High Performance Computing in Ancient History

(~Fifty Years Ago)

Marty Itzkowitz, Software Consultant

itzkowitzmarty@gmail.com

(Many slides and content are thanks to Richard Friedman)

Program Agenda

- Introduction
- The IBM 709x Series
- The CDC 6600
- The CDC 7600
- Mass Storage Technologies
- Conclusion

High-Performance Computing in Ancient History

- This talk arose from a brown-bag session at LBL/NERSC, February 3, 2016
    - Arranged by Richard Friedman, who many of you know
    - Eight alumni of the systems group at LBL from 1960-1975 presented
    - Forty folks from NERSC showed up; here are the eight old-timers:

Jeremy Knight Bill Gage Bryan Higgins Eric Beals
Richard Friedman Marty Itzkowitz Bill Benson Bill Johnston

High Performance Computing Fifty Years Ago

- This talk gives my personal perspective on three eras and their machines:
  - Late 1950's through late 1960's
    - Scientific Computing dominated by IBM 709x
  - Mid 1960's through early 1970's
    - Scientific Computing dominated by CDC 6600
  - Early 1970's through late 1970's
    - Scientific Computing dominated by CDC 7600

After that came the rise of the minicomputers and the ascendance of Cray

Note that both the CDC machines were also designed by Seymour Cray

The Programming World Fifty Years Ago

- Operating Systems:
    - Batch programming only
    - Every site rolled their own OS, all proprietary and mutually incompatible
    - All written in assembler
    - No UNIX
- Programming Languages:
    - Assembler and FORTRAN (Some Pascal, PL/1, APL, COBOL, Algol -- not used for HPC)
    - No lower case: 6-bit characters
    - No C, C++, or Java
- Data and programs stored on tapes or card decks
    - No remote access; no interactive terminals
- No email, no internet (but we did have SneakerNet)

Program Agenda

- Introduction
- The IBM 709x Series
- The CDC 6600
- The CDC 7600
- Mass Storage Technologies
- Conclusion

The IBM 709x Machines, I

- 709 was vacuum-tube-based; 7090 and 7094 were transistor based
- Cycle time 7090: ~10 µsec. == 100 KHz; 7094: ~2 µsec. == 500 KHz
- 32K 36-bit words, four fields/word:

Opcode	Decrement	Tag	Address
3	15	3	15

- One 36-bit accumulator and one 36-bit MQ register
- 15-bit Index registers, specified by Tag field
  - 7090: 3 registers, specifying multiple bits or'd the registers
  - 7094: 7 registers, specifying multiple bits designated the register
    - “Multiple Tag Mode” made a 7094 behave like a 7090
  - Effective address was sum of address and contents of index register(s)
    - Or'ing the registers allowed a 2-D table lookup in one instruction

The IBM 709x Machines, II

- Programming Languages: FORTRAN and Assembler
  - Very early Fortran -- no logical if's, among other things
  - Only upper-case characters could be shown; 6-bit BCD characters
- Machine had no OS to speak of
  - System disk held compiler, assembler, libraries
  - Jobs linked with library for tape I/O, and a few other things
  - No protection or privilege
    - Running job had all of memory
    - Job could only read or write a tape that was mounted
      - To prevent corrupting someone else's data/program

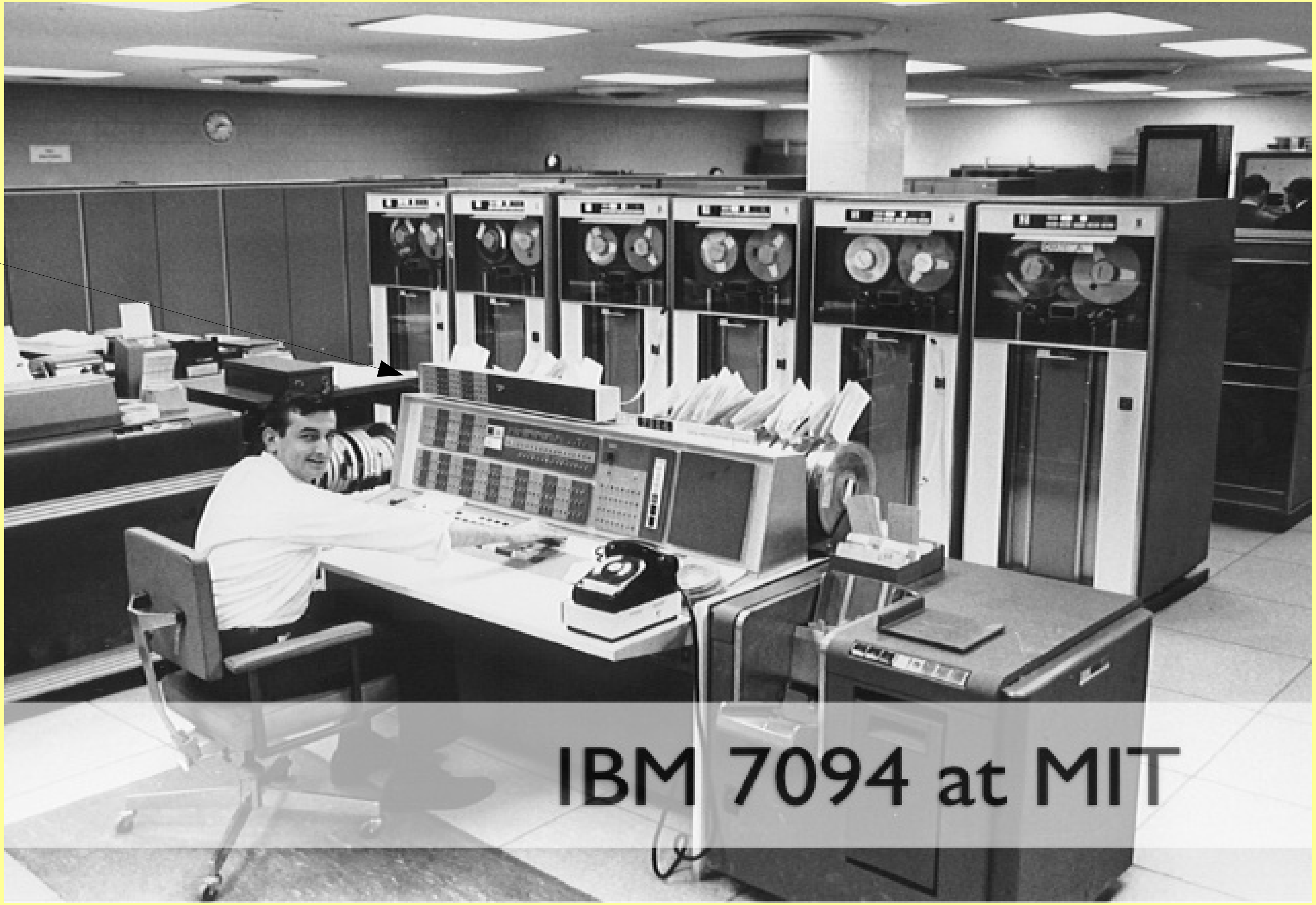
The IBM 709x Machines, III

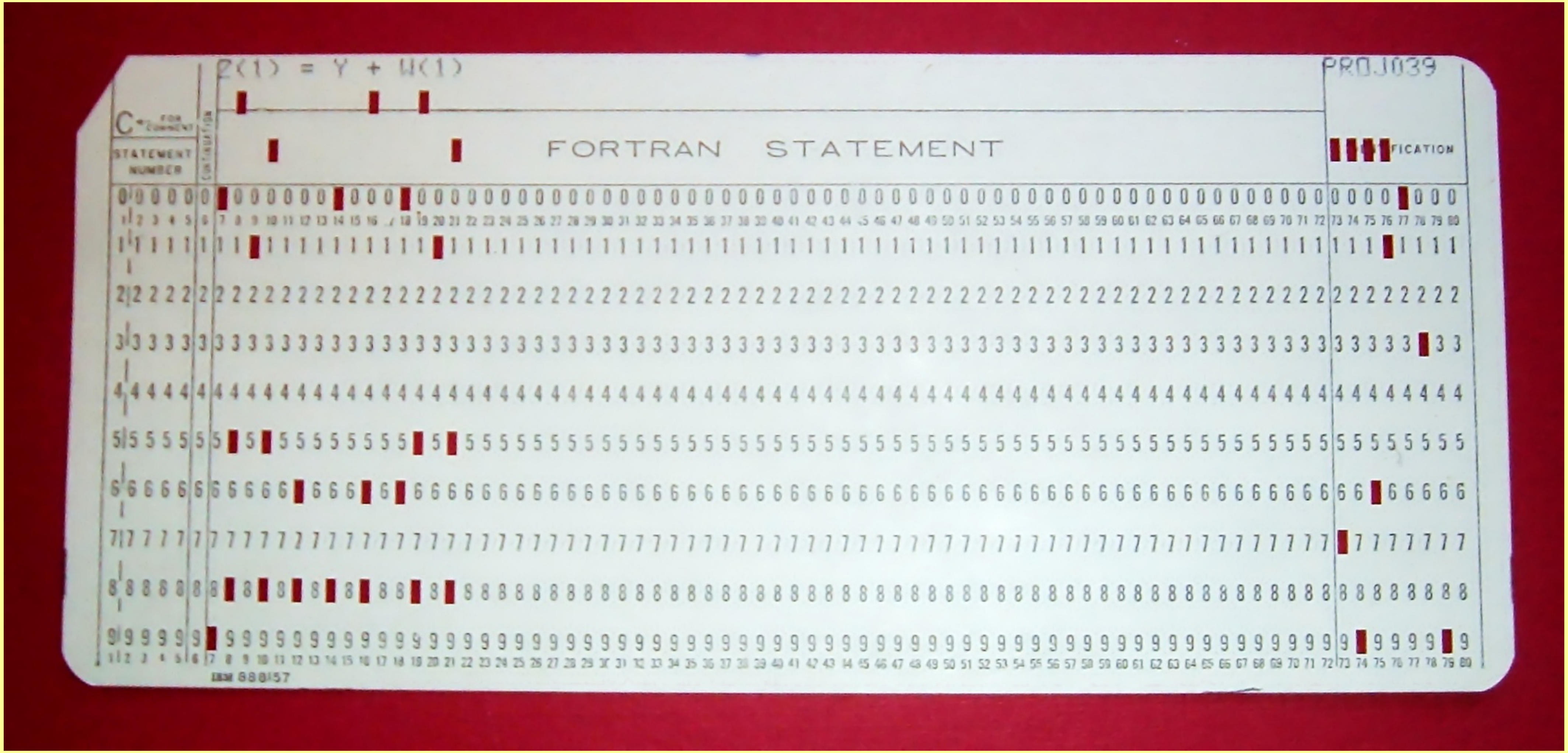
- Jobs were staged in and out via IBM 1403
  - Interesting machine in its own right
    - Variable word length, delimited by word-mark
    - Had single most powerful instruction I know of:
      - Read, write, punch, and branch
        - Read card from card reader into fixed buffer
        - Write line to line printer from another fixed buffer
        - Punch card from a third fixed buffer
        - Branch for next instruction

The IBM 709x Machines, IV

- Historical note: the Lisp language was developed on the IBM 709x
  - A list element was stored in two fields of a single word
    - Machine had instructions to load and store the individual fields of a word
    - Hence the Lisp names:
      - `car`: Contents of Address Register
      - `cdr`: Contents of Decrement Register
- Installed at NYU, Columbia, Stanford, NASA, ...
  - The first machine I ever programmed was the 7090 at NASA
    - Used for Martin Karplus' *ab initio* molecular wave function calculations for H_3
      - The most complicated molecule that could be handled at the time!
      - Karplus was awarded the Nobel Prize in Chemistry in 2013 for this work

The four extra index registers on the 7094 not on the 7090





Alphanumeric punched card - 1 character/column
Columns 73-80 were used for serialization

Program Agenda

- Introduction
- The IBM 709x Series
- The CDC 6600
- The CDC 7600
- Mass Storage Technologies
- Conclusion

The CDC 6600 Machines, I

- 100 nsec. 4-phase clock, 1000 nsec. cycle time = 1 MHz.
- 128K 60-bit words of memory; 2 Megawords of ECS (Extended Core Storage)
- 4 parcels/word; 1 and 2 parcel instructions; 1, 2, 3, or 4 instructions/word
  - Nine functional units, controlled by a scoreboard
    - Functional parallelism: allowed overlapping arithmetic operations
  - 8 word instruction stack (i-cache)
- 8 60-bit X registers, 8 18-bit A registers, 8 18-bit B registers
  - Setting A1-A5 loaded X1-X5 from the address in the A register
  - Setting A6, A7 stored from X6, X7 to the address in the A register
  - B0 was a hard-wired zero
- No paging or segmentation: base/bounds registers

