

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:\Exp5** Destino: **Y:\Turma T**, onde *T* é o número da turma (1, 2, 3 ou 4)
- Abra e siga o arquivo **Y:\Turma T\Exp5\3303_19Exp5_Roteiro.pdf**
- Descomprima o arquivo **Y:\Turma T\Exp5\3303_E5.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.
 (Por ser pasta de rede, o Explorer não atualiza a janela automaticamente – dê um “refresh” ou reabra a pasta).

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

- Jumpers* JP2 na posição 2-3 (USB): para alimentar a placa pelo conector USB
Jumpers 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 xxFF

- No Vivado, clique em “Create New Project” e em seguida em “Next”. Preencha os campos:
 “Project location”: **Y:/Turma T/Exp5/seugrupo** (*T* é a turma e *seugrupo* é o nome de guerra do grupo)
 “Project name”: algo como **xxFF**, 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: “xc7a35tcbg236-1”.
 Clique em “Next” e em seguida em “Finish”.

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

Atividade 3 Criação do módulo FF_DE

Vamos agora criar o módulo que implementa o flip-flop D descrito na seção 5.1.1 .

- No menu “Flow Navigator” (esquerda da tela), submenu “Project Manager”, clique em “Add Sources”.
 Clique em “Add or create design sources” e clique em “Next”.
 Clique em “Create File”.
 No campo “File name:”, digite **FF_DE**. Clique “OK”.

- Na janela “Add Sources”, clique em “Finish”. Isso abrirá a janela “Define Module”. Configure os sinais de E/S:
Inputs: C, D, En e Clr, todos de 1 bit (deixe a caixa “Bus” desmarcada).
Outputs: Q de 1 bit também.
 Clique em “OK.”
- O módulo FF_DE deve ter sido incluído na hierarquia do projeto:
 Para editá-lo, clique duas vezes sobre ele (em “Project Manager”, subquadro “Sources”, pasta “Desing Sources”). Preencha os comentários “Engineer”, “Project Name”, “Description” (algo como “Flip-flop D com enable”) e “Dependencies” (neste caso, nenhum)
- Complete o módulo com a descrição procedural do flip-flop D, conforme mostra a Figura 5.2, e salve.

Atividade 4 Criação do módulo xxCB4

Vamos agora criar o módulo que implementa um contador binário de 4 bits usando instanciações do flip-flop D criado na atividade anterior.

- No menu “Flow Navigator”, submenu “Project Manager”, clique em “Add Sources”.
 Crie um novo módulo de nome **xxCB4**, substituindo **xx** pelas iniciais do seu grupo. Resumidamente: “Add or create design sources” / “Next” / “Create File” / “File name:” = **xxCB4** / “OK” / “Finish”.
- Na janela “Define Module”, defina os seguintes sinais de E/S:
Inputs: C, En e Clr, todos de 1 bit (deixe a caixa “Bus” desmarcada).
Outputs: Q de 4 bit (Bus [3:0]) e **RCO** de 1 bit.
 Clique em “OK.”
- Para editar o **xxCB4.v**, clique duas vezes sobre ele na hierarquia do projeto (em “Project Manager”, subquadro “Sources”, pasta “Desing Sources”) e preencha os comentários “Engineer”, “Project Name”, “Description” (algo como “Contador binário de 4 bits”) e “Dependencies” (neste caso, **FF_DE**). A figura abaixo mostra resumidamente o arquivo criado

```

module xxCB4(
    input C,
    input En,
    input Clr,
    output [3:0] Q,
    output RCO
);
endmodule
    
```

Exercício 1 Faça o diagrama lógico de um contador síncrono de quatro bits usando 4 flip-flops D conforme o módulo xxCB4 descrito nesta atividade. NÃO USE dois contadores CB2 em cascata! Não esqueça a saída RCO.



Exercício 2 Complete o módulo xxCB4 com a codificação em Verilog do contador de 4 bits projetado no exercício anterior, usando instanciações do módulo FF_DE e declarações **assign** para gerar os sinais de interface entre os flip-flops. Não é necessário escrever o módulo inteiro, o que falta para completar o esqueleto acima. Se preferir, use uma folha anexa.

