

## SDR SDRAM 的上电初始化原理与控制电路的设计实现

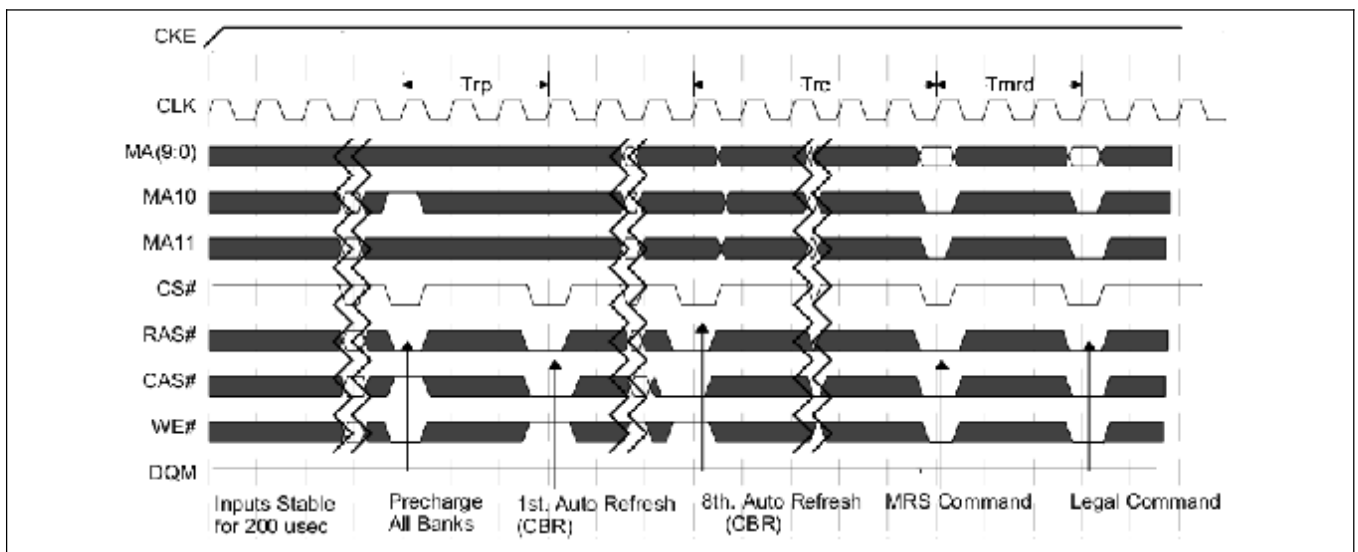
### 1. SDR SDRAM 的上电初始化原理

SDR SDRAM 必须要用规定的方式进行上电与初始化，以保证器件正常工作。同时，对模式寄存器的设置也发生在初始化期间。

#### 1.1 Jedec 21-C 标准与 Intel PC100 标准的规定

早期的 SDRAM、当代的韩系、日系、台系 SDRAM 都符合这里的描述。为保证兼容性，本次设计采用这个传统初始化标准，但 tMRD 周期取为 2 以匹配当代的器件。

下图为 1999 年 Intel 定义的初始化时序图，取自 Intel PC100 标准：



上电步骤：

1. 同时施加 VDD 与 VDDQ，并把 CKE 置为低。
2. 等待电源稳定。
3. 等待时钟稳定后把 CKE 置为高。

初始化步骤：

1. 保持 200us 的 NOP 指令发送。
2. 发送 Precharge All 指令。
3. 发送至少 8 次 Refresh 指令。
4. 发送 Mode register set 指令，设置模式寄存器。

备注：

1. 初始化时不可忽视交流时序参数(trp、trfc、tmrs)，必要时要添加 NOP 指令。
2. 当代器件一般规定 tMRS 为 2 个时钟周期，而不是 Jedec 或 PC100 中定义的 3 个周期。

