

# PCS3335 - Laboratório Digital A - Experiência 2

por Bruno de Carvalho Albertini

28/02/2025

Na experiência 2 vamos construir dois blocos básicos sequenciais que serão usados durante o projeto.

## Experiência 2

Leia este arquivo completamente antes de começar.

Na experiência 1 você se familiarizou com circuitos combinatórios e com a placa de prototipação. Nesta experiência, vamos construir dois blocos básicos sequenciais, que serão usados em quase todas experiências subsequentes: o registrador e o contador. Caso não se lembre o que são, por favor consulte a referência para lembrar.

Você deve montar um circuito digital composto por um contador e um registrador. O registrador, quando solicitado, deve guardar o valor que está no momento no contador. Para visualização, usaremos os *displays*, mostrando o valor em hexadecimal, sendo três dígitos para o valor de saída do contador e três dígitos para o valor de saída do registrador.

### Contador

O contador é síncrono, alimentado por um *clock* de 1Hz, e pode contar para cima ou para baixo. Também conta com *enable* que para a contagem, um *reset* síncrono ativo alto e carga paralela síncrona. Parametrizável no tamanho em bits ( $N$ ), sempre conta toda sequência (contador módulo completo  $2^N$  de forma circular para ambas direções). Não possui RCO para cascadeamento.

A prioridade é: *reset* mesmo sem *enable*, com *enable*: *load* e contagem.

```
entity counter is
  generic(
    WIDTH: natural := 8 -- Tamanho em bits
  );
  port (
    clock, reset: in std_logic; -- Clock e reset síncrono ativo alto
    enable: in std_logic; -- Habilita a contagem
    load: in std_logic; -- Carga paralela
    up: in std_logic; -- 0: contagem decrescente, 1: crescente
    data_i: in std_logic_vector(WIDTH-1 downto 0); -- Entrada paralela
    data_o: out std_logic_vector(WIDTH-1 downto 0) -- Saída paralela
  );
end counter;
```

### Registrador

O registrador é alimentado por um *clock* de 50MHz. Possui carga paralela e saída paralela, entrada serial única e saídas seriais segregadas (direita e esquerda), deslocamento para direita ou para esquerda. Conta ainda com um *reset* assíncrono ativo alto. Parametrizável no tamanho em bits (*N*).

```
entity shiftregister is
  generic(
    WIDTH: natural := 8 -- Tamanho em bits
  );
  port (
    clock, reset: in std_logic; -- Clock e reset assincrono ativo alto
    loadOrShift: in std_logic_vector(1 downto 0);
    -- 00: nada acontece
    -- 01: deslocamento para direita
    -- 10: deslocamento para esquerda
    -- 11: carga externa paralela
    serial_i: in std_logic; -- Entrada serial
    data_i: in std_logic_vector(WIDTH-1 downto 0); -- Entrada paralela
    data_o: out std_logic_vector(WIDTH-1 downto 0); -- Saída paralela
    serial_o_r: out std_logic; -- Saída serial direita
    serial_o_l: out std_logic -- Saída serial esquerda
  );
end shiftregister;
```

### Orientações para a Montagem

Você deve usar os *displays* conforme descrição (três dígitos para cada módulo). A entrada da carga paralela é pela chave e corresponde aos bits menos significativos. Note que a capacidade dos módulos é maior que as chaves. Todos os demais sinais devem ser via GPIO (*General Purpose Input/Output*) da placa. Você deve conectar o dispositivo Analog Discovery da sua bancada na GPIO, que será usada para duas funções:

- Ferramenta *Patterns*: para gerar o *clock* de 1Hz para o contador;
- Ferramenta *StaticIO*: para gerar os sinais de *reset* (comum aos módulos), carga paralela, deslocamento, entrada serial, direção de contagem ou deslocamento, todos sinais independentes para cada módulo.

Ao todo, a Analog Discovery possui 16 entradas/saídas digitais que podem ser utilizadas, porém elas são compartilhadas. O manual da Analog Discovery está disponível no e-Disciplinas, certifique-se de lê-lo e entendê-lo antes de planejar as ligações. Caso necessário, peça explicação para os técnicos do laboratório durante os horários de *Open-Lab* ou consulte seu professor.

Ao usar uma saída digital para gerar o *clock* você tem uma a menos a disposição para os demais sinais.

### Planejamento

Você deve apresentar o seu módulo registrador e seu módulo contador separadamente para o juiz eletrônico. A submissão para o juiz vale 1 ponto (0,5 pontos cada módulo) e a correção do seu planejamento pelo professor vale 4 pontos. Para esta experiência, o juiz aceita qualquer estrutura de VHDL e não tem limite de submissões, mas ambos da dupla precisam enviar o arquivo. A entidade dos módulos deve ser exatamente o que consta neste enunciado ou o juiz não conseguirá instanciar o seu módulo. Para o juiz, você deve apresentar somente o arquivo VHDL contendo a entidade do módulo (há dois *links* diferentes no e-Disciplinas, um para cada módulo).

No documento do seu planejamento, apresente **no mínimo** um RTL do circuito no Quartus, a ligação da GPIO com a Analog Discovery e a tabela de testes. Não esqueça de enviar o arquivo .QAR do seu projeto no Quartus junto com o planejamento.

[5 pontos]

O *link* para o juiz está no e-Disciplinas.

O limite são 99 submissões.

### Execução

Quando chegar no laboratório, veja se há observações no planejamento. Caso não tenha nenhuma, apresente-se ao professor antes de começar a montagem. Proceda com a montagem ligando a Analog Discovery na placa FPGA, configure seu projeto no Quartus, sintetize, configure a placa e faça os testes, tomando notas.

Resolva qualquer problema que encontrar, relatando no relatório a solução. Quando estiver satisfeito, chame o professor para ser avaliado.

[4 pontos]

### Relatório

Suba no e-Disciplinas um relatório contendo o resultado dos seus testes.

[1 ponto]

### Desafio

Não há desafio padrão para esta experiência. Caso seu professor proponha um desafio, a nota será incluída na execução. Note que, se for proposto um desafio e você opte por não realizá-lo, o professor poderá descontar até 3 pontos da sua execução, a depender da complexidade do desafio proposto.