

# Приложение Platform Designer

# Приложение Platform Designer

## Часть 4

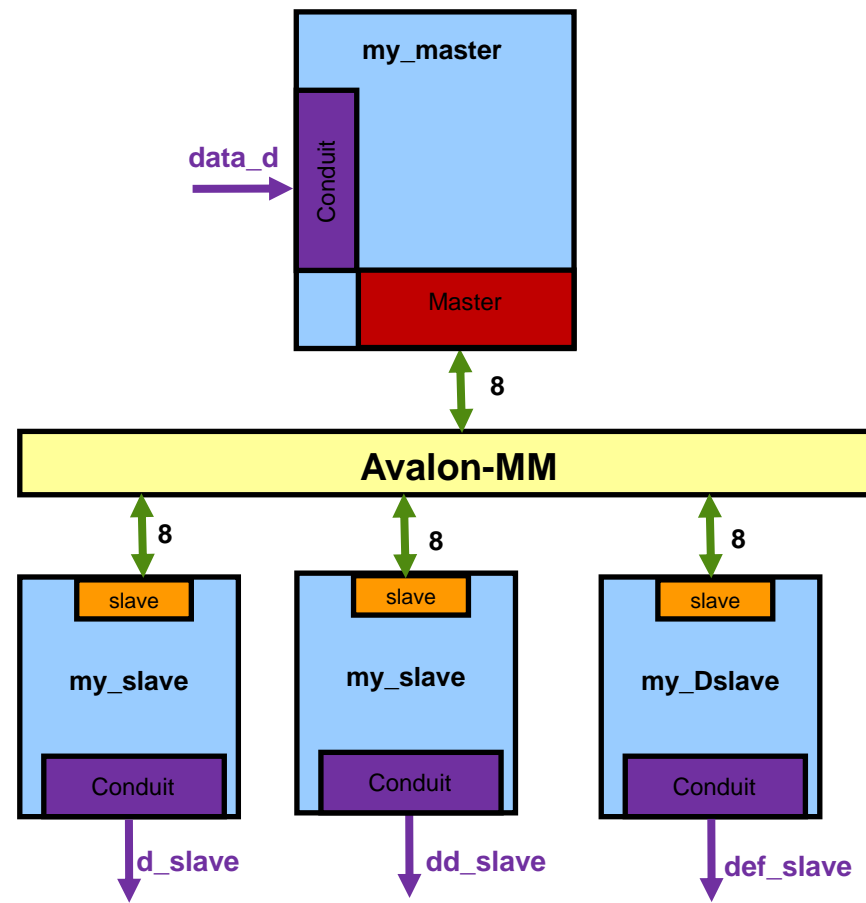
Создание и интеграция в PD  
пользовательских компонентов



# Лабораторная 4

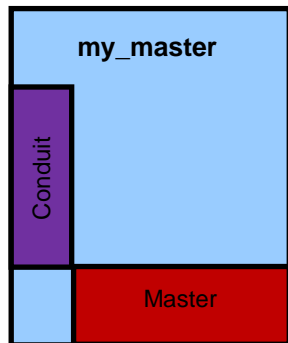


# Структура проекта





# Модуль my\_master (1)



- Осуществляет обращение (запись) по адресам от 0 до 255.
- Записываемые данные: текущий адрес + данные со входа coe\_c0\_DA.
- По сигналу rsi\_reset:
  - сброс автомата в исходное состояние.
  - сброс счетчика адреса.

```
1  `timescale 1 ns / 1 ns
2  module my_master (
3      //clock and reset
4      input bit      csi_clk,           // clock clk
5      input bit      rsi_reset,        // reset reset
6      //Avalon MM master
7      output bit [7:0] avm_m0_address, // MM master address
8      output bit      avm_m0_write,    // MM master write
9      output bit [7:0] avm_m0_writedata, // MM master writedata
10     input bit      avm_m0_waitrequest, // MM master waitrequest
11     //conduit
12     input bit [7:0] coe_c0_DA
13 );
14     typedef enum bit[1:0] {initSM, del1, wr1D, del2 } fsm_type;
15     fsm_type fsm_MM;
16     bit [7:0] cnt_intA;
17     always_ff @ (posedge csi_clk)
18     if (rsi_reset)
19     begin
20         fsm_MM      <= initSM;
21         cnt_intA    <= 8'd0;
22     end
23     else
24     case (fsm_MM)
25         initSM : fsm_MM <= del1;
26         del1   : fsm_MM <= wr1D;
27         wr1D   : if (avm_m0_waitrequest) fsm_MM <= wr1D;
28                 else fsm_MM <= del2;
29         del2   : begin fsm_MM <= initSM;
30                       cnt_intA <= cnt_intA + 8'd1; end
31     endcase
```

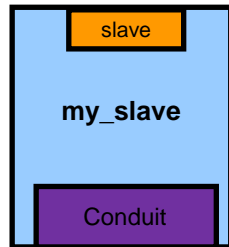


# Модуль my\_master()

```
32  always_comb
33  begin
34      case (fsm_MM)
35          wr1D:
36              begin
37                  avm_m0_address      = cnt_intA;
38                  avm_m0_write        = 1'd1;
39                  avm_m0_writedata    = cnt_intA+coe_c0_DA;
40              end
41          default
42              begin
43                  avm_m0_address      = 8'd255;
44                  avm_m0_write        = 1'd0;
45                  avm_m0_writedata    = 8'd255;
46              end
47      endcase
48  end
49  endmodule
```



# Модуль my\_slave

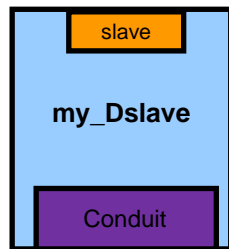


- Поддерживает обращения «запись»:
  - По сигналу `avs_s0_write` записывает данные со входа `avs_s0_writedata` во внутренний регистр `rg_DATA`
- Значение регистра `rg_DATA` передается на выход `coe_s0_Dout`.
- По сигналу `rsi_reset`:
  - Обнуляется внутренний регистр `rg_DATA`

```
1  `timescale 1 ns / 1 ns
2  module my_slave (
3  //clock and reset
4  input bit          csi_clk,          // clock clk
5  input bit          rsi_reset,        // reset reset
6  //MM Slave
7  input bit [7:0]    avs_s0_writedata, //MM Slave writedata
8  input bit          avs_s0_write,     //MM Slave write
9  output bit         avs_s0_waitrequest, //MM Slave waitrequest
10 //Conduit
11 output bit [7:0]    coe_s0_Dout
12 );
13 bit [7:0] rg_DATA;
14 assign avs_s0_waitrequest = 1'b0;
15 always_ff @(posedge csi_clk)
16     if (rsi_reset)      rg_DATA <= 8'd0;
17     else if (avs_s0_write) rg_DATA <= avs_s0_writedata;
18 assign coe_s0_Dout = rg_DATA;
19 endmodule
```



# Модуль my\_Dslave



- Поддерживает обращения «запись»:
  - По сигналу `avs_s0_write` значение счетчика числа обращений `cnt_` увеличивается на 1.
- Значение счетчика `cnt_` передается на выход. `coe_s0_Dout`.
- По сигналу `rsi_reset`:
  - Счетчик `cnt_` обнуляется.

```
1  `timescale 1 ns / 1 ns
2  module my_Dslave (
3      //clock and reset
4      input bit          csi_clk,          // clock clk
5      input bit          rsi_reset,        // reset reset
6      //MM Slave
7      input bit [7:0]    avs_s0_writedata, //MM Slave writedata
8      input bit          avs_s0_write,      //MM Slave write
9      output bit         avs_s0_waitrequest, //MM Slave waitrequest
10     //Conduit
11     output bit [7:0]    coe_s0_Dout
12 );
13     bit [7:0] cnt_;
14     assign avs_s0_waitrequest = 1'b0;
15     always_ff @(posedge csi_clk)
16     if (rsi_reset) cnt_ <= 8'd0;
17     else if (avs_s0_write) cnt_ <= cnt_ + 8'd1;
18     assign coe_s0_Dout = cnt_;
19 endmodule
```



# Порядок работы (1)

- В QP создайте проект
  - Имя проекта: lab\_PD4
  - Модуль верхнего уровня: lab\_PD4
  - Тип проекта: Empty Project
  - Файлы не добавляются
  - Микросхема: может быть любой
    - Плата DE1-SOC - 5CSEMA5F31C6N
    - Плата SoC Kit - 5CSXFC6D6F31
    - Плата MAX10\_NEEK - 10M50DAF484C6G
    - Плата **miniDilabCIV** (выбирается по умолчанию) - **EP4CE6E22C8**
    - Плата DE0-nano - EP4CE22F17C6
- EDA Tool Settings: Simulation => ModelSim Altera Starter Edition=> SystemVerilog


# Порядок работы (2)

- По представленным ранее образцам создайте описания модулей
  - my\_slave.
  - my\_Dslave.
  - my\_master.
- Осуществите их компиляцию в режиме Анализ и Синтез – убедитесь в том, что в модулях нет синтаксических ошибок.
- Для каждого модуля:
  - Приведите снимки экрана из RTL Viewer.
    - Проверьте их и поясните логику работы модуля.



# Порядок работы (3)

## В QR запустите приложение PD

- **Команда:** Tools => Platform Designer или иконка 
- **В PD:** сохраните систему под именем **lab\_DP4.qsys** в рабочей папке проекта
- **Убедитесь, что Ваша система выглядит так же, как показано на рисунке**

Qsys - lab\_PD4.qsys (C:\Intel\_trn\Q\_PD\lab\_PD4\lab\_PD4.qsys)

File Edit System Generate View Tools Help

IP Catalog System Contents Address Map Interconnect Requirements

System: lab\_PD4

Use	Co...	Name	Description	Export	Clock
<input checked="" type="checkbox"/>		clk_0	Clock Source		
		clk_in	Clock Input	clk	
		clk_in_reset	Reset Input	reset	
		clk	Clock Output	Double-click to	clk_0
		clk_reset	Reset Output	Double-click to	

The screenshot shows the Qsys Platform Designer interface. The title bar indicates the project is 'lab\_PD4.qsys' located at 'C:\Intel\_trn\Q\_PD\lab\_PD4\lab\_PD4.qsys'. The 'IP Catalog' pane on the left shows a project tree with 'lab\_PD4' highlighted by a green arrow. The 'System Contents' pane on the right displays a table of components for 'System: lab\_PD4'. The table lists components like 'clk\_0', 'clk\_in', 'clk\_in\_reset', 'clk', and 'clk\_reset' with their descriptions and export settings. The 'clk' component is highlighted, showing its export settings as 'Double-click to'.



