

---

---

## Laboratório 7

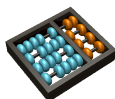
### Instruções:

Antes de iniciar o laboratório faça o download do arquivo 'lab07\_material\_v2018.1.zip' no moodle, ele contém arquivos de apoio para a execução dos exercícios, incluindo modelos para arquivos da entrega. Caso algum arquivo da entrega não possua um modelo, o formato é livre, desde que respeitados os nomes dos arquivos.

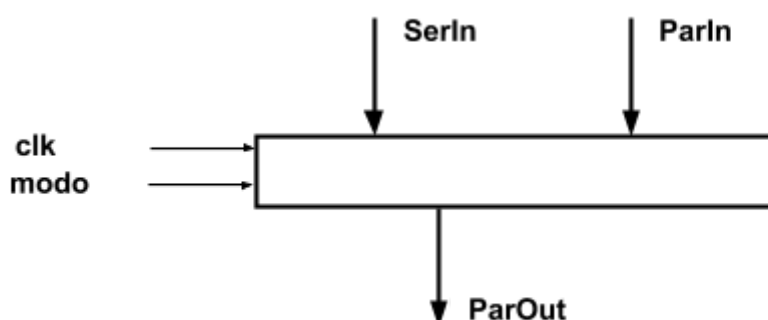
Os enunciados dos exercícios descrevem os nomes e tipos dos arquivos, entidades e conexões. Quaisquer entregas que não seguirem o padrão fornecido serão **desconsideradas**.

Deixe os arquivos que serão gravados na placa para demonstração em projetos separados e já pré-compilados, de forma que não seja necessária a recompilação do projeto no momento da demonstração. Assim, basta abrir o projeto correspondente e programar a placa, economizando o tempo da aula dedicado às avaliações.

Os arquivos que estiverem carregados no Moodle no momento do prazo final de entrega serão considerados como finais, mesmo que estejam marcados como rascunho. Portanto, não é necessário e nem recomendável marcar o envio como final, para que seja possível alterar os arquivos da submissão antes do prazo.



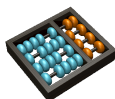
1. Seja um Registrador de Deslocamento Universal com as seguintes funções:
- Entradas: *par\_in* com N bits; modo com 2 bits; *clk*; *ser\_in* com 1 bit;
  - Saídas: *par\_out* com N bits;
  - Carga paralela síncrona, deslocamento síncrono para direita ou esquerda, de acordo com o sinal de modo:
    - *mode* = 00: NOP (manter valor atual);
    - *mode* = 01: shift left (bit de entrada serial = *ser\_in*);
    - *mode* = 10: shift right (bit de entrada serial = *ser\_in*);
    - *mode* = 11: carga paralela síncrona.



- a) Implemente em VHDL a entidade *shift\_register* <shift\_register.vhd>, um Registrador de Deslocamento Universal de N bits (genérico). Elabore uma simulação para testar o funcionamento do seu circuito, instanciado para N=6. **Entregue o arquivo shift\_register.vhd e um screenshot dos resultados da simulação em shift\_register.png.**
- b) Implementar na DE1 este registrador de 6 bits e testar o seu funcionamento. Utilizar os seguintes dispositivos de entrada e saída
1. *par\_out*: Leds
  2. *mode*: chaves - sugestão: SW(7), SW(8);
  3. *ser\_in*: chave - sugestão: SW(6);
  4. *par\_in*: chaves - sugestão: SW(5)..SW(0);
  5. *clk*: push button - sugestão: KEY(0).

**Não é necessário entregar nenhum arquivo, essa configuração será avaliada na demonstração.**

- c) Repetir o item c, agora utilizando uma chave - sugestão SW(9) - para acionar o clock ao invés de um push button
- d) Analisar o comportamento dos circuitos nos itens (b) e (c). **Entregar um arquivo shift\_register.txt descrevendo e explicando o que observou.**



---

## 2. Projeto e implementação de um relógio digital de acordo com a especificação

### Entidades requeridas:

- Entidade *clock* <clock.vhd>: *top-level* do projeto do relógio.
- Entidade *clk\_div* <clk\_div.vhd>: divisor de clock - recebe um sinal de 50 MHz (*clk*) e devolve um sinal de 1 Hz (*clk\_hz*).

