

Logos 系列产品 HMIC_H IP

用户指南

(UG022001, Version 1.2)

(2019.08.30)

文档版本修订记录

日期	版本	修订记录
2019/3/1	1.0	初始版本
2019/5/8	1.1	1.更新第四章第一节中配置 Logos HMIC_H IP 参数界面的截图 2.更新第四章第一节中 Basic Options 页面的截图 3.更新第四章第一节中 Memory Option 页面的截图 4.更新第四章第一节中 Interface Options 页面的截图 5.更新第四章第一节中 Summary 页面的截图
2019/8/30	1.2	1.IP 更名为 HMIC_H 2.更新第四章第一节中配置 Logos HMIC_H IP 参数界面的截图 3.更新第四章第一节中 Basic Options 页面的截图 4.更新第四章第一节中 Memory Option 页面的截图 5.更新第四章第一节中 Interface Options 页面的截图 6.更新第四章第一节中 Summary 页面的截图

名词术语解释

HMIC: High performance Memory Interface Controller

WR: Write

RD: Read

DFI: DDR PHY Interface

DDR: Double Data Rate

AXI: AdvancedeXtensible Interface

APB: Advanced Peripheral Bus

MC: Memory Controller

DDC: Dedicated DQS Circuit

目 录

一 概述.....	1
二 系统介绍.....	2
三 接口说明.....	3
(一) Memory 接口.....	3
(二) 全局信号	3
(三) AXI4 接口	4
(四) APB 接口	7
四 模块例化.....	8
(一) 模块配置	8
1. 选择 IP.....	8
2. IP 参数设置.....	8
3. 生成 IP.....	9
4. 配置说明	11
(二) 运行与仿真.....	15
(三) 综合与布局布线.....	16
五 接口时序描述.....	17
(一) AXI4 接口	17
1. AXI4 读时序	17
2. AXI 写时序	18
(二) APB 接口	19
1. APB 接口写时序.....	19
2. APB 接口读时序.....	19
六 Example Design	21
(一) 设计框图.....	21
(二) 测试方法.....	21
七 配置寄存器.....	23
(一) MSTR: DDRC 工作模式配置寄存器	23
(二) STAT: DDRC 工作状态寄存器	24
八 注意事项.....	25

(一) 时序约束.....	25
(二) 位置约束.....	27
九 附录.....	28

表目录

表 1	Memory 接口	3
表 2	全局信号	3
表 3	AXI4 Port0	4
表 4	AXI4 Port1	5
表 5	AXI4 Port2	6
表 6	APB 接口	7
表 7	Basic Options 参数说明	11
表 8	Memory Options 参数说明	13
表 9	Interface Options 参数说明	14
表 10	MSTR 各 bit 定义	23
表 11	STAT 各 bit 定义	24

图目录

图 1	HMIC_H IP 系统框图	2
图 2	选择 Logos HMIC_H IP 界面	8
图 3	配置 Logos HMIC_H IP 参数界面	9
图 4	Logos HMIC_H IP 生成报告界面	10
图 5	Basic Options 页面.....	11
图 6	Memory Options 页面.....	12
图 7	Interface Options 页面	14
图 8	Summary 页面.....	15
图 9	打开 ModelSim 指令	15
图 10	ModelSim 执行仿真脚本	16
图 11	AXI4 单次读时序	17
图 12	AXI4 连续读时序	18
图 13	AXI4 单次写时序	18
图 14	APB 接口 write 数据	19
图 15	APB 接口 read 数据.....	20
图 16	Example 设计系统框图	21
图 17	测试流程图	22
图 18	IP 时钟关系示意图.....	25
图 19	IP 时钟连接示意图.....	26

一 概述

HMIC_H IP 是深圳市紫光同创电子有限公司 FPGA 产品中用于实现对 SDRAM 读写而设计的 IP，通过公司 Pango Design Suite 套件（后文简称 PDS）中 IP Compiler 工具（后文简称 IPC）例化生成 IP 模块。

- 支持LPDDR、DDR2、DDR3;
- 支持x8 x16 Memory Device;
- 用户接口
 - 标准的AXI4总线接口
 - ◆ 一组128bit的AXI4 Host Port
 - ◆ 两组64bit的AXI4 Host Port
 - 标准的APB总线接口
 - ◆ DDRC配置接口
- 支持可配低功耗模式：Self-Refresh 和 Power Down;
- 支持 DDR3 的最高数据速率达到 1066Mbps;
- 支持 DDR2 的最高数据速率达到 800Mbps;
- 支持 LPDDR 的最高数据速率达到 400Mbps;
- Burst Length 8 和单 Rank;

本文将从 HMIC_H IP 的接口说明,接口时序,IP 的生成以及 IP 使用等几个方面进行描述。

二 系统介绍

HMIC_H IP 系统框图如下图所示：

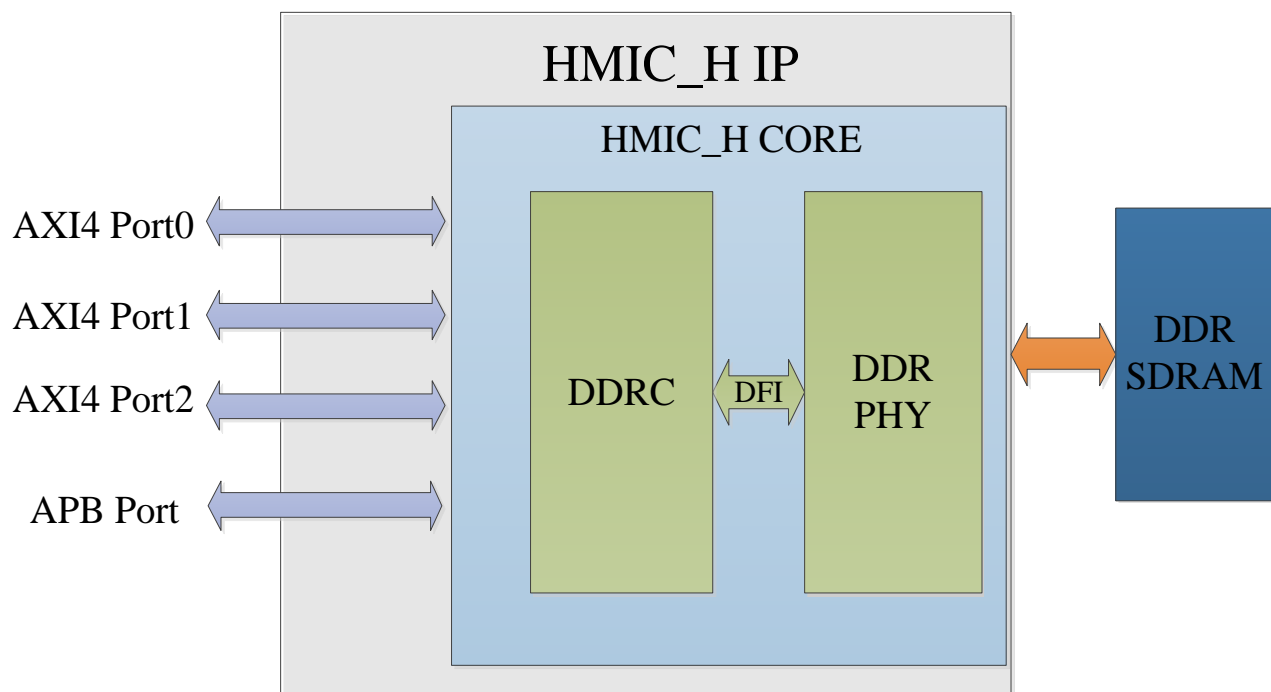


图 1 HMIC_H IP 系统框图

HMIC_H IP 包括了 DDR Controller、DDR PHY 和 PLL，用户通过 AXI4 接口实现数据的读写，通过 APB 接口可配置 DDR Controller 内部寄存器，PLL 用于产生需要的各种时钟。

- AXI4 接口：HMIC_H IP 提供三组 AXI4 Host Port：AXI4 Port0(128bit)、AXI4 Port1(64bit)、AXI4 Port2(64bit)。用户通过 HMIC_H IP 界面可以选择使能这三组 AXI4 Port。三组 AXI4 Host Port 均为标准 AXI4 接口。
- APB 接口：HMIC_H IP 提供一个 APB 配置接口，通过该接口，可配置 DDR Controller 内部寄存器。HMIC_H IP 初始化完成后使能该接口。

三 接口说明

(一) Memory 接口

表 1 Memory 接口

端口名	输入/输出	位宽	有效值	描述
pad_addr_ch0	输出	15		Memory 地址总线
pad_ba_ch0	输出	3		Bank 地址总线
pad_ddr_clk_w	输出	1		Memory 差分时钟正端
pad_ddr_clkn_w	输出	1		Memory 差分时钟负端
pad_cke_ch0	输出	1	高电平	Memory 差分时钟使能
pad_dm_rdq_s_ch0	输出	2	高电平	数据 Mask
pad_odt_ch0	输出	1		On Die Termination
pad_csn_ch0	输出	1	低电平	Memory 片选
pad_rasn_ch0	输出	1	低电平	行地址 strobe
pad_casn_ch0	输出	1	低电平	列地址 strobe.
pad_wen_ch0	输出	1	低电平	写使能
pad_rstn_ch0	输出	1	低电平	Memory 复位
pad_dq_ch0	输入/输出	16		数据总线
pad_dqs_ch0	输入/输出	2		数据时钟正端
pad_dqsn_ch0	输入/输出	2		数据时钟负端
pad_loop_in	输入	1		低位温度补偿输入
pad_loop_in_h	输入	1		高位温度补偿输入
pad_loop_out	输出	1		低位温度补偿输出
pad_loop_out_h	输出	1		高位温度补偿输出

(二) 全局信号

表 2 全局信号

端口名	输入/输出	位宽	有效值	描述
pll_refclk_in	输入	1		外部参考时钟输入
top_rst_n	输入	1	低电平	外部复位输入
pll_lock	输出	1	高电平	HMIC_H 内部 PLL lock 信号。
ddrc_rst	输入	1	高电平	DDRC 的复位输入
ddrphy_rst_done	输出	1	高电平	DDRPHY 复位完成标志
ddrc_init_done	输出	1	高电平	DDRC 的初始化完成标志
pll_aclk_0	输出	1		Axi4 Port0 的时钟
pll_aclk_1	输出	1		Axi4 Port1 的时钟
pll_aclk_2	输出	1		Axi4 Port2 的时钟
pll_pclk	输出	1		APB Port 的时钟

端口名	输入/输出	位宽	有效值	描述
csysreq_ddrc	输入	1	低电平	DDRC 低功耗请求输入
csysack_ddrc	输出	1	低电平	DDRC 低功耗响应
cactive_ddrc	输出	1	高电平	DDRC 激活标志

