基本概况：

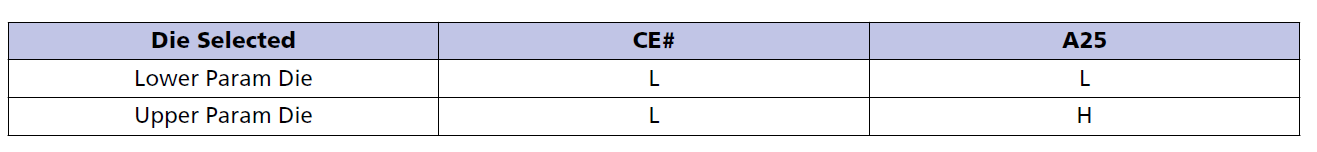
FLASH（闪存）是一种非易失性存储器，用于存储数据和程序。一般情况下FPGA对Flash读写需要借助通信协议，来封装其中复杂的引脚配置细节。FPGA需要配置相应的通信协议，以便与FLASH设备进行交互。不同类型的FLASH使用不同的通信协议，如SPI、I2C、Quad SPI等。用户端需要编写适当的FPGA逻辑，以实现与FLASH之间的通信协议。

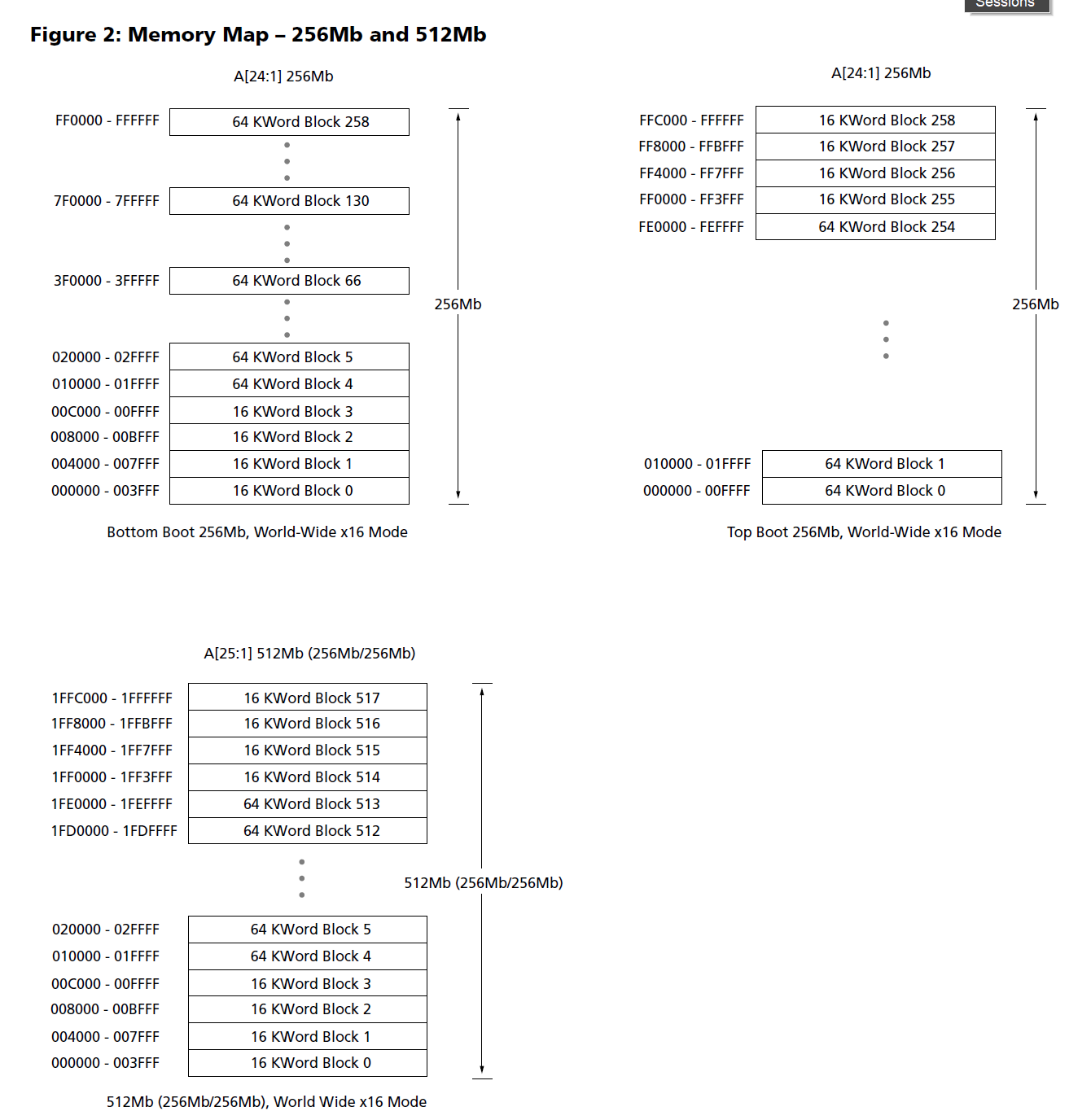
如果不通过通信协议进行交互，FPGA与FLASH之间的交互将变得更加复杂。在没有通信协议的情况下，需要直接控制FPGA引脚以进行读写操作。

