IBM Systems & Technology Group

A SHAREd History of the Mainframe – Chronicles, Artifacts, and Stories

John Eells
IBM Poughkeepsie
SHARE 116, Session 9022
March 2, 2011

Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.

AIX*
AnyNet*
CICS*
DB2*
developerWorks*
DFSMSdfp
DFSMSdss
DFSMShsm
DFSMSrmm
DFSORT*
DRDA*
DS8000
ESCON*
FICON*
FlashCopy*
GDPS*
HiperSockets
Hiperspace
HyperSwap
IBM*
IBM eServer
IBM logo*
IMS
Infoprint*
IP PrintWay
iSeries
Language Environment*
MVS
Notes*
OMEGAMON*
Open Class*
Parallel Sysplex*
PR/SM
Processor Resource/ Systems Manager
pSeries*
RACF*
Redbook
RMF
System i
System Storage
System z
System z9
System z10
SystemPac*
Tivoli*
VTAM*
WebSphere*
z/Architecture*
z/OS*
z/VM*
zSeries*

* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies.

InfiniBand® is a registered trademark of the InfiniBand Trade Association (IBTA).
Intel is a trademark of the Intel Corporation in the United States and other countries.
Linux is a trademark of Linux Torvalds in the United States, other countries, or both.
Java and all Java-related trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc., in the United States and other countries.
Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation.
UNIX is a registered trademark of The Open Group in the United States and other countries.
All other products may be trademarks or registered trademarks of their respective companies.
The Open Group is a registered trademark of The Open Group in the US and other countries.

The Trademark Blackberry® is owned by Research In Motion Limited and is registered in the United States and may be pending or registered in other countries. IBM is not endorsed, sponsored, affiliated with or otherwise authorized by Research In Motion Limited.

Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

This presentation and the claims outlined in it were reviewed for compliance with US law. Adaptations of these claims for use in other geographies must be reviewed by the local country counsel for compliance with local laws.

Where we are today…

z/OS 1.13 Preview*

A smarter operating system with designs for:

Improving Usability and Skills

z/OSMF Software Deployment and Storage Management applications, User-level mount command for z/OS UNIX System Services, Automatic F4DSCB updates, SDSF Sysplex functions to work without MQ, Catalog parmlib member, Better O/C/EOV Messages, Health Checks, …

Integrating new Applications and Supporting Industry and Open Standards

Java/COBOL interoperability, Improved Support for unnamed sections, ISPF Edit Macros, Subsystem and Unauthorized XTIOT support, dbx hookless debug, DFSORT improvements, Job level return codes, …

Scalability & Performance

Fully-shared zFS in a sysplex, RMODE 64 extensions, IFASMFDL improvements, 500K+ aliases per user catalog, Larger VVDSs, FREE=EOV, FTP support for large format data sets and EAS,…

Enhancing Security

RRSF over TCP/IP, LDAP improvements, SAF security for z/OSMF, NAS address checking and encryption negotiation, New restricted QNAMEs, PKI support for DB2 backstore, ICSF support for new HMACs, FTP & TN3270 password phrase support, …

Improving Availability

Warn before TIOT exhaustion, CMDS enhancements, Parallel FTP for dump transfers, PFA ENQ tracking, RTD improvements, zFS Refresh, DADSM Dynamic Exits, JES3 dynamic spool addition, Better channel recovery, More ASID reuse, …

Self Managing Capabilities

WLM and RMF to provide response time distribution for all goals, DFSMShsm Journal Backup and space management improvements, …

Extending the Network

IDS IPv6 support, NAT Traversal for IKEV2, NMI extensions, More VLANs per OSA port, more 64-bit TCP/IP, EE improvements, …

* All statements regarding IBM future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

z/OS and IBM zEnterprise Functions and Features[1]

Five hardware models
Increased capacity processors
Up to 15 subcapacity CPs at capacity settings 4, 5, or 6
Up to 3 TB RAIM (real) memory[2]
6.0 GB/sec InfiniBand® I/O interconnect
8 slot, 2 domain I/O drawer
Concurrent I/O drawer add, remove, replace
Optional water cooling
Optional High Voltage DC power
Optional overhead I/O cable exit
Up to 80 processors per server configurable as CPs, zAAPs, zIIPs, IFLs, ICFs, or SAPs (up to 32-way on R7, 64-way on R9, 80-way on R11)
New and enhanced instructions

z/OS exploitation in blue

Capacity Provisioning enhanced[4]
Three subchannel sets per LCSS[3]
Platform Management from HMC
CFCC Level 17 enhancements[4]
Up to 128 Coupling Link CHPIDs
Improved processor cache design
Power save functions
Crypto Express3 enhancements[5]
Secure key HMAC Support
Elliptic Curve Cryptography (ECC) Digital Signatures[3]
CPACF enhancements[5]
Out of order instruction execution
z/OS discovery and auto-configuration (zDAC)[3]
OSA-Express3 Inbound Workload Queuing (IWQ)[3]

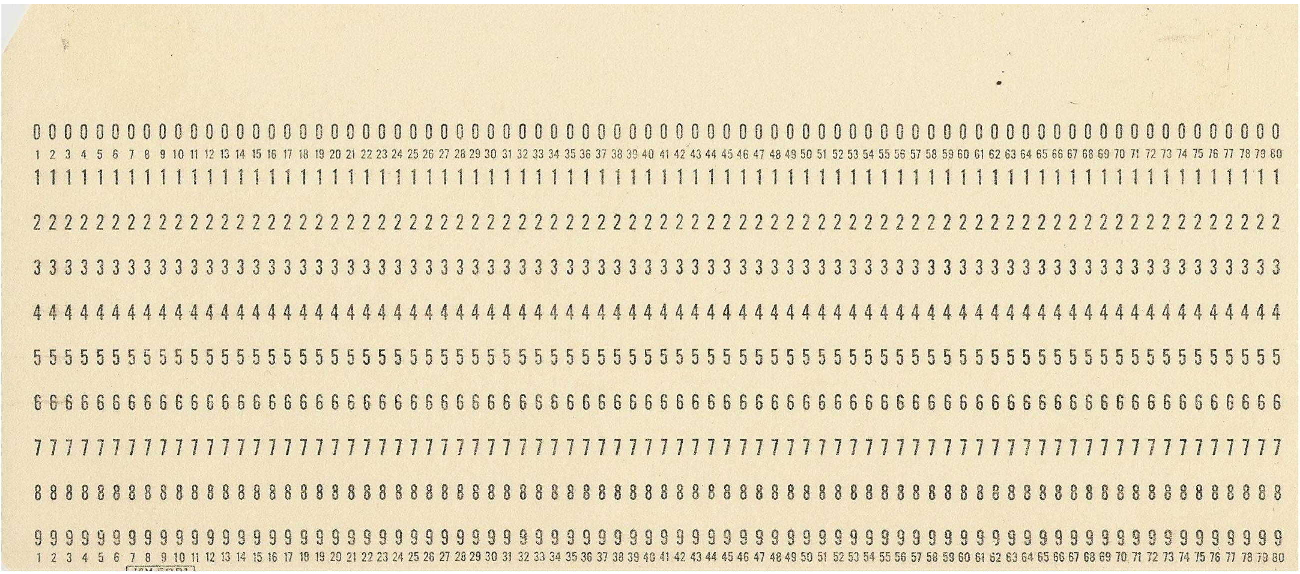
1. z/OS R7 and z/OS R8 support require IBM Lifecycle Extension for z/OS (5637-A01 or 5638-A01). PTFs required for z/OS R8-R12; refer to the PSP.
2. Maximum of 1 TB per LPAR. Maximum supported by z/OS R7 is 512 GB. z/OS R8 and later are designed to support up to 4 TB per image.
3. z/OS R12 required
4. z/OS R12, or R10 or later with PTFs required
5. Cryptographic Support for z/OS V1.10 through z/OS V1.12 Web deliverable with the PTF for APAR OA33260 required.

Where we've been…

It all started with…

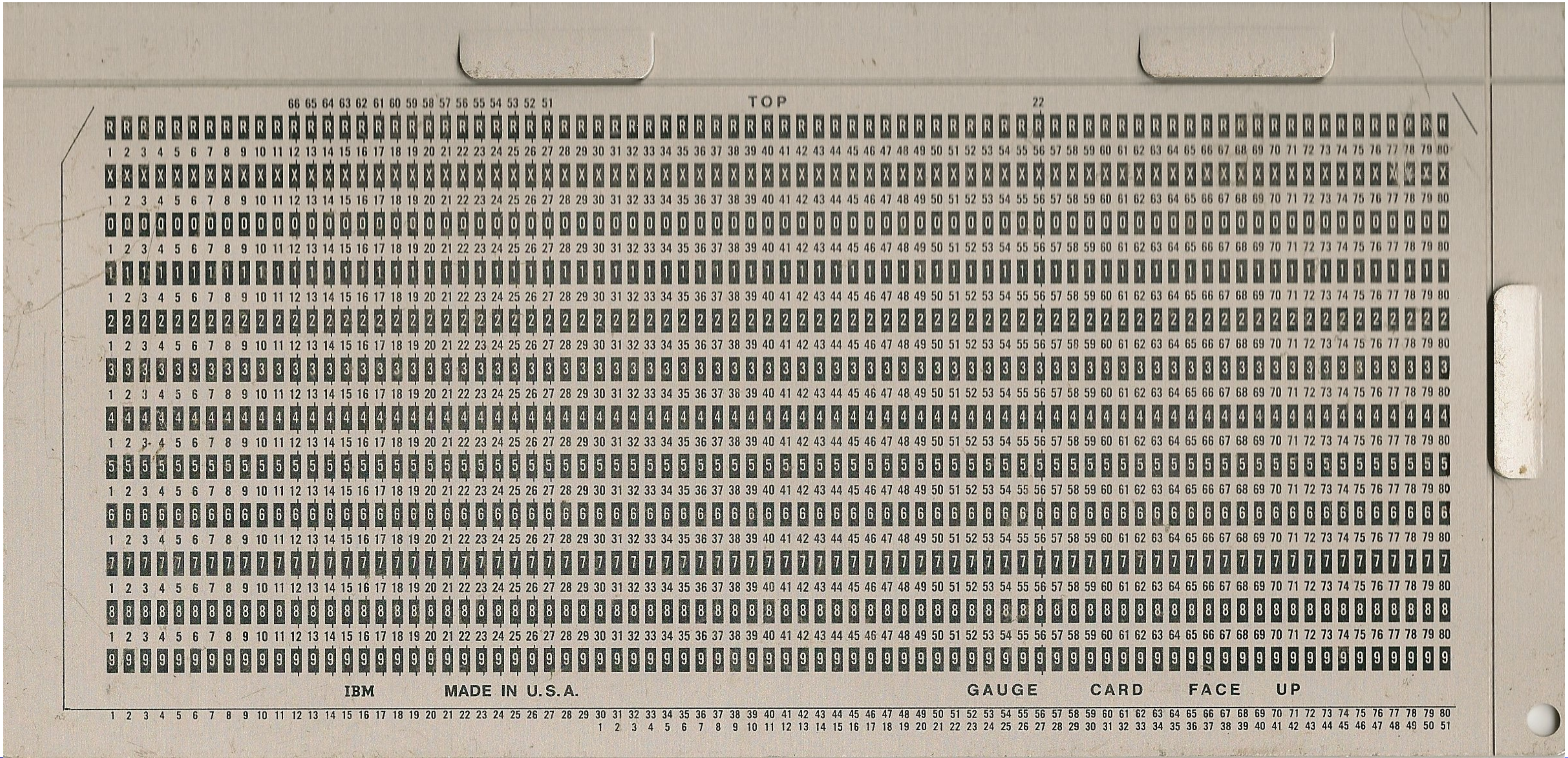
- …Herman Hollerith’s punched cards…
- …and their influence continues to affect us today!

- Ever wonder…
  - Why the 3270 default screen size is 24x80?
  - Why we have a “block size” concept?
  - Why we have data sets with sequence numbers?



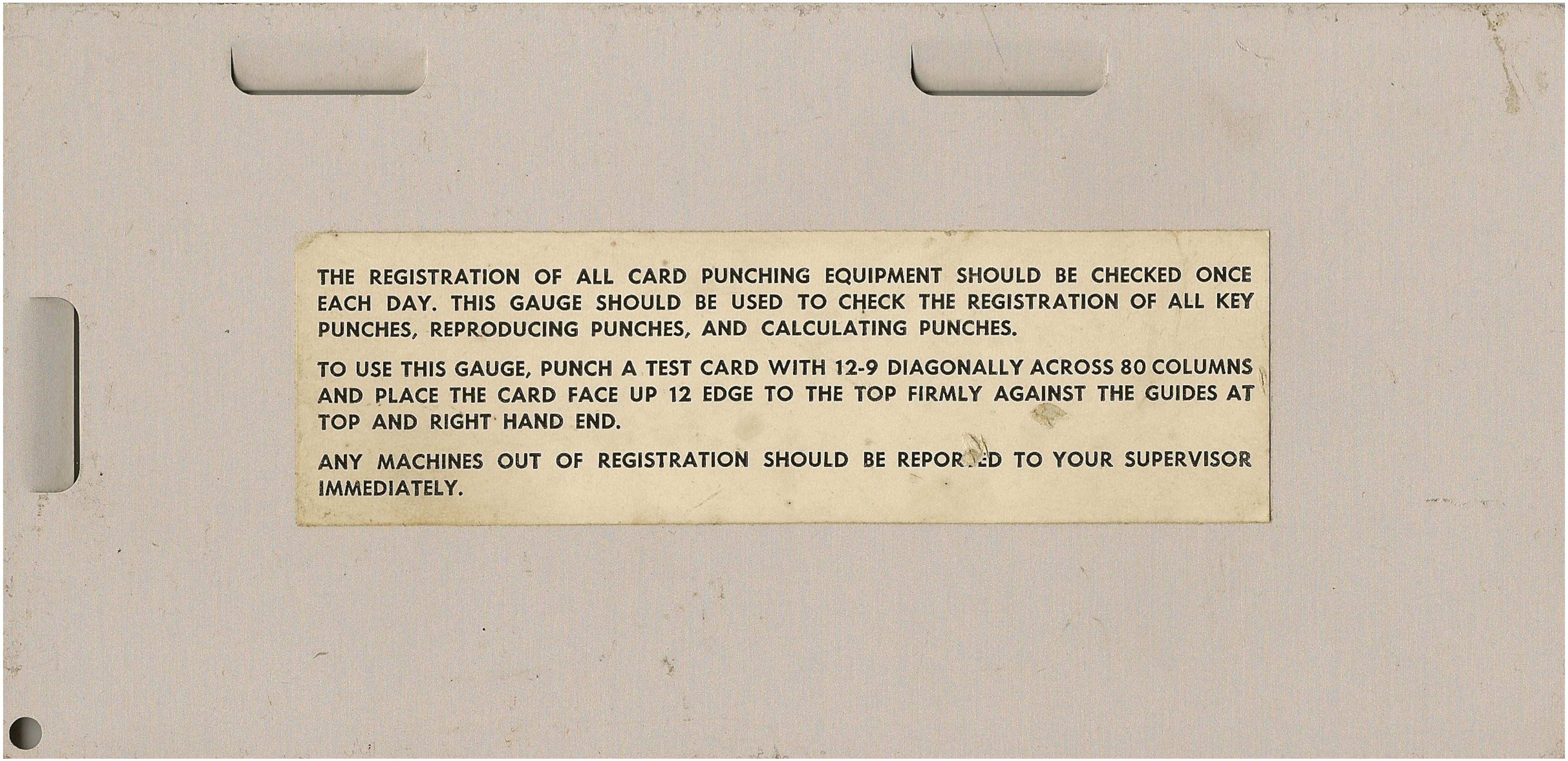
Punched Cards, Continued…

- How did you know the holes were in the right places?
- With a card registration plate, of course!
- Still standard issue in the 1970's



Punched Cards, Continued…

Rear view of card registration plate



Punched Cards, Continued…

- It's hard to believe this now, but punched cards were <u>pervasive!</u>
- Many bills and warranty cards were printed on punched cards
- "Do not fold, spindle, or mutilate…"
- This card came with my garbage disposal many moons ago:

What's a Card Jam?

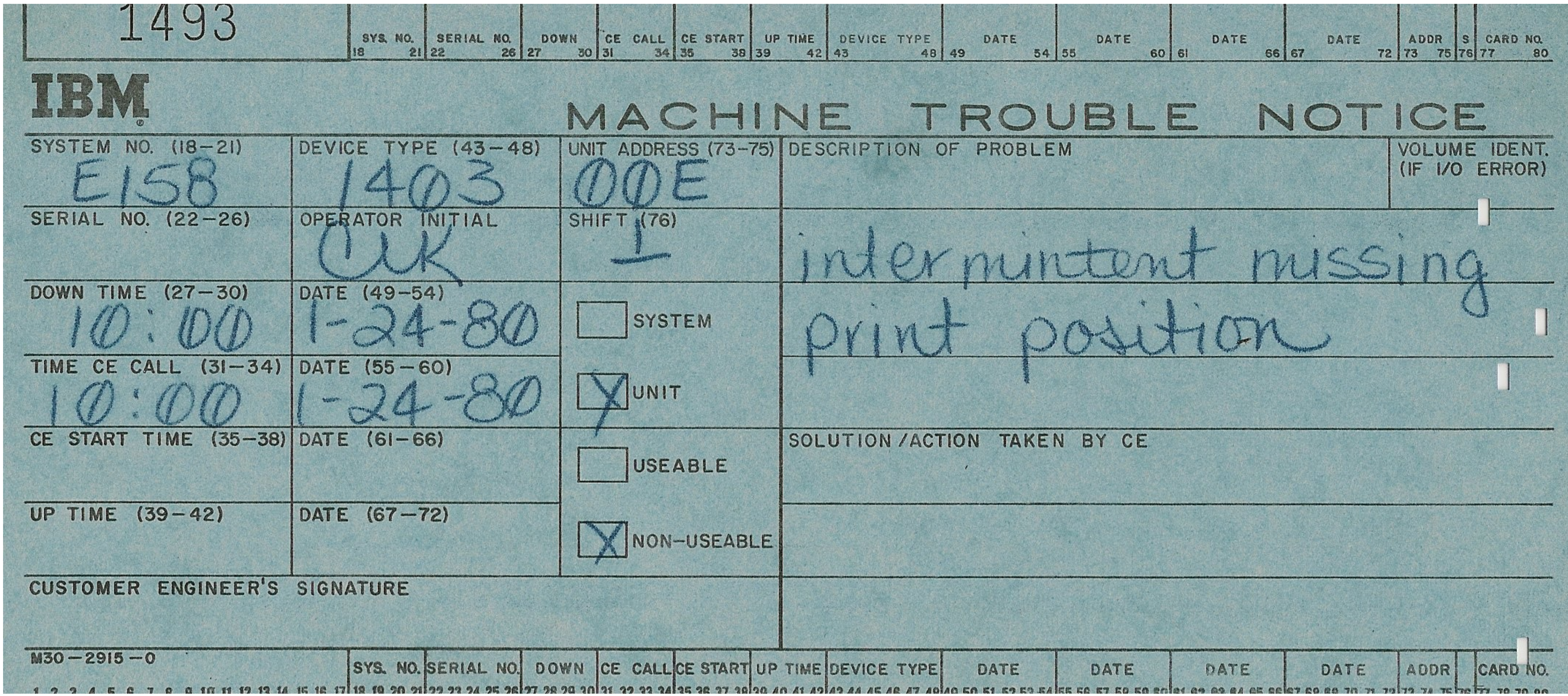
• When two cards tried to fit into the space meant for one, how did you get them out?

• You used a card saw…

• Once standard issue in CE tool bags, the thin (.010" or so), spring-steel card saw was essential if you worked on card readers, punches, or keypunch machines.

• It would clear out the card jam…eventually.

- Of course, IBM used punched cards, too:



1493

SYS. NO. 18 21 | SERIAL NO. 22 26 | DOWN 27 30 | CE CALL 31 34 | CE START 35 38 | UP TIME 39 42 | DEVICE TYPE 43 48 | DATE 49 54 | DATE 55 60 | DATE 61 66 | DATE 67 72 | ADDR 73 75 | S 76 | CARD NO. 77 80

IBM

MACHINE TROUBLE NOTICE

SYSTEM NO. (18–21): E158

DEVICE TYPE (43–48): 1403

UNIT ADDRESS (73–75): 00E

DESCRIPTION OF PROBLEM: intermintent missing print position

VOLUME IDENT. (IF I/O ERROR)

SERIAL NO. (22–26)

OPERATOR INITIAL: CK

SHIFT (76): 1

DOWN TIME (27–30): 10:00

DATE (49–54): 1-24-80

SYSTEM

TIME CE CALL (31–34): 10:00

DATE (55–60): 1-24-80

X UNIT

CE START TIME (35–38)

DATE (61–66)

USEABLE

SOLUTION / ACTION TAKEN BY CE

UP TIME (39–42)

DATE (67–72)

X NON–USEABLE

CUSTOMER ENGINEER'S SIGNATURE

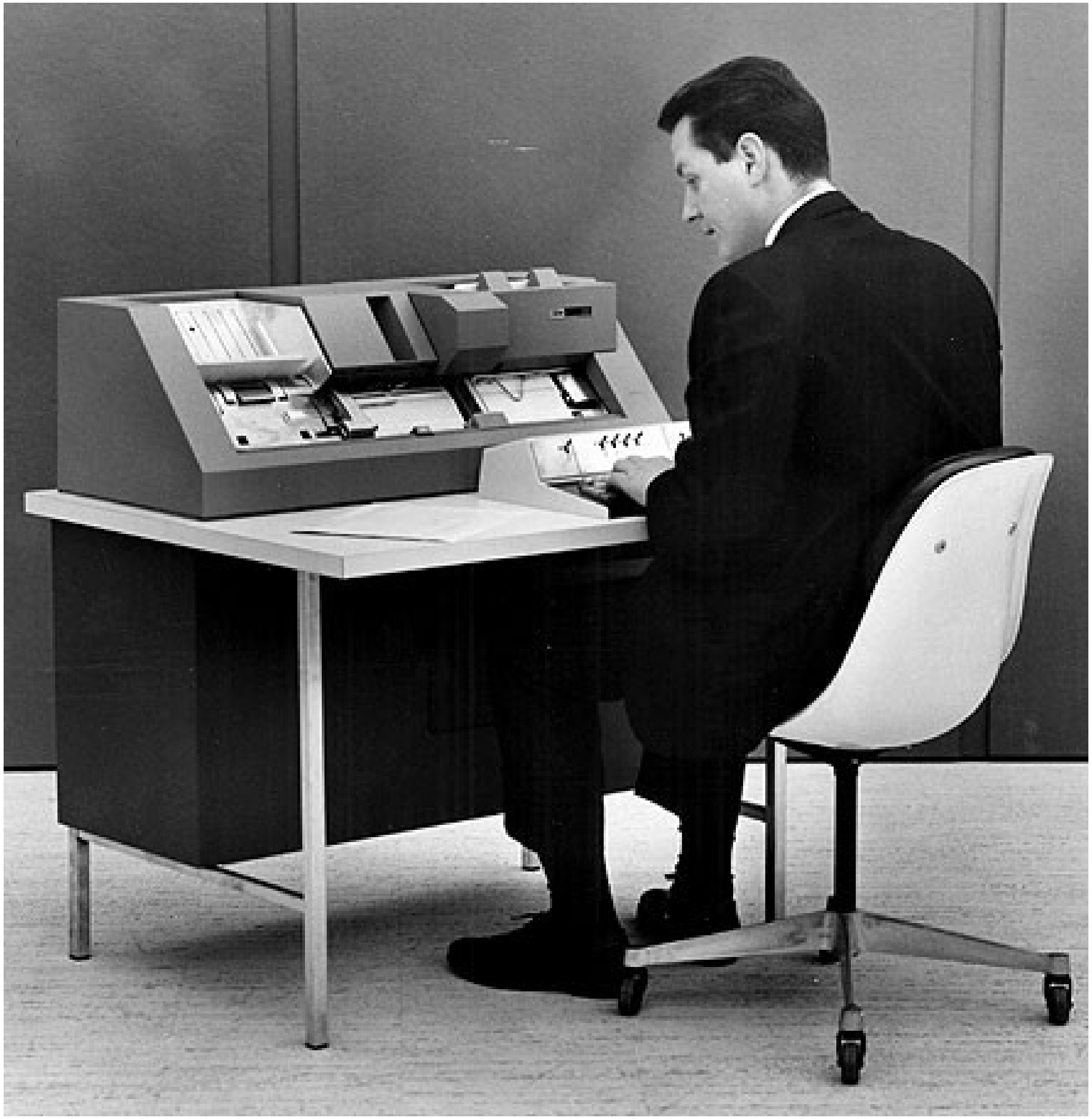
M30–2915–0

SYS. NO. | SERIAL NO. | DOWN | CE CALL | CE START | UP TIME | DEVICE TYPE | DATE | DATE | DATE | DATE | ADDR | CARD NO.

An operator named Carol K. wrote this MTN against a printer I fixed in 1980…and I obviously forgot to return the card because I found it in my old tool bag in 2007!

