

单周期CPU

通路设计考虑：

哈佛体系结构：使用指令存储区（IM）和数据存储区（DM）分别保存指令和数据

一个时钟周期内部没有状态保存 故不能有回路
即一个时钟周期内部是组合逻辑

数据通路设计表格（Excel）

状态元件：程序计数器 指令存储器 寄存器文件 数据存储器

PC

信号名	方向	位宽	描述
clk	I	1	时钟信号
reset	I	1	同步复位信号
NPC	I	32	下一条要被执行的指令的地址
PC	O	32	当前正在执行的指令的地址

NPC

信号名	方向	位宽	描述
PC	I	32	当前正在执行的指令的地址
zero	I	1	ALU计算结果是否为零
Branch	I	1	是否为beq指令
SignExt	I	32	16位立即数零扩展/符号扩展的32位
instr_index	I	26	当前指令数据的0:25，用于计算jal所要跳转的地址
jr	I	1	是否为jr指令
PC_RD1_jr	I	32	当前指令数据的21:25所表示rs寄存器中存储的jr指令跳转地址
jal	I	1	是否为jal指令
PC_4	I	32	PC+4的值，用于实现jal中的地址存储
NPC	O	32	下一条要被执行的指令的地址

IM

信号名	方向	位宽	描述
PC	I	32	当前正在执行的指令的地址
instruction	O	32	当前正在执行的指令
Op	O	6	当前正在执行的指令31:26 Op字段
Func	O	6	当前正在执行的指令5:0 Func字段
rs	O	5	当前正在执行的指令25:21 rs寄存器
rt	O	5	当前正在执行的指令20:16 rt寄存器
rd	O	5	当前正在执行的指令15:11 rd寄存器
instr_index	O	26	当前正在执行的指令25:0 用于计算jal所要跳转的地址
Imm	O	16	当前正在执行的指令15:0 立即数

EXT

信号名	方向	位宽	描述
extendSigna	I	1	符号扩展(1)/零扩展(0)
in	I	16	16位立即数
out	O	32	扩展后32位结果

ALU

信号名	方向	位宽	描述
A	I	32	第一个运算数
B	I	32	第二个运算数
ALU_operation	I	3	选择运算方式(扩展要点)
zero	O	1	ALU_result为零(1)/不为零(0)
ALU_result	O	32	运算结果

GRF

信号名	方向	位宽	描述
clk	I	1	时钟信号
reset	I	1	同步复位信号 1：复位信号有效 0：复位信号无效
PC	I	1	当前指令地址

信号名	方向	位宽	描述
WE	I	1	写使能信号 1: 写入有效 0: 写入失效
WA	I	5	地址输入信号, 指定32个寄存器中的一个, 将其作为写入目标
WD	I	32	数据输入信号
RA1	I	5	地址输入信号, 指定32个寄存器中的一个, 将其中的数据读出到 RD1
RA2	I	5	地址输入信号, 指定32个寄存器中的一个, 将其中的数据读出到 RD2
RD1	O	32	输出A1指定的寄存器中的32位数据
RD2	O	32	输出A2指定的寄存器中的32位数据
PC_RD1_jr	O	32	输出A1指定的寄存器中的32位数据(用于jr指令跳转)

DM

信号名	方向	位宽	描述
clk	I	1	时钟信号
reset	I	1	同步复位信号 1: 复位信号有效 0: 复位信号无效
PC	I	1	当前指令地址
MemWrite	I	1	写使能信号 1: 写入有效 0: 写入失效
A	I	12	内存地址输入信号
D	I	32	数据输入信号
RD	O	32	输出A指定的内存中的32位数据

控制器设计考虑:

控制器分成两部分: **主控单元**和**ALU控制单元**

主控单元Control

输入: 指令操作码字段 Op (指令31:26位)

输出: 8个控制信号 ALU控制单元所需的2位输入ALUOp

- ALUOp指明ALU的运算类型
- 00: 访存指令所需加法
- 11: beq 指令所需减法
- 10: R型指令功能码决定
- 01: ori指令所需或

控制信号	
Branch	是否需要跳转分支
RegWrite	Reg写数据使能
WriteAofRegByC	寄存器堆写入端地址选择
WriteDtoReg	寄存器堆写入端数据源选择
MemRead	DM读数据使能
MemWriteByC	DM写数据使能
ALUOP	用于ALU控制单元ALU_Control
ALUB	ALU输入端B数据源选择
ExtendSel	符号拓展(1)还是零拓展(0)
jal	指令是jal(1)/不是(0)

ALU控制单元ALU_Control

输入：主控单元生成的ALUOP 功能码字段Func（指令5:0位）

输出：ALU运算控制信号 ALU_operation（4位） 指令是否是jr

		ALU_operation	ALU运算
A	B	000	A & B
A	B	001	A B
A	B	010	A + B
A	B	011	A - B

注：ALU运算可以是多项式