3. 对于 Samsung、Zentel 的器件，可以只 Refresh 两次。

4. Elpida、Zentel 等公司建议在初始化时把 DQM 置为 1，以防止可能的数据冲突。这一点在 Intel 定义的时序图中也有体现。PC100 规定在第一条有效指令发送后，DQ 自动变为高阻，但这之前的状态未规定。

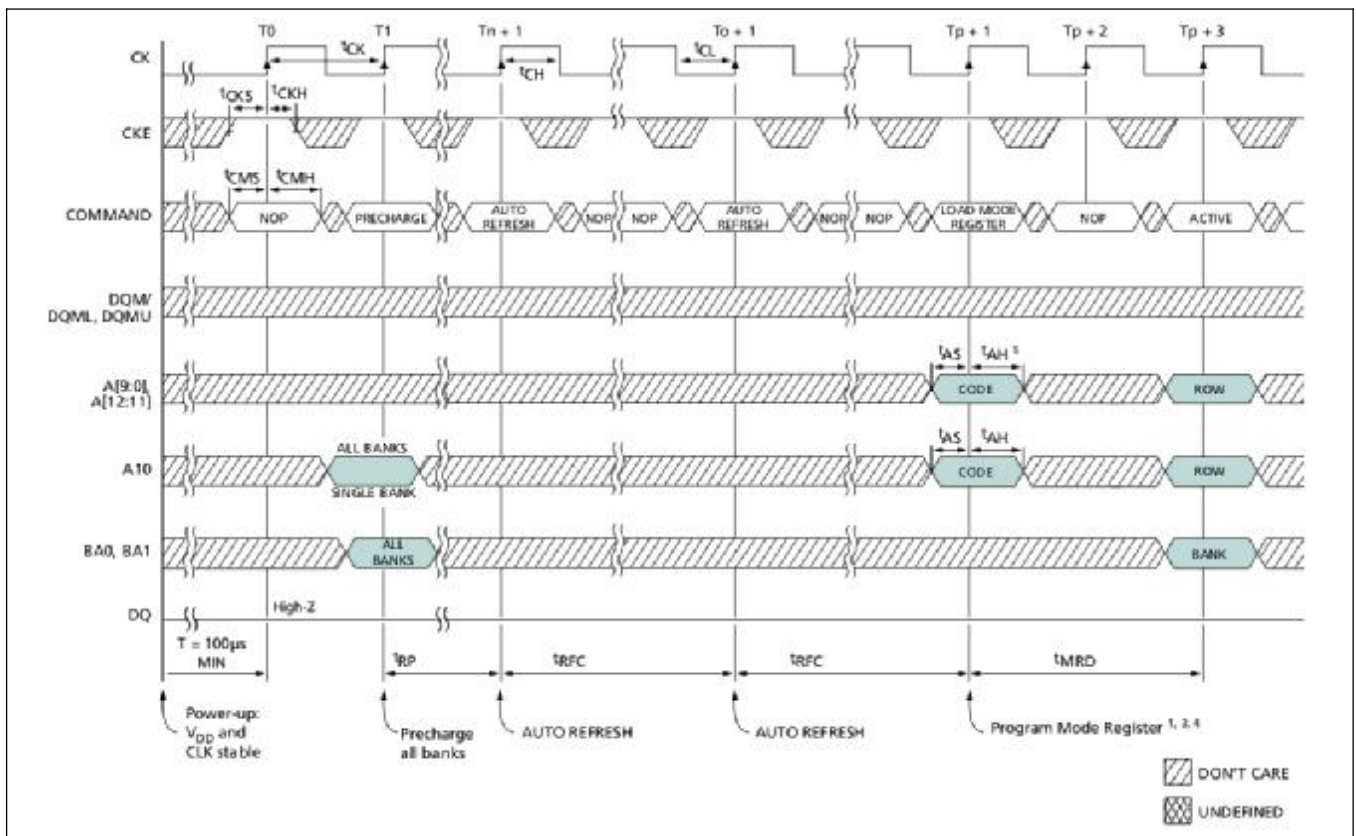
5. 理论上只要所有 Bank 都空闲，就可以设置 Mode register。但一般情况下，初始化后 Mode register 不再需要改变。Mode register 中设置了 CAS Latency、Burst Length 等重要参数，这些通过数据通过地址线传送，详细过程将在后面的章节中描述。

## 1.2 Micron、ISSI 等当代美系 SDR SDRAM 的改进

Micron、ISSI 等公司的 SDRAM 产品中，时钟稳定后的初始等待时间缩短为 100us，同时 refresh 的次数减少为至少 2 次。对于 Micron 器件，时钟稳定后，CKE 可以在 100us 的等待期间上升，然后保持 NOP 发送；而对于 ISSI 器件，时钟稳定后，CKE、DQM 必须上升为 1，然后才开始 100us 的 NOP 发送。

