

CPU的工作原理

程晨闻

东南大学电气工程学院



- 微型计算机的定义
- 计算机的发展（摩尔定律）
- 微机的分类（通用和专用，嵌入式系统）
- 嵌入式系统的组成（硬件和软件）
- 嵌入式微处理器（MPU，MCU，DSP，SoC）
- 冯诺依曼体系结构（五大组成部分）
- 哈佛结构
- 系统总线（地址总线AB，数据总线DB，控制总线CB）
- 流水线技术
- C语言基础知识



- **CPU的组成**
 - 运算器
 - 控制器
 - 寄存器
- **CPU的功能**
- **指令集**
- **程序的工作流程**



- 计算机的工作过程就是计算机执行程序的过程
- 程序是一个指令序列，这个序列明确告诉计算机应该执行什么操作，在什么地方能够找到用来操作的数据
- 一旦把程序装入主存储器，计算机就可以自动执行取出指令和执行指令的任务
- 用来完成此项工作的计算机部件称为中央处理器（CPU）
- 指令是计算机硬件能够识别并直接执行操作的命令
 - 加、减、乘、除、开方、函数、传送



三地址指令	OP	D1	D2	D3
二地址指令	OP	目的	源	
一地址指令	OP	D	单操作数指令和隐含操作数指令	
零地址指令	OP	无操作数指令和隐含操作数指令		

- 一台计算机中所有指令的集合构成了该计算机的**指令系统**，也称为**指令集**

	复杂指令系统计算机 (Complex Instruction Set Computer, CISC)	精简指令系统计算机 (Reduced Instruction Set Computer, RISC)
设计初衷	存储器贵 ，希望指令系统生成 最短指令	降低 由 硬件 执行指令的复杂度
指令集	指令长度 不固定	通常 单周期 一条指令
流水线	需调用 微程序 ， 吞吐量小	流水线处理能 高效率 执行
寄存器	寄存器用于 特定目的	寄存器及其操作 多
LOAD-STORE结构	能直接在 存储器 中运行	只能处理 寄存器 中的数据（除LOAD和STORE指令）
寻址方式	复杂	简化
特点	硬件资源浪费	对 编译器 有更高要求

➤ **CPU**控制整个程序的执行，主要包括**控制器**和**运算器**两个主要部件，以及多个**寄存器**部件，它具有以下基本功能：

程序控制	控制指令的 执行顺序	程序是指令的有序集合，相互顺序不能任意颠倒，必须严格按程序规定的顺序执行。保证计算机按一定顺序执行程序是CPU的首要任务。
操作控制	控制指令进行 操作	一条指令的功能往往由若干个操作信号的组合来实现。因此，CPU管理并产生每条指令的操作信号，把各种操作信号送往相应的部件，从而控制这些部件按指令的要求进行操作。
时间控制	对各种操作实施 定时 控制	在计算机中，各种指令的操作信号和一条指令的整个执行过程都受到严格定时。只有这样，计算机才能有条不紊地工作。
数据加工	对数据进行 加工处理	数据加工就是对数据进行算术、逻辑运算。完成数据的加工处理，是CPU的根本任务。

➤ 控制器

- 控制器是整个计算机系统的指挥中心。在控制器的指挥控制下，运算器、存储器和输入/输出设备等部件协同工作，构成了一台完整的通用计算机
- 控制器通常由以下部分组成：
 - **程序计数器 (Program counter, PC)**
 - 指出下一条指令在主存储器中的单元地址；
 - 在程序执行之前，首先将程序的第一条指令所在的主存单元地址送入PC。因此PC的内容即是从主存提取的第一条指令的地址；
 - 当执行指令时，CPU能自动递增PC的内容，使其始终保持将要执行的下一条指令的主存地址，为取下一条指令做好准备。若为单字长指令，则 $(PC)+1 \rightarrow PC$ ，若为双字长指令，则 $(PC)+2 \rightarrow PC$ ，以此类推；
 - 当遇到转移指令时，下一条指令的地址将由转移指令的地址码字段指定，而不是像通常那样通过顺序递增PC的内容来取得；
 - 程序计数器的结构应当是具有寄存信息和计数两种功能的结构。



➤ 控制器

- 控制器是整个计算机系统的指挥中心。在控制器的指挥控制下，运算器、存储器和输入/输出设备等部件协同工作，构成了一台完整的通用计算机
- 控制器通常由以下部分组成：
 - 程序计数器 (Program counter, PC)
 - 指令寄存器 (Instruction register, IR) 和指令译码器 (Instruction decoder, ID)
 - 保存当前正在执行的一条指令；
 - 当执行一条指令时，先把该指令从主存读取到数据寄存器中，然后再传送至指令寄存器；
 - 一条指令被划分为操作码和地址码2个字段。为了执行任何给定的指令，必须对操作码进行测试，以便识别所要求的操作。指令译码器 (Instruction Decoder, ID) 就是完成这项工作的；
 - 指令译码器对来自指令寄存器的操作码部分进行译码，以产生操作性质的控制电位，并将其送到微操作控制线路上，在时序部件定时信号作用下，产生具体的操作控制信号；
 - 指令寄存器中操作码字段的输出就是指令译码器的输入。操作码一经译码后，即可向操作控制器发出具体操作的特定信号。



➤ 控制器

- 控制器是整个计算机系统的指挥中心。在控制器的指挥控制下，运算器、存储器和输入/输出设备等部件协同工作，构成了一台完整的通用计算机
- 控制器通常由以下部分组成：
 - 程序计数器 (Program counter, PC)
 - 指令寄存器 (Instruction register, IR)
 - 指令译码器 (Instruction decoder, ID)
 - 地址寄存器 (Address register, AR)
- 保存CPU当前所访问的主存单元的地址；
- 由于在主存和CPU之间存在操作速度上的差异，所以必须使用地址寄存器来暂时保存主存的地址信息，直到主存的存取操作完成为止；
- 当CPU和主存进行信息交换，即CPU向主存存入/取出数据时，或者CPU从主存中读出指令时，都要使用地址寄存器和数据寄存器；
- 如果把外围设备的设备地址作为像主存的地址单元那样来看待，那么，当CPU和外围设备交换信息时，我们同样要使用地址寄存器和数据寄存器。



➤ 控制器

- 控制器是整个计算机系统的指挥中心。在控制器的指挥控制下，运算器、存储器和输入/输出设备等部件协同工作，构成了一台完整的通用计算机
- 控制器通常由以下部分组成：
 - 程序计数器 (Program counter, PC)
 - 指令寄存器 (Instruction register, IR)
 - 指令译码器 (Instruction decoder, ID)
 - 地址寄存器 (Address register, AR)
 - 时序发生器 (Timing generator, TG)
- 控制操作时间，根据周期，信号等控制计算机有序工作。



➤ 控制器

- 控制器是整个计算机系统的指挥中心。在控制器的指挥控制下，运算器、存储器和输入/输出设备等部件协同工作，构成了一台完整的通用计算机
- 控制器通常由以下部分组成：
 - 程序计数器 (Program counter, PC)
 - 指令寄存器 (Instruction register, IR)
 - 指令译码器 (Instruction decoder, ID)
 - 地址寄存器 (Address register, AR)
 - 时序发生器 (Timing generator, TG)
 - 操作控制器 (Operating controller, OC)
- 根据指令操作码与时序信号，产生各种操作控制信号，以便正常建立数据通路，完成取指与执行的控制。



