PARTE B PRÁTICA

Arguição: _____
Relatório: _____

- **Exercícios**: faça-os com antecedência ANTES DA AULA.
- **Anotações**: devem ser feitas DURANTE A AULA.
- ☐ : passos das atividades – para não se perder, MARQUE-AS depois que completá-las

Nome: _____ n. USP: _____ Turma: ____

Colega de equipe (Nome / n. USP): _____

Bancada: _____ Data: _____ Hora de início: _____

Atividade 1 Inicialização do computador e teste da placa

☐ Ligue o computador e os DOIS MONITORES. Faça login e inicie a máquina virtual **pmr3303**.

Se sua máquina virtual estiver configurada corretamente, ela terá as seguintes unidades de rede disponíveis:
\\ts02-00\PMR3303 – unidade **S:**
\\vmware-host\Shared Folders\Public\PMR3303 – unidade **Y:**

☐ Copie do servidor a pasta indicada e todo o seu conteúdo (SOBREESCREVA caso já existam):
Origem: **S:\Exp4** Destino: **Y:\Turma*T***, onde *T* é o número da turma (1, 2, 3 ou 4)

☐ Para fazer a experiência, abra e siga com atenção o arquivo **Y:\Turma*T*\Exp4\3303_19Exp4_Roteiro.pdf**

Teste a placa Basys 3 com um circuito que já está gravado na memória *flash*.

☐ *Jumper* JP2 na posição 2-3 (USB): para alimentar a placa pelo conector USB
Jumper JP1 em 1-2 (QSPI): para configurar a FPGA com o conteúdo do memória flash

☐ Ligue o cabo USB e a chave ON/OFF da placa.
Teste as chaves, os leds e os botões. Ao final, desligue a chave ON/OFF da placa.

Atividade 2 Criação do projeto *xx*AE e do módulo *xx*AE3

Vamos criar projeto simples e passar por todas as etapas necessárias para implementar um circuito na placa Basys 3 usando o programa Vivado.

☐ No Vivado, clique em "Create New Project" e em seguida em "Next". Preencha os campos:
"Project location": **Y:\Turma*T*\Exp4*seugrupo* (***T*** é a turma e *seugrupo* é o nome de guerra do grupo)
"Project name": algo como ***xx*AE**, onde *xx* são as 2 primeiras letras do nome do grupo (ATENÇÃO: o primeiro ***x*** deve ser uma letra!)
Caixa "Create project subdirectory": deixe marcada e clique em "Next".

☐ Selecione "RTL Project" e marque a caixa "Do not specify sources at this time". Clique em "Next".

☐ Selecionar o modelo de FPGA presente na placa: "xc7**a35**tcpg236-1".
Clique em "Next" e em seguida em "Finish".

Anotação 2a Anote o nome do seu diretório de trabalho: _____

Vamos agora criar o módulo que implementa *Aritmetic Extender* de 3 bits, AE3.

☐ No menu "Flow Navigator" (esquerda da tela), submenu "Project Manager", clique em "Add Sorces".
Clique em "Add or create design sources" e clique em "Next".
Clique em "Create File".
No campo "File name:", digite ***xxAE3***, onde *xx* são as 2 primeiras letras do nome do grupo. Clique "OK".

☐ Na janela "Add Sources", clique em "Finish". Isso abrirá a janela "Define Module". Configure os sinais de E/S: **M:** input, 1 bit ("Bus" em branco); **s:** input**,** Bus [1:0]; **b**: input**,** Bus **[2:0]**; **y:** output, Bus [2:0].
Clique em "OK."

☐ O módulo *xx*AE3 deve ter sido incluído na hierarquia do projeto:
(quadro "Project Manager", subquadro "Sources", pasta "Design Sources")
Clique duas vezes no módulo *xx*AE3 para abri-lo para edição.

☐ No topo do arquivo, há várias linhas de comentário destinadas a documentar o módulo. Em prol dos bons hábitos de documentação, preencha algumas delas.

```
// Engineers:  nomes dos membros do grupoZ
// Module Name:  xxAE3 (já preenchido)
// Project Name:  xxULA (onde xx... você sabe, não?)
// Description: algo como "aritmetic extender de 3 bits"
// Dependencies: nenhuma (isto é, este módulo não depende de sub-módulos)
```

A Figura 4.14 mostra resumidamente o conteúdo do arquivo criado

```
module xxAE3(
    input M,
    input [1:0] s,
    input [2:0] b,
    output [2:0] y
    );

endmodule
```

Figura 4.14 Esqueleto do módulo *xx*AE3

☐ Se sua versão do Exercício 1 já estiver feita, revise-a juntamente com a equipe e aponte possíveis erros. Caso contrário, faça o exercício ANTES de digitar o módulo no computador.

