

计算机组成与嵌入式系统

--CPU设计实验

老师：徐文辉

QQ: 4127164

电话: 18202799815

Email: xuwenhui@hust.edu.cn

实验环境

- Logisim仿真软件

实验内容

- 多周期MIPS
 - 微程序
 - 硬布线



多周期MIPS CPU实验

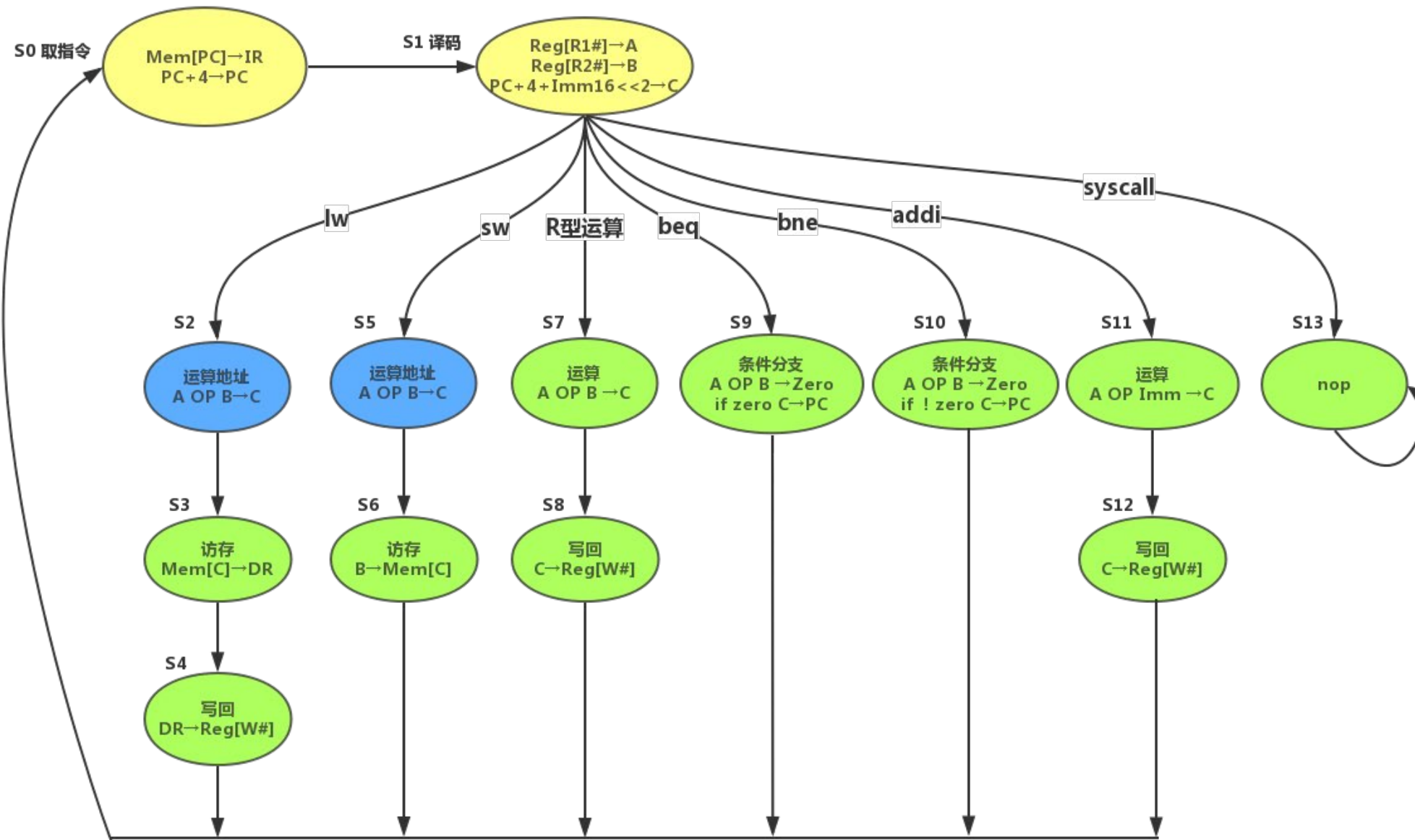
• 实验目的

- 理解多周期MIPS CPU的基本原理
 - 能在Logisim平台中设计实现多周期MIPS CPU
 - 硬布线控制器，微程序控制器两种方案
 - 8条核心指令，能运行冒泡排序测试程序

