



WP260 (v1.0) 2007 年 2 月 16 日

利用 Xilinx FPGA 和存储器接口 生成器简化存储器接口

作者: Adrian Cosoroaba

FPGA 设计人员在满足关键时序余量的同时力争实现更高性能, 在这种情况下, 存储器接口的设计是一个一向构成艰难而耗时的挑战。Xilinx FPGA 提供 I/O 模块和逻辑资源, 从而使接口设计变得更简单、更可靠。尽管如此, I/O 模块以及额外的逻辑还是需要由设计人员在源 RTL 代码中配置、验证、执行, 并正确连接到其余的 FPGA 上, 经过仔细仿真, 然后在硬件中验证, 以确保存储器接口系统的可靠性。

本白皮书讨论各种存储器接口控制器设计所面临的挑战和 Xilinx 的解决方案, 同时也说明如何使用 Xilinx 软件工具和经过硬件验证的参考设计来为您自己的应用 (从低成本的 DDR SDRAM 应用到像 667 Mb/s DDR2 SDRAM 这样的更高性能接口) 设计完整的存储器接口解决方案。

© 2007 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.

存储器接口趋势和 Xilinx 解决方案

20 世纪 90 年代后期，存储器接口从单倍数据速率 (SDR) SDRAM 发展到了双倍数据速率 (DDR) SDRAM，而今天的 DDR2 SDRAM 运行速率已经达到每引脚 667 Mb/s 或更高。当今的趋势显示，这些数据速率可能每四年增加一倍，到 2010 年，随着 DDR3 SDRAM 的出现，很可能超过每引脚 1.2 Gb/s。见图 1。

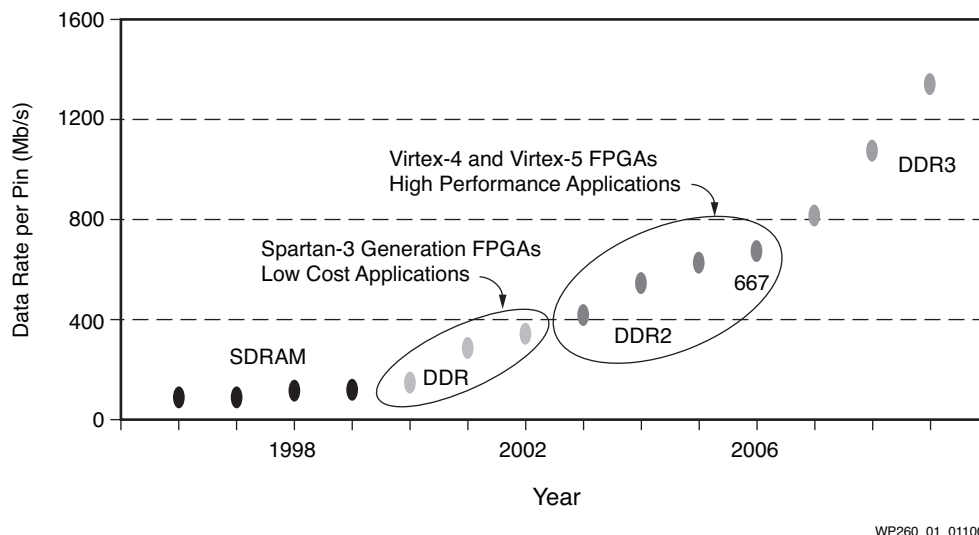


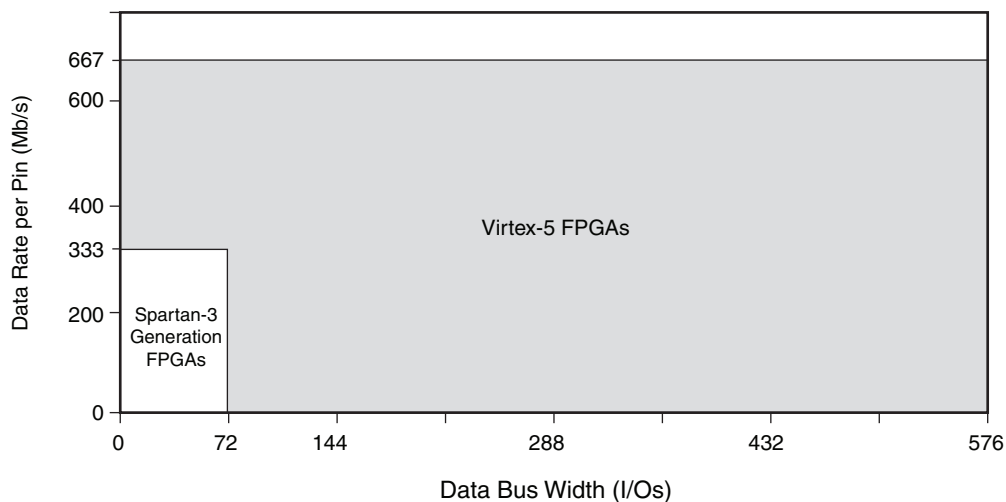
图 1: DRAM 数据速率趋势和 Xilinx 的 FPGA 解决方案

应用通常可分为两类：一类是低成本应用，降低器件成本为主要目的；另一类是高性能应用，首要目标是谋求高带宽。

运行速率低于每引脚 400 Mb/s 的 DDR SDRAM 和低端 DDR2 SDRAM 已能满足大多数低成本系统存储器的带宽需求。对于这类应用，Xilinx 提供了 Spartan-3 系列 FPGA，其中包括 Spartan-3、Spartan-3E 和 Spartan-3A 器件。

高性能应用把每引脚 533 和 667 Mb/s 的 DDR2 SDRAM 这样的存储器接口带宽推到了极限；对于这类应用，Xilinx 推出了 Virtex™-4 和 Virtex-5 FPGA，能够充分满足今天大多数系统的最高带宽需求。

带宽是与每引脚数据速率和数据总线宽度相关的一个因素。Spartan-3 系列、Virtex-4、Virtex-5 FPGA 提供不同的选项，从数据总线宽度小于 72 位的较小的低成本系统，到 576 位宽的更大的 Virtex-5 封装（见图 2）。



