

计算机系统三

课程回顾

内容

- Ch1 概要
- Ch2 指令
- Ch3 运算
- Ch4 处理器
- Ch5 存储

Ch1

- 八个伟大思想

面向摩尔定律的设计，使用抽象简化设计

加速大概率事件，通过并行提高性能，通过流水线提高性能，通过预测提高性能

存储器层次，通过冗余提高可靠性

- 性能 \rightarrow 1/执行时间 指令平均时钟周期：Clock Cycle per Instruction, CPI

- $T_{cpu} = N_{cpu} \times 1/f_{clk}$

- $N_{cpu} = N_i \times CPI_i$

$$N_{cpu} = \sum_{i=1}^n (N_i \times CPI_i)$$

Ch2

• 寻址模式

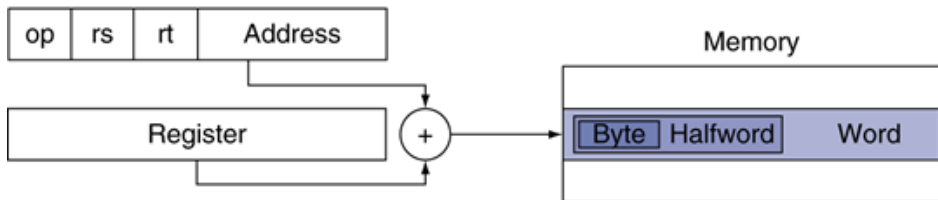
1. Immediate addressing



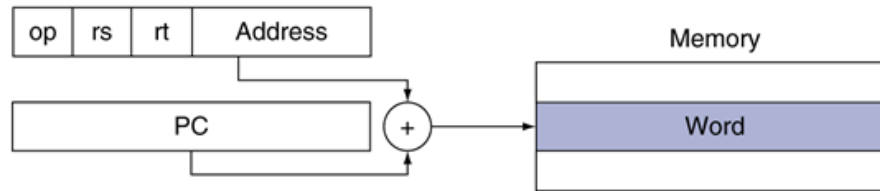
2. Register addressing



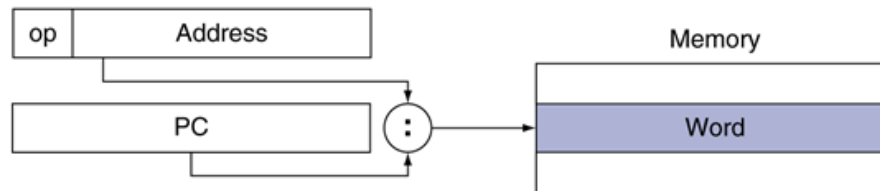
3. Base addressing



4. PC-relative addressing



5. Pseudodirect addressing



• 指令



6 bits 5 bits 5 bits 5 bits 5 bits 6 bits



6 bits 5 bits 5 bits 16 bits



6 bits 26 bits

简单 \leftrightarrow 规整 (Simplicity favors regularity)

寄存器比内存要更快地存取数据
算术运算指令 **只** 使用寄存器操作
指令/数据都被表示为二进制 (补码)