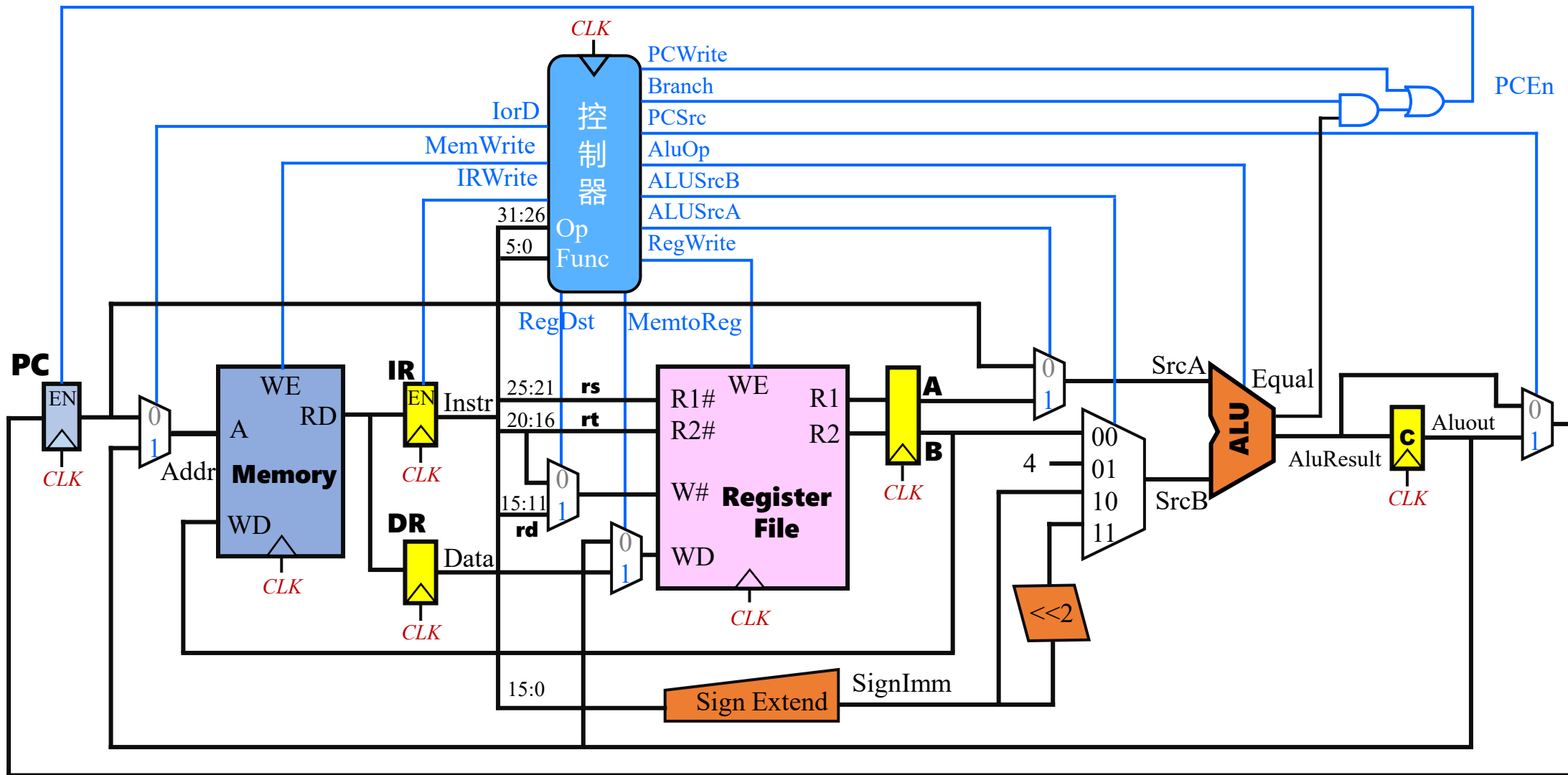
多周期MIPS CPU实验 — 8条核心指令

#	MIPS指令	RTL功能描述
1	<code>add rd,rs,rt</code>	$R[rd] \leftarrow R[rs] + R[rt]$ 不考虑溢出
2	<code>slt rd,rs,rt</code>	$R[rd] \leftarrow R[rs] < R[rt]$ 小于置1, 有符号比较
3	<code>addi rt,rs,imm</code>	$R[rt] \leftarrow R[rs] + \text{SignExt}_{16b}(\text{imm})$ 不考虑溢出
4	<code>lw rt,imm(rs)</code>	$R[rt] \leftarrow \text{Mem}_{4B}(R[rs] + \text{SignExt}_{16b}(\text{imm}))$
5	<code>sw rt,imm(rs)</code>	$\text{Mem}_{4B}(R[rs] + \text{SignExt}_{16b}(\text{imm})) \leftarrow R[rt]$
6	<code>beq rs,rt,imm</code>	$\text{if}(R[rs] = R[rt]) \text{ PC} \leftarrow \text{PC} + \text{SignExt}_{18b}(\{\text{imm}, 00\})$
7	<code>bne rs,rt,imm</code>	$\text{if}(R[rs] \neq R[rt]) \text{ PC} \leftarrow \text{PC} + \text{SignExt}_{18b}(\{\text{imm}, 00\})$
8	<code>syscall</code>	系统调用, 这里用于停机

