AN1090

NoBL™：快速 SRAM 架构

相关项目：无
相关器件系列：所有 NoBL™ SRAM 器件
软件版本：无
相关应用笔记：无

AN1090 介绍了 NoBL™ SRAM 的运行操作，并解释了它们适合网络应用的理由。

简介

使用于高性能通信设备和网络应用的处理器需要高速的存储器。需要根据使用的系统架构、应用和处理器来确定需要使用的存储器类型。如果存储器子系统不能满足处理器的要求，那么，系统性能会降低。

本应用笔记描述了赛普拉斯NoBL SRAM架构，通过它能够提高存储器子系统的性能。

NoBL SRAM 说明

NoBL表示No Bus Latency（无总线延迟）。NoBL SRAM专用于消除切换读取和写入操作之间所发生的总线反转延迟。这些器件也被称为zero bus turnaround（无总线反转）。

采用NoBL架构，可去除读和写操作间总线上未用（或死）的周期，从而可以使总线的利用率达到100%，使之适合经常发生READ/WRITE转换的网络应用。

NoBL SRAM提供了两个选择：Pipelined（流水线式）和Flow through（输出式）。Pipelined式器件适用于对频率要求更高的应用，而Flow-through式器件适用于对延迟要求更高的应用。

对于pipelined式器件，输入地址后，从SRAM读取的数据在两个周期内可用。对于flow-through式器件，输入地址后，从SRAM读取的数据在单周期内可用；执行地址和数据操作期间仅有一个单周期。

NoBL SRAM 操作

图1显示的是一个Read/Write/Read/Write序列中的Pipelined NoBL SRAM时序图。

在第一个时钟周期内，读取访问的地址被锁存到SRAM中。由于包括了内部流水线寄存器，所以SRAM在第三个时钟周期中提供数据。但在第二个时钟周期初始化写入访问，如下所示。在第四个时钟周期内为相应的地址提供写入数据。所有访问均是对称的（需要三个时钟周期来完成一个READ或WRITE操作）。因此，可对所有访问设置完整的流水线，而在读和写操作之间没有浪费任何周期。

图 2 显示的是 Flow-through NOBL SRAM 的时序图。在第一个周期中初始化读操作，SRAM 在第二个周期中驱动数据。可以在第二个周期中初始化写操作，并在第三个周期中提供相应的数据。对于 Pipelined NOBL SRAM，读操作和写操作都需要相同数量的周期（即两个周期）。NoBL SRAM 使用命名为 Advance/$\overline{\text{Load}}$ pin (ADV/$\overline{\text{LD}}$)的信号。当 ADV/$\overline{\text{LD}}$被置为低电平时，根据 Read/Write 引脚($\overline{\text{WE}}$)的状态可以将读或写指令传输到 SRAM。如果三个芯片使能中的一个处于无效状态，也会执行 Deselect（取消选择）指令。 当 ADV/$\overline{\text{LD}}$被置为高电平时，将执行突发指令。对于 NoBL SRAM，将使用$\overline{\text{WE}}$引脚和字节写入($\overline{\text{BWx}}$)信号来执行写操作。

