

2022春系统结构期末

判断

1. 程序的顺序执行,是造成时间局部性原因之一。✗ 空间
2. 组相联映射是对全相联和直接映射的一种折中,组间采用全相联映射,组内采用直接映像。✗ 反了
3. 采用多种寻址方式可以减少程序的指令条数,但可能增加处理器实现的复杂度以及指令的CPI。✓
4. 流水线属于时间重叠/并行技术。✓
5. 数据总线宽度对于高级程序员来说是透明的。✓
6. 某型号处理器增加几条媒体处理指令后,不会影响其软件的向后兼容性。✓
7. 在顺序发射、顺序完成的指令单流水线中,不存在WAW和WAR冲突。✓
8. 在指令字长确定的情况下,指令系统设计需综合考虑寄存器个数、寻址方式、立即数长度等需求。✓
9. 动态多功能流水线通过不同连接方式允许在同一时刻执行不同功能。✓
10. 减小CPI是RISC架构的精华,而只有Load和Store指令可以访存则是RISC架构的根本。✓
11. 从系统结构角度看,增加流水线级数可以提高处理器频率。✓
12. 增加Cache组相联度将会减少Cache的冲突缺失次数,但会增加命中时间。✓
13. 多级层次存储系统一定要满足包含关系,即上层缓存中的数据一定可以在其下级存储器中找到。✓
14. 采用写回(write-back)的Cache,当发生读缺失时,有时会引起写主存操作。✓
15. DRAM的密度比SRAM的密度大。✓
16. 在Cache-主存存储层次中,Cache块越大,则强制缺失次数 (compulsory miss)越少。✓
17. 为了支持精确中断,Tomasulo算法需要ROB(Reorder Buffer)支持。✓
18. 平均每条指令的执行周期数与程序无关。✗
19. Victim Cache是位于CPU和Cache间的又一级Cache。✗ 位于Cache和下级Cache
20. 通过寄存器重命名,可以消除WAW和WAR相关。✓
21. 关键字优先或早启动(early restart)技术可减少Cache缺失率。✗ 缺失开销
22. 较小而简单的Cache有利于减少命中时间,但会提高缺失率。✓
23. INTEL安腾处理器采用的是VLIW架构。✓
24. 采用写通过(write-through)策略,需要使用写缓冲器(write buffer)来提高性能,而采用写回策略的Cache则没有必要使用写缓冲器。✗ 写回法也可使用写缓冲器
25. 乱序执行的处理器采用多路组相联Cache时,为了减少功耗,一般采用Tag比较和数据访问串行的方式,并通过指令乱序执行来弥补由于存储访问延迟造成的性能下降问题。✗ 并行
26. 伪相联(Pseudo-Associativity)Cache由于变化的命中时间对CPU流水线影响较大,一般适合离CPU远的Cache。✓
27. 向量指令间存在针对同一向量寄存器的RAR(read after read)冲突。✓
28. 半性能向量长度 $n_{1/2}$ 等于向量寄存器长度的1/2。✗
29. 通过编译将循环展开成4个循环体时,一般可减少6条指令(测试和分支)。✓
30. 循环展开通过寄存器重命名和指令调度可有效开发指令级并行。✓

单选

31. 假定load和store指令占有所有指令的比例为30%,指令Cache的缺失率是2%,数据Cache的缺失率是5%,Cache缺失代价都为100个周期。如果没有任何访存缺失,处理器的CPI为2。如果考虑访存缺失,那么该处理器的CPI是:

- A. 5.5
B. 3.5

A.

$$2 + 30\% \times 5\% \times 100 + 2\% \times 100 \\ = 2 + 1.5 + 2$$

C. 6

D. 3

32. 在执行一段对数组元素求和的循环代码时, 以下说法最合适的是:

B/D.

A. 会体现出数据访问的~~时间~~局部性和空间局部性

B. 会体现出代码访问的时间局部性和空间局部性

C. 既会体现出数据访问的~~时间~~局部性和空间局部性, 也会体现出代码访问的时间局部性和空间局部性

D. 以上说法皆不对

33. 在发展高性能单处理机过程中, 起主导作用的是:

B.

