

《给定指令系统的处理器设计》

实验报告

郭李毅*

计算机科学与技术学院

2015年6月

摘要

本人于该实验中，独立设计了一个固定指令周期的简单指令集处理器，指令周期分为4个机器周期，分别为取指周期、运算周期、访存或访问外设周期及回写周期。该处理器为非流水线处理器，但由于指令信息在处理器中不断向下个模块传递，是一个可以改造为流水线处理器的锥形。该处理器能够实现具有10条指令的指令系统。处理器分为时钟管理模块、取指管理模块、运算管理模块、存取管理模块、回写管理模块及访存控制模块。实验台测试时，运行自己设计的一系列循环指令，覆盖指令集10条指令的许多组合方式，结果均正确，处理器内部暂存器较少，并可通过切换开关显示节拍、pc、ir、地址总线、数据总线、8个通用寄存器及模拟的I/O输出端口等信息。

*学号 1133710213 邮箱 475751511@qq.com

目录

1	系统整体框图	1
2	处理器接口说明	2
3	各模块说明	2
3.1	时钟管理模块	2
3.1.1	功能描述	2
3.1.2	接口说明	3
3.2	取指管理模块	3
3.2.1	功能描述	3
3.2.2	接口说明	4
3.3	运算管理模块	4
3.3.1	功能描述	4
3.3.2	接口说明	5
3.4	存取管理模块	5
3.4.1	功能描述	5
3.4.2	接口说明	5
3.5	回写管理模块	6
3.5.1	功能描述	6
3.5.2	接口说明	6
3.6	I/O控制模块	6
3.6.1	功能描述	6
3.6.2	接口说明	7
3.7	访存控制模块	7
3.7.1	功能描述	7
3.7.2	接口说明	8

4 仿真波形	8
4.1 测试方案	8
4.2 时钟管理模块	9
4.2.1 仿真波形	9
4.2.2 测试过程	9
4.3 取指管理模块	9
4.3.1 仿真波形	9
4.3.2 测试过程	9
4.4 运算管理模块	10
4.4.1 仿真波形	10
4.4.2 测试过程	10
4.5 存取管理模块	11
4.5.1 仿真波形	11
4.5.2 测试过程	11
4.6 回写管理模块	12
4.6.1 仿真波形	12
4.6.2 测试过程	12
4.7 I/O控制模块	12
4.7.1 仿真波形	12
4.7.2 测试过程	13
4.8 访存控制模块	13
4.8.1 仿真波形	13
4.8.2 测试过程	13
4.9 处理器整体	14
4.9.1 仿真波形	14
4.9.2 测试过程	15

5 问题及解决方法	15
5.1 接口定义、双向接口不熟悉	16
5.2 仿真时可实时跟踪的信息较少	16
5.3 设计时逻辑考虑不周全	16
5.4 bit文件由于提示电路太大无法生成	16
5.5 硬件测试时信号难以跟踪	16
5.6 回写及OUT指令输出出错	17
5.7 算术运算回写时下一个节拍出错	17
5.8 取指时指令不能维持输出	17
5.9 IF语句中多个上跳沿不能综合	18
5.10 RST绑定在开关上不能生成bit文件	18
5.11 访存读写时数据冲突	18
5.12 某些信号不能及时赋值	18
5.13 FPGA下载后内存无法修改	18
5.14 连续进行跳转指令时出现问题	19
6 未来工作	19

