Storage Interfaces

Introduction

Tom Gardner

IEEE Silicon Valley Technology History Committee

- Part of IEEE Santa Clara Valley Section
- Webinars on history of Silicon Valley history

www.SiliconValleyHistory.com

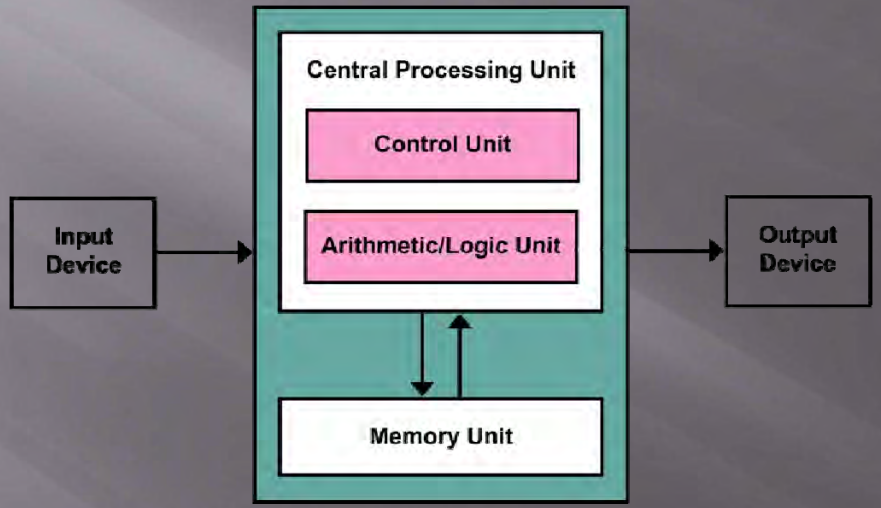
Storage Interfaces Agenda

Speaker	Subject
Tom Gardner	Introduction & History
Grant Saviers	Evolution of Storage Interfaces
Jai Menon	Evolution of Block Storage System Interfaces
Amber Huffman	Fast & Efficient Interfaces for the SSD Era
Questions & Answers	

Presentations are in two files
Gardner and Saviers: Storage Interfaces 20210511 Part 1.pdf
Menon and Huffman: Storage Interfaces 20210511 Part 2.pdf