(三) AXI4 接口

HMIC_H IP 可提供三组 AXI4 Host Port，一组 128bit，两组 64bit，关于 AXI4 的接口描述如下：

表 3 AXI4 Port0

端口名	输入/输出	位宽	有效值	描述
areset_0	输入	1	高电平	AXI Port0 复位
aclk_0	输入	1		AXI Port0 输入时钟
awid_0	输入	8		AXI Port0 Write address ID
awaddr_0	输入	32		AXI Port0 Write address
awlen_0	输入	8		AXI Port0 Write Burst length
awsize_0	输入	3		AXI Port0 Write Burst size
awburst_0	输入	2		AXI Port0 Write Burst type
awlock_0	输入	1		AXI Port0 Write Lock type
awvalid_0	输入	1	高电平	AXI Port0 Write address valid
awready_0	输出	1	高电平	AXI Port0 Write address ready
awurgent_0	输入	1	高电平	AXI Port0 Write Urgent, 使能时, 该 Port 的 Write address 指令优先执行
awpoison_0	输入	1	高电平	AXI Port0 Write Poison, 使能时, 该 Port 的 Write address 指令无效
wdata_0	输入	128		AXI Port0 Write data
wstrb_0	输入	16		AXI Port0 Write strobes
wlast_0	输入	1		AXI Port0 Write last
wvalid_0	输入	1	高电平	AXI Port0 Write data valid
wready_0	输出	1	高电平	AXI Port0 Write data ready
bid_0	输出	8		AXI Port0 Write response ID
bresp_0	输出	2		AXI Port0 Write response
bvalid_0	输出	1	高电平	AXI Port0 Write response valid
bready_0	输入	1	高电平	AXI Port0 Write response ready
arid_0	输入	8		AXI Port0 Read address ID
araddr_0	输入	32		AXI Port0 Read address
arlen_0	输入	8		AXI Port0 Read Burst length
arsize_0	输入	3		AXI Port0 Read Burst size
arburst_0	输入	2		AXI Port0 Read Burst type
arlock_0	输入	1		AXI Port0 Read Lock type
arvalid_0	输入	1	高电平	AXI Port0 Read address valid

端口名	输入/输出	位宽	有效值	描述
aready_0	输出	1	高电平	AXI Port0 Read address ready
arurgent_0	输入	1	高电平	AXI Port0 Read Urgent, 使能时, 该 Port 的 Read address 指令优先执行
arpoison_0	输入	1	高电平	AXI Port0 Read Poison, 使能时, 该 Port 的 Read address 指令无效
rid_0	输出	8		AXI Port0 Read data ID
rdata_0	输出	128		AXI Port0 Read data
rresp_0	输出	2		AXI Port0 Read response
rlast_0	输出	1		AXI Port0 Read last
rvalid_0	输出	1	高电平	AXI Port0 Read data valid
rready_0	输入	1	高电平	AXI Port0 Read data ready
csysreq_0	输入	1	低电平	AXI Port0 进入低功耗请求
csysack_0	输出	1	低电平	AXI Port0 进入低功耗响应
cactive_0	输出	1	高电平	AXI Port0 active

表 4 AXI4 Port1

端口名	输入/输出	位宽	有效值	描述
areset_1	输入	1	高电平	AXI Port1 复位
aclk_1	输入	1		AXI Port1 输入时钟
awid_1	输入	8		AXI Port1 Write address ID
awaddr_1	输入	32		AXI Port1 Write address
awlen_1	输入	8		AXI Port1 Write Burst length
awsize_1	输入	3		AXI Port1 Write Burst size
awburst_1	输入	2		AXI Port1 Write Burst type
awlock_1	输入	1		AXI Port1 Write Lock type
awvalid_1	输入	1	高电平	AXI Port1 Write address valid
awready_1	输出	1	高电平	AXI Port1 Write address ready
awurgent_1	输入	1	高电平	AXI Port1 Write Urgent, 使能时, 该 Port 的 Write address 指令优先执行
awpoison_1	输入	1	高电平	AXI Port1 Write Poison, 使能时, 该 Port 的 Write address 指令无效
wdata_1	输入	64		AXI Port1 Write data
wstrb_1	输入	8		AXI Port1 Write strobes
wlast_1	输入	1		AXI Port1 Write last
wvalid_1	输入	1	高电平	AXI Port1 Write data valid
wready_1	输出	1	高电平	AXI Port1 Write data ready
bid_1	输出	8		AXI Port1 Write response ID
bresp_1	输出	2		AXI Port1 Write response
bvalid_1	输出	1	高电平	AXI Port1 Write response valid
bready_1	输入	1	高电平	AXI Port1 Write response ready
arid_1	输入	8		AXI Port1 Read address ID

端口名	输入/输出	位宽	有效值	描述
araddr_1	输入	32		AXI Port1 Read address
arlen_1	输入	8		AXI Port1 Read Burst length
arsize_1	输入	3		AXI Port1 Read Burst size
arburst_1	输入	2		AXI Port1 Read Burst type
arlock_1	输入	1		AXI Port1 Read Lock type
arvalid_1	输入	1	高电平	AXI Port1 Read address valid
arready_1	输出	1	高电平	AXI Port1 Read address ready
arurgent_1	输入	1	高电平	AXI Port1 Read Urgent, 使能时, 该 Port 的 Read address 指令优先执行
arpoison_1	输入	1	高电平	AXI Port1 Read Poison, 使能时, 该 Port 的 Read address 指令无效
rid_1	输出	8		AXI Port1 Read data ID
rdata_1	输出	64		AXI Port1 Read data
rresp_1	输出	2		AXI Port1 Read response
rlast_1	输出	1		AXI Port1 Read last
rvalid_1	输出	1	高电平	AXI Port1 Read data valid
rready_1	输入	1	高电平	AXI Port1 Read data ready
csysreq_1	输入	1	低电平	AXI Port1 进入低功耗请求
csysack_1	输出	1	低电平	AXI Port1 进入低功耗响应
cactive_1	输出	1	高电平	AXI Port1 active

表 5 AXI4 Port2

端口名	输入/输出	位宽	有效值	描述
areset_2	输入	1	高电平	AXI Port2 复位
ack_2	输入	1		AXI Port2 输入时钟
awid_2	输入	8		AXI Port2 Write address ID
awaddr_2	输入	32		AXI Port2 Write address
awlen_2	输入	8		AXI Port2 Write Burst length
awsize_2	输入	3		AXI Port2 Write Burst size
awburst_2	输入	2		AXI Port2 Write Burst type
awlock_2	输入	1		AXI Port2 Write Lock type
awvalid_2	输入	1	高电平	AXI Port2 Write address valid
awready_2	输出	1	高电平	AXI Port2 Write address ready
awurgent_2	输入	1	高电平	AXI Port2 Write Urgent, 使能时, 该 Port 的 Write address 指令优先执行
awpoison_2	输入	1	高电平	AXI Port2 Write Poison, 使能时, 该 Port 的 Write address 指令无效
wdata_2	输入	64		AXI Port2 Write data
wstrb_2	输入	8		AXI Port2 Write strobes
wlast_2	输入	1		AXI Port2 Write last
wvalid_2	输入	1	高电平	AXI Port2 Write data valid
wready_2	输出	1	高电平	AXI Port2 Write data ready

端口名	输入/输出	位宽	有效值	描述
bid_2	输出	8		AXI Port2 Write response ID
bresp_2	输出	2		AXI Port2 Write response
bvalid_2	输出	1	高电平	AXI Port2 Write response valid
bready_2	输入	1	高电平	AXI Port2 Write response ready
arid_2	输入	8		AXI Port2 Read address ID
araddr_2	输入	32		AXI Port2 Read address
arlen_2	输入	8		AXI Port2 Read Burst length
arsize_2	输入	3		AXI Port2 Read Burst size
arburst_2	输入	2		AXI Port2 Read Burst type
arlock_2	输入	1		AXI Port2 Read Lock type
arvalid_2	输入	1	高电平	AXI Port2 Read address valid
arready_2	输出	1	高电平	AXI Port2 Read address ready
arurgent_2	输入	1	高电平	AXI Port2 Read Urgent, 使能时, 该 Port 的 Read address 指令优先执行
arpoison_2	输入	1	高电平	AXI Port2 Read Poison, 使能时, 该 Port 的 Read address 指令无效
rid_2	输出	8		AXI Port2 Read data ID
rdata_2	输出	64		AXI Port2 Read data
rresp_2	输出	2		AXI Port2 Read response
rlast_2	输出	1		AXI Port2 Read last
rvalid_2	输出	1	高电平	AXI Port2 Read data valid
rready_2	输入	1	高电平	AXI Port2 Read data ready
csysreq_2	输入	1	低电平	AXI Port2 进入低功耗请求
csysack_2	输出	1	低电平	AXI Port2 进入低功耗响应
cactive_2	输出	1	高电平	AXI Port2 active

(四) APB 接口

表 6 APB 接口

端口名	输入/输出	位宽	有效值	描述
preset	输出	1	高电平	APB 复位
pclk	输出	1		APB 时钟
pwdata	输入	32		APB 写数据
pwrite	输入	1	高电平	APB 读写方向
penable	输入	1	高电平	APB 使能.
psel	输入	1	高电平	APB Peripheral Select.
paddr	输入	12		APB 地址.
prdata	输出	32		APB 读数据
pready	输出	1	高电平	APB Ready.