1 系统整体框图

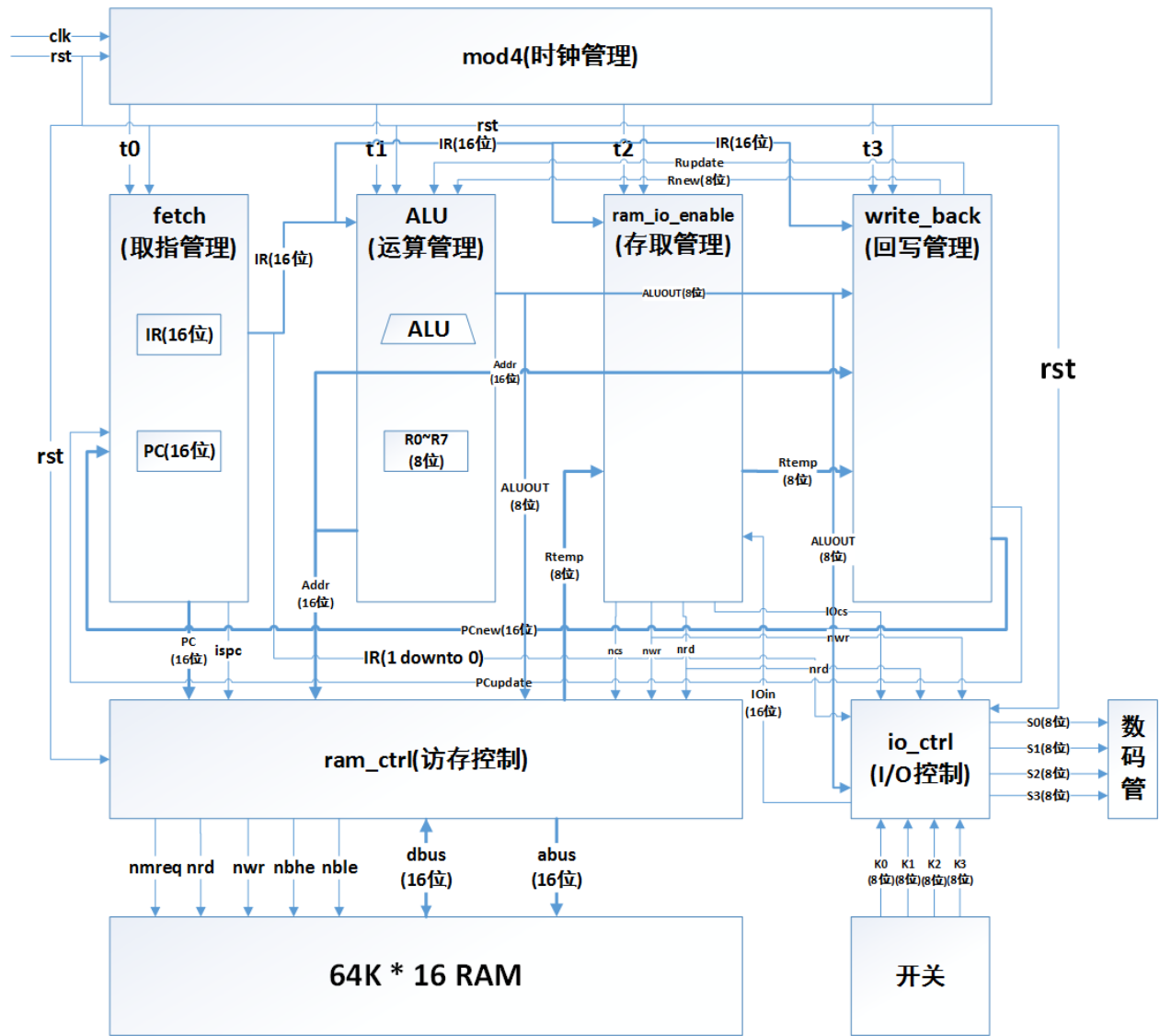


图 1: 系统设计框图

2 处理器接口说明

为仿真测试方便，仿真时增加了一部分输出端口，此部分输出端口单独列出。

表 1: 处理器接口说明

信号名	位数	方向	来源/去向	意义
clk	1	I	处理器板时钟按钮	系统时钟
rst	1	I	处理器板复位按钮	高电平复位
choice	2	I	拨动开关	选择输出至数码管的数据
abus	16	O	主存储器	地址总线
dbus	16	I/O	处理器/主存储器	数据总线
nmreq	1	O	主存储器	存储器片选
nrd	1	O	主存储器	存储器读
nwr	1	O	主存储器	存储器写
nbhe	1	O	主存储器	高字节访问允许
nble	1	O	主存储器	低字节访问允许
K0,K1,K2,K3	8	O	拨动开关	模拟I/O数据输入
output	48	O	数码管	choice选择输出不同数据

表 2: 仿真时所增加接口说明

信号名	位数	方向	意义
s3_o, s2_o, s1_o, s0_o	8	O	I/O端口数据
alureg_o	64	O	8个通用寄存器数据
ir_o	16	O	ir内容
pcoutside_o	16	O	pc内容

3 各模块说明

端口说明仅说明硬件测试时保留的接口，仿真波形中为了方便观察数据，增加了一些输出接口，没有实际的逻辑意义。增加的接口的内容在下一章仿真波形分析波形时会说明，这里不再赘述。

3.1 时钟管理模块

3.1.1 功能描述

1. 时钟管理模块接收系统时钟，依次产生t0,t1,t2,t3四个节拍脉冲，即指令周期中的

四个节拍，分别送入取指管理模块、运算管理模块、存取管理模块及回写管理模块，触发这四个模块进行相应的工作。

2. 复位信号为高电平时节拍计数器置0，且不产生脉冲信号。

3.1.2 接口说明

表 3: 时钟管理模块接口说明

信号名	位数	方向	来源/去向	意义
clk	1	I	处理器板时钟按钮	系统时钟
rst	1	I	处理器板复位按钮	高电平复位
t0	1	O	取指管理模块	取指周期节拍
t1	1	O	运算管理模块	运算周期节拍
t2	1	O	存取管理模块	存取周期节拍
t3	1	O	回写管理模块	回写周期节拍

3.2 取指管理模块

3.2.1 功能描述

1. 于第一个节拍到来时取出指令，并将指令送至各个模块，指令输出持续4个节拍，直到新的指令被取出，取代本条指令。
2. 于第一个节拍下降沿进行PC加1。
3. 当需要跳转时，由pcupdate高电平信号触发该模块，执行PC值的改变，跳转为相对跳转，由公式 $pc = pc + pcnew - 1$ 算出跳转后的PC值，待取指周期到来时进行取指，减1是由于跳转指令PC也在第一个节拍的下降沿执行了加1操作，故需扣去。

3.2.2 接口说明

表 4: 取指管理模块接口说明

信号名	位数	方向	来源/去向	意义
clk_all	1	I	处理器板时钟按钮	系统时钟
clk(t0)	1	I	时钟管理模块	取指周期节拍
rst	1	I	处理器板复位按钮	高电平复位
pcupdate	1	I	回写管理模块	PC更新触发信号
pcnew	16	I	回写管理模块	更新的PC内容(偏移量)
irin	16	I	访存控制模块	取出的指令
irout	16	O	运算、存取、回写及I/O控制模块	输出的指令
ispc	1	O	访存控制模块	取指信号
pcout	16	O	访存控制模块	所取指令地址