WP260_02_020507

图 2: Xilinx FPGA 和存储器接口带宽

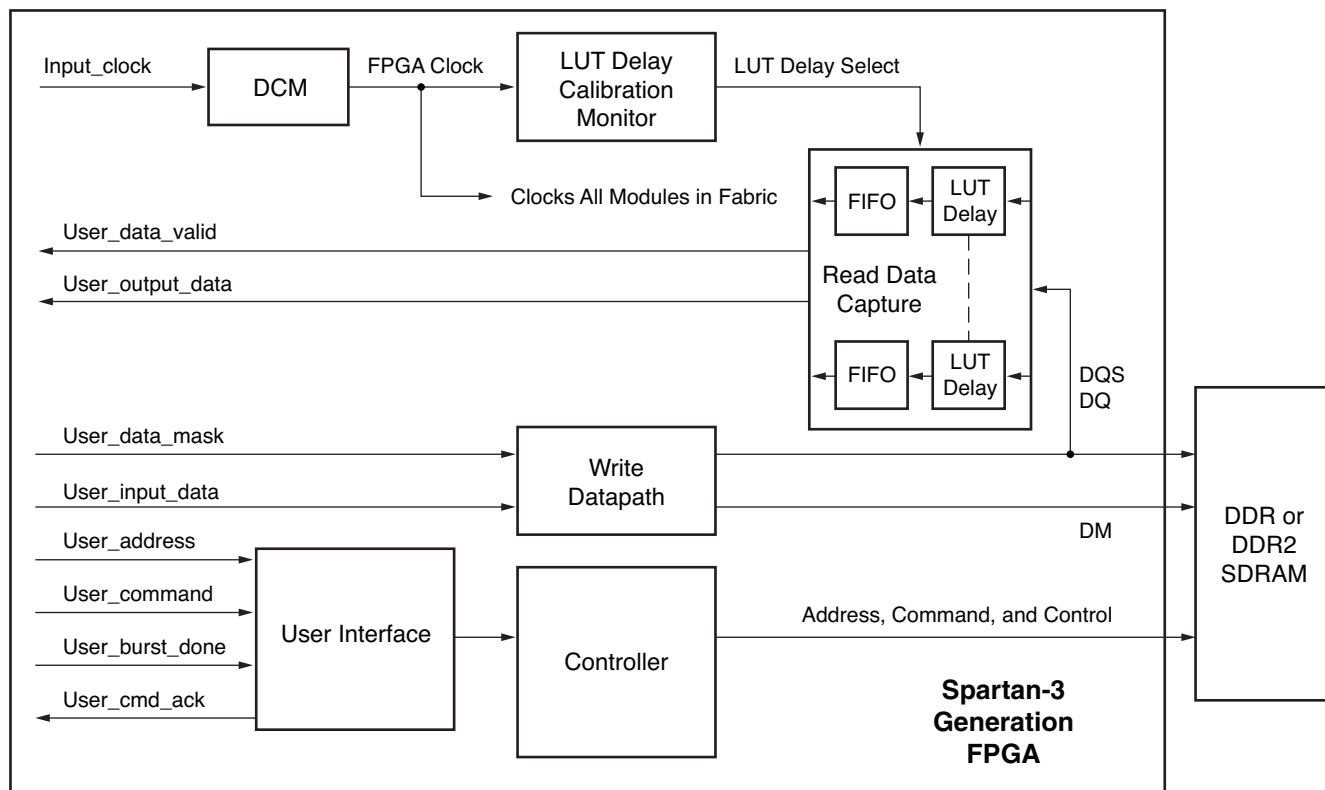
高于 400 Mb/s 速率的更宽总线使得芯片到芯片的接口愈益难以开发，因为需要更大的封装、更好的电源和接地 – 信号比率。Virtex-4 和 Virtex-5 FPGA 的开发使用了先进的稀疏锯齿形 (Sparse Chevron) 封装技术，能提供优良的信号 – 电源和接地引脚比率。每个 I/O 引脚周围都有足够的电源和接地引脚和板，以确保良好的屏蔽，使由同步交换输出 (SSO) 所造成的串扰噪音降到最低。

低成本存储器接口

今天，并不是所有的系统都在追求存储器接口的性能极限。当低成本是主要的决定因素，而且存储器的比特率达到每引脚 333 Mb/s 已经足够时，Spartan-3 系列 FPGA 配以 Xilinx 软件工具，就能提供一个易于实现、低成本的解决方案。

基于 FPGA 设计的存储器接口和控制器由三个基本构建模块组成：读写数据接口、存储器控制器状态机，以及将存储器接口设计桥接到 FPGA 设计的其余部分的用户界面 (图 3)。这些模块都在 FPGA 资源中实现，并由数字时钟管理器 (DCM) 的输出作为时钟来驱动。在 Spartan-3 系列实现中，DCM 也驱动查找表 (LUT) 延迟校准监视器 (一个确保读数据采集具有正确时序的逻辑块)。延迟校准电路用来选择基于 LUT 的延迟单元的数量，这些延迟单元则用于针对读数据对选通脉冲线 (DQS) 加以延迟。延

迟校准电路计算出与 DQS 延迟电路相同的一个电路的延迟。校准时会考虑所有延迟因素，包括所有组件和布线延迟。



WP260_03_011007

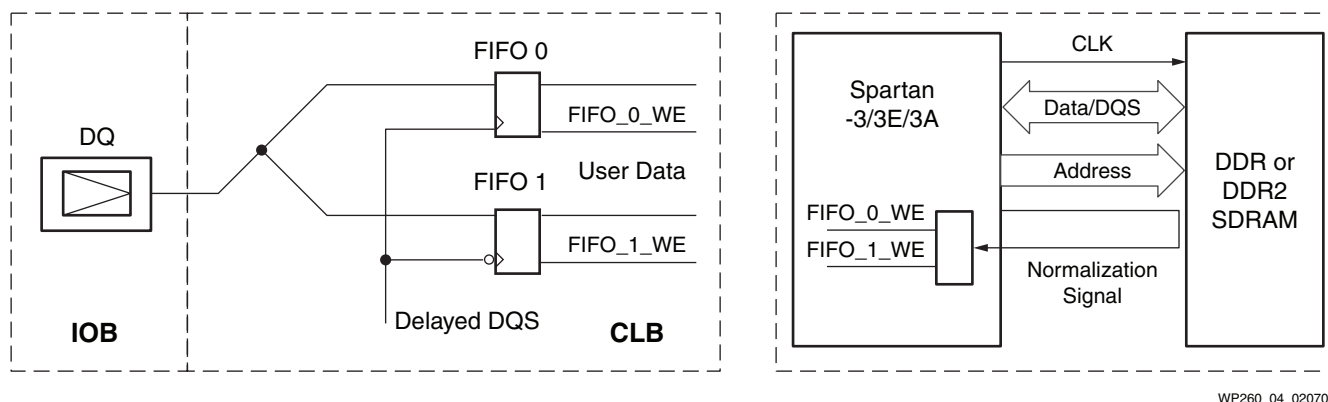
图 3: Spartan-3 系列 FPGA 的 DDR 和 DDR 2 SDRAM 接口实现

