



## Seal ( 海豹 ) SA5Z-30 FPGA

# DDR2 控制器用户指南

西安智多晶微电子有限公司 XIAN Intelligence Silicon Technology  
西安市高新区科技二路 72 号西安软件园西岳阁 102 室 邮编 710075  
<http://www.isilicotech.com>



## 文档修订历史

| 日期      | 版本  | 修订内容                          |
|---------|-----|-------------------------------|
| 2021.09 | 1.0 | 首次发布。                         |
| 2022.06 | 1.1 | 增加 3.2 章节 DDR2 控制器 IP 核的使用方法。 |

西安智多晶微电子有限公司  
SA5Z-30 FPGA DDR2 控制器用户指南



# 目录

|                             |    |
|-----------------------------|----|
| 1. 概述.....                  | 2  |
| 2. 设计说明.....                | 5  |
| 2.1. 结构框图.....              | 5  |
| 2.2. 各模块功能简述.....           | 6  |
| 2.3. 接口说明.....              | 10 |
| 2.4. 用户端操作说明.....           | 17 |
| 2.5. 用户端地址映射说明.....         | 22 |
| 2.6. 模式寄存器配置说明.....         | 23 |
| 3. DDR2 控制器使用说明.....        | 25 |
| 3.1 DDR2 控制器系统介绍.....       | 25 |
| 3.2 DDR2 控制器 IP 核的使用方法..... | 26 |



## 1. 概述

SA5Z-30 DDR2 Controller 硬核支持连接具有标准 DDR2 接口的存储设备或模块，同时以异步 FIFO 作为用户端接口。用户可以通过 bit-stream 方便的配置 DDR2 Controller 的工作模式和时序参数，从而可以快速开发出适用于不同规格型号存储器的设计。

SA5Z-30 DDR2 Controller 具有以下特征：

- ◆ 支持存储器数据位宽为 8bits 和 16bits；
- ◆ 支持 x8 和 x16 两种存储器规格；
- ◆ 支持 Unbuffered DIMM 和 Registered DIMM 两种模式；
- ◆ 支持通过 fuse 对控制器内部以下时序参数的配置：
  - tRAS、tRC、tRRD、tRCD、tRP、tFAW、tCKE、tRFC、tMRD、tWR、tWTR、tRTP、tCKESR、tZQOPER、tZQS、tPD、tXPDLL(tXARDs)、tXPR、tMOD、tZQINIT 和 tREFI；
  - 用户可以根据使用的存储器的 speed bin 来选择对应的预设参数值 或者完全自定义设置。
- ◆ 支持通过 fuse 配置 DDR2 SDRAM 内部模式寄存器(Mode Register)的初始值；
- ◆ 用户的命令和数据将通过两组 FIFO 传递给 DDR2 Controller，每一组 FIFO 的构成如下所示：
  - 1 个数据位宽为 43，深度为 16 的 cmd/addr FIFO，其中 44bit 数据的 bit[29:0]为 addr，bit[30]为 DDR3 模式使用的 on-the-fly burst length，bit[35:31]为 cmd burst count，bit[39:36]为 cmd，bit[42:40]为 command ID；
  - 1 个数据位宽为 73，深度为 64 的 Write Data FIFO，其中 73bit 数据的 bit[63:0]为 write data，bit[71:64]为 write data mask，bit[72]为 write data last 指示信号；
  - 1 个数据位宽为 68，深度为 64 的 Read Data FIFO，其中 68bit 数据的 bit[63:0]为 read data，bit[64]为 read data last 指示信号，bit[68:65]为 command ID；
  - 每个 FIFO 都将输出溢出指示信号、空满指示信号给用户，同时 Write Data FIFO 和 Read Data FIFO 还将输出 space 信号来告诉用户 FIFO 的剩余空间。
- ◆ DDR2 Controller 将以轮询 ( Round Robin ) 的方式，依次读取两组 FIFO 中保存的命令；
- ◆ DDR2 Controller 支持用户命令如表 1-1 所示：

表 1-1 DDR2 Controller 用户命令编码

| Command                   | Mnemonic  | cmd[3:0] |
|---------------------------|-----------|----------|
| Read                      | READ      | 0001     |
| Write                     | WRITE     | 0010     |
| Read with Auto Precharge  | READA     | 0011     |
| Write with Auto Precharge | WRITEA    | 0100     |
| Power down Entry          | PDOWN_ENT | 0101     |



|                                     |              |      |
|-------------------------------------|--------------|------|
| Load Mode Register                  | LOAD_MR      | 0110 |
| Self Refresh Entry                  | SEL_REF_ENT  | 1000 |
| Self Refresh Exit                   | SEL_REF_EXIT | 1001 |
| Power down Exit                     | PDOWN_EXIT   | 1011 |
| ZQ Calibration Long<br>(Only DDR3)  | ZQ_LNG       | 1100 |
| ZQ Calibration Short<br>(Only DDR3) | ZQ_SHRT      | 1101 |

- ◆ 用户命令输入支持 Burst 模式 , Burst Size 的取值为 1~32 ;
- ◆ 用户输入的访问地址 addr 支持 row\_bank\_col 和 bank\_row\_col 两种地址组合方式 ;
- ◆ 支持对 DDR2 SDRAM 自动进行初始化操作 ( 复位释放后自动进行 ) ;
- ◆ 支持在初始化结束后对 DQSW ( 即 Write DQS ) 和 DQSI ( 即 Read DQS ) 信号进行校准 :
  - 支持对每个 DQSI 信号进行 read training 操作 , 以滤除 DQSI 上的毛刺 ;
  - 支持对每个 DQSI 信号的 90° 相移进行实时校准 : 当不对 DDR 存储器进行读写操作时 , 会自动根据当前电压温度环境 , 计算新的 delay code ;
  - 支持对每个 DQSI 信号相对于 DQ 信号的延迟进行自动校准 , 从而保证对读数据的采样正确 ;
  - 支持对每个 DQSW 信号相对于 CK 信号的延迟进行校准 ( 通过 fuse 实现 ) ;
- ◆ 支持 DQLS 和 DQSU 两组读数据的自动对齐 ,DQLS 和 DQSU 之间的最大延迟为 1.75 个时钟周期 ;
- ◆ 支持自动对存储器进行刷新操作 ( 刷新间隔可配置 ) , 或用户自主发起刷新操作 ( 两种模式二选一 ) ;
  - 支持闲时自动刷新功能 : 当命令流水闲置超过 1~2tRFC 时间时 , 控制器将自动对存储器发送 MC1\_AR\_BURST\_EN 个 Refresh 命令 ;
- ◆ 支持智能 Bank 管理功能 :
  - 在控制器接收到用户的读写命令后 , 会根据 Bank 当前状态自动发送 Precharge 或 Active 命令 ;
  - 支持最多 8 个 Bank 同时处于打开状态。
- ◆ 支持 DDR2 SDRAM 的 DM 接口 ( 即支持对写数据的按字节 mask 写入 ) ;
- ◆ 支持访问地址跨越 page 边界 ( 即在 Column 地址写满后 , 控制器会自动切换 row/bank 地址 ) ;
- ◆ 在控制器内部使用了四级流水 , 因而对存储器具有很高的数据吞吐效率 :
  - 例如 : 若存储器的 Bank0 Row1 、 Bank3 Row8 、 Bank5 Row4 已经处于打开状态 , 那么用户的连续多个跨 Bank 读命令 , 将对应存储器 DQS 和 DQ 总线上的无缝读操作 ( seamless read operation ) 。



除了上述共同特征外，在 DDR2 和 DDR3 模式下，控制器还分别具有以下特征：

DDR2 模式下：

- ◆ 支持符合 DDR2 工业标准接口(JESD79-2F)的器件；
- ◆ DDR2 1.8V；
- ◆ 支持以下 DDR2 速率：  
400Mbps(4-4-4/3-3-3)/533Mbps(4-4-4/3-3-3)/667Mbps(5-5-5/4-4-4)/800Mbps(6-6-6/5-5-5/4-4-4)；
- ◆ 支持 Burst Length 为 4 或 8；
- ◆ 支持 CAS latency 设置为 3、4、5、6 ( nCK ) ；
- ◆ 支持所有的 JEDEC 标准 DDR2 命令；
- ◆ 不支持 OCD 功能；
- ◆ 支持 ODT 功能；

## 2. 设计说明

### 2.1. 结构框图

如下图所示为 DDR2 Controller 的结构框图：

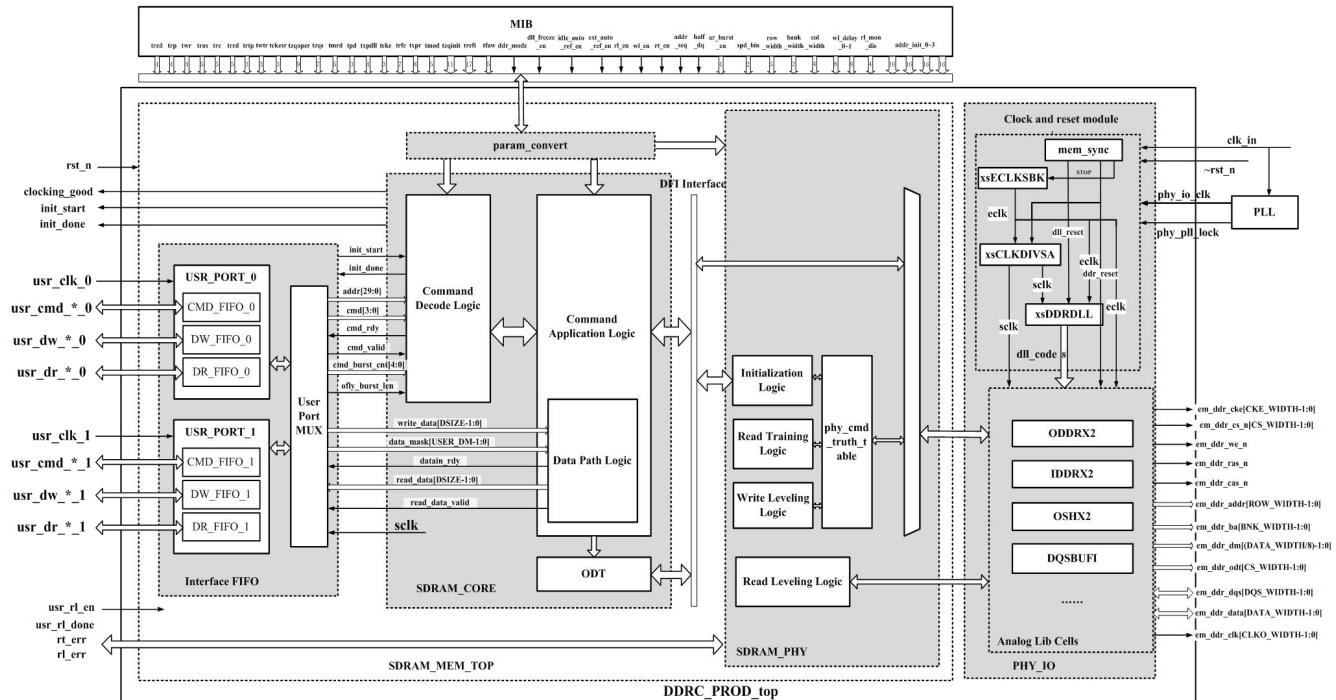


图 2-1 DDR2 Controller 结构框图

图 2-1 中 DDRC\_PROD\_top 是整个 DDR2 Controller 顶层模块，其中 SDRAM\_MEM\_TOP 模块为 DDR2 Controller 的数字逻辑部分。PHY\_IO 模块是 DDR2 Controller 的模拟电路部分。DDR2 Controller 的时序参数和功能参数，需通过图中的 MIB 部分进行配置。

DDR2 Controller 可以采用两种时钟域转换模式： X2 Gear 模式和 X4 Gear 模式。在 Seal30K 中采用的是 X2 GEAR 模式。

举例说明 X2 Gear 模式和 X4 Gear 模式下各模块的工作频率：

在 X2 Gear 模式下，对于输入时钟为 400MHz 的 DDR2 Memory( 读写频率为 800Mbps )，SDRAM\_MEM\_TOP 模块工作在 200MHz 的 SCLK 时钟下，PHY\_IO 中的 I/O 逻辑工作在 400MHz 的 ECLK 时钟下。用户端口处的数据位宽是存储器端口数据位宽的 4 倍。例如，存储器侧的 16-bits DQ，要对应用户侧的 64-bits 写总线和 64-bits 读总线。

