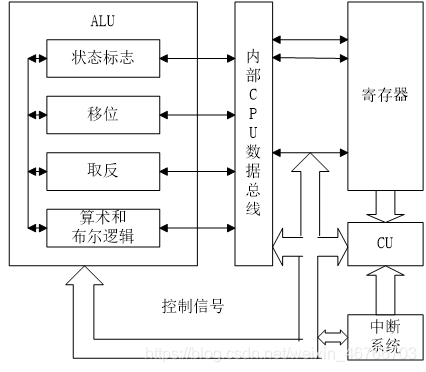
1. CPU的主要功能是执行存放在主存储器中的程序即机器指令

结构图如下：



各部分作用：

ALU：实现算术和逻辑运算

寄存器：存放操作数

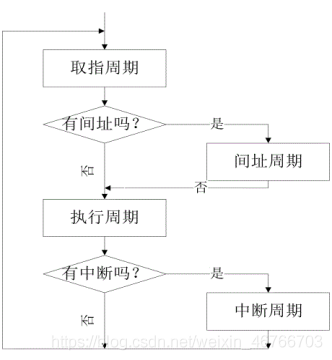
CU：发出各种操作命令序列的控制部件

中断系统：处理异常情况和特殊请求

1. 指令周期是取出并执行完一条指令所需的时间

指令周期没有固定值，因为计算机中各种指令执行所需的时间差异很大，因此为了提高CPU运行效率，即使在同步控制的机器中，不同指令的指令周期长度都是不一致的，也就是说指令周期对于不同的指令来说不是一个固定值

1. 流程图如下：



各子周期作用：

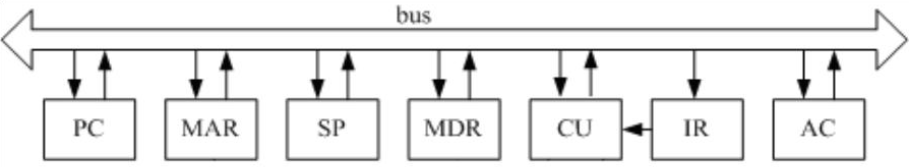
取指周期： 取指令

间址周期： 取有效地址

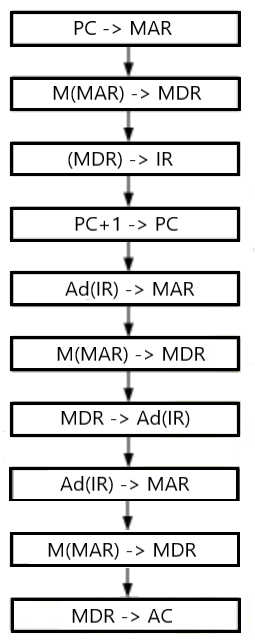
执行周期： 取操作数（当指令为访存指令时）

中断周期： 保存程序断点

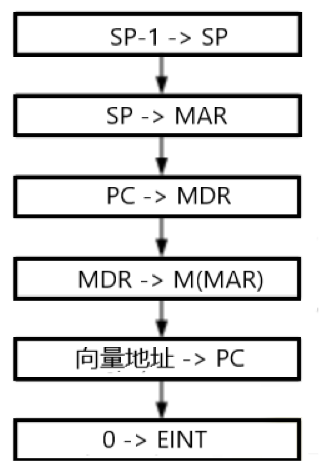
1. 设采用单总线将所给部件连接，连接图如下：



1. 指令周期数据流程图



1. 中断周期数据流程图：



7.

并行性：包含同时性和并发性。同时性指两个或两个以上的事件在同一时刻发生，并发性指两个或多个事件在同一时间段发生。即在同一时刻或同一时间段内完成两个或两个以上性质相同或性质不同的功能，只要在时间上存在相互重叠，就存在并行性

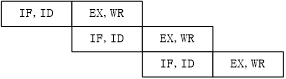
粗粒度并行是指多个处理机上分别运行多个进程，由多台处理机合作完成一个程序，一般算法实现。

细粒度并行是指在处理机的指令级和操作级的并行性。

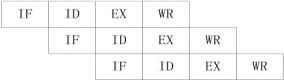
8.

