

复习

回顾

- 第一章 引言.ppt

自底向上——第一部分

- 第六章~第九章
- 理解——如何由简单的元件组成计算机，并能执行计算机语言编写的程序

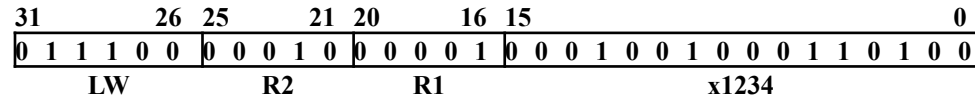
- 第六章：数据的机器级表示
 - 二进制整数/浮点数、十六进制、ASCII
 - ASCII表不需记忆
 - 表示及运算

- **第七章： 逻辑电路**
 - 晶体管级电路、门级电路
 - 译码器、多路选择器、存储器

- 第八章：冯·诺依曼（Von Neumann）模型
 - 基本组件
 - 指令处理（结合第九章指令格式、数据通路）

- 第九章：指令集结构
 - 指令格式
 - DLX指令集：指令格式、数据通路（结合第八章指令处理阶段理解）
 - 不需记忆操作码的机器表示

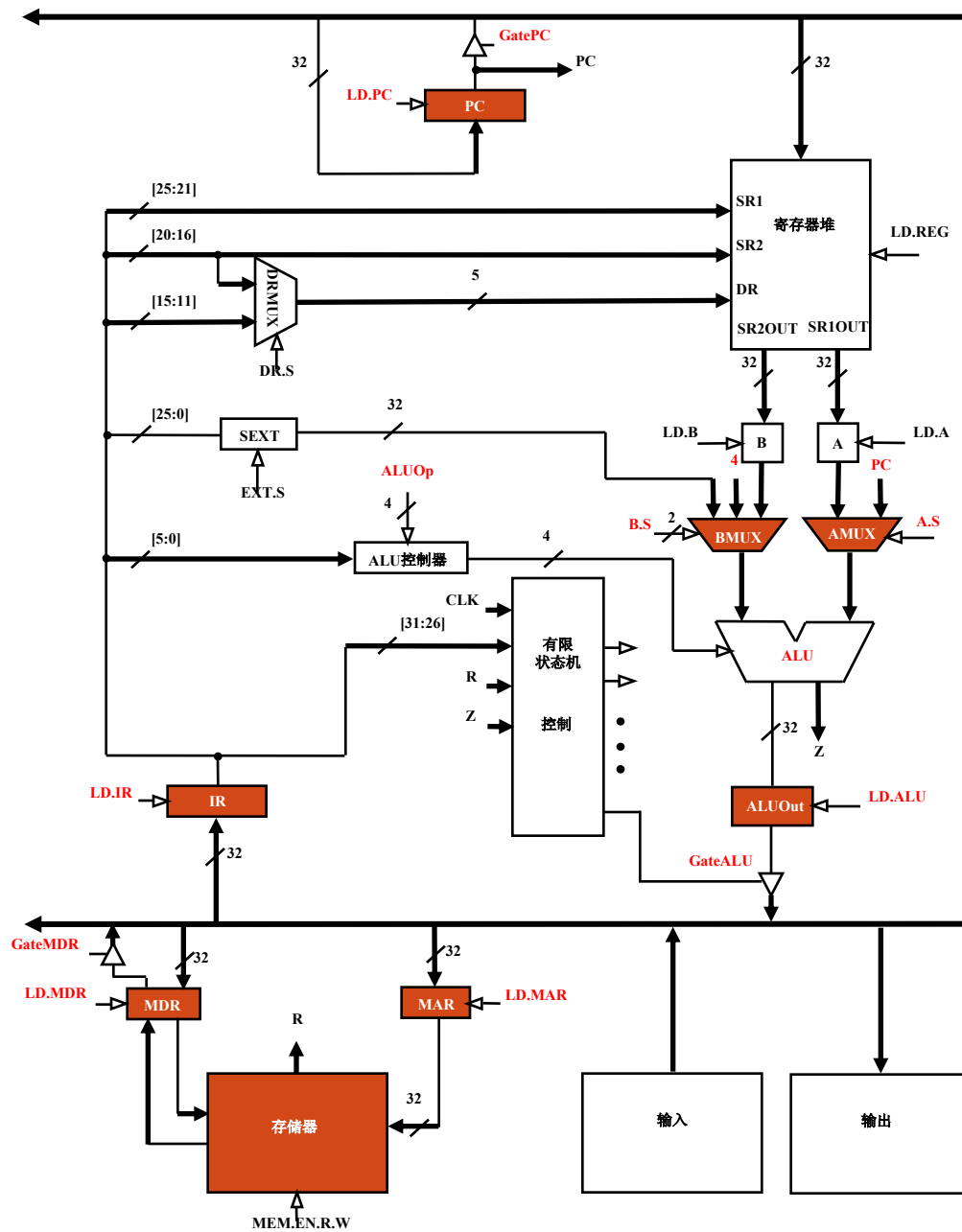
DLX数据通路



- 示例：
 - PC包含的是x40000000
 - x40000000~x40000003中保存的是011100 00010
00001 0001 0010 0011 0100

