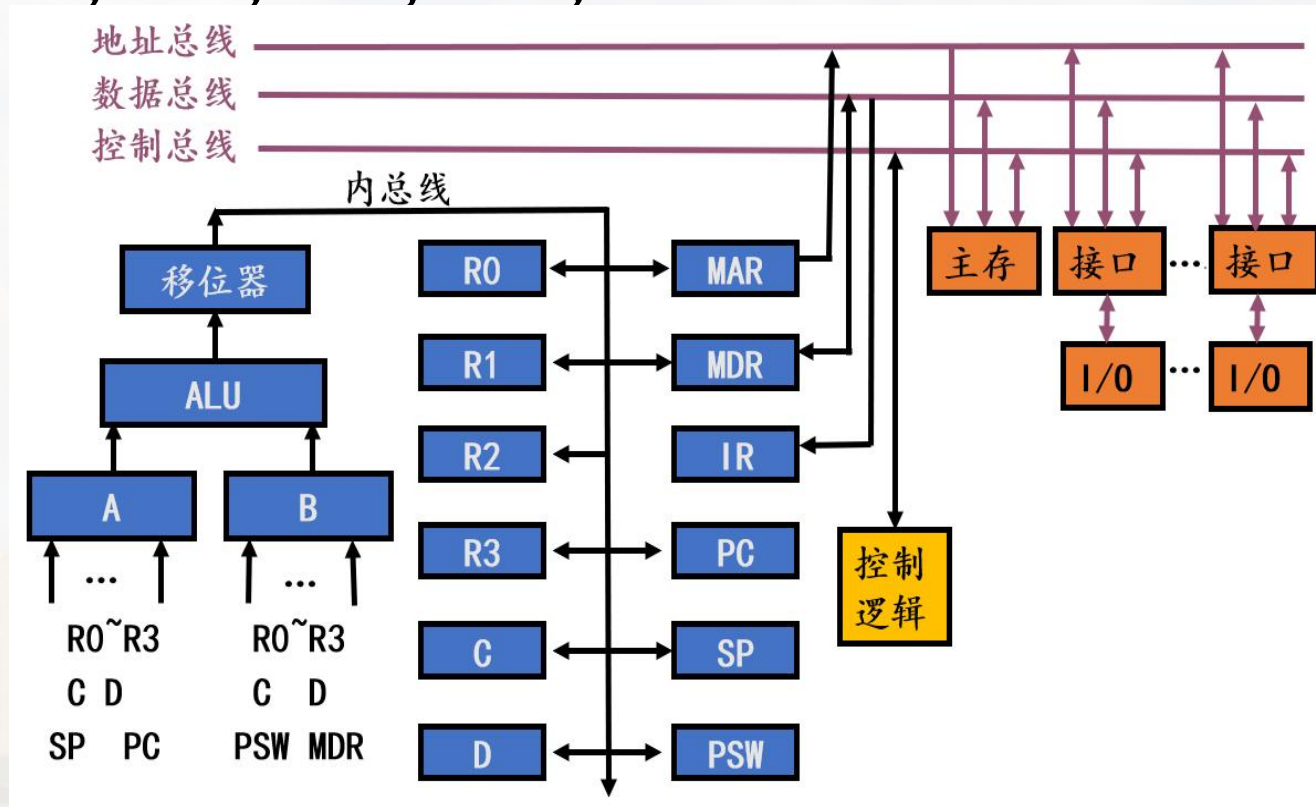
(3) 操作时间表

EMAR, R, SIR; $PC \rightarrow A$, $A+1$, DM, CPPC,

① $1 \rightarrow ST$, CPST;

② $1 \rightarrow DT$, CPDT;

③ $1 \rightarrow ET$, CPET;



三、指令流程与操作时间表

2、ST:寻址方式(MOV、双操作数指令) (参看第三章CPU (一))

PPT相关内容)

① (R) 型:

时序信号	指令流程	电平 (微命令)	脉冲 (微命令)
ST_0	$R_i \rightarrow MAR$	$R_i \rightarrow A/B$	
		A/B	
		DM	CPMAR
		T+1	CPT
ST_1	$M \rightarrow MDR \rightarrow C$	EMAR	
		R	
		SMDR	
		$MDR \rightarrow B$	
		B	
		DM	CPC
		$1 \rightarrow DT?$	CPDT?
		$1 \rightarrow ET?$	CPET?