指令流水：指将一条指令的执行过程分为n个操作时间大致相等的阶段，每个阶段由一个独立的功能部件来完成，这样n个部件可以同时执行n条指令的不同阶段，从而大大提高CPU的吞吐率

二级流水示意图：



四级流水示意图：



四级流水更能提高CPU速度，原因如下：

假设IF、ID、EX、WR每个阶段耗时为t，则连续执行n条指令

采用二级流水线时，耗时为：4t+(n-1)2t = (2n+2)t

采用四级流水线时，耗时为：4t+(n-1)t = (n+3)t

在n>1时，n+3<2n+2，可见四级流水线耗时比二级流水线耗时短，因此更能提高处理机速度。

10.

三种数据相关：写后读相关、读后写相关、写后写相关

* 1. 写后读：指令j试图在指令i写入寄存器之前就写入该寄存器，这样，指令i就错误地读出该寄存器旧的内容
  2. 读后写：指令j试图在指令i读出寄存器之前就写入该寄存器，这样，指令i就错误地读出该寄存器新的内容
  3. 写后写：指令j试图在指令i写入寄存器之前就写入该寄存器，这样，两次写的先后次序被颠倒，就会错误地使由指令i写入的值成为该寄存器的内容

15.

中断： 指当出现需要时，CPU暂时停止当前程序的执行转而执行处理新情况的程序和执行过程。即在程序运行过程中，系统出现了一个必须由CPU立即处理的情况，此时，CPU暂时中止程序的执行转而处理这个新的情况的过程就叫做中断

设计中断系统需解决的问题：

(1) 各中断源如何向CPU提出中断请求

(2) 当对个中断源同时提出中断请求时，中断系统如何确定优先响应哪个中断源的请求

(3)CPU在什么条件、什么时候、以什么方式来响应中断

(4)CPU响应中断后如何保护现场

(5)CPU响应中断后，如何停止原程序的执行而转入中断服务程序的入口地址

(6)中断处理结束后，CPU如何恢复现场，如何返回到原程序的间断处。

(7)在中断处理过程中又出现了新的中断请求，CPU该如何处理。

17.

INTR——中断请求触发器，用来登记中断源发出的随机性中断请求信号，以便为CPU查询中断及中断排队判优线路提供稳定的中断请求信号 。

EINT——中断允许触发器，CPU中的中断总开关。当EINT=1时，表示允许中断（开中断），当EINT=0时，表示禁止中断（关中断）。其状态可由开、关中断等指令设置 。

INT——中断标记触发器，控制器时序系统中周期状态分配电路的一部分，表示中断周期标记。当INT=1时，进入中断周期，执行中断隐指令的操作。

18.

中断隐指令： CPU响应中断之后，经过某些操作，转去执行中断服务程序。这些操作是由硬件直接实现的

功能：

(1) 保存断点

(2) 暂不允许中断

(3) 引出中断服务程序

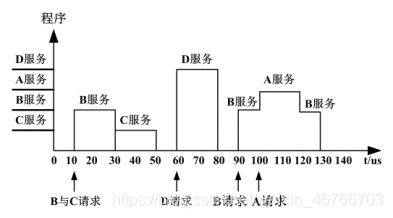
21.

(1)硬件向量法

(2)软件查询法

24.

A、B、C、D的响优先级即处理优先级。 CPU执行程序的轨迹图如下：



25.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 中断源 | 屏蔽字 | | | | |
| 0 | 1 | 2 | 3 | 4 |
| L0 | 1 | 0 | 0 | 1 | 0 |
| L1 | 1 | 1 | 1 | 1 | 1 |
| L2 | 1 | 0 | 1 | 1 | 0 |
| L3 | 0 | 0 | 0 | 1 | 0 |
| L4 | 1 | 0 | 1 | 1 | 1 |

26.

A、B、C设备的响应优先级为A最高、B次之、C最低，处理优先级为A最高、C次之、B最低。 CPU执行程序的轨迹图如下：

