

# 流水线 CPU 例题

## 流水线原理

1. Branch prediction in a RISC-V pipelined processor improves performance by guessing the outcome of branch instructions and pre-fetching the next instructions, but it introduces a risk of performance degradation due to branch mispredictions.
2. The execution time of a single instruction in a pipelined RISC-V processor is faster than in a single-cycle processor because it moves through multiple stages more quickly.
3. What is the minimum number of cycles needed to completely execute  $n$  instructions on a CPU with a  $k$  stage pipeline? Justify your formula.
4. Add NOP instructions to the code below so that it will run correctly on a pipeline that does not handle data hazards.

```
addi x11, x12, 5
add x13, x11, x12
addi x14, x11, 15
add x15, x13, x12
```

- 5.

## 结构冒险

1. 如果我们采用 a mixed cache 用来取指和读内存，会有什么冒险？
  - A. 啥都没有
  - B. Data hazard
  - C. Control hazard
  - D. Structural hazard

## 数据冒险

1. Consider the following loop.

```
LOOP: ld x10, 0(x13)
      ld x11, 8(x13)
      add x12, x10, x11
      subi x13, x13, 16
      bneq x12, LOOP
```

Assume that perfect branch prediction is used (no stalls due to control hazards), that there are no delay slots, that the pipeline has full forwarding support, and that branches are resolved in the EX (as opposed to the ID) stage. Show a pipeline execution diagram for the first two iterations of this loop.

2. Problems in this exercise refer to the following sequence of instructions, and assume that it is executed on a five-stage pipelined datapath:

```
add x15, x12, x11
ld x13, 4(x15)
ld x12, 0(x2)
or x13, x15, x13
sd x13, 0(x15)
```

1. If there is no forwarding or hazard detection, insert NOPs to ensure correct execution.
2. Now, change and/or rearrange the code to minimize the number of NOPs needed. You can assume register x17 can be used to hold temporary values in your modified code.  
请写出指令间的数据冒险原因。
3. If the processor has forwarding, but we forgot to implement the hazard detection unit, what happens when the original code executes?
4. If there is forwarding, for the first seven cycles during the execution of this code, specify which signals are asserted in each cycle by hazard detection and forwarding units in Figure 4.59.

Mux control	Source	Explanation
ForwardA = 00	ID/EX	The first ALU operand comes from the register file.
ForwardA = 10	EX/MEM	The first ALU operand is forwarded from the prior ALU result.
ForwardA = 01	MEM/WB	The first ALU operand is forwarded from data memory or an earlier ALU result.
ForwardB = 00	ID/EX	The second ALU operand comes from the register file.
ForwardB = 10	EX/MEM	The second ALU operand is forwarded from the prior ALU result.
ForwardB = 01	MEM/WB	The second ALU operand is forwarded from data memory or an earlier ALU result.

## 控制冒险

1. Consider a five-stage pipeline CPU. It does not consider structural hazard and allows the register file to be read and written in the same clock cycle. Branches are resolved in the ID

Stage. And consider the following code segment:

Loop: lw x1, 0(x4)	Instr. 1/7
lw x2, 400(x4)	Instr. 2
add x3, x1, x2	Instr. 3
sw x3, 0(x4)	Instr. 4
addi x4, x4, -4	Instr. 5
bne x0, x4, loop	Instr. 6

a) No forwarding mechanism is used. Show a pipeline execution diagram for the first loop iteration.

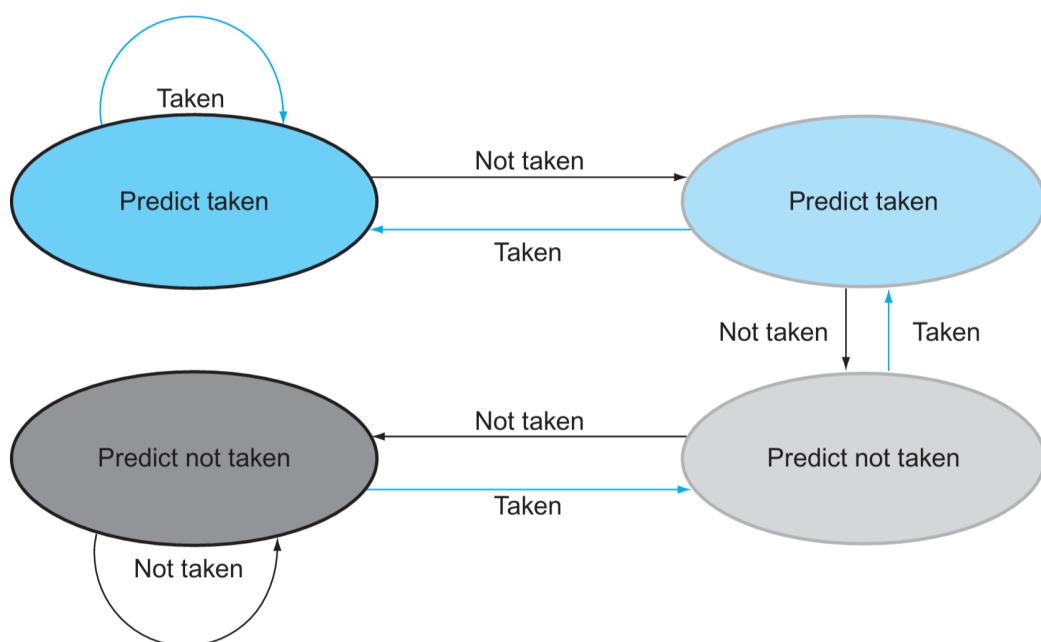
b) Forwarding mechanism is used. Show a pipeline execution diagram for the first loop iteration.

Pipeline execution table sample: 不考虑对 ID 阶段支持 forwarding 这点原题没有提到, 请务必在考场问监考老师

Clock Cycle	1	2	3	4	5	...
1	F	D	E	M	W	
2		F				

2. This exercise examines the accuracy of various branch predictors for the following repeating pattern (e.g., in a loop) of branch outcomes: T, NT, T, T, NT.

1. What is the accuracy of always-taken and always-not-taken predictors for this sequence of branch outcomes?
  
2. What is the accuracy of the 2-bit predictor for the first four branches in this pattern, assuming that the predictor starts off in the bottom left state from Figure 4.61 (predict not taken)?



3.

# 冒险

---

1.

**流水线：**先给一张五级流水线CPU的图，再给一段汇编程序 I1: addi x5, x0, 1 I2: addi x6, x0, 1 I3: add x7, x5, x6 I4: add x5, x6, x7 I5: add x6, x7, x5 I6: add x7, x5, x6 ..... 如是作无限循环  
(1) 这段程序是干啥的 (2) 运行这段程序会在哪里发生hazard，在五级流水线没有任何针对hazard的改进的情况下，求CPI，为什么 (3) 求用前递解决数据冲突之后的 CPI (4) 分别问 I2 I3 I4 的两个操作数的前递来源，从ID/EX、EX/MEM、MEM/WB种选择 (5) 给你一个新指令Jr r1, r2, imm，要求能实现  $r1 \leftarrow PC + 4$ ,  $PC \leftarrow Memory[r2+imm]$ ，指令的格式应当如何设计，参数都放到哪个位置，需要对流水线作何改动，在原图上改