图 1. 流水线式 NoBL SRAM 时序图

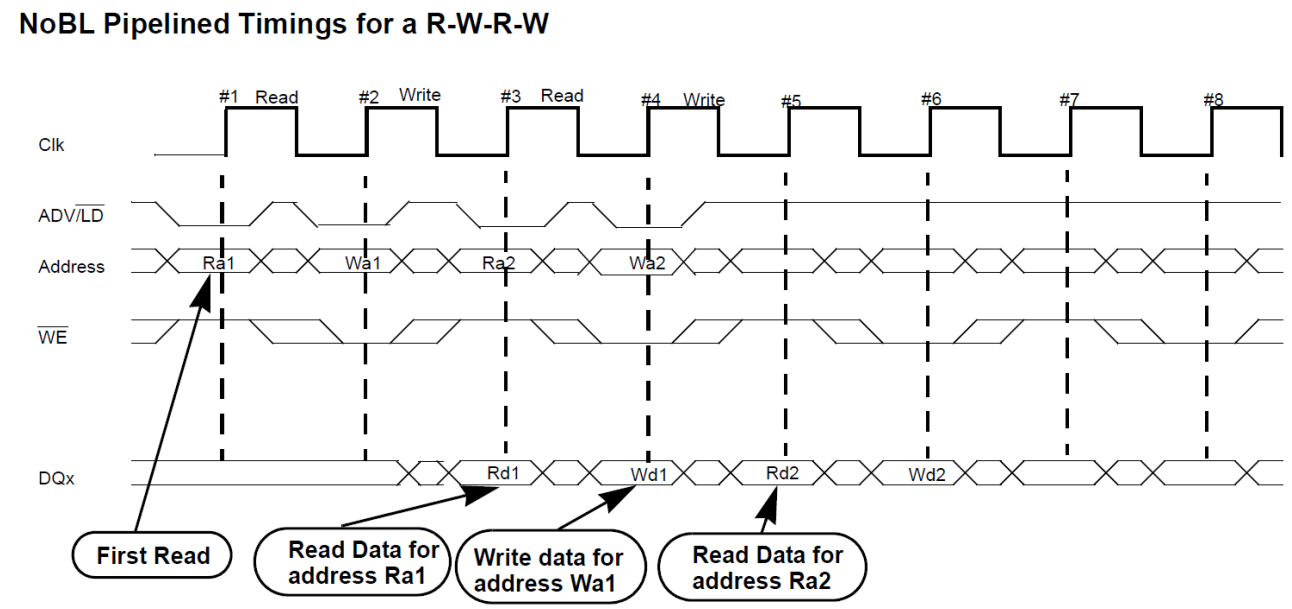
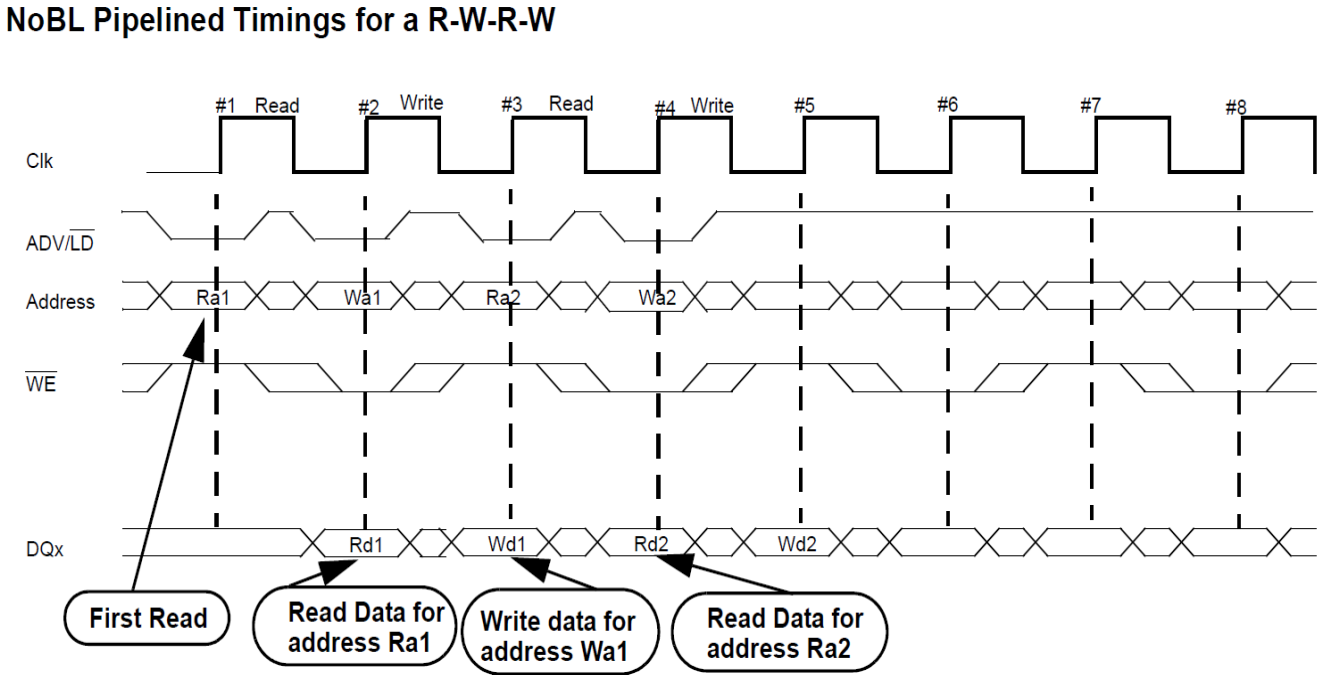


