

《计算机组成原理》第五次作业

网络空间安全学院 信息安全 陆皓喆 2211044

4.1

考虑如下指令：

指令： `and [rd],rs1,rs2`

解释： `Reg[rd]=Reg[rs1]+Reg[rs2]`

4.1.1

题目

对上述指令，图4-10中的控制单元将产生哪些控制信号？

解答

会产生以下几种控制信号：

RegWrite	ALUSrc	ALUoperation	MemWrite	MemRead	MemToReg
true	0	"and"	false	false	0

4.1.2

题目

对上述指令，将用到哪些功能单元？

解答

1. Registers
2. ALUsrc mux
3. ALU
4. the MemToReg mux

4.1.3

题目

哪些功能单元会产生输出，但输出不会被以上指令用到？对以上指令，哪些功能单元不产生任何输出？

解答

每个功能单元都会产生一些输出，但是不一定会被指令用到，比如说在本题中，我们没有使用 DataMemory 和 ImmGen

4.3

考虑下面的指令组合：

R-type	I-type(non-lw)	Load	Store	Branch	Jump
24%	28%	25%	10%	11%	2%

4.3.1

题目

多少指令会用到数据存储器？

解答

只有load指令和store指令会使用数据存储器，所以应该是25%+10%=**35%**

4.3.2

题目

多少数据会用到指令寄存器？

解答

所有的指令都需要使用指令寄存器，因为每一条指令在执行前，都需要从指令存储器中取出，本题的答案应该是**100%**

4.3.3

题目

多少指令会用到符号扩展？

解答

只有R型指令和J型指令不需要符号扩展，所以答案应该是100%-24%-2%=74%

4.3.4

题目

不需要符号扩展输出的时钟周期内，符号扩展部件在做什么？

解答

符号扩展在每个周期中产生一个输出，如果不需要符号扩展的话，程序一般会直接忽略它，所以如果不需要符号扩展，该部件依旧是产生了结果的

4.5

本题将详细考察一条指令在单周期数据通路中如何执行。本题中的问题都是在一个时钟周期内，其中处理器取入的指令字为：0x00c6ba23

4.5.1

题目

对该指令，ALU控制单元的输入值是什么？

解答

该指令应该是：sd x12, 20(x13)

所以输入的值应该是：

ALUOp	ALU Control Lines
00	0010

4.5.2

题目

该指令执行后，新的PC地址是多少？高亮标出该值决定的路径。

解答

新的PC地址应该是PC+4

该指令从原始的PC出发，通过ADD加法器，与4相加，得到PC+4，然后通过branch mux，返回到PC

4.5.3

题目

给出该指令执行过程中，每个多路选择器的输入和输出值。列出寄存器Reg[xn]的输出值。

解答

ALUsrc: 输入Reg[x12]和0x0000000000000014, 输出0x0000000000000014

MemToReg: 输入Reg[x13] + 0x14 and undefined, 输出undefined

Branch: 输入PC+4和PC+0x28

4.5.4

题目

ALU和两个加法单元的输入值是什么？

解答

ALU输入 Reg[x13]和0x0000000000000014

第一个加法单元(ADD)输入PC和4

第二个加法单元(分支)输入PC和0x0000000000000028

4.5.5

题目

寄存器单元的所有输入值是什么？

解答

读寄存器1: 输入0x13, 这是基址

读寄存器2: 输入0x12, 这是需要被存储的数据

写寄存器: 一般来说不需要回写, 一般赋值为0x0

写数据: 不需要进行回写

RegWrite:输入false, 一般来说不需要回写

4.7

本题假定在实现一个处理器的数据通路时，逻辑模块的延时如下：

I-Mem/D-Mem	Register File	Mux	ALU	Adder	Single gate	Register Read	Register Setup	Sign extend	Control
250ps	150ps	25ps	200ps	150ps	5ps	30ps	20ps	50ps	50ps

“寄存器读”是指时钟上升沿到来后寄存器值出现在输出端所需的时间。这个值只适用于PC。“寄存器建立”时间是指时钟上升沿到来之前，寄存器输入数据必需保持稳定的时间。这个值同时适用PC和寄存器堆。