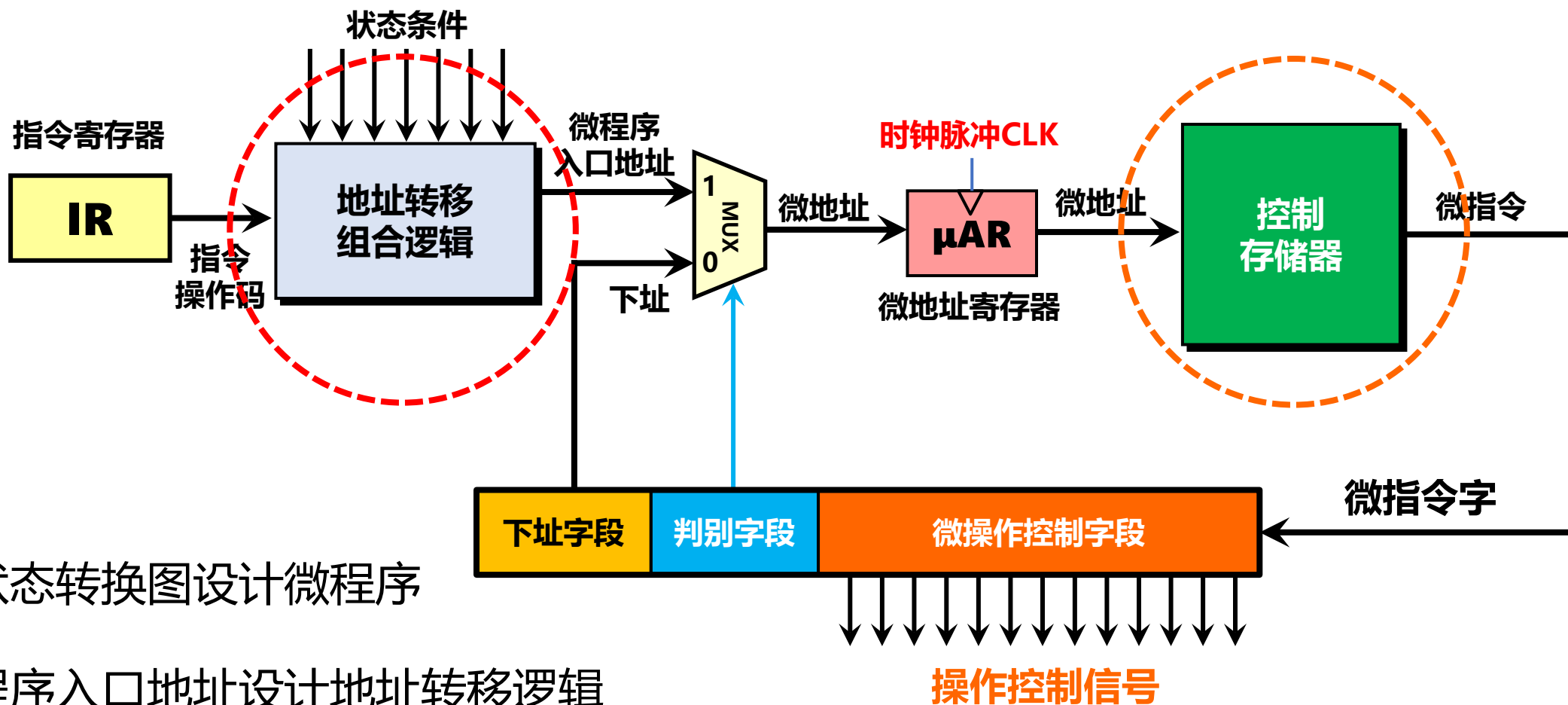
多周期MIPS CPU实验 — 指令周期状态转移图



多周期MIPS CPU实验 — 顶层视图

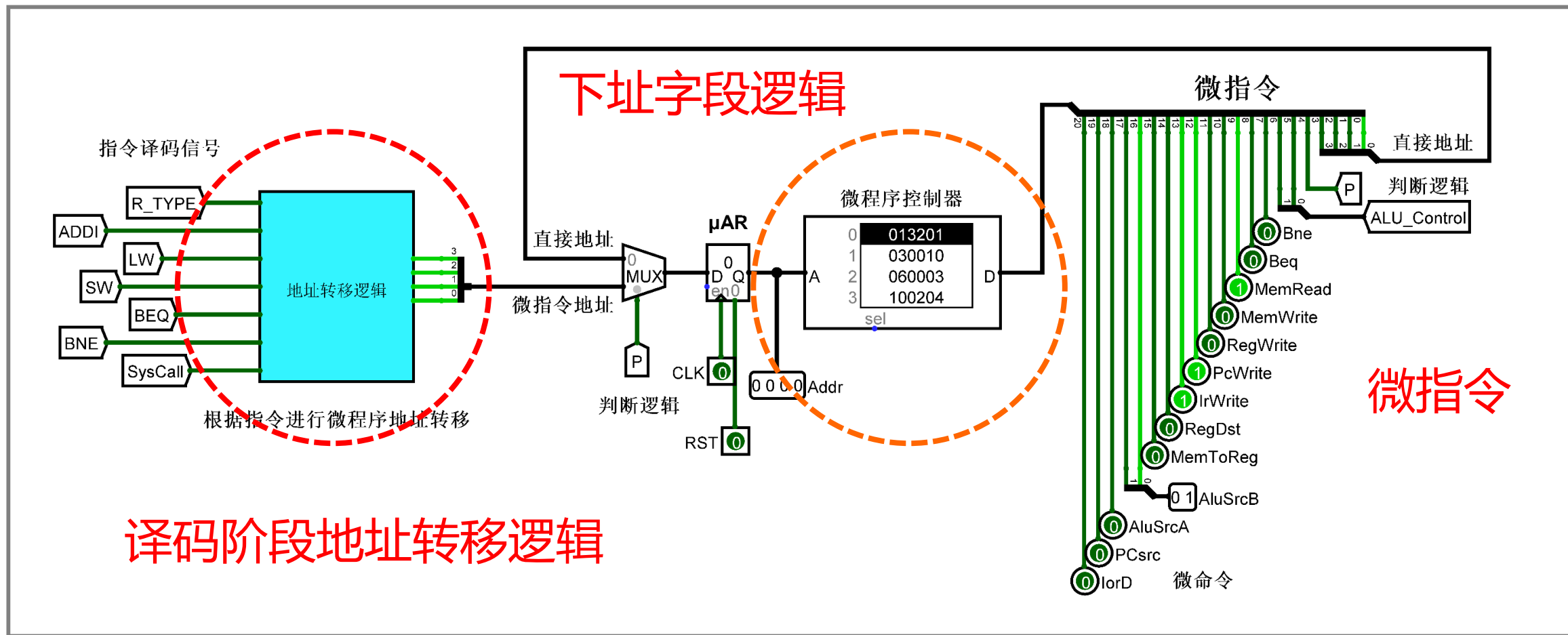


多周期MIPS CPU实验 — 构建微程序控制器

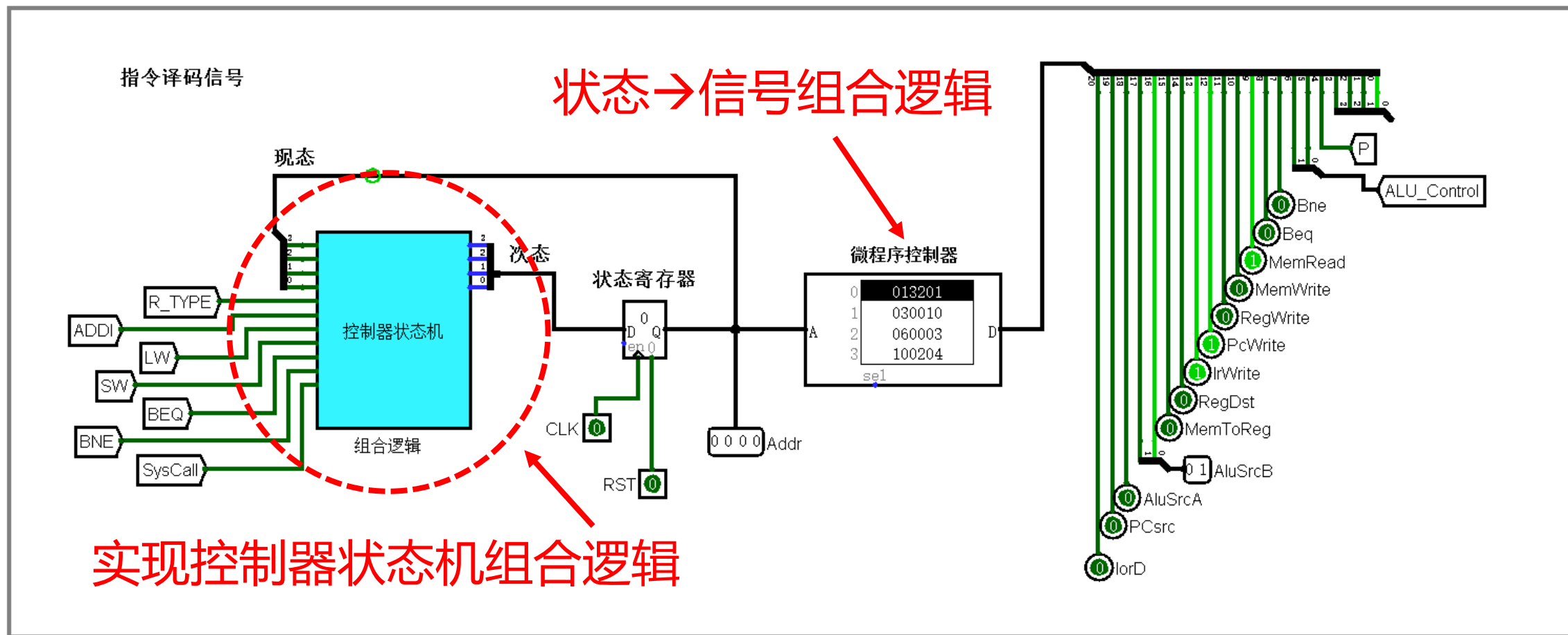


- 根据状态转换图设计微程序
- 按微程序入口地址设计地址转移逻辑
- 构造微程序控制器

多周期MIPS CPU实验 — 微程序控制器Logisim实现



多周期MIPS CPU实验 — 硬布线控制器Logisim实现



多周期MIPS CPU实验 —— 微程序控制器微指令

微指令自动生成(2019-4-22) [受保护的视图] - Excel																					
文件 开始 插入 页面布局 公式 数据 审阅 视图 帮助 Acrobat 操作说明搜索																					
受保护的视图 请注意 - 来自 Internet 的文件可能包含病毒。除非需要编辑，否则保持在受保护视图中比较安全。 启用编辑(E)																					
N10 0																					
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	V
1																					
2	微指令功能	状态	微指令地址	lorD	PcSrc	AluSrcA	AluSrcB	MemToReg	RegDst	IrWrite	PcWrite	RegWrite	MemWrite	MemRead	BEQ	BNE	AluControl	P	下址字段	微指令	十六进制
3	取指令	0	0000	0	0	0	01	0	0	1	1	0	0	1	0	0	00	0	0001	0000100110010000000001	13201
