



XAPP709 (v2.0) 2006 年 10 月 27 日

使用 Virtex-4 FPGA 器件实现 DDR SDRAM 控制器

作者：Rich Chiu

提要

本应用指南描述了在 Virtex™-4 XC4VLX25 FF668 -10C 器件中实现的 DDR SDRAM 控制器。该实现运用了直接时钟控制技术来实现数据采集，并采用自动校准电路来调整数据线上的延迟。

DDR SDRAM 器件是低成本、高密度的存储资源，在很多存储器供应商处均可获得。本参考设计使用 SDRAM 器件和 DIMM 开发而成。

DDR SDRAM 描述

DDR SDRAM 规范的详情可在电子工业联盟 (Electronic Industries Alliance, EIA) 成员 JEDEC 机构的网站上获得，网址为 <http://www.jedec.org/>。DDR SDRAM 规范刊载在参考文件 JESD79E 下的 JEDEC 文档。

DDR SDRAM 器件是硅片存储器资源，现在最经常用于各种系统，包括从消费产品到视频系统的各类应用。DDR SDRAM 器件的频率高达 200 MHz 或 DDR400。DRAM 可以使用器件或模块的配置。

DDR 控制器命令

表 1 所示为控制器发出的命令。这些命令通过使用下列控制信号传输到存储器：

- 行地址选择 ($\overline{\text{RAS}}$)
- 列地址选择 ($\overline{\text{CAS}}$)
- 写使能 ($\overline{\text{WE}}$)
- 时钟使能 (CKE) (器件配置后始终置为高)
- 芯片选择 ($\overline{\text{CS}}$) (器件运行期间始终置为低)

表 1: DDR SDRAM 命令

信号编号	功能	$\overline{\text{RAS}}$	$\overline{\text{CAS}}$	$\overline{\text{WE}}$
1	加载模式寄存器 (Load Mode Register)	L	L	L
2	自动刷新 (Auto Refresh)	L	L	H
3	预充电 (Percharge) ⁽¹⁾	L	H	L
4	选择组激活行 (Select Bank Activate Row)	L	H	H
5	写命令 (Write Command)	H	L	L
6	读命令 (Read Command)	H	L	H

© 2004–2006 Xilinx, Inc. All rights reserved. All Xilinx trademarks, registered trademarks, patents, and further disclaimers are as listed at <http://www.xilinx.com/legal.htm>. All other trademarks and registered trademarks are the property of their respective owners. All specifications are subject to change without notice.

NOTICE OF DISCLAIMER: Xilinx is providing this design, code, or information "as is." By providing the design, code, or information as one possible implementation of this feature, application, or standard, Xilinx makes no representation that this implementation is free from any claims of infringement. You are responsible for obtaining any rights you may require for your implementation. Xilinx expressly disclaims any warranty whatsoever with respect to the adequacy of the implementation, including but not limited to any warranties or representations that this implementation is free from claims of infringement and any implied warranties of merchantability or fitness for a particular purpose.

表 1: DDR SDRAM 命令 (续表)

信号编号	功能	$\overline{\text{RAS}}$	$\overline{\text{CAS}}$	$\overline{\text{WE}}$
7	空操作 (No Operation, NOP)	H	H	H

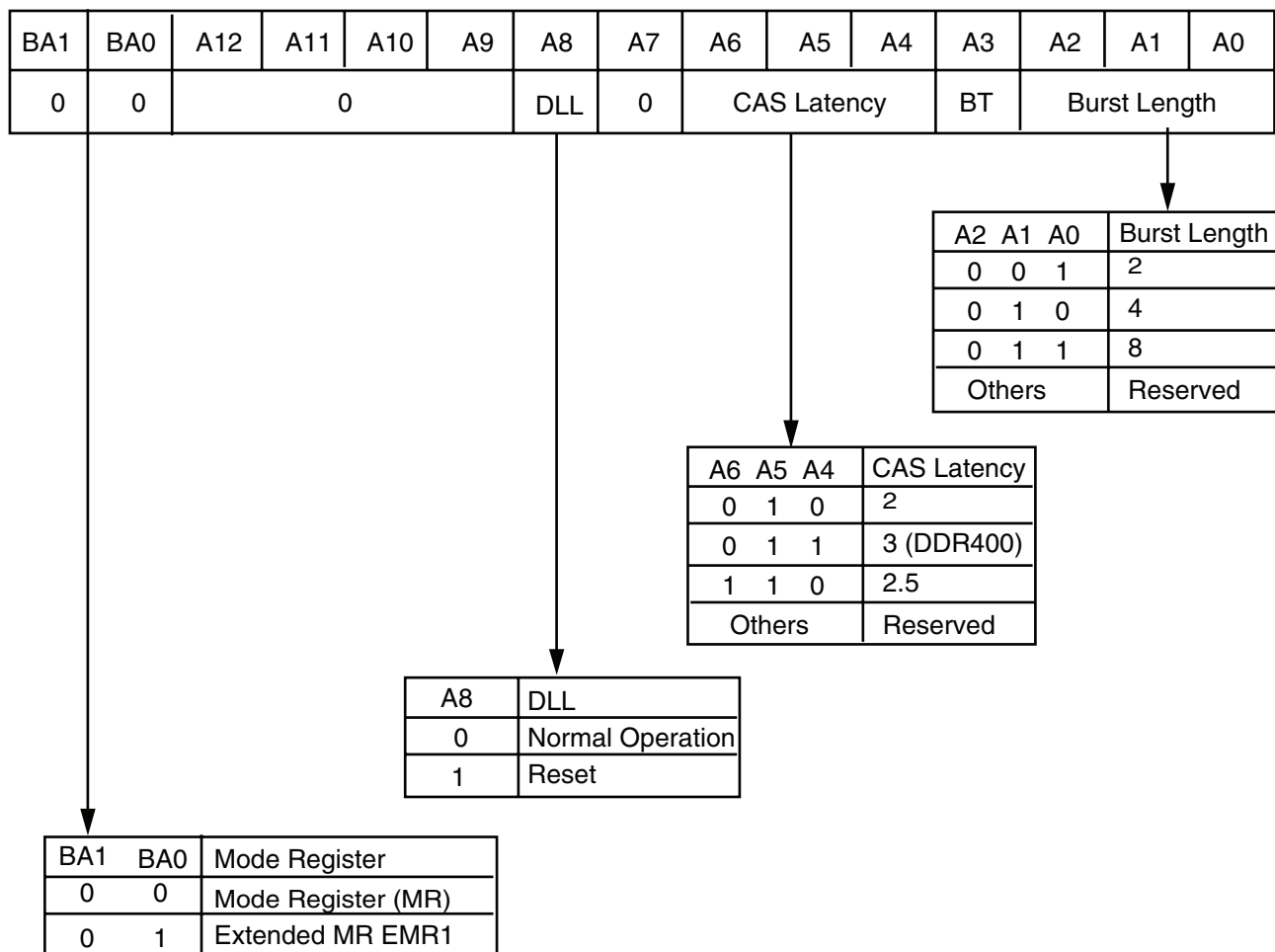
注:

- 地址信号 A10 在预充电所有组期间设定为高, 在单个组预充电期间设定为低。

命令功能

模式寄存器 (Mode Register)

模式寄存器用于定义 DDR SDRAM 特定的运行模式, 包括突发长度 (burst length)、突发类型、CAS 延迟 (CAS latency) 和运行模式的选择。图 1 所示为此控制器所用的模式寄存器的功能。



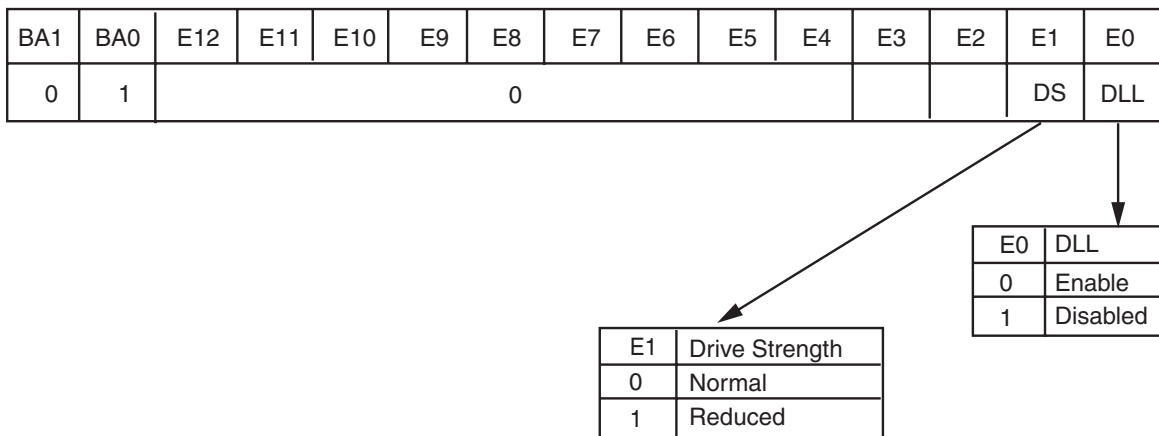
x709_01_090304

图 1: DDR400 的模式寄存器定义

组地址 (Bank Address) BA1 和 BA0 用于选择模式寄存器。图 1 显示了组地址位配置。

扩展模式寄存器

模式寄存器控制范围之外的功能由扩展模式寄存器控制。这些附加功能是由于 DDR SDRAM 接口的 DLL 使能 / 禁能和输出驱动强度, 如图 2 所示。

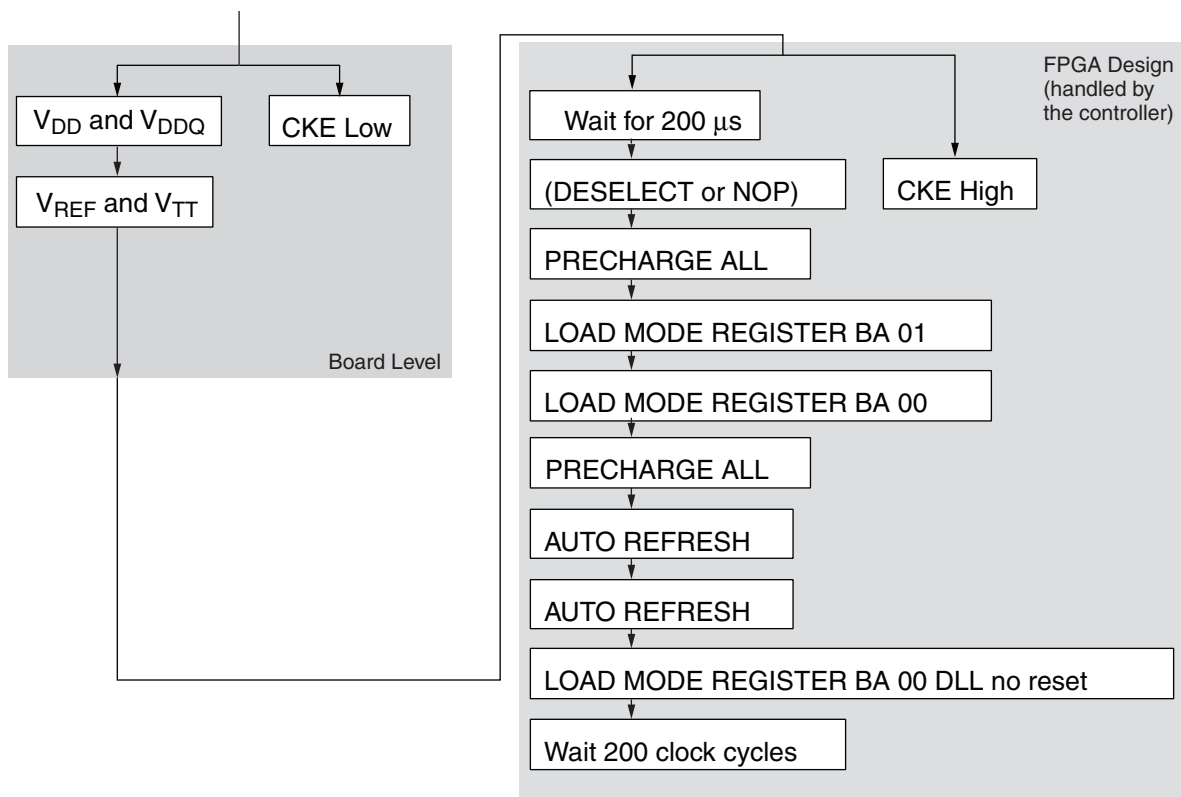


x709_02_032405

图 2: DDR400 的扩展模式寄存器

初始化顺序

此控制器状态机使用的初始化顺序遵循 DDR SDRAM 规范。配置顺序分为两步：第一步由硬件在上电时处理；第二步由 FPGA 存储器控制器设计处理。图 3 所示为针对初始化而发出的命令顺序。



x709_03_020206

图 3: 使用 Virtex-4 器件的 DDR SDRAM 初始化顺序