# Порядок работы (4) *интеграция my\_master в PD*

- В окне закладки IP Catalog дважды щелкните строчку ***New Component...***
- В появившемся окне, на закладке Component Type введите данные о компоненте так, как показано на рисунке.
- Перейдите на закладку Files

Component Editor - my\_master\_hw.tcl\*

File Templates Beta View

Component Type	Block Symbol	Files	Parameters
About Component Type			
Name:	my_master		
Display name:	my_master		
Version:	1.0		
Group:	Training		
Description:			
Created by:			
Icon:			
Documentation:	Title	URL	

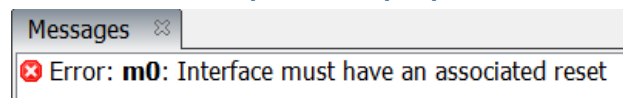


# Порядок работы (5) интеграция *my\_master* в PD

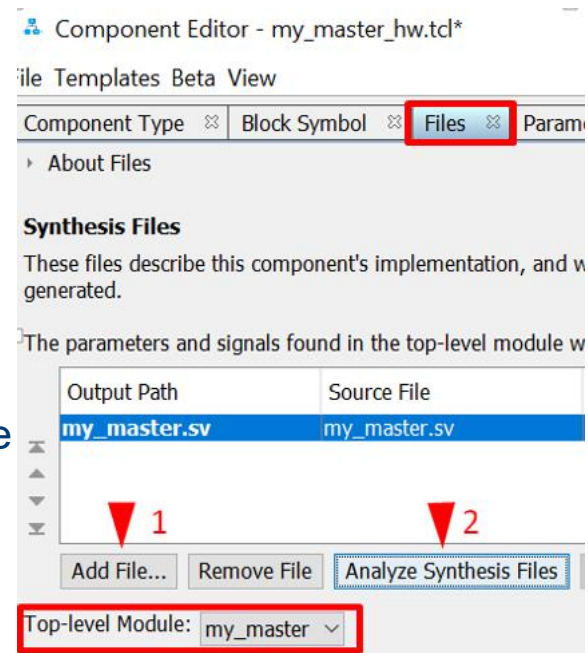
- На закладке Files, в разделе Synthesis Files, щелкните Add File...
- В появившемся окне выберите файл *my\_master.sv* и нажмите Open
- В разделе Synthesis Files, щелкните Analyze Synthesis Files
- При успешном окончании процедуры появится сообщение

✓ Analyzing Synthesis Files: completed successfully.

- при появлении ошибок – исправьте их в исходном файле
- указанное ниже сообщение об ошибке, появившееся в окне Message, пока можно проигнорировать



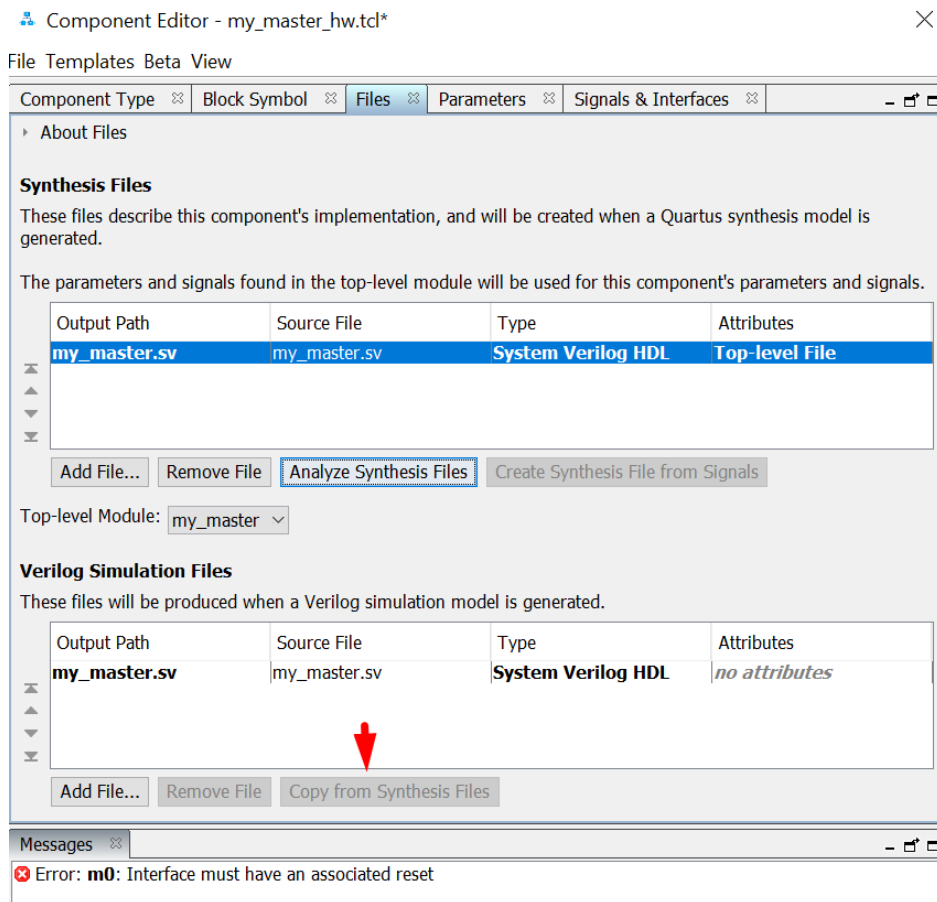
- Убедитесь в том, что в поле Top-Level Module указан модуль *my\_master*





# Порядок работы (6) интеграция *my\_master* в PD

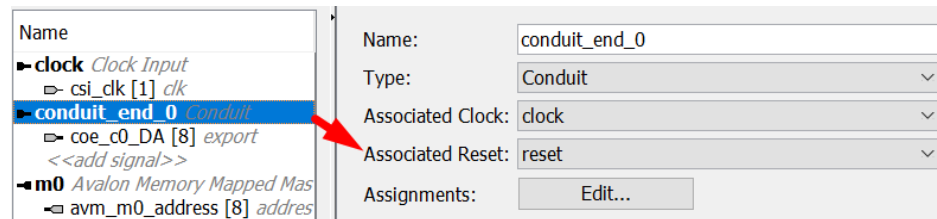
- На закладке Files, в разделе Verilog Simulation Files, щелкните Copy from Synthesis Files...
- Убедитесь в том, что закладка Files выглядит так же, как показано на рисунке
- Перейдите на закладку Signal & Interfaces



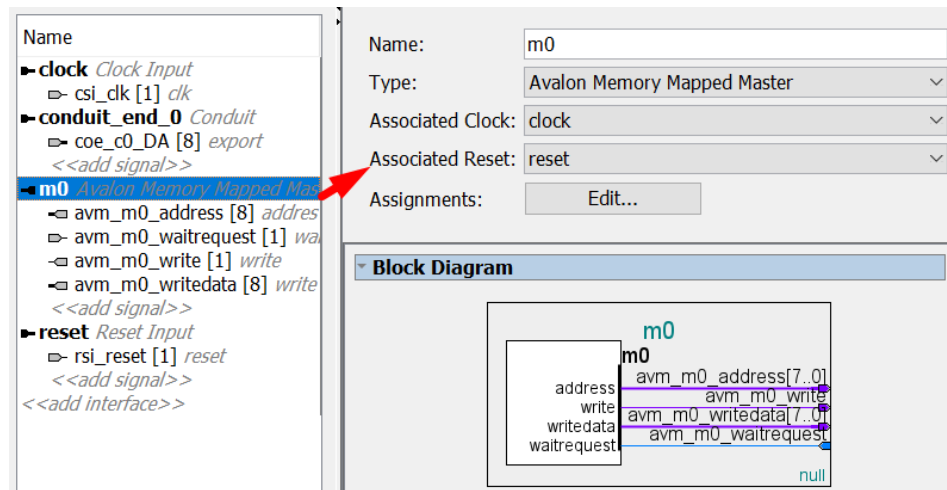


# Порядок работы (7) интеграция *my\_master* в PD

- На закладке Signal & Interfaces
  - Для интерфейса **Conduit\_end\_0** в поле Associated Reset выберите **reset**



- Для интерфейса **m0** в поле Associated Reset выберите **reset**

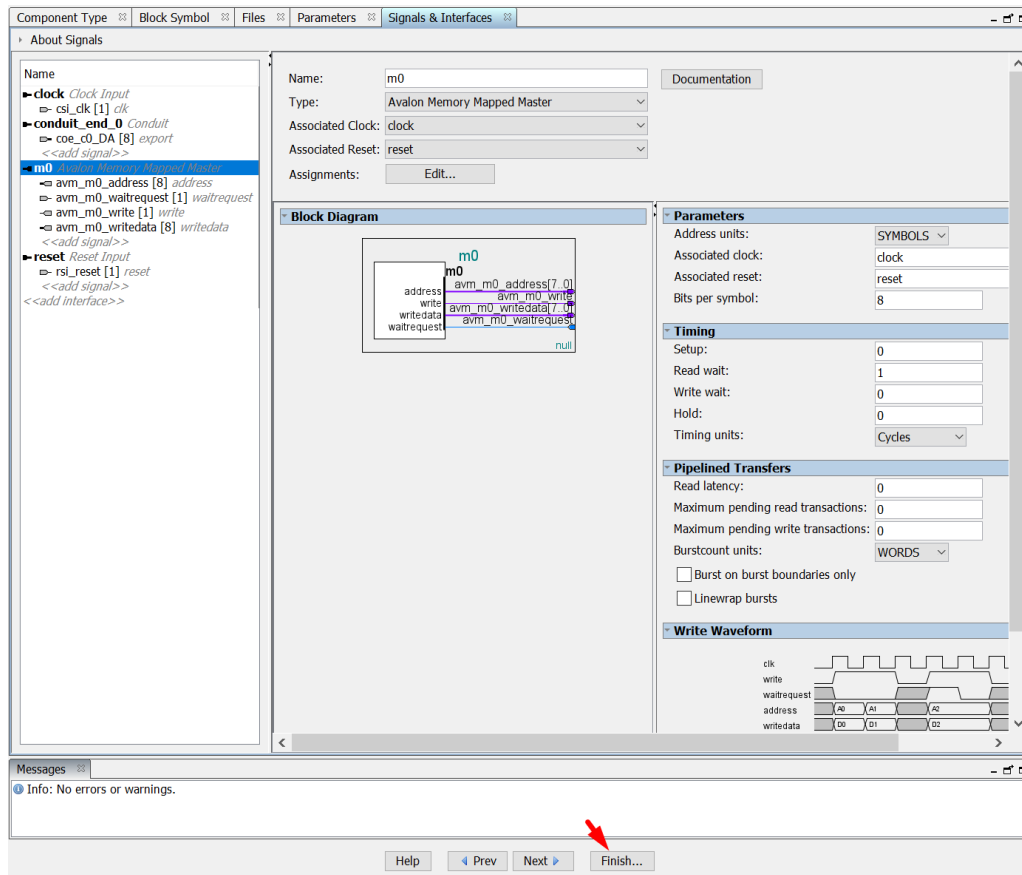


*После этого сообщение об ошибке, в окне Message должно исчезнуть.*



# Порядок работы (8) интеграция *my\_master* в PD

- Убедитесь в том, что закладка Signal & Interfaces выглядит так же, как показано на рисунке
- Нажмите кнопку Finish,
  - затем кнопку Yes, Save







# Порядок работы (9) интеграция *my\_master* в PD

- Убедитесь в том, что
  - в рабочей папке проекта появился файл ***my\_master\_hw.tcl***
  - Ваша система выглядит так же, как представленная на рисунке.

