目录

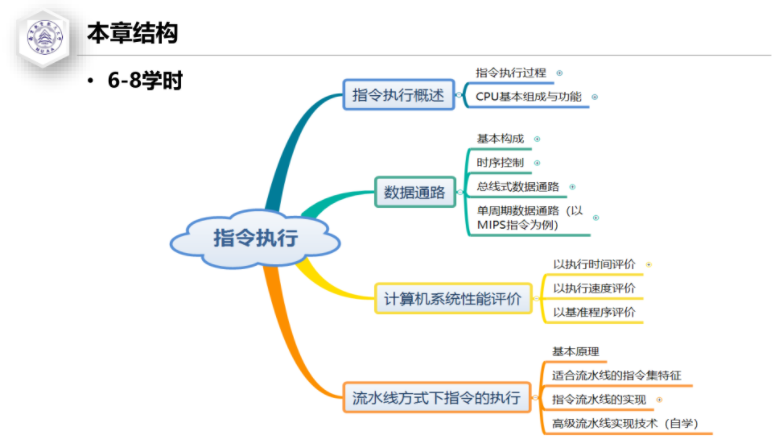
[指令执行概述 1](#_Toc71553729)

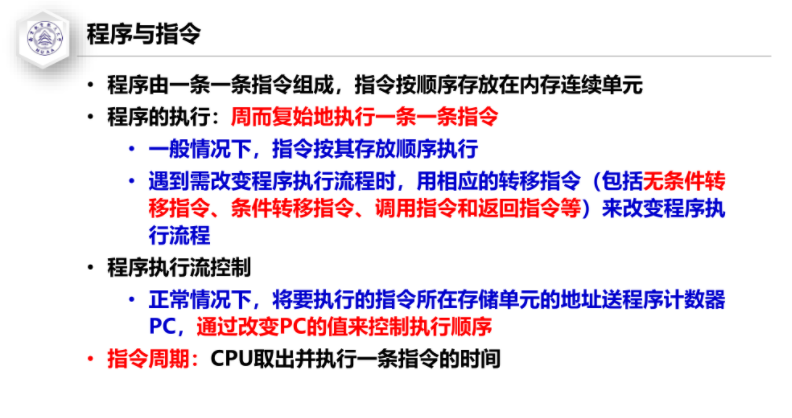
[数据通路 5](#_Toc71553730)

[总线式数据通路 8](#_Toc71553731)

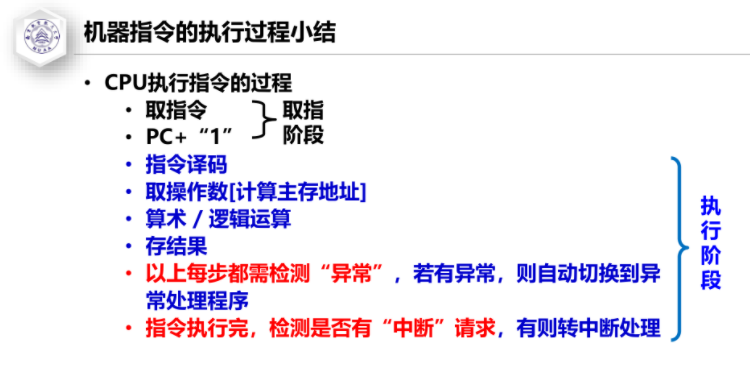
[单周期数据通路（MIPS为例） 11](#_Toc71553732)

