

ICS Seminar Week6 Prep

吴思衡 王效乐 许珈铭

2023.10.23

Rules

remainder <- ordinal number in WeChat Group % 4

for all questions do

 if question number % 4 == remainder then

 you should work on it

 end

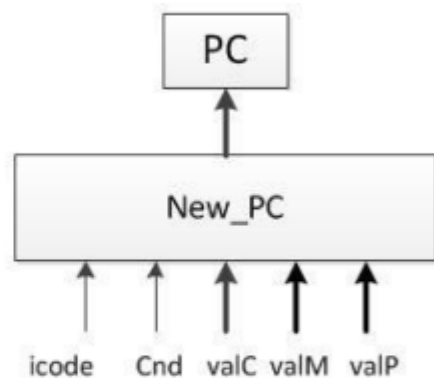
end

Sequential

14. 在 Y86 的 SEQ 实现中, PC (Program Counter, 程序计数器) 更新的逻辑

结构如下图所示, 请根据 HCL 描述为①②③④选择正确的数据来源。

```
Int new_pc = [  
  # Call.  
  Icode = ICALL : ①;  
  # Taken branch.  
  Icode = IJXX && Cnd : ②;  
  # Completion of RET instruction.  
  Icode = IRET : ③;  
  # Default.  
  1 : ④;
```



其中: Icode 为指令类型, Cnd 为条件是否成立, valC 表示指令中的常数值, valM 表示来自返回栈的数据, valP 表示 PC 自增。

- A. valC, valM, valP, valP
- B. valC, valC, valP, valP
- C. valC, valC, valM, valP
- D. valM, valC, valC, valP

C

10. 在书中 Y86 的 SEQ 实现下, 以下哪一条指令是现有信号通路能完成的:

A. `iaddq rA, V`: 将立即数 `V` 与 `R[rA]` 相加, 其中 `rB` 域设为 `F`, 结果存入寄存器 `rA`

B. `mmovq rA, rB`: 将 `R[rA]` 存的地址开始的 8 字节数据, 移动到 `R[rB]` 存的地址

C. `leave`: 相当于先执行 `rrmovq %rbp, %rsp`, 再执行 `popq %rbp`

D. `enter`: 相当于先执行 `pushq %rbp`, 再执行 `rrmovq %rsp, %rbp`

Pipeline

10、下面对流水线技术的描述，正确的是：（ ）

- A. 流水线技术不仅能够提高指令的吞吐率，还能减少单条指令的执行时间。
- B. 不断加深流水线级数，总能获得性能上的提升。
- C. 流水级划分应尽量均衡，吞吐率会受到最慢的流水级影响。
- D. 指令间数据相关可能引发数据冒险，可通过数据转发或暂停流水线来解决。

13. 关于流水线技术的描述，错误的是：

- A. 流水线技术能够提高执行指令的吞吐率，但也同时增加单条指令的执行时间。
- B. 减少流水线的级数，能够减少数据冒险发生的几率。
- C. 指令间数据相关引发的数据冒险，都可以通过 data forwarding 来解决。
- D. 现代处理器支持一个时钟内取指、执行多条指令，会增加控制冒险的开销。

9. 下面对流水线技术的描述，正确的是：
- A. 流水线技术不仅能够提高执行指令的吞吐率，还能减少单条指令的执行时间。
 - B. 不断加深流水线级数，总能获得性能上的提升。
 - C. 流水级划分应尽量均衡，吞吐率会受到最慢的流水级影响。
 - D. 指令间的数据相关可能会引发流水线停顿，但总是可以通过调度指令来解决。

11. 关于流水线技术的描述，错误的是：

- A. 流水线技术能够提高执行指令的吞吐率，但也同时增加单条指令的执行时间
- B. 增加流水线级数，不一定能获得总体性能的提升
- C. 指令间数据相关引发的数据冒险，不一定可以通过暂停流水线来解决。
- D. 流水级划分应尽量均衡，吞吐率会受到最慢的流水级影响，均衡的流水线能提高吞吐量。

