Laboratório 08

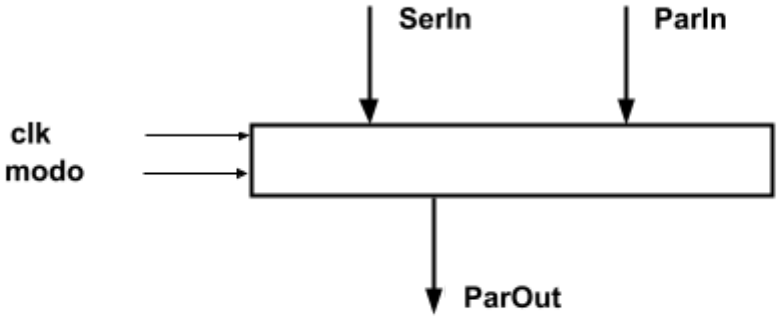
Instruções:

- Quando for demonstrar seu trabalho, tome nota do número da placa utilizada. O número da placa será utilizado para atribuir a nota ao grupo.
- A última página deste documento contém um checklist com todos os arquivos que fazem parte da entrega.
- Os nomes dos arquivos devem ser seguidos, e isso faz parte da avaliação.
- A entrega deverá estar em único arquivo .ZIP, com o nome **T_Lab08_RA.zip**, **T** é a turma, e **RA** é o RA do componente do grupo que fará a entrega. Por exemplo, B_Lab08_123456.zip é a entrega do grupo do aluno com o RA 123456, na turma B.
- Não divida ou agrupe em pastas os arquivos dentro do .ZIP.
- A entrega deve ser feita pelo Google Forms (https://forms.gle/pUcMCjhBm7faUMWWA). Você deve estar autenticado com uma conta do Google - pode ser uma conta pessoal ou da DAC.
- Apenas um integrante do grupo precisa fazer a entrega.
- Preste especial atenção aos nomes das entidades e sinais (entradas e saídas) descritos nos laboratórios. Isso também faz parte da avaliação.
- Se mais do que um arquivo for recebido para a mesma entrega, o último recebido será considerado. Utilize o mesmo RA do aluno entregando.
- Faça o download do arquivo **lab08_material_v2023.1.zip**. Esse arquivo já contém as descrições de *entity* necessárias para implementar os circuitos. Utilize elas, e não as altere.

Parte I - Registrador de deslocamento

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.



I.1. 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.

ENTREGA: Arquivo **shift_register.vhd.**

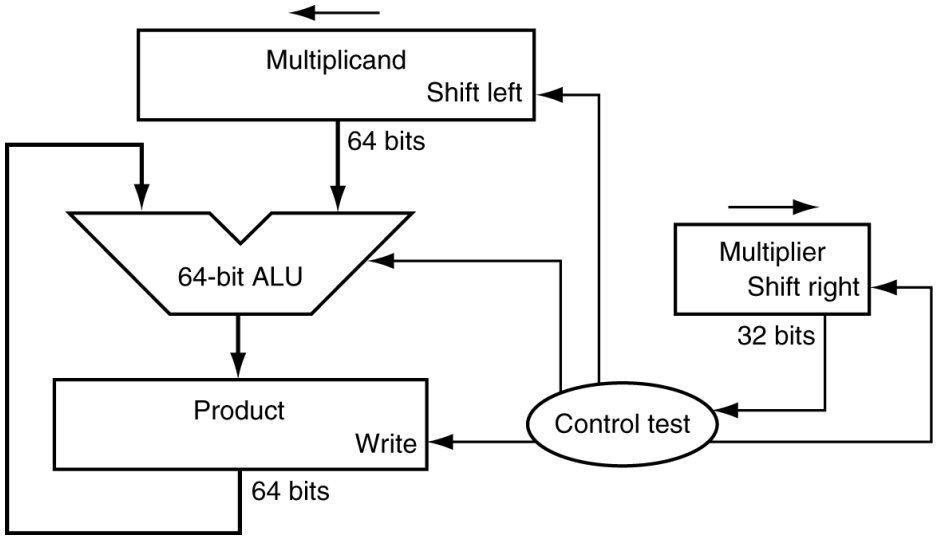
I.2. Implementar na DE1 este registrador de 6 bits e testar o seu funcionamento. Utilizar os seguintes dispositivos de entrada e saída

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).

I.3. Repetir o item c, agora utilizando uma chave - sugestão SW(9) - para acionar o clock ao invés de um push button. O que aconteceu de diferente em comparação com o item I.2? Procure uma explicação para o fenômeno observado. *Dica: utilize o manual da placa.*

Parte II - Multiplicador multiciclo

O diagrama a seguir representa o mecanismo de multiplicação de dois números inteiros sem sinal em múltiplos ciclos. Nele, considerando o resultado (produto) da multiplicação inicializado em zero, o multiplicando é estendido para o dobro do tamanho da palavra e deslocado para a esquerda, e o multiplicador deslocado para a direita a cada ciclo de clock. Também a cada ciclo de clock, o bit menos significativo do multiplicador e avaliado e, se for igual a 1, o multiplicando é somado no produto. A operação de multiplicação estará pronta após, no máximo, N ciclos de clock, onde N é o comprimento (em bits) dos operandos.



II.1. Descreva em VHDL o hardware multiplicador multiciclo acima para operandos de N bits (e produto de 2*N bits), obedecendo a seguinte interface:

```
entity multiplier is
      generic (
            N : integer := 4
      );
      port (
            a, b : in std_logic_vector(N-1 downto 0);
            -- Operandos (multiplicador e multiplicando)
            r : out std_logic_vector(2*N-1 downto 0);
            -- Resultado (produto)
            clk : in std_logic;
            -- Clock
            set : in std_logic
            -- Operandos foram alterados
      );
end multiplier;
```

Na interface, o sinal de entrada **set** em 1 indica que os operandos **a** e **b** foram alterados, ou seja, sinaliza o início de uma nova operação de multiplicação.

Você pode dividir o seu código em vários arquivos *.vhd e utilizar componentes auxiliares, se julgar necessário. Não utilize bibliotecas (*libraries*) customizadas auxiliares. Ao final, reúna todos os arquivos *.vhd necessários para o projeto em uma única pasta chamada **multiplier**.

ENTREGA: Pasta **multiplier** com todos os arquivos ***.vhd**.

II.2. Instancie seu multiplicador para operandos de 5 bits e mapeie as entradas e saídas para a placa. Utilize os switches para as entradas **a** e **b**, um dos *push-buttons* para a entrada **set** e um dos sinais de clock interno da placa para **clk** (consulte o manual). Exiba o resultado nos visores de 7 segmentos em hexadecimal. **Esta parte será avaliada na demonstração.**

Bônus: Exiba o resultado nos visores de 7 segmentos, em decimal.

- ENTREGA -

Entregue um único arquivo comprimido em formato **ZIP** de nome **T_Lab08_RA.zip**, onde **RA** é o RA do aluno entregando e **T** é a turma, contendo:

- Arquivo **shift_register.vhd** do item **I.1**.
- Pasta **multiplier** contendo todos os arquivos ***.vhd** necessários para compilar o multiplicador do item **II.1**.