

控制器设计

计52 王纪霆 2015011251

计52 于志竟成 2015011275

计52 周京汉 2015011245

控制信号

控制信号名	作用	位数
regSrcA	控制选择哪一寄存器作为ALU操作数A	4
regSrcB	控制选择哪一寄存器作为ALU操作数B	4
immeCtrl	控制原指令中立即数的位置	3(共5种情况)
dstSrc	控制写回的目标寄存器地址	4
immeExt	控制立即数应当符号扩展(1)或零扩展(0)	1
oprSrcB	选择操作数B是来自寄存器(0)还是来自立即数(1)	1
ALUop	ALU的操作码	4
isBranch	控制是否需要跳转	1
isCond	控制是条件跳转(1)或无条件跳转(0)	1
isRelative	控制跳转位置来源(PC+immediate(1)/ALUres(0))	1
isMFPC	比较特殊，控制ALU结果是否来自PC	1
ramWrite	数据写入信号(1表示写入)	1
ramRead	数据读出信号(1表示读出)	1
wbSrc	写回时数据来源是ram(0)或是ALU计算结果(1)	1
wbEN	是否需要写回数据(1为是)	1

具体说明

为了简洁，以下我们用[a:b]表示指令码中a-b段的数据。

基本的8个寄存器(0000-0111)，加上IH(1000)，SP(1001)，T(1010)使用4位编码即可。

立即数分为5种情况，为其分别编码：

000	001	010	011	100
[7:0]	[3:0]	[4:2]	[10:0]	[4:0]

ALUop控制ALU需要进行的操作。4位控制码应当足够了，需要的操作包括：

ALUop	操作	ALUop	操作
0000	加法	0111	判断A=B
0001	减法	1000	判断A!=B
0010	AND	1001	输出A
0011	OR	1010	输出B
0100	逻辑左移	1011	判断A>=B（无符号数）
0101	算术右移		
0110	逻辑右移		

指令分析

我们将相似控制信号合并为一列,定义：

- regSrc = regSrcA, regSrcB
- branch = isBranch + isCond+ isRelative
- ramCtrl = ramWrite + ramRead
- wbCtrl = wbSrc + wbEN

immeExt仅在指令为LI时为0，下表中不再列出。

isMFPC仅在指令为MFPC时为1，下表中不再列出。

基本指令：

指令	regSrc	immeCtrl	dstSrc	oprSrcB	ALUop	branch	ramCtrl	wbCt
ADDIU	0[10:8],0000	000	0[10:8]	1	0000	000	00	11
ADDIU3	0[7:5],0000	001	0[10:8]	1	0000	000	00	11
ADDSP	1001,0000	000	1001	1	0000	000	00	11
ADDU	0[10:8],0[7:5]	000	0[4:2]	0	0000	000	00	11
AND	0[10:8],0[7:5]	000	0[10:8]	0	0010	000	00	11
B	0000,0000	011	0000	1	1010	101	00	10
BEQZ	0[10:8],0000	000	0000	0	0111	111	00	00
BNEZ	0[10:8],0000	000	0000	0	1000	111	00	00
BTEQZ	1010,0000	000	0000	0	0111	111	00	00
CMP	0[10:8],0[7:5]	000	1010	0	1000	000	00	11
JR	0[10:8],0000	000	0000	0	1001	100	00	00
LI	0000,0000	000	0[10:8]	1	1010	000	00	11
LW	0[10:8],0000	100	0[7:5]	1	0000	000	01	10
LW_SP	1001,0000	000	0[10:8]	1	0000	000	01	10
MFIH	1000,0000	000	0[10:8]	0	1001	000	00	11
MFPC	0000,0000	000	0[10:8]	0	1001	000	00	11
MTIH	0[10:8],0000	000	1000	0	1001	000	00	11
NOP	0000,0000	000	0000	0	0000	000	00	00
OR	0[10:8],0[7:5]	000	0[10:8]	0	0011	000	00	11
SLL	0[7:5],0000	010	0[10:8]	1	0100	000	00	11
SRA	0[7:5],0000	010	0[10:8]	1	0101	000	00	11
SUBU	0[10:8],0[7:5]	000	0[4:2]	0	0001	000	00	11
SW	0[10:8],0[7:5]	100	0000	1	0000	000	10	00
SW_SP	1001,0[10:8]	000	0000	1	0000	000	10	00

扩展指令：

指令	regSrc	immeCtrl	dstSrc	oprSrcB	ALUop	branch	ramCtrl	wbCtrl
SLLV	0[7:5],0[10:8]	000	0[7:5]	0	0100	000	00	11
SRL	0[7:5],0000	010	0[10:8]	1	0110	000	00	11
SLTU	0[10:8],0[7:5]	000	1010	0	1011	000	00	11
BTEQZ	1010,0000	000	0[4:2]	0	0111	110	00	00
SRLV	0[7:5],0[10:8]	000	0[7:5]	0	0110	000	00	11

流水线存储数据设计

为了存储流水线数据，包括4个控制部件，包括IF/ID, ID/EX, EX/MEM, MEM/WB四个寄存器部件。

其需要保存的信号如下：

IF/ID

信号名	说明	位数
IFPC	保存PC	16
inst	保存指令	16

ID/EX

信号名	说明	位数
regA	保存寄存器A的值	16
regB	保存寄存器B的值	16
regAN	保存寄存器A的编号（用于冲突检测）	4
regBN	保存寄存器B的编号（用于冲突检测）	4
immediate	保存立即数	16
IDPC	保存PC	16

此外还需要保存以下控制器信号：dstSrc, oprSrcB, ALUop, isBranch, isCond, isRelative, isMFPC, ramWrite, ramRead, wbSrc, wbEN.

EX/MEM

信号名	说明	位数
regB	保存寄存器B的值	16
ALUres	保存ALU计算结果	16

此外还需要保存以下控制器信号：ramWrite, ramRead, wbSrc, wbEN, dstSrc.

MEM/WB

信号名	说明	位数
ramData	保存ram读出数据	16
ALUres	保存ALU计算结果	16

此外还需要保存以下控制器信号：wbSrc, wbEN, dstSrc.