4	译码	1	0001	0	0	0	00	0	0	0	0	0	0	0	0	0	00	0	0000	0000000000000000000000	0
5	LW1	2	0010	00	0	0	00	0	0	0	0	0	0	0	0	0	00	0	0000	0000000000000000000000	0
6	LW2	3	0011	0	0	0	00	0	0	0	0	0	0	0	0	0	00	0	0000	0000000000000000000000	0
7	LW3	4	0100	0	0	0	00	0	0	0	0	0	0	0	0	0	00	0	0000	0000000000000000000000	0
8	SW1	5	0101	0	0	0	00	0	0	0	0	0	0	0	0	0	00	0	0000	0000000000000000000000	0
9	SW2	6	0110	0	0	0	00	0	0	0	0	0	0	0	0	0	00	0	0000	0000000000000000000000	0
10		7	0111	0	0	0	00	0	0	0	0	0	0	0	0	0	00	0	0000	0000000000000000000000	0
11		8	1000	0	0	0	00	0	0	0	0	0	0	0	0	0	00	0	0000	0000000000000000000000	0
12		9	1001	0	0	0	00	0	0	0	0	0	0	0	0	0	00	0	0000	0000000000000000000000	0
13		10	1010	0	0	0	00	0	0	0	0	0	0	0	0	0	00	0	0000	0000000000000000000000	0
14		11	1011	0	0	0	00	0	0	0	0	0	0	0	0	0	00	0	0000	0000000000000000000000	0