Ch2

- 过程→函数调用

```
jal ProcedureLabel  
jr $ra
```

- 程序执行

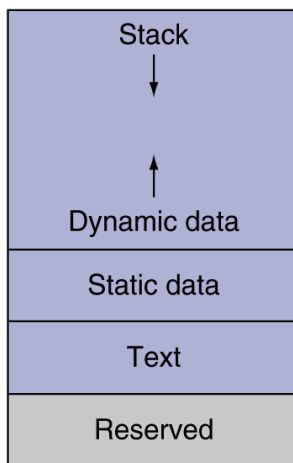
\$sp → 7fff fffc_{hex}

\$gp → 1000 8000_{hex}

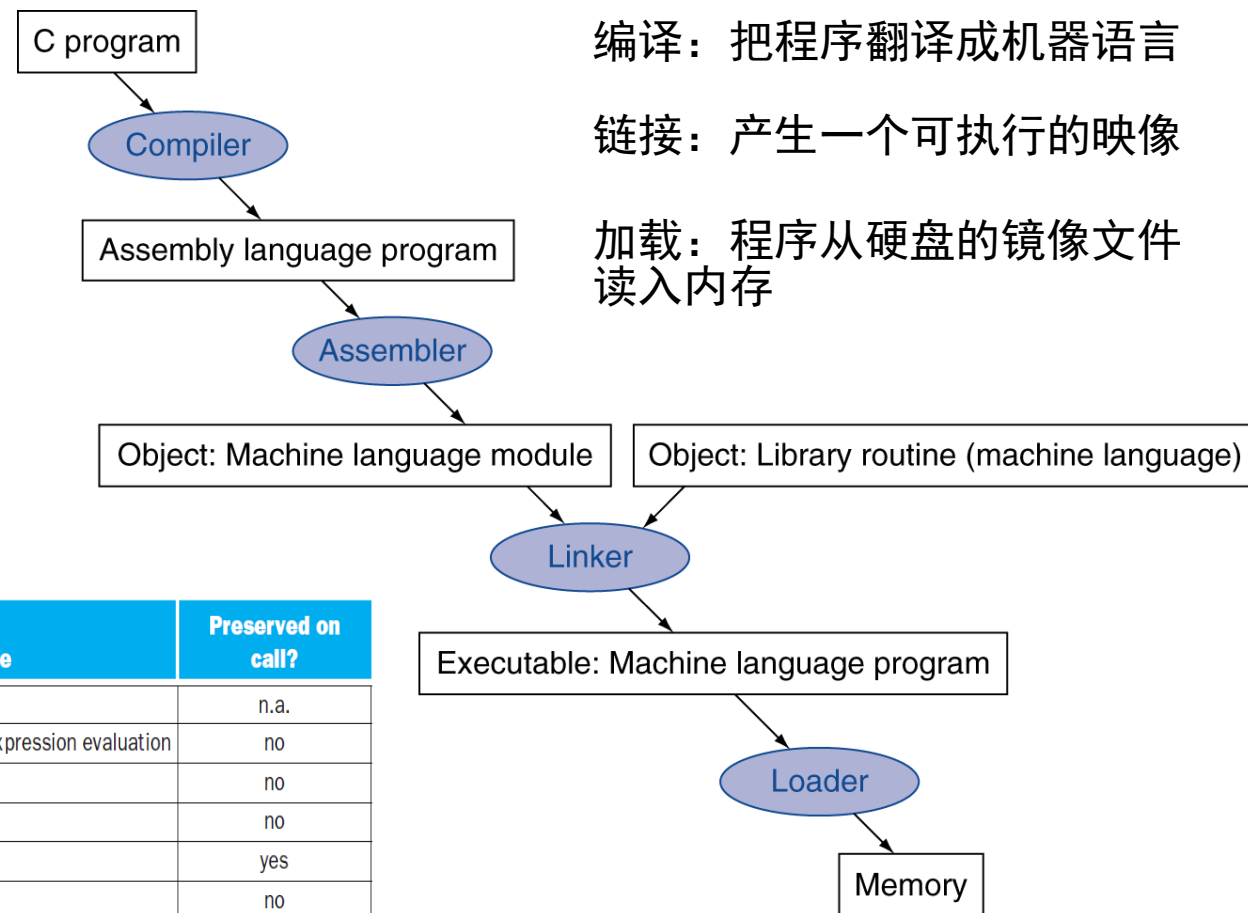
1000 0000_{hex}

pc → 0040 0000_{hex}

0

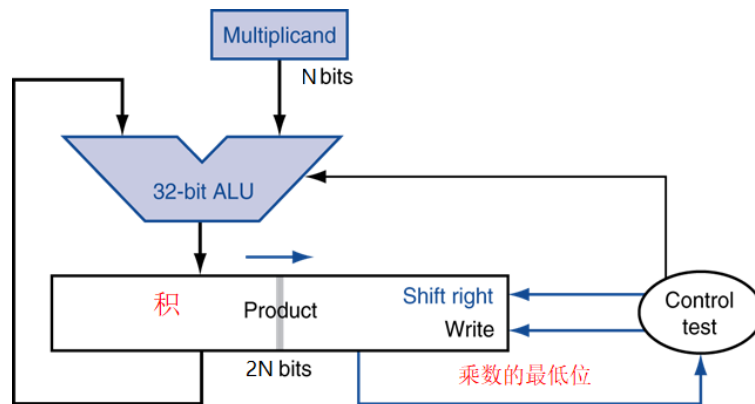
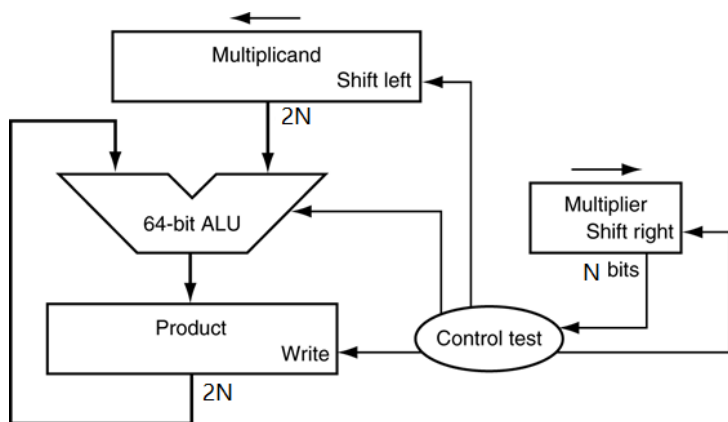


Name	Register number	Usage	Preserved on call?
\$zero	0	The constant value 0	n.a.
