第八章 冯·诺依曼模型

本章重点

- 基本结构
  - 存储器
  - 处理单元
  - 控制单元
  - 输入/输出设备
- 示例——DLX
- 指令处理（移到第九章）

基本结构

- 简介
- 存储器
- 处理单元
- 控制单元
- 输入/输出设备

冯·诺依曼模型

- 1943：ENIAC（第一台电子计算机）
  - 电子管，体积庞大、耗电量大……
  - 十进制、没有存储器、硬连线程序、运算慢……
- 1944：EDVAC（第二台电子计算机）开始研制
  - 电子离散变量自动计算机（ Electronic Discrete Variable Automatic Computer）
  - 二进制、程序存储于内存中
- 1945：约翰·冯·诺依曼（John von Neumann）
  - 关于EDVAC的报告草案（First Draft of a Report on EDVAC）
  - 提出冯·诺依曼体系结构/模型，也叫普林斯顿结构
  - 存储器与中央处理器分离，程序和数据存储不分离
  - 存储程序计算机模型，现代计算机的构建思想

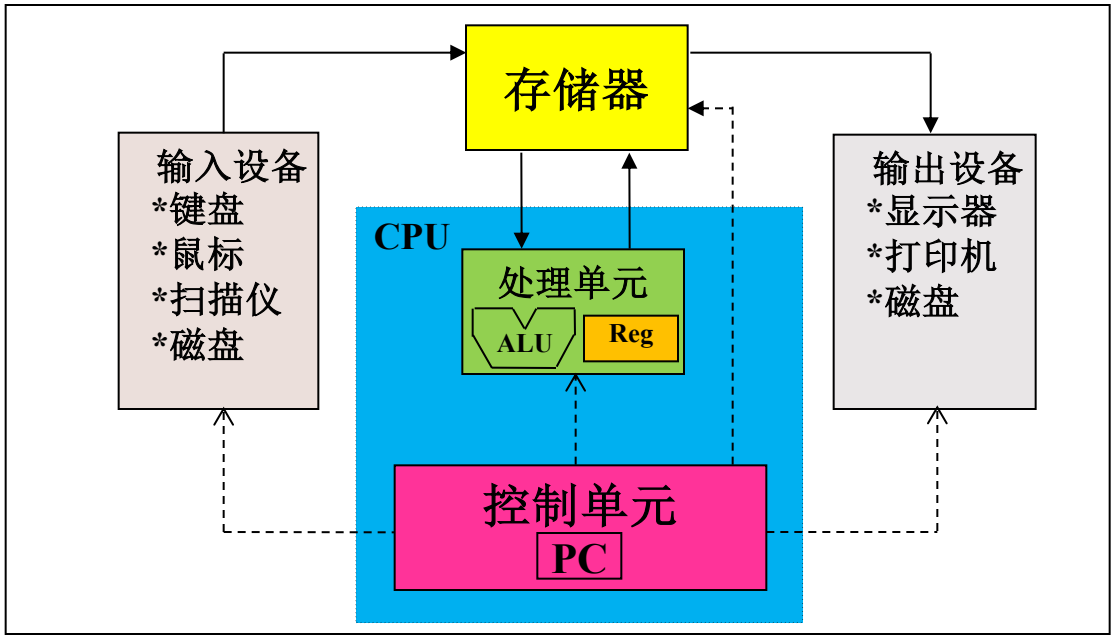
冯·诺依曼

- 美籍匈牙利数学家
- 理论计算机科学与博弈论的奠基者
- 在泛函分析、遍历理论、几何学、拓扑学和数值分析等众多数学领域及计算机科学、量子力学和经济学等领域都作过重大贡献

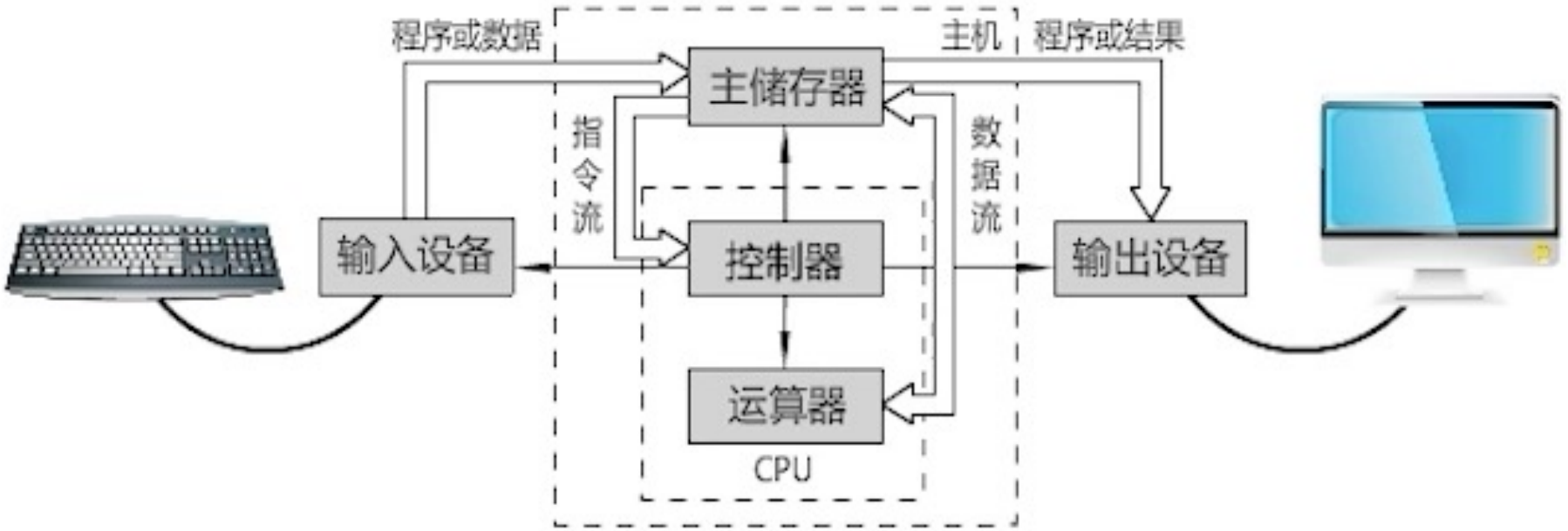
冯·诺依曼(1903~1957)

冯·诺依曼模型

- 由指令组成的**程序**和程序所需的**数据**位于**存储器**中
- 指令的执行由**运算器处理单元**完成
- 指令执行的顺序由**控制单元**来控制
- **输入设备**将程序和所需的数据送入计算机
- **输出设备**将执行结果送出计算机之外
- 注意：处理单元和控制单元是**CPU**的主要组成部分