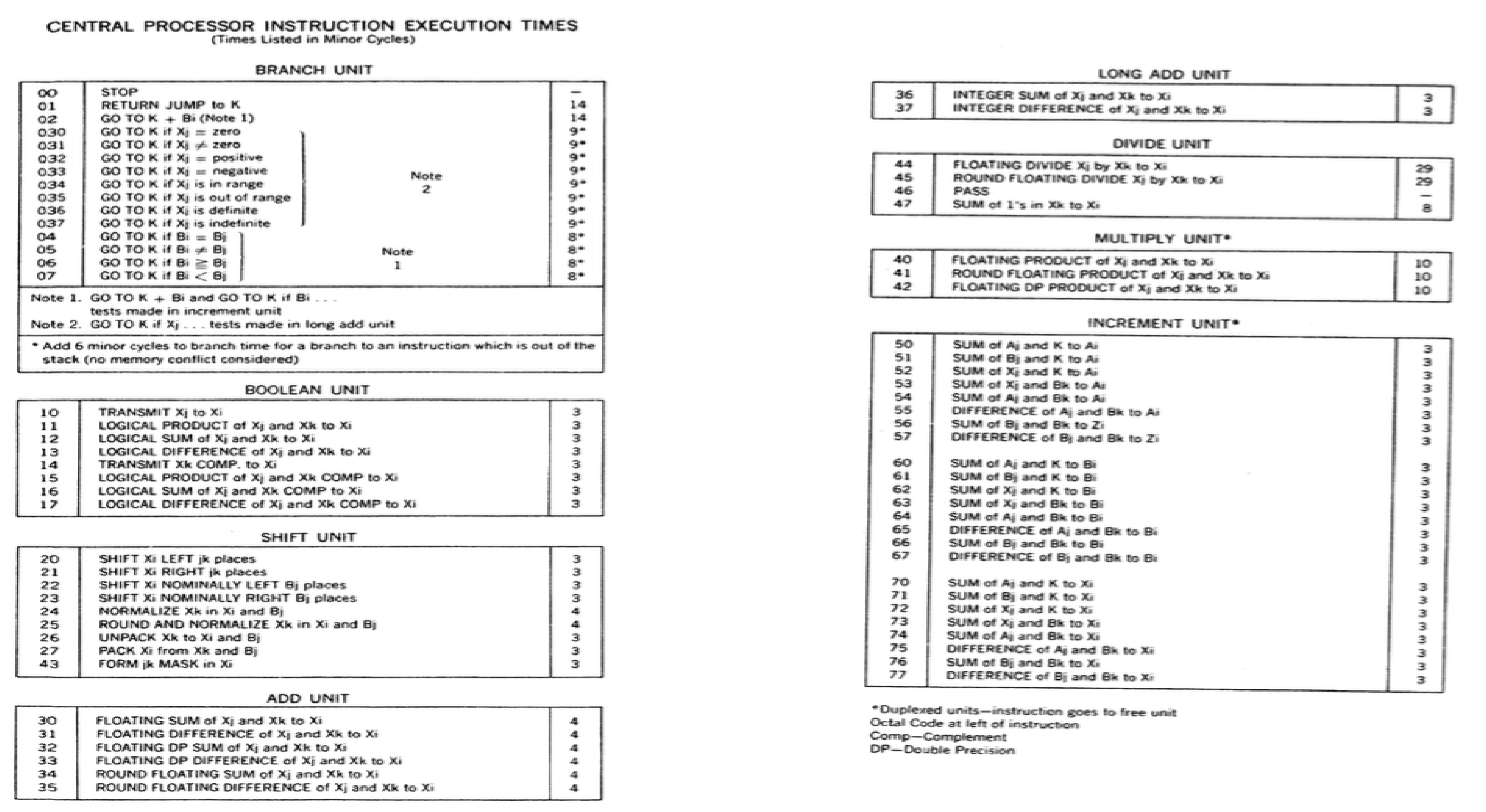
The CDC 6600 Machines, II

- 10 (later 20) Peripheral Processors (PPUs), each with 4K 12-bit words
  - 5 PPU words/central memory word
  - Each PPU could read and write anywhere in memory
  - PPUs did not run in parallel; each got a slot in the “barrel” to execute
  - PPUs could read the PC of the CPU
  - PPUs handled all I/O
- OS ran in the PPUs
  - All communication done by polling memory
  - One PPU was the monitor
    - CPU system call made by storing a command into a specific word
    - Forced CPU context switches
  - One PPU drove the operator's console
  - Other PPUs controlled I/O devices, dispatched by monitor

The CDC 6600 Machines, III

- Fastest computer in the world from 1964 to 1969 (until the 7600)
- Used for one of the first ever profilers: SPY
  - Dedicated PPU monitored the PC of the CPU and produced a histogram
  - Developed by the late Dave Stevens
    - A manager who did real work!
- Machines installed at LBL, LLL, Cal, NYU, ...

CDC 6600 Instruction Set



CENTRAL PROCESSOR INSTRUCTION EXECUTION TIMES

(Times Listed in Minor Cycles)

BRANCH UNIT

00	STOP		—
01	RETURN JUMP to K		14
02	GO TO K + Bi (Note 1)		14
030	GO TO K if Xj = zero	Note 2	9*
031	GO TO K if Xj ≠ zero		9*
032	GO TO K if Xj = positive		9*
033	GO TO K if Xj = negative		9*
034	GO TO K if Xj is in range		9*
035	GO TO K if Xj is out of range		9*
036	GO TO K if Xj is definite		9*
037	GO TO K if Xj is indefinite		9*
04	GO TO K if Bi = Bj	Note 1	8*
05	GO TO K if Bi ≠ Bj		8*
06	GO TO K if Bi ≧ Bj		8*
07	GO TO K if Bi < Bj		8*

Note 1. GO TO K + Bi and GO TO K if Bi . . .
tests made in increment unit
Note 2. GO TO K if Xj . . . tests made in long add unit

* Add 6 minor cycles to branch time for a branch to an instruction which is out of the stack (no memory conflict considered)

BOOLEAN UNIT

10	TRANSMIT Xj to Xi	3
11	LOGICAL PRODUCT of Xj and Xk to Xi	3
12	LOGICAL SUM of Xj and Xk to Xi	3
13	LOGICAL DIFFERENCE of Xj and Xk to Xi	3
14	TRANSMIT Xk COMP. to Xi	3
15	LOGICAL PRODUCT of Xj and Xk COMP to Xi	3
16	LOGICAL SUM of Xj and Xk COMP to Xi	3
17	LOGICAL DIFFERENCE of Xj and Xk COMP to Xi	3

SHIFT UNIT

20	SHIFT Xi LEFT jk places	3
21	SHIFT Xi RIGHT jk places	3
22	SHIFT Xi NOMINALLY LEFT Bj places	3
23	SHIFT Xi NOMINALLY RIGHT Bj places	3
24	NORMALIZE Xk in Xi and Bj	4
25	ROUND AND NORMALIZE Xk in Xi and Bj	4
26	UNPACK Xk to Xi and Bj	3
27	PACK Xi from Xk and Bj	3
43	FORM jk MASK in Xi	3

ADD UNIT

30	FLOATING SUM of Xj and Xk to Xi	4
31	FLOATING DIFFERENCE of Xj and Xk to Xi	4
32	FLOATING DP SUM of Xj and Xk to Xi	4
33	FLOATING DP DIFFERENCE of Xj and Xk to Xi	4
34	ROUND FLOATING SUM of Xj and Xk to Xi	4
35	ROUND FLOATING DIFFERENCE of Xj and Xk to Xi	4

LONG ADD UNIT

36	INTEGER SUM of Xj and Xk to Xi	3
37	INTEGER DIFFERENCE of Xj and Xk to Xi	3

DIVIDE UNIT

44	FLOATING DIVIDE Xj by Xk to Xi	29
45	ROUND FLOATING DIVIDE Xj by Xk to Xi	29
46	PASS	—
47	SUM of 1's in Xk to Xi	8

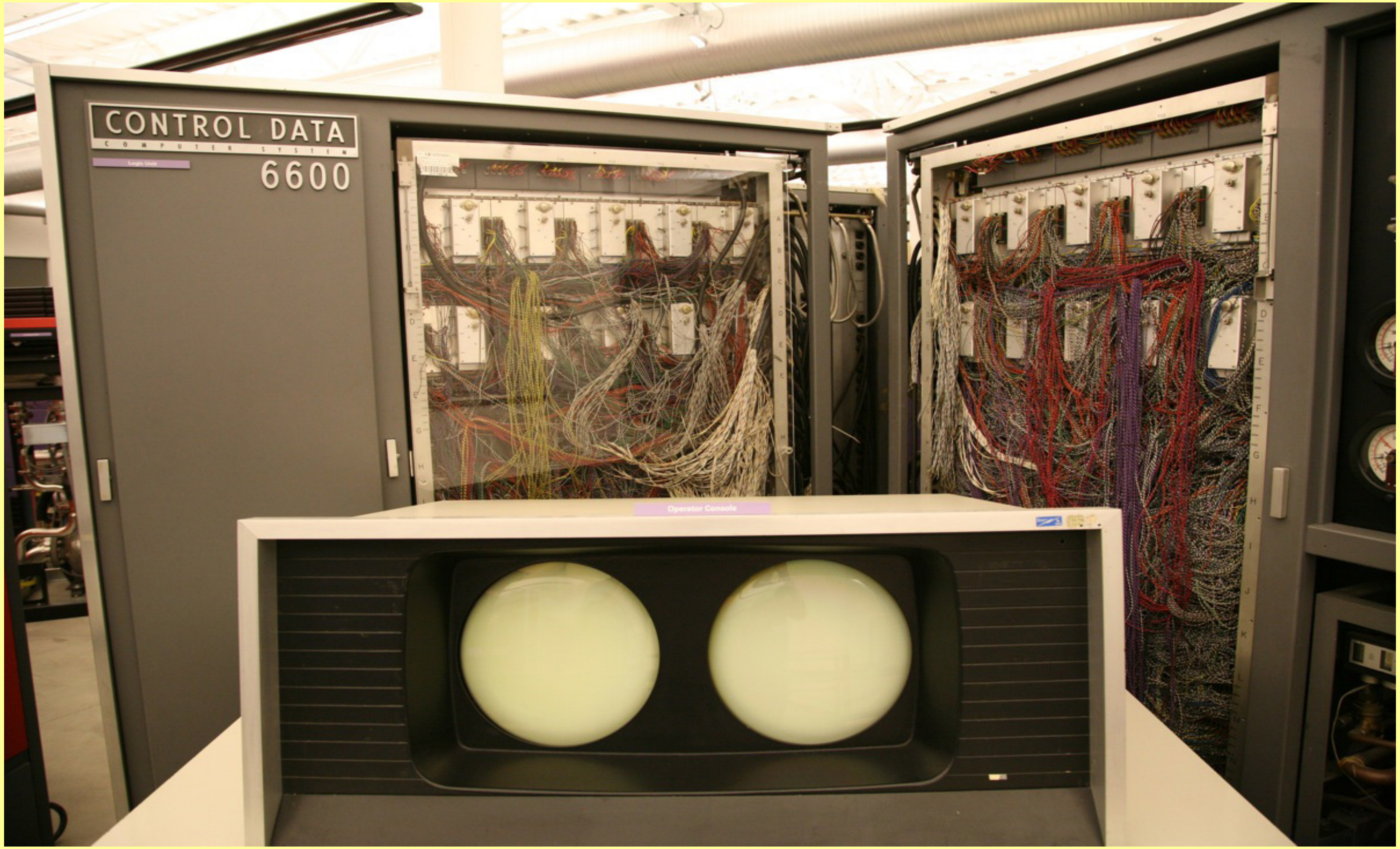
MULTIPLY UNIT*

40	FLOATING PRODUCT of Xj and Xk to Xi	10
41	ROUND FLOATING PRODUCT of Xj and Xk to Xi	10
42	FLOATING DP PRODUCT of Xj and Xk to Xi	10