- 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.
O simulador vai imprimir as saídas dos comandos \$write e \$strobe na janela **“Tcl Console”**
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”.
- Por default, o inteiro k é impresso com 32 bits e não cabe dentro da forma de onda. Mude o formato:
Clique com o botão direito sobre o sinal k, selecione “Radix” e em seguida clique em “unsigned decimal”.
- Por default, o barramento Q[3:0] é mostrado de forma condensada. Expanda os bits clicando no sinal “+” à esquerda dele.
- Verifique se o resultado da simulação está correto. Caso contrário, revise o módulo do seu contador xxCB4.

Anotação 5a Mostre os resultados e suas conclusões para o professor. Anote o horário atual: _____.

Atividade 6 Teste do contador xxCB4 na placa Basys3.

Vamos testar o módulo xxCB4 que você criou, ligando-o a chaves e leds da placa Basys 3 para vê-lo funcionando.

O módulo de topo TestCB4_top.v mostrado na Figura 5.16 já está pronto. Inclua-o no seu projeto, e aproveite para incluir também outros módulos que usaremos na próxima atividade:

- 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”.
Vá para a pasta Y:/TurmaT/Exp5/3303_E5/, segure a tecla **Ctrl** do teclado pressionada e clique nos arquivos: **CB2.v**, **CB4_2CB2.v** e **TestCB4_top.v**.
Clique e em “OK”.
MARQUE a opção **“Copy sources into project”**. Em seguida, clique em “Finish”.
- Abra o arquivo clicando duas vezes em **TestCB4_top.v** (“Project Manager”, subquadro “Sources”, pasta “Design Sources”. Clique nele com o botão direito e selecione “Set as top” (ficará no topo da hierarquia).
- Localize a instanciação do módulo xxCB4 e substitua “xx” pelas iniciais do seu grupo. Salve o arquivo.
Precisamos incluir também o arquivo de *constraints*
- No menu “Flow Navigator”, sub-menu “Project Manager”: clique em “Add Sources”.
Selecione “Add or create **constraints**” e clique em “Next”. Em seguida, clique em “Add Files”.
Na pasta Y:/TurmaT/Exp5/3303_E5/, clique no arquivo **CB4.xdc**. Clique e em “OK”.
MARQUE a opção **“Copy sources into project”**. Em seguida, clique em “Finish”.

O arquivo de *constraints* CB4.xdc contém a configuração de vários dispositivos da placa Basys 3, mas estão comentados. No arquivo de *constraints*, linhas com comentários começam com ‘#’.

Para comentar ou descomentar linhas: selecione-as e clique no botão “//” na lateral do editor.

- Descomente a linha que relaxa a verificação de *loops* temporais (como o que ocorre no latch SR):
set_property SEVERITY {Warning} [get_drc_checks LUTLP-1]
- Descomente os **pares** de linha que configuram os dispositivos de E/S que vamos usar:
chave **sw[15]**, todos os leds (**ld[0] a ld[15]**), botões **bntL**, **bntD** e **bntR**. Salve o arquivo

Configure 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”.

- Gere 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.
Você pode acompanhar alguma ação na janela “Project Summary” e na aba “Log” no parte inferior da janela.
Ao final, ignore *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”.
Conectar a placa Basys 3 e configure 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 “...TestCB4_top.bit”.
Clique em “Program”. O led ld19 (Done) deve acender, indicando que a FPGA foi configurada.

Exercício 3 Faça o diagrama lógico do circuito de teste do contador TestCB4_top (listagem da Figura 5.16). Represente o *latch* SR positivo pelo seu símbolo usual. Represente o módulo xxCB4 por seu símbolo. Represente as chaves por retângulos e o led por um círculo, como feito nas experiências anteriores. Não é necessário incluir os identificadores dos pinos da FPGA.



Teste o circuito:

- Mantenha a chave sw15 (En) em 1, pressione alternadamente os botões btnL e btnR para gerar o sinal de *clock* e acompanhe as saídas nos leds. Coloque a chave sw15 em 0 algumas vezes antes de fazer o *clock* subir, e verifique se o contador fica desabilitado corretamente. O botão btnD zera o contador.
- Teste a seguinte situação: com En = 1 (chave sw15), incremente o contador até a contagem atingir o valor máximo $Q[3:0] = 1111$. RCO (led led4) deve acender também. Mude então En para 0 e alterne o sinal de *clock* C (botões btnL e btnR) algumas vezes. A contagem não deve se alterar. Volte En para 1 e incremente o contador

Anotação 6a Mostre o circuito funcionando para o professor. Anote o horário atual: _____.