➤ 控制器

- 控制器是整个计算机系统的指挥中心。在控制器的指挥控制下，运算器、存储器和输入/输出设备等部件协同工作，构成了一台完整的通用计算机
- 控制器通常由以下部分组成：
 - 程序计数器 (Program counter, PC)
 - 指令寄存器 (Instruction register, IR)
 - 指令译码器 (Instruction decoder, ID)
 - 地址寄存器 (Address register, AR)
 - 时序发生器 (Timing generator, TG)
 - 操作控制器 (Operating controller, OC)
- 功能包括：
 1. 从主存中取出一条指令，并指出下一条指令在主存中的位置；
 2. 对指令进行译码，并产生相应的操作控制信号，以便启动规定的动作；
 3. 指挥并控制CPU、主存和输入/输出设备之间数据流动的方向。

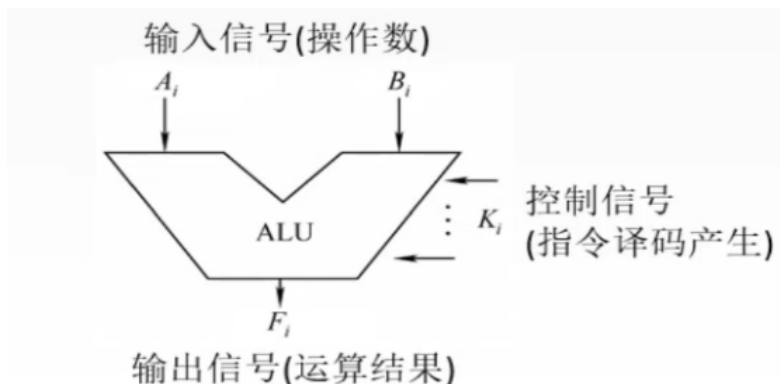


➤ 运算器

- 运算器是计算机中用于实现数据加工处理等功能的部件，它接受控制器的命令，负责完成对操作数据的加工处理任务，其核心部件是算术逻辑单元(Arithmetic Logic Unit, ALU)。
- 运算器接受控制器的命令而进行动作，即运算器所进行的全部操作都是由控制器发出的控制信号来指挥的，所以它是执行部件。
- 运算器由以下部分组成

• 算术逻辑单元 (Arithmetic logic unit, ALU)

- ALU的基本功能包括加，减，乘，除四则运算，与，或，非，异或等逻辑运算，以及移位，求补等操作；
- 运算器的操作和操作种类由控制器决定。



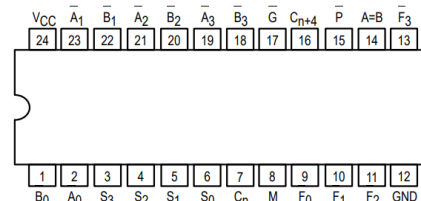
https://blog.csdn.net/qg_37291934

4-BIT ARITHMETIC LOGIC UNIT

The SN54/74LS181 is a 4-bit Arithmetic Logic Unit (ALU) which can perform all the possible 16 logic, operations on two variables and a variety of arithmetic operations.

- Provides 16 Arithmetic Operations Add, Subtract, Compare, Double, Plus Twelve Other Arithmetic Operations
- Provides all 16 Logic Operations of Two Variables Exclusive — OR, Compare, AND, NAND, OR, NOR, Plus Ten other Logic Operations
- Full Lookahead for High Speed Arithmetic Operation on Long Words
- Input Clamp Diodes

CONNECTION DIAGRAM DIP (TOP VIEW)



➤ 运算器

- 运算器是计算机中用于实现数据加工处理等功能的部件，它接受控制器的命令，负责完成对操作数据的加工处理任务，其核心部件是算术逻辑单元(Arithmetic Logic Unit, ALU)。
- 运算器接受控制器的命令而进行动作，即运算器所进行的全部操作都是由控制器发出的控制信号来指挥的，所以它是执行部件。
- 运算器由以下部分组成
 - 算术逻辑单元 (Arithmetic logic unit, ALU)
 - 累加寄存器 (Accumulator, AC)
 - AC是一个通用寄存器。
 - 功能：当运算器的算术逻辑单元(ALU)执行算术或逻辑运算时，为ALU提供一个工作区，可以为ALU暂时保存一个操作数或运算结果。
 - 运算器中至少要有一个累加寄存器。



➤ 运算器

- 运算器是计算机中用于实现数据加工处理等功能的部件，它接受控制器的命令，负责完成对操作数据的加工处理任务，其核心部件是算术逻辑单元(Arithmetic Logic Unit, ALU)。
 - 运算器接受控制器的命令而进行动作，即运算器所进行的全部操作都是由控制器发出的控制信号来指挥的，所以它是执行部件。
 - 运算器由以下部分组成
 - 算术逻辑单元 (Arithmetic logic unit, ALU)
 - 累加寄存器 (Accumulator, AC)
 - 数据寄存器 (Data register, DR)
- 主要作为CPU和主存、外设之间信息传输的中转站，以弥补CPU和主存、外设之间操作速度上的差异；
 - 数据寄存器用来暂时存放由主存储器读出的一条指令或一个数据字；反之，当向主存存入一个数据字时，也暂时将它们存放在数据寄存器中；
 - 在单累加器结构的运算器中，数据寄存器还可兼作操作数寄存器。



➤ 运算器

- 运算器是计算机中用于实现数据加工处理等功能的部件，它接受控制器的命令，负责完成对操作数据的加工处理任务，其核心部件是算术逻辑单元(Arithmetic Logic Unit, ALU)。
- 运算器接受控制器的命令而进行动作，即运算器所进行的全部操作都是由控制器发出的控制信号来指挥的，所以它是执行部件。
- 运算器由以下部分组成
 - 算术逻辑单元 (Arithmetic logic unit, ALU)
 - 累加寄存器 (Accumulator, AC)
 - 数据寄存器 (Data register, DR)
 - 程序状态字寄存器 (Program status word, PSW)
- 用来表征当前运算的状态及程序的工作方式;
- 程序状态字寄存器保存由算术指令和逻辑指令运行或测试的结果所建立起来的各种条件码内容，如运算结果进/借位标志(C)、运算结果溢出标志(O)等，这些标志位通常分别用1位触发器来保存;
- 保存中断和系统工作状态等信息，以便CPU和系统及时了解机器运行状态和程序运行状态;
- 因此，程序状态字寄存器是一个保存各种状态条件标志的寄存器。



