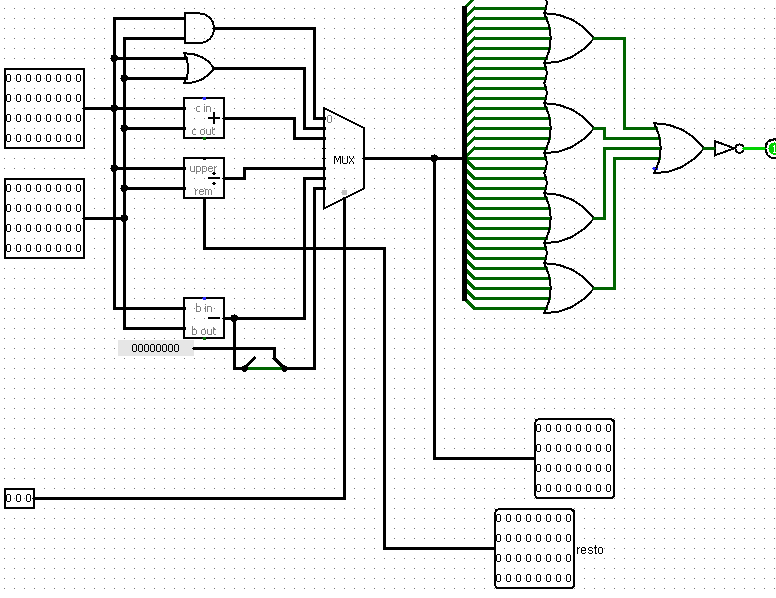
INF01113 – Organização de Computadores B (2023/1)

Trabalho pratico 1 - Grupo 4

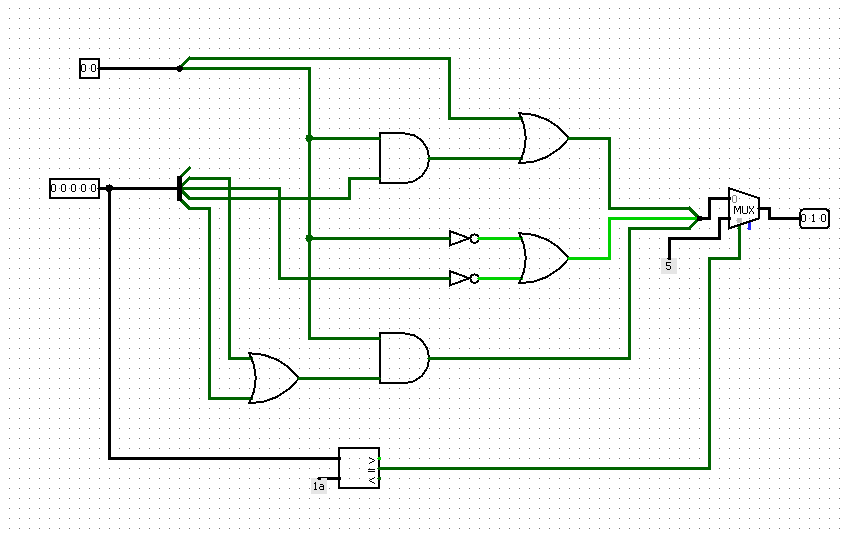
**Monociclo**

**Div**

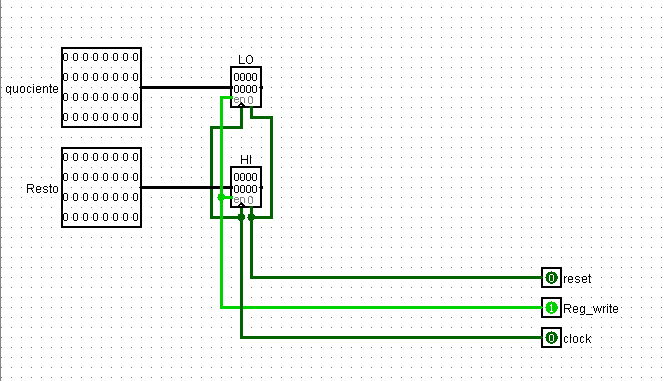
Primeiro adicionamos um divisor na ALU, com o rs sendo dividido por rt, e conectamos o quociente na porta 101 do mux da ALU e o resto num registrador(pino)



