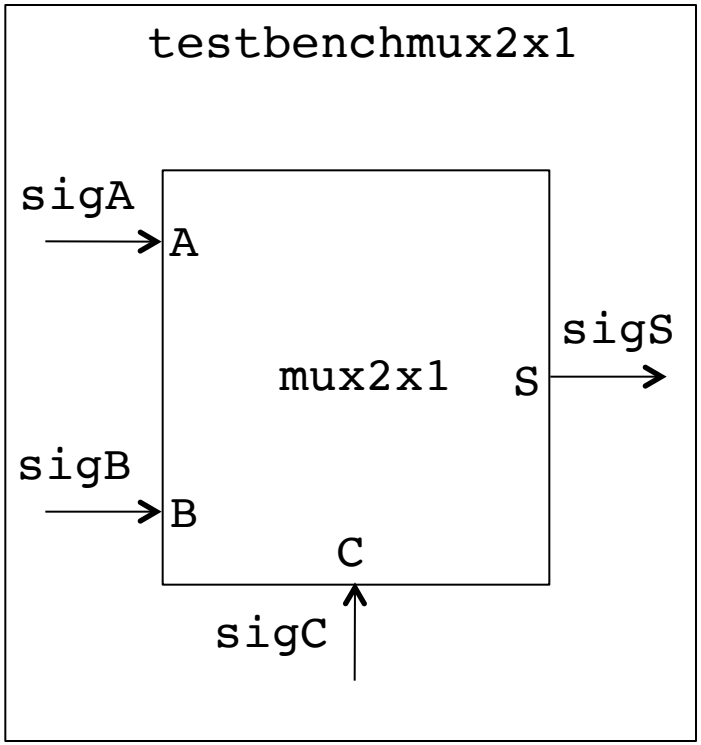
Aula 2 − Validação Funcional

Se fossemos para a banca, testar o componente, colocaríamos fios nas entradas e saídas. Injetaríamos sinais de estímulo nos fios de entrada e conectaríamos um osciloscópio nos fios de saída, permitindo a visualização do comportamento destes.

Como não temos o componente disponível fisicamente, mas sim um modelo deste, utilizamos simulação para validar o seu funcionamento.

Como tudo em VHDL é componente, criamos um artifício em que o componente a ser testado (`mux2x1`) passa a ser visto como um subcomponente do componente de teste (`testbenchmux2x1`).

Obs: Ao longo do texto, selecionar significa *clicar* uma vez!



Dessa forma, a simulação do componente `testbenchmux2x1` permite gerar os estímulos (`sigA`, `sigB`, `sigC`) para o `mux2x1` e observar o resultado obtido (`sigS`).

Deve-se observar que `A`, `B`, `C` e `S` são sinais internos do `mux2x1`, acessíveis através de `sigA`, `sigB`, `sigC` e `sigS`, respectivamente!

O componente `testbenchmux2x1` tem como característica a ausência de sinais de interface:

```
entity testbenchmux2x1 is
end testbench;
```

Como `testbenchmux2x1` é constituído pelo componente `mux2x1`, sua descrição será necessariamente estrutural:

```
architecture estrutural of testbenchmux2x1 is

component mux2x1
port (A,B,C: in bit; S: out bit);
end component;

signal sigA,sigB,sigC,sigS: bit;
```

No corpo da arquitetura do `testbenchmux2x1`, instanciamos o componente a ser testado (`componente`), como sendo do tipo `mux2x1`, e especificamos os estímulos a este ao longo do tempo:

```
begin
     componente: mux2x1 port map(sigA,sigB,sigC,sigS);
     sigA <= '0', '1' after 10 ns, '0' after 15 ns;
     sigB <= '1', '0' after 12 ns, '1' after 17 ns;
     sigC <= '0', '1' after 5 ns, '0' after 13 ns;
end estrutural;
```

Observe que `sigS` não é acionado, uma vez que está associado à resposta do componente e não a um dos estímulos!

A sequência de atribuições é escalonada no tempo, tendo como referência o tempo t = 0, *i.e.* `sigA` assume '0' em t = 0ns, '1'em t = 10ns e '0'em t = 15ns, logo 5ns após a última atribuição ('1')!

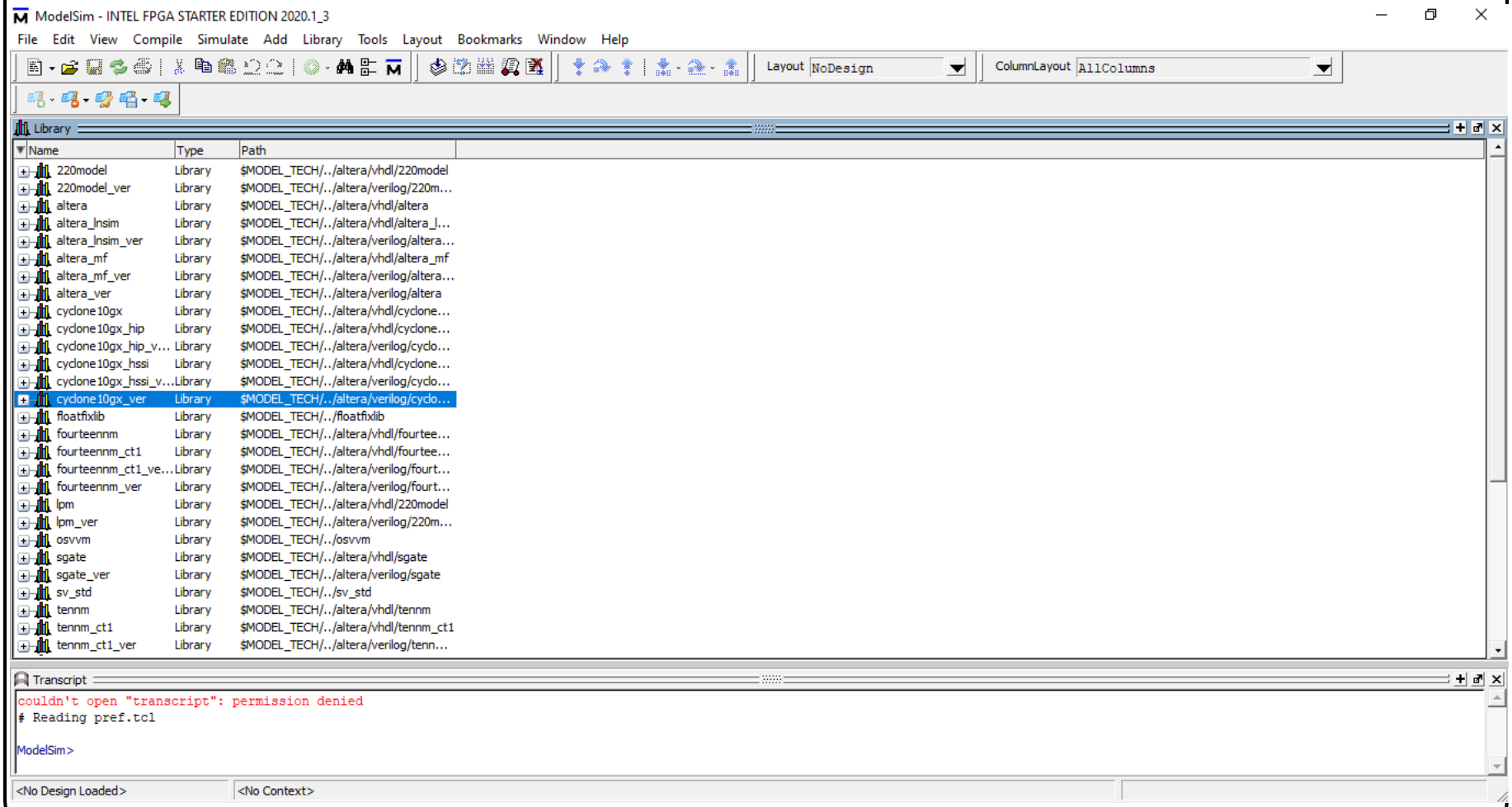
Para a simulação, vamos utilizar o simulador Modelsim-Intel FPGA Starter Edition (https://www.intel.com/content/www/us/en/software/programmable/quartus-prime/model-sim.html), disponível gratuitamente para Windows e Linux.

Para ter acesso à área dos arquivos, é necessário se registrar junto à Intel FPGA, para obter um nome de usuário e uma senha, assim que fizer acesso à página.

Há dois arquivos a serem instalados: ModelSimProSetup e ModelSimProSetup-part2. Em seguida, execute ModelsimProSetup, que irá utilizar o ModelSimProSetup-part2 automaticamente.

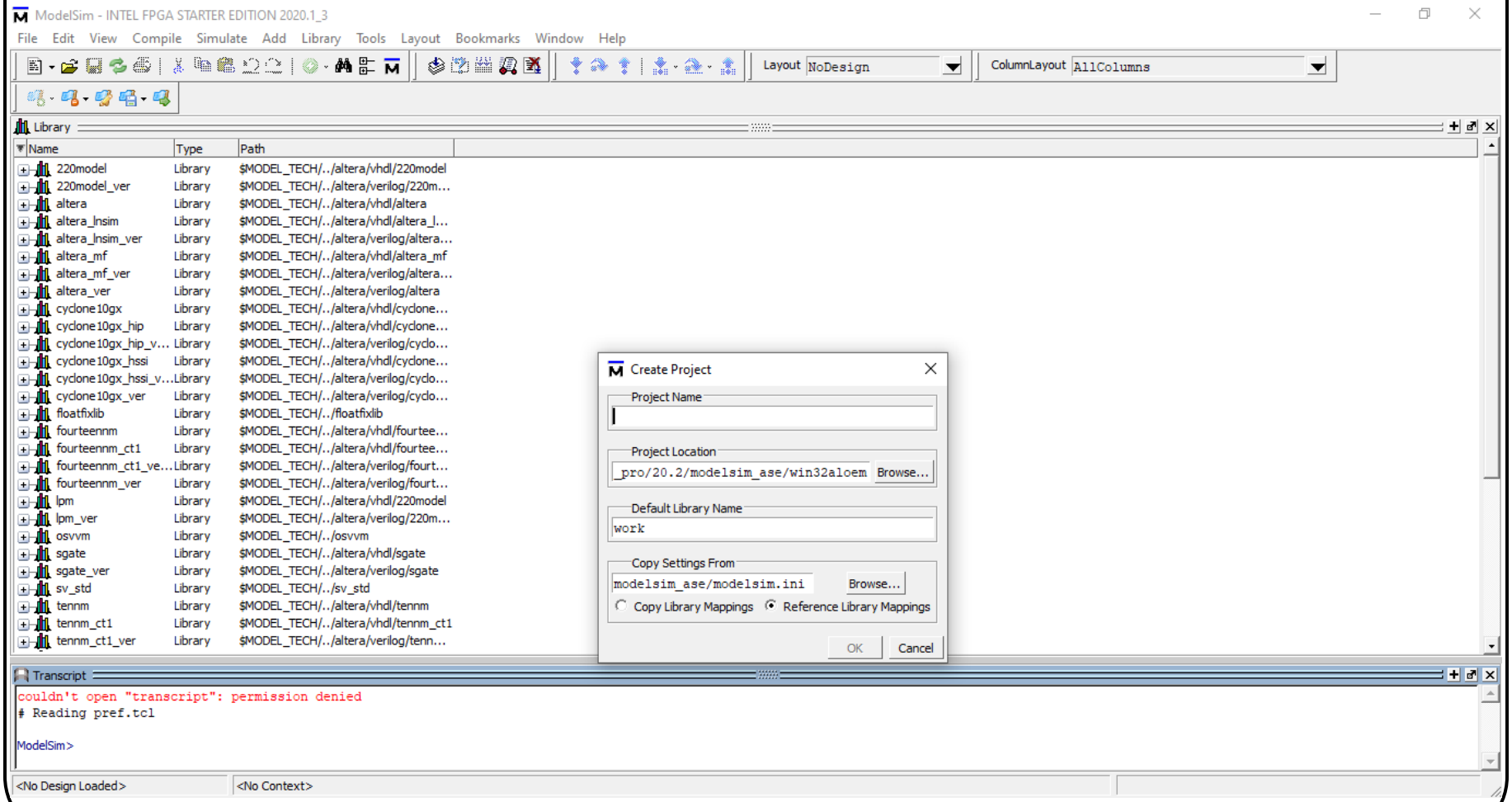
No Windows, o simulador (modelsim.exe) encontra-se na pasta intelFPGA_pro\20.2\modelsim_ase\win32aloem!

A figura abaixo corresponde à janela de abertura do Modelsim. Nela, identificamos a guia de comandos (topo), a janela da biblioteca (Library) e a janela de mensagens (Transcript), na parte inferior.