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

Atividade 7 Cascateamento de contadores

Vamos analisar o funcionamento do contador de quatro bits CB4_2CB2, construído com dois contadores de 2 bits em cascata, por meio de uma simulação. Os módulos necessários para isso já foram incluídos no projeto.

- Abra o arquivo de *testbench* CB4_sim.v. Substitua a instanciação do contador `xxCB4` (o seu) pelo `CB4_2CB2`.
- Modifique o arquivo para que reproduza as formas de onda previstas no pré-relatório, seguindo a listagem da Figura 5.13. Em resumo:
 Modifique o nome do módulo (opcional).
 Modifique o comando `$strobe`.
 Modifique a edição do sinal `En` (dentro do bloco *always*).
 Salve o arquivo
- Execute a simulação: Em “Flow Navigator”, submenu “Simulation”: clique em “Run Simulation”.
 Clique em “Run Behavioral Simulation” e aguarde o simulador trabalhar.
- Amplie a janela com as formas de onda e verifique se os resultados da simulação são os esperados. LEMBRE-SE: o valor das entradas que valem são os de ANTES das bordas de *clock*!
Para incluir o sinal EnU3: veja as instruções no Roteiro (3303_19Exp5_Roteiro.pdf)

Anotação 7a Anote as formas de onda da simulação no quadro da próxima página (**a mão livre!**) entre 0 e 22 ms. Desenhe os sinais `C`, `k` (contador da simulação), `Clr`, `En`, `Q[1:0]`, `EnU3`, `Q[3:2]` e `RCO`. Represente o sinal de *clock* por um trem de impulsos e `k` por barramento condensado em decimal. Represente também `Q[3:2]` e `Q[1:0]` por barramentos condensados (em binário).

Anotação 1b Anote a hora atual: _____. Apresente o resultado da simulação e suas conclusões para o professor.

- Feche o projeto `xxFF` no Vivado: no menu superior, clique em “File” | “Close Project”, e em seguida em “OK”.

Atividade 8 Acionador `xxDsDrv` e teste com cronômetro de 4 dígitos

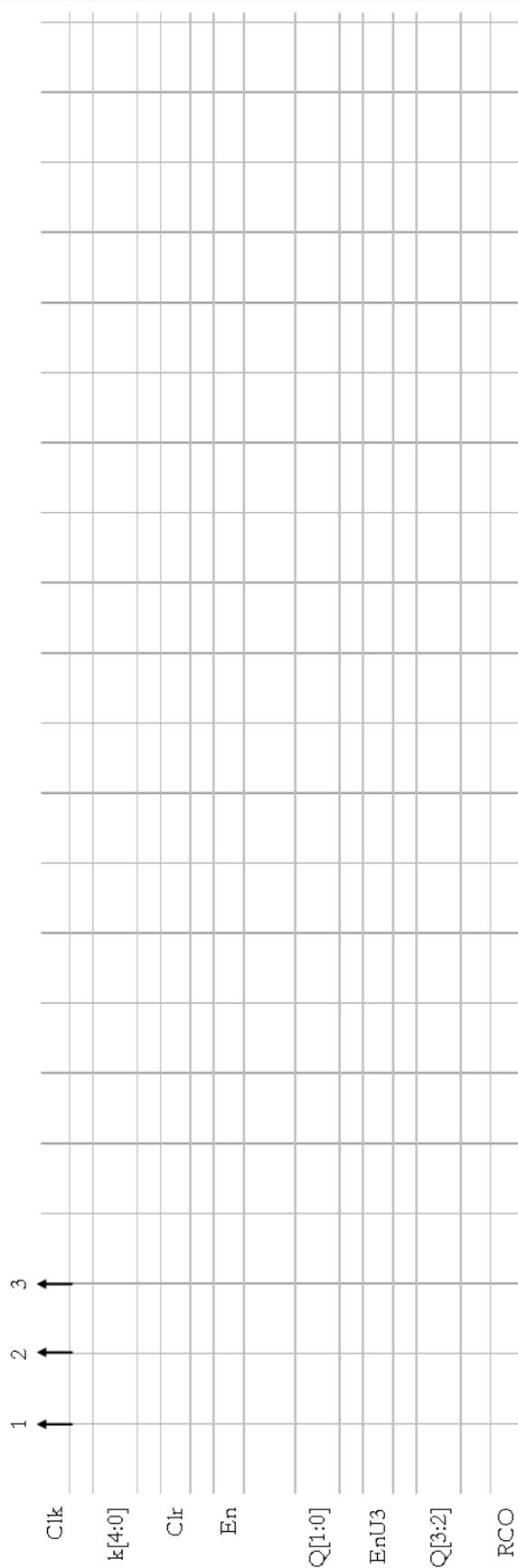
Vamos implementar o acionador de display testar usando o módulo do cronômetro `Cron_top` mostrado na Figura 5.22. Para isso, vamos usar o projeto instalado no seu computador no início da experiência.