➤ 运算器

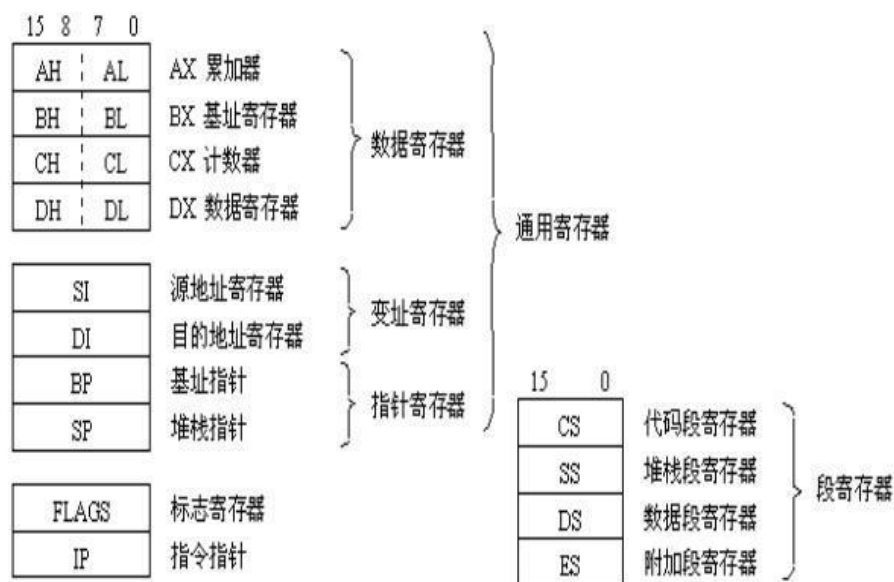
- 运算器是计算机中用于实现数据加工处理等功能的部件，它接受控制器的命令，负责完成对操作数据的加工处理任务，其核心部件是算术逻辑单元(Arithmetic Logic Unit, ALU)。
- 运算器接受控制器的命令而进行动作，即运算器所进行的全部操作都是由控制器发出的控制信号来指挥的，所以它是执行部件。
- 运算器由以下部分组成
 - 算术逻辑单元 (Arithmetic logic unit, ALU)
 - 累加寄存器 (Accumulator, AC)
 - 数据寄存器 (Data register, DR)
 - 程序状态字寄存器 (Program status word, PSW)
- 两个主要功能:
 1. 执行所有的算术运算;
 2. 执行所有的逻辑运算，并进行逻辑测试。



