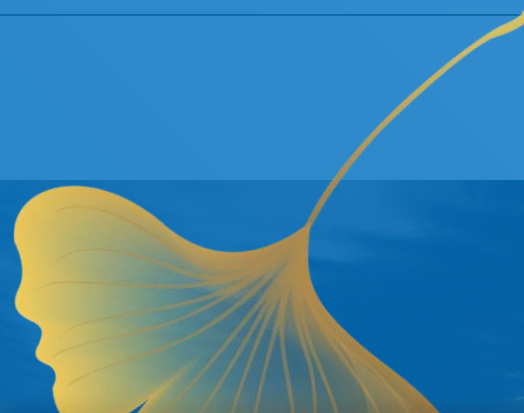




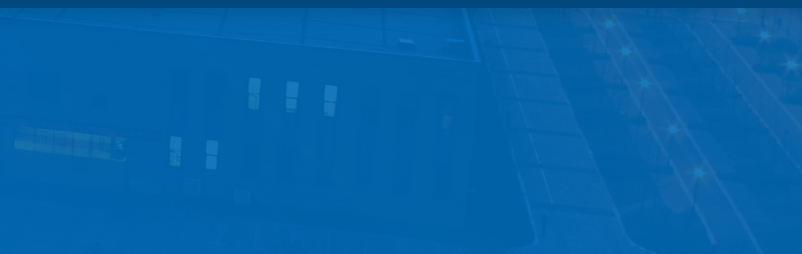
南京大学120周年校庆
120th ANNIVERSARY
NANJING UNIVERSITY
1902 - 2022

1902 2022



课程回顾

Course Review





课程回顾

C1-C8:

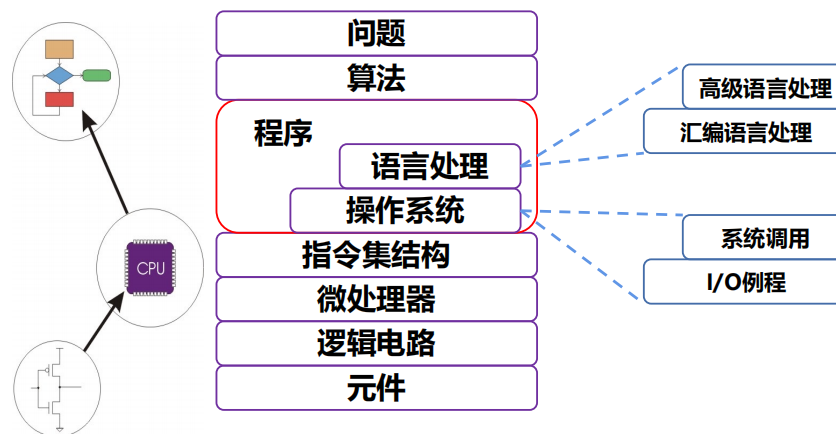
- 理解——如何由简单的元件组成计算机，并能执行计算机语言编写的程序
- 期末考试：笔试（40%-45%）
- 题型：计算、简答（理解与分析、设计）



课程回顾

C1（计算系统概述）：

- 什么是计算机？
 - 通用电子数字
 - 图灵机
- 什么是计算机系统？
 - 冯诺依曼模型
- 什么是计算系统？
 - 七个层级





课程回顾

C2-3（数据的机器级表示，约8%）

:

