# 分支跳转指令行为描述

本文档适用于以下指令：BEQ, BEQL, BGEZ, BGTZ, BLEZ, BLTZ, BNE, J, JAL, JR 共10条指令

其中BEQ , BEQL, BGEZ, BGTZ, BLEZ, BLTZ, BNE 7条branch指令为I-type指令。

J, JAL 为I-type指令。

JR指令为特殊的R-type指令。

# IF段

从PC中获得指令地址送入指令存储器，获得指令之后送入IF/ID流水线寄存器。

**对于7条branch指令：**

PC\_out送到BPU。由BPU来预测这条branch指令是跳转还是不跳转，BPU的输出送到一个选择器，由CU来控制选择下一条PC地址。

**对于JR指令：**

跟普通指令相同，没有特殊地方

# IF/ID段

IF/ID段流水线寄存器对指令进行分割，输出完整的指令ifid\_instr[31:0]提供给译码器。

**对于7条branch指令：**

输出ifid\_rs\_addr[4:0] , ifid\_rt\_addr[4:0]，由branch指令使用。

ifid\_rd\_addr[4:0]输出为无关输出。

输出ifid\_imm[15:0]作为偏移量提供branch指令使用。

**对于j,jr,jal指令：**

没有特别的地方，只是简单地传送ifid\_rs\_addr[4:0] , ifid\_rt\_addr[4:0], ifid\_rd\_addr[4:0], ifid\_imm[15:0]到后面去使用

# ID段

译码器从IF/ID流水段寄存器获得完整的指令，进行译码。

**对于7条branch指令：**

Id\_rt\_data\_sel输出为1’b1，选择从通用寄存器组中送出的值。

id\_rd\_addr\_sel输出为2’b01，选择从IF/ID流水段寄存器中输出的ifid\_rt\_addr[4:0]。（实际上分支指令并不会把计算结果送回该值所对应的寄存器中）。

id\_rt\_addr\_sel输出为1’b0，选择从IF/ID流水段寄存器中输出的ifid\_rt\_addr[4:0]。Id\_imm\_ext[1:0]信号有效，输出为2’b01。

Id\_ctrl输出其他相关的控制信号，其中的id\_branch信号有效，各个branch指令对应的condition[2:0]如下表：

|  |  |
| --- | --- |
| Branch指令 | Condition[2:0](3’bxxx) |
| BEQ | 001 |
| BEQL | 001 |
| BGEZ | 011 |
| BGTZ | 100 |
| BLEZ | 101 |
| BLTZ | 110 |
| BNE | 010 |

位扩展单元将所有的ifid\_imm[15:0]进行符号扩展。

**对于J, JAL两条指令：**

Decoder中将id\_jump置为1

输出id\_jump\_addr[31:0]到PC前的选择器.

对于jal指令，将id\_rd\_addr\_sel赋值为3代表选择31号寄存器，即$ra，将来要写入返回地址,其他decoder信号无影响.

**对于JR指令：**

在decoder中将id\_ir置为1，id\_jump应为0，其它输出为无关输出,同时GPR中输出的rs\_data传送到pc选择器的选择端。

# ID/EX段

**对于7条branch指令：**

Idex\_condition[2:0]输出从译码器输出的id\_ctrl信号中的condtion[2:0]。

Idex\_exres\_sel[1:0]输出为2’b00，选择ALU的输出为计算结果。

Idex\_movz和idex\_movnz输出均为0，表明当前指令不是movz或者movnz指令。

Idex\_B\_sel选择1’b1。

Idex\_ALU\_op[3:0]输出从译码器中输出的id\_ctrl信号中的op[3:0]，均为4’b0001，表示作减法运算。

Idex\_imm\_ext[31:0]输出ID/EX流水寄存器的输入id\_imm\_ext[31:0]。

Idex\_op\_A[31:0]输出ID/EX流水寄存器的输入gpr\_rs[31:0]。

Idex\_op\_B[31:0]输出ID/EX流水寄存器的输入gpr\_rt[31:0]。

Idex\_rs\_addr[4:0]输出ID/EX流水寄存器的输入id\_ctrl信号中的相关内容，输出到转发单元。

Idex\_rt\_addr[4:0]输出ID/EX流水寄存器的输入id\_ctrl信号中的相关内容，输出到转发单元。

Idex\_branch输出1’b1。

Idex\_pc[31:0]输出到CU，用来判断branch是否预测正确。

**对于jal指令：**

Id\_reg\_w赋值为1, idex\_of\_w\_disen应该置为1，表示这条指令写寄存器并且溢出对其无影响，id\_reg\_w传送到exmem段寄存器

Idex\_mem\_w,idex\_mem\_r赋值为0，其他无关信号应该为默认的无效值

**对于j和jr指令：**

这个阶段发生跳转，跳转到相应的目标地址中

# EX段

由转发单元选择ALU的输入。

Id/ex流水段寄存器的输出idex\_ALU\_op[3:0]作为ALU的输入，控制ALU的行为。

ALU将计算结果输出（branch指令并不关心此结果），同时输出lf , zf , of标志，If, zf的结果送入ex/mem流水段寄存器。

Id/ex流水段寄存器的输出idex\_pc\_4[31:0]以及idex\_imm\_ext[31:0]相加，计算跳转的目标地址，结果送入ex/mem流水段寄存器的adder\_target[31:0]输入端口。

# EX/MEM

**对于7条branch指令：**

Exmem\_branch输出Idex流水段寄存器输出的branch信号。

Exmem\_condition[2:0]输出idex流水段寄存器输出的condition信号。

Exmem\_target[31:0]输出adder\_target[31:0]输入的值。

Exmem\_pc\_4[31:0]输出idex\_pc\_4[31:0]的值。

Exmem\_If和Exmem\_zf输出前面ALU计算后所得的标志位。

Exmem\_pc[31:0]为真正的跳转地址，输出到BPU。

**对于jal,j,jr指令：**

这个阶段不做任何事情,jal指令只是简单地将idex\_rd\_addr传送到下一个流水段寄存器

# MEM

**对于7条branch指令：**

ex/mem流水段寄存器输出的Exmem\_branch, Exmem\_condition[2:0], Exmem\_target[31:0], Exmem\_pc\_4[31:0], Exmem\_If, Exmem\_zf六个信号输出到分支地址计算模块，分支地址计算模块根据这些信号计算输出branch指令下一条指令真正的地址，结果送入CU。

CU判断branch指令是否预测正确，如果预测正确，流水段正常执行；如果预测错误，bpu\_write\_en写使能有效，修改BPU中的内容，将正确的跳转地址写入，同时ifid\_flush, idex\_flush, exmem\_flush信号有效，冲刷流水线。

**对于jal指令：**

Memwb\_reg\_w信号应该赋值为1，memwb\_rd\_addr应该是从上段传来的rd地址exmem\_rd\_addr