如果单纯想验证flash的读写功能是否正常，可以使用verilog写一个读写验证模块，我们在其中只需要对flash的外界引脚信号进行合理配置，并且按照文档要求配置总线、命令以及读写时序（通过状态机实现）数据可以由模块内部自行产生。但是如果需要向其中烧写外部数据比如mac地址，则需要通过C语言控制CPU，通过CPU向FPGA中配置寄存器的方式向flash中烧写外部数据。

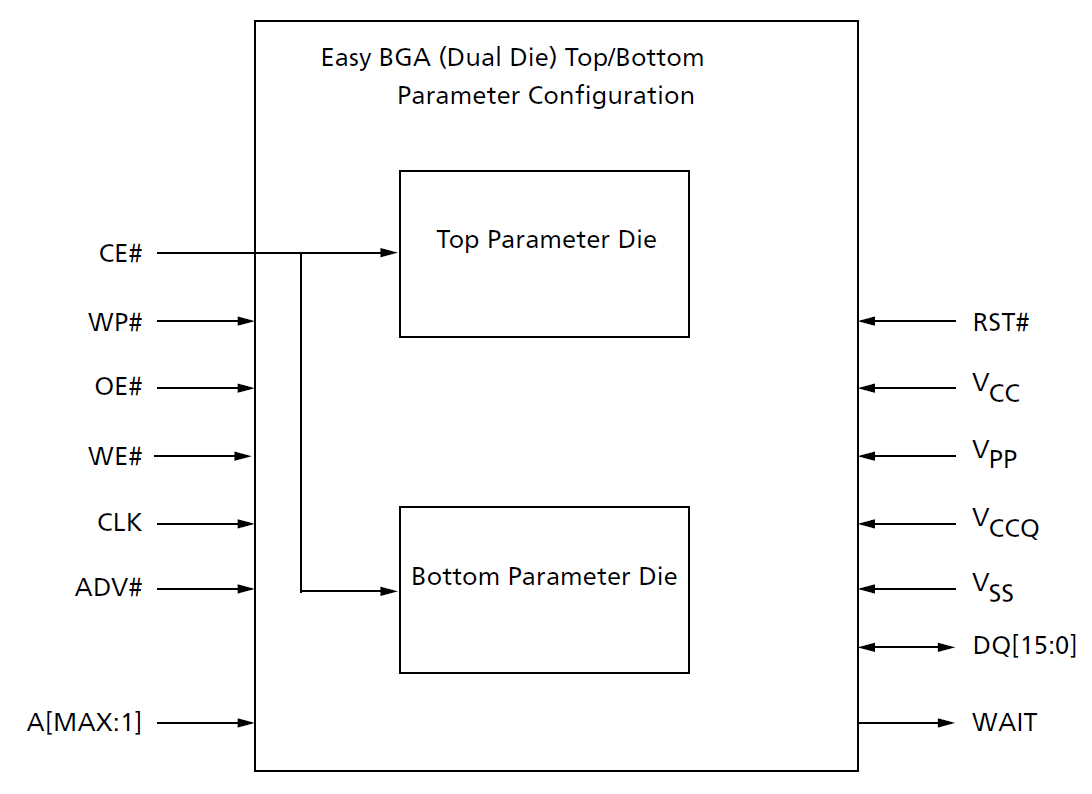
根据镁光flash文档：“命令用户接口（CUI）是系统处理器和设备（flash）的所有内部操作之间的接口。设备（flash）自动执行块擦除和编程所需的算法和时序，状态寄存器指示擦除或编程完成以及可能发生的任何错误。”