3.3 运算管理模块

3.3.1 功能描述

1. 访存指令、跳转指令在运算周期计算出访存地址或跳转地址的偏移量，通过addr输出，信号维持4个机器周期，不会与下一条指令相冲突，同时能保证回写模块pcnew得到正确的跳转地址偏移量。
2. 运算指令在运算周期计算出计算结果，通过aluout送出，信号维持4个周期，原因与上同。
3. 回写寄存器时，由rupdate高电平信号触发该模块，并通过此时ir的10到8位得到需要回写的寄存器的地址，将rdata的内容写入该寄存器。

3.3.2 接口说明

表 5: 运算管理模块接口说明

信号名	位数	方向	来源/去向	意义
clk(t1)	1	I	时钟管理模块	运算周期节拍
rst	1	I	处理器板复位按钮	高电平复位
ir	16	I	取指管理模块	当前指令
rupdate	1	I	回写管理模块	通用寄存器回写触发信号
rdata	8	I	回写管理模块	通用寄存器回写内容
aluout	8	O	访存控制、回写及I/O控制模块	运算结果
addr	16	O	访存控制模块	访存地址

3.4 存取管理模块

3.4.1 功能描述

1. 访存指令lda、sta，于存取周期置相应的使能信号，并关闭(低有效，置高电平)I/O使能信号，所有使能信号持续该周期便关闭，防止数据误读或误写。
2. I/O指令in、out，于存取周期置相应的使能信号，并关闭访存使能信号，所有使能信号持续该周期便关闭，防止数据误读或误写。

3.4.2 接口说明

表 6: 存取管理模块接口说明

信号名	位数	方向	来源/去向	意义
clk(t2)	1	I	时钟管理模块	存取周期节拍
rst	1	I	处理器板复位按钮	高电平复位
ir	16	I	取指管理模块	当前指令
rin	8	I	访存控制模块	从主存储器取出的8位数据
ioin	8	I	I/O控制模块	I/O输入数据
ncs	1	O	访存控制模块	存储器片选信号
ioncs	1	O	I/O控制模块	I/O控制模块使能信号
nrd	1	O	访存及I/O控制模块	通用读信号
nwr	1	O	访存及I/O控制模块	通用写信号
rout	8	O	回写管理模块	访存所取待回写数据

3.5 回写管理模块

3.5.1 功能描述

1. 回写通用寄存器：lda,in,add,sub,mov,mvi指令，将rupdate置高电平，同时将送入的rtemp从rdata端口送出；此时，运算管理模块接收到回写触发信号，去ir的10到8位解析回写寄存器地址，将数据写入相应寄存器。
2. 回写程序计数器：jmp及jz指令，jmp指令回写时，将pcupdate置高电平，同时将跳转偏移地址addr从pcnew端口送出；此时，取指模块接收到回写触发信号，改变pc寄存器的值。jz指令需要跳转时情况与jmp指令相同，不需要跳转时也进行回写，但此时addr为“0000000000000001”，pcnew输出“0000000000000001”，代入新pc计算公式时，效果相当于没有跳转，相对该条jz指令，取下一条指令时pc值比取jz指令的pc值多1。

3.5.2 接口说明

表 7: 回写管理模块接口说明

信号名	位数	方向	来源/去向	意义
clk(t3)	1	I	时钟管理模块	回写周期节拍
rst	1	I	处理器板复位按钮	高电平复位
ir	16	I	取指管理模块	当前指令
aluout	8	I	运算管理模块	运算结果
rtemp	8	I	存取管理模块	访存所取待回写数据
addr	16	I	运算管理模块	跳转地址偏移量
rupdate	1	O	运算管理模块	通用寄存器回写触发信号
pcupdate	1	O	取指管理模块	PC更新触发信号
rdata	8	O	运算管理模块	通用寄存器回写内容
pcnew	16	O	取指管理模块	PC的偏移量

3.6 I/O控制模块

3.6.1 功能描述

1. IN指令：I/O控制模块接收到存取管理模块的使能信号及读信号时，将端口地址为adin的I/O端口数据由rout口送出至存取管理模块，于回写周期将数据回写至指定寄存器中。地址adin由IR的末两位作为输入，rout上的I/O输入数据只持续该周期，防止误写。I/O输入用拨动开关模拟。

2. OUT指令：I/O控制模块接收到存取管理模块的使能信号及写信号时，将从aluout端口输入的数据，即相关寄存器的数据，输出到端口地址为adin的I/O端口上，输出数据至I/O用数码管显示模拟。

3.6.2 接口说明

表 8: I/O控制模块接口说明

信号名	位数	方向	来源/去向	意义
rst	1	I	处理器板复位按钮	高电平复位
npreq	1	I	存取管理模块	I/O模块使能信号
nprd	1	I	存取管理模块	读信号
npwr	1	I	存取管理模块	写信号
adin	2	I	取指管理模块	I/O端口地址
aluout	8	I	运算管理模块	输出I/O数据
K0,K1,K2,K3	8	I	拨动开关	模拟I/O输入
rout	8	O	存取管理模块	I/O输入待回写数据
S0,S1,S2,S3	8	O	数码管	模拟输出I/O的数据

3.7 访存控制模块

3.7.1 功能描述

1. 取指：取指周期取指管理模块发出取指信号，访存控制模块接收取指信号ispc高电平后置存储器使能信号及读信号为低电平(低电平有效)，送地址，经数据总线取出从输入端口pc输入的指令地址处的指令，由输出端口ir送至取指管理模块。有效电平均只持续一个周期。
2. LDA指令：取数时在运算周期生成addr，addr端口持续输入形式地址；于存取周期由存取管理模块给访存使能、读使能，访存控制模块置存储器使能信号及读信号为低电平，送地址，经数据总线取出地址addr的数据，该地址的低8位，由rtemp输出端口送至存取管理模块，于回写周期回写相应寄存器。
3. STA指令：存数时在运算周期生成addr，addr端口持续输入形式地址，aluout送入相应寄存器的数据；于存取周期由存取管理模块给访存使能、写使能，访存控制模块置存储器使能信号及写信号为低电平，送地址，通过数据总线将数据送入主存储器相应地址位置的低8位。