四 模块例化

(一) 模块配置

1. 选择 IP

打开 IPC 软件，进入 IP 选择界面，如下图所示，选取 System/DDR/Hard 目录下的 Logos HMIC_H，然后在右侧页面设置 Instance Name 名称，并选择 FPGA 的器件类型。

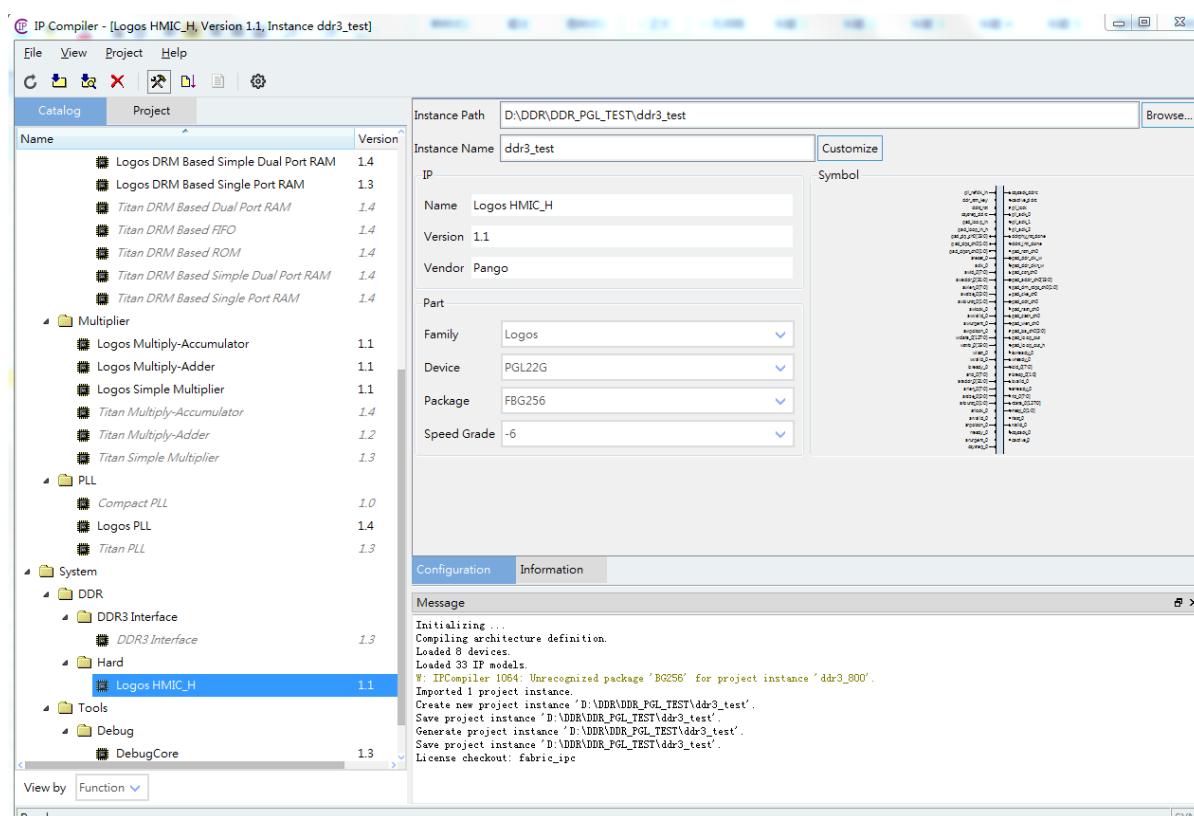


图 2 选择 Logos HMIC_H IP 界面

2. IP 参数设置

IP 选择完成后点击 Customize 进入 Logos HMIC_H IP 参数设置界面，如下图所示，左边 Symbol 为接口框图，右边为参数配置窗口：

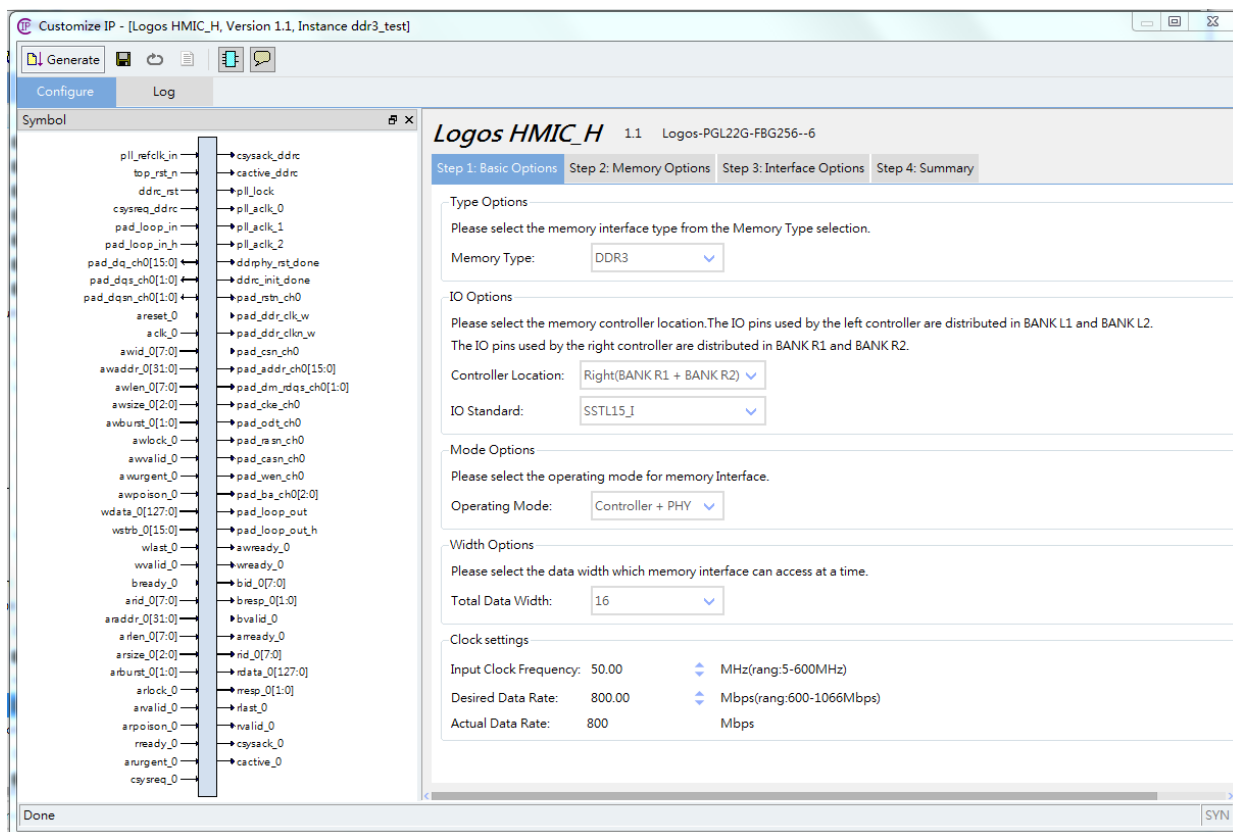
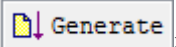


图 3 配置 Logos HMIC_H IP 参数界面

3. 生成 IP

参数配置完成后点击左上角的  按钮，生成 IP，即可生成相应于用户特定设置的 HMIC_H IP 代码。生成 IP 的信息报告界面如下图所示：

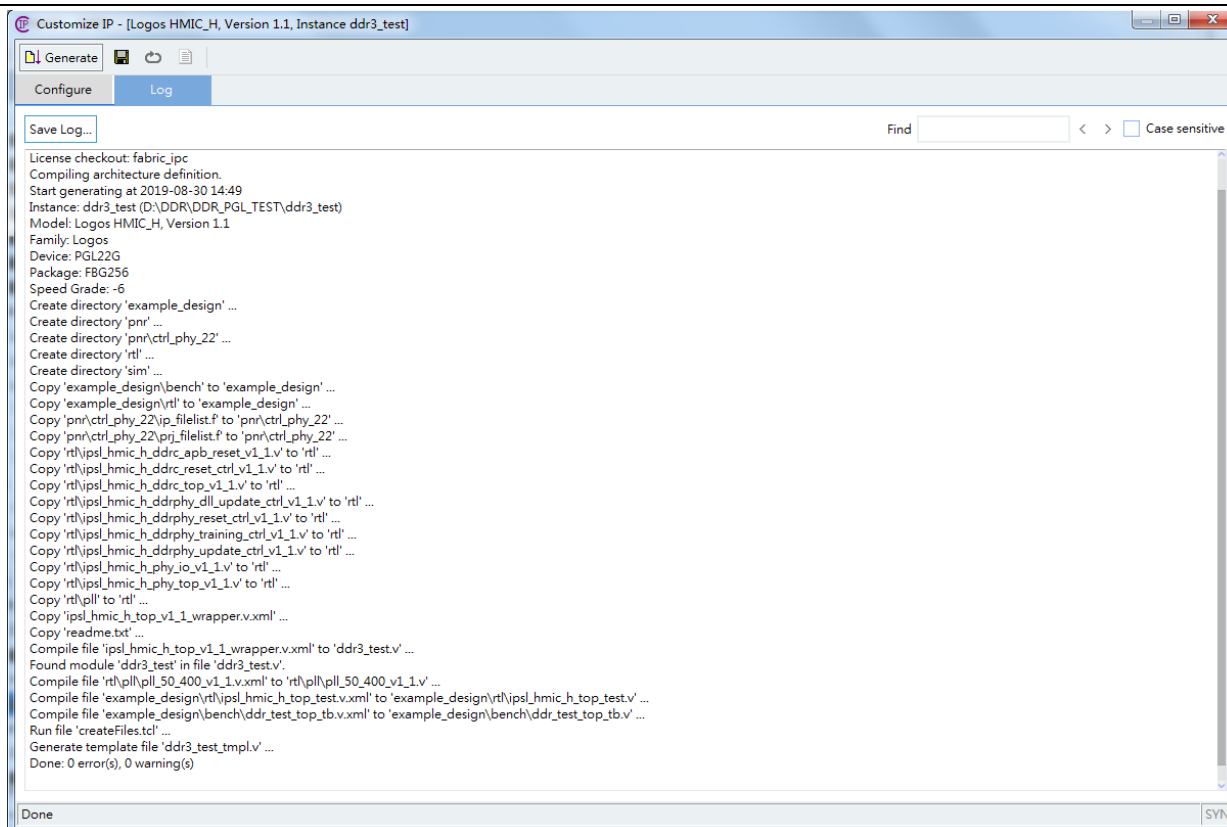


图 4 Logos HMIC_H IP 生成报告界面

注：IP 自带生成的.pds 文件和.fdc 文件仅供参考，需要根据实际单板进行修改。

成功生成 IP 后会在图 2 中指定的 project 路径下输出如下文件：

- <instance_name>.v -所生成 IP 的顶层.v 文件。
- <instance_name>.vm -所生成 IP 的综合后输出.vm 文件(如果关闭了综合器,则没有该文件)。
- <instance_name>_tmpl.v 文件-所生成文件的例化举例.v 文件，不可以参与编译。
- Inst.idf-所生成 IP 的配置文件。
- <project_path>/rtl/-所生成 IP 的 RTL 代码文件。
- <project_path>/example_design/-所生成 IP 的 example_design 所使用的 test bench 以及相应 memory simulation model 文件
- <project_path>/pnr/-所生成 IP 的 example_design 相应的工程文件.pds 与管脚约束文件.fdc。

注意：IP 生成的管脚约束文件.fdc 仅供参考，使用时请根据实际的管脚连接更改管脚约束。

- <project_path>/sim/-所生成 IP 的仿真目录。ctrl_phy_sim.tcl 是仿真脚本，sim_file_list.f 是仿

真文件 list。

4. 配置说明

HMIC_H IP 配置分为四个页面,分别为 Step1: Basic Options, Step2: Memory Options, Step3: Interface Options, Step4: Summary, 请务必按照该页面顺序配置。

Step 1: Basic Options, 是 IP 的基本配置页面, 页面如下图所示:

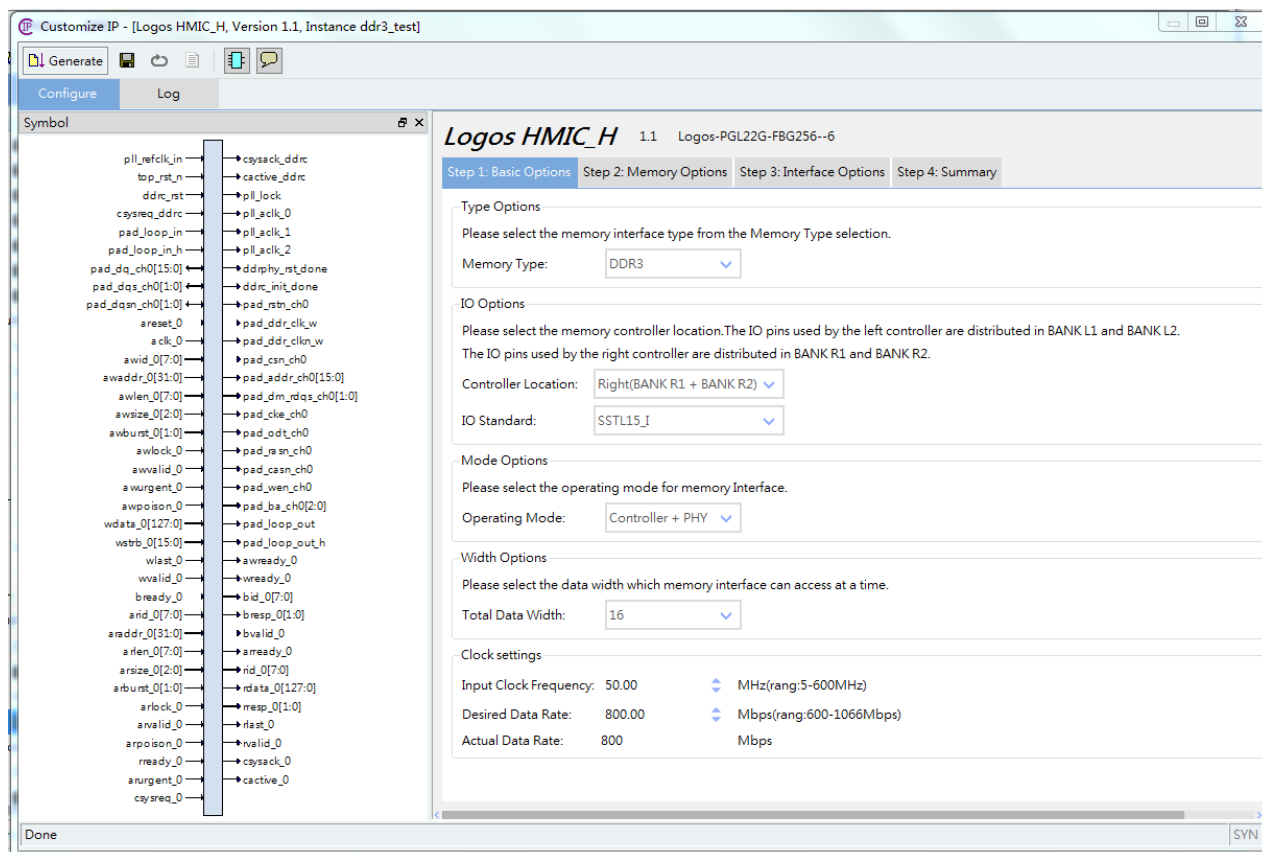


图 5 Basic Options 页面

Basic Options 详细参数见下表:

表 7 Basic Options 参数说明

选项名/参数名	说明
Memory Type	使用的SDRAM类型, 目前支持的类型为 1) DDR3; 2) DDR2; 3) LPDDR
Controller Location	Controller在FPGA芯片的位置, 目前PGL22支持的位置为 1) Right (BANK R1 + BANK R2); 2) Left (BANK L1 + BANK L2); 注: HMIC_H为硬核, 位置固定不可变动。

选项名/参数名	说明
IO Standard	接口标准选项，DDR3支持的接口标准为 1) SSTL15_I 2) SSTL15_II DDR2和LPDDR支持的接口标准为 1) SSTL18_I 2) SSTL18_II
Operating Mode	HMIC_H运行模式选择。目前只支持Controller + PHY模式
Total Data Width	与HMIC_H连接的片外SDRAM总共的DQ宽度。目前支持的总宽度为 1) 16 2) 8
Input Clock Frequency	HMIC_H的输入时钟，单位MHz
Desired Data Rate	期望的数据速率，DDR3支持的最高速率为1066Mbps，DDR2支持的最高速率为800Mbps，LPDDR支持的最高速率为400Mbps。
Actual Data Rate	实际能达到的数据速率，尽可能接近期望的速率。

Step 2: Memory Options，是 Memory 参数的配置页面，页面如下图所示：

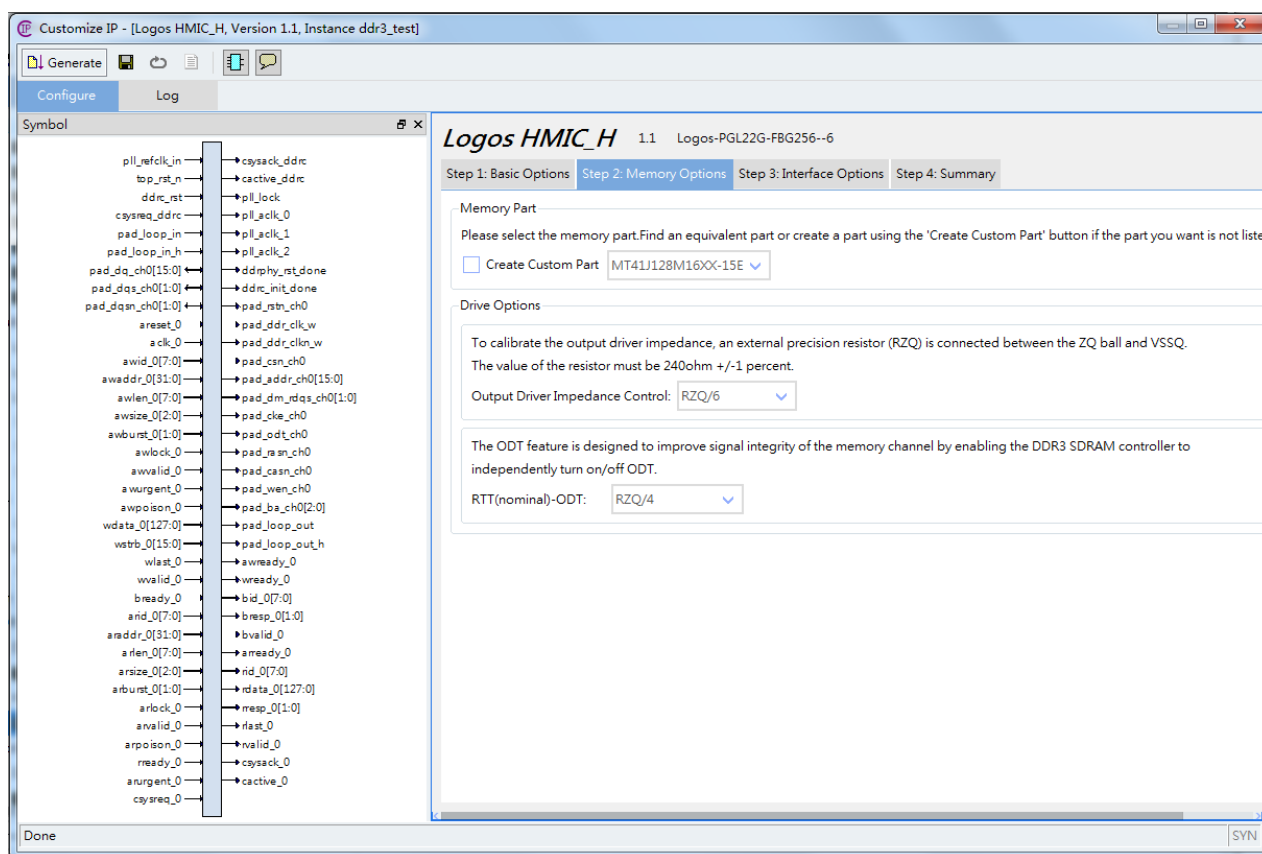


图 6 Memory Options 页面

Memory Options 详细参数见下表：

表 8 Memory Options 参数说明

选项名/参数名	说明
Memory Part	<p>SDRAM的具体型号，DDR3支持的型号为</p> <ol style="list-style-type: none"> 1) MT41J128M8XX-15E 2) MT41J64M16XX-15E 3) MT41J256M8XX-15E 4) MT41J128M16XX-15E 5) MT41J512M8XX-15E 6) MT41J256M16XX-15E <p>DDR2支持的型号为</p> <ol style="list-style-type: none"> 1) MT47H128M8XX-25E 2) MT47H64M16XX-25E 3) MT47H128M16XX-25E 4) MT47H256M4XX-25E 5) MT47H64M8XX-25E-IT <p>LPDDR支持的型号为</p> <ol style="list-style-type: none"> 1) MT46H128M16XXXX-5L-IT 2) MT46H64M16XXXX-5L-IT 3) MT46H64M16XXXX-6L-IT 4) MT46H32M16XXXX-5 5) MT46H16M16XXXX-6-IT 6) MT46H16M16XXXX-75-IT <p>若这些均不是所需要的器件型号，则可以勾选Create Custom Part选项，然后在Custom Memory Part选项框中定制新的SDRAM类型。</p>
Custom Memory Part	<p>该选项框在勾选Create Custom Part选项时显示，用于定制新的SDRAM类型。包含选项：Base Part、Timing Parameters、Row Address、Column Address、Bank Address。</p> <p>Base Part：定制SDRAM的参考型号。</p> <p>Timing Parameters：定制SDRAM的Timing参数。</p> <p>Row Address：行地址。</p> <p>Column Address：列地址。</p> <p>Bank Address：Bank地址</p>
Drive Options	<p>驱动能力选项，DDR3支持的驱动选项为（详细说明见DDR3协议）</p> <ol style="list-style-type: none"> 1) Output Driver Impedance Control 2) RTT(nominal)-ODT <p>DDR2支持的驱动选项为（详细说明见DDR2协议）</p> <ol style="list-style-type: none"> 1) Output Driver Strength 2) RTT(nominal)-ODT <p>LPDDR支持的驱动选项为（详细说明见LPDDR协议）</p> <ol style="list-style-type: none"> 1) Driver Strength