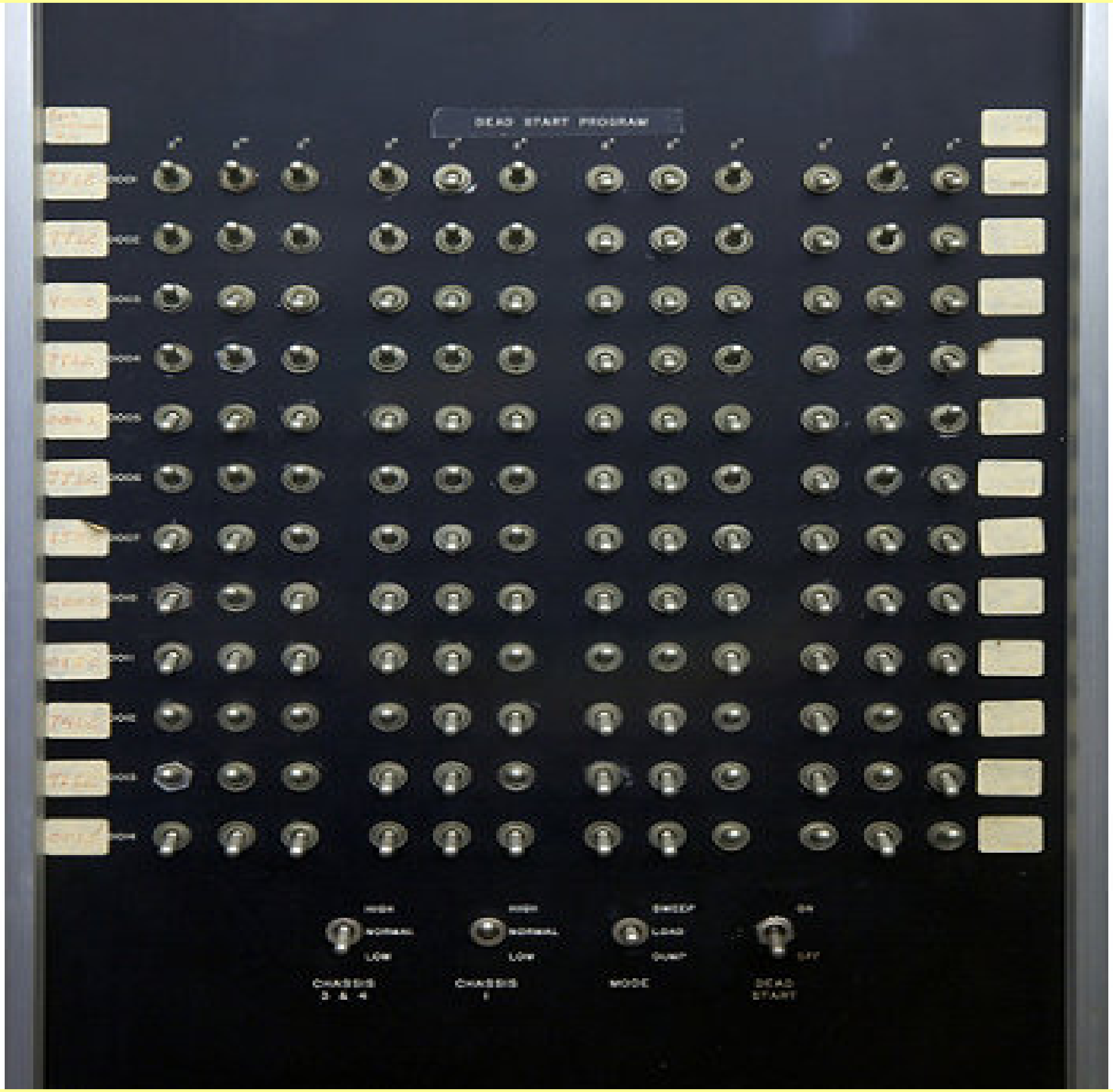
INCREMENT UNIT*

50	SUM of Aj and K to Ai	3
51	SUM of Bj and K to Ai	3
52	SUM of Xj and K to Ai	3
53	SUM of Xj and Bk to Ai	3
54	SUM of Aj and Bk to Ai	3
55	DIFFERENCE of Aj and Bk to Ai	3
56	SUM of Bj and Bk to Zi	3
57	DIFFERENCE of Bj and Bk to Zi	3
60	SUM of Aj and K to Bi	3
61	SUM of Bj and K to Bi	3
62	SUM of Xj and K to Bi	3
63	SUM of Xj and Bk to Bi	3
64	SUM of Aj and Bk to Bi	3
65	DIFFERENCE of Aj and Bk to Bi	3
66	SUM of Bj and Bk to Bi	3
67	DIFFERENCE of Bj and Bk to Bi	3
70	SUM of Aj and K to Xi	3
71	SUM of Bj and K to Xi	3
72	SUM of Xj and K to Xi	3
73	SUM of Xj and Bk to Xi	3
74	SUM of Aj and Bk to Xi	3
75	DIFFERENCE of Aj and Bk to Xi	3
76	SUM of Bj and Bk to Xi	3
77	DIFFERENCE of Bj and Bk to Xi	3

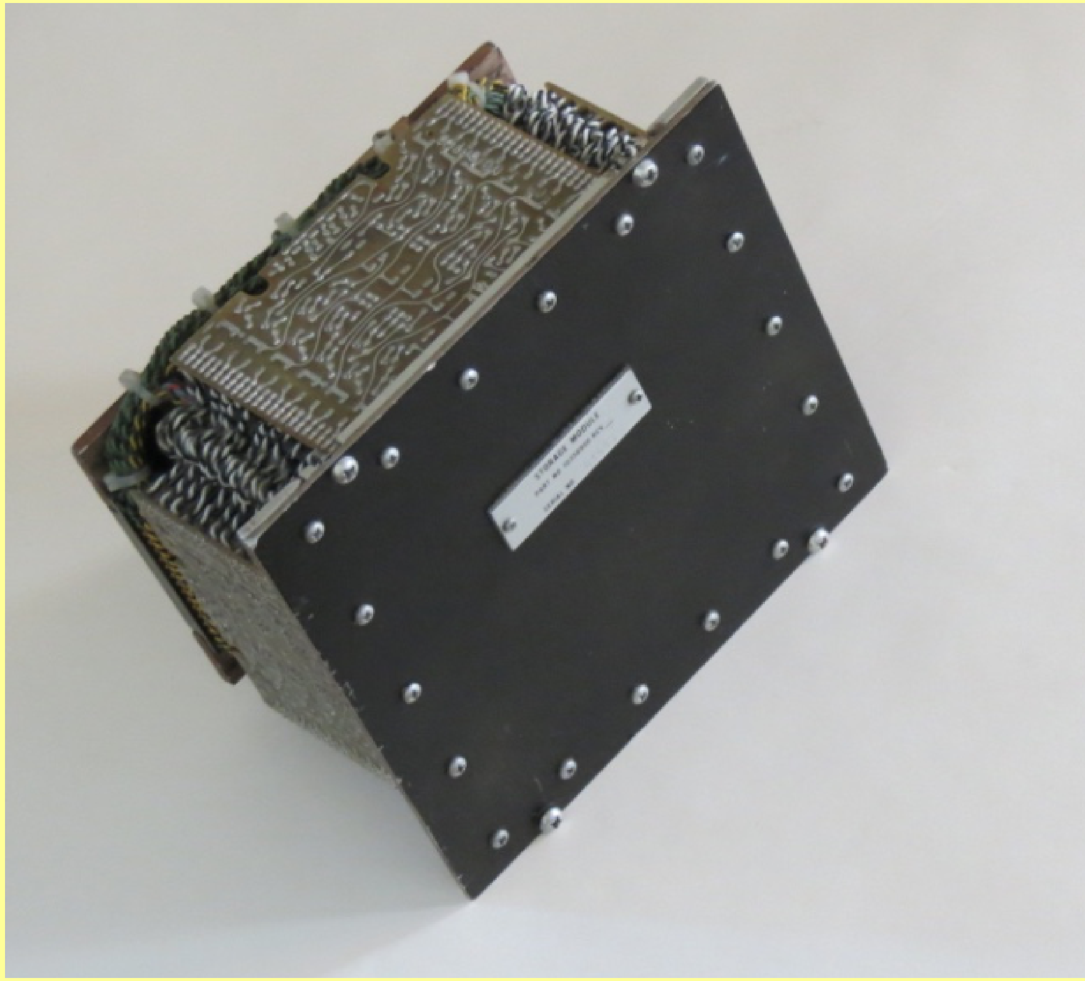
*Duplexed units—instruction goes to free unit
Octal Code at left of instruction
Comp—Complement
DP—Double Precision



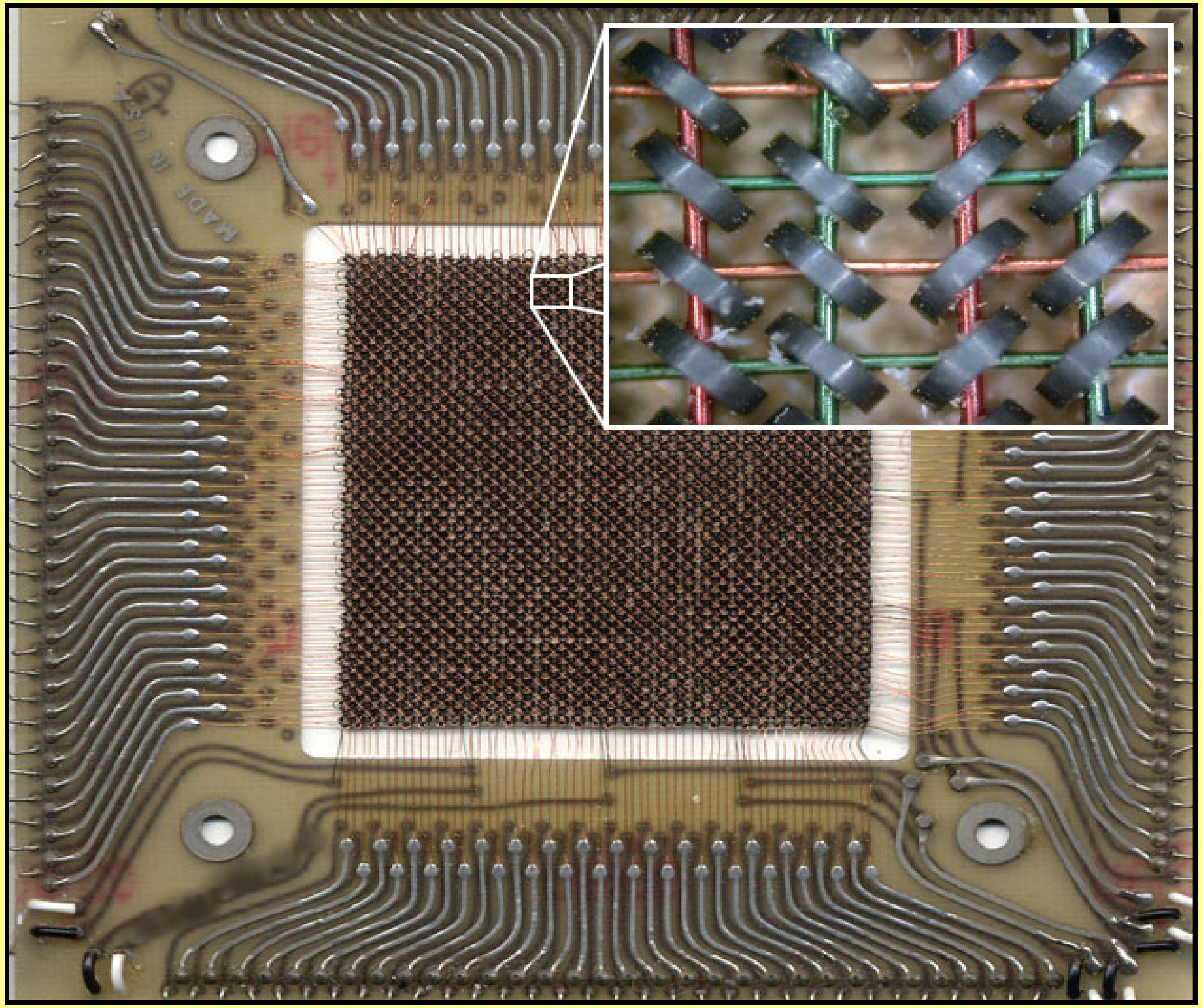
The CDC 6600 Operator's Console
In 1970, LBL had ~ 50 operators, running 24/7



