CSE 301
History of Computing

Large Scale Computer Projects

Four Major Initiatives

- Whirlwind
- Semi-Automatic Ground Environment (SAGE)
- Semi-Automatic Business Research Environment (SABRE)
- The Apollo Moon Mission

Whirlwind

- Developed at MIT for the US Navy
- Flight Simulator system to train bomber crews
- Jay Forrester was in charge of the project, design completed in 1947 with Robert Everett
- Perry Crawford of MIT in 1945 sees a demo of the ENIAC and sees digital computer as solution
- 1st computer to operate in real time and to use video displays for output
- Development led directly to the US Air Force's SAGE system, and indirectly to minicomputers of the 1960's

Whirlwind

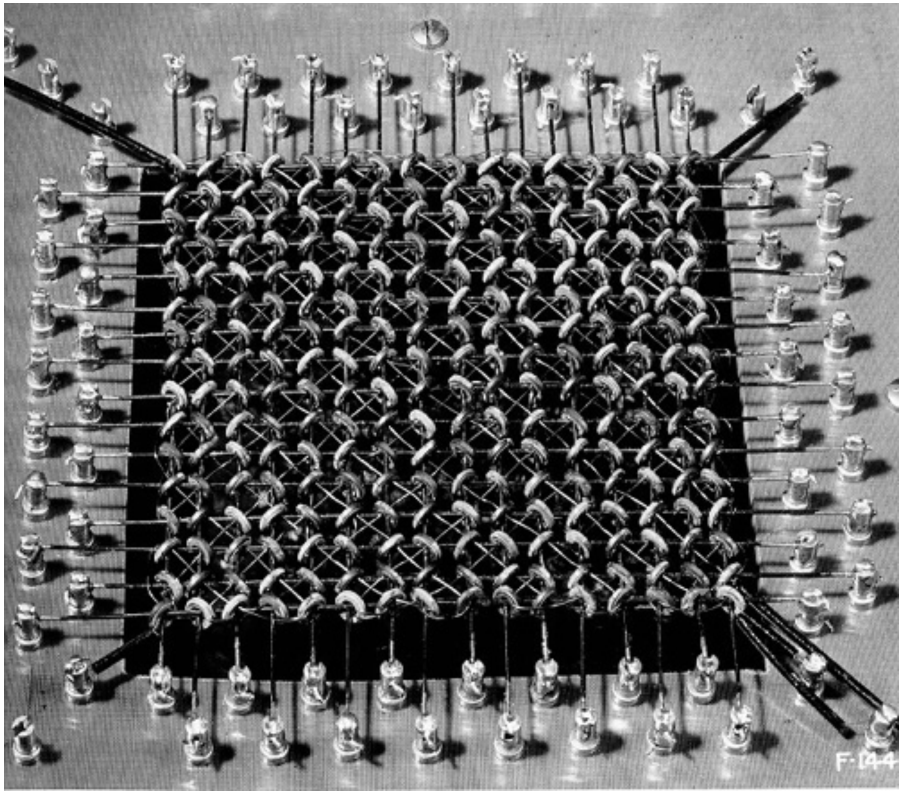
- Most computers of the era operated in bit-serial mode, not fast enough for real-time
- Whirlwind had 16 math units operating on 16-bit words in bit-parallel mode (16X faster)
- CPU’s of today do arithmetic in bit-parallel mode, sometimes on 32- or 64-bit words
- Whirlwind took 3 years to build and went online April 20, 1951
- 175 people on the project, incl. 70 engineers & technicians. $1 million annual budget.