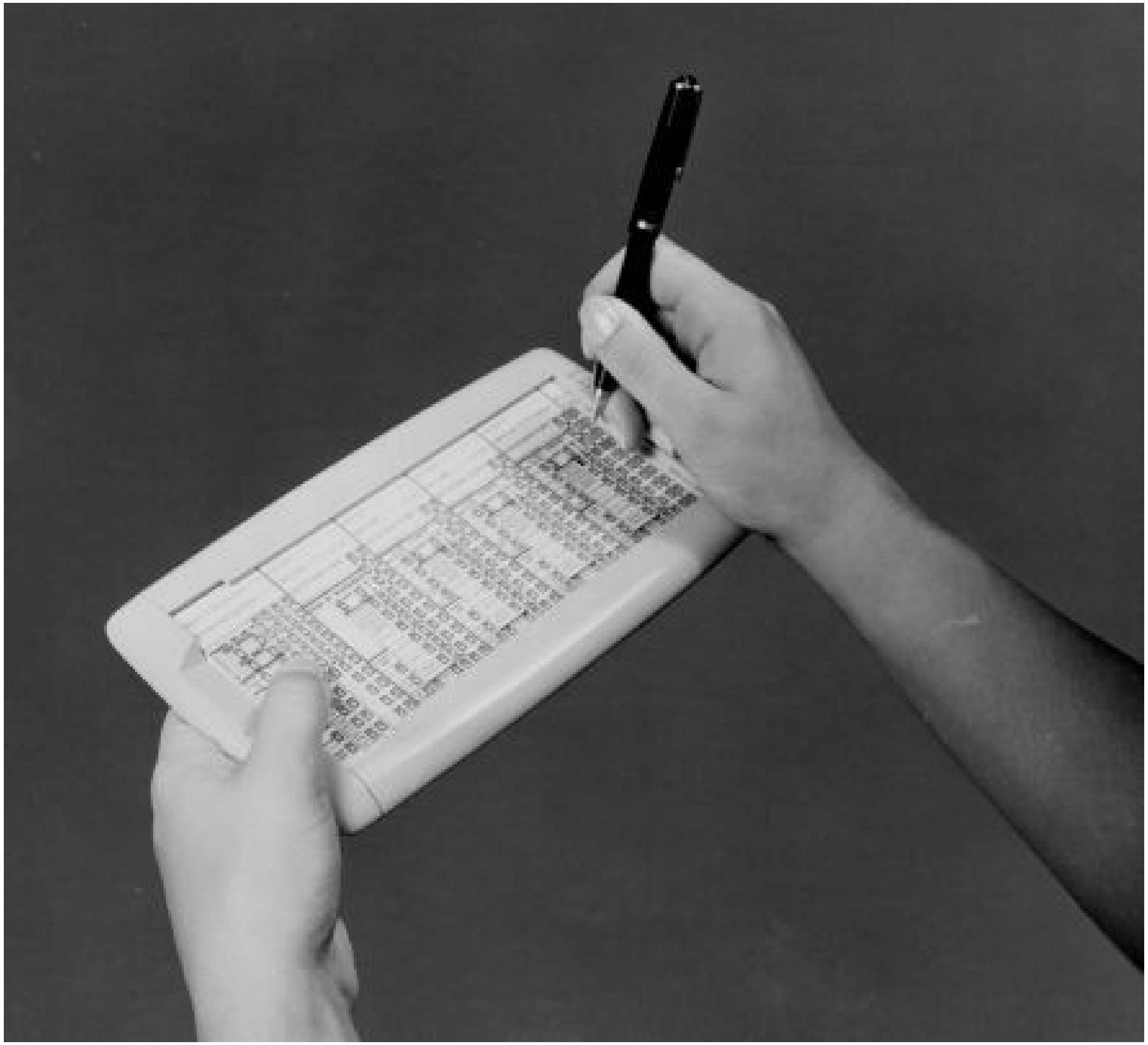
Punched Cards, Continued…

- An IBM 029 Keypunch, 1964
- Not exactly a laptop!
- It existed only to punch holes in cards
- Blank cards in feeder on top right; punched ones in stacker on left; chad bin underneath
- No error correction, of course; cards with typos went into the trash can (which is conspicuously absent in this photo)
- This is actually a model with an optional drum-mounted “template card” (I can’t recall the actual name) to speed things up



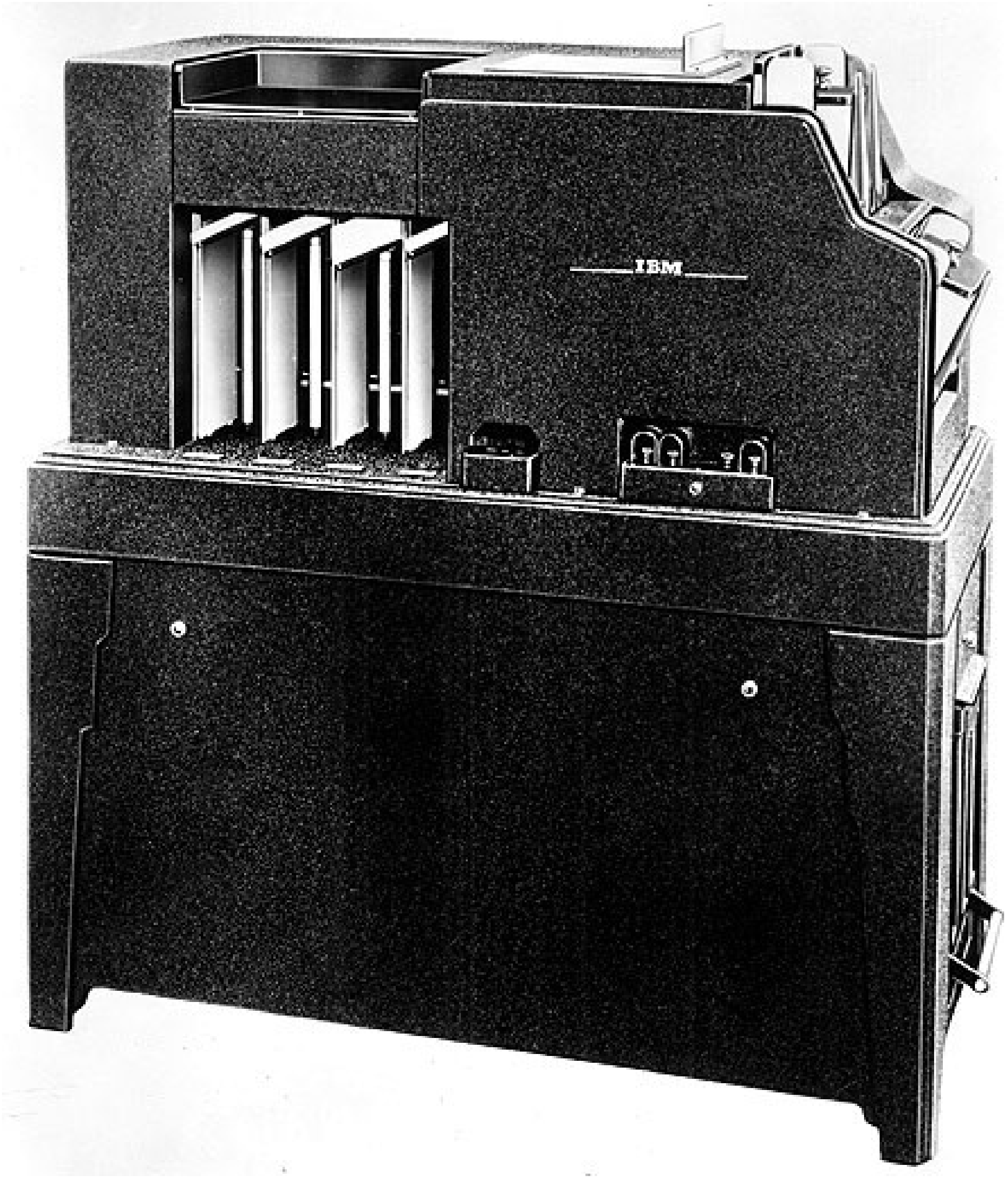
Punched Cards, Continued…

- There were no PDAs, then, but there was…
- The Port-A-Punch!
- "Designed to fit in the pocket"
  - I suppose pockets might have been larger then (some things were even before my time, after all)
- Not exactly a BlackBerry® handheld device!



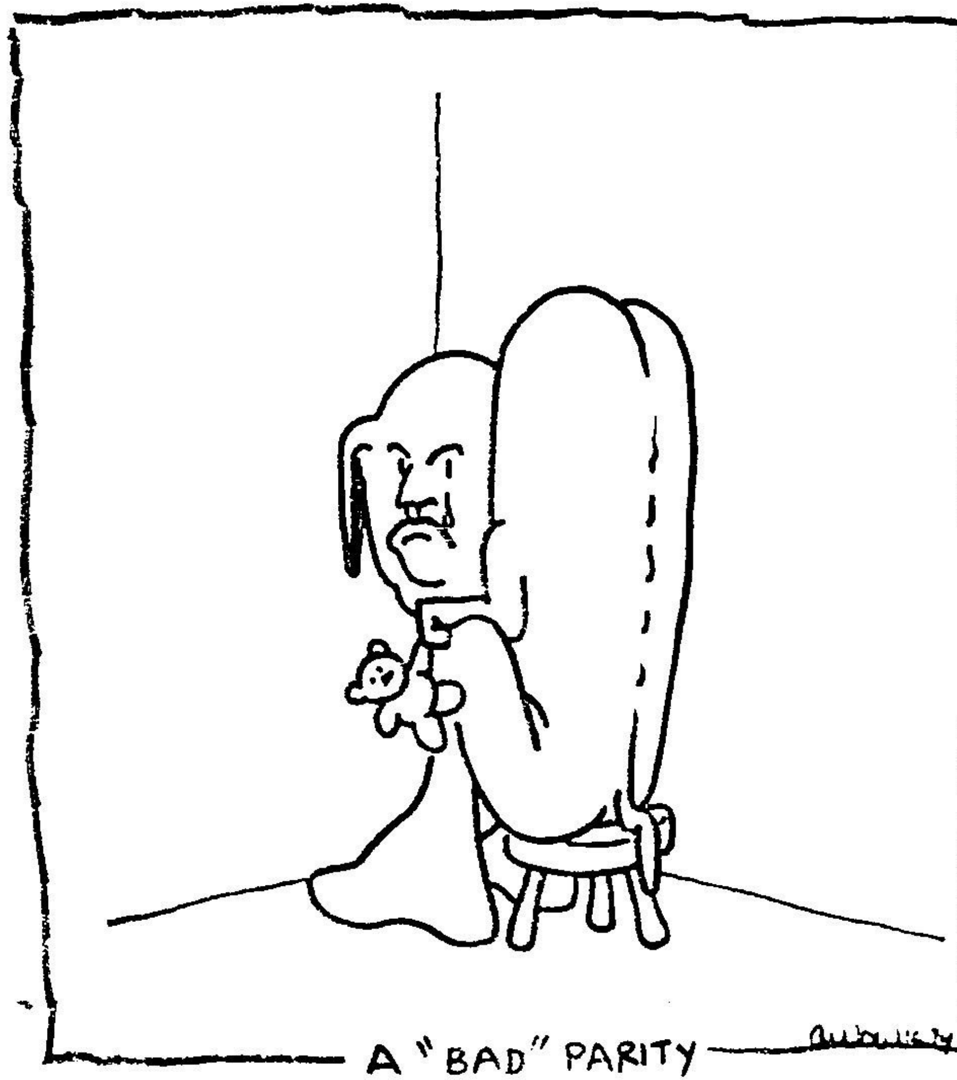
Punched Cards, Continued…

- An IBM 77 Collator
- A collator is the opposite of a sorter
- For some things, you didn't need a computer to make punched cards useful
- Today, we use things like SORT/MERGE's descendant, DFSORT



Hardware

- The industry did have to invent some things:
  - Parity (IBM uses odd parity)
  - NRZI recording for tape
  - CRC and LRC checking algorithms
  - ECC memory
  - Microcode
  - …the list goes on and on



Printers

- Printers like this 1403 came with a print train or print chain
- A hydraulic unit—sort of a 2-speed mini-automatic transmission—drove the tractors to move the paper
- Spacing and skips were controlled by a 12-channel carriage tape
  - It was just *amazing* how fast a box of paper could empty when one broke…
- Don’t leave a cup of coffee on top!
  - Some models raised the cover automatically when out of paper to catch an operator’s attention

Disk drives

- The 2314
- 9 drawers to a “bank” (because they were high maintenance, you could only use up to 8 at a time)
- Hydraulic pistons moved heads in & out of removable disk packs
- Removable disk packs and address plugs moved together to keep the same address for the same volume
- Don’t drop a disk pack! (The bits would fall off.)

Disk drives

- The 3330, 1970
- Much more reliable than the 2914, so 8 drawers to a bank
- Voice coil electromagnet and large static magnet used to replace hydraulic unit to drive access mechanism
- Can still move disk pack and unit address plug together
- 101 MB/volume (3330-1) or 202 MB/volume (3330-11)

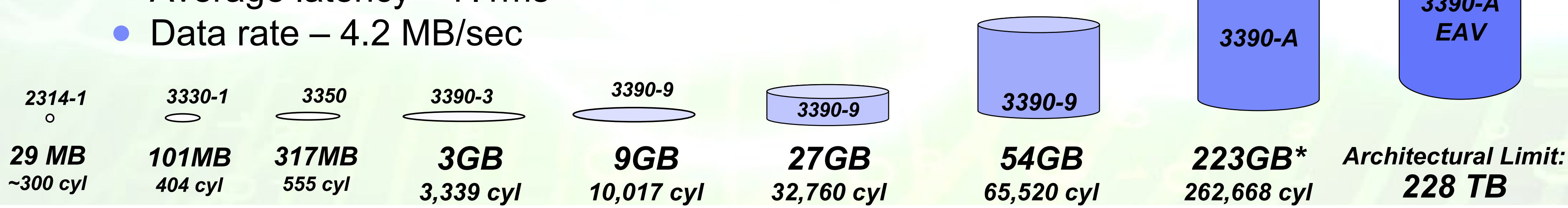
Disk drives

- A 3330 disk pack, called a 3336, alongside two Mass Storage Subsystem (3851) cartridges
- 10 2-sided data platters
- 19 data tracks per cylinder with 1 servo track

DASD Scale

- 2314:
  - Average seek time – 75ms
  - Average latency – 12.5ms
  - Data rate – 291 KB/sec
- 3330:
  - Average seek time – 30ms
  - Average latency – 8.4ms
  - Data rate – 806 KB/sec
- 3350:
  - Average seek time – 25ms
  - Average latency – 8.4ms
  - Data rate – 1.2 MB/sec
- 3380:
  - Average seek time – 17ms
  - Average latency – 8.3ms
  - Data rate – 3 MB/sec
- 3390:
  - Average seek time – 12ms
  - Average latency – 7.1ms
  - Data rate – 4.2 MB/sec

- DS8000™:
  - Seek time and rotational latency do not apply to SSD-based drives; but for 15K RPM disk:
    - Average seek time – 3.5ms
    - Average latency – 2ms
    - Data rate – 2-3.7 GB/sec



Tape drives

- The IBM 2420
- 1600 bpi!
- “Stubby” triangular vacuum columns at the top helped reduce start/stop inertia
- “Autoloading” tape covers
- Don’t forget the Write Ring! (Ever wonder where “RING” and “NORING” came from in the JES3 mount messages?)

Write Ring

Tape drives

- The IBM 3420, 1970
- Up to 6250 bpi!
- Odd models (3, 5, 7) were 1600 bpi only
- Even models (4, 6, 8) were 1600/6250 “Dual Density”
- Models 7 and 8 moved tape at 800 IPS
- High-speed rewind was *fast!* Cracked or broken, off-balance tape reels could disintegrate spectacularly, spreading plastic shrapnel throughout much of the machine.

The 3168—IBM's de facto flagship in 1977

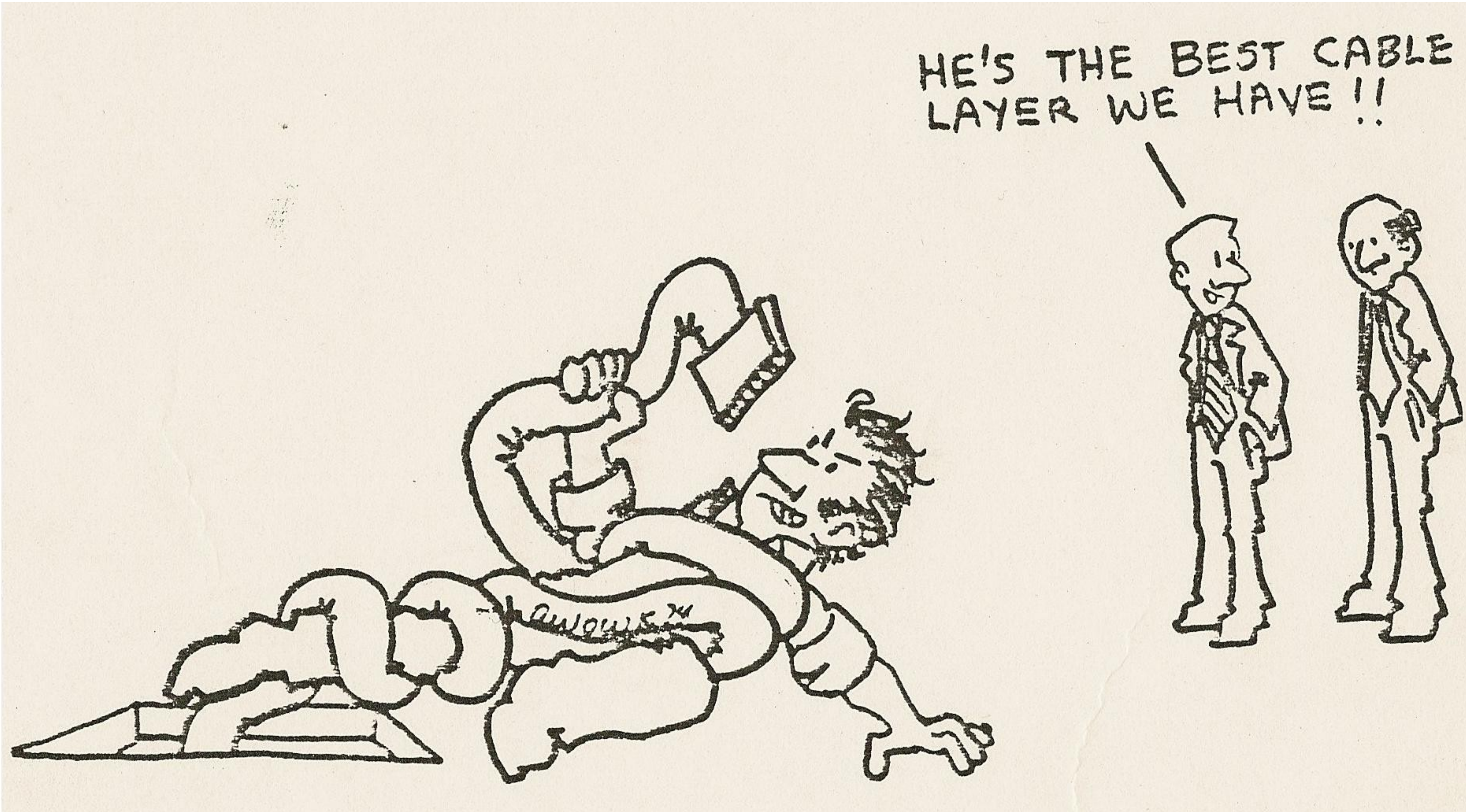
•This picture does not convey the sheer scale of this machine:

- CPU frames ~7' high
- Processor alone weighed nearly 3 ½ tons (6,881 lbs)
- It took time to walk by the CPU, console, CDU, PDU, and channel frames—MPs took more than double the space
- Channel-to-CPU cables nearly filled the space between 18" high raised floor posts across two floor tiles (4' wide)
- 6.3 KVA for the CPU alone

•Could run an MP as two "physically partitioned" UPs

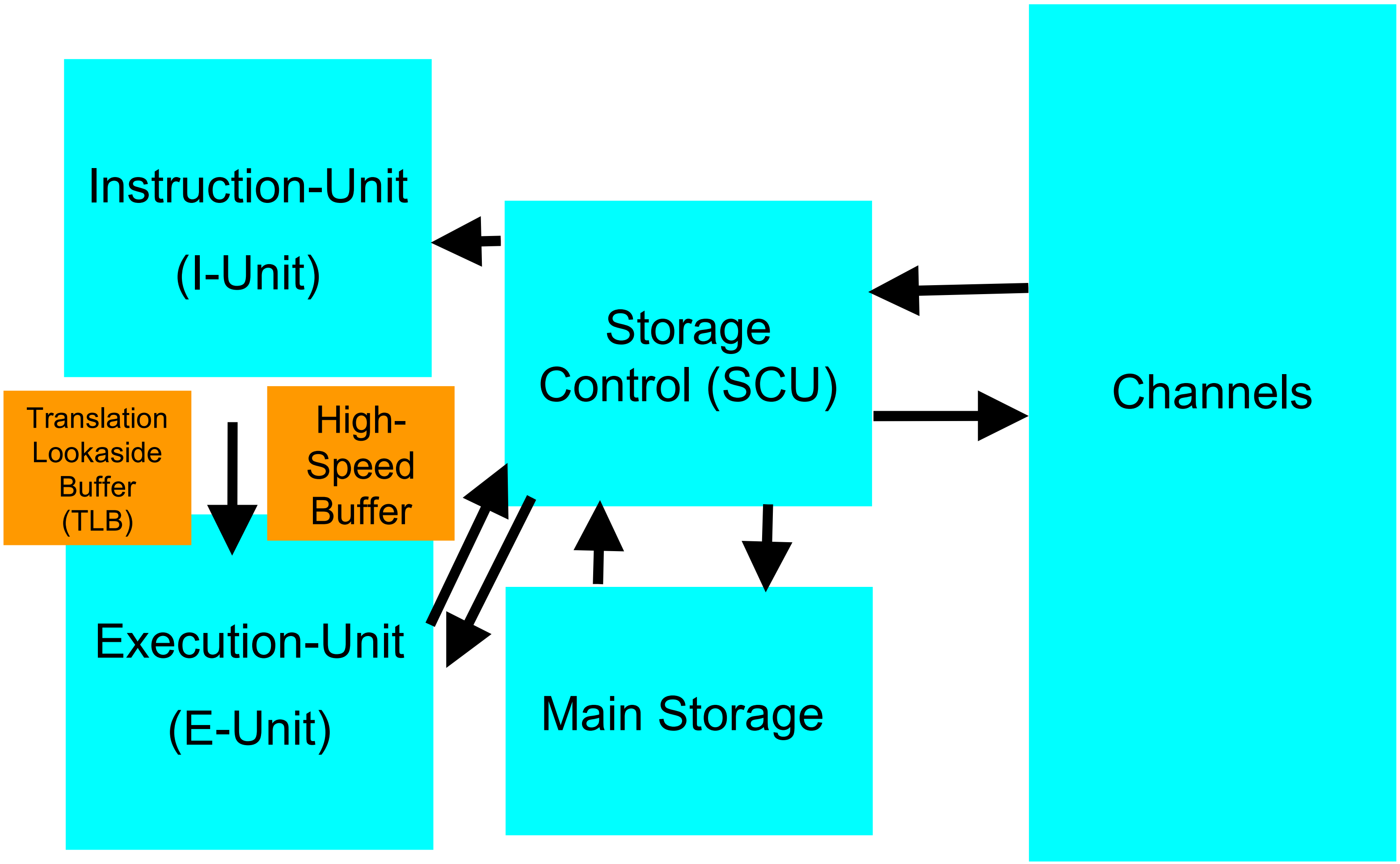
3168 Trivia

Did I mention the number of cables? Some over 100’ long, and heavy. The rule for which end to plug in where was, “Light grey away.”



3168 Block Diagram

The 168 was the basis for many later designs, and elements of its design still persist in today's servers



Hardware Diagnosis Wasn't Always Easy…

Shooting the bugs…

• Components were discrete before SLT packaging:

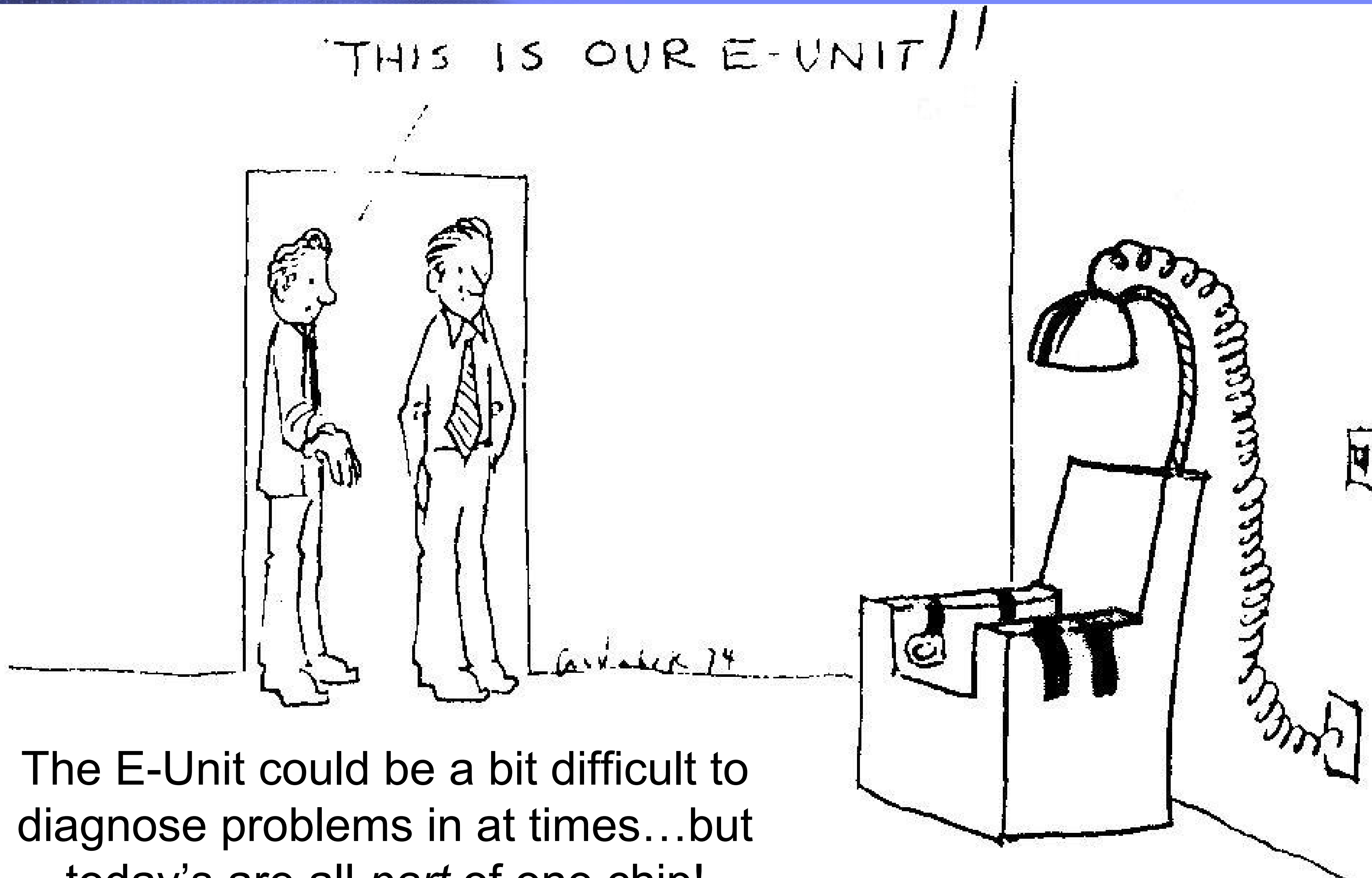
  - Transistors
  - Resistors
  - Capacitors
  - Inductors

• CEs used microcode and software diagnostics, printed logic diagrams and oscilloscopes

• Processor diagnosis often involved putting instructions into memory manually from the console using switches and dials…

• …and then following the bug circuit by circuit until the problem was found

• Intermittent problems could be solved over time by using the 'scope's "single sweep" mode to monitor events one at a time



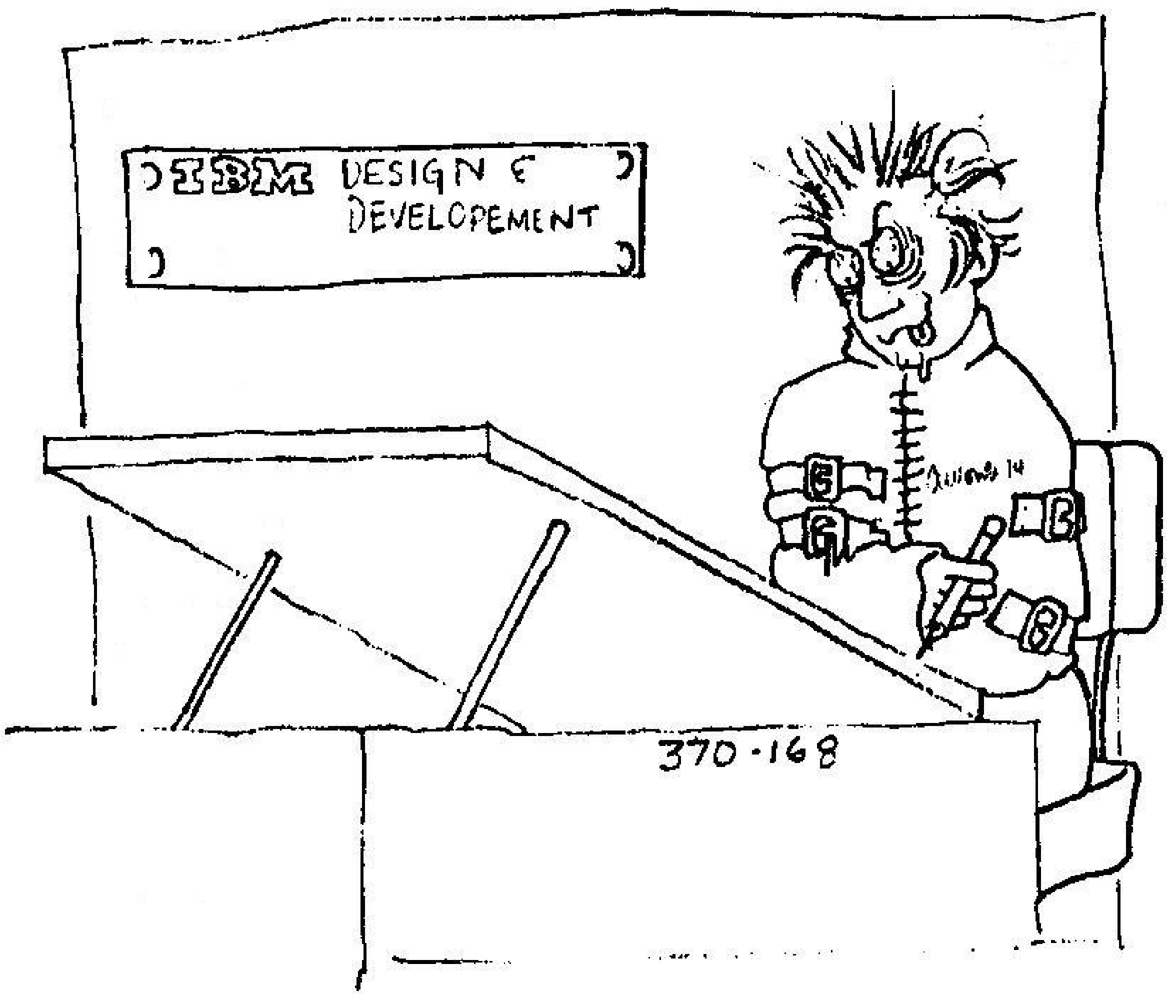
The E-Unit could be a bit difficult to diagnose problems in at times…but today's are all *part* of one chip!