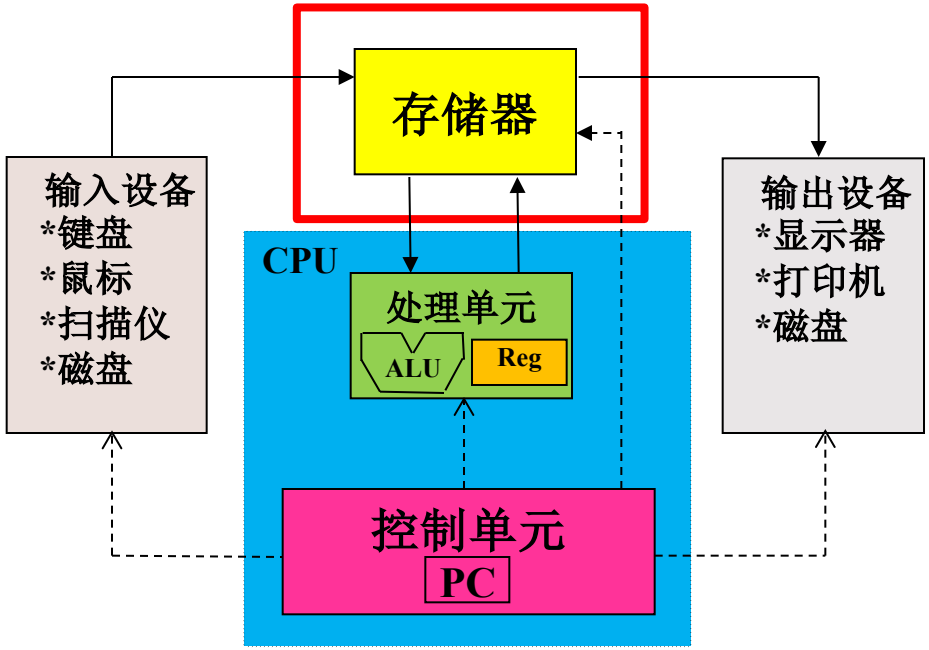
冯·诺依曼模型



基本结构

- 简介
- 存储器
- 处理单元
- 控制单元
- 输入/输出设备

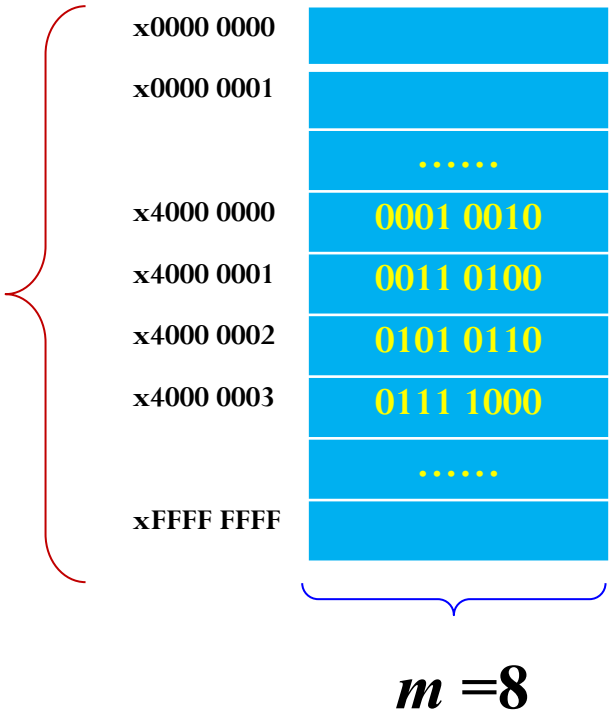
存储器



- 基本存储元件
  - 锁存器、触发器
  - 存储一位信息
- 寄存器
  - 存储多位信息
- 内存/主存储器（Memory）
  - 存储信息的二维阵列，2^n行，每一行存储m位
  - 每一行是一个存储单元（Memory Location）
  - 包含一定大小的内容（指令和数据）

存储器

- 地址
  - 和每一个单元联系在一起的唯一的标识符
  - 二进制地址　　$n = 32$
  - 总数：地址空间　　2^{32}
- 寻址能力
  - 存储在每一个单元中的信息的位数



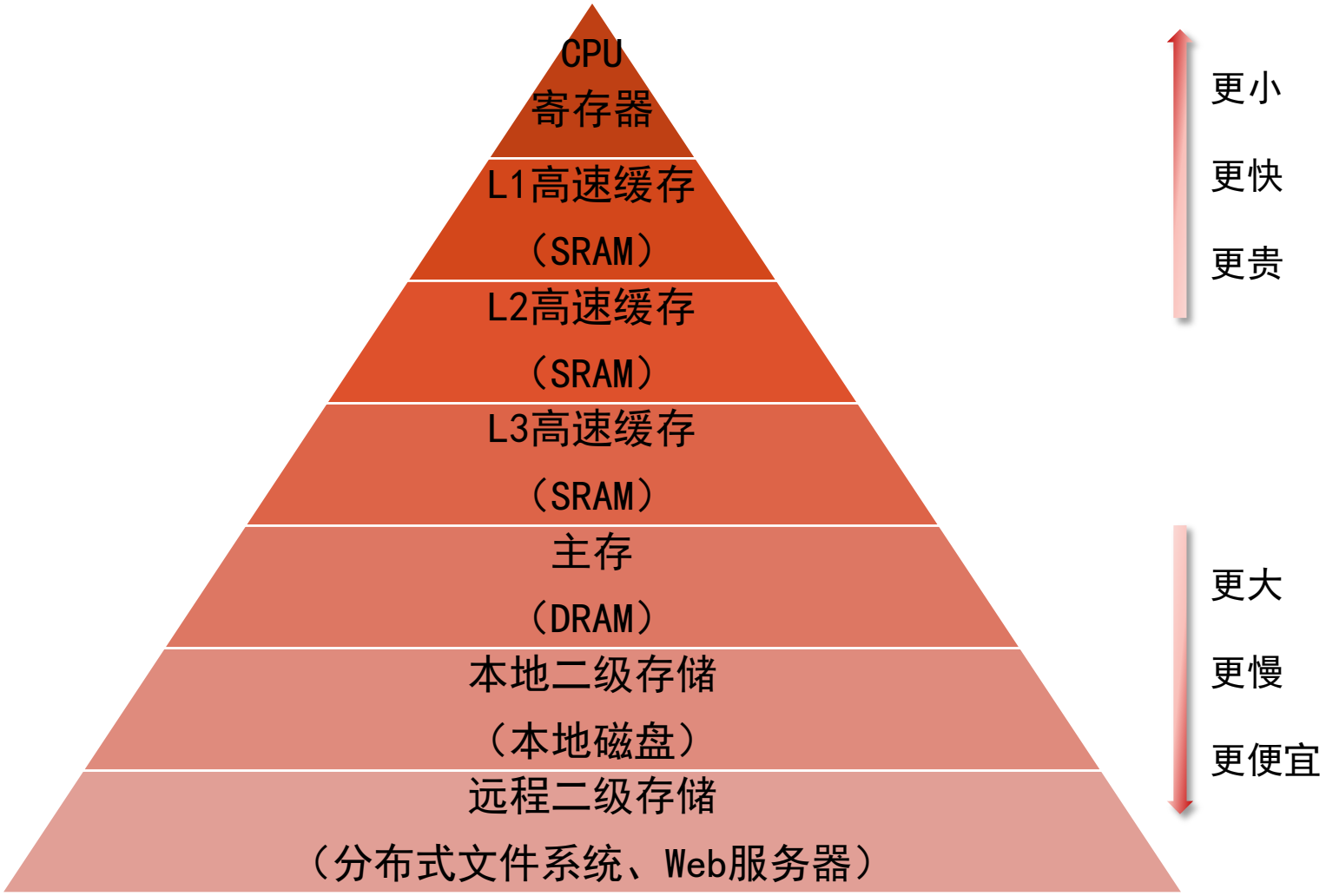
地址空间

- 唯一可识别的**单元总数**
- 2^{32}×8位，有2^{32}个不同存储单元的地址空间和8位的寻址能力，4G字节（缩写为GB）的存储器
- 要确定2^{32}个地址，需使用32位二进制数表示
  - 2^{10}，1024，1KB
  - 2^{20}，1MB
  - 2^{30}，1GB
  - 2^{32}，4GB，4294967296（约40亿）

寻址能力

- 存储在每个单元中的位数
- 大多数的存储器，字节可寻址
  - B（byte，字节），表示8位的信息量
  - 原始操作数据，键盘上键入的某个字符
    - 对应的ASCII码在存储器中占一个单元
    - 访问一个存储单元即可读写一个字符
- 64位可寻址
  - 用于科学计算的计算机

存储器


CPU
寄存器
L1高速缓存
（SRAM）
L2高速缓存
（SRAM）
L3高速缓存
（SRAM）
主存
（DRAM）
本地二级存储
（本地磁盘）
远程二级存储
（分布式文件系统、Web服务器）
更小
更快
更贵
更大
更慢
更便宜