指令执行概述

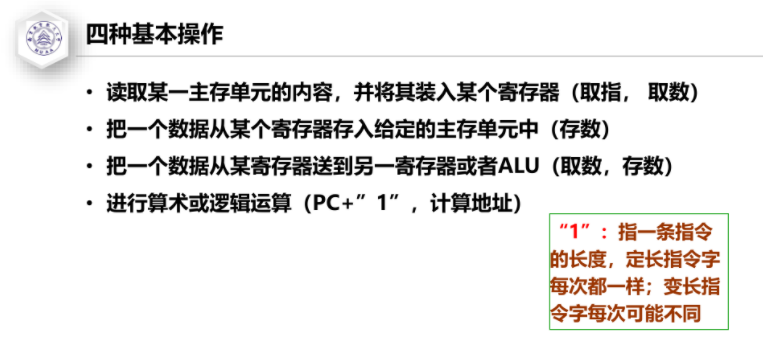




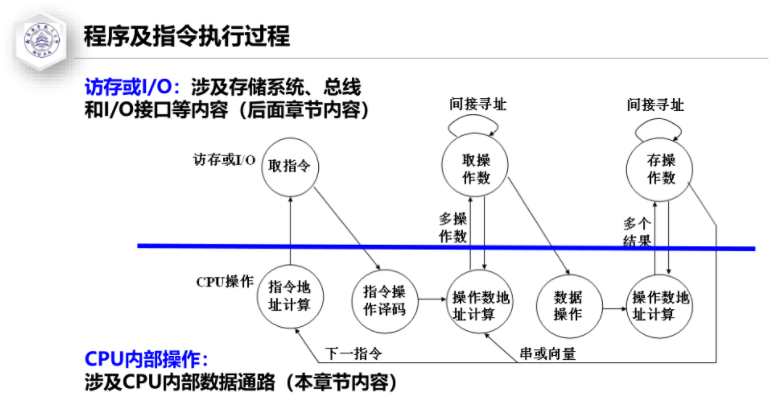


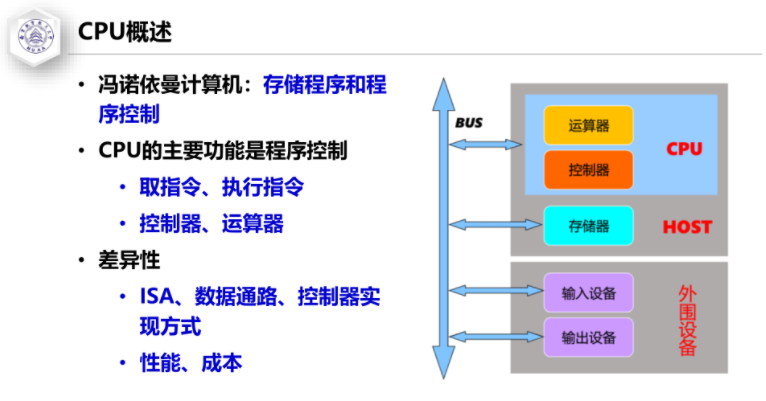


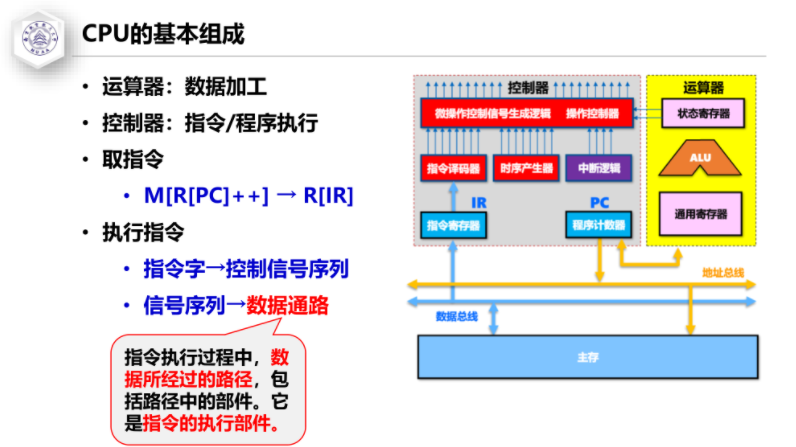
这里的”1”指的是指令长度，MIPS中是4b，IA-32中是不定长

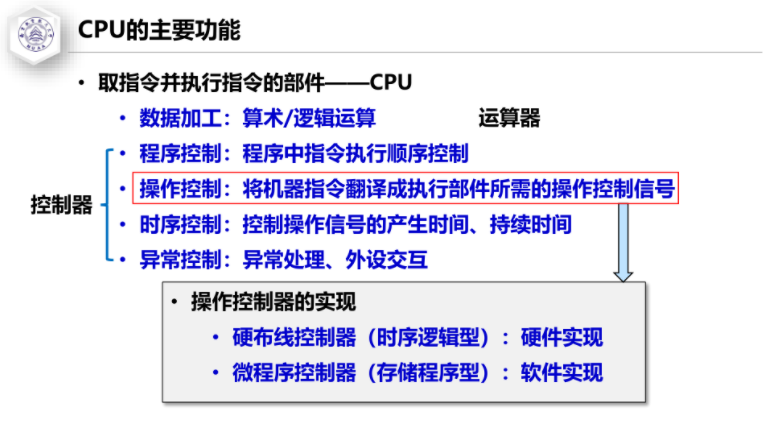


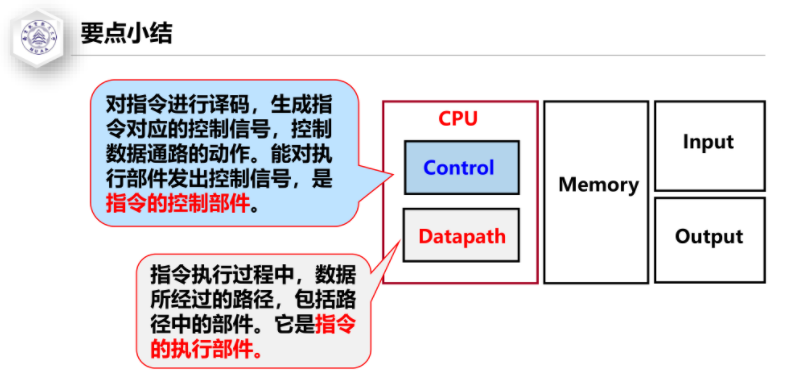
1. 主存单元 -> 寄存器
2. 寄存器 -> 主存单元
3. 寄存器之间
4. 算术或逻辑运算



