

# TRABALHO 2. CIRCUITOS SEQUENCIAIS

## 1 OBJETIVOS

Comprovar os conhecimentos relativos a circuitos combinacionais e sua implementação no Vivado utilizando VHDL.

## 2 GRUPOS

O Trabalho poderá ser feito em grupos de **até 3 pessoas**. Preencha a planilha cujo link foi disponibilizado na sala da disciplina do Aprender 3.

## 3 COMPOSIÇÃO DO TRABALHO

O Trabalho é compreendido pelos exercícios dispostos no Apêndice. O grupo deve enviar, à tarefa adequada no Aprender3, **um arquivo de identificação do grupo** (veja a Seção 3.1 para mais detalhes) e, **para cada exercício, os seguintes arquivos**:

1. **os arquivos gerados pelo Vivado** (veja a Seção 3.2 para mais detalhes); e
2. **um vídeo explicativo** (veja a Seção 3.3 para mais detalhes).

O grupo que enviar **arquivos sem identificação** – seja pela ausência de arquivo de identificação, seja pela ausência de identificação nome do arquivo (um exemplo de nome correto seria “Grupo 3.zip”); veja as Seções 3.1, 3.2 e 3.3 – **será penalizado com 1 ponto na nota final do trabalho**.

### 3.1 ARQUIVO DE IDENTIFICAÇÃO

O arquivo de identificação do grupo pode ser composto em qualquer editor de textos usualmente conhecido (extensões .txt, .doc, .docx ou .pdf, preferencialmente) e deve conter **o número do grupo e os dados dos integrantes** (nome, matrícula e e-mail).

### 3.2 ARQUIVOS DO VIVADO

Os arquivos de cada exercício gerados pelo Vivado devem ser enviados. Para isso, **para cada exercício**, no seu navegador de arquivos (por exemplo, Windows Explorer), vá à pasta criada pelo Vivado para o projeto e a **compacte** em alguns dos formatos usuais (.zip, .rar etc). Depois, **renomeie-o com a frase “GRUPO X” em que X é o número do seu grupo (por exemplo, “Grupo 3.zip”)**.

### 3.3 VÍDEO EXPLICATIVO

**Para cada exercício**, o grupo deve enviar um vídeo explicativo. Depois de gerar o vídeo, **compacte-o juntamente com o arquivo de identificação**.

Nesse vídeo o grupo deve:

1. **Introduzir o vídeo dizendo a Turma, o número do grupo e os nomes dos integrantes do grupo.** Pode ser útil mostrar na tela gravada a planilha disponibilizada indicando essas informações.
2. **Explicar todas as linhas de código implementadas** (qual a razão de cada linha) de **todos os arquivos**, ou seja, de todos o arquivos de extensão **.vhdL**.
3. **Explicar a implementação feita no laboratório remoto.** Para isso, basta **relacionar as entradas com as saídas** em cada período de tempo.

As explicações serão as principais fontes de avaliação, de modo que não basta apenas fazer o exercício e não explicar de modo satisfatório.

O Aprender tem um limite de 100Mb para upload de arquivos. Se o tamanho do vídeo ficar muito grande de modo a impedir o envio, converta-o para um formato ou resolução menor. Além disso, lembre-se de compactar o arquivo final. Em último caso, se não for possível por esse método, entrem em contato com o professor para que outro método seja disponibilizado.

## 4 ENTREGA

O Trabalho deve ser enviado **até às 23h59min do dia 23 de maio de 2021**. Cada dia de atraso acarretará decréscimo de 2,00 pontos da nota adquirida no Trabalho. Salvo indisponibilidade geral do Aprender3, **não se anularão os pontos suprimidos da nota final em razão de falha de conexão, mau funcionamento de computador ou motivo semelhante**. Portanto, **recomenda-se não deixar para enviar somente nos minutos finais: envie versões parciais**, ainda que incompletas, antes do prazo final, e as substitua quando novas versões forem sendo desenvolvidas.

## 5 NOTA

O Trabalho receberá nota entre 0 e 10 pontos.

## APÊNDICE

Para todos os exercícios deste Apêndice, **desenvolva um código em VHDL do circuito requisitado e o implemente na placa Basys 3 do laboratório remoto.**

### EXERCÍCIO 1. CRONÔMETRO

Projete um cronômetro (timer) capaz de funcionar de 0min: 00seg a 9min: 59seg (Figura 1). O circuito deve ter botões de iniciar, parar e reiniciar. Os números devem aparecer nos displays de 7 segmentos). Para isso você precisa fazer as seguintes etapas:

1. Criar um divisor de clock com saída de, aproximadamente, 1 segundo.
2. Criar um codificador para 7 segmentos.
3. Habilitar 3 anodos dos displays de 7 segmentos.
4. Multiplicar os três displays utilizados.