3168 Trivia

- Cycle time: 80ns
  - About 1/416th the speed of a single z196 CP
- Max storage 8MB (16MB for an MP!)
  - That's less than 25% of the space needed to store the scans of the cartoons in this presentation…and 1/2000th of my phone's memory
- Worst-case storage access time: About 480ns (6 machine cycles, for a partial Store; most storage operations took 4 machine cycles, or 320ns)
- Board-to-board and frame-to-frame interconnections were done with "trileads," a semi-shielded three-part wire with push-on connectors
- Power-on-Reset (POR) set the TOD clock to 0 and loaded the microcode
- Console characters were drawn on the screen with continuous lines (not pixelated)

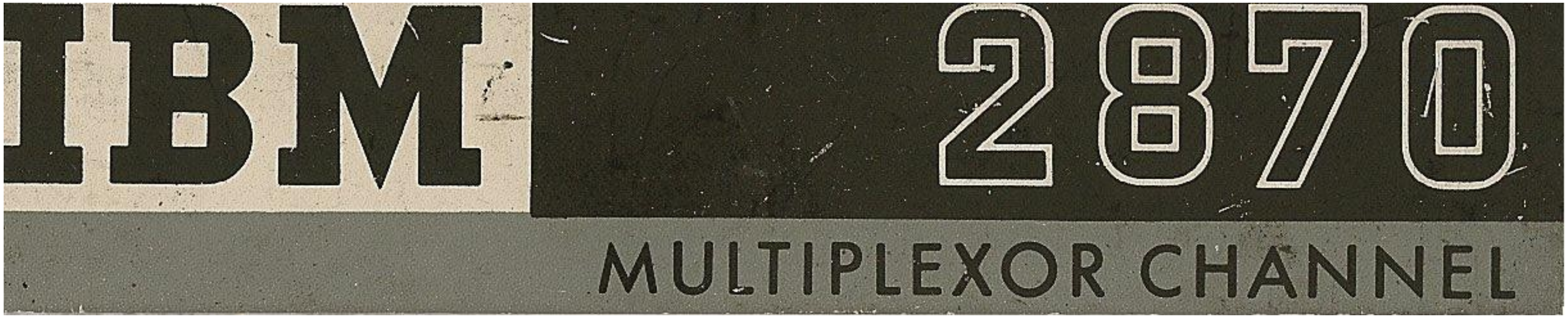
3168 Trivia

Some people apparently thought the overall design could stand improvement…



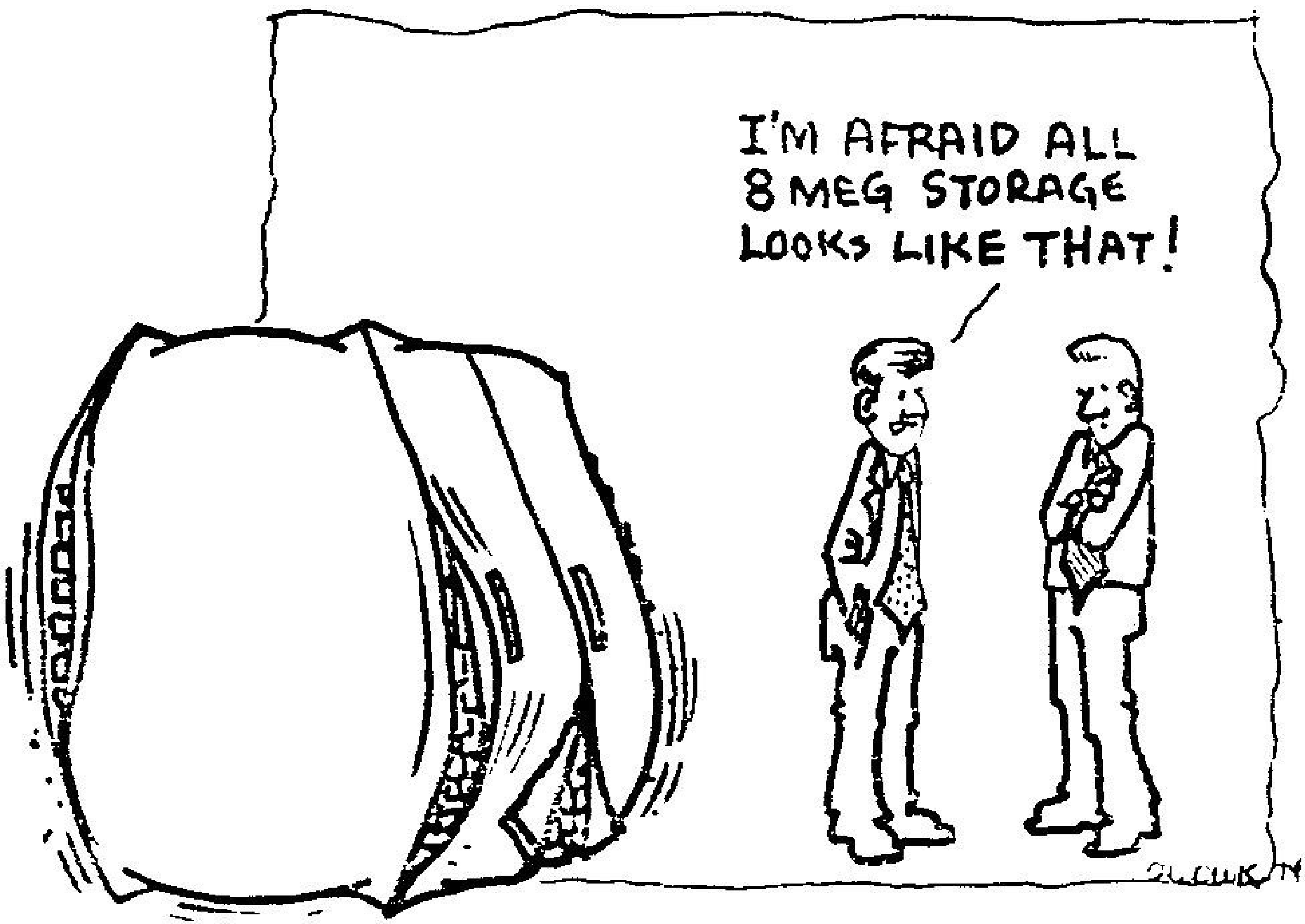
More 3168 Trivia

- Fed by a big, honkin' 240V 415Hz AC motor generator
- Cooled with (lots of!) chilled water (~30 Gal/m @ 52°F) through a large water-to-water heat exchanger in the Coolant Distribution Unit (CDU) and water-to-air exchangers inside the CPU frames
- HSM (High-Speed Multiply) internals were a trade secret, not disclosed even to CEs
- Up to 12 channels in separate frames connected with cables
  - Maximum channel data rate was 1.5 MB/sec
  - Only tape drives and 2305s could get close to that (3420's could read 6250 BPI at 800 IPS, minus overhead for IBGs, etc.)



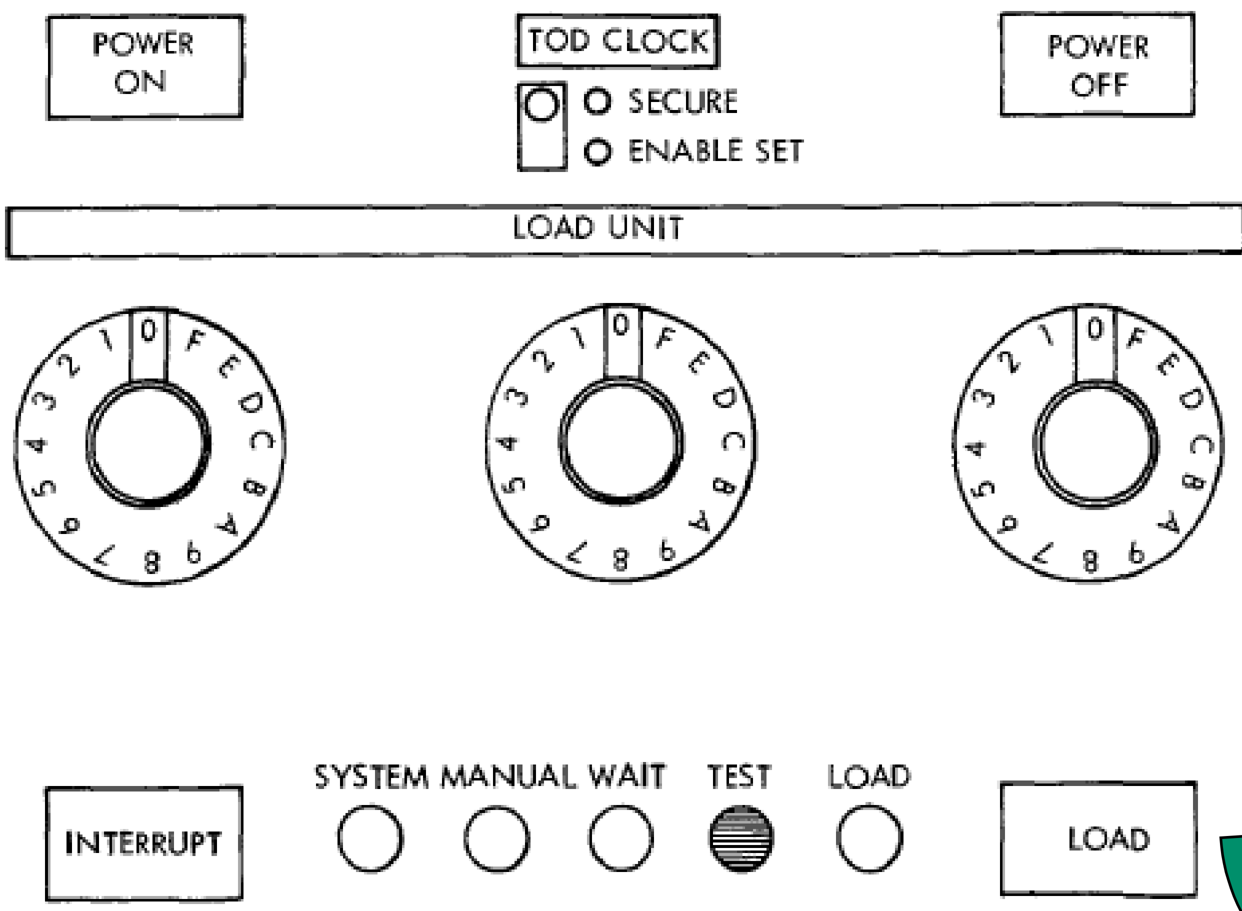
3168 Trivia

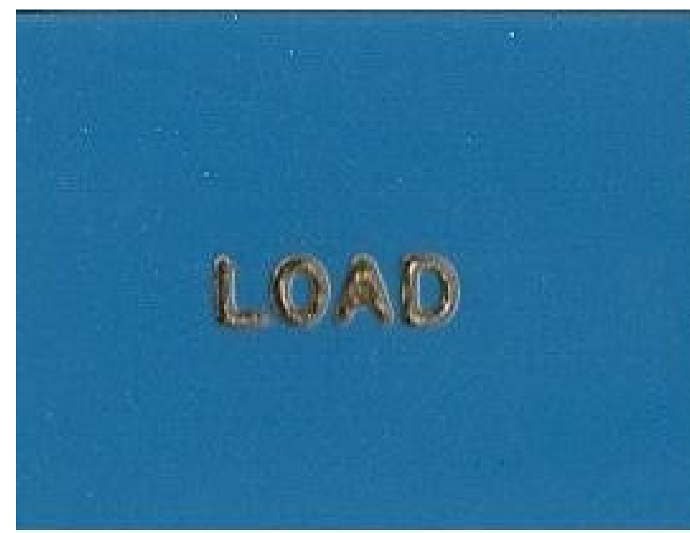
Needless to say, 8 MB was a lot of storage back then…



From a 168

Ever hear a coworker say “Hit Load” and wonder….?





More CPUs

- The 3033's were almost as big
  - But the new, 3270-based console was a bit smaller
  - No separate channel frames; "directors" were introduced, putting the channel subsystem entirely under the covers of the CPU, which reduced the total system footprint quite a bit
  - Maximum channel speed doubled to 3 MB/sec (the new cables were blue instead of grey but still had dark grey and light grey ends)
  - Maximum memory 16MB (32 for an MP!)
  - Cycle time: 57ns
  - Memory access time: 5 or 8 machine cycles (285 / 456 ns)

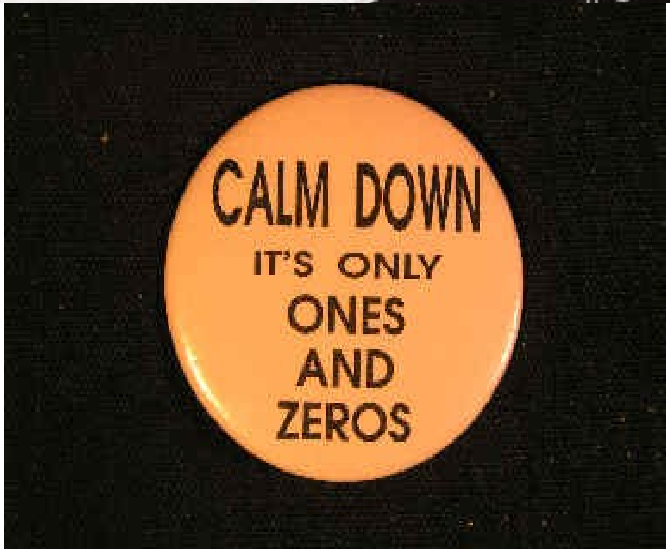
Then, there were operating systems

- But…what about the software?

- PCP
- MFT
- MVT
- SVS
- MVS™
- MVS/SP™
- MVS/XA™
- MVS/ESA™
- OS/390®
- z/OS ®

In the beginning…there were punched cards…

- …and Core Storage and PCP
  - No, nothing to do with Phencyclidine
  - PCP was the Primary Control Program
  - 32KB of main storage
  - Available in March 1966
- The life of an operator using PCP:
  - IPL from a card deck
  - Read in a job from another card deck
  - Job starts to process
  - Go hang tapes, feed printer, etc. as needed
  - Job finishes, machine goes into a wait state
  - Repeat



PCP

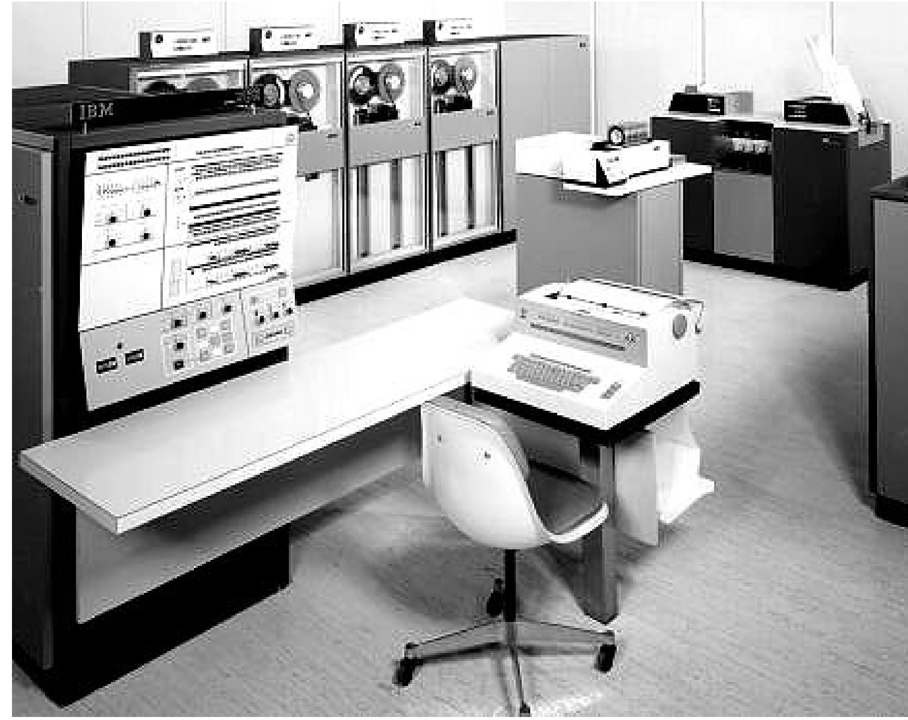
- Despite its one-job-at-a-time programming model, PCP was well-designed for its time:
  - The machines were incredibly slow by today's standards, with cycle times in the microsecond range
  - Machine utilization was actually pretty good except when the system was doing I/O because:

    - There wasn't that much data at first
      - A typical YouTube video might have required millions of punched cards…which could be read at the rate of about 300/minute
    - There wasn't enough memory to do any more, anyway!
- "Job scheduling" had to do only with whose job ran when

OS/MFT

- Still on core storage…
- But we learned how to make more of it, faster
- We could multi-task as we waited for I/O…at last!
- OS/MFT was born in 1966
  - Fence off storage areas in real memory, called "partitions"
  - Run a separate job in every partition
  - Re-IPL to change the number of partitions or their sizes
  - 64KB of storage!
  - "Job scheduling" took on a whole new meaning; not every job could run in the order it was handed to an operator as it could have been under PCP. IPLs often scheduled at specific times every day.

OS/MVT



- MFT was "Multiprogramming (with a) Fixed (number of) Tasks"
- MVT supported <u>Variable</u> tasks
- No IPL to change partition sizes any more!

- OS/MVT, 1967:
  - Still real storage based – 128KB
  - Still ran a separate job in every region
  - Job scheduling became a bit easier since one need not re-IPL to change partition sizes, but still highly limited by the machine's capacity
  - Online work begins to "interfere with" batch work at about this time

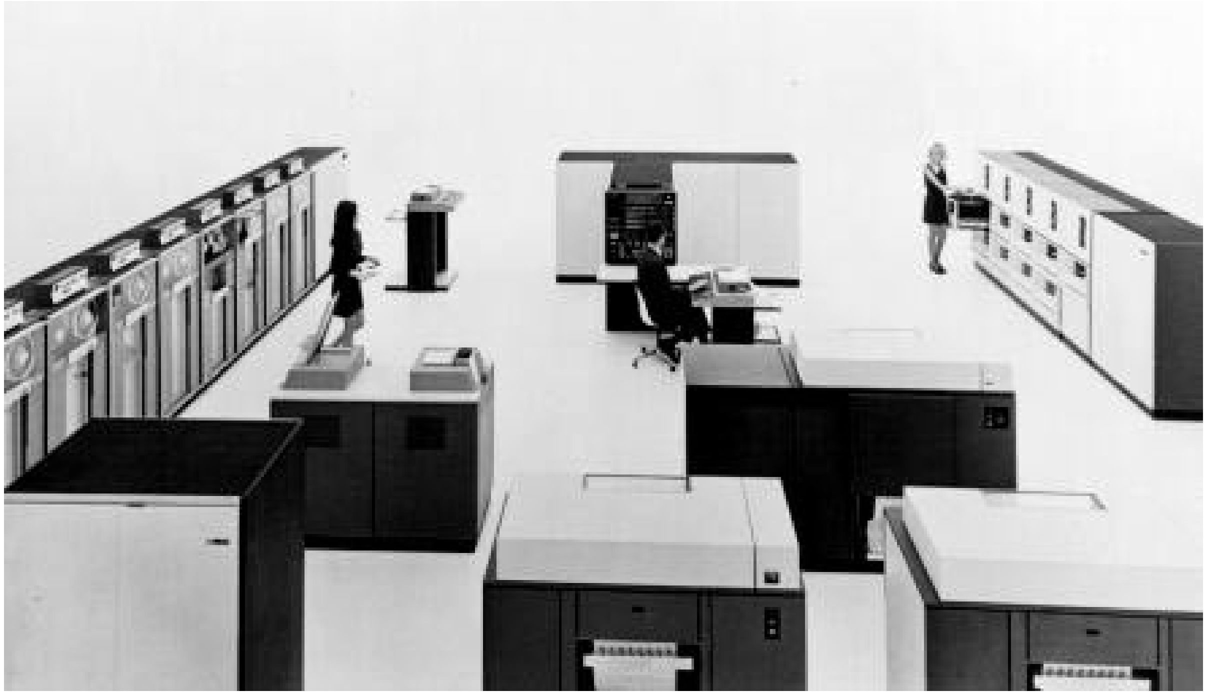
Layout of real memory under MVT

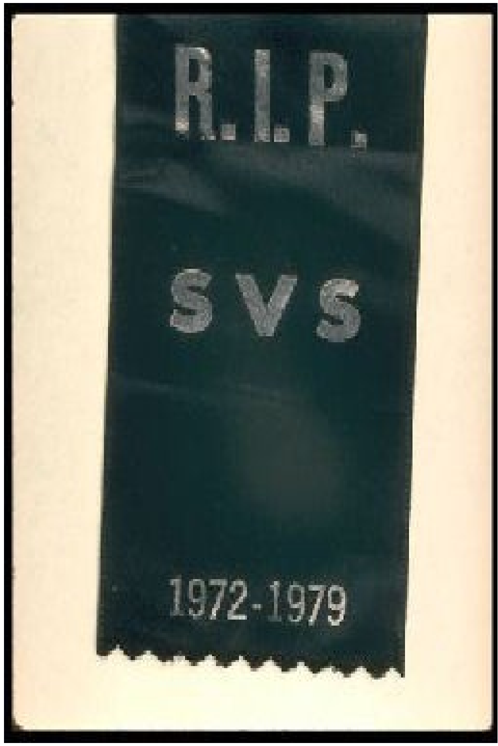
MVT could run up to 15 jobs concurrently.

The initators selected jobs from the queue, carved out the real memory to satisfy the region requirements, allocated data sets and passed control to the application programs.

The World Goes Virtual with SVS

- SVS (Single Virtual Storage) was introduced in 1972
- Virtual storage! One 16 MB address space
- Partition it however you want
- Back it with enough real storage and paging to make it all work
- The OS didn't take much storage back then, so most virtual storage was available for programs





Multiple Virtual Worlds with MVS

- MVS, 1974
- Multiple Virtual Storage = Multiple address spaces!
- Memory fragmentation (which forced frequent IPLs of SVS by today's standards) was <u>much</u> less a problem because initiators could be stopped and restarted to clean up when necessary
- Symmetric Multiprocessing introduced (3158, 3168, perhaps 3165 and 3155)
- JES2 and JES3 introduced
  - JES2 based on HASP
  - JES3 based on ASP

Setting the bar for serious business

- The Philosophy of MVS
  - ☒ Assume the work is mission critical
  - ☒ Allow no undetected errors
  - ☒ Isolate all failures to the smallest affected unit of work
  - ☒ Provide diagnostics from the first failure sufficient to debug the problem
  - ☒ Allow no program access to data it is not authorized to access

Reliability, Availability, Serviceability

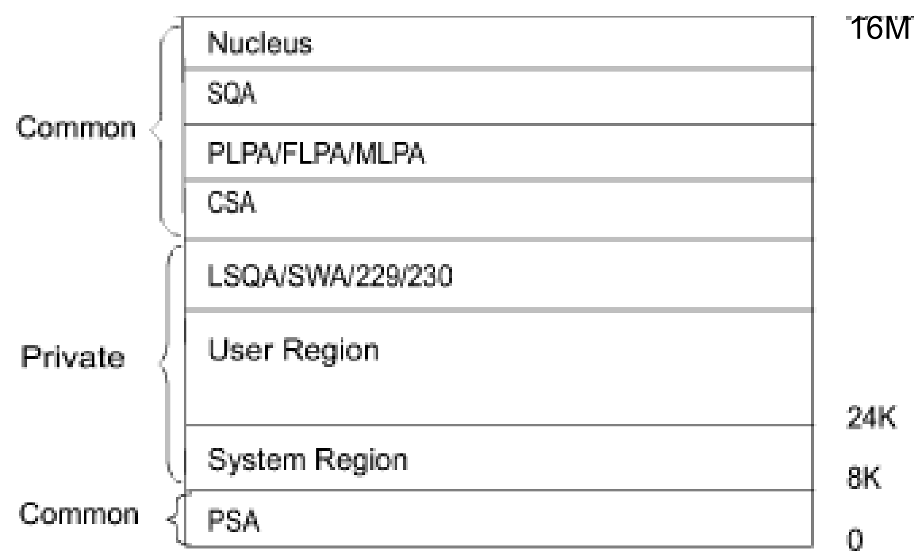
- Availability is intrinsic to the design
  - ☒ System (and subsystem) code is "covered" by a recovery routine.
  - ☒ Critical code has "nested recovery" to cover the recovery routines.
  - ☒ Diagnostic data ***specific to the error*** is gathered and reported.
  - ☒ Retry is attempted whenever possible after repairing damage and isolating the failure.

RAS is big business

- MVS RAS Guidelines
  - ☒ A lot of the code of any component (or subsystem) is devoted to RAS.
  - ☒ Hundreds of thousands of lines of RAS infrastructure code.
  - ☒ Tremendous synergy with hardware platform
  - ☒ Commitment to first failure data capture
  - ☒ Industry-unique commitment to system integrity

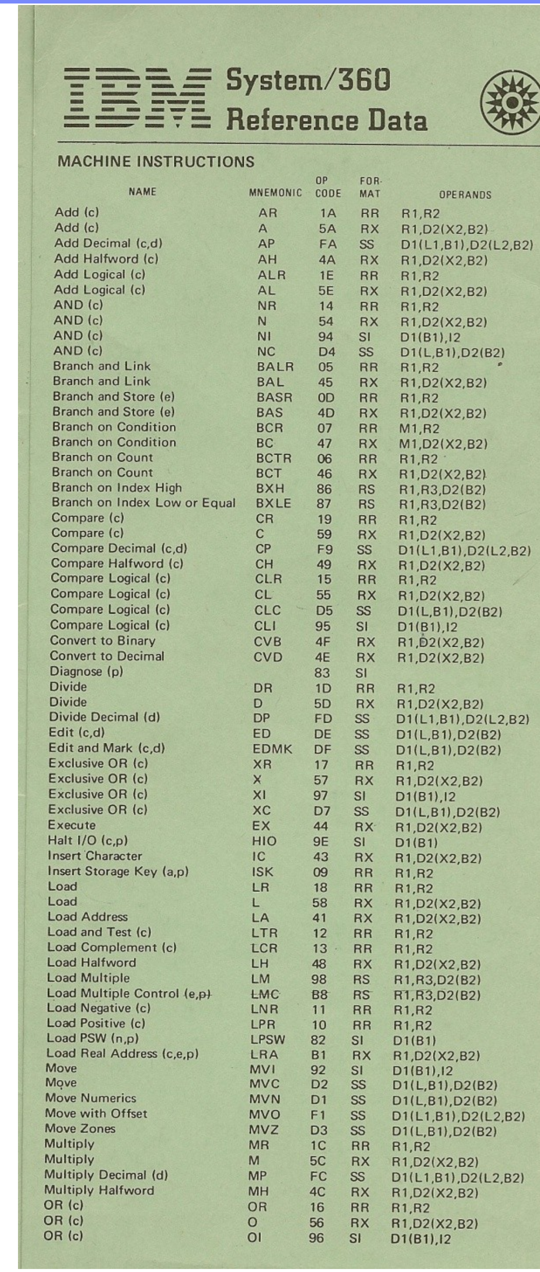
The MVS/SP Version 1 Virtual Storage Map

- Hey, back then 16 MB was a lot!



Reference Cards

- Nobody had yet imagined GUIs.
- There was a lot to remember.
- So, we had reference cards. Lots of reference cards.
- You’ve probably heard of this one—the “Green Card.”