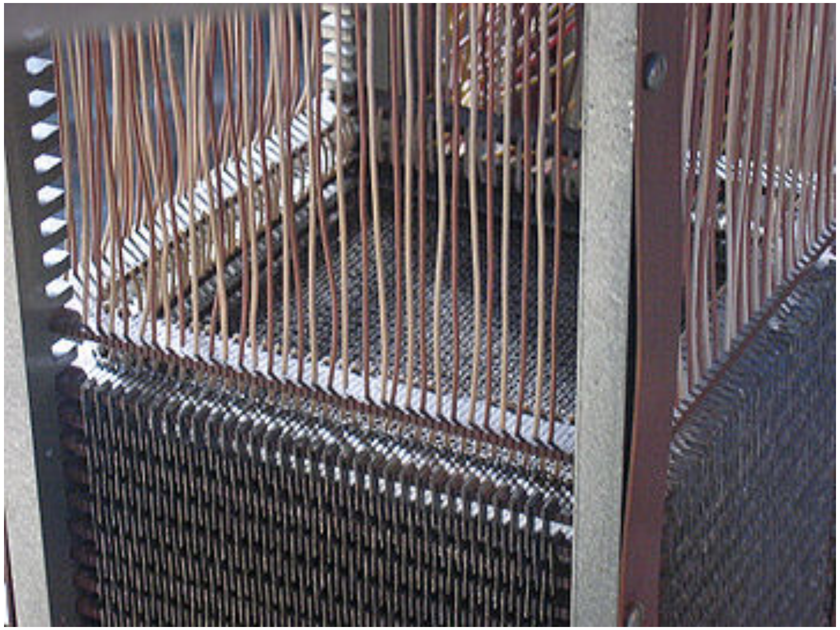
Whirlwind

- USAF takes over the project from Navy, and uses the Whirlwind computer in the Cape Cod System, a prototype of SAGE
- Addition 8 microseconds, multiplication 25.5 microseconds, division 57 microseconds
- 5000 vacuum tubes
- Effort to transistorize led to the TX-0, led by Ken Olsen
- Olsen leaves MIT to start Digital Equipment Corp. (DEC)