3.7.2 接口说明

表 9: 访存控制模块接口说明

信号名	位数	方向	来源/去向	意义
rst	1	I	处理器板复位按钮	高电平复位
ispc	1	I	取指管理模块	取指使能信号
ncs	1	I	存取管理模块	存储器片选信号
nrd	1	I	存取管理模块	读信号
nwr	1	I	存取管理模块	写信号
addr	16	I	运算管理模块	访存地址
aluout	8	I	运算管理模块	待存储数据
rtemp	8	O	存取管理模块	从存储器取出的数据
ir	16	O	取指管理模块	取出的指令
neq_o	1	O	主存储器	存储器使能信号
nrd_o	1	O	主存储器	读信号
nwr_o	1	O	主存储器	写信号
nbh	1	O	主存储器	高位字节访问允许
nbl	1	O	主存储器	低位字节访问允许

4 仿真波形

4.1 测试方案

1. 由于取指管理模块、运算管理模块、存取管理模块及回写管理模块在指令周期的四个节拍内扮演着不同的角色，为了尽量减少实验台硬件测试时逻辑设计上出现的问题，测试时指令为测试用例单位进行测试。具体步骤为先进行复位，然后模拟一条一条指令的时序进行测试，手动输入所需的必要信号，经整体波形及实验台验证，这种测试方案与实际情况相似度很高。访存控制模块同样以指令为测试用例单位进行测试。
2. 时钟管理模块：测试其能否正确输出各个节拍，复位功能是否正常。
3. I/O控制模块：由于其不是由节拍信号使能的，测试时只模拟I/O过程信号的时序输入情况。经硬件测试验证，同样减少了在实验台上大量时序逻辑、组合逻辑问题上的调试。
4. 处理器整体：同样以指令为测试用例单位进行测试，需要访存时，手动置数据总线的值。

4.2 时钟管理模块

4.2.1 仿真波形

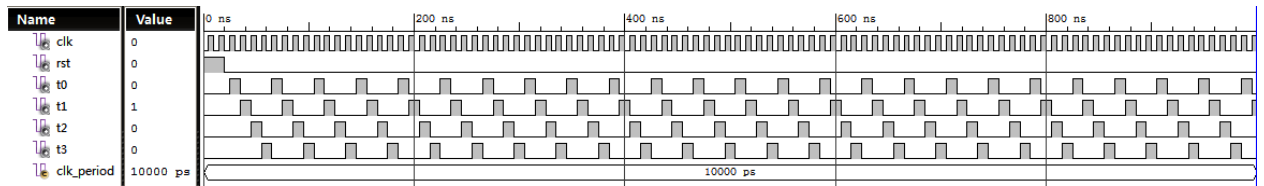


图 2: 时钟管理模块测试波形

4.2.2 测试过程

1. 先置复位信号为高电平，时钟管理模块复位，所有输出端均为低电平。
2. 置复位信号为低电平，时钟管理模块依次、循环输出t0、t1、t2及t3四个节拍高电平信号。

4.3 取指管理模块

4.3.1 仿真波形

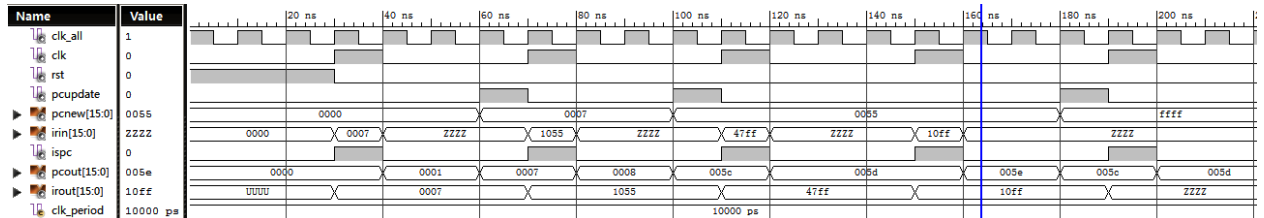


图 3: 取指管理模块测试波形

4.3.2 测试过程

依次测试无条件跳转、有条件跳转指令及其他指令所进行的取指及pc加1操作，由于本人设计cpu中有条件跳转指令均需回写且回写过程完全相同，且判断寄存器是否为0的逻辑与本模块无关，故无需同时测试是否跳转两种情况，波形中测试了正、负跳转的情况。

1. 复位信号为高电平时，该模块处于复位状态，不工作。
2. 取指周期ispc置高电平，送取指信号，所有测试指令均在取指周期取出并持续输出4个节拍。

3. 进行存数操作测试，波形可看出，在运算周期，相应寄存器的数据及存放数据的地址r7/ir(7 downto 0)分别送上aluout及addr，持续4个周期，可保证写数据时，地址与数据均为稳定状态。
4. 进行算术运算、mov、mvi等非访存指令测试，于运算周期，运算结果被送于aluout，addr置高阻，不妨存；回写周期该模块接收回写使能信号，将运算结果存入相应寄存器，寄存器地址由解析ir指令得到。
5. 跳转指令测试中，无条件跳转于运算周期生成指令地址偏移量，送addr；有条件跳转，于运算周期，若寄存器值为0，则计算地址偏移量送addr；若寄存器不为0，则送“0000000000000001”至addr，波形显示与预期一致。
6. 最后进行IN指令测试，IN指令于回写周期将I/O输入数据进行回写，将aah写入寄存器r6中。