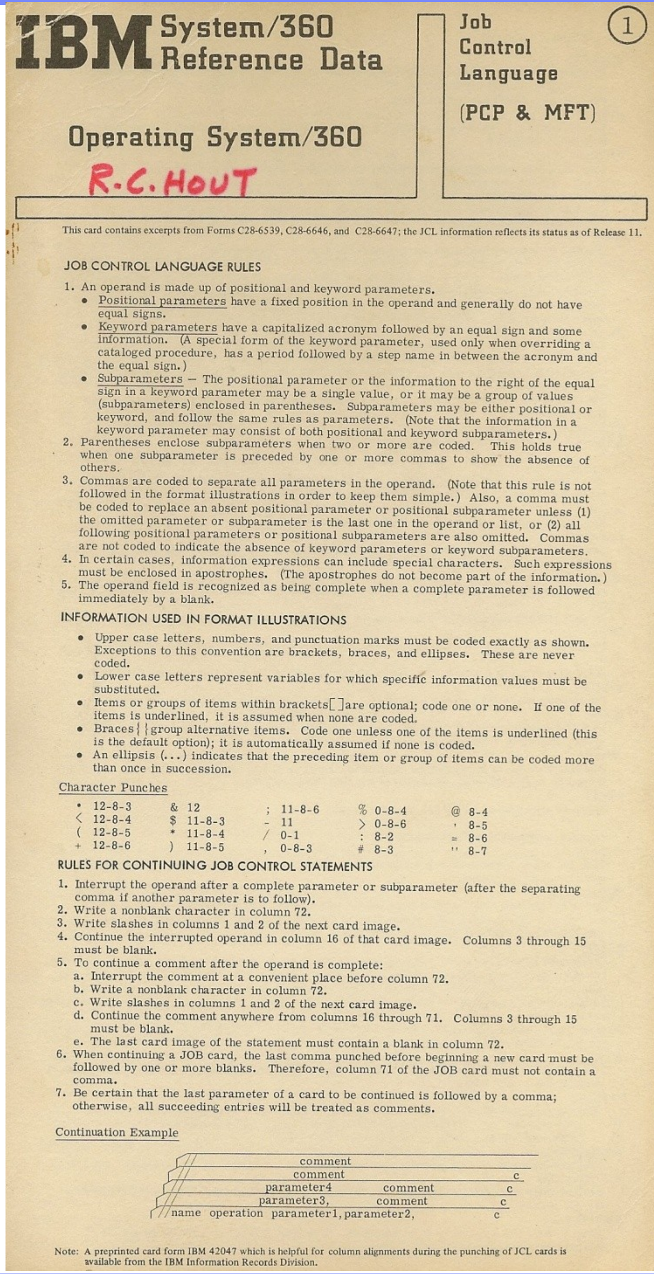
IBM System/360 Reference Data

MACHINE INSTRUCTIONS

NAME	MNEMONIC	OP CODE	FORMAT	OPERANDS
Add (c)	AR	1A	RR	R1,R2
Add (c)	A	5A	RX	R1,D2(X2,B2)
Add Decimal (c,d)	AP	FA	SS	D1(L1,B1),D2(L2,B2)
Add Halfword (c)	AH	4A	RX	R1,D2(X2,B2)
Add Logical (c)	ALR	1E	RR	R1,R2
Add Logical (c)	AL	5E	RX	R1,D2(X2,B2)
AND (c)	NR	14	RR	R1,R2
AND (c)	N	54	RX	R1,D2(X2,B2)
AND (c)	NI	94	SI	D1(B1),I2
AND (c)	NC	D4	SS	D1(L,B1),D2(B2)
Branch and Link	BALR	05	RR	R1,R2
Branch and Link	BAL	45	RX	R1,D2(X2,B2)
Branch and Store (e)	BASR	0D	RR	R1,R2
Branch and Store (e)	BAS	4D	RX	R1,D2(X2,B2)
Branch on Condition	BCR	07	RR	M1,R2
Branch on Condition	BC	47	RX	M1,D2(X2,B2)
Branch on Count	BCTR	06	RR	R1,R2
Branch on Count	BCT	46	RX	R1,D2(X2,B2)
Branch on Index High	BXH	86	RS	R1,R3,D2(B2)
Branch on Index Low or Equal	BXLE	87	RS	R1,R3,D2(B2)
Compare (c)	CR	19	RR	R1,R2
Compare (c)	C	59	RX	R1,D2(X2,B2)
Compare Decimal (c,d)	CP	F9	SS	D1(L1,B1),D2(L2,B2)
Compare Halfword (c)	CH	49	RX	R1,D2(X2,B2)
Compare Logical (c)	CLR	15	RR	R1,R2
Compare Logical (c)	CL	55	RX	R1,D2(X2,B2)
Compare Logical (c)	CLC	D5	SS	D1(L,B1),D2(B2)
Compare Logical (c)	CLI	95	SI	D1(B1),I2
Convert to Binary	CVB	4F	RX	R1,D2(X2,B2)
Convert to Decimal	CVD	4E	RX	R1,D2(X2,B2)
Diagnose (p)		83	SI	
Divide	DR	1D	RR	R1,R2
Divide	D	5D	RX	R1,D2(X2,B2)
Divide Decimal (d)	DP	FD	SS	D1(L1,B1),D2(L2,B2)
Edit (c,d)	ED	DE	SS	D1(L,B1),D2(B2)
Edit and Mark (c,d)	EDMK	DF	SS	D1(L,B1),D2(B2)
Exclusive OR (c)	XR	17	RR	R1,R2
Exclusive OR (c)	X	57	RX	R1,D2(X2,B2)
Exclusive OR (c)	XI	97	SI	D1(B1),I2
Exclusive OR (c)	XC	D7	SS	D1(L,B1),D2(B2)
Execute	EX	44	RX	R1,D2(X2,B2)
Halt I/O (c,p)	HIO	9E	SI	D1(B1)
Insert Character	IC	43	RX	R1,D2(X2,B2)
Insert Storage Key (a,p)	ISK	09	RR	R1,R2
Load	LR	18	RR	R1,R2
Load	L	58	RX	R1,D2(X2,B2)
Load Address	LA	41	RX	R1,D2(X2,B2)
Load and Test (c)	LTR	12	RR	R1,R2
Load Complement (c)	LCR	13	RR	R1,R2
Load Halfword	LH	48	RX	R1,D2(X2,B2)
Load Multiple	LM	98	RS	R1,R3,D2(B2)
Load Multiple Control ~~(e,p)~~	~~LMC~~	~~B8~~	~~RS~~	~~R1,R3~~,D2(B2)
Load Negative (c)	LNR	11	RR	R1,R2
Load Positive (c)	LPR	10	RR	R1,R2
Load PSW (n,p)	LPSW	82	SI	D1(B1)
Load Real Address (c,e,p)	LRA	B1	RX	R1,D2(X2,B2)
Move	MVI	92	SI	D1(B1),I2
Move	MVC	D2	SS	D1(L,B1),D2(B2)
Move Numerics	MVN	D1	SS	D1(L,B1),D2(B2)
Move with Offset	MVO	F1	SS	D1(L1,B1),D2(L2,B2)
Move Zones	MVZ	D3	SS	D1(L,B1),D2(B2)
Multiply	MR	1C	RR	R1,R2
Multiply	M	5C	RX	R1,D2(X2,B2)
Multiply Decimal (d)	MP	FC	SS	D1(L1,B1),D2(L2,B2)
Multiply Halfword	MH	4C	RX	R1,D2(X2,B2)
OR (c)	OR	16	RR	R1,R2
OR (c)	O	56	RX	R1,D2(X2,B2)
OR (c)	OI	96	SI	D1(B1),I2

Reference Cards

- But there were cards before the green one
- Like this one, a JCL reference card for PCP and MFT
- It was originally white but has yellowed with age
- (Bob Hout gave me this card when he retired several years ago)



IBM System/360 Reference Data

Job Control Language (PCP & MFT) ①

Operating System/360

R.C. HOUT

This card contains excerpts from Forms C28-6539, C28-6646, and C28-6647; the JCL information reflects its status as of Release 11.

JOB CONTROL LANGUAGE RULES

1. An operand is made up of positional and keyword parameters.
   - Positional parameters have a fixed position in the operand and generally do not have equal signs.
   - Keyword parameters have a capitalized acronym followed by an equal sign and some information. (A special form of the keyword parameter, used only when overriding a cataloged procedure, has a period followed by a step name in between the acronym and the equal sign.)
   - Subparameters – The positional parameter or the information to the right of the equal sign in a keyword parameter may be a single value, or it may be a group of values (subparameters) enclosed in parentheses. Subparameters may be either positional or keyword, and follow the same rules as parameters. (Note that the information in a keyword parameter may consist of both positional and keyword subparameters.)
2. Parentheses enclose subparameters when two or more are coded. This holds true when one subparameter is preceded by one or more commas to show the absence of others.
3. Commas are coded to separate all parameters in the operand. (Note that this rule is not followed in the format illustrations in order to keep them simple.) Also, a comma must be coded to replace an absent positional parameter or positional subparameter unless (1) the omitted parameter or subparameter is the last one in the operand or list, or (2) all following positional parameters or positional subparameters are also omitted. Commas are not coded to indicate the absence of keyword parameters or keyword subparameters.
4. In certain cases, information expressions can include special characters. Such expressions must be enclosed in apostrophes. (The apostrophes do not become part of the information.)
5. The operand field is recognized as being complete when a complete parameter is followed immediately by a blank.

INFORMATION USED IN FORMAT ILLUSTRATIONS

- Upper case letters, numbers, and punctuation marks must be coded exactly as shown. Exceptions to this convention are brackets, braces, and ellipses. These are never coded.
- Lower case letters represent variables for which specific information values must be substituted.
- Items or groups of items within brackets [ ] are optional; code one or none. If one of the items is underlined, it is assumed when none are coded.
- Braces { } group alternative items. Code one unless one of the items is underlined (this is the default option); it is automatically assumed if none is coded.
- An ellipsis (...) indicates that the preceding item or group of items can be coded more than once in succession.

Character Punches

.	12-8-3	&	12	;	11-8-6	%	0-8-4	@	8-4
<	12-8-4	$	11-8-3	-	11	>	0-8-6	'	8-5
(	12-8-5	*	11-8-4	/	0-1	:	8-2	=	8-6
+	12-8-6	)	11-8-5	,	0-8-3	#	8-3	"	8-7

RULES FOR CONTINUING JOB CONTROL STATEMENTS

1. Interrupt the operand after a complete parameter or subparameter (after the separating comma if another parameter is to follow).
2. Write a nonblank character in column 72.
3. Write slashes in columns 1 and 2 of the next card image.
4. Continue the interrupted operand in column 16 of that card image. Columns 3 through 15 must be blank.
5. To continue a comment after the operand is complete:
   a. Interrupt the comment at a convenient place before column 72.
   b. Write a nonblank character in column 72.
   c. Write slashes in columns 1 and 2 of the next card image.
   d. Continue the comment anywhere from columns 16 through 71. Columns 3 through 15 must be blank.
   e. The last card image of the statement must contain a blank in column 72.
6. When continuing a JOB card, the last comma punched before beginning a new card must be followed by one or more blanks. Therefore, column 71 of the JOB card must not contain a comma.
7. Be certain that the last parameter of a card to be continued is followed by a comma; otherwise, all succeeding entries will be treated as comments.

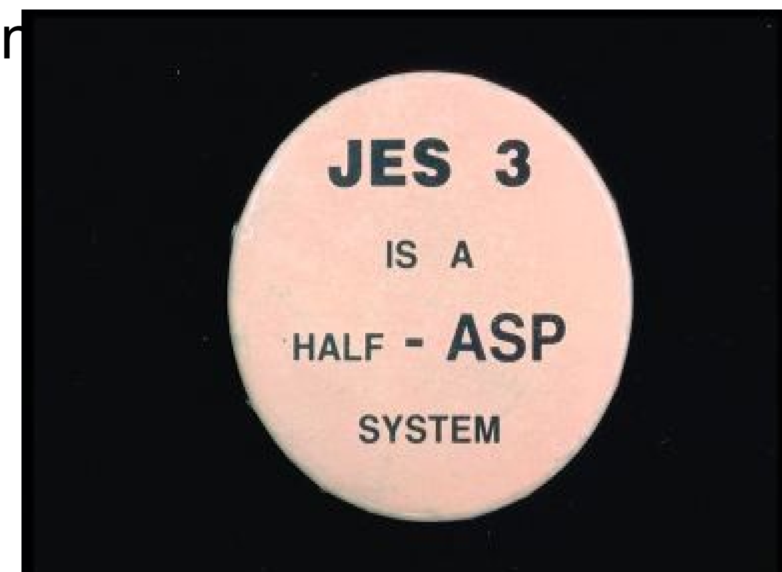
Continuation Example

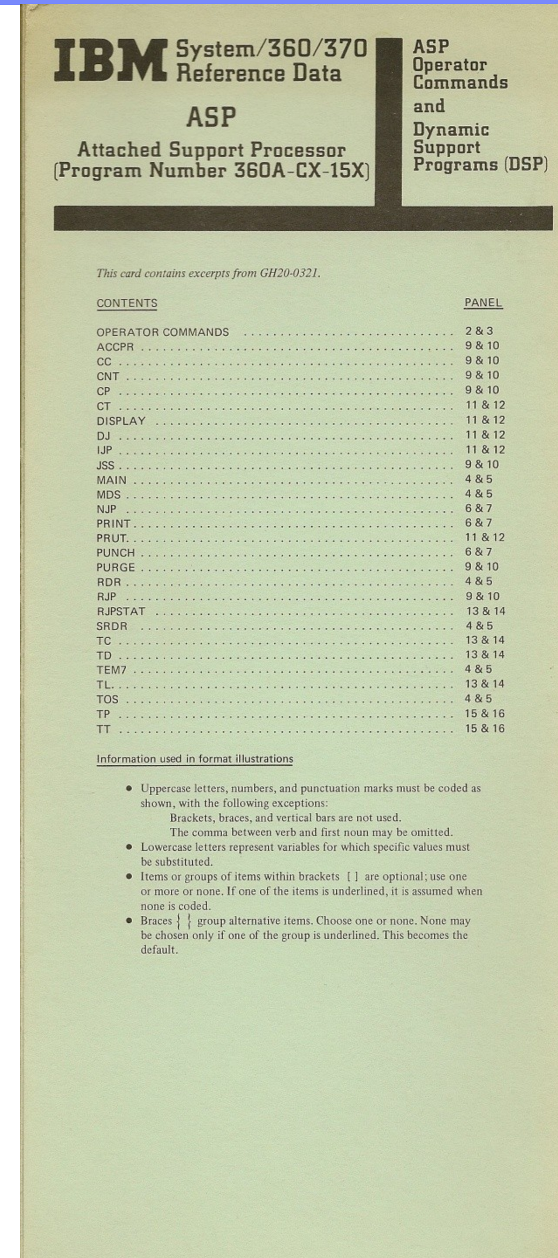
```
//            comment
//            comment                      c
//        parameter4        comment        c
//        parameter3,       comment        c
//name  operation  parameter1,parameter2,  c
```

Note: A preprinted card form IBM 42047 which is helpful for column alignments during the punching of JCL cards is available from the IBM Information Records Division.

Reference Cards

- Here’s an Attached Support Processor (ASP) reference card
- ASP was JES3’s forerunner, and it included Dynamic Support Programs (DSPs)
- As with JES3, commands started with an asterisk (*)
- If you’ve ever heard anyone say (or seen the SHARE button that says) “JES3 is a ‘half-ASP’ system,” n ant





IBM System/360/370 Reference Data

ASP Operator Commands and Dynamic Support Programs (DSP)

ASP

Attached Support Processor
(Program Number 360A-CX-15X)

This card contains excerpts from GH20-0321.

Information used in format illustrations

- Uppercase letters, numbers, and punctuation marks must be coded as shown, with the following exceptions:
  Brackets, braces, and vertical bars are not used.
  The comma between verb and first noun may be omitted.
- Lowercase letters represent variables for which specific values must be substituted.
- Items or groups of items within brackets [ ] are optional; use one or more or none. If one of the items is underlined, it is assumed when none is coded.
- Braces { } group alternative items. Choose one or none. None may be chosen only if one of the group is underlined. This becomes the default.

Reference Cards

- Here are a few more:

IBM OS/360 TSO Command Language Reference Summary

GX28-6781-1

Second Edition (August, 1972)

This reference summary will be updated from time to time; however, the basic documentation is the authoritative source and will be first to reflect changes. Effective system level is OS Release 21.6 Information herein is extracted from GC28-6732-2.

Requests for copies of this and other IBM publications should be made to your IBM representative or to the IBM branch office serving your locality. Please direct any comments on the contents of this publication to the address on the front cover. All comments and suggestions become the property of IBM.

Key to Symbols in Command Definitions

1. UPPERCASE, digits and special characters - must appear as shown.
2. Lowercase - information supplied by the user.
3. Item... - you may list the item more than once.
4. { } - you must specify one item.
5. [ ] - optional item; you may specify one.
6. KEYWORD - default item if you do not specify one.
7. Stacked items - alternatives; specify only one item from the stack.
8. **BOLDFACE** or **boldface** - information which must be given for a command.
9. Data-set-list - can be either a data-set-name or a list of data-set-names.

IBM Corporation, Publications Development, Dept. D58, Bldg. 706-2, PO Box 390, Poughkeepsie, New York 12602

IBM System/360 Reference Data — DASD Capacity and Transmission Time

2314 Direct Access Storage Facility

Models:	1	A1
Average Access Time	75 ms	60 ms
Average Rotational Delay	12.5 ms	12.5 ms

The formulas used to determine capacity and transmission time assume the use of programming systems developed and supported by IBM and are in agreement with Systems Reference Library A26-3599-2, N26-0203 and N26-0230.

These systems use eight bytes of the first record on each track. The formulas are:

a. Bytes per record, except last record on track:
$$[2137\ (KL+DL)/2048]^{*} + C + 101$$

b. Bytes per record, last record on track only:
$$KL+DL+C$$

c. Capacity per track in bytes: 7294

d. Records per track:
$$\left[\frac{c-b}{a}\right]^{*} + 1$$

e. Data rate (ms per byte): 0.0032051

f. Transmission time (ms per record):
(bytes per record) x (data rate)

KL = Key Length
DL = Data Length
C = 0 when KL = 0
C = 45 when KL ≠ 0

*Truncate any fraction

(Reprint 2/70) Printed in U.S.A. GX20-1710-2

IBM System/370 Model 165 Operator's Reference Card

For a detailed description of these operations, see *IBM System/370 Model 165 Operating Procedures*, GA22-6969.

TURNON
- Check doors, feeds, cards and/or paper.
- Check tapes, disks, and two-channel switch (if applicable).
- Check coolant and MG power (if applicable).

1. Press POWER ON (turns red).
2. Wait; POWER ON (turns white).
3. If *manual* light is not turned on, check for red CNSL FILE light. If on:
   a. Set RSDT/NON RSDT to RSDT.
   b. Set FILE SECTION SELECT to 0.
   c. Press LOAD MD.
4. If *manual* light is on, check I/O:
   a. 2250—Press POWER ON (backlight).
   b. Disks—Set to ENABLE and START.
   c. 2701—Set to ENABLE.
5. Perform IPL.

TURNOFF
1. Issue WRITELOG and HALT (if applicable).
2. Press STOP to turn *manual* light on:
3. Perform two-channel switch procedure (if applicable).
4. Check tapes; press RESET and LOAD REWIND. After rewind, press UNLOAD and RESET.
5. Check disks; switch to STOP.
6. Press POWER OFF (backlight off).
7. Check coolant and MG power (if applicable).

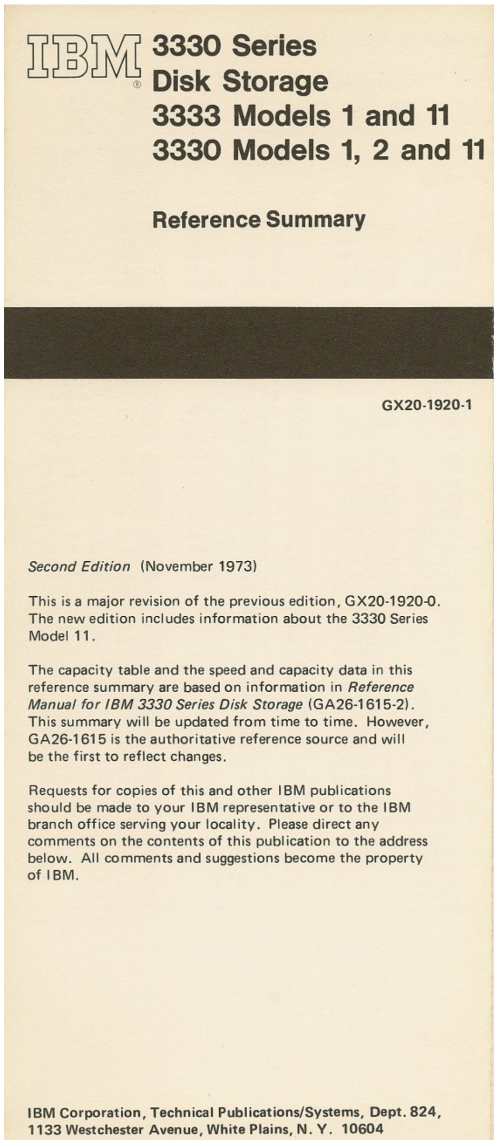
CLEAR STORAGE
1. Hold SYSTEM CLEAR; press SYSTEM RESET.
2. Release SYSTEM CLEAR; *manual* light turns on.
3. Perform IPL.

INITIAL PROGRAM LOADING (IPL)
1. Set LOAD UNIT switch to residence-volume address.
2. Hold SYSTEM CLEAR; press LOAD.

(For IPL completion, see *IBM System/360 Operating System: Operator's Procedures*, GC28-6692.)

GX22-6984-2

- …and some more:


IBM 3330 Series Disk Storage
3333 Models 1 and 11
3330 Models 1, 2 and 11

Reference Summary

GX20-1920-1

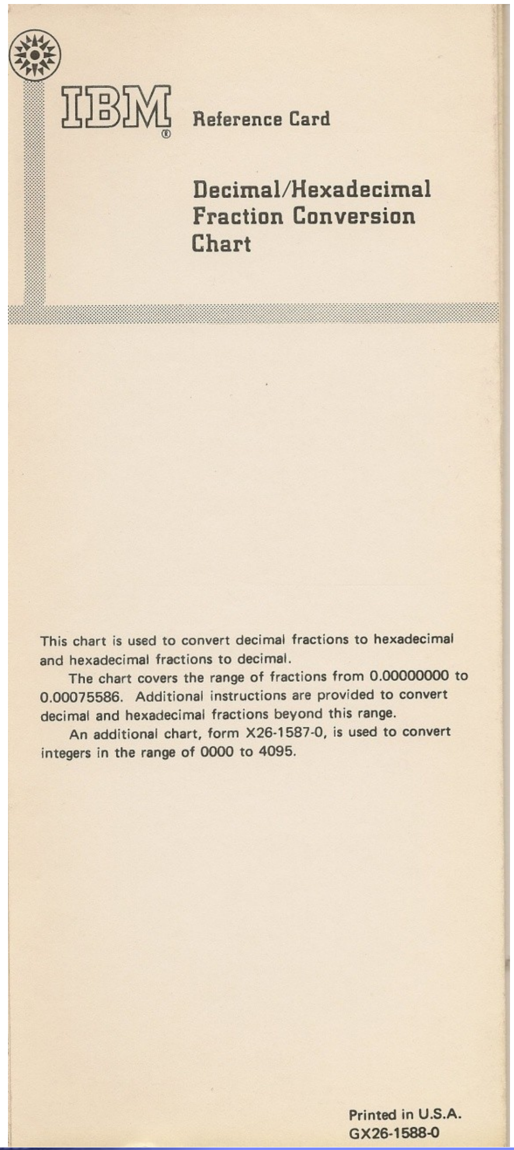
Second Edition (November 1973)

This is a major revision of the previous edition, GX20-1920-0. The new edition includes information about the 3330 Series Model 11.

The capacity table and the speed and capacity data in this reference summary are based on information in *Reference Manual for IBM 3330 Series Disk Storage* (GA26-1615-2). This summary will be updated from time to time. However, GA26-1615 is the authoritative reference source and will be the first to reflect changes.

Requests for copies of this and other IBM publications should be made to your IBM representative or to the IBM branch office serving your locality. Please direct any comments on the contents of this publication to the address below. All comments and suggestions become the property of IBM.

IBM Corporation, Technical Publications/Systems, Dept. 824, 1133 Westchester Avenue, White Plains, N. Y. 10604


IBM Reference Card

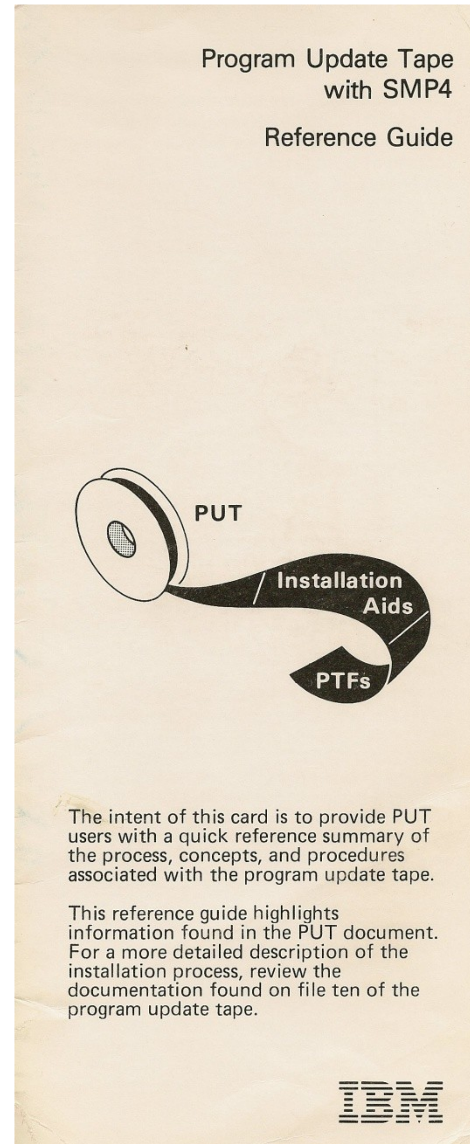
Decimal/Hexadecimal Fraction Conversion Chart

This chart is used to convert decimal fractions to hexadecimal and hexadecimal fractions to decimal.

The chart covers the range of fractions from 0.00000000 to 0.00075586. Additional instructions are provided to convert decimal and hexadecimal fractions beyond this range.

An additional chart, form X26-1587-0, is used to convert integers in the range of 0000 to 4095.

Printed in U.S.A.
GX26-1588-0


Program Update Tape with SMP4

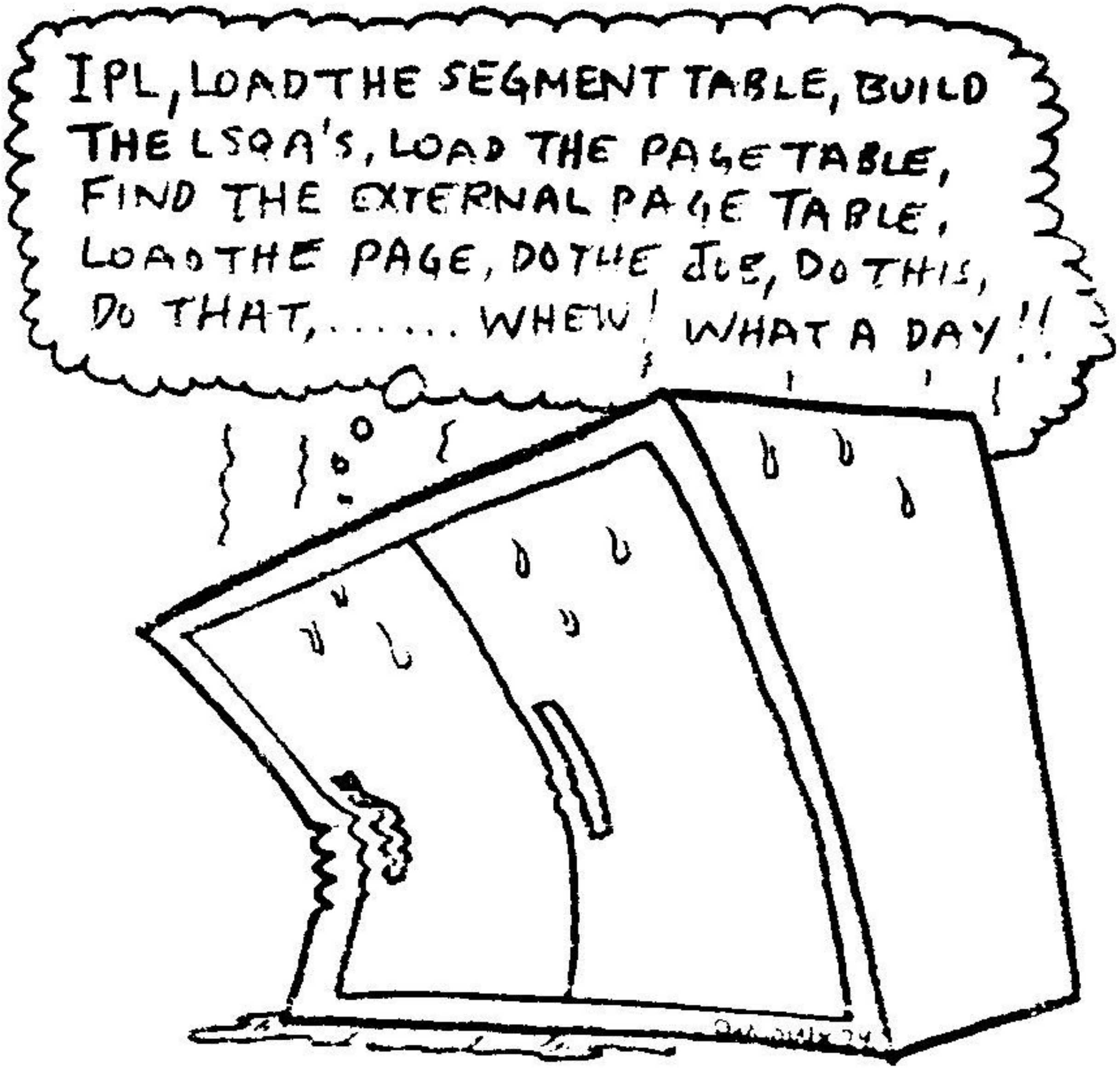
Reference Guide

The intent of this card is to provide PUT users with a quick reference summary of the process, concepts, and procedures associated with the program update tape.