在 X4 Gear 模式下，对于输入时钟为 400MHz 的 DDR2 Memory( 读写频率为 800Mbps )，SDRAM\_MEM\_TOP 模块工作在 100MHz 的 SCLK 时钟下，PHY\_IO 中的 I/O 逻辑工作在 400MHz 的 ECLK 时钟下。用户端口的数据位宽是存储器端口数据位宽的 8 倍。例如，存储器侧的 16-bits DQ，要对应用户侧的 128-bits 写总线和 128-bits 读总线。



## 2.2. 各模块功能简述

SDRAM\_MEM\_TOP 模块可分为四部分：Interface FIFO 模块、Parameter Convert 模块、SDRAM\_CORE 模块和 SDRAM\_PHY 模块。SDRAM\_CORE 模块和 SDRAM\_PHY 模块两个模块之间使用标准的 DDR PHY 接口（DFI 接口）。Interface FIFO 模块中包含了两组异步 FIFO 和路径仲裁逻辑。SDRAM\_CORE 模块中包含 Command Decode Logic ( CDL ) 模块、Command Application Logic( CAL )模块和 ODT 模块。SDRAM\_PHY 模块中包含 Initialization Logic、Read Training Logic 和 Write Leveling Logic 等子模块。

### 2.2.1. Interface FIFO

本 DDR2 控制器设计中包含了两组用户接口，即 USR\_PORT\_0 和 USR\_PORT\_1。每组用户接口均由以下 3 个异步 FIFO 构成：

- 1 个数据位宽为 43，深度为 16 的 cmd/addr FIFO，其中 43bit 数据的 bit[29:0] 为 addr，addr 的各 bit 位的含义参见 2.6.1 节；bit[30] 为 DDR3 模式使用的 on-the-fly burst length，bit[35:31] 为 cmd burst count，bit[39:36] 为 cmd，bit[42:40] 为 command ID；
- 1 个数据位宽为 73，深度为 64 的 Write Data FIFO，其中 73bit 数据的 bit[63:0] 为 write data，bit[71:64] 为 write data mask，bit[72] 为 write data last 指示信号；
- 1 个数据位宽为 68，深度为 64 的 Read Data FIFO，其中 68bit 数据的 bit[63:0] 为 read data，bit[64] 为 read data last 指示信号，bit[67:65] 为 command ID。

每个 FIFO 都将输出溢出指示信号、空满指示信号给用户，同时 Write Data FIFO 和 Read Data FIFO 还将输出 usr\_dw\_space\_\* 和 usr\_dr\_space\_\* 信号来告诉用户 FIFO 的剩余空间。

DDR2 Controller 将以轮询（Round Robin）的方式，轮流读取两组 FIFO 中保存的命令。同时，如果两个 cmd/addr FIFO 中有 1 个为空，那么 DDR2 Controller 将自动持续访问非空的 FIFO。当某组用户接口发送读命令时，相应的读数据也将被保存在该组用户接口的 Read Data FIFO 中。

当用户使用 READ/READA 命令读取存储器数据时，读出数据的 usr\_read\_cmd\_id 值，将和对应的 READ/READA 命令的 usr\_cmd\_id 值相同。用户使用其它命令时，usr\_cmd\_id 的输入无意义，可以输入 0。

USR\_PORT\_0 中异步 FIFO 所使用的用户侧时钟为 usr\_clk\_0，USR\_PORT\_1 中异步 FIFO 所使用的用户侧时钟为 usr\_clk\_1。控制器侧时钟均为 sclk。

用户可以通过 MIB 配置 mc1\_dqs\_width 信号为 1，来访问单个 x8 的器件。这种情况下，有效的写数据为 Write Data FIFO 的 bit[31:0]，有效的读数据为 Read Data FIFO 的 bit[31:0]。

### 2.2.2. Parameter Convert

Parameter Convert 模块用于对 MIB 输入的时序参数（如 mc1\_param\_trcd）进行时钟域转换。MIB 输入的时序参数的单位是 1 个 dqs ( eclk ) 周期，而控制器内部逻辑所使用的时钟为 sclk。在 X2 模式下 sclk 是 eclk 的二分频，因此要将时序参数统一除以 2（左移一位）以后向上取整。

部分逻辑也会使用到转换前的时序参数，以便达到更高的控制粒度。



### 2.2.3. Command Decode Logic (CDL)

SDRAM\_CORE 模块中的 Command Decode Logic ( CDL ) 模块，用于接收用户接口处输入的命令并进行解码。CDL 模块会根据用户命令的类型以及当前 DDR2 存储器中 Bank 和 Row 的状态，来生成各命令对应的内部标志信号，并发送给 CAL。

CDL 中包含智能 Bank 管理逻辑，该逻辑会实时追踪当前存储器中每个 Bank 的开闭状态，以及保存处于 Open 状态的 Bank 中 Active 的 Row 的地址。CDL 将使用这些信息来判断何时向存储器发送 PRECHARGE 和 ACTIVE 命令。

CDL 中包含了自动刷新逻辑，该逻辑会根据用户的配置自动生成刷新命令。

CDL 中包含了跨 Page 命令自动处理逻辑。当用户输入的读写命令的访问地址跨越 Page 边界时，会将用户输入的命令自动分割为两个命令，然后依次执行。在用户设定 mc1\_row\_width、mc1\_bank\_width 和 mc1\_col\_width 后，控制器将自动计算当前的 Page Size 值，以确定 Page 边界。

CDL 模块还提供了用户命令输入的 burst 功能。该功能可以在用户输入一个读或者写命令时，令 DDR2 Controller 自动向存储器重复发送 N 次该命令，其中次数 N 由用户输入 cmd\_burst\_cnt 指定。若使用 burst 功能发送 WRITEA 或 READA 命令，那么只有该 burst 中的最后一个命令执行之后，才会执行 AutoPrecharge。

CDL 中使用了四个队列来缓存解码后的命令，以便在 CAL 中实现四级流水，从而提高命令处理效率。

### 2.2.4. Command Application Logic (CAL)

CAL 模块会轮流接收 CDL 的四个队列中缓存的解码后命令，随后将这些解码后命令按照 JESD79 协议的时序要求，依次转换为相应的 CS、RAS、CAS 和 WE 等信号。最终转换得到的 CS、RAS、CAS 和 WE 等信号将输出给 SDRAM\_PHY 模块。CDL 和 CAL 模块并行工作，分别负责填充和取出四个队列中的命令。

CAL 将严格的按照 MIB 提供的时序参数处理这一转换过程，以保证转换后得到的命令时序符合协议规定。

Data Path Logic(DPL)模块也位于 CAL 模块中。DPL 模块负责保证从 Interface FIFO 的控制器侧接口输出的数据会在正确的时间稳定的抵达存储器接口，以及从存储器读出的数据能正确的抵达 Interface FIFO 的控制器侧接口。

### 2.2.5. On-Die Termination (ODT)

ODT 功能的作用是提升传输通道的信号完整性。DDR2 Controller 中的 ODT 模块负责根据 CAL 输出的命令，确定输出的 odt 信号的电平。

em\_ddr\_odt 信号是与时钟同步输出的信号，将只在发送写命令时拉高，其余时间都将保持低电平在这种设定下，ODT 相关的各个时序都能够满足。

相较于 DDR2 模式，DDR2 Controller 还满足了 DDR3 模式下额外要求的 ODTH4 和 ODTH8 时序参数。

### 2.2.6. SDRAM PHY

SDRAM PHY 模块和 PHY\_IO 模块组合起来就是广义上的 DDR2 PHY。DDR2 PHY 通

过 DFI 接口与 SDRAM\_CORE 模块相连。

在 Seal 30K 中，我们按照“数字逻辑”和“模拟电路单元”这两个类型，将广义上的 DDR2 PHY 分为只包含数字逻辑的 SDRAM PHY 模块和只包含模拟电路单元的 PHY\_IO 模块。

SDRAM PHY 模块中包含了初始化逻辑、Read Training 逻辑、Read Leveling 逻辑、Write Leveling 逻辑以及命令、数据路径的选通逻辑。PHY\_IO 模块负责生成时钟以及进行单边沿数据和双边沿数据之间的转换，包含了 Clock Synchronization Module ( CSM )、用于 X2 Gearing 的 xsDDR2MDQIN 和 xsDDR2MDQOUT 模块、用于生成 DQS 信号的 xsDQSBUF 模块以及其它模拟电路单元。

从存储器读回的数据还会在 SDRAM PHY 模块中进行跨周期对齐处理。即当读回的两组 dq 之间的相对延迟为 0~1.75 个 eclk 周期时，DDR2 Controller 会自动对齐两组 dq 数据后再读出。该功能要求 Read Training 模块使能。

### 2.2.7. Initialization Logic

初始化模块用于生成 JESD79-2F 与 JESD79-3F 协议规定的 DDR2 存储器初始化命令序列。在上电或者复位之后，存储器在发送数据前必须先经过初始化。DDR2 Controller 会在复位释放后自动启动存储器初始化命令序列。在初始化、Read Training、Write Leveling( DDR2 模式下不进行 Write Leveling ) 和 Read Leveling 都完成后，DDR2 Controller 会开始读取 Interface FIFO 中的命令与数据，开始正常工作。图 2-2 是 DDR2 Controller 初始化流程。

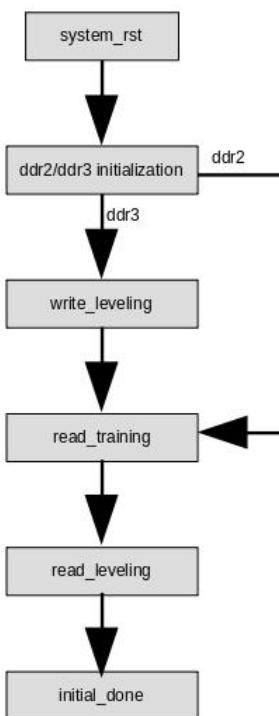


图 2-2 DDR2 Controller 初始化流程

### 2.2.8. Write Leveling Logic

Write Leveling 模块只在 DDR3 模式下起作用。在 DDR3 模式下，当 MIB 输出的



mc1\_wl\_en 信号使能时 ,Write Leveling 将在 DDR3 初始化完成后开始进行。当 Write Leveling 完成后，同时用户接口处的 wl\_err 信号会表明 Write Leveling 过程是否成功。

当有多块 DDR3 SDRAM 颗粒使用时，为了避免同步切换噪声 ( SSN )，在板级布线时多采用 fly-by 布线，而非 T 型布线。fly-by 布线是指地址、命令和时钟的布线依次经过每一颗 DDR3 SDRAM 芯片。fly-by 布线虽然减少了 SSN，但较长的布线延迟导致了 DDR3 Controller 在同一时刻输出的 DQS 和 CK 信号，不会同时到达 DDR3 SDRAM。因此采用 Write Leveling 来调整 DDR3 SDRAM 的 DQS 差分对和 CK 差分对的相对位置，使 DQS 信号和 CK 信号的边沿对齐。对 DIMM 应用来说，必须使能 Write Leveling 模块 ( 使能 wl\_en 信号 )。对将 DDR3 SDRAM 芯片放在和 SEAL 芯片同一块 PCB 板上的应用，如果 DDR3 SDRAM 芯片之间使用 fly-by 拓扑结构，必须使能 Write Leveling 模块；如果 DDR3 SDRAM 芯片之间并未使用 fly-by 拓扑结构，必须关闭 Write Leveling 模块，这是因为 DQS 和 CK 边沿之间没有 fly-by 对应的时序关系时，Write Leveling 过程有可能会失败。

Seal DDR2 Controller 的 Write Leveling 过程遵循 JEDEC 规范所规定的步骤，可以查阅 JESD79-3F 协议获取更多关于 Write Leveling 的详细信息。

在 DDR2 模式下，Write Leveling 模块不工作，但是用户可以使用 MIB 配置 mc1\_wl\_delay\_\* 信号来调整 DQS 和 DQ 信号相对于 CK 信号的相位位置。

### 2.2.9. Read Training Logic

每当 DDR2 Controller 进行读操作时 ,PHY\_IO 中的 DQSBUFI 模块都会在接收到 READ 指令后 , 确定接收到的 DQS 信号的前导码( preamble )的位置。确定前导码位置后 xsDQSBUF 模块会根据输入的 DQS 信号生成去除毛刺的 “clean DQS” ，同时拉高 DATAVALID 信号来指示读数据有效的时间窗口。每一个 DQS 信号都会独立的进行这一 Read Training 过程。