A. 资源重复

B. 时间重叠/并行

C. 资源共享

D. 局部性原理

34. 关于计算机系统的性能指标: 执行时间和吞吐率, 下列说法错误的是:

C

A. 管理员更关注吞吐率

B. 用户更关注执行时间

~~C. 执行时间短意味着吞吐率高~~

D. 用户看到的程序执行时间不仅仅是CPU执行该程序的时间

35. 指令系统采用不同寻址方式的目的主要是:

B

A. 实现程序存储与程序控制

☒ B. 缩短指令长度, 扩大寻址空间, 提高编程灵活性

C. 可直接访问外存

D. 提供扩展操作码的可能并降低指令译码难度

36. 下列关于指令系统说法错误的是:

C.

A. 指令系统不仅仅是定义了各种指令

B. 具有相同指令系统的计算机其程序可相互兼容

☒ C. 具有相同指令系统的计算机其微架构必然相同

D. 指令系统定义了计算机软硬件接口

37. 关于流水线分类, 以下说法错误的是:

A

A. 单功能流水线可分为静态与动态流水线

☒ B. 流水线可分为单功能与多功能流水线

☒ C. 流水线可分为线性与非线性流水线

☒ D. 流水线可分为顺序与乱序流水线

→ 多功能分静(动态)

38. 以下哪个指标 不是 评价流水线性能的常用指标?

C.

A. 吞吐率

B. 加速比

C. 功耗

D. 效率

39. 各流水段的执行时间并不完全相等的流水线中, 最大吞吐率受限于

B

A. 排空时间

B. 瓶颈段的执行时长

C. 建立时间

D. 段数

40. 以下关于数据相关的说法错误的是:

A.

- A. 相关一定会转化为流水线冲突
- B. 相关是程序固有的属性
- C. 相关分为数据相关、控制相关与名相关
- D. 名相关可分为输出相关与反相关

41. 理论上, 以下哪个选项不会影响分支指令性能?

D.

- ☒ A. 何时算出分支是否成功
- ☒ B. 分支成功时, 何时算出分支目标地址
- ☒ C. 分支成功时, 目标指令是否在指令Cache中
- ☒ D. 分支成功时, 分支指令后续指令的处理

42. 层次存储系统中寄存器容量 << Cache容量 << 主存容量, 其主要因素是:

B.

- A. 速度
- ☒ B. 成本
- C. 密度
- D. 复杂性

43. 以下哪种结构能够避免取指与访问操作数之间的访存冲突?

B.

- A. 冯诺依曼结构
- ☒ B. 哈佛结构
- C. 层次存储
- D. 并行存储器

44. 在一个多级存储系统M1、M2、...Mn 中。下面关于存储系统期望达到的目标说法错误的是:

B.

- ☒ A. 整个存储系统的容量应该接近于Mn的容量
- ☒ B. 整个存储系统的访问时间应该接近于Mn的访问时间
- ☒ C. 整个存储系统的平均每位价格应该接近于Mn的平均每位价格
- ☒ D. 整个存储系统的访问时间应该接近于M1的访问时间

45. 采用指令Cache与数据Cache分离的主要目的是:

D.

- A. 降低Cache的缺失代价
- B. 提高Cache的命中率
- C. 降低CPU平均访存时间
- ☒ D. 减少指令流水线中Cache的访问冲突

46. 下列关于Cache的映像规则中, 哪种映像规则的Cache利用率最高?

A.

- ☒ A. 全相联
- B. 组相联
- C. 直接映射
- D. 段相联

47. 一般情况下, 下面哪种并行主存系统的访存效率最高?

D.

- A. 单体单字存储器
- B. 多体高位交叉存储器
- C. 单体多字存储器
- ☒ D. 多体低位交叉存储器

48. 以下说法错误的是:

D.

- A. 用间址寻址方式可以缩短地址码长度
- B. 用变址寻址方式可以缩短地址码长度
- C. 用寄存器间接寻址方式可以缩短地址码长度
- ☒ D. 一般说来, 用RISC计算机编程比用CISC计算机编程所占程序存储空间更小

49. RISC计算机LOAD和STORE指令平均占总指令数的比例分别是18%和6%，则每执行一条指令，平均访问存储器次数是：

- A. 1.24
- B. 0.18
- C. 0.06
- D. 0.24

A.

$$1 + (18\% + 6\%)$$

50. 下面哪种说法具有不确定性？

E

- A. ☒ 增加 Cache 容量, 会减小容量缺失, 增加命中时间
- B. ☒ 增加Cache相联度, 会减少冲突缺失, 增加命中时间 (假设 Cache 容量固定)
- C. ☒ 增加 Cache 块大小, 会减少强制缺失, 增加命中时间 (假设 Cache 容量固定)
- D. ☒ 增加 Cache 块大小, 会增加缺失代价
- E. ☒ 增加L1(第一级) Cache 相联度, 对增加其容量有利

51. 以下哪种技术不能减少Cache缺失代价(miss penalties)?