Whirlwind Computer

Whirlwind Core Memory





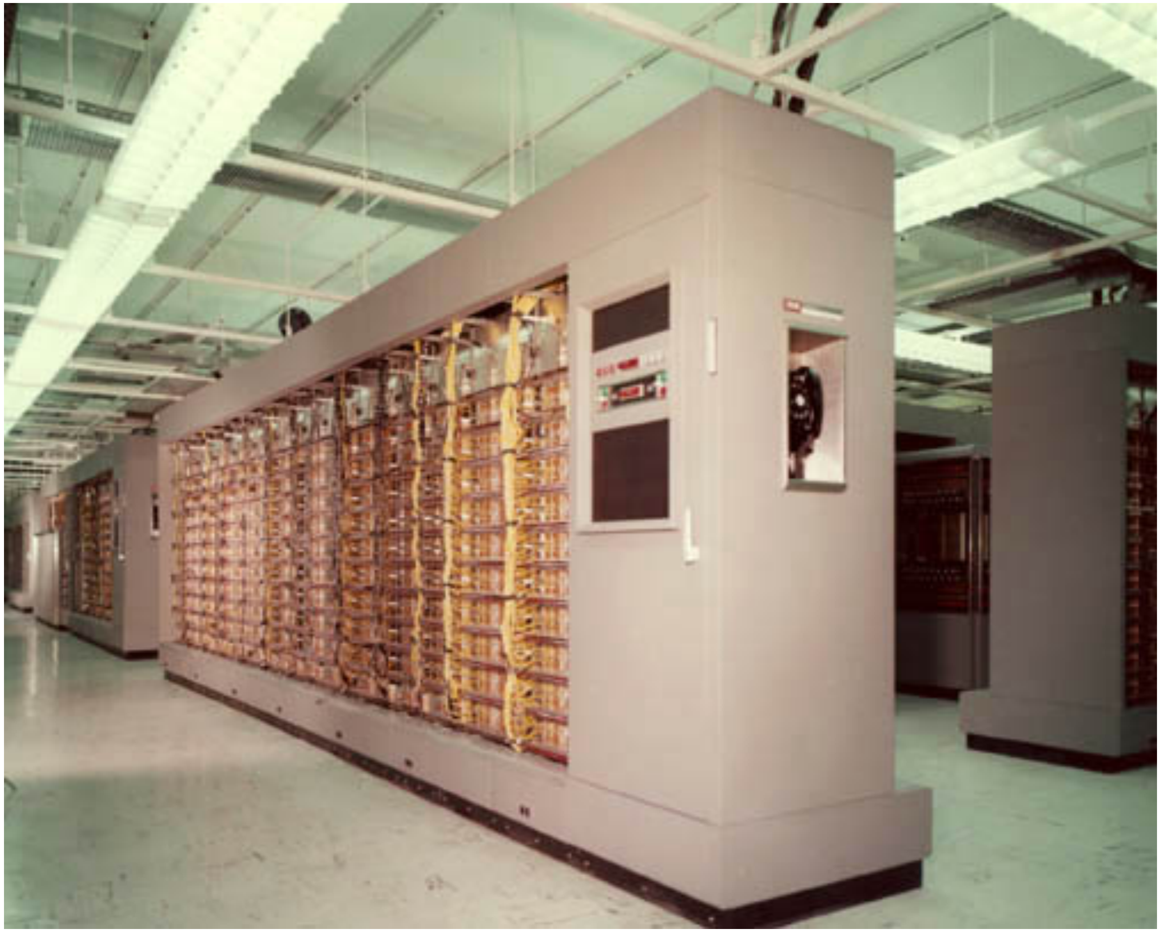
SAGE

- Automated control system for tracking and intercepting enemy jet bomber aircraft from the 1950's into the 80's.
- Real time computing, data communication using modems.
- IBM wins contract to develop SAGE's AN/FSQ-7 computer
- Largest computer ever built
- Bomber threat became missile threat before SAGE was operational

SAGE

- 55,000 vacuum tubes
- ½ acre of floor space, 275 tons and consumes up to 3 megawatts of power.
- Two systems for redundancy for each SAGE center
- Hot-swappable components (vacuum tube trays)
- CRT-based real-time user interface
- Between $8 & $12 B in 1964 dollars ($55 B in 2000 dollars)
- Influenced design of FAA’s air traffic control system and led to SABRE project with American Airlines

SAGE computer



SAGE interface

SABRE

- Computer reservation system used by airlines, railways, hotels, travel agents.
- Developed to help American Airlines improve booking operations.
- Replaced manual system in place since 1920's
- Serendipity: top IBM salesman, Blair Smith, seated next to AA president C R Smith, on a flight from LA to NY in 1953. Common family name begins conversation that ends up with an IBM proposal to build SABRE 30 days later.

SABRE

- 3 years in development (1957-1960) at a cost of $40M ($350M 2000 dollars)
- Takes over all booking operations in 1964. Opened to travel agents in 1976. Spun off by AA in 2000, taken private in 2007.
- Travelocity web site introduced in 1996. System today connects 3 million consumers to 30,000 travel agents, 400 airlines, 50 car-rental companies, and 35,000 hotels.