12、关于流水线技术的描述，正确的是：

- A. 指令间数据相关引发的数据冒险，一定可以通过暂停流水线来解决。
- B. 流水线技术不仅能够提高执行指令的吞吐率，还能减少单条指令的执行时间。
- C. 增加流水线的级数，一定能获得性能上的提升。
- D. 流水级划分应尽量均衡，不均衡的流水线会增加控制冒险。

答：（ ）

7、下列说法正确的是：

- A. 在SEQ机器中，我们采用预测跳转总是选择（always taken）的策略比从不选择（never taken）的策略要略好。
- B. 流水级划分应尽量均衡，不均衡的流水线会增加控制冒险。
- C. 如果一台机器的CPI小于1，则它必然不是普通流水线结构。
- D. 由于rrmovq %rax, %rax不影响标记位，所以可使用其代替nop指令。

12. 一个功能模块包含组合逻辑和寄存器，组合逻辑单元的总延迟是 100ps ，单个寄存器的延时是 20ps ，该功能模块执行一次并保存执行结果，理论上能达到的最短延时和最大吞吐分别是多少？
- A. 20ns , 50GIPS
 - B. 120ns , 50GIPS
 - C. 120ns , 10GIPS
 - D. 20ps , 10GIPS

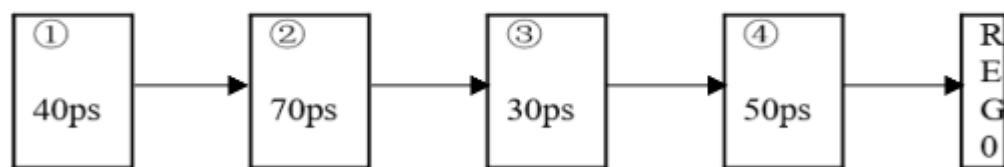
10、若处理器实现了三级流水线，每一级流水线实际需要的运行时间分别为 2ns 、 2ns 和 1ns ，则此处理器不停顿地执行完毕 10 条指令需要的时间为：

- A. 21ns
- B. 22ns
- C. 23ns
- D. 24ns

答：（ ）

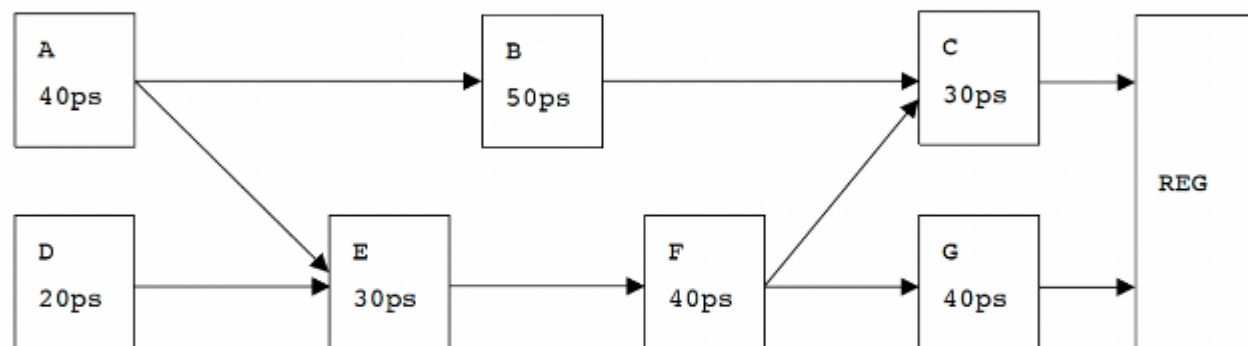
12. 若处理器实现了三级流水线，每一级流水线实际需要的运行时间分别为 1ns 、 2ns 和 3ns ，则此处理器不停顿地执行完毕 10 条指令需要的时间为：
- A. 21 ns B. 12 ns C. 24 ns D. 36 ns

