

SNIA DEVELOPER CONFERENCE



By Developers FOR Developers

Hyatt Regency Santa Clara, CA
September 15-17, 2025

A decorative graphic consisting of a series of dots forming a wave that flows from left to right across the middle of the slide. The dots are colored in a gradient from purple on the left to yellow in the middle, and then to light blue on the right.

SDXI v1.1 Data Movement Accelerator Interface Update

William Moyes

AMD, SNIA SDXI TWG Editor

Shyam Iyer

Dell, SNIA SDXI TWG Chair

www.sniadeveloper.org

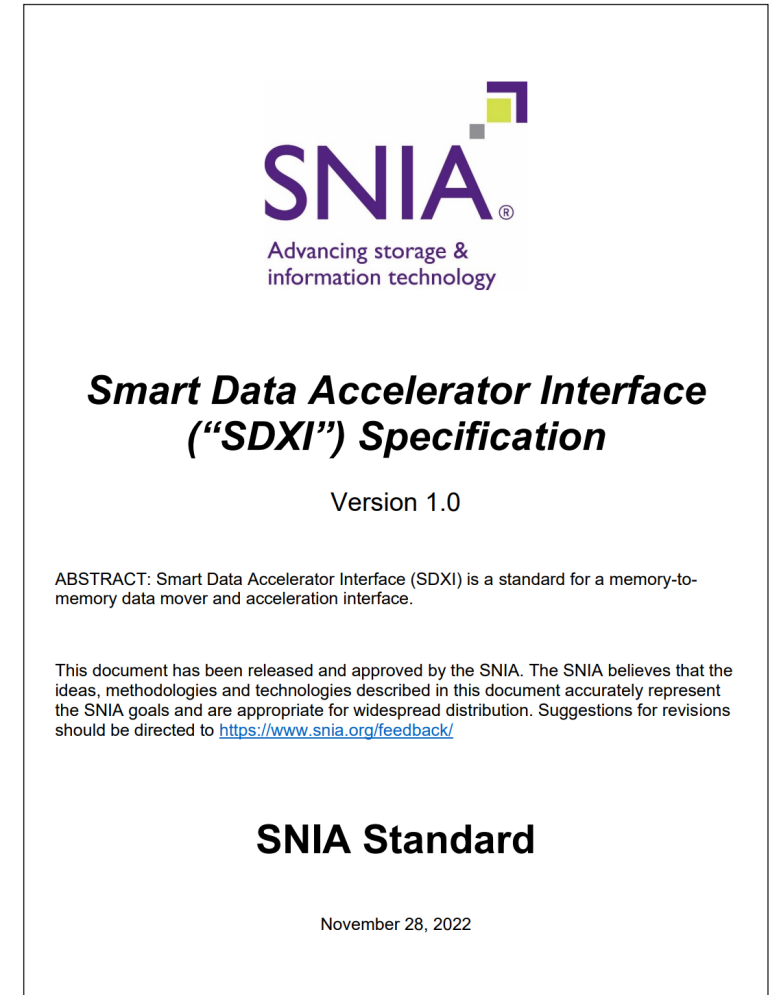
Agenda

- Introduction
- SDXI v1.0 -> v1.1 Update
- SDXI v1.1 New Features Overview
- SDXI and Storage.AI
- Q&A

Introduction

What is SDXI?

- A vendor neutral software to hardware interface for memory-to-memory data mover engines
- A modern design that supports Shared Virtual Memory, Virtualization, and User Mode accelerator use
- Supports App-to-App and VM-to-VM data movement without context switching
- Architected to be easily extended for additional off-load capabilities



Visit <http://snia.org/sdxi>

SDXI v1.0 to v1.1 Changes

- **Memory Synchronization**
 - New Read Barrier support
 - New atomicity guarantees for the 'Write Immediate' command
- **Extensibility / Flexibility**
 - Function Assignable Opcodes
 - Support for 2048 Opcode Groups per function
 - Mailbox scheme now optional
 - Peer-to-Peer DMA support now optional
 - Support for Privileged Mode Requested (PMR)
- **New operations**
 - Extended DMA Operations (memset, memcpy, memcpyimm).
 - Double Copy support (single source, dual target)
 - CRC Operation Group
 - PI Operation Group
- **Important text clarifications and improvements from v1.0**
 - Confidential Memory access support
 - Admin commands failure and implications to non-admin contexts
 - Improvements to Descriptor enqueueing example code

