

1. Overview

手里有DDR4的板子，但是在网上没有找到合适的开源仿真模型，目前全网可靠的开源仿真模型只有MT48LC64M4A2，这是镁光的SDRAM，所以决定从该仿真模型开始，之后直接在板子上实现DDR4读写访问。

主要打算分以下几块内容来学习：

- i. 结合JEDEC标准协议介绍 DDR3/4 引脚、各类寄存器以及相关操作:
  - 1) 初始化
  - 2) 自刷新
  - 3) 预充电
  - 4) 寄存器访问
  - 5) 读写
- ii. 手撸能够实现读写SDRAM简单功能的Controller
- iii. CTR, DFI, PHY三层结构介绍
- iv. FPGA MIG核使用
- v. Xilinx MIG 仿真EXAMPLE介绍
- vi. LPDDR4介绍？

2. DDR3和DDR4在协议上主要区别：

- 1. 在DDR4的RAS/CAS/WE与A16/A15/A4复用端口中，通过ACTN进行区分，而在DDR3中不存在这个概念。
- 2. DDR3的IO电压为1.5V，而DDR4的IO电压为1.2V。
- 3. DDR3只有3个MR寄存器，而DDR4有6个MR寄存器。
- 4. DDR3不可读颗粒信息，而DDR4可以读出MR的基本信息。
- 5. DDR3的读训练MPR只有一种pattern，且不可写。而DDR4的读训练MPR有4种pattern，分三种输出形式，可写。
- 6. DDR4支持CS和CMD/ADDR之间相差tCAL延迟，用于关闭接收电路，节省功耗。而DDR3在这方面则有限制。