Exercício 1 Complete o módulo *xx*AE3 com a codificação em Verilog do circuito do extensor aritmético de 3 bits AE3. Você pode usar declarações ***assign*** ou um bloco ***for***. Não é necessário escrever o módulo inteiro, o que falta para completar o esqueleto. Lembre-se: use parêntesis para melhorar a legibilidade das expressões lógicas!)

☐ Complete o módulo: digite a descrição do circuito em Verilog feita o exercício do pré-relatório e salve.

Anotação 2b Descreva os erros encontrados em seu código Verilog, se houver, e as correções feitas. Indique também Indique de qual membro da equipe foi o módulo implementado.

Atividade 3 Simulação do módulo *xx*AE3

Vamos testar o módulo *xx*AE3 por meio de uma simulação funcional (ou *behavioral*). Para isso, é necessário criar um módulo que contenha a instanciação de *xx*AE3 e gere sinais de estímulo para a simulação.

- ☐ No menu "Flow Navigator" (esquerda da tela), submenu "Project Manager": clique em "Add Sources".
  Selecione "**Add or create simulation sources**", clique em "Next" e em seguida em "Create File".
  No campo "File name:", digite algo como ***xx*AE3_sim**, substituindo xx pelas iniciais do seu grupo.
  Clique em "OK" e em seguida em "Finish".
  Na janela "Define Module", NÃO defina entradas ou saídas: elas serão geradas na simulação.
  Clique "OK" e em seguida em "Yes".
- ☐ ***xx*AE3_sim.v** foi incluído na hierarquia do projeto ("Project Manager", subquadro "Sources", pasta "Simulation Sources/sim_1". Clique nele com o botão direito e selecione "Set as top" (ficará no topo da hierarquia).
- ☐ Clique duas vezes sobre ele para abri-lo no editor.
- ☐ Preencha os comentários "Engineer", "Project Name", "Description" (algo como "simulação do módulo xxAE3.v") e "Dependencies" (neste caso, ***xx*AE3.v**).

A Figura 4.15 mostra um modelo que você pode usar para fazer o seu módulo de simulação.

```
module xxAE3_sim( );

    // Declare nets e variáveis aqui;
    // ...

    xxAE3 UUT ( M, s, b, y );  // Instanciação do módulo a simular

    initial begin
        $timeformat(-9, 0, "ns", 3);  // formato para imprimir $time
        // Descreva como gerar os sinais M, s, b
        // ...
        $finish;
    end // initial

endmodule // xxAE3_sim
```

Figura 4.15 Modelo do módulo de simulação *xx*AE3_sim.

- ☐ Se sua versão do módulo já estiver feita, revise-a juntamente com a equipe e aponte possíveis erros. Caso contrário, faça o exercício ANTES de digitar o módulo no computador.

Exercício 2 Escreva em Verilog o módulo de simulação xxAE3_sim( ) completo. Se preferir, faça em uma folha à parte e anexe. Gere as combinações possíveis de M e s[1:0], e para cada uma coloque em b[2:0] três bits iguais (3'b000 ou 3'b111). Imprima o valor da saída y[2:0] para cada combinação de entradas, usando o comando:

```
$write( "Tempo=%t: M=%b, s=%2b, b=%b  =>  y=%b\n", $time, M, s, b, y);
```

Vamos agora configurar a simulação:

☐ No menu "Flow Navigator", submenu "Simulation", clique em "Simulation Setting".
Na janela "Project Settings", com "Simulation" selecionado na barra lateral, clique na aba "Simulation".
Verifique o valor em frente ao parâmetro "xsim.simulate.runtime". deve ser pelo menos **160ns**.
Na aba "Advanced", verifique se a opção "Include all design sources..." está marcada. Clique "OK".

