

# 基于 Innovus 工具的 28 nm DDR PHY 物理设计方法

王秋实, 张 杰, 孟少鹏

(中国电子科技集团公司第三十八研究所, 安徽合肥 230088)

**摘 要:** 随着 CPU、DSP 等器件的处理速度迅速提高,对内存的速度和各方面的需求迅速增加。早期的 SDRAM 工作频率发展到 133 MHz 已到极限,成为系统性能的瓶颈。DDR(双倍数据率)技术随之应运而生,目前 DDR4 的性能已经可以达到 3 200 Mbps 级别。DDR PHY 作为存储控制器和 DRAM 颗粒物理接口之间的通用接口,是制约 DDR 读写速度提升的关键。本文以 TSMC 28 nm 工艺的 DDR PHY 设计为例,结合 Innovus 工具,在描述流程之外,重点研究解决了后端物理设计中时序路径的时间预算、延时优化、路径对齐等问题。最后该 DDR PHY 在一款工业级 DSP 中成功集成,并且板级测试结果表明其物理设计结果达到指标要求。

**关键词:** DDR PHY; 物理设计; Innovus; 时间预算; 延时优化; 路径对齐

**中图分类号:** TN47 **文献标志码:** A **文章编号:** 1672-2337(2020)04-0457-04

## Physical Design Methods of a 28 nm DDR PHY Based on Innovus

WANG Qiushi, ZHANG Jie, MENG Shaopeng

(The 38th Research Institute of China Electronics Technology Group Corporation, Hefei 230088, China)

**Abstract:** With the rapid increase in the processing speed of CPU and DSP, the demand for speed and other various aspects of memory has increased rapidly. The early SDRAM operating frequency has developed to the limit of 133 MHz, which became the bottleneck of system performance. Afterwards, DDR (double data rate) technology came into being, and the performance of DDR4 can reach 3 200 Mbps. DDR PHY, as a common interface between the memory controller and high-speed DDR DRAM device, is the key to restricting the read-write speed. This paper describes physical design methods of a 28 nm DDR PHY based on Innovus, with focus on timing budget, delay optimization and path balancing. Finally, this DDR PHY is integrated into an industrial DSP. The board-level test results show that the DDR PHY performance meets the requirements.

**Key words:** DDR PHY; physical design; Innovus; timing budget; delay optimization; path balancing

## 0 引 言

DDR 技术起始于 2001 年,之后广泛应用于各类芯片并成为主流。基于这个前提,双倍数据率物理层接口(DDR PHY)技术应运而生。DDR PHY 成为存储控制器逻辑和 DRAM 颗粒物理接口间的通用规范接口,加速了 DDR 技术的发展。在芯片中,DDR PHY 通常作为一个独立模块单独设计<sup>[1]</sup>。

集成电路设计分为前端设计和后端设计。前端设计包含系统架构设计、子模块设计、硬件描述语言实现及功能仿真验证等。后端设计是将前端设计映

射到可加工的硅工艺的过程,包括综合、布局、电源布线、标准单元摆放、时钟树综合、绕线等<sup>[2]</sup>。

DDR PHY 的物理设计流程与全芯片的物理设计流程相同。针对 28 nm 工艺的特殊性,我们特别定制了一些设计流程,包括在布局阶段插入特殊的物理单元(Physical Cell)、时钟树绕线层(NDR)的选择、绕线策略的更改等<sup>[3-4]</sup>。DDR PHY 的物理设计的关键在于时序路径的设计,时序路径质量的好坏将严重影响 DDR 系统的性能。

本文首先在第一节介绍所用 DDR PHY 的架构,第二节描述了 DDR PHY 的时序路径及其后端设计的难点,第三节提出了基于 Innovus 工具的解决办法,最后总结全文。

## 1 DDR PHY 简介

本文中的 DDR PHY 是我单位采购的商用“半硬核”IP, 基于 28 nm 流片生产。它能够提供与 JEDEC 标准的 DDR3 和 DDR4 SDRAM 通信的完整物理接口, 最高数据率为 1 400 Mbps。

之所以称之为“半硬核”IP, 是因为整个 DDR PHY 基于 slice 构成, IP 开发商提供已经固化完成的 slice 模块。slice 模块是 DDR 控制器和 DRAM 颗粒的“桥梁”, 整个 DDR PHY 使用到的 slice 模块种类包括:

(1) Data slice (8 DQ/1 DM/1 DQS), 负责数据信号的串并转换;

(2) Address/control slice (4 control/address), 负责地址信号的转换;

(3) Address/control master slice, 负责控制各个 Address/control slice;