SRAM

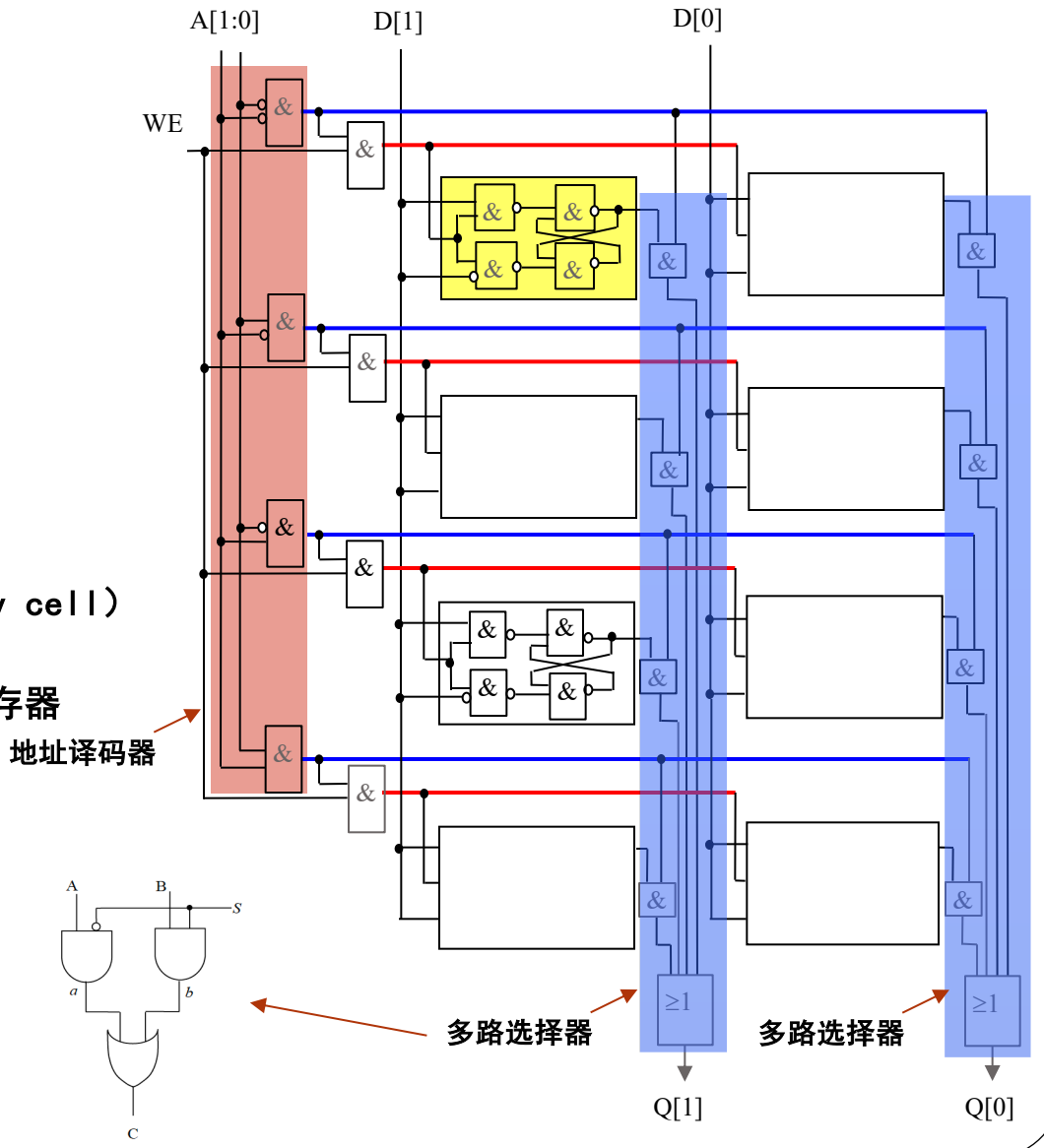
- SRAM（Static Random Access Memory，静态随机访问存储器）
- 结构相对简单
- “静态”：只要给它供电，其内部数据就不会丢失，可以一直保存
- “随机访问”：可以以任意顺序访问，而不必关心前一次访问的是哪一个单元
- 存储元：可以用更少的晶体管实现

DRAM

- DRAM（Dynamic Random Access Memory，动态随机访问存储器）
- “动态”：使用电容存储电荷保存数据
  - 必须隔一段时间刷新（refresh）一次，
  - 如果存储单元没有被刷新，存储的信息就会丢失
  - 关机也会丢失数据
- “随机访问”：可以以任意顺序访问，而不必关心前一次访问的是哪一个单元
- 存储元：采用动态存储单元，最常见的系统内存

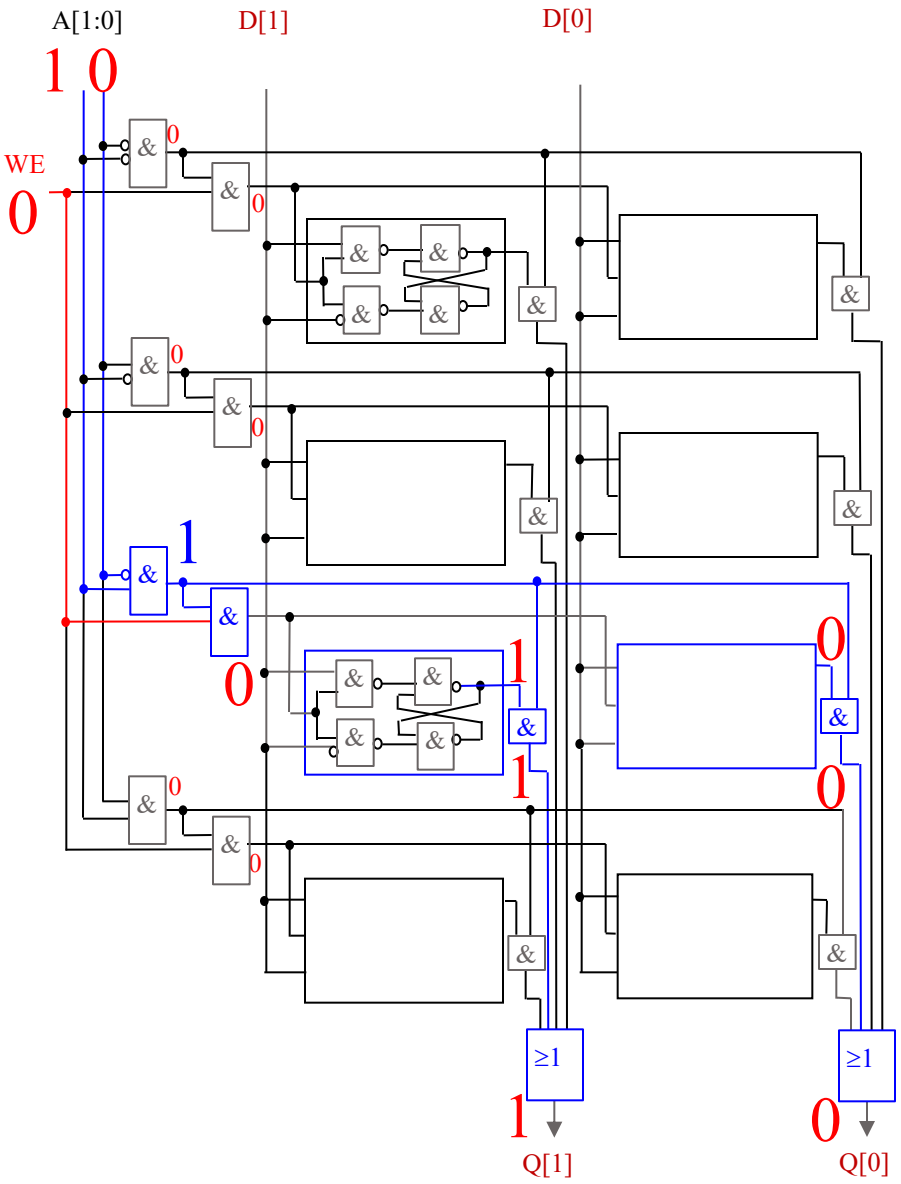
一个4×2的存储器

- 地址空间：4
- 寻址能力：2
- 地址：A[1:0]
- 地址译码器（红色阴影）
- 字线（蓝线）
- 字（2位）
  - 每一位由一个存储元（Memory cell）组成
  - 存储元：可以是一个门控D锁存器（黄色阴影）
- 多路选择器（蓝色阴影）
- 输出：Q[1:0]
- 输入：D[1:0]
- WE，字WE（红线）