3. Pinout Description

Symbol	Type	Function
CK, CK#	Input	时钟CK和ck#是差分时钟输入。所有的地址和控制输入信号都在CK的正沿和ck#的负沿的交叉处采样。
CKE, (CKE0), (CKE1)	Input	时钟使能CKE高有效，拉高时激活内部时钟信号和设备输入BUFFER和输出驱动。 CKE拉低时提供预充电断电和自刷新操作(所有Bank空闲)，或主动下电(任意Bank的已激活行)。 对于自刷新退出操作，CKE是异步的。 VREFCA和VREFDQ在上电和初始化顺序中稳定后，在所有操作过程中都需要保持稳定(包括Self-Refresh)。 <b>CKE必须在整个读写访问过程中保持高位。</b> Input buffers, excluding CK, CK#, ODT and CKE, are disabled during power-down. Input buffers, excluding CKE, are disabled during Self-Refresh.
CS#, (CS0#), (CS1#), (CS2#), (CS3#)	Input	芯片选择信号:当CS#拉高时，所有命令都被屏蔽。 CS#在多RANK系统中被用于选择外部Rank。 <b>CS#也是命令代码的一部分。</b>
ODT, (ODT0), (ODT1)	Input	On Die Termination: Die指的就是DDR的颗粒芯片。 在DDR中，ODT (On-Die Termination) 的作用是将端接电阻移植到芯片内部，这个功能只在DDR2以上的数据信号才有。有ODT功能，原本需要在PCB板上加串联电阻的数据信号就不需要再额外添加端接，只需要芯片内部打开ODT的端接功能，且这个端接可调。ODT的主要目的是减少信号反射，提高信号品质，从而产生更高的内存时钟频率速度。将终结电阻设计在内存芯片之上还可以简化主板的设计，降低主板的成本，同时终结电阻器可以和内存颗粒的“特性”相符，从而减少内存与主板的兼容问题的出现。一根内存条往往会使用多片DDR颗粒，在级联的过程中数据信号（电子）能量没有消耗殆尽，所以在终端会往前端回流产生震荡/反射。在时钟频率越来越高的情况下，会导致信号不稳定，所以以前的方案是在级联终端加上串联电阻。 在ODT拉高时表示使用DDR内部端接电阻。 启用后，ODT仅适用于每个DQ、DQS、DQS#和DM/TDQS，NU/TDQS#(当TDQS通过MR1中的模式寄存器A11=1使能时)信号用于x4/x8配置。对于x16配置，ODT应用于每个DQ、DQSU、DQSU#、DQSL、DQSL#、DMU和DML信号。如果MR1和MR2被设置为禁用RTT，那么ODT引脚将被忽略。
RAS#, CAS#, WE#	Input	命令输入:RAS#，CAS#和WE#(以及CS#)组合后定义要输入的命令。
DM, (DMU), (DML)	Input	Data Mask: DM是用于写入数据的输入掩码信号。在写访问期间，当DM被采样为高时，输入数据被屏蔽。DM在DQS的上下沿采样。对于x8设备，DM或TDQS/TDQS#功能通过模式寄存器使能MR1中的A11来设置。
BA0 - BA2	Input	Bank地址: BA0-BA2定义具体正在进行激活，读写访问，预充电命令的Bank地址。Bank地址还决定了在MRS周期中访问哪种模式寄存器。
A0 - A15	Input	地址输入:为Active命令提供行地址，为读写访问提供列地址从各bank的内存数组中选择一个位置。(A10/AP和A12/ bc#具有附加功能:见下文)。地址输入还提供模式寄存器集命令期间的操作码。
A10 / AP	Input	自动预充电:在读/写命令期间采样A10以确定是否应在读/写操作后对被访问Bank进行自动充电。(高:Autoprecharge;LOW:无自动充电功能)。 在Precharge命令期间对A10进行采样，以确定Precharge是否适用于某一个Bank(A10 LOW)或所有Bank(A10 HIGH)。如果只有一个Bank要充电，则按Bank地址进行选择。
A12 / BC#	Input	Burst Chop: 在读写命令期间对A12 / BC#进行采样，以确定是否削减突发长度(on_the_fly模式)。(低有效)。详细信息请参见命令真值表。
RESET#	Input	低有效，异步复位。
DQ	InOut	双向数据总线
DQS	InOut	Data Strobe:读数据时为输出并边沿对齐，写数据时为输入并中心对齐。DDR3 SDRAM只支持差分，不支持单端。 DQS的主要作用包括数据同步、数据定时以及数据的可靠性传输。 在DDR3内存控制器和内存模块之间的数据交互过程中，DQS信号的生成和处理是一项重要的交互过程。在数据发送过程中，内存控制器通过发送DQS信号来定时同步传输数据的时钟。而在数据接收过程中，内存模块的DQS接口用于提供时钟信号，以使数据能够正确地读取并进行进一步的处理。 DQS的另一重要功能是数据选取。在读取和写入数据时，DQS用于在每个数据位之间提供一个同步标记，以确保数据传输的准确性和可靠性。
TDQS, TDQS#	Output	Termination Data Strobe: TDQS/TDQS# is applicable for x8 DRAMs only. When enabled via Mode Register A11 = 1 in MR1, the DRAM will enable the same termination resistance function on TDQS/TDQS# that is applied to DQS/DQS#. When disabled via mode register A11 = 0 in MR1, DM/TDQS will provide the data mask function and TDQS# is not used. x4/x16 DRAMs must disable the TDQS function via mode register A11 = 0 in MR1.

4. DDR3 SDRAM Addressing

1. 512Mb

Configuration	128Mb x 4	64Mb x 8	32Mb x 16
# of Banks	8	8	8
Bank Address	BA0 - BA2	BA0 - BA2	BA0 - BA2
Auto precharge	A10/AP	A10/AP	A10/AP
BC switch on the fly	A12/BC#	A12/BC#	A12/BC#
Row Address	A0 - A12	A0 - A12	A0 - A11
Column Address	A0 - A9,A11	A0 - A9	A0 - A9
Page size <sup>1</sup>	1 KB	1 KB	2 KB

i.