Step 3: Interface Options，是接口参数的配置页面，页面如下图所示：

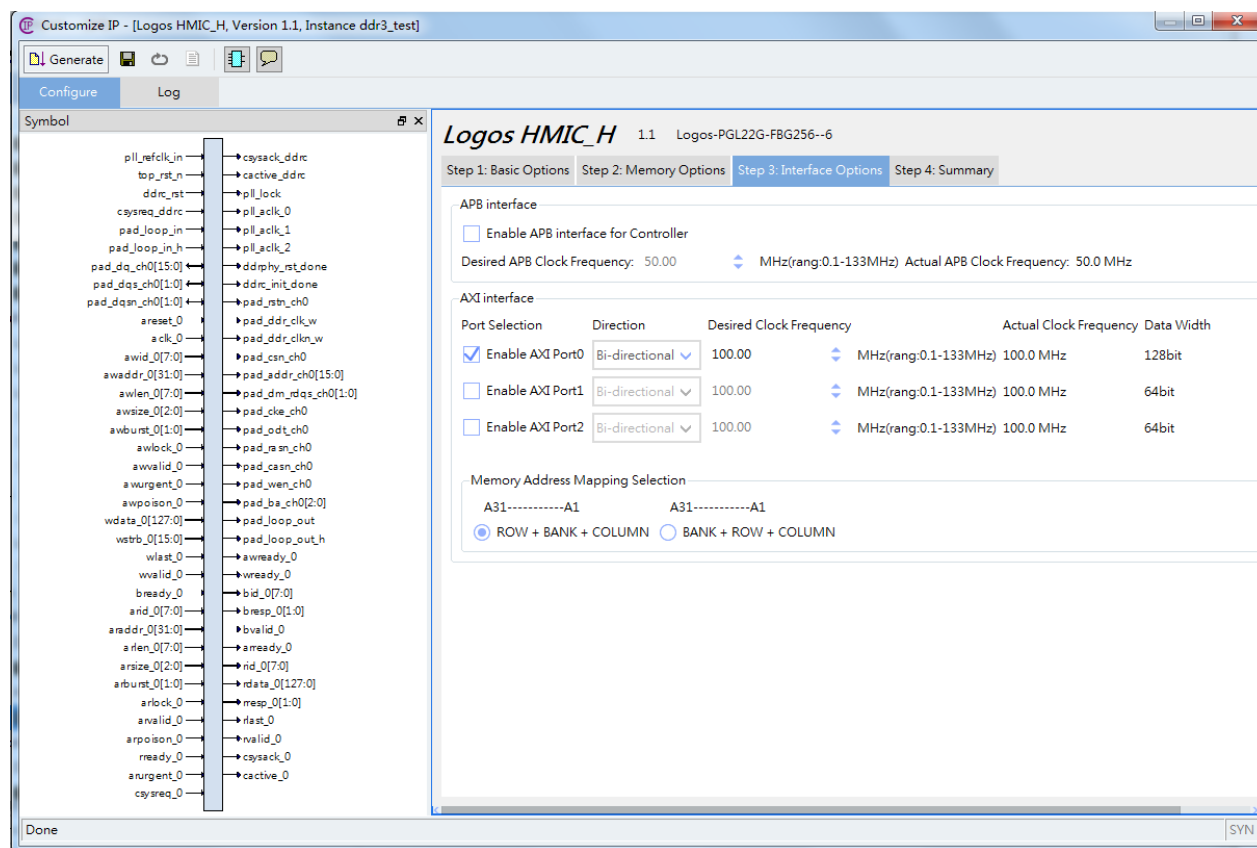


图 7 Interface Options 页面

Interface Options 详细参数见下表:

表 9 Interface Options 参数说明

选项名/参数名	说明
APB interface	APB接口配置，包括使能和APB接口的时钟频率。
AXI interface	三组AXI4接口的配置，包括使能、读写方向、时钟频率。 读写方向配置如下 1) Bi-directional 可读可写 2) Read 只读 3) Write 只写
Memory Address Mapping Selection	SDRAM的地址与AXI4地址映射选项 1) ROW + BANK + COLUMN 2) BANK + ROW + COLUMN AXI4的读写地址为32位，但是只有A1-A31有效，A0为无效位。例如，选择ROW+BANK+COLUMN时，SDRAM列地址c0对应AXI地址A1，c1对应A2，其他地址以此类推。

Step 4: Summary 页面，用于打印当前的配置信息，不需要配置参数，页面如下图所示:

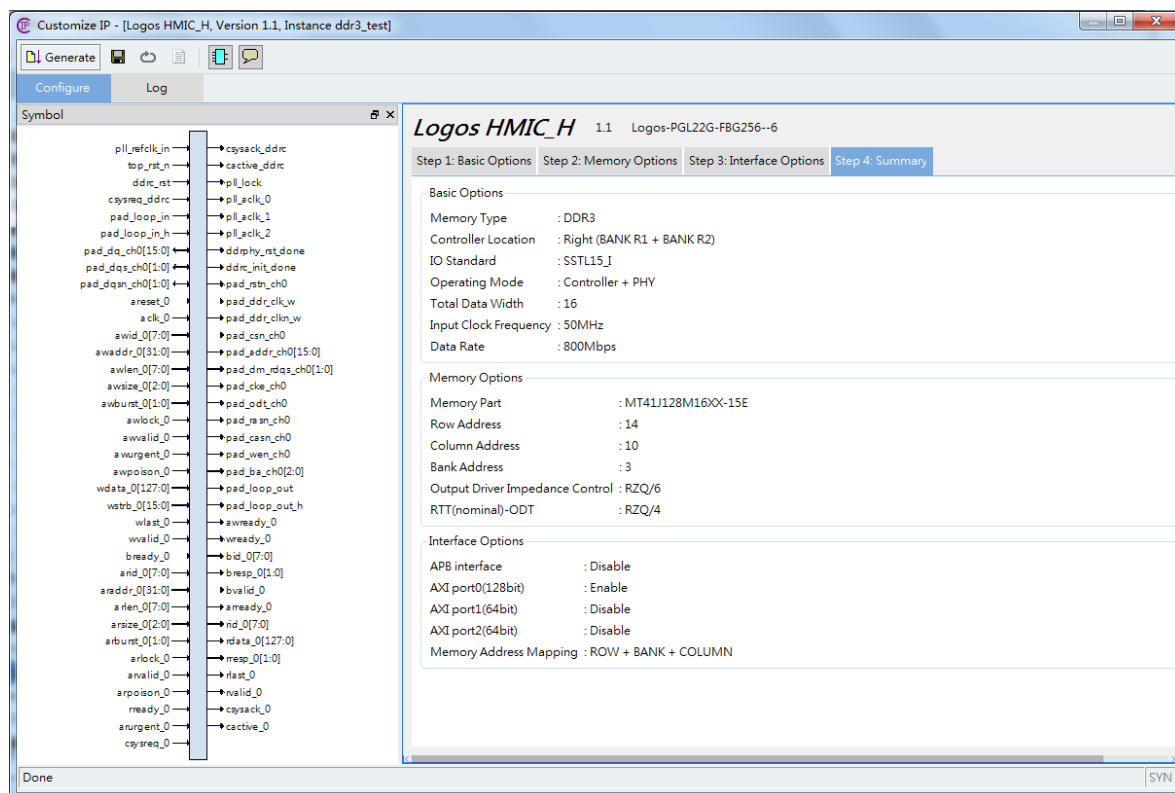


图 8 Summary 页面

(二) 运行与仿真

HMIC_H IP 的仿真是基于 example design 的 test bench 进行的。有关 example design 的详细信息请参考第六章。

仿真运行步骤：打开 cmd.exe，在命令行中将当前目录切换到 IP 生成目录下的 sim，运行 vsim，打开 ModelSim 仿真软件，如下图所示：

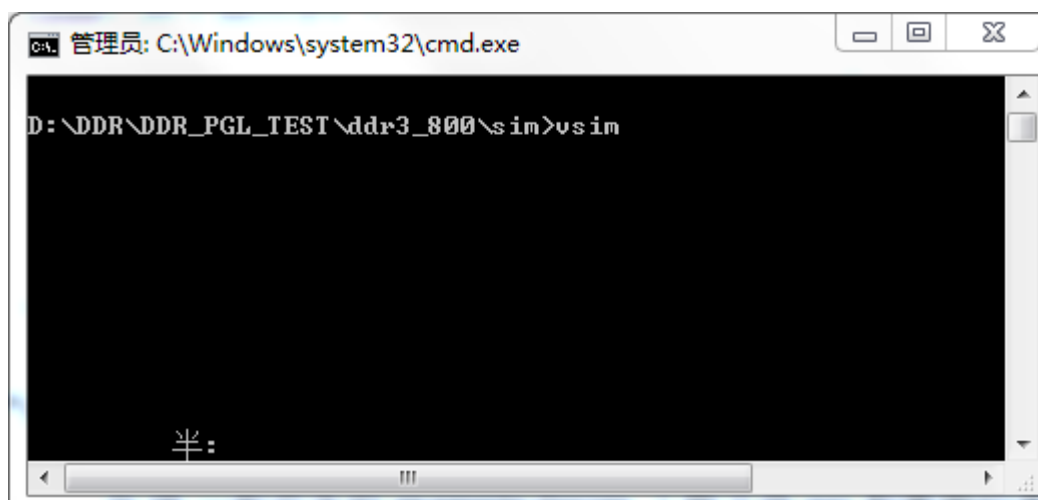


图 9 打开 ModelSim 指令

在 ModelSim 仿真软件中运行 ctrl_phy_sim.tcl 脚本即可进行仿真，如下图所示：

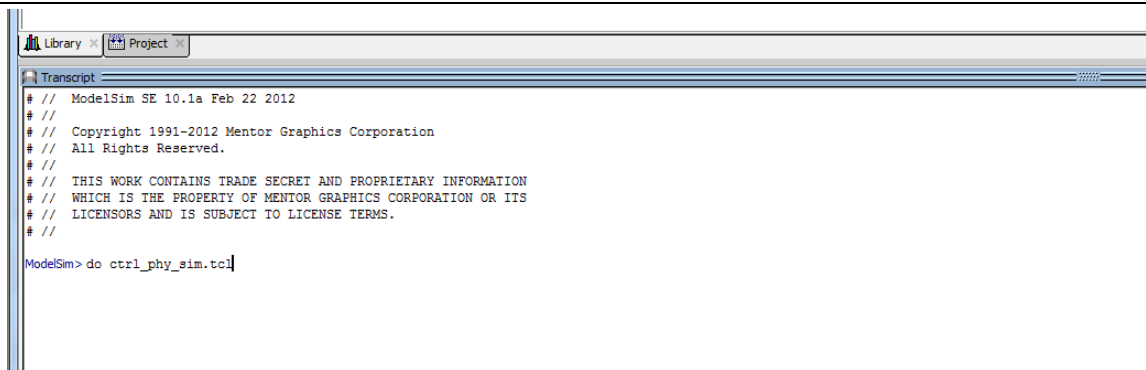


图 10 ModelSim 执行仿真脚本

(三) 综合与布局布线

与 IP 一起产生的 example design 工程文件.prj 和管脚约束文件.fdc 存放在/pnr/目录下。

目前只有在使用 HMIC_H IP 验证板才可以直接导入并使用这些文件进行综合与布局布线，使用其他板调试，则需要修改物理约束，具体请参考第八章。

五 接口时序描述

(一) AXI4 接口

本设计 AXI4 接口为标准的 AXI4 协议接口，接口时序可参考 AXI4 协议。

1. AXI4 读时序

以 AXI4 Port0 为例，AXI4 接口单次读操作的时序如图 11 所示，主设备发送地址，一个周期后从设备接收。主设备在发送地址的同时也发送了一些控制信息标志了 Burst 的程度和类型，为了保持图的清晰性，在此省略这些信号。地址总线上出现地址之后，在读数据通道上发生数据的传输。从设备一直保持 rvalid_0 信号为低，直到读数据准备好。从设备发送 rlast_0 信号标志此次读操作中最后一个数据的传输。