- 位和数据类型
- 进位计数制
- 整数数据类型
- 浮点数数据类型
- ASCII编码
- C语言中的数据类型

ASCII表不需记忆

掌握数据的表示方法和类型

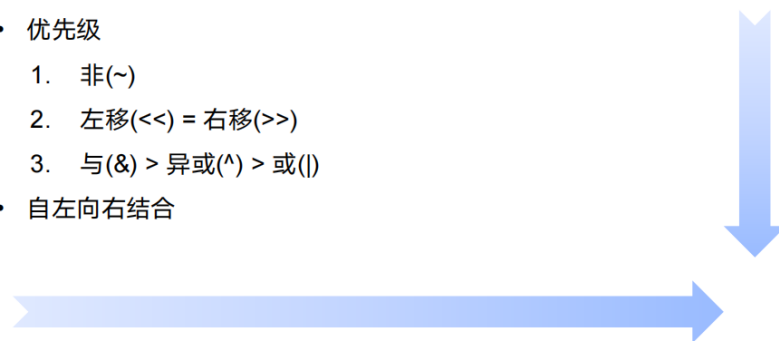


课程回顾

C4（数据的运算，约8%）：

- 按位逻辑运算
 - 与、或、非、异或、左移、右移
 - 优先级与结合性
- C语言的逻辑运算
- 算数运算
 - 无符号/有符号（补码）整数的加减乘除运算
 - 定点/浮点数加减乘除运算
 - IEEE 754-单/双精度浮点数

- 优先级
 1. 非(~)
 2. 左移(<<) = 右移(>>)
 3. 与(&) > 异或(^) > 或(|)
- 自左向右结合





课程回顾

C5（数字逻辑电路，约8%）：

- 晶体管级电路、门级电路
- 组合逻辑电路
 - 译码器、多路选择器、加法器
- 时序逻辑电路
 - 锁存器、触发器、存储器
- 电路设计/分析

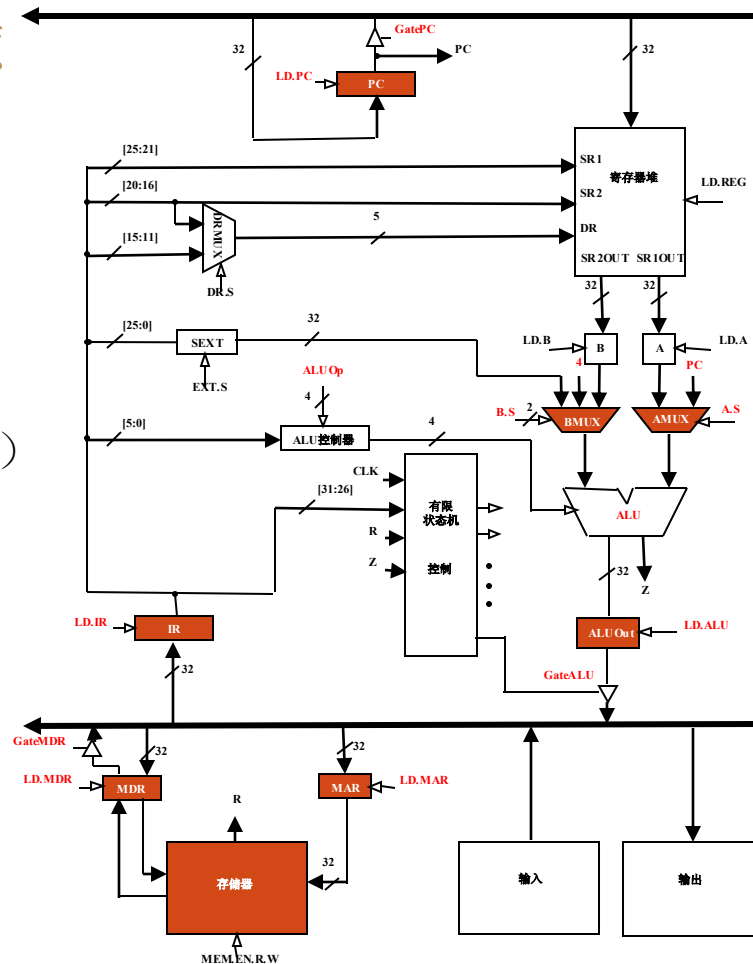


- 冯·诺依曼（Von Neumann）模型
- 中央处理器
- 指令处理（结合C7指令格式、数据通路）

- 按照DLX指令执行的步骤，将处理指令所需的操作划分为以下阶段：

- 取指令 (Instruction fetch)
- 译码/取寄存器 (Instruction decode/Register fetch)
- 执行/有效地址/完成分支 (Execution/Effective address/Branch completion)
- 访问内存 (Memory access)
- 存储结果 (Write-back)