15		12	1100	0	0	0	00	0	0	0	0	0	0	0	0	0	00	0	0000	0000000000000000000000	0
16		13	1101	1	0	0	00	0	0	0	0	0	0	0	0	0	11	0	0000	10000000000001100000	100060
17																					
18	第1步： 在第1列安排微程序，通常取指令部分放置在0号单元，同一指令的微程序中的微指令顺序存放																				
19	第2步： 填写D到S列的微指令控制信号，注意其中aluSrcB，AluControl为2位，下地址字段4位																				
20	第3步： 完成第2步后，最后一列微指令16进制会自动更新																				
21	第3步： 将最后一列的16进制编码复制粘贴到Logisim中的控制存储器中																				

微指令十六进制编码直接复制粘贴到控存中

多周期MIPS CPU实验 — 微程序控制器微指令

Logisim 2.15.0.2.exe: 多周期微程序控制器 of cpu

File Edit Project Simulate Window Help



cpu

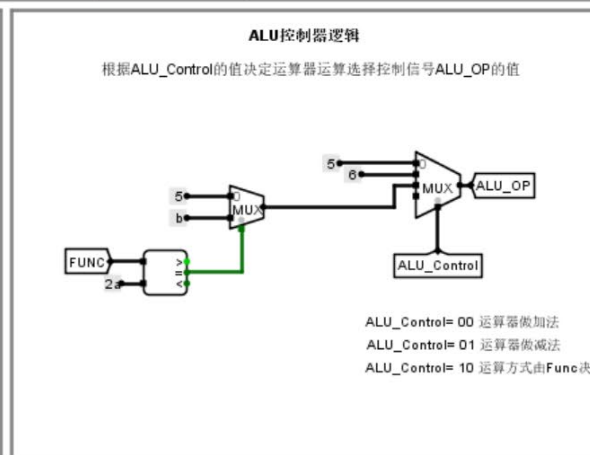
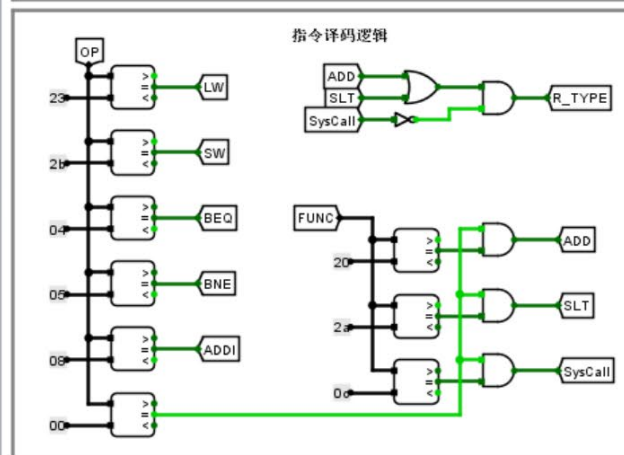
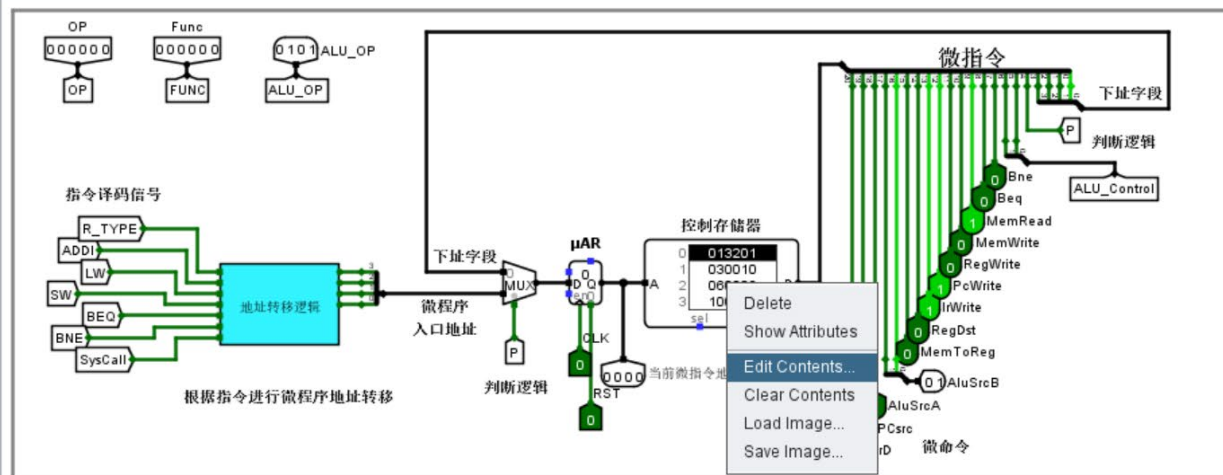
- ◇ 单周期MIPS(硬布线)
- ◇ 单周期硬布线控制器
- ◇ 多周期MIPS(微程序)
- ◇ 多周期微程序控制器
- ◇ 微程序地址转移逻辑
- ◇ 多周期MIPS(硬布线)
- ◇ 多周期硬布线控制器
- ◇ 状态机FSM
- ◆ MIPS Regfile
- ◆ MIPS ALU