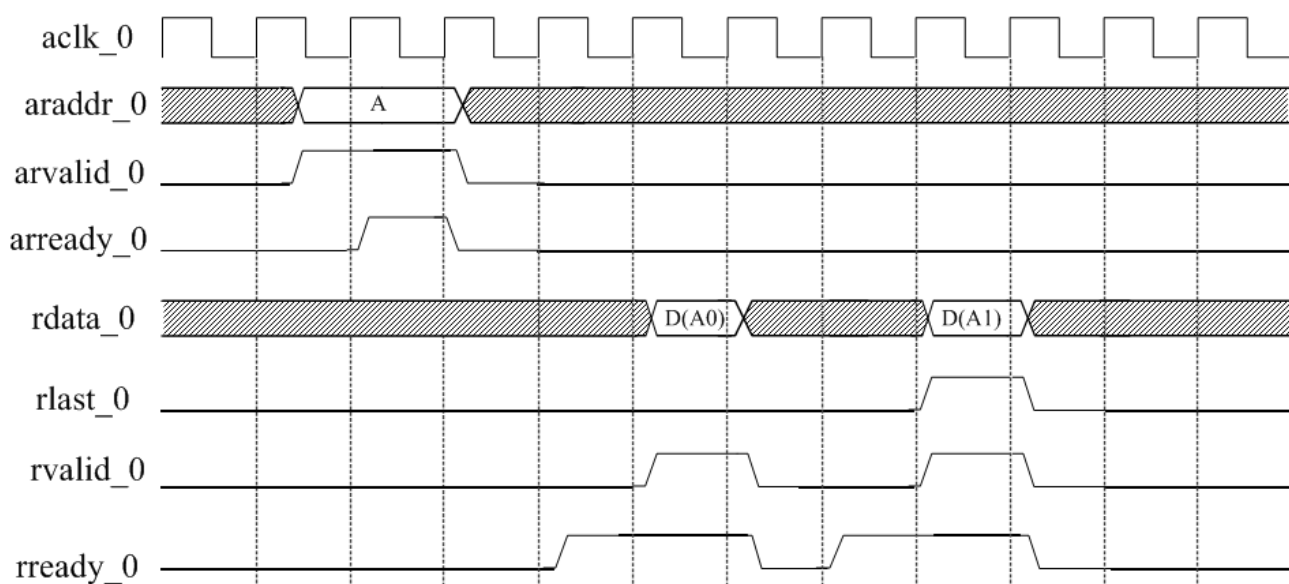


图 11 AXI4 单次读时序

以 AXI4 Port0 为例，AXI4 接口连续读操作的时序如图 12 所示，主设备在从设备接收第一次读操作的地址后发送下一次读操作的地址。这样可以保证一个从设备在完成第一次读操作的同时可以开始处理第二次读操作的数据。

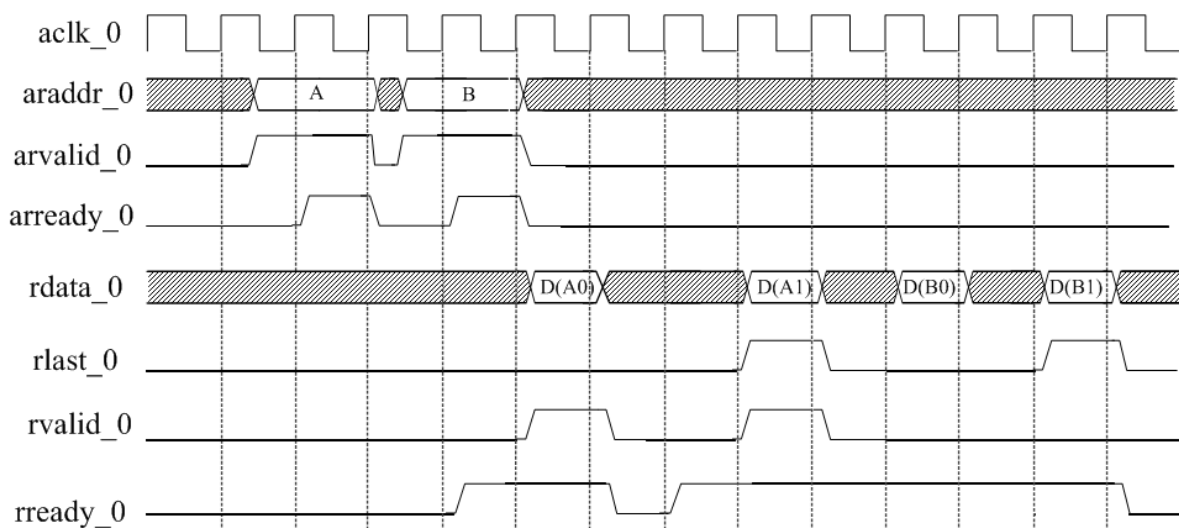


图 12 AXI4 连续读时序

2. AXI 写时序

以 AXI4 Port0 为例，AXI4 接口单次写操作的时序如图 13 所示，当主设备发送地址和控制信息到写地址通道之后，写操作开始。然后主设备通过写数据通道发送每一个写数据，当为最后一个需要发送的数据时，主设备将 `wlast_0` 信号置高。当从设备接收完所有的数据时，从设备返回给主设备一个写响应标志本次写操作的结束。连续写操作与连续读操作类似，即主设备在从设备接收第一次写操作的地址后发送下一次写操作的地址。