SABRE computer

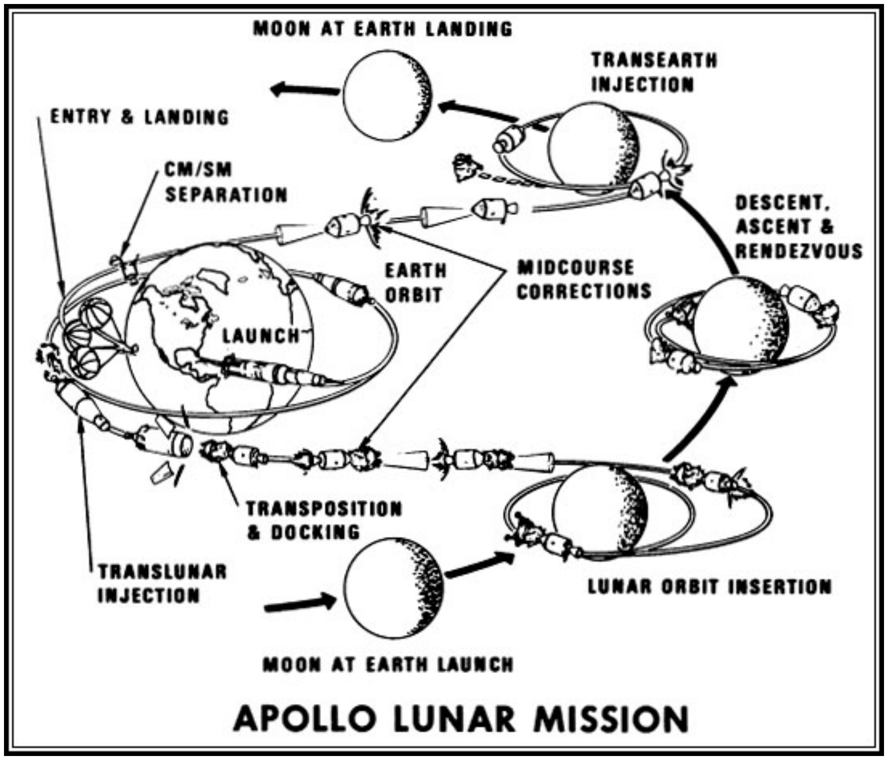
Apollo Space Program

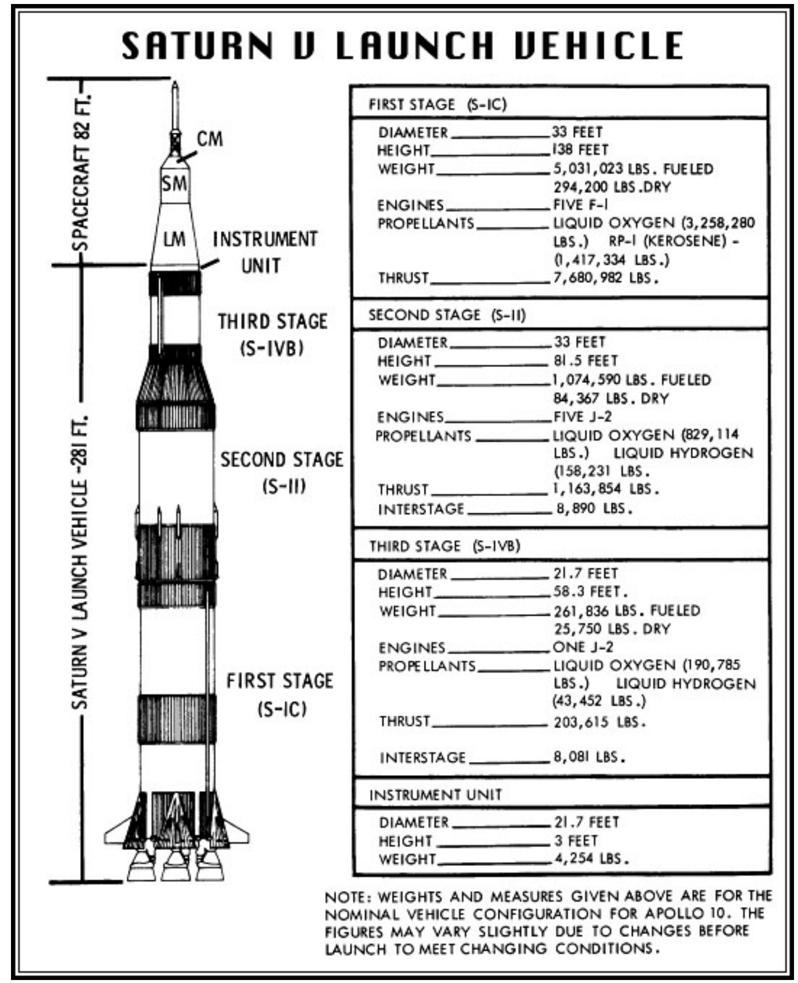
- Origin of Computer Simulation and Modeling (changing role of mathematical analysis)
- Era of Dual Computer Environments (scientific and business)
- Software Engineering (evolving recognition of need for software engineering processes)