ii. 这边仅以64Mb\*8为例说明：列地址10bits数据位宽，所以有2^10个列地址（1024即1K），行地址13bits数据位宽，有2^13个行地址（2^13=8192,8192/1024=8K），Bank地

址为3bis有8个Bank，所以共计1K\*8K\*8=64Mb，数据位宽为8bis，所以是64Mb\*8=512Mb。

## 2. Page Size

页大小是指在执行激活命令时（Active Command）从存储阵列传送到DDR内部感测放大器的数据字节数。每个Bank的页面大小，计算如下：

page size = 2 ^ COLBITS \* ORG ÷ 8

COLBITS = the number of column address bits

ORG = the number of I/O (DQ) bits

## 3. Functional Description

### 3.1简化状态图

这个简化的状态图旨在提供一些常见的状态转换和控制它们的命令概述。不包括涉及多个bank的情况、启用或禁用ODT以及一些其他事件的详细操作。

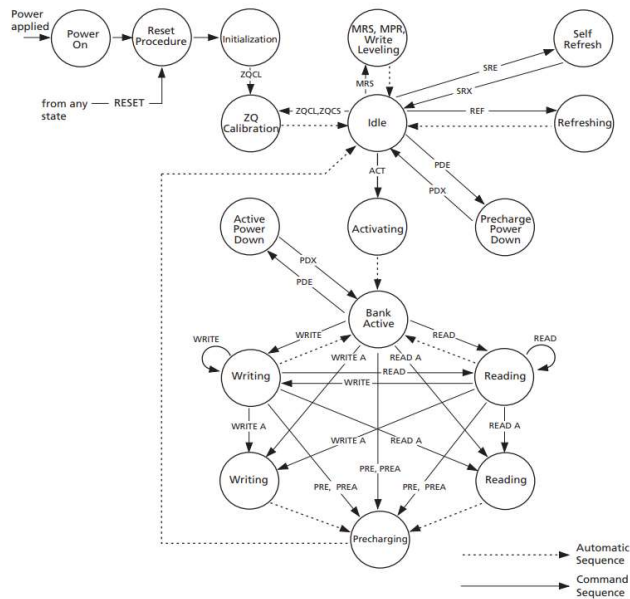


Figure 4 — Simplified State Diagram

状态图简写命令对应表

Abbreviation	Function	Abbreviation	Function	Abbreviation	Function
ACT	Active	Read	RD, RDS4, RDS8	PDE	Enter Power-down
PRE	Precharge	Read A	RDA, RDAS4, RDAS8	PDX	Exit Power-down
PREA	Precharge All	Write	WR, WRS4, WRS8	SRE	Self-Refresh entry
MRS	Mode Register Set	Write A	WRA, WRAS4, WRAS8	SRX	Self-Refresh exit
REF	Refresh	RESET	Start RESET Procedure	MPR	Multi-Purpose Register
ZQCL	ZQ Calibration Long	ZQCS	ZQ Calibration Short	-	-

**NOTE:** See “Command Truth Table” on page 33 for more details.

ZQ Calibration简介

DDR3中的ZQ Calibration是指一种校准过程，用于确保DDR3 SDRAM的稳定性和可靠性。ZQ Calibration的目的是在校准DDR3 SDRAM的终结电阻，以确保数据传输的稳定性和可靠性。

具体流程如下：

a) ZQ CL (ZQ Calibration Long) 命令被最常用于初始系统上电或器件处于复位状态。

1. 命令接收阶段：当收到ZQ校准命令后，校准控制模块开始启动校准程序。

2. 校准阶段：在ZQ校准过程中，控制模块通过调整VOH[0:4]，来控制P Channel device的导通状态，进而改变终结电阻的阻值。同时，比较VPULL-UP和VDDQ/2的电压，当二者相等时，DQ上下两侧的电阻相等，均为240Ω，此时校准完成。