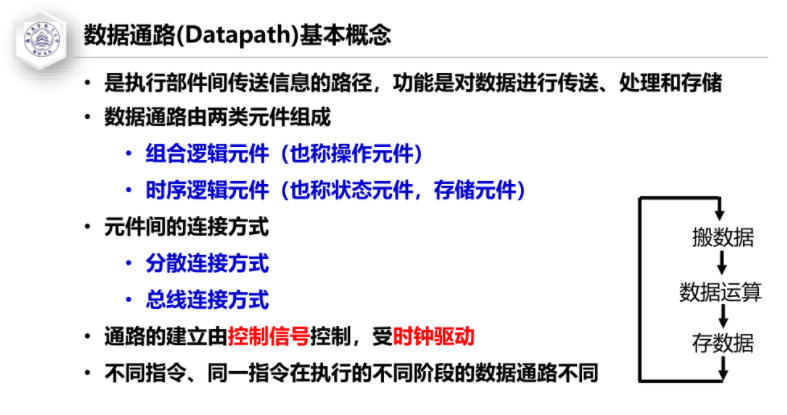


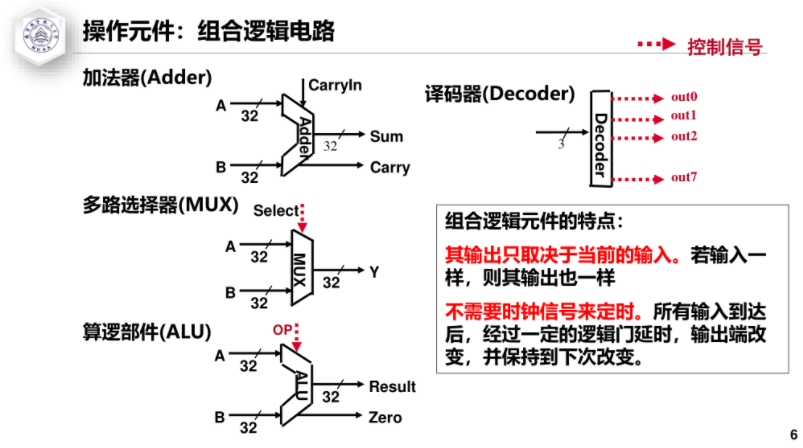


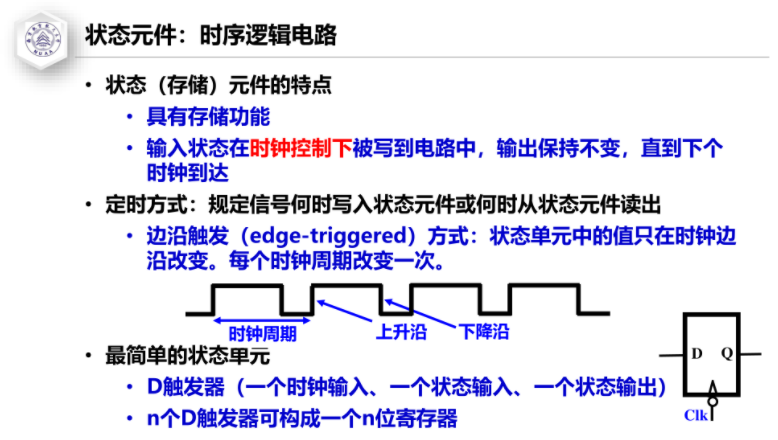


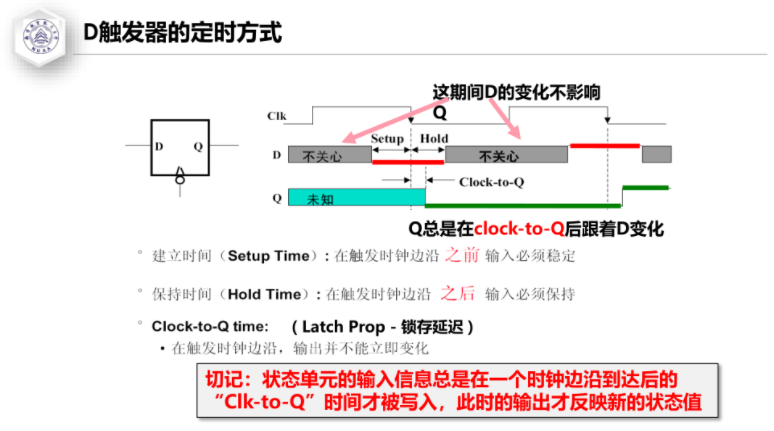


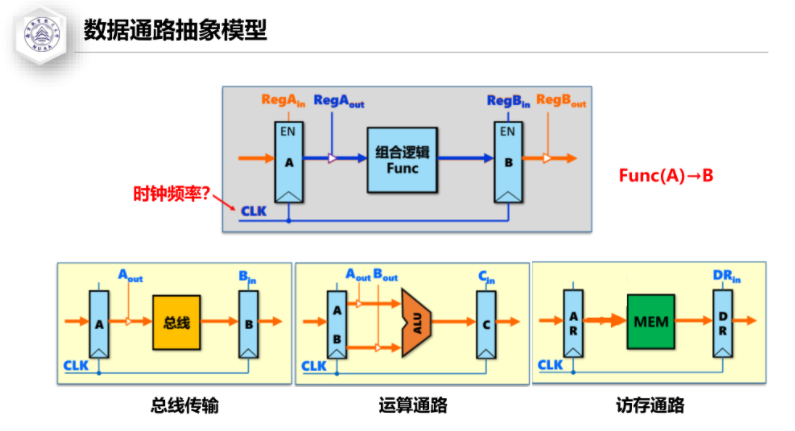
数据通路

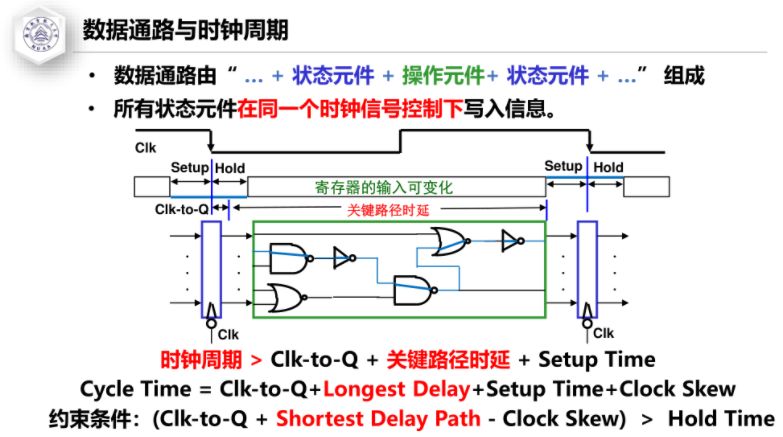






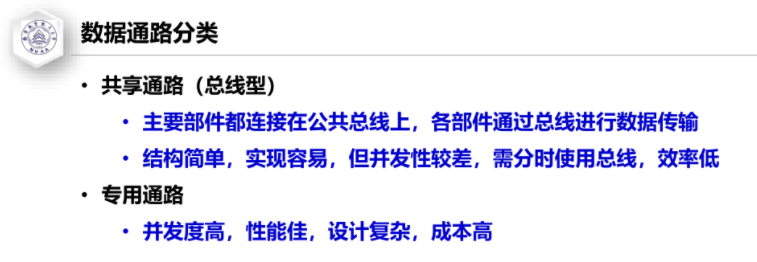






时钟周期的倒数为时钟频率

用最长的路径时延算时钟周期



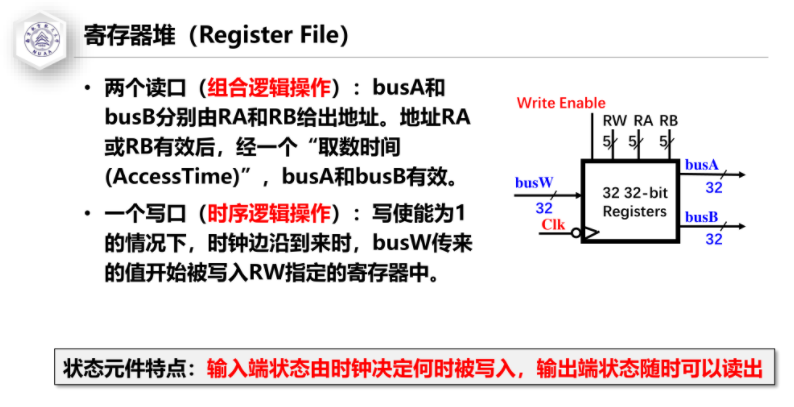
