

《存储器的基本知识与应用》

出品: Fraser Innovation Inc.

主讲: 庄乾起

实验平台: FII系列FPGA开发板

淘宝店铺: <https://shop296232497.taobao.com/>

知识库网站: <http://www.icfedu.cn/>



8

DDR3 SDRAM 与FPGA 接口设计



MT41J128M16HA-125

实例详解-FPGA接口设计

1. 新建FPGA工程

(1) 在Vivado下新建工程

File-->Project-->New

(a) 工程名称(DDR3_intf)及存储目录(D:/FPGA_DDR3_IO)如图8-1

(b) 选择创建子目录
--> Next

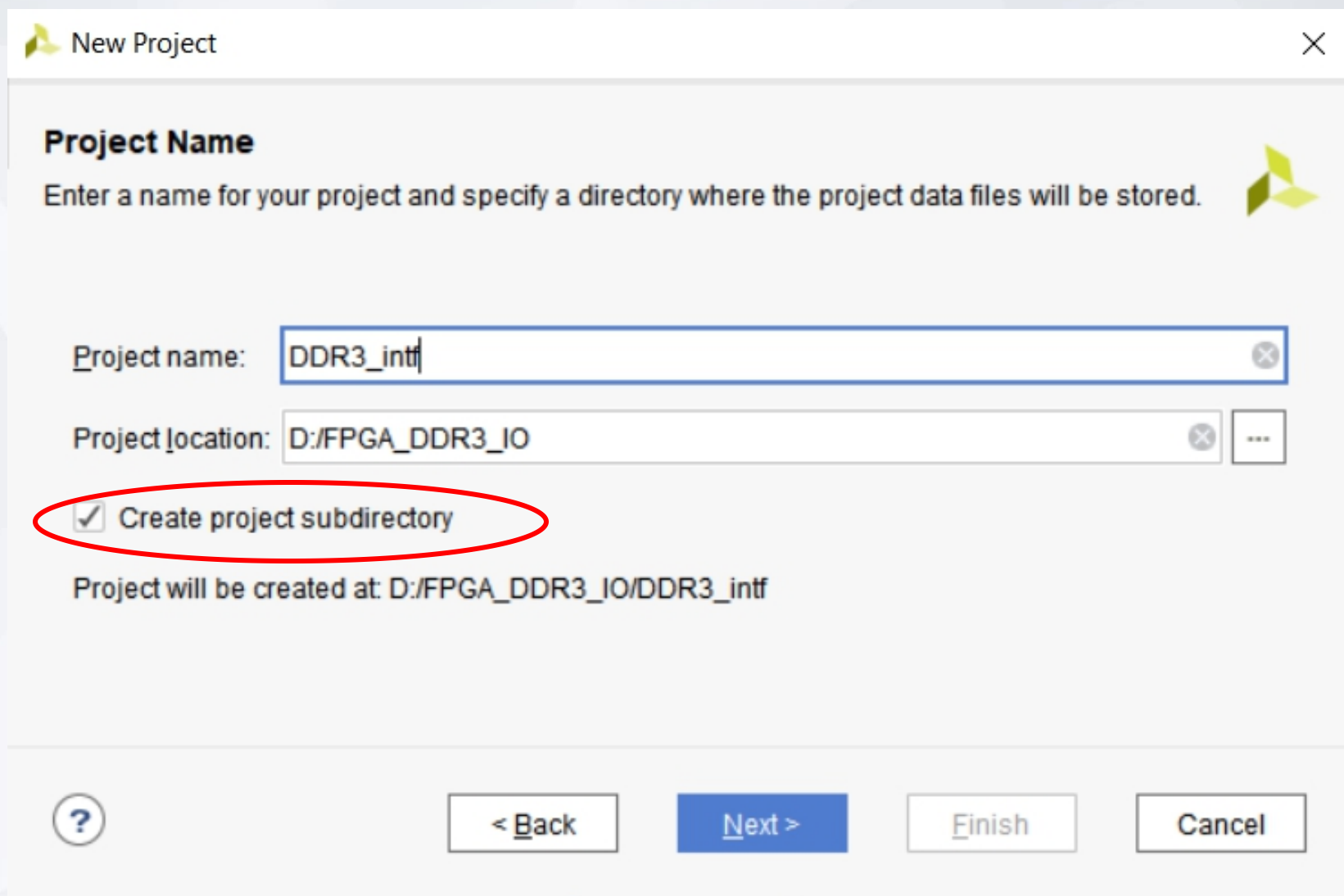


图 8-1 新建FPGA工程-1

MT41J128M16HA-125

实例详解-FPGA接口设计

1. 新建FPGA工程

(2) 选择RTL Project

在生成工程文件时不
添加额外的源文件

如图：8-2

--> Next

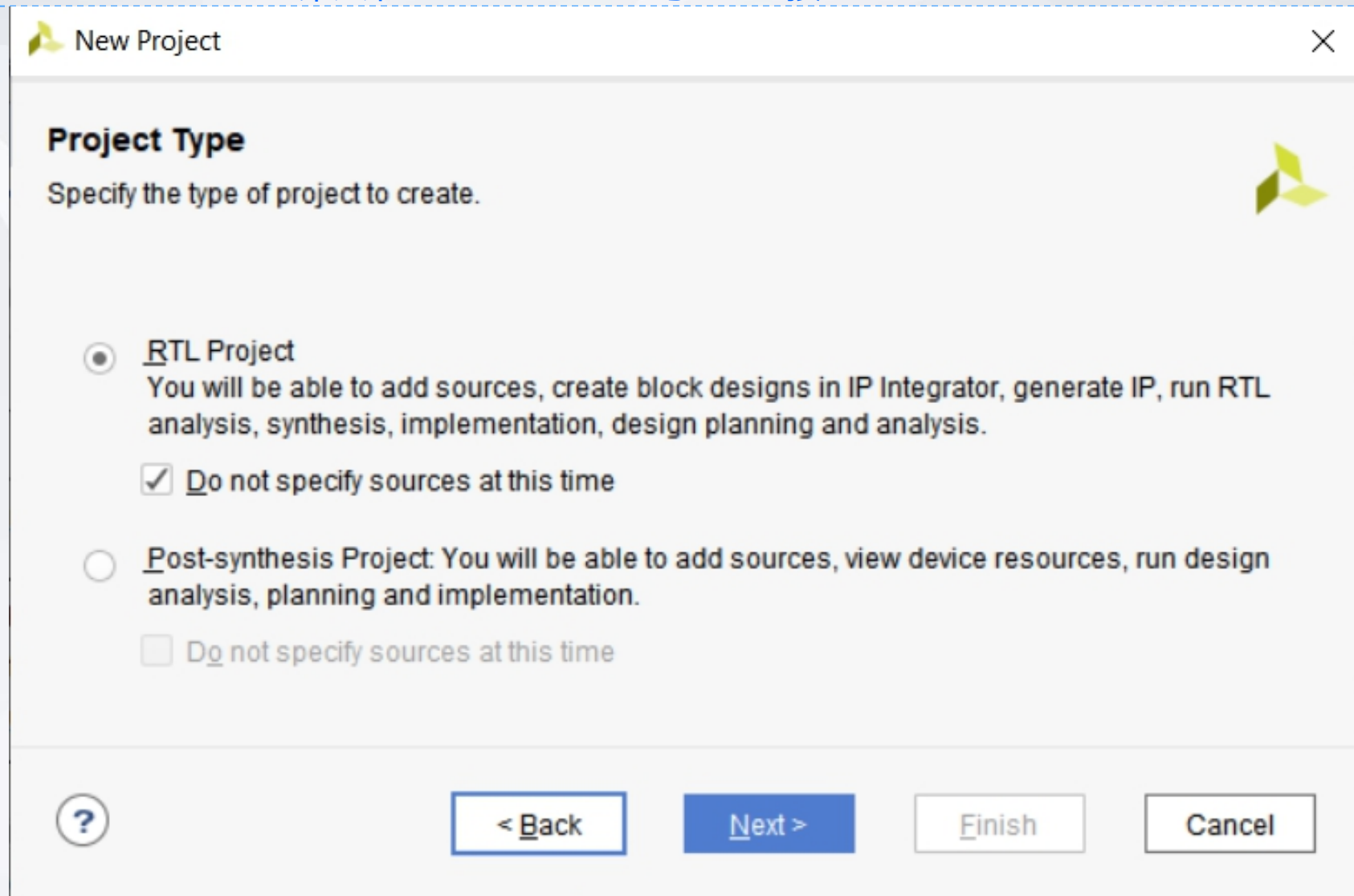


图 8-2 新建FPGA工程-2

MT41J128M16HA-125

实例详解-FPGA接口设计

1. 新建FPGA工程

(3) 选择器件:

这里选择 xc7a100fpg676-2

如图8-3

--> Next

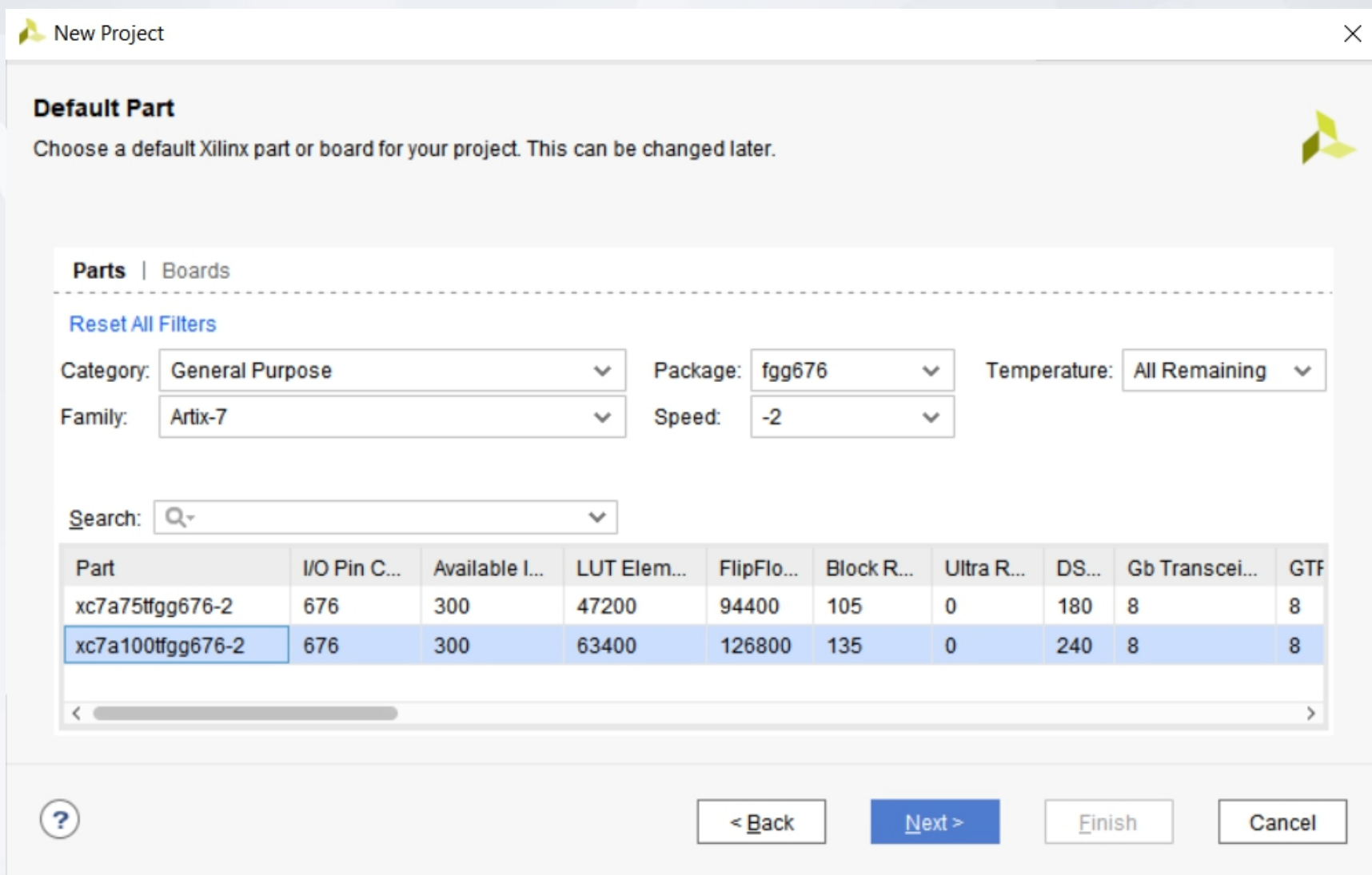


图 8-3 新建FPGA工程-3

第八节 DDR3 SDRAM 与FPGA 接口

MT41J128M16HA-125

实例详解-FPGA接口设计

1. 新建FPGA工程

(4) 工程概述: 如图8-4

--> Finish

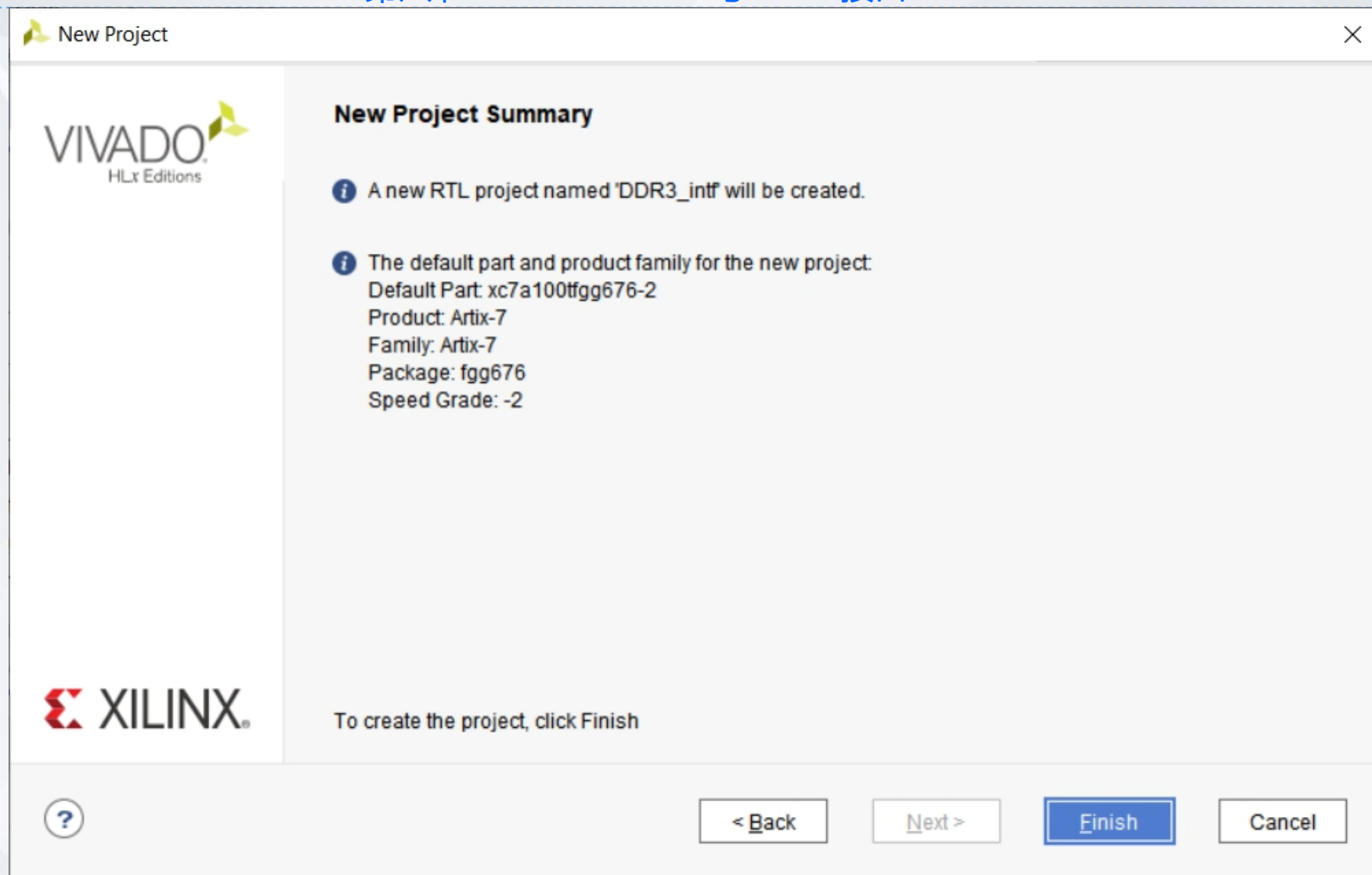


图 8-4 新建FPGA工程-4

第八节 DDR3 SDRAM 与FPGA 接口

MT41J128M16HA-125

实例详解-FPGA接口设计

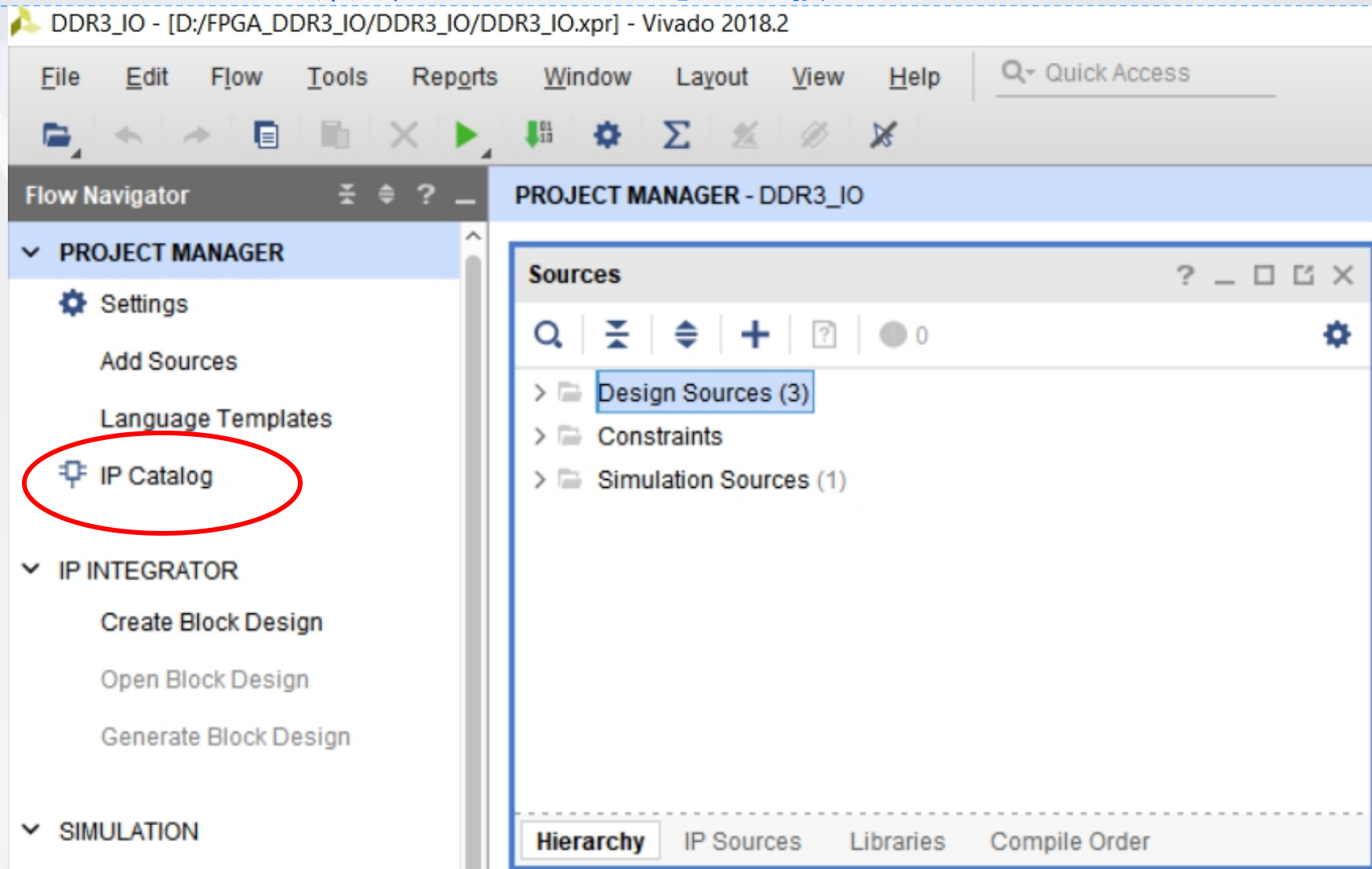
2. 在新建FPGA工程添加DDR3
控制器及接口(1) 打开IP catalog 窗口，
双击IP Catalog

图 8-5 打开IP Catalog窗口

第八节 DDR3 SDRAM 与FPGA 接口

MT41J128M16HA-125

实例详解-FPGA接口设计

2. 在新建FPGA工程添加

DDR3 控制器及接口

(1) 在IP catalog 窗口内,
输入MEM,搜索

选择Memory

Interface Generator-->

Memory Interface generator

双击进入下一个界面

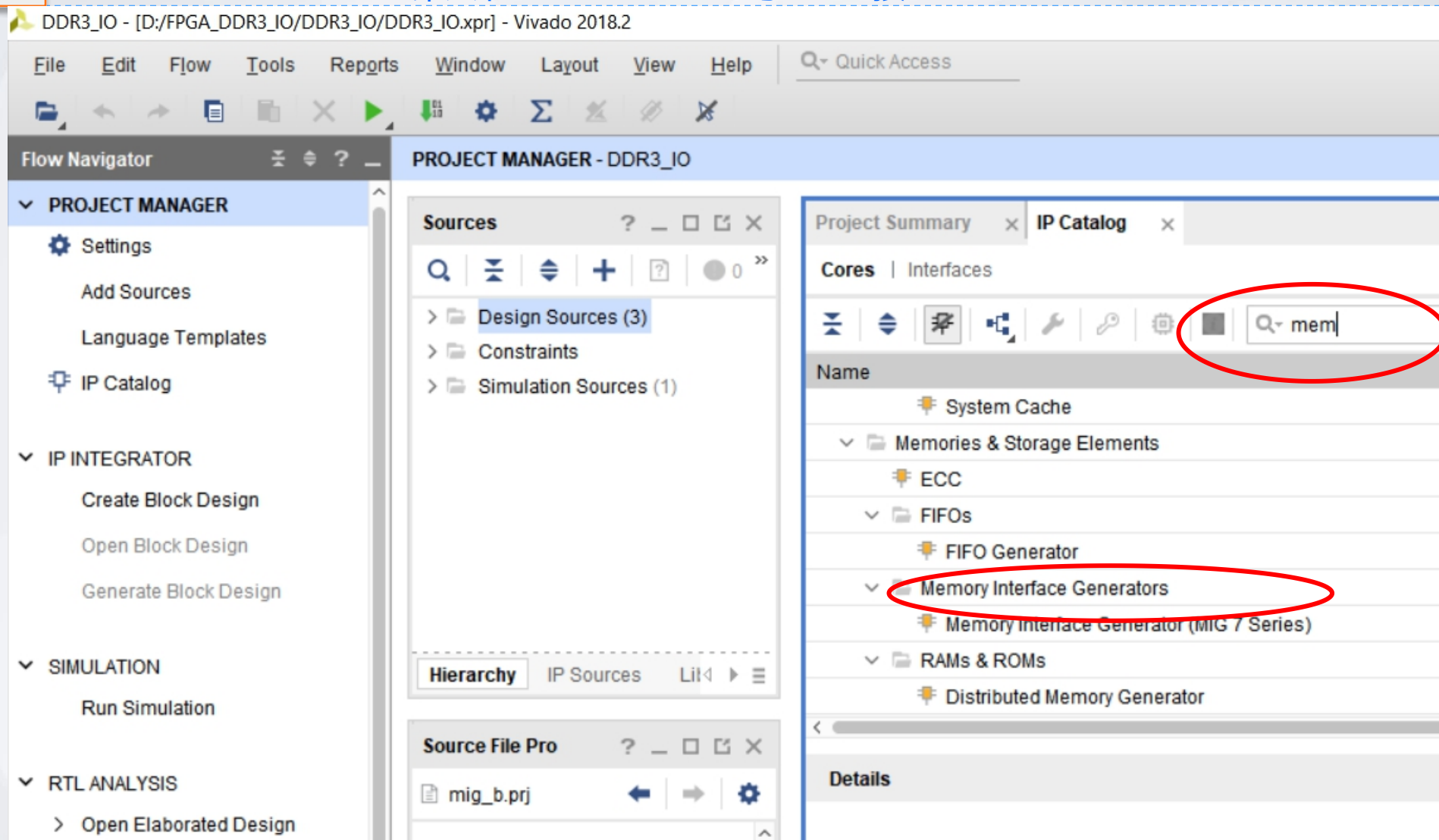


图 8-6 创建DDR3接口-1

MT41J128M16HA-125

实例详解-FPGA接口设计

2. 在新建FPGA工程添加
DDR3 控制器及接口
(2) 在新界面中对MIG
(Memory Interface
Generator)及项目中使用的
器件 (FPGA)，软件工
具，开发语言 (Verilog)
进行了描述。