三、指令流程与操作时间表

② - (R) 型：

时序信号	指令流程	电平（微命令）	脉冲（微命令）
ST ₀	$R_{i-1} \rightarrow R_i, MAR$	$R_i \rightarrow A/B$	
		A-1	
		DM	CPMAR、CPR _i
		T+1	CPT
ST ₁	$M \rightarrow MDR \rightarrow C$	EMAR	
		R	
		SMDR	
		$MDR \rightarrow B$	
		B	
		DM	CPC
		$1 \rightarrow DT?$	CPDT?
		$1 \rightarrow ET?$	CPET?

三、指令流程与操作时间表

③ I / (R) +:

时序信号	指令流程	电平 (微命令)	脉冲 (微命令)
ST₀	$R_i \rightarrow \text{MAR}$	$R_i \rightarrow A/B$	
		A/B	
		DM	CPMAR
		T+1	CPT
ST₁	$M \rightarrow \text{MDR} \rightarrow C$	EMAR	
		R	
		SMDR	
		$\text{MDR} \rightarrow B$	
		B	
		DM	CPC
		T+1	CPT
ST₂	$R_i + 1 \rightarrow R_i$	$R_i \rightarrow A/B$	
		A+1	
		DM	CPR_i
		$1 \rightarrow \text{DT?}$	CPDT?
		$1 \rightarrow \text{ET?}$	CPET?

三、指令流程与操作时间表

④ @ (R) +:

时序信号	指令流程	电平（微命令）	脉冲（微命令）
ST ₀	R _i → MAR	R _i → A/B	
		A/B	
		DM	CPMAR
		T+1	CPT
ST ₁	M → MDR → C	EMAR	
		R	
		SMDR	
		MDR → B	
		B	
		DM	CPC
		T+1	CPT



三、指令流程与操作时间表

④ @ (R) +:

时序信号	指令流程	电平（微命令）	脉冲（微命令）
ST ₂	R _i +1→R _i	R _i →A/B	
		A+1	
		DM	CPR _i
		T+1	CPT
ST ₃	C→MAR	C→A/B	
		A/B	
		DM	CPMAR
		T+1	CDT
ST ₄	M→MDR→C	EMAR	
		R	
		SMDR	
		MDR→B	
		B	
		DM	CPC
		1→DT?	CPDT?
		1→DT?	CPDT?

三、指令流程与操作时间表

⑤ X(R) :

时序信号	指令流程	电平 (微命令)	脉冲 (微命令)
ST ₀	PC→MAR	PC→A	
		A	
		DM	CPMAR
		T+1	CPT
ST ₁	M→MDR→C	EMAR	
		R	
		SMDR	
		MDR→B	
		B	
		DM	CPC
		T+1	CPT

三、指令流程与操作时间表

⑤ X(R) :

时序信号	指令流程	电平（微命令）	脉冲（微命令）
ST ₂	PC+1→PC	PC→A	
		A+1	
		DM	CPPC
		T+1	CPT
ST ₃	R _i +C→MAR	R _i →A/B、C→B/A	
		A+B	
		DM	CPMAR
		T+1	CPT
ST ₄	M→MDR→C	EMAR	
		R	
		SMDR	
		MBR→B	
		B	
		DM	CPC
		1→DT?	CPDT?
		1→ET?	CPET?

3、DT:寻址方式(MOV、双操作数指令)

与ST寻址方式相似

区别: MOV指令: 在DT中, 缺少取出目的操作数一步,
即 $M \rightarrow MDR \rightarrow D$ 。(为什么?)

