# 概述

在我们的CPU设计中，为了使流水线更加均匀，我们选择用4个周期完成branch指令，并且通过分支预测来降低控制冒险带来的cost，在EX段计算出分支目标地址（加法器）和分支条件标志位（ALU），在MEM段得到最终的目标地址。在MEM段的一个分支处理单元负责选择最终的目标地址。

# 模块信号说明

|  |  |
| --- | --- |
| Exmem\_branch | 是否为分支指令 |
| Exmem\_condition | 分支条件 |
| Exmem\_target | 分支目标地址 |
| Exmem\_pc\_4 | 当前PC+4 |
| Exmem\_lf | Less Flag |
| Exmem\_zf | Zero Flag |
| Final\_target | 真正的目标地址 |

# 具体功能

本模块要完成的工作与单周期CPU类似，但是有1点不同：

由于MIPS有延迟槽，而我们要做分支预测，所以我们会让编译器保证分支指令的下条指令一定为nop，以免发生错误。为了避免这些nop指令带来吞吐量下降，我们在分支预测时，如果预测不跳转，我们会将下条指令的pc设置为当前pc+8。所以在此单元中，如果这是一条分支指令，但是分支条件不满足，那么输出的Final\_target是pc+8而不是pc+4；

用伪代码表示：

If not branch:

Final\_target = pc +4

Else if condition\_satisfy(condition,lf,zf):

Final\_target = Exmem\_target

Else:

Final\_target = pc+8