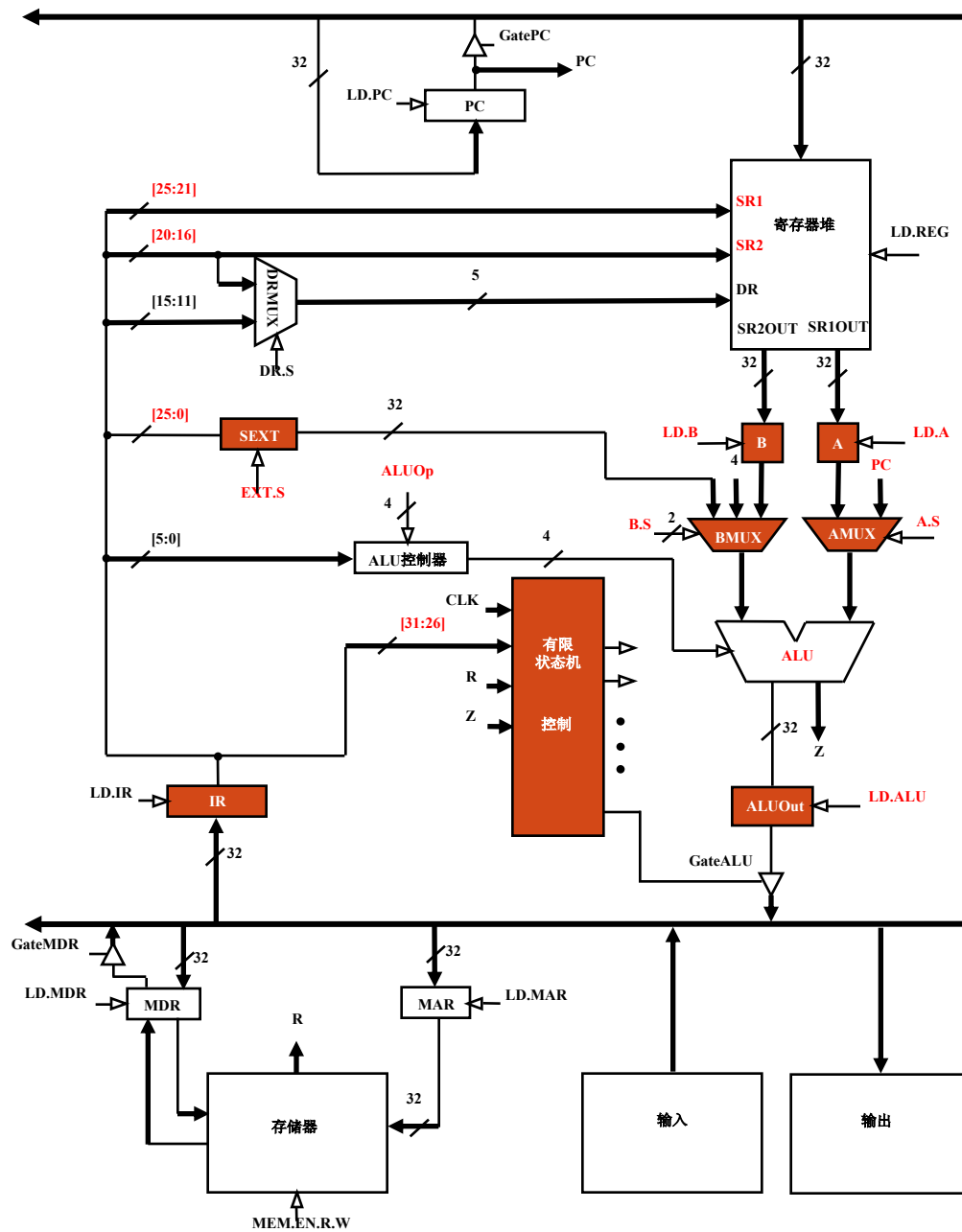
取指令

- 第一个时钟周期
 - PC中的内容通过总线被加载到MAR中，并且选择PC与4在ALU中执行加法运算
- 下一个时钟周期（如果存储器可以在一个时钟周期里提供信息）
 - 存储器被读取，指令011100 00010 00001 0001 0010 0011 0100被加载到MDR，PC加4的结果加载到PC（x4000 0004）
- 接下来的一个时钟周期
 - MDR中的值被加载到指令寄存器（IR）