每条DLX指令需要其中的3到5个阶段





回顾

C7-8（指令集架构&机器语言程序设计

- 指令格式、分类
- DLX指令集：指令格式、分类、数据通路
 - 结合指令处理的五个阶段
- 机器语言程序的设计与理解问题

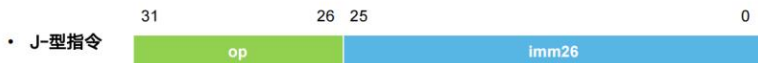
指令分类（按操作个数，0-4）



- 操作码op为000000，SR1、SR2源操作数，DR目标操作数，funct为函数（实际操作）



- 一般SR1和立即数为源操作数，DR目标操作数；条件转移操作SR1和DR为源操作数



- 跳转指令，[25:0]立即数为计算跳转目标地址的一部分

	31	26	25	21	20	16	15	11	10	6	5	0
ADD	000000		SR1		SR2		DR		未用			000001
ADDI	000001		SR1		DR				Imm16			
SUB	000000		SR1		SR2		DR		未用			000011
SUBI	000011		SR1		DR				Imm16			
AND	000000		SR1		SR2		DR		未用			001001
ANDI	001001		SR1		DR				Imm16			
OR	000000		SR1		SR2		DR		未用			001010
ORI	001010		SR1		DR				Imm16			
XOR	000000		SR1		SR2		DR		未用			001011
XORI	001011		SR1		DR				Imm16			
LHI	001100		未用		DR				Imm16			
SLL	000000		SR1		SR2		DR		未用			001101
SLLI	001101		SR1		DR				Imm16			
SRL	000000		SR1		SR2		DR		未用			001110
SRLI	001110		SR1		DR				Imm16			
SRA	000000		SR1		SR2		DR		未用			001111
SRAI	001111		SR1		DR				Imm16			
SLT	000000		SR1		SR2		DR		未用			010000
SLTI	010000		SR1		DR				Imm16			
SLE	000000		SR1		SR2		DR		未用			010010
SLEI	010010		SR1		DR				Imm16			
SEQ	000000		SR1		SR2		DR		未用			010100
SEI	010100		SR1		DR				Imm16			
LB	010110		SR1		DR				Imm16			
SB	010111		SR1		DR				Imm16			
LW	011100		SR1		DR				Imm16			
SW	011101		SR1		DR				Imm16			
BEQZ	101000		SR1		未用				Imm16			
BNEZ	101001		SR1		未用				Imm16			
J	101100								PCOffset26			
JR	101101		SR1		未用				未用			
JAL	101110								PCOffset26			
JALR	101111		SR1		未用				未用			
TRAP	110000								Vector26			

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

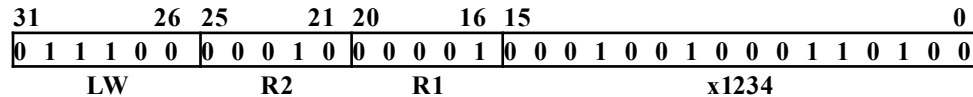
- 第六章：数据的机器级表示
 - 二进制整数/浮点数、十六进制、ASCII
 - ASCII表不需记忆
 - 表示及运算
 - 10%

- **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
 - 10%

- 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）
 - 10%

- 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，5%）
 - 不需记忆操作码的机器表示
 - 10%