Hora de rodar a simulação e verificar os resultados:

☐ Em "Flow Navigator", submenu "Simulation": clique em "Run Simulation".
Clique em "Run Behavioral Simulation" e aguarde o simulador trabalhar.
Se a simulação falhar, analise as mensagens de erro na aba "Messages". Corrija, salve e tente simular de novo.
Se tudo correr bem, o simulador vai imprimir as saídas do comando $write na janela ***"Tcl Console"***

Anotação 1b Descreva os erros encontrados em seu código Verilog, se houver, e as correções feitas. Indique também Indique de qual membro da equipe foi o módulo implementado

Exercício 3 Faça o mapa de Karnaugh da saída *yi* do extensor aritmético AE1. Para sua comodidade, reproduzimos a expressão aqui: $y_i = \overline{M}.\left(\overline{b_i}.s_1 \; + \; b_i.\overline{s_0}\right)$.

Anotação 3a Compare os resultados da simulação (impressos na janela *Tcl Console*) com o previsto no pré-relatório. Lembre-se que o módulo simulado xxAE3 é composto por três circuitos iguais ao do bloco AE1. COMENTE.

O simulador também abre uma nova janela contendo as ***formas de onda*** da simulação

☐ Desacople a tela gráfica do simulador: clique no ícone ‘ ’ na barra superior e amplie a janela.
Clique com o botão da direita sobre as formas de onda e clique em “Full view”.

Anotação 3b Mostre os resultados e suas conclusões para o professor. Anote o horário atual: _____.

Atividade 4 Teste do módulo *xx*AE3 na placa Basys 3

Vamos testar o módulo *xx*AE3 que você criou, ligando-o a chaves e leds da placa Basys 3 para vê-lo funcionando. Vamos criar um módulo no topo da hierarquia do projeto, no qual instanciaremos o módulo *xx*AE3.

☐ No menu “Flow Navigator”, submenu “Project Manager”, clique em “Add Sources”.
Crie um novo módulo de nome ***xx*AE3_top**, substituindo ***xx*** pelas iniciais do seu grupo. Resumidamente: “Add or create design sources” / “Next” / “Create File” / “File name:” = ***xx*AE3_top** / “OK” / “Finish”.

☐ Na janela “Define Module”, defina os seguintes sinais de entrada (“Direction” = “input”):
sw2 / **sw1** / **sw0 / sw15**.
Declare o sinal de saída (“Direction” = “output”): **ld15**. Clique em “OK”.

☐ No subquadro “Sources” do quadro “Project Manager”, o módulo *xx*AE3_top.v deve ter sido incluído na pasta “Design Sources”. Clique duas vezes sobre ele para abri-lo no editor.
Documente o módulo: preencha os comentários “Engineer”, “Project Name”, “Description” (algo como “módulo de topo para teste do xxAE3”) e “Dependencies” (neste caso, **xxAE3**).

☐ Complete o módulo seguindo a listagem mostrada na Figura 4.11. Neste caso, o módulo conterá apenas a instanciação do módulo *xx*AE3, com as chaves conectadas às entradas e o led à saída.

Exercício 4 Encontre no manual da placa Basys 3 os identificadores dos pinos aos quais estão conectadas as chaves e o led que usaremos nesta Atividade. Faça o diagrama lógico do circuito de teste do módulo *xx*AE3. Represente o módulo *xx*AE3 por um símbolo (caixa-preta). Represente as chaves por retângulos e o led por um círculo, como feito nas experiências anteriores. Anote ao lado de cada chave e do led o pino da FPGA ao qual está conectado.

Exercício 5 Para exemplificar a forma que Vivado usa para associar um sinal de entrada ou de saída do circuito a um pino da FPGA, escreva os dois pares de linhas que precisam ser incluídas no arquivo de *constraints* para definir:: a) a chave SW15; b) o led LD15

Agora é necessário declarar em que pinos da FPGA deverão ser conectadas as entradas e as saídas do circuito. Isso é feito por meio do arquivo de restrições (*constraints*).

☐ No menu "Flow Navigator", submenu "Project Manager", clique em "Add Sources".
Selecione "Add or create constraints" (1º item), clique em "Next" e em seguida em "Create File".
No campo "File name:", digite algo como ***xx*AE**, substituindo *xx* pelas iniciais do seu grupo.
Clique em "OK" e em seguida em "Finish".