(4) Memory clock slice, 负责时钟信号的转换。

图 1 是 DDR PHY 的结构图。除 slice 模块外, 整个 DDR PHY 还包括一些数字模块, 称之为 soft module, 包括 DFI 接口、配置接口、内部寄存器等, 构成存储控制器和 slice 模块之间的电路。

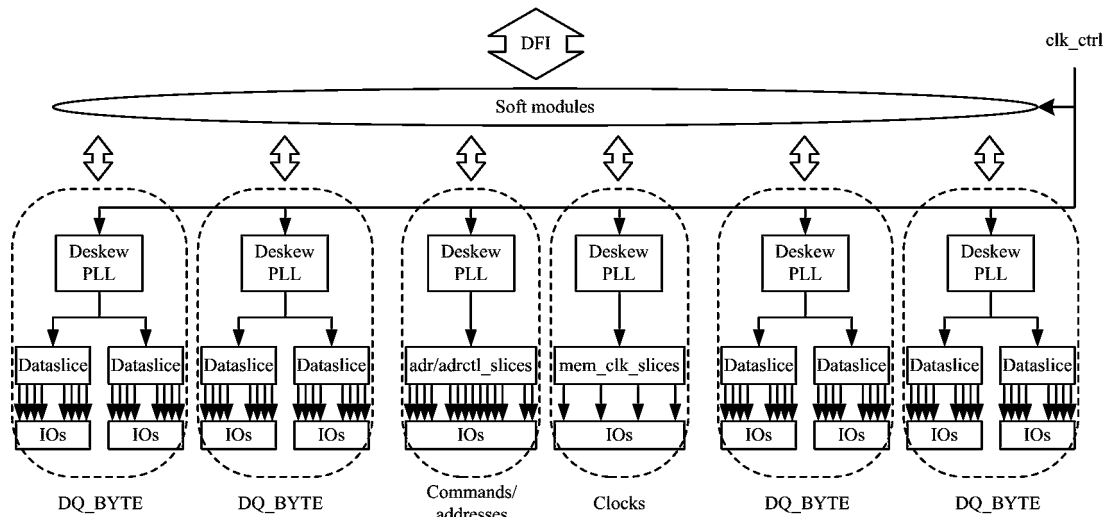


图 1 DDR PHY 的结构图

## 2 时序路径的设计难点

DDR PHY 的输出信号包括数据信号、地址信号、地址控制信号、时钟信号等, 每一条从时钟输入管脚到这些输出信号的路径都是需要重点关注的时序路径。

归纳来说, DDR PHY 中的时序路径可以分为如下 5 段:

Stage 1: 时钟输入端 `clk_ctrl` 到 `deskew PLL`;

Stage 2: `deskew PLL` 到 slice 模块时钟端;

Stage 3: slice 模块时钟端到内部寄存器 CK 端;

Stage 4: slice 模块内部寄存器到 slice 模块输出端;

Stage 5: slice 模块输出端到 IO。

DDR PHY 需要为存储控制器逻辑和 DRAM

颗粒之间提供高速、稳定、可靠的数据通信, 这对时序路径的整体延时、不同路径之间的延迟差等设计指标提出了苛刻的要求。

### 2.1 时间预算

时间预算就是在时序路径中分配延时的过程。由于各个 slice 模块已经固化, 因此 Stage 3 和 Stage 4 的延迟已经固定。为保证达到设计目标, 在设计初期需要时序路径中其他阶段的延时进行合理的规划。

### 2.2 延时优化

DDR 的存取时间代表了读取数据所延迟的时间, 与时序路径的整体延时直接相关。从设计指标上来说, 存取时间越小则越优。

在布局阶段, 通过将 `deskew PLL` 与 slice 模块就近摆放来减小 Stage 2 的延时; 通过将 slice 模

块与 IO 单元就近摆放来减小 Stage 5 的延时。

### 2.3 路径对齐

路径对齐是 DDR PHY 时序路径设计的最大难点。如图 2 所示,在所有延迟角落下,路径对齐要求如下:同一个 data slice 和同一个 address slice 的输出信号间的延迟偏移均不能超过 75 ps,同一个 address control slice 的输出信号间的延迟偏移不能超过 85 ps;memory clock slice 的输出信号的延迟要大于所有 data slice 和 address slice 的输出信号间的延迟,且它们之间的延迟偏移不能超过 125 ps; address slice、address control slice 和 memory clock slice 的输出信号间的延迟偏移不能超过 150 ps;所有 DDR PHY 的输出信号之间的延迟偏移不能超过 170 ps。