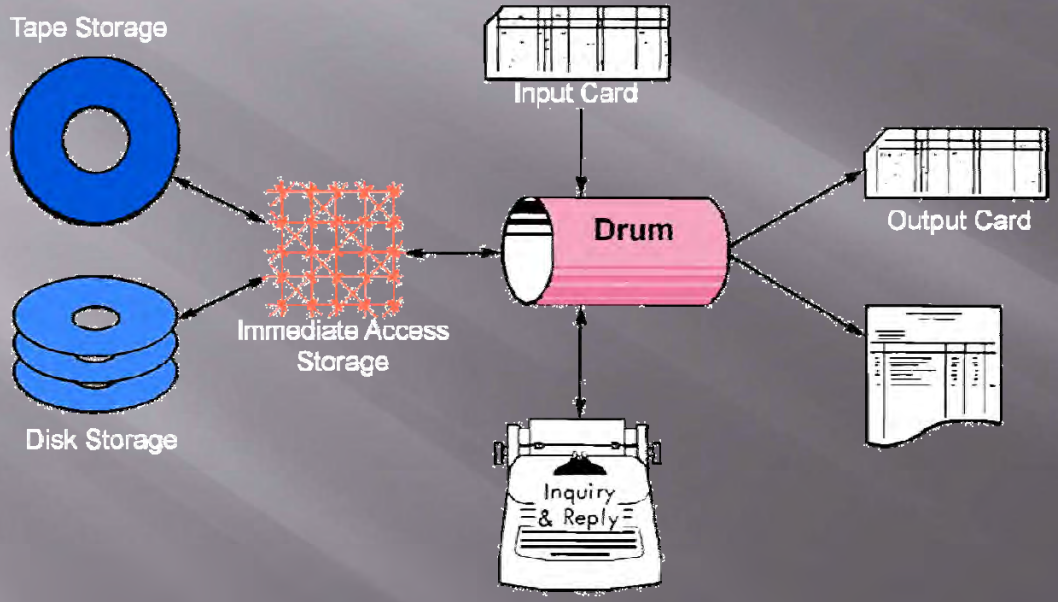
Layers of memory and storage

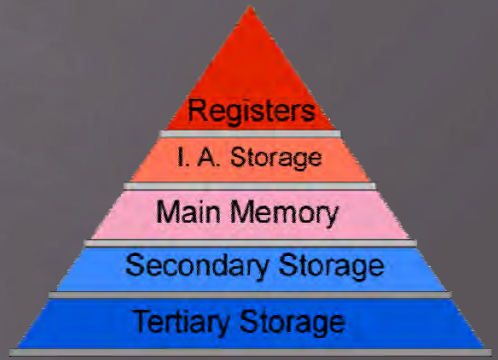
In the beginning



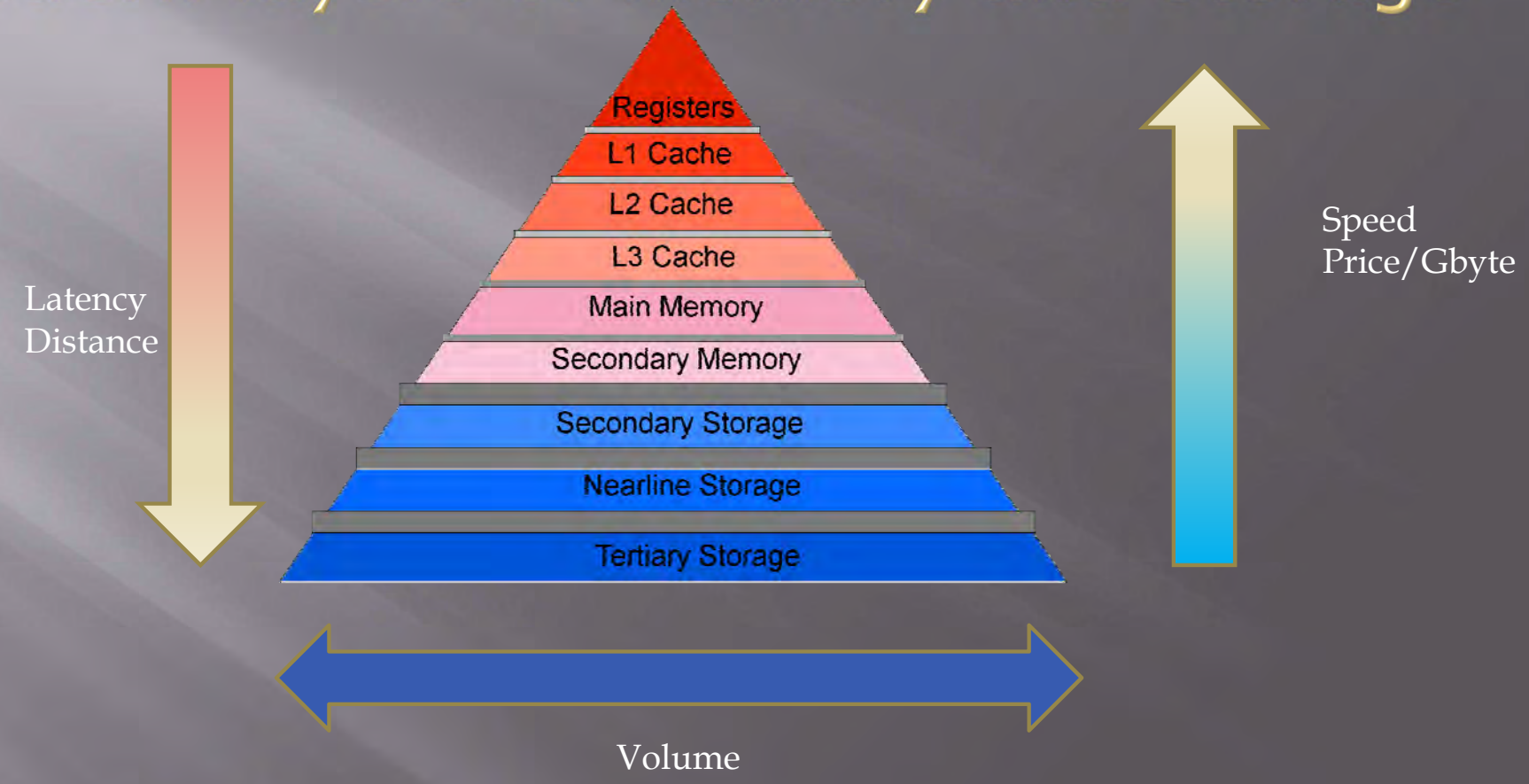
Never enough memory – UNOBTANIUM

Early memory hierarchy – IBM 650

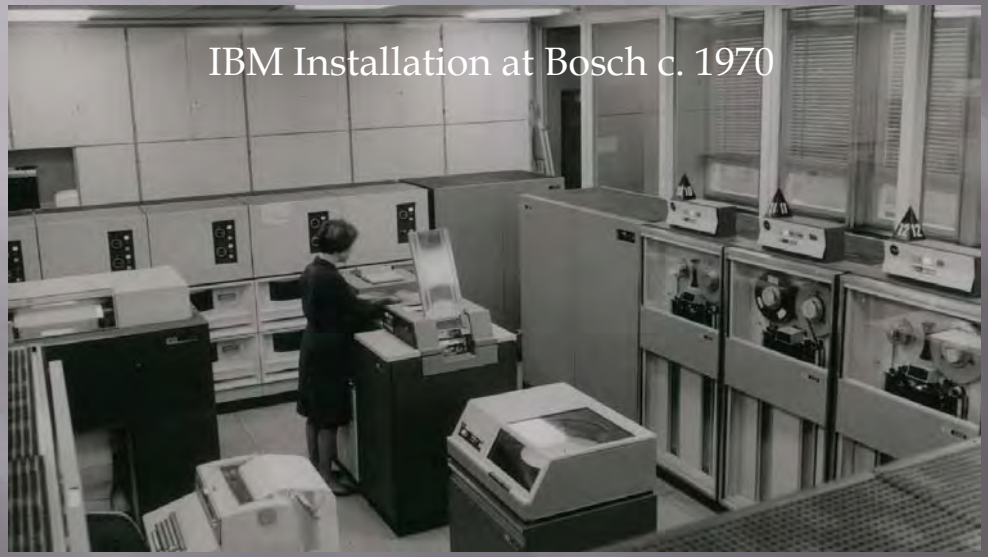




Modern layers of memory and storage

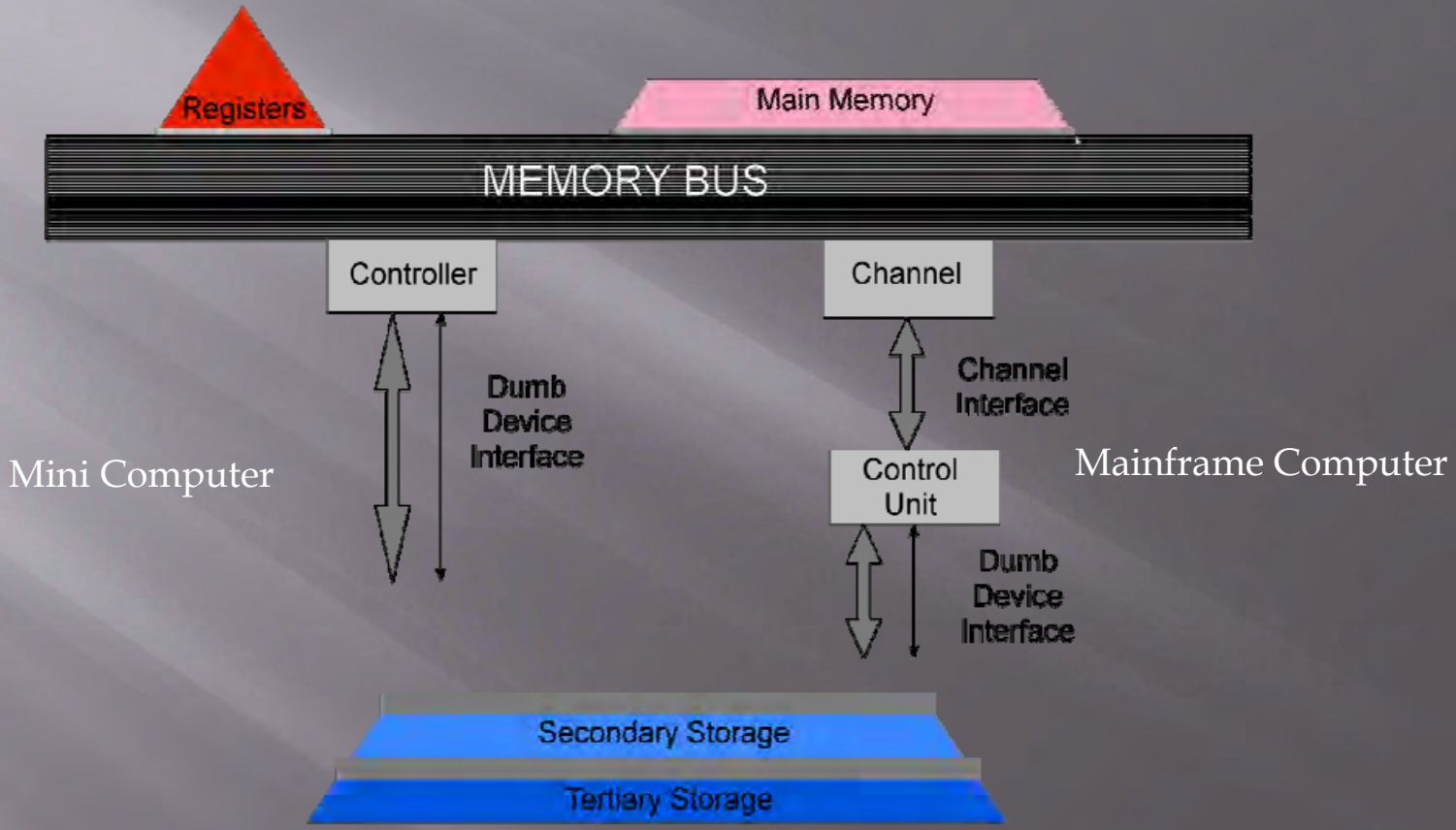


JBODs with "dumb" drive interfaces
(1960s – 1980s)

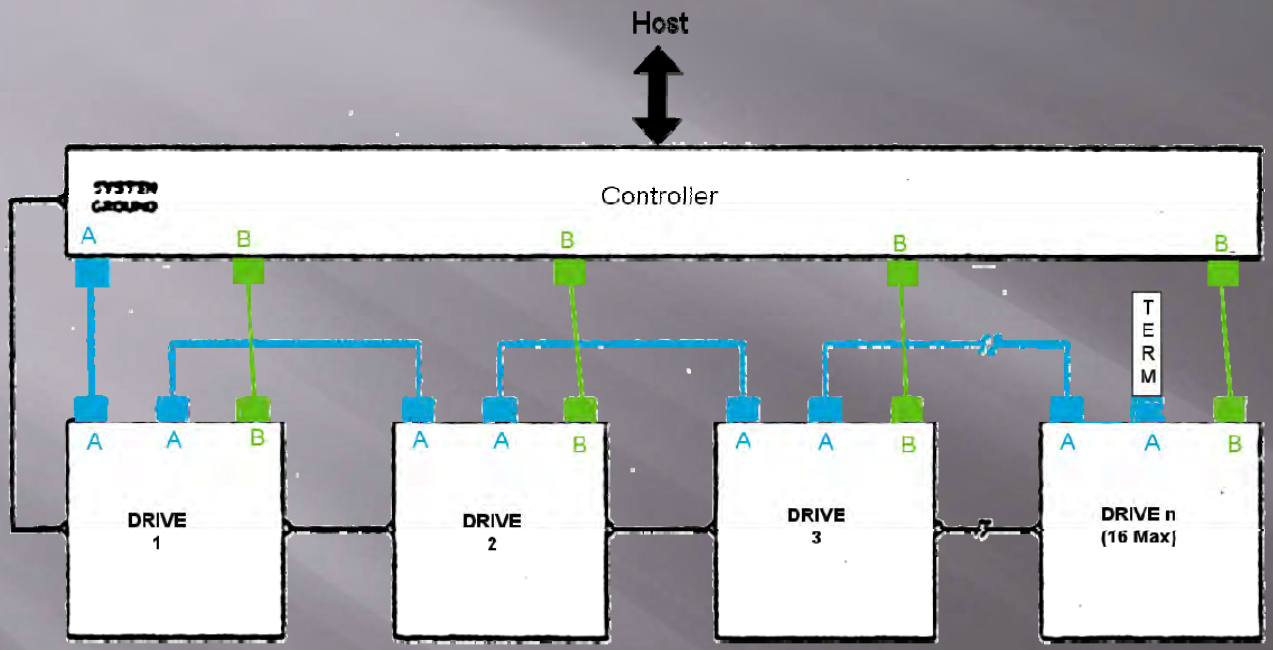

IBM Installation at Bosch c. 1970


CDC Installation at LRL, early 70's

Early defacto "dumb" drive interfaces



"Dumb" drive interface characteristics



Typically three cables:

- Power
- **Control**
  - Selection
  - Movement
  - Status
  - Start/Stop
  - Power control
- **Data**
  - Unidirectional
  - Synchronization
  - Safety

Maximum individual control cable (A) lengths = 100 ft

Maximum individual data cable (B) length = 50 ft

"Dumb" drive interfaces

"Standard" Dumb Drive Interfaces		
Date	Command Interface	Step/direction Interface
1961	7361 SCU/ 1301	
1965	2841 SCU 2302,2303, 2311,2321	
1969	DEC Controllers+ MRX 630-1 DEC RP01-3	
1973		SA900
1975	CDC SMD	
1978		SA4000
1979		ST506/ST412
1984	ESDI	