--> Next

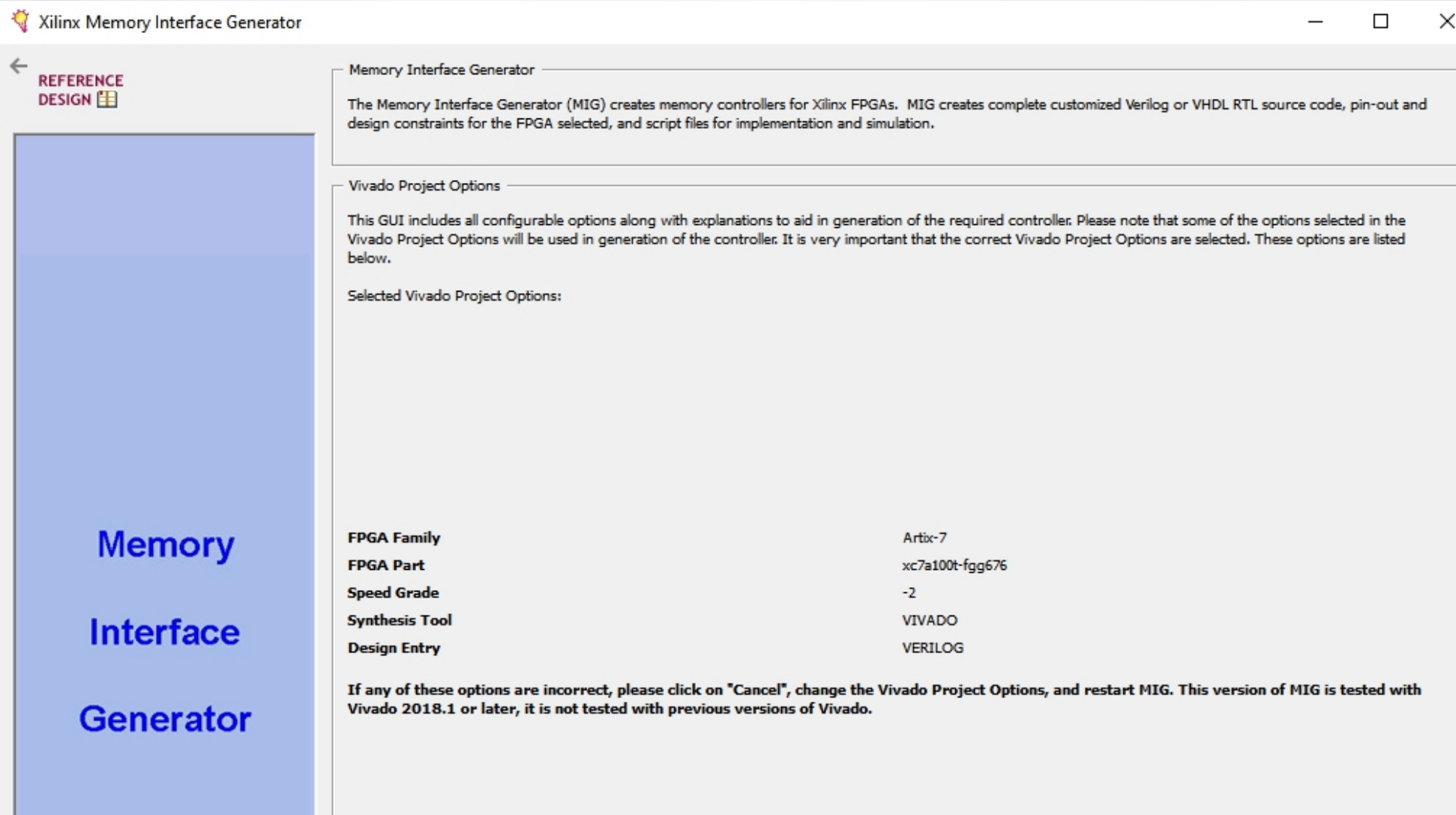


图 8-7 新建FPGA工程-2



MT41J128M16HA-125

实例详解-FPGA接口设计

2. 在新建FPGA工程添加DDR3 控制器及接口

(3) 在新界面中对MIG (Memory Interface Generator) 的选项如下:

(a) MIG Output Option

选择: Create Design

包含RTL, XDC, implementation 及仿真文件, 可以用来仿真测试, 也可以用在其他文件中。

(b) 组件名称:

将mig_7series_0修改成

DDR3_INTF

(c) 多控制器:

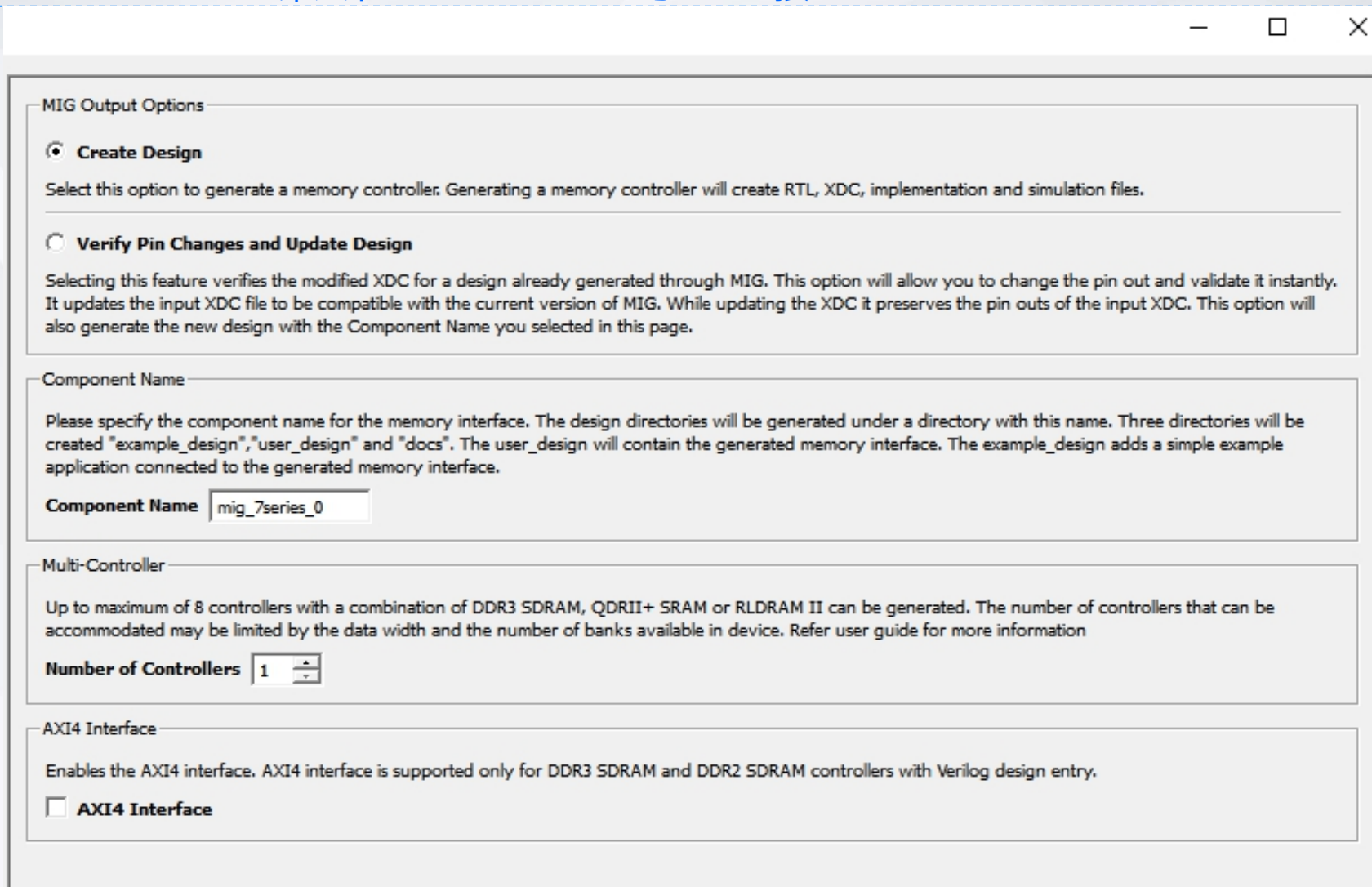
选择一个: 1

(d) AXI4 接口:

不选

如图8-8

--> Next



MIG Output Options

☒ Create Design

Select this option to generate a memory controller. Generating a memory controller will create RTL, XDC, implementation and simulation files.

☐ Verify Pin Changes and Update Design

Selecting this feature verifies the modified XDC for a design already generated through MIG. This option will allow you to change the pin out and validate it instantly. It updates the input XDC file to be compatible with the current version of MIG. While updating the XDC it preserves the pin outs of the input XDC. This option will also generate the new design with the Component Name you selected in this page.

Component Name

Please specify the component name for the memory interface. The design directories will be generated under a directory with this name. Three directories will be created "example_design", "user_design" and "docs". The user_design will contain the generated memory interface. The example_design adds a simple example application connected to the generated memory interface.

Component Name

Multi-Controller

Up to maximum of 8 controllers with a combination of DDR3 SDRAM, QDRII+ SRAM or RLDRAM II can be generated. The number of controllers that can be accommodated may be limited by the data width and the number of banks available in device. Refer user guide for more information

Number of Controllers

AXI4 Interface

Enables the AXI4 interface. AXI4 interface is supported only for DDR3 SDRAM and DDR2 SDRAM controllers with Verilog design entry.

☐ AXI4 Interface

图 8-8 新建FPGA工程-3

第八节 DDR3 SDRAM 与FPGA 接口

MT41J128M16HA-125

实例详解-FPGA接口设计

2. 在新建FPGA工程添加DDR3 控制
器及接口

(4) 在新界面中对MIG (Memory
Interface Generator) 的选项如下:

- (a) 兼容器件选择:
选择兼容器件xc7a75t-fgg676
- (b) 选择DDR的类型
DDR3_SDRAM

DDR3L :低电压系列, 接口电压1.35V
LPDDR3:是一种用在移动设备上的功耗
更低, 体积更小的器件, 电压1.2V

如图: 8-9

--> Next

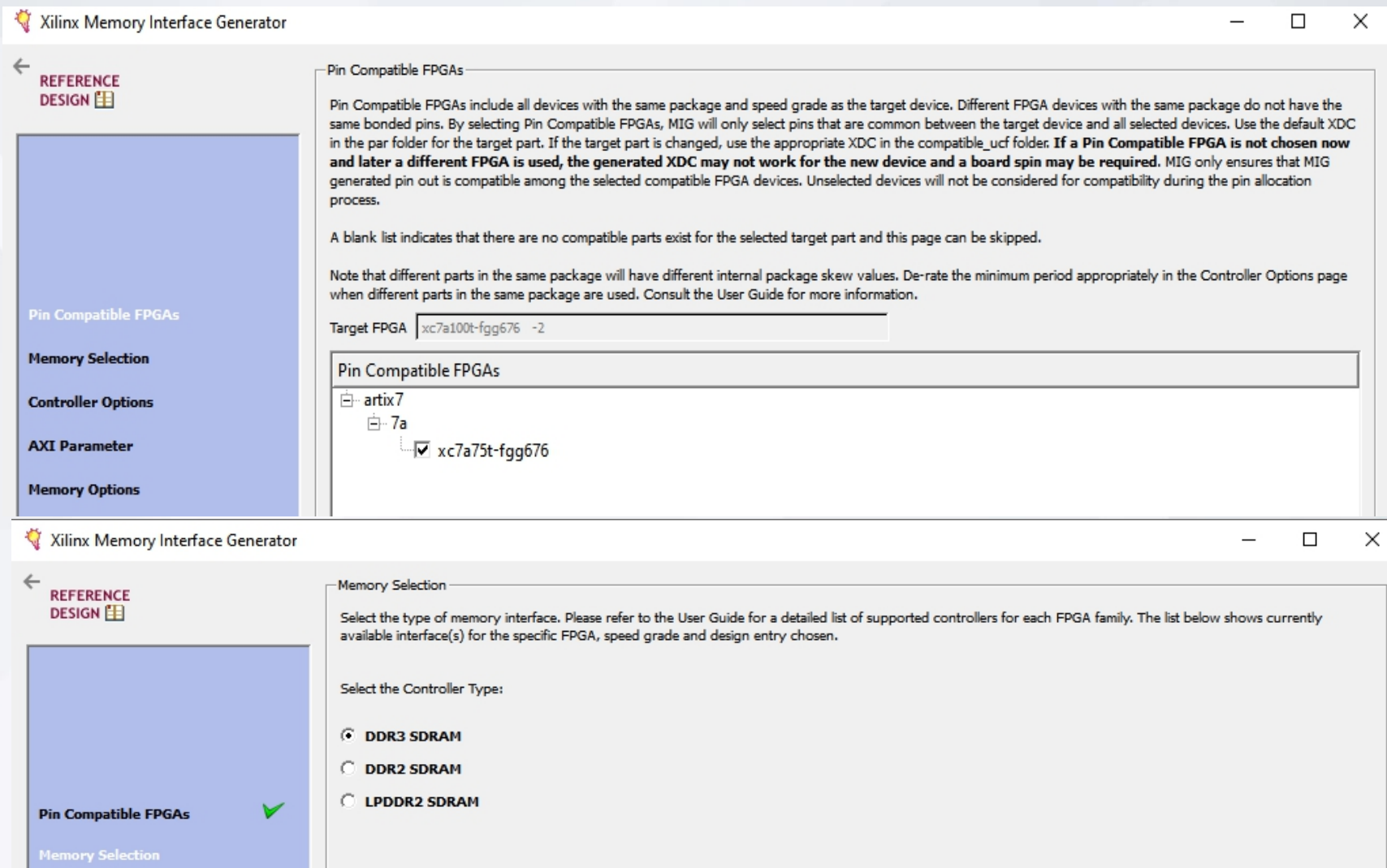


图 8-9 新建FPGA工程-4



第八节 DDR3 SDRAM 与FPGA 接口

MT41J128M16HA-125

实例详解-FPGA接口设计

2. 在新建FPGA工程添加DDR3 控制器及接口

(5) 在新界面中对MIG (Memory Interface Generator) 的选项如下:

DDR3参数选项:

(a) 时钟周期2500PS,400MHz

(b) 控制器与PHY (物理层) 的时钟比例4: 1

(c) VCCAUX的电压设为1.8V

(d) Memory的类型选择

Components (非RDIMMS, LRDIMMS, UDIMMS, SODIMMS, 区别?)