用户界面是一种握手型的界面。用户发出一条读或写命令，如果是写命令的话还包括地址和数据，而用户界面逻辑以 User_cmd-ack 信号回应，于是下一条命令又可发出。

在 Spartan-3 系列实现中，使用可配置逻辑块 (CLB) 中的 LUT 来实现读数据采集。在读事务过程中，DDR 或 DDR2 SDRAM 器件将读数据选通脉冲 (DQS) 及相关数据按照与读数据 (DQ) 边沿对齐的方式发送给 FPGA。在高频率运行的源同步接口中采集读数据是一项颇具挑战性的任务，因为数据在非自由运行 DQS 的每个边沿上都会改变。读数据采集的实现使用了一种基于 LUT 的 tap 延迟机制。DQS 时钟信号被适量延迟，使其放置后在读数据有效窗口中具有足够的余量，以在 FPGA 内被采集。

读数据的采集是在基于 LUT 的双端口分布式 RAM 中完成的（见图 4）。LUT RAM 被配置成一对 FIFO，每个数据位都被输入到上升边沿 (FIFO 0) 和下降边沿 (FIFO 1) 的

FIFO 中，如图 4 所示。这些深度为 16 个输入的 FIFO 异步运行，具有独立的读写端口。



WP260_04_020707

图 4: Spartan-3 系列 FPGA 读数据采集 FIFO 的实现

来自存储器的读数据写到经过延迟的 DQS 上升边沿的 FIFO_0 中，并写到经过延迟的 DQS 下降边沿的 FIFO_1 中。将读数据从 DQS 时钟域传输到存储器控制器时钟域，就是通过这些异步 FIFO 完成的。在存储器控制器的时钟域中，可以从 FIFO_0 和 FIFO_1 同时读出数据。FIFO 的读指针在 FPGA 的内部时钟域中生成。写使能信号 (FIFO_0_WE 和 FIFO_1_WE) 的生成通过 DQS 和一个外部回送 (亦即归一化) 信号完成。外部归一化信号作为输出传送至输入 / 输出模块 (IOB)，然后通过输入缓冲器作为输入取出。这种技术可补偿 FPGA 与存储器器件之间的 IOB、器件和迹线延迟。从 FPGA 输入管脚发出的归一化信号在进入 LUT 延迟电路之前使用与 DQS 相似的布线资源，以与布线延迟相匹配。环路之迹线延迟应为发送给存储器的时钟和 DQS 之迹线延迟的总和 (图 4)。

写数据命令和时序由写数据接口生成并控制。写数据接口使用 IOB 触发器和 DCM 的 90 度、180 度和 270 度输出，发送按照 DDR 和 DDR2 SDRAM 的时序要求与命令位和数据位正确对齐的 DQS。

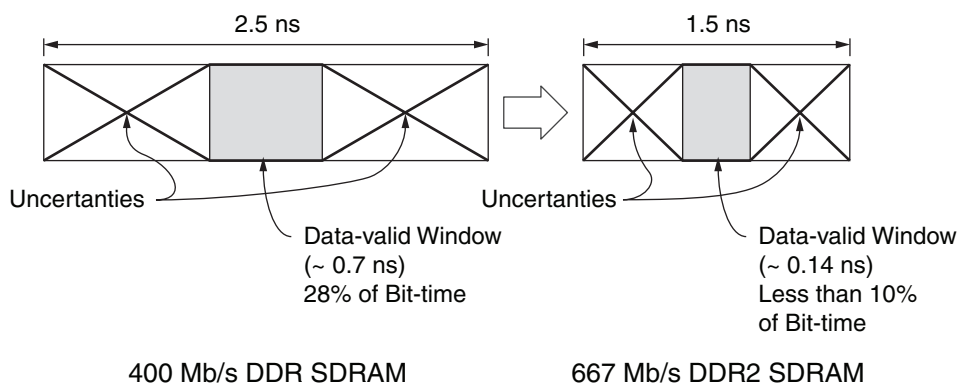
用于 Spartan-3 系列 FPGA 的一种 DDR 和 DDR2 SDRAM 存储器接口实现已通过硬件进行了充分验证。一个利用 [Spartan-3A 入门套件](#) 的低成本 DDR2 SDRAM 参考设计示例已完成。此设计为板上 16 位宽 DDR2SDRAM 存储器器件而开发，并使用了 XC3S700A-FG484。此参考设计仅利用了 Spartan-3A FPGA 器件可用资源的一小部分：13% 的 IOB、9% 的逻辑 Slice、16% 的 BUFG MUX 和八个 DCM 中的一个。这一实现为其余部分 FPGA 设计所需的其他功能留下了可用资源。

使用存储器接口生成器 (MIG) 软件工具 (本白皮书后面的部分有说明)，设计人员可以很容易地定制 Spartan-3 系列的存储器接口设计，以适合自己的应用。

高性能存储器接口

随着数据速率的提高，满足接口时序方面的要求变得愈益困难了。与写入存储器相比，从存储器中读数据时，存储器接口时钟控制方面的要求通常更难满足。追求更高数据速率的趋势使得设计人员面临巨大挑战，因为数据有效窗口（此为数据周期内的一段时间，其间可获得可靠的读数据）比数据周期本身缩小得快。造成这种情况的原因是，影响有效数据窗口尺寸大小的系统和器件性能参数具有种种不确定性，它们缩小的速率与数据周期不同。

如果比较一下运行速度为 400 Mb/s 的 DDR SDRAM 数据有效窗口和运行速度为 667 Mb/s 的 DDR2 存储器技术，这种情况就一目了然了。数据周期为 2.5 ns 的 DDR 器件拥有 0.7 ns 的数据有效窗口，而数据周期为 1.5 ns 的 DDR2 器件仅有 0.14 ns 的数据有效窗口（图 5）。