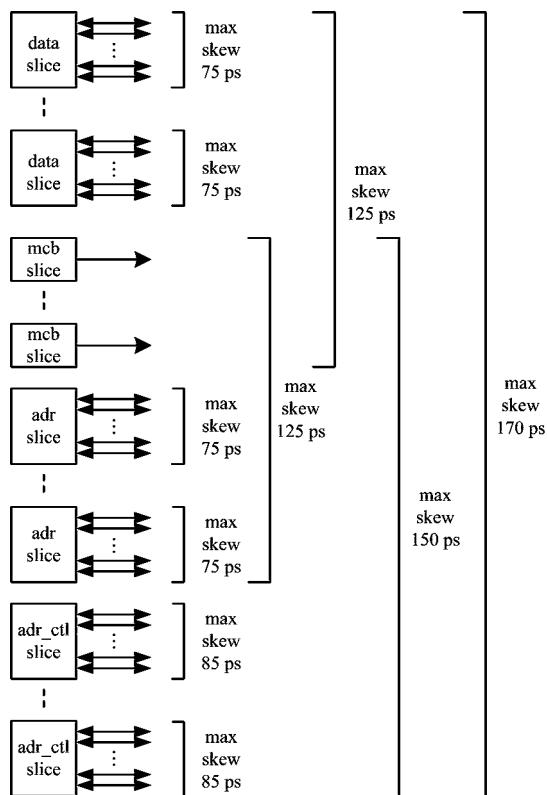


图 2 DDR PHY 路径对齐要求

## 3 基于 Innovus 的设计方法

Innovus 为 CADENCE 公司于 2015 年发布的全新后端物理设计平台,基于该平台可以进行从 RTL 到 GDSII 的全流程实现,采用平台内嵌命令结合 LINUX 可以很方便地实现对数字电路网表文件的编辑<sup>[5]</sup>。

本节将结合项目实例从 4 个方面来阐述 DDR PHY 时序路径设计难点的解决方法。

### 3.1 使用 DCCK 单元

DCCK 单元为专门设计的带耦合电容的时钟树驱动单元,具有如下优点:单元延迟经过优化,保证时钟路径的延迟尽可能的小;具有对称的上升时间和下降时间,保证了延时链的精度;由于耦合电容的存在,使时钟树的时钟延迟、传输时间、时钟偏移等参数不易受电源波动影响。

项目选取 DCCK 单元中阈值电压为 LVT、驱动能力为 X8/12/16 的缓冲器(buffer)和反相器(inverter)作为 DDR PHY 时钟路径的驱动单元,方法如图 3 所示。

```
set_ccopt_mode -cts_buffer_cells { \
  DCCKBD8BWP12T35P140LVT \
  DCCKBD12BWP12T35P140LVT \
  DCCKBD16BWP12T35P140LVT \
}

set_ccopt_mode -cts_inverter_cells { \
  DCCKND8BWP12T35P140LVT \
  DCCKND12BWP12T35P140LVT \
  DCCKND8BWP12T35P140LVT \
}
```

图 3 使用 DCCK 单元的设置脚本

### 3.2 利用 insertion delay

如前文中所说,各个 slice 模块已经固化,slice 模块时钟输入端到 slice 模块内部寄存器 CK 端的时钟树路径(即 Stage 3)已经固定,项目使用 insertion delay 代表该部分时钟树的延迟。

项目使用 create\_ccopt\_macro\_model\_spec 命令将传统 cts 引擎的 macro model delays 转换成 ccopt 引擎中的 insertion delay。需要注意的是,为了保证时钟树设计的精确性,需要在多模式多角落(MMMC)下设置 insertion delay。图 4 为使用 insertion delay 的部分设置脚本。

```
set_ccopt_property insertion_delay 0.252 \
  -pin ${ds_hier}/inst_data_slice_num_0/clk
-delay_corner dc_cworst_T_125c

set_ccopt_property insertion_delay 0.260 \
  -pin ${ds_hier}/inst_data_slice_num_0/clk
-delay_corner dc_rcworst_T_125c

set_ccopt_property insertion_delay 0.270 \
  -pin ${ds_hier}/inst_data_slice_num_0/clk
-delay_corner dc_cworst_125c

set_ccopt_property insertion_delay 0.288 \
  -pin ${ds_hier}/inst_data_slice_num_0/clk
-delay_corner dc_rcworst_125c
```

图 4 使用 insertion delay 的设置脚本

### 3.3 创建 skew group

在 ccopt 中, skew group 可以看作整个时钟树的子时钟树, 用户可以根据需要定制 leaf pin、insertion delay、target skew 等参数。