4.5 存取管理模块

4.5.1 仿真波形

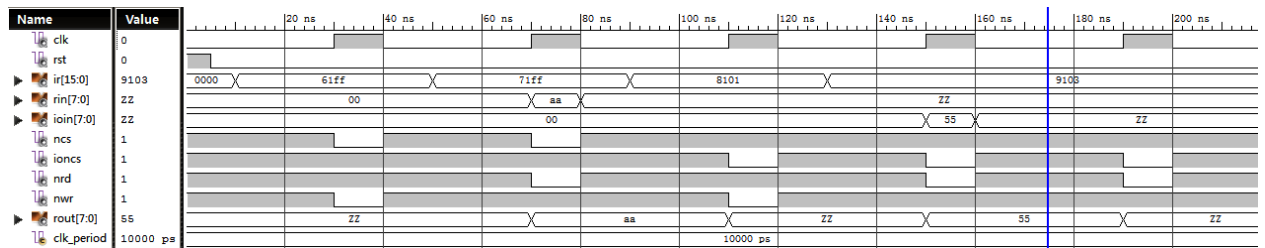


图 7: 存取管理模块测试波形

4.5.2 测试过程

1. 先置复位信号为高电平，该模块使能信号输出均为高电平。
2. 分别测试STA、LDA、OUT、IN指令，四条指令均在存取周期将相关使能信号置低电平(见波形使能信号凹陷处)，信号持续该周期；读指令(IN、LDA)在存储周期还将相应数据送至rout输出。

4.6 回写管理模块

4.6.1 仿真波形

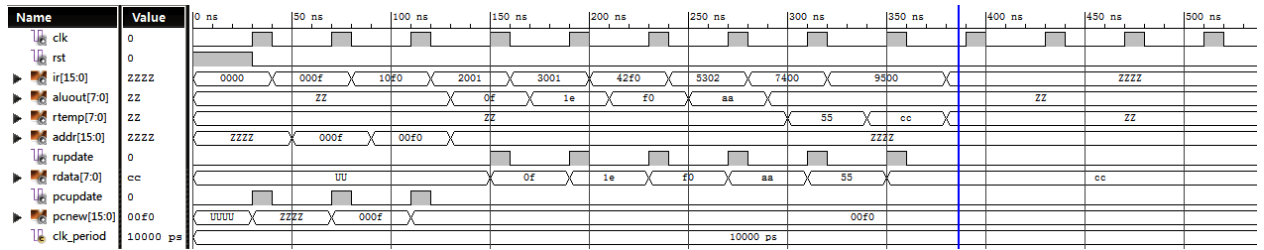


图 8: 回写管理模块测试波形

4.6.2 测试过程

1. 复位时，系统使能输出均为无效(低电平)，一些信号置高阻。
2. 跳转指令：于运算周期手动置addr；回写周期，该模块置pcupdate为高电平，同时将addr的值通过pcnew端口送出。rupdate为低电平(无效)。
3. 非访存指令：于运算周期手动置aluout；回写周期置rupdate为高电平，将aluout输入的数据从rdata输出，回写至相应寄存器。pcupdate为低电平(无效)。
4. 访存、I/O的输入指令：与存取周期手动置rtemp；回写周期置rupdate为高电平，将rtemp输入的数据从rdata输出，回写至相应寄存器。pcupdate为低电平(无效)。

4.7 I/O控制模块

4.7.1 仿真波形

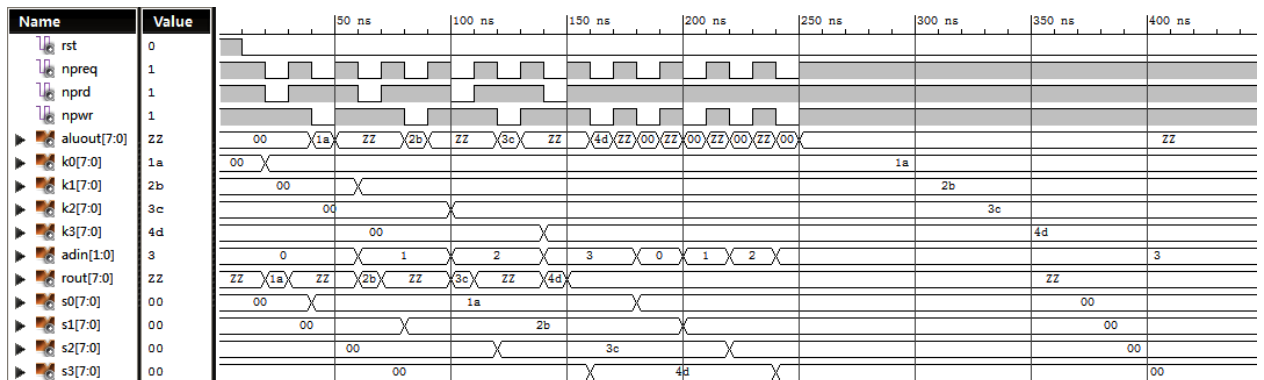


图 9: I/O控制模块测试波形

4.7.2 测试过程

1. 复位信号置1(有效)时，该模块处于不工作状态。
2. IN指令：手动置使能信号，端口地址为adin的端口将数据送至rout，数据保持一个周期，K0 K3模拟数据输入。
3. OUT指令：手动置使能信号，aluout输入的数据被送至端口地址为adin的端口，用S0 S3模拟，数据在改变之前始终保持，确保数码管常亮。