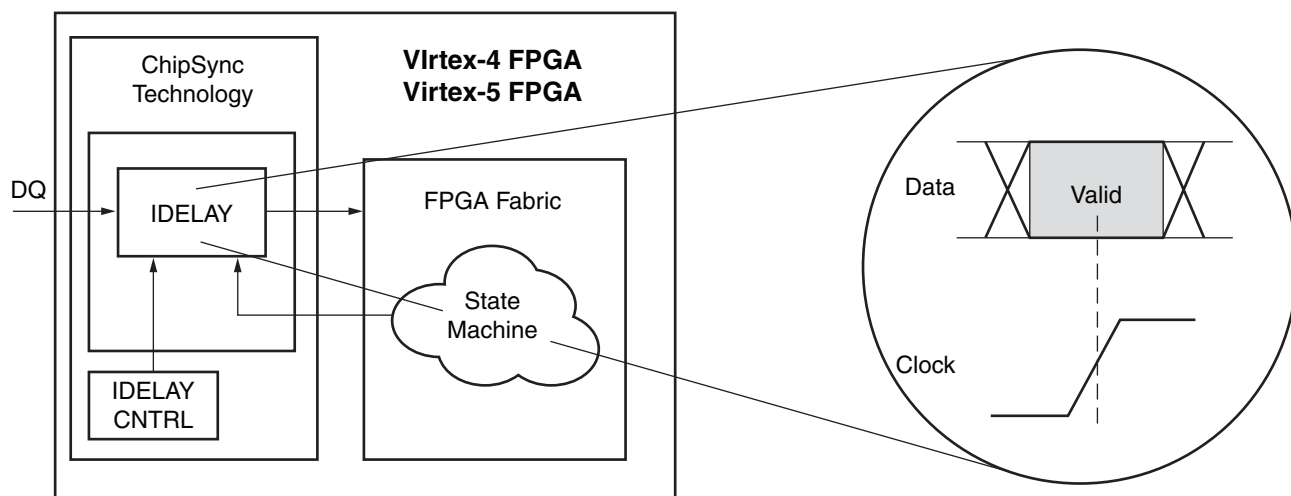
WP260_05_020507

图 5: 缩小的数据有效窗口

显然，数据有效窗口的加速减损给 FPGA 设计人员带来了一堆全新的设计挑战，要创建和维护可靠的存储器接口性能，就得采用更有效的方法。

正如 Spartan-3 系列 FPGA 中所实现的那样，使用读数据 DQS 可以把读数据采集到可配置逻辑块 (CLB) 中，但是使用 LUT 把 DQS 或时钟与数据有效窗口中心对齐时，所用的延迟 tap 却很粗糙。CLB 中实现的延迟 tap 具有大约几百微微秒 (ps) 的分辨率，然而，对于超过 400 Mb/s 的数据速率的读取采集时序，所需的分辨率要比基于 CLB 的 tap 高一个数量级。Virtex-4 和 Virtex-5 FPGA 采用 I/O 模块中的专用延迟和时钟资源（称为 ChipSync™ 技术）来解决这一难题。内置到每个 I/O 中的 ChipSync

模块都含有一串延迟单元（tap 延迟），在 Virtex-4 中称为 IDELAY，而在 Virtex-5 FPGA 中称为 IODELAY，其分辨率为 75 ps（见图 6）。



WP260_06_020507

图 6: Virtex-4 和 Virtex-5 FPGA 采用 75 ps tap 延迟进行时钟 - 数据对齐

此实现的架构基于几个构建模块。用户界面负责把存储器控制器和物理层接口桥接到其余 FPGA 设计，它使用 FIFO 架构（图 7）。FIFO 有三套：命令 / 地址 FIFO、写 FIFO、读 FIFO。这些 FIFO 保存着命令、地址、写数据和读数据。主要的控制器模块