☐ Verifique no quadro "Project Manager", subquadro "Sources": o arquivo *xx*AE.xdc deve ter sido incluído na pasta "Constraints/constrs_1". Clique duas vezes sobre ele para abri-lo no editor.

Lembre-se: cada pino de E/S é definido por um par de linhas. Veja o exemplo descrito na seção 4.4.4.

☐ Copie as duas linhas "`set_property ...`" do exemplo no arquivo *xx*AE.xdc.

☐ Inclua a definição dos pinos conectados às demais chaves: sw 1, sw0 e sw15. Repita para o led ld15.
Use o (velho e bom) CTRL-C/CTRL-V para não ter que redigitar tudo várias vezes. Ao final, salve o arquivo.

Pode-se gerar um diagrama lógico (um tanto limitado) correspondente ao código Verilog digitado, que permite verificar se o circuito gerado é o esperado:

☐ No menu "Flow Navigation", submenu "RTL Analysis", clique em "Elaborate Design" e em "Schematic".

Será aberta a janela com o diagrama lógico do circuito. O diagrama é hierárquico:

- No primeiro nível, temos os pinos de I/O e o símbolo do circuito *xx*AE3. Clicando sobre um *pad*, abre-se a janela "I/O Port Properties" (à esquerda do diagrama), que permite conferir a configuração do *pad*.
- Clicando-se sobre o "+" do símbolo, pode-se ver o diagrama lógico do circuito.

Vamos agora configurar alguns parâmetros de síntese e implementação. A ideia é desativar algumas opções de otimização do circuito para que o programa rode mais rápido (em compensação, o circuito final provavelmente será mais lento, ocupará mais elementos da FPGA e consumirá mais energia...).

☐ No menu "Flow Navigation", submenu "Synthesis", clique em "Synthesis Settings".
Clique no campo "Strategy" e selecione a opção "Flow_RuntimeOptimized".

☐ Na lateral esquerda, clique em "Implementation" (triângulo verde).
Clique no campo "Strategy" e selecione a opção "Flow_Quick" (última linha).

☐ Clique no menu "Opt_Design" e DESMARQUE a opção "is_enabled" (para desabilitar otimizações).
Clique em "OK".

Estamos prontos para gerar o arquivo *bitstream* de configuração da FPGA

☐ No menu "Flow Navigation", submenu "Program and Debug", clique em "Generate Bitstream".
Clique em "Yes" e... relaxe. Isso deve levar em torno de 1 min.

O programa parece meio morto, mas você pode acompanhar alguma ação na janela "Project Summary" e na aba "Log" no parte inferior da janela.

Ao final, ignore três *warnings* a respeito de falta *time constraints*, *clock* e configurações de voltagem.

☐ Se não houver erros, deve abrir-se a janela "Bitstream Generation Completed" para selecionar o próximo passo (mas isso pode ser desabilitado). Como faremos mais a frente, por hora clique em "Cancel".

Hora de conectar a placa Basys 3 e configurar a FPGA:

☐ Com a placa **desligada** (sw16), passe o *jumper* JP1 para a posição 2-3 (JTAG). Em seguida, **ligue** a placa.

☐ A máquina virtual PMR3303 está configurada para conectar dispositivos USB novos, e o Vivado deve reconhecer a placa. Se isso não acontecer:
Na aba superior do VMWARE PLAYER (mova o ponteiro para o topo da tela), clique em "Player" | "Removable Devices" | "Future Devices..." | "Connect"

☐ No submenu "Program and Debug", clique em "Open Target" (abaixo do item "Hardware Manager") e em seguida clique em "Auto Connect". Se aparecer uma mensagem de erro "open-hw", veja o item anterior.

☐ No submenu "Program and Debug", clique em "Program Device" e em seguida em "xc7a35t".
Na janela "Program Device" o campo "Bitstream file" já deve estar preenchido com "...xxAE_top.bit".
Clique em "Program". O led ld19 (Done) deve acender, indicando que a FPGA foi configurada.

☐ A entrada M está ligada à chave sw2, e s1 e s0 à sw1 e sw0. A entrada b[0] é acionada pela chave sw15.
Verifique se a saída y[0] (led ld15), reproduz a função projetada no pré-relatório.
Lembre-se que
yi = 0 para M = 1 (modo lógico).

