Stored Program Considered Harmful: History & Historiography

Thomas Haigh
University of Wisconsin—Milwaukee

Paper presented at CIE 2013 Special Session on History of Computing

Broader Project

- History of ENIAC in use
  - ENIAC built 1943-45 at University of Pennsylvania for US Army
  - Converted to new “stored program” mode in 1948
  - Used intensively until 1955
- Historical status unclear in new mode

Thomas Haigh.
www.tomandmaria.com/tom

Early Computing

- History of Computing Emerges from 1970s
  - Endless early arguments over “the first computer”
  - What determines? “General purpose,” programmability, electronic construction, etc.
- Historians agree to nuance
  - Each early machine gets a “prize”
- Mike Williams in introduction to *The First Computers:*
  - “If you add enough adjectives to a description you can always claim your own favorite. For example ENIAC is often claimed to be the ‘first electronic, general purpose, large scale, digital computer’ and you certainly have to add all those adjectives before you have a correct statement.”
- But “first stored program computer” is the biggest remaining one

Electronic Computing Conventional History of "Firsts"

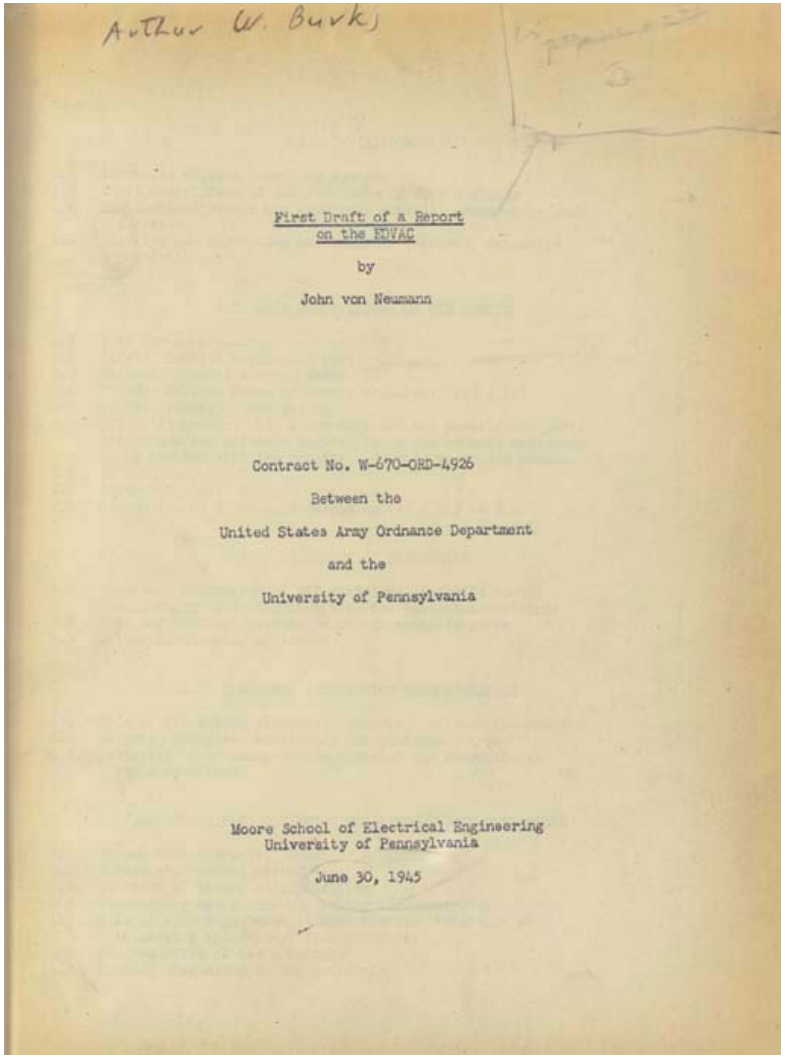
- 1943 Atanasoff-Berry Computer
  - Electronic, digital, special purpose, never quite worked
- 1944 Colossus (UK)
  - electronic, digital, special purpose
- 1945 ENIAC (Philadelphia)
  - electronic, digital, general purpose, but not "stored program" – configure by wires & switches. 18,000 tubes.
- 1948 (June) Manchester "Baby"
  - Electronic, digital, stored program, but test bed for memory device so no useful function
- 1949 EDSAC (Cambridge)
  - Electronic, digital, stored program, "usable" – runs many real programs. 3,000 tubes.
  - Many similar computers soon follow

Doron Swade's Confession

- Former senior curator at Science Museum, London
- For years he "assumed that the significance of the stored program must be self-evident" and attributed his own confusion to personal inadequacy, until finally he "became bold and began asking" among computer historians and pioneers what it actually was.
- Their answers were "all different," with the question of whether "the primary benefit was one of principle or practice frustratingly blurred."
  - "no one challenged the status of the stored program as the defining feature of the modern digital electronic computer…. "
  - "But it seems that we struggle when required to articulate its significance in simple terms and the apparent mix of principle and practice frustrates clarity."

