

注意：本文档仅用于说明本课程的期末考试题型，完整的知识点按复习要求进行。期末考试一般无选择题、填空题、判断题，以综合问答为主，可参考教材每章的课后作业题。

示例 1：考察第三章浮点表示。

假设有一个特殊的浮点表示结构，符号位 1 位，5 位阶码，8 位尾数；阶码全 0 表示指数-15，阶码全 1 表示指数 16；尾数部分隐含整数位为 1（整数位 1 不保存在尾数部分）；舍入采用直接截断。一个十进制数-26.62510，

1) 请说阶码是如何表示指数，说明映射规则？

2) 请说明表示这个十进制数据的 14 位浮点的二进制数填入下表（请在答题纸上画表填入）。

符号位	阶码					尾数							

3) 如果要提高舍入操作的精度，请说明舍入操作方法如何优化。

4) 如果采用 IEEE 754 标准的单精度浮点表示方法，如何计算两个 32 位单精度浮点的乘法，请画出流程图并简要说明。

参考答案：

1) 阶码的编码为移码。映射方式：移码=5 位阶码原值+15。

2)

$$26.625_{10} = 11010.101_2 = 1.1010101 \times 2^4$$

符号位	阶码					尾数							
1	1	0	0	1	1	1	0	1	0	1	0	1	0

3) 主要说明使用保护位，粘贴位，采用 0 舍 1 入也可以。

4) 主要说明阶码（指数）相加，尾数相乘，规格化、舍入、规格化迭代等关键步骤。

示例 2：考察第四章处理器设计。

在某冯诺依曼结构 RISC 处理器中，指令和数据宽度都是 16 位，流水线分为五级（取指、译码（同时取数）、执行、访存、写回）。如下表所示程序表达式和对应的指令序列，在一个标准 5 级流水处理器中运行。该处理器分支决断发生在执行级，不使用延时槽方法，预测分支永远不跳转；该处理器没有任何前递和旁路的结构；所有存储器访问 1 拍可以获得数据。下表左侧 c 程序所有变量均为 16 位整形，其中数组

$X[4]=\{4, 2, 0, -2\}$, $Y[4]=\{1, 2, -3, 0\}$ 。la、li 为伪指令, 其中 la s1, X 将数组 X 的指针加载给寄存器 s1, li s3 4 将立即数 4 加载给寄存器 s3。

c = 0;	1: la s1, X
for (i = 0; i < 4; i++) { // B3	2: la s2, Y
x = X[i];	3: li s3, 4
y = Y[i];	4: li s4, 0 # c
if (x == 0) // B1	5: loop: lw s5, 0(s1) # x
c++;	6: lw s6, 0(s2) # y
if (y == 0) // B2	7: bnez s5, skip1 # B1
c--;	8: addi s4, s4, 1
	9: skip1: bnez s6, skip2 # B2
	10: addi s4, s4, -1
	11: skip2: addi s1, s1, 2
	12: addi s2, s2, 2
	13: addi s3, s3, -1
	14: bnez s3, loop # B3

1) 如果没有 Cache, 请说明该指令序列中存在的冒险情况以及对应的冒险类型, 如果是数据冒险请给出产生相关的指令和寄存器。如果发生结构冒险, 如何避免结构冒险?

2) 如果有 I-cache 和 D-cache, 请说明右侧汇编代码加粗部分 loop 至 skip2 之间第 5 行至第 11 行 7 行代码在循环第一次迭代时实际执行指令的数量, 并画出对应的时空图?

3) 该处理器执行这段程序需要多少个周期以及对应的 CPI 是多少, 如果处理工作频率 200MHz, 这段程序执行时间是多少?

4) 如果该处理器使用完全前递和仅采用 ALU-ALU 的前递两种方法优化数据冒险, 三种设计的电路工作频率如下表, 三种电路执行上述程序, 比较执行时间, 哪种会更快?

原始设计	完全前递	仅 ALU-ALU 前递
200MHz	150MHz	170MHz

5) 如果能够采用循环展开并进行指令级并行设计, 可以实现最多 4 条指令级并行执行, 但是其中有 30% 的指令因无法并行需串行执行, 执行这段程序的加速比是多少?