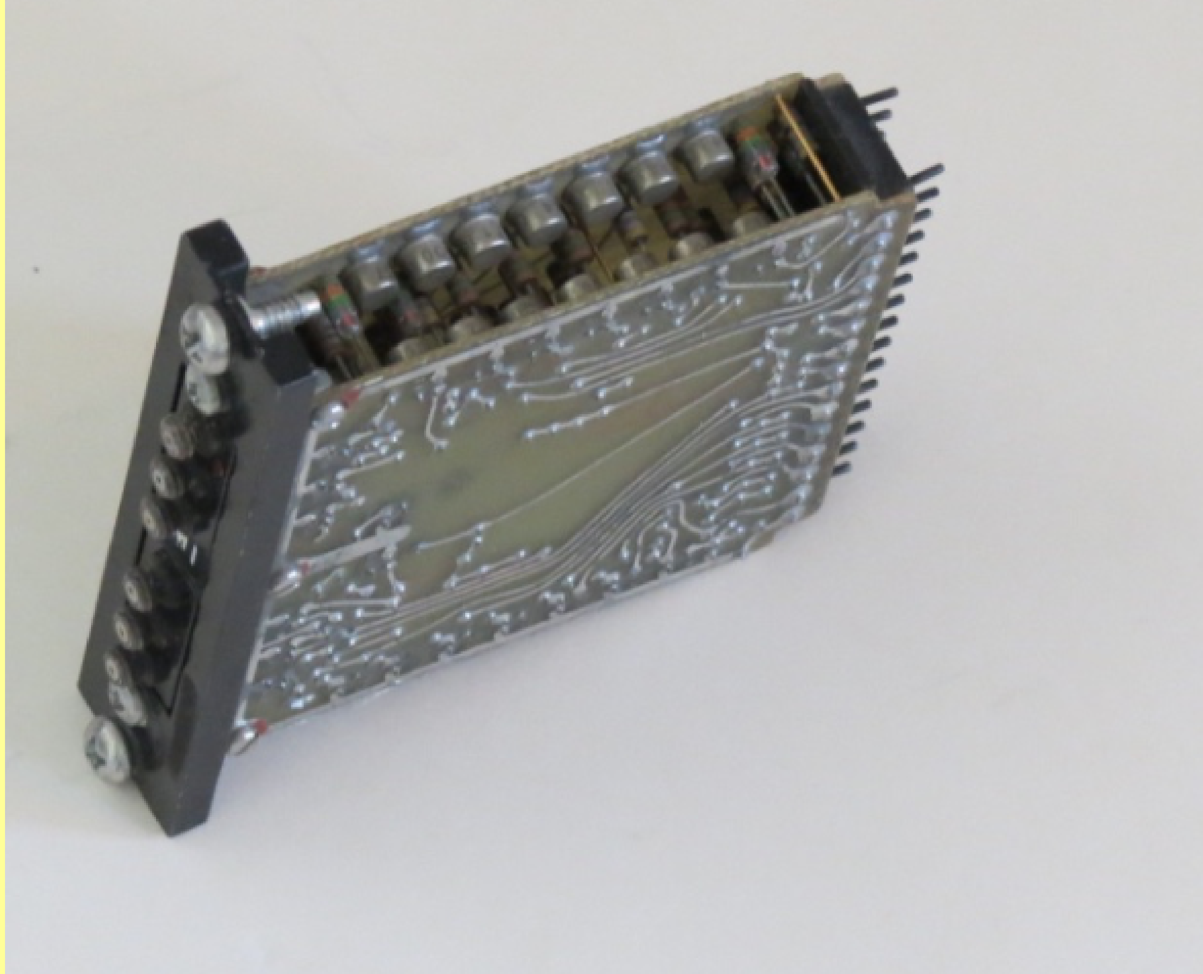
The CDC 6600 Dead-start panel: 12 12-bit instructions



CDC 6600 Memory Module
4K 12-bit words = 6 KB
Weight ~5 pounds



CDC 6600 Memory Module
Ferrite Cores



CDC 6600 “MI” module

I/O Devices

- 6603 Disks
  - Fourteen ~ 3' dia. platters: 12 data, 1 clock, 1 spare ==> 55 MB. total
  - Each 12-bit PPU word was distributed across all twelve data platters
  - Hydraulically operated; floor would shake when heads moved; kitty litter under drive
- Pneumatic card reader; card punch; Line printers
  - Lawrence Livermore Labs had a printer that printed at 60 miles/hour
- CalComp plotters; Microfiche output
- Remote terminals
  - At long last!
  - First 300 baud, then 1200, then 9600 (seemed like heaven!)

6603 Disk

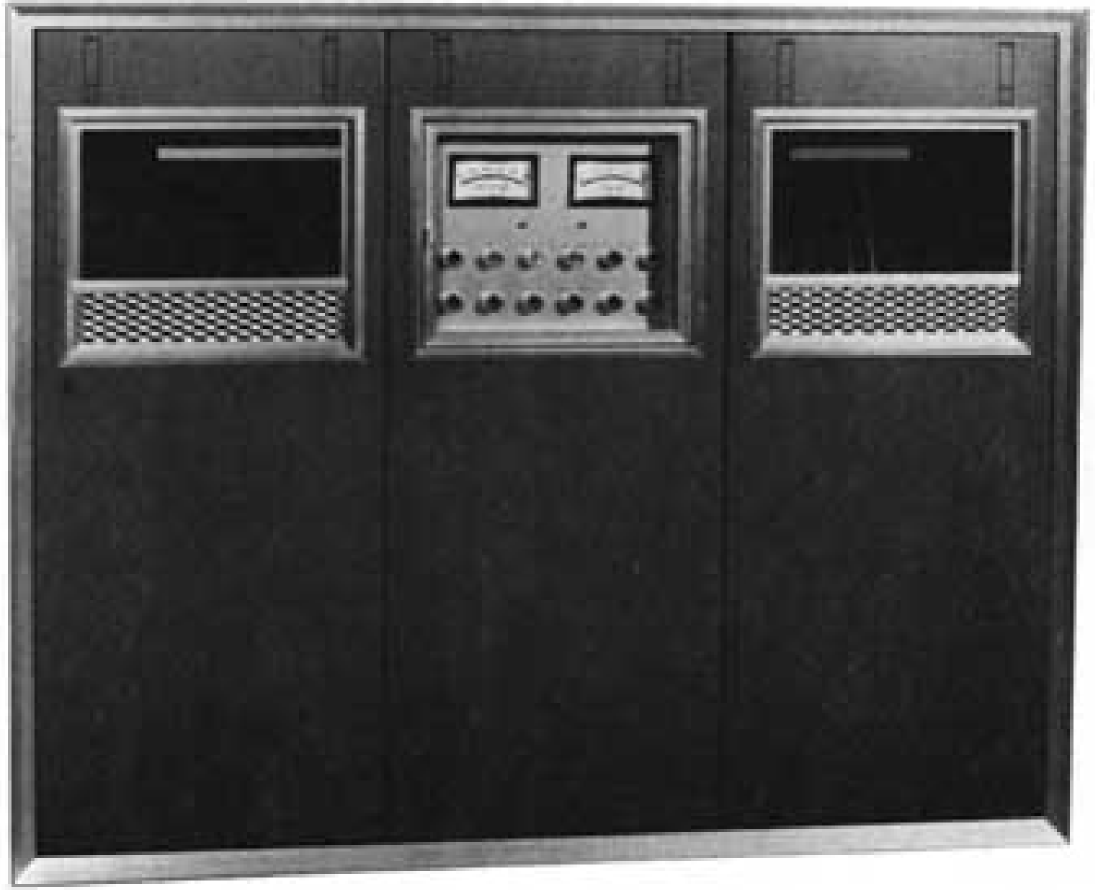
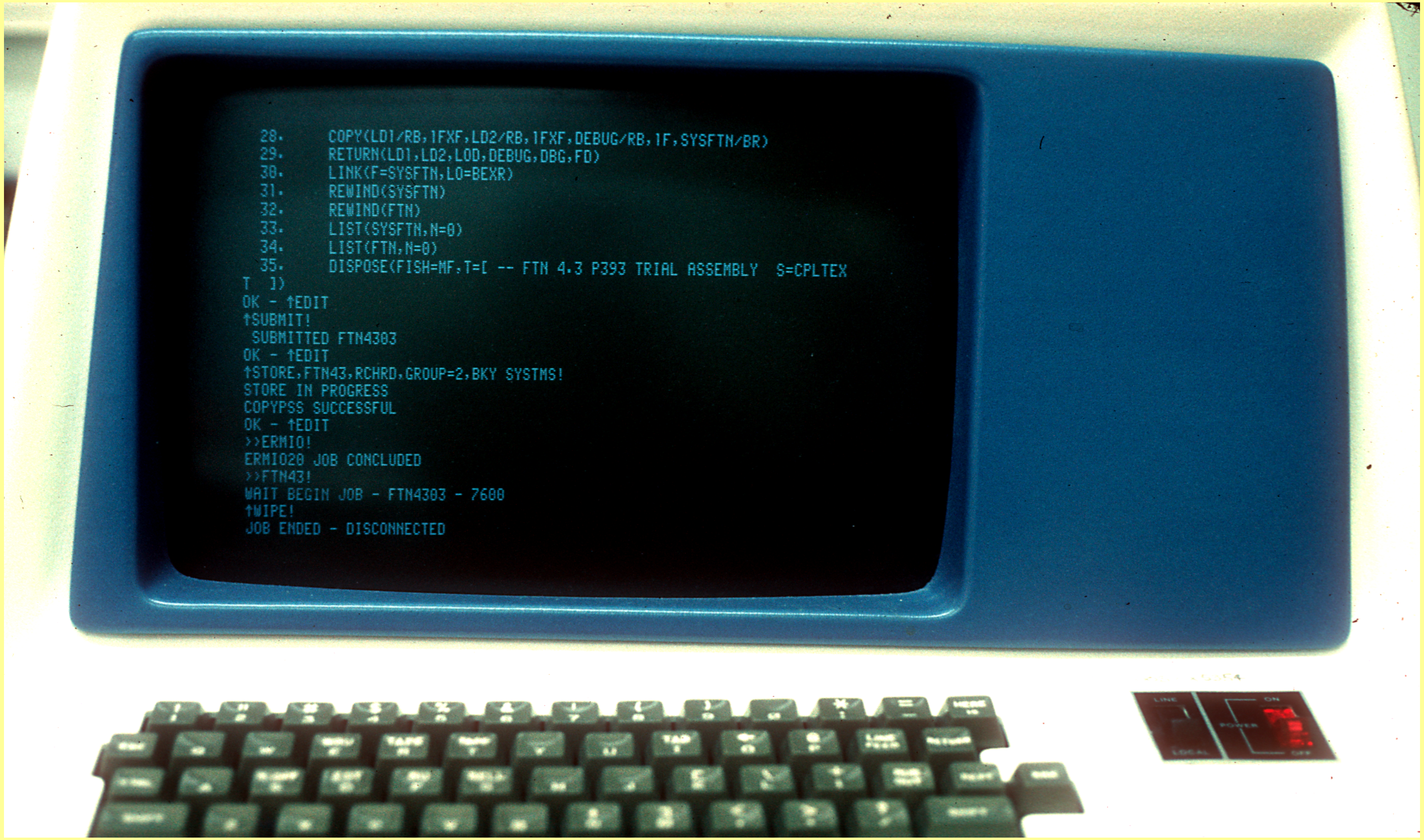
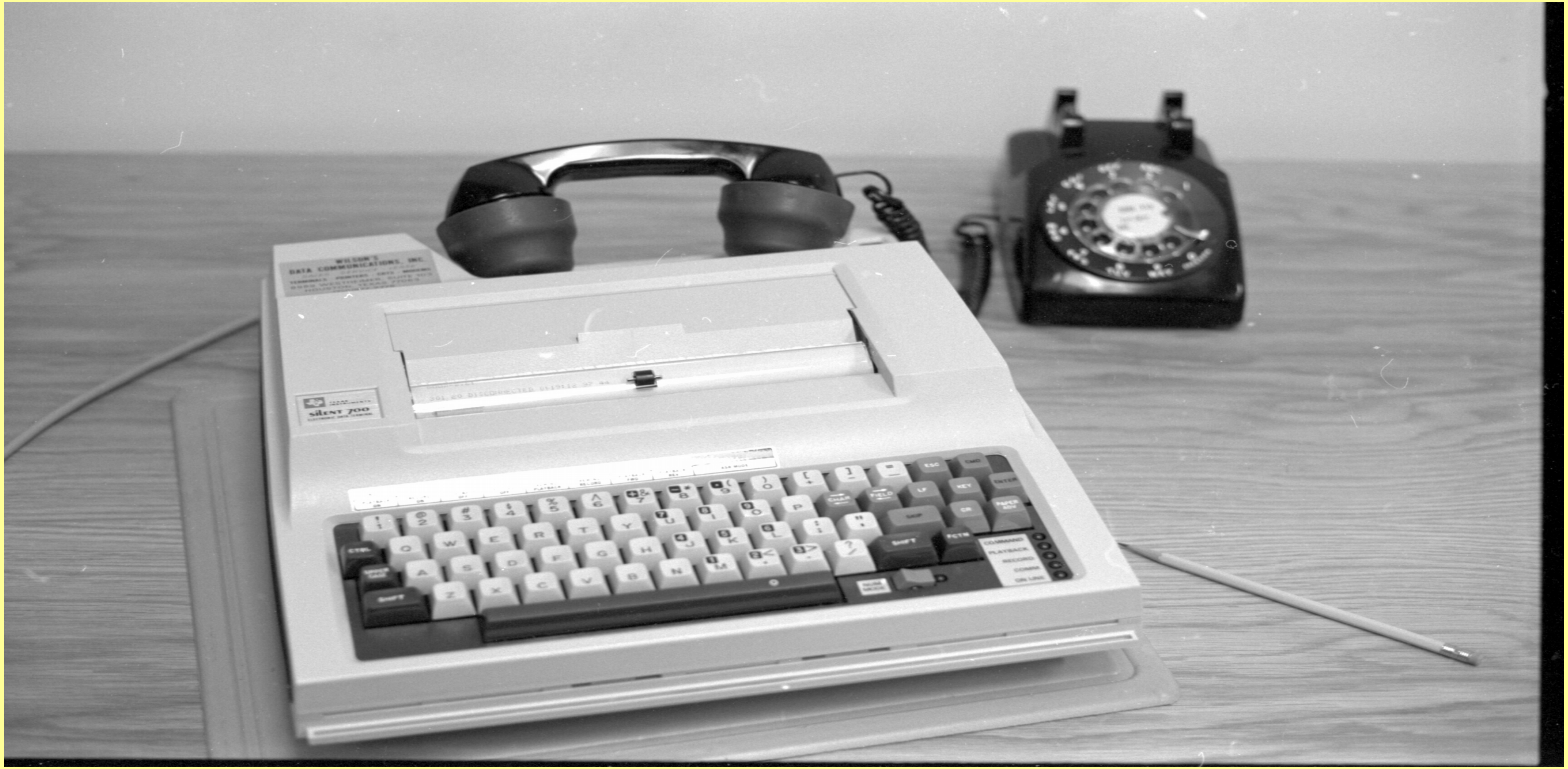