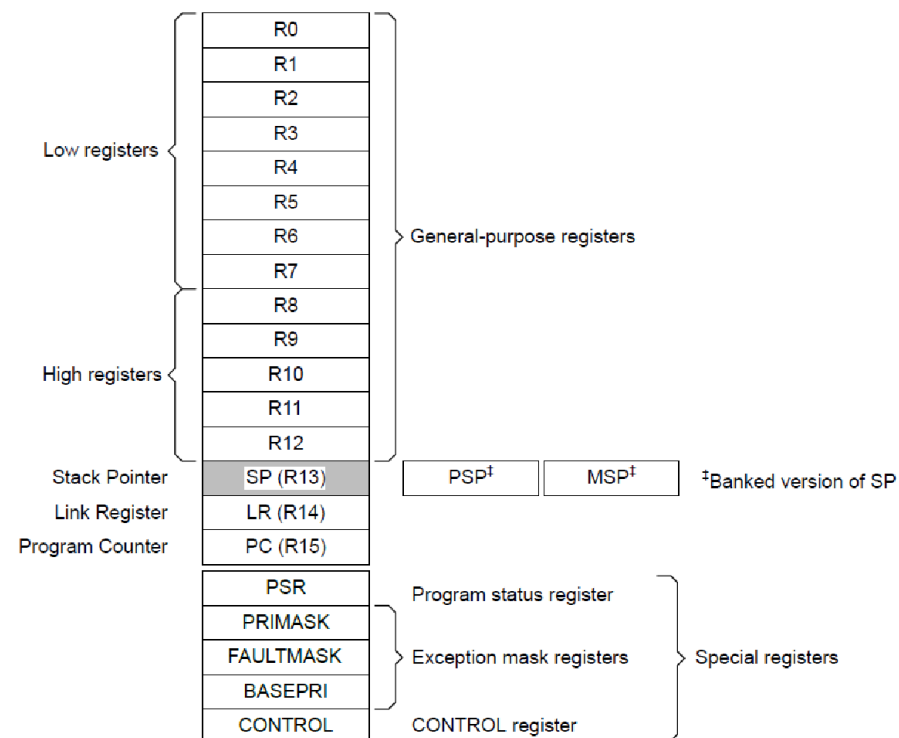
➤ 通用寄存器

- 通用寄存器可用于传送和暂存数据，也可参与算术逻辑运算，并保存运算结果。除此之外，它们还各自具有一些特殊功能。

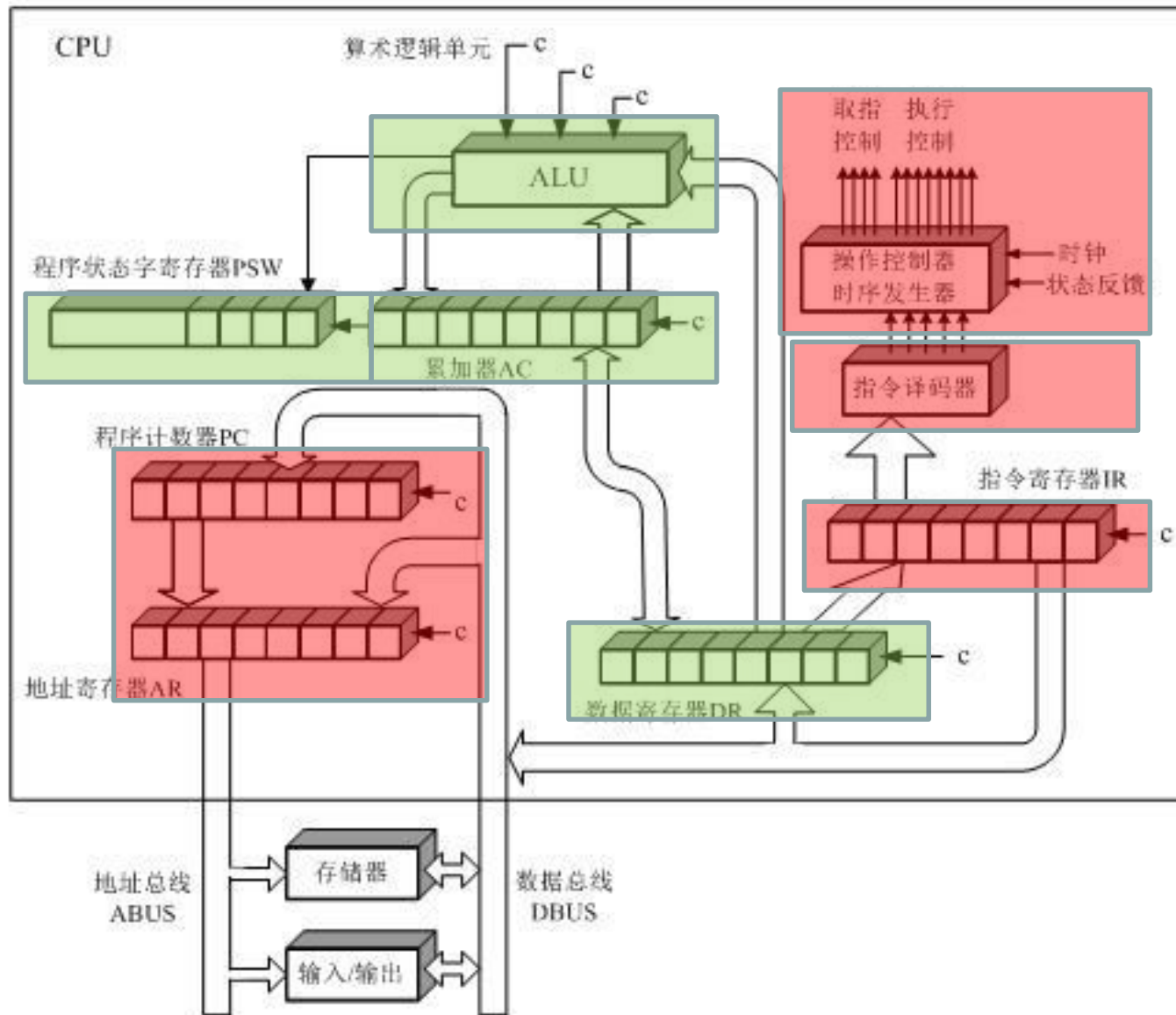
8086的通用寄存器组：



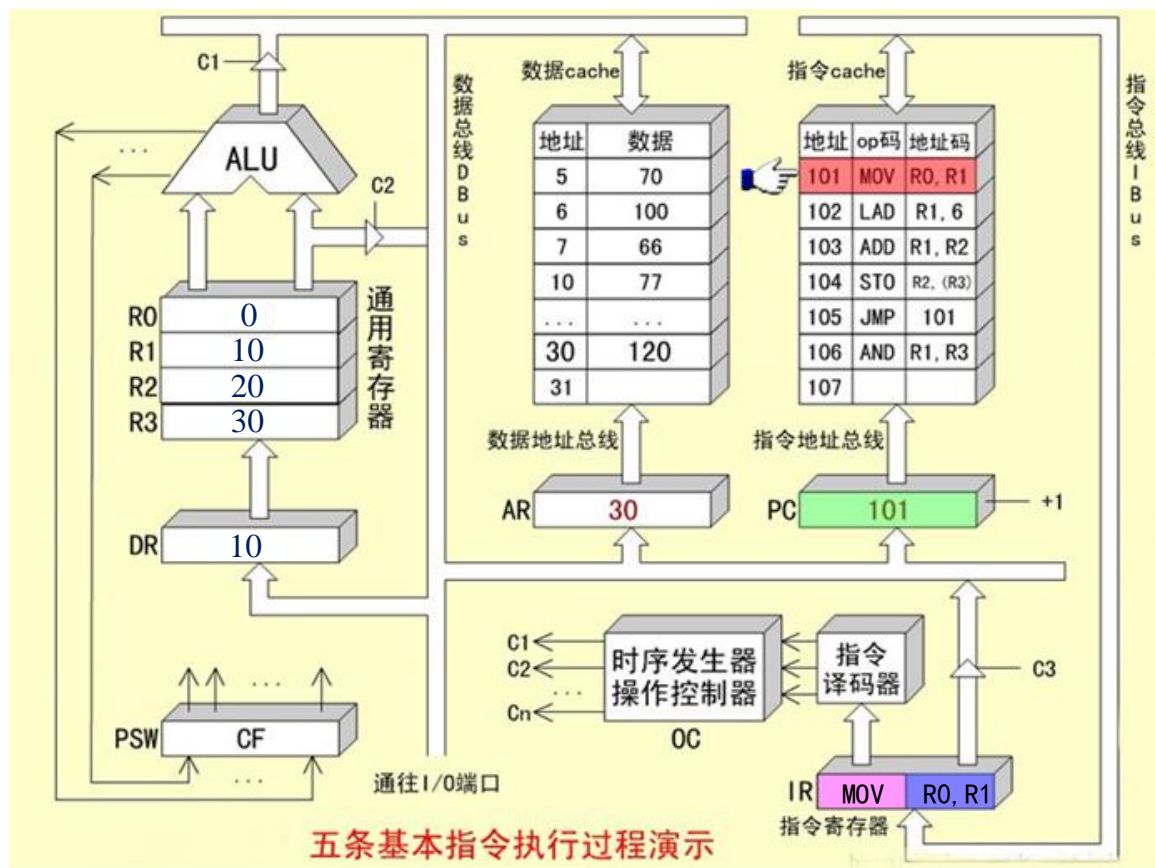
Cortex M4的通用寄存器组



冯诺依曼结构



➤ 哈佛结构

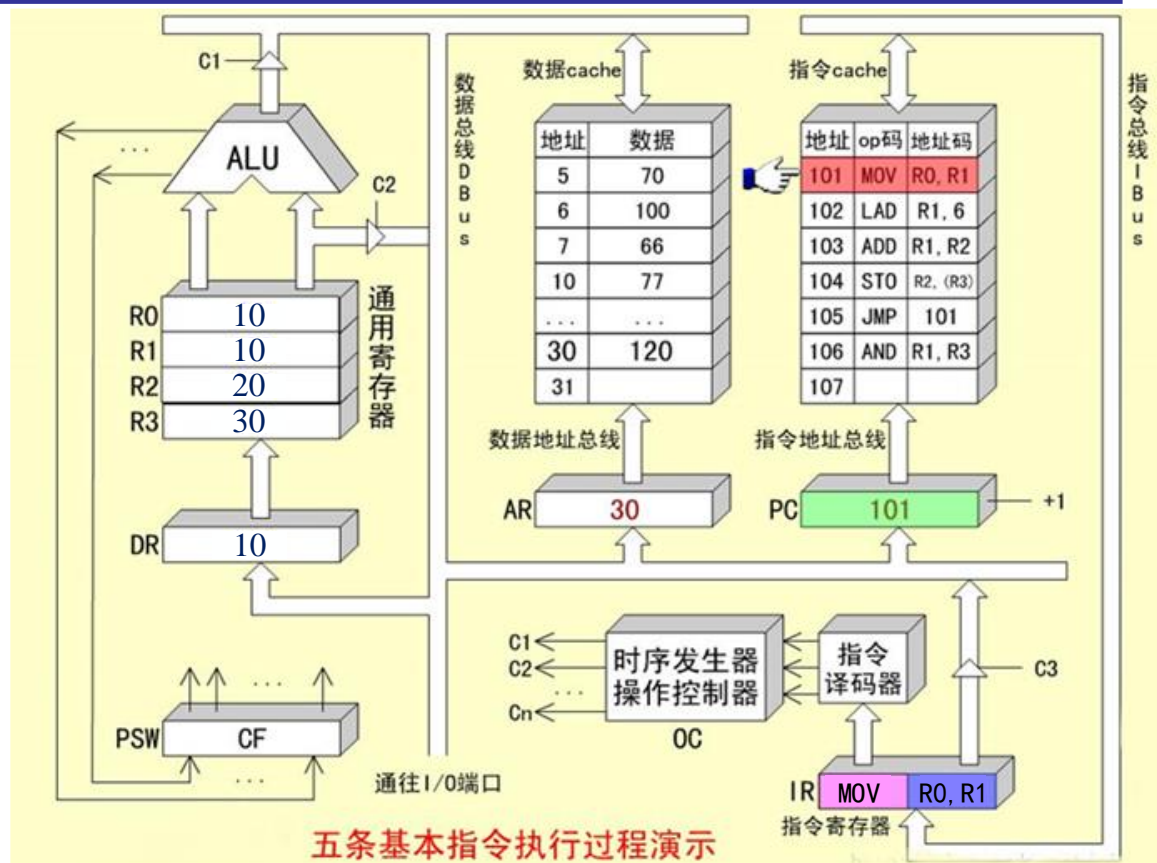


101 MOV指令

1. 取指译码阶段:

