计算机科学与技术专业

计算机组成

MIPS单周期控制

高小鹏

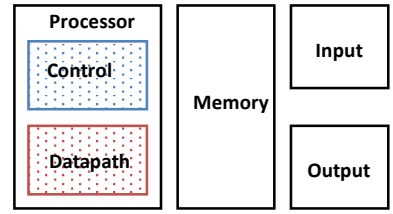
北京航空航天大学计算机学院
系统结构研究所

目录

- 实现控制
- 时钟方法

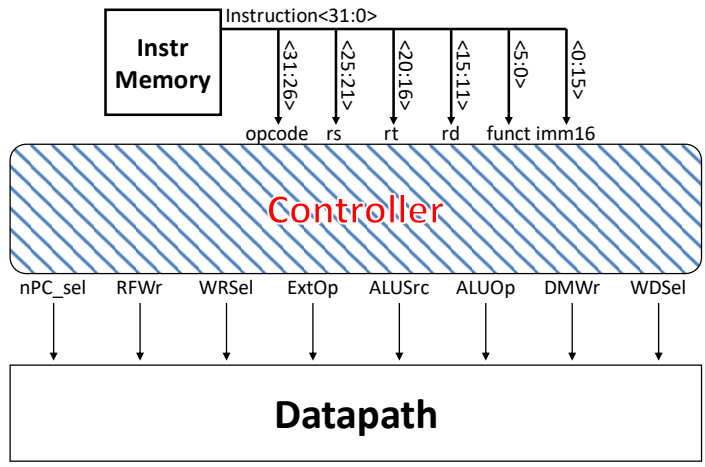
Processor Design Process

- Five steps to design a processor:
  1. Analyze instruction set → datapath requirements
  2. Select set of datapath components & establish clock methodology
  3. Assemble datapath meeting the requirements
  4. Analyze implementation of each instruction to determine setting of control points that effects the register transfer
  5. Assemble the control logic
     - Formulate Logic Equations
     - Design Circuits



Now

Purpose of Control



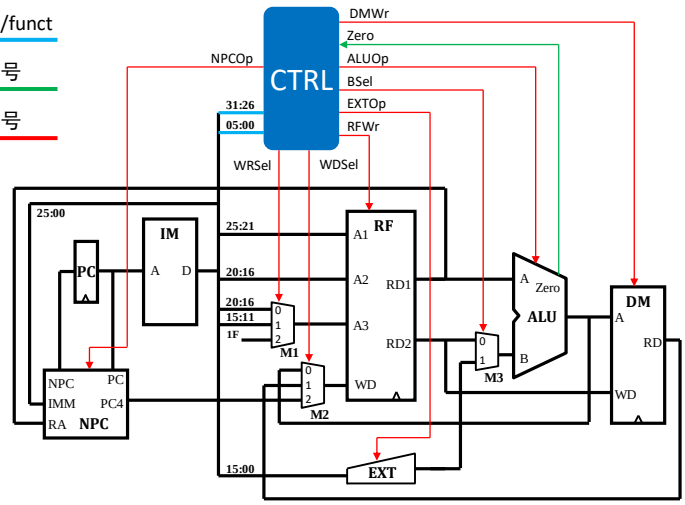
数据通路与控制器

- 控制器的职责：根据各指令的opcode（指令的31:26位）和funct（指令的05:00位），产生各功能部件的控制信号表达式

输入：opcode/funct

输入：状态信号

输出：控制信号



合成各指令的控制信号取值矩阵

- 示例：addu指令

环节	步骤	语义	连接关系	功能部件	控制信号取值
1	读指令	读取指令 计算下条指令地址 更新PC	<PC.DO,IM.A> <PC.DO,NPC.PC> <NPC.NPC,PC.DI>	IM PC NPC	**NPCOp:PC+4**
2	读操作数		<IM.D[25:21],RF.A1> <IM.D[20:16],RF.A2>	RF	
3	执行	ALU执行加法	<RF.RD1,ALU.A> <RF.RD2,ALU.B>	ALU	**ALUOp:ADD**
4	回写	计算结果回写至rd寄存器	<ALU.C,RF.WD> <IM.D[15:11],RF.A3>	RF	**RFWr:1**

- 对于不涉及的组合逻辑功能部件，如EXT，其控制信号可以设置为X
  - X有助于化简。一般来说，令其为0会得到更简单的控制信号表达式
- 对于不涉及的存储功能部件，如DM，其写使能必须设置为0
  - 如果DM的写使能不设置为0，则意味着addu指令也会导致DM被写入

指令	NPCOp	RFWr	EXTOp	ALUOp	DMWr	M1Sel	M2Sel	M3Sel
addu	PC+4	1	X	ADD	0	?	?	?