- Wiring
- Gates
- Plexers
- Arithmetic
- Memory
- Input/Output
- Base
- CS3410 Components
- HUST2020

Circuit: 多周期微程序控制器

Circuit Name ◇ 多周期微程序...
Shared Label 微程序控制器
Shared Label F... East
Shared Label F... SansSerif Bold ...
Label Color #000000

Zoom: 81%



给出简单的逻辑实现对应指令译码信号, LW、SW、BEQ、BNE、ADDI、ADD、SLT、SYSCALL、R_TYPE。

注意_R_TYPE表示R型运算指令, SYSCALL是特殊的R型指令, 不属于这个类别

多周期MIPS CPU实验 —— 微程序地址转移

微程序地址转移逻辑自动生成ext(2020-9-1) [受保护的视图] - Excel

登录

文件

开始

插入

页面布局

公式

数据

审阅

视图

帮助

Acrobat

操作说明搜索

共享

J37

✕

✓

\sum

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

1

机器指令译码信号

微程序入口地址

2

R_Type

ADDI

LW

SW

BEQ

BNE

SYSCALL

xxx1

xxx2

xxx3

入口地址
10进制

S4

S3

S2

S1

S0

3

1

7

0

0

1

1

1

4

1

11

0

1

0

1

1

5

1

9

0

1

0

0

1

6

1

0

0

0

0

7

1

8

1

</

多周期MIPS CPU实验 —— 微程序地址转移

微程序地址转移逻辑自动生成ext(2020-9-1) [受保护的视图] - Excel																	
文件 开始 插入 页面布局 公式 数据 审阅 视图 帮助 Acrobat 操作说明搜索																	
R41																	
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	R_Type	ADDI	LW	SW	BEQ	BNE	SYSCALL	xxx1	xxx2	xxx3	最小项表达式	S4	S3	S2	S1	S0	
2	R_Type&										R_Type			R_Type+	R_Type+	R_Type+	
3		ADDI&									ADDI		ADDI+		ADDI+	ADDI+	
4			LW&								LW		LW+			LW+	
5				SW&							SW						
6					BEQ&						BEQ						
7						BNE&					BNE						
8							SYSCALL&				SYSCALL	SYSCALL+					
9																	
10																	
11																	
12																	
31												SYSCALL	ADDI+LW	R_Type	R_Type+ADDI	R_Type+ADDI+LW	
33																	
34																	
35																	
36																	
37																	
38																	
39																	
40																	
41																	
42																	
43																	
44																	
45																	
46																	
47																	
48																	
49																	

此表一般不需要修改，如需修改，在审阅中撤销工作表保护

这是最终的表达式，复制到Logisim中即可

多周期MIPS CPU实验 — 微程序地址转移

Logisim 2.15.0.2.exe: 微程序地址转移逻辑 of cpu

File Edit Project Simulate Window Help

Project menu options:

- Add Circuit...
- Load Library
- Unload Libraries...
- Move Circuit Up
- Move Circuit Down
- Set As Main Circuit
- Remove Circuit
- Revert To Default Appearance
- View Toolbox
- View Simulation Tree
- Edit Circuit Layout
- Edit Circuit Appearance
- Analyze Circuit
- Get Circuit Statistics
- Options...