图 2. 输出式 NoBL 时序图



总线效率

总线效率是用于衡量器件通过总线传输数据的效率度量。对于 SRAM，该图展示了连续传输 READ/WRITE 数据时所使用的周期数量。

总线效率= 数据传输的周期数 / 总周期数。

对于一个类似于 R-W-R-W 的操作，当传输数据时，可在每一个时钟内达到最大的总线效率。换言之，不管是否操作，当数据在每个时钟周期内被传输，总线效率将达到 100%。NoBL SRAM 可在每一个周期内完成一个数据传输操作，因此，它可以达到 100%的总线效率。

在完成写操作后执行读操作或完成读操作后执行写操作，这样可增大可用带宽。

图 3. Pipelined NoBL SRAM 的详细时序

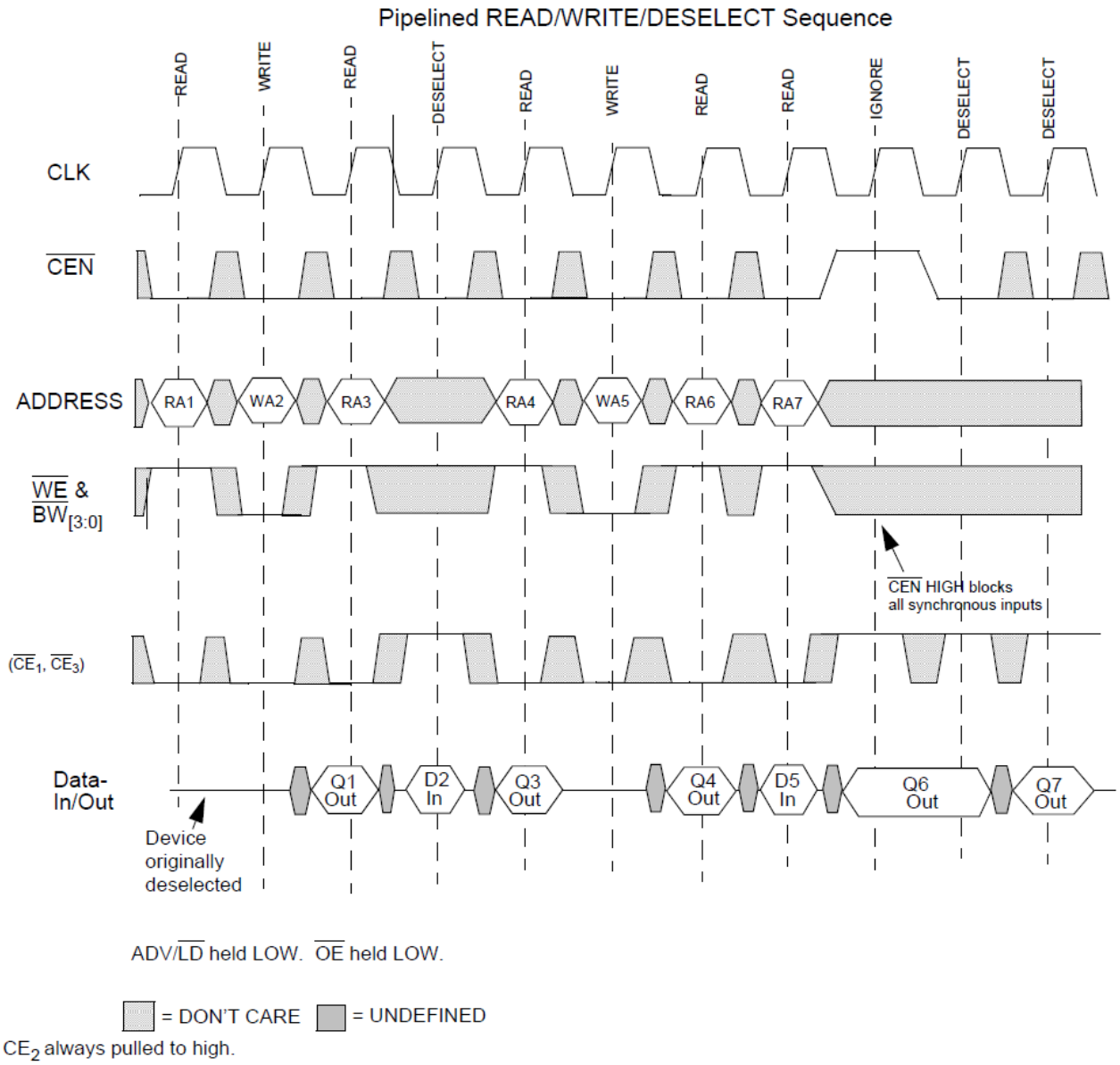
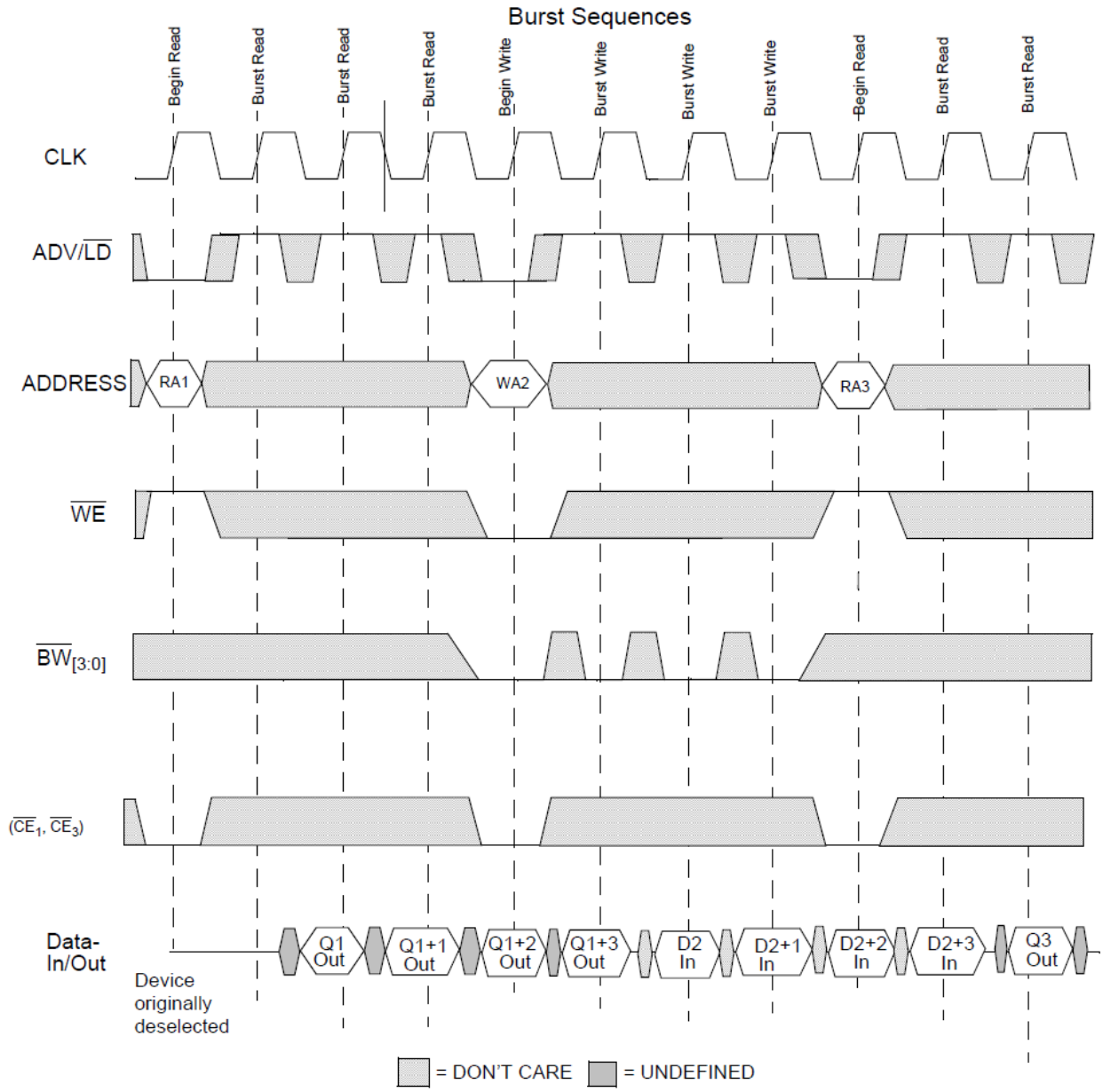


图 4. Pipelined NoBL 器件的突发时序



CE_2 is always pulled high.