初始化顺序完成后，控制器向 DDR SDRAM 存储器器件发出虚读命令。该过程允许数据通路模块在 Virtex-4 输入延迟模块中选择正确的 tap 数量。数据通路模块使用存储器在此虚读命令周期发出的数据选通脉冲 (DQS) 来确定输入的 DQS 和内部系统时钟 (CLK0) 之间的关系。数据通

路模块确定所要求的正确数量的延迟 tap 后，一个 Tap_select_done 信号便被发送到控制器。然后控制器进入空闲状态。

读使能校准

DQS 校准后，控制器发出一个命令，向存储器写入一个预定义数据模式 (pattern)。写操作结束后，数据被读回，并与已写的数据进行比较。一个模块会计算发出读命令后，在该模块中接收到有效数据所需的时钟周期数。该数值用于在读使能信号上建立正确的内部延迟，以保证正常的读操作。

预充电 (PRECHARGE) 命令

预充电命令用于取消特定组中活动行的活动状态。在预充电命令发出的一定时间 (t_{RP}) 后，此组在接下来的行激活命令中可以使用。输入 A10 确定对一个组还是全部组进行预充电。

自动刷新 (AUTO REFRESH) 命令

DDR 器件需要每 7.8 μs 刷新一次。要求自动刷新计数器的电路放置在控制器内。控制器将 DCM 的 CLKDV 输出用于刷新计数器。该输出提供自动刷新计数器需要的低频率时钟。要节省 DCM 的 CLKDV 输出使用的 BUFG，设计人员可以使用 DCM 的高频 CLK0 输出或 DCM 的 CLK/4 输出（用于 IDELAY 电路）作为时钟来驱动刷新计数器。如果自动刷新电路的时钟改变，mem_interface_top_parameters_0.v 文件中的 max_ref_count 也应做相应改变。