Anotação 4a Anote a hora:_____. Apresente o circuito funcionado para o professor.

☐ Desligue a chave ON/OFF (sw16) da placa Basys 3.

☐ Feche o projeto *xx*AE no Vivado: no menu superior, clique em "File" | "Close Project", e em seguida em "OK".

Atividade 5 ULA de 3 bits

O projeto da ULA de 3 bits está parcialmente feito em um arquivo compactado copiado no início da aula.

☐ Abra no "Explorer" do Windows a pasta **Y:\Turma*T*\Exp4** (*T*: número da turma).
Descomprima o arquivo **3303_E4.zip**: clique com o botão direito, clique em "7-zip" e clique em "Extrair aqui".
Caso apareça uma janela de confirmação, clique em "Sim para Todos" para substituir os arquivos existentes.

Vamos abrir o projeto e copiá-lo com outro nome para preservar o original.

☐ Na tela de abertura do Vivado, clique em "Open Project"
Localize a pasta Y:/Turma***T***/Exp4/3303_E4 e dê duplo clique no arquivo **3303_E4.xpr**. Aguarde a carga do projeto.

☐ Clique em "File" na barra de menus superior. Clique em "Save Project as..." e complete os seguintes campos:
"Project location": Y:/Turma***T***/Exp4/***seugrupo*** , onde *seugrupo* é o nome de guerra do grupo.
"Project name": ***xx*ULA** onde *xx* são as 2 primeiras letras do nome do grupo.
A opção "Create project subdirectory" deve estar marcada. Clique em "Ok".
A cópia do projeto será aberto no lugar do original.

Vamos precisar também o arquivo xxAE3 criado nas atividades anteriores.

☐ No menu "Flow Navigator", sub-menu "Project Manager": clique em "Add Sources".
Selecione "Add or create design sources" e clique em "Next". Em seguida, clique em "Add Files".
Em Y:/Turma*T*/Exp4/*seugrupo*/*xx*AE/*xx*AE.srcs/source_1/new , clique no arquivo ***xx*AE3.v** e em "OK".
Em seguida, clique em "Finish".

O projeto usa o módulo *xx*AE3 que você implementou. É preciso editar o arquivo ULA3.v para trocar o "xx" pelas iniciais do seu grupo.

☐ No quadro "Projec Manager", subquadro "Sources", pasta "Design Sources": expanda a hierarquia do módulo "ULA_top" (clique duas vezes sobre ele).
Clique duas vezes sobre o módulo ULA3.v para abri-lo no editor. A Figura 4.12 mostra o seu conteúdo original.
Localize a instanciação do módulo `xxAE3` e substitua "*xx*" pelas iniciais do seu grupo. Salve o arquivo.

Como o Verilog é capaz de gerar somadores binários com número variável de bits, usamos essa facilidade para somar os *arrays* de três bits x[2:0] e y[2:0] e o vem-um c_0E. O resultado de quatro bits é atribuído ao array {c3, f}, composto pela concatenação do bit c3 com o array f[2:0].

A Figura 4.13 mostra o módulo de topo, que descreve como os sinais da ULA serão conectados a chaves e leds da placa Basys 3. Confira o conteúdo do arquivo clicando duas vezes sobre o módulo ULA_top.v (quadro "Project Manager", subquadro "Sources", pasta "Design Sources").

O módulo declara as 16 chaves do painel, mas não usa todas, pulando algumas – as não usadas ficarão em aberto. O mesmo acontece com os 16 leds. Isso porque as chaves e leds foram declarados na forma de *arrays* (ao invés de pontos individuais, como fizemos para testar o módulo *xx*AE3) e o Verilog não permite declarar faixas não contínuas de índices.

Exercício 6 Faça o diagrama lógico do circuito de teste do módulo ULA3 descrito nesta Atividade (represente este módulo por seu símbolo). Represente as chaves por retângulos e o led por um círculo, como feito nas experiências anteriores (desta vez, não é preciso anotar os identificadores dos pinos da FPGA). Não inclua as chaves em aberto e nem os leds aterrados (em zero).

Você pode conferir o diagrama lógico do seu circuito:

☐ No menu "Flow Navigation", submenu "RTL Analysis", clique em "Elaborate Design" e em "Schematic".
No nível mais alto, confira as conexões de I/O com as chaves e leds.
Abra o bloco ULA3: ele deve reproduzir a estrutura da ULA que projetamos, com um somador, LE (*Logic Extender*), AE (*Arithmetic Extender*) e CE (*Carry-in Extender*).
Ao final, feche a janela "Schematic".