有的时候专用通路可以看作是多个总线

总线式数据通路

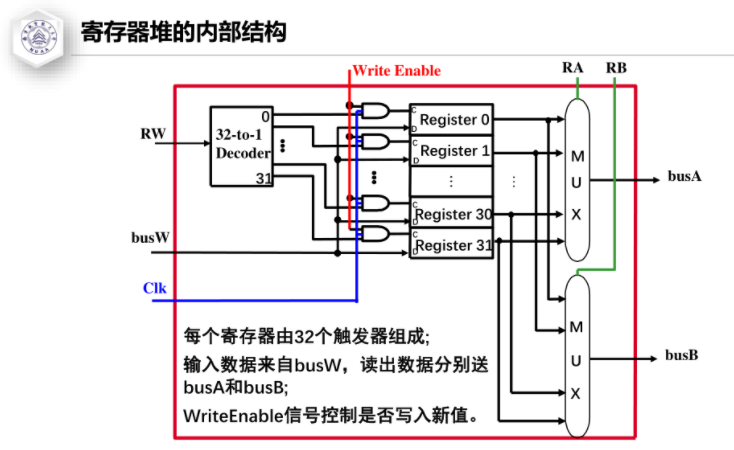


三态门：开关 高电平时可以送数据 低电平时不行

输出带三态门的寄存器：输入输出都有开关，与总线相连

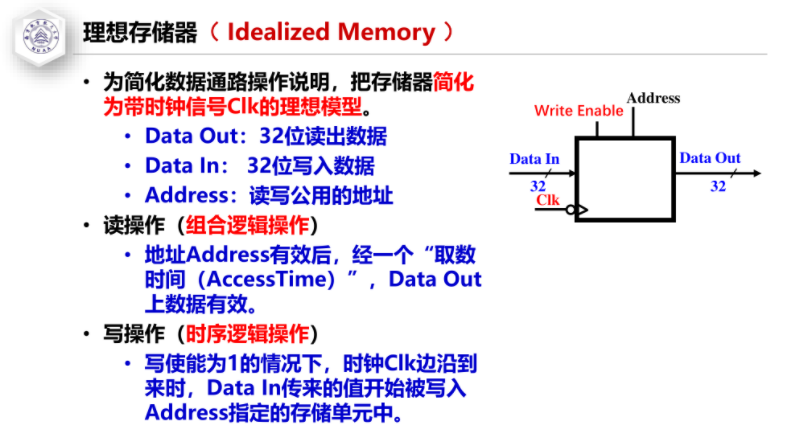


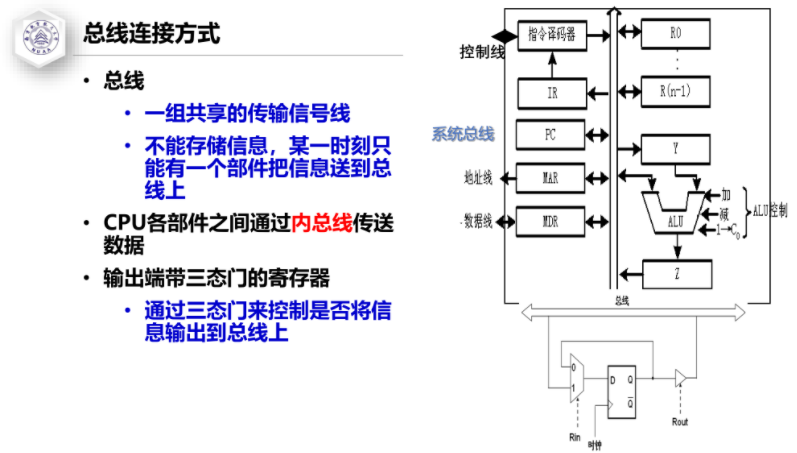
Write Enable：写使能



Decoder：译码器

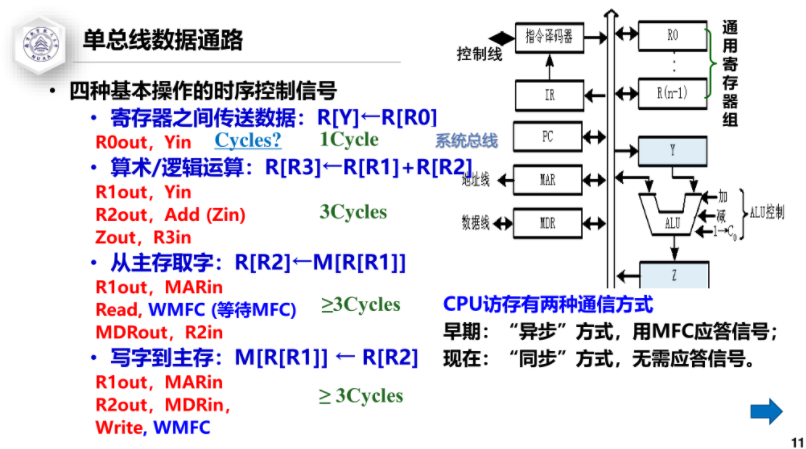
写口用的译码器 读口用的选择器





图中的总线称作CPU的内部总线

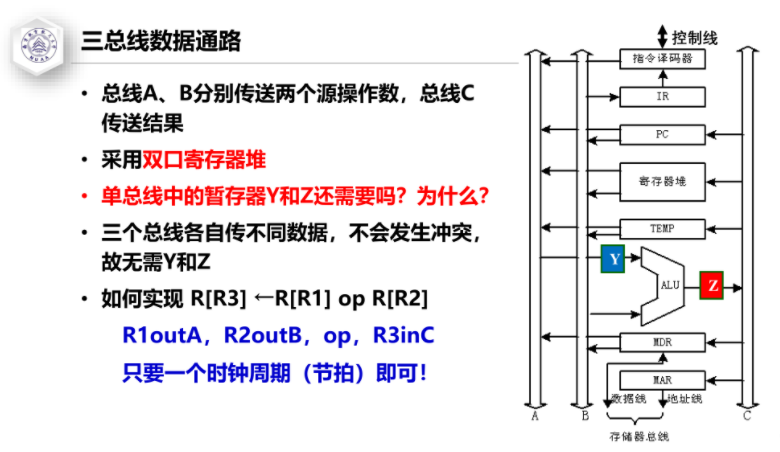
最底下的小部件是输出端带三态门的寄存器

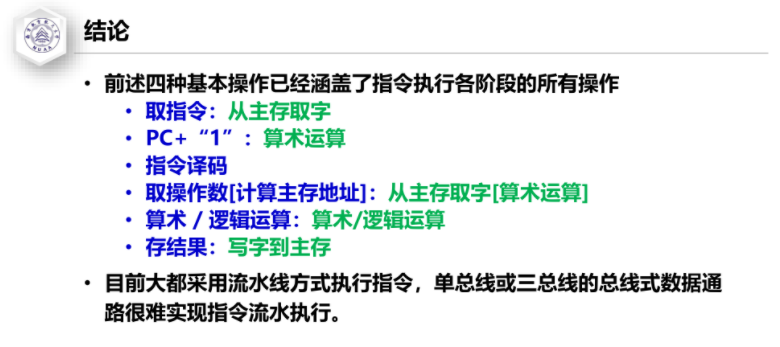