Use	Co...	Name	Description	Export	Clock
<input checked="" type="checkbox"/>		<b>clk_0</b>	Clock Source		
		clk_in	Clock Input	<b>clk</b>	<b><i>exported</i></b>
		clk_in_reset	Reset Input	<b>reset</b>	
		clk	Clock Output	<i>Double-click to</i>	clk_0
		clk_reset	Reset Output	<i>Double-click to</i>	



# Порядок работы (10) *интеграция my\_slave в PD*

- В окне закладки IP Catalog дважды щелкните строчку ***New Component...***
- В появившемся окне, на закладке Component Type введите данные о компоненте так, как показано на рисунке.
- Перейдите на закладку Files

Component Editor - my\_slave\_hw.tcl\*

File Templates Beta View

Component Type	Block Symbol	Files	Parameters				
▸ About Component Type							
Name:	my_slave						
Display name:	my_slave						
Version:	1.0						
Group:	Training						
Description:							
Created by:							
Icon:							
Documentation:	<table border="1"><thead><tr><th>Title</th><th>URL</th></tr></thead><tbody><tr><td colspan="2"></td></tr></tbody></table>			Title	URL		
Title	URL						

+ -



# Порядок работы (11) интеграция *my\_slave* в PD

- На закладке Files, в разделе Synthesis Files, щелкните Add File...

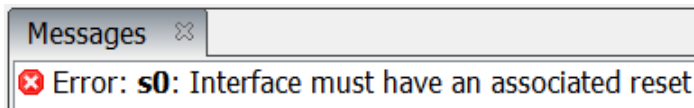
- В появившемся окне выберите файл *my\_slave.sv* и нажмите Open

- В разделе Synthesis Files, щелкните Analyze Synthesis Files

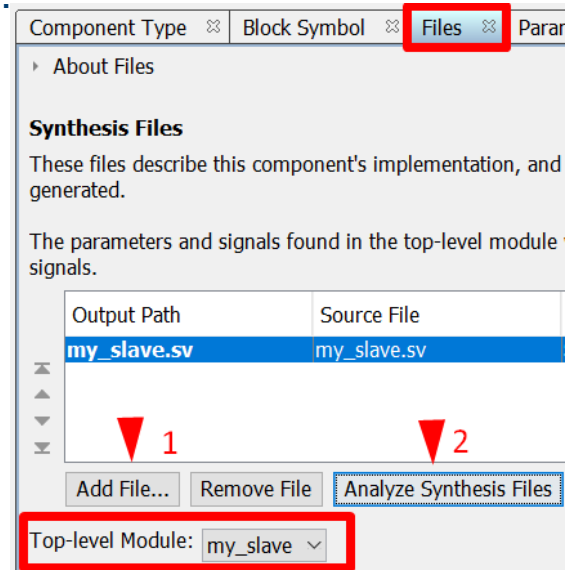
- При успешном окончании процедуры появится сообщение

✓ Analyzing Synthesis Files: completed successfully.

- при появлении ошибок – исправьте их в исходном файле
- указанное ниже сообщение об ошибке, появившееся в окне Message пока можно проигнорировать



- Убедитесь, что в поле Top-Level Module указан модуль *my\_slave*





# Порядок работы (12) интеграция *my\_slave* в PD

- На закладке Files, в разделе Verilog Simulation Files, щелкните Copy from Synthesis Files...
- Убедитесь в том, что закладка Files выглядит так же, как показано на рисунке
- Перейдите на закладку Signal & Interfaces

Component Editor - my\_slave\_hw.tcl\*

File Templates Beta View

Component Type ⓘ Block Symbol ⓘ Files ⓘ Parameters ⓘ Signals & Interfaces ⓘ

▸ About Files

## Synthesis Files

These files describe this component's implementation, and will be created when a Quartus synthesis is generated.

The parameters and signals found in the top-level module will be used for this component's parameters signals.

Output Path	Source File	Type	Attributes
my_slave.sv	my_slave.sv	System Verilog HDL	Top-level File

Add File...

Remove File

Analyze Synthesis Files

Create Synthesis File from Signals

Top-level Module: my\_slave ▾

## Verilog Simulation Files

These files will be produced when a Verilog simulation model is generated.

Output Path	Source File	Type	Attributes
my_slave.sv	my_slave.sv	System Verilog HDL	no attributes

Add File...

Remove File

Copy from Synthesis Files

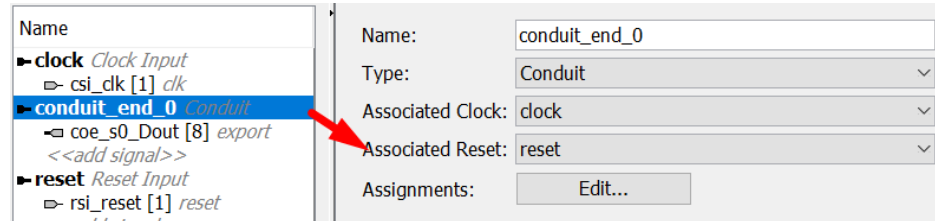
Messages ⓘ

✖ Error: s0: Interface must have an associated reset

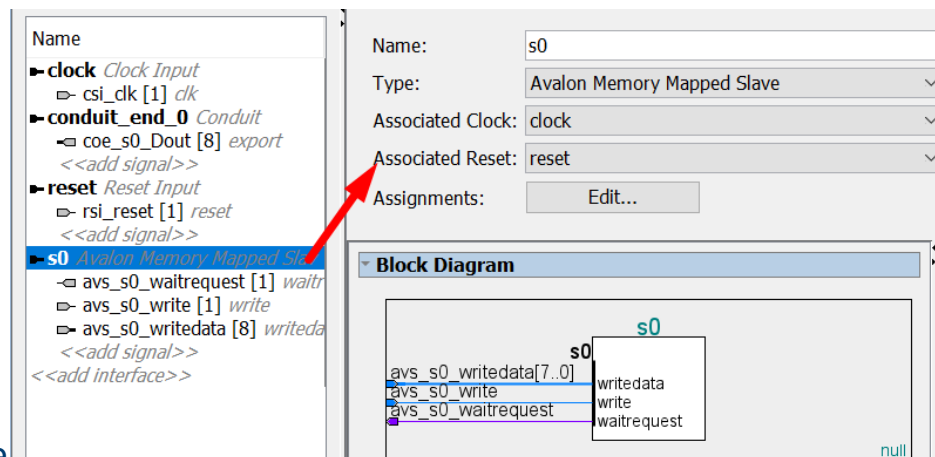


# Порядок работы (13) интеграция *my\_slave* в PD

- На закладке Signal & Interfaces
  - Для интерфейса **Conduit\_end\_0** в поле Associated Reset выберите **reset**



- Для интерфейса **s0** в поле Associated Reset выберите **reset**



После этого сообщение об ошибке, в окне Message должно исчезнуть.



# Порядок работы (14) интеграция *my\_slave* в PD

- Убедитесь в том, что закладка Signal & Interfaces выглядит так же, как показано на рисунке
- Нажмите кнопку Finish,
  - затем кнопку Yes, Save

The screenshot shows the 'Signals & Interfaces' configuration window in Xilinx Vivado. The 'About Signals' pane on the left lists the component's signals: `clock` (Clock Input), `cs1_clk [1]` (clk), `conduit_end_0` (Conduit), `coe_s0 Dout [8]` (export), `reset` (Reset Input), and `rs1_reset [1]` (reset). The 'Block Diagram' pane shows a block named 's0' with ports for `avs_s0_writedata [7:0]`, `avs_s0_write`, and `avs_s0_waitrequest`. The 'Parameters' pane on the right is configured with: Address units: WORDS, Associated clock: clock, Associated reset: reset, Bits per symbol: 8, Burstcount units: WORDS, and Explicit address span: 00000000000000000000. The 'Timing' section shows Setup: 0, Read wait: 1, Write wait: 0, and Hold: 0, with units in Cycles. The 'Pipelined Transfers' section shows Read latency: 0, Maximum pending read transactions: 0, and Maximum pending write transactions: 0. The 'Read Waveforms' section shows 'Read Not Supported'. The 'Write Waveforms' section shows a timing diagram for `clk`, `write`, `waitrequest`, and `writedata` across three data beats (D0, D1, D2). At the bottom, the 'Messages' pane shows 'Info: No errors or warnings.' and the 'Finish...' button is highlighted with a red arrow.



# Порядок работы (15) интеграция *my\_slave* в PD

- Убедитесь в том, что
  - в рабочей папке проекта появился файл ***my\_slave\_hw.tcl***
  - Ваша система выглядит так же, как представленная на рисунке.

Use	Co...	Name	Description	Export	Clock
<input checked="" type="checkbox"/>		<b>clk_0</b>	Clock Source		
		clk_in	Clock Input	<b>clk</b>	<b><i>exported</i></b>
		clk_in_reset	Reset Input	<b>reset</b>	
		clk	Clock Output	<i>Double-click to</i>	clk_0
		clk_reset	Reset Output	<i>Double-click to</i>	



# Порядок работы (16) интеграция *my\_Dslave* в PD

- В соответствии с рассмотренной выше процедурой интегрируйте в PD компонент *my\_Dslave*.
- Убедитесь в том, что
  - в рабочей папке проекта появился файл *my\_Dslave\_hw.tcl*
  - Ваша система выглядит так же, как представленная на рисунке.

Use	Co...	Name	Description	Export	Clock
<input checked="" type="checkbox"/>		<b>clk_0</b>	Clock Source		
		clk_in	Clock Input	<b>clk</b>	
		clk_in_reset	Reset Input	<b>reset</b>	<b><i>exported</i></b>
		clk	Clock Output	<i>Double-click to</i>	clk_0
		clk_reset	Reset Output	<i>Double-click to</i>	





# Порядок работы (17)

- Добавьте компоненты к системе
  - my\_master
  - my\_slave (два компонента)
  - my\_Dslave

*В появляющемся окне настройки каждого компонента нажмите Finish не изменяя настройки компонента - настройка компонентов будет осуществлена после их добавления к системе*

*При добавлении компонентов на закладке Messages будут появляться сообщения об ошибках. На данном этапе на них можно не обращать внимание.*



# Порядок работы (18)

- Убедитесь в том, что Ваша система выглядит так же, как представленная на рисунке
- Сохраните файл.

The screenshot displays the Vivado IDE interface with three main windows:

- IP Catalog:** Shows a project hierarchy with 'my\_Dslave', 'my\_master', and 'my\_slave' components. The 'Library' section lists various IP blocks like Basic Functions, DSP, Interface Protocols, etc.
- System Contents:** Displays the system hierarchy for 'lab\_PD4' with components like 'clk\_0', 'my\_master\_0', 'my\_slave\_0', 'my\_slave\_1', and 'my\_Dslave\_0'.
- Address Map:** Shows the address map for the selected component 'my\_Dslave\_0'. It lists the component's internal components and their addresses.