由于 PCB 布线延迟和芯片 I/O 延迟的存在 ,xsDQSBUF 模块接收到的 DQS 前导码的位置是可能与标准位置有偏移。SDRAM PHY 中的 Read Training 模块将输出 READ[3:0] 、 READCLKSEL[2:0] 给 xsDQSBUF 模块，用以确定准确的 DQS 前导码的位置。 xsDQSBUF 模块使用 READ 信号生成脉冲信号 read\_ena ，使用 READCLKSEL 信号调整 read\_ena 信号的位置。每当 READCLKSEL 信号改变时， xsDQSBUF 模块都会检测 read\_ena 脉冲是否覆盖了 DQS 的有效区域。如果检测成功 ,xsDQSBUF 模块会生成 clean DQS，并将 BURSTDET 信号拉高以告知 Read Training 模块；如果检测失败， xsDQSBUF 模块会等待 Read Training 模块调整 READCLKSEL 值之后，再次开始检测。Read Training 模块在接收到 BURSTDET 高电平后，会保存当前的 READCLKSEL 值，之后 DDR2 Controller 进行读操作时都将使用该保存值。

可以通过 MIB 输出的 mc1\_rt\_en 信号来控制是否进行 Read Training 过程。Read Training 在存储器初始化完成和 Write Leveling 完成后开始进行。当 Read Training 过程完成后， rt\_err 信号将指示 Read Training 过程是否出现错误。在 Read Training 过程中的读写访问操作的 Burst Length 随 mc1\_addr\_init\_0 中设置的初始值而定。

### 2.2.10 Read Leveling Logic

Read Leveling 用于补偿读路径上 dqs 和 dq 间的负载不均衡。理想情况下，读路径上 dqs



和对应的 dq 是边沿对齐的。当 dqs 和对应的 dq 之间存在延迟时，可使用 Read Leveling 进行调整。

PHY\_IO 中的 IDDR 模块会检测每个 dq 信号边沿与 dqs90 信号(由 xsDQSBUF 模块对 dqs90 相移 90°后得到)边沿之间的距离，并生成 edge\_plus 和 edge\_minus 信号。Read Leveling 模块会根据 edge\_plus 和 edge\_minus 信号，向 PHY\_IO 中的 xsDQSBUF 模块输出 rdmove、rdloadn 和 rddirection 信号，来对生成 dqs90 的 delay code 进行调整。

可以通过 MIB 输出的 mc1\_rl\_en 信号来控制是否进行 Read Leveling 过程。Read Leveling 在 Read Training 完成后进行。Read Leveling 过程中，控制器将先向存储器的地址 0 写入数据 64'hFFFF\_0000\_FFFF\_0000，随后连续读取地址 0 的数据。Read Leveling 模块会根据读过程中 xsDDR2MDQIN 反馈的 edge\_plus 和 edge\_minus 信号，调整 dqs90 的相位，直到 edge\_plus || edge\_minus 为 0 (表明 Read Leveling 成功) 或 edge\_plus & edge\_minus 为 1 (表明 Read Leveling 出错)。当 Read Leveling 完成后，rl\_err 信号将指示 Read Leveling 过程是否出现错误。

除了在初始化过程中执行 Read Leveling 过程外，用户还可以通过 usr\_rl\_en 信号在 DDR2 存储器工作过程中进行 Read Leveling：

在 DDR3 模式下，用户首先需要通过 LMR 命令使能存储器的 MPR 模式，随后连续发送读命令。当读出 MPR 指定值时，再拉高 usr\_rl\_en 信号。Read Leveling 模块会开始自动开始进行 Read Leveling 过程，并在完成后将 usr\_rl\_done 信号设为 1。用户在检测到 usr\_rl\_done 信号后应立即将 usr\_rl\_en 拉低，否则控制器会再次执行 Read Leveling。

## 2.3. 接口说明

表 2-1 DDR2 Controller 接口说明

| Port Name  | Width | Active State | I/O | Description  |
|--|-------|--------------|-----|--|
| clk_in   | 1     | N/A          | I   | DDR2 控制器启动时钟。该时钟为低速时钟，需独立提供一个连续的低频时钟即可，如 FPGA 内部 OSC 或其他连续时钟（不可使用 DDR2 的 sclk 及 eclk 产生的时钟）。 |
| rst_n  | 1     | Low          | I   | 异步复位信号，低电平时将对整个 DDR2 Controller 和存储器复位。  |
| Local User Interface (the wildcard character '*' in the following chart represents 0 or 1) |       |              |     |  |
| Port Name  | Width | Active State | I/O | Description  |
| init_start   | 1     | High         | O   | init_start 为高表明正在进行初始化流程   |
| init_done  | 1     | High         | O   | init_done 为高表明初始化已完成，电平锁存信号  |
| phy_io_clk   | 1     | N/A          | I   | PHY 接口所需高速时钟 eclk 输入，如：ddr2 控制器  |



|                     |    |      |   |   |
|---------------------|----|------|---|---|
|                     |    |      |   | 运行在 800Mbps 时 ,该端口需要输入 400MHz 时钟。   |
| phy_pll_lock        | 1  | N/A  | I | 产生 eclk 的 PLL 输出的 LOCK 信号。  |
| usr_clk_*           | 1  | N/A  | I | USR_PORT_* 用户输入时钟   |
| usr_cmd_we_*        | 1  | High | I | USR_PORT_* 中 cmd/addr FIFO 的写使能   |
| usr_cmd_*           | 4  | N/A  | I | USR_PORT_* 用户命令输入   |
| usr_cmd_id_*        | 3  | N/A  | I | USR_PORT_* 用户输入的读命令的 ID 值   |
| usr_addr_*          | 30 | N/A  | I | USR_PORT_* 用户读地址/写地址/LMR 命令写入值的输入信号   |
| usr_cmd_burst_cnt_* | 5  | N/A  | I | USR_PORT_* 用户输入命令的 burst 数目输入。在用户输入读或写命令时 , 控制器将自动重复执行该命令 cmd_burst_cnt 次。控制器会自动生成每次命令的读或写地址 ( 连续自增 ) 。 cmd_burst_cnt 的取值范围是 0~31 , 当 cmd_burst_cnt 取 0 时代表重复 32 次。 |
| usr_cmd_full_*      | 1  | High | O | USR_PORT_* 中 cmd/addr FIFO 输出的写满指示位 , 位于 usr_clk_* 时钟域  |
| usr_cmd_overflow_*  | 1  | High | O | USR_PORT_* 中 cmd/addr FIFO 输出的写溢出指示位 , 位于 usr_clk_* 时钟域下  |
| usr_cmd_space_*     | 1  | High | O | USR_PORT_* 中 cmd/addr FIFO 输出的已用空间指示位 , 位于 usr_clk_* 时钟域  |
| usr_write_data_*    | 64 | N/A  | I | USR_PORT_* 用户写数据输入。用户端的写数据位宽是存储器端数据总线位宽的 4 倍 ( X2 Gear 下 ) 或 8 倍 ( X4 Gear 下 ) 。  |
| usr_write_mask_*    | 8  | High | I | USR_PORT_* 用户写数据输入的掩膜信号。usr_write_mask_* 中的每一位对应了 usr_write_data_* 中的一个 byte。   |
| usr_dw_we_*         | 1  | High | I | USR_PORT_* 中 write data FIFO 的写使能   |
| usr_dw_full_*       | 1  | High | O | USR_PORT_* 中 write data FIFO 输出的写满指示位 , 位于 usr_clk_* 时钟域  |
| usr_dw_overflow_*   | 1  | High | O | USR_PORT_* 中 write data FIFO 输出的写溢出指示位 , 位于 usr_clk_* 时钟域下  |
| usr_dw_space_*      | 6  | N/A  | O | USR_PORT_* 中 write data FIFO 输出的已用空间指示位 , 位于 usr_clk_* 时钟域  |



| usr_dr_re_*                 | 1     | High         | I   | USR_PORT_*中 read data FIFO 的读使能   |
|-----------------------------|-------|--------------|-----|---|
| usr_dr_empty_*              | 1     | High         | O   | USR_PORT_*中 read data FIFO 输出的读空指示位 , 位于 usr_clk_*时钟域   |
| usr_dr_underflow_*          | 1     | High         | O   | USR_PORT_*中 read data FIFO 输出的读溢出指示位 , 位于 usr_clk_*时钟域  |
| usr_dr_space_*              | 6     | N/A          | O   | USR_PORT_*中 read data FIFO 输出的已用空间指示位 , 位于 usr_clk_*时钟域   |
| usr_read_last_*             | 1     | High         | O   | usr_read_last_* 为高时表明 : 当前输出的 usr_read_data_* 为相应读命令的最后一笔数据 ; 位于 usr_clk_*时钟域                         |
| usr_read_cmd_id_*           | 3     | N/A          | O   | 当前时刻输出给用户的 usr_read_data_* 所对应的读命令的 ID 值  |
| usr_read_data_*             | 64    | N/A          | O   | USR_PORT_*输出给用户的读数据   |
| usr_rl_en                   | 1     | High         | I   | Read Leveling 使能信号 ; 该信号仅在 MIB 输入 mc1_rl_en 为高时有效。  |
| usr_rl_done                 | 1     | High         | O   | 在控制器完成 usr_rl_en 请求的 Read Leveling 操作后 , 将会拉高 usr_rl_done 信号告知用户。该信号仅在 MIB 输入 mc1_rl_en 为高时有效。电平锁存信号。 |
| rt_err                      | 1     | High         | O   | Read Training 错误指示信号。当 rt_err 为高时 , 表明 Read Training 过程失败 , 控制器未能正常工作。电平锁存信号。                         |
| rl_err                      | 1     | High         | O   | Read Leveling 错误指示信号。当 rl_err 为高时 , 表明 Read Leveling 过程失败 , 控制器未能正常工作。电平锁存信号。                         |
| clocking_good               | 1     | High         | O   | clocking_good 为高表明当前复位已释放 , 目 DLL 与 PLL 均已处于 Lock 状态  |
| DDR2 SDRAM Memory Interface |       |              |     |   |
| Port Name                   | Width | Active State | I/O | Description   |
| em_ddr_clk                  | 1     | N/A          | O   | 控制器输出给 DDR2 SDRAM 的时钟。DDR2 模式下最大频率 400 MHz, DDR3 模式下最大频率 533MHz。                                      |
| em_ddr_clk_n                | 1     | N/A          | O   | em_ddr_clk 的差分时钟 , 由 PHY_IO 中的模拟电路单元生成。   |



|              |    |      |     |  |
|--------------|----|------|-----|--|
| em_ddr_cke   | 1  | High | O   | 控制器输出给 DDR2 SDRAM 的时钟使能信号。                                   |
| em_ddr_addr  | 16 | N/A  | O   | 控制器输出给 DDR2 SDRAM 的地址总线 ( Row Address 和 Column Address 合用 )。 |
| em_ddr_ba    | 3  | N/A  | O   | 控制器输出给 DDR2 SDRAM 的 bank 地址总线。                               |
| em_ddr_data  | 16 | N/A  | I/O | 控制器输出给 DDR2 SDRAM 的双向数据总线。                                   |
| em_ddr_dm    | 2  | High | O   | 控制器输出给 DDR2 SDRAM 的写数据掩膜信号。                                  |
| em_ddr_dqs   | 2  | N/A  | I/O | 控制器输出给 DDR2 SDRAM 的双向数据选通信号。                                 |
| em_ddr_dqs_n | 2  | N/A  | I/O | em_ddr_dqs 的差分信号，由 PHY_IO 中的模拟电路单元生成。                        |
| em_ddr_cs_n  | 1  | Low  | O   | 控制器输出给 DDR2 SDRAM 的片选 ( Chip Select ) 信号。                    |
| em_ddr_cas_n | 1  | Low  | O   | 控制器输出给 DDR2 SDRAM 的列地址 ( Column Address ) 选通信号。              |
| em_ddr_ras_n | 1  | Low  | O   | 控制器输出给 DDR2 SDRAM 的行地址 ( Row Address ) 选通信号。                 |
| em_ddr_we_n  | 1  | Low  | O   | 控制器输出给 DDR2 SDRAM 的写使能信号。                                    |
| em_ddr_odt   | 1  | High | O   | 控制器输出给 DDR2 SDRAM 的 ODT 控制信号。                                |