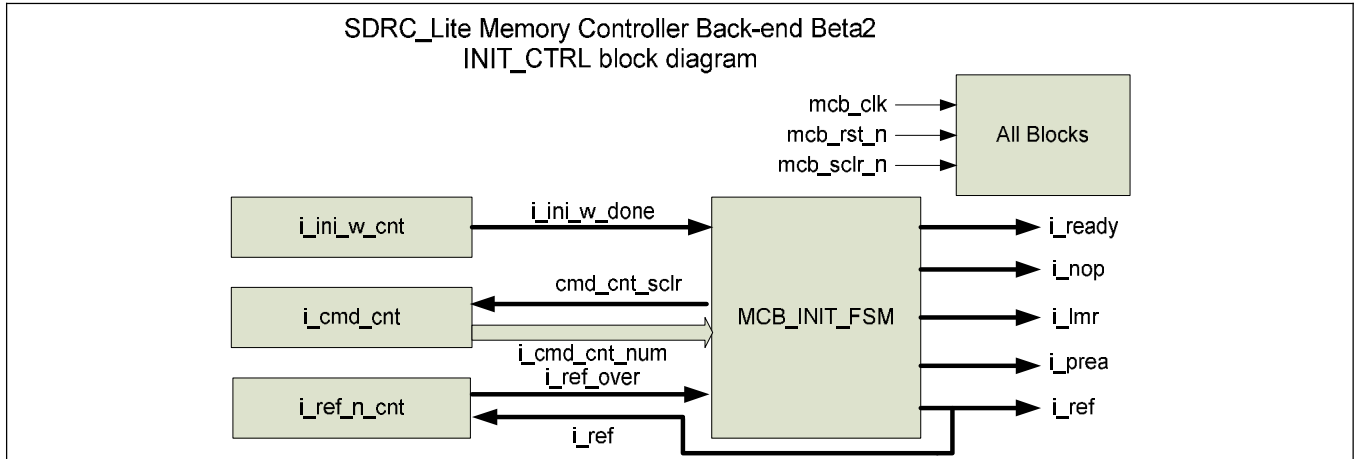
Micron 为 SDRAM 做的 verilog 模型流传甚广，Xilinx、Altera、Lattice 等 FPGA 厂商提供的参考设计一般都遵循 Micron 手册中的初始化步骤。

下图为 Micron SDRAM 的初始化时序图，取自 Micron 的器件手册：



## 2. 上电初始化控制电路(MCB\_INI\_CTRL)的设计概要

本次设计使用的初始化控制电路，包括初始化状态机(MCB\_INI\_FSM)、初始化等待计数器(i\_ini\_w\_cnt)、初始化 refresh 次数计数器(i\_ref\_n\_cnt)、初始化命令延时计数器(i\_cmd\_cnt)四个模块。它们的连接关系如下图：



初始化等待计数器(i\_ini\_w\_cnt)、初始化 refresh 次数计数器(i\_ref\_n\_cnt)、初始化命令延时计数器(i\_cmd\_cnt)，与初始化状态机协同工作，共同完成初始化的控制。初始化等待计数器控制 100us 或 200us 的初始化等待；初始化 refresh 次数计数器控制 refresh 的次数；初始化命令延时计数器保证 trp、trfc、tcmd 等指令延时参数被满足。用户可以通过 verilog 参数改变这三个计数器的设置。