图 3 和图 4 分别显示了 pipelined NoBL SRAM 的时序。

新访问周期是由 ADV/$\overline{LD}$信号控制的。Advance/Load 输入用于增加片上地址计数器或加载新的地址。当该输入在时钟的上升沿上被置为高电平，并且$\overline{CEN}$被置为低电平时，可增大内部突发计数器。当该信号在时钟的上升沿上被置为低电平时，可通过多个地址线将一个新地址加载到器件内，以执行访问操作。

当$\overline{WE}$信号在时钟的上升沿上处于高电平时，将初始化

读周期。在后续周期内，如果 ADV/$\overline{LD}$一直处于高电平，则器件开始执行突发访问操作，如图 4 所示。

选好器件后，如果$\overline{WE}$在时钟的上升沿上被置为低电平，则 NoBL 器件将开始执行写访问操作。

突发计数器的序列由 MODE 输入信号决定。MODE 上的低电平输入选择线性突发模式，而高电平输入选择交错突发序列。这两个突发计数器在突发序列中都使用 A0 和 A1，并且在充分递增时执行环绕式处理。突发访问的操作序列如下：将 ADV/$\overline{LD}$, $\overline{CEN}$, $\overline{CE}_1$, $\overline{CE}_3$置为低电平，并将 CE_2 置为高电平，以将新地址加载到 SRAM 内。在后续的时钟周期内，无论 CE 或$\overline{WE}$引脚的状态如何，将 ADV/$\overline{LD}$置为高电平都会增加内部突发计数器。$\overline{WE}$在开始突发周期时被锁存。因此，在突发序列中将保持访问的类型（读取或写入）。表 1 显示的是交错突发序列，表 2 显示的是线性突发序列。

表 1. 交错突发序列

第一地址	第二地址	第三地址	第四地址
Ax+1, Ax	Ax+1, Ax	Ax+1, Ax	Ax+1, Ax
00	01	10	11
01	00	11	10
10	11	00	01
11	10	01	00

表 2. 线性突发序列

第一地址	第二地址	第三地址	第四地址
Ax+1, Ax	Ax+1, Ax	Ax+1, Ax	Ax+1, Ax
00	01	10	11
01	10	11	00
10	11	00	01
11	00	01	10

使用 NOBL SRAM 时要注意的事项

1. t_{CHZ} 和 t_{CLZ}。t_{CHZ} 参数指定 NoBL 器件在时钟的上升沿后将其输出驱动置为高阻态的时间。t_{CLZ} 参数指定 NoBL 器件开始将数据驱动到数据总线（低阻抗状态）的时间。表 3 显示了 t_{CHZ} 和 t_{CLZ} 的最大和最小时序，如数据手册中所述。

   表 3. t_{CHZ} 和 t_{CLZ} 的值

参数	最小值	最大值
t_{CHZ}	1.5	3.5
t_{CLZ}	2.5	

   好像指定了器件，以允许共享共同的数据总线的 SRAM 之间发生冲突（其原因是 $t_{CHZ(max)}$和 $t_{CLZ(min)}$的重叠）。事实并非如此。需要在全部过程、温度和电压范围下保证这两个参数的规范。$t_{CHZ(max)}$在过程的底角、高温度和低工作电压被确认。 $t_{CLZ(min)}$ 则在相反的工作范围（快速过程、低温度、高电压）被确认。

   很明显，在相同的电路板上，这两个极限值并非同时存在。进入高阻态前，在所有工作条件下（就是说，不管处理过程的变化），delta 均约为 1 ns，NoBL 能够将总线驱动到高阻态。因此，各 NoBL SRAM 器件之间不会发生任何数据总线冲突。

2. NoBL SRAM 的芯片选择

   NoBL SRAM 的芯片选择运行方式与同步流水线式 SRAM 的芯片选择运行方式不一样。NoBL SRAM 包括三个芯片选择$\overline{CE}_1$、CE_2 和 CE_3。在同步流水线式 SRAM，当芯片选择无效，$\overline{ADSP}$信号将被忽略。

   在 NoBL SRAM 中，CE 引脚在时钟的上升沿上被采样。NoBL SRAM 的引脚将不被任何其他输入掩码。因此，需要激活使能所有芯片，以便选择器件，另外这三个芯片选择中的任何一个可以取消选择器件。