Figure 18. 6603 Disk File



```
  28.     COPY(LD1/RB,1FXF,LD2/RB,1FXF,DEBUG/RB,1F,SYSFTN/BR)
  29.     RETURN(LD1,LD2,LOD,DEBUG,DBG,FD)
  30.     LINK(F=SYSFTN,LO=BEXR)
  31.     REWIND(SYSFTN)
  32.     REWIND(FTN)
  33.     LIST(SYSFTN,N=0)
  34.     LIST(FTN,N=0)
  35.     DISPOSE(FISH=MF,T=[ -- FTN 4.3 P393 TRIAL ASSEMBLY  S=CPLTEX
T  ])
OK - ↑EDIT
↑SUBMIT!
 SUBMITTED FTN4303
OK - ↑EDIT
↑STORE,FTN43,RCHRD,GROUP=2,BKY SYSTMS!
STORE IN PROGRESS
COPYPSS SUCCESSFUL
OK - ↑EDIT
>>ERMIO!
ERMI020 JOB CONCLUDED
>>FTN43!
WAIT BEGIN JOB - FTN4303 - 7600
↑WIPE!
JOB ENDED - DISCONNECTED
```

ADM3a Remote Terminal



Remote Computing -- 300 baud modem at first
then 1200 baud, and then 9600 baud

Program Agenda

- Introduction
- The IBM 709x Series
- The CDC 6600
- The CDC 7600
- Mass Storage Technologies
- Conclusion

The CDC 7600 Machines, I

- 27.5 ns. clock = 36.36 MHz.
- CPU compatible with the 6600
- 15 PPUs
- 65 KWord "Small Core Memory"; 512 KWord "Large Core Memory"
  - Only SCM could be used for instruction execution
  - Register load/stores can go to SCM or directly to LCM
    - Also could do block copy between SCM and LCM
  - LCM was used to hold overlays, data, disk buffers
- 3360 Modules in machine; 120 miles of wiring
- Cost: $5,100,000

The CDC 7600 Machines, II

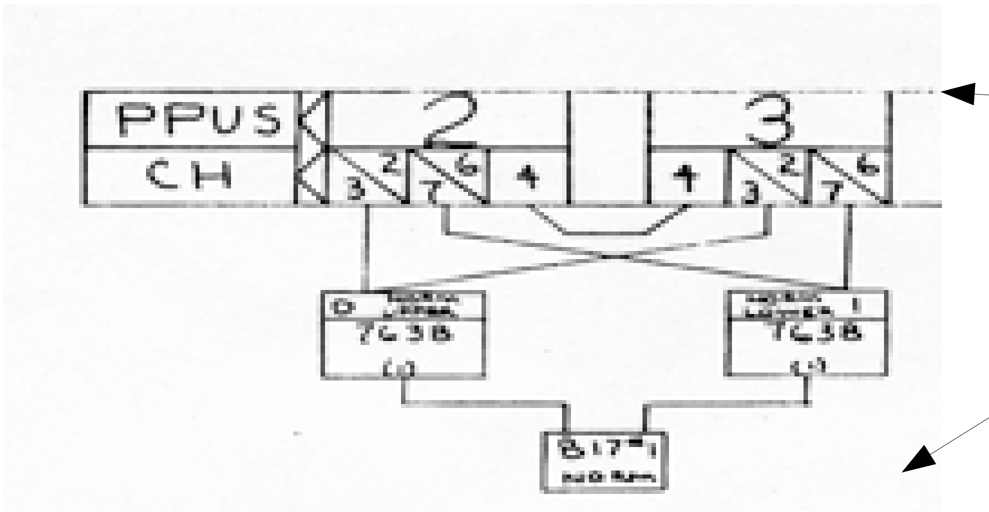
- Jobs and data files were staged in and out via a CDC 6600
- OS ran in the CPU, unlike the 6600
  - Almost every installation ran their own home-brew OS
    - LBL ran BKY, with 2X the throughput of CDC's SCOPE
- Original OS supported two jobs at a time (upper and lower jobs)
  - Three-year project to have BKY support six (could recompile for more, but not needed)
    - Utilization over 24 hours -- User CPU: 94%; OS xeq: 3%; OS moving memory: 3%; idle: 0% (!)
    - Delivered 2 days late -- because we could bring up a new OS only on a Tuesday
  - One remaining known bug: the BCD 1-second adder would fail in year 2100
    - OS would think it's a leap year, but it's not
  - Several years later, a change in files/job required recompile to change table sizes
    - Change was not understood or explained, but it was real

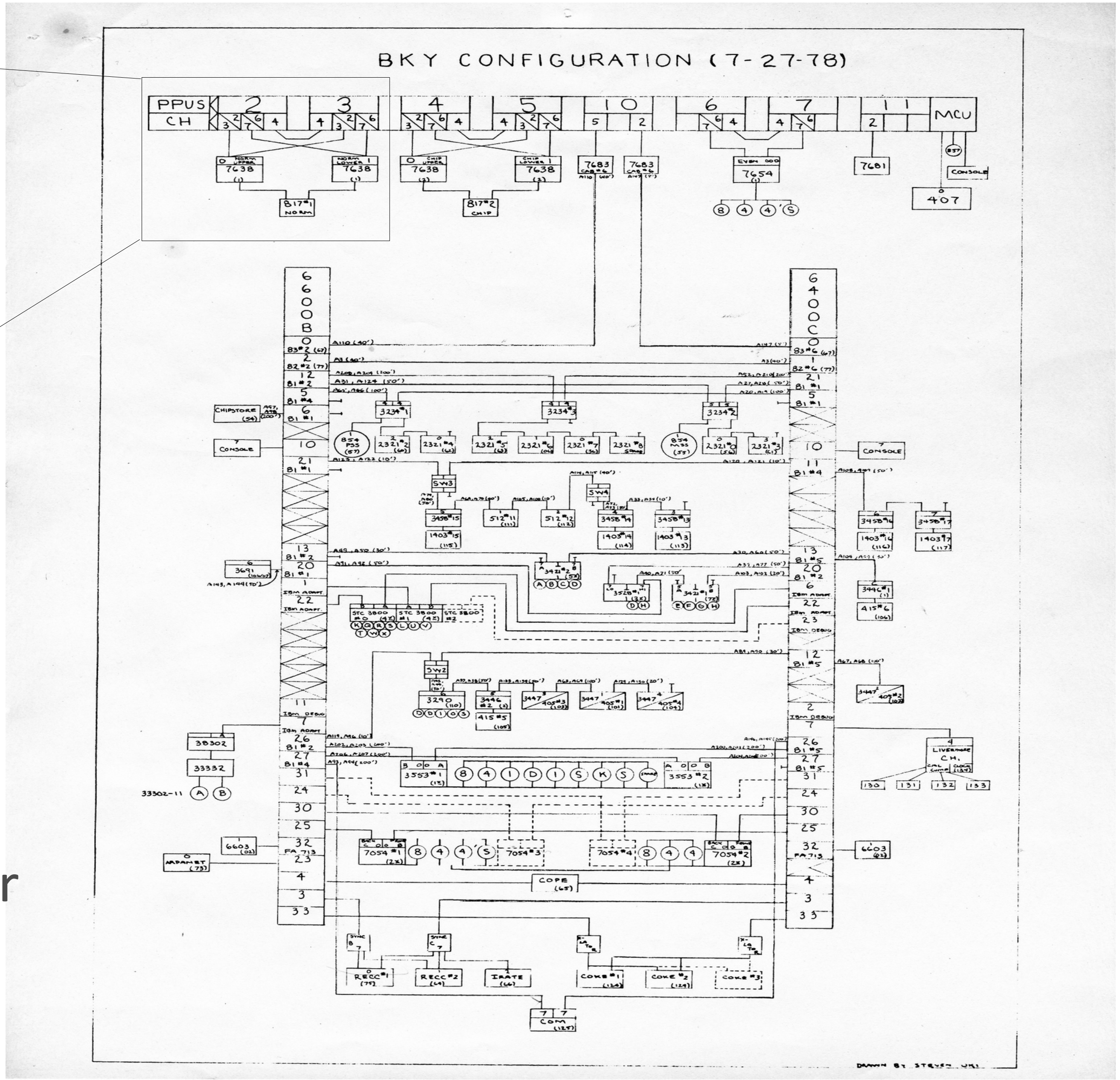
CDC 6600/7600 Development Languages

- CPU Languages
  - FORTRAN 66 with extensions for small/large memory, overlays
  - CDC FTN4 and legacy RUN compilers
  - FTN5 (Fortran 77) introduced in 1980
- CDC COMPASS 7600/6600 Assembler
  - Assembler had advanced macro programming capabilities
- PAS – Peripheral Processor Assembler
- Various math libraries from NYU, CERN and others
- UPDATE – CDC source code maintainer

CDC 7600 OS Development

- OS entirely written in assembler; macros came later
- The machine cost ~$3600 per hour to rent
  - System time: Monday and Thursday 6AM-8AM, Saturday 5AM-8AM
- Much development done on a simulator
  - Could boot the OS, and run a job on it, but very slowly
  - One had to bench-check code very carefully
    - Turnaround for a compilation could be hours or overnight!
- During system time, very little time for actual debugging
  - Take a crash dump, and figure out what happened later
  - A dump was a ~2” stack of 11”X17” fan-fold paper
  - No symbolic interpretation, everything debugged by reading the octal dump
- 1970s: OS's and compilers maintained at LBL by ~15 systems programmers





7600 disk required 2 PPUs to read at full disk speed

One PPU reads sector from disk
Tells partner to read next sector
Dumps data to central memory
Waits for partner to tell it next sector