3. 数据处理与存储阶段：校准完成后，将记录下每个电阻对应的VOH[0:4]的值，这些数据将被用于后续的数据处理和存储。

b) DDR3 SDRAM有一个引脚需要外接240R电阻，这就是ZQ校准引脚。这个引脚与ODT (On-Die Termination) 相连，用于控制ODT。

c) ODT是一种动态片上端接，通过使用新的合并驱动器 (Merged Driver) 来降低电容并增强性能。合并驱动器实际上是一种使用内部电阻 (MOS实现) 阵列来控制内部阵列电阻值改变的方法，最终达到改变信号线路驱动能力的目的，从而获得更好的性能。

d) 终结电阻的设定是通过控制ODT来实现的，这样可以使DDR3 SDRAM的数据传输更加稳定和可靠。

### 3.2基本功能

DDR3 SDRAM是一种高速动态随机存取存储器，内部配置为8组DRAM，DDR3 SDRAM采用8n预取架构，实现高速运行。8n预取架构与一个特殊设计的接口相结合，该接口设计为在I/O引脚上每个时钟周期传输两个数据字 (two data words)。

DDR3 SDRAM的读写操作都是面向突发传输的 (burst)，从一个选定的位置开始，并在一个可配置的顺序中继续进行8或4 (上文A12 / BC#提到的突发削减模式) 的burst length。读写访问开始于一次激活命令 (Active command)，之后跟着读/写命令。地址信息与激活命令同时产生，以确定选择对应的Bank与行地址 (BA0-BA2选择Bank; A0-A15选择Row)。Address确定了突发传输操作起始位置，并通过A10确定是否要发出自动预充命令，如果在模式寄存器中启用Onthefly模式的话，还可以通过A12选择BC4或BL8的突发长度。

DDR3 SDRAM在正常工作前，必须先上电并按预先设定的方式进行初始化。

下面的部分提供了详细的信息，包括设备复位和初始化、寄存器定义、命令描述和设备操作。

## 4. 初始化SDRAM

### i. 基于MT48LC64M4A2仿真模型

1) 这是镁光公司20多年前的256Mb高速CMOS SDRAM，不是DDR3 SDRAM。但是操作流程更加方便用于学习。

### ii. 仿真环境

1) 基于win11下iverilog+gtkwave仿真环境

2) 安装方式参考<https://zhuanlan.zhihu.com/p/95081329>

### iii. MT48LC64M4A2命令真值表

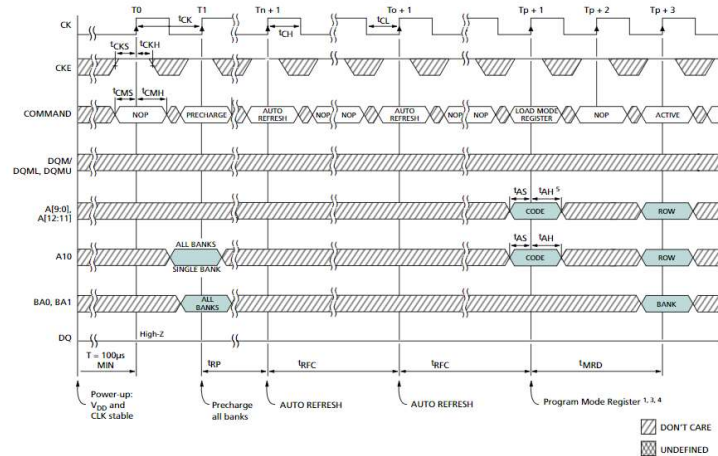
由前面内容可知Command命令由CS#, RAS#, CAS#, WE#信号组成

Command	Value (CS#, RAS#, CAS#, WE#)
COMMAND INHIBIT (NOP)	CS#拉高
NO OPERATION (NOP)	7
ACTIVE (select bank and activate row)	3
READ (select bank and column, and start READ burst)	5
WRITE (select bank and column, and start WRITE burst)	4
BURST TERMINATE	6
PRECHARGE (Deactivate row in bank or banks)	2
AUTO REFRESH or SELF REFRESH (enter self refresh mode)	1