SDXI v1.1 Command Overview

Type	Sub-Type	Operation	Spec	Notes
0x001	0x01	DSC_DMAB_NOP	SDXI 1.0	No Operation
0x001	0x02	DSC_DMAB_WRT_IMM	SDXI 1.0	Write 1 to 32 bytes (signaling)
0x001	0x03	DSC_DMAB_COPY	SDXI 1.0	Copy Memory (memcpy)
0x001	0x04	DSC_DMAB_REPCOPY	SDXI 1.0	Repeat Copy (4K)
0x002	many	Administrative Ops	SDXI 1.0	Administrative commands (start, stop, flush, discover, etc.)
0x003	many	DSC_ATM_*	SDXI 1.0	Atomic Swap / Add / Subtract / And / Or / Xor / Min(signed) / Max(signed) / Min(unsigned) / Max (unsigned) / Increment with Rollover / Decrement with Rollover / Compare and Swap
0x004	0x00	DSC_INTR	SDXI 1.0	Generate Interrupt
Func Assigned	0x01	DSC_XDMA_FILL_IMM	SDXI 1.1	Memory Fill (memset)
Func Assigned	0x02	DSC_XDMA_CMP	SDXI 1.1	Memory Compare (memcmp)
Func Assigned	0x03	DSC_XDMA_CMP_IMM	SDXI 1.1	Memory Compare against Immediate Byte
Func Assigned	0x01	DSC_DBLCOPY	SDXI 1.1	Single source, dual target copy
Func Assigned	many	DSC_CRCM1_*	SDXI 1.1	CRC16 / CRC32 / CRC64 over multiple blocks or a single stream, with or without copying the data
Func Assigned	many	DSC_PI_*	SDXI 1.1	Protection Information DIF/DIX operations (Check, Insert, Strip, Update, Compare)

SDXI 1.1: Function Assigned Opcodes

Why?

- SDXI 1.0 allowed only 32 Operation Groups out of 2048 possible
- Innovations had to wait for SNIA standardization. The four vendor defined opcodes lacked robust discovery and namespace mechanisms

What's New:

- Functions now advertise supported operations via a list of GUIDs
- Existing and future SNIA defined operation groups have GUIDs
- Vendors can introduce their own extensions by creating their own GUIDs
- SNIA can adopt/ratify extensions without ecosystem disruption

Extended DMA commands

'C' equivalents:

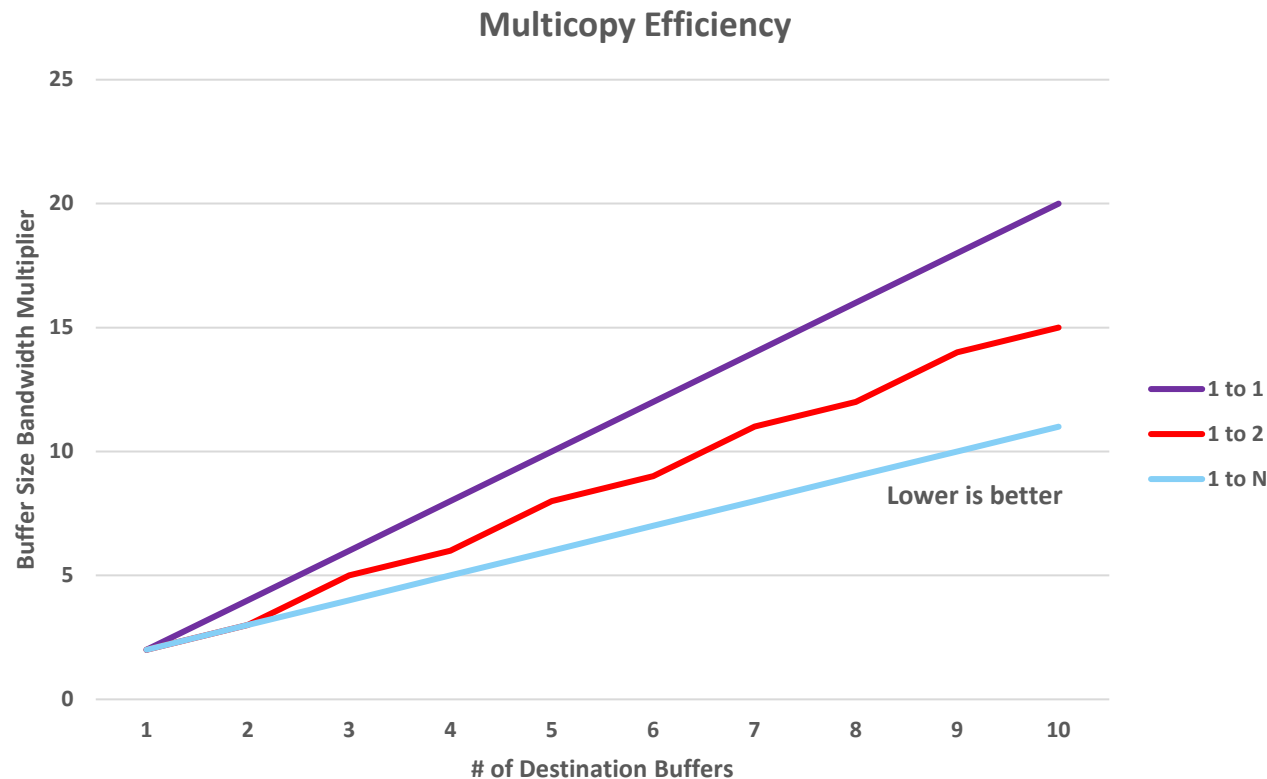
- memcpy - SDXI 1.0 'Copy'
- memset - SDXI 1.1 'Fill Immediate' - fills a buffer with a repeated byte
- memcmp - SDXI 1.1 'Compare' - Compares two buffers, returning a C like comparison result, and the offset of the earliest difference