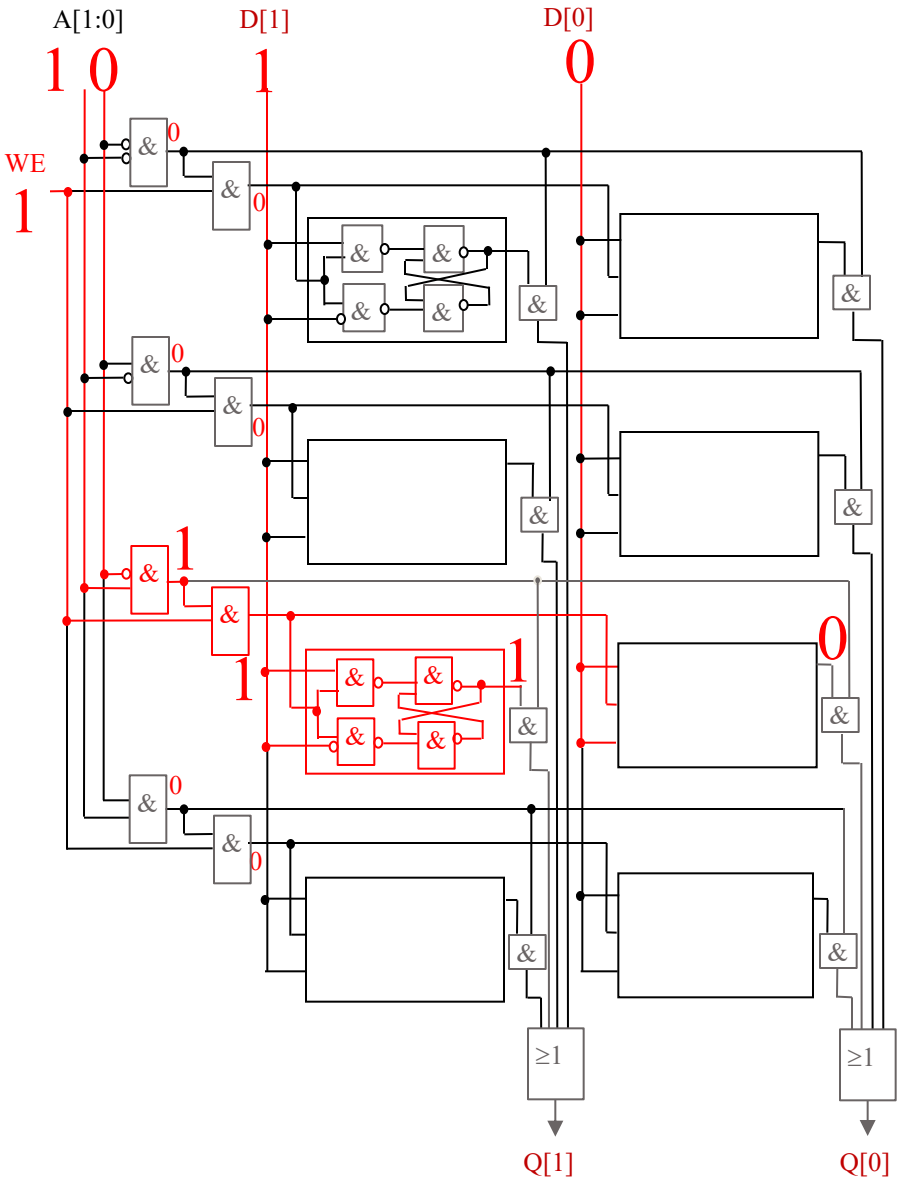
读单元2

- 地址
  - A[1:0]
- WE=0
- 输出
  - Q[1:0]



写/存储

- 地址
  - A[1:0]
- 输入
  - D[1:0]
- WE=1
  - 字WE=1



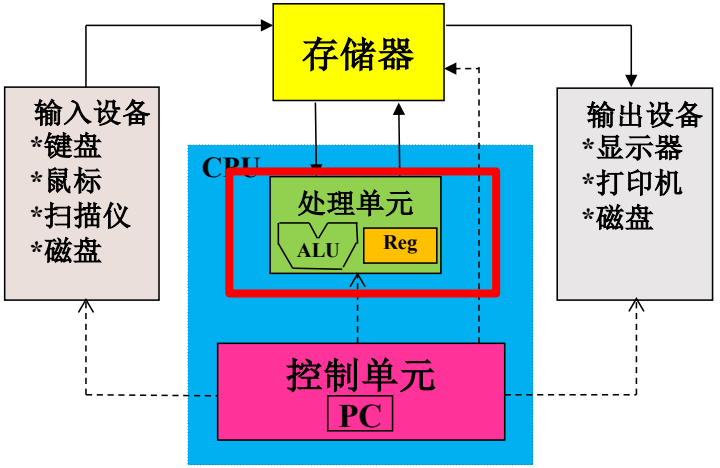
习题（1）

- 书面作业
- 7. 14
- 7. 15
- 7. 16

基本结构

- 简介
- 存储器
- 处理单元
- 控制单元
- 输入/输出设备

处理单元

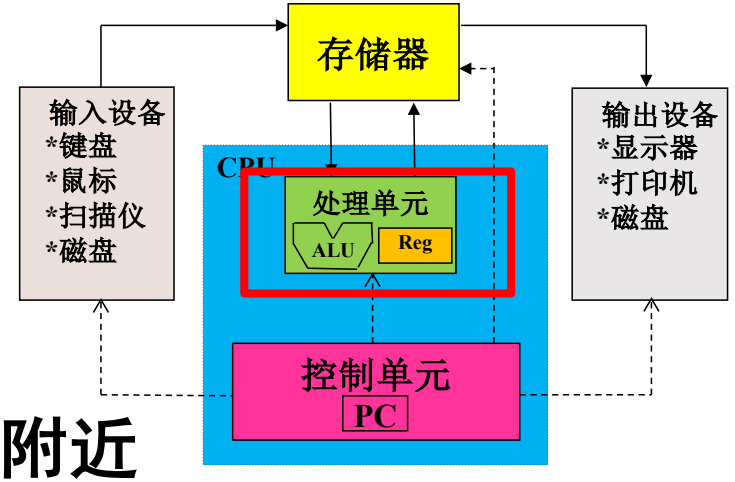


- 算术和逻辑单元ALU
  - Arithmetic and Logic Unit
  - 最简单的处理单元
- 功能：“执行信息的实际处理”
- 现代计算机包含许多复杂功能的处理单元，执行一个特定的运算（除法，平方根等）

字

- ALU一次正常处理的信息量大小通常被称为计算机的字长(word length)，每一次被处理的元素被称为一个字（word）
- 取决于计算机的不同用途，每一个指令集结构都拥有自己的字长
  - Intel的Pentium Ⅳ处理器，32位
  - Sun的SPARC-V9和Intel的Itanium处理器，64位

处理单元

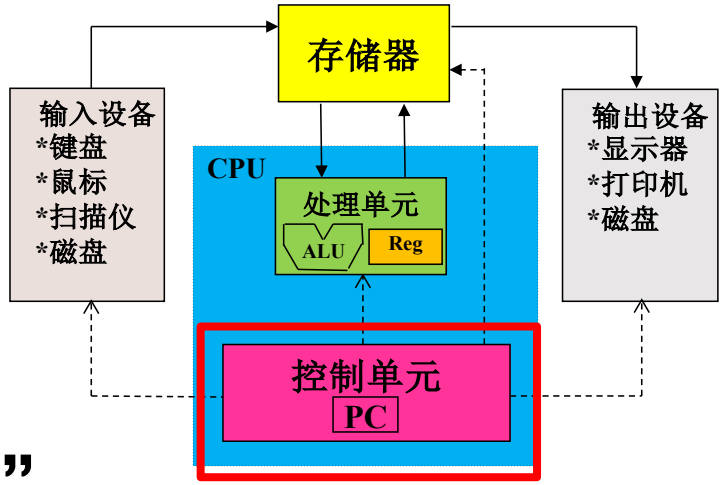


- 寄存器堆/文件Reg，ALU附近
- 功能：临时存取数据
  - 计算(A+B)×C，先在存储器中存储A+B的结果，随后读取出来，再和C相乘
  - 访问存储器的时间远长于执行加法或乘法的时间
  - 使用临时存储空间Reg存储A+B的结果
- 典型寄存器的大小等于ALU一次处理数据位数
  - 每个寄存器都包含一个字

基本结构

- 简介
- 存储器
- 处理单元
- 控制单元
- 输入/输出设备

控制单元



- 功能：“指挥信息的处理”
- 其具体工作包括：
  - 在执行程序的过程中，跟踪存储器中的指令
  - 在处理指令的过程中，跟踪指令的处理阶段
- 控制单元可以是多个控制器，从属于各个部件
  - ALU控制器用于控制ALU执行何种运算
  - 对于输入和输出则有专门的I/O控制器
  - ……

PC

- 程序计数器/指令指针
- Program Counter，简称PC
- 控制单元中容纳下一条指令所在地址的寄存器
- 功能：跟踪存储器中的指令，确切地说是跟踪要处理的下一条指令

基本结构

- 简介
- 存储器
- 处理单元
- 控制单元
- 输入/输出设备

输入/输出设备

- 要使计算机处理信息，信息必须被送入计算机中
- 为了能够使用处理后的结果，它必须能以某种形式显示在计算机以外
- 为输入和输出的目的而出现的设备在计算机术语中被称为外围设备（peripherals）

