Insper 2018

Design de Computadores:

Relógio Microprogramado

Hugo Mendes

Descrição do funcionamento do Circuito

A instrução do que deve ser feito está contido em cada linha da memória de instruções (ROM), e a partir desses bits ativa-se os pontos de controle do circuito, definindo quais registradores usar, seletores de funções, de entradas etc.

Minha ROM é dividida em 2 blocos, do bloco 0 ao 15 refere-se à lógica aplicada antes de dar 20 horas, a partir do momento que são 20 horas ou mais o programa passa para o segundo bloco em que se zera a unidade de hora quando chega em 4, não mais em 10. Segue a baixo o formato do dado contido em cada linha da ROM (instruções de 12 bits):

000 0001 1 0010

Seguindo o formato little-endian, os primeiros 3 bits mais significativos são responsável por dizer qual registrador está sendo operado no momento, os próximos 4 bits são o valor imediato a ser usado na segunda entrada da ALU, o bit único isolado é o que define qual a função que a ALU vai fazer, e os últimos 4 bits (menos significativos) são o endereço da próxima instrução para qual o program conter deve apontar. Afim de só atualizar o relógio a cada 1s, para cada “ciclo” de instruções realizadas na ROM, o ultimo endereço é um endereço nulo, ou seja, nada acontece nele, e a cada 1s o program conter é jogado ao primeiro estado da ROM afim de realizar as operações necessárias novamente. O relógio inteiro funciona com essa lógica. Entretanto, as funcionalidades extras, são como “interrupções” desse ciclo, elas ignoram o que aconteceria com a linha de instrução atual e realizam a instrução pedida pelo usuário, como aumentar a frequência ou ajustar o horário do relógio.

Dificuldades Enfrentadas

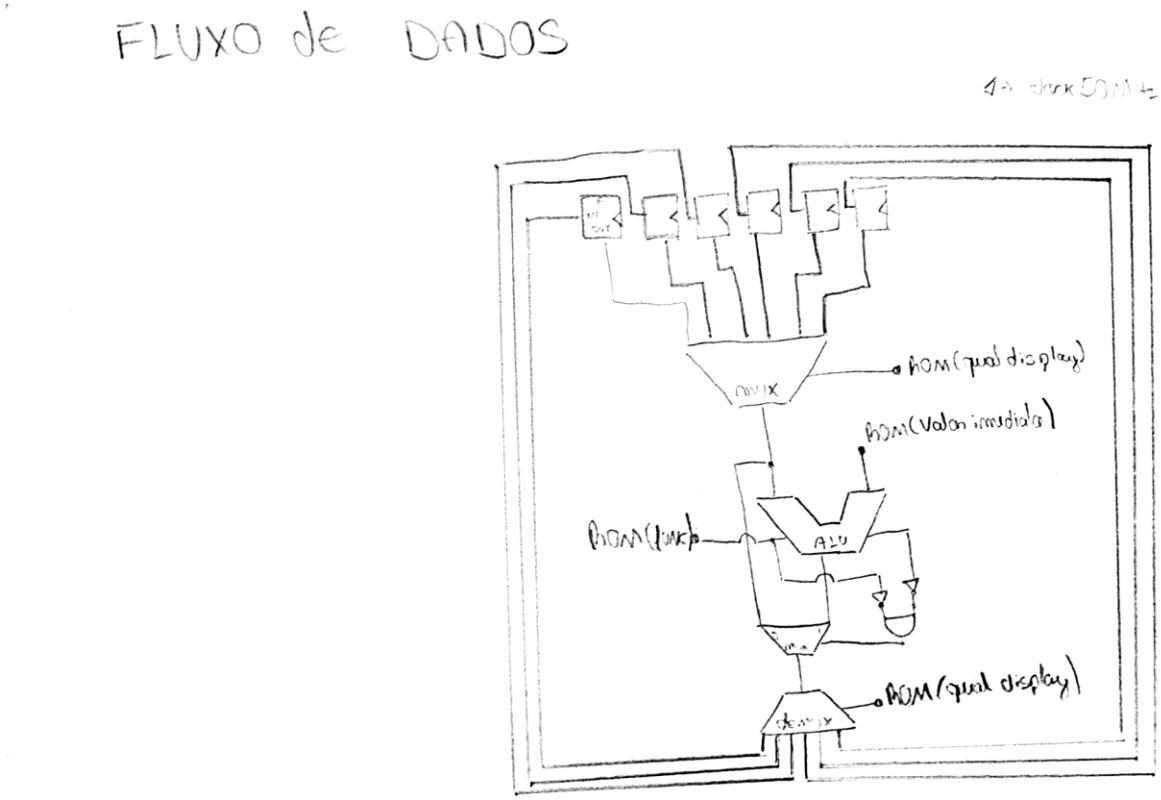
Basicamente fiz 3 relógios implementados de jeitos diferentes, pois nos primeiros 2 modos eu implementei com um circuito específico para atendar tais linhas de comando da ROM o que não era a ideia do projeto que consistia em montar um fluxo de dados genérico e utilizar-se das instruções da ROM para chegar no resultado desejado.

Outra dificuldade foi em lidar com o clock de 50mhz, demorou para entender que o meu fluxo de dados poderia rodar nessa frequência, e o clock de 1s serviria basicamente para reiniciar o ciclo de volta a primeira instrução (soma 1s).

Por fim, a terceira maior dificuldade foi debugar o que estava acontecendo quando não obtinha o resultado esperado uma vez que fazer as simulações era

demorado e complicado, inclua inclusive adaptar o código só para conseguir testar. Isso de fato atrapalhou bastante.

Imagem do Fluxo de Dados



Para não poluir o desenho não adicionei os mux que cuidam do enable de cada registrador, e a saída dos registradores ligadas aos conversores de 7 segmentos que mandam o dado para os displays da FPGA. É um circuito simples e que dá conta do recado.

Diagrama de Blocos do Circuito