参考答案:

1)

存在数据冒险、控制冒险、结构冒险。

数据冒险: 5、7 行, 寄存器 s5 冒险。

避免结构冒险措施: 使用 cache 或者增加预取缓冲等。

2)

根据跳转条件判断，实际执行 5 条指令，时空图如下（黄色部分为气泡）：

lw s5, 0(\$t1)	IF	ID	EX	MEM	WB										
lw s6, 0(\$t2)		IF	ID	EX	MEM	WB									
bnez s5, skip1			IF				ID	EX	MEM	WB					
bnez s6, skip2									IF	ID	EX	MEM	WB		
addi \$1, \$1, 2												IF	ID	EX	MEM

3)

Loop 前面执行 4 条指令

Loop 第一次迭代 8 条指令

第二次迭代 8 条指令

第三次迭代 9 条指令

第四次迭代 9 条指令

指令流水节拍 $4+8+8+9+9=38$ 拍

发生跳转冒险与无跳转冒险相同，寄存器 S5 数据冒险每次迭代额外增加 2 拍， $2 \times 4=8$ 拍

所有分支指令发生四次，额外增加 $(2+2+2) \times 4=24$ 拍

流水线建立/退出需要额外 4 拍。

总计 $38+8+24+4=74$ 拍，执行时间为：

$74/200\text{MHz}=0.37\text{us}$ 或 $70/200\text{MHz}=0.35\text{us}$ (不计算流水线拆除时间)

4) 没有前递的电路会更快。

完全前递可以消除存储器访问对应的数据冒险，ALU-ALU 前递无法消除存储器访问对应冒险。

因此，完全前递 $38+24+4=66$ 拍， $66/150\text{MHz}=0.44\text{us}$ 或 $62/150\text{MHz}=0.41\text{us}$ (不计算流水线拆除时间)

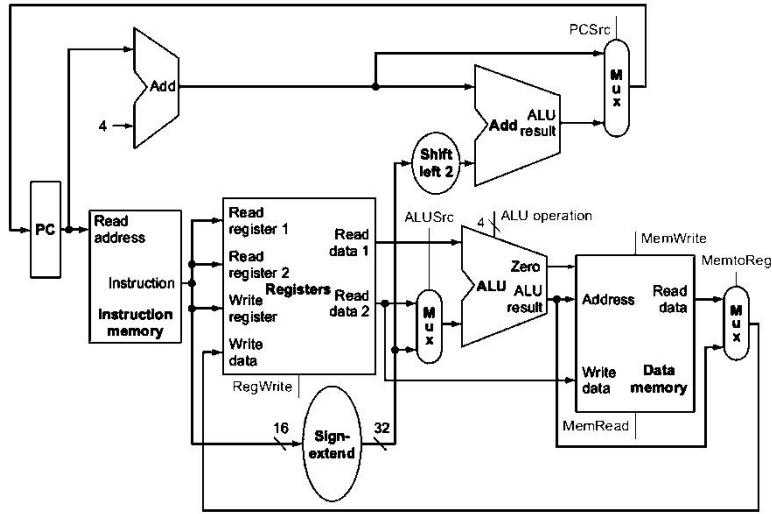
ALU-ALU 前递 $74/170\text{MHz}=0.435\text{us}$ $70/170\text{MHz}=0.47\text{us}$

结论：没有前递的电路会更快。

5) 根据 Amdahl 定律，加速比= $1/((1-0.3)/4+0.3)=40/19 \approx 2.105$

示例 3：考察第四章处理器设计。

如下图支持 MIPS 简化指令集（仅包括 add、sub、ori、lw、sw 以及 beq 六条指令）的单周期处理器数据通路，其各单元的延时关系如下表所示。如果忽略所有的线延时，电路中所有延时仅为下表中内容。



I-Mem/ D-Mem	Registers File Access	Mux	ALU	Adder	单一逻辑门	Register Access	Register Setup (RF 同)	Sign Extend	Decode
200ps	200ps	25ps	150ps	100ps	10ps	30ps	50ps	45ps	40ps

- 1) 请说明 R-Type 指令的延时由哪些部分的延时组成，总延时是多少？
- 2) 通过计算，说明关键路径是由哪条指令造成的？该处理器的最大工作频率是多少？
- 3) 如果使用上表中的寄存器延时参数对应的寄存器构建 5 级流水的处理器，分别是取指、译码（同时取数）、执行、存储器访问以及寄存器文件写回 5 级，请画出数据通路结构图，并计算工作频率。