This reference guide highlights information found in the PUT document. For a more detailed description of the installation process, review the documentation found on file ten of the program update tape.

IBM

IPL processing got a bit more complicated…



MVS

As did diagnosis…

Diagnosis

• Notwithstanding the prior chart…

• AMDPRDMP was the forerunner to IPCS

• A standalone dump took about half a box of paper on a large, busy system…at least, until XA came along:

  • With MVS/XA, a typical standalone dump was like Midwestern corn by the 4th of July—that is, "knee high."

  • With MVS/ESA, with dataspace storage, they could be…well…bigger!

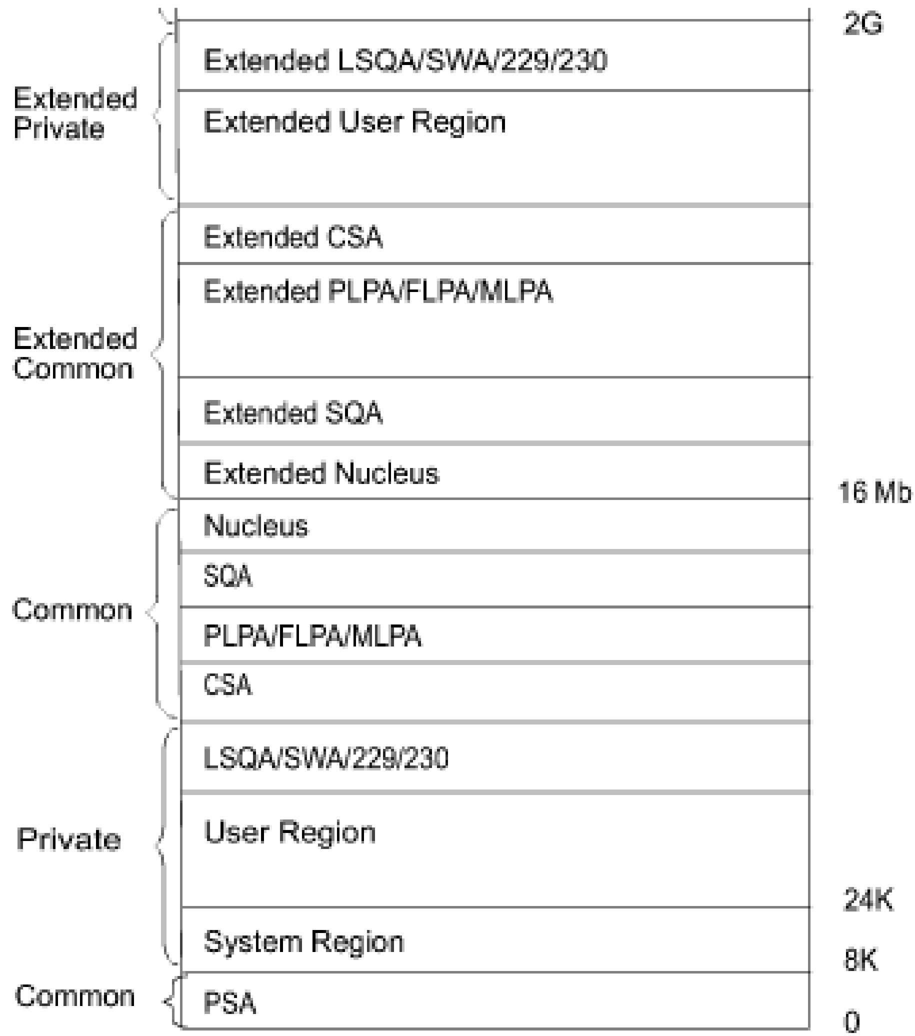
  • With 64-bit z/OS, it's a Good Thing we have IPCS.

MVS/XA

- MVS/XA Version 2 was introduced in 1981
- 31-bit addressing hits the streets
- 2 GB of virtual storage per address space looked like infinite space back then (you can trust me on this)
- Required new hardware, 3083, 3081 or 3084 (3084 pictured)
- Introduction of Dynamic Path Selection on DASD controllers
- Hardware and software both incredibly reliable by early 1980's standards
    - Months between IPLs vs. days or weeks
    - We found out we had been <u>relying</u> on frequent IPLs for some business processes
    - And, we found new problems related to the longer life of an IPL—like initiator fragmentation, now (mostly) a thing of the past


The MVS/XA Version 2 Virtual Storage Map

- After 16 MB…
- …2 GB looked like it would last us a long time…
- …which, of course, it did not.



MVS/ESA

• MVS/ESA Version 3, 1988

• Introduced access registers, linkage stacks, data spaces, and Hiperspace™

• PR/SM introduced on 3090 at about the same time, creating LPAR mode (later the only mode) in addition to Basic Mode

• System-Managed Storage (SMS) introduced, with the Interactive Storage Management Facility (ISMF)

• LLA and VLF introduced, along with what is now z/OS UNIX System Services

MVS/ESA, continued

- MVS/ESA SP Version 4, 1991
- Extended Multi-Image Facility (EMIF) introduced for PR/SM™
- MVS/ESA SP 4.1 introduced:
  - Sysplex
  - HCD
- Available on:
  - 1600 or 6250 bpi open-reel tape
  - 3480 (uncompressed only)

MVS/ESA, continued

- MVS/ESA SP Version 5, 1994
  - Parallel Sysplex® introduced
    - IMS™ data sharing is first exploiter
  - Workload Manager (WLM) is introduced
    - Defined policies for the system's workloads
    - Goal-based performance management
    - CICSPlex®/SM and VTAM® are first cross-system exploiters

OS/390 – Putting it All Together

- OS/390, 1996
  - 72 products in one*!
    - Former products became “elements” of OS/390, some at no additional charge, some not
  - All tested together, at the same time
    - Intended to improve quality by letting us focus our test efforts
  - Ordering became simpler—checklist was much shorter because there were fewer options
  - New installation vehicle, ServerPac, replaced CBIPO

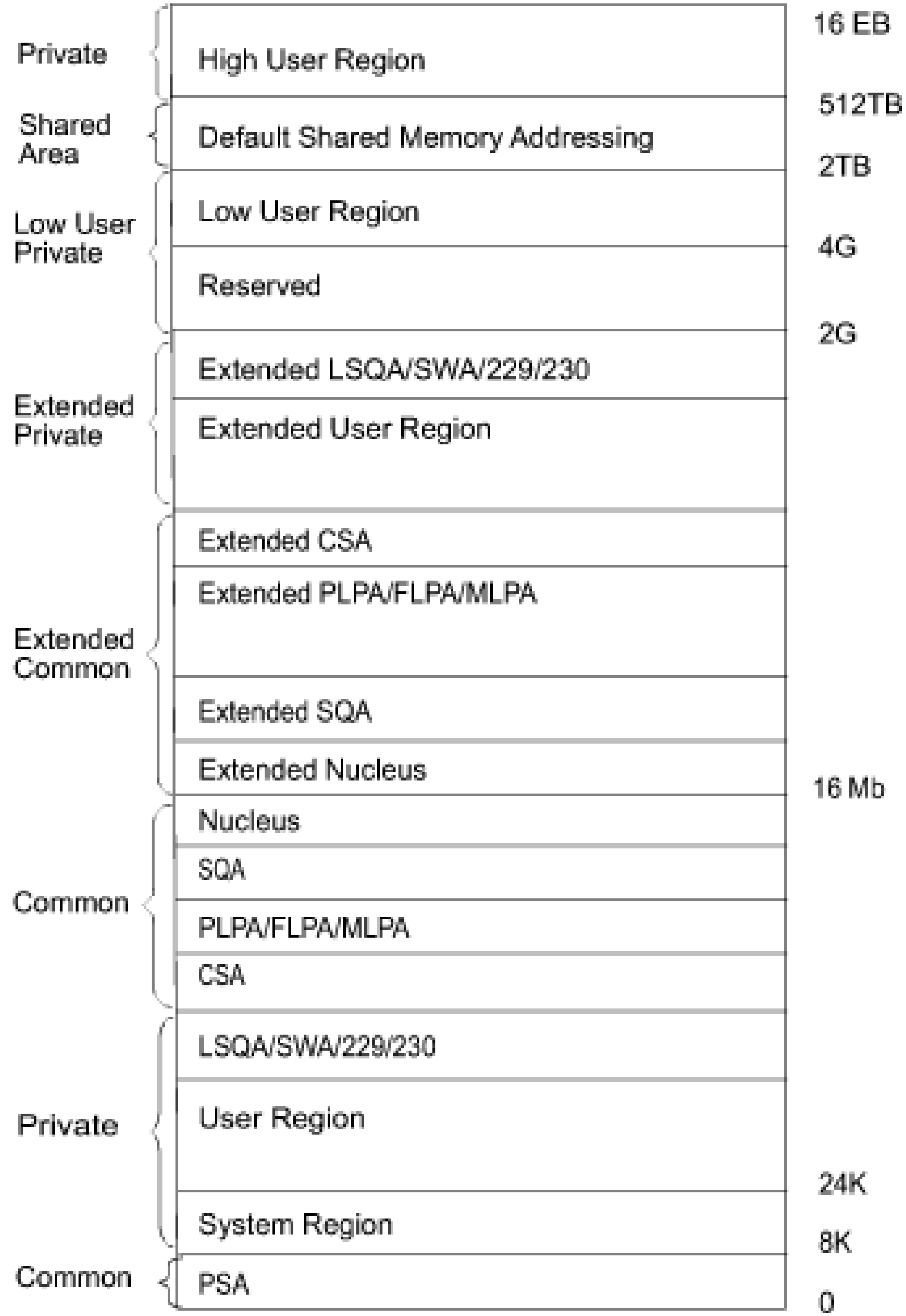
*OK, so not all of them were separate products.

z/OS – The Next Generation

- z/OS, 2000
  - The beginning of 64-bit addressing
    - First exploiters were access methods, HFS, XRC, and DB2
  - Support for IRD
    - WLM moves work to resources
    - IRD moves resources to work
      - LPAR CPU management
      - I/O priority queueing
      - Dynamic Channel Path Management
  - Workload license charges introduced
  - SNA Master Console Support
  - ServerPac provides a Recommended System Layout function to automatically place data sets on volumes

The z/OS R9 Virtual Storage Map

- We are hopeful that 16 EB will last us *at least* a few more months…
- …and if it's not, I'm not sure what we'll do next…
- One wag noted that 128-bit addressing would require more silicon atoms than we think there might be in the known universe

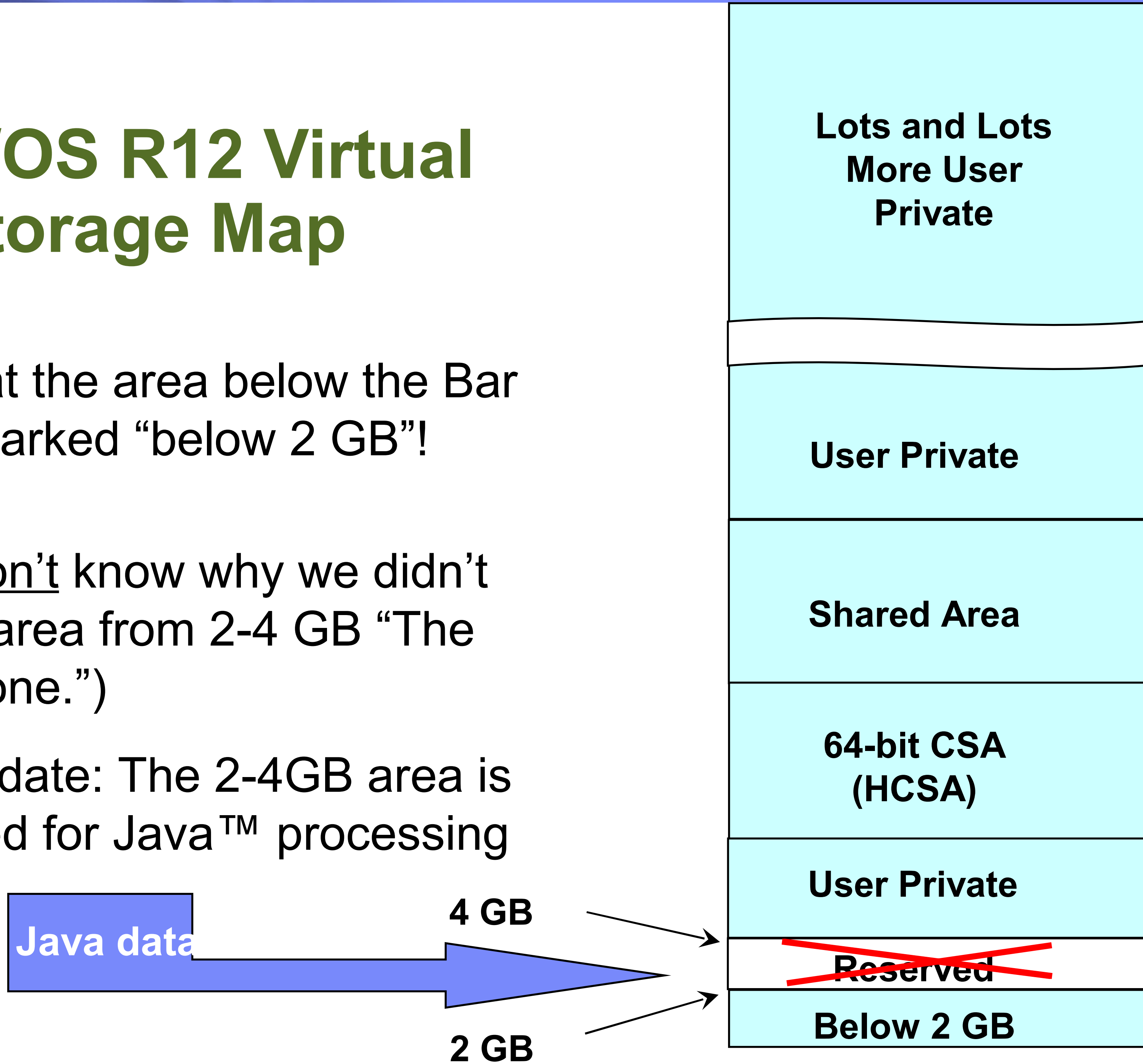


The z/OS R12 Virtual Storage Map

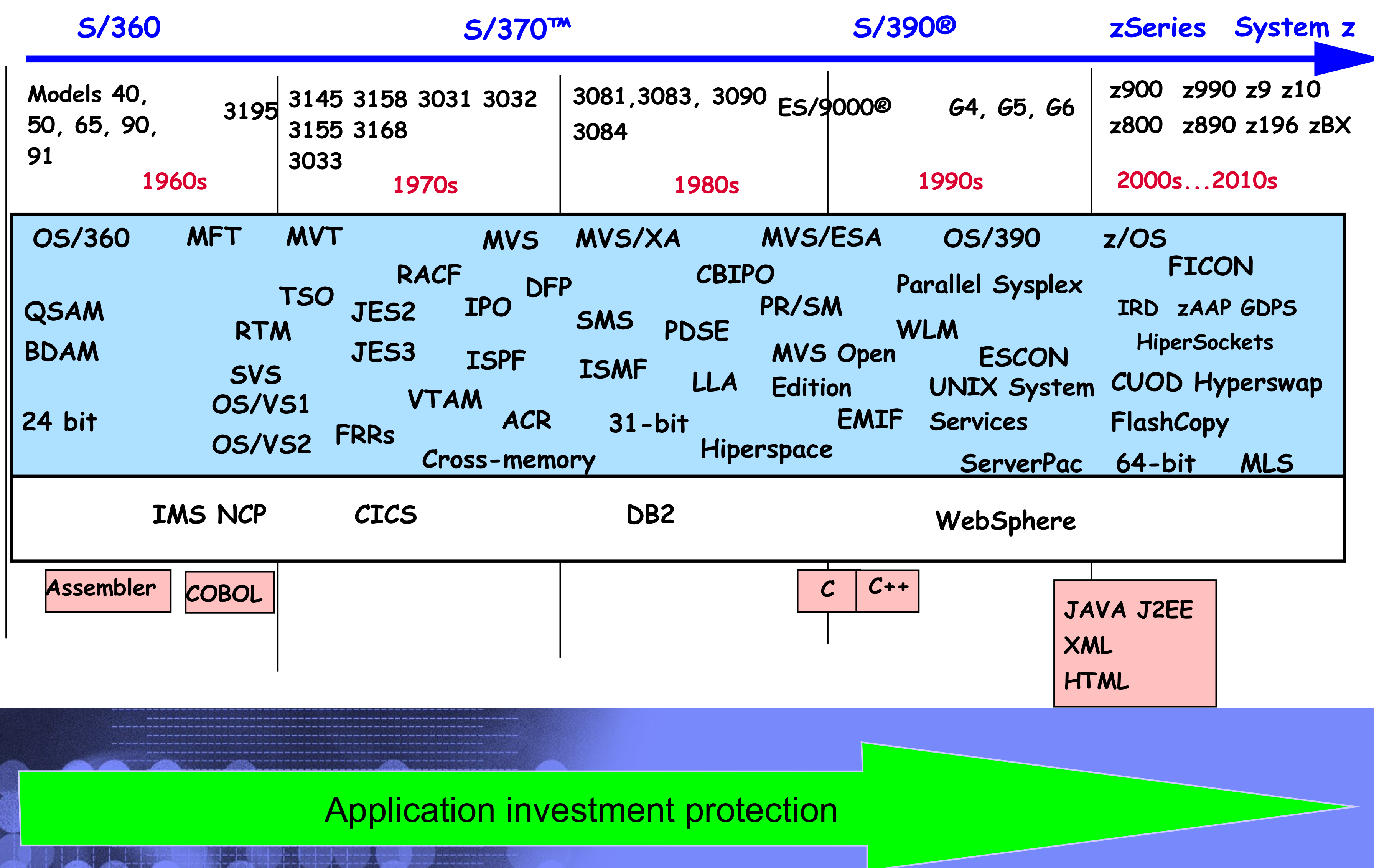
Note that the area below the Bar is just marked “below 2 GB”!

(No, I <u>don’t</u> know why we didn’t call the area from 2-4 GB “The Dead Zone.”)

2011 update: The 2-4GB area is now used for Java™ processing

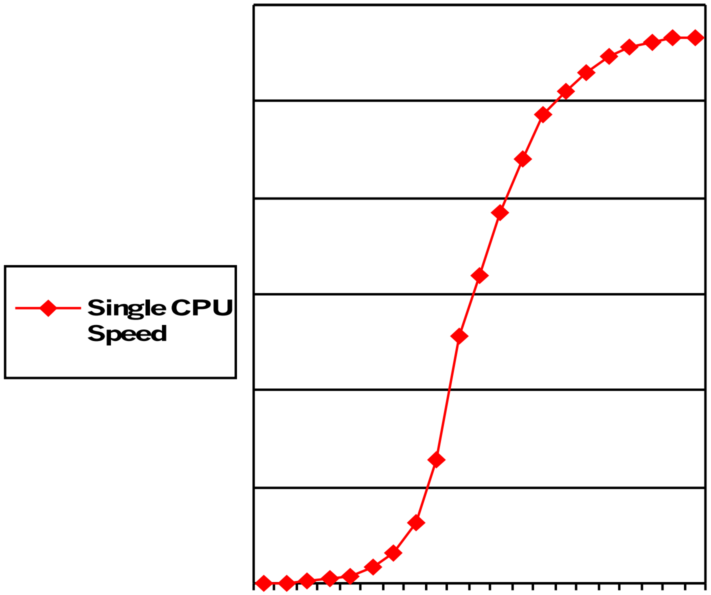


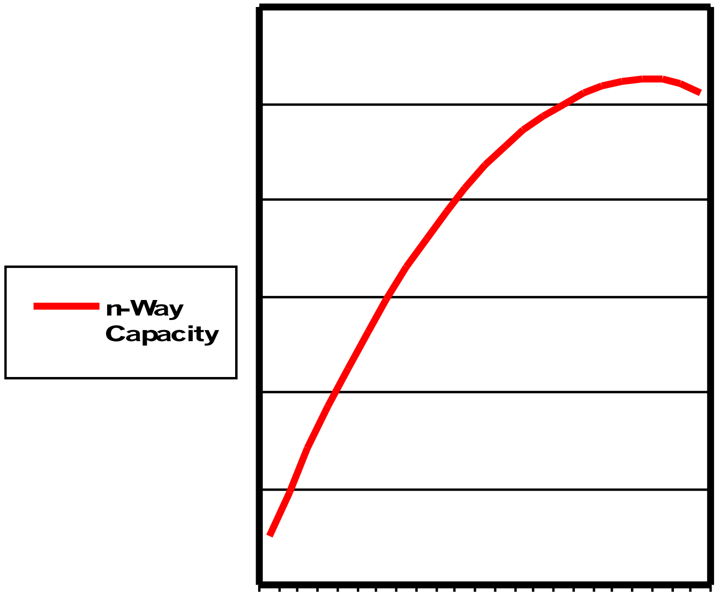
Over 40 Years of Innovation



Speed, speed, and more speed

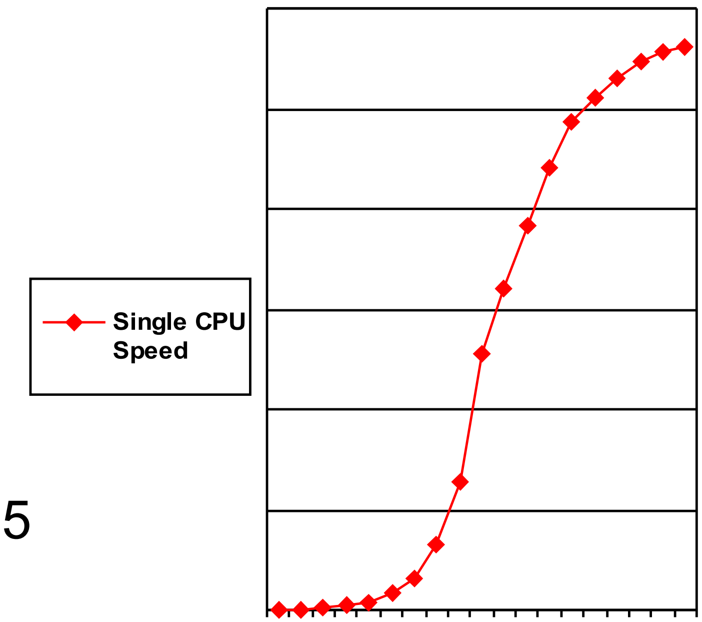
- Stealing a trick out of Grace Hopper's book…
  - **Here's a nanosecond's worth of wire…**
  - **Here's the cycle time of a 3168…**
  - **Here's the cycle time of a z196**
- You can start to see one of the problems facing the industry this way
- n-way scalability seems likely to become the order of the day for a while…
  - **When you last shopped for a home computer, how many were single-core?**
- …with horizontal scalability becoming necessary at some point
- The NextGen message? Learn to multiprogram and multithread well!