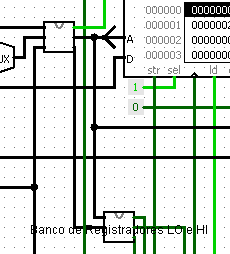
Agora, precisamos implementar uma lógica no ALU\_Control para que saia 101 dado o campo func seja igual a 11010, para isso apenas adicionamos um comparador para que compare a entrada do ALU\_Control com 11010, se for igual, o bit de saída aciona o mux para que a saída seja 101



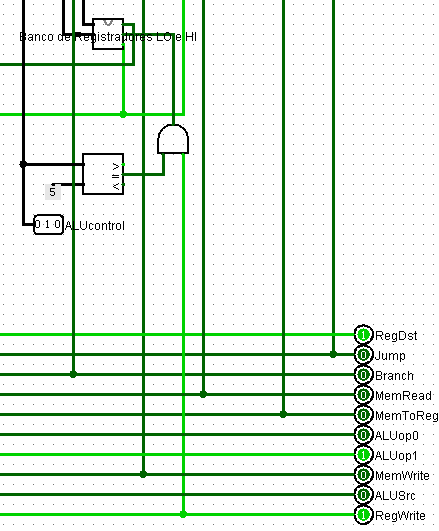
Agora só precisamos escrever o resultado, a operação DIV guarda o quociente e o resto da divisão em dois registradores especiais, LO e HI, respectivamente, logo criamos um banco de registrador auxiliar, BancoReg\_DIV, com esses dois registradores



E no Datapath ligamos a saída da ALU nesse banco de Registradores auxiliar



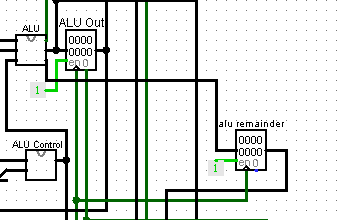
Porém, só queremos escrever nos registradores especiais quando for uma instrução DIV, então adicionamos um comparador entre a saída do ALU\_Control e 101, e o bit resultante do comparador numa porta AND junto com regWrite para saber que precisamos escrever nesses registradores especiais. Com isso temos a instrução de divisão no MIPS monociclo



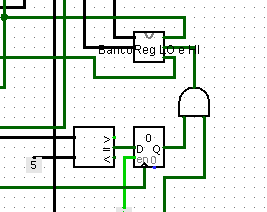
**Multiciclo**

**Div**

Mantemos quase todos os componentes do monociclo para o multiciclo, a diferença é que agora precisamos manter o resultado do resto da ALU entre um ciclo e outro, para isso adicionamos um registrador “alu remainder” na saída do resto da ALU



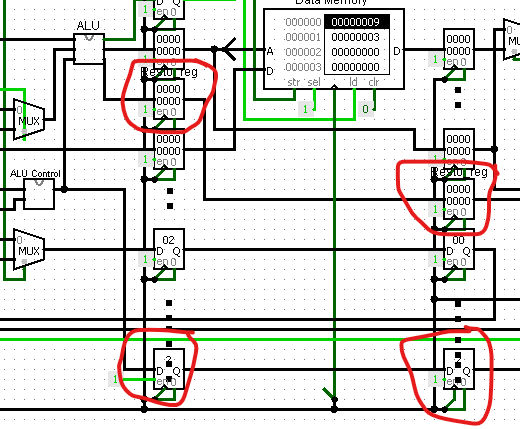
Tambem é necessário manter a informação que a operação realizada foi um DIV, para gravarmos nos registradores especiais, para isso apenas foi adicionado um registrador na saída do comparador da ALU\_Contro e 101



**Pipeline**

**Div**

No pipeline, aproveitamos várias partes dos componentes, as diferenças são que agora precisamos guardar o resto da alu e a saída do ALU\_Control por dois passos. Fazemos isso colocando um registrador intermediário para cada saída, em cada passo, ou seja, 2 regs por passo



A operação de escrever nos registradores LO e HI voltou a ser como era no monociclo, pois agora temos o dado da ALU\_Control preservado pelo segundo registrador intermediário do ALU\_Control