#### MIB Interface

| Port Name            | Width | Default value | I/O | Description  |
|----------------------|-------|---------------|-----|--|
| mc1_ddr_mode         | 1     | 0             | I   | DDR2 Controller 工作模式选择：<br>0 : DDR2 模式 ; 1 : DDR3 模式。  |
| mc1_wl_delay         | 16    | 0             | I   | DDR2 模式下 , mc1_wl_delay[7:0] 是对 dqsw0 和 dqa[7:0] 的延迟码; mc1_wl_delay[15:8] 是对 dqsw1 和 dqa[15:8] 的延迟码        |
| mc1_dll_freeze_en    | 1     | 0             | I   | 如果 mc1_dll_freeze_en 为高电平 , 那么当 DDR2 存储器进入 Power Down 或 Self Refresh 状态时 , DLL 会进入 freeze 状态。              |
| mc1_idle_auto_ref_en | 1     | 0             | I   | 如果 mc1_idle_auto_ref_en 为高电平 , 那么当 DDR2 Controller 的命令队列空闲超过 1~2 个 tREFI 时间后 , DDR2 Controller 会自动向存储器连续发送 |



|                     |   |       |   |   |
|---------------------|---|-------|---|---|
|                     |   |       |   | mc1_ar_burst_en 个 Refresh 命令。   |
| mc1_rl_en           | 1 | 1     | I | DDR2 Controller 中初始化后 Read Leveling 流程的使能信号。  |
| mc1_rl_mon_dis      | 4 | 4'b0  | I | <p>Read Leveling 过程中调整的范围，其中 bit[1:0]对应 dqs0 和 dq[7:0]的调整范围，bit[3:2]对应 dqs1 和 dq[15:8]的调整范围。例如当 mc1_rl_mon_dis=4' b0 时，只有当 dqs90 边沿和对应的 dq90 边沿之间的距离小于 100/2=50ps 时，Read Leveling 模块才会相应移动 dqs90 的相位。</p> <p>2'b00: 100ps;<br/>2'b01: 200ps;<br/>2'b10: 300ps;<br/>2'b11: 400ps</p> |
| mc1_wl_en           | 1 | 0     | I | DDR2 Controller 中初始化后 Write Leveling 流程的使能信号。DDR3 模式下( ddr_mode 为高电平 ) ,wl_en 为高电平时 ,DDR2 Controller 中的 Write Leveling 模块才会工作。DDR2 模式下 ,wl_en 为高电平时 ,mc1_wl_delay_0 和 mc1_wl_delay_1 的配置才会生效。   |
| mc1_rt_en           | 1 | 1     | I | DDR2 Controller 中初始化后 Read Training 流程的使能信号。当 rt_en 为高电平时 ,DDR2 Controller 才会对存储器进行 Read Training 操作。   |
| mc1_ext_auto_ref_en | 1 | 0     | I | <p>DDR2 Controller 的外部自动刷新使能信号：</p> <p>0 : DDR2 Controller 使用内部的自动刷新逻辑 , 每隔 mc1_ar_burst_en*trefi 时间就向存储器发送 mc1_ar_burst_en 次刷新命令；</p> <p>1 : 当用户输入 ext_auto_ref 高电平时 , DDR2 Controller 会向存储器发送 mc1_ar_burst_en 次刷新命令。</p>  |
| mc1_ar_burst_en     | 4 | 0     | I | Auto Refresh Control Burst Count 信号。DDR2 Controller 内部自动刷新逻辑使用的刷新间隔计数值。mc1_ar_burst_en 的有效取值为 1~9 时 , 取值更高时应相应减小 mc1_param_trfc 的值 , 否则会导致 DDR2 存储器刷新间隔超出协议规定时间。  |
| mc1_spd_bin         | 2 | 2'b00 | I | Speed Bin 选择信号。根据存储器工作频率 , 确定 DDR2 Controller 初始化流程中延时计数器的目标  |



|                 |   |          |   |  |
|-----------------|---|----------|---|--|
|                 |   |          |   | 值。<br>2'b00 : SG25E ( 800MHz )<br>2'b01 : SG187E ( 1066MHz )<br>2'b10 : SG3E ( 667MHz )<br>2'b11 : SG37E ( 533MHz ) or lower frequency |
| mc1_addr_seq    | 1 | 0        | I | 选择用户输入的地址排序类型：<br>0 : Row-Bank-Column<br>1 : Bank-Row-Column   |
| mc1_dqs_width   | 3 | 2        | I | 选择用户使用 x8 还是 x16 的存储器<br>3'b010 : 使用 x16 存储器 , dqs0/dqs1 和 dq[15:0] 均会使用<br>3'b001 : 使用 x8 存储器 , 只使用 dqs0 和 dq[7:0]                    |
| mc1_row_width   | 5 | 5'b10000 | I | 所使用的存储器的 Row 地址的位宽   |
| mc1_bank_width  | 2 | 2'b11    | I | 所使用的存储器的 Bank 地址的位宽  |
| mc1_col_width   | 4 | 4'b1011  | I | 所使用的存储器的 Column 地址的位宽  |
| mc1_rdimm       | 1 | 1'b0     | I | 选择 DIMM 类型为 Un-buffered DIMM 还是 Registered DIMM：<br>0 : Un-buffered DIMM<br>1 : Registered DIMM  |
| mc1_dual_rank   | 1 | 1'b0     | I | 选择 DIMM 中 RANK 的数目：<br>0 : Single RANK<br>1 : Dual RANK<br><br>注意 30K 中仅支持 Single RANK   |
| mc1_addr_mirror | 1 | 1'b0     | I | 选择在 Dual RANK 模式下，是否使用地址镜像：<br>( Single RANK 下该配置无效 )<br>0 : 地址镜像关闭<br>1 : 地址镜像使能  |
| mc1_mpr_calib   | 1 | 1'b0     | I | 利用 DDR 存储器的 MPR 功能来进行 Read Leveling。   |



|                   |    |          |   |   |
|-------------------|----|----------|---|---|
| mc1_addr_init_0   | 16 | 16'h0a52 | I | 初始化完成后对存储器写入的 MR/MR0 寄存器的初始值  |
| mc1_addr_init_1   | 16 | 16'h0020 | I | 初始化完成后对存储器写入的 EMR1/MR1 寄存器的初始值  |
| mc1_addr_init_2   | 16 | 16'h0000 | I | 初始化完成后对存储器写入的 EMR2/MR2 寄存器的初始值  |
| mc1_addr_init_3   | 16 | 16'h0000 | I | 初始化完成后对存储器写入的 EMR3/MR3 寄存器的初始值  |
| mc1_param_trcd    | 4  | 4'h5     | I | tRCD: ACT to RD(A) or WT(A) Delay。<br><br>注：前缀为 param_* 的参数的计量单位是一个 sclk 时钟周期；计算不同 speed bin 下的 param_* 参数值时，应将协议中对应的时序参数值先转换为 em_ddr_clk ( eclk ) 的周期数，再除以 2 并向上取整 ( Round Up )。   |
| mc1_param_trp     | 4  | 4'h5     | I | PRE command period  |
| mc1_param_twr     | 4  | 4'h6     | I | WRITE recovery time   |
| mc1_param_tras    | 5  | 5'h12    | I | ACT to PRE command period   |
| mc1_param_trc     | 5  | 5'h17    | I | ACT to ACT or REF command period (same bank)  |
| mc1_param_trrd    | 5  | 5'h3     | I | ACTIVE to ACTIVE command period (different bank)  |
| mc1_param_trtp    | 3  | 3'h3     | I | Internal READ Command to PRECHARGE Command delay  |
| mc1_param_twtr    | 3  | 3'h3     | I | Delay from start of internal write transaction to internal read command   |
| mc1_param_tckesr  | 3  | 3'h3     | I | Minimum CKE low width for Self Refresh entry to exit timing<br><br>(This parameter only occurs in DDR3 protocol, but the DDR2 protocol also require the DDR2 SDRAM must remain in Self Refresh mode at least tCKE time. In DDR3 protocol, tCKESR = tCKE(min) + 1nCK. So it's safe to use tCKESR in DDR2 mode) |
| mc1_param_tzqoper | 9  | 9'h102   | I | Normal operation Full calibration time ( 在 DDR2 模式中未使用 )  |
| mc1_param_tzqs    | 7  | 7'h50    | I | Normal operation short calibration time ( 在 DDR2 模式中未使用 )   |
| param_tzqinit     | 11 | 11'h200  | I | Power-up and RESET calibration time ( 在 DDR2 模式中未使用 )   |



|                 |    |         |   |   |
|-----------------|----|---------|---|---|
| mc1_param_mrd   | 5  | 5'h2    | I | Mode Register Set command cycle time  |
| mc1_param_pd    | 3  | 3'h3    | I | Power Down Entry to Exit Timing (This parameter only occurs in DDR3 protocol, but in DDR2 protocol this timing should be larger than tCKE. So param_pd should be set the same value with tCKE in DDR2 mode)         |
| mc1_param_xpdll | 4  | 4'h4    | I | <p>DDR3 模式中为参数 tXP DLL :</p> <p>Exit Precharge Power Down with DLL frozen to commands requiring a locked DLL</p> <p>DDR2 模式中为参数 tXARDS :</p> <p>Exit active power down to read command (slow exit, lower power)</p> |
| mc1_param_cke   | 3  | 3'h3    | I | CKE minimum pulse width   |
| mc1_param_rfc   | 7  | 7'h1e   | I | REF command to ACT or REF command time  |
| mc1_param_xpr   | 8  | 8'ha0   | I | Exit Reset from CKE HIGH to a valid command ( DDR3 协议中为 512nCK , 在 DDR2 协议中无此时序名称 , 但实际上仍存在要求 tXPR=400ns )  |
| mc1_param_mod   | 5  | 5'hc    | I | Mode Register Set command update delay ( 在 DDR2 模式中未使用 )  |
| mc1_param_refi  | 17 | 17'hc30 | I | Average periodic refresh interval   |
| mc1_param_tfaw  | 5  | 5'hd    | I | Four Activate Window  |

在 DDR2 模式下 , 由于 DDR2 存储器没有 MPR 模式 , 因此需要用户首先向某一地址写入适当数据 , 再连续读出该地址。随后再执行余下的步骤。

## 2.4. 用户端操作说明

### 2.4.1. 用户命令编码

用户被允许输入给 DDR2 Controller 的 cmd 信号编码如下表所示 :

表 2-2 DDR2 Controller 用户命令编码

| Command                   | Mnemonic  | cmd[3:0] |
|---------------------------|-----------|----------|
| Read                      | READ      | 0001     |
| Write                     | WRITE     | 0010     |
| Read with Auto Precharge  | READA     | 0011     |
| Write with Auto Precharge | WRITEA    | 0100     |
| Power down Entry          | PDOWN_ENT | 0101     |
| Load Mode Register        | LOAD_MR   | 0110     |



|                      |              |      |
|----------------------|--------------|------|
| Self Refresh Entry   | SEL_REF_ENT  | 1000 |
| Self Refresh Exit    | SEL_REF_EXIT | 1001 |
| Power down Exit      | PDOWN_EXIT   | 1011 |
| ZQ Calibration Long  | ZQ_LNG       | 1100 |
| ZQ Calibration Short | ZQ_SHRT      | 1101 |

DDR2 Controller 只接受上表中所列的命令，其余 cmd 值均为无效命令。如果存储器已经进入 Self Refresh 模式或 Power Down 模式，那么此时用户再输入的 SEL\_REF\_ENT 和 PDOWN\_ENT 命令将被无视。如果存储器已经退出了 Self Refresh 模式或 Power Down 模式，那么此时用户再输入的 SEL\_REF\_EXIT 和 PDOWN\_EXIT 命令将被无视。