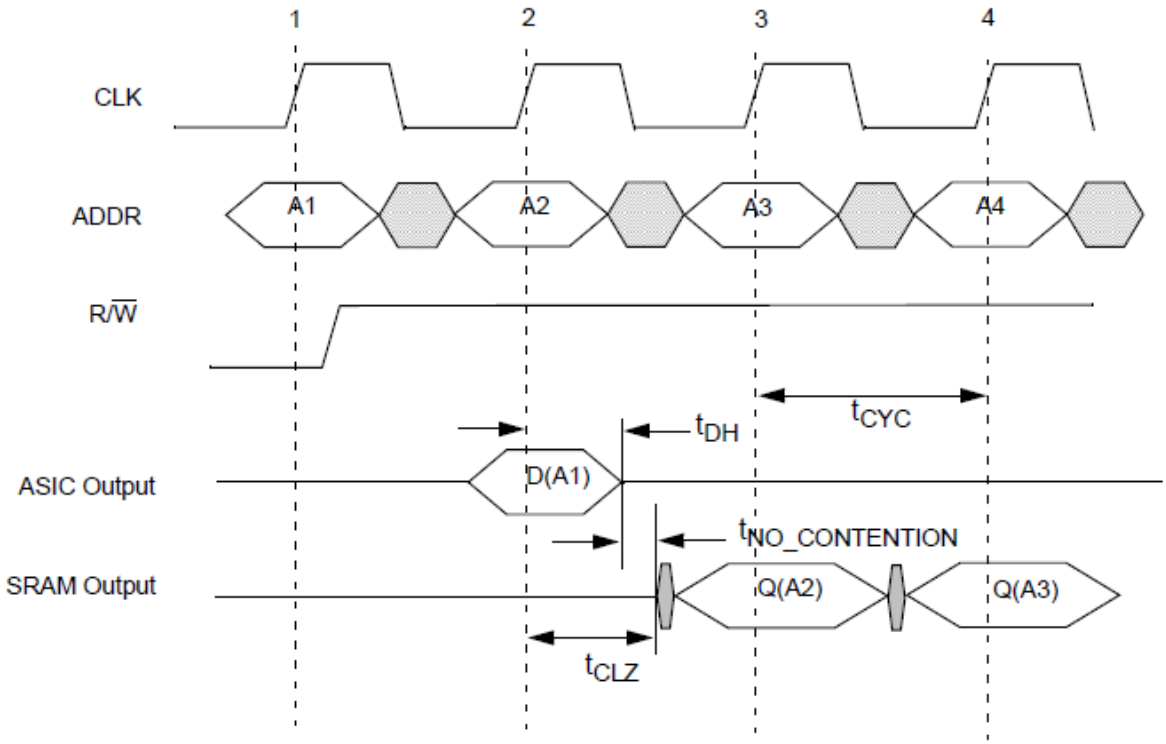
3. $\overline{OE}$ 控制

   NoBL 器件是通用的 I/O 器件。因此当输出处于工作状态时，不应将数据驱动到器件内。将数据传输到 DQ_0–DQ_{31} 输入前，输出使能($\overline{OE}$)可以取消激活高电平。这样可使输出驱动模块处于三态。然而，当初始化写操作时，内部逻辑将被确认，并且会同步禁用输出驱动模块，从而允许传输些数据。该特性大大简化了写序列，并在所有情况中，执行写操作器件可以被免除使用$\overline{OE}$。

4. 总线冲突

   系统设计员担心发生总线冲突，特别是发生高频率冲突。应特别注意如何在切换写指令和读指令时不存在总线死周期。SRAM 输出驱动程序开启下一个读周期前，SRAM 控制器执行的写周期必须处于高阻抗状态。

图 5. 如果 ASIC 的关闭时间大于 $t_{NO\_CONTENTION}$，将发生总线冲突。



不能完全消除总线冲突。图 5 演示了总线冲突是如何发生的。执行写周期时，驱动到 SRAM（在这种情况下，其由 ASIC）的数据必需满足 t_{DH} 保持时间或 0.5 ns。为了确保保持时间为 0.5 ns，以补偿温度，V_{DD} 的变化和时钟时滞， ASIC 驱动总线的时间必需大于 0.5 ns。满足保持时间后，如果该时间超过 $t_{NO\_CONTENTION}$，那么，将发生总线冲突。ASIC 驱动总线的时间是所使用的 ASIC 的处理技术和其输出驱动模块的功能。ASIC 尝试关闭并且 SRAM 尝试打开器件时，可能会发生多次总线冲突。当然，t_{CLZ} 等于 1.5 ns 是 SRAM 的最差条件，但在大多数温度和电压的情况下，它大于 1.5 ns。但仍使用 1.5 ns 来计算最差情况。