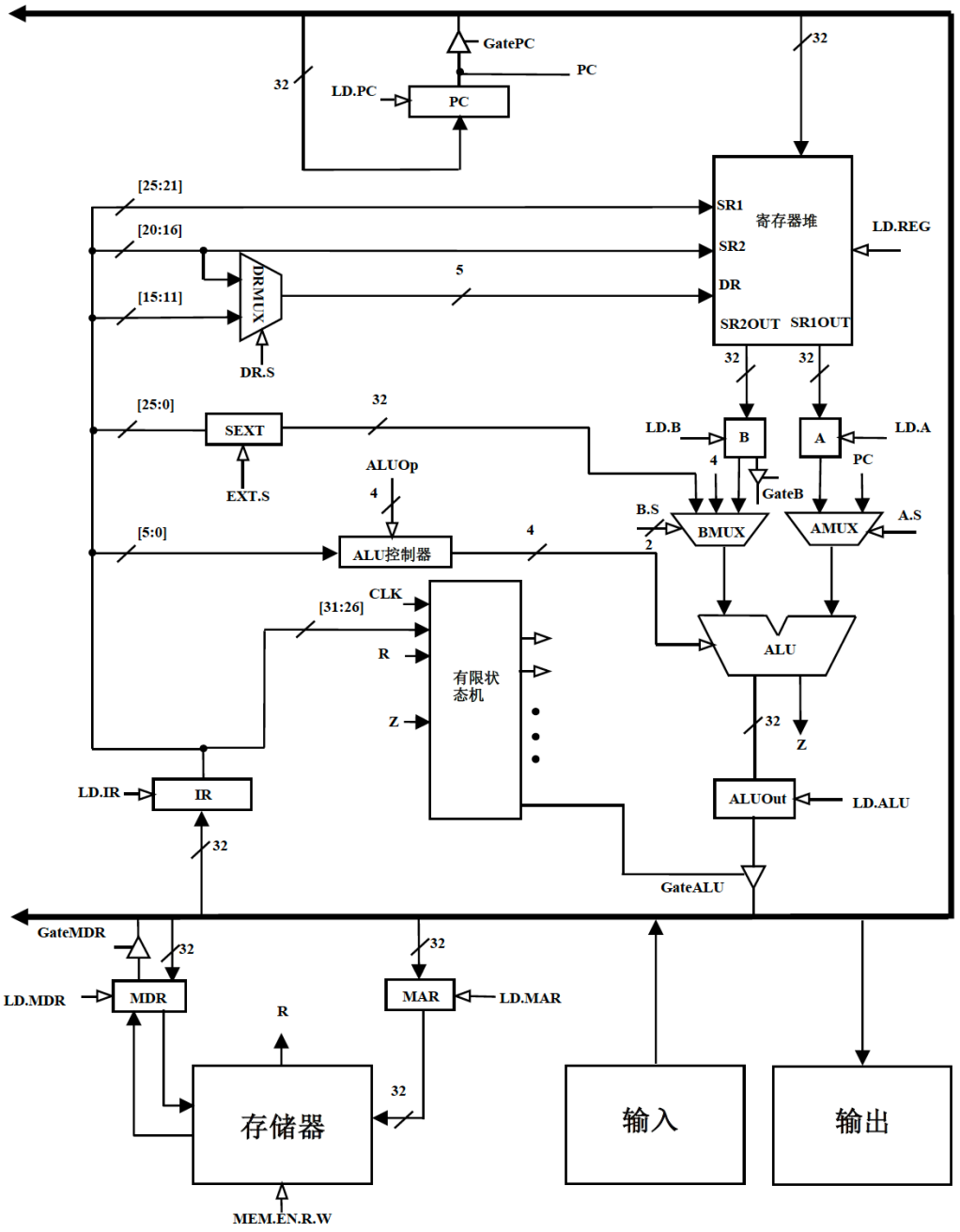
输入/输出设备

- 外围设备（peripherals）
  - 键盘——输入；监视器（显示器）——输出
  - 输入：鼠标，数字扫描仪、磁盘
  - 输出：打印机，磁盘

示例——DLX

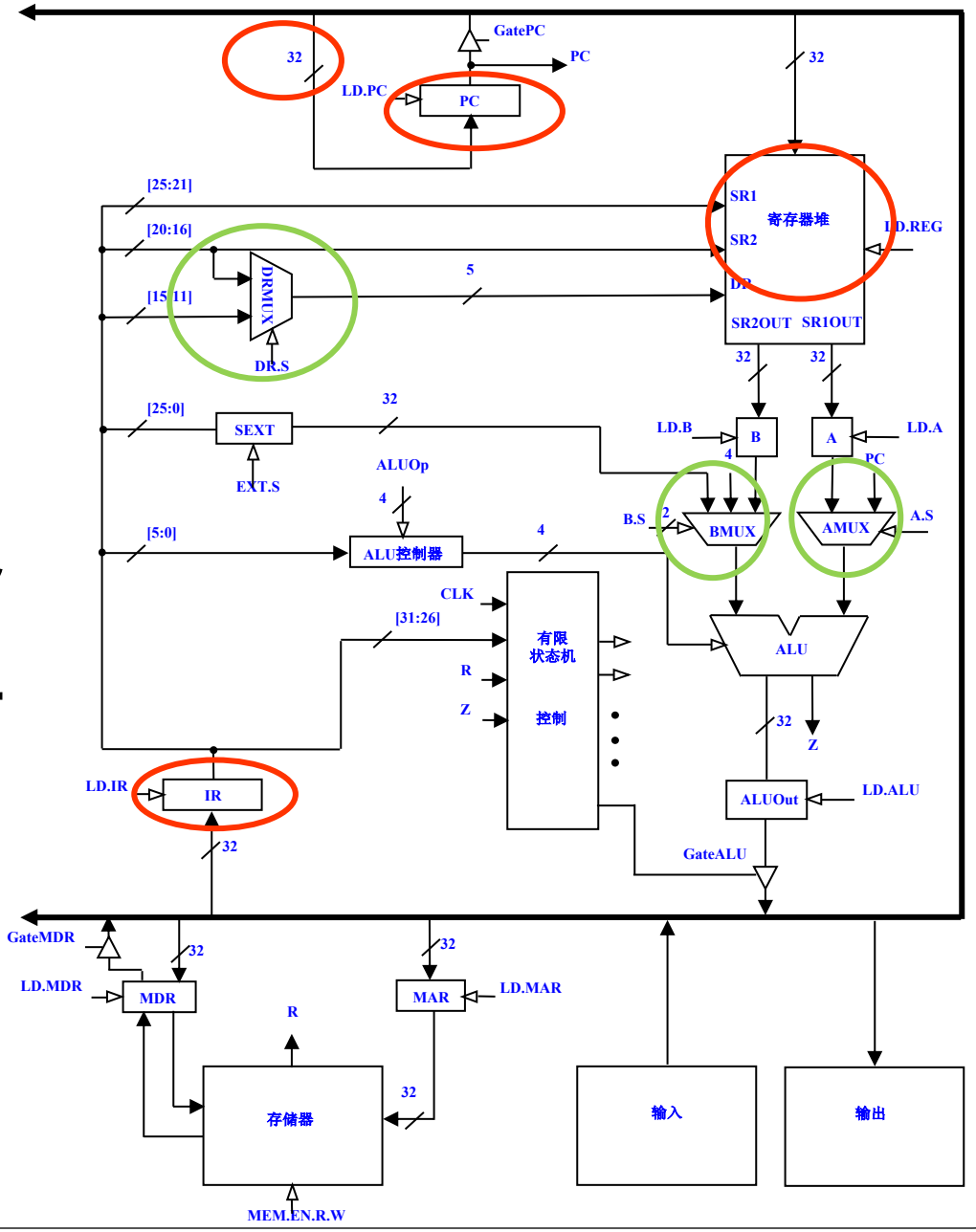
DLX数据通路

- 一个采用**总线结构**、**多时钟周期**的实现方案
- “数据通路”，在计算机内部用于处理信息的所有元件的总和



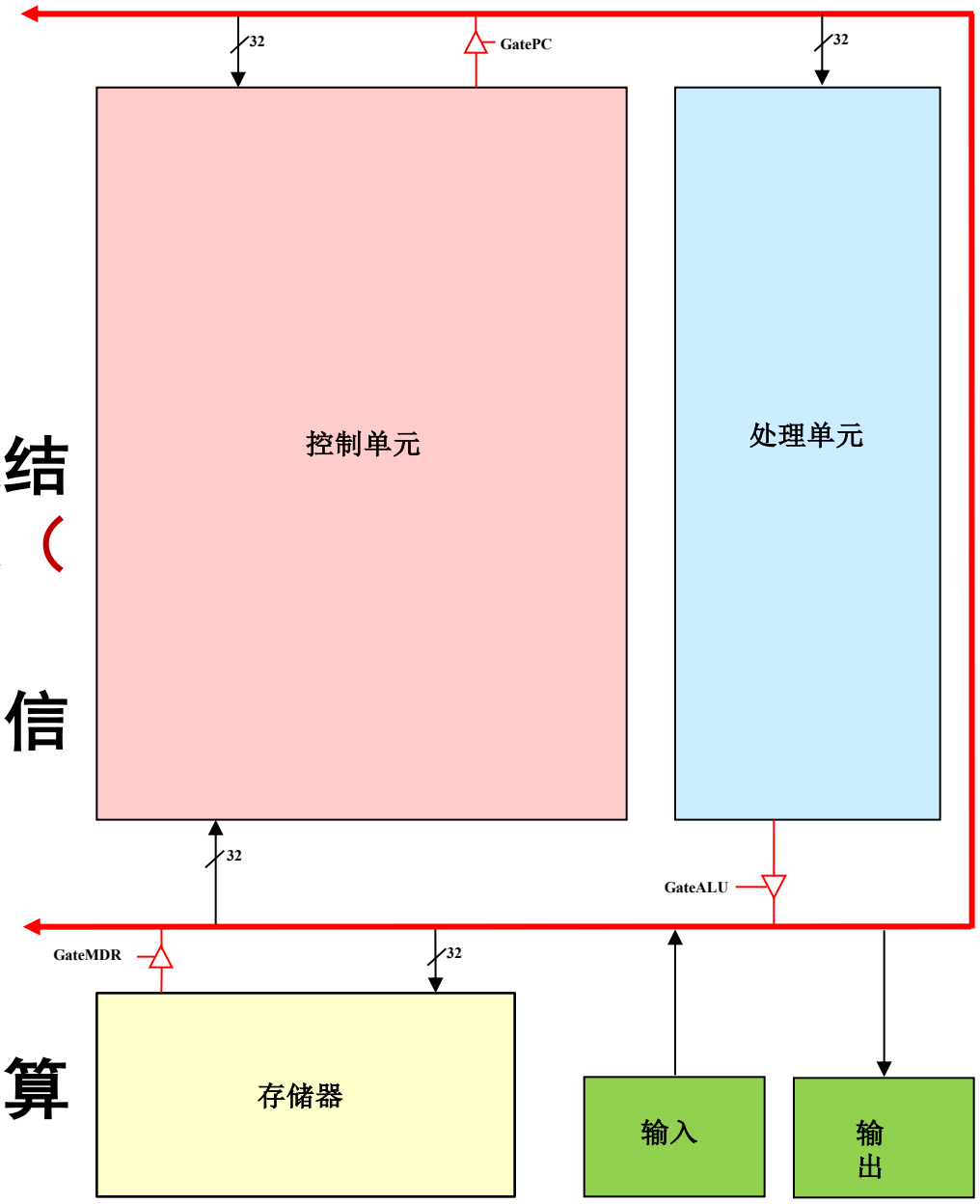
DLX数据通路