## 2.4.2. 命令与地址时序

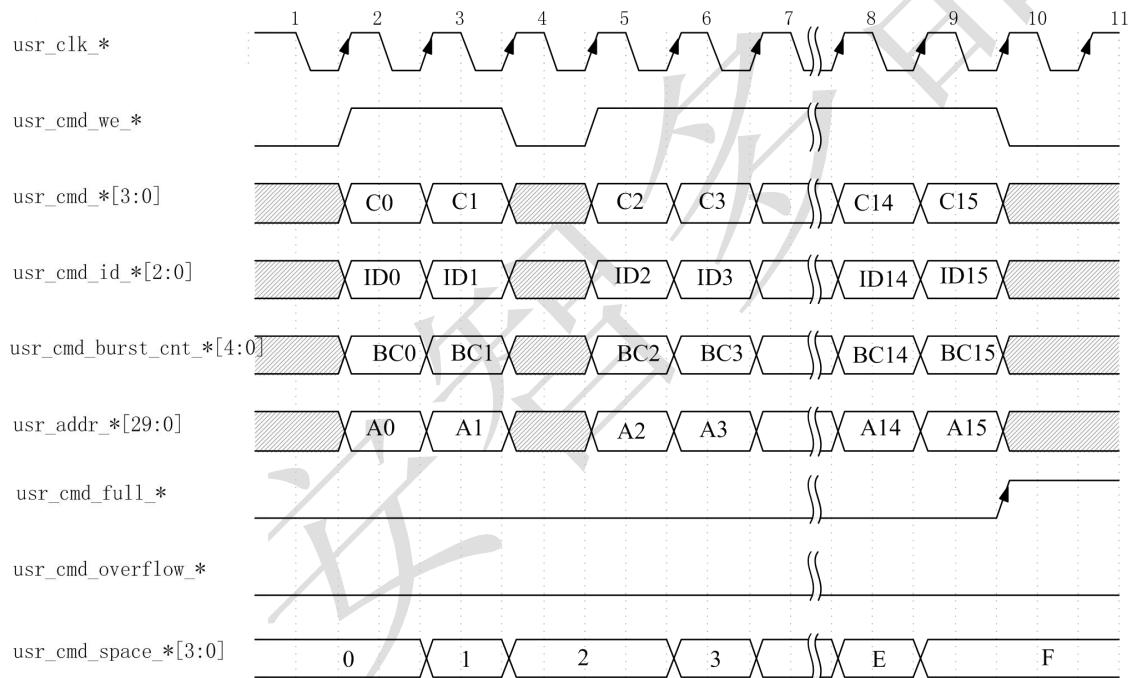


图 2-3 命令与地址时序

在复位释放后，用户就可以向 USR\_PORT\_\*中的 FIFO 写入命令、地址或写数据。图 2-3 为 cmd/addr FIFO 的写入时序。

当用户向 cmd/addr FIFO 写入数据时（usr\_cmd\_we\_\*为高电平时），FIFO 的满信号 usr\_cmd\_full\_\*必须为低电平，否则写入的数据会覆盖掉之前 FIFO 中保存的数据。如果这种覆盖有效数据的情况发生，FIFO 将拉高 usr\_cmd\_overflow\_\*信号报告用户。

DDR2 Controller 支持 command burst 功能。这个功能是指，DDR2 Controller 可以重复发送同一读/写命令，且发送的个数由 cmd\_burst\_cnt 指定。cmd\_burst\_cnt 的值将会和 cmd 命令一起被 DDR2 Controller 采样。在 DDR2 Controller 处理 burst command 时（如果控制器内部命令队列已经被填满），cmd\_rdy 将保持低电平，并且访问存储器的地址会自动增加，直到指定次数的读/写操作都已完成。cmd\_burst\_cnt[4:0]表示命令重复的次数，当 cmd\_burst\_cnt 值大于 1 时，表示使用 command burst 模式；当 cmd\_burst\_cnt 值为 0 时，表



示命令重复的次数为 32。举个例子，如果 DQ 的宽度是 16-bit，cmd\_burst\_cnt 是 2，BL(Memory's Burst Length)是 8，那么一个这样的读命令，将读取的数据量会是：DQ Width \* cmd\_burst\_cnt \* BL = 16\*2\*8=256 bits=32Byte。

DDR2 Controller 允许使用 command burst 功能跨越 Page 边界，但前提条件是用户访问的起始地址要根据读写命令的 Burst Length 与边界对齐（见表 2-3），否则跨越 Page 边界处的访问可能写入或读取错误值。

表 2-3 对用户访问的起始地址的要求

| Burst Length | Start Address[3:0] |
|--------------|--------------------|
| 4            | 0x0                |
|              | 0x4                |
|              | 0x8                |
|              | 0xC                |
| 8            | 0x0                |
|              | 0x8                |

### 2.4.3. 写操作时序

用户在向 cmd/addr FIFO 中写入 WRITE/WRITEA 命令之前，应当先向 Write Data FIFO 中写入足够的写数据。否则，控制器有可能读空 Write Data FIFO(usr\_dw\_underflow\_\*拉高)，导致对相应地址写入了未知数据。

用户向 Write Data FIFO 中写入一笔 burst count 为 32，burst length 为 8 的写数据的时序如图 2-4。

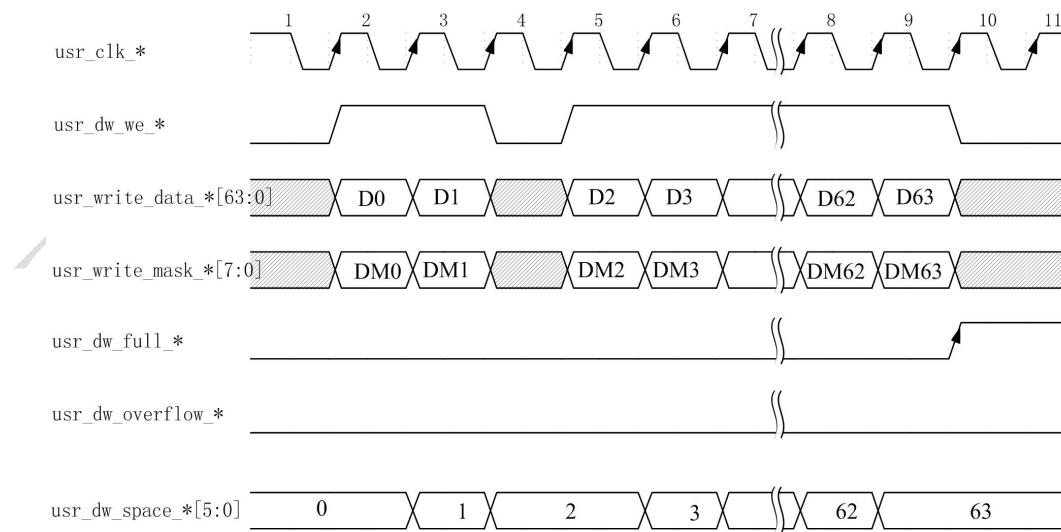


图 2-4 写操作时序

当用户向 Write Data FIFO 写入数据时 (usr\_dw\_we\_\*为高电平时)，FIFO 的满信号usr\_dw\_full\_\*必须为低电平，否则写入的数据会覆盖掉之前 FIFO 中保存的数据。如果这种覆盖有效数据的情况发生，FIFO 将拉高usr\_dw\_overflow\_\*信号报告用户。



usr\_dw\_space\_\*指示的是 Write Data FIFO 中已使用的容量。当控制器从 Write Data FIFO 中取出写数据时，usr\_dw\_space\_\*的值会相应减小，但是由于 usr\_clk\_\*和 sclk 的频率不同，FIFO 中的真实已用容量会在一定的延迟后才反映到 usr\_dw\_space\_\*信号上。同样由于 usr\_clk\_\*和 sclk 的频率不同，一个 usr\_clk\_\*时钟周期内，usr\_dw\_space\_\*的减小值有可能大于 1，例如从 63 直接跳到 61。用户可以将 usr\_dw\_space\_\*作为 Write Data FIFO 的将满信号使用。

DDR2 Controller 从 cmd/addr FIFO 中读取出写命令和写地址后，会从中解析出当前要访问的 row 地址和 bank 地址，并检查该 row 地址在存储器中是否已经是 open 状态。如果该 bank 中没有处于打开状态的 row，那么 DDR2 Controller 将发送给存储器一个 ACTIVE 命令来打开该 row，然后再发送 WRITE 命令进行写操作。如果该 bank 中已经有打开状态的 row，但处于打开状态的 row 不是当前要访问的 row，那么 DDR2 Controller 将先发送给存储器一个 PRECHARGE 命令来关闭这个 row，再发送一个 ACTIVE 命令来打开要访问的 row，最后再发送 WRITE 命令进行写操作。如果要访问的 row 已经处于打开状态，那么 DDR2 Controller 将直接给存储器发送 WRITE 命令（不再发送 ACTIVE 或 PRECHARGE 命令）。

DDR2 Controller 对 WRITEA 命令的处理与 WRITE 基本相同，除了会向存储器通过 A10 位声明这一写命令附带了 Auto Precharge 要求。存储器在接收到 WRITEA 命令后会自动在写操作后附带对当前 bank 的 PRECHARGE 操作。

#### 2.4.4. 读操作时序

用户在向 cmd/addr FIFO 中写入 READ/READA 命令之前，应保证 Read Data FIFO 中有足够的空间来保存读出的数据。用户从 Read Data FIFO 中读出一笔 burst count 为 8，burst length 为 4 的数据的时序如图 2-5 所示。当读出最后一个数据 D7 时，Read Data FIFO 会将 usr\_read\_last\_\*指示信号拉高。

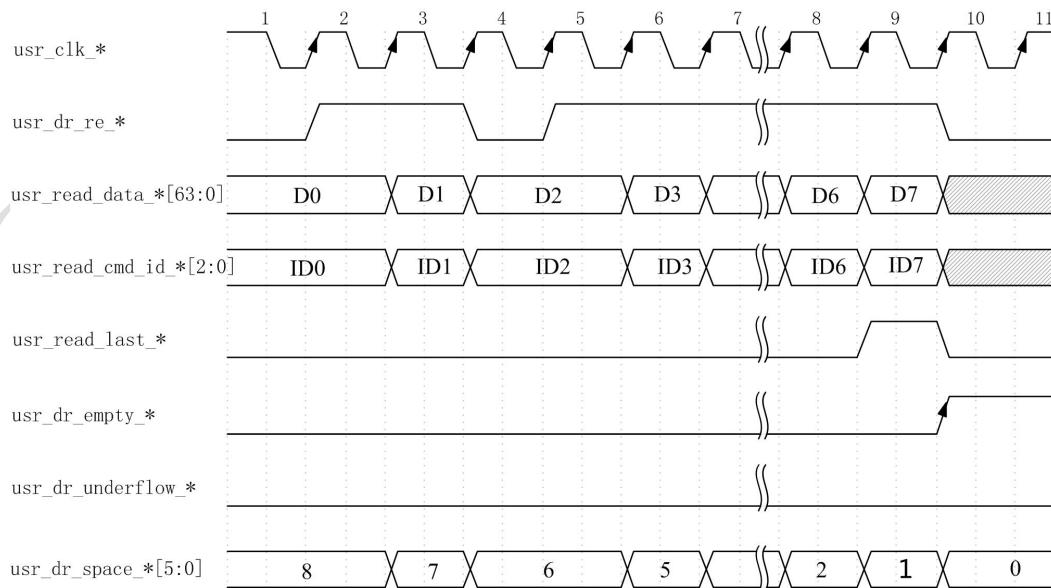


图 2-5 读操作时序

当用户从 Read Data FIFO 读出数据时（usr\_dr\_re\_\*为高电平时），FIFO 的空信号 usr\_dr\_empty\_\*必须为低电平，否则读出的数据是无效信号。如果这种下溢情况发生，FIFO



将拉高 usr\_dr\_underflow\_\*信号报告用户。

usr\_dr\_space\_\*指示的是 Read Data FIFO 中已使用的容量。当控制器向 Read Data FIFO 中写入读数据时，usr\_dr\_space\_\*的值会相应增大，但是由于 usr\_clk\_\*和 sclk 的频率不同，FIFO 中的真实已用容量会在一定的延迟后才反映到 usr\_dr\_space\_\*信号上。同样由于 usr\_clk\_\*和 sclk 的频率不同，一个 usr\_clk\_\*时钟周期内，usr\_dr\_space\_\*的增加值有可能大于 1，例如从 5 直接跳到 7。用户可以将 usr\_dr\_space\_\*作为 Read Data FIFO 的将空信号使用。

在 DDR2Controller 中，读操作和写操作有相同的 row 状态检查设计。DDR2Controller 在进行读操作时，将根据要访问的 row 的状态来自动判断是否要先发送 ACTIVE 或 PRECHARGE 命令。

DDR2 Controller 对 READA 命令的处理与 READ 基本相同，除了会向存储器通过 A10 位声明这一写命令附带了 Auto Precharge 要求。存储器在接收到 READA 命令后会自动在写操作后附带对当前 bank 的 PRECHARGE 操作。