双操作数指令: 在DT中, 需要取出目的操作数 (R寻址除外), 所以, 多一步操作, 即 $M \rightarrow MDR \rightarrow D$ 。

三、指令流程与操作时间表

4、ET

①ET周期分支：SR:Source Register

DR:Destination Register

	寻址方式
SR	源操作数采用R寻址，表明源操作数在CPU内的寄存器中
$\overline{\text{SR}}$	源操作数寻址采用(R)、-(R)、(R)+、@(R)+、X(R)中任意一种，表明源操作数在主存中
DR	目的操作数采用R寻址，表明目的操作数在CPU内的寄存器中
$\overline{\text{DR}}$	目的操作数寻址采用(R)、-(R)、(R)+、@(R)+、X(R)中任意一种，表明目的操作数在主存中

4、ET:②举例:

序号	类型	操作码	目的寻址方式	源寻址方式
1	SR · DR	MOV	R_0	R_1
		ADD	R_0	R_1
2	SR · \overline{DR}	MOV	$(R_0) +$	R_1
		ADD	$@(R_0) +$	R_1
3	\overline{SR} · DR	MOV	R_0	$X(R_1)$
		ADD	R_0	$-(R_1)$
4	\overline{SR} · \overline{DR}	MOV	$X(R_0)$	$@(R_1) +$
		ADD	(R_0)	$(R_1) -$

4、ET:②举例 ET周期分支一: SR · DR

指令: **MOV** R_0 , R_1

周期	指令流程
FT:	$M \rightarrow IR$, $PC+1 \rightarrow PC$
ST:	无
DT:	无
ET:	$R_1 \rightarrow R_0$
	$PC \rightarrow MAR$

指令: **ADD** R_0 , R_1

周期	指令流程
FT:	$M \rightarrow IR$, $PC+1 \rightarrow PC$
ST:	无
DT:	无
ET:	$R_1 + R_0 \rightarrow R_0$
	$PC \rightarrow MAR$

三、指令流程与操作时间表

4、ET:②举例 ET周期分支二: $SR \cdot \overline{DR}$

指令: **MOV** $(R_0)+, R_1$

周期	指令流程
FT:	$M \rightarrow IR, PC+1 \rightarrow PC$
ST:	无
DT:	$R_0 \rightarrow MAR$
	$R_0+1 \rightarrow R_0$
ET:	$R_1 \rightarrow MDR$
	$MDR \rightarrow M$
	$PC \rightarrow MAR$

指令: **ADD** $@(R_0)+, R_1$

周期	指令流程
FT:	$M \rightarrow IR, PC+1 \rightarrow PC$
ST:	无
DT:	$R_0 \rightarrow MAR$
	$M \rightarrow MDR \rightarrow D$
	$R_0+1 \rightarrow R_0$
	$D \rightarrow MAR$
	$M \rightarrow MDR \rightarrow D$
ET:	$R_1+D \rightarrow MDR$
	$MDR \rightarrow M$
	$PC \rightarrow MAR$

三、指令流程与操作时间表

4、ET:②举例 ET周期分支三: $\overline{SR} \cdot DR$

指令: $MOV\ R_0,\ X(R_1)$

周期	指令流程
FT:	$M \rightarrow IR,\ PC+1 \rightarrow PC$
ST:	$PC \rightarrow MAR$
	$M \rightarrow MDR \rightarrow C$
	$PC+1 \rightarrow PC$
	$R_1+C \rightarrow MAR$
	$M \rightarrow MDR \rightarrow C$
DT:	无
ET:	$C \rightarrow R_0$
	$PC \rightarrow MAR$

指令: $ADD\ R_0,\ -(R_1)$

周期	指令流程
FT:	$M \rightarrow IR,\ PC+1 \rightarrow PC$
ST:	$PC \rightarrow MAR$
	$R_1-1 \rightarrow MAR,\ R_1$
DT:	无
ET:	$R_0+C \rightarrow R_0$
	$PC \rightarrow MAR$

三、指令流程与操作时间表

4、ET:②举例 ET周期分支四: $\overline{SR} \cdot \overline{DR}$ (一)