- Na tela de abertura do Vivado, clique em “Open Project”
 Localize a pasta `Y:/Turma T/Exp5/3303_E5` e dê duplo clique no arquivo **3303_E5.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/Exp5/seugrupo`, onde *seugrupo* é o nome de guerra do grupo.
 “Project name”: **xxCron** 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 aberta no lugar do original.
- Crie e complete o módulo `xxDsDrv` :
 No menu “Flow Navigator”, submenu “Project Manager”, clique em “Add Sources”. Em seguida:
 “Add or create design sources” / “Next” / “Create File” / “File name:” = **xxDsDrv** / “OK” / “Finish”
- Na janela “Define Module”, configure os sinais de E/S:
Inputs: `C` (1bit); `Da`, `Db`, `Dc` e `Dd` (Bus [3:0]); e `En` (1bit).
Outputs: `An` (Bus [3:0]); `Seg` (Bus [0:6]) – ATENÇÃO ao definir o bus `Seg`!.
 Clique em “OK.”
- Abra o módulo `xxDsDrv` no editor. A figura abaixo mostra resumidamente o arquivo criado. Complete a descrição do módulo conforme o projeto do pré-relatório. Salve e corrija possíveis erros de sintaxe.

```

|module xxDsDrv(
|    input C,
|    input [3:0] Da, Db, Dc, Dd,
|    input En,
|    output [3:0] An,
|    output [0:6] Seg
|);
|endmodule

```



Exercício 4 Desenhe o diagrama lógico do circuito de acionamento do mostrador de sete segmentos xxDsDrv, que se encontra delimitado pela linha tracejada na Figura 5.19 (o display de 7 segmentos é um elemento externo e não faz parte do circuito). Use os elementos mostrados na Figura 5.21. A entrada Clr do contador deve ser ligada à 0 V (GND), já que não será usada. Em seguida, descreva o circuito em Verilog por meio de um módulo xxDsDrv.



Exercício 5 Complete o módulo `xxDsDrv` com a codificação em Verilog do circuito projetado no exercício anterior. Não é necessário escrever o módulo inteiro, o que falta para completar o esqueleto.



Instancie o acionador `xxDsDrv` no circuito do cronômetro `Cron_top`.

- Clique duas vezes sobre o módulo `Cron_top` para abri-lo no editor. Complete o módulo com a instanciação do acionador `xxDsDrv`. Salve e corrija possíveis erros de sintaxe.
- Gere o *bitstream* e transfira para a FPGA.
- Verifique o funcionamento do circuito.

Exercício 6 Faça o diagrama lógico do circuito do cronômetro `Cron_top` (listagem da Figura 5.22). Represente os módulos internos (`Clock400`, `CB2`, etc) por seus símbolos. Represente as chaves por retângulos e o led por um círculo, como feito nas experiências anteriores. Não é necessário incluir os identificadores dos pinos da FPGA.



Anotação 8a Mostre o **circuito funcionando** para o professor. Anote a **hora atual**: _____.