auto_ref 信号标志出需要一个待发的自动刷新命令。在控制器发出自动刷新命令之前，此信号一直为高。在发出自动刷新命令前，控制器要完成当前活动组中的事务。

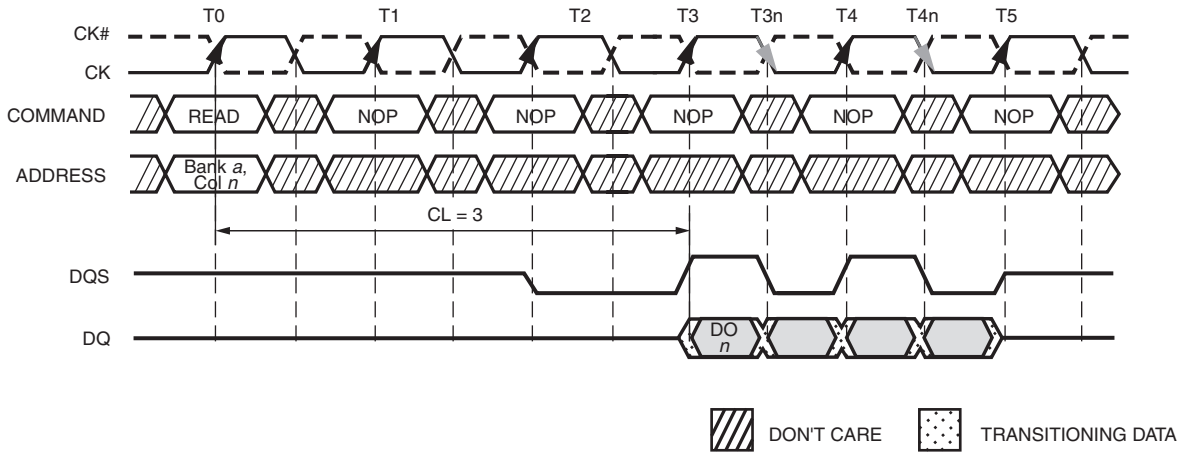
激活命令

激活命令激活一个组中的一行，允许任何读或写命令发送到存储器阵列中的一个组。行被激活后，读或写命令就可以按照 t_{RCD} 规范发送到行。当控制器发现输入的地址是关于组内某行而非当前活动时，即会发出地址冲突信号。预充电命令也由控制器发出，取消活动行的活动状态。控制器还会向新行发出另一个激活命令。

读命令

读命令用于发起对活动行的突发式读访问。BA0 和 BA1 上的值选择组地址，而 A0 – Ai 上提供的地址输入选择起始列位置。读突发结束后，只要还未预充电，此行仍可用于后面的访问。

图 4 所示为一个附加延迟为零的读命令示例。因此，在该示例中，读延迟与 CAS 延迟相同，在 DDR400 规范中为三个时钟周期。



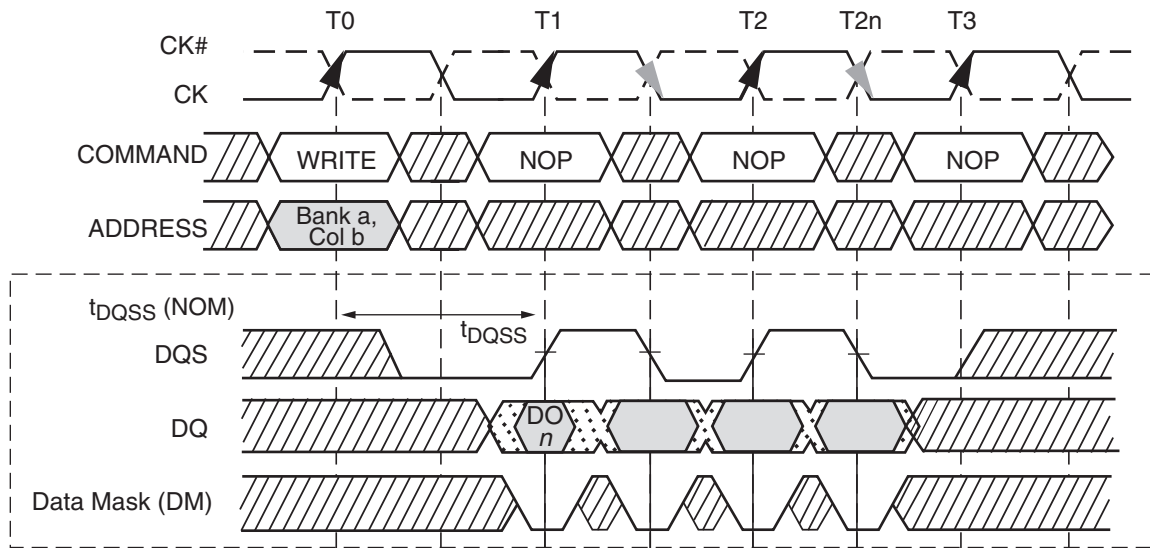
X709_04_013006

图 4: DDR SDRAM 读访问波形

写命令

写命令用于发起对活动行的突发式访问。BA0 和 BA1 上的值选择组地址，而地址输入 A0 - Ai 的值选择活动行内的起始列位置。写延迟的值等于一个时钟周期。

图 5 所示为一个写延迟为 1 的写突发。写命令和 DQS 信号的第一个上升沿之间的时间间隔由写延迟确定。



X709_03_110805

图 5: DDR SDRAM 写命令波形

时序分析

本 Virtex-4 DDR 参考设计利用该器件独特的 I/O 和时钟控制功能，可最大限度地提高性能和时序余量。

作为此参考设计基础的物理层采用了直接时钟控制方法。

有关详情请参阅 XAPP701: <http://www.xilinx.com/cn/bvdocs/appnotes/xapp701.pdf>。

本部分针对地址 / 控制通路、写数据通路及读（或采集）数据通路给出一个时序分析示例。

地址 / 控制通路

地址和控制信号在 CLK180 上保持同步，为与时钟 CLK0 有关的存储器器件确保充足的建立及保持时间余量。

表 2 所示为频率为 165 MHz 时，DDR 接口的地址和控制信号时序分析。设计人员必须把其它与特定电路板相关的情况考虑进来，例如在速度级别为 -10 的 Virtex-4 器件中的实现，时钟和地址 / 控制网络的负载会有差别。

表 2: 以 165 MHz 运行时的地址和控制信号时序分析

参数	值 (ps)	前沿不确定度	后沿不确定度	描述
T _{CLOCK}	6061			时钟周期。
T _{SETUP}	600	600	0	–
T _{HOLD}	600	–	600	–
T _{PACKAGE_SKEW}	±20	20	20	封装歪斜。
T _{DCCD_MEMORY_DLL}	±150	150	150	CLK0 的下降沿（也即 CLK180 的上升沿）用于驱动 DDR CK 的上升沿。CLK0 的上升沿用于驱动地址 / 控制。
T _{JITTER}	±0	0	0	同一个 DCM 输出，用于生成 CK 和地址 / 控制。
T _{CLOCK_TREE_SKEW}	±100	100	100	采用了小值。假定 CK/CK# 输出距控制 / 地址很近。
T _{CLKOUT_PHASE}	±140	140	140	任何 DCM 输出参数值间的相移，参见 DS302 : 《Virtex-4 数据手册》。
T _{PCB_LAYOUT_SKEW}	±400	400	400	地址 / 控制线路和电路板上相关联的 CK/CK# 迹线间的歪斜。
总不确定度		1410	1410	–
时序余量		1620	1620	–