Dumb because drive specific functions in controller

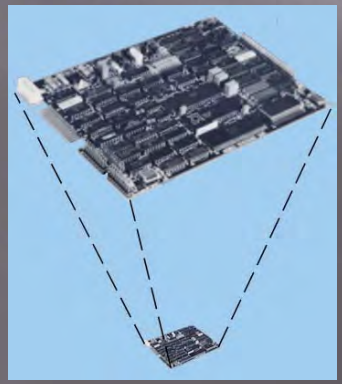
- Serializing and deserializing
- ECC/EDC generation and correction/detection
- Blocking and unblocking
- Write encoding and data separation

Barrier to technical innovation

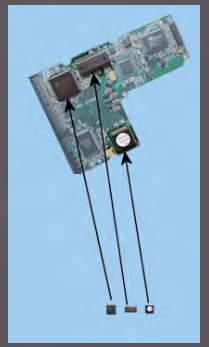
Parallel intelligent drive interfaces
(1972 – 2000s)
Move the intelligence into the drives
three examples

IBM Proprietary DCI
DEC Proprietary Massbus
Sun "Standard" IPI

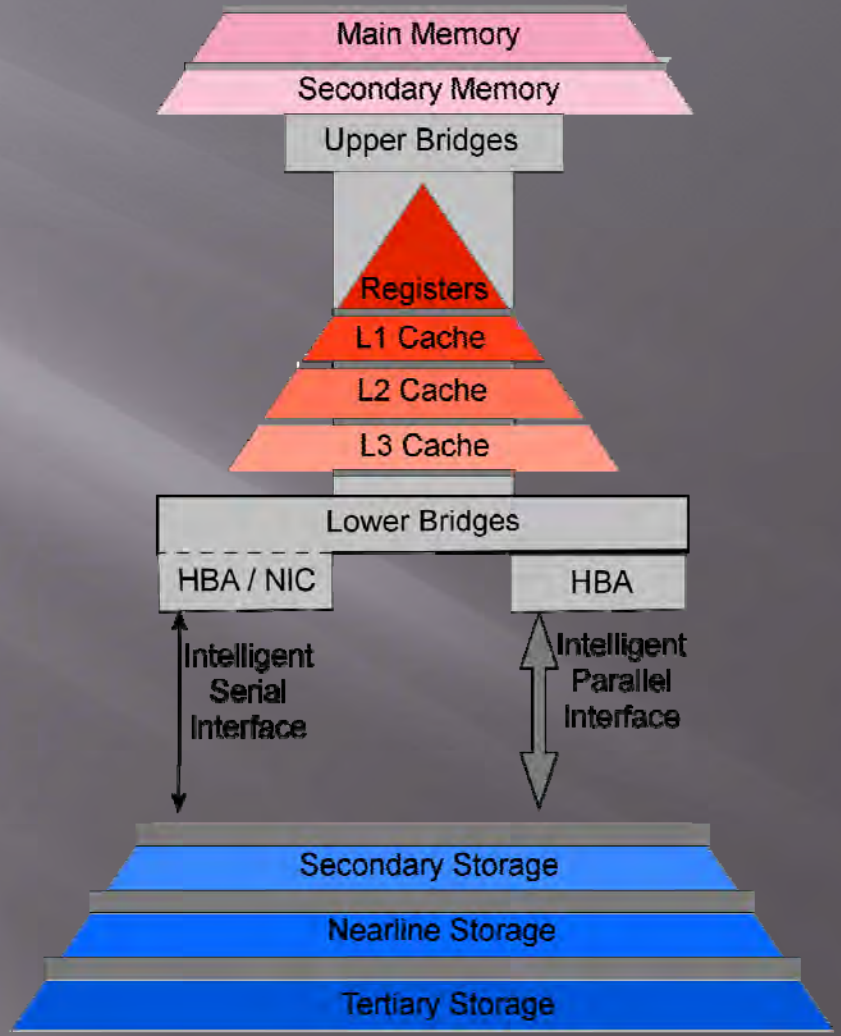
Sort of standard
SASI/SCSI
Bridge controller