控制读、写和刷新操作。其他两个逻辑模块执行读操作的时钟 – 数据对齐：初始化控制器和校准逻辑。

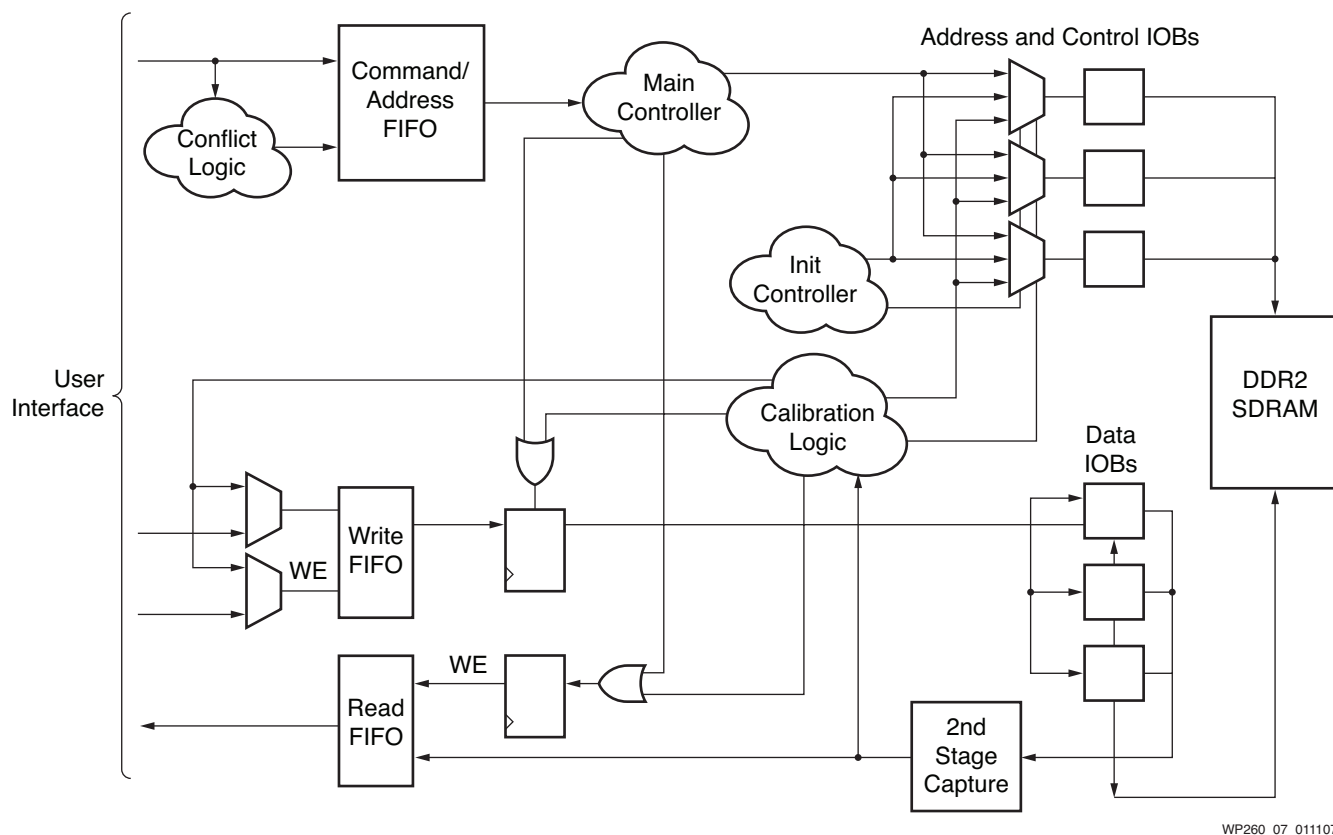


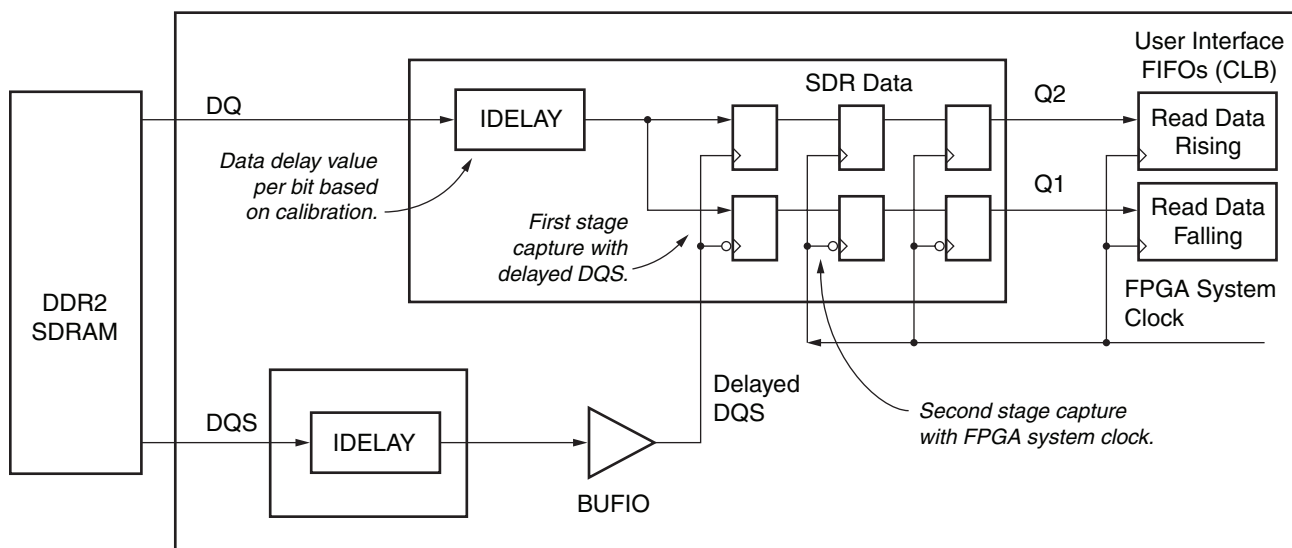
图 7: Virtex-5 FPGA 存储器接口架构

用于地址、控制 and 数据的物理层接口在 I/O 模块 (IOB) 中实现。读数据在锁存器的第二级 (也是 IOB 的一部分) 重新采集。

Virtex-4 和 Virtex-5 FPGA 存储器接口参考设计支持两种读数据采集技术。Virtex-4 FPGA 支持的直接时钟技术延迟了读数据，因而使用 IOB 的输入 DDR 触发器中的系统时钟可直接寄存读数据。为将 FPGA 时钟对齐到最佳状态，对每个读数据位都会单独进行校验。这种技术为高达 240 MHz 的时钟速率提供了足够的性能。

第二种技术称为基于 DQS 的技术。此技术用于更高的时钟速率，Virtex-4 和 Virtex-5 FPGA 二者都支持此技术。它使用存储器 DQS 来采集相应的读数据，数据被此 DQS 的延迟信号 (通过一个局部 I/O 时钟缓冲器 (BUFIO) 分配) 寄存。此数据然后在触发

器的第二级与系统的时钟域同步。IOB 中的输入串行器 / 解串器功能用于读数据采集；第一对触发器把数据从延迟的 DQS 域中传输到系统的时钟域（图 8）。



WP260_08_020507

图 8: 使用基于 DQS 技术的 Virtex-5 读数据采集

两种技术都涉及到 tap 延迟 (IDELAY) 单元的应用，在由校验逻辑实现的校验程序中，这些延迟单元会有所变化。在系统初始化期间，会执行此校准程序以设置 DQS、数据和系统时钟之间的最佳相位。这样做的目的是使时序余量最大化。校准会消除任何由过程相关的延迟所导致的不确定性，从而补偿对于任何一块电路板都不变的那些通路延迟成分。这些成分包括 PCB 迹线延迟、封装延迟和过程相关的传播延迟成分（存储器和 FPGA 中都有），以及 FPGA I/O 模块中采集触发器的建立 / 保持时间。有的延迟是由系统初始化阶段的过程、电压和温度所决定的，校准即负责解决这些延迟的变动。

在校准过程中会增加 DQS 和数据的延迟 tap 以执行边沿检测，检测方式是通过连续从存储器中读回数据并对预编写培训模式或存储器 DQS 本身进行采样，直到确定数据选通脉冲 (DQS) 的前沿或前后两沿。之后数据或 DQS 的 tap 数被设定，以提供最大的时序余量。对“基于 DQS”的采集而言，DQS 和数据可以有不同的 tap 延迟值，因为同步实质上分为两个阶段：一个先在 DQS 域中采集数据，另一个把此数据传输到系统时钟域。

在更高的时钟频率下，“基于 DQS”的采集方法就变得十分必要，其二阶段方法能提供更好的采集时序余量，因为 DDR 时序的不确定性主要限于 IOB 中触发器的第一级。此外，因为使用 DQS 来寄存数据，与时钟 - 数据 (Tac) 变化相比较，DQS - 数据变化的时序不确定性要小一些。例如，对于 DDR2 而言，这些不确定性就是由器件的 tDQSQ 和 tQHS 参数给出的。

正如 Spartan-3 系列 FPGA 中所实现的那样，Virtex-4 和 Virtex-5 FPGA 的写时序由 DCM 所支持，此 DCM 生成系统时钟的两相输出。存储器的 DQS 由一个输出 DDR 寄存器来输出，这个 DDR 寄存器由系统时钟的同相时钟驱动。写数据则由超前系统时钟 90° 的一个 DCM 时钟输出进行时钟控制。这种技术确保了在 FPGA 的输出部分，DQS 与写操作的数据中心对齐。