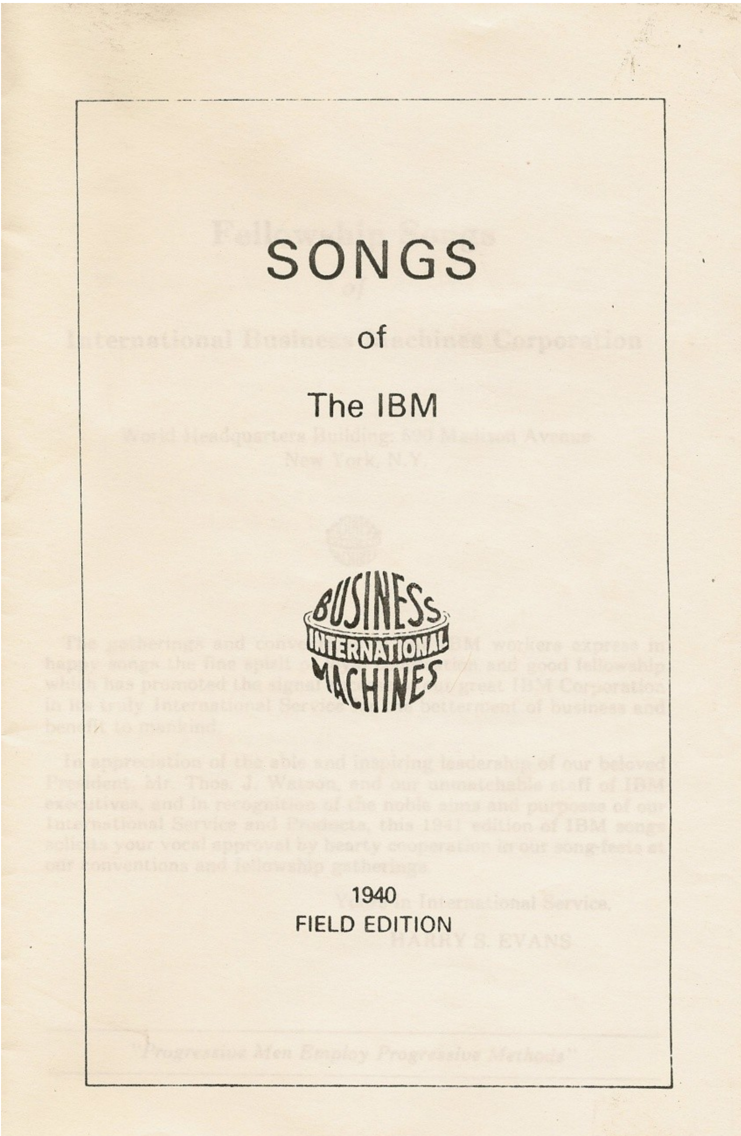


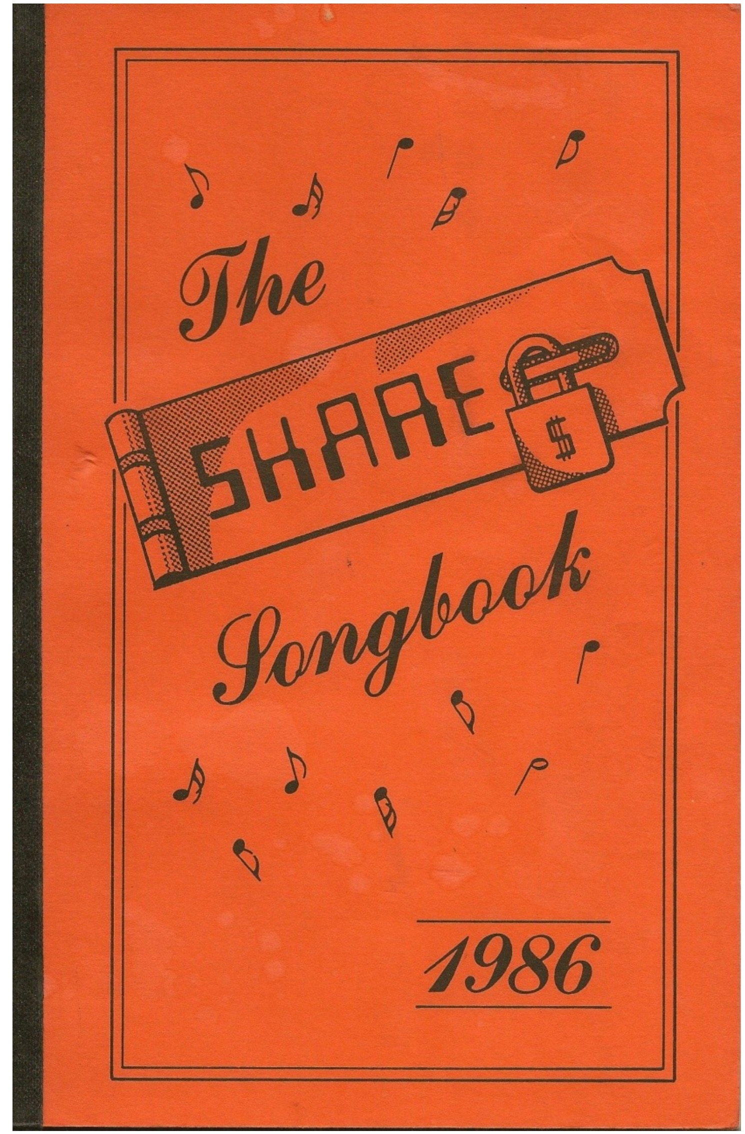
N-way Scale on MVS – z/OS

- MP (2-way) support with OS/VS2 Release 2 in 1974
  - **zAAPs and zIIPs didn't exist then)**
- 16-way support with MVS/XA™ in 1983
  - **3084 MP was 4-way**
  - **3090-600 was 6-way**
  - **(No zAAPs or zIIPs then, either!)**
- 32-way support with z/OS R6 on z990 servers in 2005
  - **Sum of CPs, zIIPs, and zAAPs in one z/OS LPAR**
- 54-way support with z/OS R9 on IBM System z9 EC servers in 2007
  - **Likewise, the sum of CPs, zIIPs, and zAAPs in one z/OS LPAR**
- 64-way support with z/OS R9 on IBM System z10 EC servers in 2008
  - **Still the sum of CPs, zIIPs, and zAAPs in one z/OS LPAR**
- 80-way support with z/OS R11 on IBM zEnterprise 196 servers in 2010
  - **This remains the sum of CPs, zIIPs, and zAAPs in one z/OS LPAR**



Other Trivia

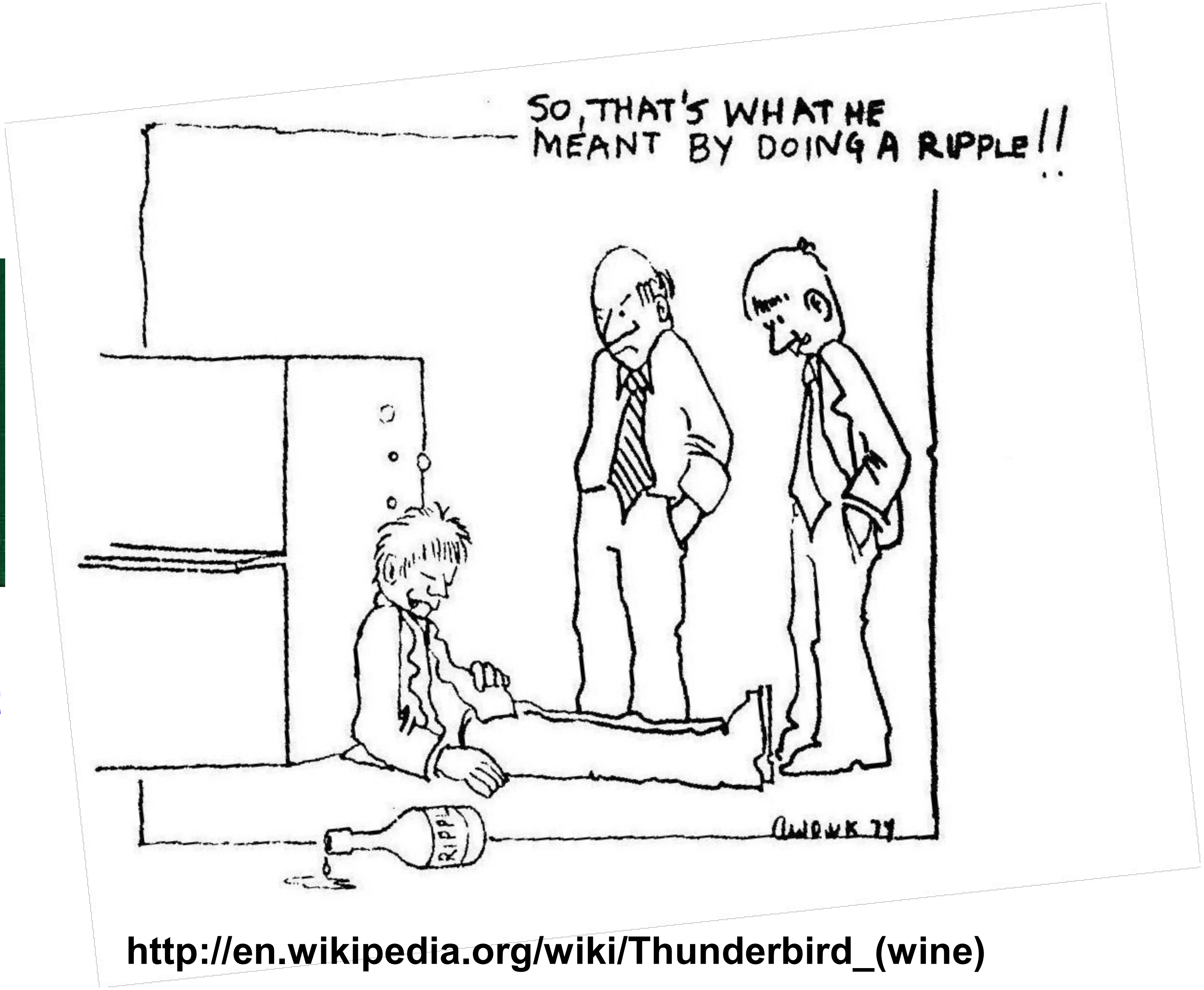




Other Trivia

Another 168 console pushbutton:

Nothing to do with "fortified" wine: It set a 1-byte data pattern from eight separate console toggle switches into every byte of real memory



http://en.wikipedia.org/wiki/Thunderbird_(wine)

Thanks for Coming

- Thanks for attending
- Hope you had fun…I did!

The Future Runs on System z

A SHAREd History of the Mainframe – Chronicles, Artifacts, and Stories

John Eells
IBM Poughkeepsie
SHARE 116, Session 9022
March 2, 2011

Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.

AIX*	FlashCopy*	MVS	System z
AnyNet*	GDPS*	Notes*	System z9
CICS*	HiperSockets	OMEGAMON*	System z10
DB2*	Hiperspace	Open Class*	SystemPac*
developerWorks*	HyperSwap	Parallel Sysplex*	Tivoli*
DFSMSdfp	IBM*	PR/SM	VTAM*
DFSMSdss	IBM eServer	Processor Resource/ Systems Manager	WebSphere*
DFSMShsm	IBM logo*	pSeries*	z/Architecture*
DFSMSrmm	IMS	RACF*	z/OS*
DFSORT*	Infoprint*	Redbook	z/VM*
DRDA*	IP PrintWay	RMF	zSeries*
DS8000	iSeries	System i	
ESCON*	Language Environment*	System Storage	
FICON*			

* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies.

The Trademark Blackberry® is owned by Research In Motion Limited and is registered in the United States and may be pending or registered in other countries. IBM is not endorsed, sponsored, affiliated with or otherwise authorized by Research In Motion Limited.

InfiniBand® is a registered trademark of the InfiniBand Trade Association (IBTA).
Intel is a trademark of the Intel Corporation in the United States and other countries.
Linux is a trademark of Linux Torvalds in the United States, other countries, or both.
Java and all Java-related trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc., in the United States and other countries.
Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation.
UNIX is a registered trademark of The Open Group in the United States and other countries.
All other products may be trademarks or registered trademarks of their respective companies.
The Open Group is a registered trademark of The Open Group in the US and other countries.

Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

This presentation and the claims outlined in it were reviewed for compliance with US law. Adaptations of these claims for use in other geographies must be reviewed by the local country counsel for compliance with local laws.

Where we are today…

z/OS 1.13 Preview*

A smarter operating system with designs for:

Improving Usability and Skills
z/OSMF Software Deployment and Storage Management applications, User-level mount command for z/OS UNIX System Services, Automatic F4DSCB updates, SDSF Sysplex functions to work without MQ, Catalog parmlib member, Better O/C/EOV Messages, Health Checks, …

Integrating new Applications and Supporting Industry and Open Standards
Java/COBOL interoperability, Improved Support for unnamed sections, ISPF Edit Macros, Subsystem and Unauthorized XTIOT support, dbx hookless debug, DFSORT improvements, Job level return codes, …

Scalability & Performance
Fully-shared zFS in a sysplex, RMODE 64 extensions, IFASMFDL improvements, 500K+ aliases per user catalog, Larger VVDSs, FREE=EOV, FTP support for large format data sets and EAS,…

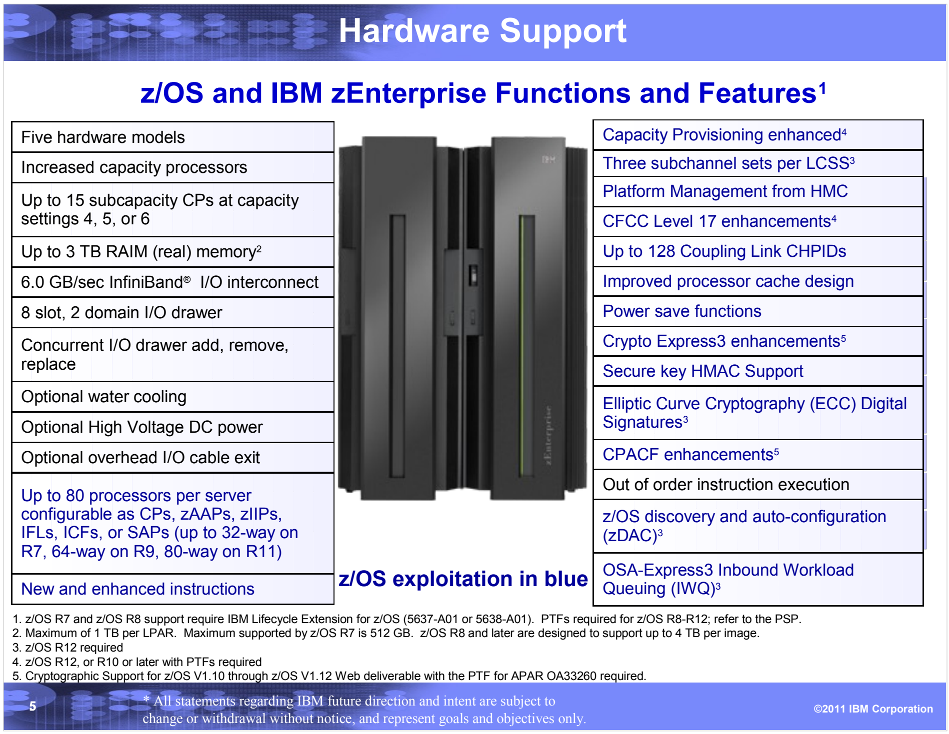
Enhancing Security
RRSF over TCP/IP, LDAP improvements, SAF security for z/OSMF, NAS address checking and encryption negotiation, New restricted QNAMEs, PKI support for DB2 backstore, ICSF support for new HMACs, FTP & TN3270 password phrase support, …

Improving Availability
Warn before TIOT exhaustion, CMDS enhancements, Parallel FTP for dump transfers, PFA ENQ tracking, RTD improvements, zFS Refresh, DADSM Dynamic Exits, JES3 dynamic spool addition, Better channel recovery, More ASID reuse, …

Self Managing Capabilities
WLM and RMF to provide response time distribution for all goals, DFSMShsm Journal Backup and space management improvements, …

Extending the Network
IDS IPv6 support, NAT Traversal for IKEV2, NMI extensions, More VLANs per OSA port, more 64-bit TCP/IP, EE improvements, …

* All statements regarding IBM future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.



Integration and exploitation of IBM zEnterprise System

•IBM introduces the IBM zEnterprise System -- a system that combines the gold standard of enterprise computing with built-in function to extend IBM's mainframe-like governance and qualities of service to special-purpose workload optimizers and general-purpose application serving. End-to-end management is enabled for this heterogeneous environment by the IBM zEnterprise Unified Resource Manager, which provides energy monitoring and management, goal-oriented policy-based workload monitoring and management, increased security, virtual networking, and data management, consolidated in a single interface that can be tied to business requirements. An IBM zEnterprise System is composed of the IBM zEnterprise 196, the IBM zEnterprise Unified Resource Manager, the IBM zEnterprise BladeCenter Extension (zBX), and optimizers or blades.

•The IBM zEnterprise 196 server adds additional scalability and performance capabilities for your z/OS environment.

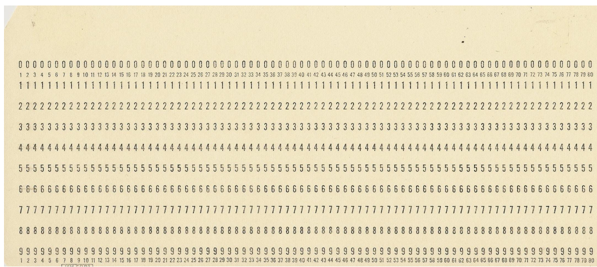
- The new 96-way core design (with 80 cores that are customer configurable) delivers massive scalability for secure data serving and transaction processing for large-scale businesses. The performance of a z196 (2817) processor is expected to be 1.3 to 1.5 times the performance of a z10 EC (2097) based on workload and model. The largest z196 (2817-780) is expected to exceed 1.6 times the capacity of the largest z10 (2097-764). It has up to twice the available real memory, 3 terabytes (TB) per server (with up to 1 TB real memory per LPAR) compared to the z10 EC Model E64. New quad-core 5.2 GHz processor chips, with more than 100 new instructions to enable improved code efficiency, are also designed to help improve the execution of Java and CPU-intensive workloads. For example,

Where we've been…

It all started with…

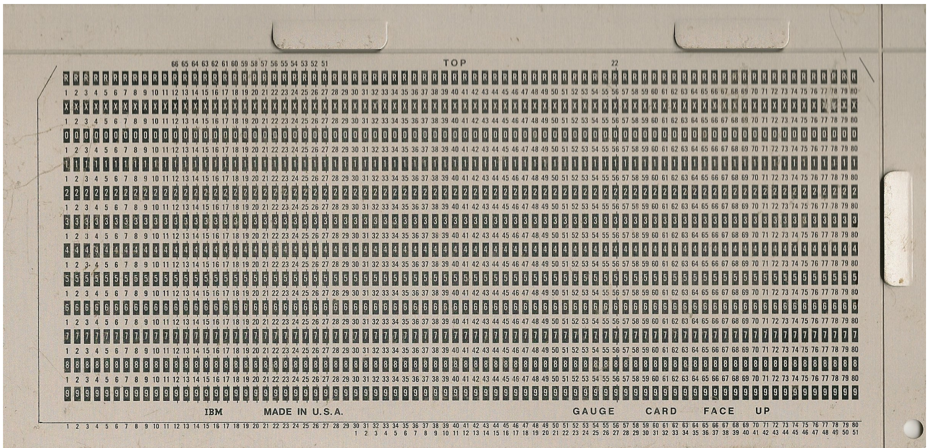
- …Herman Hollerith's punched cards…
- …and their influence continues to affect us today!

- Ever wonder…
  - Why the 3270 default screen size is 24x<u>80</u>?
  - Why we have a "block size" concept?
  - Why we have data sets with sequence numbers?



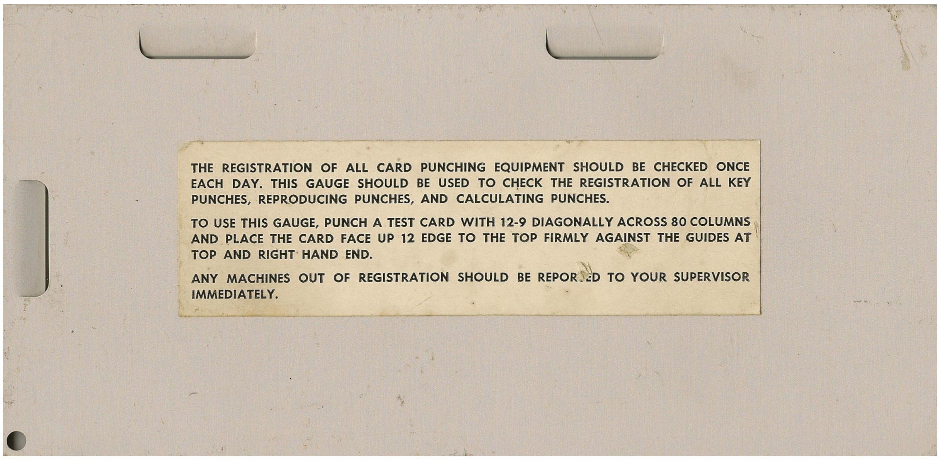
Punched Cards, Continued…

- How did you know the holes were in the right places?
- With a card registration plate, of course!
- Still standard issue in the 1970’s



Punched Cards, Continued…

Rear view of card registration plate



Punched Cards, Continued…

- It’s hard to believe this now, but punched cards were pervasive!
- Many bills and warranty cards were printed on punched cards
- “Do not fold, spindle, or mutilate…”
- This card came with my garbage disposal many moons ago:


IMPORTANT TO PURCHASER
GFC310 GFC310--02 VG125726B 42
CONSUMER PRODUCT OWNERSHIP REGISTRATION
YOUR PROMPT COMPLETION AND RETURN OF THIS CARD WILL FACILITATE OUR CONTACTING YOU IN THE UNLIKELY EVENT A SAFETY MODIFICATION IS ISSUED FOR YOUR PRODUCT UNDER THE CONSUMER PRODUCT SAFETY ACT.
OWNER REGISTRATION
GENERAL ELECTRIC COMPANY
LOUISVILLE, KY. 40225
DATE PLACED IN USE MONTH DAY YEAR
(PLEASE PRINT)
NAME ______ AREA CODE ______ TEL. NO. ______
APT. ______ STREET ______
CITY ______ COUNTY ______
STATE ______ ZIP ______
DEALER/BUILDER NAME ______
CITY ______ STATE ______
IMPORTANT–FILL IN AND MAIL THIS CARD TODAY!
DEALERS & BUILDERS: DO NOT REMOVE THIS CARD FROM THE PRODUCT

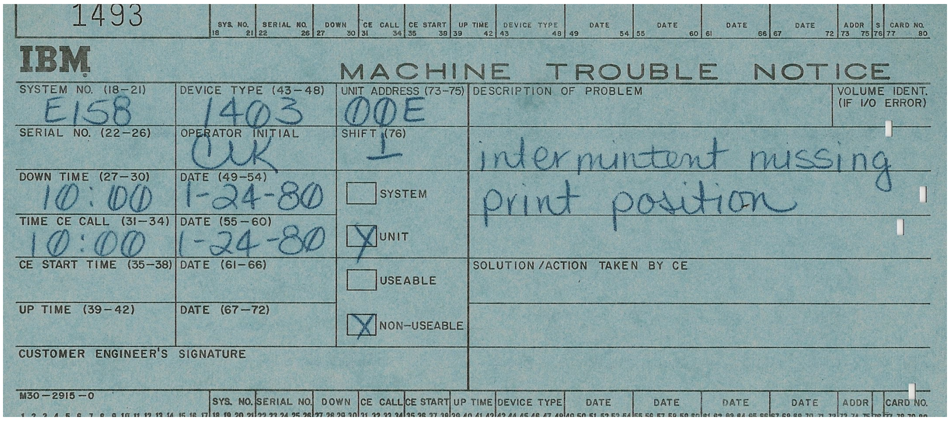
What's a Card Jam?

- When two cards tried to fit into the space meant for one, how did you get them out?
- You used a card saw…
- Once standard issue in CE tool bags, the thin (.010" or so), spring-steel card saw was essential if you worked on card readers, punches, or keypunch machines.
- It would clear out the card jam…eventually.

IBM 129 Card Removal Tool image courtesy of Mike Loewen, Pennsylvania State University (PSU)

Punched Cards, Continued…

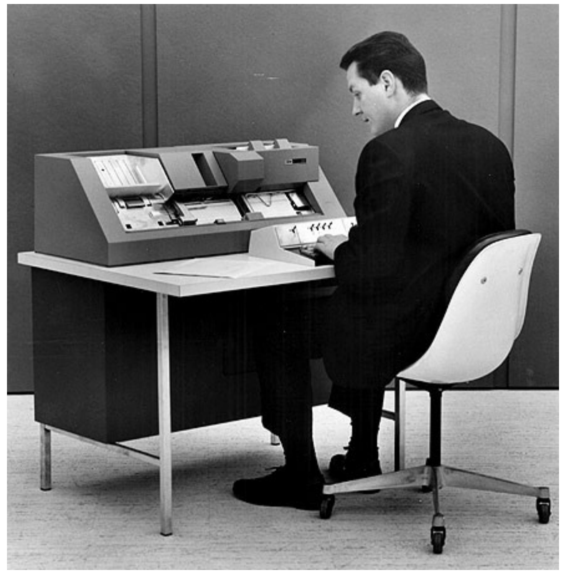
- Of course, IBM used punched cards, too:



An operator named Carol K. wrote this MTN against a printer I fixed in 1980…and I obviously forgot to return the card because I found it in my old tool bag in 2007!

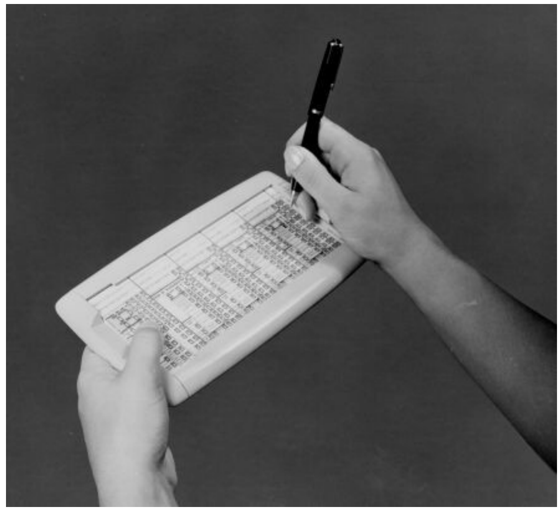
Punched Cards, Continued…

- An IBM 029 Keypunch, 1964
- Not exactly a laptop!
- It existed only to punch holes in cards
- Blank cards in feeder on top right; punched ones in stacker on left; chad bin underneath
- No error correction, of course; cards with typos went into the trash can (which is conspicuously absent in this photo)
- This is actually a model with an optional drum-mounted “template card” (I can’t recall the actual name) to speed things up



Punched Cards, Continued…

- There were no PDAs, then, but there was…
- The Port-A-Punch!
- “Designed to fit in the pocket”
  - I suppose pockets might have been larger then (some things <u>were</u> even before my time, after all)
- Not exactly a BlackBerry® handheld device!

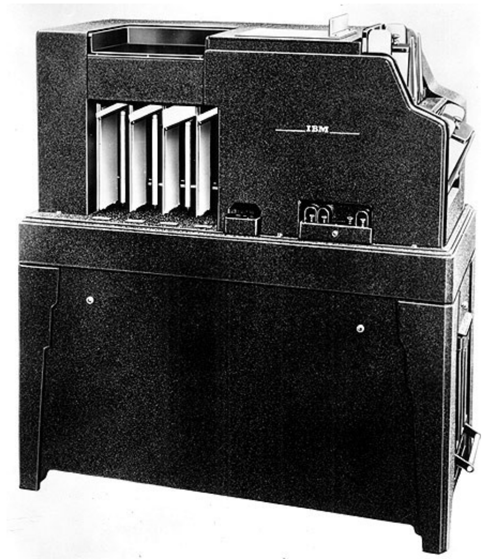


Port-A-Punch

IBM's Supplies Division introduced the Port-A-Punch in 1958 as a fast, accurate means of manually punching holes in specially scored IBM punched cards. Designed to fit in the pocket, Port-A-Punch made it possible to create punched card documents anywhere. The product was intended for "on-the-spot" recording operations -- such as physical inventories, job tickets and statistical surveys -- because it eliminated the need for preliminary writing or typing of source documents.

Punched Cards, Continued…

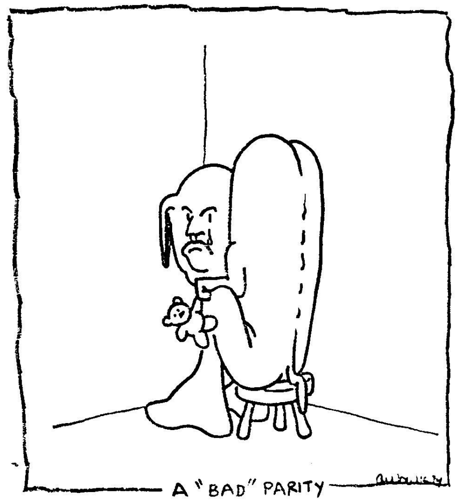
- An IBM 77 Collator
- A collator is the opposite of a sorter
- For some things, you didn't need a computer to make punched cards useful
- Today, we use things like SORT/MERGE's descendant, DFSORT



Hardware

•The industry did have to invent some things:

- Parity (IBM uses odd parity)
- NRZI recording for tape
- CRC and LRC checking algorithms
- ECC memory
- Microcode
- …the list goes on and on



Printers

- Printers like this 1403 came with a print train or print chain
- A hydraulic unit—sort of a 2-speed mini-automatic transmission—drove the tractors to move the paper
- Spacing and skips were controlled by a 12-channel carriage tape
  - It was just *amazing* how fast a box of paper could empty when one broke…
- Don't leave a cup of coffee on top!
  - Some models raised the cover automatically when out of paper to catch an operator's attention

Disk drives

- The 2314
- 9 drawers to a “bank” (because they were high maintenance, you could only use up to 8 at a time)
- Hydraulic pistons moved heads in & out of removable disk packs
- Removable disk packs and address plugs moved together to keep the same address for the same volume
- Don’t drop a disk pack! (The bits would fall off.)

Introduced in 1965, the IBM 2314 Direct Access Storage Facility provided eight independently operating disk drives and a spare along with a control unit in one facility. Users of large-scale computer systems could attach enough 2314s to provide nearly 10 billion bytes of data storage.

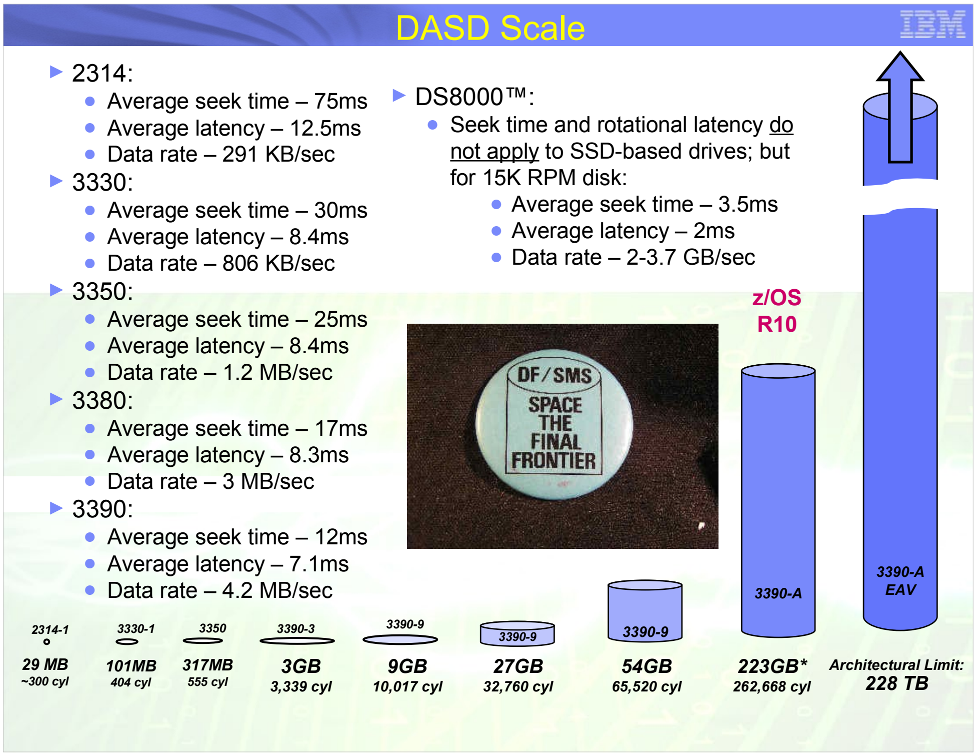
Disk drives

- The 3330, 1970
- Much more reliable than the 2914, so 8 drawers to a bank
- Voice coil electromagnet and large static magnet used to replace hydraulic unit to drive access mechanism
- Can still move disk pack and unit address plug together
- 101 MB/volume (3330-1) or 202 MB/volume (3330-11)

The IBM 3330 Data Storage (seen here in a design model) was a high-performance, high-capacity direct access storage subsystem for use with all IBM System/370 models as well as the IBM System/360 Model 195. Each 3330 subsystem could have from two to 16 drives, giving users up to 1.6 billion bytes of online storage. Developed and manufactured at IBM's facilities in San Jose, Calif., the 3330 was announced in 1970 and withdrawn from marketing 13 years later.