写数据通路

写数据通路与 CLK90 保持同步。但是，写数据字是作为 DDR 值被发送的，所以对于 CLK180 的上升沿和下降沿都必须有充足的建立及保持时间余量。

相应地，写数据通路的时序分析结合了存储器时钟的最大占空比失真，如表 3 所示。该分析也适于速度级别为 -10 的 Virtex-4 器件。

表 3: 以 165 MHz 运行时的写数据通路时序分析

参数	值 (ps)	前沿不确定度	后沿不确定度	描述
T _{CLOCK}	6061	–	–	时钟周期。
T _{DCCD}	±303	303	303	存储器时钟的占空比失真（时钟周期的 5%）。

表 3: 以 165 MHz 运行时的写数据通路时序分析 (续表)

参数	值 (ps)	前沿不确定度	后沿不确定度	描述
$T_{\text{CLOCK_PHASE}}$	2727	–	–	数据周期是时钟周期的一半，再减去 10% 的占空比失真。
T_{SETUP}	400	400	0	存储器数据手册上的建立时间。
T_{HOLD}	400	0	400	存储器数据手册上的保持时间。
$T_{\text{PACKAGE_SKEW}}$	± 20	20	20	封装歪斜。
T_{JITTER}	± 50	50	50	同一个 DCM 用于生成 CLK0/CLK180 和 CLK90。只需考虑 CLK0 输出和 CLK90 输出间抖动的差异。
$T_{\text{CLOCK_SKEW_FPGA}}$	± 100	100	100	为全局时钟线路上的歪斜采用了小值，因为检测到的 DQS 和相关联的 DQ 彼此相邻。
$T_{\text{CLKOUT_PHASE}}$	± 140	140	140	任何 DCM 输出参数值间的相移，参见《Virtex-4 数据手册》。
$T_{\text{PCB_LAYOUT_SKEW}}$	± 50	50	50	板上的数据线和相关联的 DQS 之间的歪斜。
总不确定度		760	760	–
时序余量		603	603	

读数据通路

通过使用此前介绍过的直接时钟控制技术，读数据通路值将被直接采集到 FPGA 时钟域中。因此，必须就 CLK0 进行数据采集时序分析，还必须考虑 IDELAY tap 延迟的分辨率。另外，也必须考虑 DQS 选通脉冲和总线上数据位间潜在的歪斜。表 4 所示为读数据采集的时序分析，假定采用的器件还是速度级别为 -10 的 Virtex-4。《XAPP701》详细讨论了用于读时序分析的延迟参数。

表 4: 以 165 MHz 运行时的读数据通路时序分析

不确定度参数	值 (ps)	描述
T_{CLOCK}	6061	时钟周期。
$T_{\text{CLKOUT_DUTY_CYCLE_DLL}}$	150	从时钟相位（等于半个时钟周期）减去 DCM 输出占空比失真可确定 $T_{\text{DATA_PERIOD}}$ 。
$T_{\text{DATA_PERIOD}}$	2880	数据周期是时钟周期的一半，再减去占空比失真。
T_{AC}	1400	指定的存储器数据输出访问时间。

表 4: 以 165 MHz 运行时的读数据通路时序分析 (续表)

不确定度参数	值 (ps)	描述
T _{PACKAGE_SKEW}	20	为封装歪斜采用了一个小值, 因为 PCB 迹线长度的调整可以补偿这一歪斜。
T _{SAMP}	550	该参数在 Virtex-4 源同步数据手册中定义, 包括不确定度来源的数量。
T _{IDELAYPAT_JIT}	732	以 165 MHz 运行, 在最差的情况下, tap 的总数为 $3/4 \times \text{clock_period} = 61$ taps。模式抖动为 $61 \times 12 = 732$ ps。
T _{CLOCK_TREE_SKEW} - 最大值	100	为全局时钟线路上的歪斜采用了小值, 因为 DQS 和相关联的 DQ 彼此相邻。
T _{PCB_LAYOUT_SKEW}	50	板上的数据线和相关联的 DQS 之间的歪斜。
不确定度	2852	
窗口	28	