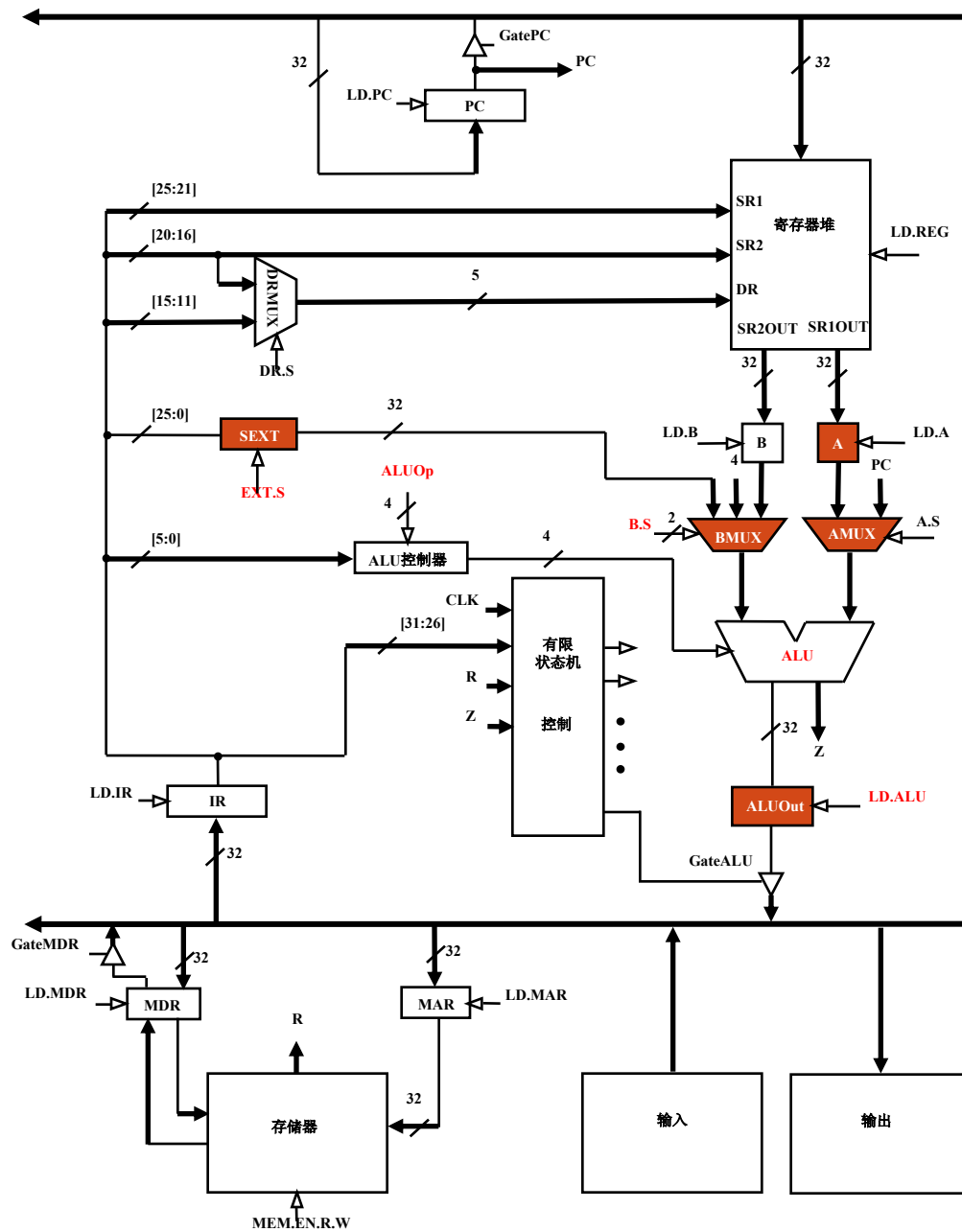
译码/取寄存器

- 下一周期
 - IR中的指令被译码（操作码是011100，为LW指令），使得控制逻辑发出正确的控制信号（空心箭头），从而控制指令的执行
 - 将IR[25:21]（即R2）读取出来，写到寄存器A中
 - 读取IR[20:16]的内容，写到寄存器B中
 - 在ALU中执行PC+SEXT(IR[15:0])，结果存储于ALUOut中