系统总线包括控制线、地址线、数据线

Zin挂在ALU上，可以省略

异步方式访存，效率很低

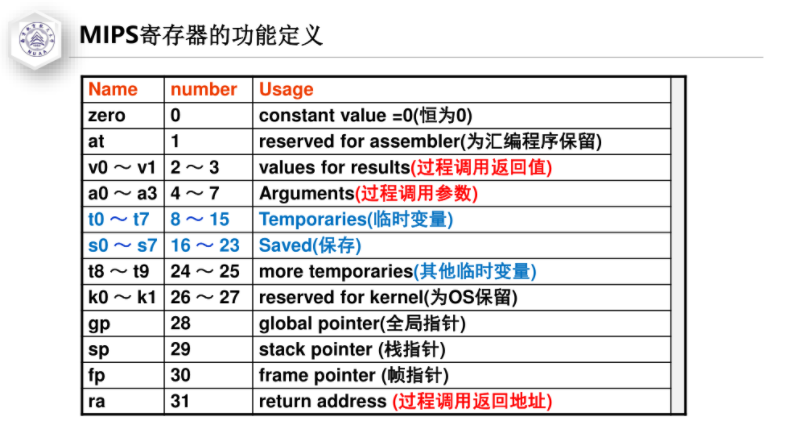




流水线可以同时间多条指令同时执行

单周期数据通路（MIPS为例）

**MIPS指令集概述**



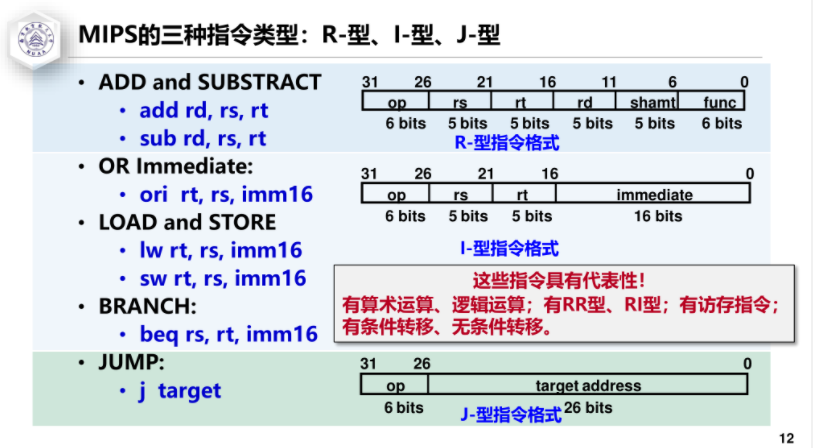


31个32位通用寄存器，1个全0的32位通用寄存器

32个32位浮点寄存器，成对做double

除了load/store可以访存，其它指令都只能是访问寄存器

地址加是无符号加



R型：op全0，func确定操作类型，rs、rt、rd确定三个寄存器，rd是目的寄存器，shamt用于移位。

I型：

ori：或逻辑运算，imm16 是无符号数，0扩展，rt是目的寄存器