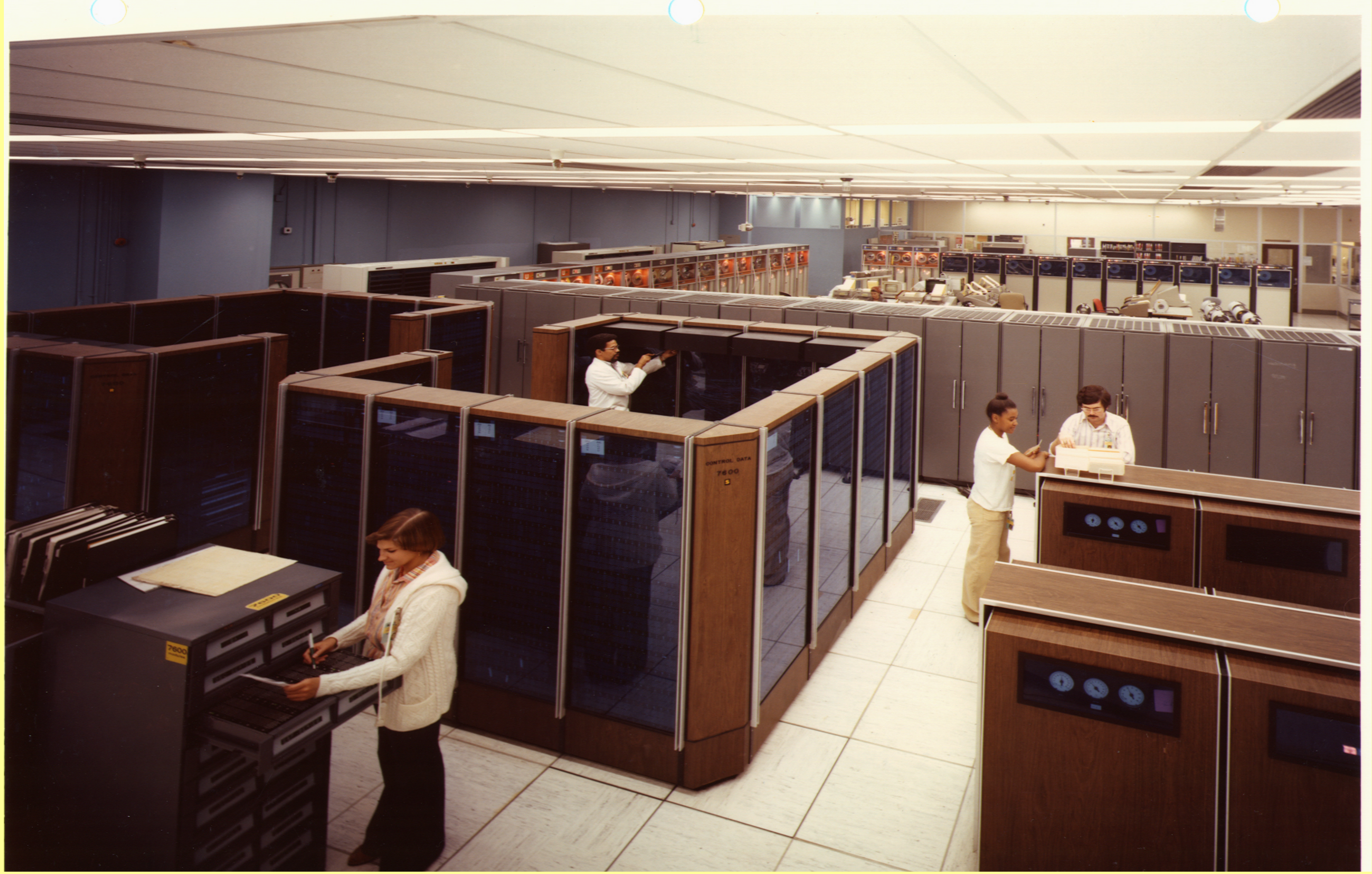
Rinse and repeat

CDC 7600 Application Development

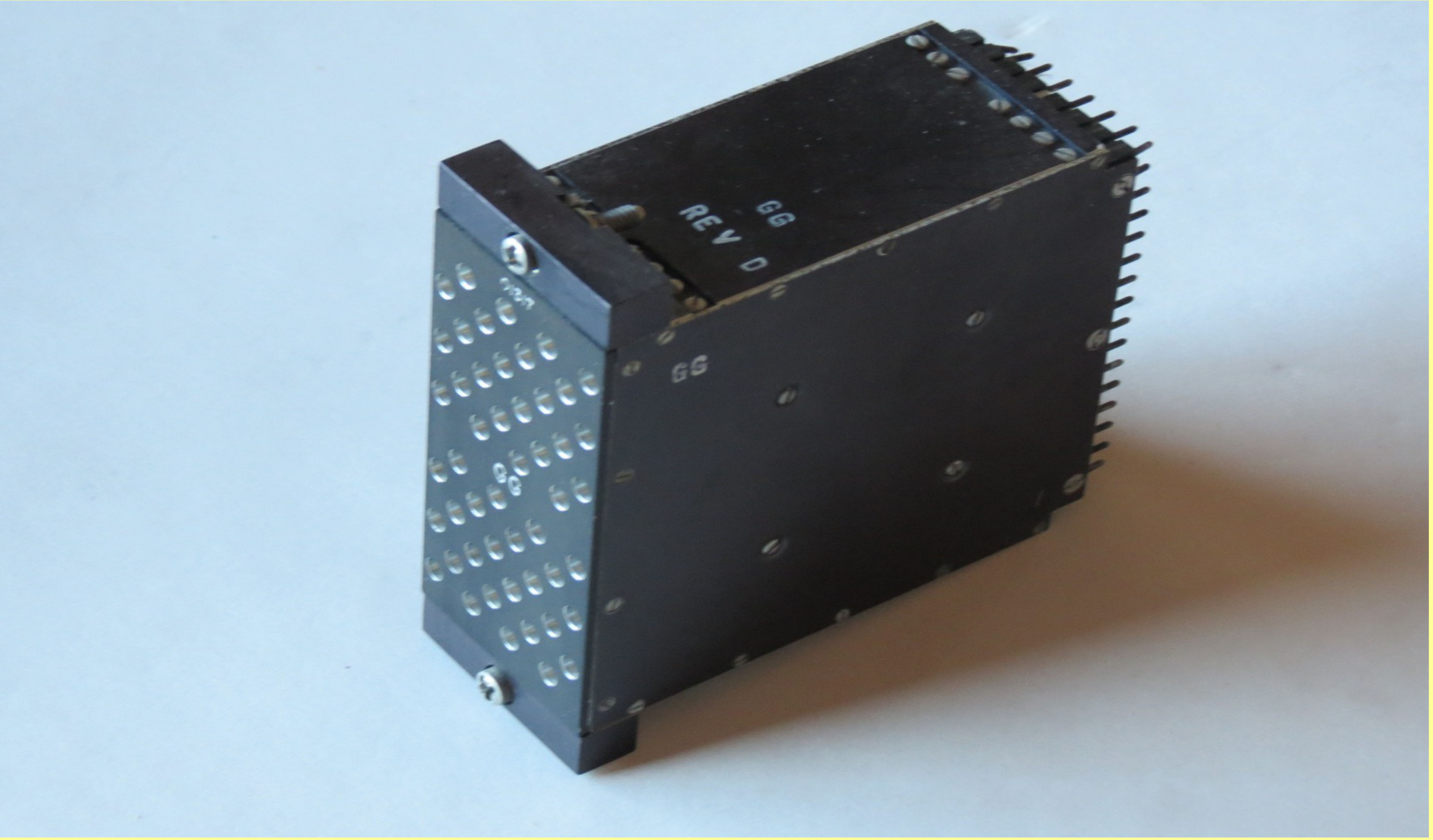
- Jobs were submitted on punched cards
  - Specified priority, time-limit, [some other parameter], account number
- Data had to be staged in and out to the 7600 disks
  - From tape, data-cell, or chipstore
- Printouts were put in an array of cubbyholes
- Debugging, as with the OS, was done by reading the octal dump
- Job status terminals to query any particular job
- Remote job submission via card reader output via printer
  - Leased phone lines connected the sites

CDC 7600 Input Job Structure

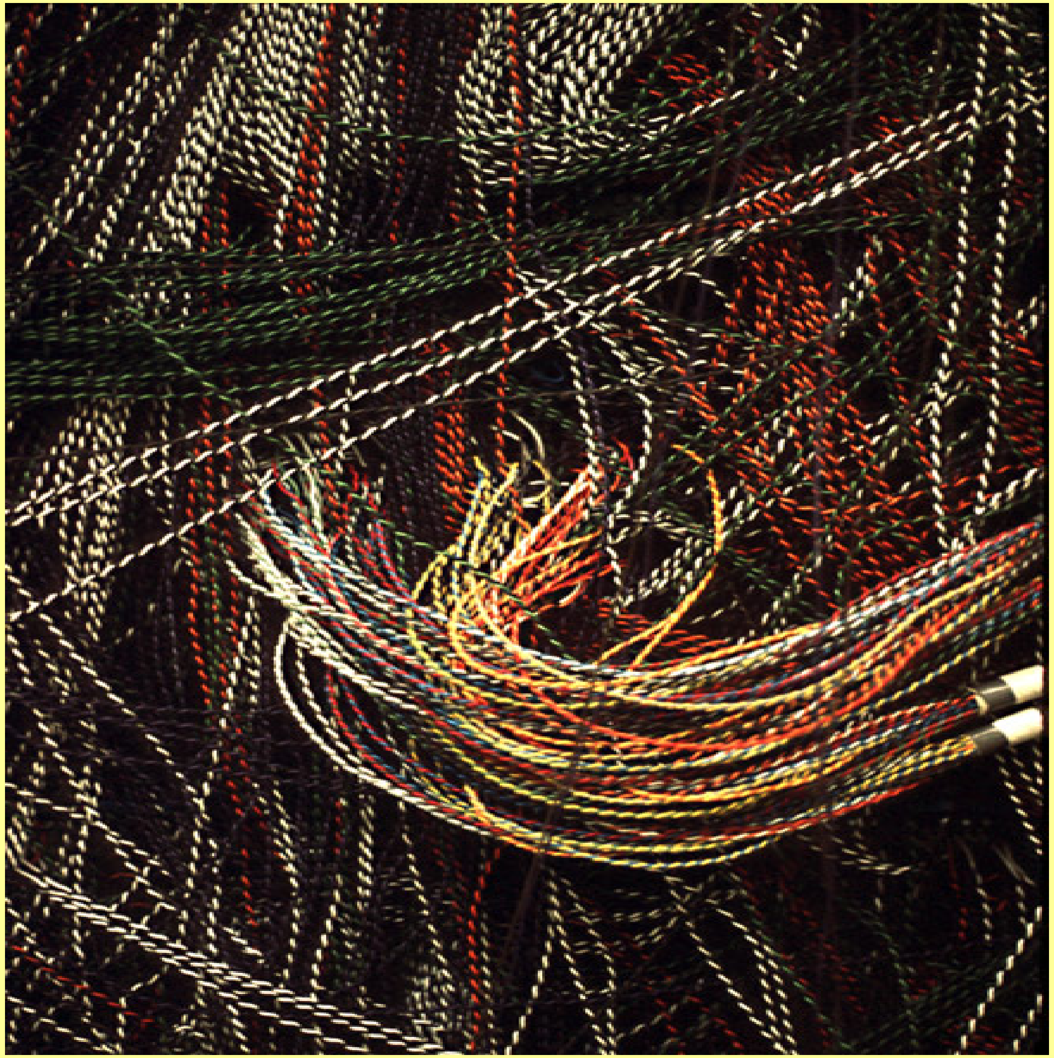
```
JOBNAME,5,100,50000.123456,JOE USER (SPECIAL DIRECTIVES)
(COMPILER CALL)
(CALL TO LOAD AND EXECUTE)
EXIT.
(ERROR PROCESSING)
7-8-9 CARD
PROGRAM MAIN (INPUT,OUTPUT)
****
7-8-9 CARD
(DATA CARDS)
6-7-8-9 CARD
```



CDC 7600 at LLNL (ca. 1970)



CDC 7600 “GG” module



CDC 7600 Backplane Wiring
Wires were color-coded by length

Comparison: CDC 7600 Machine *vs*. iPhone 5

The CDC 7600 was the world's fastest machine in its day

iPhone 5, 1.3 GHz, 1 GB Memory

- The iPhone 5 runs 36.1 times faster than the 7600
- The iPhone 5 has 227 times the memory as the 7600
- The iPhone 5 costs 0.0001 times as much as the 7600
  - Without taking inflation into account!

