

MAC0329 – Álgebra booleana e aplicações

DCC / IME-USP — Primeiro semestre de 2018

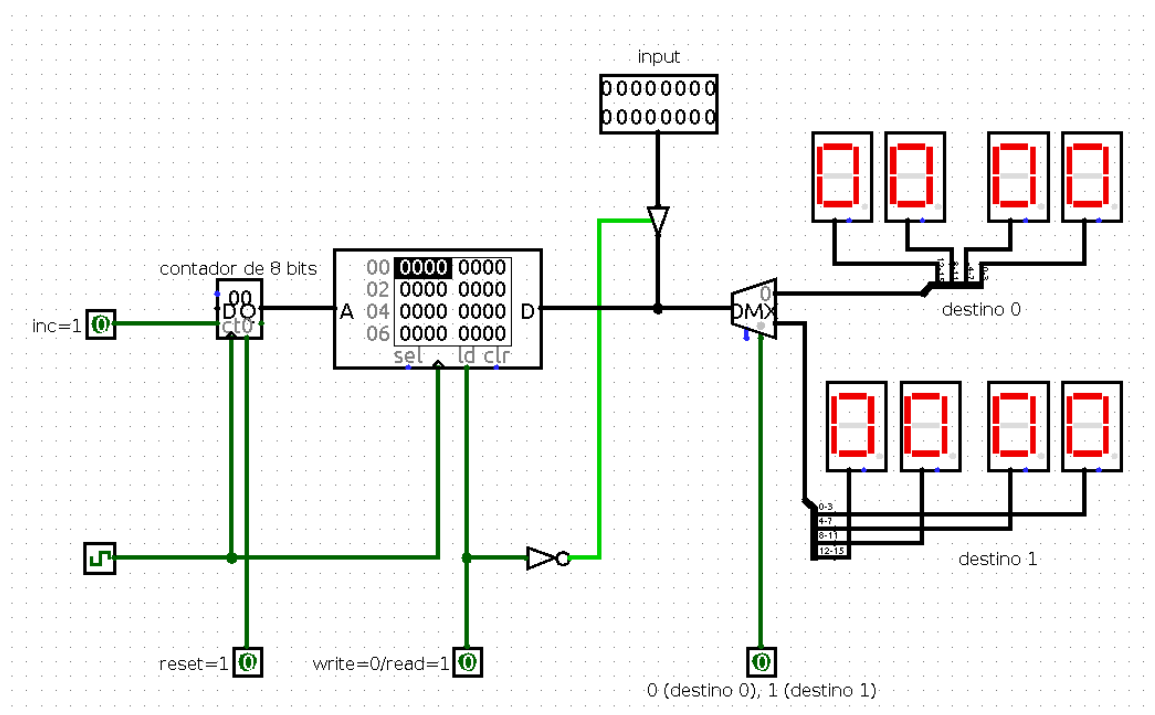
Projeto de circuito 3 – parte 1 (aquecimento)

Esta é a parte 1 do EP3. O objetivo do EP3 é construir e testar um circuito que implementa um ciclo de execução da CPU. Nesta primeira parte, iremos trabalhar alguns dos componentes desse circuito, que serão posteriormente complementados na parte 2. Se preferir, esta primeira parte pode ser pensada como uma etapa de aquecimento para o EP3.

O trabalho deve ser desenvolvido em grupo, mantendo-se preferencialmente os mesmos membros do EP anterior. Caso o EP anterior tenha sido desenvolvido individualmente ou em grupo de dois membros, neste projeto o arranjo pode ser alterado para se formar grupos de 3 membros. O trabalho pode ser dividido entre os membros, mas todos devem ter ciência sobre os detalhes do projeto.

1 Detalhamento

Uma tarefa importante da CPU é “ler” instruções ou dados armazenados na memória RAM. Igualmente importante é “escrever” (armazenar) dados na memória RAM. Para entender como isso funciona, considere o circuito abaixo.



O circuito contém vários componentes:

1. pino de entrada (input)
2. memória RAM
3. contador de 8 bits
4. DMUX
5. *display* de dígitos (que podem ser substituídos por pinos de saída)
6. *clock*
7. pinos de controle diversos

A memória RAM é composta de posições que são identificadas sequencialmente por números de 0 em diante. Cada posição pode armazenar n bits. A RAM possui uma porta (entrada) de endereço (A) e uma porta (entrada/saída) de dados (D). Dados podem ser escritos ou lidos da RAM, em uma posição específica A. O modo de operação leitura/escrita da RAM pode ser alterado por meio do pino `ld`. Uma operação é executada a cada pulso do *clock*.

No circuito acima, o endereço A é de 8 *bits*, e cada posição da memória armazena 16 *bits*. Quando o modo de operação é de escrita (`write`, logo, `ld=0`), o valor do `input` é armazenado na posição da RAM cujo endereço é A. Por outro lado, quando o modo de operação é de leitura (`read`, logo, `ld=1`), o dado armazenado na posição A da RAM é enviado para um dos destinos. O destino é controlado por um DMUX. Note que há um *buffer* controlado ligando os pinos de entrada (`input`) ao barramento de dados conectado à porta D da memória RAM. Este controle é importante pois o barramento de dados é usado para tráfego de dados em duas direções: no modo de operação escrita, os dados trafegam para esquerda no sentido de D; no modo de operação leitura, os dados trafegam para direita em sentido oposto a D. Assim, durante o modo leitura, o dado do `input` não deve ser enviado para o barramento de dados.

Tarefas

1. Construa o circuito acima. Podem ser usados os componentes disponíveis no Logisim.
2. Simule o circuito, repetindo os seguintes passos:
 - (a) prepare os dados (`input` e/ou dados nas posições de memória) manualmente
 - (b) acerte manualmente as *flags* de controle
 - (c) clique no *clock*
3. Altere o circuito para que no modo de operação `write` da RAM, em vez do valor que está no `input`, seja armazenado o valor do contador.
Após essa alteração, experimente habilitar o pulso (no menu `Simular`), i.e., “colocar o *clock* no automático”. Use uma frequência baixa para poder “ver” a dinâmica do processamento.
4. Altere o circuito para que no modo de operação `write`, seja possível escolher de onde vem o dado a ser armazenado; faça de modo que o dado possa vir ou do `input` ou do contador. Para possibilitar essa escolha, adicione mais um pino de controle.

Comentários finais

1. As tarefas acima visam promover a familiarização com os componentes, assim como com alguns conceitos importantes tais como a noção de controle e passo de execução (atrelado a um pulso de *clock*).
2. antes de um pulso de *clock*, todos os dados e sinais de controle devem estar ajustados de acordo com o processamento que se deseja executar.
3. Note que as linhas que conectam os diversos componentes transportam *bits*. Essas linhas são chamadas de barramentos e possuem largura (em *bits*) que depende do tipo da informação transportada. No nosso exemplo, a linha conectando a porta D da RAM, o `input` e o DMUX é um barramento de dados de largura 16 *bits*. Já a linha ligando a saída do contador à porta A da RAM é um barramento de endereço de 8 *bits*. As linhas ligadas aos pinos de controle são os barramentos de controle, no caso todos de largura 1 *bit*.