- 寄存器（32位）
  - 寄存器堆/文件
  - 程序计数器PC
  - 指令寄存器IR
- 多路选择器
  - DRMUX提供一个5位的地址给寄存器堆
  - AMUX和BMUX分别提供一个32位的数值给ALU
- 每根用交叉斜线标记32的线表示该线内共有32条线，每条用来传送1位的信息



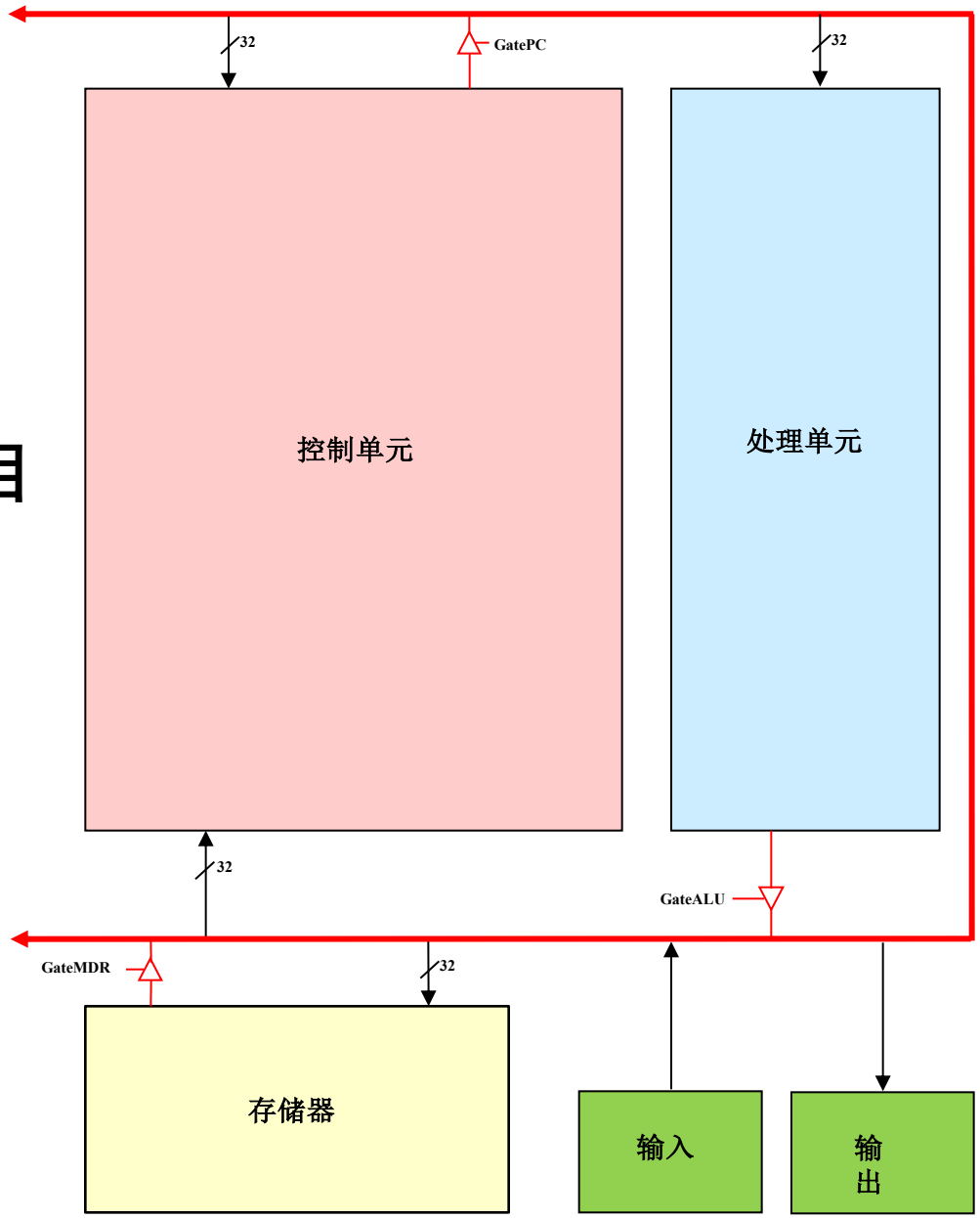
DLX总线

- 两端都有箭头的粗黑线结构代表数据通路的**总线（物理线或导线）**
- 连接模型各组件之间，信息通信高速公路
- 数据/地址/控制总线
- 优点：功能多、成本低
- 缺点：性能和带宽对计算机性能有重要影响