(e) 电压1.5V, 位宽16 (? , 32?)

(f) ECC (disable)

(g) BANK Machine -4

(h) ORDERING --strict

如图: 8-10

--> Next

REFERENCE DESIGN

Options for Controller 0 - DDR3 SDRAM

Clock Period: Choose the clock period for the desired frequency. The allowed period range(2500 - 3300) is a function of the selected FPGA part and FPGA speed grade. Refer to the User Guide for more information. 2500 ps 400.00 MHz

PHY to Controller Clock Ratio: Select the PHY to Memory Controller clock ratio. The PHY operates at the Memory Clock Period chosen above. The controller operates at either 1/4 or 1/2 of the PHY rate. The selected Memory Clock Period will limit the choices. 4:1

Vccaux_io: Vccaux_io must be set to 2.0V in the High Performance banks for the highest data rates. Vccaux_io is not available in the High Range banks. Note that Vccaux_io is common to groups of banks. Consult the 7 Series Datasheets and FPGA SelectIO Resources User Guide for more information. 1.8V

Memory Type: Select the memory type. Type(s) marked with a warning symbol are not compatible with the frequency selection above. Components

Memory Part: Select the memory part. Part(s) marked with a warning symbol are not compatible with the frequency selection above. Find an equivalent part or create a part using the "Create Custom Part" button if the part needed is not listed here. The "Create Custom Part" feature is not supported for RDRAM II. MT41J128M8JC-125 Create Custom Part

Memory Voltage: Select the Voltage of the Memory part selected. 1.5V

Data Width: Select the Data Width. Parts marked with a warning symbol are not compatible with the frequency and memory part selected above. 16

ECC: MIG supports ECC for 72 bit data width configuration. To be able to select ECC, select a data width that has ECC supported. Disabled

Data Mask: Enable or disable the generation of Data Mask (DM) pins using this check box. This option can be selectable only if the memory part selected has DM pins. Uncheck this box to not use data masks and save FPGA I/Os that are used for DM signals. ECC designs (DDR3 SDRAM, DDR2 SDRAM) will not use Data Mask. ☒

Number of Bank Machines: This parameter defines the number of bank machines. A given bank machine manages a single DRAM bank at any given time. 4

ORDERING: Normal mode allows the memory controller to reorder commands to the memory to obtain the highest possible efficiency. Strict mode forces the controller to execute commands in the exact order received. Strict

Memory Details: 1Gb, x8, row:14, col:10, bank:3, data bits per strobe:8, with data mask, single rank, 1.5V

User Guide Version Info

< Back Next > Cancel

图 8-10 新建FPGA工程-5

第八节 DDR3 SDRAM 与FPGA 接口

MT41J128M16HA-125

实例详解-FPGA接口设计

2. 在新建FPGA工程添加DDR3控制器及接口

(6) 在新界面中对MIG (Memory Interface Generator) 的选项如下:

DDR3控制器参数选项:

(a) DDR3控制器的PLL的输入时钟频率100M

(b) 突发类型及长度8, sequential。

(c) 输出电阻 (可编程) RZQ/7

(d) CS# disable

(e) RTT--nominal RZQ/4

(f) Memory Address Mapping

Selection:

[ROW:BANK:COLLUM]?

如图: 8-10

--> Next

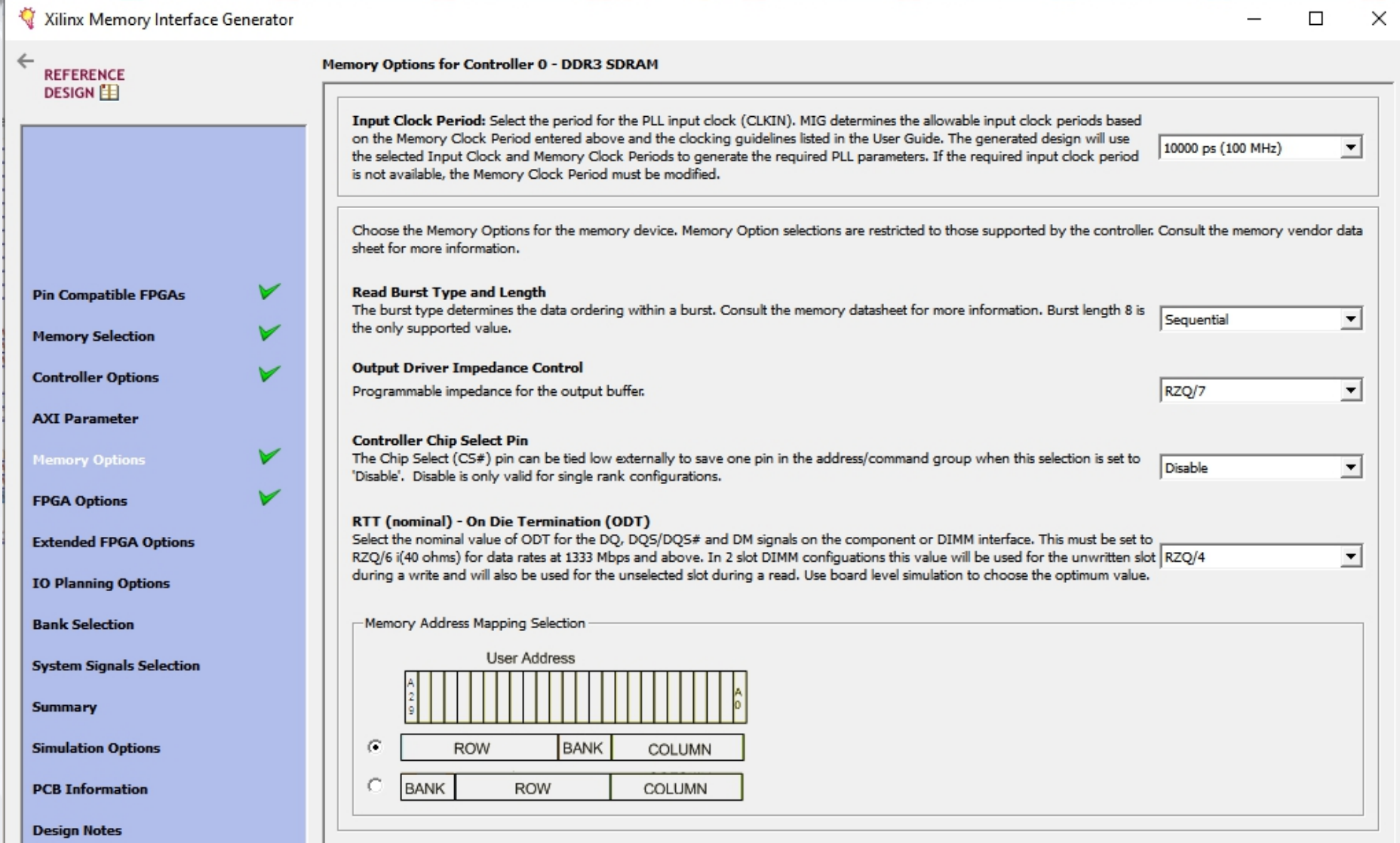


图 8-11 新建FPGA工程-6

第八节 DDR3 SDRAM 与FPGA 接口

MT41J128M16HA-125

实例详解-FPGA接口设计

2. 在新建FPGA工程添加DDR3
控制器及接口

(7) 在新界面中对MIG (Memory Interface Generator) 的选项如下:

DDR3控制器参数选项:

- (a) 系统时钟: No buffer
 - (b) 参考时钟: No buffer
 - (c) 系统复位的极性
 - (d) Debug Signals Control: Disable
 - (e) 内/外部 Vref, ---内部
 - (f) IO降低功耗---ON
 - (g) 使用XADC, 作为温度测量, 用来调整DQS窗口
- Enable