The “First Draft” Report, 1945

- Universally cited as first and definitive statement of stored program concept.
- Informally circulated by Herman Goldstine from notes by John von Neumann
  - Credit for ideas within intensely disputed
- Often cited, little read
  - Hard to obtain until 1990s
  - Unfinished
  - Uneven – highly detailed in places
  - Confusing terminology – e.g. “organs,” “neurons”



Arthur W. Burks

First Draft of a Report
on the EDVAC

by

John von Neumann

Contract No. W-670-ORD-4926

Between the

United States Army Ordnance Department

and the

University of Pennsylvania

Moore School of Electrical Engineering
University of Pennsylvania

June 30, 1945

Thomas Haigh.
www.tomandmaria.com/tom

“Stored Program” in First Draft

- Musings at early state for new U. Penn. Contract on an ENIAC successor, codenamed EDSAC
  - Not really a standards document!
  - Not intended for publication
- Phrase “stored program” does not occur
  - Neither does the word “program” (JvN liked “code”)
- Does discuss code storage, but
  - “instructions must be given in some form which the device can sense: Punched into a system of punchcards or on teletype tape, magnetically impressed on steel tape or wire, photographically impressed on motion picture film, wired into one or more fixed or exchangeable plugboards—this list being by no means necessarily complete.”

What IS in First Draft?

- Key architectural ideas are there
  - Interchangable storage of code and data in numbered memory locations is “tempting”
  - Minimal instruction set of resusable operations
  - Modification of address fields of operations
  - Separation of control, memory, storage
    - All transfers and arithmetic via special purpose registers
- But also masses of detail, speculation, musing on requirements, notes on tubes, etc.

Initial Reception

- Ideas from report accepted quickly
- Understanding of the benefits of “EDVAC-type” machines varies in 1946-50 era
  - Simplicity of hardware is key benefit
  - Interchangeability of program and data storage is mentioned, largely from efficiency
  - Modifications to code in memory sometimes mentioned
    - EDVAC would use for all loop termination, conditional branch, etc.
- Connections to Turing machine model, analysis of universality, etc. not yet relevant

The Phrase “Stored Program”

- Enters with IBM, 1949
- Building a “Test Assembly” incorporating
  - Electronic tabulating machine
  - Magnetic drum memory
  - New logic and control unit with electronic memory
- “Stored program” specifies the program stored in memory vs. wired on the plug board
- Phrase has some use, mostly by IBM people, in public in the 1950s but not that common

Conflation with “Universal Computer”

- In recent decades, historians and other writers tend to treat as synonymous
  - Stored program computer
  - Universal computer
  - Von Neumann architecture
- Popular histories also tend to write as if early computer projects were directly inspired by Turing’s work of the 1930s

Different Aims of History versus Theory

- Theoretical work looks for minimum logically sufficient abstract machine
  - i.e. what is the LEAST we can take from “First Draft” and still get the computational power
  - In this case RASM for “register machine”
  - RASP – Random Access Stored Program Machine, etc.
- Historians are interested in tracing the actual influence of different aspects of the first draft over time
  - i.e. what is the MOST that machines of the early 1950s took from the “First Draft”

Paper Proposes Three More Precisely Definable Alternatives to SPC

- All found in 1945 “First Draft”
- All influential on later machines, though frequently adopted separately
  1. Modern Code Paradigm
  2. von Neumann Architecture
  3. Hardware Paradigm
- Why “Paradigms”
  - Kuhn’s original sense of a tangible exemplar, built on and extended as the basis for a new scientific community