此设计的其他方面包括整体控制器状态机的逻辑生成和用户接口。为了使设计人员更容易完成整个设计，Xilinx 开发了存储器接口生成器 (MIG) 工具。

控制器设计和集成

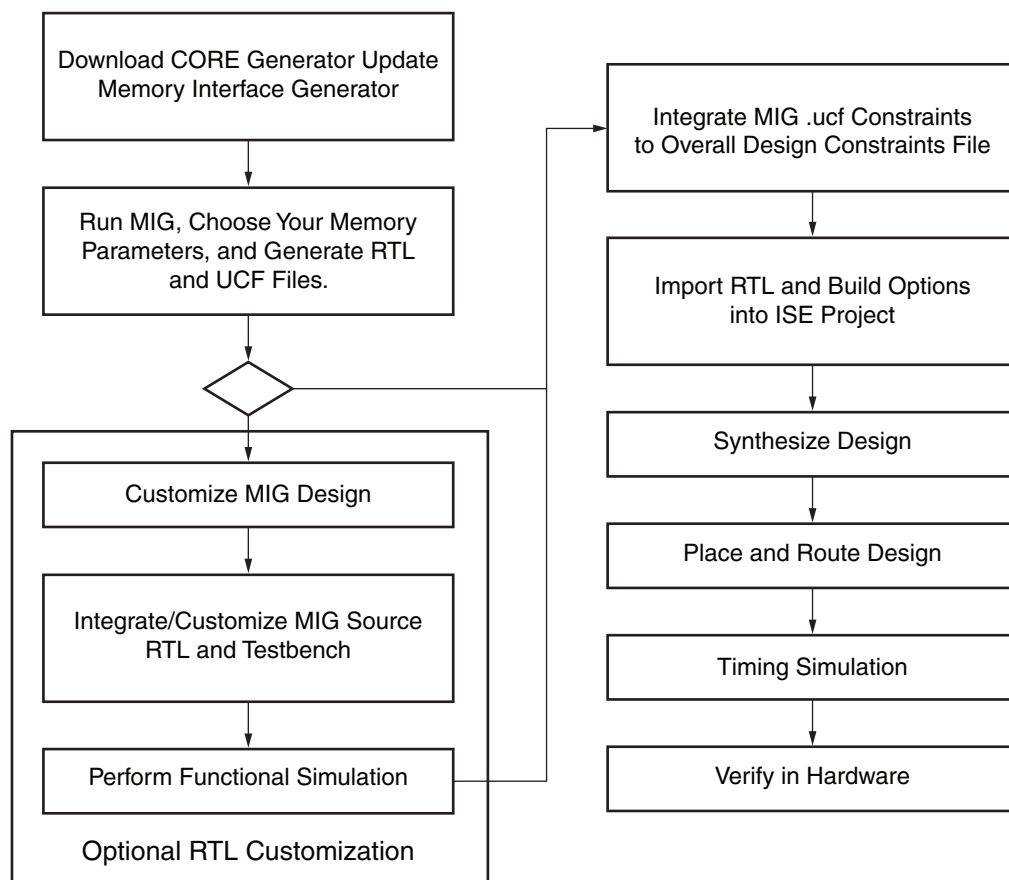
创建存储器控制器是一项极其复杂、精细的任务，FPGA 设计人员要解决面临的一道难题，就需要 FPGA 随附的工具提供更新水平的集成支持。

为设计的完整性起见，对包括存储器控制器状态机在内的所有构建模块加以集成，十分必要。控制器状态机因存储器架构和系统参数的不同而异。状态机编码也可以很复杂，它是多个变量的函数，例如：

- 架构（DDR、DDR2、QDR II、RLDRAM 等）
- 组 (bank) 数（存储器器件之外或之内）
- 数据总线宽度
- 存储器器件的宽度和深度
- 组和行存取算法

最后，数据与 DQS 比 (DQ/DQS) 这类参数会进一步增加设计的复杂性。控制器状态机必须按正确顺序发出命令，同时还要考虑存储器器件的时序要求。

使用 MIG 软件工具可生成完整的设计。该工具作为 CORE Generator™ 参考设计和知识产权套件的一部分，可从 Xilinx 免费获取。MIG 设计流程（图 9）与传统 FPGA 的设计流程非常相似。MIG 工具的优点是不必再为物理层接口或存储器控制器从头生成 RTL 代码。



WP260_08_020707

图 9: MIG 设计流程

MIG 图形用户界面 (GUI) 可用于设置系统和存储器参数 (图 10)。例如, 选定 FPGA 器件、封装方式和速度级别之后, 设计人员可选择存储器架构, 并挑选实际存储器器件或 DIMM。同是这一个 GUI, 还可用于选择总线宽度和时钟频率。同时, 对于某些

FPGA 器件，它还提供拥有多于一个控制器的选项，以适应多个存储器总线接口的要求。另外一些选项可提供对时钟控制方法、CAS 延迟、突发长度和引脚分配的控制。

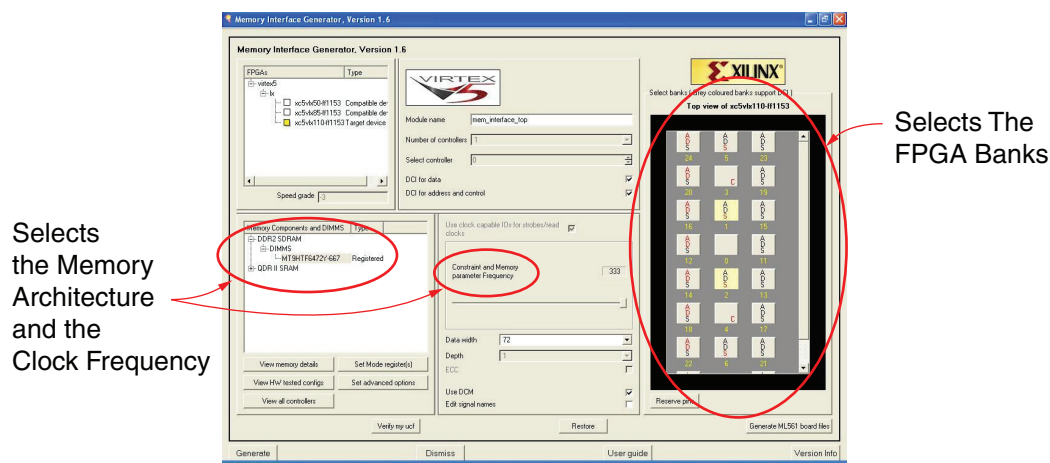


图 10: MIG 图形用户界面 (GUI)