在使用 LMR 命令对存储器中的 DLL 进行复位操作（对 MR0 的 A8 写 1，存储器在完成 DLL 复位后会自动清除 MR0 的 A8）后，要等待 200（DDR2）或 tDLLK=512（DDR3）个 eclk 时钟周期后，才可以进行 READ 操作。在初始化过程中进行的 DLL 复位操作已保证了该延迟，但初始化结束后再进行 DLL 复位时，需要用户注意发送 READ 命令的时间。

## 2.4.5. 刷新操作时序

JEDEC 规范中说明 DDR2 存储器可以累计推迟最多 8 个 refresh 操作，因此 DDR2 Controller 可以一次性发送一个 burst 的 Refresh 命令，Burst Length 最大为 9。

DDR2 Controller 内部包含了一个 Refresh 命令的自动产生逻辑，可以每隔 mc1\_ar\_burst\_en\*tREFI 时间，就向存储器发送连续 mc1\_ar\_burst\_en 个 Refresh 命令。mc1\_ar\_burst\_en 与 tREFI 参数均由 MIB 输入。例如，当 ar\_burst\_en 的值为 9 时，Refresh Burst Length 是 9，那么 DDR2 Controller 将在间隔 9 \* tREFI 时间后，连续向存储器发送 9 个 Refresh 命令。较大的 Refresh Burst Length 有助于提高存储器接口的访问效率。

在连续发送多个 Refresh 命令时，DDR2 Controller 会在第一个 Refresh 命令发送前先发送一个 PRECHARGE 命令，随后的 Refresh 命令前则不会再发送 PRECHARGE 命令。这有助于提高存储器接口的吞吐率。

如果某些具体应用中不需要对存储器进行 Refresh 操作，用户将 ext\_auto\_ref\_en 设为 1，但保持 ext\_auto\_ref 信号为低。

DDR2 Controller 还具有闲时自动刷新功能，即当命令流水闲置超过 1~2tRFC 时间时，控制器将自动对存储器发送 mc1\_ar\_burst\_en 个 Refresh 命令。无论用户选择使用内部自动刷新还是使用外部 ext\_auto\_ref 刷新，在配置 MIB 输出 mc1\_idle\_auto\_ref\_en 为高后，闲时自动刷新功能都会起作用。

## 2.4.6. 自刷新操作时序

用户通过向 cmd/addr FIFO 中写入 SEL\_REF\_ENT 和 SEL\_REF\_EXT 命令，来使存储器



进入或退出自刷新状态。用户输入 SEL\_REF\_ENT 命令后，DDR2 Controller 将先依次向存储器发送 Precharge all Banks 和 Refresh 命令，之后再发送 Self Refresh Entry 命令。DDR2 Controller 在向存储器发送 Self Refresh Exit 命令后，会自动等待 tXSDL( DDR3 )或 tXSRD ( DDR2 )时间后，再开始发送其它命令。

DDR2 Controller 在向存储器发送 Self Refresh Entry 命令之前，会自动先发送一个 Refresh (REF)命令。

### 2.4.7. Read Leveling 操作时序

当 mc1\_rl\_en 使能时，用户可以通过 usr\_rl\_en 信号在 DDR2 存储器工作过程中进行 Read Leveling：

在 DDR3 模式下，用户首先需要通过 LMR 命令使能存储器的 MPR 模式，随后连续发送读命令。当读出 MPR 指定值时，再拉高 usr\_rl\_en 信号。Read Leveling 模块会开始自动开始进行 Read Leveling 过程，并在完成后将 usr\_rl\_done 信号设为 1。用户在检测到 usr\_rl\_done 信号后应立即将 usr\_rl\_en 拉低，否则控制器会再次执行 Read Leveling，参见图 2-6。

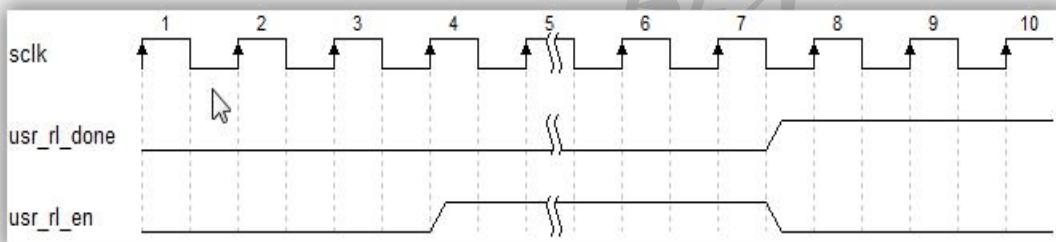


图 2-6 Read Leveling 操作时序

在 DDR2 模式下，由于 DDR2 存储器没有 MPR 模式，因此需要用户首先向某一地址写入适当数据，再连续读出该地址。随后再执行余下的步骤。

### 2.5. 用户端地址映射说明

用户端地址 addr 信号的位宽为 ADDR\_WIDTH，在读/写命令时，其与存储器侧地址的映射关系如图 2-7、2-8 所示：当 mc1\_addr\_seq=0 时，用户端地址为 Row-Bank-Column 排列：

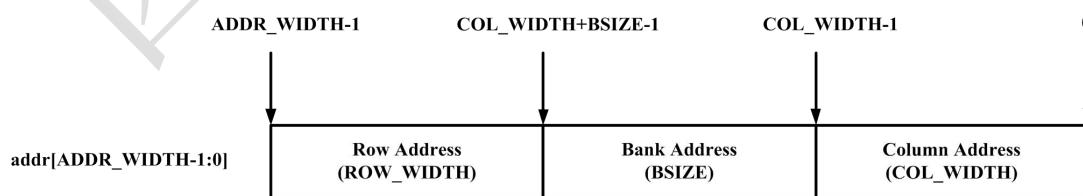


图 2-7 mc1\_addr\_seq=0 时用户端地址映射

当 mc1\_addr\_seq=1 时，用户端地址为 Bank-Row-Column 排列：

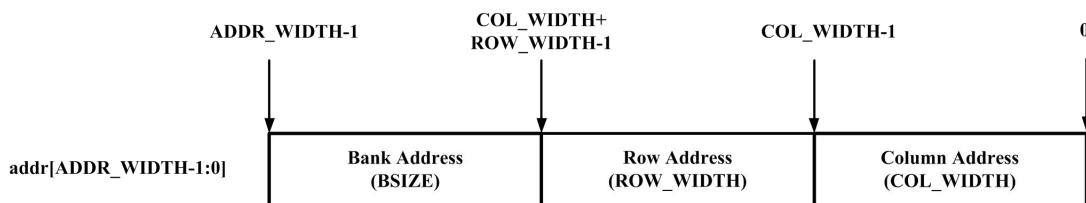


图 2-8 mc1\_addr\_seq=1 时用户端地址映射

在 Seal30K 中， ADDR\_WIDTH 为 30，其中 ROW\_WIDTH 为 16， BSIZE 为 3， COL\_WIDTH 为 11。

## 2.6. 模式寄存器配置说明

### 2.6.1. addr 映射关系

用户使用 LOAD\_MR 命令配置模式寄存器时，应当在 addr 总线上输入目标模式寄存器的地址和要写入的数据。配置模式寄存器时的 addr 的映射关系如图 2-9 所示：

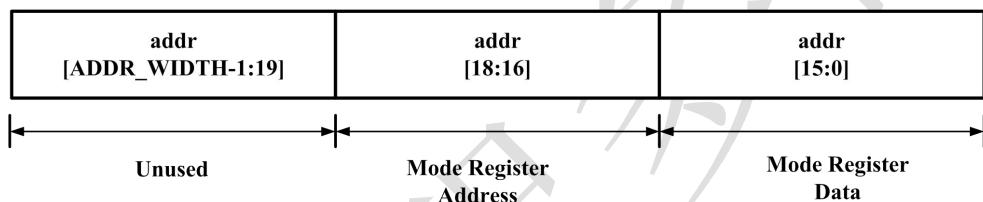


图 2-9 配置模式寄存器时 addr 映射

在使用 LOAD\_MR 命令时，addr[15:0]须输入要配置的 16 位数据，addr[18:16]须输入要配置的模式寄存器的地址（地址分配见表 2-3），addr[29:19]输入无效。

表 2-3 模式寄存器地址

| Mode Register | Addr[18:16] |
|---------------|-------------|
| MR0/MR        | 000         |
| MR1/EMR1      | 001         |
| MR2/EMR2      | 010         |
| MR3/EMR3      | 011         |

### 2.6.2. 模式寄存器初始值

在存储器的初始化过程中（包括 Read Training 和 Write Leveling 过程）将使用 MIB 输入的模式寄存器的初始值（addr\_init0~3）。如果在初始化结束后，用户不使用 LOAD\_MR 命令对模式寄存器进行修改，那么模式寄存器将一直保持初始值。表 2-4 所示为 addr[15:0] 与模式寄存器中参数的对应关系。



表 2-4 模式寄存器初始值

| Type                      | Registers                       | Value  | Description                         | User Address |
|---------------------------|---------------------------------|--------|-------------------------------------|--------------|
| <b>DDR2 Mode Register</b> |                                 |        |                                     |              |
| DDR2 MR<br>(BA=000)       | Burst Length                    | 3'b010 | BL = 4                              | addr[2:0]    |
|                           | Burst Type                      | 1'b0   | Sequential                          | addr[3]      |
|                           | CAS Latency                     | 3'b101 | CL = 5 Cycles                       | addr[6:4]    |
|                           | Test Mode                       | 1'b0   | Normal                              | addr[7]      |
|                           | DLL Reset                       | 1'b0   | DLL Reset = No                      | addr[8]      |
|                           | WR Recovery for autoprecharge   | 3'b101 | WR = 6 Cycles                       | addr[11:9]   |
|                           | Active Power Down Exit          | 1'b0   | Fast exit (use tXARD)               | addr[12]     |
|                           | All Others                      | 3'b000 | \                                   | addr[15:13]  |
| DDR2 EMR<br>(BA=001)      | DLL Enable                      | 1'b0   | Enable                              | addr[0]      |
|                           | Output Driver Impedance Control | 1'b0   | Full strength                       | addr[1]      |
|                           | RTT0                            | 1'b0   | ODT Disabled                        | addr[2]      |
|                           | Additive Latency                | 3'b100 | AL = 4 Cycles<br>(AL $\geq$ tRCD-1) | addr[5:3]    |
|                           | RTT1                            | 1'b0   | ODT Disabled                        | addr[6]      |
|                           | OCD                             | 3'b000 | OCD Not Applicable                  | addr[9:7]    |
|                           | DQS Mode                        | 1'b0   | Differential Disabled               | addr[10]     |
|                           | RDQS Enable                     | 1'b0   | Disabled                            | addr[11]     |
|                           | Qoff                            | 1'b0   | Output buffer enabled               | addr[12]     |
|                           | All Others                      | 3'b000 | \                                   | addr[15:13]  |
| DDR2 EMR2<br>(BA=010)     | All                             | 16'b0  | \                                   | addr[15:0]   |
| DDR2 EMR3<br>(BA=011)     | All                             | 16'b0  | \                                   | addr[15:0]   |



### 3. DDR2 控制器使用说明

SA5Z-30 FPGA 器件通过 3D 封装方式将 DDR2 memory 器件合封在 FPGA 内部。

#### 3.1 DDR2 控制器系统介绍

DDR2 控制器系统包括了 DDR2 控制器硬核 , PHY 以及 DDR2 memory 器件 , 结构示意  
图如图 3-1 所示。其中 , FPGA die 内部集成了 DDR2 硬核控制器及 PHY。