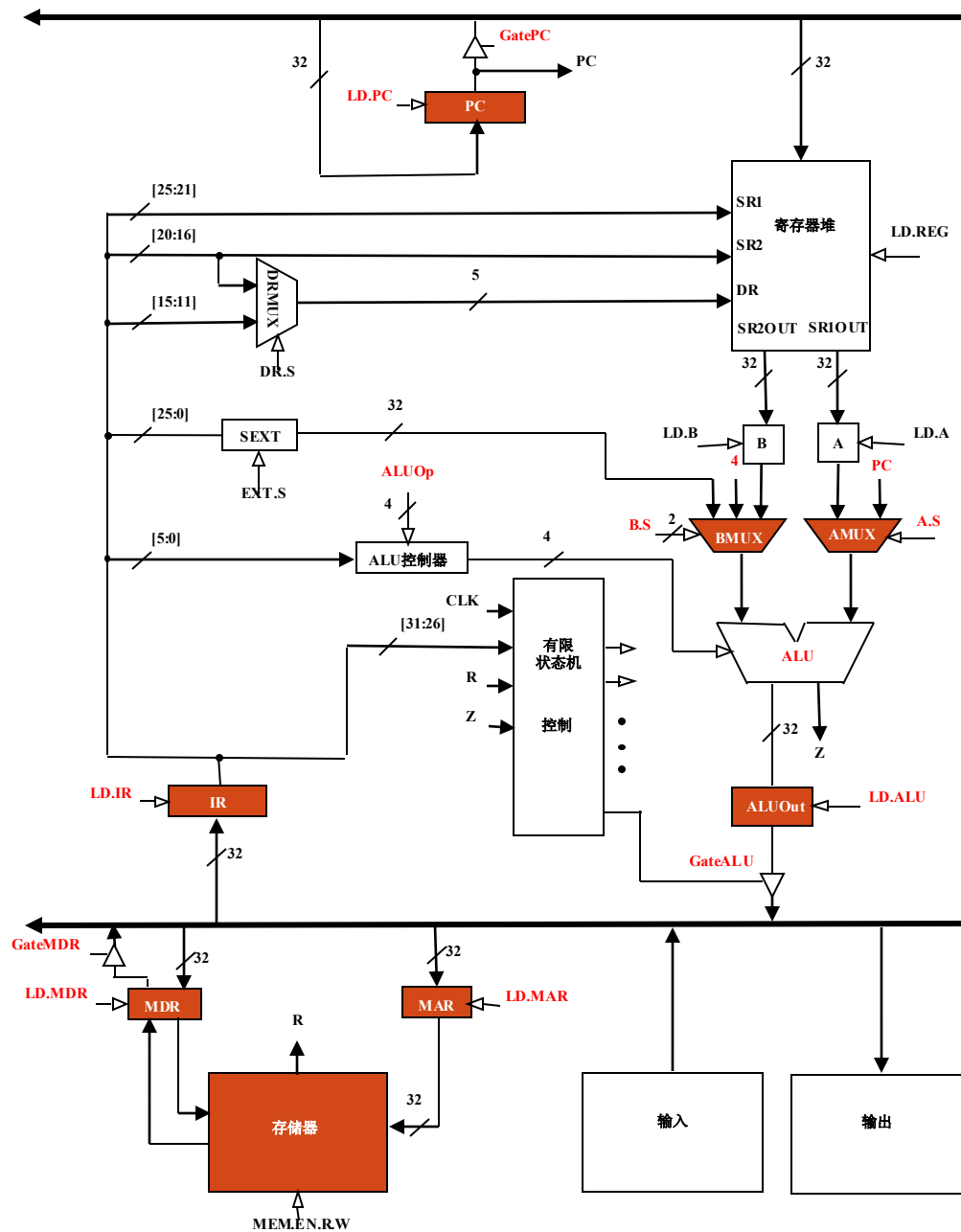
DLX数据通路



- 示例：
 - PC包含的是x400000000
 - x400000000~x400000003中保存的是011100 00010
00001 0001 0010 0011 0100