4.8 访存控制模块

4.8.1 仿真波形

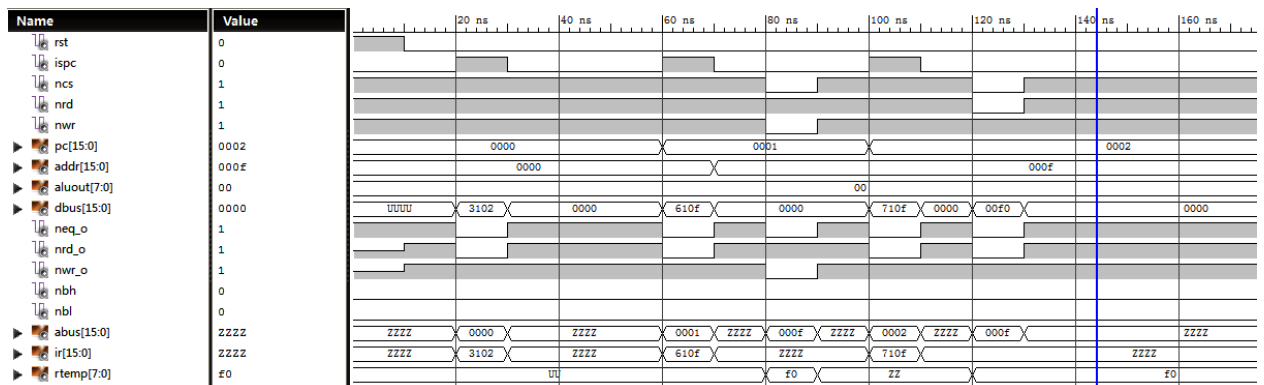


图 10: 访存控制模块测试波形

4.8.2 测试过程

1. 复位信号有效时，存储器片选信号失效，不访问存储器。
2. 无论执行何种指令，均在取指周期进行取指。取指模块给ispc高电平，本模块置存储器片选有效，给读使能，abus为pc值，dbus的数据送ir口输出。离开取指周期，片选失效。
3. LDA、STA指令，在存取周期，根据接收的从存取管理模块送来的使能信号，置相应的使能信号，abus为addr的值，仅持续访存周期，STA指令dbus低八位为aluout的数据，LDA指令dbus将数据低八位送rtemp输出。

4.9 处理器整体

4.9.1 仿真波形

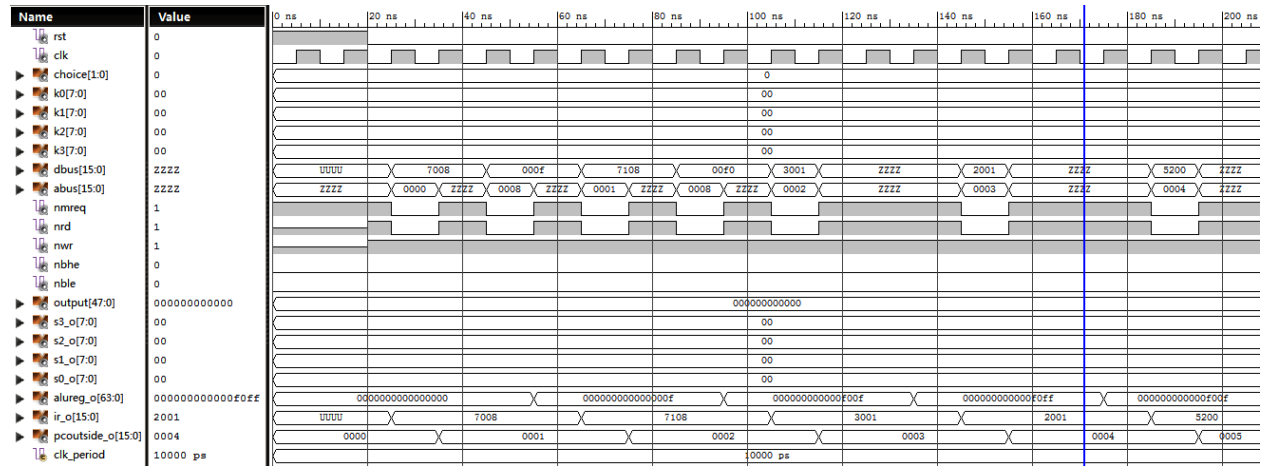


图 11: 处理器整体测试波形(1)