load&store：imm16 是偏移量，作有符号数，符号扩展

rt <- rs + imm16

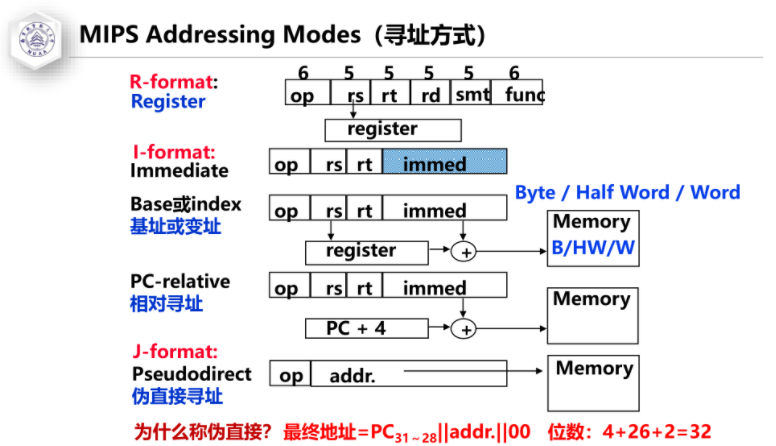
BRANCH：条件转移指令，imm16相对寻址

J型：

JUMP无条件跳转指令，target绝对寻址







R：寄存器寻址

I：立即数、寄存器寻址

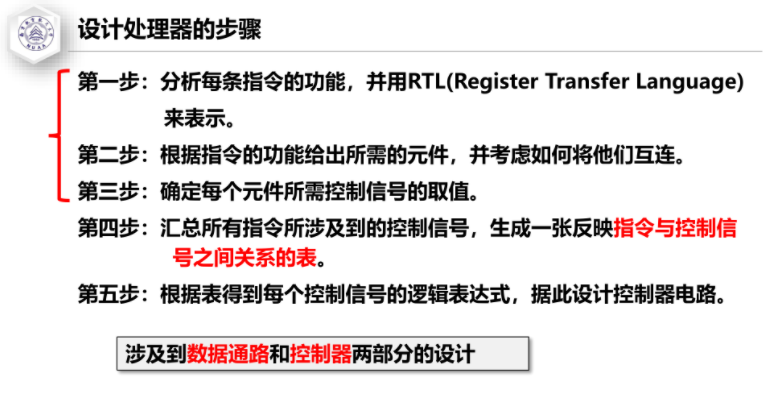
基址或变址

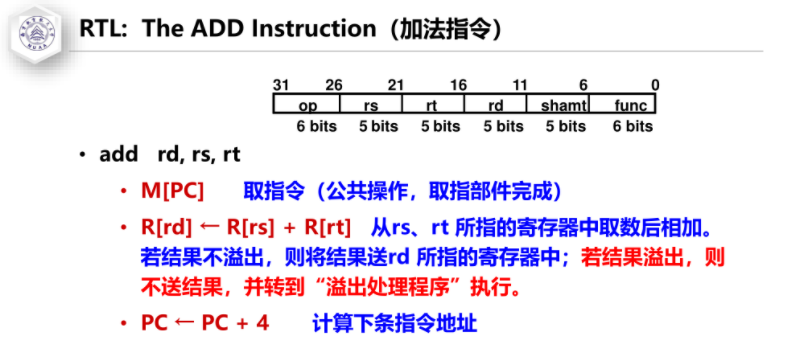
相对寻址（电路比较简单）

J：伪直接寻址

地址为4的倍数，所以后两位一定是0

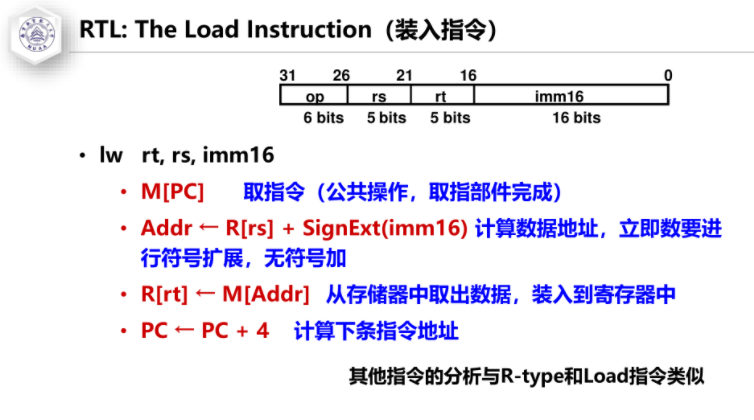
**单周期数据通路实现**

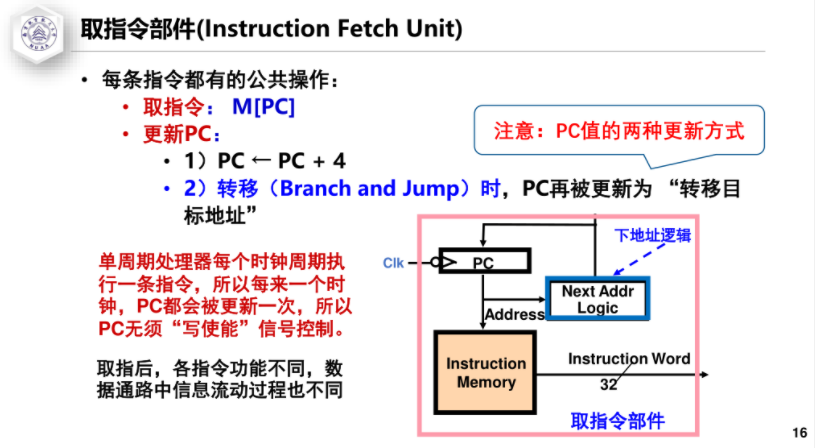