- 程序计数器PC装入第一条指令的地址101;
- PC的内容被放到指令地址总线上, 通过指令总线IBus装入指令寄存器IR;
- 程序计数器PC内容加1, 变成102, 为下一条指令做好准备;
- 从101号地址读出MOV指令, 指令寄存器IR中的操作码被译码, CPU识别出是MOV指令, 至此取指译码阶段完成。

➤ 哈佛结构

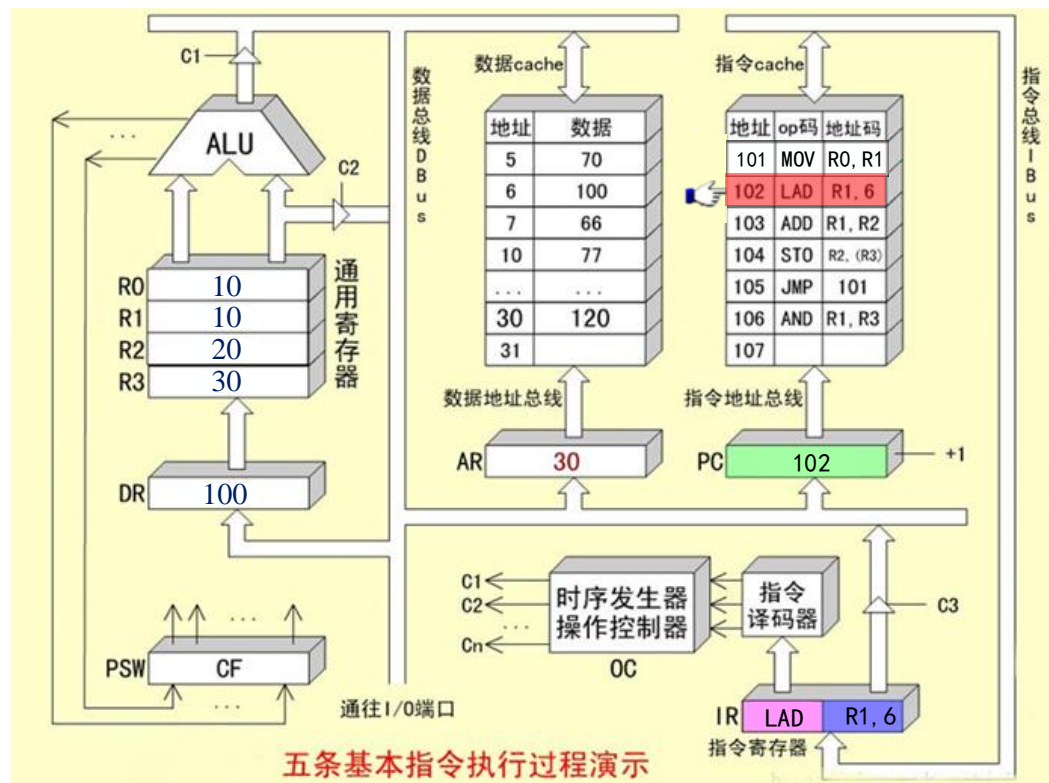


101 MOV指令

2. 执行阶段:

- 操作控制器OC送出控制信号到通用寄存器，选择R1（10）为源寄存器，R0（00）为目标寄存器；
- OC送出控制信号到ALU，指定ALU做传送操作，打开ALU输出三态门，将ALU输出（10）送到数据总线DBus上；
- 将DBus上的数据打入数据缓冲寄存器DR，将DR中的数据打入目标寄存器R0，R0的内容由0变为10，至此MOV指令执行完毕。

➤ 哈佛结构

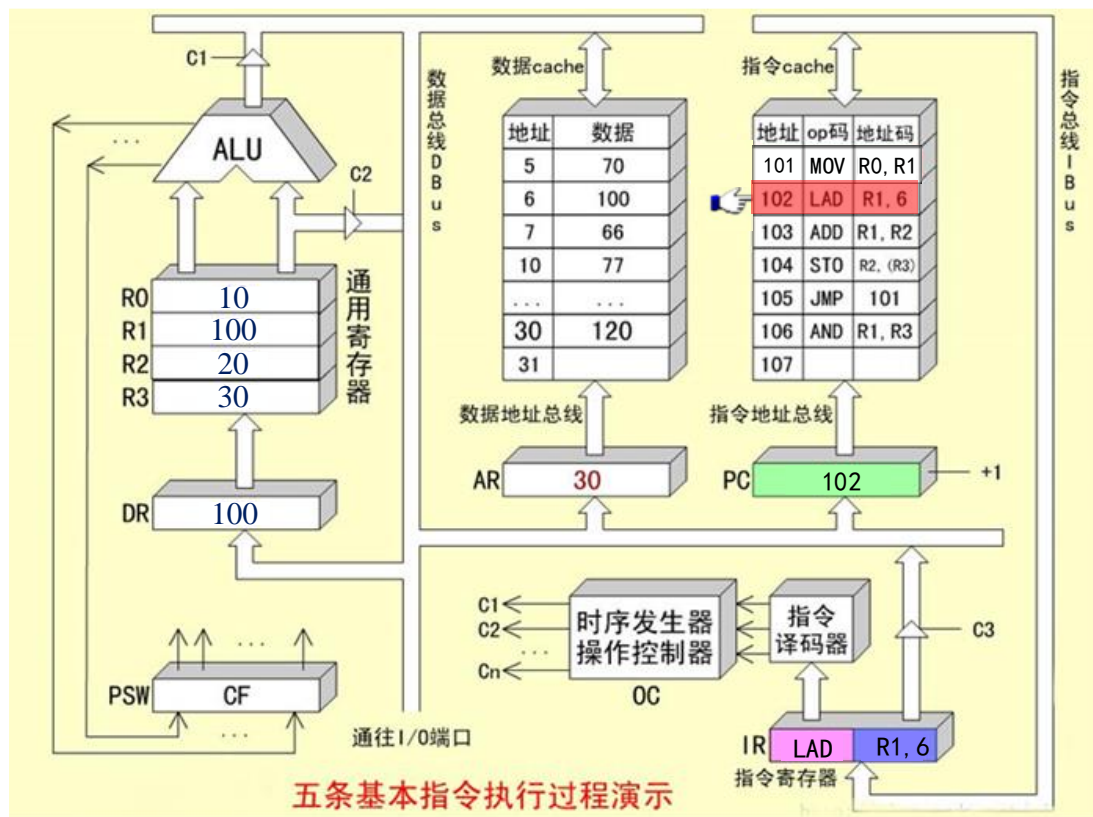


102 LAD指令（存储区->寄存器）

1. 取指译码阶段：

- LAD指令的取指阶段和MOV指令完全相同；
- PC（102）的内容被放到指令地址总线上，通过指令总线IBus装入指令寄存器IR
程序计数器PC内容加1，变成103，为下一条指令做好准备；
- 指令寄存器IR中的操作码被译码，CPU识别出是LAD指令，至此取指译码阶段完成。