## iv. 初始化时序

1) Figure

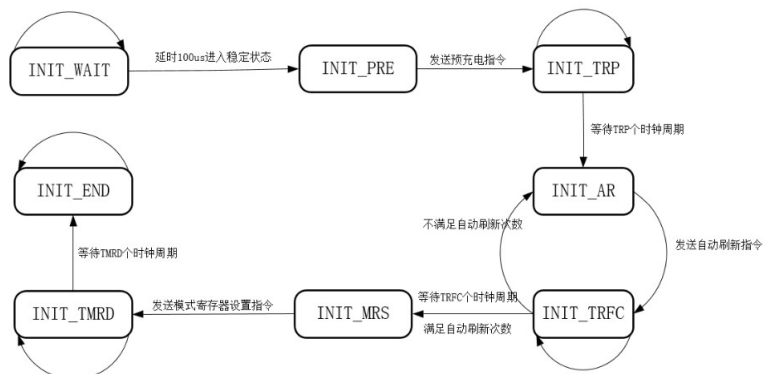
Figure 18: Initialize and Load Mode Register



2) 初始化流程

- 等待上电100us,在此期间拉高CKE, 同时发送NOP命令 (7);
- 对所有Bank进行PRECHARGE操作 (2), A10拉高即为选中所有Bank;
- 等待tPR, 同时发送NOP (7), 仿真中将tPR设为1个CK周期;
- 发送AUTO REFRESH命令 (1);
- 等待tRFC, 同时发送NOP (7), 根据文档可知tRFC(min)=7.813us,tRFC设为1个CK周期; (在文档中搜索tRFC可以在后面的参数表中找到, 步骤c里面的tPR同理)
- 重复a和e, 共计2次, 重复刷新次数根据不同颗粒决定;
- 发送LMR命令 (0), 同时地址线设置为操作码 (后面介绍), 根据地址线不同的值实现不同的寄存器配置;
- 等待tMRD,同时发送NOP (7);
- 初始化操作完成。

3) 初始化状态机



## v. 代码解读

直接看注释, 写的很详细

- tb\_sdram\_init
- sdram\_init
- sdram\_model\_plus

## vi. 操作流程

1) bat脚本

```

::echo "开始编译"
::cd .\Desktop\test
::iverilog -g2005-sv -o wave padup.v tb.sv
::echo "编译完成"
::echo "生成波形文件"
::vvp -n wave -lxt2
::echo "打开波形文件"
::gtkwave wave.vcd
::pause
@echo off
echo "生成filelist"
dir .\*.* /b /s > filelist.f
dir .\*.sv /b /s >> filelist.f
if exist *.gtkw (
    echo "存在gtkw波形文件"
    echo "打开gtkw波形文件"
    gtwave *.gtkw
) else (
    if exist "wave.lxt" (
        echo "存在lxt波形文件"
        echo "打开lxt波形文件"
        gtwave wave.lxt
    ) else (
        echo "无波形文件"
        echo "开始编译"
        iverilog -g2005-sv -o wave.vvp -f filelist.f
        echo "编译完成"
        echo "生成lxt波形文件"
        vvp -n wave.vvp -lxt2
    )
  
```

```
copy wave.vcd wave.lxt
echo "打开lxt波形"
gtkwave wave.lxt
)
)
pause
```

2) 将4个文件放在一个单独的文件夹

名称	修改日期
filelist.f	2023/10/8 16:42
open_wave.bat	2023/10/7 17:17
sdram_init.v	2023/10/8 16:27
sdram_model_plus.v	2023/10/7 17:42
tb_sdram_init.v	2023/10/8 16:42

a)

