Elaborada por Jardel Silveira e Vanessa Rodrigues

## Executando o exemplo CL hello\_world com a GUI Vivado

#### Descrição

Nesta prática vamos executar o exemplo CL hello\_world, que foi usado na prática <u>Criação de uma Amazon FPGA Image (AFI) do exemplo CL hello\_world</u>. Porém não usaremos o modo batch como anteriormente, ao invés disso, faremos o uso da interface gráfica do vivado para a sintetização e a simulação do projeto.

#### Objetivos de Aprendizagem

- Acesso à interface gráfica da instância t2.2xlarge.
- Acesso a interface gráfica do vivado instalado na instância t2.2xlarge.
- Simulação do exemplo hello\_world.
- Implementação e sintetização (geração do arquivo .tar pela interface do Vivado).

# Parte 1: Configurar a máquina local e a instância para permitir o acesso a interface gráfica

- 1. Execute uma instância EC2 t2.2xlarge
- 2. Na máquina local, execute o seguinte comando para permitir que seja acessada por qualquer host

```
xhost +
```

3. Na instância t2.2xlarge, execute o seguinte comando para verificar se o SSH está configurado para aceitar que conexões através do modo gráfico

```
cat /etc/ssh/sshd_config | grep X11Forwarding
```

O resultado deste comando deverá ser "X11Forwarding yes".

- 4. Ainda na instância t2.2xlarge, execute o seguinte comando para instalar o xauth sudo yum install xauth
- 5. Feche a sessão SSH e em seguida reconecte-se adicionando no comando de conexão o parâmetro -X, como no exemplo abaixo

```
ssh -X -i "jardel-key-pair-useast1.pem" 

<u>centos@ec2-54-209-133-69.compute-1.amazonaws.com</u>
```

6. Execute o comando

```
export DISPLAY=localhost:10.0
```

7. Instale o editor de texto gedit com o seguinte comando

```
sudo yum install gedit
```

8. Execute o comando gedit. A interface do gedit deverá abrir. Isso significa que a configuração para o uso da interface foi realizada com sucesso.

#### Parte 2: Configurando o IP Integrator

1. Configure o HDK

```
git clone https://github.com/aws/aws-fpga.git $AWS_FPGA_REPO_DIR
cd $AWS_FPGA_REPO_DIR
source hdk_setup.sh
```

Entre no diretório ~/.Xilinx/Vivado e verifique se o arquivo init.tcl ou o arquivo Vivado\_init.tcl existe dentro desse diretório. Caso nenhum destes dois arquivos existam, crie o arquivo com o seguinte comando

```
touch Vivado init.tcl
```

3. Obtenha o path absoluto de \$HDK SHELL DIR com o seguinte comando

```
echo $HDK_SHELL_DIR
```

4. No arquivo init.tcl ou Vivado init.tcl, adicione a seguinte linha

```
source <saida do comando echo $HDK_SHELL_DIR>/hlx/hlx_setup.tcl
```

Toda vez que o Vivado for executado, esse script sempre será fornecido e os recursos do integrador de IP serão carregados automaticamente.

Obs: Você poderá editar o arquivo usando o gedit previamente instalado, executando o comando gedit Vivado\_init.tcl

5. Execute o vivado com o comando vivado

# Parte 3: Simulação do exemplo hello\_world usando a interface gráfica do Vivado

- 1. Uma vez que o vivado tenha sido executado com a interface gráfica, feche-o e retorne ao terminal
- Adicione as seguintes variáveis de ambiente para clock recipes e IDs para o exemplo hello\_world

```
export CLOCK_A_RECIPE=0
```

```
export CLOCK_B_RECIPE=0

export CLOCK_C_RECIPE=0

export device_id=0xF000

export vendor_id=0x1D0F

export subsystem_id=0x1D51

export subsystem_vendor_id=0xFEDD
```

- 3. Mude para o diretório /home/centos/src/project\_data/aws-fpga/hdk/cl/examples/hello\_world\_hlx
- 4. Execute o vivado com o comando vivado. Uma vez que a GUI tiver sido carregada, clique no console TCL
- 5. Digite o comando aws::make\_rtl -examples cl\_hello\_world para criar o exemplo hello\_world\_hlx. O exemplo será gerado no diretório cl/examples/hello\_world\_hlx/example\_projects. O projeto do vivado será o arquivo hello\_world.xpr contido na pasta examples\_projects.
- 6. Clique em Simulation->Run Simulation->Run Behavioral Simulation
- 7. Adicione os sinais necessários para a simulação
- 8. Digite o comando -run all no TCL console