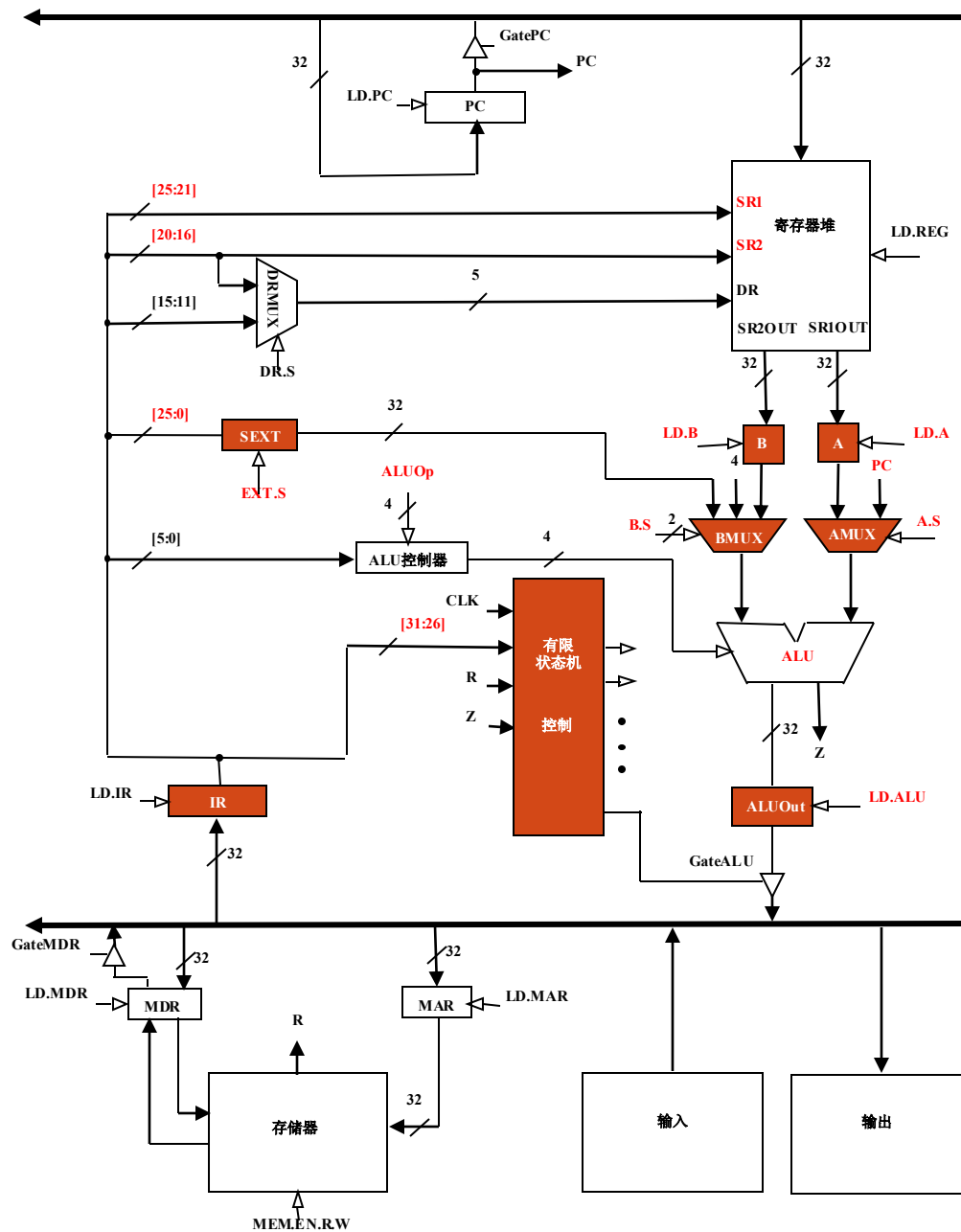
取指令

- 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）



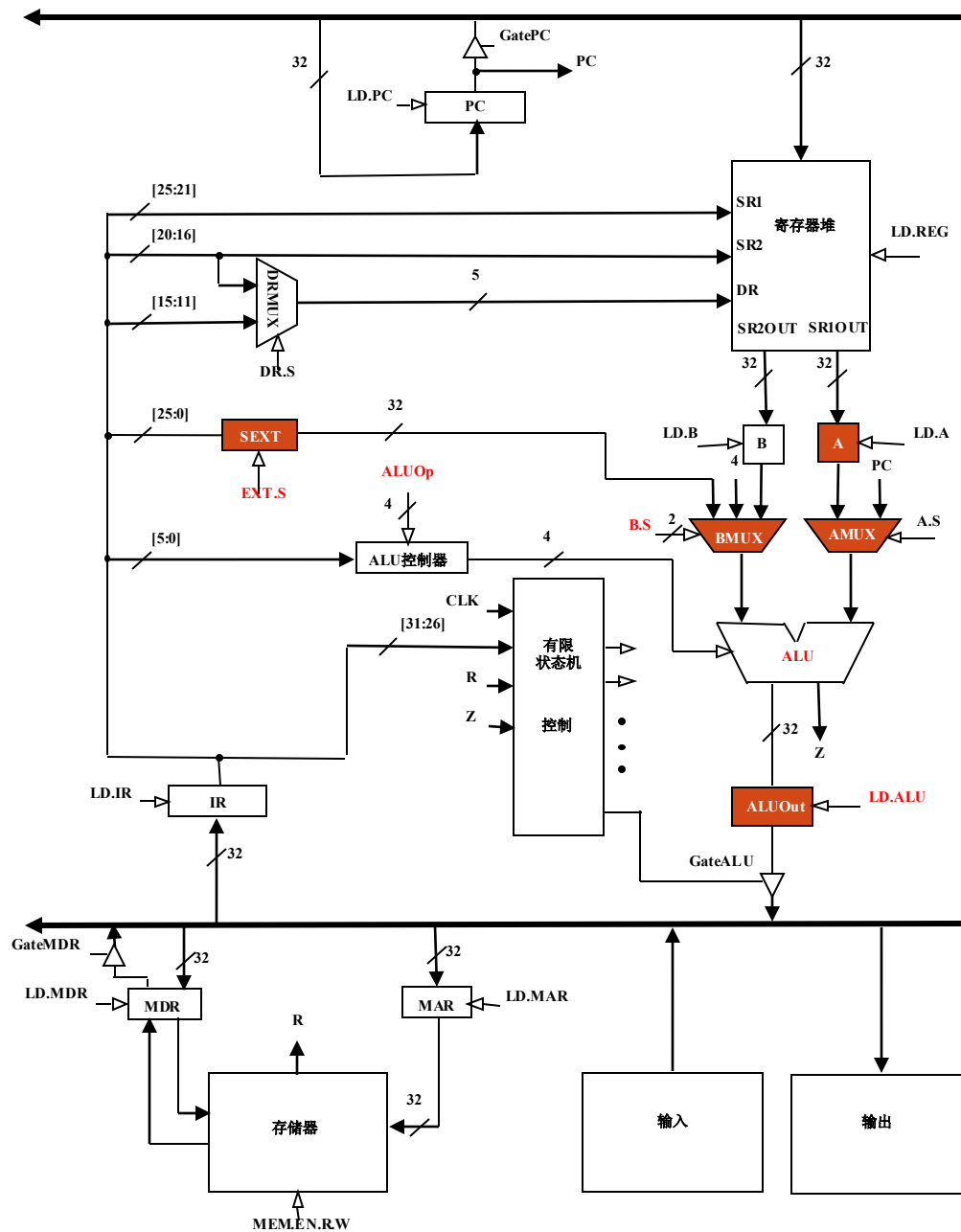
译码/取寄存器

- 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 $PC + \text{SEXT}(\text{IR}[15:0])$ ，结果存储于ALUOut中