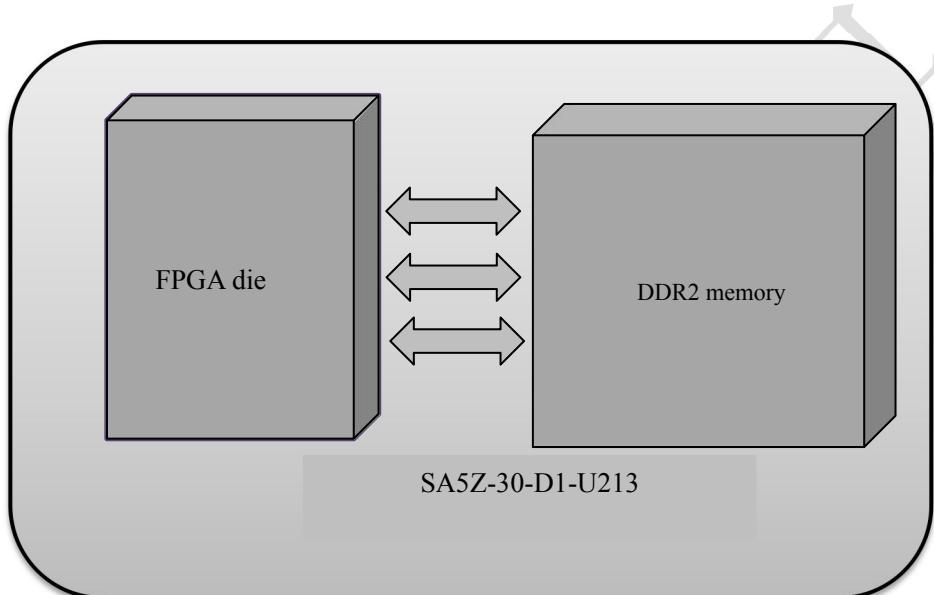


图 3-1 SA5Z-30D1-8U213 结构

参照图 3-1 , DDR2 与 FPGA 之间的接口信号说明如表 3-1 所示。

表 3-1 DDR2 MEMORY 与 FPGA 之间的接口信号

| 名称          | 说明                   | 方 向            |                | 备注   |
|-------------|----------------------|----------------|----------------|--|
|             |                      | 对<br>于<br>DDR2 | 对<br>于<br>FPGA |  |
| CLK<br>nCLK | 时钟<br>Clock          | 输入             | 输出             | 差分时钟(由于使用 SSTL18D_I , 所以<br>nCLK 不应该出现在顶层模块) |
| nCS         | 片选<br>Chip Select    | 输入             | 输出             | (为低电平时)使能 , 或(高电平时)禁用器件<br>工作                |
| CKE         | 时钟使能<br>Clock Enable | 输入             | 输出             |  |
| A[12:0]     | 行/列地址                | 输入             | 输出             | 行地址 : RA0-RA12                               |



|                              |   |    |    |   |
|------------------------------|---|----|----|---|
|                              | row/column address                          |    |    | 列地址 : CA0-CA9   |
| BA[1:0]                      | bank 选择地址<br>bank select address            | 输入 | 输出 | BANK  |
| nRAS                         | 行地址选通<br>Row Address Strobe                 | 输入 | 输出 | 低电平有效   |
| nCAS                         | 列地址选通<br>Column Address Strobe              | 输入 | 输出 | 低电平有效   |
| nWE                          | 写使能<br>Write Enable                         | 输入 | 输出 | 低电平有效   |
| LDM<br>UDM                   | DM: 数据输入/输出<br>掩码<br>Data input/output mask | 输入 | 输出 | LDM 用于掩码 DQ0~DQ7,<br>UDM 用于掩码 DQ8~DQ15.   |
| LDQS,<br>nLDQS<br>UDQS,nUDQS | DQS: 差分数据触发<br>Data Strobe                  | 双向 | 双向 | LDQS,nLDQS 用于 DQ0-DQ7,<br>UDQS,nUDQS 用于 DQ8-DQ15<br>(由于使用 SSTL18D_I , 所以<br>nLDQS/nUDQS 不应该出现在顶层模块) |
| DQ[15:0]                     | 数据输入/输出<br>Data input/output                | 双向 | 双向 |   |
| ODT                          | 片内信号终结<br>On-Die-Termination.               | 输入 | 输出 | ODT 仅作用于 DQ, DM, DQS and nDQS   |
| VREF                         | 参考电压<br>Reference Voltage                   | 输入 | 输入 | 外部专用引脚 不应该出现再顶层模块中。   |

### 3.2 DDR2 控制器 IP 核的使用方法

为了让用户方便、高效的使用 Seal 5000 系列 DDR2 控制器，XIST 为用户提供了 DDR2 控制器 IP。用户可以通过 XIST 提供的 HQ-FPGA 软件生成 DDR2 控制器 IP，将其例化到自己的工程，完成设计。

Seal 5000 系列根据是否内置 DDR2 MEMORY 以及内置 DDR2 MEMORY 容量大小，分为 SA5Z-30-D0 不含内置 DDR2 MEMORY、SA5Z-30-D1 内置 DDR2 MEMORY 容量是 128Mbit、SA5Z-30-D2 内置 DDR2 MEMORY 容量是 512Mbit。用户在生成 DDR2 控制器 IP 时需要确认自己使用的器件是 Seal 5000 系列的哪一型号，在新建工程的时候选择对应的器件型号。



用户需要根据自己所用的器件型号建立相应的工程，这里以 SA5Z-30-D1 为例介绍 DDR2 控制器 IP 生成步骤。

首先，打开 HQ-FPGA 软件，点击新建工程按钮，选择工程存放的路径，确定工程名称以及目标器件型号，如图 3-2 所示，因为这里介绍的是 DDR2 控制器 IP 生成步骤，因此不需要添加源文件，直接点击完成，如图 3-3。

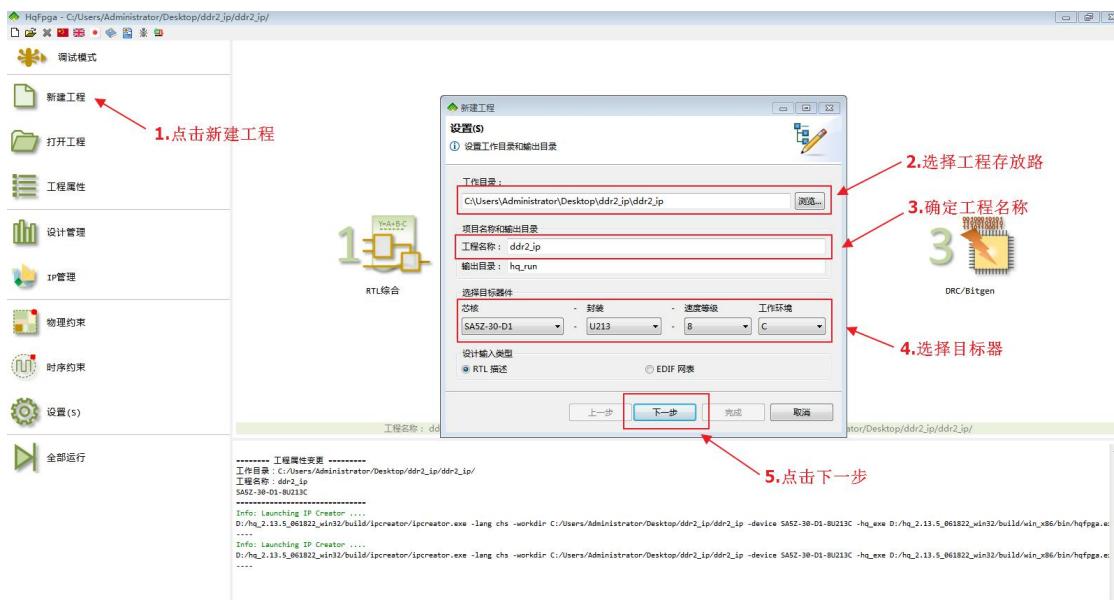


图 3-2 建立工程



图 3-3 添加工程源文件

其次，点击 IP 管理按钮，选择 DDR2\_CONTROLLER，如图 3-4 所示。

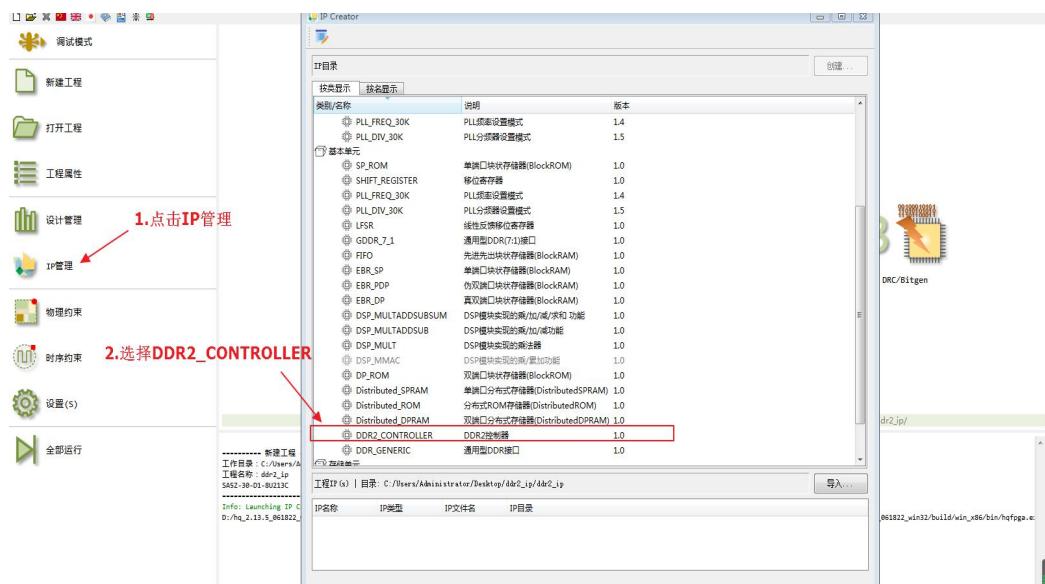


图 3-4 打开 DDR2 控制器 IP 配置界面

然后，用户就可以看到 DDR2 控制器 IP 的配置界面，如图 3-5 所示，其中配置信息包含时钟主频、突发模式、RRT 值和地址序列等配置项。时钟主频为 DDR2 MEMORY 工作频率，用户可以根据自己的设计需求设置不同的频率输入，但是需要确保 DDR2 CONTROLLER 输入接口 phy\_io\_clk 的频率与设置频率保持一致；突发模式可设置为 4 突发 Burst4 和 8 突发 Burst8 两种模式，控制连续性数据传输长度；RRT 值可配置为 0、50、75 和 150，分别代表不同的终端匹配电阻阻值；地址序列可以配置成 Bank\_Row\_Column 和 Row\_Bank\_Column 两种不同选项，需要注意的是 XIST 在设计 DDR2 CONTROLLER 之初就是为了尽可能多的匹配不同容量的 DDR2 MEMORY 器件，所以 DDR2 CONTROLLER 输入接口 user\_addr 一共 30bit，其中 Bank 占 3bit，Row 占 16bit，Column 占 11bit，但是 Seal 5000 系列中的 SA5Z-30-D1 内置了 128Mbit DDR2 MEMORY 器件，所以实际 Bank 占 2bit，Row 占 12bit，Column 占 9bit。如图 3-6 所示，用户在使用 SA5Z-30-D1 器件时，无需设置 Bank，Row，Column 位宽，但是需要在 user\_addr 相应位上补 0，如表 3-2 所示，其中 user\_addr\_in 是用户输入的地址。

