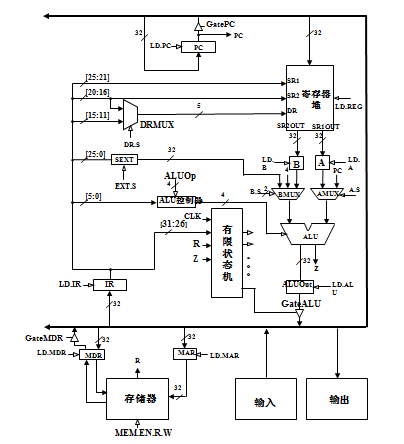
**第八章 冯·诺依曼模型**

1.**数据通路**：计算机内部用于处理信息的所有元件总和。

2.**DLX子集数据通路**



3.**PC**：**程序计数器**（**指令指针**），32位，存储下一条指令所在地址的寄存器。

4.**IR**：**指令寄存器**，32位。保存正在处理的指令。

5.多路选择器

6.DLX数据通路采用**总线结构**，**多时钟周期**实现方案。

7.**总线**：存储器和处理器，处理器和I/O设备间的通信。用黑色箭头框表示。

优点：功能多成本低。

缺点：产生通信瓶颈（总线上一次只传输一个值）。

8.**三态设备**（三角形）：使计算机的控制逻辑一次只允许一个提供者向总线提供信息。位于每一个提供数据给总线的组件的输入箭头处。

9.**LD.**X（**加载使能**）：使信息允许加载时为1，不允许加载时为0。

10.**Gate**.X：使数据与总线相连。

11.实心箭头：通信流动数据元素。

12.空心箭头：控制数据元素处理的控制信号。

13.**冯诺依曼模型主要思想**：把程序和数据都作为一个二进制序列存储在存储器里，在控制单元的引导下一次执行一条指令。

14.由指令组成的程序和程序所需的数据位于存储器中。

15.**MAR**（**地址寄存器**）：32位（DLX有232个存储单元）。

16.**MDR**（**数据寄存器**）：32位，存储连续4个单元的数据，或单元数据符号扩展的结果。

17.**处理单元执行信息的实际处理。**DLX只包含ALU和寄存器堆。

18.**ALU**：**算数逻辑单元**。

19.**字长**：ALU正常处理的信息量大小。ALU有32位字长。

20.**字**：信息中的每一个元素。**1个字=4个字节**，32位。

21.**寄存器堆**：靠近ALU提供少量存储空间用于临时存取数据的一组寄存器。

避免对存储器不必要的长时间访问。

每个寄存器包含一个字。

共32个通用寄存器（R0，R1，…，R31），32个浮点寄存器（F0，F1，…，F31）。

22.**控制单元指挥信息的处理。**

23.指令包括操作码和操作数。

24.寻址模式：计算机将要读取/存储的存储单元的地址的机制。

25.**多周期实现方案**

①**取指令**

查询存储器，把指令放进MDR。

把MDR的内容加载入IR。

②**译码**：识别指令而确定下一步。

**取寄存器**：为后面的阶段进行获取操作数的操作。（有的操作结果后面不会用到，但不浪费时间，原因：操作同时进行）

③**执行**：对上一阶段得到的寄存器的值执行算数/逻辑运算。

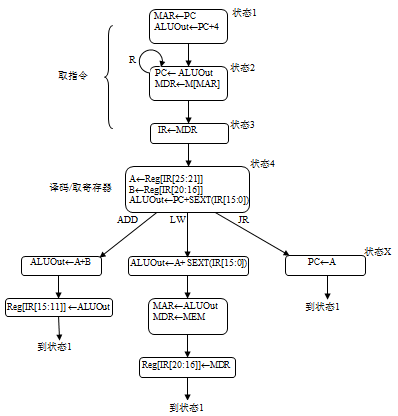
**有效地址**：计算出处理指令所需存储单元的地址。

**完成分支**：完成分支跳转。

④**访问内存**：获取内存中的数据。

⑤**存储结果**：将结果写入指定目标。

26.DLX状态图（需要多个时钟周期，一个状态内的操作同时执行）



**状态①**

**MAR←PC**

有限状态机使Gate.PC和LD.MAR设为1，PC与总线相连，当前时钟周期结束时将总线内容写入MAR。

**ALUOut←PC+4**

有限状态机使A.S（AMUX选择信号）设为1，选择来自PC的输入；

有限状态机使B.S（BMUX选择信号）设为01，选择输入4；

有限状态机使ALUOp设为0001，在ALU中执行加法；

有限状态机使LD.ALU设为1，结果存储与ALUOut中。

**状态②**

**PC←ALUOut**

有限状态机使Gate.ALU和LD.PC设为1，将ALUOut的值写入PC。

**MDR←M[MAR]**

有限状态机使MEM.EN.R.W（写使能）设为0，LD.MDR设为1，读存储器。

**R**

读存储器需要多个时钟周期，读取时就绪信号R设为0，读取结束时设为1。

**状态③**

**IR←MDR**

R为1时，有限状态机使Gate.MDR和LD.IR设为1，当前时钟周期结束时，IR被写入。

**状态④**

**A←Reg[IR[25:21]]**，**B←Reg[IR[20:16]]**

有限状态机使LD.A和LD.B设为1，从IR[25:21]和IR[20:16]分别获得源操作数传给ALU的A、B寄存器。

**ALUOut←PC+SEXT(IR[15:0])**

有限状态机使A.S设为1，选择来自PC的输入；

有限状态机使EXT.S设为0，B.S设为00，选择IR[15:0]符号扩展结果；

有限状态机使ALUOp设为0001，在ALU中执行加法；

有限状态机使LD.ALU设为1，结果存储于ALUOut中。

27.**时钟电路**

|  |  |
| --- | --- |
| Q=1 | 时钟电路输出与时钟发生器输出一样 |
| Q=0（清0） | 时钟电路输出为0，即停止指令运行。（HALT） |