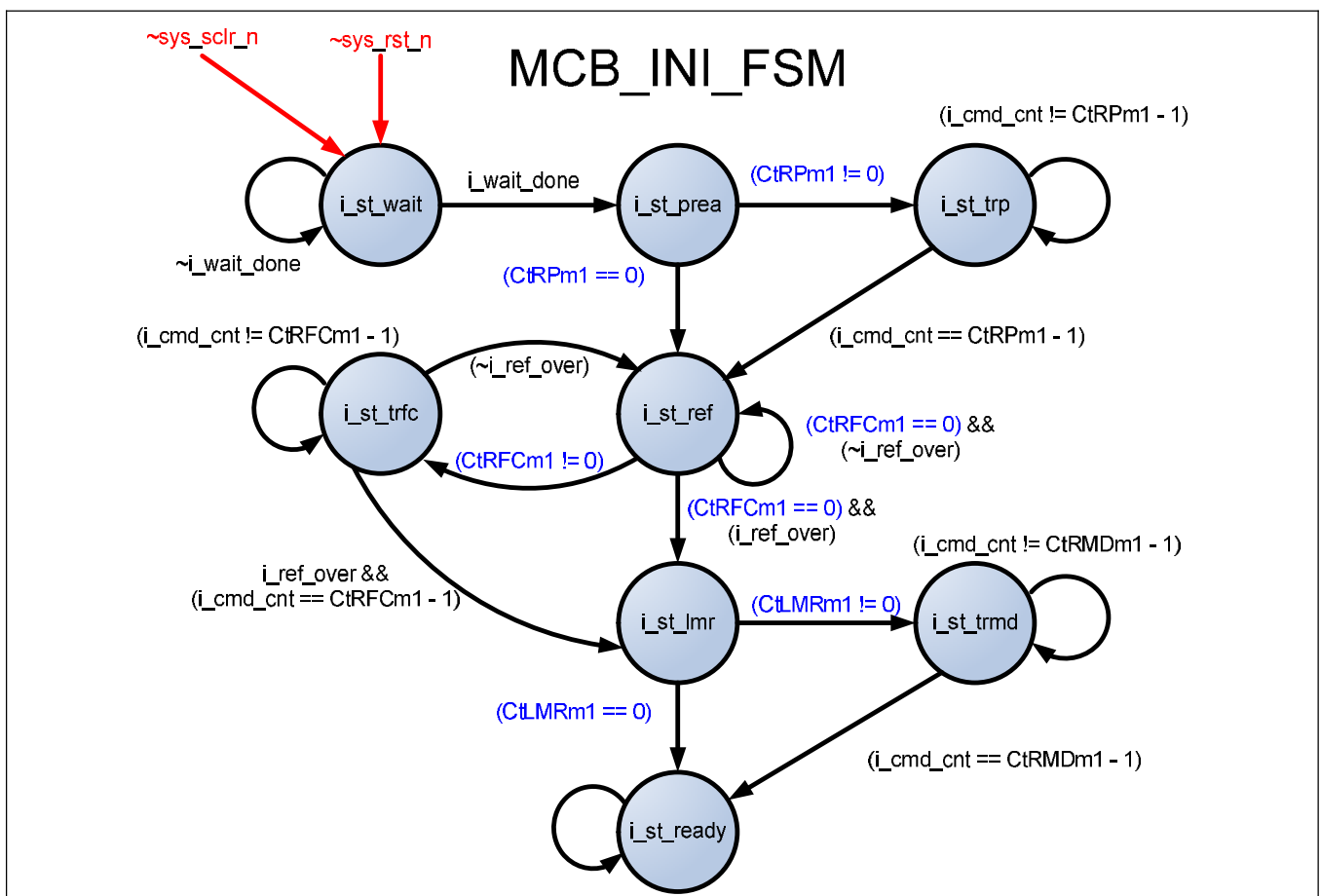
初始化状态机(MCB\_INI\_FSM)采用 Moore 模型，输出只与当前状态有关，以保证输出控制信号与时钟同步。状态机代码风格将采用 Altera 推荐的 2 段式写法，时序的状态更新为一段，组合的 Next State Logic 与 Output Logic 为一段。初始化状态机的状态跳转顺序基本按照 Jedec 的 SDRAM 初始化规定。此处假设在系统复位后，电源与时钟信号已经稳定，状态机主要控制初始化部分而非上电部分。各个状态与对应的输出如下表：

状态名	输出的指令信号	输出的其他控制信号
i_st_nop	(延时状态)	
i_st_prea	i_prea	i_cmd_cnt_sclr
i_st_trp	(延时状态)	
i_st_ref	i_ref	i_cmd_cnt_sclr
i_st_trfc	(延时状态)	
i_st_lmr	i_lmr	i_cmd_cmr_sclr
i_st_tcmd	(延时状态)	
i_st_ready	(初始化完成状态)	i_ready