Apollo Program

- In 1961, President Kennedy announced the goal of landing a man on the moon before the end of the decade (60s)
- Program goals:
  - To establish the technology to meet other national interests in space.
  - To achieve preeminence in space for the United States.
  - To carry out a program of scientific exploration of the Moon.
  - To develop man's capability to work in the lunar environment

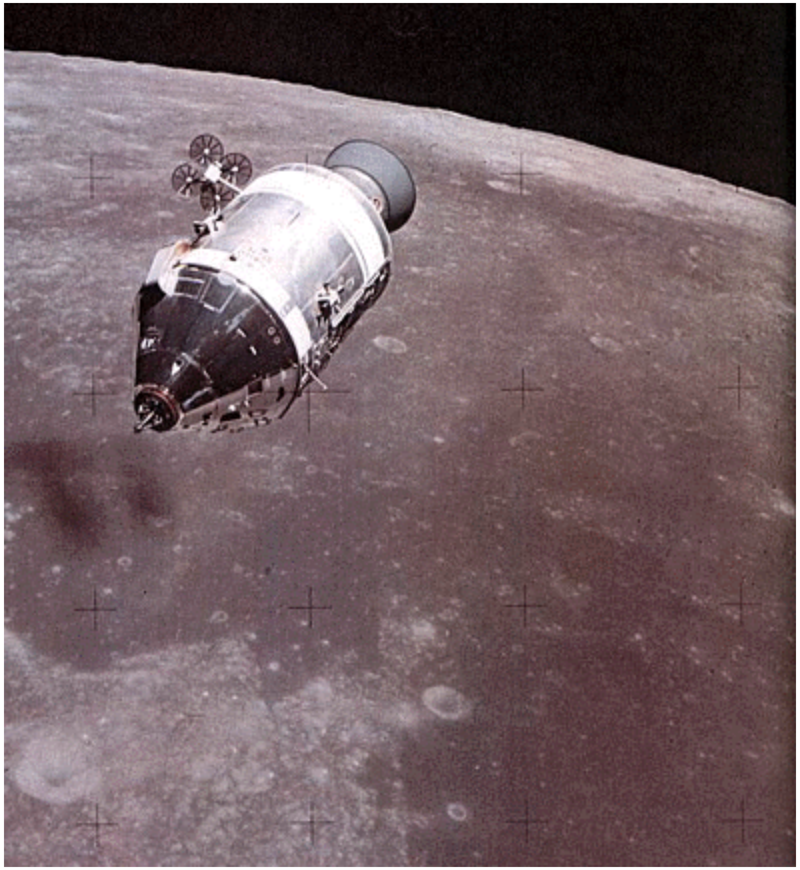
Apollo Mission





Use of Computing in Apollo Program

- On-board control (e.g., navigation, propulsion, instrumentation)
- Analysis (e.g., structural, thermal, simulation, operations analysis)
- Project management (parts lists, inventory, payroll)



Apollo Guidance Computer (AGC)

- Controlled all navigational functions
- First computer to use integrated circuits (ICs)
- Each IC contained a 3-input nor gate
- Magnetic core memory
  - 16 bit word
  - 4K words
  - 12 microsecond access time
- Software – AGC Assembler

IBM 7094

- Typical scientific computer
- $3M
- Transistor technology
- 2 microsecond cycle time
- Optimized for floating point arithmetic
- 32K words memory – magnetic core