为了满足 2.3 节提到的路径对齐要求, 项目创建不同的 skew group 来调整不同类别时序路径的延时。主要进行延时调整的时序路径为时钟输入管脚 clk\_ctrl 到 deskew PLL 的时钟树路径 (即 Stage 1)。图 5 所示为将所有 data slice 的 deskew PLL 的参考时钟输入端创建 skew group。

```
create_ccopt_skew_group -name ds_pll_ref \
  -target_insertion_delay 0.70 -target_skew 0.01 \
  -from_delay_corners dc_cworst_T_125c \
  -sources clk_ctrl \
  -shared_sinks { \
    ${ds_pll_0_hier}/FREF \
    ${ds_pll_1_hier}/FREF \
    ${ds_pll_2_hier}/FREF \
    ${ds_pll_3_hier}/FREF }
```

图 5 创建 skew group 脚本

### 3.4 手工调整

为了使 deskew PLL 到 slice 模块时钟端 (即 Stage 2) 的延时尽可能小, 需要手工调整该段路径上时钟缓冲器的位置, 使时钟走线结果达到最优。

在最终的设计结果仍然无法达到路径对齐的要求但不是很大的情况下, 可以根据静态时序分析 (STA) 结果手工添加缓冲器来调整, 主要进行手工调整的时序路径为 slice 模块输出端到 IO 的数据路径 (即 Stage 5)。

### 3.5 设计结果

DDR PHY 最终的 PPA (Performance, Power, Area) 结果为:

- (1) 尺寸为: 宽 8.7  $\mu\text{m}$ , 高 1.4  $\mu\text{m}$ ;
- (2) 工作频率 700 MHz, 速率 1 400 Mbps;
- (3) 功耗 1.5 W, 静态电压降 1.3%。

图 6 为 DDR PHY 完成物理设计后的结果局部图。

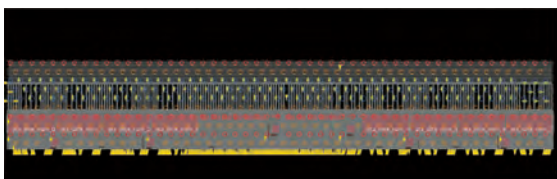


图 6 DDR PHY 物理设计结果图

表 1 为延迟偏移最大的工艺角情况下各时序路径的延时信息。从表中可以看出, 所有 DDR

PHY 的输出信号之间的延迟偏移为 122 ps, 满足 170 ps 的设计要求。

表 1 时序路径的时序预算

路径	延时/ns			
	min rise	max rise	min fall	max fall
DQ0	1.330	1.383	1.313	1.365
DQ1	1.339	1.389	1.321	1.371
DQ2	1.339	1.387	1.331	1.377
DQ3	1.342	1.393	1.333	1.383
DQ4	1.355	1.412	1.337	1.392
DQ5	1.354	1.408	1.336	1.388
DQ6	1.324	1.379	1.307	1.361
DQ7	1.317	1.370	1.300	1.351
ADR	1.309	1.341	1.396	1.329
ACTL	1.309	1.343	1.396	1.331
MCB	1.414	1.422	1.405	1.414

## 4 结束语

集成了该 DDR PHY 的一款工业级 DSP 已经成功流片并通过测试, 在测试板上, DDR 系统能够稳定运行在 1 400 Mbps 速率上。测试结果进一步证明了本文提出的 DDR PHY 物理设计方法的可行性和有效性。另外, 随着工艺推进到 12 nm 及对 DDR PHY 的速率要求提升到 3 200 Mbps, 本文提出的物理设计方法还有进一步的提升空间。

### 参考文献:

- [1] 卢俊. 基于 65nm DDR PHY 数字后端设计方法的研究 [D]. 上海: 复旦大学, 2010.
- [2] 秦晓凌, 潘中平. ASIC/SOC 后端设计作业流程剖析 [J]. 中国集成电路, 2002, 38(7): 79-85.
- [3] 沈良伟. 基于 28 纳米工艺的光通信芯片低功耗物理设计 [D]. 成都: 电子科技大学, 2014.
- [4] 高明. 28 nm 工艺下双核 Cortex-A9 处理器芯片的物理设计 [D]. 南京: 东南大学, 2016.
- [5] 戈喆, 王志鸿, 厉媛玥. 基于 Innovus 的低功耗物理设计 [J]. 电子技术应用, 2016, 42(8): 21-24.

### 作者简介:



王秋实 男, 1989 年 10 月生, 安徽阜南人, 2010 年于北京大学获得学士学位, 2013 年于北京大学获得理学硕士学位, 现为中国电子科技集团公司第三十八研究所后端设计工程师, 主要研究方向为大规模集成电路的物理实现。  
E-mail: iamwangqiushi@163.com