O primeiro passo é criar um projeto, que irá agregar todos os componentes associados a ele, posteriormente:

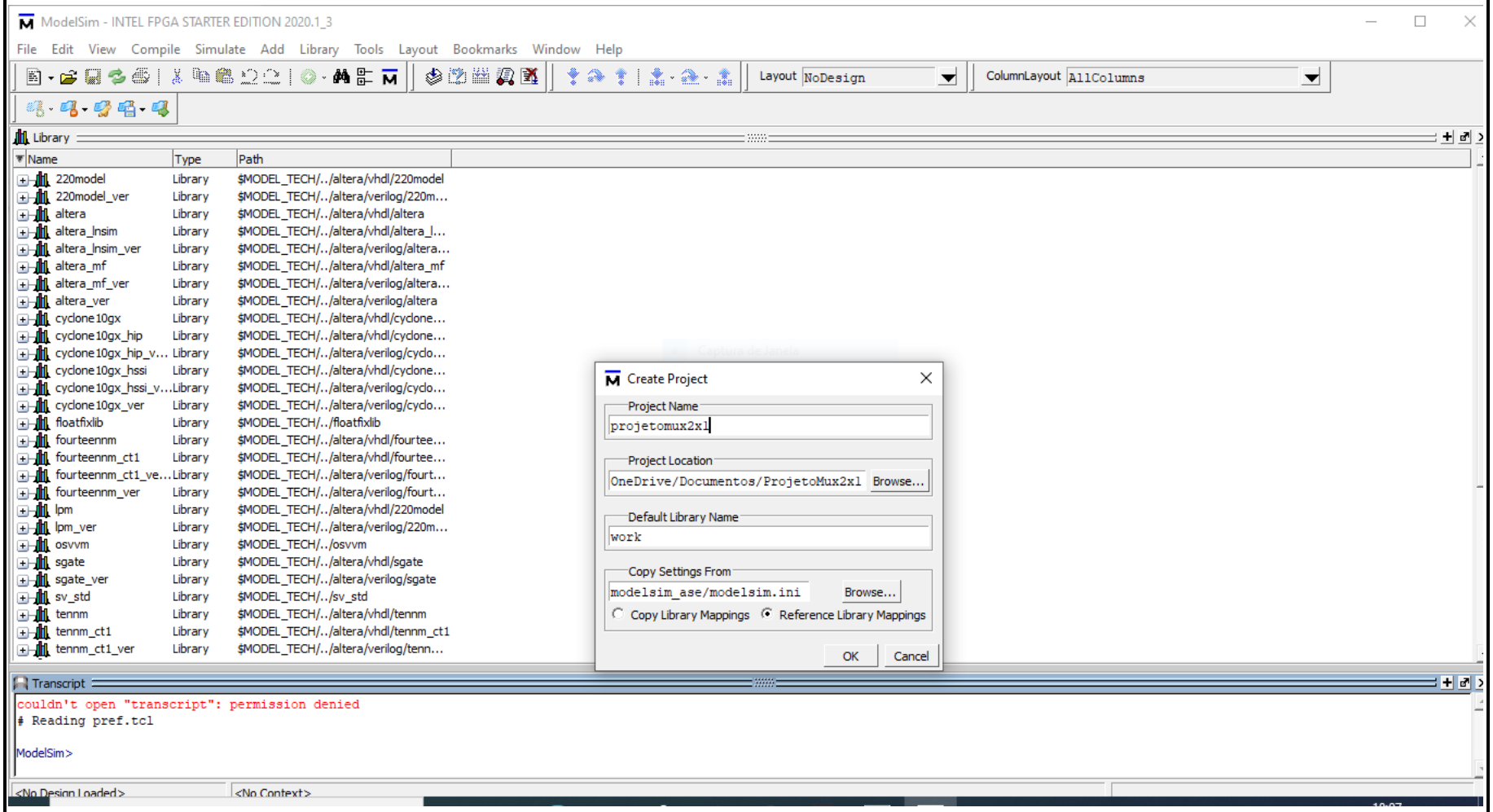
- File>New>Project



- Project Name: insira o nome do projeto, *e.g.* projetomux2x1
- Project Location>Browse... e selecione a pasta onde o projeto irá ser armazenado, utilizando uma pasta diferente da que contém o Modelsim (intelFPGA_pro); neste momento, podemos criar a pasta com o nome ProjetoMux2x1
- Default Library: work
- Copy Settings From: modelsim_ase/modelsim.ini

Selecione Reference Library Mappings.

No exemplo, o arquivo estará na pasta C:/Users/ldmm/Documents/ProjetoMux2x1, com o nome projetomux2x1.mpf. Nunca o abra!



- OK

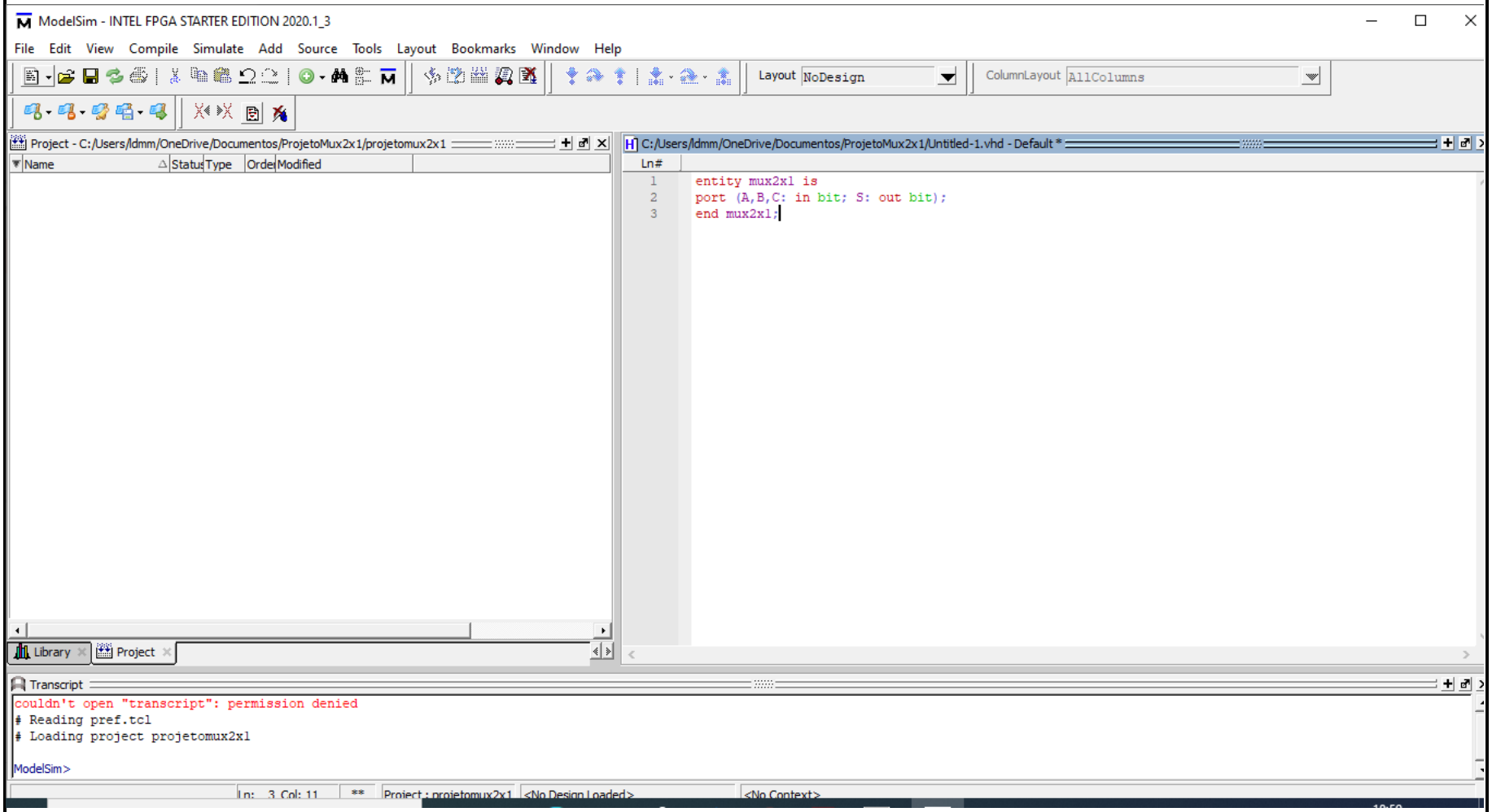
Feche a janela Add items to the Project.

Agora temos uma nova janela, designada Project, ao lado de Library. Selecione a janela Library, onde poderá observar a presença da pasta work, com a informação de que está vazia (empty). É nessa biblioteca que todos os componentes que comporão o `mux2x1` serão montados. Não faça alterações nessa pasta!

Agora vamos criar o nosso primeiro componente. Selecione a janela Project e, em seguida:

- File>New>Source>VHDL

Uma nova janela (Untitled-1.vhd) é disponibilizada à direita de Project, onde iremos editar a descrição da entidade do `mux2x1`, conforme elaborado anteriormente.


ModelSim - INTEL FPGA STARTER EDITION 2020.1_3
File Edit View Compile Simulate Add Source Tools Layout Bookmarks Window Help
Layout NoDesign
ColumnLayout AllColumns
Project - C:/Users/ldmm/OneDrive/Documentos/ProjetoMux2x1/projetomux2x1
Name Status Type Order Modified
C:/Users/ldmm/OneDrive/Documentos/ProjetoMux2x1/Untitled-1.vhd - Default *
Ln#
1 entity mux2x1 is
2 port (A,B,C: in bit; S: out bit);
3 end mux2x1;
Library
Project
Transcript
couldn't open "transcript": permission denied
Reading pref.tcl
Loading project projetomux2x1
ModelSim>
Ln: 3 Col: 11
Project : projetomux2x1
<No Design Loaded>
<No Context>

Em seguida, vamos salvá-lo:

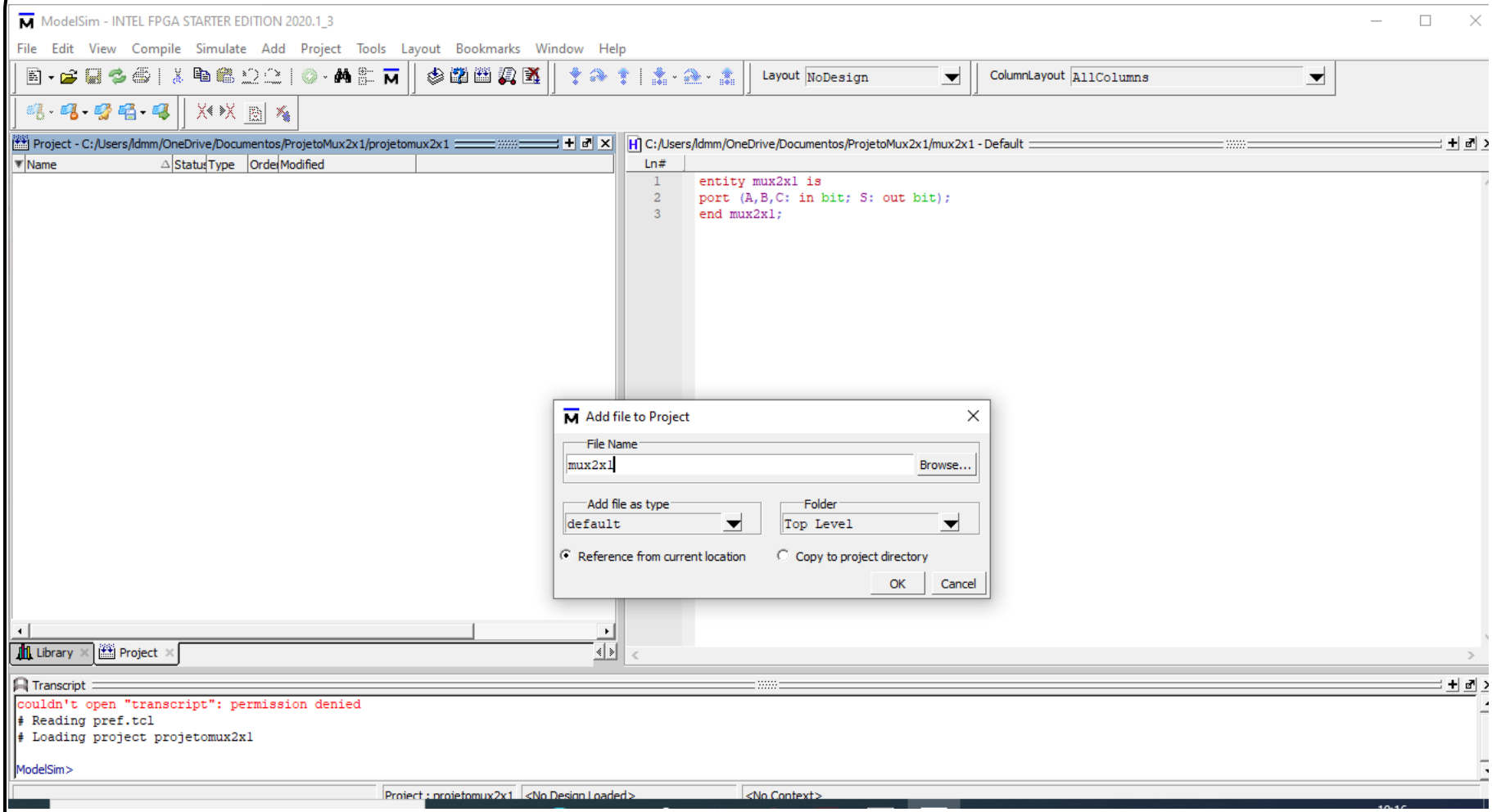
- File>Save As...

Na janela Save As, observe que a pasta corresponde à que foi criada para o projeto (ProjetoMux2x1). Atribua o nome ao componente, *i.e.* `mux2x1`, e salve na pasta do projeto. Não guarde nada na pasta work!

Selecione a janela Project e, em seguida:

- Project>Add to Project>Existing File...

Na janela Add file to Project, no campo File Name, insira o nome do arquivo que contém a descrição da entidade do componente `mux2x1`, *i.e.* `mux2x1`. Você também pode utilizar o botão Browse, que permitirá visualizar a pasta ProjetoMux2x1, onde se encontra o arquivo mux2x1.vhd.


ModelSim - INTEL FPGA STARTER EDITION 2020.1_3
File Edit View Compile Simulate Add Project Tools Layout Bookmarks Window Help
Layout NoDesign
ColumnLayout AllColumns
Project - C:/Users/ldmm/OneDrive/Documentos/ProjetoMux2x1/projetomux2x1
Name Status Type Order Modified
C:/Users/ldmm/OneDrive/Documentos/ProjetoMux2x1/mux2x1 - Default
Ln#
1 entity mux2x1 is
2 port (A,B,C: in bit; S: out bit);
3 end mux2x1;
Add file to Project
File Name
mux2x1
Browse...
Add file as type
default
Folder
Top Level
Reference from current location
Copy to project directory
OK
Cancel
Library
Project
Transcript
couldn't open "transcript": permission denied
Reading pref.tcl
Loading project projetomux2x1
ModelSim>
Project : projetomux2x1
<No Design Loaded>
<No Context>

Na janela Project, observe que o arquivo torna-se visível, com um ponto de interrogação em Status, significando que ainda não foi compilado. O processo de compilação verifica se o código está correto e, neste caso, gera a representação interna utilizada pelo simulador. Selecione o arquivo mux2x1 e compile-o:

➢Compile>Compile Selected

Na janela Transcript, aparece a mensagem "Compile of mux2x1 was successful" e, na janela Project, o atributo Status recebe ✓.