➤ 哈佛结构

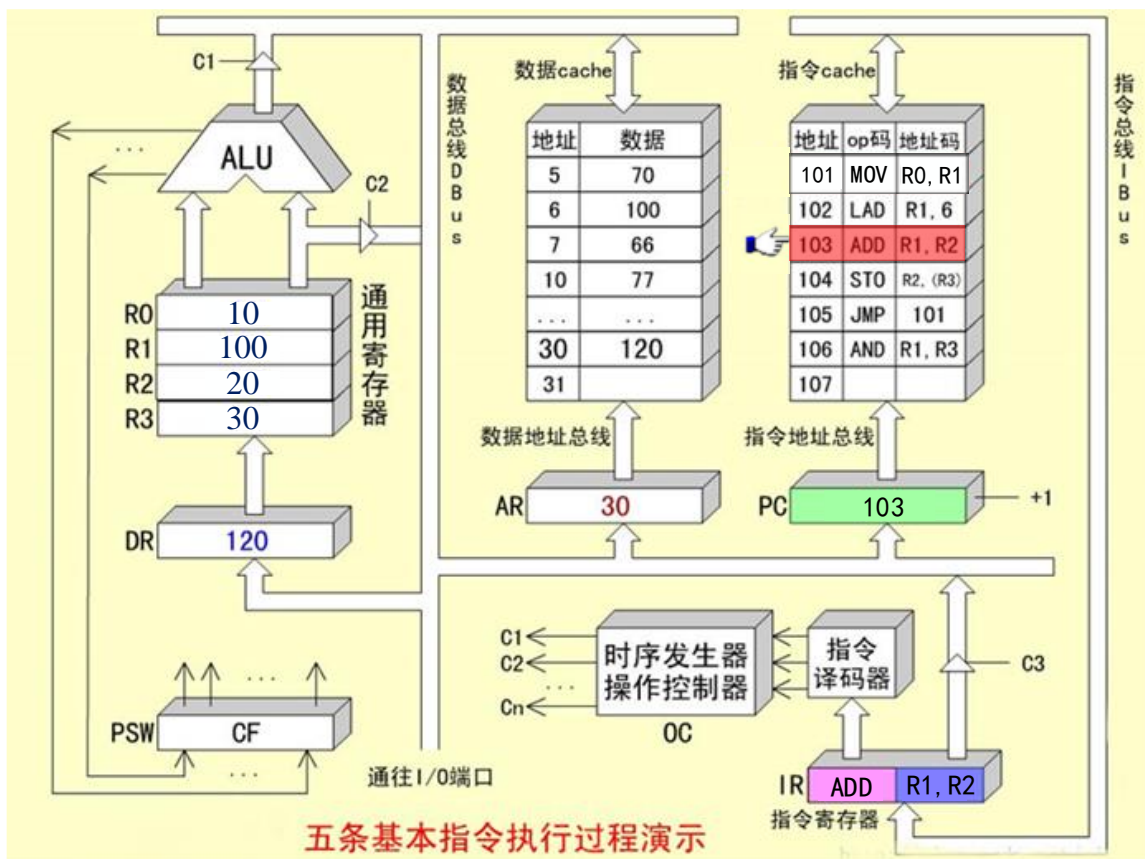


102 LAD指令 (存储区->寄存器)

2. 执行阶段:

- OC发出控制命令，打开IR输出三态门，将指令中的直接地址码6放到数据总线DBus上;
- 装入地址寄存器AR，将内存6号单元中的数100读出到DBus上，装入缓冲寄存器DR;
- 将DR中的数100装入通用寄存器R1，原来R1中的值10被覆盖，至此LAD指令执行完毕。

➤ 哈佛结构

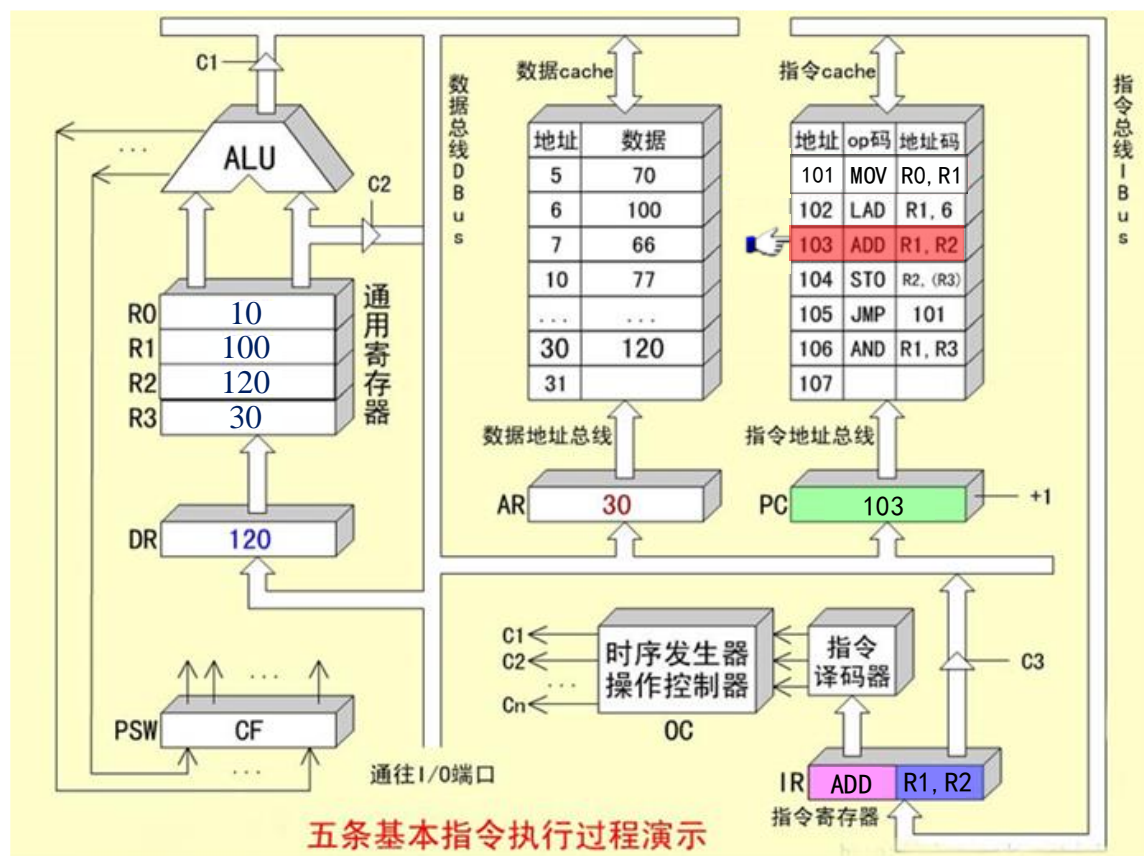


103 ADD指令

1. 取指译码阶段：

- ADD指令的取指译码阶段和其他指令相同；
- PC (103) 的内容被放到指令地址总线上通过指令总线IBus装入指令寄存器IR程序计数器PC内容加1，变成104，为下一条指令做好准备；
- 指令寄存器IR中的操作码被译码，CPU识别出是ADD指令，至此取指译码阶段完成。

