
CS 61C RISC-V单周期数据路径2020年秋季讨论

7：2020年10月12日

1 预检查

本节旨在对您进行概念性检查，以确定您是否从概念上理解此主题以及是否对此主题有任何误解。请回答以下问题的对/错，并附上解释：

1.1 单周期数据路径利用每个指令的所有硬件单元。

错。所有单元在每个周期中都是活动的，但它们的输出可能被控制信号忽略（选通）。

1.2 可以并行执行单周期数据路径的各级以加速单指令的执行。

错。每个级取决于由其之前的级产生的值（例如，指令解码取决于所获取的指令）。

1.3 组合逻辑仅用于指令解码阶段。

错。其它级也执行组合逻辑（用于指令获取、存储器写入、寄存器更新的多路复用器;执行期间的ALU操作）。

2 单周期CPU

2.1 对于本工作表，我们将使用最后一页的单周期CPU数据路径。

- (a) 在数据路径上，用数据路径组件的名称填入每个圆框，用控制信号的名称填入每个方框。
- (b) 解释每个数据路径阶段发生的情况。

IF 指令获取

将地址发送到指令存储器（IMEM），并在该地址读取IMEM。

ID指令解码

从指令位生成控制信号，从RegFile生成立即数和读取寄存器。

EX执行

执行ALU操作，并进行分支比较。

MEM存储器

读取或写入数据存储器（DMEM）。

WB回写

将 PC + 4（ALU 操作的结果）或内存中的数据写回 RegFile。

2.2 根据上一页的数据路径，用每条指令的控制信号填写下表。在可能的情况下，使用 * 表示此信号是什么无关紧要（例如，让值成为它想要的任何值不会影响指令的执行）。如果信号的值对正确执行很重要，但可以变化，请列出所有值（例如，对于可能具有0和1值的信号，写入0/1）。

	溴当量	BrLT	PCSel	伊梅尔	布鲁尼	ASel	BSEL	ALUSEL	内存RW	注册WEn	WBSel
加上	*	*	0 (PC + 4)	*	*	0 (注册)	0 (注册)	加上	0	1	1 (ALU)
ori	*	*	0	I	*	0 (注册)	1 (免疫)	或	0	1	1 (ALU)
长波	*	*	0	I	*	0 (注册)	1 (免疫)	加上	0	1	2 (MEM)
SW	*	*	0	S	*	0 (注册)	1 (免疫)	加上	1	0	*
贝克	1/0	*	1/0	小型	*	1 (PC)	1 (免疫)	加上	0	0	*
贾尔	*	*	1 (ALU)	乌杰	*	1 (PC)	1 (免疫)	加上	0	1	0 (PC + 4)
蓝色的	*	1/0	1/0	小型	1	1 (PC)	1 (免疫)	加上	0	0	*

2.3 计时方法

- 状态元素是连接到时钟的元素（由底部的三角形表示）。每个状态元件的输入信号必须在每个上升沿之前稳定。
- 关键路径是电路中状态元件之间的最长延迟路径。电路的时钟不能比这更快，因为任何更快的速度都意味着不能保证正确值在分配的时间内到达状态元件。如果我们将寄存器放置在关键路径中，则可以通过减少寄存器之间的逻辑数量来缩短周期。

在本练习中，假设数据路径中每个阶段的延迟如下：

如果：200 ps ID：100 ps EX：200 ps MEM：200 ps WB：100磅/秒

(a) 标记以下指令使用的数据路径的阶段，并计算执行该指令所需的总时间。

	如果	编号	出口	内存	全文	总时间
加	X	X	X		X	600磅/秒
上	X	X	X		X	600磅/秒
ori	X	X	X	X	X	800磅/秒
长	X	X	X	X		700磅/秒
波	X	X	X			500磅/秒
S	X	X	X			600磅/秒
W	X	X	X			500磅/秒
贝	X	X	X			500磅/秒
克	X	X	X			500磅/秒
贾	X	X	X			500磅/秒
尔	X	X	X			500磅/秒
蓝	X	X	X			500磅/秒
色	X	X	X			500磅/秒
的	X	X	X			500磅/秒

- (b) 哪些指令执行关键路径?加载字（lw），使用所有5个阶段。
- (c) 这条单周期数据路径的最快时钟是多少?

$\frac{1}{800}$ 皮秒= $\frac{1}{800 \times 10^{-12}}$ 秒=1,250,000,000s=1.25GHz

(d) 为什么单周期数据路径效率低下?

在任何给定时间，单周期数据路径的大部分都处于闲置状态。此外，即使不是每条指令都执行关键路径，数据路径的时钟只能与最慢的指令一样快。

(e) 如何提高其性能?流水线的目的是什么?

性能可以通过流水线技术来提高，或者在级之间放置寄存器，从而减少寄存器之间的组合逻辑数量，从而实现更快的时钟时间。

4 RISC-V单周期数据路径