b) 直接双击bat文件即可生产filelist文件，并开始自动编译，生产波形文件，进行波形转化，使用gtkwave打开波形。

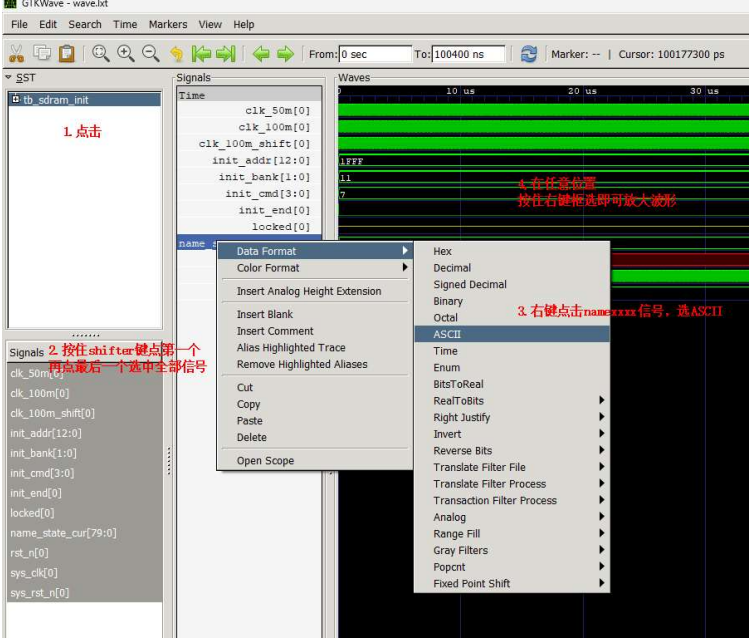
c)

```
"生成filelist"
找不到文件
"无波形文件"
"开始编译"
"编译完成"
"生成lxt波形文件"
LXT2 info: dumpfile wave.vcd opened for output.
at time 100215 ns PRE : Bank = ALL
at time 100235 ns AREF : Auto Refresh
at time 100255 ns AREF : Auto Refresh
at time 100255 ns ERROR: tRC violation during Auto Refresh
at time 100275 ns LMR : Load Mode Register
CAS Latency = 3
Burst Length = Full
Burst Type = Sequential
Write Burst Mode = Programmed Burst Length
at time 100275 ns ERROR: tRC violation during Load Mode Register
已复制 1 个文件。
"打开lxt波形文件"
```

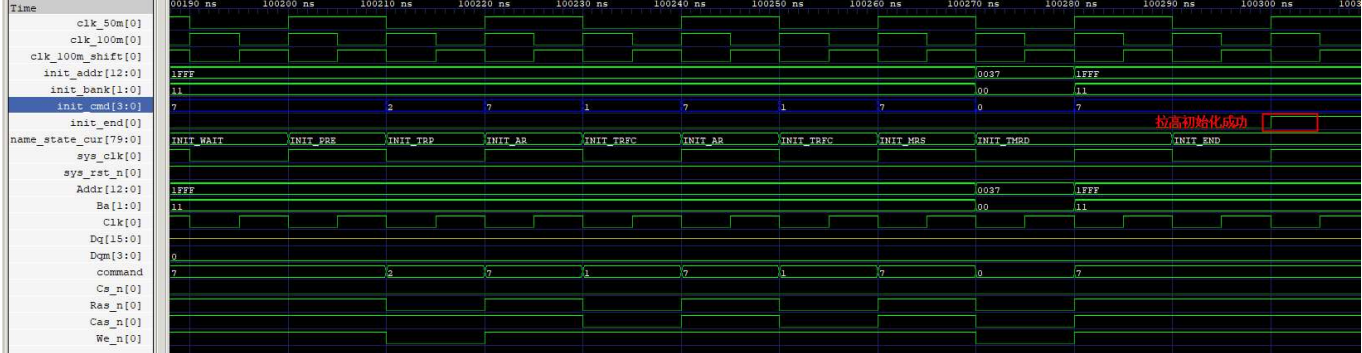
d) 在终端中出现如上log后即表示初始化操作完成，等待后会自动打开wave波形。

vii. 波形对照

操作介绍



拉到最后并放大



波形与状态机效果一致，init\_end信号拉高，初始化成功。