ANSI Standard
SCSI
ATA

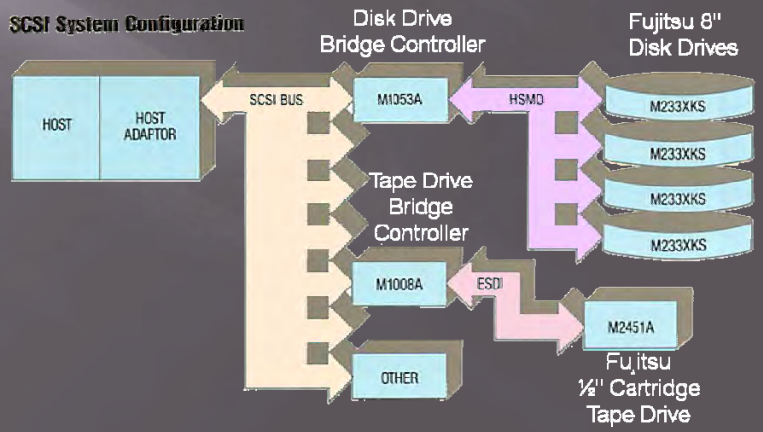


Modern Storage Interfaces



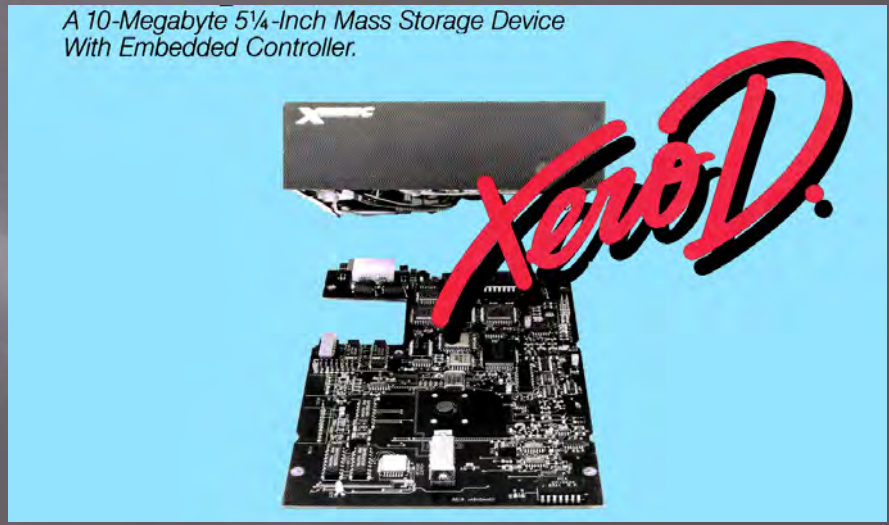
Parallel SCSI

1979	SA SASI Paper SA and NCR at ANSI
Beginning 1981?	Bridge controllers



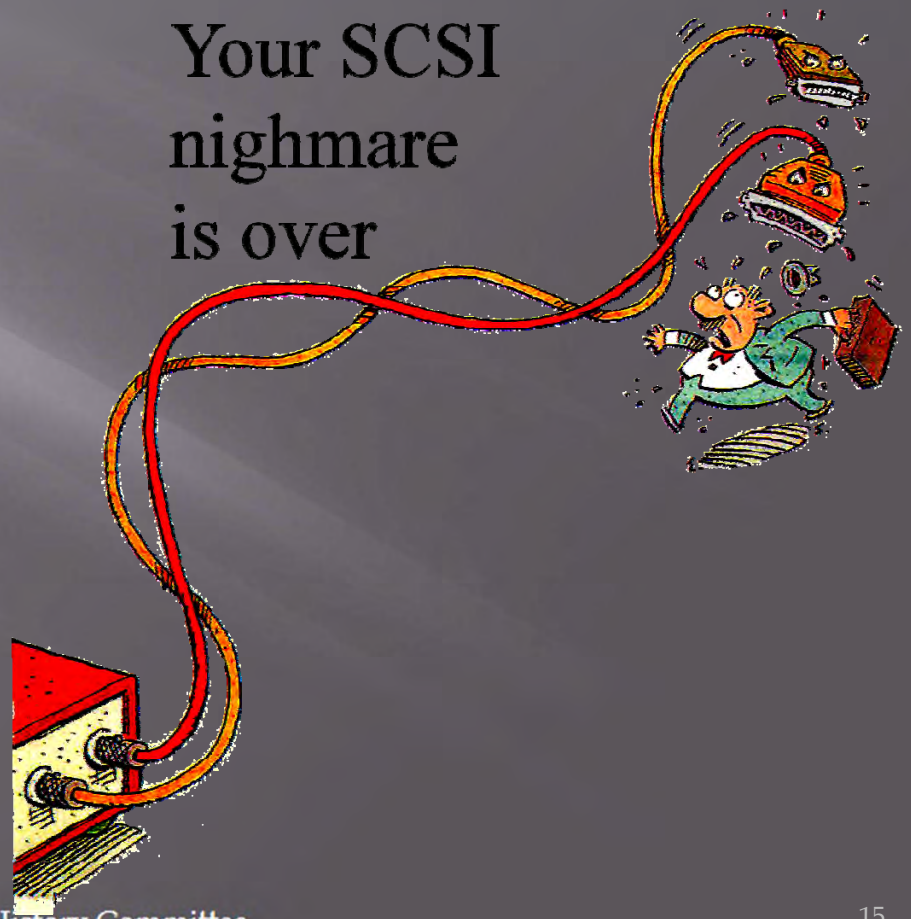
Parallel SCSI

1979	SA SASI Paper SA and NCR at ANSI
Beginning 1981?	Bridge controllers
Beginning 1984	Embedded SCSI



Parallel SCSI

1979	SA SASI Paper SA and NCR at ANSI
Beginning 1981?	Bridge controllers
Beginning 1984	Embedded SCSI
1986 (1994!)	Common Command Set Rev 4b
1990 (1995!)	CAM Committee ASPI (Adv. SCSI Programming Intf.)