Vamos introduzir um erro no modelo, para ver como tratá-lo. Na janela de edição, retiramos o ; ao final da linha 2:

```
entity mux2x1 is
port (A,B,C: in bit; S: out bit)
end mux2x1;
```

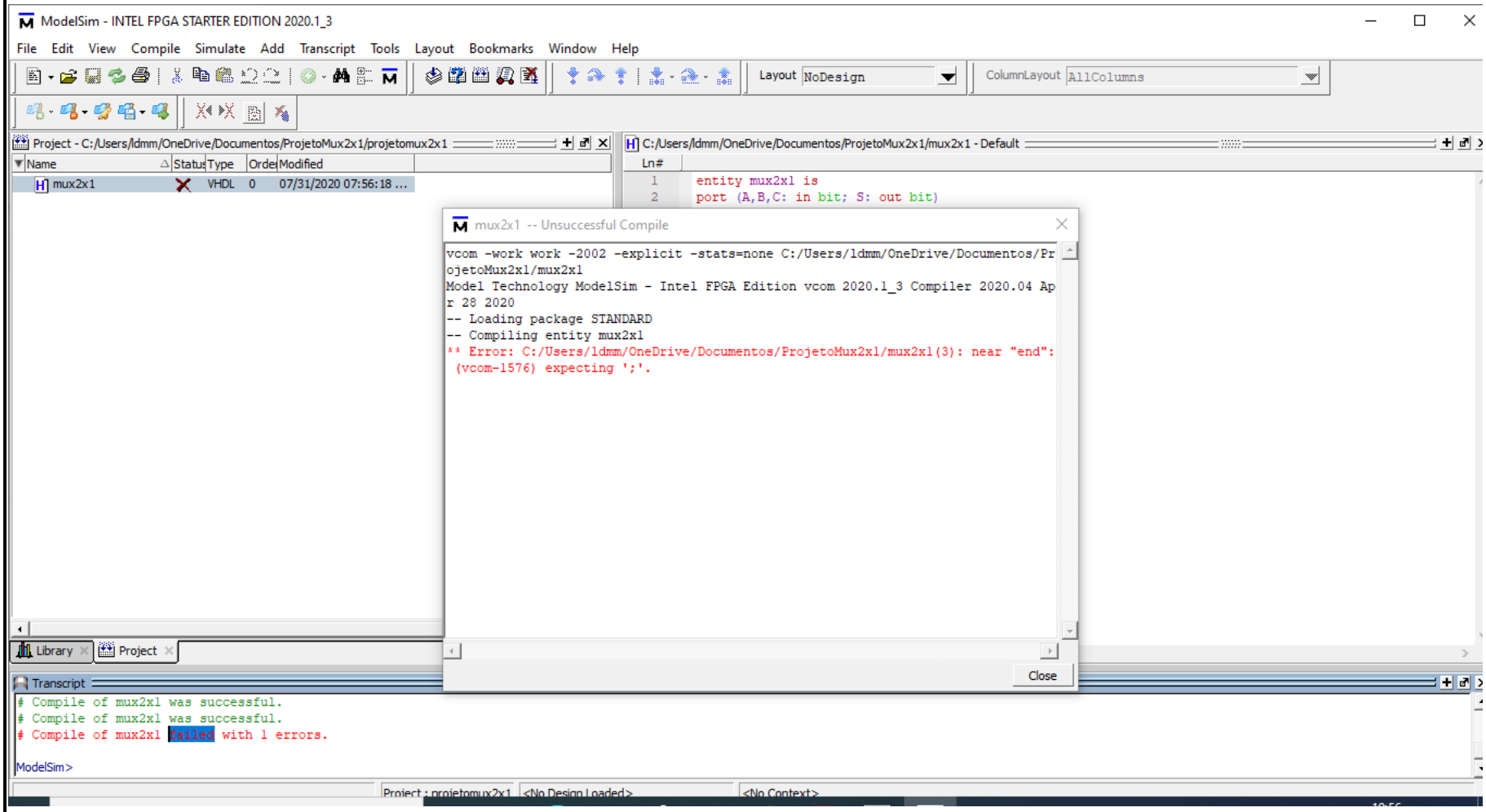
Em seguida, vamos salvá-lo:

- File>Save

Na janela Project, observe que o arquivo mux2x1 apresenta um ? no atributo Status, indicando que requer compilação. Selecione o arquivo mux2x1 e o compile novamente:

- Compile>Compile Selected

Na janela Transcript, aparece a mensagem "Compile of mux2x1 failed with 1 errors" e, na janela Project, o atributo Status recebe ✗. Ao selecionar duas vezes a mensagem, aparece a janela "mux2x1 − Unsuccessful Compile", indicando o erro na linha (3) e sugerindo que seja a falta do ;.


ModelSim - INTEL FPGA STARTER EDITION 2020.1_3
File Edit View Compile Simulate Add Transcript Tools Layout Bookmarks Window Help
Layout NoDesign
ColumnLayout AllColumns
Project - C:/Users/ldmm/OneDrive/Documentos/ProjetoMux2x1/projetomux2x1
Name Status Type Order Modified
mux2x1 VHDL 0 07/31/2020 07:56:18 ...
C:/Users/ldmm/OneDrive/Documentos/ProjetoMux2x1/mux2x1 - Default
Ln#
1 entity mux2x1 is
2 port (A,B,C: in bit; S: out bit)
mux2x1 -- Unsuccessful Compile
vcom -work work -2002 -explicit -stats=none C:/Users/ldmm/OneDrive/Documentos/ProjetoMux2x1/mux2x1
Model Technology ModelSim - Intel FPGA Edition vcom 2020.1_3 Compiler 2020.04 Apr 28 2020
-- Loading package STANDARD
-- Compiling entity mux2x1
** Error: C:/Users/ldmm/OneDrive/Documentos/ProjetoMux2x1/mux2x1(3): near "end": (vcom-1576) expecting ';'.
Close
Library
Project
Transcript
Compile of mux2x1 was successful.
Compile of mux2x1 was successful.
Compile of mux2x1 failed with 1 errors.
ModelSim>
Project : projetomux2x1
<No Design Loaded>
<No Context>

Feche a janela “mux2x1 − Unsuccessful Compile” e retorne para a janela de edição. Observe que o erro não está na linha 3, mas nas imediações desta. Neste caso, ao final da linha 2. Vamos corrigir, incluindo o ; e, em seguida, salvá-lo:

- File>Save

Na janela Project, selecione o arquivo mux2x1 e compile-o novamente:

- Compile>Compile Selected

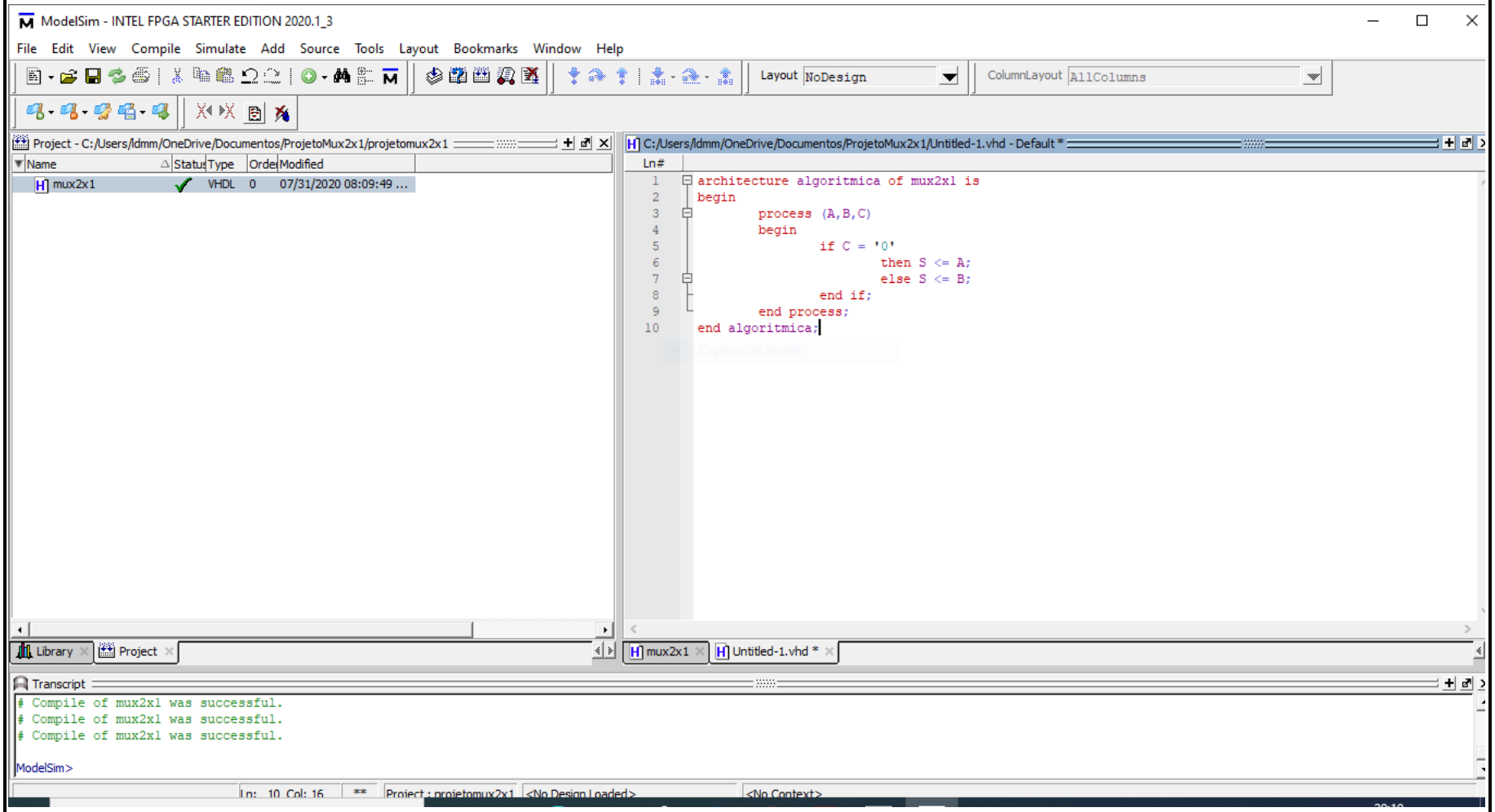
Na janela Transcript, aparece a mensagem “Compile of mux2x1 was successful” e, na janela Project, o atributo Status recebe ✓.

Na janela Library, observe que a work não está mais vazia. Selecione o botão + ao lado da work. Agora temos, nesta biblioteca, a entidade (E) `mux2x1`. Novamente, não faça qualquer alteração/manipulação em qualquer elemento desta janela! Você pode somente consultar, selecionando o botão +.

Agora, vamos especificar a arquitetura do componente `mux2x1`. Selecione a janela Project e, em seguida:

- File>New>Source>VHDL

Observe que uma nova janela de edição aparece (Untitled-1.vhd), mantendo a anterior (`mux2x1`). Na nova janela, vamos editar a descrição da arquitetura do `mux2x1`, no domínio comportamental algorítmico, conforme elaborado anteriormente.


ModelSim - INTEL FPGA STARTER EDITION 2020.1_3
File Edit View Compile Simulate Add Source Tools Layout Bookmarks Window Help
Layout NoDesign
ColumnLayout AllColumns
Project - C:/Users/ldmm/OneDrive/Documentos/ProjetoMux2x1/projetomux2x1
Name Status Type Order Modified
mux2x1 VHDL 0 07/31/2020 08:09:49 ...
C:/Users/ldmm/OneDrive/Documentos/ProjetoMux2x1/Untitled-1.vhd - Default *
Ln#
1 architecture algoritmica of mux2x1 is
2 begin
3 process (A,B,C)
4 begin
5 if C = '0'
6 then S <= A;
7 else S <= B;
8 end if;
9 end process;
10 end algoritmica;
Library
Project
mux2x1
Untitled-1.vhd *
Transcript
Compile of mux2x1 was successful.
Compile of mux2x1 was successful.
Compile of mux2x1 was successful.
ModelSim>
Ln: 10 Col: 16
Project : projetomux2x1
<No Design Loaded>
<No Context>

- File>Save As...

Vamos atribuir o nome mux2x1algoritmica para o arquivo que irá conter a especificação da arquitetura algoritmica.

Selecione a janela Project e, em seguida:

- Project>Add to Project>Existing File...

Na janela Add file to Project, no campo File Name, insira o nome do arquivo que contém a especificação da arquitetura algorítmica do componente `mux2x1`, *i.e.* `mux2x1algoritmica` (sem acento!).

Na janela Project, selecione o arquivo mux2x1algoritmica e, em seguida, compile-o:

- Compile>Compile Selected

Na janela Transcript, aparece a mensagem “Compile of mux2x1algoritmica was successful” e o atributo Status recebe ✓.