需要注意的是，镁光的512Mb大小的flash内部分成了两个逻辑上的存储部分，地址位的最高位比特是作为片选比特来用的，如下图所示（L表示low/0，H表示high/1）CE#信号是整个flash芯片的使能信号。当CE#信号置位之后，flash才被使能，准备好传输数据。然后再使用地址最高位进行存储部分的片选。（die在这里是名词，意思是模具，模块）





上图中给出了flash的清晰的存储单元的结构



另外，不同的封装方式封装出来的芯片，引出的引脚和引脚上对应的信号是不一样的，所以引出的信号前面一般要加上特定的封装方式作为限定语。

注意，上图中带有#的信号指的是低电平有效（所谓active low）；VCCQ是可编程电源电压、VSS是地GND

对应着上图看信号：

左侧：

CE#（chip enable）是整个flash芯片的使能信号，置位（有效位）之后才将芯片激活。当芯片不用的时候该信号必须置无效位（高电平）

WP#（write protect）是用于控制是否允许对存储器进行写入操作，从而保护存储器中的数据免受误操作的影响。

WP# 信号通常用于保护特定的 Flash 存储区域，以防止意外或未经授权的数据修改。当 WP# 信号被激活时，存储器中的数据将被锁定，不允许进行写入操作。这在某些应用场景中是非常有用的，例如防止固件被非法篡改或保护存储在 Flash 中的重要配置信息。（如果不需要用到这个信号，可以将该引脚接到VCCQ或者VSS上）

OE#（output enable）是flash的输出使能信号，置位时将在读周期期间启用芯片内部的输出数据缓冲区（buffer）

WE#（write enable）是写操作使能信号，控制对flash的写入过程。置位时地址和数据锁定在WE#或CE#的上升沿（以先发生者为准，其实就是在这些边沿对数据线采样）

CLK 时钟信号，不用的话要接VCCQ或者地，反正不能悬空

ADV#是地址有效信号，ADV#变为高电平时，地址线上的地址被锁存。为什么要锁存？是因为从地址线上采样到的地址信号需要在flash接收和解析中保持稳定。

A[MAX:1]#是flash的地址输入信号线，其中可能会有未使用的有效地址引脚，这些引脚应该按照具体设计的要求绑定在VCCQ或者VSS上，不应该保持悬空。

右侧：RST#（Reset）复位信号，复位时禁止写入操作，信号为高时启动正常操作，从复位退出时会将设备置于异步读取模式。

Vcc：设备核心电源，逻辑源电压。当VCC小于VLKO时禁止对flash的写入。

Vpp：擦除和编程电源（待进一步）

Vccq：输出驱动器源电压

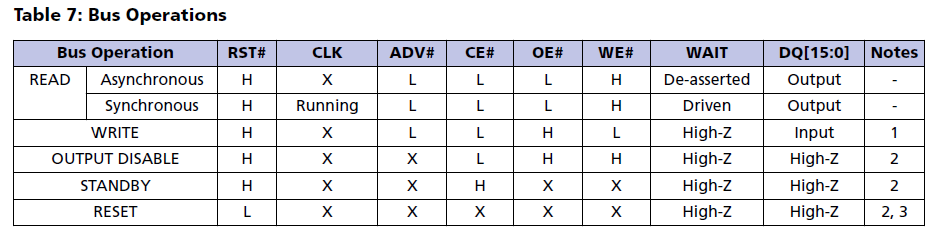
Vss：系统接地

DQ[15:0]（Data input/output/data queue）数据总线，在写入周期内输入数据和命令、在存储器、状态寄存器、保护寄存器和配置寄存器读取周期期间输出数据(有个data ball不知道是干嘛的)

WAIT：用来指示读取的数据的有效与否（GPT说它是用来调和设备和flash之间速度的差异）

在了解了flash外部接入的各个信号引脚之后，接下来要了解的是数据传输的总线过程和时序。

CE# \_low和RST#\_high情况下启动读取操作，flash在内部解码地址输入（上述两个信号先到来的边沿的时刻，地址线上采样到的地址）以确定访问的块，ADV#\_low打开内部的地址锁存器，地址即保持稳定，OE#\_low启用输出数据buffer，并将所选数据算入I/O总线。