DDR SDRAM 接口的实现

这一部分介绍了 DDR SDRAM (DDR SDRAM controller) 控制器和接口的参数、接口框图 (图 6) 和控制器状态机 (图 7)。

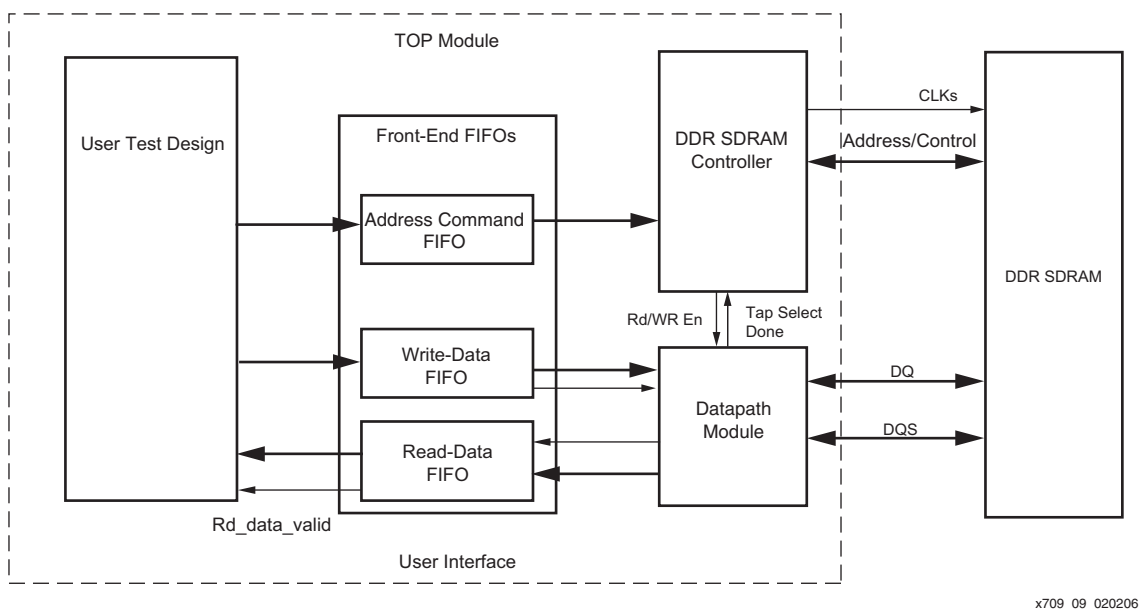


图 6: 控制器设计框图

硬件测试平台

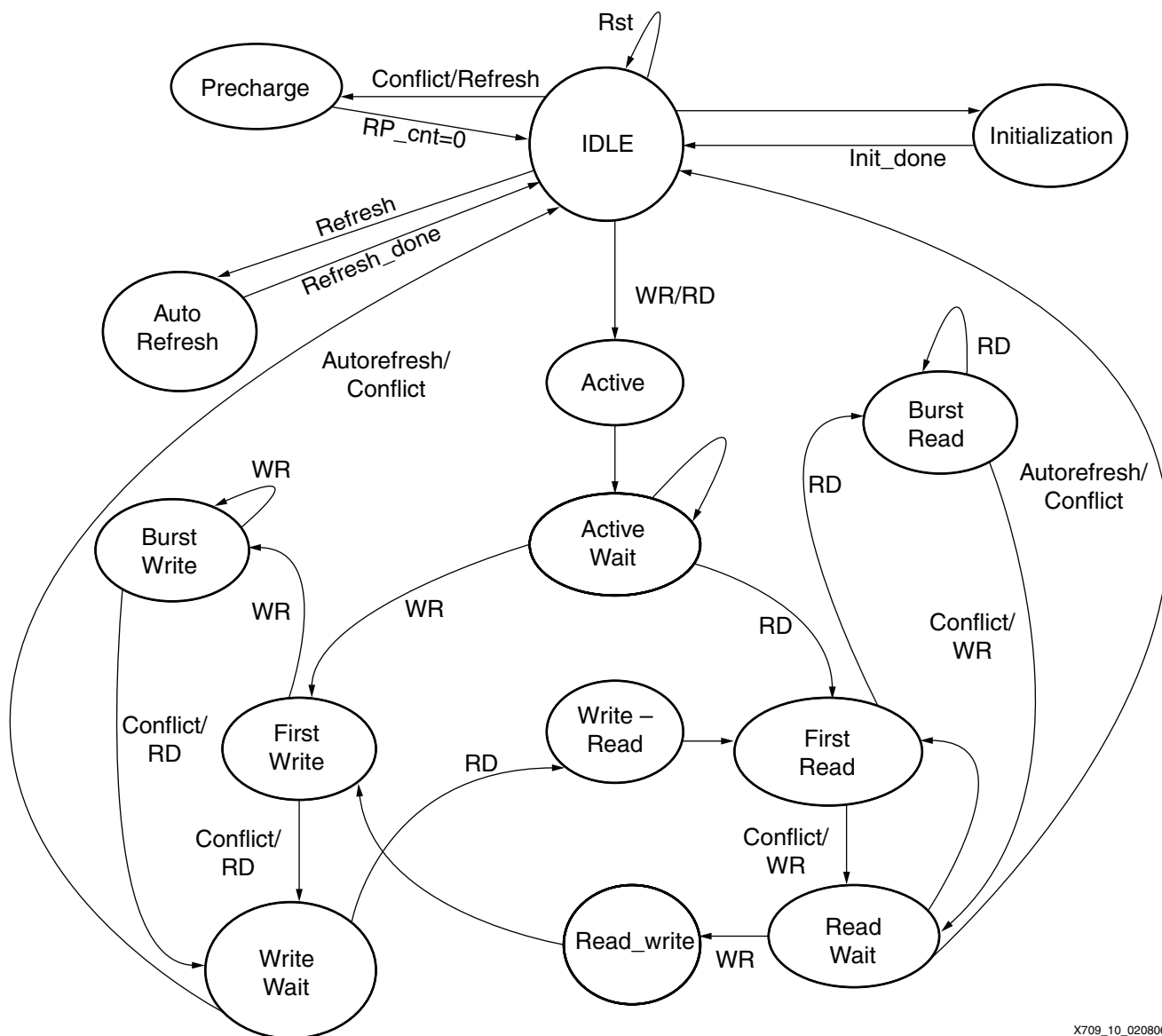
硬件测试平台提供地址和数据模式, 对 DDR SDRAM 控制器的各个设计方面进行测试。用户后端包括以下模块: 后端状态机模块、读数据比较器模块和数据生成器模块。数据生成器模块生成各种写到存储器器件的地址和数据模式。地址位置预存储在 Block RAM 中, 在这里用作只读存储器 (ROM)。存储的地址值被选择后, 用于测试对 DDR SDRAM 器件中的不同行和组的访问。数据模式生成器包括一个状态机, 用于发送数据模式。后端状态机充当用户设计, 并发出写使能或读使能信号, 以确定数据生成器模块需要访问哪一个 FIFO。