$trp$ 、 $trfc$ 、 $tcmd$  分别是 Precharge、Refresh、Mode Register Set 后的延时。在此期间不能发送 NOP 以外的指令。按照器件手册的规定， $trp$ 、 $trfc$ 、 $tcmd$  的单位从 ns 折算为时钟周期时，应该向上取整。假设  $Trp$  折合为  $x$  个时钟周期，那么从  $i\_st\_prea$  进入  $i\_st\_trp$  后，就应停留  $x-1$  个时钟周期（因为  $i\_st\_prea$  自身也占用了周期）。如果  $x$  恰好为 1，则  $i\_st\_trp$  状态不存在，电路参数决定了， $x$  的大小就决定了，是否可能进入  $i\_st\_trp$  只与电路参数有关，这个判断不会映射为实体逻辑电路，在下文中的状态转换图中用蓝色条件标出。 $trfc$ 、 $tcmd$  对状态跳转的影响同理于  $trp$ 。

在系统复位时，进入 IDEL 状态（即  $i\_st\_nop$ ）；完成 100us 或 200us 的初始化等待后，进行 Precharge All；满足  $trp$  延时后，进行 refresh；满足  $trfc$  延时后，如果达到需要的 refresh 次数，就进行 mode register set，否则，继续 refresh；mode register set 后，经过  $tcmd$  延时，最终进入 ready 状态，初始化完成。

所有的延时等待状态中，都不发指令信号； $i\_st\_prea$  中发起  $i\_prea$ ； $i\_st\_ref$  中发起  $i\_ref$ ； $i\_st\_lmd$  中发起  $i\_lmr$ ； $i\_prea$ 、 $i\_ref$ 、 $i\_lmr$  信号将通过信号通路 (MCB\_SIG\_FF) 编码为 SDRAM 命令。初始化完成后，发出  $i\_ready$ ，这之后用户的请求才会被响应。初始化状态机的状态转换图如下：



注：上图中，蓝色的条件与器件的时序和系统工作频率参数有关，不会映射为电路逻辑。

## A1 版本历史:

版本	时间	说明
SDRC_lite spec alpha2	2012/03/30	参照 SDRAM 的业界标准与 alpha2 版初始化单元的设计实现编写。
SDRC_lite spec beta2.1	2012/04/14	包含 beta2.1 版设计的一些更新; verilog 仿真、FPGA 综合部分将另外在设计实现文档中描述。

## A2 参考文献:

- [1] Intel, PC SDRAM Specification, Revision 1.7, November 1999. (Intel PC100)
- [2] Jedec, Jedec standard No.21-C. (Jedec SDRAM standard)
- [3] TI, TMS626812 datasheet, 1997. (早期的 SDRAM, 只有两个 bank)
- [4] Micron, 64Mb x4, x8, x16 SDR SDRAM Features, Rev. Q, February 2012.
- [5] ISSI, 1 Meg Bits x 16 Bits x 4 Banks (64-MBIT) SYNCHRONOUS DYNAMIC RAM, Rev. I, Dec. 2011.
- [6] Elpida, User's manual HOW TO USE SDRAM, rev 8.1, March 2009.
- [7] Hynix, SDRAM DEVICE OPERATION, rev1.1, sep. 2003.
- [8] Samsung, CMOS SDRAM Device Operations, rev 0.2, 1999.
- [9] windbond, w9864G6JT 1M \* 4 BANKS \* 16 BITS SDRAM, Revision A01, dec. 2011.
- [10] Nanya, NT5SV32M8CS NT5SV16M16CS 256Mb Synchronous DRAM, REV 1.4, dec.2011
- [11] Xilinx, xapp134, Synthesizable High Performance SDRAM Controller, 2000.
- [12] Altera, white paper SDR SDRAM Controller, version 1.1, February 2001.
- [13] Lattice, RD1010, SDR SDRAM Controller, rev 04.6, April 2011.
- [14] Zentel, A3V64S40FTP 64Mb Synchronous DRAM Specification, rev 1.0, Aug. 2011.
- [15] Altera, Quartus II Handbook, Ch10, Recommended HDL Coding Styles, rev 10.1, Dec. 2011.
- [16] Xilinx, UG627 XST user guide, Chapter 03:XST HDL Coding Techniques, rev 12.4, Dec. 2010.
- [17] Pong P. Chu, FPGA Prototyping Using Verilog Examples, Wiley-Interscience, June 2008.