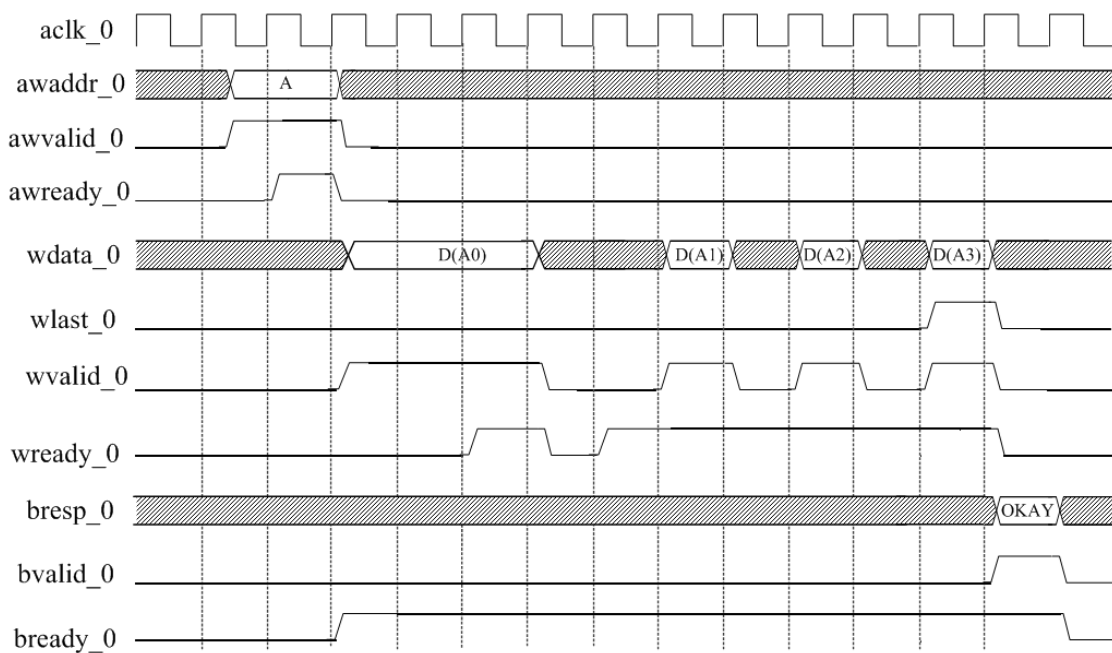


图 13 AXI4 单次写时序

(二) APB 接口

APB 接口是半双工通信，读写数据线独立，控制线和地址线复用。每次握手至少需要消耗 2 个 pclk 的周期。APB 接口包含的信号见 APB 接口说明。

1. APB 接口写时序

第一个时钟周期：psel, pwrite 拉高，paddr, pwidth 给初值，它们需维持稳定直到握手完成后释放。

第二个时钟周期：penable 拉高直到握手完成后释放。

APB 接口写操作的典型时序如下图所示：

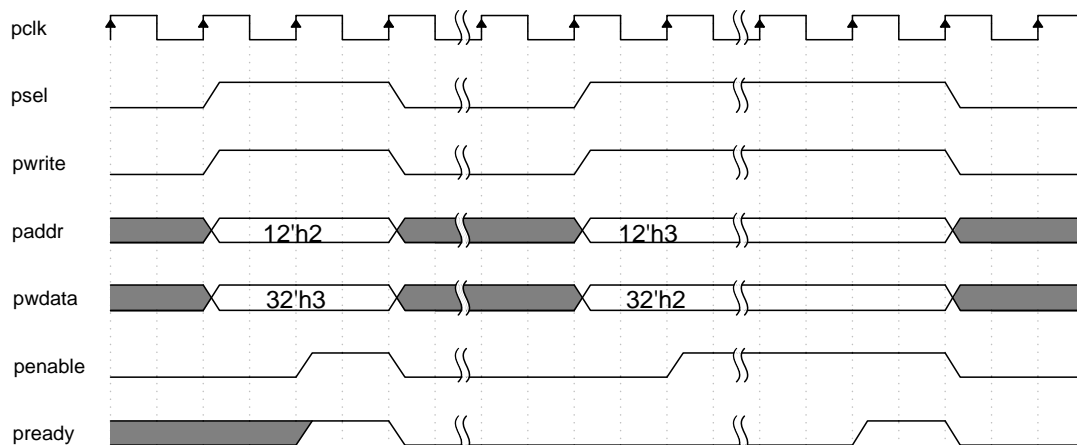


图 14 APB 接口 write 数据

2. APB 接口读时序

第一个时钟周期：psel 拉高，pwrite 拉低，paddr 给初值，它们需维持稳定直到握手完成后释放。

第二个时钟周期：penable 拉高直到握手完成后释放。

有效数据：prdata 只在握手时有效。

APB 接口读操作的典型时序如下图所示：

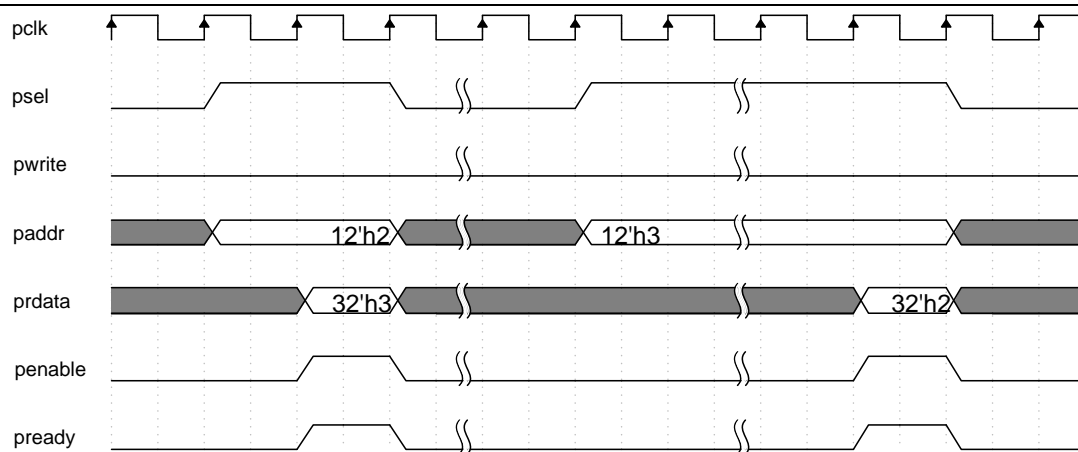


图 15 APB 接口 read 数据

六 Example Design

本节主要介绍基于 HMIC_H IP 的 Example 设计方案。本应用方案用户逻辑作为 AXI Master，HMIC_H IP 作为 AXI Slave，用户逻辑通过 AXI 接口的 Write 通道写入数据，在 AXI 接口的 Read 通道接收数据并进行数据比对，若数据错误，将点亮 Error LED。

(一) 设计框图

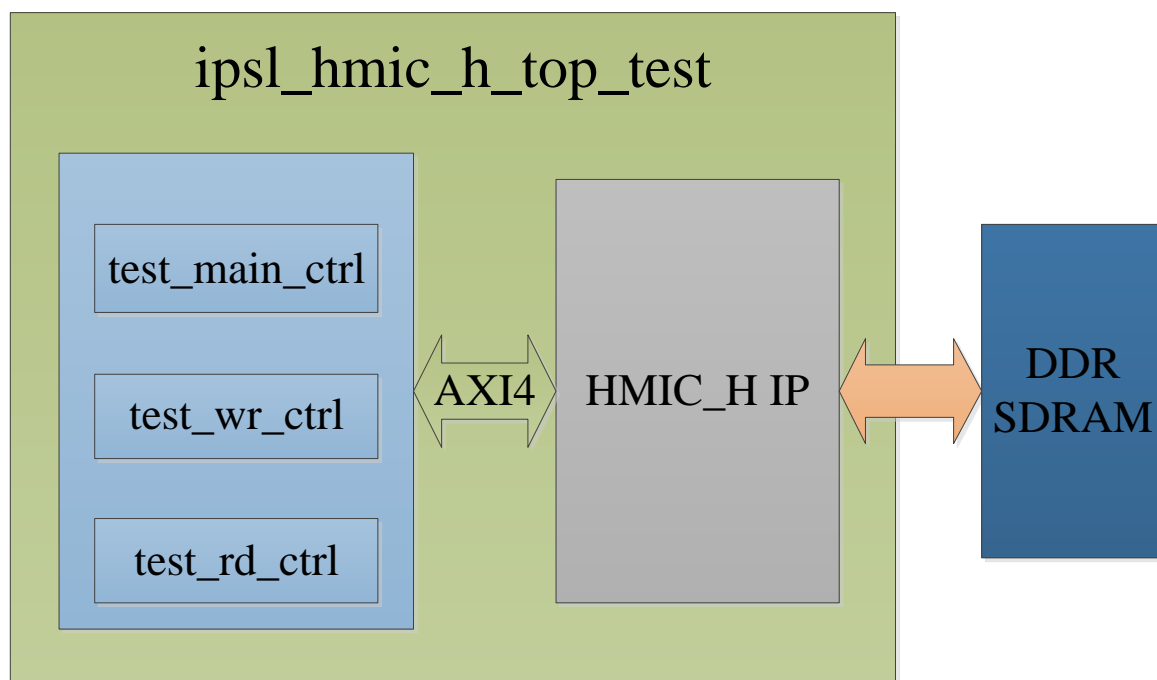


图 16 Example 设计系统框图

上图为 Example 设计的系统框图，其中 test_main_ctrl 模块为 AXI 读写指令的控制模块，test_wr_ctrl 模块为 AXI 写指令和写数据的控制模块，test_rd_ctrl 模块为 AXI 读指令和读数据的控制模块。以上三个模块都有 64bit 和 128bit 两种位宽，当使用 AXI Port0(128bit)时，调用 128bit 的模块，使用 AXI Port1(64bit)或者 AXI Port2(64bit)时，调用 64bit 的模块。

(二) 测试方法

在 Example 设计中，用户逻辑对 HMIC_H IP 进行读写，并对回读数据校验，详细的测试流程如下图所示：

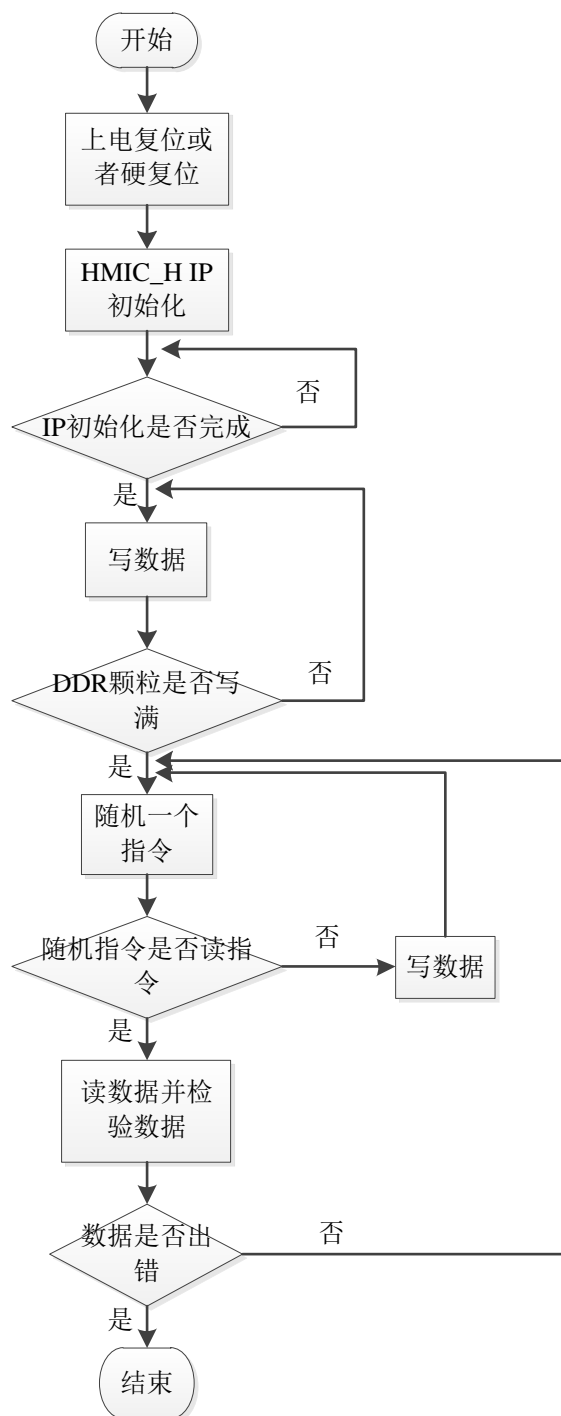


图 17 测试流程图

系统上电或者硬复位启动后，HMIC_H IP 开始执行初始化，待初始化完成（ddrc_init_done 拉起），test_main_ctrl 模块控制 test_wr_ctrl 模块产生写指令和写数据对 DDR 颗粒的数据初始化，写满后，test_main_ctrl 开始进行随机读写，test_rd_ctrl 对回读的数据进行检验，判断数据是否出错。

七 配置寄存器

HMIC_H IP 可通过 APB 接口对内部 DDRC 配置寄存器进行读写，在初始化阶段，IP 将配置 DDRC 内部的配置寄存器，如果用户需要读写 DDRC 内部寄存器，需要在初始化完成后进行操作。由于 IP 初始化阶段已将 DDRC 内部寄存器进行了正确的配置，因此不建议用户在初始化完成后随意更改配置寄存器的值。

(一) MSTR: DDRC 工作模式配置寄存器

位宽：32bit，访问地址：0x00。

表 10 MSTR 各 bit 定义

Bits	名称	复位值	访问类型	描述
31:20				保留
19:16	burst_rdw	0x4	R/W	SDRAM burst length used: 0001: Burst length of 2 0010: Burst length of 4 0100: Burst length of 8 1000: Burst length of 16 IP 只支持 Burst length 为 8。
15	dll_off_mode	0x0	R/W	DRAM 的 DLL 模式: 0: DLL 打开 1: DLL 关闭
14				保留
13:12	data_bus_width	0x0	R/W	数据的总线的模式: 00: Full DQ bus width 01: Half DQ bus width 10: Quarter DQ bus width
11:9				保留
8	burst_mode	0x0	R/W	burst_mode 0: Sequential burst mode 1: Interleaved burst mode
7:6				保留
5	lpddr4	0x0	R/W	是否选择 lpddr4 0: 否 1: 是
4	ddr4	0x0	R/W	是否选择 ddr4 0: 否 1: 是

Bits	名称	复位值	访问类型	描述
3	lpddr3	0x0	R/W	是否选择 lpddr3 0: 否 1: 是
2	lpddr2	0x0	R/W	是否选择 lpddr2 0: 否 1: 是
1	mobile ddr	0x0	R/W	是否选择 mobile ddr 0: 否 1: 是
0	ddr3	0x1	R/W	是否选择 ddr3 0: 否 1: 是

注：R/W 代表可读可写，R 代表只读。

(二) STAT: DDRC 工作状态寄存器

位宽：32bit，访问地址：0x00。

表 11 STAT 各 bit 定义

Bits	名称	复位值	访问类型	描述
31:10				保留
9:8	selfref_state	0x0	R	Self refresh state. 00: SDRAM is not in Self Refresh 01: Self refresh 1 10: Self refresh power down 11: Self refresh 2
7:6				保留
5:4	selfref_type	0x0	R	标识 SDRAM 是否进入 Self refresh 或者 SR-Powerdown 00: 否 11: 是
3:2				保留
1:0	operating_mode	0x0	R	DDRC 运行状态 00: init 01: normal 10: power-down 11: self refrash