➤ 哈佛结构

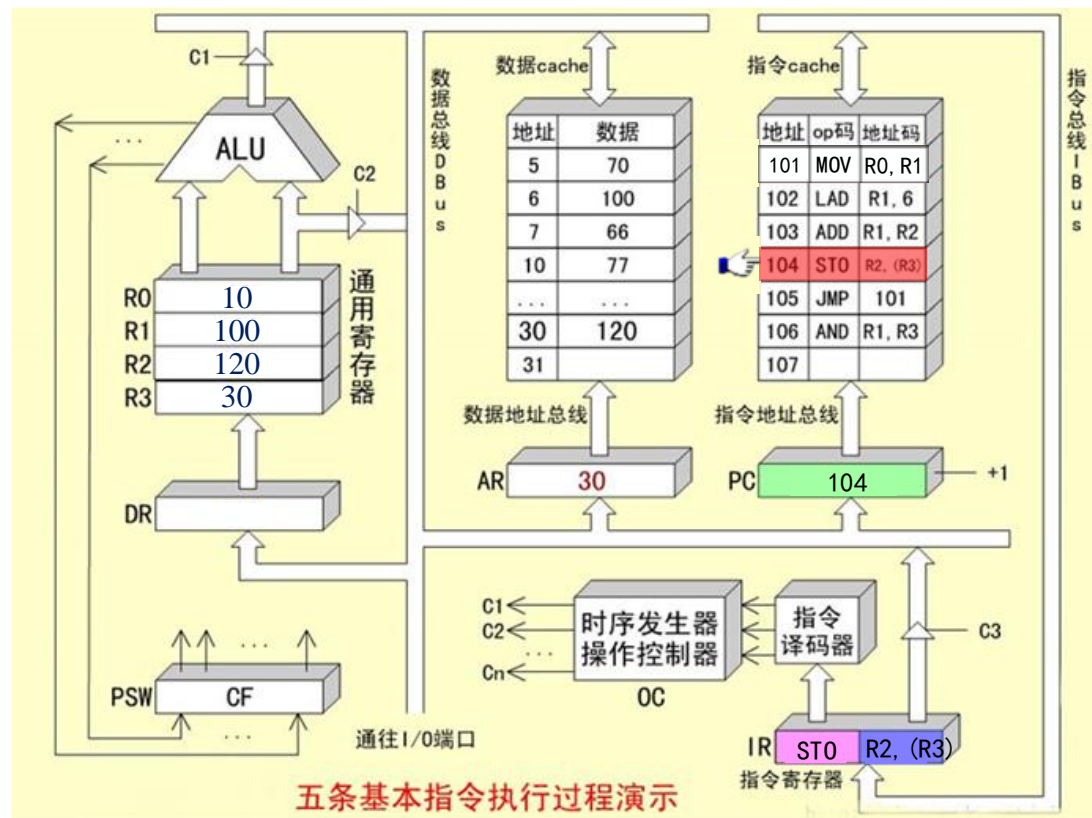


103 ADD指令

2. 执行阶段:

- 操作控制器OC送出控制信号到通用寄存器，选择R1（100）为源寄存器，R2（20）为目标寄存器；
- ALU做R1和R2的加法运算，打开ALU输出三态门，将运算结果120放到数据总线DBus上，然后打入缓冲寄存器DR；
- 将DR中数值120装入R2中，R2原来的数20被覆盖。到此ADD指令执行结束。

➤ 哈佛结构

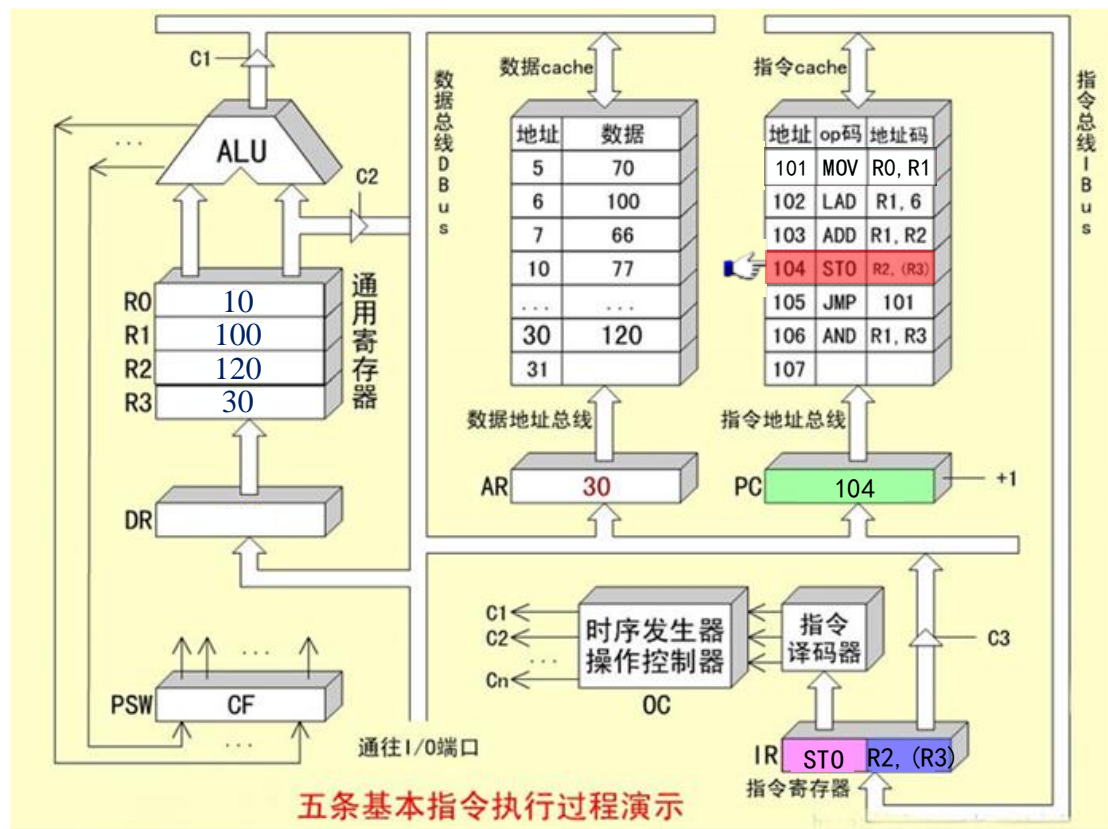


104 STO指令 (寄存器->存储区)

1. 取指译码阶段:

- STO指令的取指阶段和其他指令相同。
- PC (104) 的内容被放到指令地址总线上通过指令总线IBus装入指令寄存器IR
- 程序计数器PC内容加1, 变成105, 为下一条指令做好准备。
- 指令寄存器IR中的操作码被译码, CPU识别出是STO指令, 至此取指译码阶段完成。