Atividade 6 Teste da ULA

Exercício 7 Para simular o funcionamento da ULA, faça manualmente as operações aritméticas descritas na tabela abaixo para M=0. Nas colunas a[2:0] e b[2:0] escreva os valores de A e B em complemento de 2. Na coluna "decimal", escreva o resultado f[2:0] em **decimal com sinal** (ex: –3). Por fim, complete a tabela para M=1

	decimal	decimal				Modo Aritmético (M = 0)				Modo Lógico (M = 1)		
s1 s0	A	B	c0	a[2:0]	b[2:0]	operação	c3	f[2:0]	decimal	operação	c3	f[2:0]
0 0	–3	–1	1									
0 1	2	0	1									
1 0	1	2	0									
1 1	1	1	1									

É chegada a hora da verdade:

- ☐ Sintetize o projeto: no menu "Flow Navigator", submenu "Synthesis", clique em "Run Synthesis".
  Ignore os *warnings* a respeito dos leds forçados a valores constantes (0 ou 1) e chaves em aberto.
  Mas... se aparecerem mensagens diferentes, verifique se há erros, corrija e sintetize de novo
- ☐ Implemente: menu "Flow Navigator", submenu "Implementation", clique em "Run Implementation".
  Ignores os *warnings* a respeito de *time constraints*, *clock* e voltagem.
  Mensagens diferentes destas vão requerer sua atenção. Verifique se há erros e corrija. Nesse caso, será necessário sintetizar de novo: volte ao passo anterior.
- ☐ Gere o *bitstream*: menu "Flow Navigator", submenu "Program and Debug", "Generate Bitstream"
- ☐ Ligue a chave ON/OFF da placa.
  Configure para a FPGA: clique em "Open Target" e em seguida em "Auto Connect".
  Clique em "Program Device" e em seguida em "xc7a35t".
  Na janela "Program Device", o campo "Bitstream file" deve terminar com "...**ULA_top.bit**" (se ainda constar o arquivo *xx*AE_top.bit, corrija). Clique em "Program".
  O led ld19 (Done) deve acender, indicando que a programação da FPGA teve sucesso!
- ☐ Teste o funcionamento da ULA.

Anotação 6a Teste a ULA com os casos previstos no exercício do pré-relatório. Compare com os resultados do pré-relatório e COMENTE. Preste especial atenção no bit c3.

Anotação 6b Anote a hora atual:_____. Apresente o circuito funcionando e suas conclusões para o professor.

(Atividade 8 - Opcional: em anexo)

Atividade 7 Finalização

Se quiser, você pode copiar os arquivos dos projetos em um pendrive. Copie as pastas xxAE e xxULA.

- ☐ Encerre o programa Vivado.
- ☐ Encerre (*shutdown*) a máquina virtual PMR3303.
- ☐ Aguarde o PLAYER terminar. DESLIGUE O COMPUTADOR E OS MONITORES.
  Deixe a bancada em ordem. Falhas nesse procedimento serão penalizadas.

Check list: verifique cada um dos itens abaixo.

- ☐ *Check list*: verifique cada um dos itens abaixo.
  - ☐ **Equipamentos** Verifique se estão todos desligados. Em especial, certifique-se que o computador E OS MONITORES estejam desligados (NÃO os deixe em STAND-BY)
  - ☐ **Placa Basys 3** Deve estar protegida com a espuma anti-estática e guardada na caixinha.
  - ☐ **Cabo USB-micro** Guarde na caixa de componentes de PMR3303.
  - ☐ **Multímetro** Desligado e com os cabos das pontas de prova arrumados. Deixe-o no tampo inferior da bancada, para que possamos conferir facilmente se está desligado.
  - ☐ **Outros componentes e cabinhos** Verifique se ficou alguma coisa na bancada ou no chão.
  - ☐ **Empréstimos** Se usou alguma coisa de outra bancada, devolva e liste: _____
  - ☐ **Defeitos** Se encontrou algum defeito, preencha a Comunicação de Defeito e liste:
    _____
  - ☐ **Limpeza** Limpe a bancada.