用不了一分钟，MIG 工具即可生成 RTL 和 UCF 文件，前者是 HDL 代码文件，后者是约束文件。这些文件是用一个经过硬件验证的参考设计库生成的，并根据用户输入进行了修改。

设计人员享有完全的灵活性，可进一步修改 RTL 代码。与提供“黑匣子”实现方法的其他解决方案不同，此设计中的代码未加密，设计人员完全可以对设计进行任意修改和进一步定制。输出文件按模块分类，这些模块被应用于此设计的不同构建模块：用户界面、物理层、控制器状态机等等。因此，设计人员可选择对控制组存取算法的状态机进行自定义。由 MIG 工具生成的 Virtex-4 和 Virtex-5 DDR2 的组存取算法彼此不同。Virtex-5 设计采用一种最近最少使用 (LRU) 算法，使多达四组中的一行总是打开，以缩减因打开 / 关闭行而造成的开销。如果需要在一个新组中打开一行，控制器会关闭最近最少使用组中的行，并在新组中打开一行。而在 Virtex-4 控制器实现中，任何时候只有单个组有一个打开的行。每个应用都可能需要有自己的存取算法来最大化吞吐量，设计人员可通过改变 RTL 代码来修改算法，以更加适合其应用的访问模式。

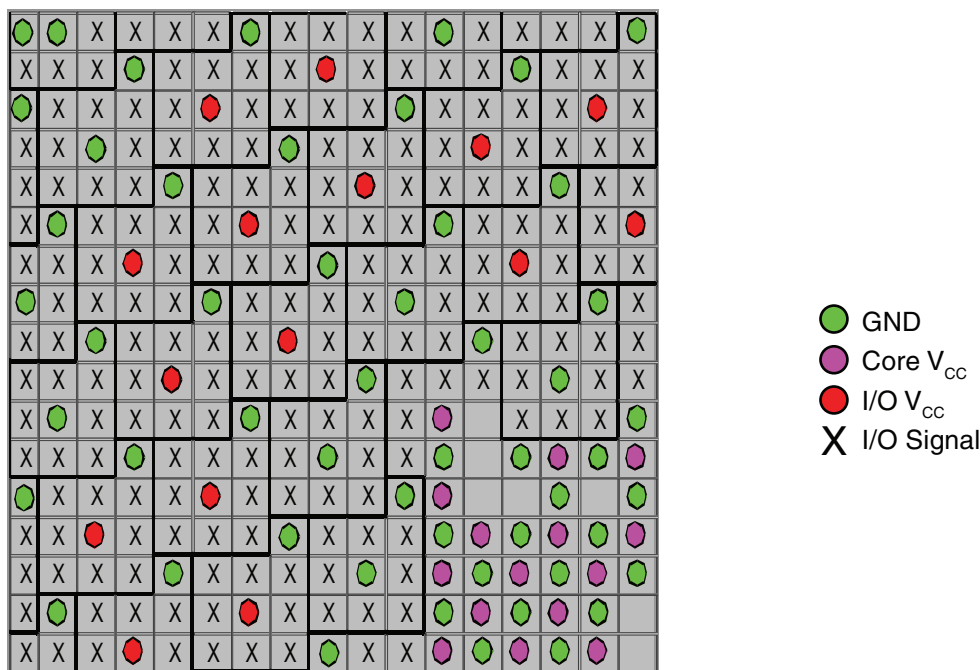
修改可选代码之后，设计人员可再次进行仿真，以验证整体设计的功能。MIG 工具还可生成具有存储器校验功能的可综合测试平台。该测试平台是一个设计示例，用于 Xilinx 基础设计的功能仿真和硬件验证。测试平台向存储控制器发出一系列写和读回命令。它还可以用作模板，来生成自定义的测试平台。

设计的最后阶段是把 MIG 文件导入 ISE 项目，将它们与其余 FPGA 设计文件合并，然后进行综合、布局和布线，必要时还运行其他时序仿真，并最终进行硬件验证。MIG 软件工具还会生成一个批处理文件，包括相应的综合、映射以及布局和布线选项，以帮助优化生成最终的 bit 文件。

高性能系统设计

实现高性能存储器接口远远不止实现 FPGA 片上设计，它需要解决一系列芯片到芯片的难题，例如对信号完整性的要求和电路板设计方面的挑战。

信号完整性的挑战在于控制串扰、地弹、振铃、噪声容限、阻抗匹配和去耦合，从而确保可靠的信号有效窗口。Virtex-4 和 Virtex-5 FPGA 所采用的列式架构能使 I/O、时钟、电源和接地引脚部署在芯片的任何位置，而不光是沿着外围排列。此架构缓解了与 I/O 和阵列依赖性、电源和接地分布、硬 IP 扩展有关的问题。此外，Virtex-4 和 Virtex-5 FPGA 中所使用的稀疏锯齿形封装技术能对整个封装中的电源和接地引脚进行均匀分配。这些封装提供了更好的抗串扰能力，使高性能设计中的信号完整性得以改善。图 11 所示为 Virtex-5 FPGA 封装管脚。圆点表示电源和接地引脚，叉号表示用户可用的引脚；在这样的布局中，I/O 信号由足够的电源和接地引脚环绕，能确保有效屏蔽 SSO 噪音。



