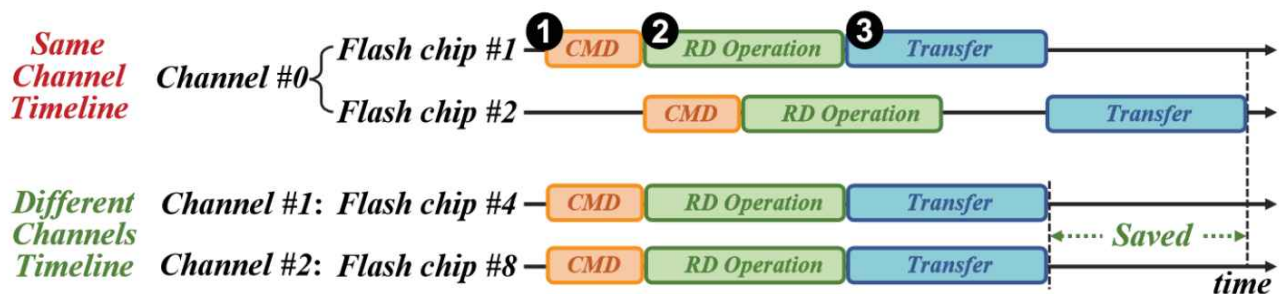


# Venice: Improving Solid-State Drive Parallelism at Low Cost via Conflict-Free Accesses

## Motivation

### Problem

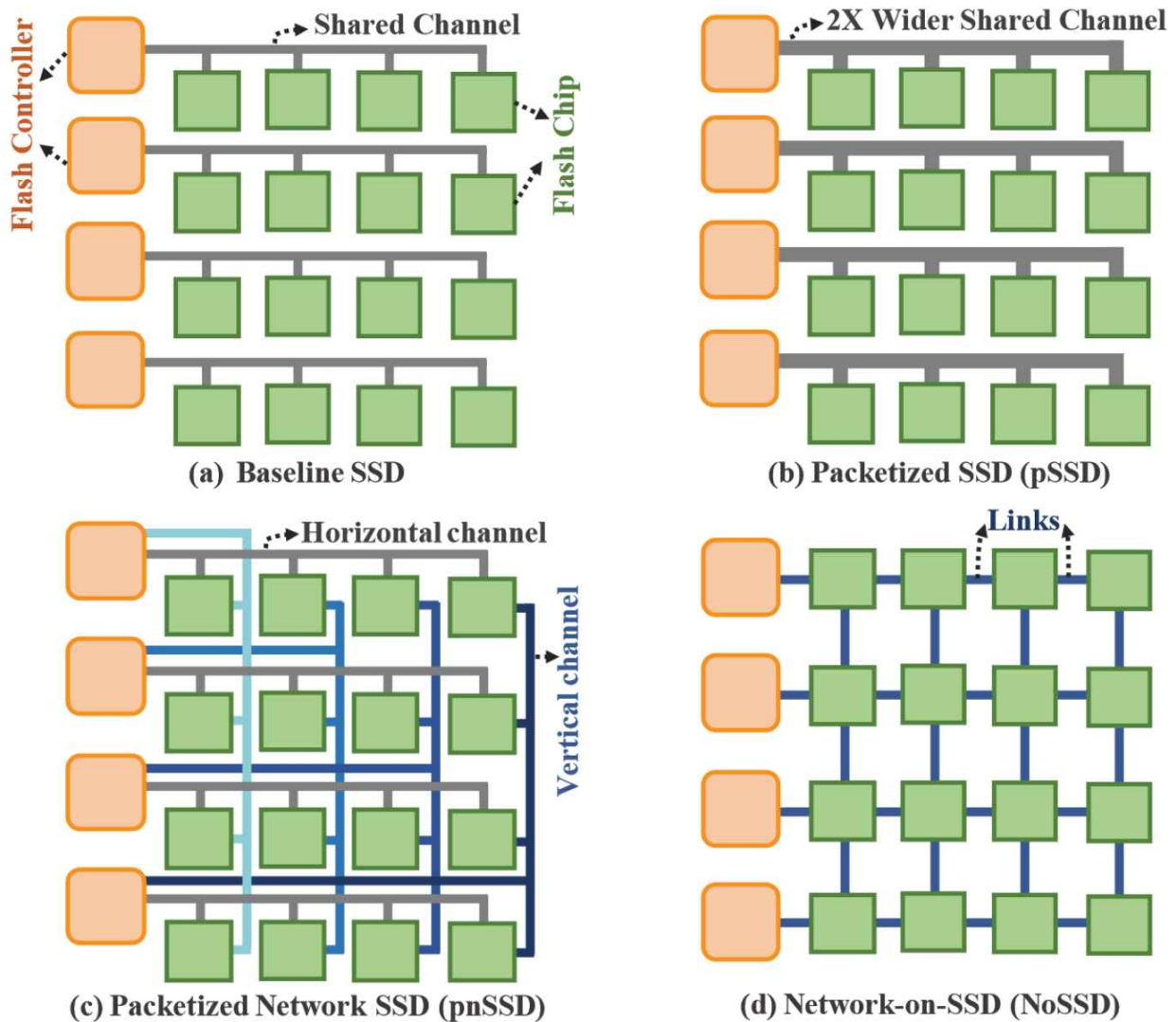
在SSD内，当多个IO请求访问同一个channel时，路径冲突就会产生，导致性能降低。增加channel的数量虽然可以一定程度上缓解这个问题，但是同时也会导致控制器更加复杂（例如，需要更多的IO pins来服务更多并行通道）。增加通道的带宽导致每个闪存晶圆20%额外的电路面积开销并且只是减轻，不能解决路径冲突问题。