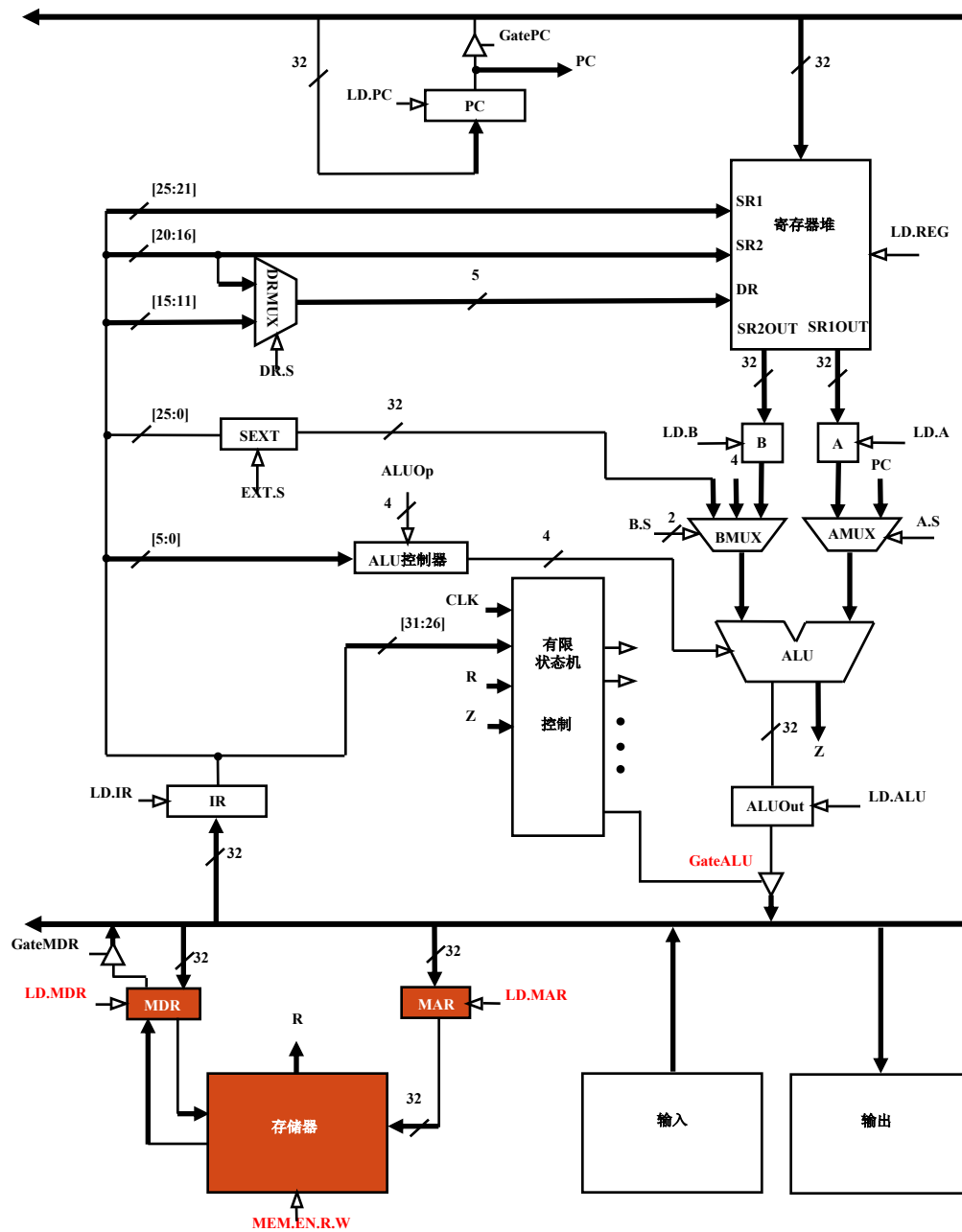
执行/**有效地址**/完成分支

- 下一周期
 - 有限状态机将AMUX和BMUX的选择信号A. S和B. S分别设为0和00，将EXT. S设为0（SEXT逻辑将执行IR[15:0]的符号扩展操作），选择来自寄存器A（即基址寄存器R2）和来自IR[15:0]的符号扩展的值；
 - 将ALUOp设为0001（加法），在ALU中进行加法运算，即计算“基址+偏移量”，形成有效地址；
 - 将LD. ALU设为1，结果存储于ALUOut寄存器中