合成各指令的控制信号取值矩阵

- 完整的控制信号取值矩阵（不包含MUX的控制信号）

指令	NPCOp	RFWr	EXTOp	ALUOp	DMWr	M1Sel	M2Sel	M3Sel
addu	PC+4	1	X	ADD	0	?	?	?
subu	PC+4	1	X	SUB	0	?	?	?
ori	PC+4	1	0	OR	0	?	?	?
lw	PC+4	1	1	ADD	0	?	?	?
sw	PC+4	0	1	ADD	1	?	?	?
beq	0b01	0	X	SUB	0	?	?	?
jal	0b10	1	X	X	0	?	?	?
jr	0b11	0	X	X	0	?	?	?

合成各指令的控制信号取值矩阵

- 构造MUX控制信号真值表
  - 根据指令与输入信号关系及端口与输入源关系，就可以“自动”确定控制信号表达式。以sw指令为例
    - ①指令的20:16位指定寄存器堆的回写编号，由此可以查出MUX的端口编号
    - ②MUX的端口编号与控制信号取值是一一对应的

指令与RF.A3

指令	RF.A3
addu	IM.D[15:11]
subu	IM.D[15:11]
ori	IM.D[20:16]
lw	IM.D[20:16]
sw	IM.D[20:16]
beq	
jal	0x1F
jr	

RF.A3的MUX

端口编号	输入源
0	IM.D[15:11]
1	IM.D[20:16]
2	0x1F

①　②

MUX控制信号真值表

指令	端口编号	控制信号取值
add	0端口	0b00
sub	0端口	0b00
ori	1端口	0b01
lw	1端口	0b01
sw	1端口	0b01
slt	0端口	0b00
beq		
jal	2端口	0b10
jr		

合成各指令的控制信号取值矩阵

- 完整的控制信号取值矩阵（包含MUX的控制信号）

指令	NPCOp	RFWr	EXTOp	ALUOp	DMWr	M1Sel	M2Sel	M3Sel
addu	PC+4	1	X	ADD	0	00	00	0
subu	PC+4	1	X	SUB	0	00	00	0
ori	PC+4	1	0	OR	0	XX	00	1
lw	PC+4	1	1	ADD	0	XX	01	1
sw	PC+4	0	1	ADD	1	XX	XX	1
beq	0b01	0	X	SUB	0	XX	XX	0
jal	0b10	1	X	X	0	XX	10	X
jr	0b11	0	X	X	0	XX	XX	X

- 对比NPCOp/ALUOp与M1Sel/M2Sel/M3Sel，显然宏定义的表达方式更好：不仅易于理解，而且有利于工程维护

合成各指令的控制信号取值矩阵

- 完整的控制信号取值矩阵（包含MUX的控制信号）

指令	NPCOp	RFWr	EXTOp	ALUOp	DMWr	M1Sel	M2Sel	M3Sel
addu	PC+4	1	X	ADD	0	00	00	0
subu	PC+4	1	X	SUB	0	00	00	0
ori	PC+4	1	0	OR	0	XX	00	1
lw	PC+4	1	1	ADD	0	XX	01	1
sw	PC+4	0	1	ADD	1	XX	XX	1
beq	0b01	0	X	SUB	0	XX	XX	0
jal	0b10	1	X	X	0	XX	10	X
jr	0b11	0	X	X	0	XX	XX	X

- Q：该如何将这个表格转换为组合逻辑表达式？

计算机组成与实现

构造控制信号的布尔表达式

- 第1步：将每条指令用一个变量与之对应
  - 使用指令的opcode与funct产生变量
  - 为了提高可读性，变量名就是指令名
  - 由于R型指令的opcode都为0，因此需产生一个单独变量Rtype
- 示例

```
beq = op[5]'·op[4]'·op[3]'·op[2]·op[1]'·op[0]'
Rtype = op[5]'·op[4]'·op[3]'·op[2]'·op[1]'·op[0]'
add = Rtype·funct[5]·funct[4]'·funct[3]'
    ·funct[2]'·funct[1]'·funct[0]'
```

指令	opcde	funct
…	…	…
addu	000000	100000
beq	000100	
…	…	…

构造控制信号的布尔表达式

- 第1步：将每条指令用一个变量与之对应
  - 使用指令的opcode与funct产生变量
  - 为了提高可读性，变量名就是指令名
  - 由于R型指令的opcode都为0，因此需产生一个单独变量Rtype
- 示例：Verilog表达式

```
assign beq = (op==`BEQ);
assign Rtype = (op==6'b000000) ;
assign add = Rtype&(funct==`ADDU) ;
```