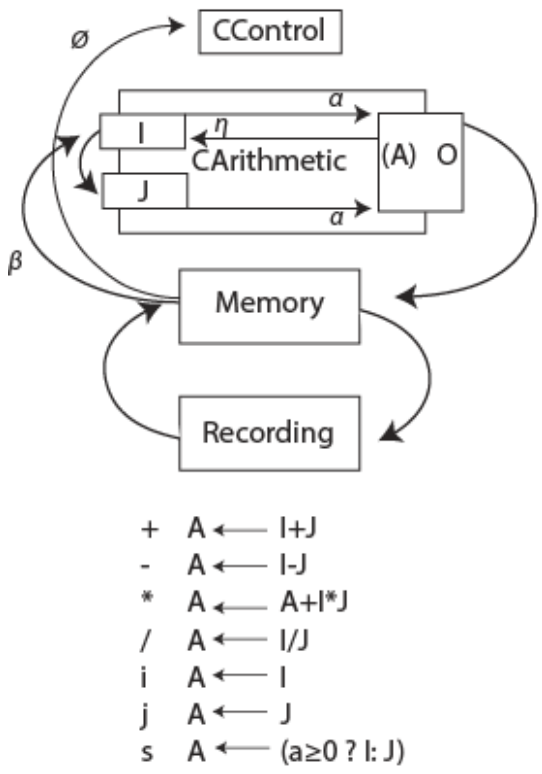
EDVAC Hardware Paradigm

- Entirely electronic storage
- Binary coding
- Mercury delay line storage, sufficient for code and data
- Minimal use of special purpose hardware
  - Multiplier: OK
  - Dedicated circuits for table lookup, square roots, loop termination, floating point, etc: not OK
- Radical minimalism of design
  - Means cheaper, more reliable, far more feasible

Von Neumann Architecture Paradigm

- Serialization of computation
- Structure of “Organs” separate
  - Memory (large and fast)
  - Control
  - Arithmetic
- All memory transfers via arithmetic unit
- Special purpose registers
  - Program counter
  - Instruction register
  - Fixed source and destination registers for arithmetic

John von Neumann *et al.*, EDVAC Architecture



Modern Code Paradigm 1

- **The program is executed completely automatically.**
  - "Once these instructions are given to the device, it must be able to carry them out completely and without any need for further intelligent human intervention."
  - Essential for electronic machines, whereas manual intervention at branch points had been workable with slower devices such as the Harvard Mark I.

Modern Code Paradigm 2

- **The program is written as a single sequence of instructions, known as “orders” in the First Draft, which are stored in numbered memory locations along with data.**
  - These instructions control all aspects of the machine’s operations.
  - The same mechanisms are used to read code and data. However the “First Draft” design used flag on locations holding code and prevent overwriting!

Modern Code Paradigm 3

- **Each instruction within the program specifies one of a set of atomic operations made available to the programmer.**
  - **This was usually done by beginning the instruction with one of a small number of operation codes.**
  - Some operation codes are followed by argument fields specifying a memory location with which to work or other parameters.
  - Altogether, orders required between 9 and 22 bits to express.

Modern Code Paradigm 4

- **The program's instructions are usually executed in a predetermined sequence.**
  - the machine "should be instructed, after each order, where to find the next order that it is to carry out."
  - represented implicitly by the sequence in which they were stored, as in "normal routine" it "should obey the orders in the temporal sequence in which they naturally appear."

Modern Code Paradigm 5

- **A program can instruct the computer to depart from this ordinary sequence and jump to a different point in the program.**
  - "There must, however, be orders available which may be used at the exceptional occasions referred to, to instruct CC to transfer its connection [i.e. fetch the next instruction from] any other desired point" in memory."
  - This provided capabilities such as jumps and subroutine returns.

Modern Code Paradigm 6

- **The address on which an instruction acts can change during the course of the program's execution.** That applies to the source or destination of data for calculations or the destination of a jump.
  - the final sentence of the First Draft noted that when a number was transferred to a memory location holding an instruction only the final thirteen digits, representing the address μp, should be overwritten.
- Actual computers achieved functionally equivalent capability through some combination of unrestricted code modification, indirect addressing mechanisms, and conditional branch instructions.

Initial Adoption Separate

- Turing's Ace design (1945, versions operational from 1950)
  - Code: Partially (no instruction codes)
  - Architecture: Partially (different conception of registers)
  - Hardware: Yes (delay line memory, all electronic. 800 tubes!)
- Bromley's ARC (claimed operational May 1948)
  - Code: Yes
  - Architecture: Yes
  - Hardware: No (relay based rather than electronic)
- ENIAC when reconfigured (operational April 1948)
  - Code: Yes (address modification via indirect addressing)
  - Architecture: No (still very baroque)
  - Hardware: No (memory still limited)

Later Trajectories Separate

- Modern Code Paradigm
  - Still basically describes machine language, though increasingly rarely written by humans
- Von Neumann Architecture Paradigm
  - Eroded gradually over the years, but change largely evolutionary
- EDSAC Hardware Paradigm
  - Rapidly becomes obsolete
  - E.g. transistors, core memory

Conclusions

- "Stored Program Concept" has been given too many different meanings and associations over the years
  - Literal meaning is unclear
  - Often used to include ideas added post-1945
  - Not always distinguished vN Architecture
- Solution: historians should separate this fuzzy concept into the three stated paradigms. Each is
  - Clearly grounded in the 1945 "First Draft"
  - Capable of precise definition
  - Shown to have a separate history of adoption, evolution, and eclipse