WP60_11_020707

图 11: 采用分布式电源和接地引脚的 Virtex-5 FPGA 封装

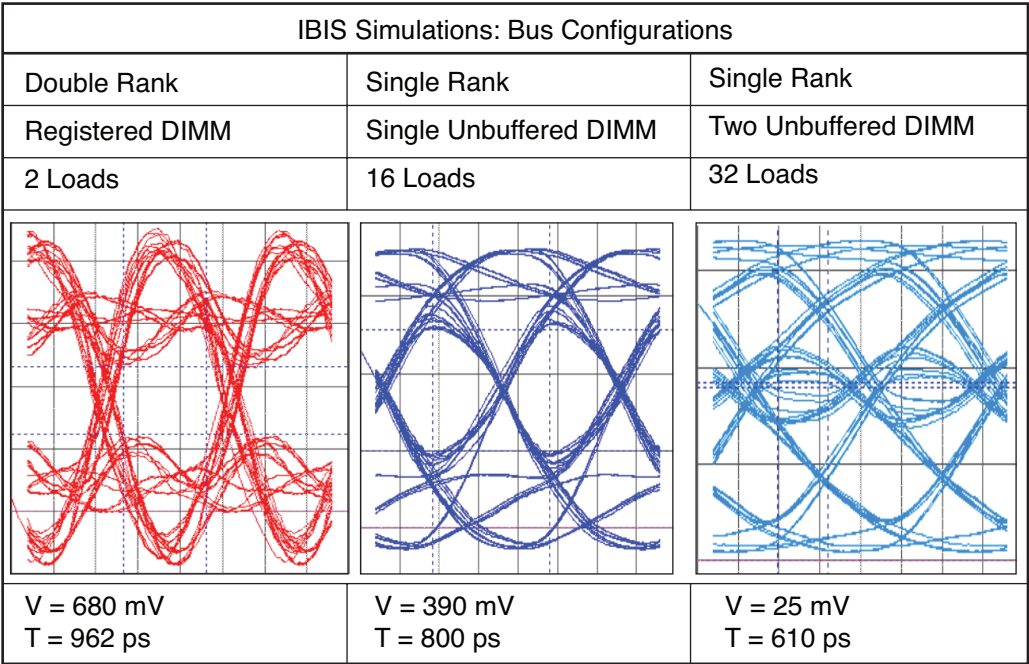
对于高性能存储器系统来说，增加数据速率并不总能满足需求；要达到希望的带宽，就需要有更宽的数据总线。今天，144 或 288 位的接口已经随处可见。多位同时切换可导致信号完整性问题。对 SSO 的限制由器件供应商标明，它代表器件中用户可为每组同时使用的信号引脚的数量。凭借稀疏锯齿形封装技术良好的 SSO 噪音屏蔽优势和同质的 I/O 结构，宽数据总线接口完全可能实现。

表 1 列出了 Virtex-5 LX 器件和满足 600 Mb/s 数据速率下的 SSO 需求的最大数据总线宽度。

表 1: Virtex-5 LX 器件满足 600 Mb/s 数据速率下 SSO 需求的最大数据总线宽度

Virtex-5 器件	封装	最大数据宽度
LX30、LX50	FF324	108
LX30、LX50、LX85、LX110	FF676	243
LX50、LX85、LX110	FF1153	432
LX110、LX220、LX330	FF1760	648

设计大容量或密集型存储器系统的另一个挑战是容量负载。高性能存储器系统可能需要由地址和命令信号共用的一条总线驱动的多存储器器件。大容量无缓冲 DIMM 接口就是一个例子。如果每个单列 DIMM 拥有 18 个组件，那么包含两个 72 位无缓冲 DIMM 的接口可以在地址和命令总线上拥有多达 36 个接收器。由 JEDEC 标准推荐，并在通用系统中常见的最大负载是两个无缓冲 DIMM。总线上所产生的容量负载会极其庞大，导致信号边沿上升和下降需要多于一个时钟周期，从而使存储器器件的建立和保持出错。图 12 所示为 IBIS 仿真所提供的眼图，使用的是不同配置：一个寄存 DIMM、一个无缓冲 DIMM 和两个单列无缓冲 DIMM。容量负载的范围从使用寄存 DIMM 时的 2 个接收器到使用无缓冲 DIMM 时的 36 个接收器不等。



WP60_13_011107

图 12: 利用对 DIMM 接口的 IBIS 仿真获得的地址和命令信号眼张开度，数据总线上具有不同负载

这些眼图清楚地显示了地址总线的容量负载效果；寄存 DIMM 提供地址和命令总线上一个打得很开的有效窗口。一个 DIMM 的眼张开度在 267 MHz 下仍然不错。然而，当负载为 32 时，地址和命令信号有效窗口便大为缩小，而传统的实现方法已不足以可靠地与两个无缓冲 DIMM 接口。

这个简单的测试示例说明负载会导致边沿明显变慢的同时，眼图在更高的频率下闭上。对于总线负载不可减少的系统，降低操作的时钟频率不失为使信号完整性维持在可接受水平上的一种方法。然而，还有其它方法可以在不降低时钟频率的情况下解决容量负载问题：

- 在可以往接口添加一个时钟周期的延迟的应用中，使用寄存 DIMM 可以是一个不错的选择。这些 DIMM 使用一个寄存器来缓冲地址和命令一类信号，从而降低容量负载。
- 使用基于在地址和命令信号上采用两个时钟周期（称为 2T 时序）的设计技术，地址和命令信号可以用系统时钟频率的一半发送。

控制好存储器系统的成本和达到要求的性能一样，也是一个很大的挑战。降低电路板设计的复杂性并减少材料费用的一个方法是使用片上终端而不是电路板上的电阻器。Virtex-4 和 Virtex-5 系列 FPGA 提供一种称为“数控阻抗 (DCI)”的功能，在设计中实现该功能可减少电路板上的电阻器数量。MIG 工具具有一个内置选项，允许设计人员在实现存储器接口设计时包含针对地址、控制或数据总线的上述功能。此时要考虑的一个权衡因素是当终端在片上实现时，片上与片外功耗孰优孰劣。

存储器接口的开发板

对参考设计进行硬件验证是确保解决方案严密可靠的重要最终步骤。Xilinx 已经验证了 Spartan-3 系列、Virtex-4 和 Virtex-5 FPGA 的存储器接口设计。表 2 所示为对于每一个开发板，所支持的存储器接口。

表 2: 存储器接口的开发电路板

Xilinx FPGA	Spartan-3	Spartan-3E	Spartan-3A	Virtex-4	Virtex-5
开发 电路板	SL-361	入门套件	入门套件	ML-461	ML-561
支持的 存储器 接口	DDR	DDR	DDR2	DDR DDR2 QDR-II RLDRAM II	DDR DDR2 QDR-II RLDRAM II

开发电路板的范围涵盖从低成本 Spartan-3 系列 FPGA 实现到 Virtex-4 和 Virtex-5 FPGA 系列器件所提供的高性能解决方案。

结论

有了合适的 FPGA、软件工具和开发电路板这样的利器，使用 667 Mb/s DDR2 SDRAM 进行存储器接口控制器设计便成为一个既快速又流畅的过程，无论是低成本应用还是高性能设计，都可以得心应手地完成。

欲了解这些存储器接口解决方案的详情，请参见下列网址：

www.xilinx.com/cn/memory。

修订历史

下表说明此文档的修订历史。

日期	版本	修订
2007 年 2 月 16 日	1.0	Xilinx 最初版本。