Program Agenda

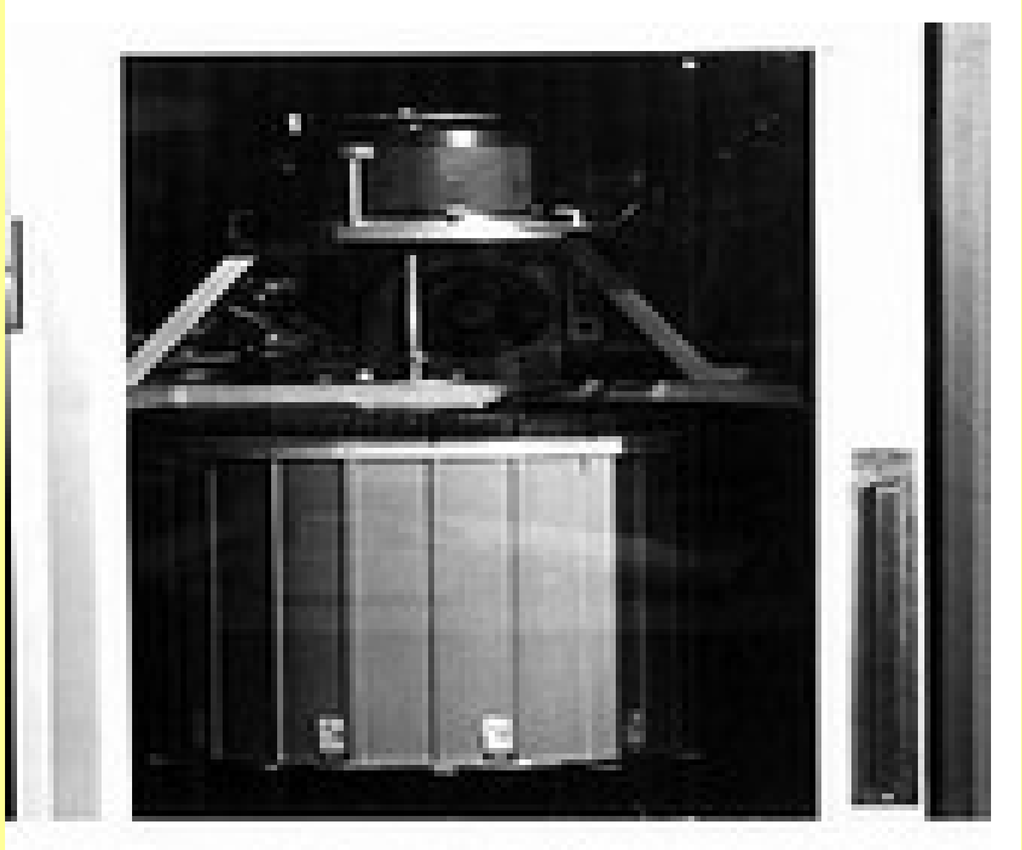
- Introduction
- The IBM 709x Series
- The CDC 6600
- The CDC 7600
- Mass Storage Technologies
- Conclusion

Mass Storage in the Late 1970's (At LBL)

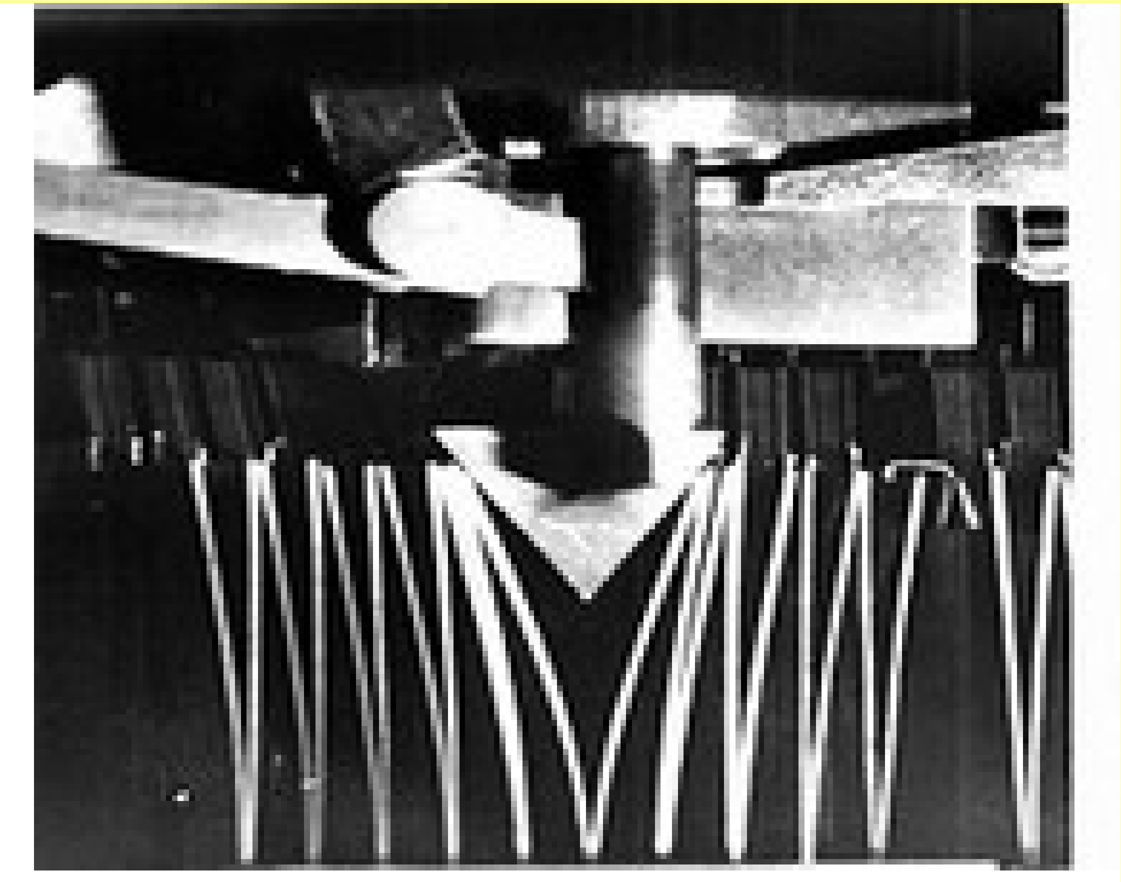
- CDC 841 and 844 multiple spindle disks: 25 to 96 million words
  - Removable spindles, but rarely used as removable
- 7-track and 9-track magnetic tape
  - Tape-hanging robot
  - LBL had many thousands of tapes assigned to users
    - When the lab started charging $1/year for tapes, over half were returned

DataCell

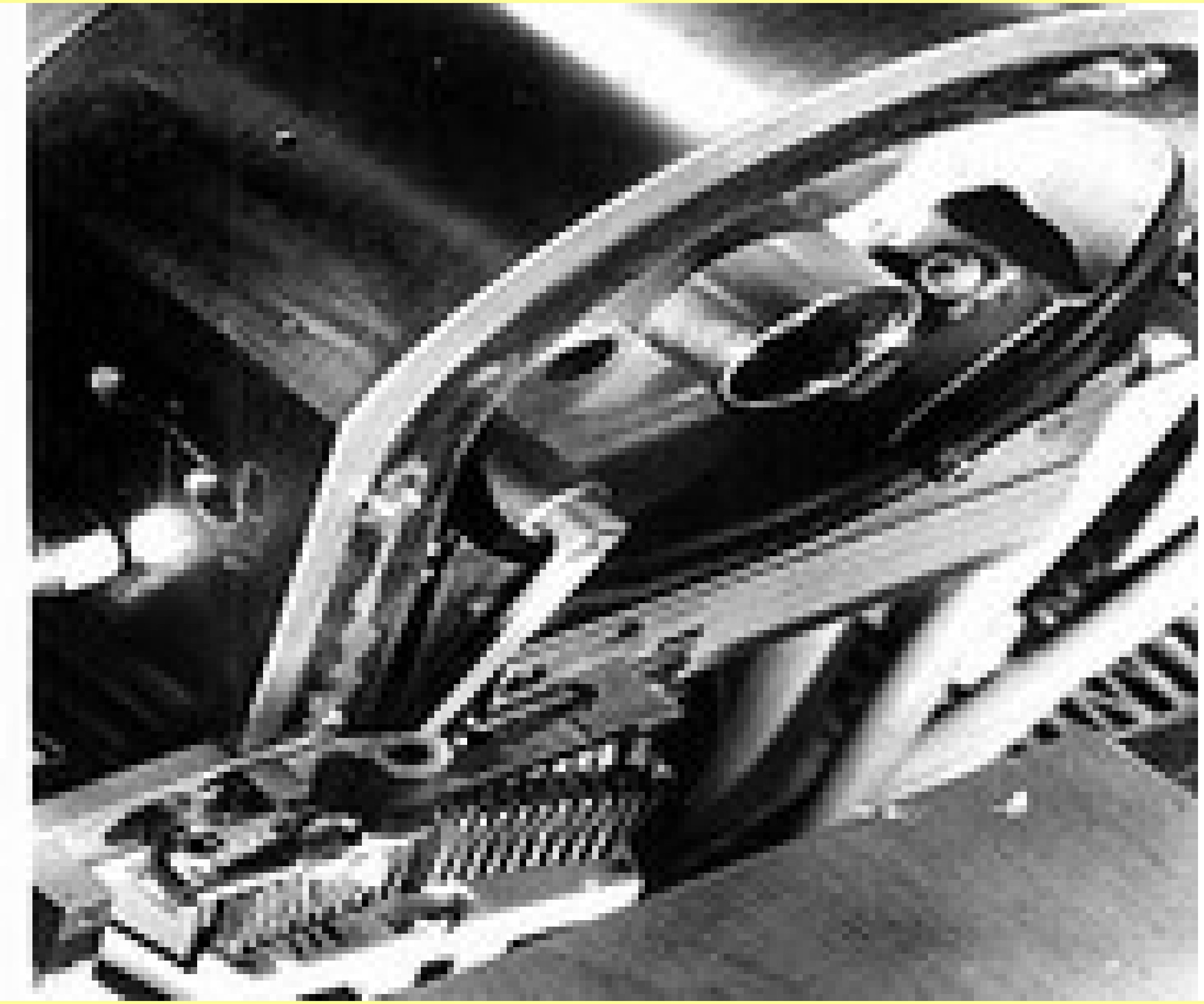
- IBM 2321 data cell (400 MB)
  - Ten wedge-shaped cells arranged on a vertical drum
  - Each cell held 200 magnetic strips -- 20 per wedge
  - Each strip held 200,000 characters (6-bit? 8-bit?)
  - Strip extracted and wound around a drum for reading and writing



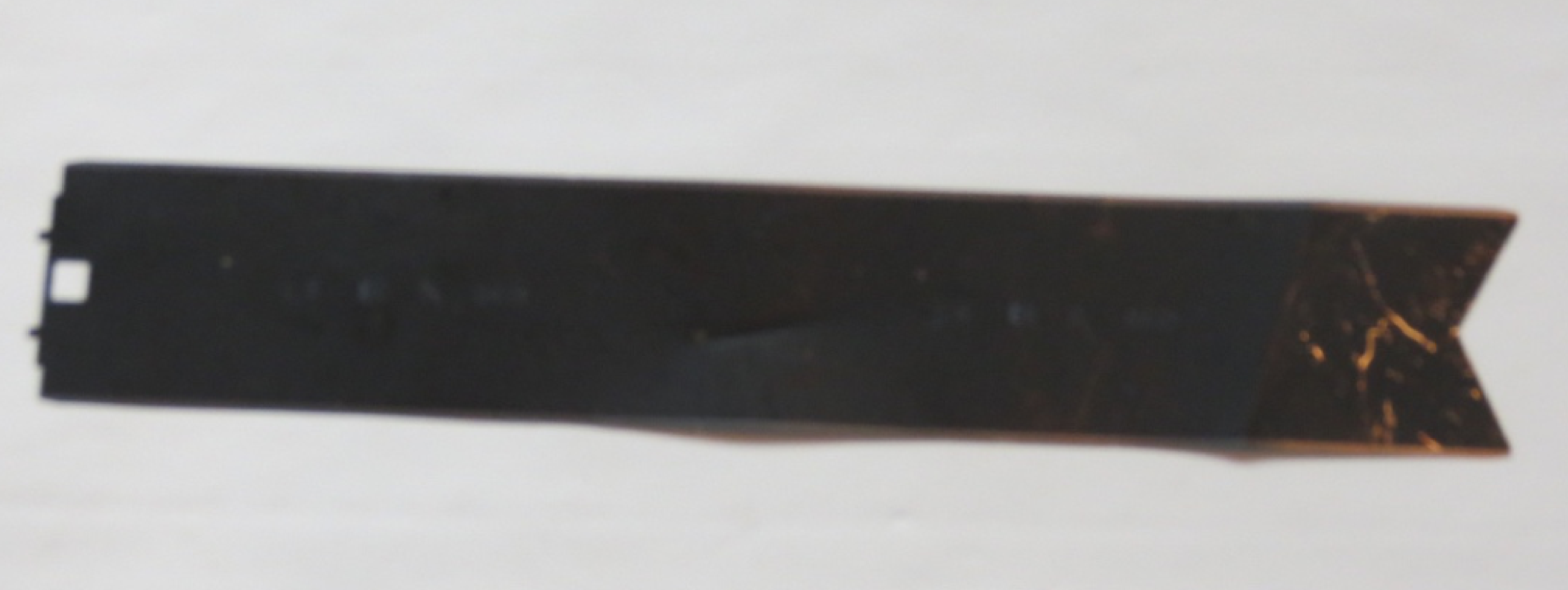
IBM Data Cell 2321
Vertically-mounted Drum