Use	Connec...	Name	Description	Export	Clock	Base
<input checked="" type="checkbox"/>		clk_0	Clock Source			
<input checked="" type="checkbox"/>		clk_in	Clock Input	clk	exported	clk_0
<input checked="" type="checkbox"/>		clk_in_reset	Reset Input	reset		
<input checked="" type="checkbox"/>		clk	Clock Output			
<input checked="" type="checkbox"/>		clk_reset	Reset Output			
<input checked="" type="checkbox"/>		my_master_0	my_master			
<input checked="" type="checkbox"/>		clock	Clock Input			
<input checked="" type="checkbox"/>		reset	Reset Input			
<input checked="" type="checkbox"/>		m0	Avalon Memory Mapped ...			
<input checked="" type="checkbox"/>		conduit_end_0	Conduit			
<input checked="" type="checkbox"/>		my_slave_0	my_slave			
<input checked="" type="checkbox"/>		clock	Clock Input			
<input checked="" type="checkbox"/>		reset	Reset Input			
<input checked="" type="checkbox"/>		s0	Avalon Memory Mapped ...			
<input checked="" type="checkbox"/>		conduit_end_0	Conduit			
<input checked="" type="checkbox"/>		my_slave_1	my_slave			
<input checked="" type="checkbox"/>		clock	Clock Input			
<input checked="" type="checkbox"/>		reset	Reset Input			
<input checked="" type="checkbox"/>		s0	Avalon Memory Mapped ...			
<input checked="" type="checkbox"/>		conduit_end_0	Conduit			
<input checked="" type="checkbox"/>		my_Dslave_0	my_Dslave			
<input checked="" type="checkbox"/>		clock	Clock Input			
<input checked="" type="checkbox"/>		reset	Reset Input			
<input checked="" type="checkbox"/>		s0	Avalon Memory Mapped ...			
<input checked="" type="checkbox"/>		conduit_end_0	Conduit			


Current filter:

Type	Path	Message
8 Errors		



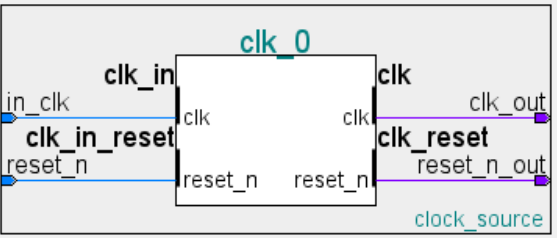
# Настройка компонента **clk\_0**

- Щелчком выберите **clk\_0**
- Нажмите правую клавишу мыши и выберите команду Edit
- В появившемся окне задайте Reset synchronous edges = Deassert

 Clock Source  
clock\_source

**Block Diagram**

☒ Show signals



**Parameters**

Clock frequency: 50000000 Hz

☒ Clock frequency is known

Reset synchronous edges: Deassert ▾



# Подключите тактовый сигнал clk

- На закладке System Contents щелчком выделите интерфейс **clk\_0.clk** (интерфейс clk компонента clk\_0)
- Выполните команду меню **View=>Connections**
- В появившемся окне закладки **Connections** выберите подключение ко всем тактовым входам
- Переключитесь на закладку **System Contents**
- Нажмите правую клавишу мыши
- Выберите команду **Filter=>Clock and Reset Interfaces**
- Убедитесь, что соединения выполнены - Ваша система выглядит так же, как представленная на рисунке

**System:** lab\_PD4    **Path:** clk\_0.clk

Connected to: clk\_0.clk

...	Connection	Clock Crossing
<input checked="" type="checkbox"/>	clk_0.clk/my_Dslave_0.clock	
<input checked="" type="checkbox"/>	clk_0.clk/my_master_0.clock	
<input checked="" type="checkbox"/>	clk_0.clk/my_slave_0.clock	
<input checked="" type="checkbox"/>	clk_0.clk/my_slave_1.clock	

**System:** lab\_PD4    **Path:** clk\_0.clk

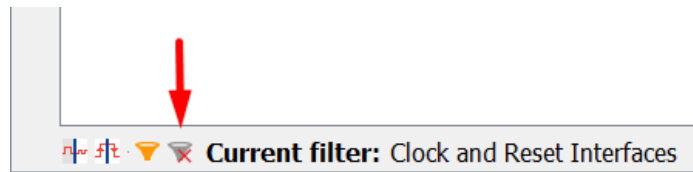
Use	Co...	Name	Description	Export	Clock	Base
<input checked="" type="checkbox"/>		<b>clk_0</b>	Clock Source			
		clk_in	Clock Input			
		clk_in_reset	Reset Input			
		<b>clk</b>	Clock Output	<b>clk</b>	<b>reset</b>	
				Double-click to	clk_0	
		clk_reset	Reset Output	Double-click to	clk_0	
<input checked="" type="checkbox"/>		<b>my_master_0</b>	my_master			
		clock	Clock Input	Double-click to	clk_0	
		reset	Reset Input	Double-click to	[clock]	
<input checked="" type="checkbox"/>		<b>my_slave_0</b>	my_slave			
		clock	Clock Input	Double-click to	clk_0	1'
		reset	Reset Input	Double-click to	[clock]	
<input checked="" type="checkbox"/>		<b>my_slave_1</b>	my_slave			
		clock	Clock Input	Double-click to	clk_0	1'
		reset	Reset Input	Double-click to	[clock]	
<input checked="" type="checkbox"/>		<b>my_Dslave_0</b>	my_Dslave			
		clock	Clock Input	Double-click to	clk_0	1'
		reset	Reset Input	Double-click to	[clock]	



# Подключите сигнал Reset

- На закладке System Contents выполните команду меню **System=>Create Global Reset Network**
- Убедитесь, что соединения выполнены - Ваша система выглядит так же, как представленная на рисунке
- Сбросьте фильтрацию – нажмите на иконку  в нижней части окна System Contents
- Сохраните файл

System: lab_PD4 Path: clk_0.clk						
Use	Co...	Name	Description	Export	Clock	Base
<input checked="" type="checkbox"/>		clk_0	Clock Source			
	└─	clk_in	Clock Input	clk	exported	
	└─	clk_in_reset	Reset Input	reset	[clk_in]	
	└─	clk	Clock Output	Double-click to	clk_0	
	└─	clk_reset	Reset Output	Double-click to	clk_0	
<input checked="" type="checkbox"/>		my_master_0	my_master			
	└─	clock	Clock Input	Double-click to	clk_0	
	└─	reset	Reset Input	Double-click to	[clock]	
<input checked="" type="checkbox"/>		my_slave_0	my_slave			
	└─	clock	Clock Input	Double-click to	clk_0	
	└─	reset	Reset Input	Double-click to	[clock]	
<input checked="" type="checkbox"/>		my_slave_1	my_slave			
	└─	clock	Clock Input	Double-click to	clk_0	
	└─	reset	Reset Input	Double-click to	[clock]	
<input checked="" type="checkbox"/>		my_Dslave_0	my_Dslave			
	└─	clock	Clock Input	Double-click to	clk_0	
	└─	reset	Reset Input	Double-click to	[clock]	





# Подключите Avalon-MM интерфейсы

- На закладке System Contents
  - щелчком выделите интерфейс **my\_master.m0**
  - Нажмите правую клавишу мыши и выберите команду **Filter=> Avalon-MM Interfaces**
  - В столбце Connections выполните подключения так, как показано на рисунке
- Сохраните файл

System: lab\_PD4 Path: my\_master\_0.m0

Use	Name	Description	Export	Clock	Base	End
<input checked="" type="checkbox"/>	my_master_0	my_master		[clock]		
<input checked="" type="checkbox"/>	m0	Avalon Memory Mapped ...	Double-click to	clk_0		
<input checked="" type="checkbox"/>	my_slave_0.s0	my_slave Avalon Memory Mapped ...	Double-click to	clk_0	0x0000	0x0000
<input checked="" type="checkbox"/>	my_slave_1.s0	my_slave Avalon Memory Mapped ...	Double-click to	clk_0	0x0000	0x0000
<input checked="" type="checkbox"/>	my_Dslave_0.s0	my_Dslave Avalon Memory Mapped ...	Double-click to	clk_0	0x0000	0x0000

Current filter: Avalon-MM Interfaces

Messages

Type	Path	Message
2 Errors		
	lab_PD4.my_master_0.m0	my_slave_1.s0 (0x0..0x0) overlaps my_slave_0.s0 (0x0..0x0)
	lab_PD4.my_master_0.m0	my_Dslave_0.s0 (0x0..0x0) overlaps my_slave_1.s0 (0x0..0x0)



# Назначьте Default Slave

- Щелкните правой клавишей мыши в поле заголовков столбцов (закладка Структура Системы)
- Выберите Show Default Slave Column
- Перетащите колонку Default Slave Column к колонке End
- Выберите check box напротив интерфейса my\_Dslave\_0.s0

Use	Name	Description	Export	Clock	Base	End	Default Slave
<input checked="" type="checkbox"/>	my_master_0	my_master		[clock]			
<input checked="" type="checkbox"/>	m0	Avalon Memory Mapped ...	Double-click	clk_0			
<input checked="" type="checkbox"/>	my_slave_0	my_slave		[clock]	0x0000		
<input checked="" type="checkbox"/>	s0	Avalon Memory Mapped ...	Double-click	clk_0			
<input checked="" type="checkbox"/>	my_slave_1	my_slave		[clock]	0x0000		
<input checked="" type="checkbox"/>	s0	Avalon Memory Mapped ...	Double-click	clk_0			
<input checked="" type="checkbox"/>	my_Dslave_0	my_Dslave		[clock]	0x0000		
<input checked="" type="checkbox"/>	s0	Avalon Memory Mapped ...	Double-click	clk_0			




Use	Name	Description	Export	Clock	Base	End	Default Slave
<input checked="" type="checkbox"/>	my_master_0	my_master		[clock]			
<input checked="" type="checkbox"/>	m0	Avalon Memory Mapped ...	Double-click	clk_0			
<input checked="" type="checkbox"/>	my_slave_0	my_slave		[clock]	0x0000	0x0000	<input type="checkbox"/>
<input checked="" type="checkbox"/>	s0	Avalon Memory Mapped ...	Double-click	clk_0			
<input checked="" type="checkbox"/>	my_slave_1	my_slave		[clock]	0x0000	0x0000	<input type="checkbox"/>
<input checked="" type="checkbox"/>	s0	Avalon Memory Mapped ...	Double-click	clk_0			
<input checked="" type="checkbox"/>	my_Dslave_0	my_Dslave		[clock]	0x0000	0x0000	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	s0	Avalon Memory Mapped ...	Double-click	clk_0			




Current filter: Avalon-MM Interfaces

Type	Path	Message
1 Error		
	lab_PD4.my_master_0.m0	my_slave_1.s0 (0x0..0x0) overlaps my_slave_0.s0 (0x0..0x0)



# Назначьте базовые адреса ведомым Avalon-MM

- Назначьте базовые адреса (дважды щелкните в поле *Base адрес* и введите значение адреса):
  - Компоненту `my_slave_0.s0` назначьте базовый адрес = 1
  - Компоненту `my_slave_1.s0` назначьте базовый адрес = 2
- Зафиксируйте адреса – нажмите на символ 
- Убедитесь, что адреса назначены правильно - Ваша система выглядит так же, как представленная на рисунке.
- Сохраните файл. 
- Сбросьте фильтрацию – нажмите на иконку  в нижней части окна System Contents.