D.

- A. ☒ 多级 Cache
- B. ☒ 关键字优先
- C. ☒ 写缓冲合并(Merging Write Buffer)
- D. ☒ 软件/硬件预取
- E. ☒ 让读不命中优先于写

降低缺失率

52. 在一个5段的流水线处理机上需经9拍才能完成一个任务, 其预约表如下表所示, 该预约表对应的原始冲突向量是：

功能段	1	2	3	4	5	6	7	8	9
S1	×								×
S2		×	×					×	
S3				×					
S4					×	×			
S5							×		

8
5 6
1

- A. 10110011
- B. ☒ 10110001
- C. 10001101
- D. 11001101

B.

53. 如上题(单选22题)所示预约表, 通过插入非计算延迟可以实现流水线的无冲突最优调度, 此时可以达到的流水线的最大吞吐率是(假设流水线时钟周期为 Δt):

- A. $(1/2) \Delta t$
- B. ☒ $(1/3) \Delta t$
- C. $(1/5) \Delta t$
- D. $(2/7) \Delta t$

B

54. 一个单处理器采用Cache-主存两层存储结构, 指令Cache和数据Cache分离, 它们的命中率分别是 H_i 与 H_d 。处理机访问两个Cache的时间都是 c 个时钟周期, 一次访问主存储器的延迟是 b 个时钟周期。在CPU进行的所有访存操作中, 访问指令Cache的百分比是 F_i 。数据Cache采用写回和写分配(write-back&write-allocate)策略, 当数据Cache访问缺失时, 被替换块(dirty位为1)需要写回主存的情况占比是 F_d 。则该处理器的平均存储器访问时间为:

H_i H_d

- A. $F_i (H_i c + (1-H_i)(b+c)) + (1-F_i) (H_d c + (1-H_d) (F_d(b+c) + (1-F_d)(b+c)))$
- B. $F_i (H_i c + (1-H_i)(2b+c)) + (1-F_i) (H_d c + (1-H_d) (F_d(2b+c) + (1-F_d)(2b+c)))$
- C. ☒ $F_i (H_i c + (1-H_i)(b+c)) + (1-F_i) (H_d c + (1-H_d) (F_d(2b+c) + (1-F_d)(b+c)))$
- D. $F_i (H_i c + (1-H_i)(b+c)) + (1-F_i) (H_d c + (1-H_d) (F_d(b+c) + (1-F_d)(2b+c)))$
- E. 以上都不对

C

填空

55. 一台单处理机可以以标量方式运行，也可以以向量方式运行。假设向量方式运算速度是标量方式的 9 倍。某基准程序在此处理机上运行时间为 T ，向量方式占整个运行时间的 25%，其余的时间则以标量方式运行。

1) 上述程序中向量化代码所占的比例是 [填空 1]。 75% $0.25 + 0.75/9$

2) 我们通过硬件优化，使向量方式运算速度加倍，此时可达到的加速比（与全部用标量方式运行相比）是 [填空 2]。 3.43 1.81 倍

3) 如果要达到与 2) 硬件优化相同的加速比，采用编译优化的软件方法，用向量化编译器支持同样的基准程序，其新的向量化代码所占的比例是 [填空 3]。 79.7% $3.43 = \frac{1}{1-\alpha+\frac{\alpha}{9}}$ $\alpha =$

56. 假设一个 Cache 共有 4 个块，每块大小为 1 个字节。Cache 采用 LRU 替换策略，初始时空。当程序执行过程中访存的字节地址序列为：0, 1, 2, 4, 1, 2, 0, 4 时，如果采用直接映射方式，Cache 的命中次数为 [填空 1] 次；如果采用 2 路组相连（用地址第 0 位作 index），Cache 的命中次数为 [填空 2] 次。

57. 假设有一个两级 Cache，L1 Cache 的命中时间是 2 个时钟周期，缺失率为 10%。L2 Cache 的命中时间为 10 个时钟周期，局部缺失率（相对于所有 L2 Cache 访问）为 20%，缺失代价为 100 个时钟周期，则平均存储器访问时间（AMAT）为 [填空 1] 个时钟周期。 $2 + 10\% \times (10 + 20\% \times 100)$