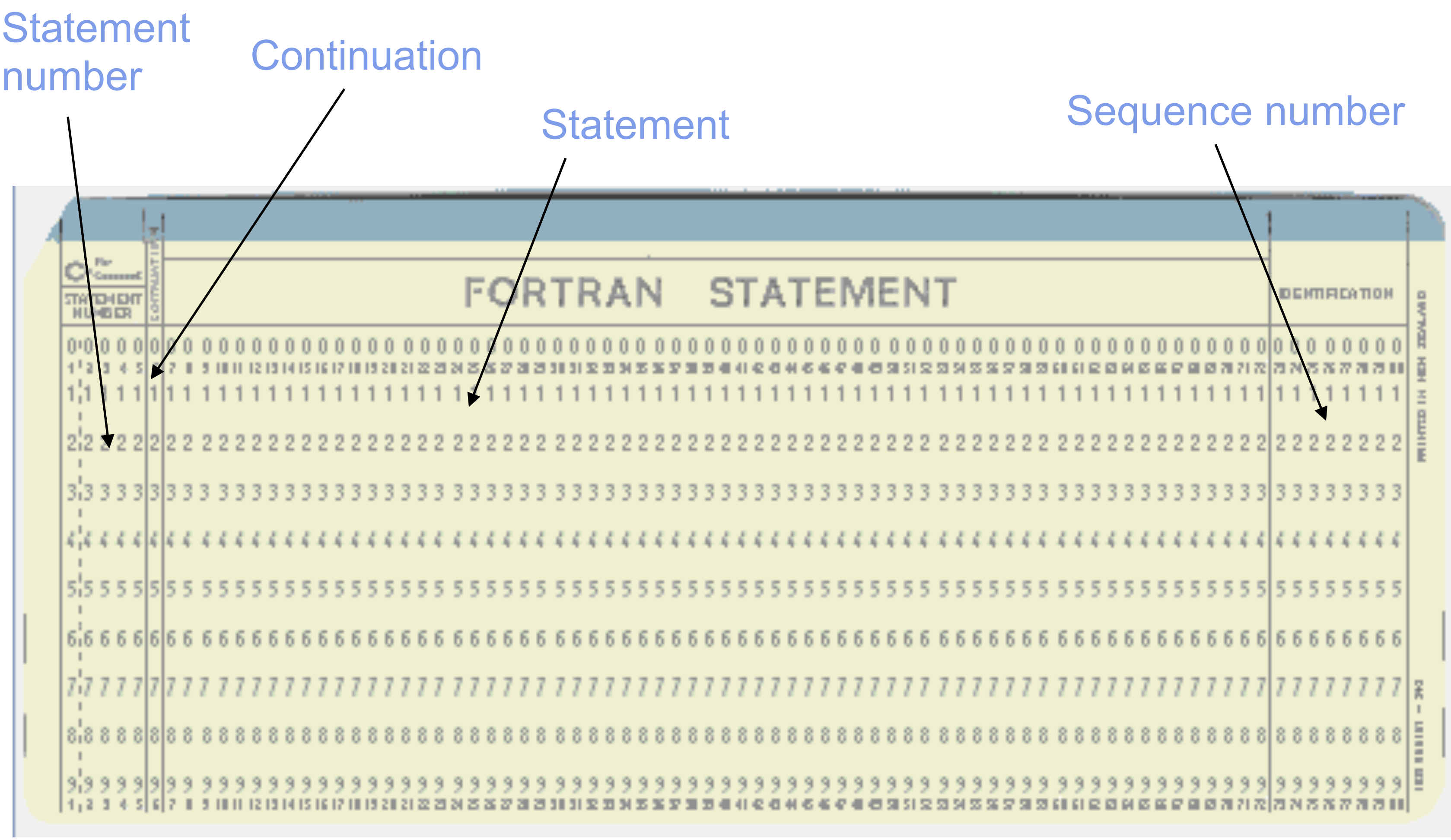
Apollo Software Development

- Software engineering, as we know it, did not exist – but software development was done within the systems engineering structure
- No design documents, code reviews, test plans, best practices, etc.
- Software was considered an extension of mathematical analysis
- Typical programs were one huge module (1 - 4 thousand lines of code)

Apollo Programming Process

- Code entry – coding sheets
- Coding sheets given to key punch operators
- Resulting punch cards were printed, verified and inserted into card deck
- Editor – listings and a box of punch cards
- One execution of the program per night – you had to be perfect the first time

FORTRAN


Statement number
Continuation
Statement
Sequence number
FORTRAN STATEMENT
STATEMENT NUMBER