➤ 哈佛结构

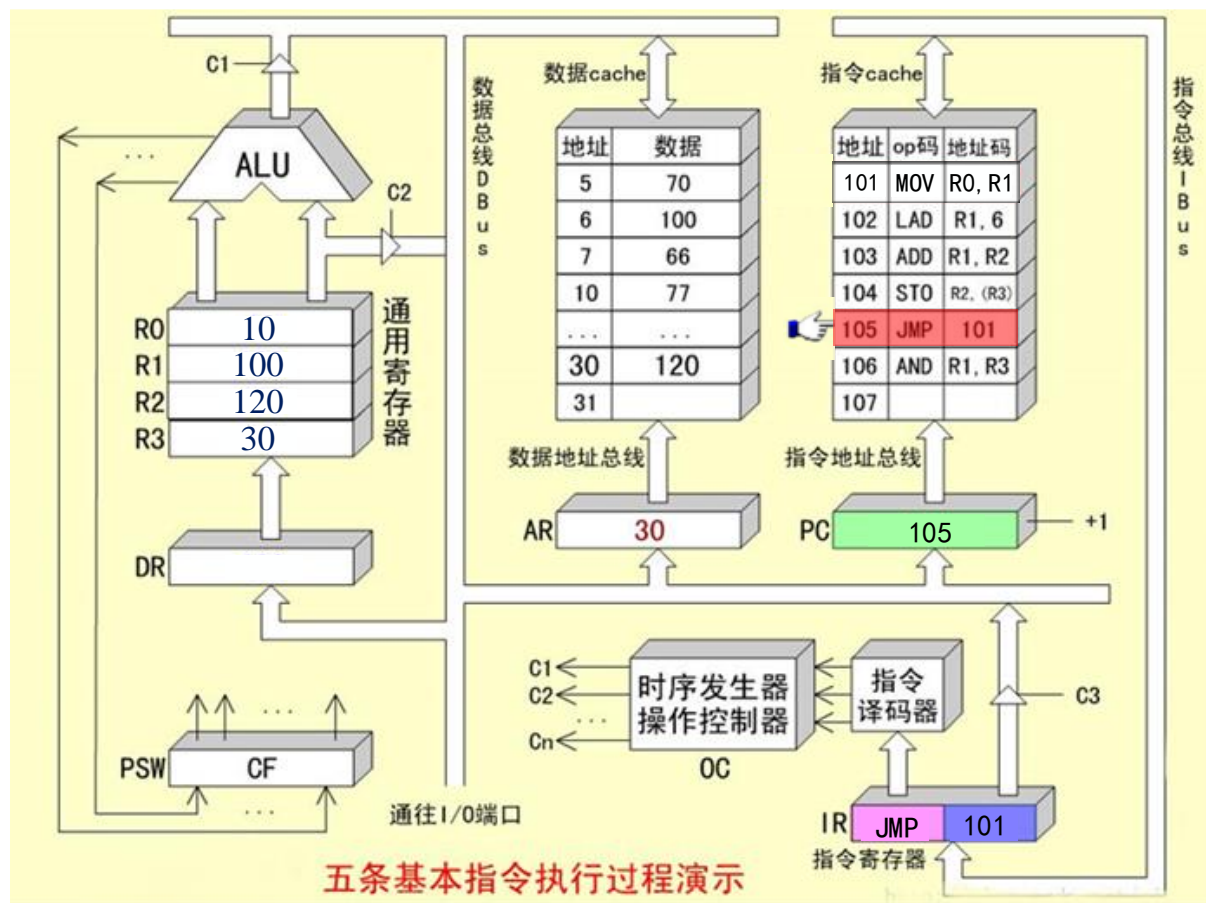


104 STO指令（寄存器->存储区）

2. 执行阶段:

- 操作控制器OC送出控制信号到通用寄存器，选择R3（30）作为数据存储器地址；
- 打开通用寄存器输出三态门，将地址30放到DBus上并装入地址寄存器AR，并进行地址译码；
- 操作控制器OC送出控制信号到通用寄存器，选择R2（120）作为数存的写入数据放到DBus上。将数值120写入数存30单元，原先的数据被冲掉。至此STO指令执行结束。

➤ 哈佛结构

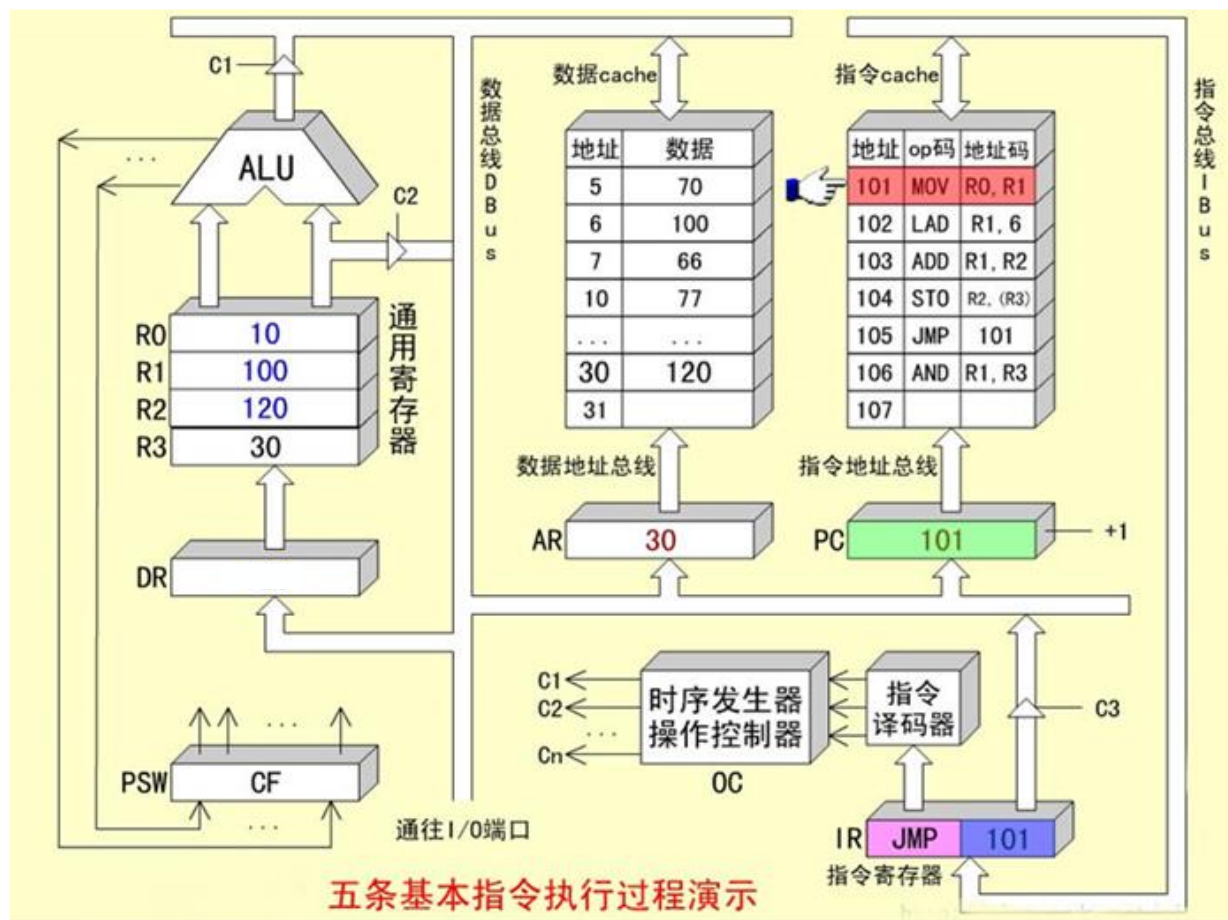


105 JMP指令

1. 取指译码阶段:

- PC (105) 的内容被放到指令地址总线上通过指令总线IBus装入指令寄存器IR;
- 程序计数器PC内容加1, 变成106, 为下一条指令做好准备。
- 指令寄存器IR中的操作码被译码, CPU识别出是JMP指令, 至此取指译码阶段完成。

➤ 哈佛结构



105 JMP指令

2. 执行阶段:

- OC发出控制命令，打开IR输出三态门，将IR中的地址码101发送到DBus上；
- 将DBus上的地址码101打入到程序计数器PC中，PC中原先的地址106被更换；
- 下一条指令不是从106单元取出，而是转移到101单元取出。至此JMP指令执行周期结束。

谢谢！