#### Parte 4: Simulação DPI com test\_hello\_world.c

- 1. Em Project Manager, clique com o botão direito em Simulation e selecione Simulation Settings
- 2. Selecione a caixa ... em Verilog options e modifique o valor de TEST\_NAME para test\_null desativar o sv stimulus

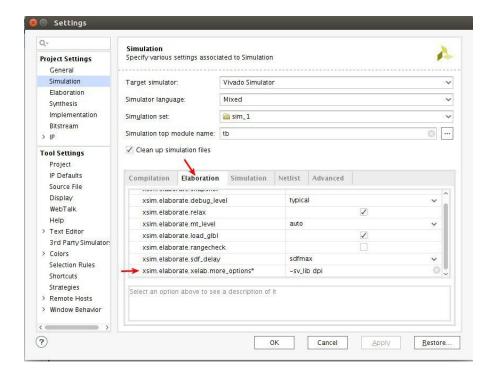
```
TEST NAME = test null
```

3. No TCL console, em Vivado Project, copie e cole o seguinte comando para setar o path para criação do .so com o script test\_hello\_world.c. Caso esteja usando simuladores de terceiros, modifique o comando para combinar o simulador e o path para dpi.tcl em vez de dpi\_xsim.tcl

```
set_property -name {xsim.compile.tcl.pre} -value
$::aws::make_faas::_nsvars::script_dir/../../hlx_examples/build/RTL/cl_hell
o_world/verif/scripts/dpi_xsim.tcl -objects [get_filesets sim_1]
```

- 4. Em Project Manager, clique com o botão direito em Simulation e selecione Simulation Settings
- 5. Na tab Elaboration adicione o seguinte valor para xsim.elaborate.xelab.more\_options

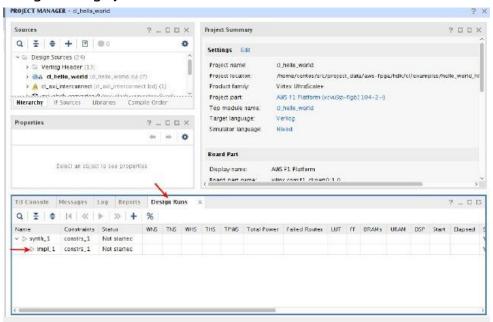
```
-sv_lib dpi
```



As configurações podem ser diferentes dependendo do simulador. Certos simuladores de terceiros podem precisar que seja incluído um path explícito o diretório de design para projetos de exemplo RTL fornecidos como cl\_hello\_world e cl\_dram\_dma. Para isso, na opção Verilog, selecione a caixa ... e clique no botão + sob Verilog Include Files Search Paths e selecione o path para o diretório cl/ cl\_example/design.

### Parte 5: Implementando o arquivo Design/Tar

- 1. Na tab Design Runs, clique com o botão direito em impl 1 e selecione Lauch Runs
- 2. Clique em OK na caixa de diálogo Launch Runs e depois clique em OK na caixa de diálogo Missing Synthesis Results



- 3. Após isso, a sintetização e a implementação serão executadas
- 4. Após terminada a execução o arquivo .Developer\_CL.tar será gerado no diretório example\_projects/cl\_hello\_world.runs/faas\_1/build/checkpoints/to\_aws/
- 5. Com o arquivo .tar gerado é possível criar uma uma AFI para carregá-la em uma instância F1. Para isso, deve-se seguir a prática <u>Criar uma Amazon FPGA Image (AFI) de um dos Exemplos CL</u>, a partir da parte 1 ítem 9.

### Referências

- Amazon Web Services. Hardware Development Kit (HDK) e Software Development Kit
   (SDK) [internet]. [Acesso em: 26 dez. 2017]. Disponível em: https://github.com/aws/aws-fpga/blob/master/hdk/docs/IPI\_GUI\_Vivado\_Setup.md
- ALMEIDA, Bruno. Acessando modo gráfico da sua instância EC2. 2013. Disponível em:
   <a href="http://blog.rivendel.com.br/2013/09/13/acessando-modo-grafico-da-sua-instancia-ec2/">http://blog.rivendel.com.br/2013/09/13/acessando-modo-grafico-da-sua-instancia-ec2/</a>>. Acesso em: 20 fev. 2018.