4.7.1

题目

R型指令的延迟是多少？（即R型指令正确执行需要多长的时钟周期？）

解答

$$30 + 250 + 150 + 25 + 200 + 25 + 20 = 700ps$$

4.7.2

题目

lw指令的延迟是多少？

解答

$$30 + 250 + 150 + 25 + 200 + 250 + 25 + 20 = 950ps$$

4.7.3

题目

sw指令的延迟是多少？

解答

$$30 + 250 + 150 + 200 + 25 + 250 = 905ps$$

4.7.4

题目

beq指令的延迟是多少？

解答

$30 + 250 + 150 + 25 + 200 + 5 + 25 + 20 = 705ps$

4.7.5

题目

一条算术指令、逻辑指令或I型的移位（非road）指令的延迟是多少？

解答

$30 + 250 + 150 + 25 + 200 + 25 + 20 = 700ps$

4.7.6

题目

该CPU的最小时钟周期是多少？

解答

一般来说，最小时钟周期都需要参考lw指令的延迟，所以最小时钟周期为：950ps

4.10

处理器设计师改进处理器的数据通路时，通常依赖于开销/性能之间的折中。在下面三个问题中，假设从图4-21给出的数据通路开始，采用和练习题4.7中相同的延迟，开销如下表示：

i-Mem	Register File	Mux	ALU	Adder	D-Mem	Single Register	Sign extend	Sign gate	Control
1000	200	10	100	30	2000	5	100	1	500

假设将通用寄存器的数量从32个增加到64个时，将把lw和sw指令的数量减少12%，但将寄存器堆的延迟从150ps增加到160ps，开销从200增加到400（采用练习题4.8中给出的指令使用频度，忽略练习题2.18中讨论的指令集体系结构的其他影响）

4.10.1

题目

改进后的性能加速比能达到多少？

解答

将lw指令和sw指令的数量减少12%，可以减少我们的指令的时间

$$(0.12) \times (0.25 + 0.11) = 0.0432$$

但是，运行一条指令的时间从950ps到了960ps，所以现在运行一条指令的时间是

$$960 \times (1 - 0.0432) = 918.528ps$$

所以，现在与先前相比，是得到了加速的，加速比为

$$\frac{950}{918.528} = 1.034$$

所以，加速比为1.034

4.10.2

题目

比较性能的变化和开销的变化。

解答

原来的CPU成本是4496，现在的CPU成本是4696

具体来说：

1. PC: 5
2. I-Mem: 1000
3. Register file: 200到400
4. ALU: 100
5. D-Mem: 2000
6. Sign Extend: 100
7. Controls: 1000
8. adders: 30*2
9. muxes: 3*10
10. single gates: 1*1

$$\frac{4696}{4496} = 1.044$$

因此，性能每提升3%，在CPU上的花费就要上升4.4%

4.10.3

题目

结合计算出的开销/性能比，分别描述哪种情况下可以增加更多的寄存器，哪种情况下增加寄存器数量是没有意义的。

解答

一般来说，增加更多的寄存器是没有意义的，新的CPU成本一般来说都会更高

但是，这个计算方法没有考虑到性能的效用问题，比如说本题中我们可能就是需要那3%的性能来完成这个项目，所以为了完成项目，我们肯定是愿意付出4.4%甚至10%的成本的。

所以，在需要性能的替身时，我们可以选择使用更多的寄存器；但是如果性能已经达到了要求，或者再上升已经没有什么意义，这样的话就不需要再增加寄存器了。

4.15

lw是4.4节CPU中延迟最长的指令。如果修改lw和sw指令，去掉偏移量（offset）计算（例如，在使用lw或sw指令之前，必须将地址计算出来并存放在rs中），那么就没有指令同时使用ALU和数据存储器。这种修改可以减少时钟周期。然而，这也增加了指令数，因为许多lw和sw指令需要换成lw/add或sw/add的指令组合。

4.15.1

题目

新的时钟周期是多少？

解答

新的时钟周期将是750。

ALU和数据存储器现在将并行运行，因此我们已经有效地从关键路径中删除了两者中更快的那个(时间为200的ALU)，留下了时间较长的那一个

4.15.2

题目

由练习题4.7中的混合指令构成的程序在这个新CPU上运行更快还是更慢？快或慢多少？（为了简化，假设每条lw和sw指令分别由连续的两条指令替代。）

解答

应该是更慢了。

原来，运行一条指令的时间是950ps，在新的机器上编译时，我们需要运行1.35条指令，虽然单条指令的时间从950ps下降到750ps，但是我们经过计算发现

$$\frac{950}{750 \times 1.35} = 0.93$$

说明现在的运算速度相比之前更慢了，慢了7%

4.15.3

题目

影响程序在新CPU上的运行速度的主要因素是什么？

解答

主要因素是**load和store的指令数量**，以及如何使用这两类指令

如果一个程序的load和store都只使用了很少的地址数量，那么也可能会提升性能

4.15.4

题目

原来的CPU（如4-21所示）与新的CPU相比，哪个的整体设计更好？

解答

我认为原来的CPU设计的较好，因为对于lw和sw指令，其实在指令中是非常常见的，两者之和可以达到36%，所以使用指令的数量变大来换取运行速度的变慢，一般情况下是不可取的，比如说在第二小题中计算出来的结果，甚至比原先要慢7%左右。