参考答案：

1) 680 ps。R-type 指令会经历的延时=

$$\begin{aligned}
 & \text{Treg-acc+TI-mem+Tdecode+TRF-acc+Tmux+Talu+Tmux+Treg-stp} \\
 & = 30 + 200 + 40 + 200 + 25 + 150 + 25 + 50 = 720 \text{ ps}
 \end{aligned}$$

2) 1.12 GHz。lw 指令是关键路径。

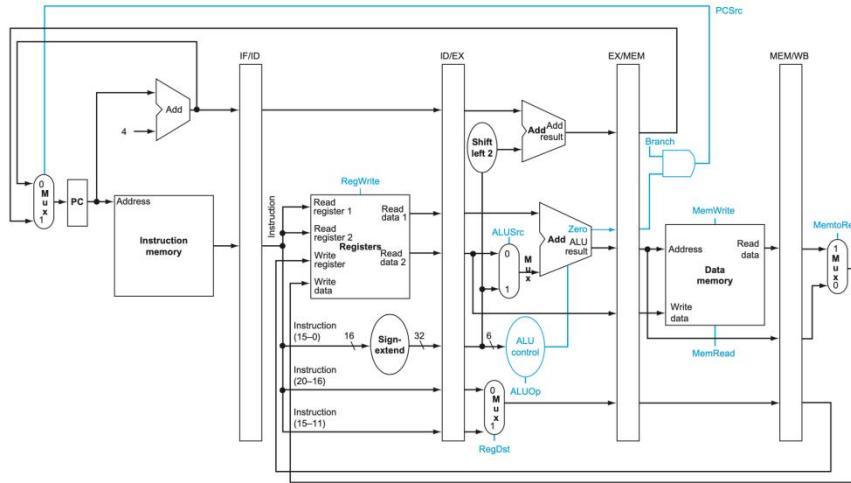
Lw 指令延迟=

$$\begin{aligned}
 & \text{Treg-acc+TI-mem+Tdecode+TRF-acc+Talu+Td-mem+Tmux+Treg-stp} \\
 & = 30 + 200 + 40 + 200 + 150 + 200 + 25 + 50 = 895 \text{ ps}
 \end{aligned}$$

$$F_{\max} = 1/895 \text{ ps} = 1.12 \text{ GHz}$$

3) 3.57 GHz

流水线数据通路参考下图，重点画出流水线寄存器：



最慢的一级是译码级，需经历 $T_{reg_acc} + T_{decode} + T_{RF_acc} + T_{reg_stp} = 30 + 40 + 200 + 50 = 320\text{ps}$ ，CPU最高频率为 $1/320\text{ps} = 3.125\text{GHz}$ 。

示例 4：考察第五章层次化存储。

1) 假设某处理器 $CPI=1.0$ ，整个系统工作频率为 1.0GHz ，该处理器使用了一级 Cache，Cache 数据读写延时均为 2 个时钟周期，该 Cache 的平均命中率为 80%；片外存储器访问延时为 100ns ；如果通过增加二级 Cache 进行优化，该二级 Cache 的读写需要 5ns ，二级 Cache 命中率 95%，片内系统仍然采用上述电路工作频率，优化后，处理器 CPI 为多少？

2) TLB 的设计目的是什么？说明一次存储访问，在 TLB、Cache 都命中的情况，请说明存储访问执行过程？如果 TLB 发生缺失，仅使用一级页表模式，页面访问可能出现哪些情况？

3) 如果一个处理器采用 16 位地址，其中地址高 4 位表示 TAG，中间 8 位用于作为 Index，后 4 位表示 Cache 块内偏移，请画出该地址分配关系对应的直接映射 Cache 的结构关系，画图中必须能够表示 Cache 的 Tag、Data 域的行列关系，命中判断和 Data 行内字选择。

4) 如果将第 3) 问中直接映射结构的 Cache 作为 I-cache 来执行第一题中的指令序列，则第二次迭代会出现多少次不命中（缺失）？

5) 如果不改变地址位段的划分映射，希望 Cache 容量增大为原来的 2 倍，Cache 结构应如何变化？，重新画出 Cache 结构图，仍需在图中标识清楚 Cache 的 Tag、Data 域的行列关系，命中判断和 Data 行内字选择关系。