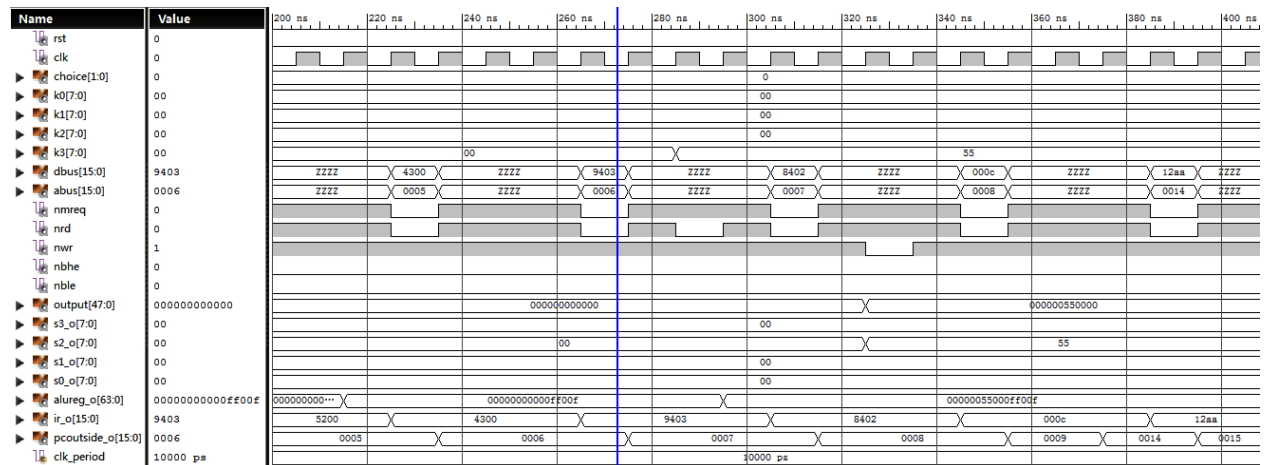


图 12: 处理器整体测试波形(2)

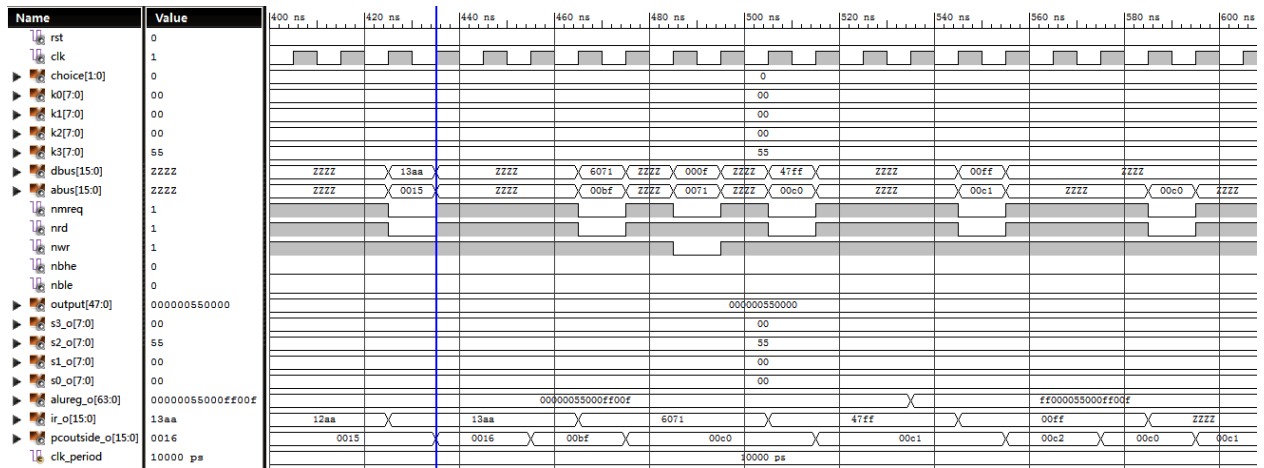


图 13: 处理器整体测试波形(3)

4.9.2 测试过程

output与choice为实验台上显示用，波形图中用增加的输出信号替代这样的显示方式。

1. 复位信号有效时，CPU处于不工作状态。
2. LDA指令：取指、回写功能正常。abus、dbus均在正确的节拍赋值。alureg_o中可以看出，寄存器r0、r1的值也相应地发生改变。
3. 非访存指令：取指、回写功能正常。寄存器的值相应地发生改变。
4. IN、OUT指令：取指功能正常，IN指令进行数据输入，并回写至相应寄存器，寄存器值发生改变；out指令输出数据，可以看到S2被为55，输出功能正常。存取周期I/O使能端的赋值均正确，有效电平均只持续一个周期。
5. 跳转指令：取指功能正常，回写时，PC与跳转指令取指时的改变量为跳转地址偏移量，说明功能正确。
6. STA指令：取指功能正常，存取周期，给出片选、读信号为低电平，持续该周期，abus、dbus赋值正确。
7. 负跳转指令：先置r7为FFh，负跳转时，PC值改变正确。

5 问题及解决方法

以下问题大部分为自己进行实验台测试时所遇到并解决的问题，后面的为数不多的举例是在帮助他人解决的问题且觉得有借鉴意义。

5.1 接口定义、双向接口不熟悉

解决方案：后来发现总体的架构还是没有问题的，接口上构思不熟练，在输入vhdl代码的时候进行微调。感觉随着开发次数的增加，开发经验的累积，在架构上对接口的设计会越来越成熟，有利于底层vhdl代码输入的实现。

5.2 仿真时可实时跟踪的信息较少

解决方案：

1. 仿真时临时增加输出端口，输出必要的信息，即几乎每次仿真都需要观察但是没有作为接口的数据。
2. 将偶尔需要观察的信号，拖入仿真波形一栏，进行Relaunch，由于Relaunch需要运行两次，且第二次往往较久，故时间效率上不如第一种高效。
3. 对某些较难分析的小时间片段有耐心地进行单步调试，了解到更多的细节，发现问题所在。

5.3 设计时逻辑考虑不周全

解决方案：书写test bench的时候，应尽量测试自己所预计到的软肋，并尽力模仿实验台上各个信号给出的顺序写test bench。这样能使仿真最真实地模拟出硬件测试的情况，刁钻的测试用例可以找出细小的bug，事实证明，仿真过程中能解决的问题，尽量在仿真处解决，硬件测试时再解决容易使效率降低，并且摸不着头脑。

5.4 bit文件由于提示电路太大无法生成

原因分析：查看了芯片的资源使用情况，原因在于CPU绑定的管脚过多，且即使有的接口没有绑定，只要声明的对外接口大于实验台的最大接口，xilinx便不能生成.bit文件。