如图: 8-10

--> Next

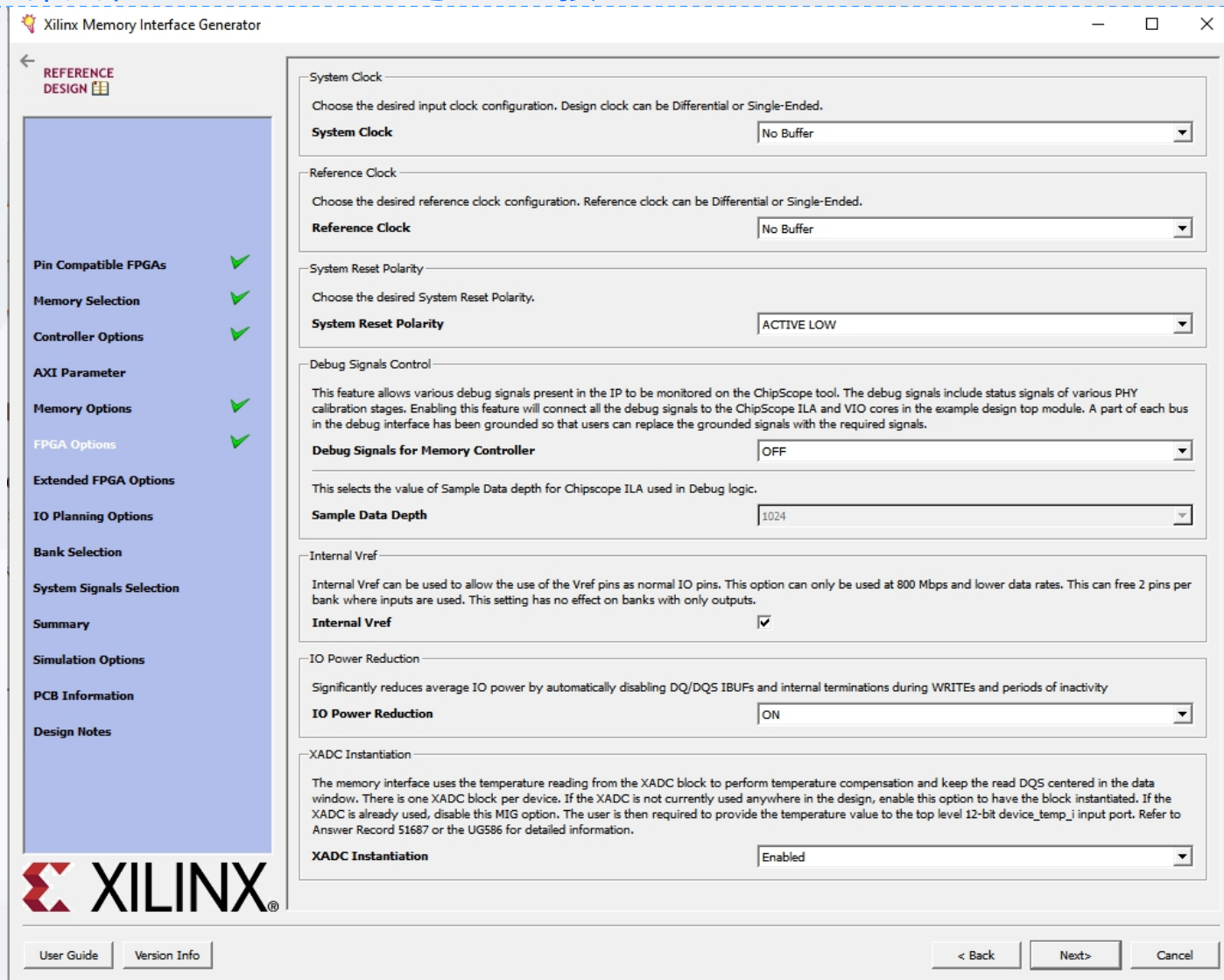


图 8-11 新建FPGA工程-6

MT41J128M16HA-125

实例详解-FPGA接口设计

2. 在新建FPGA工程添加DDR3 控制器及接口

（8）在新界面中对MIG（Memory Interface Generator）的选项如下：

DDR3控制器参数选项：

HR BANK的内部阻抗设置，选择50欧姆

思考题：HR BANK 与HP BANK的区别？

--> Next

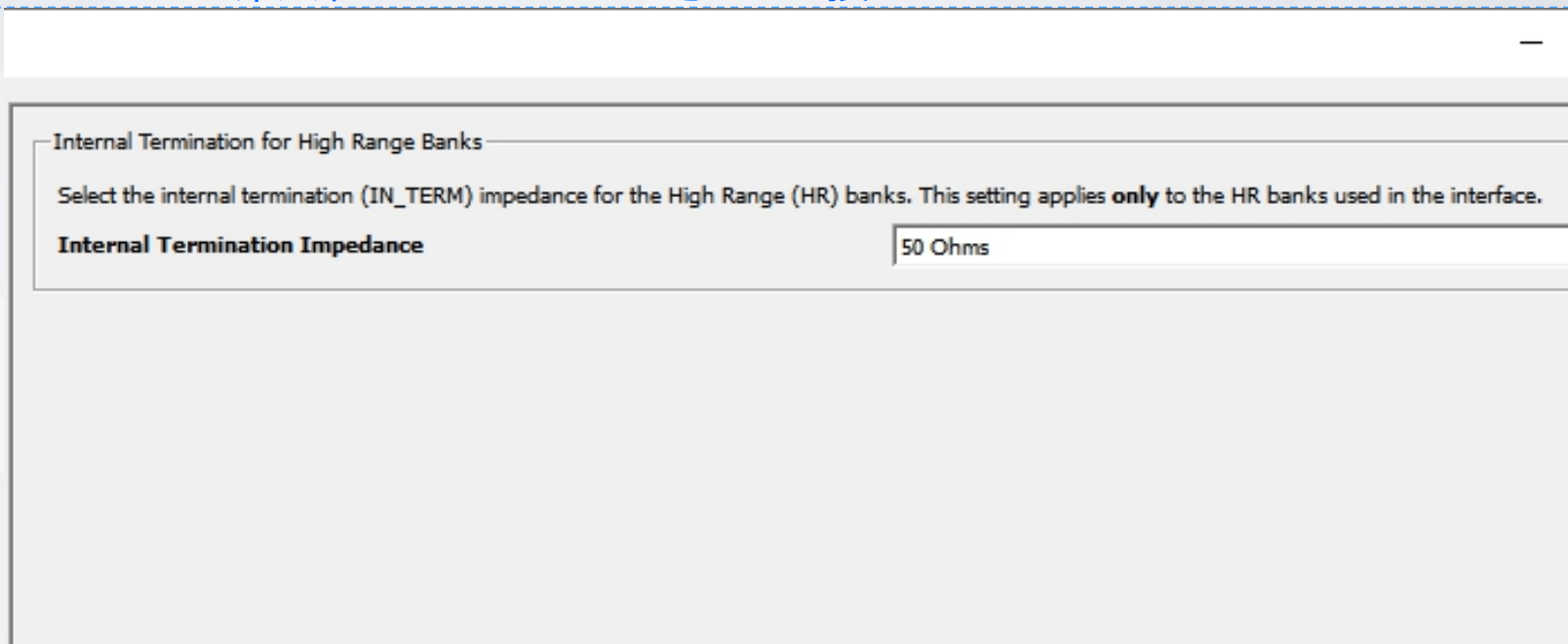


图 8-12 新建FPGA工程-7

MT41J128M16HA-125

实例详解-FPGA接口设计

2. 在新建FPGA工程添加DDR3
控制器及接口

(9) 在新界面中对MIG (Memory Interface Generator) 的选项如下:

DDR3控制器参数选项:

(a) HR BANK的内部阻抗设置,
选择50欧姆

(b) Pin/Bank 选择模式
固定管脚输出模式

思考题: HR BANK 与HP BANK的区别?

--> Next

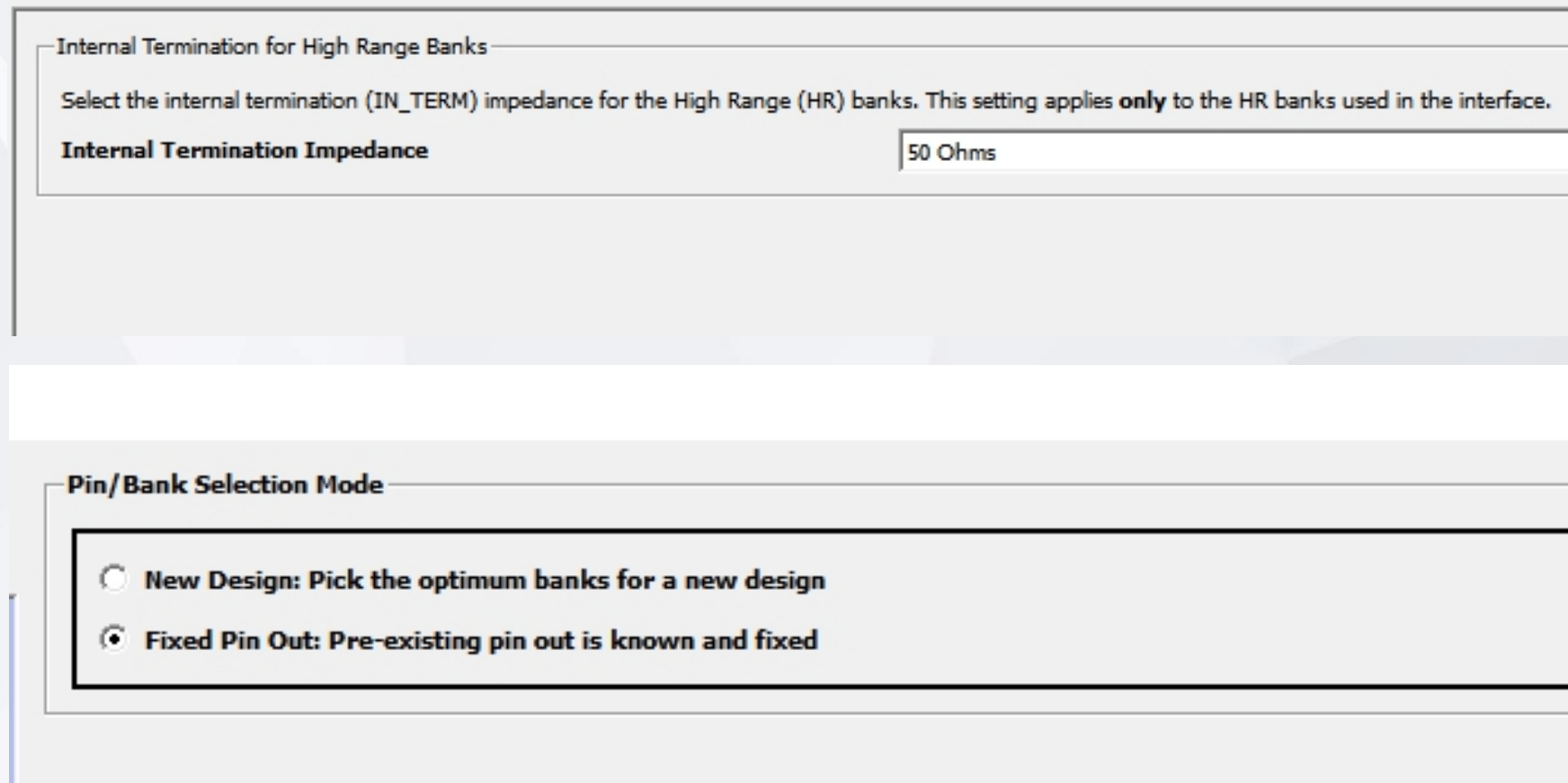


图 8-12 新建FPGA工程-7

第八节 DDR3 SDRAM 与FPGA 接口

MT41J128M16HA-125

实例详解-FPGA接口设计

2. 在新建FPGA工程添加DDR3
控制器及接口

(10) 在新界面中对MIG (Memory Interface Generator) 的选项如下:

DDR3控制器参数选项:

(a) 根据图纸锁定各个管脚。

如果已经有XDC (UCF) 文件也可以直接读取文件加载

(b) validate

--> Next

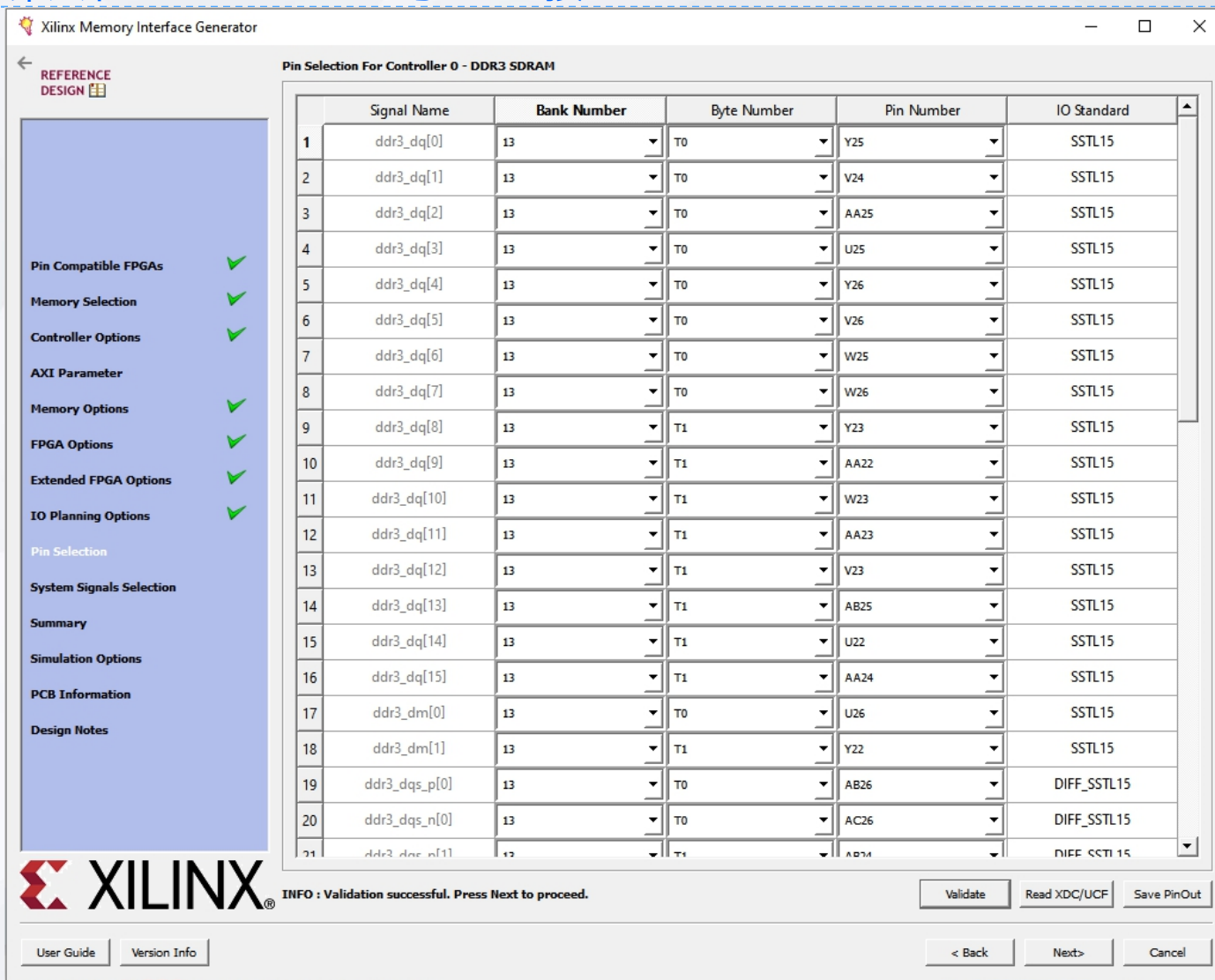


图 8-13 新建FPGA工程-8

第八节 DDR3 SDRAM 与FPGA 接口

MT41J128M16HA-125

实例详解-FPGA接口设计

2. 在新建FPGA工程添加DDR3
控制器及接口

(11) 在新界面中对MIG
(Memory Interface Generator) 的
选项如下:

DDR3控制器参数选项:

(a) 根据图纸锁定各个

管脚。

如图: 8-14

--> Next

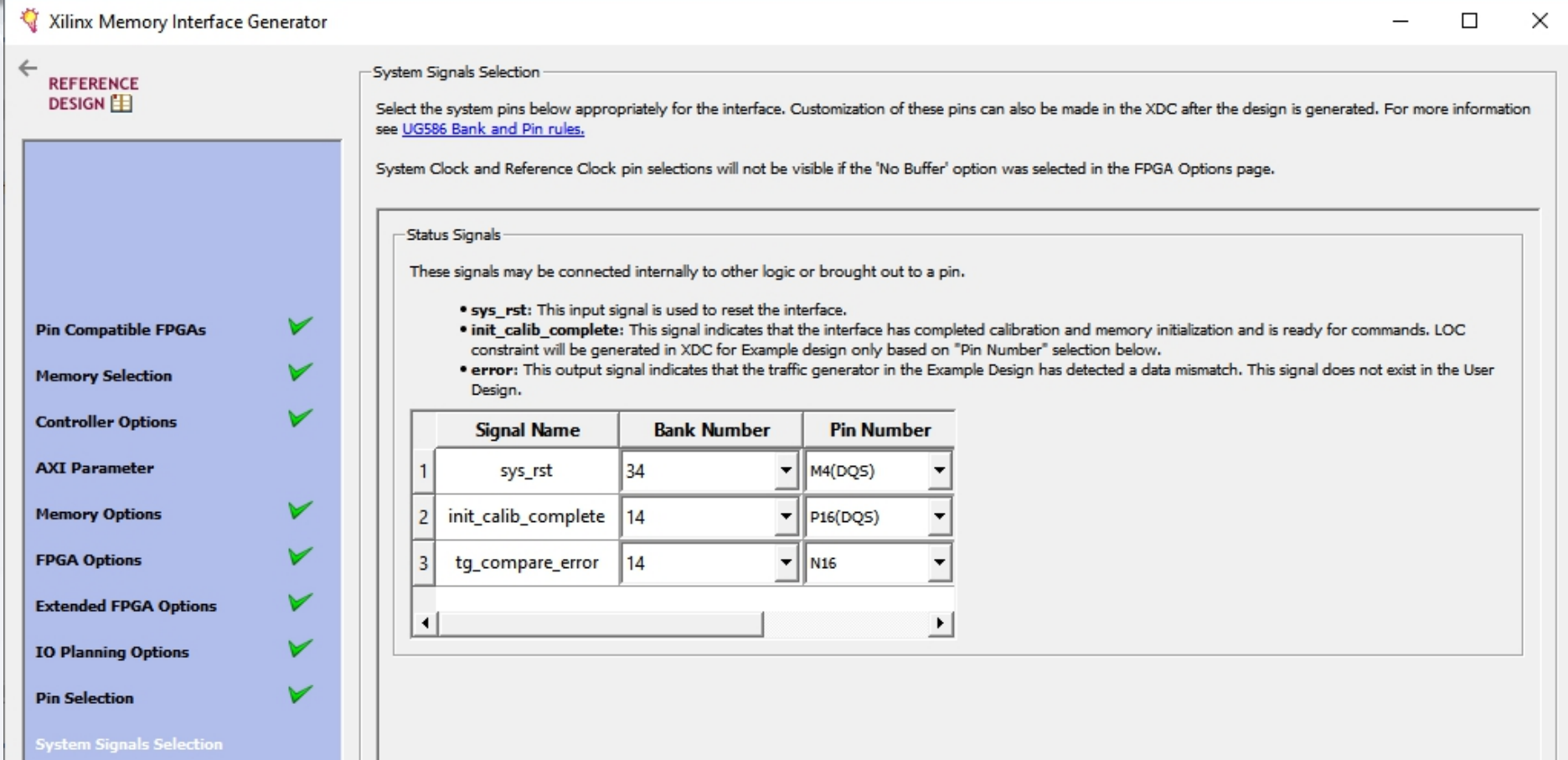


图 8-14 新建FPGA工程-9

第八节 DDR3 SDRAM 与FPGA 接口

MT41J128M16HA-125

实例详解-FPGA接口设计

2. 在新建FPGA工程添加DDR3
控制器及接口

(12) 在新界面中对MIG
(Memory Interface Generator) 的
选项如下:

DDR3控制器参数选项:

(a) Summary

如图: 8-15

--> Next

Accept

--> Next

Generate

--> Next

完成接口设计

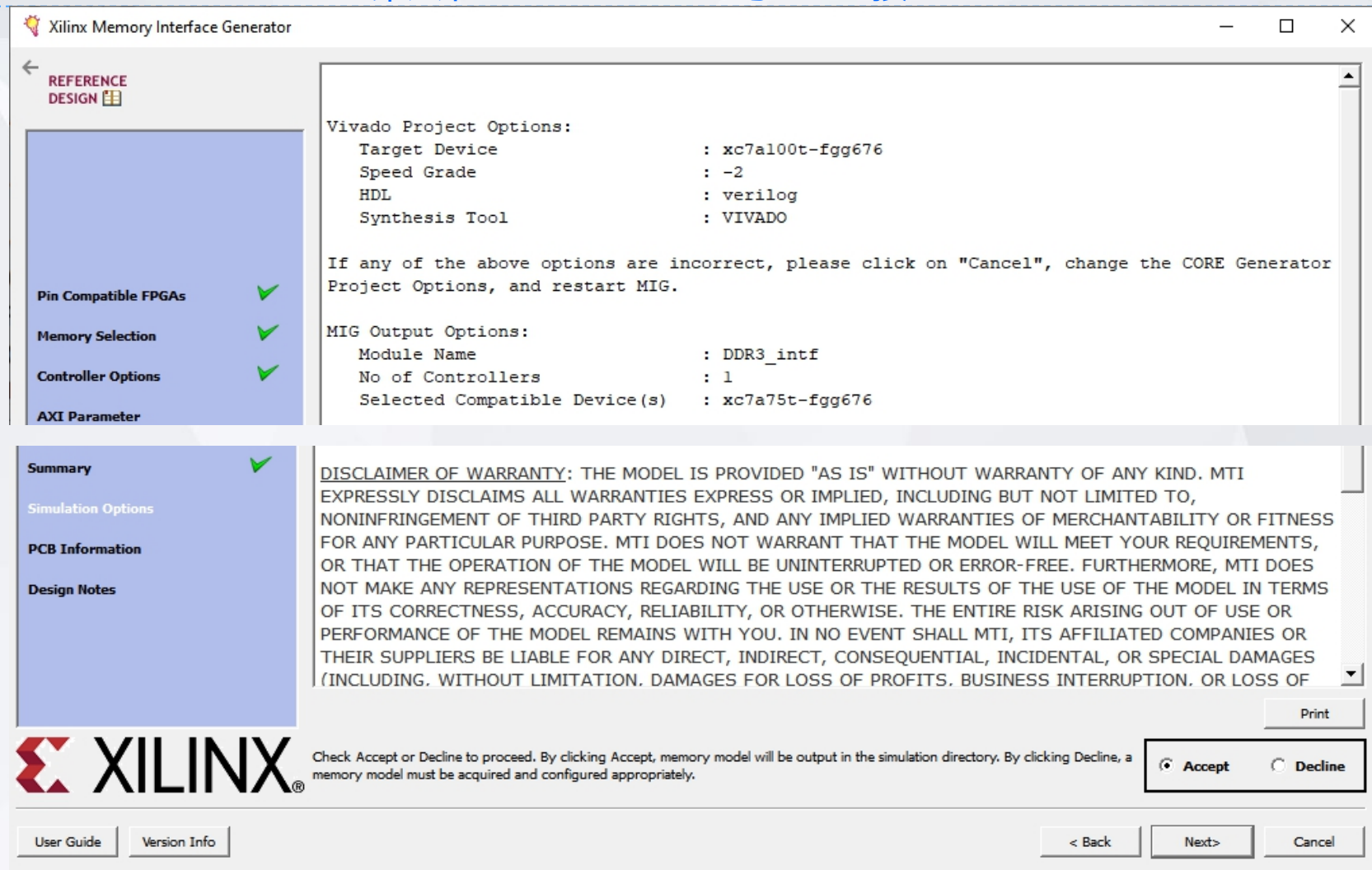


图 8-15 新建FPGA工程-10



第八节 DDR3 SDRAM 与FPGA 接口

MT41J128M16HA-125

实例详解-FPGA接口设计

2. 在新建FPGA工程添加DDR3
控制器及接口

(13) 在新界面中对MIG (Memory Interface Generator) 生成的模块进行操作

(1) 新生成的IP core (DDR3) 如右图所示 (图8-16)

- (2) File->Text Editor->new file
- (3) 打开 ddr3_intf(DDR3_intf.v)
- (4) 复制接口到新建的文件中
- (5) 存储新文件为ddr3_top.v

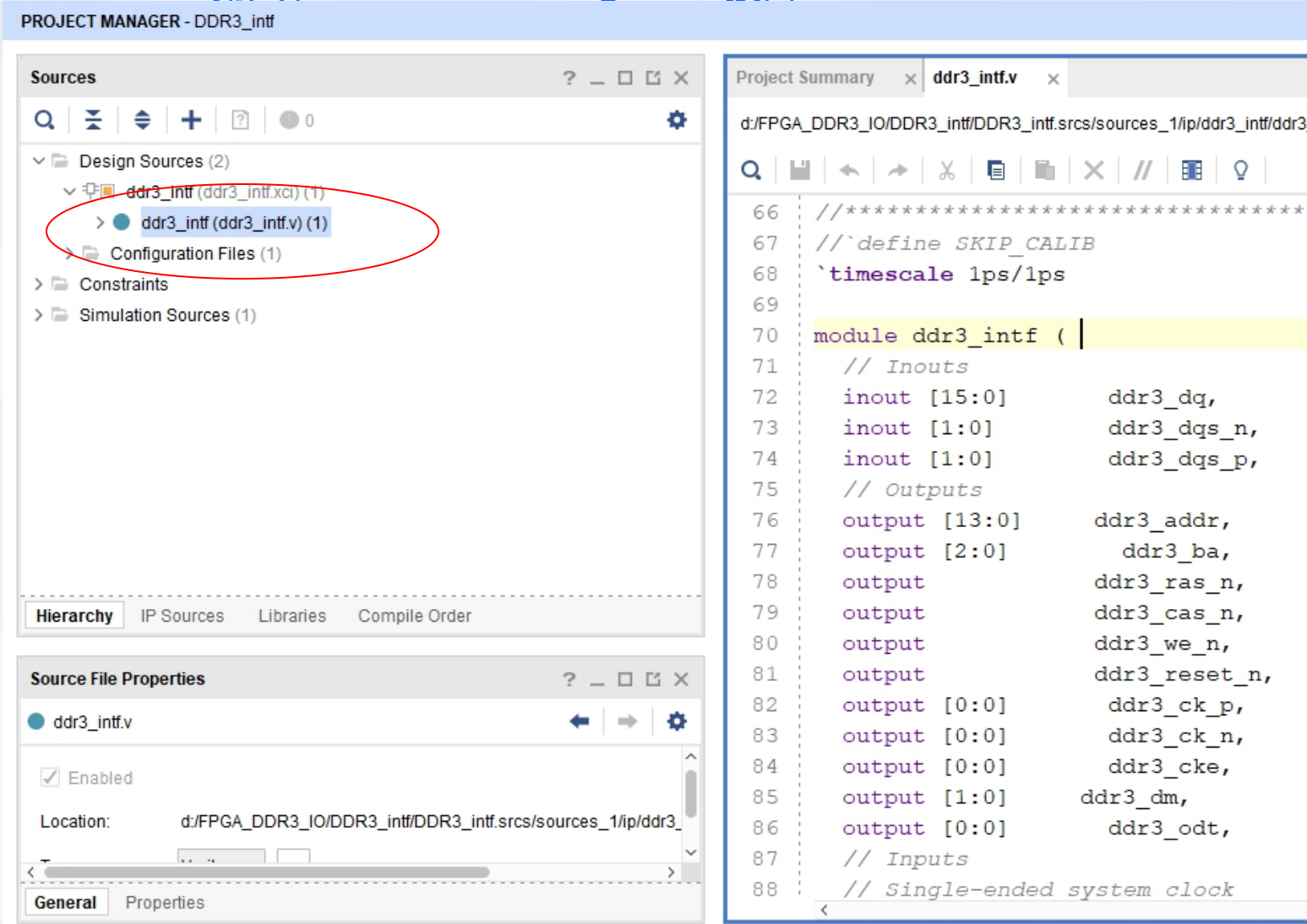


图 8-16 DDR3 IP

第八节 DDR3 SDRAM 与FPGA 接口

MT41J128M16HA-125

实例详解-FPGA接口设计

2. 在新建FPGA工程添加DDR3
控制器及接口

(13) DDR3编程及应用:

(1) 在已经创建的目录中找到ddr3_intf.veo文件并打开, 复制模块的层次文件, 如图8-17

如本机的目录如下:

D:\FPGA_DDR3_IO\DDR3_intf\DDR3_intf.srcs\sources_1\ip\ddr3_intf

(2) 添加PLL模块

- (1) 输入时钟50Mhz, 单端输入
- (2) 输出时钟100Mhz, 命名sys_clk
- (3) 输出时钟200Mhz, clk_ref, 主要用做IO delay矫正
- (4) 定义wire 变量sys_clk, clk_ref, pll_locked, nsys_rst;
- (5) 系统复位信号如下:

```
always@(posedge sys_clk)
begin
nsys_rst<=pll_locked;
end
```

```
ddr3_intf u_ddr3_intf (
// Memory interface ports
.ddr3_addr
.ddr3_ba
.ddr3_cas_n
.ddr3_ck_n
.ddr3_ck_p
.ddr3_cke
.ddr3_ras_n
.ddr3_reset_n
.ddr3_we_n
.ddr3_dq
.ddr3_dqs_n
.ddr3_dqs_p
.init_calib_complete
.ddr3_dm
.ddr3_odt
// Application interface ports
.app_addr
.app_cmd
.app_en
.app_wdf_data
.app_wdf_end
.app_wdf_wren
.app_rd_data
.app_rd_data_end
.app_rd_data_valid
.app_rdy
.app_wdf_rdy
.app_sr_req
.app_ref_req
.app_zq_req
.app_sr_active
.app_ref_ack
.app_zq_ack
.ui_clk
.ui_clk_sync_rst
.app_wdf_mask
// System Clock Ports
.sys_clk_i
// Reference Clock Ports
.clk_ref_i
.sys_rst
);

(ddr3_addr), // output [13:0] ddr3_addr
(ddr3_ba), // output [2:0] ddr3_ba
(ddr3_cas_n), // output ddr3_cas_n
(ddr3_ck_n), // output [0:0] ddr3_ck_n
(ddr3_ck_p), // output [0:0] ddr3_ck_p
(ddr3_cke), // output [0:0] ddr3_cke
(ddr3_ras_n), // output ddr3_ras_n
(ddr3_reset_n), // output ddr3_reset_n
(ddr3_we_n), // output ddr3_we_n
(ddr3_dq), // inout [15:0] ddr3_dq
(ddr3_dqs_n), // inout [1:0] ddr3_dqs_n
(ddr3_dqs_p), // inout [1:0] ddr3_dqs_p
(init_calib_complete), // output init_calib_complete

(ddr3_dm), // output [1:0] ddr3_dm
(ddr3_odt), // output [0:0] ddr3_odt

(app_addr), // input [27:0] app_addr
(app_cmd), // input [2:0] app_cmd
(app_en), // input app_en
(app_wdf_data), // input [127:0] app_wdf_data
(app_wdf_end), // input app_wdf_end
(app_wdf_wren), // input app_wdf_wren
(app_rd_data), // output [127:0] app_rd_data
(app_rd_data_end), // output app_rd_data_end
(app_rd_data_valid), // output app_rd_data_valid
(app_rdy), // output app_rdy
(app_wdf_rdy), // output app_wdf_rdy
(app_sr_req), // input app_sr_req
(app_ref_req), // input app_ref_req
(app_zq_req), // input app_zq_req
(app_sr_active), // output app_sr_active
(app_ref_ack), // output app_ref_ack
(app_zq_ack), // output app_zq_ack
(ui_clk), // output ui_clk
(ui_clk_sync_rst), // output ui_clk_sync_rst
(app_wdf_mask), // input [15:0] app_wdf_mask

(sys_clk_i),
(clk_ref_i),
(sys_rst) // input sys_rst
);
```

图 8-17 DDR3 模块层次接口



MT41J128M16HA-125-- 实例详解-FPGA接口设计信号解释

Signal name	Direction	Description	描述
app_addr[ADDR_WIDTH – 1:0]	input	This input indicates the address for the current request.	当前请求地址，[ROW+BA+COL)
app_cmd[2:0]	Input	This input selects the command for the current request.	当前请求命令，000--写，001--读
app_en	Input	This is the active-High strobe for the app_addr[], app_cmd[2:0], app_sz, and app_hi_pri inputs.	高电平有效的选通信号，使能APP的地址，命令等
app_rdy	Output	This output indicates that the UI is ready to accept commands. If the signal is deasserted when app_en is enabled, the current app_cmd and app_addr must be retried until app_rdy is asserted.	指示当前的UI准备好接收新命令，如果在app_en使能后该信号撤销，命令及地址需重新发送
app_hi_pri	Input	This active-High input elevates the priority of the current request.	提升当前命令的优先级，常规设计中一般不使用
app_rd_data [APP_DATA_WIDTH – 1:0]	Output	This provides the output data from read commands.	读到的数据
app_rd_data_end	Output	This active-High output indicates that the current clock cycle is the last cycle of output data on app_rd_data[]. This is valid only when app_rd_data_valid is active-High.zai	当前时钟的数据为最后一笔数据，只有在app_rd_data_valid 为高电平时才有效
app_rd_data_valid	Output	This active-High output indicates that app_rd_data[] is valid.	高电平指示app_rd_data[] 有效
app_sz	Input	This input is reserved and should be tied to 0.	保留，输入0

表 8-1 DDR3 接口应用层命令



MT41J128M16HA-125-- 实例详解-FPGA接口设计信号解释

Signal name	Direction	Description	描述
app_wdf_data [APP_DATA_WIDTH – 1:0]	input	This provides the data for write commands.	写命令之后跟的写数据
app_wdf_end	Input	This active-High input indicates that the current clock cycle is the last cycle of input data on app_wdf_data[].	当前时钟的写数据为最后写命令的最后一笔数据
app_wdf_mask [APP_MASK_WIDTH – 1:0]	Input	This provides the mask for app_wdf_data[].	每个字节对应的屏蔽位
app_wdf_rdy	Output	This output indicates that the write data FIFO is ready to receive data. Write data is accepted when app_wdf_rdy = 1'b1 and app_wdf_wren = 1'b1.	指示写数据缓冲区FIFO可以接收新数据，数据在app_wdf_rdy = 1'b1与app_wdf_wren = 1'b1.都有效是写数据才会被接受
app_wdf_wren	Input	This is the active-High strobe for app_wdf_data[].	写使能
app_correct_en_i	Input	When asserted, this active-High signal corrects single bit data errors. This input is valid only when ECC is enabled in the GUI. In the example design, this signal is always tied to 1.	在ECC有效的情况下，如果该位有效可以纠正一位错误
app_sr_req	Input	This input is reserved and should be tied to 0.shu	保留，接0
app_sr_active	Output	This output is reserved.	保留

MT41J128M16HA-125-- 实例详解-FPGA接口设计信号解释

Signal name	Direction	Description	描述
app_ref_ack	Output	This active-High output indicates that the Memory Controller has sent the requested refresh command to the PHY interface	高电平指示刷新命令已经送到DDR接口
app_zq_req	Input	This active-High input requests that a ZQ calibration command be issued to the DRAM.	数据端口校准命令
ui_clk	Output	This UI clock must be a half or quarter of the DRAM clock.	用户时钟，必须是一半或1/4 的DDR时钟频率
init_calib_complete	Output	PHY asserts init_calib_complete when calibration is finished.	校准完成后输出高电平
app_ecc_multiple_err[7:0]	Output	This signal is applicable when ECC is enabled and is valid along with app_rd_data_valid. The app_ecc_multiple_err[3:0] signal is non-zero if the read data from the external memory has two bit errors per beat of the read burst. The SECCDED algorithm does not correct the corresponding read data and puts a non-zero value on this signal to notify the corrupted read data at the UI.	只有在ECC使能的情况下有效，在每个突发读时app_ecc_multiple_err[3:0]在检查到数据有2位错时输出非零值。如果采用SECCDED算法，该机制并不纠错，只是输出一个非零值。
ui_clk_sync_rst	Output	This is the active-High UI reset.	高电平有效的UI复位信号
app_ecc_single_err[7:0]	Output	This signal is applicable when ECC is enabled and is valid along with app_rd_data_vali. The app_ecc_single_err signal is non-zero if the read data from the external memory has a single bit error per beat of the read burst.	只有在ECC使能的情况下有效，在每个突发读时app_ecc_multiple_err[3:0]在检查到数据有1位错时输出非零值。如果采用SECCDED算法，该机制并不纠错，只是输出一个非零值。

表 8-3 DDR3 接口应用层命令



MT41J128M16HA-125-- FPGA应用程序设计，信号定义

接口信号定义

Wire		init_calib_complete ; // output	init_calib_complete
reg	[27:0]	app_addr; // input [27:0]	app_addr
wire	[2:0]	app_cmd; // input [2:0]	app_cmd
wire		app_en; // input	app_en
wire		app_rdy; // output	app_rdy
wire	[127:0]	app_wdf_data; // input [127:0]	app_wdf_data
wire		app_wdf_end; // input	app_wdf_end
wire		app_wdf_wren; // input	app_wdf_wren
wire		app_wdf_rdy ; // output	app_wdf_rdy
wire	[127:0]	app_rd_data; // output [127:0]	app_rd_data
wire		app_rd_data_end; // output	app_rd_data_end
wire		app_rd_data_valid ; // output	app_rd_data_valid
wire	[15:0]	app_wdf_mask; // input [15:0]	app_wdf_mask
wire		ui_clk ; // output	ui_clk
wire		ui_clk_sync_rst; // output	ui_clk_sync_rst
// System Clock Ports			
wire		sys_clk;	
// Reference Clock Ports			
wire		clk_ref ;	
wire		pll_locked;	
wire		nsys_rst=pll_locked;	



MT41J128M16HA-125-- FPGA应用程序设计，PLL接口

利用PLL生成sys_clk 100MHz, clk_ref 200MHz

```
sys_pll sys_pll_inst
(
    // Clock out ports
    .clk_out1    (sys_clk),
    .clk_out2    (clk_ref),
    // Status and control signals
    .reset       (1'b1),
    .locked       (pll_locked),
    // Clock in ports
    .clk_in1      (inclk)
);
```



MT41J128M16HA-125-- FPGA应用程序设计，控制及测试接口

```
ddr_ctrl ddr_ctrl_inst
(
    .app_addr      (app_addr),
    .app_cmd       (app_cmd),
    .app_en        (app_en),
    .app_rdy       (app_rdy),
    .app_wdf_data  (app_wdf_data),
    .app_wdf_end   (app_wdf_end),
    .app_wdf_wren  (app_wdf_wren),
    .app_wdf_rdy   (app_wdf_rdy),
    .app_rd_data   (app_rd_data),
    .app_rd_data_end (app_rd_data_end),
    .app_rd_data_valid (app_rd_data_valid),
    .clk           (ui_clk),
    .rst_n         (ui_clk_sync_rst),
    .error_flag    (error_flag),
    .end_flag      (end_flag)
);
```



第八节 DDR3 SDRAM 与FPGA 接口

MT41J128M16HA-125-- FPGA应用程序设计，控制及测试接口

DDR控制器接口，分为3组，地址/命令， 写， 读

```
module ddr_ctrl(
    (* MARK_DEBUG="true" *) output reg [28:0]    app_addr           ,
    (* MARK_DEBUG="true" *) output reg [2:0]      app_cmd           ,
    (* MARK_DEBUG="true" *) output reg           app_en             ,
    (* MARK_DEBUG="true" *) input                app_rdy            ,//地址/命令组
    (* MARK_DEBUG="true" *) output reg [255:0]    app_wdf_data      ,
    (* MARK_DEBUG="true" *) output               app_wdf_end        ,
    (* MARK_DEBUG="true" *) output reg           app_wdf_wren       ,
    (* MARK_DEBUG="true" *) input                app_wdf_rdy        , //写数据组
    (* MARK_DEBUG="true" *) input [255:0]         app_rd_data       ,
    (* MARK_DEBUG="true" *) input                app_rd_data_end    ,
    (* MARK_DEBUG="true" *) input                app_rd_data_valid  ,//读数据组
    input                                         clk                ,
    input                                         rst_n              ,
    (* MARK_DEBUG="true" *) output reg           error_flag         = 0 ,
    output reg                                   end_flag            = 0
);
```