IBM Data Cell 2321
Picker Mechanism



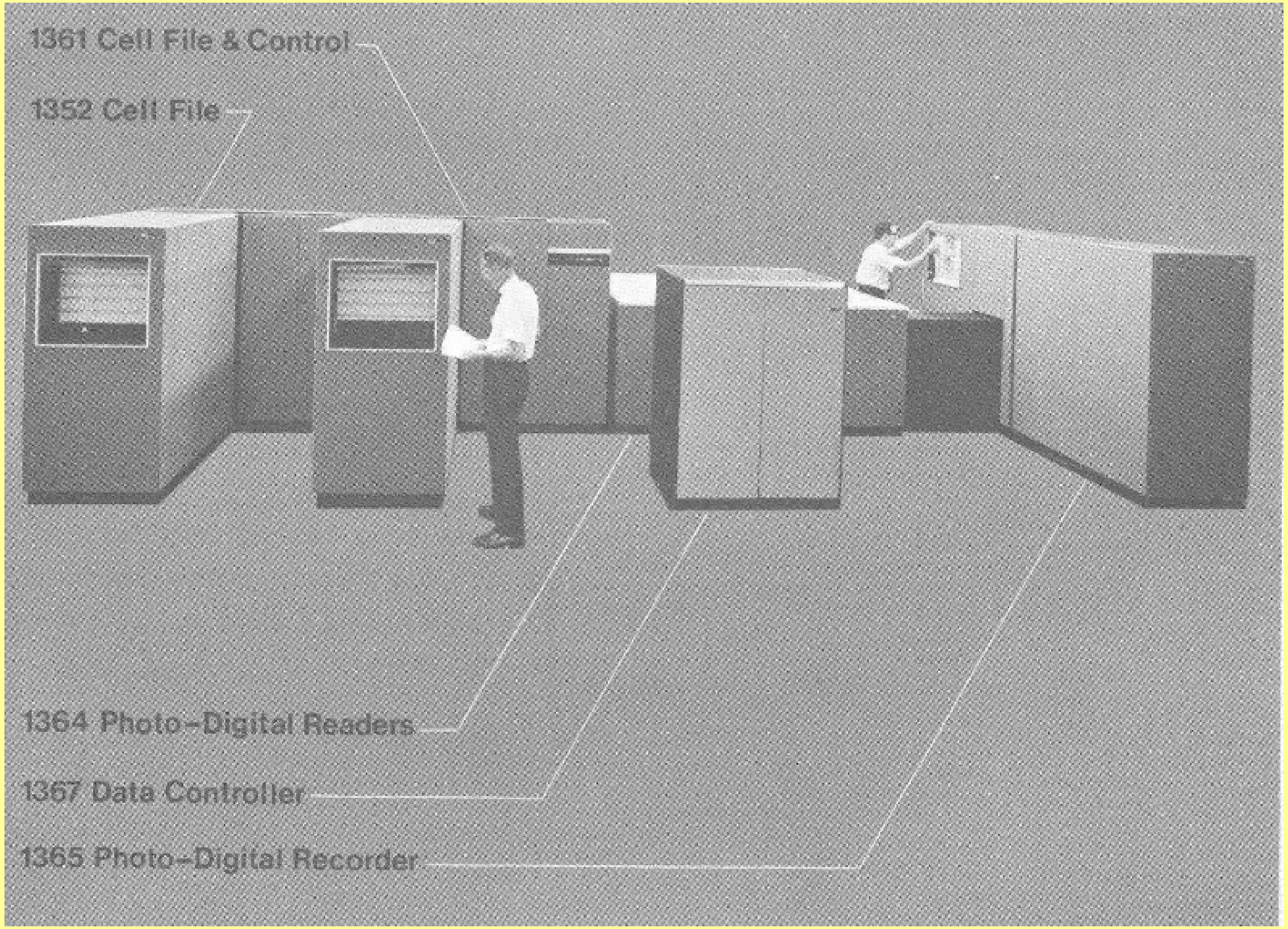
IBM Data Cell 2321
Reader/writer Head



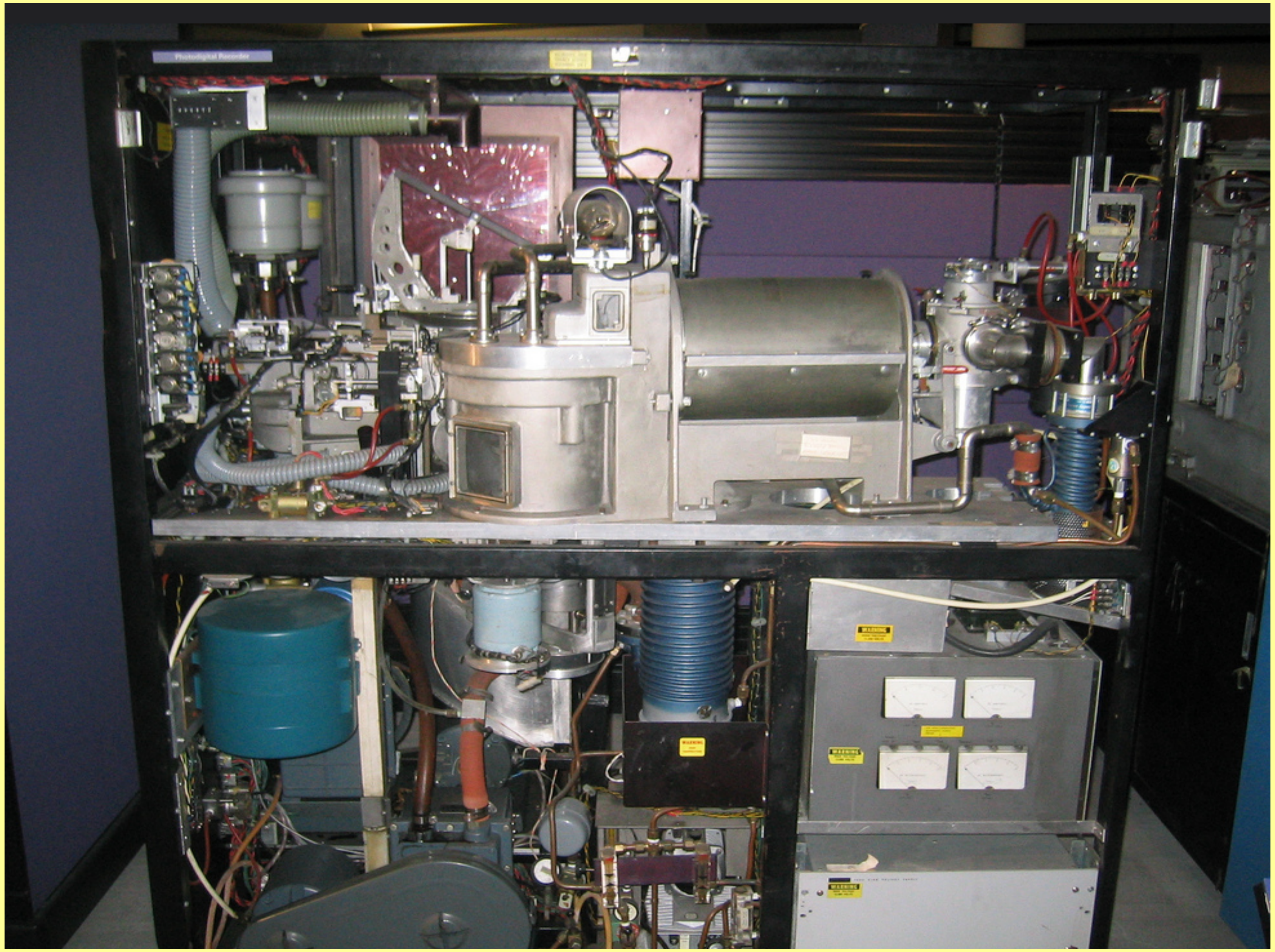
IBM Data Cell 2321 Strip
Somewhat the worse for wear

Chipstore

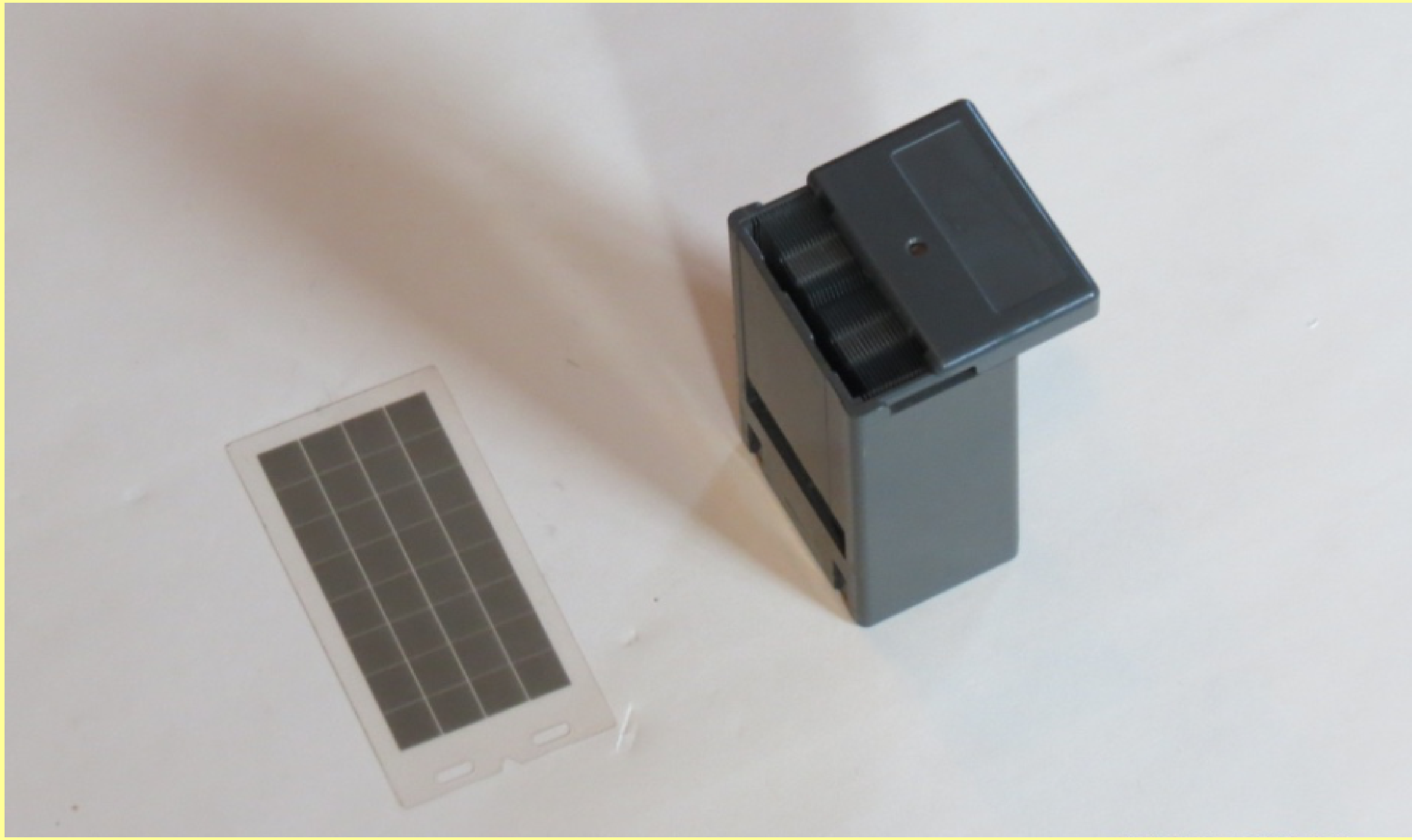
- IBM 1360 Photo-digital Chip Store
  - Wrote chip with electron beam; developed with wet chemistry
  - Each chip had 32 fields, each with 128 KB
  - 32 chips stored per cell
  - 2,250 cells in cell file unit
  - 2 Units hadTotal capacity ~ 1.2 TB -- the first TB device
  - Chips and cells were handled by robot, using pneumatic tubes
    - Write speed: 500 Kbits/sec. Read speed: 2.5 Mbits/sec.



IBM Photodigital Chip Store



IBM Photodigital Chip Store Guts



IBM Photodigital Chip Store
Chip and Cell

Program Agenda

- Introduction
- The IBM 709x Series
- The CDC 6600
- The CDC 7600
- Mass Storage Technologies
- Conclusion

Conclusion: The End of the Era

- In 1985, LBL retired its CDC Equipment
  - Owned by LBL, with no resale value
  - Present and former systems programmers invited to decommissioning party
  - At the party, the CDC machines were trashed
    - Various modules taken (usually, those with each person's initials)
      - Unfortunately, including the master timing module, one per 7600, that CDC wanted as a spare
    - Wires ripped out of the backplane
    - That's where most of the artifacts came from
  - ~$10,000,000 worth of Hardware ==> $0 in an afternoon

Plus Ça Change, Plus C'est La Même Chose

Despite many orders of magnitude change in performance and scale

- Code generation has the same issues:
  - Instruction scheduling, register allocation
    - In the OS code, it's done manually, however
- Laying code out in memory:
  - Function ordering for overlays == ordering to minimize I-cache misses
- Careful design, and bench checking still important
  - IMNSHO -- a minority opinion, at least in practice