58. 在 CRAY-1 机器上计算 $D=A \times (B+C)$ ，其中 A、B、C、D 都是长度为 64 的向量，并且 A、B、C 已经存放在相应的向量寄存器中。CRAY-1 的向量寄存器长度为 64，所用浮点功能部件的执行时间分别为：相加 5 拍，相乘 7 拍，把向量数据元素送往功能部件以及把结果存入向量寄存器都需要 1 拍时间。不采用向量链接方式，完成该计算所需的最短拍数是 [填空 1] 拍；采用向量链接方式，完成该计算所需的最短拍数是 [填空 2] 拍。

$$(1+5+1) + 64-1 + (1+7+1) + (64-1) = 7 + 63 + 9 + 63$$

主观题

59. 设计一个处理器 Cache，该 Cache 纯数据容量 64K 字节，采用 4 路组相联，Cache 块大小为 64 字节，每个块有 1 个有效 (Valid) 位和 1 个脏 (Dirty) 位。假设处理器的物理地址总共 46 位，试计算：

1、地址的 tag 域、index 域以及 block offset 域的位数。（2 分）

2、为实现该 Cache，需要 SRAM 的总的位数（含纯数据 Cache 和附加信息位）是多少（以 Kbits 为单位）？（3 分）

3、假定 Cache 初始为空，整数大小为 4 字节，数组是 Cache 块对齐的。考虑以下两种对已访问位置计数的代码，请问哪段代码 Cache 命中率更高？说明原因，指明是由于哪种类型的 Cache 缺失造成的？

$$1. \frac{64k}{64} = 1024 \text{ 块}$$

$$4 \text{ 路组相联: } \frac{1024}{4} = 256 = 2^8$$

Index: 8 位

$$64 = 2^6 \quad \text{Block offset: 6 位} \quad \text{Tag: } 46 - 8 - 6 = 32 \text{ 位}$$

$$64kB + 2kbits$$

(3分) 3. B命中更高, A结构体连续存放, 每次读 visit 会将 danger 一起读入, 造成额外强制性读失

程序 A

```

1  typedef struct {
2      ... // other struct members
3      int visited;
4      int danger;
5  } location;
6
7  location locs[NUM_LOCS];
8  for (int i = 0; i < NUM_LOCS; i++)
9      if (locs[i].visited) count++;
        
```

步长为2访问
 $\frac{1}{8}$

程序 B

```

1  int visited[NUM_LOCS];
2  int danger[NUM_LOCS];
3  for (int i = 0; i < NUM_LOCS; i++)
4      if (visited[i]) count++;
        
```

步长为1访问
 $\frac{1}{16}$

ROB: 按序提交

60. 假设一个 RISC-V 处理器, 采用支持精确中断和推测执行的 Tomasulo 算法。假设该处理器有 2 个 FP 加法器, 1 个 FP 乘法/除法器, 2 个地址加法器, 2 个整数运算单元。在一个给定的周期内, 只能发射 (Issue) 一条指令, 只有一条指令可以访问数据 Cache, 只有一条指令可以写 CDB, 只有一条指令可以提交 (Commit)。并且:

- FP 加减法运算需要 3 个周期
- FP 乘法/除法需要 10 个周期
- 整数操作和地址计算 (Load 和 Store 指令在 EXE 段执行地址计算) 需要 1 个周期
- 访问数据 Cache 需要一个周期

在下表中填写每条指令执行时各阶段的时钟周期号 (注意: CC_{执行} = CC_{结束} - CC_{开始} + 1)

指令	发 射 Issue at	执 行 EXE		读存储器 Read Mem at	写 CDB Write CDB at	提交 Commit at
		开始	结束			
1 add x3, x3, x2	1	2	2		3	4
2 fdiv.s f4, f2, f8	2	3	12		13	14
3 fadd.s f6, f10, f4	3	14	16		17	18
4 sub x3, x3, x4	4	5	5		6	19
5 fsw f6, 4(x3)	5	18	18		19	20 (write mem)
6 flw f2, 8(x3)	6	7	7	8	9	21
7 fsub.s f10, f8, f2	7	10	12		14	22
8 fadd.s f14, f6, f8	8	18	20		21	23

61. 使用 2-bit 饱和计数器实现分支预测。当计数器的值为 0 或 1 的时候预测为 not taken; 计数器的值为 2 或 3 的时候预测为 taken。实际分支为 taken 时, 计数器+1 (最大为 3), 否则, 实际分支为 not taken 时, 计数器-1 (最小为 0)。假定初始状态下饱和计数器的值是 0, 多个分支指令对应的饱和计数器相互独立, 预测结果互不干扰。现有 Fibonacci 数列前几项的计算程序如下 (左侧为高级语言, 右侧