****

****

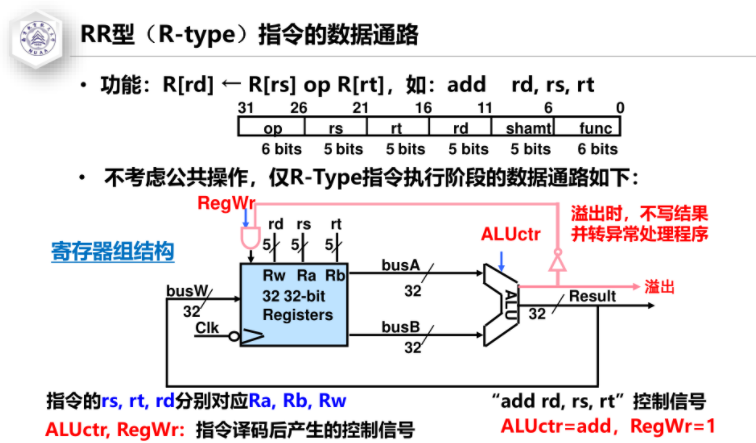
**MIPS没有状态寄存器**

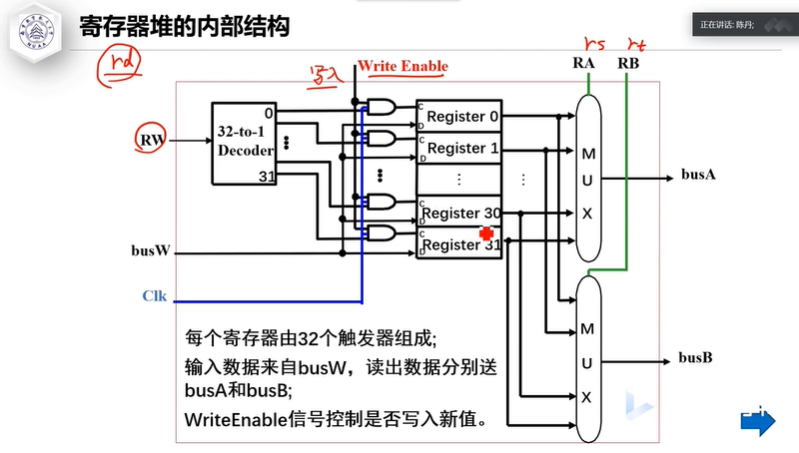
**addu、subu 无符号运算**

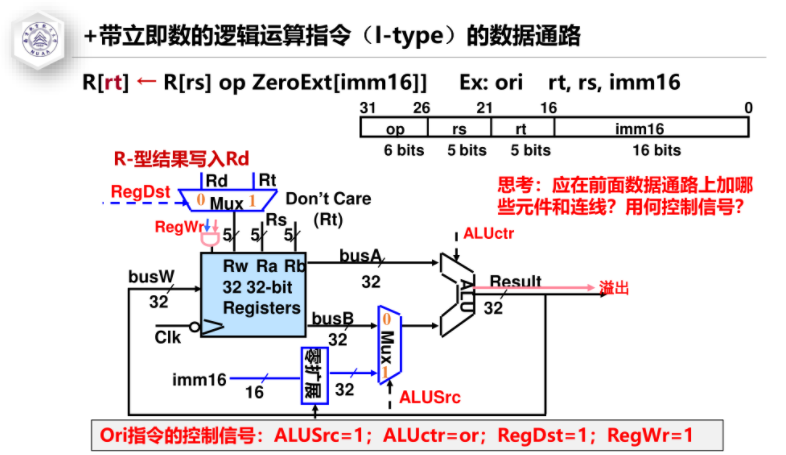
****

****

**下地址逻辑：有三种情况，所以用MUX实现**

****

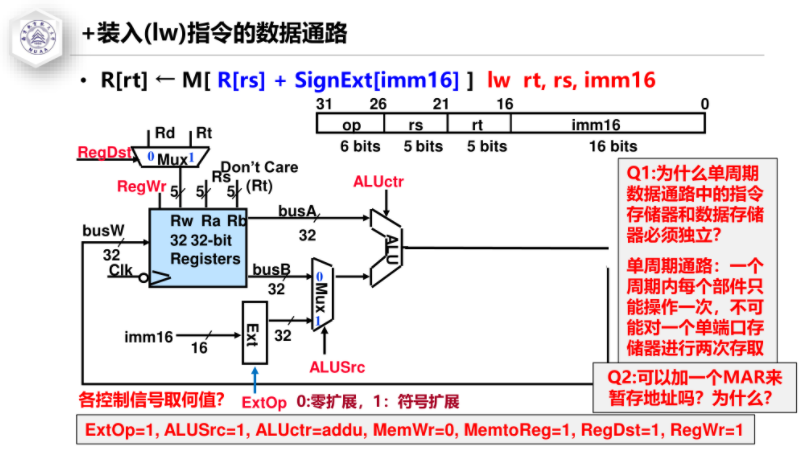
****

****

**ExtOp零扩展控制信号，零扩展/符号扩展二选一**

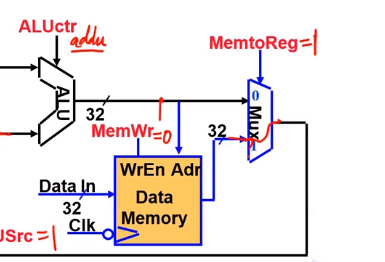