Use	...	Name	Description	Export	Clock	Base	End	Default Slave
<input checked="" type="checkbox"/>		my_master_0	my_master		[clock]			
	└─	m0	Avalon Memory Mapped ...	Double-click	clk_0			
<input checked="" type="checkbox"/>	└─	my_slave_0	my_slave		[clock]			
	└─	s0	Avalon Memory Mapped ...	Double-click	clk_0	 0x0001	0x0001	<input type="checkbox"/>
<input checked="" type="checkbox"/>	└─	my_slave_1	my_slave		[clock]			
	└─	s0	Avalon Memory Mapped ...	Double-click	clk_0	 0x0002	0x0002	<input type="checkbox"/>
<input checked="" type="checkbox"/>	└─	my_Dslave_0	my_Dslave		[clock]			
	└─	s0	Avalon Memory Mapped ...	Double-click	clk_0	 default	default	<input checked="" type="checkbox"/>





# Экспортируйте выводы

- На закладке System Contents в столбце Export задайте имена экспортируемым выводам так, как показано на рисунке

Use	Connec...	Name	Description	Export	Clock	Base	End	Default Slave
<input checked="" type="checkbox"/>		<div>clk_0</div> <div>clk_in</div> <div>clk_in_reset</div> <div>clk</div> <div>clk_reset</div>	<div>Clock Source</div> <div>Clock Input</div> <div>Reset Input</div> <div>Clock Output</div> <div>Reset Output</div>	<div>clk</div> <div>reset</div> <div>Double-click to export</div> <div>Double-click to export</div>	<div>exported</div> <div>[clk_in]</div> <div>clk_0</div> <div>clk_0</div>			
<input checked="" type="checkbox"/>		<div>my_master_0</div> <div>clock</div> <div>reset</div> <div>m0</div>	<div>my_master</div> <div>Clock Input</div> <div>Reset Input</div> <div>Avalon Memory Mapped ...</div>	<div>Double-click to export</div> <div>Double-click to export</div> <div>Double-click to export</div>	<div>clk_0</div> <div>[clock]</div> <div>[clock]</div>			
<input checked="" type="checkbox"/>		conduit_end_0	Conduit	data_d	[clock]			
<input checked="" type="checkbox"/>		<div>my_slave_0</div> <div>clock</div> <div>reset</div> <div>s0</div>	<div>my_slave</div> <div>Clock Input</div> <div>Reset Input</div> <div>Avalon Memory Mapped ...</div>	<div>Double-click to export</div> <div>Double-click to export</div> <div>Double-click to export</div>	<div>clk_0</div> <div>[clock]</div> <div>[clock]</div>	<div>0x0001</div>	<div>0x0001</div>	<input type="checkbox"/>
<input checked="" type="checkbox"/>		conduit_end_0	Conduit	d_slave	[clock]			
<input checked="" type="checkbox"/>		<div>my_slave_1</div> <div>clock</div> <div>reset</div> <div>s0</div>	<div>my_slave</div> <div>Clock Input</div> <div>Reset Input</div> <div>Avalon Memory Mapped ...</div>	<div>Double-click to export</div> <div>Double-click to export</div> <div>Double-click to export</div>	<div>clk_0</div> <div>[clock]</div> <div>[clock]</div>	<div>0x0002</div>	<div>0x0002</div>	<input type="checkbox"/>
<input checked="" type="checkbox"/>		conduit_end_0	Conduit	dd_slave	[clock]			
<input checked="" type="checkbox"/>		<div>my_Dslave_0</div> <div>clock</div> <div>reset</div> <div>s0</div>	<div>my_Dslave</div> <div>Clock Input</div> <div>Reset Input</div> <div>Avalon Memory Mapped ...</div>	<div>Double-click to export</div> <div>Double-click to export</div> <div>Double-click to export</div>	<div>clk_0</div> <div>[clock]</div> <div>[clock]</div>	<div>default</div>	<div>default</div>	<input checked="" type="checkbox"/>
		conduit_end_0	Conduit	def_slave	[clock]			



# Проверьте систему

- Убедитесь в том, что
  - Ваша система выглядит так же, как представленная на рисунке
  - Закладка сообщений (Messages) не содержит сообщений.
- Сохраните файл

System: lab\_PD4 Path: my\_master\_0.conduit\_end\_0

Use	Connec...	Name	Description	Export	Clock	Base	End	Default Slave	I...	Tags
<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/> <b>clk_0</b>	Clock Source							
		clk_in	Clock Input	clk	exported					
		clk_in_reset	Reset Input	reset	[clk_in]					
		clk	Clock Output		clk_0					
		clk_reset	Reset Output		clk_0					
<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/> <b>my_master_0</b>	my_master							
		clock	Clock Input	Double-click to export	clk_0					
		reset	Reset Input	Double-click to export	[clock]					
		m0	Avalon Memory Mapped ...	Double-click to export	[clock]					
<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/> <b>conduit_end_0</b>	Conduit	data_d	[clock]					
<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/> <b>my_slave_0</b>	my_slave							
		clock	Clock Input	Double-click to export	clk_0					
		reset	Reset Input	Double-click to export	[clock]					
		s0	Avalon Memory Mapped ...	Double-click to export	[clock]	0x0001	0x0001	<input type="checkbox"/>		
<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/> <b>conduit_end_0</b>	Conduit	d_slave	[clock]					
<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/> <b>my_slave_1</b>	my_slave							
		clock	Clock Input	Double-click to export	clk_0					
		reset	Reset Input	Double-click to export	[clock]					
		s0	Avalon Memory Mapped ...	Double-click to export	[clock]	0x0002	0x0002	<input type="checkbox"/>		
<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/> <b>conduit_end_0</b>	Conduit	dd_slave	[clock]					
<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/> <b>my_Dslave_0</b>	my_Dslave							
		clock	Clock Input	Double-click to export	clk_0					
		reset	Reset Input	Double-click to export	[clock]					
		s0	Avalon Memory Mapped ...	Double-click to export	[clock]	default	default	<input checked="" type="checkbox"/>		
<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/> <b>conduit_end_0</b>	Conduit	def_slave	[clock]					

Current filter: All Interfaces

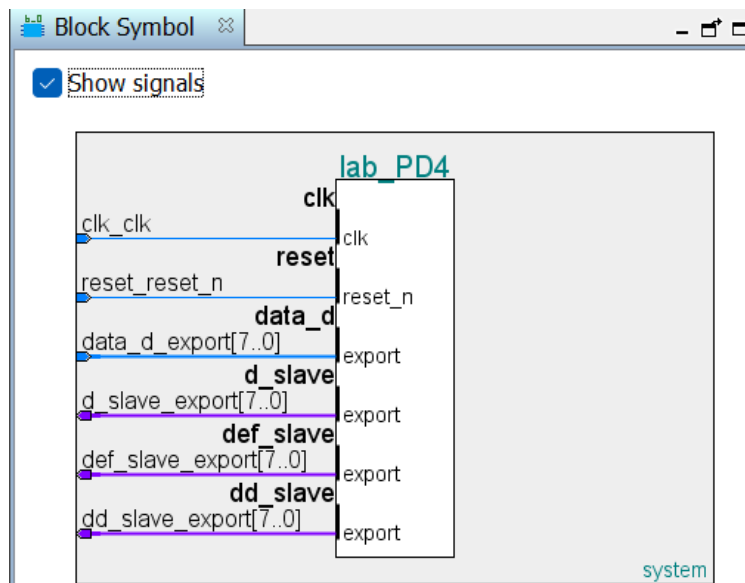
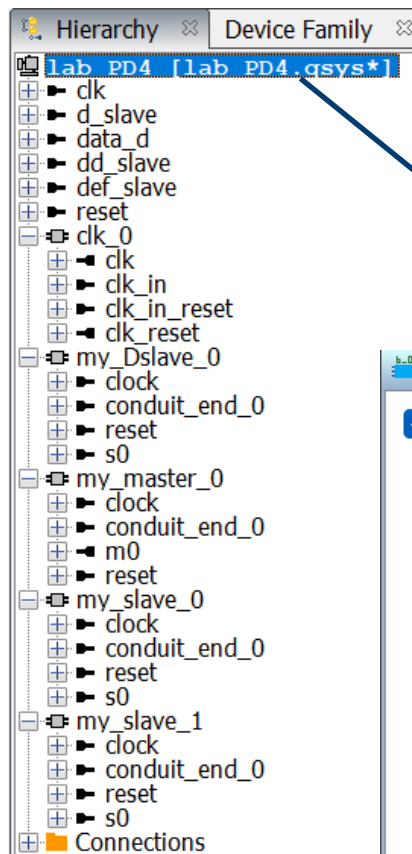
Messages

Type	Path	Message
------	------	---------



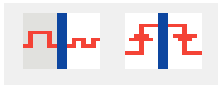

# Анализ системы

- Выполните команду: меню View=>Hierarchy
- В окне закладки Hierarchy выделите систему
- Выполните команду: меню View=>Block Symbol
- Убедитесь, что иерархия системы и символ соответствует представленным на рисунке





# Анализ системы

- Выполните команду: меню View => Clock domains Beta
- Выберите режим отображения Clocks   
**Clocks** Resets
- Убедитесь в том, что в столбце Connections нет красных точек => нет проблем подключения
- Выполните команду: меню View => Reset domains Beta
- Выберите режим отображения Reset   
**Clocks** **Resets**
- Убедитесь в том, что в столбце Connections нет красных точек => нет проблем подключения



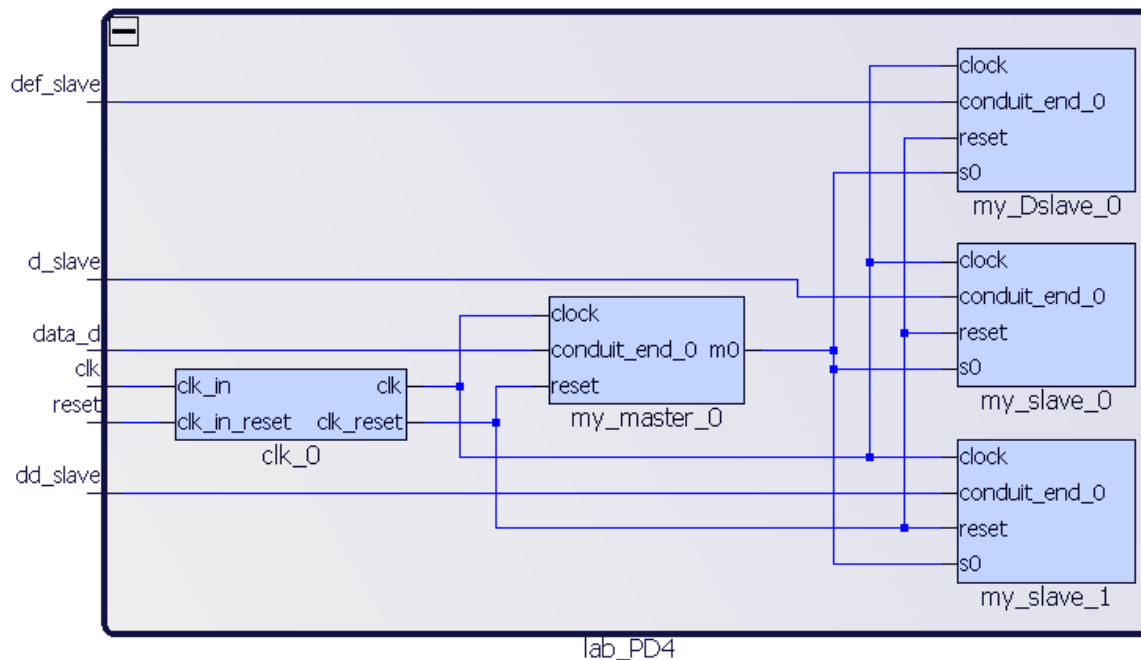
# Анализ системы

- Выполните команду:
  - меню **System => Show System with Platform Designer (или Qsys) Interconnect**
- сравните созданную Вами систему и систему с модулями добавленными PD: Убедитесь в том, что PD добавил только модуль ***mm\_interconnect\_0***