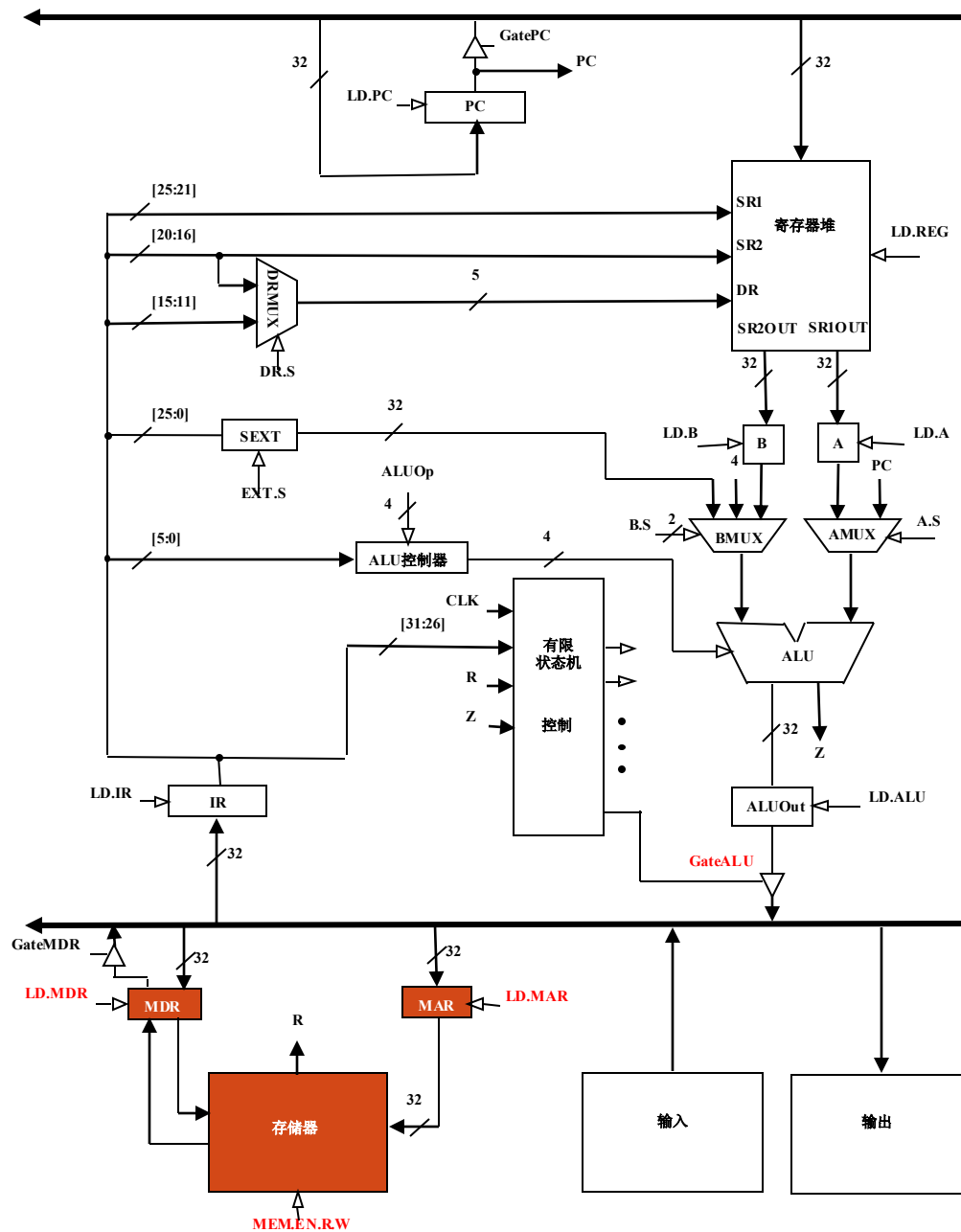
执行/**有效地址**/完成分支

- 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



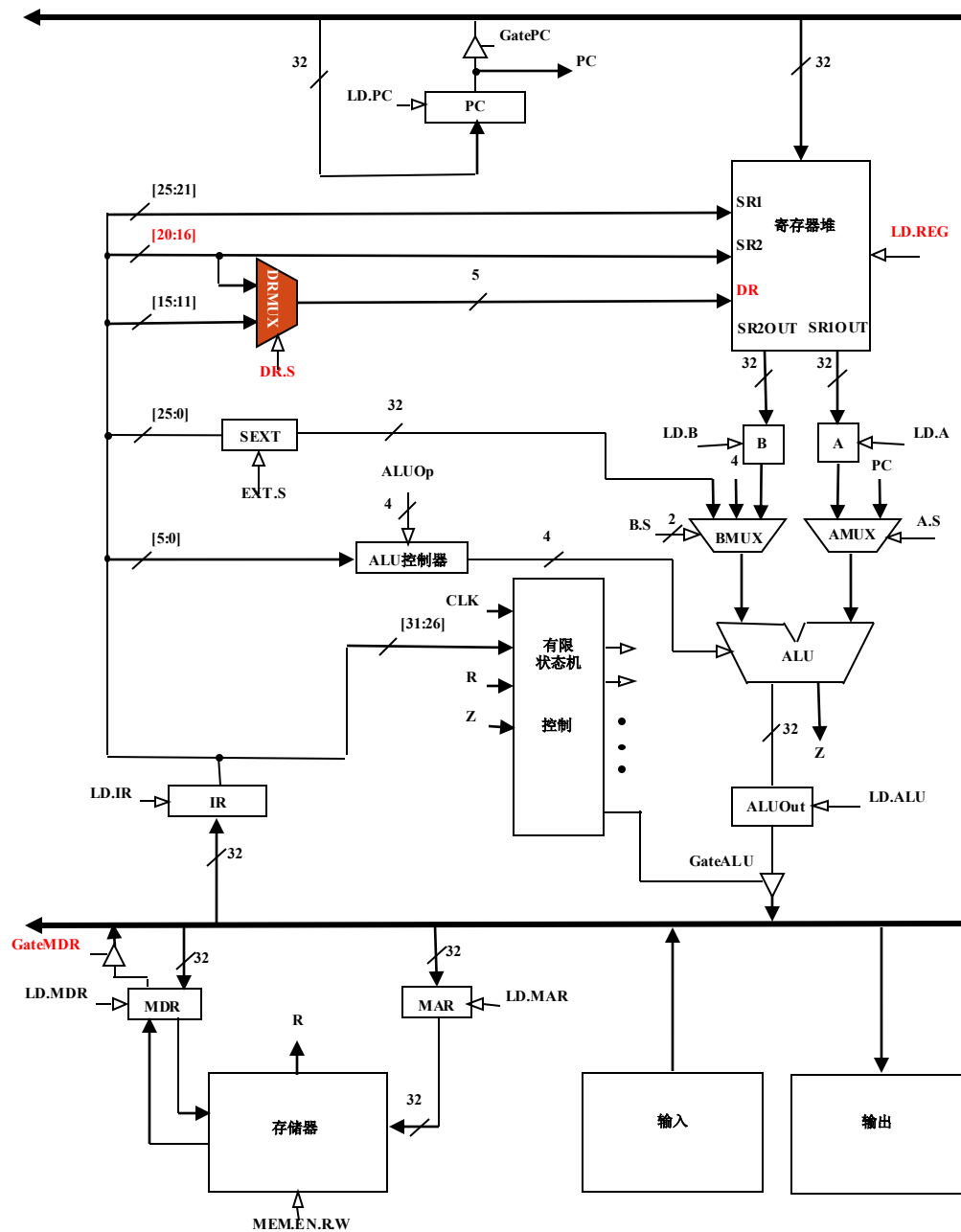
访问内存

- 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（上机，笔试填空）
 - 汇编语言处理
 - 15%
- **第十二章：输入和输出**
 - I/O基础（内存映射、轮询）
 - 5%
- **第十三章：自陷（系统调用）**
 - 5%
- **第十四章：子例程**
 - 5%

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
 - 10%
- 第十五、十六、十七章：函数、指针和数组
 - C语言程序设计（上机，笔试填空）
 - 高级语言处理（C-DLX，10%）
 - 不需记忆寄存器分配规则
 - 20%

- **考试记律**