\$v0-\$v1	2-3	Values for results and expression evaluation	no
\$a0-\$a3	4-7	Arguments	no
\$t0-\$t7	8-15	Temporaries	no
\$s0-\$s7	16-23	Saved	yes
\$t8-\$t9	24-25	More temporaries	no
\$gp	28	Global pointer	yes
\$sp	29	Stack pointer	yes
\$fp	30	Frame pointer	yes
\$ra	31	Return address	yes

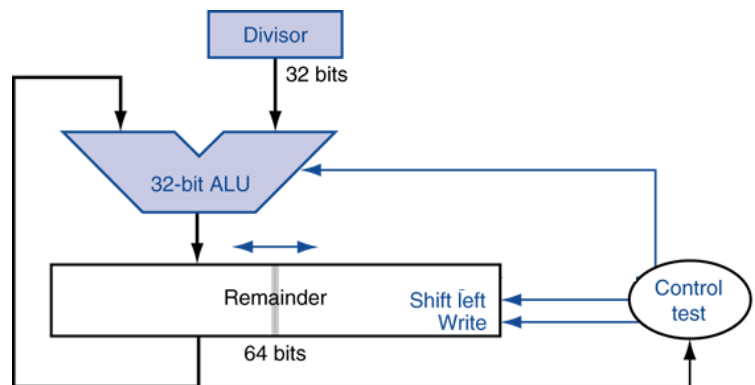
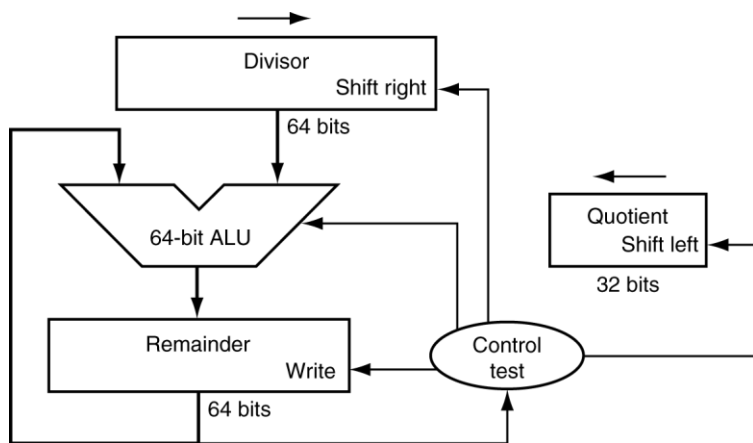