用户接口 (User Interface)

后端用户接口由三个 FIFO 构成，即地址命令 FIFO (Address Command FIFO)、写数据 FIFO (Write-Data FIFO) 和读数据 FIFO (Read-Data FIFO)。前两个 FIFO 通过用户后端模块访问，而读数据 FIFO 则通过数据通路模块 (Datapath Module) 访问，以存储采集的读数据。

DDR SDRAM 控制器接口

图 7 所示为 DDR SDRAM 命令生成状态机的状态机。



X709_10_020806

图 7: DDR SDRAM 控制器状态机

在控制器向存储器发出命令之前要执行这些步骤：

1. 控制器向读 / 写地址 FIFO 发出读使能信号。
2. 如果所有组都已预充电 (Percharge)，控制器会激活相应组中的某行；或将这些组、行地址与已处于活动状态的组、行地址进行比较。如果存在冲突，在进入读 / 写状态前，控制器会预充电活动组，然后发出一个激活 (Active) 命令。
3. 在写状态下，如果控制器检测到读命令，就会等待 write_to_read 的时间结束后再发出读命令。同样，在读状态中，检测到来自命令逻辑模块的写命令后，控制器会等待 read_to_write 的时间结束后再发出写命令。

4. 如果后端用户应用发出了动态命令请求，如预充电、自动刷新 (Auto Refresh)、激活或加载模式寄存器，控制器就会发出预充电命令。
5. 命令经过流水处理而与地址信号同步，然后被发送到 DDR 存储器。

参考设计规范

DDR SDRAM 存储器控制器使用了直接时钟数据采集技术，其参考设计中加进了存储器接口生成器 (MIG) 工具。此工具已与 Xilinx CORE Generator™ 软件集成。要获得最新版本的设计，请通过 Xilinx 网站下载 IP 更新，网址为：

http://www.xilinx.com/cn/xlnx/xil_sw_updates_home.jsp

该设计已针对高于 200 MHz 的运行频率进行了硬件描述。表 5 提供了参考设计规范。

表 5: 参考设计规范

参数	规范 / 细节
运行频率	165 MHz
Virtex-4 器件速度级别	-10
使用测试平台和 ChipScope™ 的组件设计中的器件利用率	1269 个 Slice
仅 DIMM 接口的 Slice 数量	2846 个 Slice
仅组件接口的 Slice 数量	944 个 Slice
支持的突发模式	2、4 和 8
支持的 CAS 延迟	2、2.5 和 3
HDL 语言	Verilog、VHDL
总线宽度	8-144 位
用于组件验证的器件	Micron MT46V32M16
用于 DIMM 验证的器件	Micron MT18VDDF6472AG-40BG4

设计文件

表 6 列出了参考设计文件：

表 6: 设计文件列表及对 Verilog 设计的描述

模块名称	各个模块的描述
top.v	DDR 控制器和物理层顶层模块。
data_path.v	物理层顶层。例化下列模块：tap_ctrl、data_tap_inc、idelay_ctrl、idelay_rd_en、v4_dqs_iob、v4_dq_iob 和 rd_data_fifo。
data_tap_inc.v	为与 DQS 相关联的数据位实现 tap 选择控制器。

表 6: 设计文件列表及对 Verilog 设计的描述 (续表)

模块名称	各个模块的描述
idelay_ctrl.v	在该设计中使用 IDELAY 基元时, 例化所需要的 IDELAYCTRL 基元。
tap_ctrl.v	该模块检测 DQS 信号的两个转换, 并确定相关联的数据位所需要的 tap 延迟, 以使这些数据位以内部 FPGA 时钟 CLK 为中心对齐。
v4_dm_iob.v	针对双向数据例化 IDELAY 基元和 IOB 触发器。
v4_dq_iob.v	针对双向数据例化 IDELAY 基元和 IOB 触发器。
v4_dqs_iob.v	针对双向 DQS 例化 IDELAY 基元和 IOB 触发器。
ddr_controller.v	<ul style="list-style-type: none"> 提供读使能信号给写地址、写数据和读地址 FIFO。 包括控制器状态机。提供正确的命令信号给 DDR 器件。自动刷新命令由控制器生成, 生成时考虑了自动刷新命令间隔。 提供地址信号给 DDR 器件。
test_bench.v	存储器接口的可综合测试平台。
backend_rom.v	存储要在存储器阵列中写入和访问的数据和地址。
cmp_rd_data.v	如果有位错误, 则生成错误信号。它会将读数据与预期数据值相比较。
user_interface.v	针对读数据例化 FIFO16 基元。一个 FIFO 用于上升沿数据, 另一个 FIFO 用于下降沿数据。
backend_fifo.v	针对 DDR SDRAM 接口例化后端 FIFO, 包括写地址和数据 FIFO, 以及读地址和数据 FIFO。这些 FIFO 使用 Virtex-4 FIFO16 基元或由 LUT RAM 实例产生的 FIFO 实现。
Rd_data_fifo.v	针对读数据例化 FIFO16 基元。一个 FIFO 用于上升沿数据, 另一个 FIFO 用于下降沿数据。
RAM_D.v	例化 LUT RAMS, 构建异步 FIFO。它被配置成时钟上升沿上的时钟数据。
parameter.v	用于 DDR SDRAM 参考设计 (运行频率为 200 MHz) 的值。
Infrastructure.v	Virtex-4 DCM 基元被例化。用于生成 FPGA clk_0、clk_90 以及分别与这两个时钟同步的系统复位。
Infrastructure_iobs.v	针对存储器时钟例化 ODDR 和 OBUFDS 基元。
Pattern compare.v	将来自存储器的读数据同一个固定模式进行比较, 计算从命令发出点的读使能延迟。
Main.v	例化 top.v 和 test_bench.v 模块。
Mem_interface_top.v	连接存储器接口和测试平台设计 (即时钟、复位和存储器器件) 的封装模块。
Addr_gen.v	在复位时, 对用户地址值和命令所在的 Block RAM 进行例化。
Data_gen.v	针对写和读数据生成数据模式。
Data_write.v	针对数据 IOB 生成控制信号。
Tap_logic.v	例化 tap_ctrl.v 和 data_tap_inc.v。