# Анализ системы

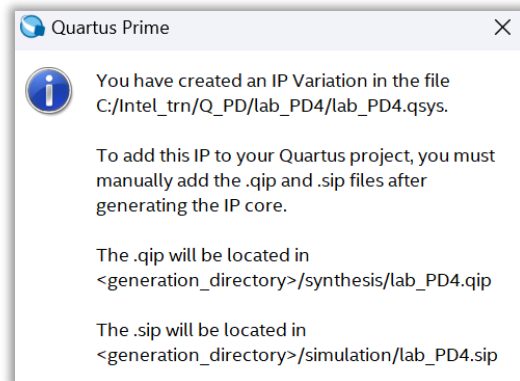
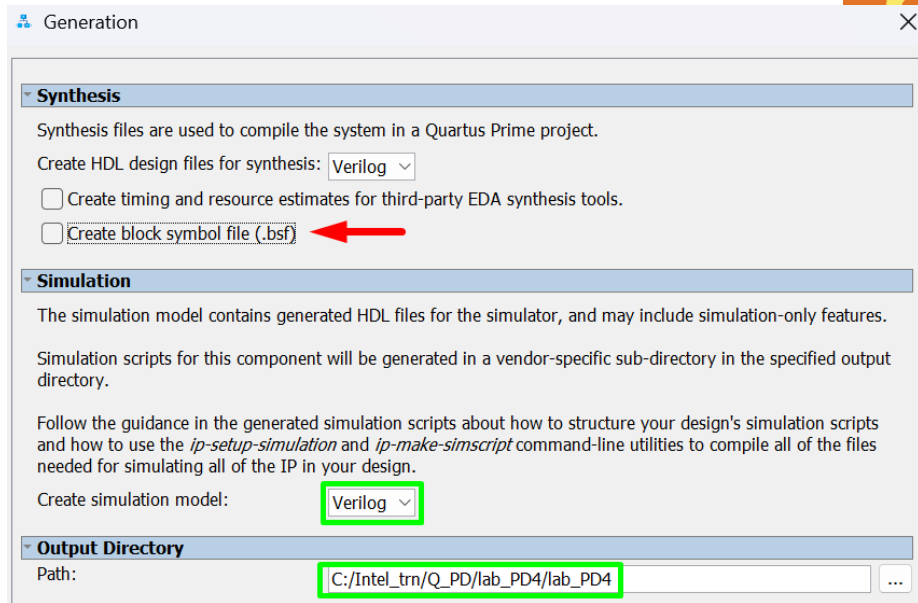
- Выполните команду: меню **View=>Schematic**
- Убедитесь в том, что система синхронизации Вашей системы выглядит так же, как представленная на рисунке



# Генерация системы

- В окне PD нажмите кнопку Generate HDL... (правый нижний угол окна)
- Установите режимы так, как показано на рисунке
- Нажмите кнопку Generate
- По окончании процедуры появится сообщение
  - Нажмите кнопку Close
- В окне PD нажмите кнопку Finish (правый нижний угол окна)
- Появится напоминание о необходимости подключить файлы к проекту пакета QP
- Нажмите кнопку ОК.

Generate: completed successfully.

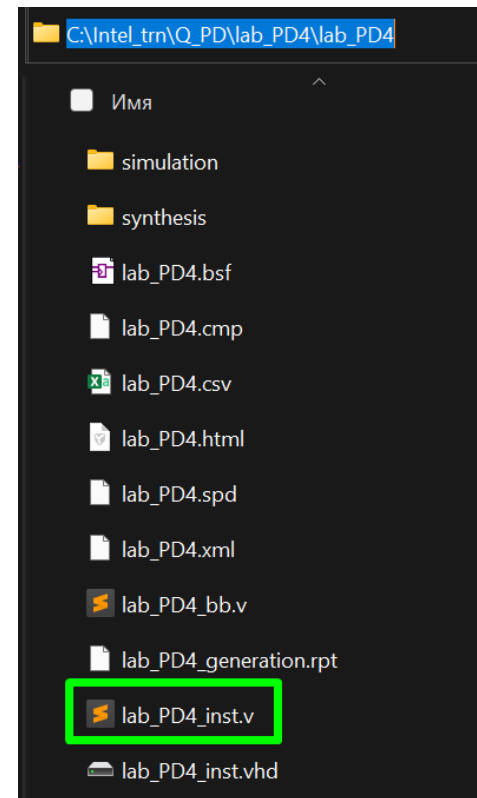




# Файл lab\_PD4\_top.sv

- Используя файла lab\_PD4\_inst.v создайте описание верхнего уровня. *Пример приведен на рисунке.*

```
1  `timescale 1 ns / 1 ns
2  module lab_PD4_top (
3      input bit clk,
4      input bit reset,
5      input bit [7:0] data_d,
6      output bit [7:0] dd_slave,
7      output bit [7:0] d_slave,
8      output bit [7:0] def_slave
9  );
10  lab_PD4 Lab4_sys_inst (
11      .clk_clk      (clk),
12      .reset_reset_n (reset),
13      .def_slave_export (def_slave),
14      .dd_slave_export (dd_slave),
15      .d_slave_export  (d_slave),
16      .data_d_export   (data_d)
17  );
18  endmodule
```







# Подключите файлы к проекту в QP

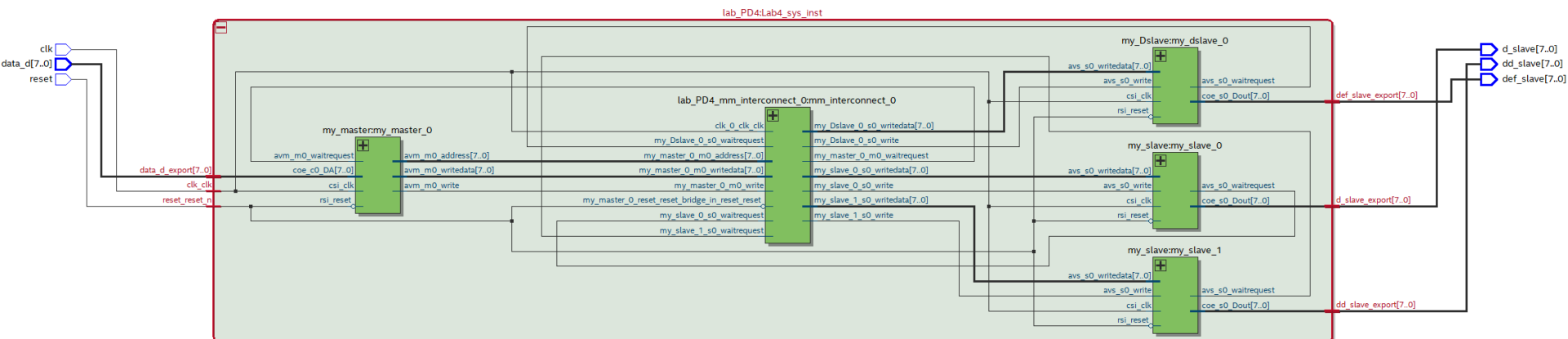
- В QP
  - Выполните **Project => Add\Remove Files from project**
  - Подключите к проекту файлы:
    - lab\_PD4\_top.sv
    - lab\_PD4.qip
    - lab\_PD4.sip

File Name	Type	Library
lab_PD4/synthesis/lab_PD4.qip	IP Variation File (.qip)	
lab_PD4/simulation/lab_PD4.sip	Quartus Prime SIP File	
lab_PD4_top.sv	SystemVerilog HDL File	



# Анализ и синтез в QP

- В QP
  - Файл lab\_PD4\_top.sv объявите файлом верхнего уровня
  - Выполните команду **Start Analysis and Synthesis**
  - **Убедитесь в том, что компиляция без ошибок**
- Выполните: меню Tools=>Netlist Viewers => RTL viewer
- Убедитесь в том, что Ваша схема похожа на схему, приведенную на рисунке





# Файл tb\_lab\_PD4\_top.sv

- Создайте тест для проверки системы, который:
  - Сбрасывает систему
  - Обеспечивает генерацию тактовых сигналов, необходимых для формирования Мастером на шине Avalon-MM 9-ти запросов на запись.
  - Задает на входе data\_d мастера число, равное Вашему номеру в списке группы.

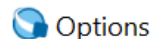
*Пример теста приведен на рисунке.*

```
1  `timescale 1 ns / 1 ns
2  module tb_lab_PD4_top ();
3      bit clk;
4      bit reset ;
5      bit [7:0] data_d;
6      bit [7:0] dd_slave;
7      bit [7:0] d_slave;
8      bit [7:0] def_slave;
9      assign data_d = 8'd1; //Ваш номер в списке группы
10     always #10 clk = ~ clk;
11     initial begin
12         #20;
13         reset = 1'b1;
14         repeat (4*9) @(negedge clk);
15         $stop;
16     end
17     Lab_PD4_top Lab4_sys_top_inst (.*) ;
18 endmodule
```



# Настройка QP для NativeLink

- Убедитесь, что правильно задана ссылка на пакет ModelSim
  - Выполните команду Tools=>Options



Category:

- ▼ General
  - EDA Tool Options**
  - Fonts
  - Headers & Footers Settings
- ▼ Internet Connectivity
  - Notifications
  - Libraries
- ▼ IP Settings
  - IP Catalog Search Locations
  - Design Templates
  - License Setup
  - Preferred Text Editor
  - Processing
  - Tooltip Settings
- ▼ Messages

### EDA Tool Options

Specify the directory that contains the tool executable for each third-party ED.