指令: $MOV\ X(R0), @(R1) +$

周期	指令流程
FT:	$M \rightarrow IR, PC+1 \rightarrow PC$
ST:	$R_1 \rightarrow MAR$
	$M \rightarrow MDR \rightarrow C$
	$R_1+1 \rightarrow R_1$
	$C \rightarrow MAR$
	$M \rightarrow MDR \rightarrow C$
DT:	$PC \rightarrow MAR$
	$M \rightarrow MDR \rightarrow D$
	$PC+1 \rightarrow PC$
	$R_0+D \rightarrow MAR$

指令: $ADD\ (R0), (R1) -$

周期	指令流程
FT:	$M \rightarrow IR, PC+1 \rightarrow PC$
ST:	$R_1-1 \rightarrow MAR, R_1$
	$M \rightarrow MDR \rightarrow C$
DT:	$R_0 \rightarrow MAR$
	$M \rightarrow MDR \rightarrow D$
ET:	$C+D \rightarrow MDR$
	$MDR \rightarrow M$
	$PC \rightarrow MAR$

三、指令流程与操作时间表

4、ET: ②举例 ET周期分支四: $\overline{SR} \cdot \overline{DR}$ (续前)

指令: MOV X(R0), @(R1) +

周期	指令流程
ET:	C→MDR
	MDR→M
	PC→MAR

三、指令流程与操作时间表

4、ET:③ 四种分支（此处以MOV指令为例：指令流程及微命令序列）

第一种：SR·DR（DR→结果存放R中）

ET:

时序信号	指令流程	电平（微命令）	脉冲（微命令）
ET ₀	R _i →R _j	R _i →A/B	
		A/B	
		DM	CPR _j
		T+1	CPT
ET ₁	PC→MAR	PC→A	
		A	
		DM	CPMAR
		(1) 1→FT?	CPFT?
		(2) 1→DMAT?	CPDMAT?
		(3) 1→IT?	CPIT?

三、指令流程与操作时间表

4、ET:③ 四种分支

第二种: $\overline{SR} \cdot \overline{DR}$: ($\overline{DR} \rightarrow$ 结果存放M中)

ET:

时序信号	指令流程	电平 (微命令)	脉冲 (微命令)
ET_0	$R_i \rightarrow MDR$	$R_i \rightarrow A/B$	
		A/B	
		DM	CPMDR
		T+1	CPT
ET_1	$MDR \rightarrow M$	EMAR	
		W	
		T+1	CPT
ET_2	$PC \rightarrow MAR$	$PC \rightarrow A$	
		A	
		DM	CPMAR
		(1) $1 \rightarrow FT?$	CPFT?
		(2) $1 \rightarrow DMAT?$	CPDMAT?
		(3) $1 \rightarrow IT?$	CPIT?

三、指令流程与操作时间表

4、ET:③ 四种分支

第三种: $\overline{SR} \cdot DR$ ($DR \rightarrow$ 结果存放R中)

时序信号	指令流程	电平 (微命令)	脉冲 (微命令)
ET:	ET_0	$C \rightarrow R_j$	$C \rightarrow A/B$
		A/B	
		DM	CPR_j
		$T+1$	CPT
	ET_1	$PC \rightarrow MAR$	$PC \rightarrow A$
		A	
		DM	CPMAR
		(1) $1 \rightarrow FT?$	CPFT?
		(2) $1 \rightarrow DMAT?$	CPDMAT?
		(3) $1 \rightarrow IT?$	CPIT?

三、指令流程与操作时间表

4、ET:③ 四种分支

第四种: $\overline{SR} \cdot \overline{DR}$: ($\overline{DR} \rightarrow$ 结果存放M中)

ET:

时序信号	指令流程	电平 (微命令)	脉冲 (微命令)
ET ₀	C→MDR	C→A/B	
		A/B	
		DM	CPMDR
		T+1	CPT
ET ₁	MDR→M	EMAR	
		W	
		T+1	CPT
ET ₂	PC→MAR	PC→A	
		A	
		DM	CPMAR
		(1) 1→FT?	CPFT?
		(2) 1→DMAT?	CPDMAT?
		(3) 1→IT?	CPIT?