Disk drives

- A 3330 disk pack, called a 3336, alongside two Mass Storage Subsystem (3851) cartridges
- 10 2-sided data platters
- 19 data tracks per cylinder with 1 servo track



SHARE button image courtesy of Barry Merrill, Merrill Consultants

Tape drives

- The IBM 2420
- 1600 bpi!
- “Stubby” triangular vacuum columns at the top helped reduce start/stop inertia
- “Autoloading” tape covers
- Don’t forget the Write Ring! (Ever wonder where “RING” and “NORING” came from in the JES3 mount messages?)

Write Ring

Tape drives

- The IBM 3420, 1970
- Up to 6250 bpi!
- Odd models (3, 5, 7) were 1600 bpi only
- Even models (4, 6, 8) were 1600/6250 "Dual Density"
- Models 7 and 8 moved tape at 800 IPS
- High-speed rewind was *fast!* Cracked or broken, off-balance tape reels could disintegrate spectacularly, spreading plastic shrapnel throughout much of the machine.

Then, there were the CPUs

The 3168—IBM’s de facto flagship in 1977

•This picture does not convey the sheer scale of this machine:

- CPU frames ~7’ high
- Processor alone weighed nearly 3 ½ tons (6,881 lbs)
- It took time to walk by the CPU, console, CDU, PDU, and channel frames—MPs took more than double the space
- Channel-to-CPU cables nearly filled the space between 18” high raised floor posts across two floor tiles (4’ wide)
- 6.3 KVA for the CPU alone

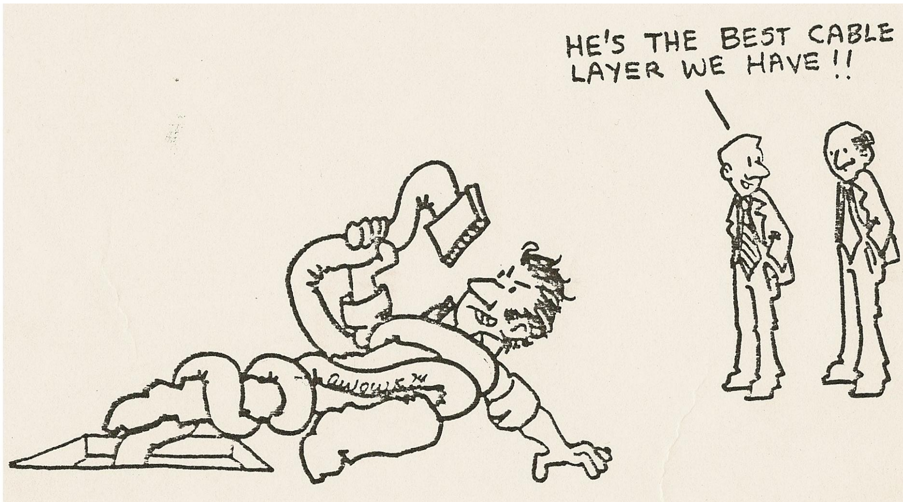
•Could run an MP as two “physically partitioned” UPs

The 3168 is shown with its Power and Cooling Distribution Units (PDU and CDU) behind it to the left and its console ahead of it; a 3803 control unit & 3420 tape drives to the far left; a 3830 control unit & 3330 disk drives on the far right; a 327x terminal, 3505 reader & 3525 punch in the left foreground; and 3211 printers in the right foreground. I can’t identify the box in the far right rear; it might be a 2701 or 2703 communications controller.

The displays on the left showed the state of various bits and pieces of hardware with indicator lights. Rotating knobs changed the labels and meanings of the lights. One light’s label was, simply, “Always On.” When the machine was powered up…it was!

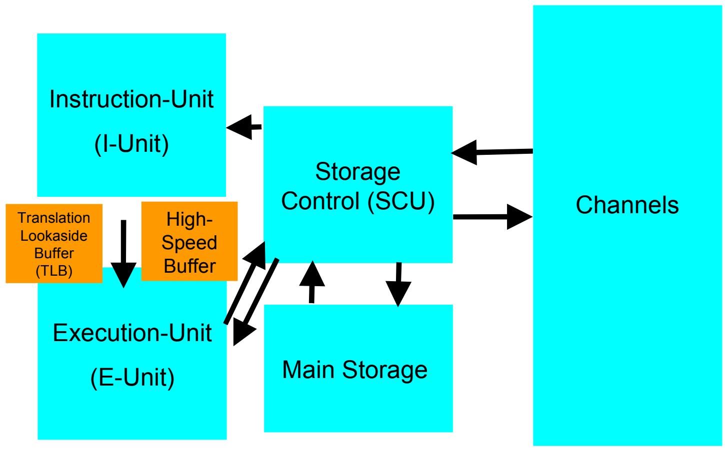
3168 Trivia

Did I mention the number of cables? Some over 100' long, and heavy. The rule for which end to plug in where was, "Light grey away."



3168 Block Diagram

The 168 was the basis for many later designs, and elements of its design still persist in today's servers



Hardware Diagnosis Wasn't Always Easy…

Shooting the bugs…

• Components were discrete before SLT packaging:

  • Transistors
  • Resistors
  • Capacitors
  • Inductors

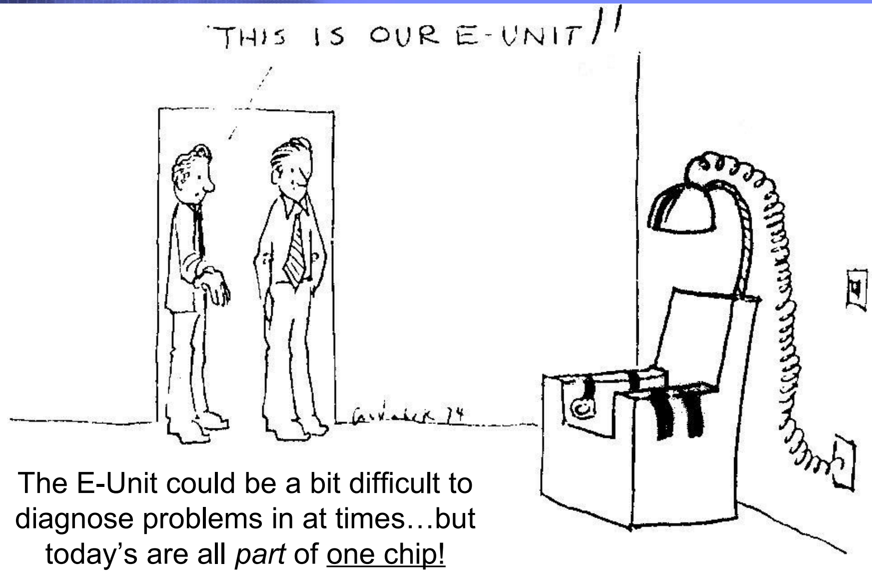
• CEs used microcode and software diagnostics, printed logic diagrams and oscilloscopes

• Processor diagnosis often involved putting instructions into memory manually from the console using switches and dials…

• …and then following the bug circuit by circuit until the problem was found

• Intermittent problems could be solved over time by using the 'scope's "single sweep" mode to monitor events one at a time

3168



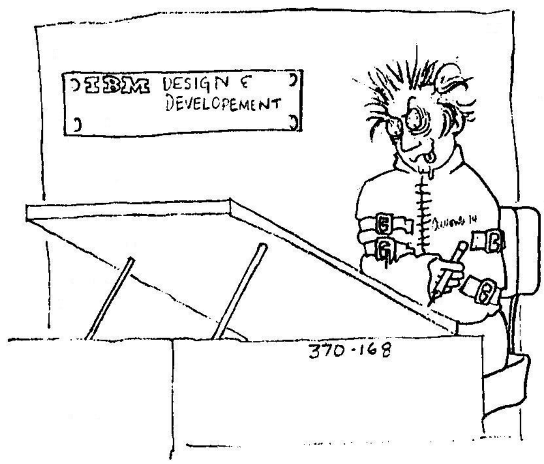
The E-Unit could be a bit difficult to diagnose problems in at times…but today's are all *part* of one chip!

3168 Trivia

- Cycle time: 80ns
    - About 1/416th the speed of a single z196 CP
- Max storage 8MB (16MB for an MP!)
    - That's less than 25% of the space needed to store the scans of the cartoons in this presentation…and 1/2000th of my phone's memory
- Worst-case storage access time: About 480ns (6 machine cycles, for a partial Store; most storage operations took 4 machine cycles, or 320ns)
- Board-to-board and frame-to-frame interconnections were done with "trileads," a semi-shielded three-part wire with push-on connectors
- Power-on-Reset (POR) set the TOD clock to 0 and loaded the microcode
- Console characters were <u>drawn</u> on the screen with continuous lines (not pixelated)

3168 Trivia

Some people apparently thought the overall design could stand improvement…

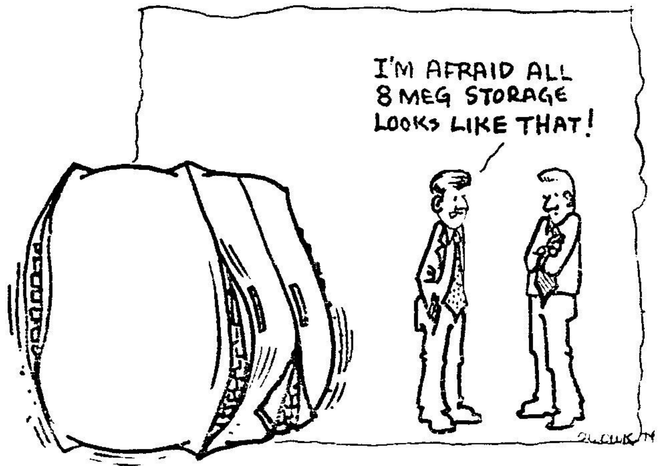


More 3168 Trivia

- Fed by a big, honkin' 240V 415Hz AC motor generator
- Cooled with (lots of!) chilled water (~30 Gal/m @ 52°F) through a large water-to-water heat exchanger in the Coolant Distribution Unit (CDU) and water-to-air exchangers inside the CPU frames
- HSM (High-Speed Multiply) internals were a trade secret, not disclosed even to CEs
- Up to 12 channels in separate frames connected with cables
  - Maximum channel data rate was 1.5 MB/sec
  - Only tape drives and 2305s could get close to that (3420's could read 6250 BPI at 800 IPS, minus overhead for IBGs, etc.)

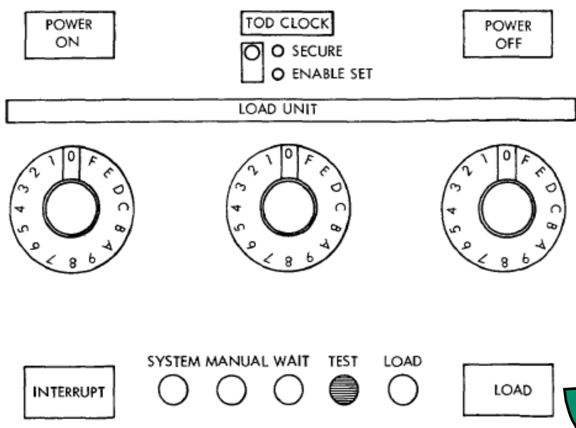
3168 Trivia

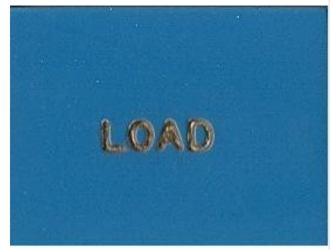
Needless to say, 8 MB was a lot of storage back then…



From a 168

Ever hear a coworker say “Hit Load” and wonder….?





More CPUs

- The 3033's were almost as big
  - But the new, 3270-based console was a bit smaller
  - No separate channel frames; "directors" were introduced, putting the channel subsystem entirely under the covers of the CPU, which reduced the total system footprint quite a bit
  - Maximum channel speed doubled to 3 MB/sec (the new cables were blue instead of grey but still had dark grey and light grey ends)
  - Maximum memory 16MB (32 for an MP!)
  - Cycle time: 57ns
  - Memory access time: 5 or 8 machine cycles (285 / 456 ns)

3033 shown with PDU and CDU to the back left and L-shaped 3270-based console in front. 3800 printer on the left has the optional Burster/Trimmer/Stacker feature; it's next to a 3505 card reader and 3525 punch. The box to the right rear might be a 3851 Mass Storage Subsystem (MSS). 3330s to the far right may have been MSS staging drives; they are flanked by 3350 disk drives.

Then, there were operating systems

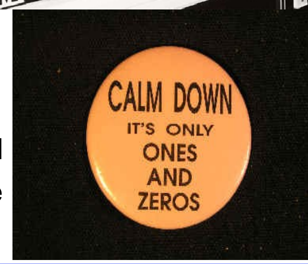
- But…what about the software?

- PCP
- MFT
- MVT
- SVS
- MVS™
- MVS/SP™
- MVS/XA™
- MVS/ESA™
- OS/390®
- z/OS ®

SHARE button images courtesy of Barry Merrill, Merrill Consultants

In the beginning…there were punched cards…

- …and Core Storage and PCP
  - No, nothing to do with Phencyclidine
  - PCP was the Primary Control Program
  - 32KB of main storage
  - Available in March 1966
- The life of an operator using PCP:
  - IPL from a card deck
  - Read in a job from another card deck
  - Job starts to process
  - Go hang tapes, feed printer, etc. as needed
  - Job finishes, machine goes into a wait state
  - Repeat



SHARE button image courtesy of Barry Merrill, Merrill Consultants

PCP

- Despite its one-job-at-a-time programming model, PCP was well-designed for its time:
  - The machines were incredibly slow by today's standards, with cycle times in the microsecond range
  - Machine utilization was actually pretty good except when the system was doing I/O because:
    - There wasn't that much data at first
      - A typical YouTube video might have required millions of punched cards…which could be read at the rate of about 300/minute
    - There wasn't enough memory to do any more, anyway!
- "Job scheduling" had to do only with whose job ran when

SHARE button image courtesy of Barry Merrill, Merrill Consultants

OS/MFT

- Still on core storage…
- But we learned how to make more of it, faster
- We could multi-task as we waited for I/O…at last!
- OS/MFT was born in 1966
  - Fence off storage areas in real memory, called "partitions"
  - Run a separate job in every partition
  - Re-IPL to change the number of partitions or their sizes
  - 64KB of storage!
  - "Job scheduling" took on a whole new meaning; not every job could run in the order it was handed to an operator as it could have been under PCP. IPLs often scheduled at specific times every day.

OS/MVT

- MFT was “Multiprogramming (with a) Fixed (number of) Tasks”
- MVT supported <u>Variable</u> tasks
- No IPL to change partition sizes any more!

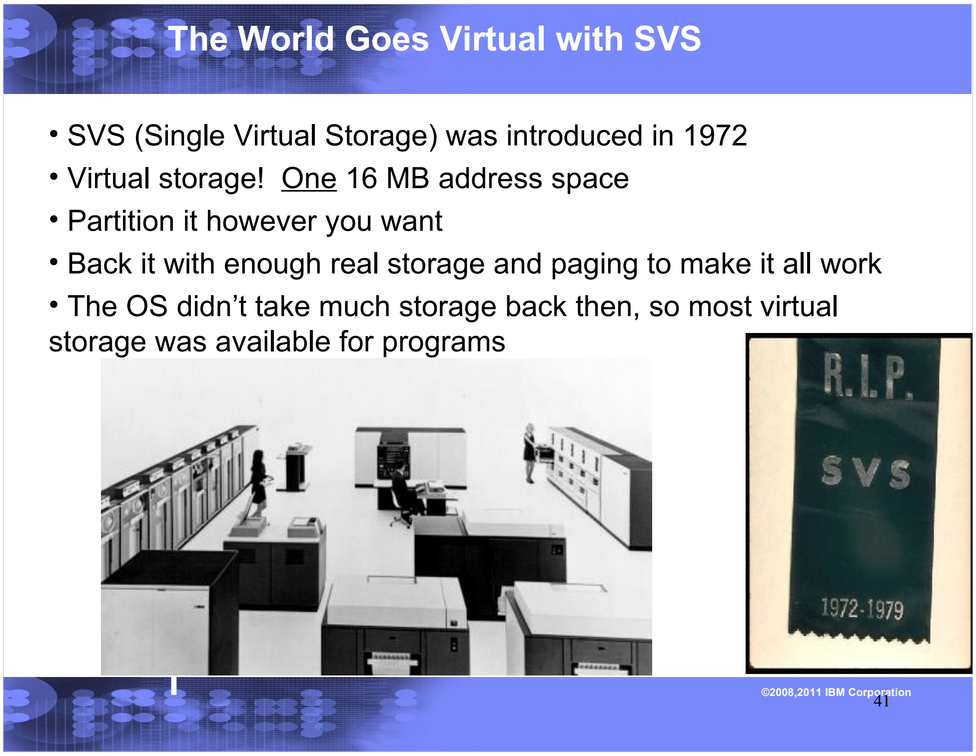
- OS/MVT, 1967:
  - Still real storage based – 128KB
  - Still ran a separate job in every region
  - Job scheduling became a bit easier since one need not re-IPL to change partition sizes, but still highly limited by the machine’s capacity
  - Online work begins to “interfere with” batch work at about this time

SHARE button image courtesy of Barry Merrill, Merrill Consultants

Layout of real memory under MVT

MVT could run up to 15 jobs concurrently.

The initators selected jobs from the queue, carved out the real memory to satisfy the region requirements, allocated data sets and passed control to the application programs.



Ribbon image courtesy of Barry Merrill, Merrill Consultants

3155 processor shown, flanked by 3330 disk drives with a 3830 control unit on right, a remote 3215 console and 3420 tape drives on left, a rear view of two 1403-N1 printers in the right foreground with a 2821 control unit to their left. Behind the 2821 is a 2540 card reader/punch; to its right is a 3211 printer.

This note is a formal non-working paper of the Project MAC Computer Systems Research Division. It should be reproduced and distributed wherever levity is lacking, and may be referenced at your own risk in other publications.

The Paging Game
By Jeff Berryman

RULES

1. Each player gets several million things.

2. Things are kept in crates that hold 4096 things each. Things in the same crate are called crate-mates.

3. Crates are stored either in the workshop or the warehouse. The workshop is almost always too small to hold all the crates.

4. There is only one workshop but there may be several warehouses. Everybody shares them.

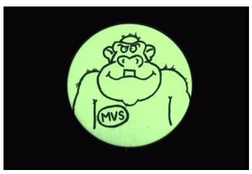
5. Each thing has its own thing number.

6. What you do with a thing is to zark it. Everybody takes turns zarking.

7. You can only zark your things, not anybody else's

Multiple Virtual Worlds with MVS

- MVS, 1974
- Multiple Virtual Storage = Multiple address spaces!
- Memory fragmentation (which forced frequent IPLs of SVS by today's standards) was much less a problem because initiators could be stopped and restarted to clean up when necessary
- Symmetric Multiprocessing introduced (3158, 3168, perhaps 3165 and 3155)
- JES2 and JES3 introduced
    - JES2 based on HASP
    - JES3 based on ASP



SHARE button image courtesy of Barry Merrill, Merrill Consultants

3032 processor shown without its PDU or CDU, with a 3851 MSS to the far left. 3505 reader and 3525 punch shown left of center, 3211 printer right of center. 3330 disk drives to the right rear. 3420 tape drives shown on the right.

Setting the bar for serious business

- The Philosophy of MVS
  - Assume the work is mission critical
  - Allow no undetected errors
  - Isolate all failures to the smallest affected unit of work
  - Provide diagnostics from the first failure sufficient to debug the problem
  - Allow no program access to data it is not authorized to access

Reliability, Availability, Serviceability

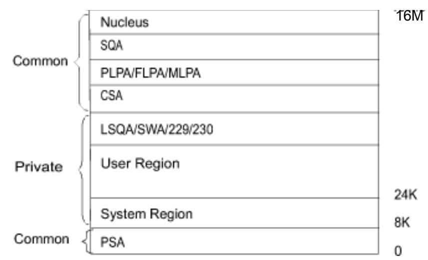
- Availability is intrinsic to the design
  - ☒ System (and subsystem) code is "covered" by a recovery routine.
  - ☒ Critical code has "nested recovery" to cover the recovery routines.
  - ☒ Diagnostic data ***specific to the error*** is gathered and reported.
  - ☒ Retry is attempted whenever possible after repairing damage and isolating the failure.

RAS is big business

- MVS RAS Guidelines
  - A lot of the code of any component (or subsystem) is devoted to RAS.
  - Hundreds of thousands of lines of RAS infrastructure code.
  - Tremendous synergy with hardware platform
  - Commitment to first failure data capture
  - Industry-unique commitment to system integrity

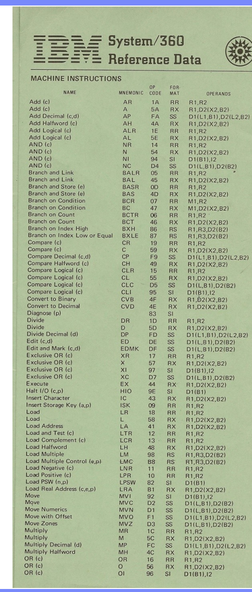
The MVS/SP Version 1 Virtual Storage Map

- Hey, back then 16 MB was a lot!



Reference Cards

- Nobody had yet imagined GUIs.
- There was a lot to remember.
- So, we had reference cards. Lots of reference cards.
- You've probably heard of this one—the "Green Card."



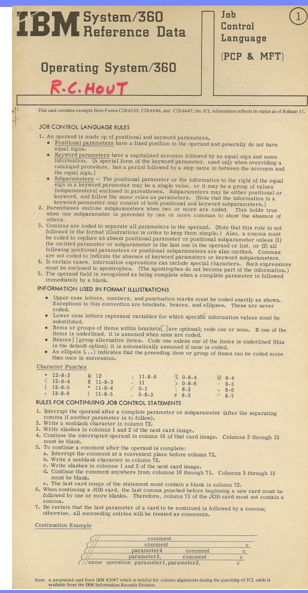
IBM System/360 Reference Data

MACHINE INSTRUCTIONS

NAME	MNEMONIC	OP CODE	FORMAT	OPERANDS
Add (c)	AR	1A	RR	R1,R2
Add (c)	A	5A	RX	R1,D2(X2,B2)
Add Decimal (c,d)	AP	FA	SS	D1(L1,B1),D2(L2,B2)
Add Halfword (c)	AH	4A	RX	R1,D2(X2,B2)
Add Logical (c)	ALR	1E	RR	R1,R2
Add Logical (c)	AL	5E	RX	R1,D2(X2,B2)
AND (c)	NR	14	RR	R1,R2
AND (c)	N	54	RX	R1,D2(X2,B2)
AND (c)	NI	94	SI	D1(B1),I2
AND (c)	NC	D4	SS	D1(L,B1),D2(B2)
Branch and Link	BALR	05	RR	R1,R2
Branch and Link	BAL	45	RX	R1,D2(X2,B2)
Branch and Store (e)	BASR	0D	RR	R1,R2
Branch and Store (e)	BAS	4D	RX	R1,D2(X2,B2)
Branch on Condition	BCR	07	RR	M1,R2
Branch on Condition	BC	47	RX	M1,D2(X2,B2)
Branch on Count	BCTR	06	RR	R1,R2
Branch on Count	BCT	46	RX	R1,D2(X2,B2)
Branch on Index High	BXH	86	RS	R1,R3,D2(B2)
Branch on Index Low or Equal	BXLE	87	RS	R1,R3,D2(B2)
Compare (c)	CR	19	RR	R1,R2
Compare (c)	C	59	RX	R1,D2(X2,B2)
Compare Decimal (c,d)	CP	F9	SS	D1(L1,B1),D2(L2,B2)
Compare Halfword (c)	CH	49	RX	R1,D2(X2,B2)
Compare Logical (c)	CLR	15	RR	R1,R2
Compare Logical (c)	CL	55	RX	R1,D2(X2,B2)
Compare Logical (c)	CLC	D5	SS	D1(L,B1),D2(B2)
Compare Logical (c)	CLI	95	SI	D1(B1),I2
Convert to Binary	CVB	4F	RX	R1,D2(X2,B2)
Convert to Decimal	CVD	4E	RX	R1,D2(X2,B2)
Diagnose (p)		83	SI	
Divide	DR	1D	RR	R1,R2
Divide	D	5D	RX	R1,D2(X2,B2)
Divide Decimal (d)	DP	FD	SS	D1(L1,B1),D2(L2,B2)
Edit (c,d)	ED	DE	SS	D1(L,B1),D2(B2)
Edit and Mark (c,d)	EDMK	DF	SS	D1(L,B1),D2(B2)
Exclusive OR (c)	XR	17	RR	R1,R2
Exclusive OR (c)	X	57	RX	R1,D2(X2,B2)
Exclusive OR (c)	XI	97	SI	D1(B1),I2
Exclusive OR (c)	XC	D7	SS	D1(L,B1),D2(B2)
Execute	EX	44	RX	R1,D2(X2,B2)
Halt I/O (c,p)	HIO	9E	SI	D1(B1)
Insert Character	IC	43	RX	R1,D2(X2,B2)
Insert Storage Key (a,p)	ISK	09	RR	R1,R2
Load	LR	18	RR	R1,R2
Load	L	58	RX	R1,D2(X2,B2)
Load Address	LA	41	RX	R1,D2(X2,B2)
Load and Test (c)	LTR	12	RR	R1,R2
Load Complement (c)	LCR	13	RR	R1,R2
Load Halfword	LH	48	RX	R1,D2(X2,B2)
Load Multiple	LM	98	RS	R1,R3,D2(B2)
Load Multiple Control (e,p)	LMC	B8	RS	R1,R3,D2(B2)
Load Negative (c)	LNR	11	RR	R1,R2
Load Positive (c)	LPR	10	RR	R1,R2
Load PSW (n,p)	LPSW	82	SI	D1(B1)
Load Real Address (c,e,p)	LRA	B1	RX	R1,D2(X2,B2)
Move	MVI	92	SI	D1(B1),I2
Move	MVC	D2	SS	D1(L,B1),D2(B2)
Move Numerics	MVN	D1	SS	D1(L,B1),D2(B2)
Move with Offset	MVO	F1	SS	D1(L1,B1),D2(L2,B2)
Move Zones	MVZ	D3	SS	D1(L,B1),D2(B2)
Multiply	MR	1C	RR	R1,R2
Multiply	M	5C	RX	R1,D2(X2,B2)
Multiply Decimal (d)	MP	FC	SS	D1(L1,B1),D2(L2,B2)
Multiply Halfword	MH	4C	RX	R1,D2(X2,B2)
OR (c)	OR	16	RR	R1,R2
OR (c)	O	56	RX	R1,D2(X2,B2)
OR (c)	OI	96	SI	D1(B1),I2

Reference Cards

- But there were cards before the green one
- Like this one, a JCL reference card for PCP and MFT
- It was originally white but has yellowed with age
- (Bob Hout gave me this card when he retired several years ago)



IBM System/360 Reference Data

Operating System/360

R.C. HOUT

Job Control Language (PCP & MFT)

1

This card contains excerpts from Forms C28-6539, C28-6646, and C28-6647; the JCL information reflects its status as of Release 11.