Punch Cards

Characters determined by punch holes in a column

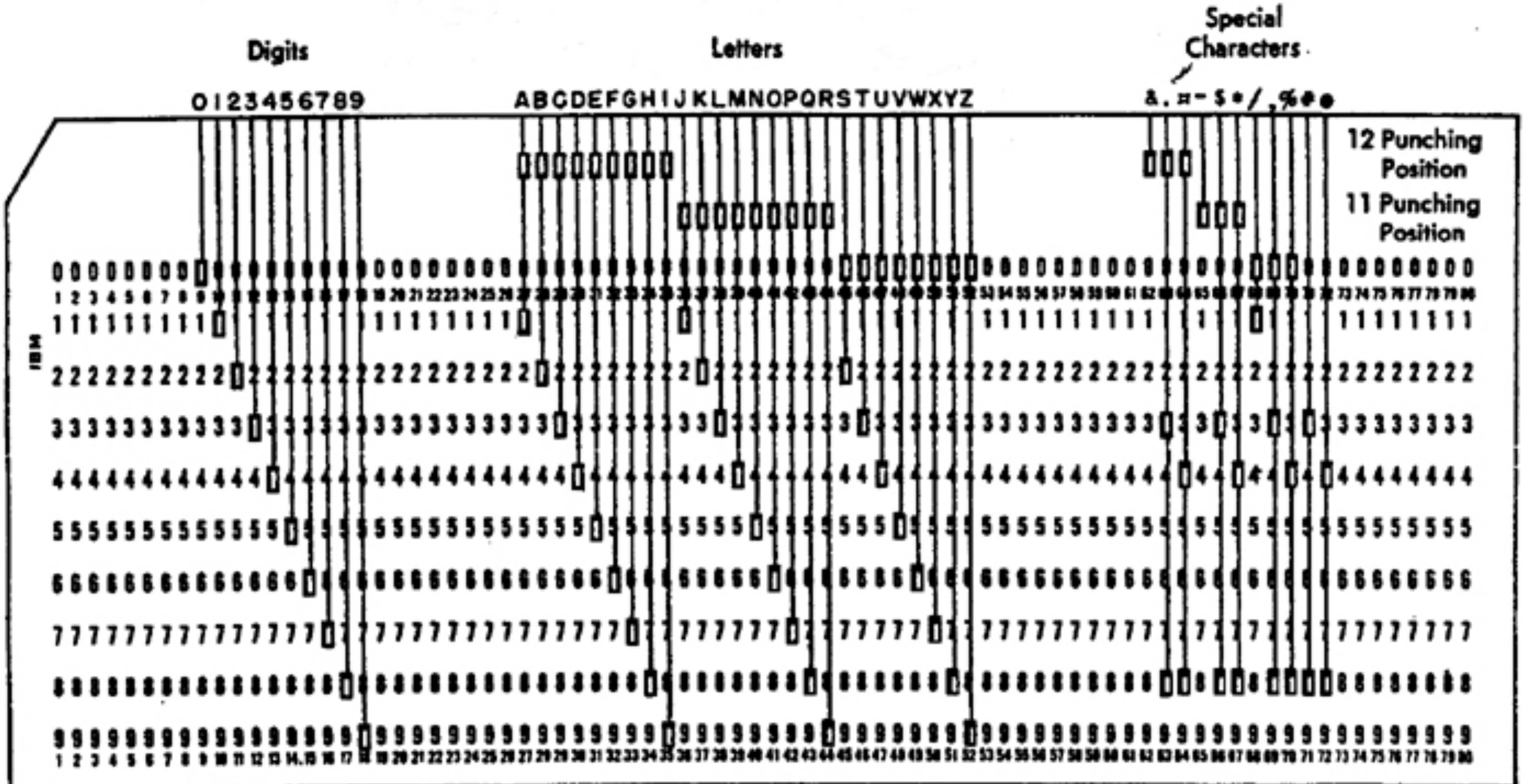


Figure 2. Punching Positions in Card

FORTRAN

- BY the late 1960s, the software battle had been won – in favor of high level languages
  - Assembler Language – high performance
  - FORTRAN – more portable
- Language features
  - Limited structuring
  - Common blocks
  - Dynamic loader
  - Fixed field instruction layout (key punch)
  - Very efficient – close to assembler language in performance