`BEQ和`ADDU是宏定义。宏有助于提高可读性，更易于工程维护

指令	opcde	funct
…	…	…
addu	000000	100000
beq	000100	
…	…	…

构造控制信号的布尔表达式

- 第2步：使用指令变量构造控制信号
  - ※构造控制信号表达式时，要确保包含了所有取值为1的指令
- 示例

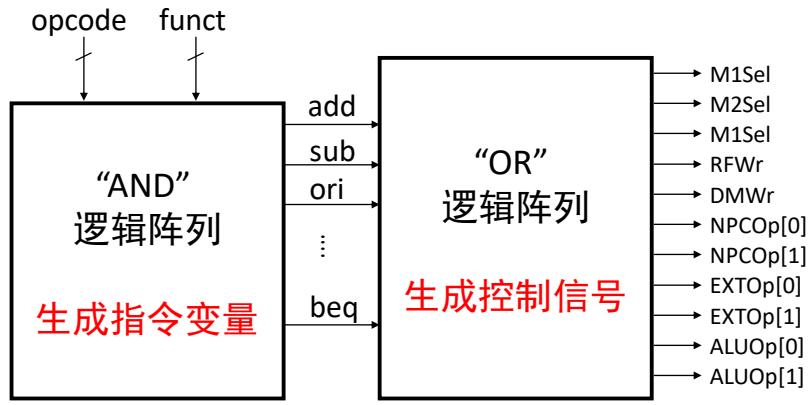
DMWr = sw

RFWr = add + sub + ori + lw + jal

指令	NPCOp	RFWr	EXTOp	ALUOp	DMWr	M1Sel	M2Sel	M3Sel
addu	PC+4	1	X	ADD	0	00	00	0
subu	PC+4	1	X	SUB	0	00	00	0
ori	PC+4	1	0	OR	0	XX	00	1
lw	PC+4	1	1	ADD	0	XX	01	1
sw	PC+4	0	1	ADD	1	XX	XX	1
beq	0b01	0	X	SUB	0	XX	XX	0
jal	0b10	1	X	X	0	XX	10	X
jr	0b11	0	X	X	0	XX	XX	X