11. 如下图所示，①~④为四个组合逻辑单元，对应的延迟已在图上标出，REG0 为一寄存器，延迟为 20ps。通过插入**额外的 2 个**流水线寄存器 REG1、REG2（延迟均为 20ps），可以对其进行流水化改造。改造后的流水线的吞吐率最大为 _____ GIPS。



- A. 7.69 B. 8.33 C. 10.00 D. 11.11

8. A-G 为 7 个基本逻辑单元，下图中标出了每个单元的延迟，以及用箭头标出了单元之间所有的依赖关系。寄存器的延迟均为 20ps ，在图中以 REG 符号表示。假设流水线寄存器只能添加在有直接依赖关系的基本逻辑单元之间，而不能在 C 或 G 与 REG 之间。以下说法正确的是：



- A. 原电路的吞吐量 (throughput) 舍入后大约是 $1000/150=6.667$ GIPS。
- B. 将该电路改造成 2 级流水线有 8 种方法
- C. 如果将该电路改造成 3 级流水线，延迟最小可以到 80ps 。
- D. 不论实现该电路时遇到怎样的数据冒险和控制冒险，一定可以对流水线寄存器使用暂停 (stalling) 解决。

10. 在 Y86 的 SEQ 实现中, 对仅考虑 IRMMOVQ, ICALL, IPOPOPQ, IRET 指令, 对 mem_addr 的 HCL 描述正确的是:

```
word mem_addr = [  
    icode in { (1), (2) } : valE;  
    icode in { (3), (4) } : valA;  
];
```

- A. (1) IRMMOVQ (2) IPOPOPQ (3) IRET (4) ICALL
- B. (1) IRMMOVQ (2) IRET (3) IPOPOPQ (4) ICALL
- C. (1) ICALL (2) IPOPOPQ (3) IRMMOVQ (4) IRET
- D. (1) IRMMOVQ (2) ICALL (3) IPOPOPQ (4) IRET

6、在Y86-64的PIPE实现中，仅考虑ICALL、IPOPQ、IPUSHQ、IRET指令，对mem_addr的HCL描述正确的是：

```
word mem_addr = [  
    M_icode in { ①, ② } : M_valE;  
    M_icode in { ③, ④ } : M_valA;  
];
```

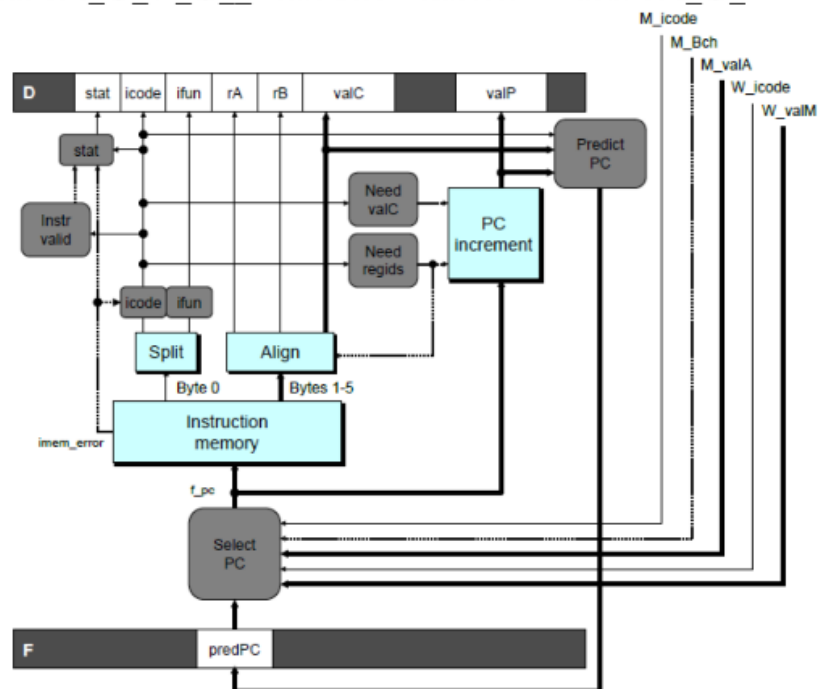
- A. ①IPUSHQ ②ICALL ③IPOPQ ④IRET
- B. ①IPUSHQ ②IRET ③ICALL ④IRET
- C. ①IPUSHQ ②IPOPQ ③IRET ④ICALL
- D. ①IPUSHQ ②IRET ③IPUSH ④ICALL