6) 如果使用优化后的 Cache 作为 I-cache 解决本题第 4) 问的问题，即重新执行

第一题指令序列，第二次迭代命中情况会如何变化？写明详细命中次数。

参考答案：

1) $4+3x$ (x 为访存指令比例)。

包含两级 cache 的存储系统 $AMAT=2+0.2*(5+5\%*100)=4$ ns。

流水线本身为访存提供了 1 个周期，访存还需要阻塞 $4-1=3$ 拍。

假设指令中访存类指令 (lw 或 sw) 占比为 x ，那么 $CPI=1+(1+x)*3=4+3x$ 拍。

备注：若不考虑访存指令比例，即认为 $x=0$ ，直接回答 $CPI=4$ 也可以。

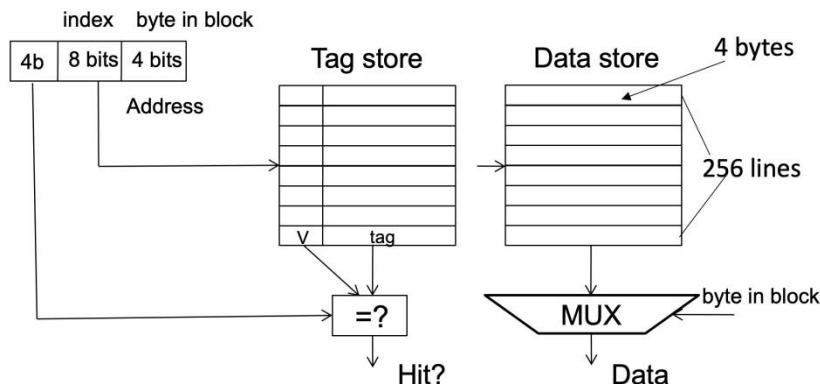
2)

TLB 的目的是为了提高页表访问的片上局部性的，提高页表访问的效率，提高虚拟地址转换计算的效率。

若一次存储访问，在 TLB、Cache 都命中的情况，则直接从 cache 获得数据。

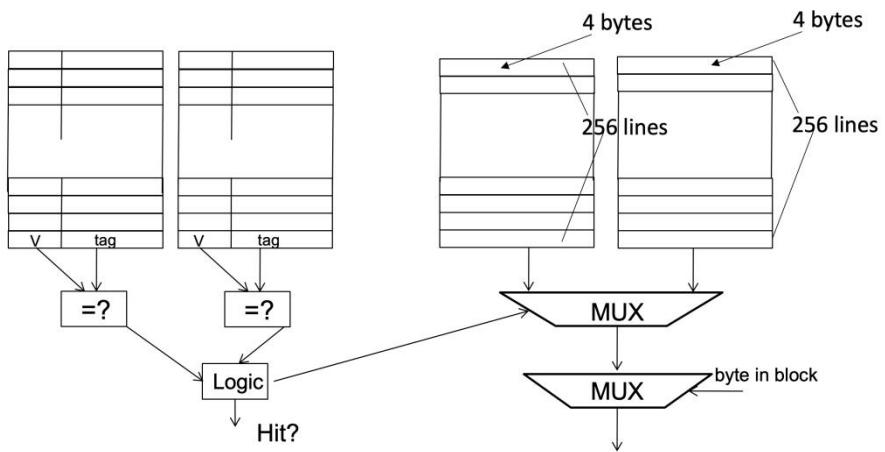
如果 TLB 发生缺失，仅使用一级页表模式，则情况有：第一，PTE 有效，使用实地址访问 cache，cache 命中直接返回数据，cache miss 需要访问内存；第二，PTE 无效，发生 Page Fault，需要操作系统从硬盘获得数据。

3) 参考下图，cache 应当为 256 行，每行 4 个字节



4) 0 次不命中。第一题中循环内地址连续，指令序列不超过 256 条，不存在 conflict miss，第二次迭代应该不会出现指令 miss，第二次迭代执行的 8 条指令全部命中。

5) 变为 2 路组相联即可，结构如下图：



6) 不会出现缺失，因为没有缩小 index 结构，仍然命中 8 次。