**Figure 3: Service timeline of two read requests to two different flash chips. The two flash chips are connected to the same channel (top) or different channels (bottom).**

### Existing Approaches

- Packetized SSD (pSSD). pSSD通过同时使用控制信号pin和数据信号pin来同时传输指令和数据增加了一倍的通道带宽。并且集成了一个片上的控制器使能控制器和芯片之间分组数据。它的缺点是引入了较大的电路面积开销。
- Packetized NetworkSSD. 为每个芯片提供两条访问路径（2 channels）
- Network-on-SSD. 提供一个2D网络互联闪存芯片。但是，它导致了额外的面积开销，主要来源于（1）路由器（2）每个闪存芯片需要4倍的I/O pins。除此之外，它也不能利用多样化的路径。



**Figure 2: Flash chip array architecture of four SSD designs: Baseline SSD, Packetized SSD (pSSD) [15], Packetized Network SSD (pnSSD) [15], and Network-on-SSD (NoSSD) [38].**

## Design

### Low-cost Interconnection Network

Venice使用Flash Node将闪存芯片和路由器分离。在一个Flash Node中，闪存芯片被连接到一个路由器，Flash Node之间通过2D网络互联。因此，闪存芯片无需额外的修改。

### Path Reservation.

对于给定的I/O请求，如果连接的通道繁忙，flash node将发送一个侦察包。该数据包通过非最小全自适应路由算法遍历flash节点，如下所示。一旦它遍历一个节点，它就会向路由器添加一个映射(入口端口到出口端口)，这意味着该路径是保留的。如果找不到出口端口，则返回并取消预订。如果报文返回到flash节点时没有成功保留路径，flash节点将立即发送另一个侦察报文。

# Non-Minimal Full-Adaptive Routing Algorithm

非最小意味着如果最近的路已经被预留，侦察包可以绕很长一段路。例如，从FC3到F2最近的路径包含F1或F7，其未连接到F2的端口处于繁忙状态。因此，使用未使用的链接，这是一个很长的路，但没有路径冲突。

额外的传输开销可以忽略不计(我认为这是因为传输是通过流水线执行的)。缺点是需要保留的路径数量增加了。

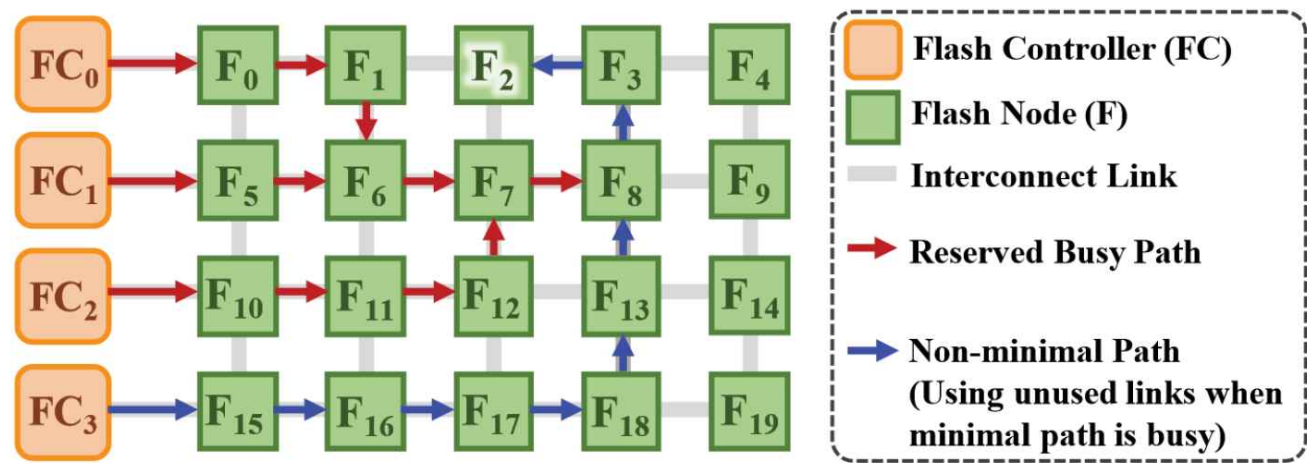


Figure 8: Example demonstrating how Venice’s non-minimal routing algorithm finds a conflict-free path in the interconnection network of flash nodes