Parallel SCSI

SAM – SCSI Architectural Model

X3T10/99D revision 18 November 27, 1995

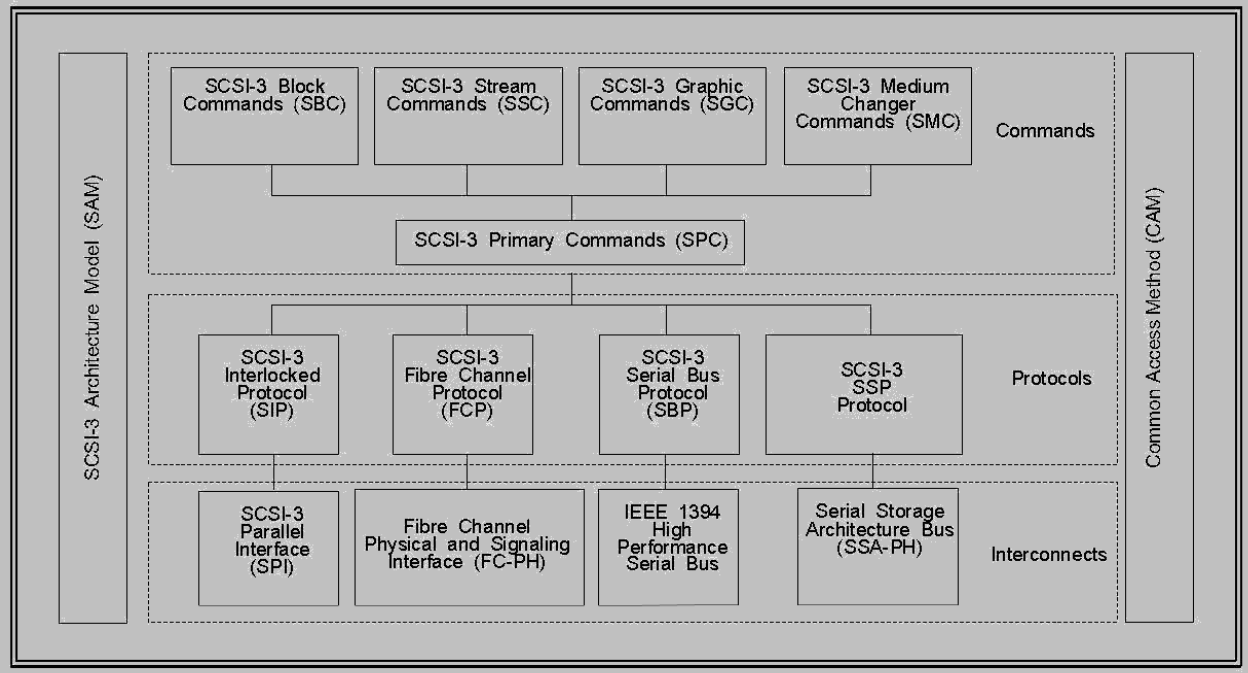


Figure 2: Functional Scope of SCSI-3 Standards

Parallel SCSI

SCSI-1 (1986) into this century

Logical Block Addressing

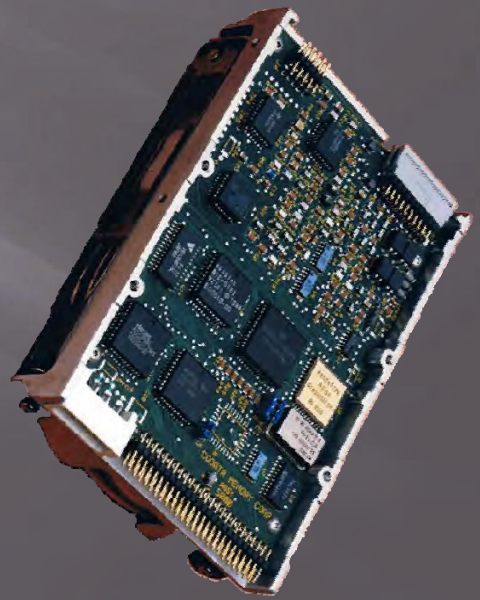
2^{24} Blocks of up to 2^{24} bytes	→	2^{64} Blocks of up to 2^{64} bytes
8 bit bus, 8 devices	→	16 bit bus, 16 devices
8 LUNs/device	→	8 byte LUN structure
5MB/sec	→	640 MB/sec?
22 HDD commands (~62 pages)	→	87 HDD commands (440 pages)

Single ended cable:	6m	→	3	→	1.5	→	0
HV differential cable:	25m			→			0
LV differential cable:			12m		→		10
2 connectors		→			8 connectors		

Parallel ATA

WD Controller w/ MiniScribe drive
Compaq Portable II
1986

Conner CP342
1987



Parallel ATA

"The ATA bus (a.k.a. IDE bus) is a disk drive interface originally designed for the ISA Bus of the IBM PC/AT"

1987	Conner publishes "Task File Interface"
1989	First presentation at CAM

CONNER PERIPHERALS INC.
CP3022
PRODUCT SPECIFICATION
REVISION IV
February, 1988

2.0 KEY FEATURES

- Block size 512 bytes.
- **IBM Task File emulation plus additional commands.**
- Emulates IBM Task File and supports additional commands.
- Up to two drives may be daisy-chained on this interface.
- Translate mode (17 sectors, 4 heads, 615 cylinders) is supported.

5 Addressing Bytes: D/H,C,C,SN,SC

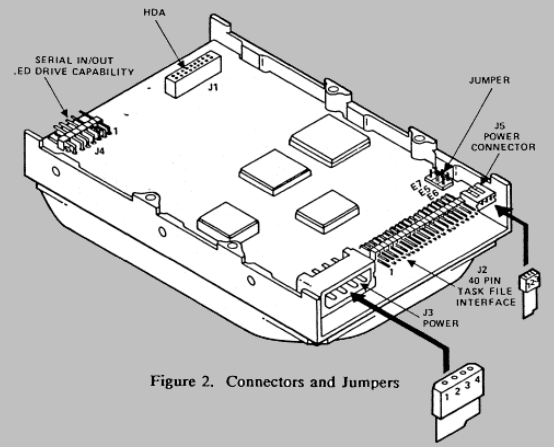


Figure 2. Connectors and Jumpers

Parallel ATA

"The ATA bus (a.k.a. IDE bus) is a disk drive interface originally designed for the ISA Bus of the IBM PC/AT"

1987	Conner publishes "Task File Interface"
1989	First presentation at CAM
1994	ATA-1 Issued
1996	ATA-2 Issued

- ATA–1 76 pages
  - 28 bit – LBA added
  - 35 Commands
  - PIO/DMA Up to 8.3 MB/sec

1994 – 1996 Spec Wars

- ATA–2 76 pages
  - 48 Commands
  - Up to 16.7 MB/sec

Parallel ATA

"The ATA bus (a.k.a. IDE bus) is a disk drive interface originally designed for the ISA Bus of the IBM PC/AT"

1987	Conner publishes "Task File Interface"
1989	First presentation at CAM
1994	ATA-1 Issued
1996	ATA-2 Issued
1998	ATA/ATAPI-4 Issued

- SFF Committee did the work ~1996
- ATA/ATAPI-4 337 pages
  - SCSI transported over ATA
  - 49 Commands
  - Up to 33.3 MB/sec

Parallel ATA

"The ATA bus (a.k.a. IDE bus) is a disk drive interface originally designed for the ISA Bus of the IBM PC/AT"

1987	Conner publishes "Task File Interface"
1989	First presentation at CAM
1994	ATA-1 Issued
1996	ATA-2 Issued
1998	ATA/ATAPI-4 Issued
2002	ATA-6 obsoletes CHS addressing

- ATA/ATAPI-6 496 pages
  - 71 non Packet Commands
  - 29 Packet Commands
  - Up to 100 MB/sec
- Last parallel spec

How the world turned serial

- 1981 – DEC SI/SDB (RA80 disk drive)
- As systems' interconnects
  - 1983 - DEC CI
  - 1988/1994: Fibre Channel
  - 1990 – IBM ESCON
- 2000-3: SATA
- 2002-4: SCSI over Fabrics (SAS)
- 2014-6: NVMe_oF

Subsystem progression

Into the 80s

JBODs

2021

Universal

Storage Server

- Fibre Channel
- FC-NVMe
- iSCSI
- FICON (CKD)

- 4.5 PB

Subsystem progression
1983 into this century

1983	DEC HSC 500	Mirror
1988	AT SS5000	Virtualization, Distributed Parity
1989	~~RAID Paper~~	
1989	Auspex 5000	NAS
1990	EMC 4200	Emulated CKD
1997	EMC Celerra	NAS/SAN Server
2001	IBM	iSCSI

The next speaker
Grant Saviers

Evolution of Storage Interfaces

Evolution of Storage Interfaces

Grant Saviers

May 2021

IEEE Storage Interfaces webinar

Which “Storage Interfaces”?