Ch3

• 乘法器



• 除法器



Ch3

• 浮点数

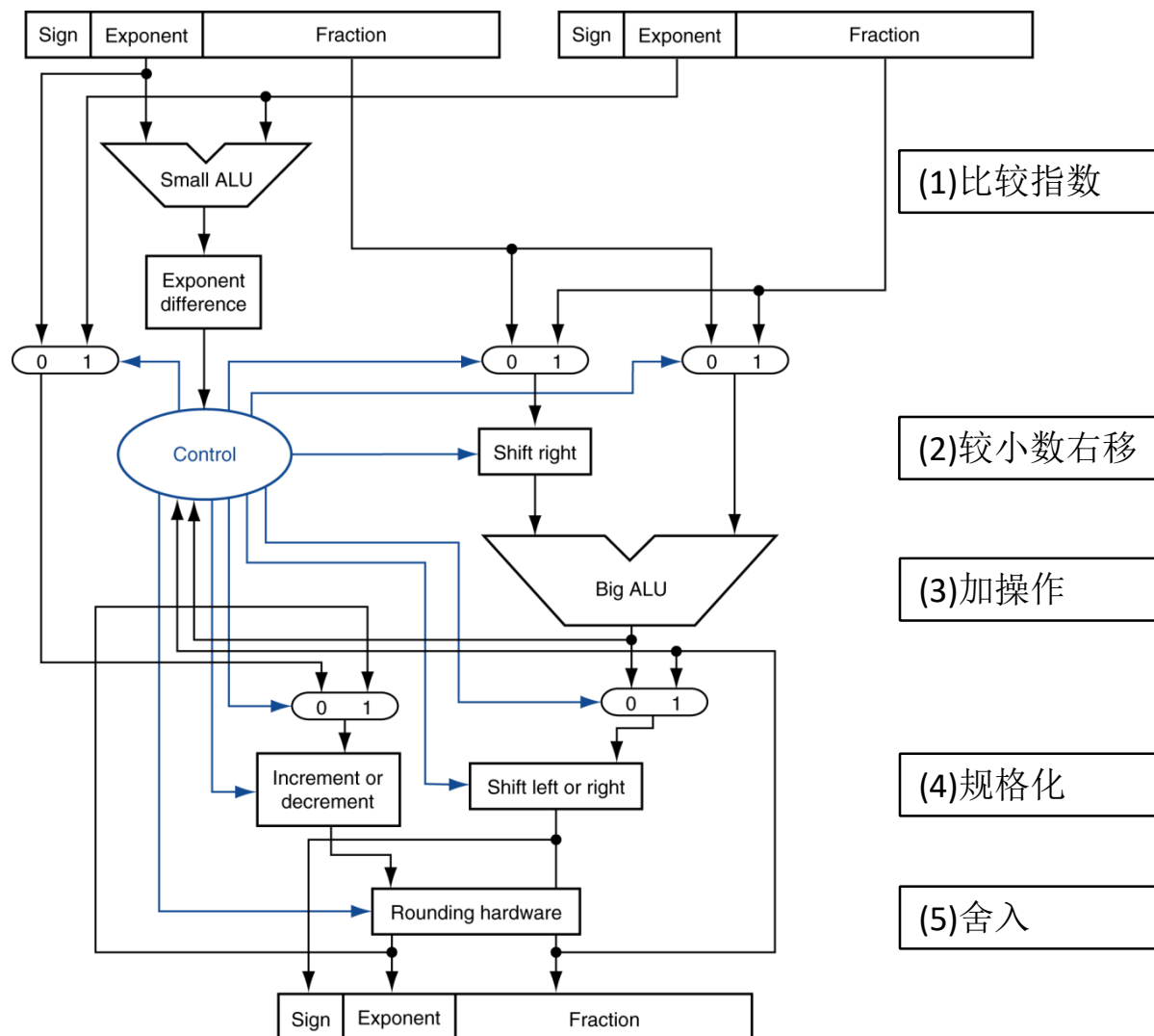
单精度single: 8 bits , 23 bits

双精度double: 11 bits, 52 bits

S	阶码 (指数+偏移)	尾数
---	------------	----

$$x = (-1)^S \times (1 + \text{尾数}) \times 2^{(\text{阶码} - \text{偏移})}$$

• 浮点加法器



• 单周期数据通路

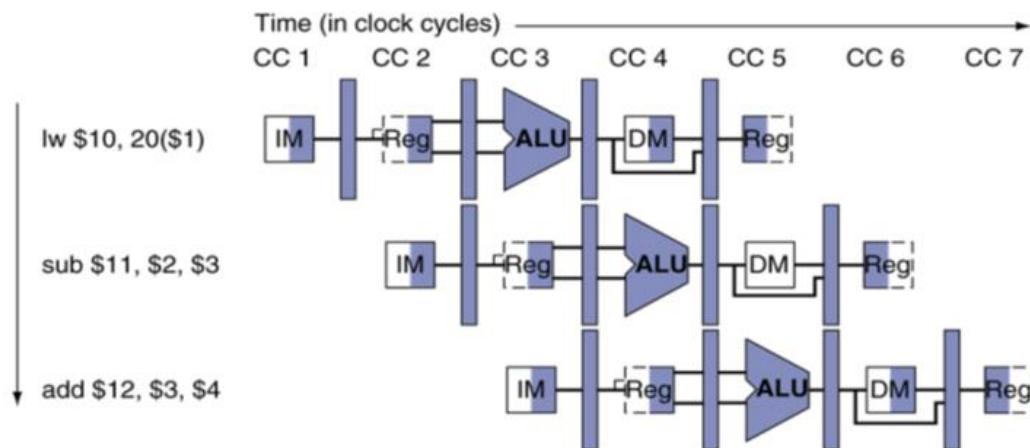


信号名	0的含义	1的含义
RegDst	写寄存器的目标号来源于指令rt字段(bits 20:16).	写寄存器的目标号来源于指令rd字段(bits 15:11).
RegWrite	无	寄存器堆写使能
ALUSrc	ALU第二个输入来源于寄存器堆的第二个输出	ALU第二个输入来源于指令的低16位(目标地址的偏移量)
Branch (PCSrc)	顺序执行, 取 $PC + 4$.	跳转, 使用目标地址替代PC+4
MemRead	无	数据存储器读使能
MemWrite	无	数据存储器写使能
MemtoReg	写入寄存器的值来源于ALU.	写入寄存器的值来源于数据存储器
ALUOP	ALU功能控制	