为对应的汇编语言)：

1 procedure Compute(void):	1 . compute:	
2 a ← 0	2. mov \$0, %eax	; %eax = 0
3 b ← 1	3. mov \$1, %ebx	; %ebx = 1
4 for j ← 3 downto 0 do	4. mov \$3, %ecx	; %ecx = 3
5 begin	5. loop:	
6 c ← a + b	6. add %eax, %ebx, %edx	; %edx = %eax + %ebx
7 b ← b + c	7. add %edx, %ebx, %ebx	; %ebx = %edx + %ebx
8 a ← c	8. mov %edx, %eax	; %eax = %edx
9 end	9. sub \$1, %ecx	; %ecx = %ecx - 1
10 procedure Main(Void):	10. jns loop	; jump to loop if %ecx >= 0
11 For i ← 100 downto 1 do	11. ret	
12 begin	13. main:	
13 call Compute	14. mov \$100, %ecx	; %ecx = 100
14 end	15. loop2:	
	16. call compute	; call subroutine
	17. sub \$1, %ecx	; %ecx = %ecx - 1
	18. jnz loop2	; jump to loop2 if %ecx != 0
	19. ret	

计数: 0 1 2 3 2 3 3 3 2 3 3
实际: 右侧的汇编代码采用 AT&T 格式, 在函数调用的时候忽视了栈的变化。\$imm 表示一个立即数, %reg 表示一个通用寄存器。请问在执行右侧汇编语言 Main 例程时, 采用 2-bit 饱和计数器进行分支预测的情况下, 右图中第 10 行 jns 指令位置会发生多少次预测错误? 请给出计算依据。
T T F 3+9

实际:

T	T	T	F	T	T	F	T	T	F	T	T	T
---	---	---	---	---	---	---	---	---	---	---	---	---

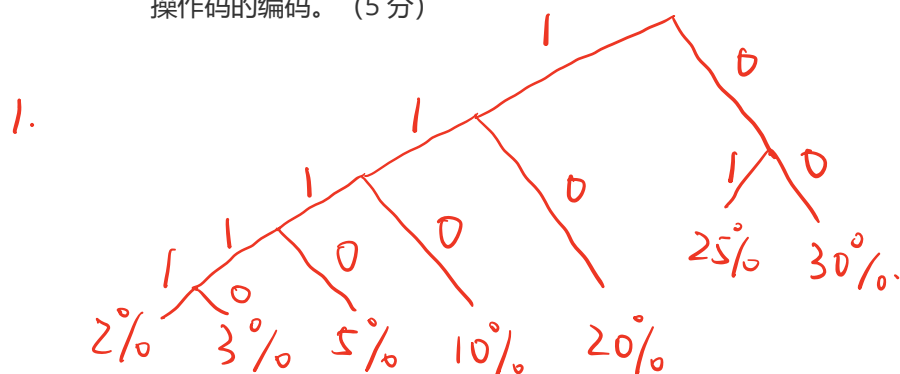
$3 + 9 \times 1 = 102\%$

预测:

F	F	T	T	T	T	T	T	T	T	T	T	T
---	---	---	---	---	---	---	---	---	---	---	---	---

= 102次

62. 一台模型机共有 7 条指令，各指令的使用频率分别为：35%，25%，20%，10%，5%，3% 和 2%，有 8 个通用数据寄存器，2 个变址寄存器。1、要求操作码的平均长度最短，请设计操作码的编码，并计算所设计操作码的平均长度。（5 分）2、设计 8 字长的寄存器-寄存器型指令 3 条，16 位字长的寄存器-存储器型变址寻址方式指令 4 条，变址范围不小于 ± 127 。请设计指令格式，并给出各字段的长度和操作码的编码。（5 分）



编码:

30%	00
25%	01
20%	10
10%	110
5%	1110

平均码长:

$$(30\% + 25\% + 20\%) \times 2 + 10\% \times 3 + 5\% \times 4 + (3\% + 2\%) \times 5 = 1.5 + 0.3 + 0.2 + 0.25$$

$$3\%: 11110 = 2.25$$

$$2\%: 11111$$

2. 寄存器-寄存器 3条:

操作码	源寄存器	目的寄存器
2位	3位	3位

操作码: 00, 01, 10

寄存器-存储器 4条:

操作码	通用寄存器	变址寄存器	地址偏移
4位	3位	1位	8位

操作码: 1100, 1101, 1110, 1111