### Saídas:

- Representações de horas (*hour\_dec* e *hour\_un*), minutos (*min\_dec* e *min\_un*) e segundos (*sec\_dec* e *sec\_un*) a serem conectadas em visores de 7 segmentos.

### Entradas:

- *clk*: entrada do sinal de relógio.
- *decimal*: valor da dezena para acertar o relógio;
- *unity*: valor da unidade para acertar o relógio;
- *set\_hour*, *set\_minute* e *set\_second*: indicador de qual valor está sendo acertado.

### Exemplo:

Para acertar o relógio para 23h 59min, por exemplo, define-se *decimal* para 2 e *unity* para 3 e salva-se no campo de horas com um pulso em *set\_hour*. Após, define-se *decimal* para 5 e *unity* para 9 e salva-se no campo de minutos com um pulso em *set\_minute*.

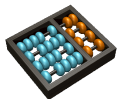
### Procedimento para o projeto e itens a serem entregues:

Projete o divisor de clock na entidade *clk\_div*. O sinal de clock resultante *clk\_hz* deverá ter um pulso em nível lógico a cada segundo, mas com largura de pulso igual à do pulso do clock padrão da placa (*clk*). Assim, seu relógio poderá ser facilmente controlado por *clk* e sincronizado por *clk\_div*.

Projete a entidade *clock*, utilizando o sinal de 1 Hz *clk\_hz* para sincronizar o funcionamento do relógio. O relógio deverá estar sempre ativo (sem “zerar”), e, quando acertado, o valor deve ser ajustado imediatamente (Dica: sincronize o acerto do relógio com *clk* ao invés de *clk\_hz*). Caso o usuário tente acertar o relógio para um valor inválido, este deverá ser ignorado, como se não houvesse uma tentativa de acerto. Além das entidades *clock* e *clk\_div*, você pode criar quantas entidades e/ou pacotes (*packages*) adicionais julgar necessário, e implementá-los em quantos arquivos VHDL achar necessário. Somente arquivos .vhd serão considerados. Não crie ou utilize *libraries* customizadas, mas você pode utilizar as *libraries* disponíveis na ferramenta.

Verifique o funcionamento do seu relógio na placa utilizando o arquivo *clock\_board.vhd* como top-level e o arquivo de assignments padrão da placa. **Esta configuração será avaliada na demonstração.**

---



---

---

Reúna todos os arquivos VHDL em um único arquivo zipado `clock.zip`. Os arquivos deverão estar na raiz da estrutura do `.zip`, e não distribuídos em pastas. Certifique-se de que você adicionou todos os arquivos VHDL para o funcionamento correto do seu projeto e de que a declaração das entidades `clock` e `clk_div` não foram alteradas.

3. Este exercício utiliza os componentes para controle de mouse `mouse_ctrl` e `ps2_iobase` disponíveis na pasta de “Material complementar > Componentes IO DE1-SoC (nova) > PS2 (mouse e teclado)” do curso no Moodle. Antes de fazê-lo, leia o tutorial “*Mouse e Teclado.pdf*” na pasta “Material complementar > Tutoriais IO DE1 (antiga)”. O tutorial foi escrito para um modelo anterior da placa utilizada, portanto existem algumas pequenas diferenças para o código disponível, que foi adaptado para a placa nova.

Faça o download dos componentes `mouse_ctrl`, `ps2_iobase` e `ps2_mouse_test` da pasta de material complementar. Inclua no mesmo projeto o seu componente `bin2hex` (Laboratório 02) e, utilizando `ps2_mouse_test` como entidade top-level e o arquivo de assignments padrão para a placa, grave o projeto na placa para testar com um mouse externo.

**Não é necessário entregar nenhum arquivo, essa configuração será avaliada na demonstração.**