Also

- 'Compare Immediate' - Compares each byte in a buffer against a constant byte value. Useful for checking if a buffer is all zero, etc.

Double Copy

- Copies a single source buffer to two target locations
- Improves efficiency when making multiple replicas of data. 1-to-2 copy offers most of the efficiency gains, without the complexity needed for a 1-to-N copy
- Two replicas is a common use case



CRC Offload Commands

The CRC Operation Group can:

- Calculate a CRC in place, or while copying data
- Extend a CRC calculation across commands for streamed data
- Calculate individual CRCs on multiple 512- or 4096-byte blocks in a buffer
- Insert or remove 8- or 16-byte metadata 'gaps' while copying blocks of data

CRC Application	crcsel	reflect	seed + invert	CRC of "123456789"
CRC-16 / T10-DIF	0x0	00b	seed=0x0000, inv=00b -or- seed=0xFFFF, inv=01b	0xD0DB
CRC-32C / iSCSI	0x1	11b	seed=0xFFFFFFFF, inv=10b -or- seed=0x00000000, inv=11b	0xE3069283
PKZIP / PNG / ZLIB	0x2	11b	seed=0xFFFFFFFF, inv=10b -or- seed=0x00000000, inv=11b	0xCBF43926
BZIP2	0x2	00b	seed=0xFFFFFFFF, inv=10b -or- seed=0x00000000, inv=11b	0xFC891918
MPEG-2	0x2	00b	seed=0xFFFFFFFF, inv=00b -or- seed=0x00000000, inv=01b	0x0376E6E7
NVM Express 64b CRC	0x3	11b	seed=0xFF...FF, inv=10b -or- seed=0x00...00, inv=11b	0xAE8B14860A799888

Multiple popular CRC polynomials and modes are supported spanning a wide variety of use cases, such as T10 DIX/DIF guard tags, file formats such as zip, png, and mpeg2, and protocols such as iSCSI

Protection Information Operation Group

- Inspired by the new CRC off-load commands
- Directly manages SCSI / NVMe Protection Information Tags
- Computes CRC, increments ref tag, checks tags for expected values
- Supports:
 - DIF Check
 - DIF Insert
 - DIF Strip
 - DIF Update
 - DIF Compare

SDXI and Storage.AI

- Architecturally, SDXI can:
 - Transfer data within and across system memory and MMIO spaces
 - CPU memory ↔ NPU memory ↔ GPU memory ↔ DPU memory
 - Access multiple address spaces
 - Intended for Guest Physical, or process virtual address spaces
 - Could be leveraged to access accelerator internal address spaces
 - Define new opcodes via Function Assigned opcodes
 - Allows SDXI extensions for new AI memory workloads
- Help create the future, Join SDXI TWG!



Q&A



Thank you for attending!

Please remember to rate this session. You get access the presentations at
<http://sniadeveloper.org/conference>



Backup