表 6: 设计文件列表及对 Verilog 设计的描述 (续表)

模块名称	各个模块的描述
jobs.v	例化 data_path_jobs.v、controller_jobs.v 和 infrastructure_jobs.v。
Data_path_jobs.v	例化 v4_dq_job、v4_dm_job 和 v4_dqs_job 模块。
Controller_jobs.v	针对存储器控制信号例化 OBUF 基元。
Wr_data_fifo.v	例化 FIFO16 基元，该基元存储用户数据和掩码信息。

设计层级

图 8 以树形图总结了参考设计中的层级。

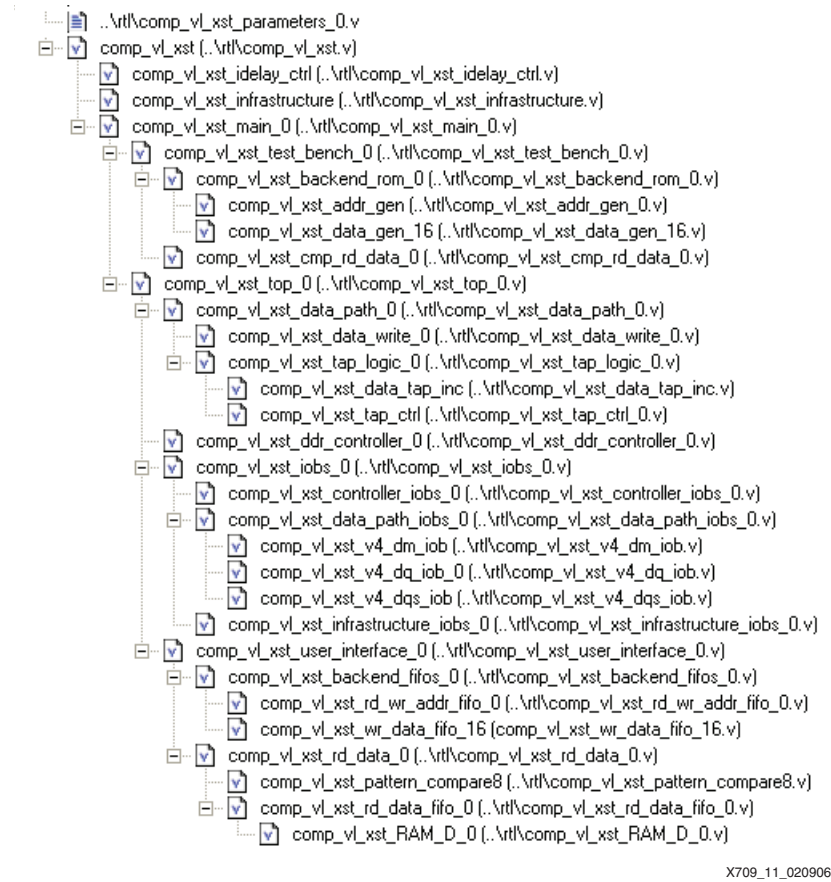


图 8: 设计层级

修订历史

下表说明此技术文档的修订历史。

日期	版本	修订
2004 年 9 月 10 日	1.0	Xilinx 最初版本。
2004 年 11 月 16 日	1.1	修改“激活命令”和“写命令”部分。添加参考设计链接。完成表 6。
2005 年 4 月 1 日	1.2	更新图 2、图 5、参考设计中含有在 ML461 平台上进行硬件测试的结果、“设计层级”、表 5 和表 6。添加“时序分析”。
2005 年 8 月 27 日	1.3	阐明表 2 和表 3。更新带有 ChipScope 文件的参考设计。
2005 年 11 月 18 日	1.4	更新参考设计链接。请参阅“参考设计规范”。更新读数据时序分析（参阅表 4）和图 8。
2006 年 3 月 27 日	1.5	更新图 3，替换图 4，更新图 6、图 7 和图 8，添加“读使能校准”部分，更新“写命令”部分，更新表 5 和表 6。在 MIG 1.5 中更新参考设计。
2006 年 10 月 27 日	2.0	更新表 2、表 3 和表 4。删除图 6、图 7 和图 8。