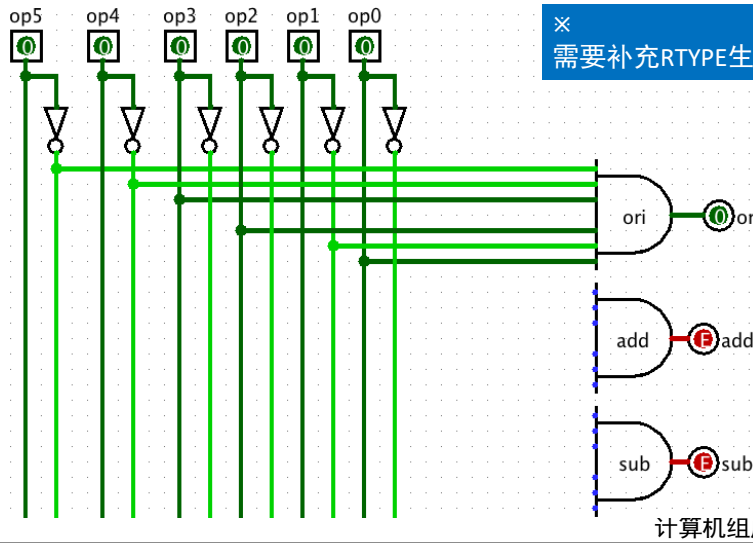
实现控制器

- 用AND和OR两个阵列来实现控制器



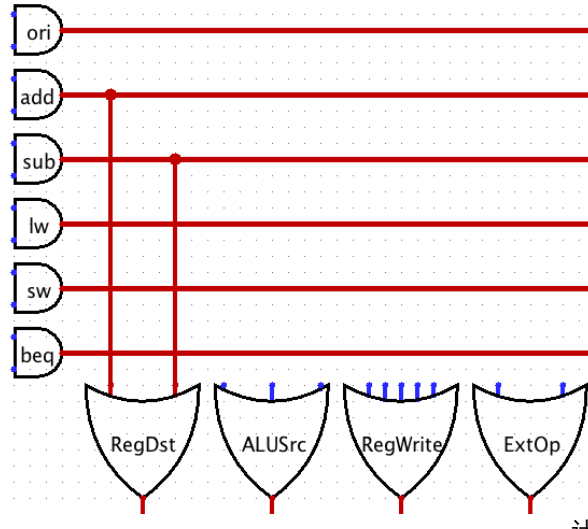
实现控制器

- 用AND和OR两个阵列来实现控制器
  - AND阵列：用Logisim实现



实现控制器

- 用AND和OR两个阵列来实现控制器
  - OR阵列：用Logisim实现



目录

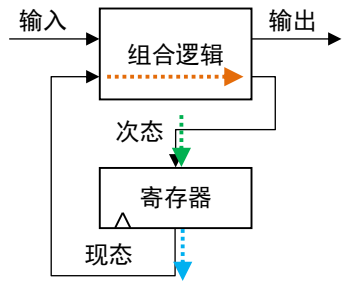
- 实现控制
- 时钟方法

寄存器时序术语

- 建立时间（Setup Time）：输入信号在时钟上升沿之前就必须有效的时间
- 保持时间（Hold Time）：输入信号在时钟上升沿之后仍然必须保持有效的时间
- 输出延迟（ CLK-to-Q）：输出信号在时钟上升沿之后输出有效值的时间

最大时钟频率

- 如何计算电路的最大频率？
  - 最大频率取决于最大延迟
  - 最大延迟取决于为了确保寄存器的正确输入所需要的时间

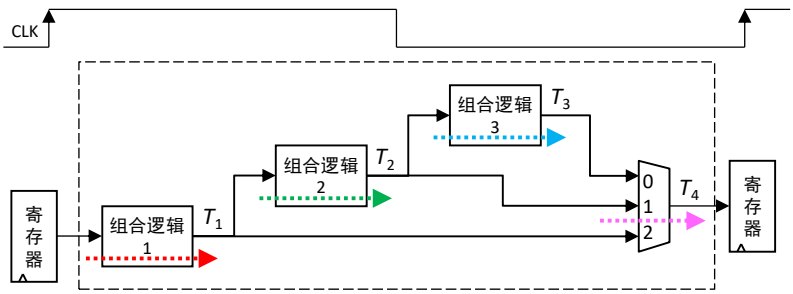


T = 建立时间 +
输出延迟 +
组合逻辑延迟

$f_{max} = 1/T$

关键路径

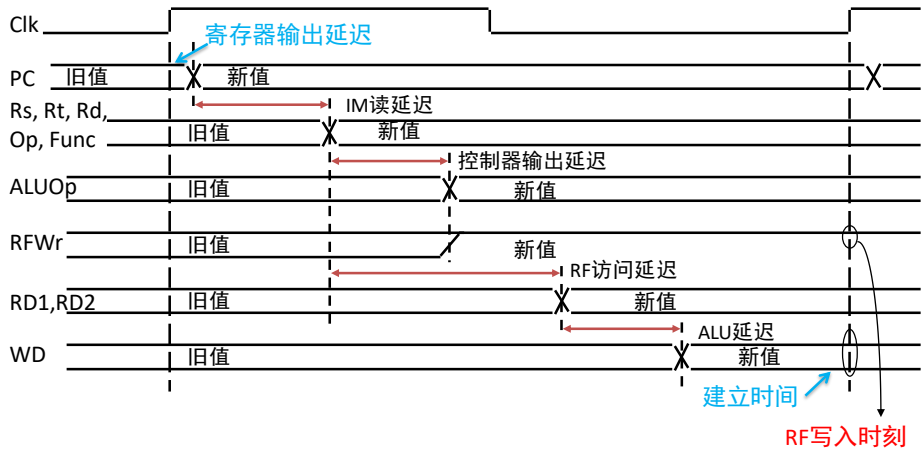
- 关键路径：电路中的任意2个寄存器之间的最大延迟
- 电路的时钟周期必须大于关键路径，否则信号将不能正确的传递到下一个寄存器



关键路径 = 组合逻辑延迟1 +
组合逻辑延迟2 +
组合逻辑延迟3 +
MUX延迟

Add指令执行延迟分析

- 执行延迟包括：PC输出延迟、IM读延迟、控制器输出延迟、ALU运算延迟、寄存器建立延迟



单周期性能

- 假设：RF的读写延迟均为100ps，其他部件延迟为200ps
- 最大时钟频率是多少？
  - lw延迟最大，即关键路径为800ns。因此，最大时钟频率为1.25GHz

指令	读取指令	读寄存器	ALU	数据存取	写寄存器	理想执行时间	实际执行时间
addu	200	100	200		100	600	800
subu	200	100	200		100	600	800
ori	200	100	200		100	600	800
lw	200	100	200	200	100	800	800
sw	200	100	200	200		700	800
beq	200	100	200			500	800
jal	200				100	300	800
jr	200	100				300	800

- 如何提高时钟频率？