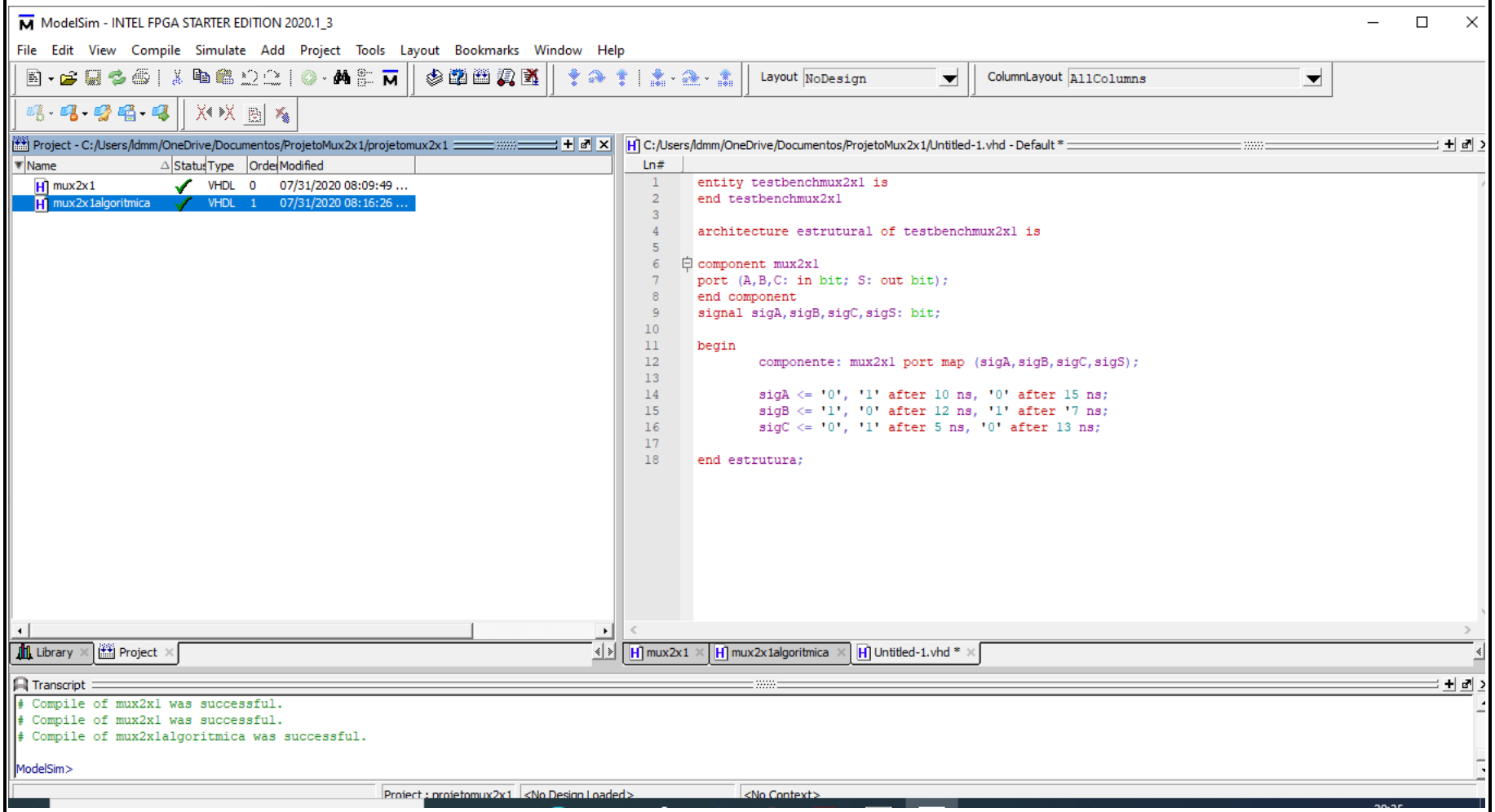
Na janela Library, selecione o botão + da work. Selecione o botão + junto à entidade `(E) mux2x1`. Agora temos a arquitetura `(A) algoritmica` do `mux2x1`. Novamente, não faça qualquer alteração/ manipulação em qualquer elemento desta janela! Você pode somente consultar, selecionando o botão +.

Pronto, o componente `mux2x1` já está pronto na biblioteca work! O próximo passo é verificar se o componente `mux2x1` funciona de acordo com a tabela verdade, apresentada anteriormente.

Para tal, vamos criar o componente `testbenchmux2x1` já elaborado. Veja que o projeto continua o mesmo (projetomux2x1.mpf), já que `testbenchmux2x1` é parte desse projeto!

Selecione a janela Project e, em seguida:

- File>New>Source>VHDL



- File>Save As...

Na janela Save As, atribua o nome ao componente, *i.e.* `testbenchmux2x1`, e salve-o.

Selecione a janela Project e, em seguida:

- Project>Add to Project>Existing File...

Na janela Add file to Project, no campo File Name, insira o nome do arquivo que contém a descrição da entidade e arquitetura do componente `testbenchmux2x1`, *i.e.* testbenchmux2x1.

Na janela Project, selecione o arquivo testbenchmux2x1:

- Compile>Compile Selected

Na janela Transcript, aparece a mensagem “Compile of testbenchmux2x1 was successful” e o atributo Status recebe ✓.

Na janela Library, selecione o botão + ao lado da work. Agora temos, nesta biblioteca, a entidade `(E) testbenchmux2x1`. Selecione o botão + junto à entidade e teremos a arquitetura `(A) estrutural` do `testbenchmux2x1`.

No caso do `testbenchmux2x1`, elaboramos a entidade e a arquitetura em um mesmo arquivo (testbenchmux2x1). No processo de compilação, as duas especificações são identificadas para a montagem da biblioteca work!

A vinculação da arquitetura à entidade se faz a partir do cabeçalho em `architecture`, onde se declara o nome da arquitetura (`estrutural`) e a qual entidade corresponde (`testbenchmux2x1`).

A simulação se faz a partir do `testbenchmux2x1`, que é onde o componente a ser testado é invocado e os estímulos especificados!

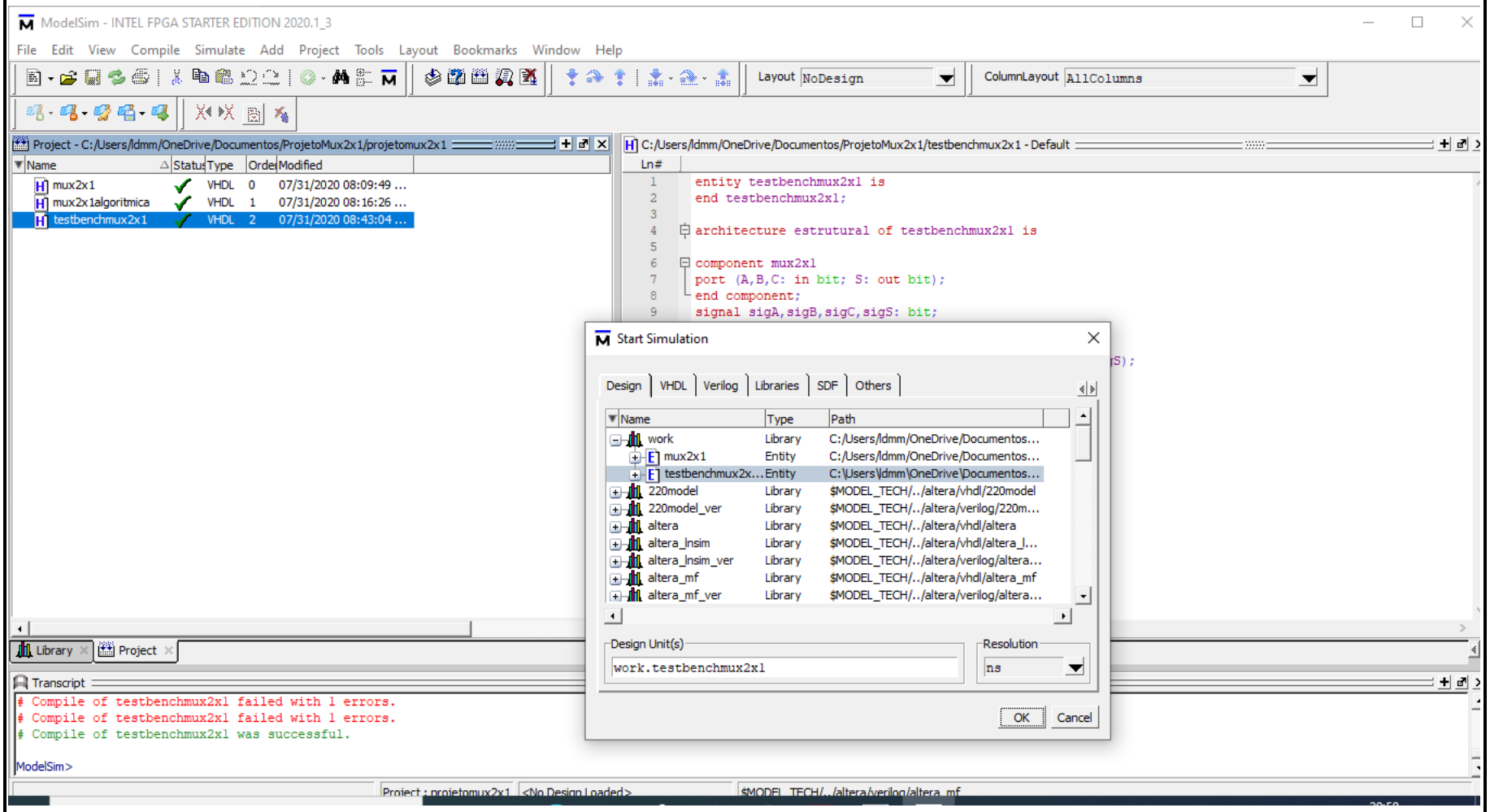
Na guia de comandos, no topo da página:

- Simulate>Start Simulation...

Na janela Start Simulation, selecione o botão + ao lado de work e, em seguida, selecione a entidade `testbenchmux2x1.`No campo Resolution, escolha ns (nanosegundos).

- OK

Aguarde até que a janela sim esteja disponível do lado esquerdo, ao lado de Library e Project.

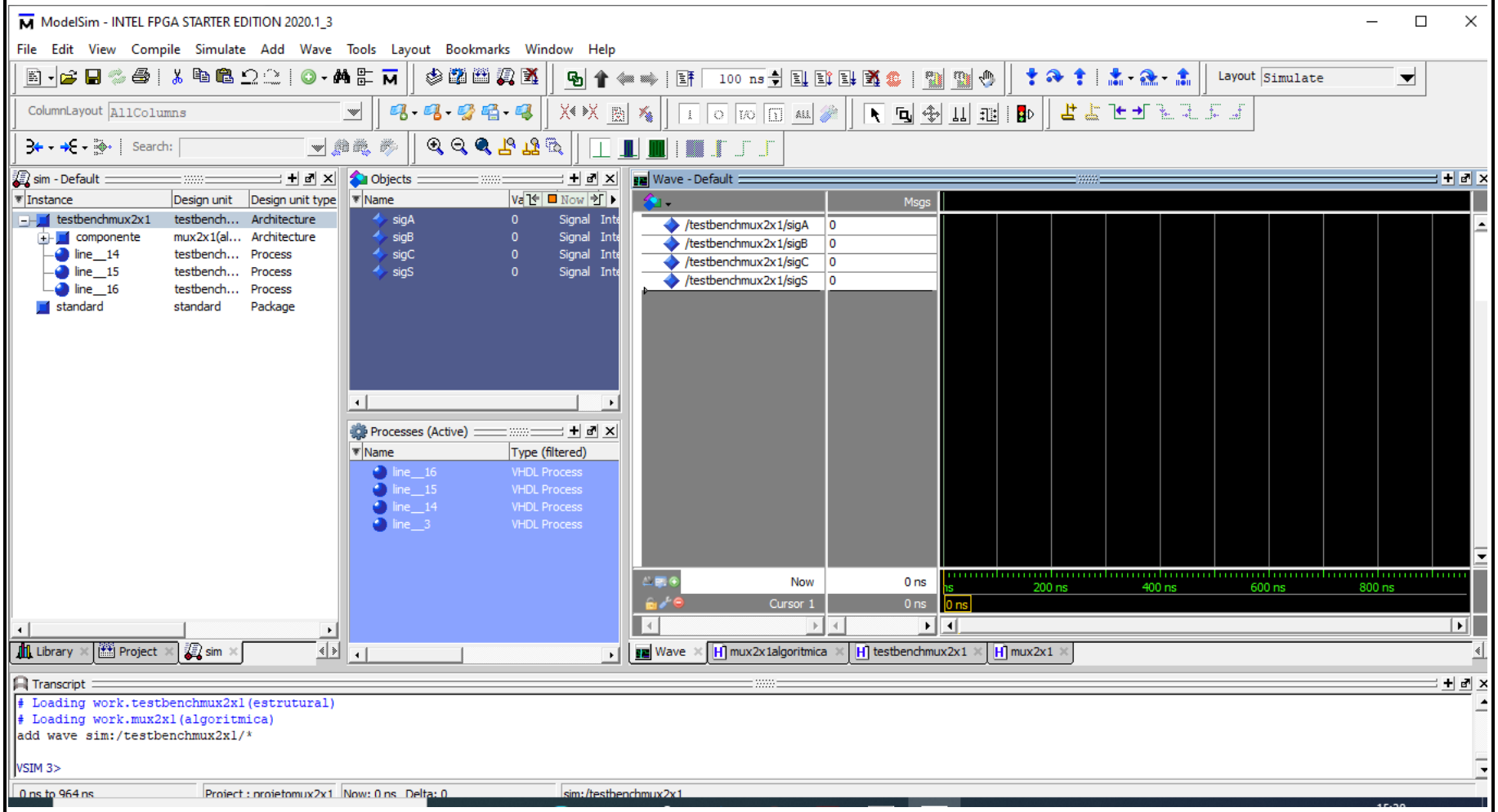

ModelSim - INTEL FPGA STARTER EDITION 2020.1_3
File Edit View Compile Simulate Add Project Tools Layout Bookmarks Window Help
Layout NoDesign
ColumnLayout AllColumns
Project - C:/Users/ldmm/OneDrive/Documentos/ProjetoMux2x1/projetomux2x1
Name Status Type Order Modified
mux2x1 VHDL 0 07/31/2020 08:09:49 ...
mux2x1algoritmica VHDL 1 07/31/2020 08:16:26 ...
testbenchmux2x1 VHDL 2 07/31/2020 08:43:04 ...
C:/Users/ldmm/OneDrive/Documentos/ProjetoMux2x1/testbenchmux2x1 - Default
Ln#
1 entity testbenchmux2x1 is
2 end testbenchmux2x1;
3
4 architecture estrutural of testbenchmux2x1 is
5
6 component mux2x1
7 port (A,B,C: in bit; S: out bit);
8 end component;
9 signal sigA,sigB,sigC,sigS: bit;
Start Simulation
Design VHDL Verilog Libraries SDF Others
Name Type Path
work Library C:/Users/ldmm/OneDrive/Documentos...
mux2x1 Entity C:/Users/ldmm/OneDrive/Documentos...
testbenchmux2x... Entity C:\Users\ldmm\OneDrive\Documentos...
220model Library $MODEL_TECH/../altera/vhdl/220model
220model_ver Library $MODEL_TECH/../altera/verilog/220m...
altera Library $MODEL_TECH/../altera/vhdl/altera
altera_lnsim Library $MODEL_TECH/../altera/vhdl/altera_l...
altera_lnsim_ver Library $MODEL_TECH/../altera/verilog/altera...
altera_mf Library $MODEL_TECH/../altera/vhdl/altera_mf
altera_mf_ver Library $MODEL_TECH/../altera/verilog/altera...
Design Unit(s)
work.testbenchmux2x1
Resolution
ns
OK
Cancel
Library
Project
Transcript
Compile of testbenchmux2x1 failed with 1 errors.
Compile of testbenchmux2x1 failed with 1 errors.
Compile of testbenchmux2x1 was successful.
ModelSim>
Project : projetomux2x1
<No Design Loaded>

Observe que, na janela de edição, temos uma aba Wave, que corresponde à janela onde os resultados de simulação serão amostrados na forma de ondas quadradas.