注：R/W 代表可读可写，R 代表只读。

八 注意事项

(一) 时序约束

IP 有 5 个时钟，分别为 pll_refclk_in、phy_clk、pll_aclk0、pll_aclk1、pll_aclk2、pll_pclk，其中 pll_refclk_in 是输入时钟，phy_clk、pll_aclk0、pll_aclk1、pll_aclk2、pll_pclk 都是 PLL 倍频得到，phy_clk 用作 HMIC_H 硬核的输入时钟，pll_aclk0 用做 AXI4 port0 的输入时钟，pll_aclk1 用做 AXI4 port1 的输入时钟，pll_aclk2 用做 AXI4 port2 的输入时钟，pll_pclk 用做 APB port 的输入时钟。phy_clk 是 HMIC_H 专用时钟，在 IP 内部使用，不允许外接使用。pll_pclk，pll_aclk_0，pll_aclk_1，pll_aclk_2 四路时钟供外部逻辑使用，彼此没有相位关系，都是异步时钟。各时钟间的关系如下图所示：

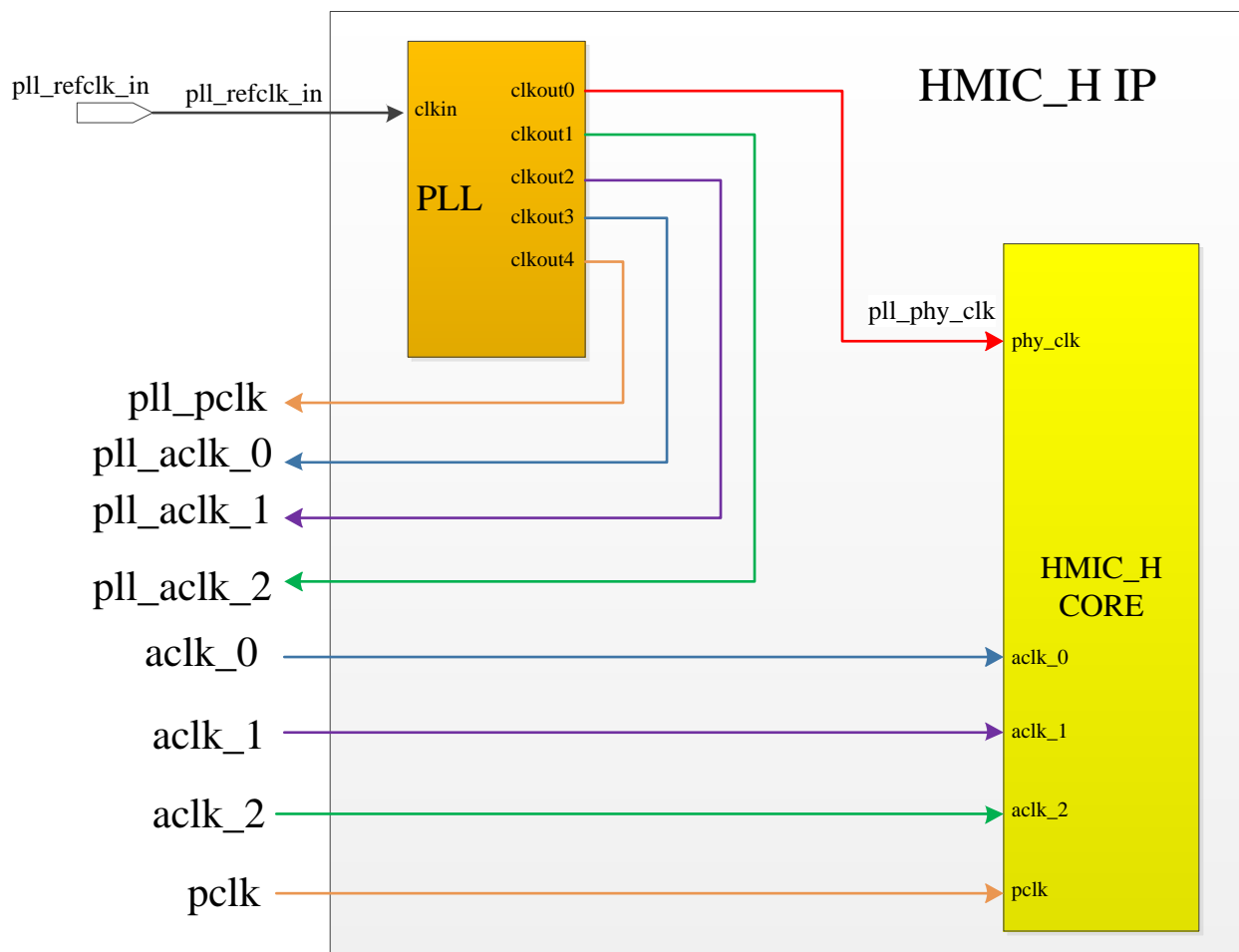


图 18 IP 时钟关系示意图

时钟接口的典型连接如下图所示：

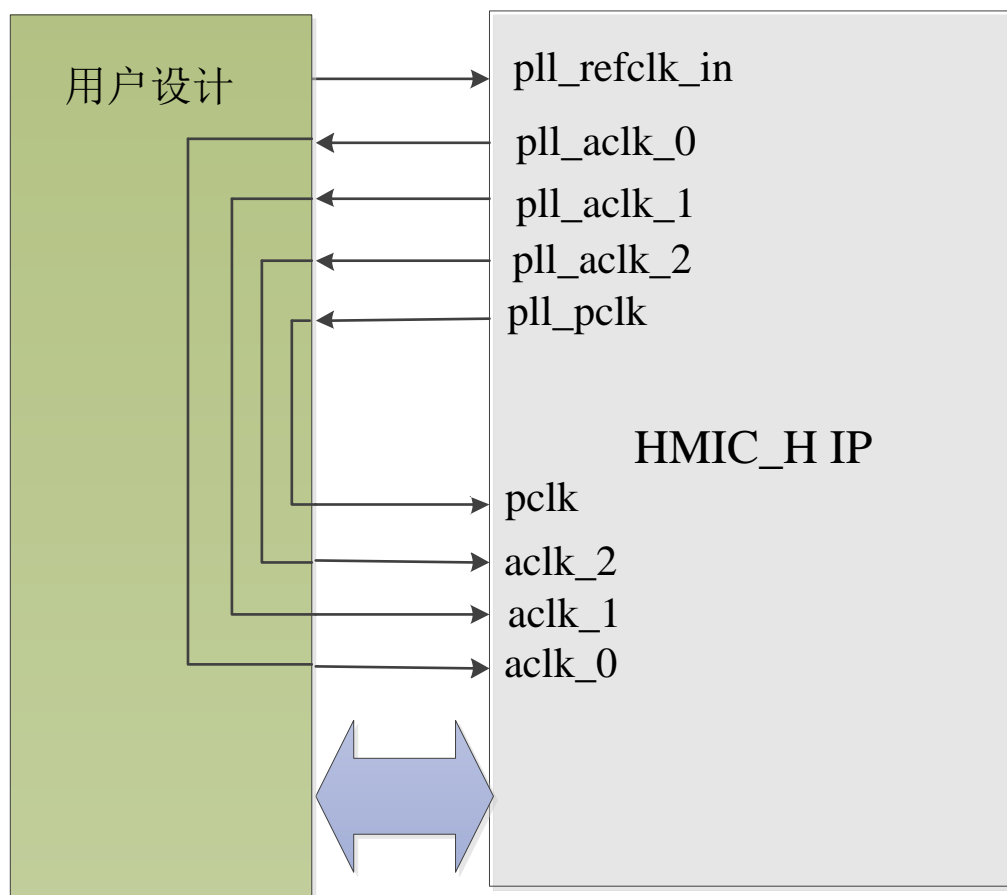


图 19 IP 时钟连接示意图

IP 中的 5 个时钟都需要时序约束，以 Example Design 为例，首先需要约束外部输入时钟源，约束语句如下：

```
create_clock -name {pll_refclk_in} {p:pll_refclk_in} -period {20} -waveform {0 10}
```

外部时钟约束语句说明：create_clock -name {时钟名} {时钟路径} -period {周期，单位 ns} -waveform {高电平时间}。

对于 PLL 的输出时钟，以 Example Design 中的 phy_clk 为例，约束语句如下：

```
create_generated_clock -name {phy_clk} -source {p:pll_refclk_in}
{t:u_ipsl_hmic_h_top.u_pll_50_400.clkout0} -multiply_by 8
```

约束语句说明：create_generated_clock -name {时钟名} -source {时钟的源头} {时钟路径} -multiply_by 与时钟源头的倍数关系。

其他时钟的具体约束方法可参考<project_path>/pnr 目录下的.fdc 文件。

(二) 位置约束

由于 HMC_H IP 为硬核 IP，故，在使用时要注意管脚与硬核的对应关系。具体注意事项如下：

1. 供 HMC_H IP 使用的 PLL 需要布局在指定位置，以 PGL226IFBG256 为例，需将 PLL 的位置固定在 PLL_82_71，约束方法可参考<project_path>/pnr 目录下的.fdc 文件；

2. 在配置 HMC_H IP 对应硬核位置时，如配置为 Right，则所有与 DDR SDRAM 有关的 IO 都需要与<project_path>/pnr/ddr_XXX_right.fdc（XXX 为具体封装，如封装为 FBG256，则 XXX 为 256）中的 IO 约束一致，不能更改；如配置为 Left，则与<project_path>/pnr/ddr_XXX_left.fdc 一致。

3. HMC_H IP 外接 DDR SDRAM 的 IO 固定与 DDR3 x16 颗粒相对应，故，在配置 HMC_H IP 对应 Memory Type 时，如选择 DDR2 和 LPDDR，则需在制作 PCB 时，将 HMC_H IP 多余的 IO 悬空。例如 DDR2 没有 RSTN 管脚，则制作 PCB 时将 HMC_H IP 的 pad_rstn_ch0 悬空。

4. HMC_H IP 的外接 DDR SDRAM 的地址线和数据线是固定位宽(x16)，如果使用的 DDR SDRAM 的地址线或者数据线少于 HMC_H IP 外接的地址线和数据线，将 HMC_H IP 多余的地址线和数据线悬空。例如 HMC_H IP 选择数据位宽为 8 时，需要将 HMC_H IP 的高位数据线和时钟线悬空。

5. pad_loop_in, pad_loop_in_h, pad_loop_out, pad_loop_out_h 为温度补偿 IO，制作 PCB 时需要悬空。

九 附录