例题：

拟定指令“**ADD @(R_1)+, X(R_3)**”的执行流程及操作时间表的安排； $X(R_3)$ 表示源寻址， $@(R_1)+$ 表示目的寻址。

三、指令流程与操作时间表

FT:

时序信号	指令流程	电平（微命令）	脉冲（微命令）
	M→IR	EMAR	
		R	
		SIR	
	PC+1→PC	PC→A	
		A+1	
	时序转换：周期转换	DM	CPPC
		1→ST	CPST
ST: ST ₀	PC→MAR	PC→A	
		A	
	时序转换：节拍转换	DM	CPMAR
		T+1	CPT

三、指令流程与操作时间表

时序信号	指令流程	电平（微命令）	脉冲（微命令）
ST ₁	M→MDR→C	EMAR	
		R	
		SMDR	
		MDR→B	
		B	
	时序转换：节拍转换	DM	CPC
		T+1	CPT
ST ₂	PC+1→PC	PC→A	
		A+1	
		DM	CPPC
	时序转换：节拍转换	T+1	CPT
ST ₃	R ₃ +C→MAR	R ₃ →A/B、C→B/A	
		A+B	
		DM	CPMAR
	时序转换：节拍转换	T+1	CPT

三、指令流程与操作时间表

时序信号	指令流程	电平（微命令）	脉冲（微命令）
ST ₄	M→MDR→C	EMAR	
		R	
		SMDR	
		MDR→B	
		B	
	时序转换：周期转换	DM	CPC
		1→DT	CPDT
时序信号	指令流程	电平（微命令）	脉冲（微命令）
DT ₀	R ₁ →MAR	R ₁ →A/B	
		A/B	
		DM	CPMAR
		T+1	CPT
DT ₁	M→MDR→D	EMAR	
		R	
		SMDR	

DT:

三、指令流程与操作时间表

时序信号	指令流程	电平（微命令）	脉冲（微命令）
		MDR→B	
		B	
		DM	CPD
		T+1	CPT
DT ₂	R ₁ +1→R1	R ₁ →A/B	
		A+1/B+1	
		DM	CPR ₁
		T+1	CPT
DT ₃	D→MAR	D→A/B	
		A/B	
		DM	CPMAR
		T+1	CPT
DT ₄	M→MDR→D	EMAR	
		R	
		SMDR	

三、指令流程与操作时间表

ET:

时序信号	指令流程	电平（微命令）	脉冲（微命令）
		MDR→B	
		B	
		DM	CPD
		1→ET	CPET
时序信号	指令流程	电平（微命令）	脉冲（微命令）
ET ₀	C+D→MDR	C→A/B、D→B/A	
		A+B	
		DM	CPMDR
		T+1	CPT
ET ₁	MDR→M	EMAR	
		W	
		T+1	CPT
ET ₂	PC→MAR	PC→A	
		A	
		DM	CPMAR

5、单操作数指令

1) FT周期（略，与MOV指令FT周期相同）

2) 无ST周期

3) DT周期（略，与双操作数指令DT周期相同）

4) ET周期：两种分支

三、指令流程与操作时间表

第一种：DR (DR→结果存放R中)

ET:

时序信号	指令流程	电平 (微命令)	脉冲 (微命令)
ET ₀	OP R _i →R _i	R _i →A/B	
		OP A/B	
		DM	CPR _i
		T+1	CPT
ET ₁	PC→MAR	PC→A	
		A	
		DM	CPMAR
		(1) 1→FT?	CPFT?
		(2) 1→DMAT?	CPDMAT?
		(3) 1→IT?	CPIT?