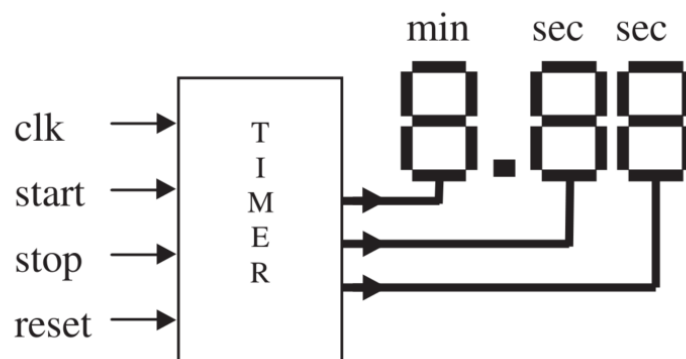


Figura 1. Esquema do cronômetro (timer) referente ao Exercício 1.

### EXERCÍCIO 2. FLIP-FLOP JK

Projeto um **flip-flop JK mestre-escravo** com **set e reset assíncronos** ativados em **nível alto** (Figura 2). As transições de estado devem ser acionadas com a **borda de subida** do clock.

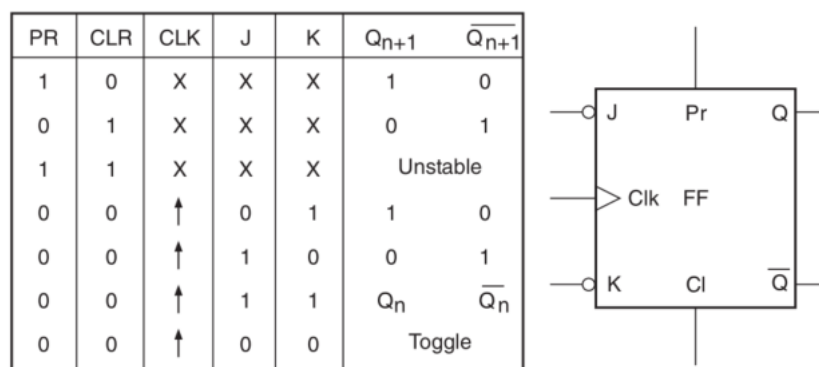


Figura 2. Esquema e tabela de verdade do flip-flop JK mestre-escravo relativo ao Exercício 2.

### EXERCÍCIO 3. REGISTRADOR DE DESLOCAMENTO UNIVERSAL

Projete um registrador de deslocamento universal de 8 bits de acordo com o esquema (Figura 3) e a descrição a seguir.

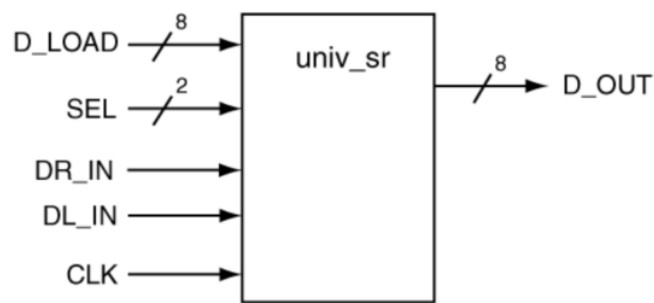


Figura 3. Esquema do registrador de deslocamento universal relativo ao Exercício 3.

Esse registrador universal tem mais de uma operação, que é selecionada pela entrada SEL (2 bits). Caso SEL seja:

- "00", o registrador não se altera. A saída D\_OUT fica inalterada.
- "01", acontece um carregamento paralelo. A saída D\_OUT recebe D\_LOAD.
- "10", acontece um deslocamento à direita. A saída D\_OUT sofre um deslocamento para a direita (cada bit recebe o valor do seu bit à esquerda) e o primeiro bit à esquerda recebe o valor de DL\_IN.
- "11", acontece um deslocamento à esquerda. A saída D\_OUT sofre um deslocamento para a esquerda (cada bit recebe o valor do seu bit à direita) e o primeiro bit à direita recebe o valor de DR\_IN.
- qualquer outro valor, acontece um reset. A saída D\_OUT é carregada com 0.

Cada operação deve ser realizada somente nas bordas de subida do clock. D\_LOAD e D\_OUT são portas de 8 bits, SEL de 2 bits e as demais de 1 bit.