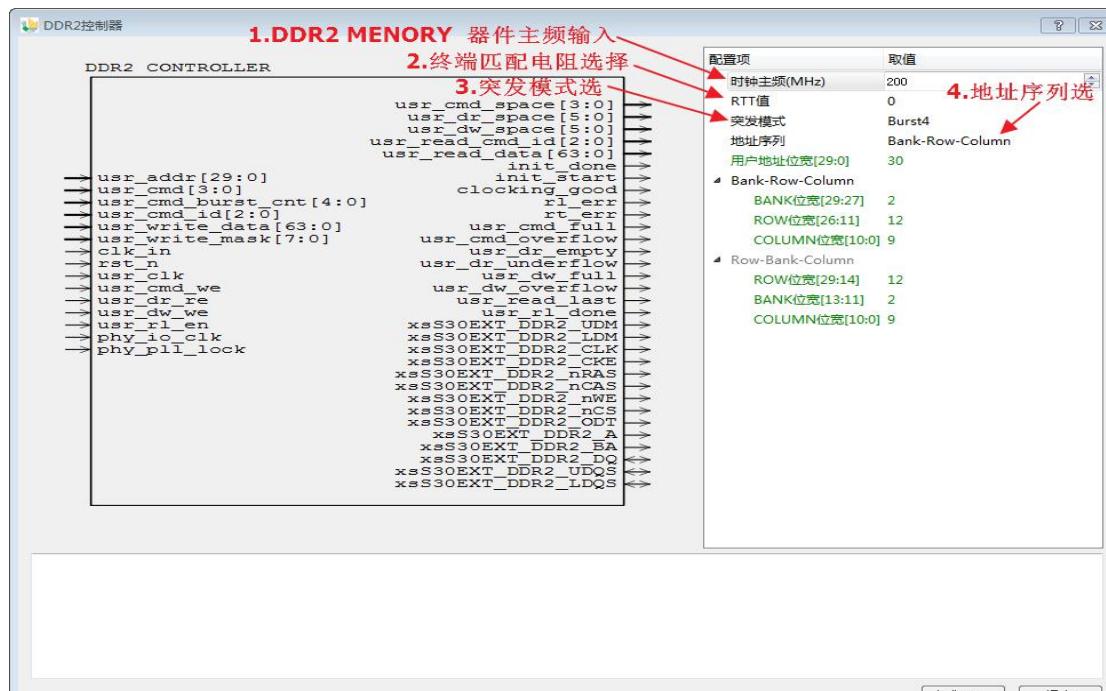


图 3-5 DDR2 控制器 IP 配置界面



图 3-6 IP 核实际支持地址宽度



表 3-2 Bank\_Row\_Column 地址顺序及格式

| Bank_Row_Column 顺序 | user_addr_in 地址格式  |
|--------------------|--|
| Bank_Row_Column    | {1'b0 ,user_addr_in[22:21] ,4'b0000 ,user_addr_in[20: 9] ,2'b00 ,user_addr_in[8:0] } |
| Row_Bank_Column    | {4'b0000 ,user_addr_in[22:11] ,1'b0 ,user_addr_in[10: 9] ,2'b00 ,user_addr_in[8:0] } |

最后，点击生成 IP 按钮，确定 DDR2 控制器 IP 的文件名称以及文件输出路径后，再点击保存按钮，如图 3-7 所示，用户就可以在相应的路径下找到 DDR2 控制器 IP 以及 DDR2 控制器 IP 使用参考例程，如图 3-8 所示。

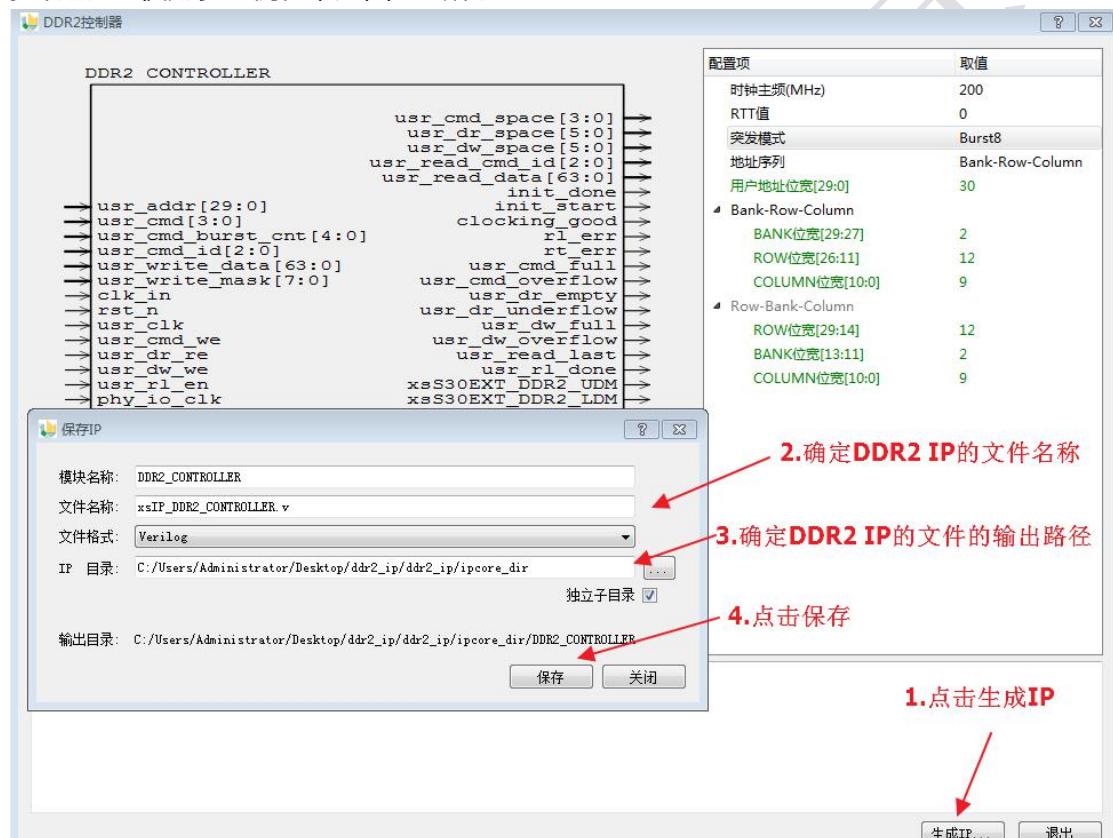


图 3-7 DDR2 控制器 IP 生成



图 3-8 DDR2 控制器 IP 存放路径

XIST 为用户提供了 DDR2 控制器 IP 使用的参考例程 example，用户在生成 DDR2 控



制器 IP 时 , HQ-FPGA 会默认将 example 存放在 DDR2 控制器 IP 生成的路径 , 如图 3-8 所示 , example 中包含了 DDR2 控制器 IP、用户参考设计、IO 物理约束以及时序约束文件 , 如图 3-9 所示。用户可以参考 example 例程来例化 DDR2 控制器 IP 完成自己的设计 , 至此已经介绍完了 DDR2 IP 核的生成步骤。



图 3-9 DDR2 控制器 IP 参考例程介绍

需要注意的是 , 在 SEAL 5000 系列 SA5Z-30-D1 、 SA5Z-30-D2 器件内部 ,FPGA 与 DDR2 memory 器件通过硬连线相连(Hard-wired) , DDR2 的端口(port)没有对外引出 , 用户需要通过访问 FPGA 的相关端口与 DDR2 memory 器件进行数据传输并对其进行控制。在用户设计描述中 , 用户可以在其顶层模块中加入特殊命名的端口来访问 DDR2 memory 器件 , 如表 3-3 。

表 3-3 DDR2 Controller 特殊端口说明

| 特殊端口名称              | 信号说明    | 方向 |
|---------------------|---------|----|
| xsS30EXT_DDR2_CLK   | CLK     |    |
| xsS30EXT_DDR2_nCLK  | nCLK    | 输出 |
| xsS30EXT_DDR2_nCS   | nCS     | 输出 |
| xsS30EXT_DDR2_CKE   | CKE     | 输出 |
| xsS30EXT_DDR2_A     | A[12:0] | 输出 |
| xsS30EXT_DDR2_BA    | BA[1:0] | 输出 |
| xsS30EXT_DDR2_nRAS  | nRAS    | 输出 |
| xsS30EXT_DDR2_nCAS  | nCAS    | 输出 |
| xsS30EXT_DDR2_nWE   | nWE     | 输出 |
| xsS30EXT_DDR2_LDM   | LDM     |    |
| xsS30EXT_DDR2_UDM   | UDM     | 输出 |
| xsS30EXT_DDR2_LDQS  | LDQS    |    |
| xsS30EXT_DDR2_nLDQS | nLDQS   |    |
| xsS30EXT_DDR2_UDQS  | UDQS    |    |
| xsS30EXT_DDR2_nUDQS | nUDQS   |    |
| xsS30EXT_DDR2_DQ    | DQ      | 双向 |
| xsS30EXT_DDR2_ODT   | ODT     | 输出 |



用户可以参照表 3-3，来在自己设计工程的顶层来定义 DDR2 MEMORY 相关的端口。

```
module my_design (
    myclk, mydatain, mydataout.... // 正常顶层设计端口声明
    // 用于访问 DDR2-SDRAM 的端口声明
    xsS30EXT_DDR2_UDM,
    xsS30EXT_DDR2_LDM,
    xsS30EXT_DDR2_CLK,
    xsS30EXT_DDR2_CKE,
    xsS30EXT_DDR2_nRAS,
    xsS30EXT_DDR2_nCAS,
    xsS30EXT_DDR2_nWE,
    xsS30EXT_DDR2_nCS,
    xsS30EXT_DDR2_ODT,
    xsS30EXT_DDR2_A0,
    xsS30EXT_DDR2_A1,
    xsS30EXT_DDR2_A2,
    xsS30EXT_DDR2_A3,
    xsS30EXT_DDR2_A4,
    xsS30EXT_DDR2_A5,
    xsS30EXT_DDR2_A6,
    xsS30EXT_DDR2_A7,
    xsS30EXT_DDR2_A8,
    xsS30EXT_DDR2_A9,
    xsS30EXT_DDR2_A10,
    xsS30EXT_DDR2_A11,
    xsS30EXT_DDR2_A12,
    xsS30EXT_DDR2_BA0,
    xsS30EXT_DDR2_BA1,
    xsS30EXT_DDR2_DQ0,
    xsS30EXT_DDR2_DQ1,
    xsS30EXT_DDR2_DQ2,
    xsS30EXT_DDR2_DQ3,
    xsS30EXT_DDR2_DQ4,
    xsS30EXT_DDR2_DQ5,
    xsS30EXT_DDR2_DQ6,
    xsS30EXT_DDR2_DQ7,
    xsS30EXT_DDR2_DQ8,
    xsS30EXT_DDR2_DQ9,
    xsS30EXT_DDR2_DQ10,
    xsS30EXT_DDR2_DQ11,
    xsS30EXT_DDR2_DQ12,
    xsS30EXT_DDR2_DQ13,
    xsS30EXT_DDR2_DQ14,
    xsS30EXT_DDR2_DQ15,
    xsS30EXT_DDR2_UDQS,
    xsS30EXT_DDR2_LDQS
    //-----);
    ...

    // 访问 DDR2-SDRAM 的端口定义
    output      xsS30EXT_DDR2_UDM;
    output      xsS30EXT_DDR2_LDM;
    output      xsS30EXT_DDR2_CLK;
    output      xsS30EXT_DDR2_CKE;
    output      xsS30EXT_DDR2_nRAS;
    output      xsS30EXT_DDR2_nCAS;
    output      xsS30EXT_DDR2_nWE;
    output      xsS30EXT_DDR2_nCS;
    output      xsS30EXT_DDR2_ODT;
    output      xsS30EXT_DDR2_A0;
    output      xsS30EXT_DDR2_A1;
    output      xsS30EXT_DDR2_A2;
    output      xsS30EXT_DDR2_A3;
    output      xsS30EXT_DDR2_A4;
```



```
output    xsS30EXT_DDR2_A5;
output    xsS30EXT_DDR2_A6;
output    xsS30EXT_DDR2_A7;
output    xsS30EXT_DDR2_A8;
output    xsS30EXT_DDR2_A9;
output    xsS30EXT_DDR2_A10;
output    xsS30EXT_DDR2_A11;
output    xsS30EXT_DDR2_A12;
output    xsS30EXT_DDR2_BA0;
output    xsS30EXT_DDR2_BA1;
inout    xsS30EXT_DDR2_DQ0;
inout    xsS30EXT_DDR2_DQ1;
inout    xsS30EXT_DDR2_DQ2;
inout    xsS30EXT_DDR2_DQ3;
inout    xsS30EXT_DDR2_DQ4;
inout    xsS30EXT_DDR2_DQ5;
inout    xsS30EXT_DDR2_DQ6;
inout    xsS30EXT_DDR2_DQ7;
inout    xsS30EXT_DDR2_DQ8;
inout    xsS30EXT_DDR2_DQ9;
inout    xsS30EXT_DDR2_DQ10;
inout    xsS30EXT_DDR2_DQ11;
inout    xsS30EXT_DDR2_DQ12;
inout    xsS30EXT_DDR2_DQ13;
inout    xsS30EXT_DDR2_DQ14;
inout    xsS30EXT_DDR2_DQ15;
inout    xsS30EXT_DDR2_UDQS;
inout    xsS30EXT_DDR2_LDQS;

...

```

图 3-10 通过特殊顶层模块端口访问 DDR2 的示例

本方法的优点是：首先，设计描述与实际硬件有直接的对应关系，方便理解。其次，没有特别的描述方式限制，对第三方综合工具比较友好。由于 DDR2 端口是顶层模块端口，第三方工具可以按需插入单向或双向 IO 缓冲器，或者 IO 寄存器，HQFPGA 开发软件最终会将这些特殊端口映射到硬件端口实现。