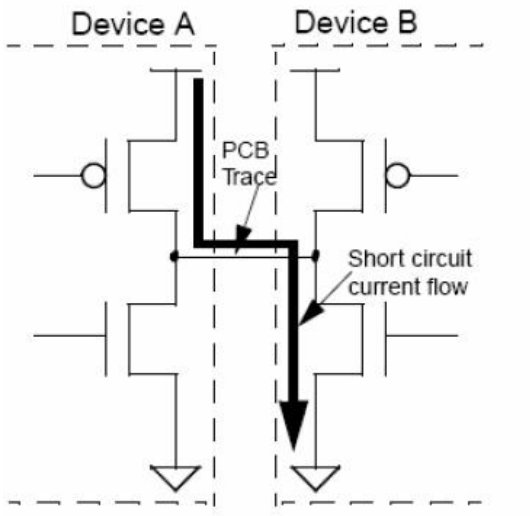
因此，在所有温度和电压条件下不能完全避免发生总线冲突。在下面的段落中，我们将分析它们对器件的影响。

假设在时钟的上升沿后 ASIC 变为高阻态的时长为 2.5 ns。这意味着，在最差条件下，会发生 1 ns 的总线冲突。假设 NoBL SRAM 在频率为 133 MHz 时或在 7.5 ns 的时钟期间（t_{CYC}）运行。仅在写周期切换为读周期时，才会发生此冲突。最差情况下，每两个周期会发生该冲突。因此，在最差的情况下，每两个时钟周期（15 ns）或者全部周期总数的 6.7%在 1 ns 时会发生总线冲突。

平均来说，一半的 DQ 引脚总数和 SRAM 都将出现于总线冲突（因为数据位可能是 1，也可能是 0）。而对于 x36 SRAM，假设 18 引脚出现冲突（平均的），我们将计算总线冲突。如果在假设冲突期间，SRAM 驱动电路的电阻为 50 Ω，ASIC 的电阻为 50 Ω，那么，在电源和接地间将有一个 100 Ω 等效电路，如图 6 所示。请注意，由于 RAM 驱动模块和 ASIC 驱动模块的电阻不断改变（即一个关闭，另一个打开），所以上面的假设只是一个粗略估计。已给 3.6 V V_{DD} (Max)，冲突期间的电流 = 3.6 / (50 + 50) = 36 mA.

冲突期间，每个 I/O 的电源和接地间的电流为 36 mA。

图 6. 冲突的器件



发生冲突时，SRAM 的功损 = 0.036^2*50 = 0.065 W。

对于 18 个 I/O，功率耗损 = 0.065*18= 1.17 W

乍一看，这简直是一个不可接受的功率。可是，它仅在周期总数的 6.7 %期间发生。

无总线冲突的功率耗损 = 内核功耗 + I/O 开关功耗。

内核功耗 = V_{DD}(Max) x I_{DD}(Max) = 3.6*0.35 = 1.26 W

假设执行读操作的 50%， I/O 开关功耗 = ½*f *C*V^2*0.5

其中，f = 133 MHz

对于 36 IO，C = 20 pF*36

V = 3.6 V

因此，I/O 开关功耗

= ½*133*10^6*20*10^{-12}*36*3.6*3.6*0.5 = 0.31 W

总功耗 = 1.26 + 0.31 = 1.57 W

添加了 6.7%时间的总线冲突功耗，1.57 + (0.067*1.17) = 1.65 W，或者最小增量。

对结温的影响：

工作结温 $T_J = T_A + \theta_{JA}$ *P ，其中 P 是由 SRAM 造成的功率损耗。

T_A (Max) = 70 ºC。

θ_{JA} = 25 ºC/W

无总线冲突时，T_J= 70 + 25*1.57 = 109 ºC

发生总线冲突时，T_J= 70 + 25*1.65 = 111 ºC

因此，结温将增加 2 ºC。这是可接受的增量。

也就是说，它是可接受小数量的总线冲突的。总线冲突不会损坏驱动模块。NoBL SRAM 具有单独的电源引脚，以供给 I/O 和内核。因此内核电源将与 I/O 环和 I/O 上浪涌的任何电流隔离。额外的电流导致小的 IR，并导致 I/O 环发生 LdI/dt 电压下降，而不对内核逻辑造成影响。该架构可防止由小量冲突导致对 SRAM 核的电压降低。

请注意，上述计算中假设每两个周期（也是最差情况）发生一次总线冲突。还假设保持上述两个驱动模块电阻不变。最后，假设 ASIC 在最差情况下被关闭，同时，SRAM 在最差情况下被打开。通常，这些情况发生在电流的对立极限值和温度中，并且不太可能发生上述情况。

