Logisim实现的单周期CPU设计文档

一、设计综述

1. 体系结构综述

   处理器为32位单周期处理器，支持指令集为：

   `add, sub, ori, lw, sw, beq, lui, nop`

   其中：

   - `nop`为空指令，机器码`0x00000000`，无实际作用
   - `add, sub` 为无符号加减法

2. 设计思路

   - 模块化层次化设计
   - 顶层驱动信号仅包括异步复位信号

二、关键模块设计

1. IFU 取指单元

   - 基本信息

序号	信号名	描述
1	clk	时钟信号
2	rst	异步复位至`0x00000000`
3	beq	检测是否是beq
4	eq	是否相等
5	imm32[31:0]	32位拓展后立即数
6	instr[31:0]	32位指令

序号	功能
1	异步复位到`0x00000000`
2	正常递增PC<=PC+4
3	在`beq`条件成立的情况下跳转

2. GRF 寄存器堆

   - 基本信息

序号	信号名	描述
1	clk	时钟信号
2	rst	异步复位信号

序号	信号名	描述
3	WE	写使能信号
4	WD	32位写入数据
5	W	5位写寄存器地址
6	R1	5位读寄存器地址
7	R2	5位读寄存器地址
8	R1(O)	R1读寄存器输出
9	R2(O)	R2读寄存器输出

序号	功能
1	从读寄存器读取数据
2	向写寄存器写入数据
3	零寄存器读取写入都不产生任何影响

3. ALU 算数逻辑单元

   - 基本信息

序号	信号名	描述
1	In1	输入端1
2	In2	输入端2
3	Op[2:0]	操作码
4	eq(O)	比较是否相等
5	zero(O)	输出是否是零
6	out(O)	输出运算结果

序号	功能
1	根据ALUop进行运算：
`000`	`addu`
`001`	`subu`
`010`	`and`
`011`	`or`
`101`	`sll in2 32`

4. DM 数据存储器

   - 基本功能

序号	信号名	描述

序号	信号名	描述
1	clk	时钟信号
2	rst	异步复位信号
3	WE	写使能信号
4	ld	读使能信号
5	WD[31:0]	32位写入数据
6	addr[4:0]	访问地址

5. Ext 拓展器： 同时支持无符号和符号拓展的拓展器

三、控制器设计

Controller 控制器设计

Signal Sheet and State:

Index	Instr	Opcode	func	RegWE	RegData	RegDes	ALUsrc	ALUop	DMRE	DMWE
1	add	000000	100000	1	1	1	1	add	No	No
2	lw	100011(i)	NA	1	0	0	0	add	Yes	No
3	sw	101011(i)	NA	0	x	x	0	add	No	Yes
4	sub	000000	100010	1	1	1	1	sub	No	No
5	ori	001101(i)	NA	1	1	0	0	or	No	No
6	beq	000100(i)	NA	0	x	x	1	sub	No	No
7	lui	001111(i)	NA	1	1	0	0	sll	No	No

思考题回答：

1. 单周期CPU与有限状态机： 在我们设计的单周期CPU中，本人设计的PC内置一个PC和一个NPC，其中使用内置的Program Counter负责存储当前状态，次态程序计数器NPC和指令存储器共同负责状态转移。
2. 关于存储器： IM使用ROM，DM使用RAM，GRF使用Register是合理的。IM在当前只需要读取，DM需要写入也需要读取，GRF需要方便的进行复位，读取，写入。这和Logisim中提供的组件是一致的。
3. 设计了一个Splitter，用于快速取得对应的机器语言字段
4. nop指令本身不会也不应当进行任何操作，因此需要排除在控制信号外。
5. 直观上看覆盖了所有操作，然而很明显没有减法sub，没有验证nop是否可用，指令之间也比较割裂，各种指令没有做到杂糅增加复杂度。检测不够强。