Na janela sim, selecione o `testbenchmux2x1`:

- Add>To Wave>All items in the region

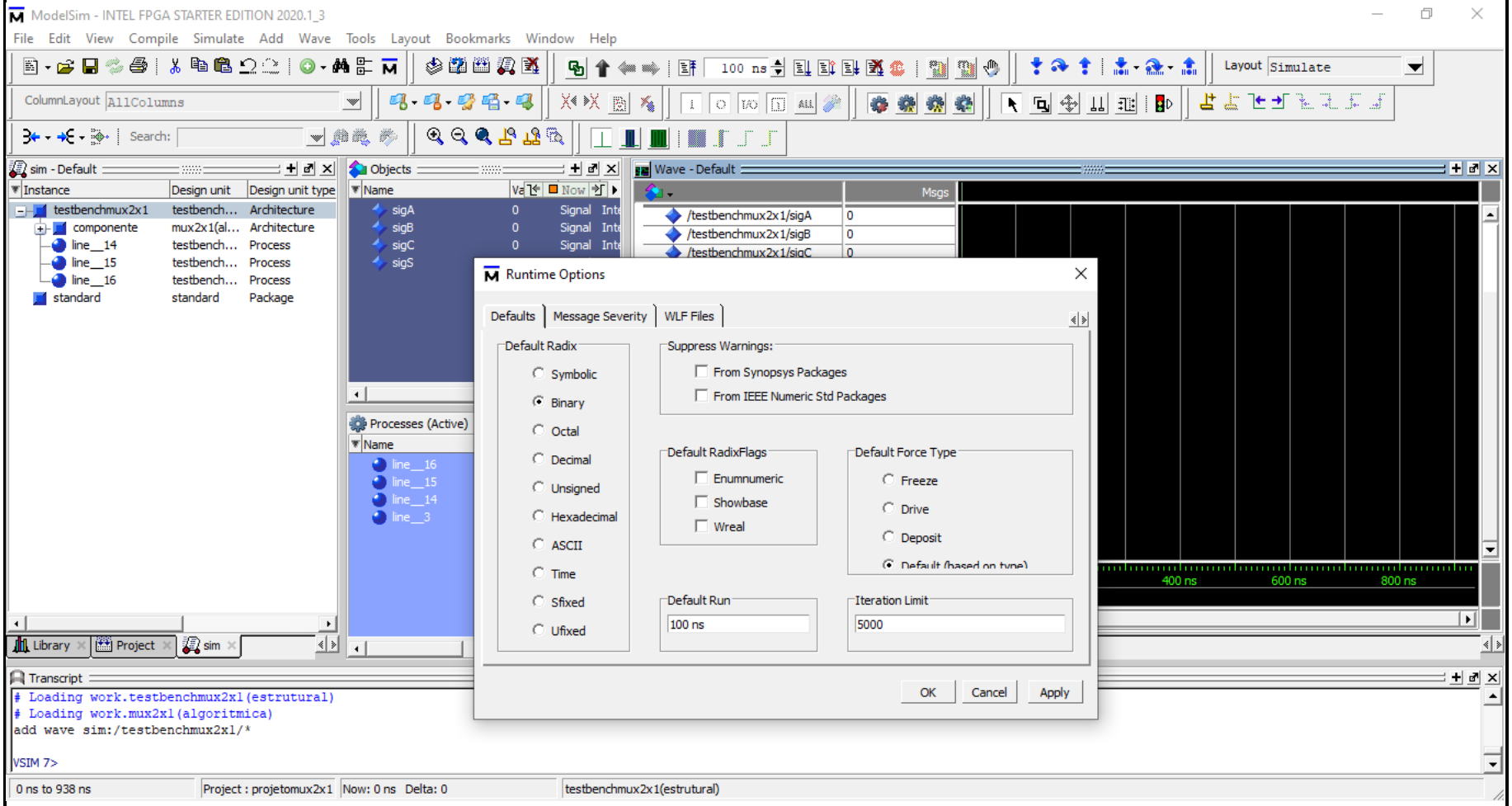
Na janela Wave, primeira coluna, aparecem os sinais descritos no `testbenchmux2x1`. Na segunda coluna, temos os valores binários dos respectivos sinais. A terceira coluna servirá para amostrar as formas de onda ao longo do tempo de simulação.


ModelSim - INTEL FPGA STARTER EDITION 2020.1_3
File Edit View Compile Simulate Add Wave Tools Layout Bookmarks Window Help
sim - Default
Instance Design unit Design unit type
testbenchmux2x1 testbench... Architecture
componente mux2x1(al... Architecture
line__14 testbench... Process
line__15 testbench... Process
line__16 testbench... Process
standard standard Package
Objects
Name
sigA 0 Signal
sigB 0 Signal
sigC 0 Signal
sigS 0 Signal
Processes (Active)
Name Type (filtered)
line__16 VHDL Process
line__15 VHDL Process
line__14 VHDL Process
line__3 VHDL Process
Wave - Default
Msgs
/testbenchmux2x1/sigA 0
/testbenchmux2x1/sigB 0
/testbenchmux2x1/sigC 0
/testbenchmux2x1/sigS 0
Now 0 ns
Cursor 1 0 ns
200 ns 400 ns 600 ns 800 ns
Library Project sim
Wave mux2x1algoritmica testbenchmux2x1 mux2x1
Transcript
Loading work.testbenchmux2x1(estrutural)
Loading work.mux2x1(algoritmica)
add wave sim:/testbenchmux2x1/*
VSIM 3>

- Simulate>Runtime Options
- Selecione Binary
- Default Run = 100 ns

Observe se o campo Iteration Limit é diferente de 0.

- OK

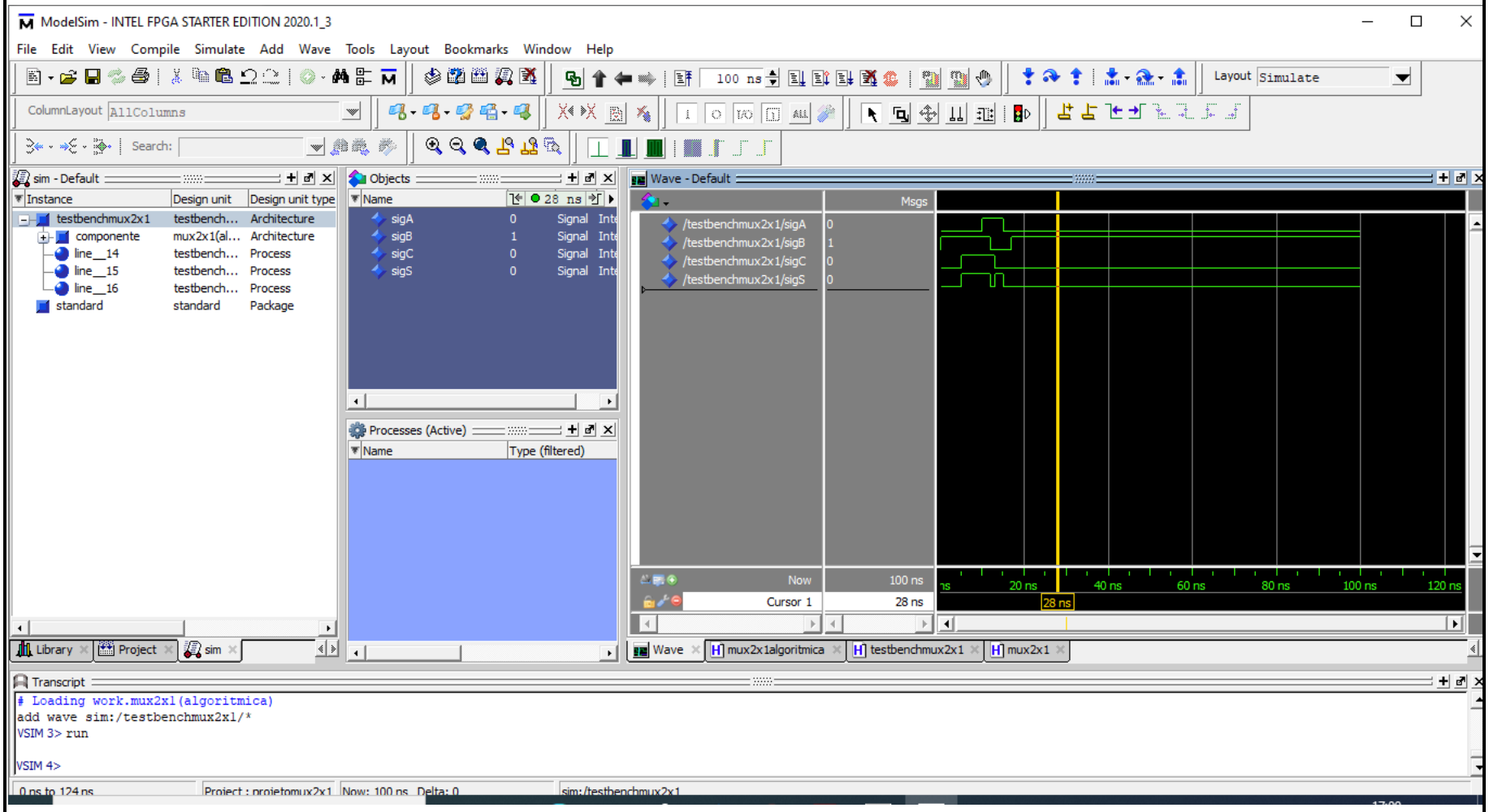

ModelSim - INTEL FPGA STARTER EDITION 2020.1_3
File Edit View Compile Simulate Add Wave Tools Layout Bookmarks Window Help
sim - Default
Instance Design unit Design unit type
testbenchmux2x1 testbench... Architecture
componente mux2x1(al... Architecture
line__14 testbench... Process
line__15 testbench... Process
line__16 testbench... Process
standard standard Package
Objects
Name
sigA 0 Signal
sigB 0 Signal
sigC 0 Signal
sigS
Processes (Active)
Name
line__16
line__15
line__14
line__3
Wave - Default
Msgs
/testbenchmux2x1/sigA 0
/testbenchmux2x1/sigB 0
/testbenchmux2x1/sigC 0
400 ns 600 ns 800 ns
Runtime Options
Defaults Message Severity WLF Files
Default Radix
Symbolic
Binary
Octal
Decimal
Unsigned
Hexadecimal
ASCII
Time
Sfixed
Ufixed
Suppress Warnings:
From Synopsys Packages
From IEEE Numeric Std Packages
Default RadixFlags
Enumnumeric
Showbase
Wreal
Default Force Type
Freeze
Drive
Deposit
Default Run
100 ns
Iteration Limit
5000
OK Cancel Apply
Library Project sim
Transcript
Loading work.testbenchmux2x1(estrutural)
Loading work.mux2x1(algoritmica)
add wave sim:/testbenchmux2x1/*
VSIM 7>
0 ns to 938 ns
Project : projetomux2x1
Now: 0 ns Delta: 0
testbenchmux2x1(estrutural)

- Simulate>Run>Run 100

Selecione a terceira coluna da janela Wave e selecione a lupa +, localizada na guia de comandos, o que permitirá aumentar o grau de visão. Ao colocar o cursor (linha amarela vertical) sobre uma determinada posição (selecionando essa posição), os valores correspondentes aos sinais são amostrados na segunda coluna.

Observe que as formas de onda terminam em 100 ns!

Verifique se, para cada combinação dos sinais de entrada, a resposta corresponde à especificação do componente!


ModelSim - INTEL FPGA STARTER EDITION 2020.1_3
File Edit View Compile Simulate Add Wave Tools Layout Bookmarks Window Help
sim - Default
Instance Design unit Design unit type
testbenchmux2x1 testbench... Architecture
componente mux2x1(al... Architecture
line__14 testbench... Process
line__15 testbench... Process
line__16 testbench... Process
standard standard Package
Objects
Name
sigA 0 Signal
sigB 1 Signal
sigC 0 Signal
sigS 0 Signal
Processes (Active)
Name Type (filtered)
Wave - Default
Msgs
/testbenchmux2x1/sigA 0
/testbenchmux2x1/sigB 1
/testbenchmux2x1/sigC 0
/testbenchmux2x1/sigS 0
Now 100 ns
Cursor 1 28 ns
20 ns 40 ns 60 ns 80 ns 100 ns 120 ns
28 ns
Library Project sim
Wave mux2x1algoritmica testbenchmux2x1 mux2x1
Transcript
Loading work.mux2x1(algoritmica)
add wave sim:/testbenchmux2x1/*
VSIM 3> run
VSIM 4>

Para terminar a simulação:

- Simulate>End Simulation

Para sair do Modelsim, selecione a janela Project e, em seguida:

- File>Close Project

Observe que a janela Project é fechada. Em seguida:

- File>Quit

É importante que se feche o projeto antes de sair do aplicativo. Isto faz com que o Modelsim feche os arquivos de simulação corretamente, a fim de abrí-los da próxima vez sem problemas!