Toolbox:

- cpu
- W
- G
- Plexers
- Arithmetic
- Memory
- Input/Output
- Base
- CS3410 Components
- HUST2020

Circuit: 微程序地址转移逻辑

Circuit Name: 微程序地址转...

Shared Label: 地址转移逻辑

Shared Label F...: North

Shared Label F...: SansSerif Plain...

Label Color: #000000

Zoom: 81 %

Combinational Analysis

File Edit Project Simulate Window Help

Inputs Outputs Table Expression Minimized

Output: S3

SYSCALL + BN S3 + ADDI

SYSCALL + BNE + BEQ + ADDI

S3

S2

S1

S0

Clear Revert Enter

Build Circuit

多周期MIPS CPU实验 — 硬布线控制器状态转换

硬布线控制器状态转换逻辑自动生成(2019-10-29) [受保护的视图] - Excel

文件 开始 插入 页面布局 公式 数据 审阅 视图 帮助 Acrobat 操作说明搜索

AC35

	A	B	C	D	E	F	G	H	I	J	K	L	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC
1	当前状态(现态)					输入信号							下一状态 (次态)															
2	S3	S2	S1	S0	现态 10进制	R_Type	LW	SW	BEQ	BNE	SYSCALL	ADDI	次态 10进制	N3	N2	N1	N0											
3	0	0	0	0	0								1	0	0	0	1											
4	0	0	0	1	1	1							7	0	1	1	1											
5	0	0	0	1	1		1						2	0	0	1	0											
6																												
7																												
8																												
9																												
10																												
11																												
12																												
13																												
14																												
15																												
16																												
17																												
18																												
19																												
20																												
21																												
22																												
23																												
24																												
25																												
26																												
27																												
28																												
29																												
30																												
31																												
32	只填0或1, 无关项不填																											
33																												
34																												
35																												
36																												
37																												

状态转换表 表达式自动生成

多周期MIPS CPU实验 — 硬布线控制器状态转换

硬布线控制器状态转换逻辑自动生成(2019-10-29) [受保护的视图] - Excel

文件 开始 插入 页面布局 公式 数据 审阅 视图 帮助 Acrobat 操作说明搜索

N18 =IF(状态转换表!O19=1,\$M18&"+", "")

	A	B	C	D	E	F	G	H	I	J	K	M	N	O	P	Q	R	S	T	
1	S3	S2	S1	S0	R_Type	LW	SW	BEQ	BNE	SYS CALL	ADDI	最小项表达式	N3	N2	N1	N0				
2	~S3&	~S2&	~S1&	~S0&								~S3&~S2&~S1&~S0				~S3&~S2&~S1&~S0+				
3	~S3&	~S2&	~S1&	S0&	R_Type&							~S3&~S2&~S1&S0&R_Type		~S3&~S2&~S1&S0&R_Type+	~S3&~S2&~S1&S0&R_Type+	~S3&~S2&~S1&S0&R_Type+				
4	~S3&	~S2&	~S1&	S0&		LW&						~S3&~S2&~S1&S0&LW			~S3&~S2&~S1&S0&LW+					
5																				
6																				
7																				
8																				
9																				
10																				
11																				
12																				
13																				
14																				
15																				
16																				
17																				
18																				
19																				
20																				
31	逻辑表达式->>>												~S3&~S2&~S1&S0&R_Type	~S3&~S2&~S1&S0&R_Type	~S3&~S2&~S1&S0&R_Type	~S3&~S2&~S1&S0&R_Type				
33																				
34																				
35																				
36	此表一般不需要修改，如需修改，在审阅中撤销工作表保护												这里是最终的表达式，复制到Logisim中即可							
37																				
38																				
39																				
40																				
41																				

状态转换表 表达式自动生成

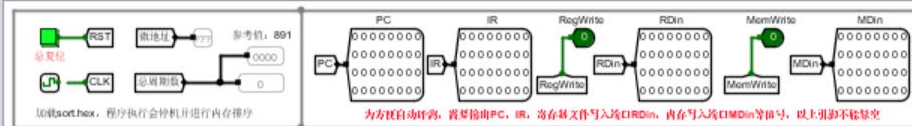
多周期MIPS CPU实验 — 测试

Logisim 2.15.0.2.exe: 多周期MIPS(微程序) of cpu

File Edit Project Simulate Window Help



- cpu
- ◇ 单周期MIPS(硬布线)
- ◇ 单周期硬布线控制器



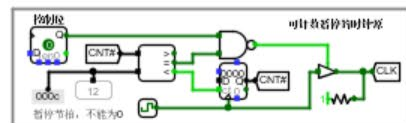
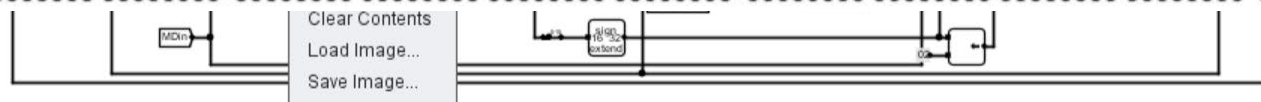
Logisim: Hex Editor