应用

如前面部分所述，使用背靠背 READ-WRITE 操作可从 NoBL SRAM 中受益匪浅。

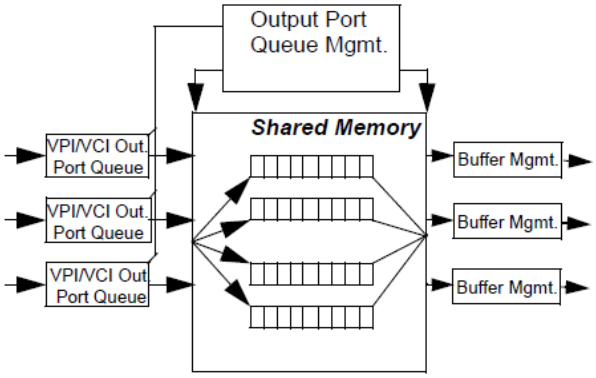
NoBL SRAM 消除数据延迟，并最大化地利用存储器带宽。

ATM 开关应用是一个通过使用 NoBL SRAM 证明系统性能得到提高的示例。

ATM 开关是需要高存储空间吞吐量的应用。 共有三种方法可实现一个 ATM 开关：共享的存储器、自身路由和共享的背板。

共享存储器架构指的是大的共享存储器被路由到输出端口前可用于缓冲输入单元。图 7 显示的是 ATM 开关的典型共享存储器架构。共享存储器的大小取决于为每一个输出端口而缓冲的单元数量。由开关支持的数据速率可确定共享存储器的数据总线宽度以及存储器工作频率。

图 7. 共享存储器 ATM 开关



大多数共享存储器开关都使用了 SRAM。对于该问题，我们使用一个共享存储器模块，能够存储高达 26 K 的单元（1 ATM 单元 l = 53 字节），其目标数据速率为 19.2 Gbps。大多数 ATM 操作都涉及到了 ATM 单元的连续写和读操作。

使用本应用的特定类型存储器时，重点的是它能否提供 19.2 Gbps 的数据速率。因此，通过下述几种方法能够达到所需的数据速率。使用较宽的数据总线宽度，能够达到所需数据速率而不要在告诉数据速率中运行 SRAM，这是其中一种方法。另一个能够达到较高数据速率的方法是运行同步

SRAM 时，使用较高的时钟频率。可以通过下面的公式来计算所需的时钟频率：

频率 = 数据速率 /（总线效率 × 数据总线带宽）

在本应用中，我们假设使用了 192 位的总线带宽。

使用 NoBL 器件的解决方案

该应用的一个可用解决方案是NoBL SRAM。这些器件在写和读操作间不需要反转时间。

该器件的流水线版本包括两个周期标准偏移（一个读操作和一个写操作）。

使用该器件，可以使总线利用率达到100%

执行该模块需要的带宽为19.2 Gbps，工作频率达到19.2 Gbps / (1.0 × 192 bits) = 100 MHz。

结论

通过使用 NoBL 架构，可以消除读和写操作间的等待周期，使 I/O 总线的利用率接近 100%。这样可大大改善在给定系统中的带宽。

文档修订记录

文档标题：NoBL™：快速 SRAM 架构 – AN1090

文档编号：001-92154

修订版	ECN	原始变更	提交日期	变更说明
**	4346097	RLIU	04/26/2014	本文档版本号为 Rev**，译自英文版 001-26399 Rev*E。

全球销售和设计支持

赛普拉斯公司拥有一个由办事处、解决方案中心、工厂代表和经销商组成的全球性网络。要找到离您最近的办事处，请访问赛普拉斯所在地。

产品

汽车	cypress.com/go/automotive
时钟与缓冲区	cypress.com/go/clocks
接口	cypress.com/go/interface
照明和电源控制	cypress.com/go/powerpsoc cypress.com/go/plc
存储器	cypress.com/go/memory
光学导航传感器	cypress.com/go/ons
PSoC	cypress.com/go/psoc
触摸感应	cypress.com/go/touch
USB 控制器	cypress.com/go/usb
无线/射频	cypress.com/go/wireless

PSoC®解决方案

psoc.cypress.com/solutions

PSoC 1 | PSoC 3 | PSoC 5

赛普拉斯开发者社区

社区 | 论坛| 博客 | 视频 | 培训

技术支持

cypress.com/go/support

赛普拉斯半导体
198 Champion Court
San Jose, CA 95134-1709

电话 ：408-943-2600
传真 ：408-943-4730
网站 ：www.cypress.com