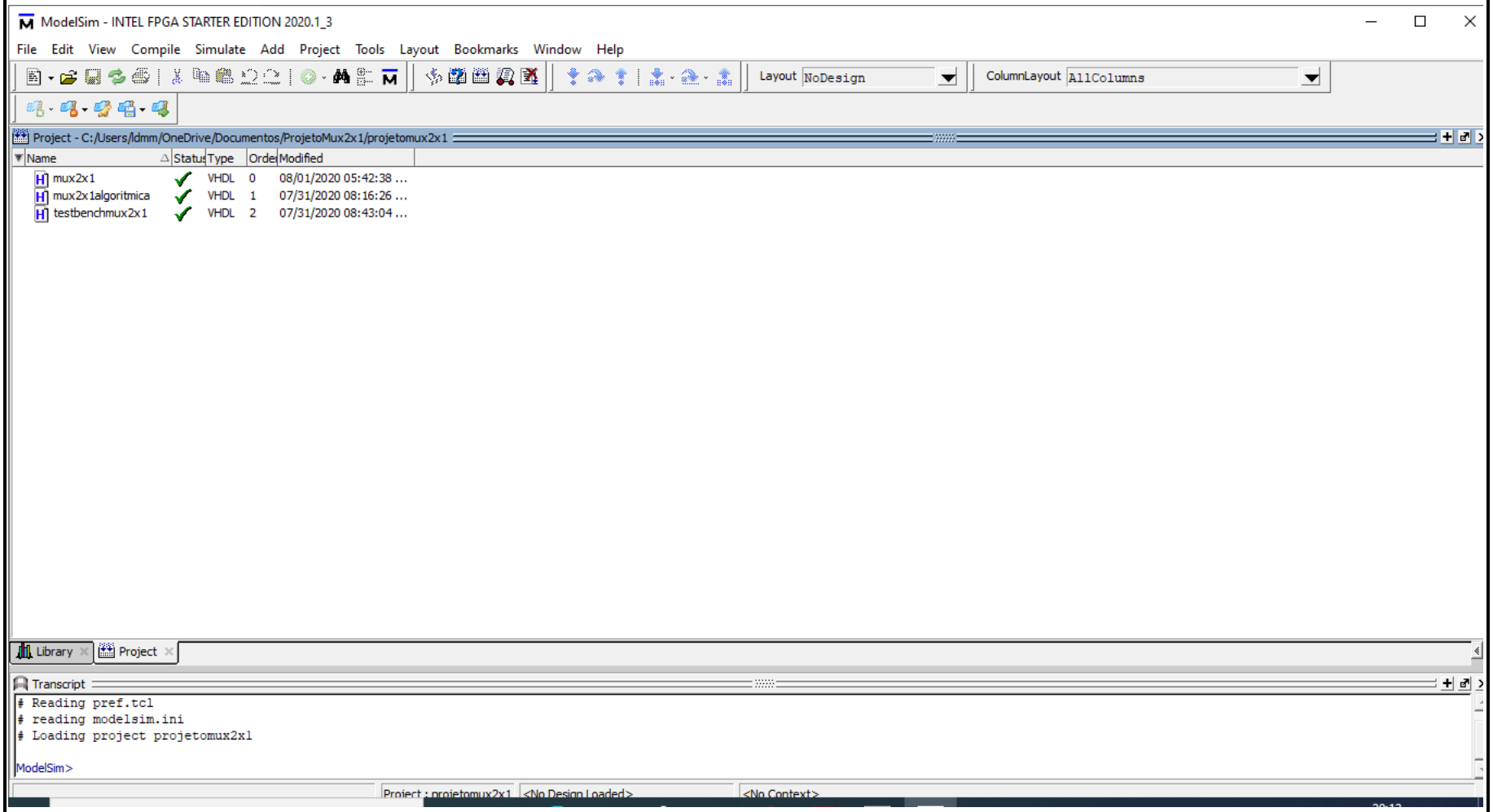
Vamos agora incorporar as outras descrições de arquitetura do `mux2x1`: `fluxodedados` e `estrutural`. Para tal, vamos executar o Modelsim novamente, só que, agora, ao invés de criar um projeto, vamos invocar o projeto que foi criado anteriormente: projetomux2x1.mpf.

- File>Open

Na janela Open File, vá para a pasta ProjetoMux2x1. Selecione Project Files, no campo ao lado de Nome, e, em seguida, selecione projetomux2x1.mpf.

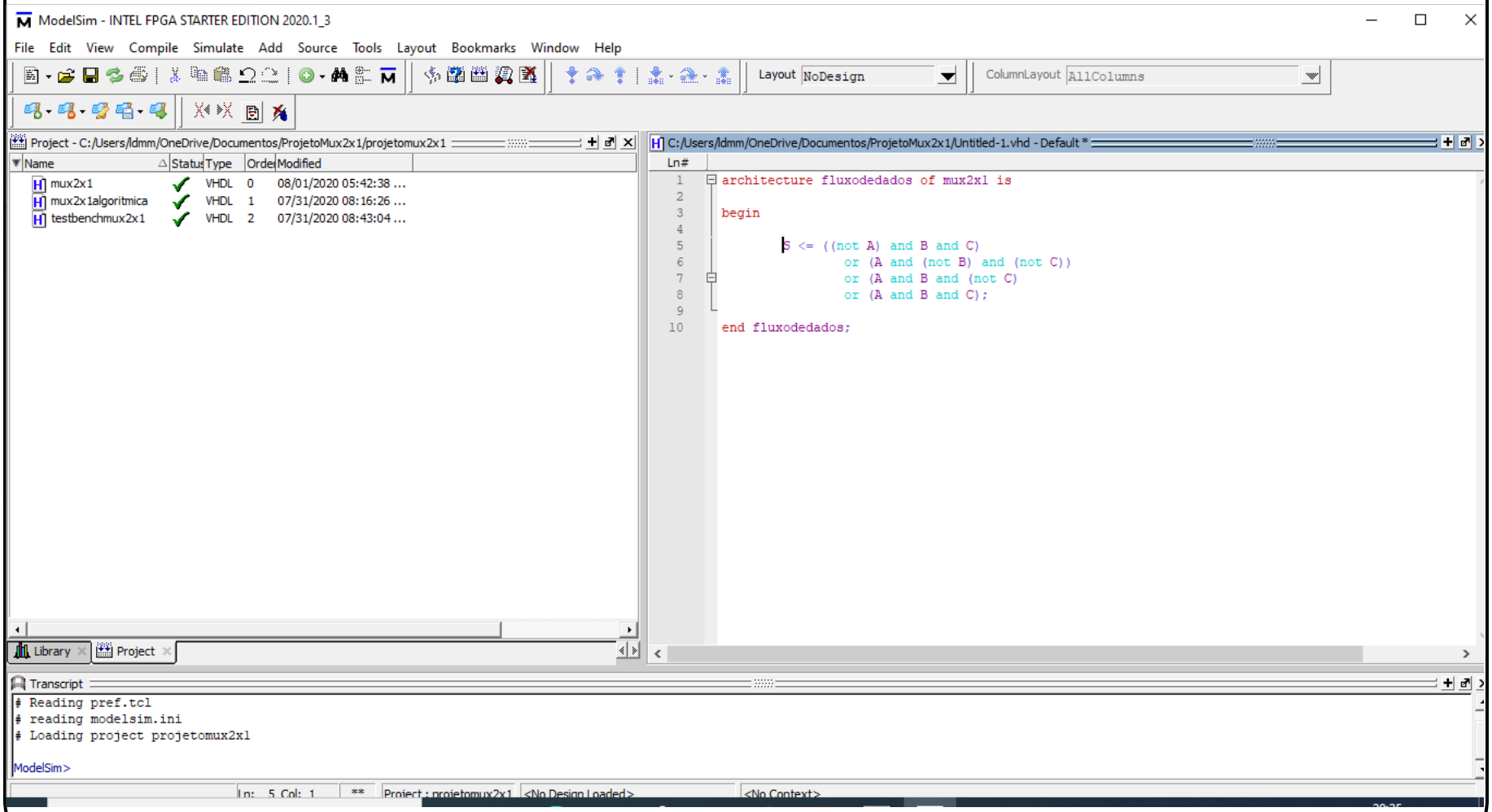
- Abrir

A janela Project torna-se disponível e os arquivos vinculados ao projeto projetomux2x1 são visualizados.



➢ File>New>Source>VHDL

Reproduzimos, então, o modelo em fluxo de dados já elaborado.



- File>Save As...

Na janela Save As, no campo Nome, inserir mux2x1fluxodedados.

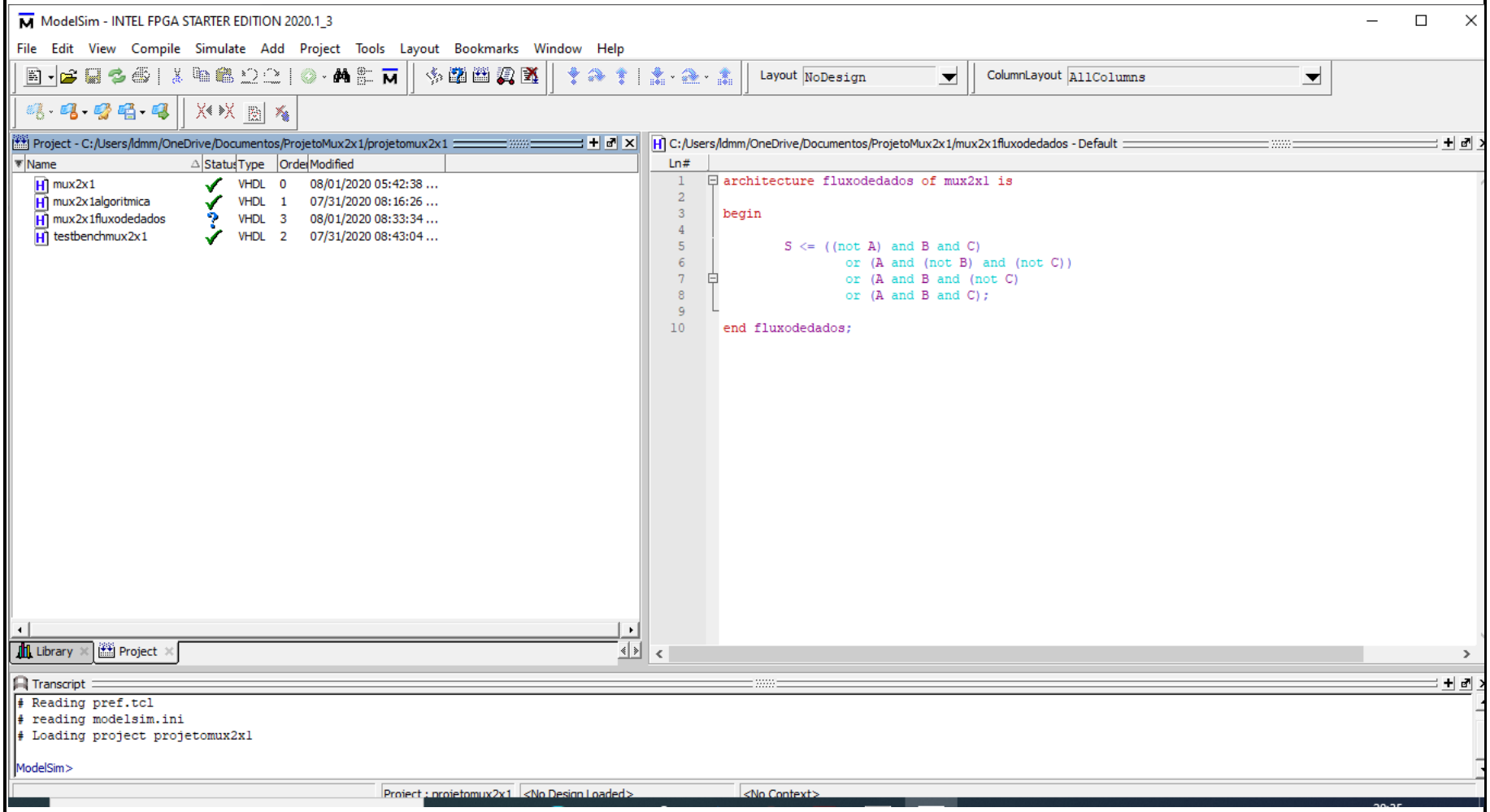
Selecione a janela Project e, em seguida:

- Project>Add to Project>Existing File...

Na janela Add file to Project, no campo File Name, inserir mux2x1fluxodedados.