MT41J128M16HA-125-- FPGA应用程序设计，控制及测试接口

参数定义，通过计时器分离读写状态。

```
parameter BURST_WR_128BIT = 16777216 ;
parameter BURST_RD_128BIT = 16777216 ;
parameter DATA            = 128'habcdef1234567890abcdef1234567890 ;    //测试数据
(* MARK_DEBUG="true" *) wire time_wrreq  = ( scnt == 28'd50_000_000 ) ;
(* MARK_DEBUG="true" *) wire time_rdreq  = ( scnt == 28'd00_000_000 ) ;
reg [27:0] scnt = 0 ;           //100M 时钟, // Read and write once every 1 second
(* MARK_DEBUG="true" *) reg[3:0] cstate;
reg[3:0] nstate;
(* MARK_DEBUG="true" *) reg [31:0] num = 0 ;
(* MARK_DEBUG="true" *) reg [31:0] wrnum = 0 ;
always @ (posedge clk)
if (!rst_n) begin
    scnt      <= 0 ;
    end_flag <= 0 ;
end
else if (scnt < 28'd199_999_999) begin
    scnt      <= scnt + 1;
    end_flag <= 0 ;
end
else begin
    scnt <= 0 ;
    end_flag <= ~end_flag ;
end
```



MT41J128M16HA-125-- FPGA应用程序设计，控制及测试接口

DDR控制器读写状态机， 0状态判断读/写触发条件，1状态为数据写状态，2状态是读状态

```
always @ (posedge clk )
if (!rst_n)      cstate <= 0 ;
else cstate <=nstate ;
always @(cstate , time_wrreq ,time_rdreq ,num ,app_rdy ,app_wdf_rdy)
case (cstate)
  0 : begin
      if ( time_wrreq )          nstate <= 1 ;
      else if (time_rdreq )      nstate <= 2 ;
      else  nstate <= 0 ;
    end
  1: begin
      if (( wrnum > BURST_WR_128BIT) && (num >BURST_WR_128BIT ) )
        nstate <= 3 ;
      else
        nstate <= 1 ;
      end
  2: begin
      if (num > BURST_RD_128BIT)
        nstate <= 3 ;
      else
        nstate <= 2 ;
      end
  3:  nstate <= 0 ;
endcase
```



MT41J128M16HA-125-- FPGA应用程序设计，控制及测试接口

应用程序与 DDR控制器交互，命令/地址 计数器

```
always @ (posedge clk)
```

```
    if (!rst_n)
        num <= 32'd0 ;
    else
        if (cstate ==1)
            begin
                if ( app_rdy & app_en )
                    num <= num+ 1 ;
            end
        else
            if ( cstate == 2)
                begin
                    if (app_rdy & app_en)
                        num <= num +1 ;
                end
            else num <= 0 ;
```



MT41J128M16HA-125-- FPGA应用程序设计，控制及测试接口

写数据计数器更新：

```
always @ ( posedge clk)
```

```
if (!rst_n) wrnum <= 32'd0 ;
```

```
else  
    if (cstate == 1)  
        begin  
            if (app_wdf_rdy & app_wdf_wren)  
                wrnum <= wrnum +1 ;
```

```
        end  
    else wrnum <= 0 ;
```



MT41J128M16HA-125-- FPGA应用程序设计，控制及测试接口

应用程序与DDR控制器交互，读命令与地址：

```
always@ (posedge clk )
if (~rst_n)
begin
    app_cmd <= 3'd0 ;
    app_en  <= 1'b0 ;
    app_addr <=27'd0 ;
end
else begin
    if ( cstate == 1)
        begin
            app_cmd <= 3'b000 ;
            if ( num < BURST_WR_128BIT )
                app_en  <= 1 ;
            else
                app_en  <= 0 ;
            if (app_rdy & app_en)
                app_addr[26:3] <= app_addr[26:3]+1 ; // app_addr[10:3] +1'b1 ;
        end
    end
```



MT41J128M16HA-125-- FPGA应用程序设计，控制及测试接口

应用程序与DDR控制器交互，写命令与地址：

```
else      begin
    if (cstate == 2)
        begin
            app_cmd <= 3'b001 ;
            if (num <BURST_RD_128BIT)
                app_en <= 1'b1 ;
            else
                app_en <= 1'b0 ;
            if (app_rdy&app_en)          //注意只有 app_rdy与app_en同时为1时，地址才能更新。
                app_addr[26:3] <= app_addr[26:3] + 1 ;
        end
    else
        begin
            app_cmd <= 3'd0 ;
            app_en    <= 1'b0 ;
            app_addr <= 27'd0 ;
        end
    end
end
end
```



MT41J128M16HA-125-- FPGA应用程序设计，控制及测试接口

写过程：应用程序与DDR3控制器交互

```
always @ ( posedge clk)
  if (!rst_n)
    begin
      app_wdf_data <= 128'd0 ;
      app_wdf_mask<=16'd0;
      app_wdf_wren <= 0 ;
    end
  else begin
    app_wdf_mask<=16'd0;
    if(cstate == 1) begin

      app_wdf_data <= DATA ;
      if (wrnum < BURST_WR_128BIT )
        app_wdf_wren <= 1'b1 ;
      else
        app_wdf_wren <= 1'b0 ;
    end

    ;
  end
else begin
  app_wdf_data <= 128'd0 ;
  app_wdf_wren <= 0 ;
end
end
```

注意：如果**DATA**是变量需要更新，一定要判断app_wdf_wren&app_wdf_rdy为真才可以，如下例：

```
if (app_wdf_wren&app_wdf_rdy)
  DATA<=DATA_NEW;
```

原因是只有app_wdf_wren&app_wdf_rdy为真时，DATA才能写到DDR控制中，DATA才能更新为新数据，否则维持元数据等待条件成立时将没有成功写入的DATA数据重新写入。



MT41J128M16HA-125-- FPGA应用程序设计，控制及测试接口

```
assign app_wdf_end = 1'b1;
```

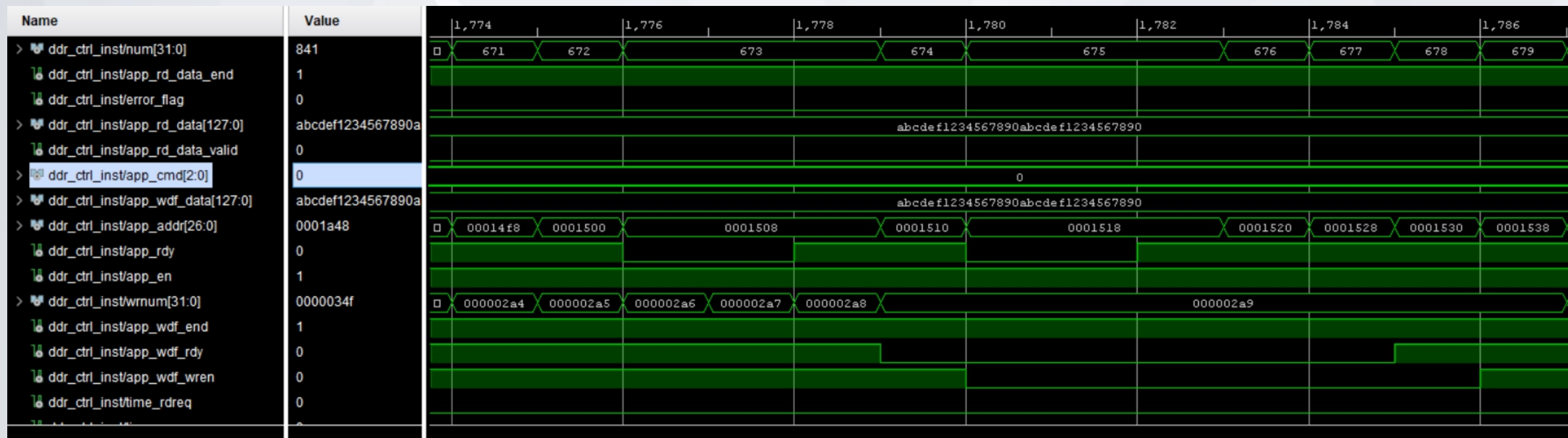
读数据与原始数据对比，检查错误

```
always @(posedge clk or negedge rst_n)
    if(!rst_n) begin
        error_flag <= 0;
    end
    else begin
        if(app_rd_data_valid&&(app_rd_data!= DATA))
            error_flag <= 1 ;
        else
            error_flag <=0 ;
    end
```



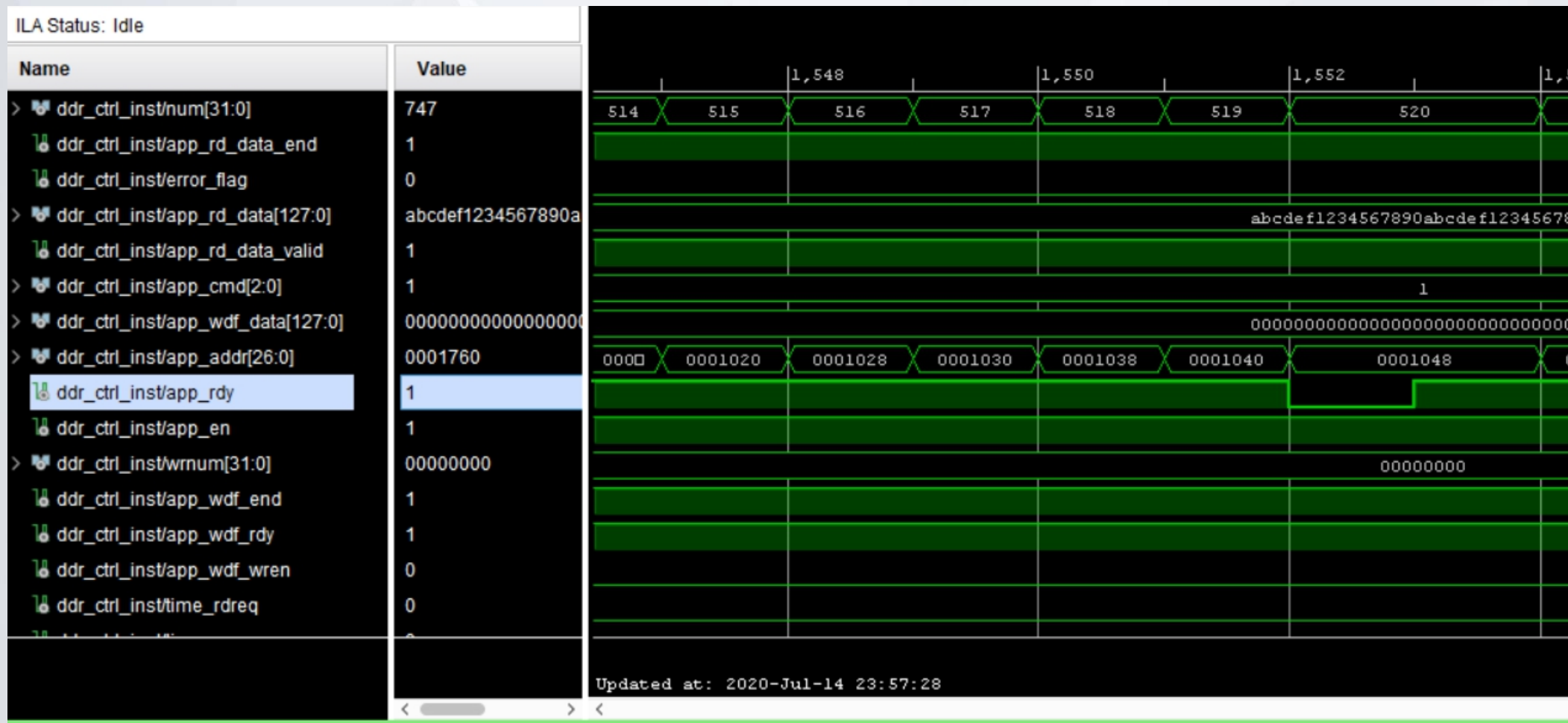
第八节 DDR3 SDRAM 与FPGA 接口

MT41J128M16HA-125-- FPGA应用程序设计，控制及测试接口



8-18 写波形图

MT41J128M16HA-125-- FPGA应用程序设计，控制及测试接口



8-19 读波形图

THANK

YOU!



知识库网站: <http://www.icfedu.cn/>



淘宝店铺: <https://shop296232497.taobao.com/>