Ch4

• 流水线

1. IF: 从内存中取指令
2. ID: 指令解码& 读寄存器
3. EX: 执行运算或计算地址
4. MEM: 访问内存操作
5. WB: 将结果写回寄存器



• 冒险

结构冒险：部件忙

分离独立的指令和数据内存

数据冒险：需要前面指令的计算结果

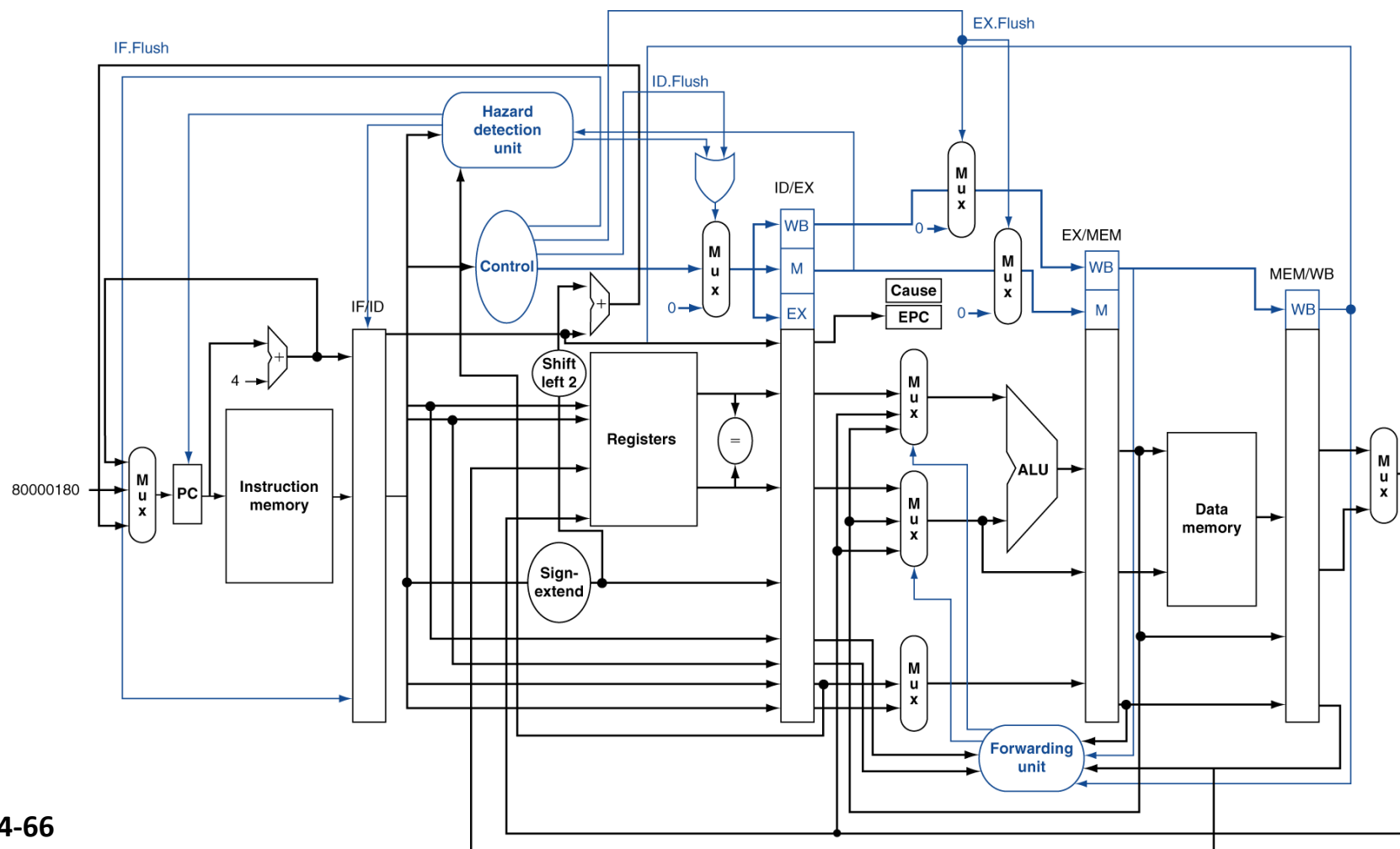
前推/阻塞/指令调整

控制冒险（控制相关）

类型和判断条件

Ch4

• 流水线数据通路

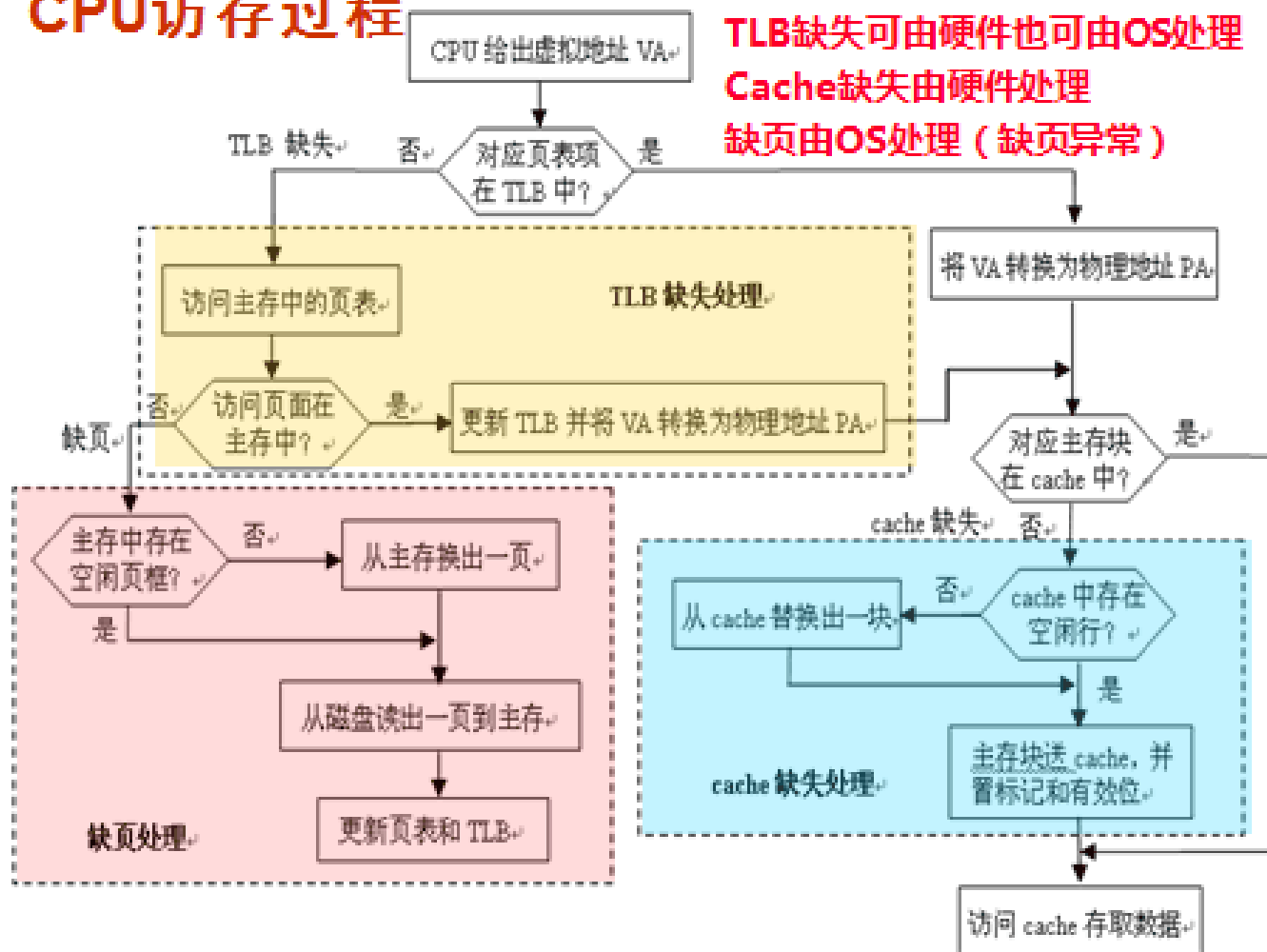


非流水基础之上增加:

- 1) 流水线寄存器
- 2) 旁路电路
- 3) 冒险电路
- 4) 分支预测及撤销