Memory and Processor Management

- Memory was very limited, so approaches were developed to deal with that limitation
  - Loader map – programmed the loading of modules into memory so you only had what you needed
  - Aliasing – using multiple identifiers to refer to the same memory location
- Secondary storage – tape storage for partial results
- Processor was very slow, so approaches were developed to deal with that limitation
  - Assembler subroutines – subroutines that were frequently invoked were coded in assembler
  - Deep understanding of compiler optimization
  - Deep understanding of relationship between source code and object code

Software Engineering Consequences

- In the 1960s, people were cheap and computers were expensive
- Software reliability was not well understood
- Software maintenance implications were not well understood
- Advantages of code portability were not well understood

Computational Analysis

- Idealized structures
- Equations of motion, etc.
- Large range of test cases
- Instrumented drop test (and crash test) used to verify mathematical and computer assumptions

Scientific Vs. Commercial Computing

Scientific

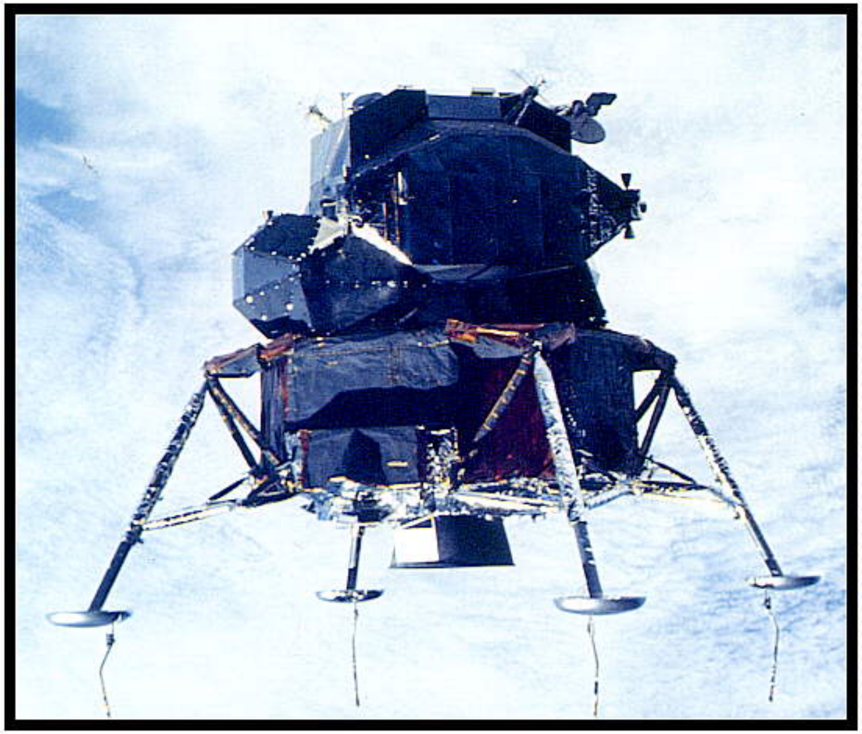
- Word computers
- Floating point arithmetic
- Double precision
- FORTRAN software
- Graphic displays (pen plotters and vector displays)
- Expensive
- CDC, IBM, etc.

Commercial

- Limited instruction set
- Character manipulation
- File IO
- COBOL, RPG software
- Card readers, paper tape
- IBM, Burroughs, etc.

Lunar Excusion Module (LEM)

- Designed to travel between lunar orbit and the lunar surface
- Designed and built on Long Island by Grumman
- Landed on the moon in 1969
- One LM is now in Smithsonian museum (Washington DC)



Lunar Excursion Module (LEM)

- Most analytical work for Lunar landing performed on 360/75
- Analytical work for subsequent landings performed on the first commercial virtual memory system (360/67)

Space Era Summary

- Importance of computer modeling and simulation
- Convergence to single architecture for commercial and scientific processing
- Growth of software engineering issues