- OK

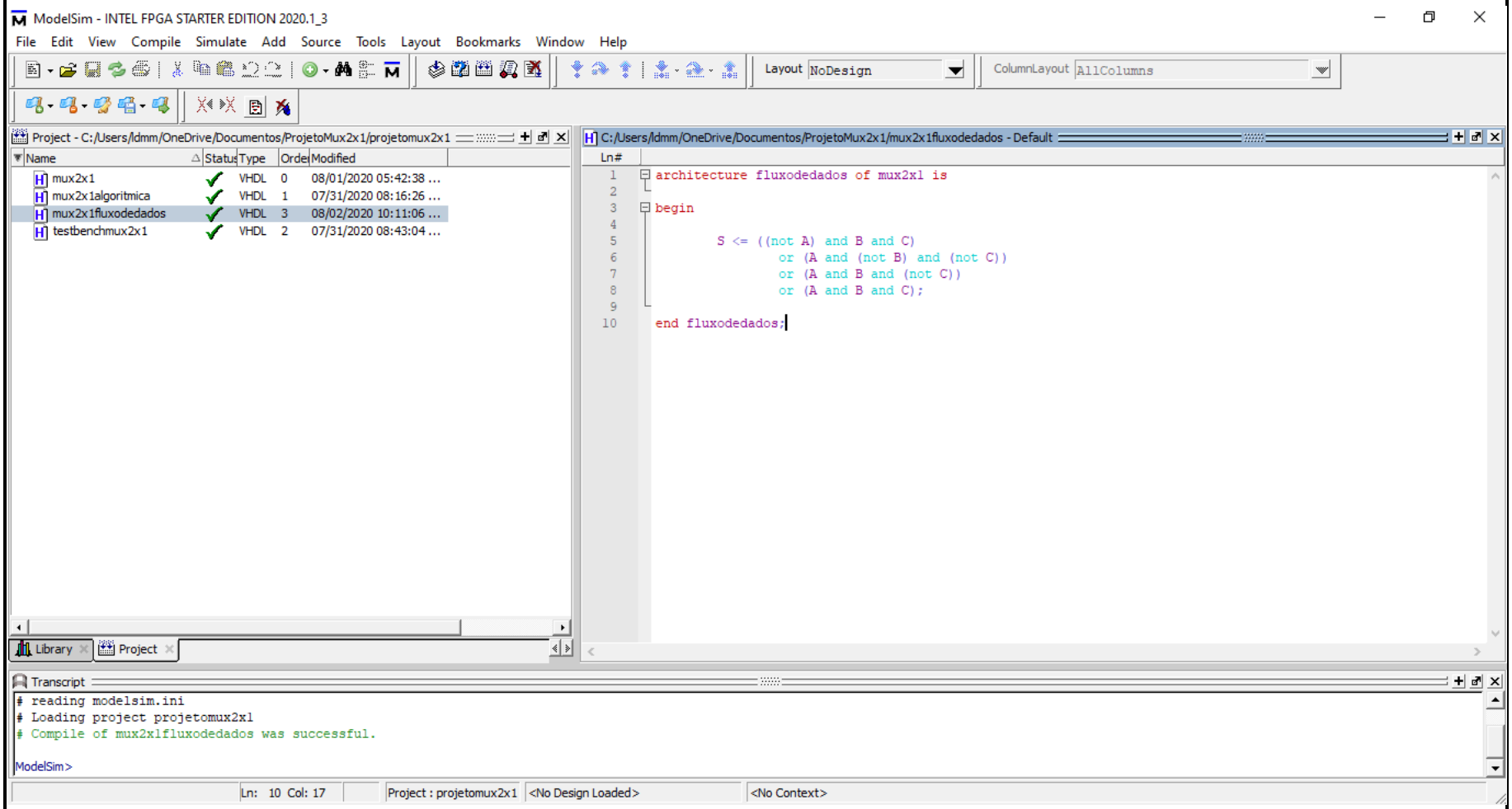
Na janela Project, o arquivo mux2x1fluxodedados torna-se disponível para ser compilado.



Selecione a janela Project, em seguida, selecione o arquivo mux2x1fluxodedados e, finalmente:

- Compile>Compile Selected

Na janela Transcript, aparece a mensagem "Compile of `mux2x1fluxodados` was successful" e, na janela Project, o atributo Status recebe ✓.


ModelSim - INTEL FPGA STARTER EDITION 2020.1_3
File Edit View Compile Simulate Add Source Tools Layout Bookmarks Window Help
Layout NoDesign
ColumnLayout AllColumns
Project - C:/Users/ldmm/OneDrive/Documentos/ProjetoMux2x1/projetomux2x1
Name Status Type Order Modified
mux2x1 VHDL 0 08/01/2020 05:42:38 ...
mux2x1algoritmica VHDL 1 07/31/2020 08:16:26 ...
mux2x1fluxodedados VHDL 3 08/02/2020 10:11:06 ...
testbenchmux2x1 VHDL 2 07/31/2020 08:43:04 ...
Library
Project
C:/Users/ldmm/OneDrive/Documentos/ProjetoMux2x1/mux2x1fluxodedados - Default
Ln#
1 architecture fluxodedados of mux2x1 is
2
3 begin
4
5 S <= ((not A) and B and C)
6 or (A and (not B) and (not C))
7 or (A and B and (not C))
8 or (A and B and C);
9
10 end fluxodedados;
Transcript
reading modelsim.ini
Loading project projetomux2x1
Compile of mux2x1fluxodedados was successful.
ModelSim>
Ln: 10 Col: 17
Project : projetomux2x1
<No Design Loaded>
<No Context>

O próximo passo é simular, utilizando o mesmo `testbenchmux2x1` já descrito e utilizado para a arquitetura algorítmica:

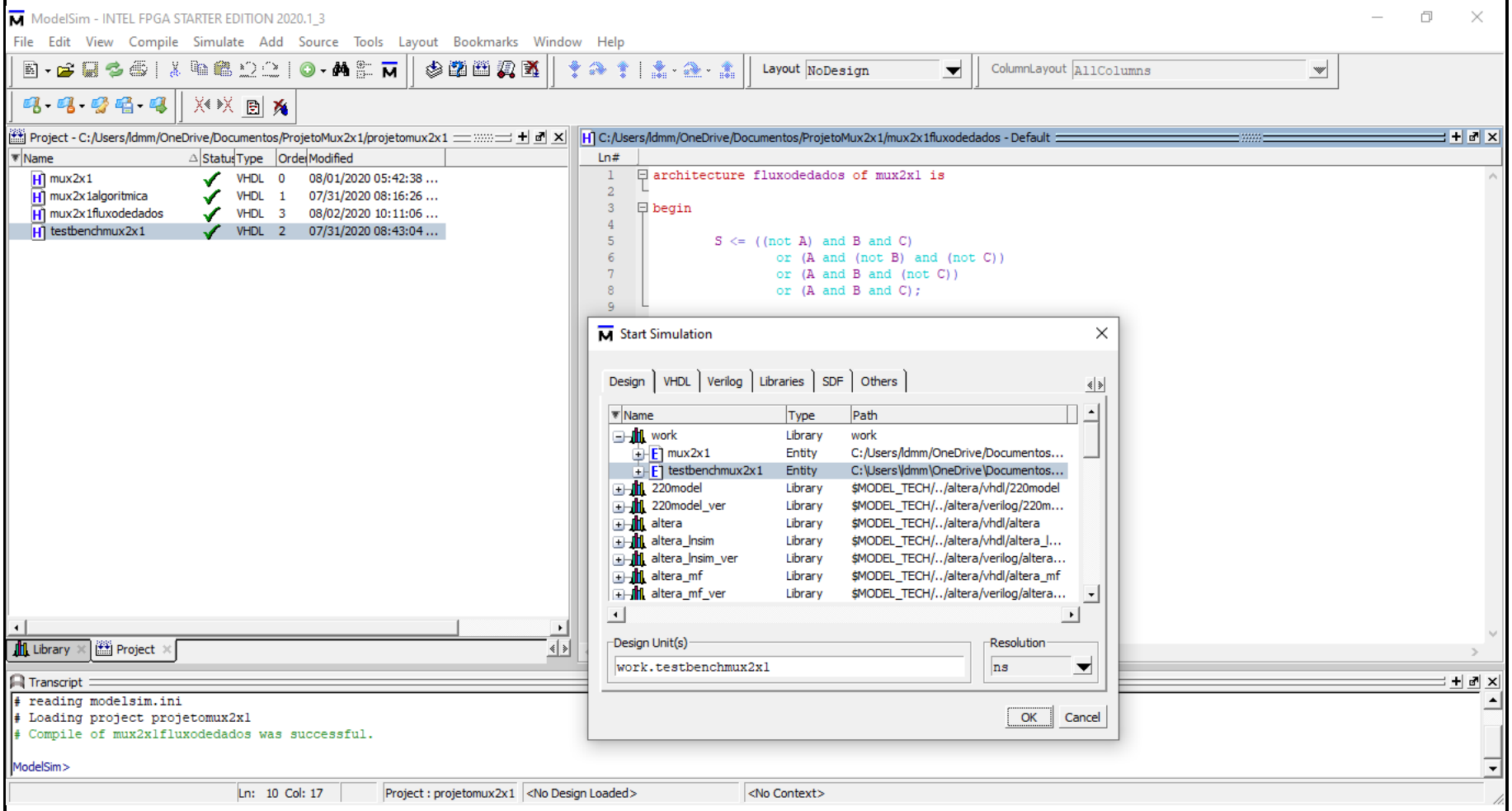
- Simulate>Start Simulation...

Na janela Start Simulation, selecione selecione o botão + ao lado de work e, em seguida, selecione `(E) testbenchmux2x1`:

- Resolution = ns

- OK

Aguarde a janela sim ser disponibilizada ao lado de Library e Project!


ModelSim - INTEL FPGA STARTER EDITION 2020.1_3
File Edit View Compile Simulate Add Source Tools Layout Bookmarks Window Help
Layout NoDesign
ColumnLayout AllColumns
Project - C:/Users/ldmm/OneDrive/Documentos/ProjetoMux2x1/projetomux2x1
Name Status Type Order Modified
mux2x1 VHDL 0 08/01/2020 05:42:38 ...
mux2x1algoritmica VHDL 1 07/31/2020 08:16:26 ...
mux2x1fluxodedados VHDL 3 08/02/2020 10:11:06 ...
testbenchmux2x1 VHDL 2 07/31/2020 08:43:04 ...
C:/Users/ldmm/OneDrive/Documentos/ProjetoMux2x1/mux2x1fluxodedados - Default
Ln#
1 architecture fluxodedados of mux2x1 is
2
3 begin
4
5 S <= ((not A) and B and C)
6 or (A and (not B) and (not C))
7 or (A and B and (not C))
8 or (A and B and C);
9
Start Simulation
Design VHDL Verilog Libraries SDF Others
Name Type Path
work Library work
mux2x1 Entity C:/Users/ldmm/OneDrive/Documentos...
testbenchmux2x1 Entity C:\Users\ldmm\OneDrive\Documentos...
220model Library $MODEL_TECH/../altera/vhdl/220model
220model_ver Library $MODEL_TECH/../altera/verilog/220m...
altera Library $MODEL_TECH/../altera/vhdl/altera
altera_lnsim Library $MODEL_TECH/../altera/vhdl/altera_l...
altera_lnsim_ver Library $MODEL_TECH/../altera/verilog/altera...
altera_mf Library $MODEL_TECH/../altera/vhdl/altera_mf
altera_mf_ver Library $MODEL_TECH/../altera/verilog/altera...
Design Unit(s)
work.testbenchmux2x1
Resolution
ns
OK
Cancel
Library
Project
Transcript
reading modelsim.ini
Loading project projetomux2x1
Compile of mux2x1fluxodedados was successful.
ModelSim>
Ln: 10 Col: 17
Project : projetomux2x1
<No Design Loaded>
<No Context>

A simulação do `mux2x1` para cada uma dessas arquiteturas utiliza o mesmo `testbenchmux2x1`, uma vez que este simplesmente fornece estímulos às entradas do `mux2x1`, independentemente da sua especificação interna (arquitetura).

Obviamente, o `mux2x1` deverá responder da mesma forma, independentemente da descrição interna.

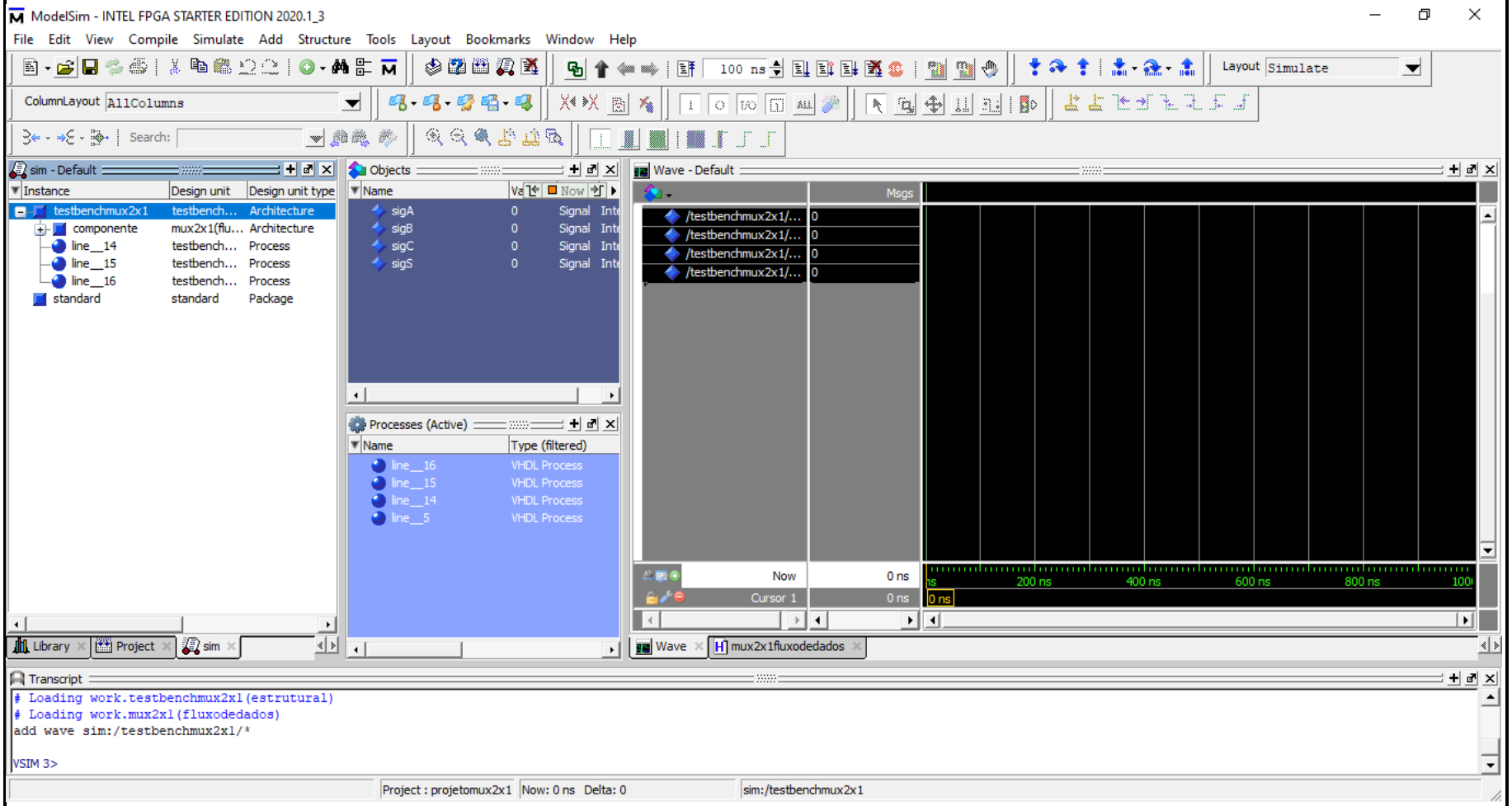
Por isso temos somente um `testbench` para cada entidade (`entity`) e não para cada arquitetura (`architecture`) de qualquer componente!