- What real time control and data functions are where?
  - Recording technology dependent? Bit modulation, clocking, detection, checks
  - Where is DMA done? CPU? Controller?
  - Error detection & recovery? Competence? Complexity?
  - RAS and failure prediction? None? 365 x 24?
- What wires & protocols at various levels? du jour? Multi-gen?
- O/S driver to controller or channel?
  - Where is logical to physical mapping done?
  - What error recovery where?
  - What performance optimization & where?
  - Application access to bits or files? TP & DB systems bits vs files

My view of the generations

- Gen 1- Logic is expensive – cost & reliability
  - “fire hose” coaxial cables & CPU dedicated to I/O
- Gen 2 – Physically big disks on big systems: > fan out, distance needs
  - Separate storage controllers, strings, channels

Sequential logic & state machines ⇧ Microprocessors + ASICs ⇩

- Gen 3 – Technology independent bus + protocols
  - Multi generation subsystem HW/SW architectures
- Gen 4 – Intelligent controllers & serial interconnects
- Gen next – what should “storage interfaces” be?

Gen 1 – IBM 1401 + 729 tape drive ~ 1957

- Processor or imbedded processor partial controller is the controller
  - CPU ALU committed to device to memory transfers
  - Program stops during I/O (except Univac I, 5 years prior)
  - Read analog signals error checked, detected, clock recovery in CPU
- Coaxial cable bus - “fire hose” size cable
  - Daisy chain, up to 8 tape drives
  - Analog read signal in same cable
  - CPU “TAU” controls real time tape motion, checks analog
  - Huge connector – “shoe” in IBM terminology, 6x6”, 10#

- Architecture and bus the same for 6 gens of 729 drives (I > VI)

Gen 2 – Disk performance ⬆ ~1965 - 83
2311, 2314, 3330, 2315, 5440 and clones

- Controller “de-jour” I/F per drive maker, size, & technology generation
  - Market leader often set the “standard” IBM, Diablo, CDC
- Motion control moves to disk drive
  - Incremental or absolute track position commands
  - Overlapped simultaneous seeks

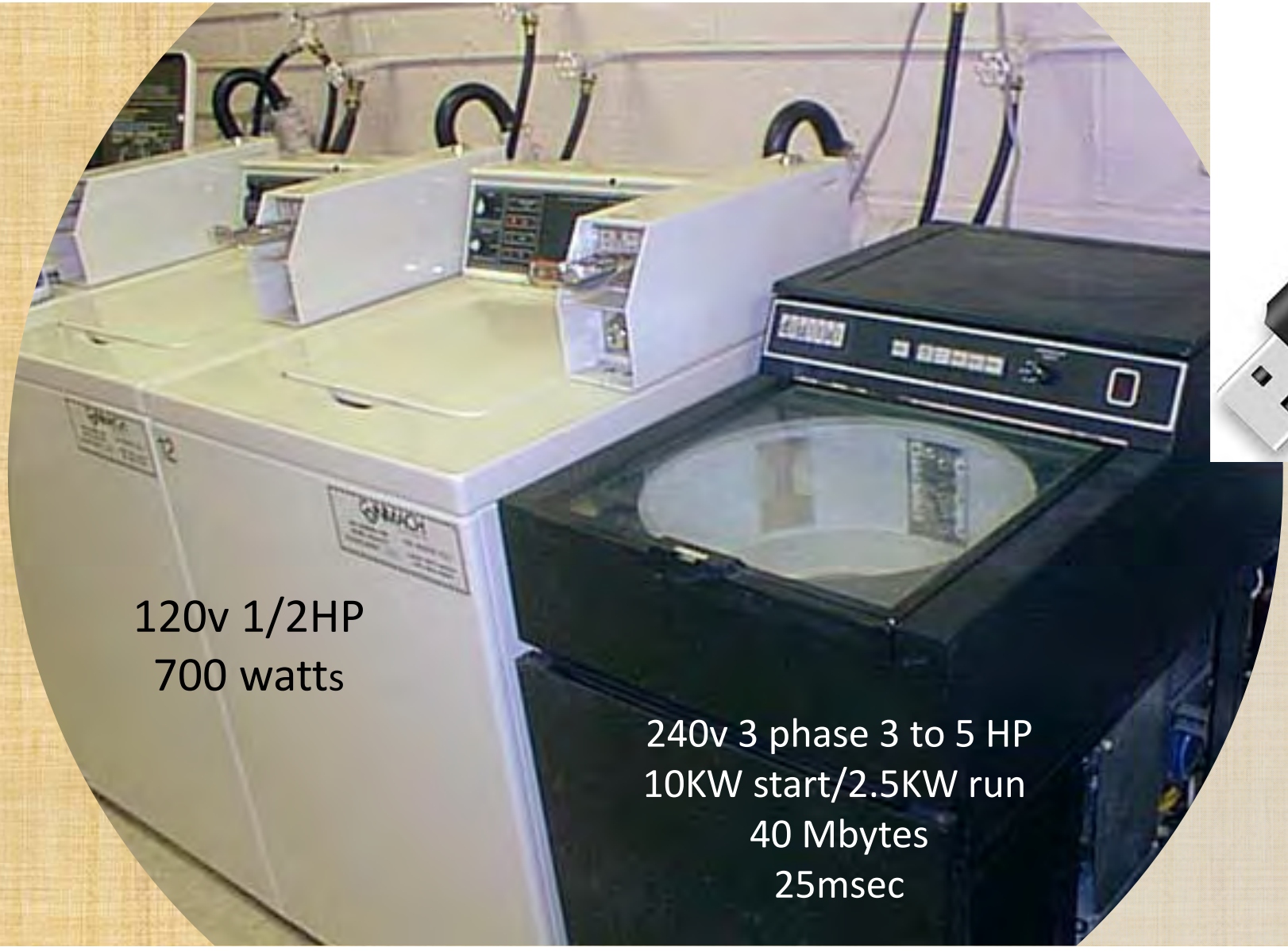
- Read peak detectors usually in device - some areal density independence
  - BUT Clock recovery in controller & write clock & encoded data from controller
  - SERDES, error detection in controller, 1 data transfer at a time
- Single ended digital daisy chain bus; typically 8 devices per contoller
- String controllers and channels emerge (high perf tape similar)
  - Large fan-outs to dozens of drives, simultaneous parallel data paths