解决方案：在生成.bit文件前，先将没有绑定的接口注释掉，这类接口设计是为仿真观察用，使得对外总接口数少于实验台上限，问题解决。

5.5 硬件测试时信号难以跟踪

解决方案：设计组合逻辑电路，通过开关的选择，将不同的信息输出到数码管上显示，最后确定输出的内容为当前节拍、PC、IR、ABUS、DBUS、I/O输出口。存储器的内容及访问存储器的使能端赋值情况可以从实验台测试使用的软件看到。信息数的增加大大加快了调试的速度，更容易发现电路内部的问题。

5.6 回写及OUT指令输出出错

问题描述：回写寄存器时，及OUT指令输出时，容易在附近一个节拍，将数据写到其他寄存器上。

原因分析：经分析，是由于同时送地址、数据，容易产生险象，即虽然仿真没有出现问题，但是真实的硬件电路并不能保证地址改变的瞬间地址是多少，故会在附近一个节拍误写入其他寄存器。

解决方案：写数据时需要平稳的地址输入，故将模块之间传递地址改为地址直接取自IR进行解析，由于IR指令持续4个周期，故地址输入平稳，问题得到解决。但是进一步分析，这样的改进，在不增加存放指令暂存器的情况下不利于指令流水的设计，故在将来的学习中应该思考并实现解决险象问题的其他方案，如：增加判断逻辑，待数据平稳之后再进行输入。

5.7 算术运算回写时下一个节拍出错

问题描述：算术运算回写时，结果正确，但在下一个节拍，结果被一个不定的值改变，出现不能预计的偏差。

原因分析：回写模块rdata只在回写周期输出，之后便置高阻，再次出现数据竞争的问题，虽然仿真没有问题，但在实验台上便可能出错。

解决方案：将相关的置高阻的语句删去后，即输出在没有改变的时候保持住，置高阻是出于误写的考虑。但在分析之后，这样的改变不会出现误写的情况。故删去置高阻相关语句之后，此问题解决。

5.8 取指时指令不能维持输出

问题描述：仿真时没有出现问题，指令在整个指令周期内持续输出；而实验台测试中，发现指令在下个节拍边改变为FFFFh。

原因分析：询问老师之后，一部分vhdl代码不能确定是否能在综合的时候用硬件实现。

解决过程：故删去设计中的两处该部分代码，用暂存器替代原来的逻辑，I/O控制处后来出现了OUT指令输出错，在上面已提到解决方案。

取指模块有出现了第二节拍暂存器仍然会被误写数据的情况，此时不再是FFFFh，而是不能确定的数。故引入全局时钟信号，在该节拍未结束时便停止输入数据。

然而，出现了IR再非取指节拍也改变的现象。原因是加入时钟信号没有完善相关逻辑，导致每次上升沿都会进行新的IR写入。

引入了相关判读逻辑，但此时IR的值一直不改变。

原因是信号的赋值有延迟，不能马上进入想要执行的语句块，改为用变量进行判断后，问题解决。

5.9 IF语句中多个上跳沿不能综合

原因及解决方案：由于没有相关电路可以综合，且逻辑分析起来也会有问题，故输入vhdl代码时，将跳沿语句放于IF语句的最后一句。且若能用敏感信号表+电平代替跳沿，且不影响相关功能，就进行改变。

5.10 RST绑定在开关上不能生成bit文件

原因及解决方案：翻阅了老师的PPT，及上网查阅了相关资料，猜测是由于RST绑定了太多的管脚，被误认为是时钟信号，加入的xilinx提示所加的一句话之后，不能解决问题，最后将RST的管脚绑定在了INT按钮上，问题解决。

5.11 访存读写时数据冲突

原因分析：由于代码是自己思考所写，没有参考类似代码，在IF语言合适的地方置高阻，但总有考虑不周到的地方。

解决方案：询问助教，学习了可以在IF语句之前统一置高阻，解决了思考良久但总有疏漏的较不好的代码风格。

5.12 某些信号不能及时赋值

此后3个问题为帮助他人解决的问题，且认为有借鉴意义，故进行记录。

原因分析：IF语句中输入信号判断可能会有延迟。

解决方案：

1. 将信号判断改为变量判断。
2. 不影响整体功能的情况下，可以将连线逻辑写在process外面，即组合逻辑实现，这样信号一改变便会输出，不必担心有潜在的寄存器使数据输出慢了一个节拍。

5.13 FPGA下载后内存无法修改

问题描述：FPGA下载后内存无法修改，且有些地方恒为0。

原因分析：访存控制模块里使能端在不访存时没有及时置无效。

解决方案：调整思路，重新设计了访存控制模块，问题解决。

5.14 连续进行跳转指令时出现问题

原因分析：回写pc时延迟了一个节拍，故其实是连续取了两次jmp指令，再跳转，一条跳转指令运行时最后效果上不会出现问题。但是IR、PC显示有明显的问题。多条跳转指令组合，问题边暴露得更加明显。

解决方案：将PC回写及加1的逻辑单独写在一个process里面，保证PC及时改变，问题解决。

6 未来工作

在改进地址数据同时传输的时候，对地址采用了直接解析模块内的IR的方式，在不增加指令暂寄存器的情况下不利于指令流水的设计，故在进一步改进该处理器，设计流水线时出现了困难。之后，考虑问题的重点在于如何通过另外一种方式解决此类问题，如：增加判断逻辑，待数据平稳之后再进行输入。进而进行指令流水的设计。