10. Y86 指令 `popl rA` 的 SEQ 实现如下图所示，其中 ❶ 和 ❷ 分别为：

Fetch	$\text{icode:ifun} \leftarrow M_1[\text{PC}]$ $\text{ra:rb} \leftarrow M_1[\text{PC}+1]$ $\text{valP} \leftarrow \text{❶}$
Decode	$\text{valA} \leftarrow R[\%esp]$ $\text{valB} \leftarrow R[\%esp]$
Execute	$\text{valE} \leftarrow \text{❷}$
Memory	$\text{valM} \leftarrow M_4[\text{valA}]$
Write Back	$R[\%esp] \leftarrow \text{valE}$ $R[\text{ra}] \leftarrow \text{valM}$
PC Update	$\text{PC} \leftarrow \text{valP}$

- A) $\text{PC} + 4 \quad \text{valA} + 4$
- B) $\text{PC} + 4 \quad \text{valA} + (-4)$
- C) $\text{PC} + 2 \quad \text{valB} + 4$
- D) $\text{PC} + 2 \quad \text{valB} + (-4)$

11. 流水线数据通路中的转移预测策略为总是预测跳转。如果转移预测错误，需要恢复流水线，并从正确的目标地址开始取值。其中，用来判断转移预测是否正确的信号是_①_和_②_，用来获得正确的目标地址的信号是_③_。



- A. ① M_icode ② M_Bch ③ M_valA
- B. ① W_icode ② M_Bch ③ M_valA
- C. ① W_icode ② M_Bch ③ W_valM
- D. ① M_icode ② M_Bch ③ W_valM

A

9. 在课本 Y86-64 的 PIPE 上执行以下的代码片段，一共使用到了（ ）次数据转发。假设在该段代码执行前和执行后 PIPE 都执行了足够多的 nop 指令。

```
mrmovq 0(%rdx), %rax
addq   %rbx, %rax
mrmovq 8(%rdx), %rcx
addq   %rcx, %rax
irmovq $10, %rcx
addq   %rcx, %rax
rmmovq %rax, 16(%rdx)
```

A. 3 B. 4 C. 5 D. 6

(11-13)、在教材所描述的流水线处理器（the PIPE processor）上分别运行如下四段Y86程序代码。请分析其中数据冒险的具体情况，并回答后续3个小题。

<pre>#Program 1: mrmovl 8(%ebx), %cdx rmmovl %cdx, 16(%ecx)</pre>	<pre>#Program 2: mrmovl 8(%ebx), %cdx nop rmmovl %edx, 16(%ecx)</pre>
<pre>#Program 3: mrmovl 8(%ebx), %edx nop nop rmmovl %cdx, 16(%ecx)</pre>	<pre>#Program 4: mrmovl 8(%ebx), %edx nop nop nop rmmovl %edx, 16(%ecx)</pre>

11、对于每段程序，请指出是否会因为数据冒险导致流水线停顿（Stall）。

Program 1: (), Program 2: (), Program 3: (), Program 4: ();

A. Stall B. No-Stall

12、对于每段程序，请指出流水线处理器内是否会产生数据转发（Forwarding）。

Program 1: (), Program 2: (), Program 3: (), Program 4: ();

A. Forwarding B. No-Forwarding

13、对于每段程序，请指出流水线处理器内使用哪个信号进行数据转发，如果不进行数据转发，则用none表示。

Program 1: (), Program 2: (), Program 3: (), Program 4: ();

A. m_valM B. W_valM C. none

A, B, B, B
A, A, A, B
A, A, B, C