EDA Tool	Directory Containing Tool Executable
Precision ...	
Synplify	
Synplify ...	
Active-HDL	
Riviera-P...	
ModelSim	
QuestaSim	
ModelSi...	C:\intelFPGA_lite\20.1\modelsim_ase\win32aloem



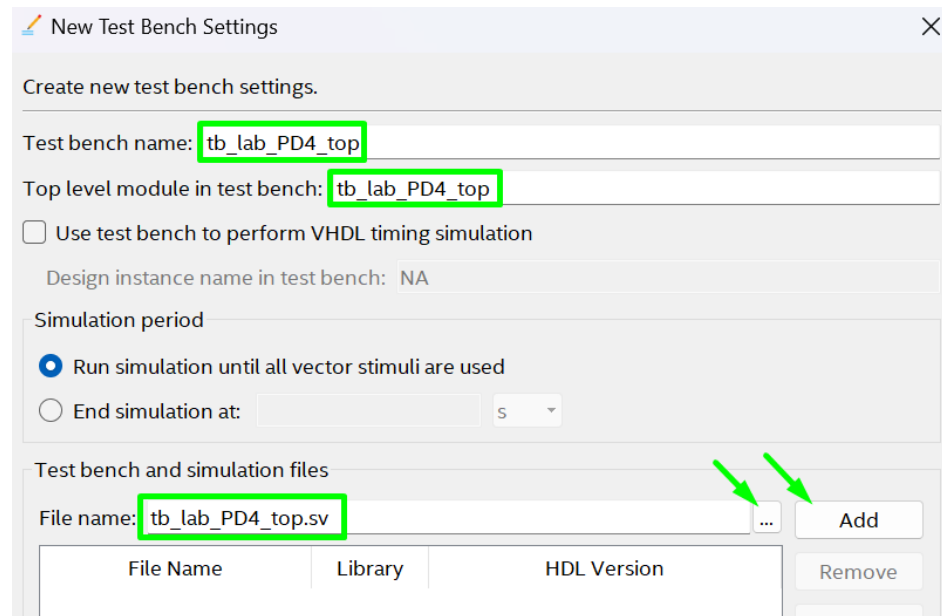
# Настройка QP для NativeLink

- Выполните команду : **меню Assignment=>Settings=>Simulation =>NativeLink settings=>кнопка Test Benches**



- Нажмите кнопку **New**

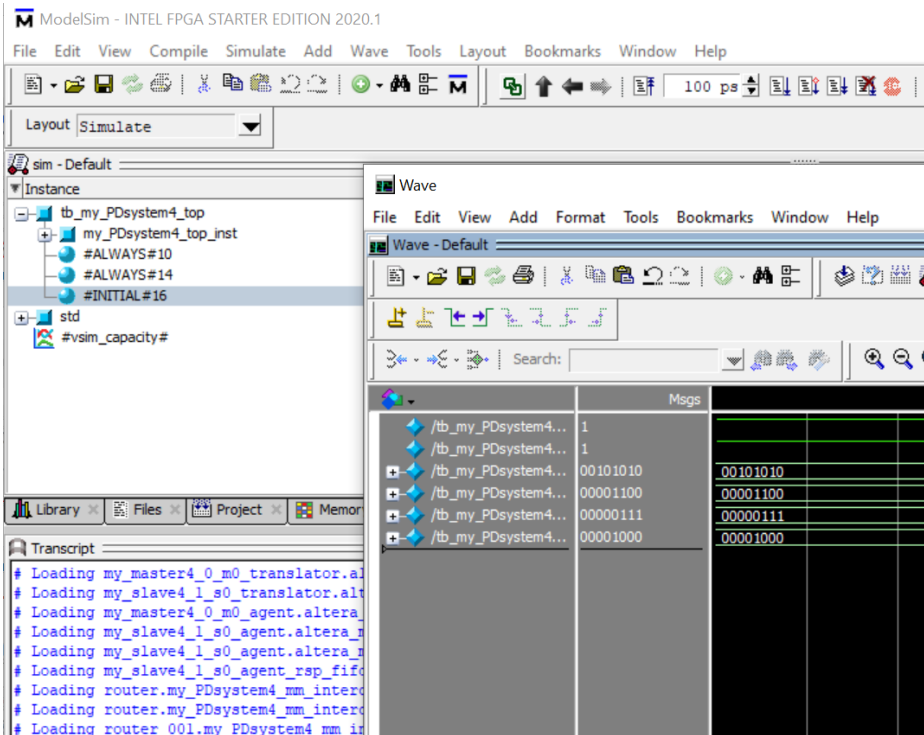
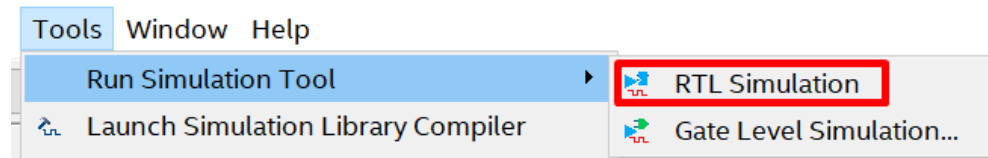
- Выполните назначения, показанные на рисунке.





# Запуск моделирования с NativeLink

- Выполните команду : меню **Tools=>Run Simulation Tool=>RTL Simulation**
- Откроется окно (окна) пакета ModelSim



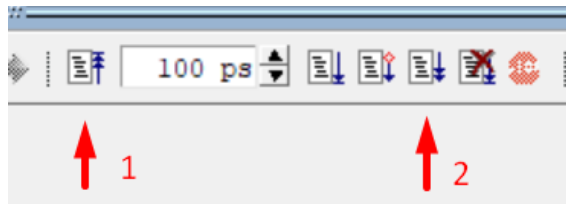


# Загрузка формата временной диаграммы

- В окне Wave пакета ModelSim
  - Удалите временные диаграммы
  - выполните команду: File=>Load и выберите файл waveLab4.do (папка C:\Intel\_trn\Q\_PD\lab\_PD4)


*Если при загрузке файла возникают ошибки => в Вашем проекте использованы имена компонентов/экземпляров компонентов/выводов, не соответствующие файлу waveLab4.do. Следует: либо исправить несоответствующие имена в do файле (он текстовый); либо создать раскладку сигналов временной диаграммы самостоятельно (образец на следующем слайде)*

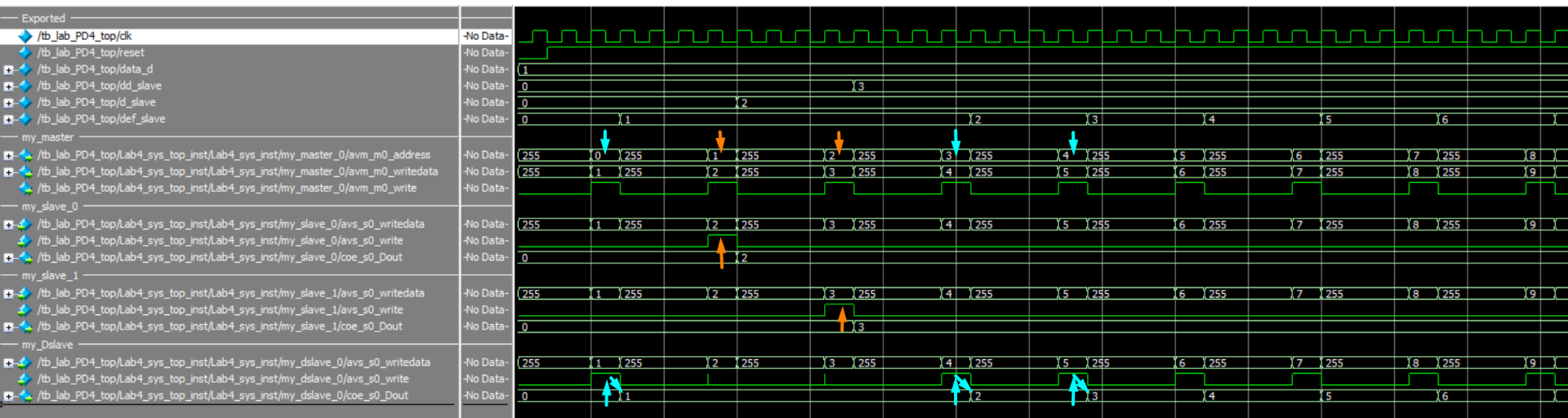
- В окне Wave пакета ModelSim нажмите кнопку Restart а затем Run -All





# Анализ временной диаграммы

- В окне Wave пакета ModelSim выполните команду Zoom Full 
- Проведите анализ полученной временной диаграммы и убедитесь в правильности работы системы.
- По какому адресу реализована запись в компоненты: my\_slave\_0; my\_slave\_1; my\_Dslave?







# Файл tb\_lab\_PD4\_top.sv

- Измените тест так, что бы он:
  - Обеспечивал генерацию числа тактовых сигналов, необходимых для формирования Мастером на шине Avalon-MM 260-ти запросов на запись.
- Перемоделируйте Вашу систему

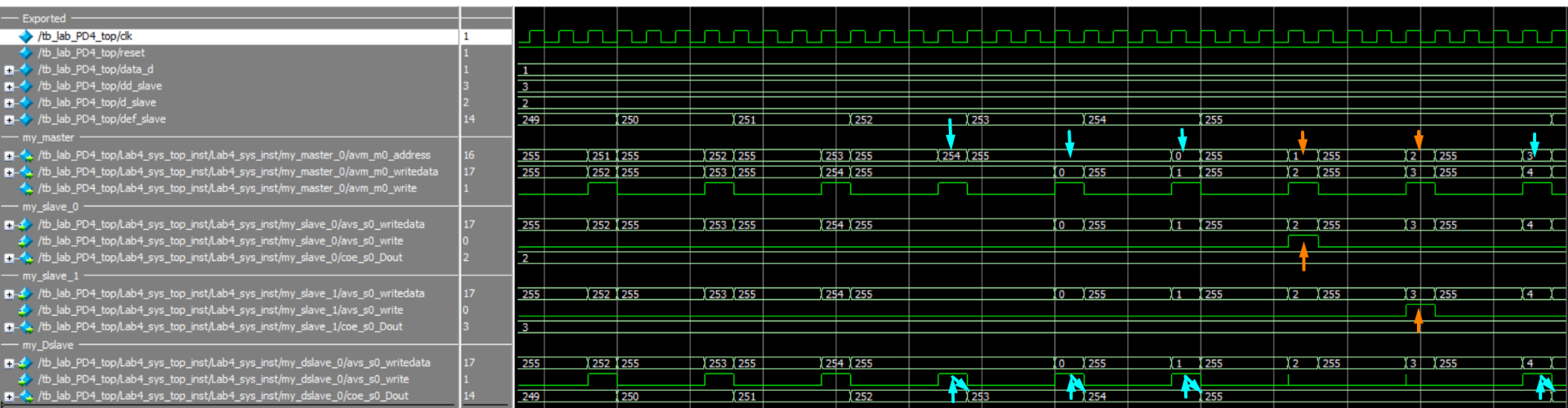
*Пример теста приведен на рисунке.*

```
1  `timescale 1 ns / 1 ns
2  module tb_lab_PD4_top ();
3      bit clk;
4      bit reset ;
5      bit [7:0] data_d;
6      bit [7:0] dd_slave;
7      bit [7:0] d_slave;
8      bit [7:0] def_slave;
9      assign data_d = 8'd1; //Ваш номер в списке группы
10     always #10 clk = ~ clk;
11     initial begin
12         #20;
13         reset = 1'b1;
14         repeat (4*260) @(negedge clk);
15         $stop;
16     end
17     Lab_PD4_top Lab4_sys_top_inst (.*);
18 endmodule
```



# Анализ временной диаграммы

- В окне Wave пакета ModelSim найдите окончание процесса моделирования
- Проведите анализ полученной временной диаграммы и убедитесь в правильности работы системы.
  - Сколько было обращений к модулю `my_Dslave` при формировании мастером на шине Avalon-MM 256-ти запросов на запись?





# Пакет QP: настройка In-System Sources & Probes

- Настройте IP компонент In-System Sources & Probes:

- Задайте имя экземпляра компонента – SP\_unit
- В окне настройки параметров задайте параметры так, как показано на рисунке:
  - Компонент будет источником сигналов
    - reset
    - data\_d [7:0]
  - Формируемые сигналы должны быть синхронизированы.

