单周期CPU 答案

数据通路

1. 参考 **指令需要的数据通路** 这个部分，计算指令的PC多路选择器并行的选了 PC+4，写入 PC 的时间短于其他，因此我们同时考虑寄存器-内存通路和寄存器-PC通路计算

$$\begin{gathered}
add: 5ns（PC\text{跳转}）+300ns（\text{取指}）+35ns（\text{取操作数}）+15ns（\text{进}ALU\text{前筛选}）+90ns（ALU）+15ns（\text{写入寄存器的筛选}）+20ns（\text{写入寄存器}）=480ns \\
addi: 5ns（PC\text{跳转}）+300ns（\text{取指}）+40ns(\text{立即数生成})+15ns（\text{进}ALU\text{前筛选}）+90ns（ALU）+15ns+20ns（\text{写入寄存器}）=485ns \\
lw: 5ns（PC\text{跳转}）+300ns（\text{取指}）+40ns(\text{立即数生成})+15ns（\text{进}ALU\text{前筛选}）+90ns（ALU）+300ns（\text{读取数据}）+15ns+20ns（\text{写入寄存器}）=785ns \\
sw: 5ns（PC\text{跳转}）+300ns（\text{取指}）+40ns(\text{立即数生成})+15ns（\text{进}ALU\text{前筛选}）+90ns（ALU）+200ns（\text{写入内存}）=650ns \\
beq: 5ns（PC\text{跳转}）+300ns（\text{取指}）+35ns（\text{取操作数}）+15ns（\text{进}ALU\text{前筛选}）+90ns（ALU）+5ns（\text{过}and\text{门}）+15ns(Mux)+15ns(PC\text{的}Mux)+20ns（\text{写}PC）=500ns \\
jal: 5ns（PC\text{跳转}）+300ns（\text{取指}）+40ns(\text{立即数生成})+20ns(PC+imm)+15ns(\text{过}MUX)+20ns（\text{写入}PC）=400ns
\end{gathered}$$

2. Single-cycle processors are typically limited in their clock speed due to the necessity of completing the longest instruction within one clock cycle, which affects the overall performance scalability of the processor. (T) 对的，时钟周期计算方法就是按照最慢的指令计算的
3. A   A需要load和store，Bstore，Cstore，Dstore
4. It is possible to execute the stages of the single cycle data path in parallel to speed up execution of a single instruction. (F) 必然造成数据错乱
5. 都不会，因为读寄存器肯定比立即数慢

控制信号取值

1.

inst	Regwrite	Immsel	ALUsrcB	Branch	ALU control	Jump	MenToReg	MemRW
Jal Label	1（要写返回地址）	11（就是UJ）	x（根本不进ALU）	0（非branch）	x（不过ALU）	1（jump指令）	2（来源是PC）	00（不要读写）
LW xd,D(xs)	1（要写寄存器xd）	00（I型指令）	1（要加立即数）	0（非branch）	0010（加法）	0（不改PC）	1（来源是memory）	10（要写不要读）

2. When silicon chips are fabricated, defects in materials (e.g., silicon) and manufacturing errors can result in defective circuits. A very common defect is for one signal wire to get “broken” and always register a logical 0. This is often called a “stuck-at-0” fault.
   - Which instructions fail to operate correctly if the MemToReg wire is stuck at 0?

     loads， jal,jalr,auipc,lui
   - Which instructions fail to operate correctly if the ALUSrc wire is stuck at 0?

     I-type, loads，stores

修改元件

1. Examine the difficulty of adding a proposed `lwi.d rd, rs1, rs2 (“Load With Increment”)` instruction to RISC-V. Interpretation: `Reg[rd]=Mem[Reg[rs1]+Reg[rs2]]`

   思路上直接用 add 的前一半通路算出 rs1+rs2，用这个结果进入 load 指令的通路

   1. Which new functional blocks (if any) do we need for this instruction?

      没有，直接用 load 的通路
   2. Which existing functional blocks (if any) require modification?

    修改控制信号即可

   3. Which new data paths (if any) do we need for this instruction?

      不需要，直接用add到ALU+load的后半段通路

   4. What new signals do we need (if any) from the control unit to support this instruction?

      不需要，直接ALU用 add 的信号

2. Examine the difficulty of adding a proposed `swap rs1, rs2` instruction to RISC-V. Interpretation: `Reg[rs2]=Reg[rs1]; Reg[rs1]=Reg[rs2]`

   将ALU的输出增加一个输出 rs2 的，ALU 结果输入 rs1，增加一条线选择 rs1 输入rs2，修改regs的输入逻辑

   当然我们也支持比如说ALU输出两个啦，直接控制信号改两个regwrite啦，都可以

   1. Which new functional blocks (if any) do we need for this instruction?

      不需要，改现有通路即可

   2. Which existing functional blocks (if any) require modification?

       寄存器变两个写端口

      ALU 改出能交换之后输出两个端口

   3. What new data paths do we need (if any) to support this instruction?

      将 rs1 结果增加控制信号输入 rs2

   4. What new signals do we need (if any) from the control unit to support this instruction?

      增加一个 reg write 信号用于第二个口子

   5. Modify Figure 4.21 to demonstrate an implementation of this new instruction

3. Examine the difficulty of adding a proposed `ss rs1, rs2, imm (Store Sum)` instruction to RISC-V. Interpretation: `Mem[Reg[rs1]]=Reg[rs2]+immediate`

   一种可行的思路是给ALU增加一个第一路选择rs1还是imm，ALU结果输入mem的writedata，这又需要一个选择器，rs1 输入 mem 的address

   1. Which new functional blocks (if any) do we need for this instruction?

      ALU加选择器和mem要加两个选择器

   2. Which existing functional blocks (if any) require modification?

      不需要改

   3. What new data paths do we need (if any) to support this instruction?

      选择器都要改，增加ALU输出到mem的，增加rs1到mem的

   4. What new signals do we need (if any) from the control unit to support this instruction?

       几个选择器都要

   5. Modify Figure 4.21 to demonstrate an implementation of this new instruction.

中断

1. D，mert可以自动返回