

Curso de Graduação em Engenharia Eletrônica - Faculdade Gama - Universidade de Brasília

Prática de Eletrônica Digital 1. Código: FGA0071. Período Letivo: 2022/2 Turmas: 7, 8, 9, 10

Professor: Henrique Marra Taira Menegaz

e-mail: [henriquemenegaz@unb.br](mailto:henriquemenegaz@unb.br)

---

# EXPERIMENTO DE VHDL 3

## 1 REGRAS DE APRESENTAÇÃO

Os grupos podem apresentar o projeto do experimento de duas maneiras, uma de forma presencial, na sala de aula, e outra por vídeo.

Aqueles que optarem pela forma presencial deverão apresentar ao professor **os exercícios implementados na Basys 3 e explicar os códigos** escritos.

Por sua vez, os que optarem por vídeo, deverão **utilizar o laboratório remoto** e enviar **um arquivo de identificação do grupo** (veja a Seção 1.1 para mais detalhes) e, **para cada exercício, os seguintes arquivos** (o método de envio será apresentado na sala da disciplina do Aprender3):

1. **os arquivos gerados pelo Vivado** (veja a Seção 1.2 para mais detalhes); e
2. **um vídeo explicativo** (veja a Seção 1.3 para mais detalhes).

### 1.1 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).

### 1.2 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.).

### 1.3 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 as linhas de código implementadas.
3. Explicar o funcionamento na Basys 3.

Para tanto, grave a tela e o som do computador enquanto se explicam as linhas de código e se acessa o laboratório remoto para utilizar a Basys 3.

Por fim, tente fazer um vídeo de no máximo 8 min. Se o tamanho ficar muito grande de modo a dificultar o envio, converta-o para um formato ou resolução menor. Além disso, lembre-se de compactar o arquivo final do vídeo.

#### 1.4 INSTRUÇÕES PARA APRESENTAÇÃO

Ao apresentar o trabalho, **tanto de forma presencial, quanto por vídeo**, o grupo deve:

1. **Explicar todas as linhas de código implementadas** (qual a razão de cada linha) de **todos os arquivos**, ou seja, dos arquivos de descrição de circuitos (.vhdl) e dos arquivos de **restrição (.xdc)**; a **explicação será a principal fonte de avaliação**, de modo que não basta apenas fazer o exercício sem explicar de modo satisfatório;
2. **Explicar o funcionamento na Basys 3**. Para isso, basta **relacionar as entradas com as saídas** de acordo com o requisitado em cada exercício.

## 2 ENTREGA

Os grupos que optarem por apresentar presencialmente deverão apresentar o Trabalho ao professor em sala de aula, **nos horários das aulas, somente**. O prazo final de apresentação será a aula seguinte à aula deste experimento.

Os que optarem por apresentar por vídeo deverão enviar os arquivos requisitados no Aprender 3 até as 11h55min do sétimo dia contado da aula correspondente a este experimento. Para cada dia de atraso, será descontado 4 pontos da nota. **Não se admitirá entrega posterior a esse prazo 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.

## 3 NOTA

O Trabalho receberá nota entre 0 e 10 pontos.

## 4 PROJETO

Projete uma ULA programável de acordo com o esquema da Figura 1 e a Tabela 1. As variáveis **A** e **B** são entradas de dados de 4 bits, **S** = **S<sub>1</sub>S<sub>0</sub>** é a entrada de seleção, **F** = **F<sub>3</sub>F<sub>2</sub>F<sub>1</sub>F<sub>0</sub>** é a saída de dados, **over** é uma

saída de 1 bit que indica a ocorrência de um *overflow*, e **c\_out** é uma saída de *carry out*. As variáveis **A**, **B** e **F** devem ser consideradas como representadas em complemento de 2. Utilize chaves (*switches*) da Basys 3 para serem os bits de entrada e LEDs para os bits de saída (não há necessidade de usar os displays de 7 segmentos).

A operação realizada pela ULA será escolhida por **S** de acordo com a Tabela 1. Modos aritméticos serão realizados para **S** = 00 e **S** = 01, e modos lógicos para **S** = 10 e **S** = 11.

De fato, caso **S** = 00, a ULA deverá efetuar a soma (em complemento de 2) de **A** com **B** e, caso **S** = 01, a subtração (em complemento de 2) de **A** por **B**. Caso a operação de soma ou de subtração resulte em *overflow*, então **over** deverá ter o valor 1 e, em caso contrário, o valor 0. De modo semelhante, caso alguma dessas operações resulte em *carry out*, então **c\_out** deverá ter o valor 1 e, em caso contrário, o valor 0.

Por outro lado, caso **S** = 10, a ULA deverá efetuar a operação lógica **A and B bit-a-bit** e, caso **S** = 11, a operação lógica **A or B bit-a-bit**. Nesses dois casos, **over** e **c\_out** não fornecem informação alguma, logo poderão ter qualquer valor (*don't care*).

**Critérios de pontuação.** Este projeto será pontuado de acordo com os seguintes critérios:

- Código correto da soma (**S** = 00): 1,5 ponto.
- Código correto da subtração (**S** = 01): 1,5 ponto.
- Código correto da operação lógica *and* (**S** = 10): 1,5 ponto.
- Código correto da operação lógica *or* (**S** = 11): 1,5 ponto.
- Funcionamento na Basys 3: 4 pontos.

Tabela 1. Tabela Verdade a ser implementada no projeto.

Operação	S <sub>1</sub> S <sub>0</sub>	F <sub>3</sub> F <sub>2</sub> F <sub>1</sub> F <sub>0</sub>	over	c_out
SOMA(A,B)	00	A+B	overflow	carry out
SUBT(A,B)	01	A-B	overflow	carry out
AND(A,B)	10	A and B	X	X
OR(A,B)	11	A or B	X	X

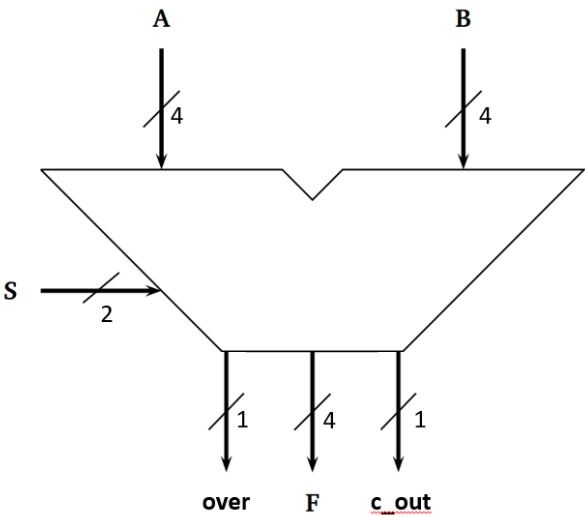


Figura 1. Esquemático da ULA a ser implementada no projeto.