JOB CONTROL LANGUAGE RULES

1. An operand is made up of positional and keyword parameters.
   - Positional parameters have a fixed position in the operand and generally do not have equal signs.
   - Keyword parameters have a capitalized acronym followed by an equal sign and some information. (A special form of the keyword parameter, used only when overriding a cataloged procedure, has a period followed by a step name in between the acronym and the equal sign.)
   - Subparameters — The positional parameter or the information to the right of the equal sign in a keyword parameter may be a single value, or it may be a group of values (subparameters) enclosed in parentheses. Subparameters may be either positional or keyword, and follow the same rules as parameters. (Note that the information in a keyword parameter may consist of both positional and keyword subparameters.)
2. Parentheses enclose subparameters when two or more are coded. This holds true when one subparameter is preceded by one or more commas to show the absence of others.
3. Commas are coded to separate all parameters in the operand. (Note that this rule is not followed in the format illustrations in order to keep them simple.) Also, a comma must be coded to replace an absent positional parameter or positional subparameter unless (1) the omitted parameter or subparameter is the last one in the operand or list, or (2) all following positional parameters or positional subparameters are also omitted. Commas are not coded to indicate the absence of keyword parameters or keyword subparameters.
4. In certain cases, information expressions can include special characters. Such expressions must be enclosed in apostrophes. (The apostrophes do not become part of the information.)
5. The operand field is recognized as being complete when a complete parameter is followed immediately by a blank.

INFORMATION USED IN FORMAT ILLUSTRATIONS

- Upper case letters, numbers, and punctuation marks must be coded exactly as shown. Exceptions to this convention are brackets, braces, and ellipses. These are never coded.
- Lower case letters represent variables for which specific information values must be substituted.
- Items or groups of items within brackets [ ] are optional; code one or none. If one of the items is underlined, it is assumed when none are coded.
- Braces { } group alternative items. Code one unless one of the items is underlined (this is the default option); it is automatically assumed if none is coded.
- An ellipsis (...) indicates that the preceding item or group of items can be coded more than once in succession.

Character Punches

. 12-8-3	& 12	; 11-8-6	% 0-8-4	@ 8-4
< 12-8-4	$ 11-8-3	- 11	> 0-8-6	' 8-5
(12-8-5	* 11-8-4	/ 0-1	: 8-2	= 8-6
+ 12-8-6	) 11-8-5	, 0-8-3	# 8-3	" 8-7

RULES FOR CONTINUING JOB CONTROL STATEMENTS

1. Interrupt the operand after a complete parameter or subparameter (after the separating comma if another parameter is to follow).
2. Write a nonblank character in column 72.
3. Write slashes in columns 1 and 2 of the next card image.
4. Continue the interrupted operand in column 16 of that card image. Columns 3 through 15 must be blank.
5. To continue a comment after the operand is complete:
   a. Interrupt the comment at a convenient place before column 72.
   b. Write a nonblank character in column 72.
   c. Write slashes in columns 1 and 2 of the next card image.
   d. Continue the comment anywhere from columns 16 through 71. Columns 3 through 15 must be blank.
   e. The last card image of the statement must contain a blank in column 72.
6. When continuing a JOB card, the last comma punched before beginning a new card must be followed by one or more blanks. Therefore, column 71 of the JOB card must not contain a comma.
7. Be certain that the last parameter of a card to be continued is followed by a comma; otherwise, all succeeding entries will be treated as comments.

Continuation Example

```
//             comment
//             comment                    c
//      parameter4        comment         c
//      parameter3,       comment         c
//name  operation  parameter1,parameter2, c
```

Note: A preprinted card form IBM 42047 which is helpful for column alignments during the punching of JCL cards is available from the IBM Information Records Division.

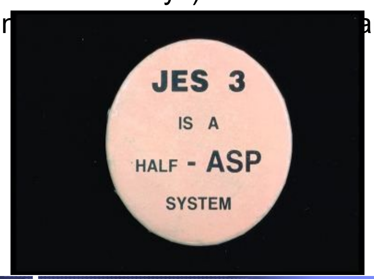
Reference Cards

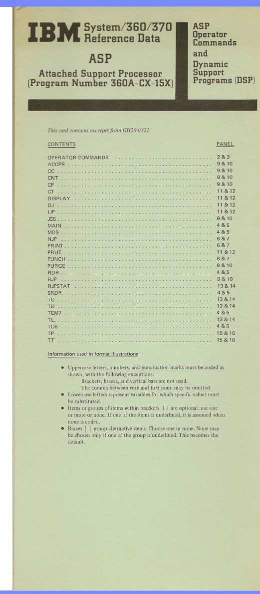
• Here's an Attached Support Processor (ASP) reference card

• ASP was JES3's forerunner, and it included Dynamic Support Programs (DSPs)

• As with JES3, commands started with an asterisk (*)

• If you've ever heard anyone say (or seen the SHARE button that says) "JES3 is a 'half-ASP' system," r ant





IBM System/360/370 Reference Data

ASP

Attached Support Processor
(Program Number 360A-CX-15X)

ASP Operator Commands and Dynamic Support Programs (DSP)

This card contains excerpts from GH20-0321.

CONTENTS	PANEL
OPERATOR COMMANDS	2 & 3
ACCPR	9 & 10
CC	9 & 10
CNT	9 & 10
CP	9 & 10
CT	11 & 12
DISPLAY	11 & 12
DJ	11 & 12
IJP	11 & 12
JSS	9 & 10
MAIN	4 & 5
MDS	4 & 5
NJP	6 & 7
PRINT	6 & 7
PRUT	11 & 12
PUNCH	6 & 7
PURGE	9 & 10
RDR	4 & 5
RJP	9 & 10
RJPSTAT	13 & 14
SRDR	4 & 5
TC	13 & 14
TD	13 & 14
TEM7	4 & 5
TL	13 & 14
TOS	4 & 5
TP	15 & 16
TT	15 & 16

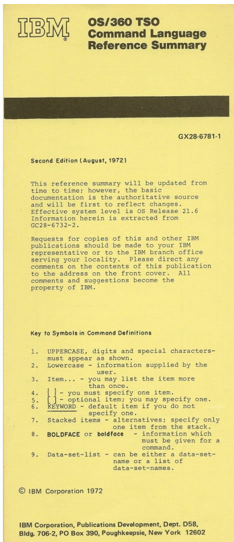
Information used in format illustrations

- Uppercase letters, numbers, and punctuation marks must be coded as shown, with the following exceptions:
  Brackets, braces, and vertical bars are not used.
  The comma between verb and first noun may be omitted.
- Lowercase letters represent variables for which specific values must be substituted.
- Items or groups of items within brackets [ ] are optional; use one or more or none. If one of the items is underlined, it is assumed when none is coded.
- Braces { } group alternative items. Choose one or none. None may be chosen only if one of the group is underlined. This becomes the default.

SHARE button image courtesy of Barry Merrill, Merrill Consultants

Reference Cards

- Here are a few more:



IBM **OS/360 TSO Command Language Reference Summary**

GX28-6781-1

Second Edition (August, 1972)

This reference summary will be updated from time to time; however, the basic documentation is the authoritative source and will be first to reflect changes. Effective system level is OS Release 21.6 Information herein is extracted from GC28-6732-2.

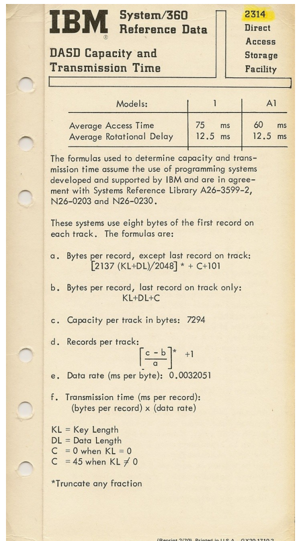
Requests for copies of this and other IBM publications should be made to your IBM representative or to the IBM branch office serving your locality. Please direct any comments on the contents of this publication to the address on the front cover. All comments and suggestions become the property of IBM.

Key to Symbols in Command Definitions

1. UPPERCASE, digits and special characters- must appear as shown.
2. Lowercase - information supplied by the user.
3. Item... - you may list the item more than once.
4. { } - you must specify one item.
5. [ ] - optional item; you may specify one.
6. KEYWORD - default item if you do not specify one.
7. Stacked items - alternatives; specify only one item from the stack.
8. **BOLDFACE** or **boldface** - information which must be given for a command.
9. Data-set-list - can be either a data-set-name or a list of data-set-names.

© IBM Corporation 1972

IBM Corporation, Publications Development, Dept. D58, Bldg. 706-2, PO Box 390, Poughkeepsie, New York 12602



IBM **System/360 Reference Data**

DASD Capacity and Transmission Time

2314 Direct Access Storage Facility

Models:	1	A1
Average Access Time	75 ms	60 ms
Average Rotational Delay	12.5 ms	12.5 ms

The formulas used to determine capacity and transmission time assume the use of programming systems developed and supported by IBM and are in agreement with Systems Reference Library A26-3599-2, N26-0203 and N26-0230.

These systems use eight bytes of the first record on each track. The formulas are:

a. Bytes per record, except last record on track:
$[2137\ (KL+DL)/2048]^{*} + C+101$

b. Bytes per record, last record on track only:
$KL+DL+C$

c. Capacity per track in bytes: 7294

d. Records per track:
$\left[\frac{c-b}{a}\right]^{*} +1$

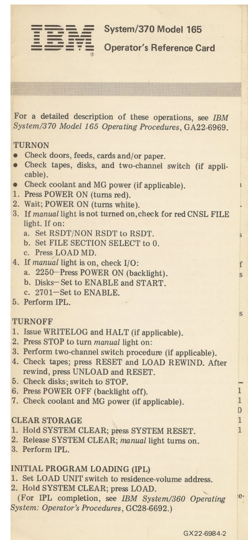
e. Data rate (ms per byte): 0.0032051

f. Transmission time (ms per record):
(bytes per record) x (data rate)

KL = Key Length
DL = Data Length
C = 0 when KL = 0
C = 45 when KL $\neq$ 0

*Truncate any fraction

(Reprint 2/70) Printed in U.S.A. GX20-1710-2



IBM **System/370 Model 165 Operator's Reference Card**

For a detailed description of these operations, see *IBM System/370 Model 165 Operating Procedures*, GA22-6969.

TURNON
- Check doors, feeds, cards and/or paper.
- Check tapes, disks, and two-channel switch (if applicable).
- Check coolant and MG power (if applicable).

1. Press POWER ON (turns red).
2. Wait; POWER ON (turns white).
3. If *manual* light is not turned on, check for red CNSL FILE light. If on:
   a. Set RSDT/NON RSDT to RSDT.
   b. Set FILE SECTION SELECT to 0.
   c. Press LOAD MD.
4. If *manual* light is on, check I/O:
   a. 2250—Press POWER ON (backlight).
   b. Disks—Set to ENABLE and START.
   c. 2701—Set to ENABLE.
5. Perform IPL.

TURNOFF
1. Issue WRITELOG and HALT (if applicable).
2. Press STOP to turn *manual* light on:
3. Perform two-channel switch procedure (if applicable).
4. Check tapes; press RESET and LOAD REWIND. After rewind, press UNLOAD and RESET.
5. Check disks; switch to STOP.
6. Press POWER OFF (backlight off).
7. Check coolant and MG power (if applicable).

CLEAR STORAGE
1. Hold SYSTEM CLEAR; press SYSTEM RESET.
2. Release SYSTEM CLEAR; *manual* light turns on.
3. Perform IPL.

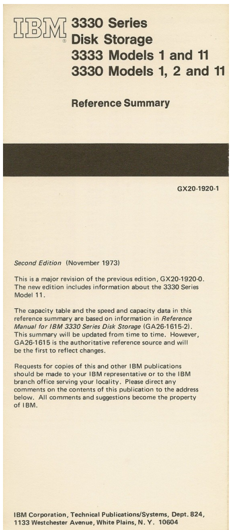
INITIAL PROGRAM LOADING (IPL)
1. Set LOAD UNIT switch to residence-volume address.
2. Hold SYSTEM CLEAR; press LOAD.

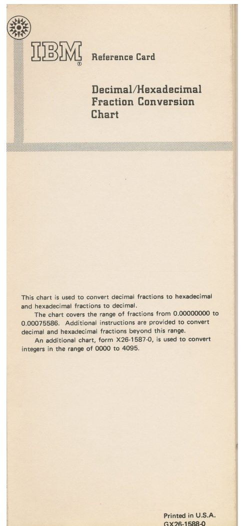
(For IPL completion, see *IBM System/360 Operating System: Operator's Procedures*, GC28-6692.)

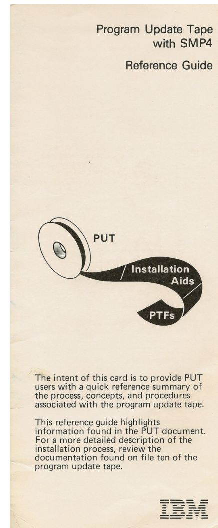
GX22-6984-2

Reference Cards

- …and some more:

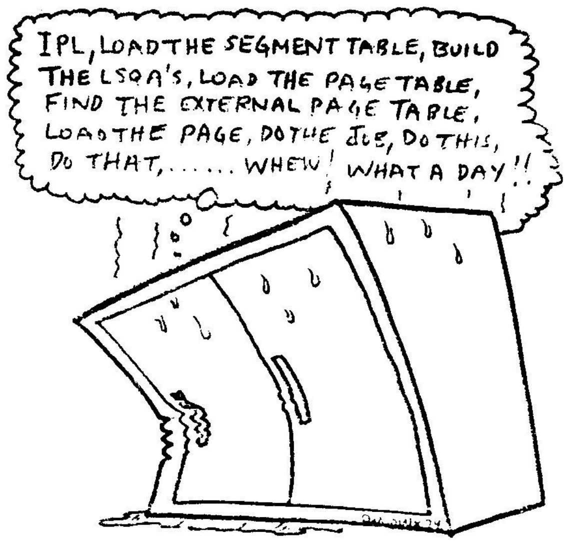






MVS

IPL processing got a bit more complicated…



MVS

As did diagnosis…

Diagnosis

- Notwithstanding the prior chart…
- AMDPRDMP was the forerunner to IPCS
- A standalone dump took about half a box of paper on a large, busy system…at least, until XA came along:
  - With MVS/XA, a typical standalone dump was like Midwestern corn by the 4th of July—that is, "knee high."
  - With MVS/ESA, with dataspace storage, they could be…well…bigger!
  - With 64-bit z/OS, it's a Good Thing we have IPCS.

SHARE button image courtesy of Barry Merrill, Merrill Consultants

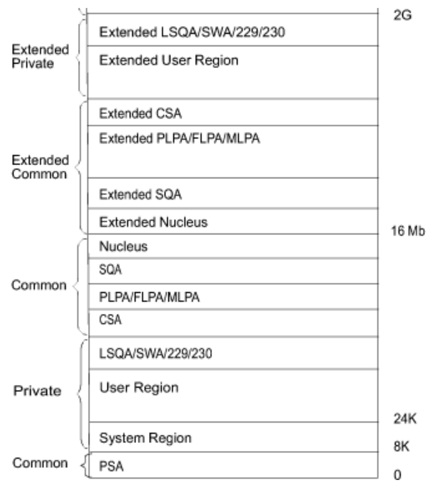
MVS/XA

- MVS/XA Version 2 was introduced in 1981
- 31-bit addressing hits the streets
- 2 GB of virtual storage per address space looked like infinite space back then (you can trust me on this)
- Required new hardware, 3083, 3081 or 3084 (3084 pictured)
- Introduction of Dynamic Path Selection on DASD controllers
- Hardware and software both incredibly reliable by early 1980's standards
    - Months between IPLs vs. days or weeks
    - We found out we had been <u>relying</u> on frequent IPLs for some business processes
    - And, we found new problems related to the longer life of an IPL—like initiator fragmentation, now (mostly) a thing of the past

SHARE button images courtesy of Barry Merrill, Merrill Consultants

The MVS/XA Version 2 Virtual Storage Map

- After 16 MB…
- …2 GB looked like it would last us a long time…
- …which, of course, it did not.



MVS/ESA

- MVS/ESA Version 3, 1988
- Introduced access registers, linkage stacks, data spaces, and Hiperspace™
- PR/SM introduced on 3090 at about the same time, creating LPAR mode (later the only mode) in addition to Basic Mode
- System-Managed Storage (SMS) introduced, with the Interactive Storage Management Facility (ISMF)
- LLA and VLF introduced, along with what is now z/OS UNIX System Services

SHARE button images courtesy of Barry Merrill, Merrill Consultants

MVS/ESA, continued

- MVS/ESA SP Version 4, 1991
- Extended Multi-Image Facility (EMIF) introduced for PR/SM™
- MVS/ESA SP 4.1 introduced:
  - Sysplex
  - HCD
- Available on:
  - 1600 or 6250 bpi open-reel tape
  - 3480 (uncompressed only)

MVS/ESA, continued

- MVS/ESA SP Version 5, 1994
  - Parallel Sysplex® introduced
    - IMS™ data sharing is first exploiter
  - Workload Manager (WLM) is introduced
    - Defined policies for the system's workloads
    - Goal-based performance management
    - CICSPlex®/SM and VTAM® are first cross-system exploiters

SHARE button image courtesy of Barry Merrill, Merrill Consultants

OS/390 – Putting it All Together

- OS/390, 1996
  - 72 products in one*!
    - Former products became "elements" of OS/390, some at no additional charge, some not
  - All tested together, at the same time
    - Intended to improve quality by letting us focus our test efforts
  - Ordering became simpler—checklist was much shorter because there were fewer options
  - New installation vehicle, ServerPac, replaced CBIPO

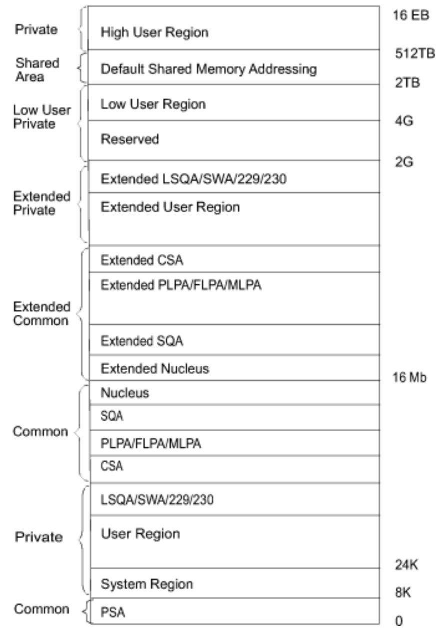
*OK, so not all of them were separate products.

z/OS – The Next Generation

- z/OS, 2000
  - The beginning of 64-bit addressing
    - First exploiters were access methods, HFS, XRC, and DB2
  - Support for IRD
    - WLM moves work to resources
    - IRD moves resources to work
      - LPAR CPU management
      - I/O priority queueing
      - Dynamic Channel Path Management
  - Workload license charges introduced
  - SNA Master Console Support
  - ServerPac provides a Recommended System Layout function to automatically place data sets on volumes

The z/OS R9 Virtual Storage Map

- We are hopeful that 16 EB will last us *at least* a few more months…
- …and if it's not, I'm not sure what we'll do next…
- One wag noted that 128-bit addressing would require more silicon atoms than we think there might be in the known universe

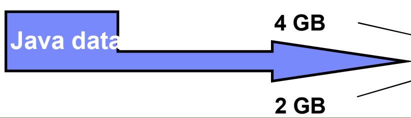


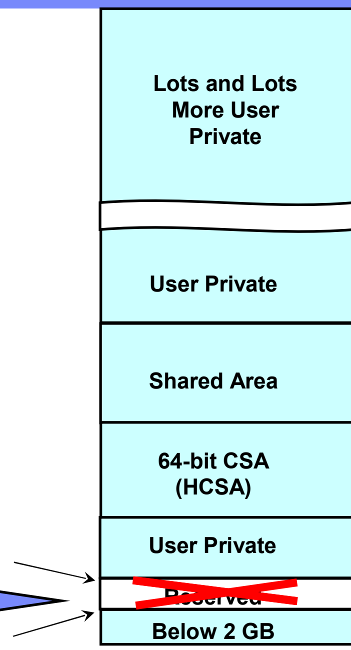
The z/OS R12 Virtual Storage Map

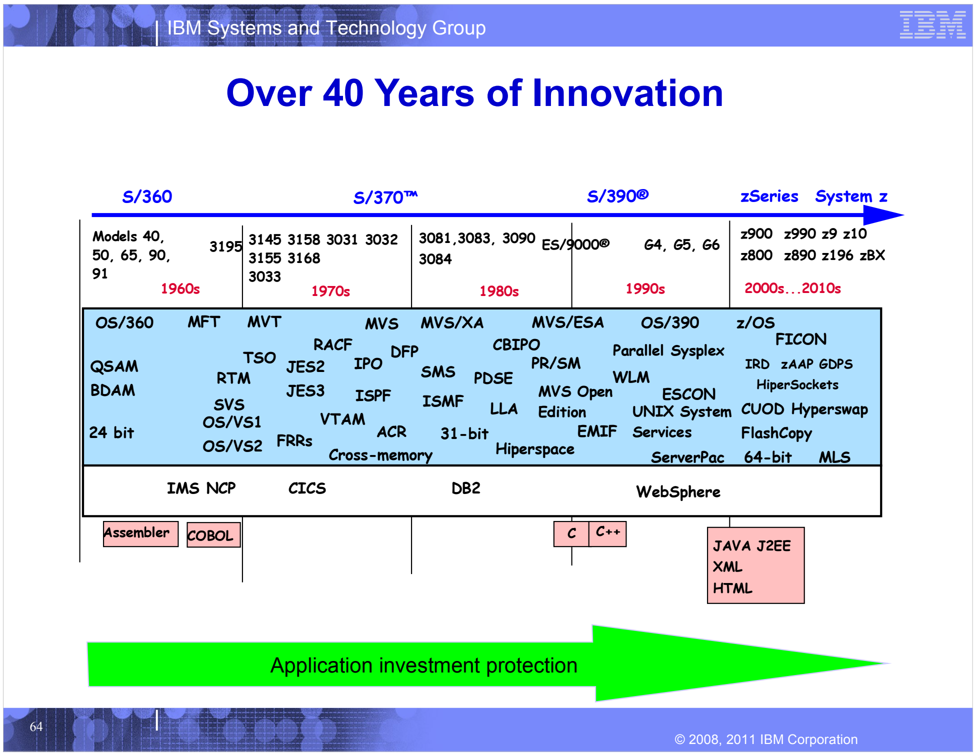
Note that the area below the Bar is just marked “below 2 GB”!

(No, I <u>don’t</u> know why we didn’t call the area from 2-4 GB “The Dead Zone.”)

2011 update: The 2-4GB area is now used for Java™ processing



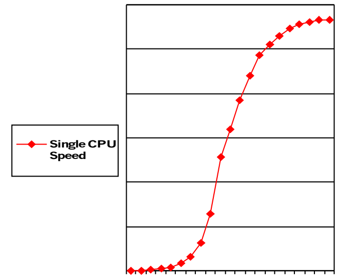


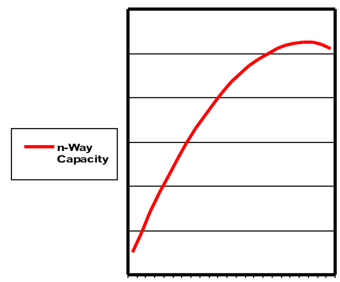


System/360 was introduced in 1964. System z, its successor, is the result of over 45 years of constant innovation and refinement. From the System/360 Model 40 to the zEnterprise 196, and from the beginnings of OS/360 to z/OS, new capabilities and technologies have been added while protecting your investment in existing applications.

Speed, speed, and more speed

- Stealing a trick out of Grace Hopper's book…
  - **Here's a nanosecond's worth of wire…**
  - **Here's the cycle time of a 3168…**
  - **Here's the cycle time of a z196**
- You can start to see one of the problems facing the industry this way
- n-way scalability seems likely to become the order of the day for a while…
  - **When you last shopped for a home computer, how many were single-core?**
- …with horizontal scalability becoming necessary at some point
- The NextGen message? Learn to multiprogram and multithread well!



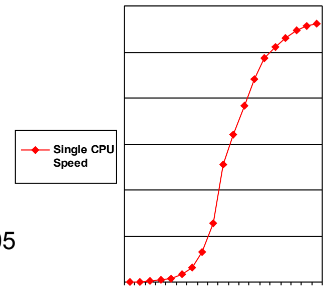


(All numbers approximate. The velocity factor of wire is variable. However, the ratios should be accurate.)

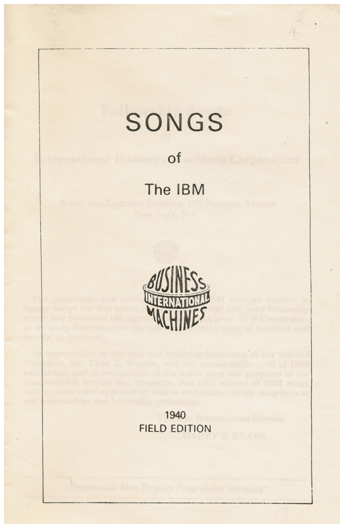
1' per nanosecond
80' for the cycle time of a 3168
2.3" for the cycle time of a z196

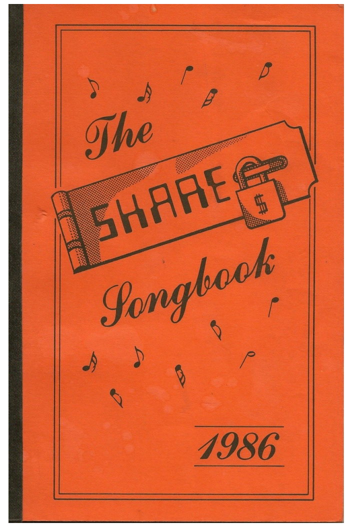
N-way Scale on MVS – z/OS

- MP (2-way) support with OS/VS2 Release 2 in 1974
  - **zAAPs and zIIPs didn't exist then)**
- 16-way support with MVS/XA™ in 1983
  - **3084 MP was 4-way**
  - **3090-600 was 6-way**
  - **(No zAAPs or zIIPs then, either!)**
- 32-way support with z/OS R6 on z990 servers in 2005
  - **Sum of CPs, zIIPs, and zAAPs in one z/OS LPAR**
- 54-way support with z/OS R9 on IBM System z9 EC servers in 2007
  - **Likewise, the sum of CPs, zIIPs, and zAAPs in one z/OS LPAR**
- 64-way support with z/OS R9 on IBM System z10 EC servers in 2008
  - **Still the sum of CPs, zIIPs, and zAAPs in one z/OS LPAR**
- 80-way support with z/OS R11 on IBM zEnterprise 196 servers in 2010
  - **This remains the sum of CPs, zIIPs, and zAAPs in one z/OS LPAR**



Other Trivia

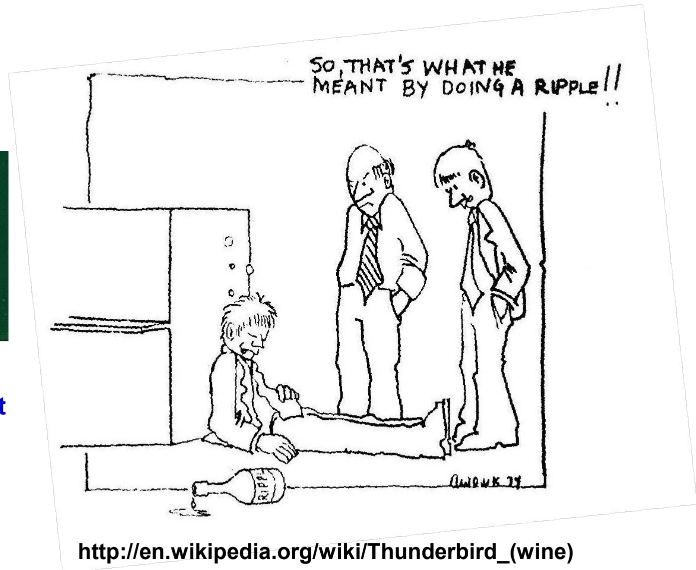




Other Trivia

Another 168 console pushbutton:

Nothing to do with "fortified" wine: It set a 1-byte data pattern from eight separate console toggle switches into every byte of real memory



http://en.wikipedia.org/wiki/Thunderbird_(wine)

Thanks for Coming

- Thanks for attending
- Hope you had fun…I did!


The Future Runs on System z
Optimize your z/OS environment
z/OS
©2008,2011 IBM Corporation