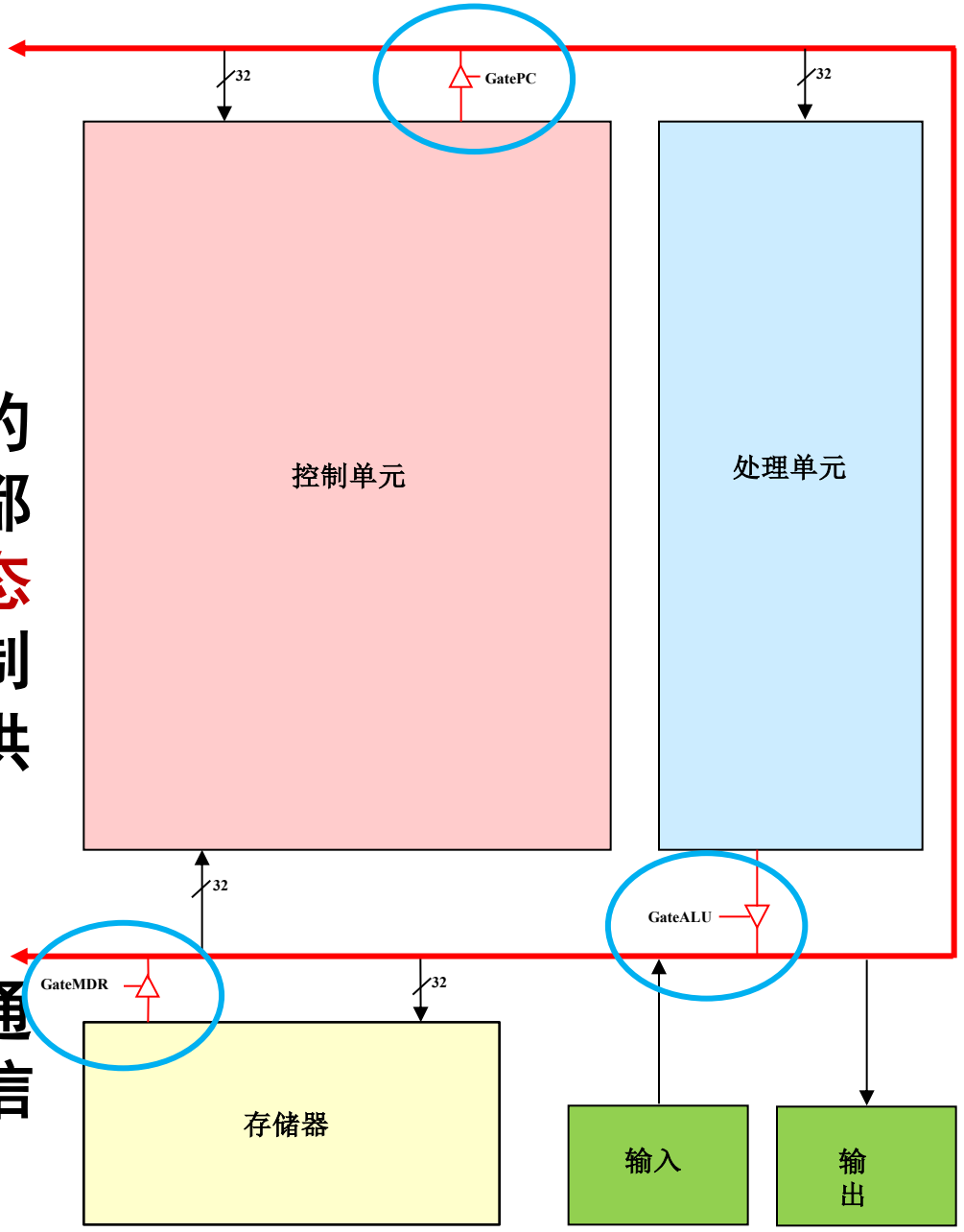
DLX总线

- DLX的总线由32根线和相关的电子元件组成
- 允许将32位信息从一个组件传输到另一个组件
- 数据/地址/控制信号
- 在总线上一次只可传输一个值

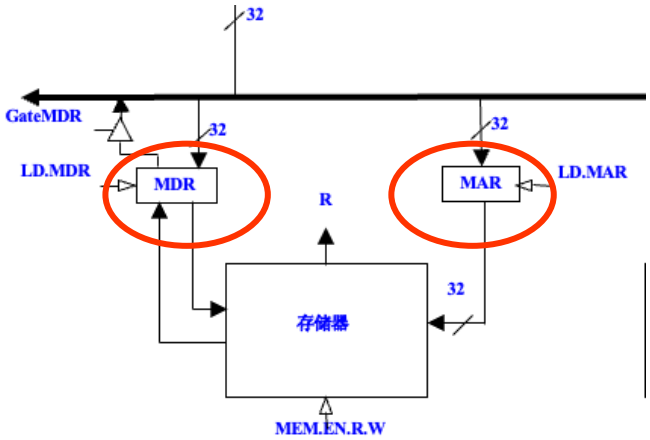


DLX总线

- 每一个提供数据给总线的组件在它的输入箭头后都有一个三角形（称为三态设备），使计算机的控制逻辑一次只允许一个提供者能提供信息给总线
- 从总线获得数据的组件通过将LD.x（加载使能）信号设为1（回忆门控锁存器），从而得到信息

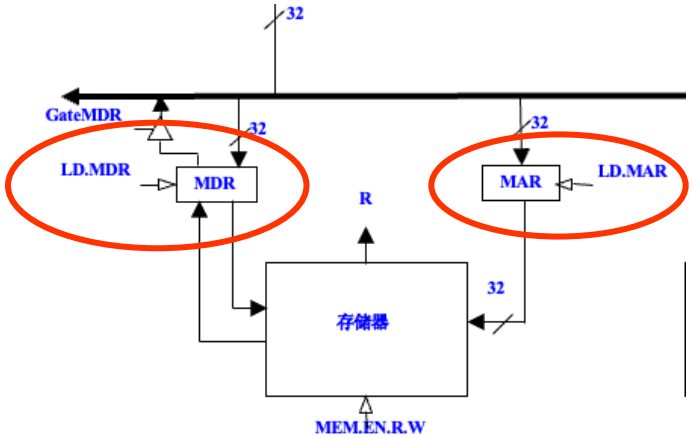


DLX存储器



- 主存地址寄存器(Memory Address Register, MAR)
  - 保存数据传输目的位置或者数据来源位置的地址
  - 32位，DLX的存储器的地址空间是2^{32}个存储单元
- 主存数据寄存器(Memory Data Register, MDR)
  - 保存要被写入地址单元或者从地址单元读入的数据
  - 32位，DLX字节可寻址，即每个单元包含8位
  - 在大多数情况下，MDR包含从MAR中的地址开始的4个连续单元的数据，有时包含的是MAR所指的单元中的数据（8位）符号扩展的结果（32位）

MAR和MDR

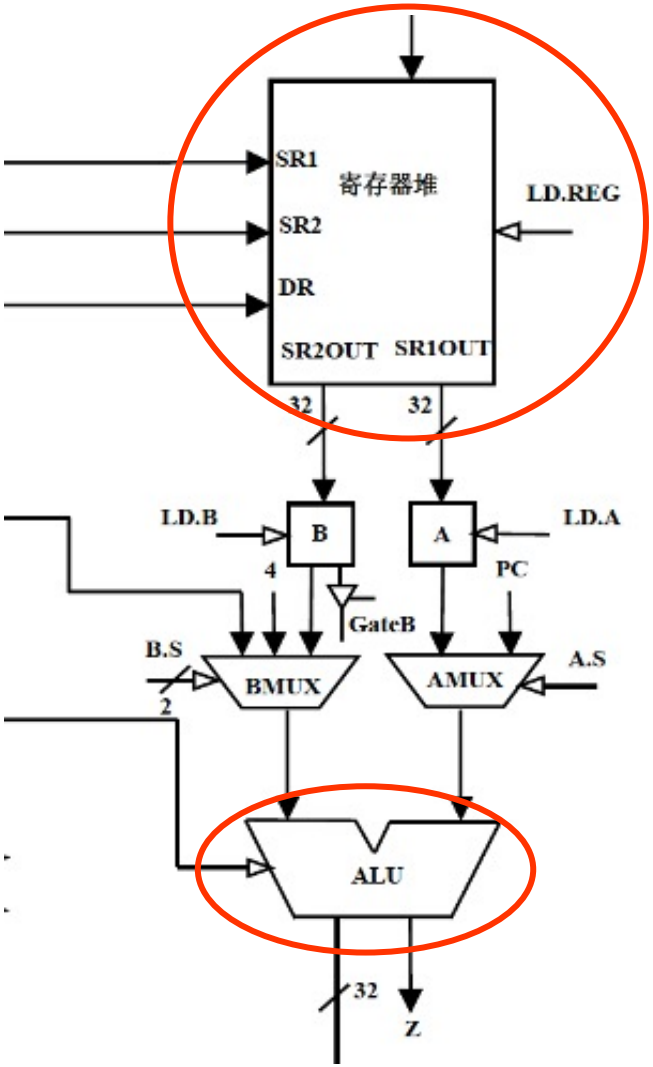


- 如果要读出某个存储单元中的内容，首先把它的地址存入地址寄存器（MAR），然后查询存储器，该地址所对应的存储单元的内容将会输出到数据寄存器（MDR）。

- 如果要写一个值到存储单元中，首先要把目的地址存入MAR，把值存入MDR中，然后设“写使能”信号为1，查询存储器，MDR里的信息就会被写到MAR中的地址所对应的存储单元里。