File Edit Project Simulate Window Help

```
000 2010ffff 20110000 ae300200 22100001 22310004 ae300200 22100001 22310004 ae300200 22100001 22310004 ae300200 22100001 22310004 ae300200 22100001
010 22310004 ae300200 22100001 22310004 ae300200 22100001 22310004 ae300200 00008020 2011001c 8e130200 8e340200 0274402a 11000002 ae330200 ae140200
020 2231fffc 1611ff8 22100004 2011001c 1611ff5 2002000a 0000000c 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
030 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
040 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
050 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
060 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
070 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
080 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
090 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0a0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
```

Circuit: 多周期MIPS(微程序)

Circuit Name ◇ 多周期MIPS(...)
Shared Label
Shared Label F... East
Shared Label F... SansSerif Plain...
Label Color #000000



数字逻辑1进入计数时钟模式，cnt+clk时钟由运行时钟节拍暂停
cnt+clk关闭时钟继续运行，时钟制位暂停，进入单步调试模式测试
用这个时钟源需要去掉电路中的其他时钟源

多周期MIPS CPU实验 — 测试

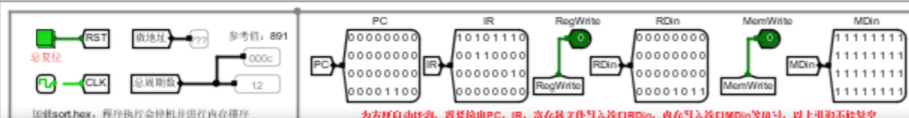
Logisim 2.15.0.2.exe: 多周期MIPS(微程序) of cpu

File Edit Project Simulate Window Help

Hand cursor, A, I, Run, Stop, Step, Undo, Redo

+ - - - - -

- cpu
 - ◇ 单周期MIPS(硬布线)
 - ◇ 单周期硬布线控制器
 - ◇ 多周期MIPS(微程序)
 - ◇ 多周期微程序控制器
 - ◇ 微程序地址转移逻辑
 - ◇ 多周期MIPS(硬布线)
 - ◇ 多周期硬布线控制器
 - ◇ 挂本例FSM



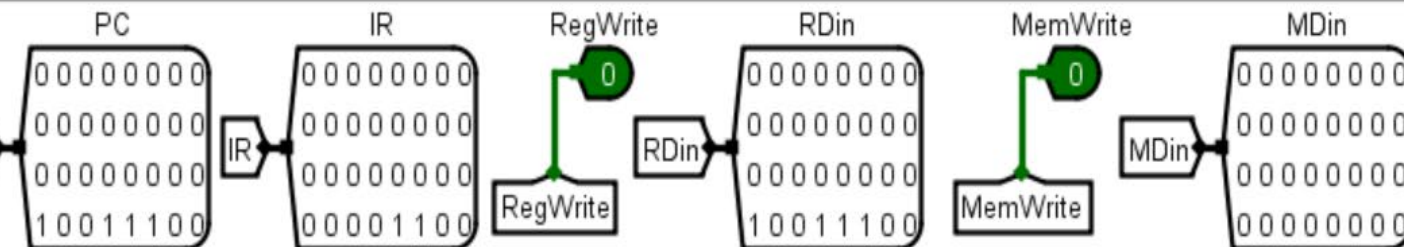
Logisim: Hex Editor

File Edit Project Simulate Window Help

```
000 2010fff 20110000 ae300200 22100001 22310004 ae300200 22100001 22310004 ae300200 22100001 22310004 ae300200 22100001 22310004 ae300200 22100001
010 22310004 ae300200 22100001 22310004 ae300200 22100001 22310004 ae300200 00008020 2011001c 8e130200 8e340200 0274402a 11000002 ae330200 ae140200
020 2231fff 1611fff8 22100004 2011001c 1611fff5 2002000a 0000000c 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
030 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
```



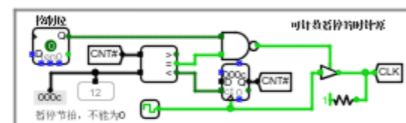
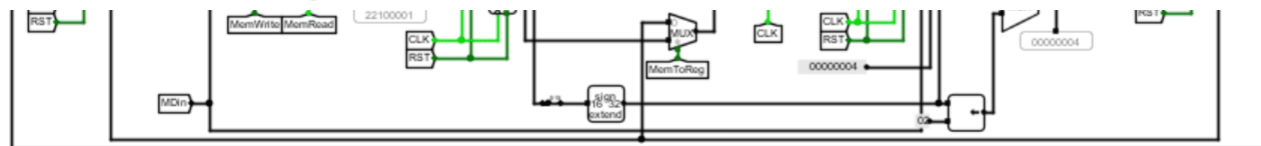
加载sort.hex, 程序执行会停机并进行内存排序



为方便自动评测, 需要输出PC, IR, 寄存器文件写入端口RDin, 内存写入端口MDin等信号, 以上引脚不能悬空

Circuit: ◇ 多周期MIPS(微程序)

Circuit Name ◇ 多周期MIPS(...)
Shared Label
Shared Label F... East
Shared Label F... SansSerif Plain...
Label Color #000000



载工具置1进入计数暂停模式, cnt+时钟自动运行到对节拍暂停
cnt+时钟连续运行, 每控制位置零, 进入单步测试模式测试
用该时钟暂停置零去停电路中其他时钟

请同学们开始实验！