了解完这些信号如何对总线操作进行调整之后，接下来了解一下用户端如何对flash进行READ、WRITE、ERASE等操作。

系统CPU（用户端）通过系统总线控制flash的所有上述操作，至于操作的各种细节（算法）则由flash自己管理并完成。

对flash操作的控制通过总线写入命令来控制。通过将特定flash命令写入命令用户界面（CUI）来启动设备操作，将任一命令写入flash的CUI都会启动一系列内部计时函数，最终完成请求的任务。（可以通过RST#或者相应的挂起命令）

操作的具体细节：

要注意的一些点：read array mode是指读取存储在存储单元中的数据，从指定的存储单元中读取数据。

Read status register mode是指读取状态寄存器模式，这种模式一般是在编程或擦除之后flash会进入此模式。状态寄存器的数据在DQ[7:0]上输出。

总之不同的命令代码会代表不同的命令，控制flash进行不同的操作

所谓CFI是指一种用于与闪存存储器进行通信和配置的标准接口协议，为不同的制造商和不同类型的闪存提供了统一的接口，以便主机可以读取存储器的特性、容量、结构和功能信息等各项信息而不需要了解闪存制造商的详细制造工艺等细节。

因此Read CFI mode就是把flash设置为读取CFI接口所提供的种种信息的模式，这些信息将通过DQ[7:0]来进行输出

需要注意的是：在flash中，write和program的意思很接近，都是将数据配置到flash中的过程。

flash通常以word（两个byte）作为读写的最小单位。使用写单字命令来写入word。写入word的过程如下：写入一个word需要两个周期，第一个周期将命令写入CUI为写操作做准备，在下一个周期中，地址和数据被锁存，flash在对应的地址位置执行写入算法。(这里的周期并不指时钟周期)

（在写入操作期间，flash仅对读取状态寄存器命令和写入挂起命令作出响应。写入完成之后，必须发出RREAD ARRAY命令以读取阵列数据）

另外的写操作是：缓冲式写入

这种写入方式涉及到两条命令：1.buffered你program（写入缓冲区命令）这个命令将可变数量的word加载到写入buffer中，最多可以容纳512个word\

2.buffered program confirm（第二步），这个命令将写入buffer的数据写入存储器阵列。

下面还有一种BEFP模式，是另一种缓冲式写入，先不看。后面还有很多其他操作，但是我们主要看的是停等式的word写入，其他的先不看

到目前我们学会了flash外部需要在各个状态下配置的引脚电平信号。以及各个状态对应需要向CUI里写入的命令（这两个必要条件）接下来我们需要考虑的是时序问题（每个周期内准确的操作和对应的准确地址）

一些代号：

地址：

DBA：（device base address）:设备基地址（该地址在双芯片512Mbflash中需要）

DnA：（address within the device）设备内地址

IA：（identification address offset）标识码地址偏移量

CFI-A:读CFI地址偏移量

WA:要写入内存的字对应要写入的地址

BA(address within the block):块内地址

OPT-RA(protection register address)保护寄存器地址

LRA(lock register address)锁定寄存器地址

RCD（read configuration register data）读取配置寄存器的数据，对于不同封装方式的flash，读取的地址位宽不一样。

数据：

ID（identifier data）标识符数据

CFI-D：CDQ上的CFI数据

SRD：状态寄存器数据

WD：字数据

N：要加载到写入缓冲区中的数据的字数

OTP-D：保护寄存器数据

LRD：锁定寄存器数据

对操作的详解：

读模式：flash支持两种读模式：异步页面模式和同步突发模式。异步页面模式是flash上电或者复位之后的默认读取模式。

Flash可以处于四种读取状态中的任何一种：读取阵列、读取标识符、读取状态或者读取CFI。上电或者复位后，默认为读取阵列模式，如果要更改读取状态，需要将对应的READ命令写入设备。

因此设备上电或者重启之后，将会处于异步页模式的读阵列状态，如果想在其他设备向flash写入数据之后进行阵列读取，必须先向flash中写入READ\_ARRAY命令。（这种模式下，16个字被定义为1页，有点类似于突发传输，给一个初始地址，在初始访问时间延迟后，从初始地址开始的一页内的16个word会存储到内部的页缓冲区中，然后由地址总线上地址的变化（最低4位的变化刚好能对应16个word）去把缓冲区里的字word从DQ输出。（flash的地址是以字word为单位的）

异步单字读

外接信号要求：CE#\_low