- 5) 异常处理

Ch5

CPU访存过程



Ch5

• Cache: 块

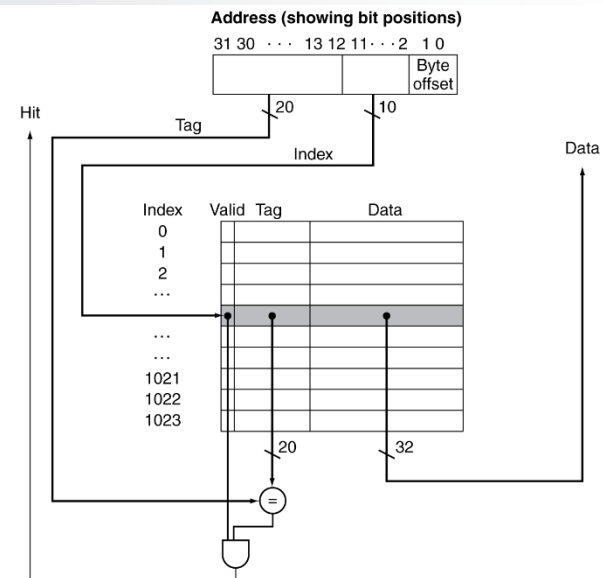
块地址=地址/块大小, 索引(块号)=块地址 mod 块数
相联:

N路组相联→组数=块数/N, 组号=块地址 mod 组数

1路组相联→组数=块数→直接映射

全相联→只有1组

Index 索引	V 有效位	Tag 标记	Data 数据
000	N		
001	N		
010	N		
011	N		
100	N		
101	N		
110	N		
111	N		



• 虚拟存储器: 页

页号= 虚拟地址/页大小

全相联

页表的缓存→TLB

命中/缺失
写直达
写回