Atividade 9 Run / Stop

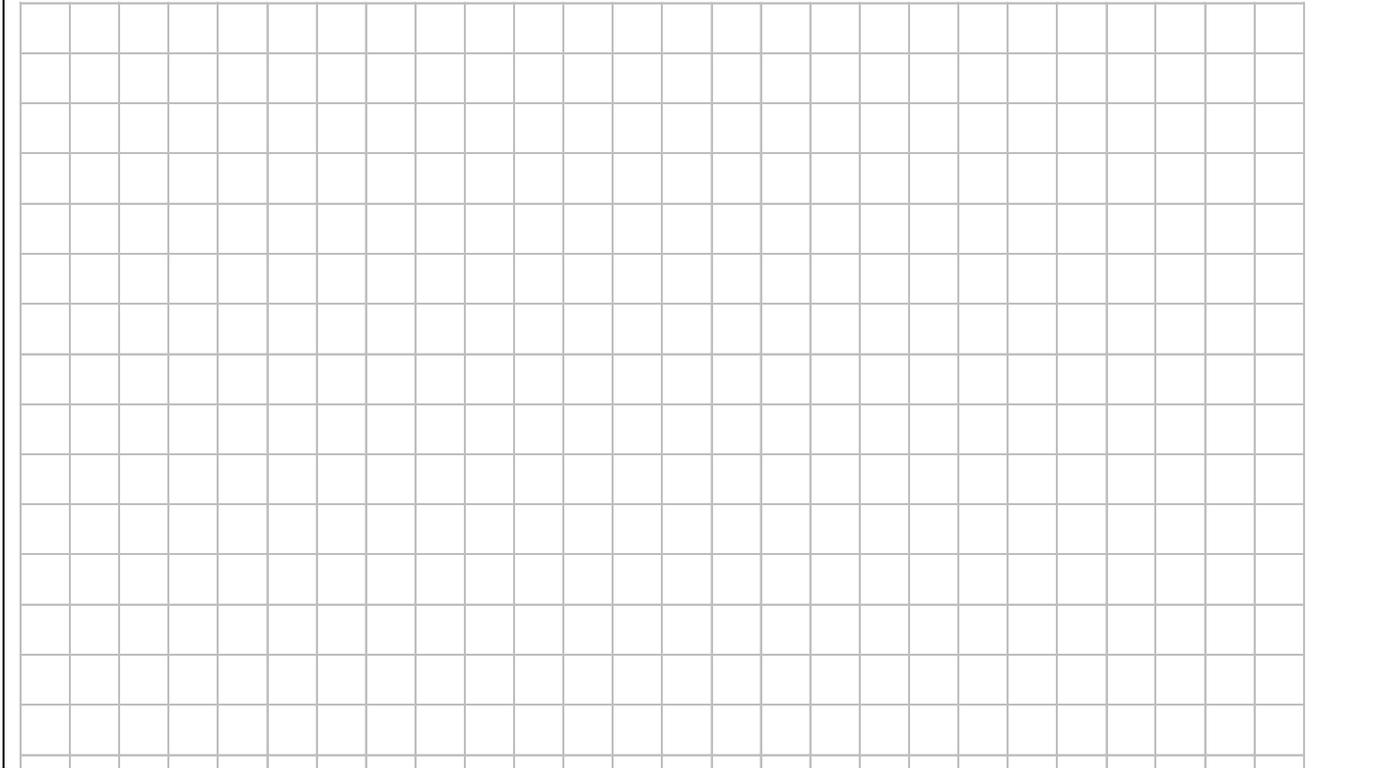
Faça uma pequena alteração no cronômetro para incluir mais dois botões: Run (btnL) e Stop (btnR).

Ao se pressionar o botão Run, o cronômetro deve disparar. Pressionando-se Stop, a cronometragem se detém; nessa condição, apertando-se o botão Run novamente o cronômetro continua a partir da contagem em que parou.

ATENÇÃO: o botão Clear (bntD) **SOMENTE** pode zerar o cronômetro quando este estiver **PARADO**.

Dica: use um latch para controlar a habilitação dos contadores. Veja o módulo LatchSR já existente no projeto.

Anotação 9a Faça o diagrama lógico do circuito modificado de forma **resumida**: havendo componentes repetidos com ligações iguais, desenhe somente o primeiro e o último com reticências entre eles. Indique os botões e leds da placa usados nas entradas e saídas do circuito. Descreva as modificações feitas no módulo Cron_top.v .



Antes de mexer no módulo Cron_top, salve-o com outro nome .

- Com o módulo aberto no editor, clique o botão da direita na janela do editor e clique em “Save File As...” Salve como **xxCronOrig_top**, substituindo xx pelas iniciais do seu grupo.
- Modifique o módulo Cron_top para incluir o circuito de Run / Stop. Salve e corrija possíveis erros de sintaxe.
- Abra o arquivo de *constraints* Cron.xdc. Descomente os pares de linhas que definem os botões btnL e btnR. Salve.

Por fim...

- Gere o bitstream e transfira para a FPGA.
- Teste o circuito.

Anotação 9b Mostre o circuito funcionando para o professor. Anote a hora atual: _____.

Atividade 10 Opcional (em anexo)

Atividade 11 Finalização

Se quiser, você pode copiar os arquivos dos projetos em um pendrive.

- 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.