三、指令流程与操作时间表

第二种： \overline{DR} ($\overline{DR} \rightarrow$ 结果存放M中)

ET:

时序信号	指令流程	电平（微命令）	脉冲（微命令）
ET_0	OP $D \rightarrow MDR$	$D \rightarrow A/B$	
		OP A/B	
		DM	CPMDR
		T+1	CPT
ET_1	$MDR \rightarrow M$	EMAR	
		W	
		T+1	CPT
ET_2	$PC \rightarrow MAR$	$PC \rightarrow A$	
		A	
		DM	CPMAR
		(1) $1 \rightarrow FT?$	CPFT?
		(2) $1 \rightarrow DMAT?$	CPDMAT?
		(3) $1 \rightarrow IT?$	CPIT?

6、转移指令JMP/返回指令RST

JMP指令（RST指令是JMP指令的一种特例）

1) FT周期（略，与MOV指令FT周期相同）

2) 无ST周期

3) 无DT周期（具体参考转移指令流程）

4) ET周期：

a) 转移不成功NJP（具体参考转移指令流程）

b) 转移成功JP（具体参考转移指令流程）

三、指令流程与操作时间表

7、转子指令JSR

分为转子不成功NJSR指令与转子成功JSR指令。

1) 转子不成功**NJSR指令**：只有FT及ET周期

2) 转子成功**JSR指令**：有FT周期、或ST周期（寻址方式决定）、ET周期。

注意：转子与转移指令的区别在于，转子指令在执行完成子程序后要返回主程序的第K+1条指令的执行，因此，在执行子程序前，在主程序的第K条指令的ET周期中要**保护断点、现场等操作**；在这点上，转子指令与后面的**中断IT周期**相似。