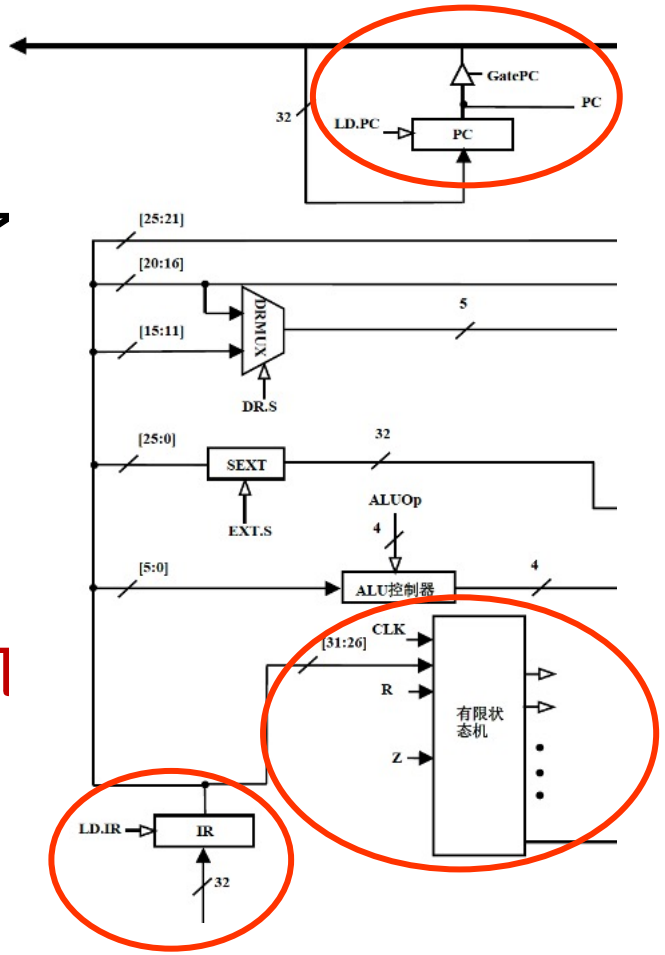
处理单元

- ALU和寄存器堆
- ALU可以做加法、减法、乘法、除法、与、或、异或、比较、移位等运算
- 32个整数寄存器、32个浮点寄存器
- DLX子集
  - 未包括整数乘法、除法及浮点数运算等操作
  - 也未包括浮点寄存器

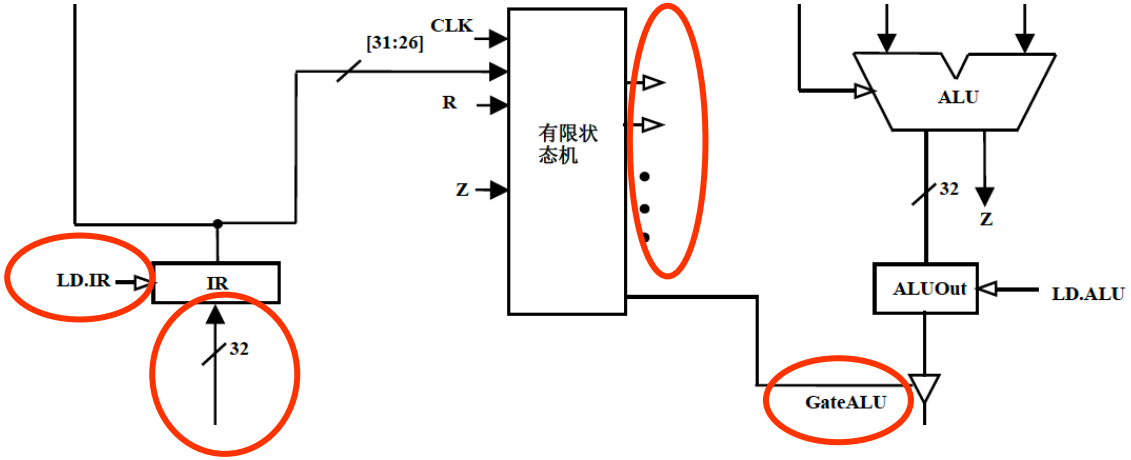


控制单元

- 所有管理计算机信息处理的组件
- 最重要是有限状态机，指挥所有行为
  - 有限状态机的一个输入是CLK，它说明了每个时钟周期持续的时间
  - 为了跟踪指令的处理阶段，控制单元还需要一个指令寄存器（Instruction Register，简称IR），用来保存正在处理的指令。IR也是有限状态机的一个输入，因为要处理的DLX指令决定了计算机要执行的行为。
- 程序计数器（PC）
  - 记录下一条要被执行的指令所在的地址



空心箭头



- 实心箭头表示沿着相应通路流动的是数据元素
- 空心箭头表示控制数据元素处理的控制信号
- 有限状态机的所有输出都是空心箭头
  - 控制了计算机的处理
  - LD. IR（1位），控制了当前时钟周期内，指令寄存器（IR）是否要从总线上加载新的指令
  - GateALU，决定ALUOut的值在当前时钟周期内是否被提供给总线

LW指令状态3

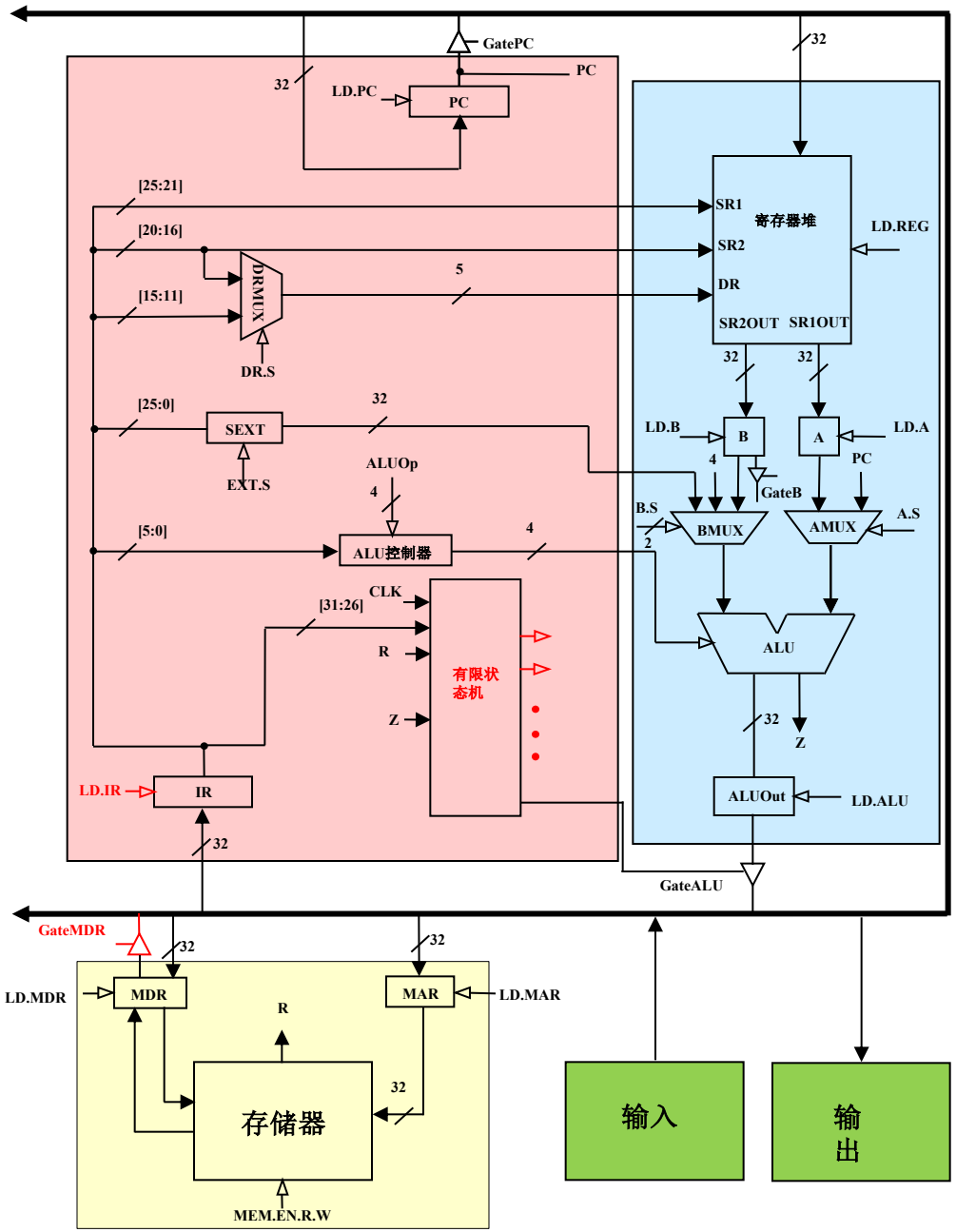
IR←MDR

- 控制信号

  GateMDR=1

  LD.IR=1

  ……



输入/输出设备

- 由键盘和显示器组成
  - 最简单的键盘需要两个寄存器，一个数据寄存器（KBDR），用来保存由键盘键入字符的ASCII码，和一个状态寄存器（KBSR），用来提供键盘键入字符的状态信息
  - 最简单的显示器同样需要两个寄存器，一个用来保存那些将被显示在显示器上的内容的ASCII码（DDR），另一个用来提供相关的状态信息（DSR）
- 第12章

指令获取阶段工作的元件