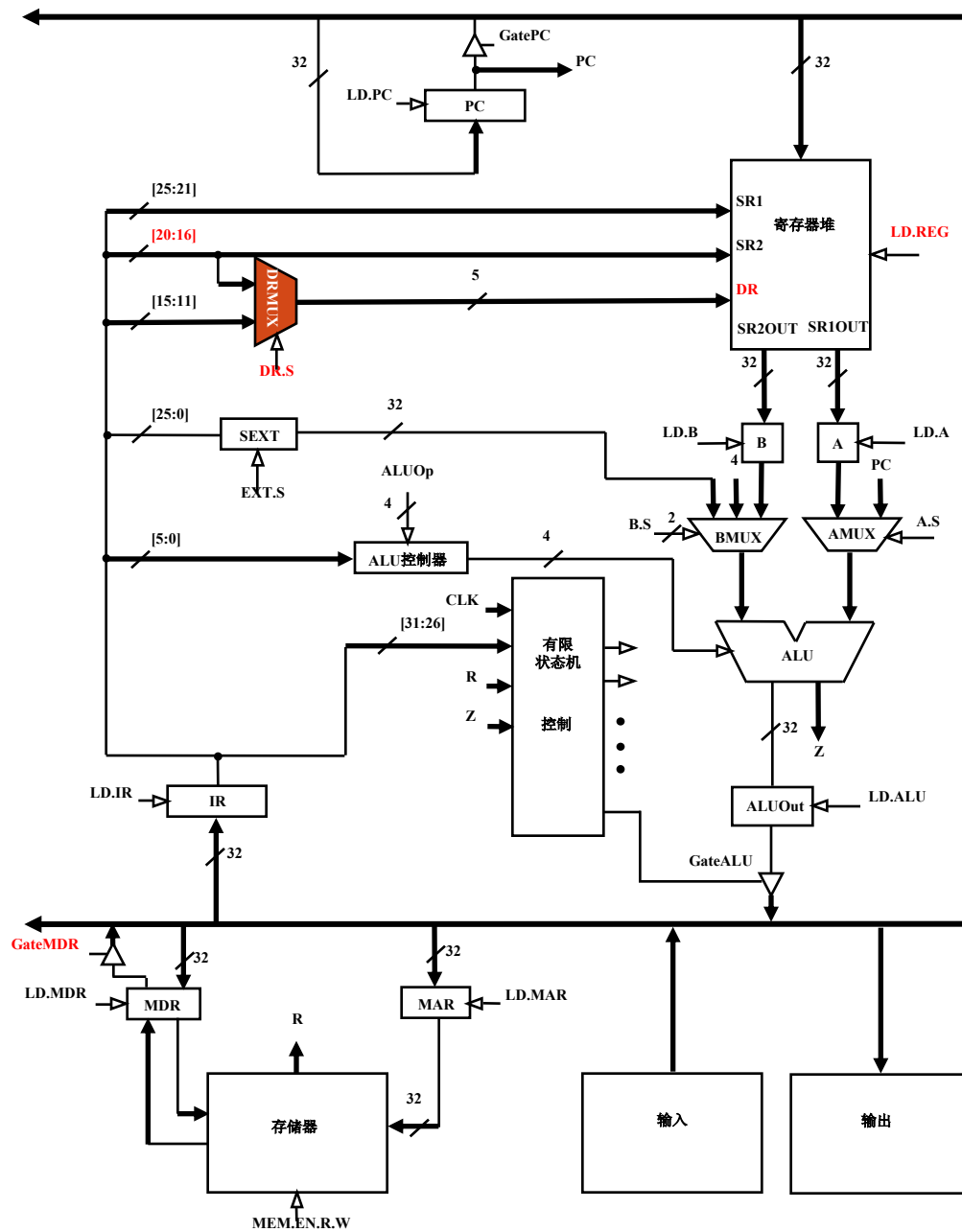
访问内存

- 下一周期（或多于一个，如果访问存储器需要多于一个时钟周期的话）
 - 有限状态机将GateALU和LD. MAR设为1，将ALUOut中的值通过总线传给MAR；
 - 将MEM. EN. R. W设为0（即读存储器），LD. MDR设为1，读取存储器，将以该地址开头的连续4个单元中的内容加载进MDR。



写回

- 最后一个周期
 - 有限状态机将DR. S设为1，选择IR的[20:16]作为目标寄存器（DR），即被加载的寄存器；
 - 将GateMDR和LD. REG设为1，在时钟周期结束时，MDR中的值被加载到R1中。



第二部分

- 一个计算机实例——DLX
- 第十章~第十四章
 - 汇编语言程序设计（包括子例程）
 - I/O基础（系统调用、轮询）
 - 汇编过程（两趟，符号表）

- 第十章：DLX机器语言
- 第十一章：DLX汇编语言
 - 汇编语言程序设计
 - 汇编语言处理
- 第十二章：输入和输出
 - I/O基础（内存映射、轮询）
- 第十三章：自陷（系统调用）
- 第十四章：子例程

第三部分

- 使用——一种高级计算机语言（C语言）编写比较复杂的程序
- 理解——这些程序是如何在计算机这一复杂机器内部执行的
- 第二章~第五章
 - C语言基础
 - 类型、变量、运算符、控制结构
- 第十五、十六、十七章：函数、指针和数组
 - C语言程序设计
 - 高级语言处理（C-DLX）
 - 不需记忆寄存器分配规则

- **考试记律**