三、指令流程与操作时间表

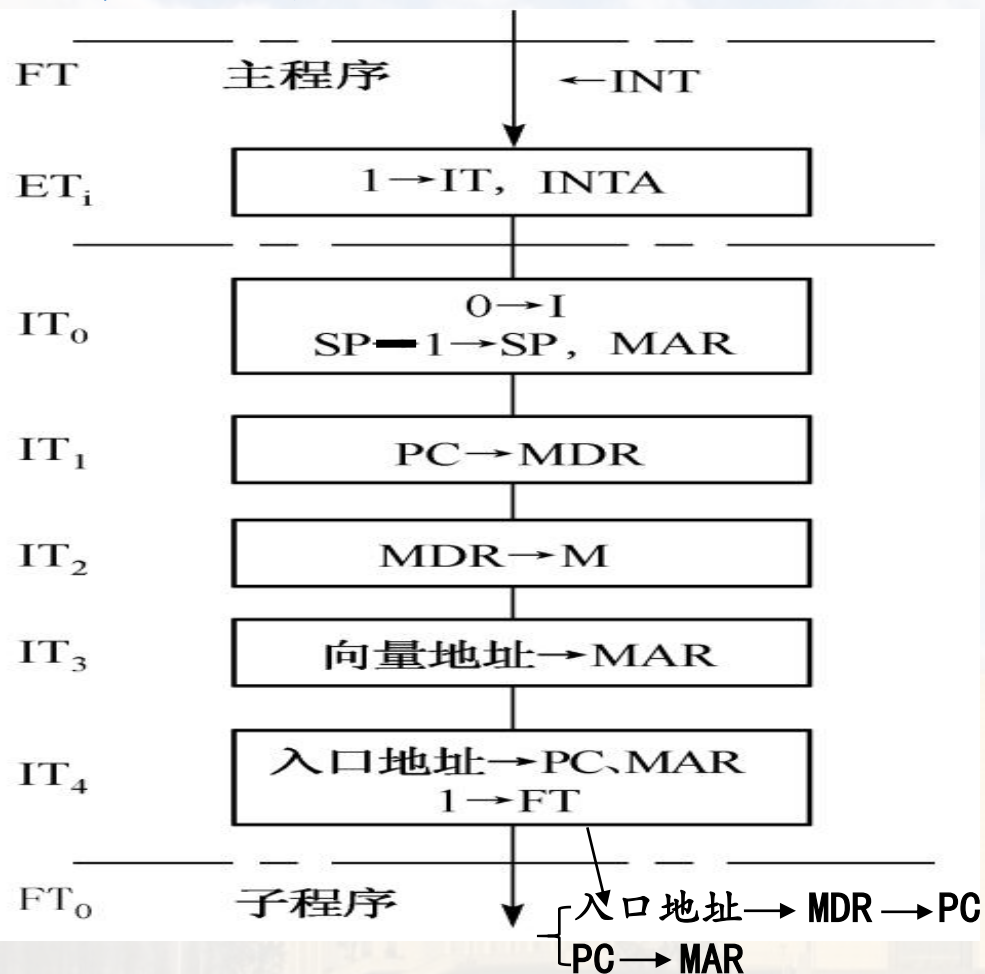
三种寻址方式：R、(R)、(R)+在ET周期中的保存断点操作

ET:

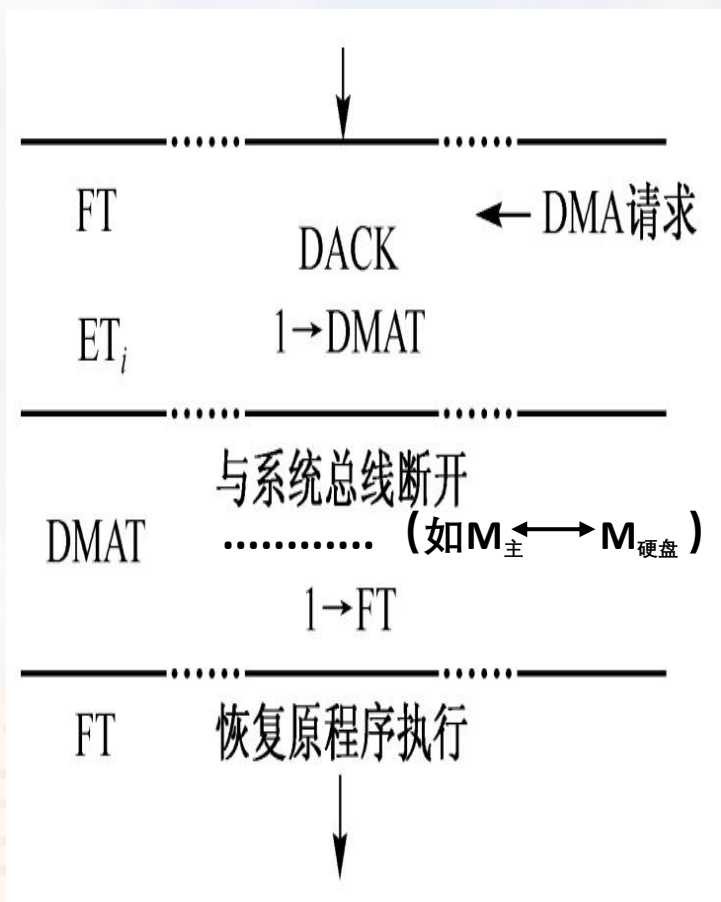
时序信号	指令流程	电平（微命令）	脉冲（微命令）
ET ₀	SP-1→SP、MAR	SP→A	
		A-1	
		DM	CPSP、CPMAR
		T+1	CPT
ET ₁	PC→MDR	PC→A	
		A	
		DM	CPMDR
		T+1	CPT
ET ₂	MDR→M	EMAR	
		W	
		T+1	CPT
.....

三、指令流程与操作时间表

8、中断周期IT



9、DMAT周期



四、组合逻辑控制方式的优缺点

缺点：

- ① **设计不规整。**组合逻辑控制方式是用许多门电路产生微命令的，而这些门电路所需的逻辑形态很不规整，因此组合逻辑控制器的核心部分比较繁琐、零乱，设计效率较低，检查调试也比较困难。
- ② **不易修改或扩展。**组合逻辑控制方式的另一缺点是不易修改或扩展指令功能。这是因为设计结果用印制电路板（硬连逻辑）固定下来以后，就很难再修改与扩展。



谢谢观看

计算机组成原理

2025-8-22

计算机组成原理--第三章 中央处理器



信息与软件工程学院
School of Information and Software Engineering