**Instance Info**

- ☒ Automatic Instance Index Assignment
- Instance Index: 0
- The 'Instance ID' of this instance (optional): SP\_0

**Probe Parameters**

- Probe Port Width [0..512]: 0

**Source Parameters**

- Source Port Width [0..512]: 9
- Hexadecimal initial value for the Source Port: 00
- ☒ Use Source Clock
- ☐ Use Source Clock Enable



# Файл db\_lab4\_PD4\_top.sv

- Файл db\_lab\_PD4\_top.sv – файл с описанием модуля верхнего уровня для отладки разработанной системы на плате.
- Файл подключает созданный модуль SP\_unit к модулю lab\_PD4\_top
- Создайте этот файл, используя файл SP\_unit\_inst.v (папка C:\Intel\_trn\Q\_PD\lab\_PD4\SP\_unit). *Пример файла приведен на рисунке*

```
1  `timescale 1 ns / 1 ns
2  module db_lab_PD4_top (
3      (* altera_attribute = "-name IO_STANDARD \"3.3-V LVC MOS\"", chip_pin = "R8" *)
4      // "23" for miniDilab-CIV
5      // "R8" for DE0_nano
6      // "N5" for MAX10 NEEK
7      input bit clk
8  );
9      bit reset ;
10     bit [7:0] data_d;
11     bit [7:0] dd_slave;
12     bit [7:0] d_slave;
13     bit [7:0] def_slave;
14     SP_unit SP_unit_inst (
15         .source      ( {reset, data_d} ),
16         .source_clk   ( clk )
17     );
18     Lab_PD4_top lab_PD4_top_inst (.*);
19 endmodule
```



# Подключите файлы к проекту в QP

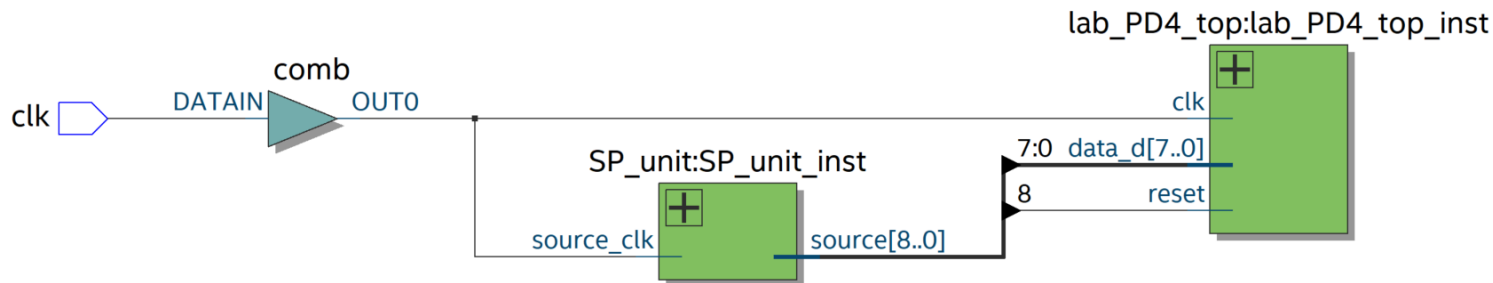
- В QP
  - Выполните **Project => Add\Remove Files from project**
  - Выберите и подключите файлы
    - db\_lab\_PD4\_top.sv
    - SP\_unit.qip

File Name	Type
SP_unit/synthesis/SP_unit.qip	IP Variation File (.qip)
db_lab_PD4_top.sv	SystemVerilog HDL File



# Анализ и синтез в QP

- В QP
  - Файл db\_lab\_PD4\_top.sv объявите файлом верхнего уровня
  - Выполните команду **Start Analysis and Synthesis**
    - Убедитесь в том, что компиляция прошла без ошибок
- Выполните: меню Tools=>Netlist Viewers => RTL viewer
- Убедитесь в том, что Ваша схема похожа на схему, приведенную на рисунке






# Настройка логического анализатора Signal TapII

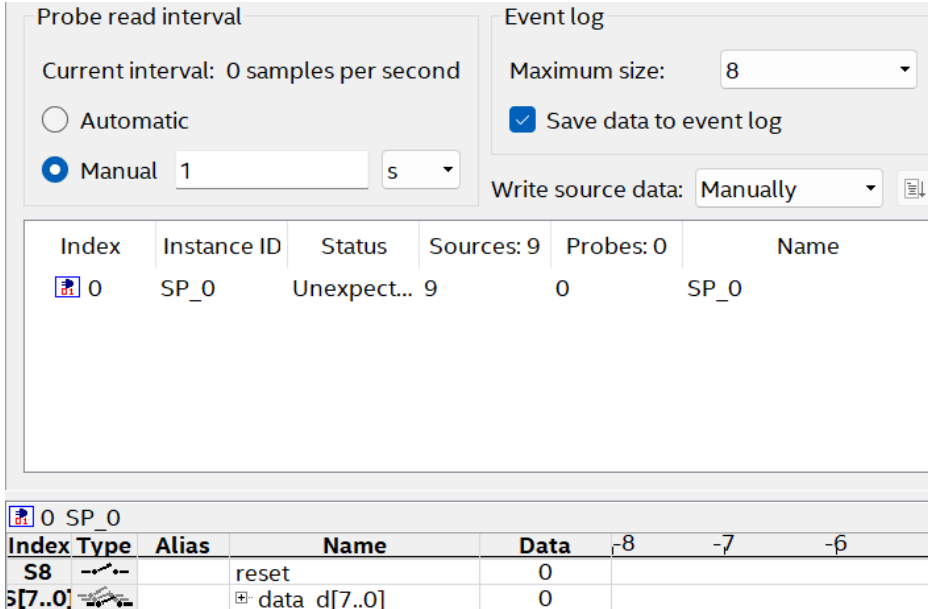
- Создайте файл SignalTapII и сохраните его под именем lab\_PD4.stp
- Настройте логический анализатор
  - Подключенные цепи должны соответствовать цепям из теста ModelSim (смотри рисунок)
  - Условие захвата – переход reset из 0 в 1
  - Длина – 64 отсчета
  - Pre-trigger position

Type	i	Node Name	Data Enable	Trigger Enable	Trigger Conditions
			101	101	1 Basic AND
		lab PD4 top:lab PD4 top inst reset	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
		lab PD4 top:lab PD4 top inst data d[7..0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	XXXXXXXXXb
		lab PD4 top:lab PD4 top inst dd slave[7..0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	XXXXXXXXXb
		lab PD4 top:lab PD4 top inst d slave[7..0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	XXXXXXXXXb
		lab PD4 top:lab PD4 top inst def slave[7..0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	XXXXXXXXXb
		lab PD4 top:lab PD4 top inst lab PD4:Lab4 sys inst my master:my master 0 avm m0 address[7..0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	XXXXXXXXXb
		lab PD4 top:lab PD4 top inst lab PD4:Lab4 sys inst my master:my master 0 avm m0 writedata[7..0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	XXXXXXXXXb
		lab PD4 top:lab PD4 top inst lab PD4:Lab4 sys inst my master:my master 0 avm m0 write	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
		lab PD4 top:lab PD4 top inst lab PD4:Lab4 sys inst my slave:my slave 0 avs s0 writedata[7..0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	XXXXXXXXXb
		lab PD4 top:lab PD4 top inst lab PD4:Lab4 sys inst my slave:my slave 0 avs s0 write	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
		lab PD4 top:lab PD4 top inst lab PD4:Lab4 sys inst my slave:my slave 0 coe s0 Dout[7..0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	XXXXXXXXXb
		lab PD4 top:lab PD4 top inst lab PD4:Lab4 sys inst my slave:my slave 1 avs s0 writedata[7..0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	XXXXXXXXXb
		lab PD4 top:lab PD4 top inst lab PD4:Lab4 sys inst my slave:my slave 1 avs s0 write	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
		lab PD4 top:lab PD4 top inst lab PD4:Lab4 sys inst my slave:my slave 1 coe s0 Dout[7..0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	XXXXXXXXXb
		lab PD4 top:lab PD4 top inst lab PD4:Lab4 sys inst my Dslave:my dslave 0 avs s0 writedata[7..0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	XXXXXXXXXb
		lab PD4 top:lab PD4 top inst lab PD4:Lab4 sys inst my Dslave:my dslave 0 avs s0 write	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
		lab PD4 top:lab PD4 top inst lab PD4:Lab4 sys inst my Dslave:my dslave 0 coe s0 Dout[7..0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	XXXXXXXXXb



# Компиляция и создание lab\_PD4.spf

- Запустите полную компиляцию проекта 
- Подключите плату и сконфигурируйте FPGA
- Создайте файл ISSPE (меню Tools=> In-System Source and Probe Editor)
- Настройте его так, как показано на рисунке
- Сохраните под именем **lab\_PD4.spf**



Index	Instance ID	Status	Sources	Probes	Name
0	SP_0	Unexpect...	9	0	SP_0

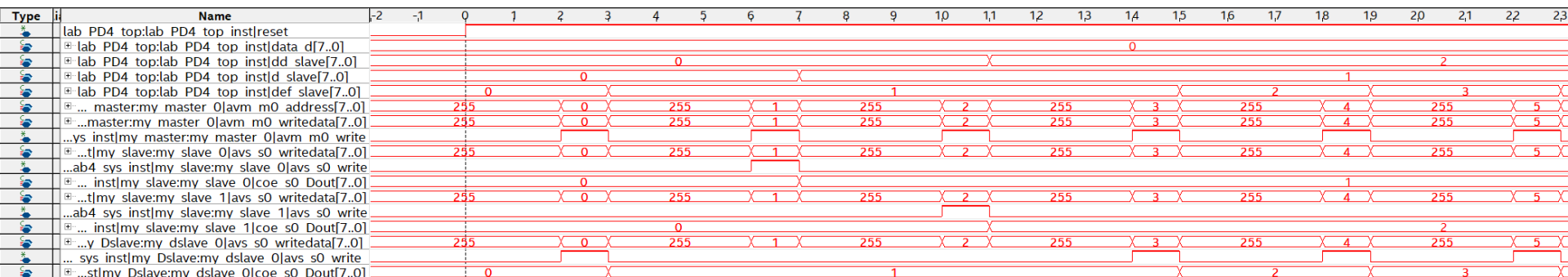
Index	Type	Alias	Name	Data			
S8			reset	0			
S[7..0]			data d[7..0]	0			





# Анализ результатов в окне Signal TapII (1)

- Используя ISSPE для
  - управления сигналом reset
  - Задания значения data\_d (значения, равного Вашему номеру в списке группы)получите в логическом анализаторе временную диаграмму, **аналогичную** приведённой ниже (показана часть диаграммы)
- Сравните ее с результатами моделирования, полученными в пакете ModelSim.





# Анализ результатов в окне Signal TapII (2)

- Настройте условие триггера (условие захвата данных) в логическом анализаторе так, чтобы захват данных начинался по появлению на шине адреса Мастера значения = 253
- С помощью SP\_unit сбросьте отлаживаемый модуль
- Получите временную диаграмму
- Сравните ее с результатами моделирования, полученными в пакете ModelSim.



# *Лабораторная 4*

## *ЗАВЕРШЕНА!*