**RegWr写使能信号**

**虽然没有溢出标志寄存器，ALU依旧有四个溢出标志**

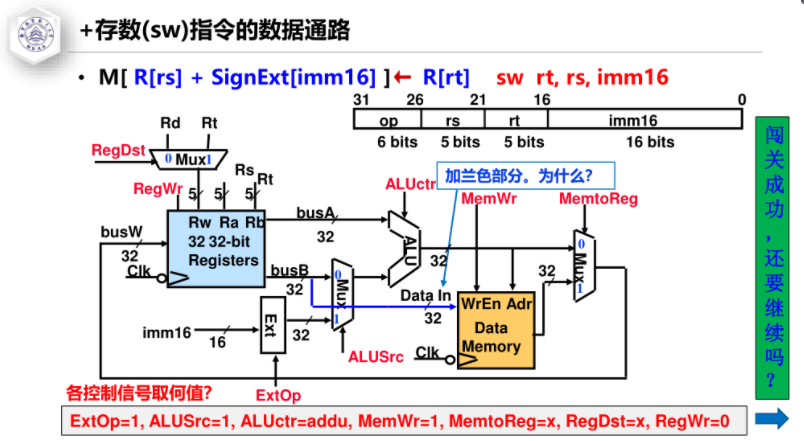
****

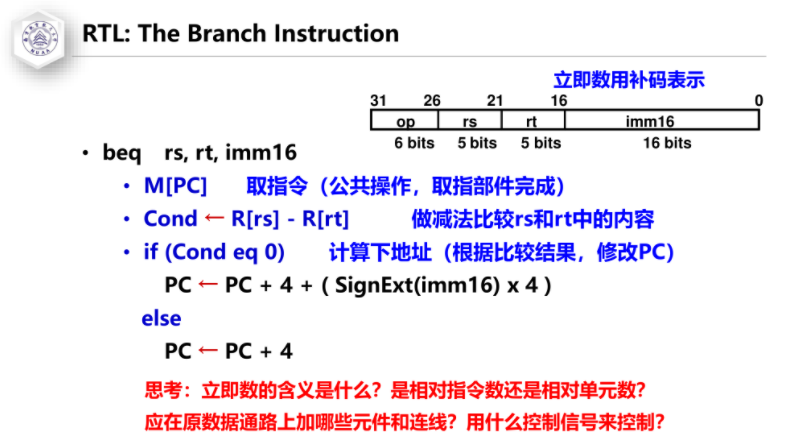
**w：一个字**

**取数据**

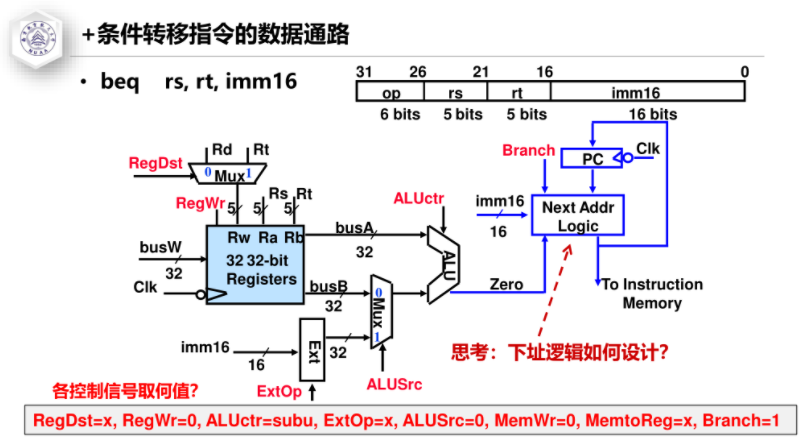
****

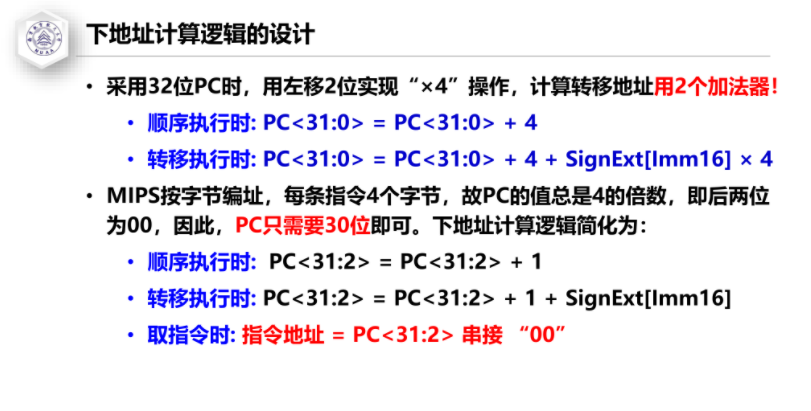
**Q2：不可以，加了之后时序控制更加复杂，需要在时间周期的中间某个位置再设置一个跳变，没有必要。**

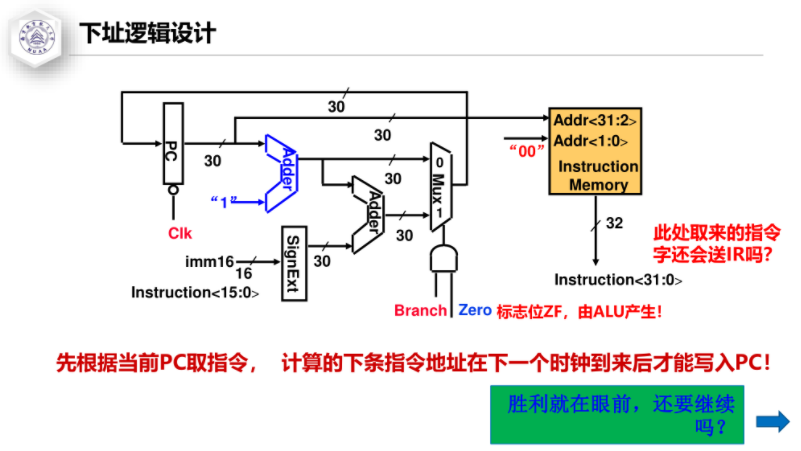
****

****

**含义：指令数**

****

****

****

**不会送IR，理由同上面的不用MAR、MDR**

**下一个时钟沿到达+Clk-to-Q时写入PC**