Na janela sim, selecione o `testbenchmux2x1`:

- Add>To Wave>All items in the region

Na janela Wave, aparecem os sinais descritos no `testbenchmux2x1` (primeira coluna). Na segunda coluna, temos os valores binários dos respectivos sinais. A terceira coluna servirá para amostrar as formas de onda ao longo do tempo de simulação.

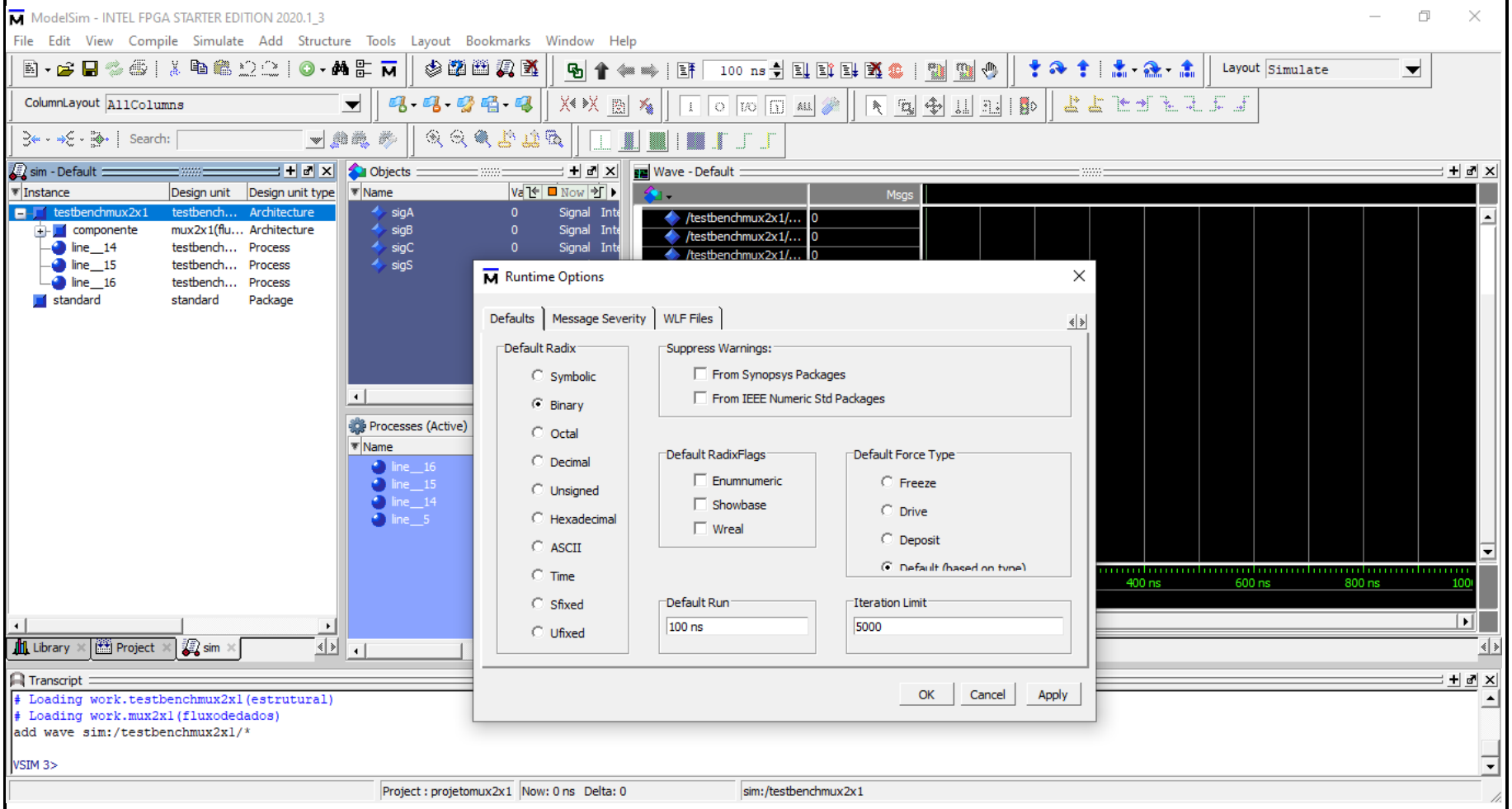
Observe que a janela é idêntica à da simulação da arquitetura algorítmica. Isto porque o componente é o mesmo (`mux2x1`), mas com uma outra descrição (fluxo de dados)!


ModelSim - INTEL FPGA STARTER EDITION 2020.1_3
File Edit View Compile Simulate Add Structure Tools Layout Bookmarks Window Help
sim - Default
Instance Design unit Design unit type
testbenchmux2x1 testbench... Architecture
componente mux2x1(flu... Architecture
line__14 testbench... Process
line__15 testbench... Process
line__16 testbench... Process
standard standard Package
Objects
Name
sigA 0 Signal
sigB 0 Signal
sigC 0 Signal
sigS 0 Signal
Processes (Active)
Name Type (filtered)
line__16 VHDL Process
line__15 VHDL Process
line__14 VHDL Process
line__5 VHDL Process
Wave - Default
Msgs
/testbenchmux2x1/... 0
/testbenchmux2x1/... 0
/testbenchmux2x1/... 0
/testbenchmux2x1/... 0
Now 0 ns
Cursor 1 0 ns
200 ns 400 ns 600 ns 800 ns
0 ns
Library Project sim
Wave mux2x1fluxodedados
Transcript
Loading work.testbenchmux2x1(estrutural)
Loading work.mux2x1(fluxodedados)
add wave sim:/testbenchmux2x1/*
VSIM 3>
Project : projetomux2x1 Now: 0 ns Delta: 0 sim:/testbenchmux2x1

- Simulate>Runtime Options
- Selecione Binary
- Default Run = 100 ns

Observe se o campo Iteration Limit é diferente de 0.

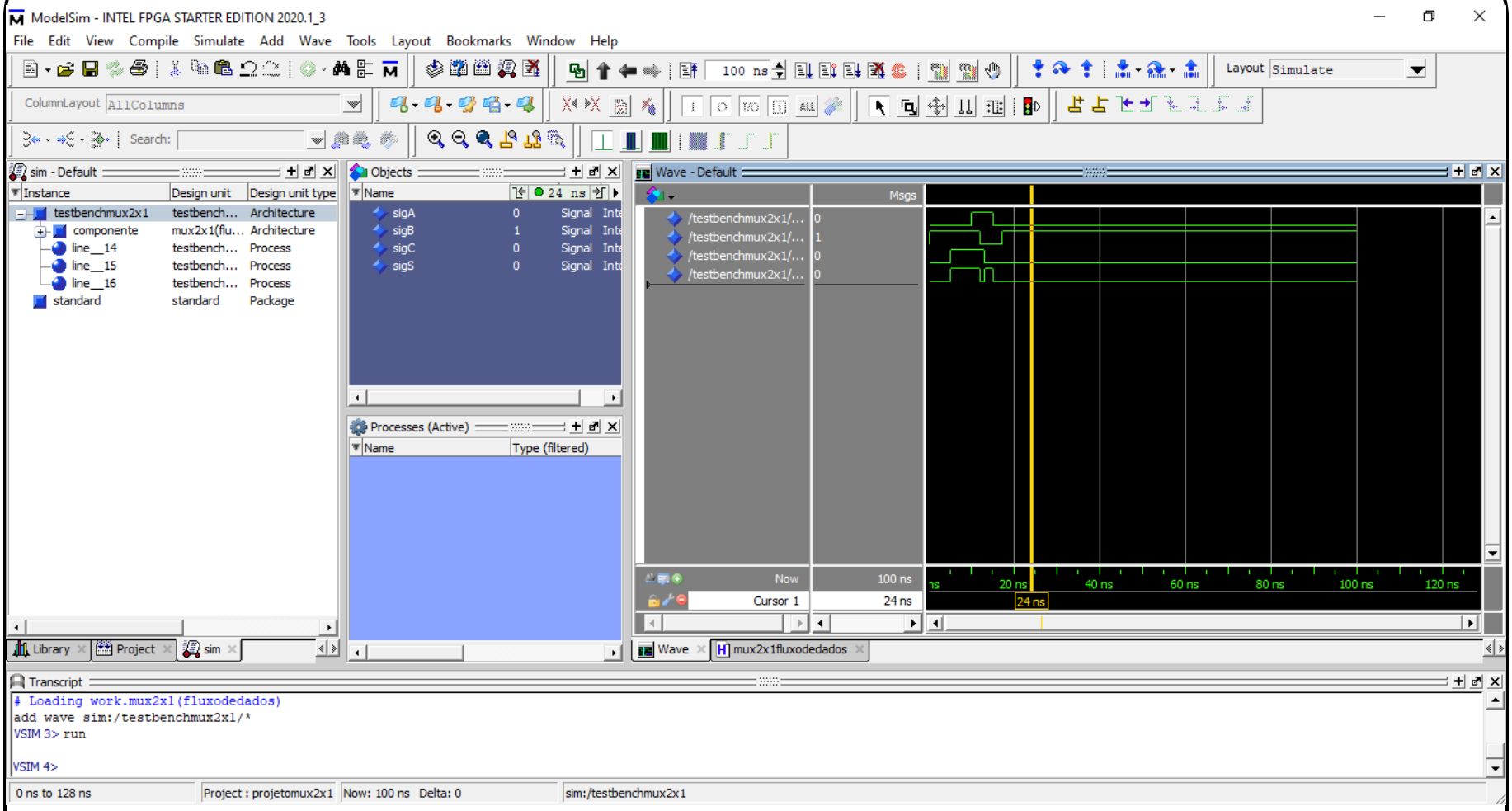
- OK


ModelSim - INTEL FPGA STARTER EDITION 2020.1_3
File Edit View Compile Simulate Add Structure Tools Layout Bookmarks Window Help
sim - Default
Instance Design unit Design unit type
testbenchmux2x1 testbench... Architecture
componente mux2x1(flu... Architecture
line__14 testbench... Process
line__15 testbench... Process
line__16 testbench... Process
standard standard Package
Objects
Name
sigA 0 Signal
sigB 0 Signal
sigC 0 Signal
sigS
Processes (Active)
Name
line__16
line__15
line__14
line__5
Wave - Default
Msgs
/testbenchmux2x1/... 0
/testbenchmux2x1/... 0
/testbenchmux2x1/... 0
400 ns 600 ns 800 ns
Runtime Options
Defaults Message Severity WLF Files
Default Radix
Symbolic
Binary
Octal
Decimal
Unsigned
Hexadecimal
ASCII
Time
Sfixed
Ufixed
Suppress Warnings:
From Synopsys Packages
From IEEE Numeric Std Packages
Default RadixFlags
Enumnumeric
Showbase
Wreal
Default Force Type
Freeze
Drive
Deposit
Default Run
100 ns
Iteration Limit
5000
OK Cancel Apply
Library Project sim
Transcript
Loading work.testbenchmux2x1(estrutural)
Loading work.mux2x1(fluxodedados)
add wave sim:/testbenchmux2x1/*
VSIM 3>
Project : projetomux2x1 Now: 0 ns Delta: 0 sim:/testbenchmux2x1

- Simulate>Run>Run 100

Selecione a terceira coluna da janela Wave e selecione a lupa +, localizada na guia de comandos, o que permitirá aumentar o grau de visão. Ao colocar o cursor (linha amarela vertical) sobre uma determinada posição (selecionando essa posição), os valores correspondentes aos sinais são amostrados na segunda coluna.

Observe que as formas de onda terminam em 100 ns!



Observe que o resultado é idêntico ao obtido a partir da simulação da descrição algorítmica!

Para terminar a simulação:

- Simulate>End Simulation

Para sair do Modelsim, selecione a janela Project e, em seguida:

- File>Close Project

Observe que a janela Project é fechada. Em seguida:

- File>Quit

É importante que se feche o projeto antes de sair do aplicativo. Isto faz com que o Modelsim feche os arquivos de simulação corretamente, a fim de abrí-los da próxima vez sem problemas!

Pode-se elaborar todo um projeto em um único arquivo, de onde o compilador extrai todos os elementos para a construção da biblioteca work. No entanto, qualquer alteração que seja necessária, fará com que todo o arquivo seja manipulado, o que pode ser inconveniente quando o tamanho do projeto for considerável. A utilização de um arquivo por componente facilita a identificação e manipulação do componente na pasta do projeto.

Outro aspecto é que a todo e qualquer componente, se faz corresponder uma única entidade e, para cada entidade, pode-se associar mais de uma arquitetura. No caso do mux2x1, elaboramos uma algorítmica, uma fluxo de dados e uma estrutural. No entanto, só temos uma única entidade. O nome da arquitetura é genérico e não está associado a qualquer modelo em especial.