Gen 3 – Technology independent storage bus

- DEC controller and driver proliferation was a big problem/mess/expensive
  - 16, 18, 32, & 36 bit CPU architectures
  - Many O/S – 16b 5x, 18b 2x, 32b 2x, 36b 2x = cost, complexity, & incomplete
  - Goal “universal storage bus”
- ~1973 Massbus – tape, disk, bulk RAM mixed on single storage device bus
  - Multi tech & multi generation – any mix per controller
    - 2314, 3330, HPT, 3350, 9370/SMD and “double density”
    - NRZI & PE & GCR tape drives
  - Daisy chain, 8 devices, twisted pair differential, 16+P control + 18+P data + 4 timing
  - One controller per CPU implementation – eg RH11, RH70, RH780 (up to 4 per)
- Mixed 2314, 2314-2, 3330, 3330-2, SMD, & 3350 on a single controller were common

DEC Massbus (MB) Implementation

- SERDES, peak detector, clock recovery moved into disk drive
- IBM PCM drive I/F's modified to DEC spec. 2314s & 3330s & SMDs
  - "Bustle" cabinet bolted to side of drive (from MRX, CDC, ISS)
- Seek, block access, in drive (not CKD capable)
  - Dual port drives for redundancy and performance
  - 1 at a time data transfer, but overlapped control
  - Tape subsystem had head of string controller
- Tape remained "du-jour" at device with MB head of string control
- DEC HPT disk and 3350 had native MB



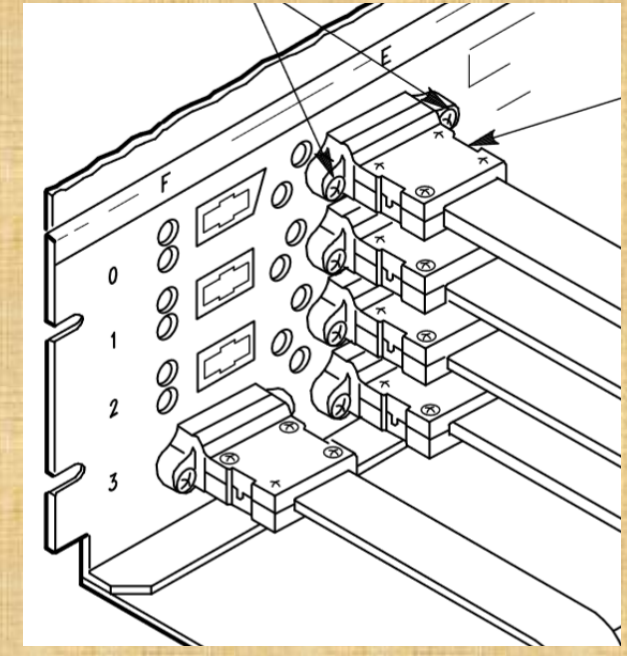
120v 1/2HP
700 watts

240v 3 phase 3 to 5 HP
10KW start/2.5KW run
40 Mbytes
25msec

5v 0.5 watts
1 Terabyte
200usec

Gen 4 – Intelligence & serial ~ 1981

- DEC Digital Storage Architecture (DSA) & Clustering
- Transactions/sec, I/Os per second leadership goals
  - More devices, further away, increased security & reliability
  - RAS including failure prediction, real time monitoring, & automatic failure recovery & redundancy
- Serial interconnects (DEC “SDI/STI”) are feasible
  - Silicon ASICs are cheaper than copper wires
  - Radial connections to devices & dual port
  - Small cables & connectors for higher fan-ins
  - Transformer coupled for EMC “bombproof”



DEC DSA 1981 – Multi tech & gen/high perf.

- Disks as perfect logical block storage (like SSD now)
  - Logical to physical mapping moved to subsystem
  - Revector bad blocks and tracks into spare space
  - Standardized “on disk” format: spares, error logs, diagnostics (0.4% data loss)
- Robust architectural definition –
  - Mass Storage Control Protocol - MSCP, Class & Port drivers
  - CPU architecture independent
  - Maximum performance optimization

DSA - Increased performance and function

- HSC = Hierarchical Storage Controller – 5 levels
  - Cache, disk, tape, bulk ram, tape libraries
- VAX Cluster Controllers (HSC40 thru HSC95)
  - N-way Mirroring, inline auto restore, rebuild, & catch-up
  - Online mirror volume backup/restore to tape
  - Storage device diagnostics, error logging, revector
  - Multi drive I/O queues, multi level access optimization
  - High fan in/out: up to 56 devices; multiple HSC per cluster
  - Multiple simultaneous data paths of 4 or 8 devices per path
  - Later gen HSCx5 incorporated SCSI interfaces
- Many storage generations – at all levels of hierarchy
- Also “simpler” controllers – Unibus, Qbus, BI bus

And there was/is SCSI

- Somewhat technology independent
  - Disk, tape, CD, & DVD storage
  - RAS - None to limited
  - Many connector “standards”
- Slogged through many generations –
  - 8, 16 or 32 bit wide bus running at 5, 10 or 20 MHz
  - PIO evolved to DMA controllers
  - Narrow evolved to wider daisy chain busses 8 to 16 devices
  - Single ended > LVDS signals
- SAS, SSA, iSCSI architectural melding
- Dumb bits/blocks, & evolved versions for “server” grade RAS & 10e6 MTBF

What should disk storage look like?

- Gordon Bell circa 1983 – “Why aren’t you putting the file system in the disk drive?”
  - VMS file system engineer – “you will get the code only from my dead hands”.
  - Auspex, then NetApp and now the cloud as “perfect” file storage
- SATA proliferation – du jour incrementalism
- Many RAID flavors (DEC sponsored research@ Berkeley)
  - Why when redundancy in cloud and mirroring are so cheap?
  - No on disk data standard means SOL for some failures.

Presentations continued in
Storage Interfaces 20210511 Part 2.pdf

Jai Menon

Evolution of Block Storage System Interfaces