WHEN
100 FLOPS/WATT
WAS A GIANT LEAP

THE APOLLO GUIDANCE COMPUTER
HARDWARE, SOFTWARE AND
APPLICATION IN MOON MISSIONS

Mark C Miller, LLNL (miller86@llnl.gov), July 17 2019

Presented in the webinar series
Best Practices for HPC Software Developers
OUTLINE

• Background
• Hardware Architecture
• The Software Effort
• Brief Detour
• Mission Applications
CURRENT GENERATION HPC/CSE SOFTWARE DEVELOPERS WILL RECOGNIZE MANY COMMON THEMES

• Flops/Watt power constraints
• Checkpoint/Restart
• Performance Portability
• Co-Design
• Sufficient Testing Resources
• Role and Impact of Software Development Processes
Virtual AGC Project: https://www.ibiblio.org/apollo/

3-Part Blog Series on Better Scientific Software Site (bssw.io)

Part 1 | Part 2 | Part 3
OUTLINE

• Background
• Hardware Architecture
• The Software Effort
• Brief Detour
• Mission Applications
WHAT WAS THE APOLLO PROGRAM?

• 10 year project, starting in 1961 to land people on the moon
  • 36 attempts 1958-1965; none survivable

• 7 Lunar Missions

• The Apollo Guidance Computer (AGC) was instrumental in the success
Early Sixties
State of the Art Computers

- 4,000 ft$^3$
- 8 tons
- 125 Kilowatts
- MTBF $\approx$ Days
- Reboot $\geq$ 30 mins
- UI = Punch Cards & Printouts
- Time slice multi-tasking
- $\sim$1 Flops/Watt
1966 BLOCK II AGC

- 1 cubic foot volume
- 70 lbs weight
- 55 Watts power
- MTBF ≥ Months
- Reboot ≈ 7 seconds
- UI = Verb/Noun ELD (DSKY)
- Priority Based Multi-Tasking
- ~259 Flops/Watt
APOLLO SPACECRAFT

- Ascent
- Descent
- Attitude Control
- Thrusters
- Main Engines
ROLE OF THE COMPUTER

• In powered or coasting flight, manage the State Vector
  • Position & Position Rates
  • X, Y, Z & ∆X, ∆Y, ∆Z
• Attitude & Attitude Rates
  • R(oll), P(itch), yaw & ∆R, ∆P, ∆W
• Real-time, Accurately, Reliably
• Autonomously

In spite of many constraints and challenges...
• Sensor bias and drift
• Avoiding orientations causing IMU gimbal lock
• Moon's lumpy gravity field
• Changing center of mass (fuel slosh & loss)
• Minimize fuel consumption
• Communication lapses and blackouts
• Allowing for failures & contingencies

Long periods of boredom punctuated by moments of extreme peril
EXAMPLE MANEUVER: LUNAR ORBIT INSERTION (LOI)

- Velocity = 2 miles/sec
- Distance from moon = 60 miles
- RT signal to Earth = 2.5 sec
- Insertion burn on far side
MIT INSTRUMENTATION LAB
PRIME CONTRACTOR ON APOLLO PGNCS

• Design the entire guidance system
• Draper Labs: Charles Stark “Doc” Draper (Apollo’s Iron Man)
• Designed the Polaris missile guidance system
• Massive R&D effort
OUTLINE

• Background
• Hardware Architecture
• Guidance Software
• Brief Detour
• Mission Applications
AGC HARDWARE OVERVIEW

SPECS
• 16 bit word size (15 + odd parity)
• 1.024 MHz Clock
• 12-pulse micro-sequence
• 4 central reg’s + ~15 special reg’s
• 2K words Erasable Memory (RAM)
• 36K words Fixed Memory (ROM)
• Both RAM and ROM were NVM
AGC ARCHITECTURE OVERVIEW

• 4 Central registers
  • A: accumulator w/overflow bit
  • Z: program counter
  • Q: div-remainder / return addr
  • L: lower-product

• Other special purposes registers
  • ROM / RAM memory banking
  • Editing (shift) registers
  • Zero / NEWJOB (000678)
  • Not directly programmable

• 8 basic + 33 extended instructions
  • Data Movement
  • Arithmetic & Logic
  • Flow Control
  • I/O & Interrupts

• Many exotic I/O devices

• Programmed in Assembly Language
ACTUAL MACHINE INSTRUCTION SET

- **CS**: clear and subtract
- **TS**: transfer to storage w/ overflow handling
- **XCH**: exchange A w/ storage
- **AD**: add
- **MASK**: bit-wise and
- **TC**: Transfer control
- **CCS**: count, compare & Skip
- **INDEX**: add (+/-) offset to next instruction
- **MP**: multiply
- **DV**: divide
- **Others**…
NUMERICS OF THE AGC

WHOLE NUMBERS

- 16 bit, 1’s complement, big endian
- 1 bit for parity (bit 16) & sign (bit 15)
- 14 bits for magnitude range 0…2^{14}-1
- Full range of -16,383_{10} to +16,383_{10}

P|S|2^{13}|2^{12}|2^{11}|…|2^{1}|2^{0}|

Fixed Point Representation

Coders must ensure proper scaling!!

Single, double, triple precision

+,-,x,÷ = 35.1, 70.2, 133.4

𝜇sec

FRACTIONAL NUMBERS

![Flops/x Computing Metrics Comparison](image-url)
THE AGC EXECUTIVE (OPERATING SYSTEM)

"TASKS"

- Short, finely tuned
  - < 5 ms (150-200 instructions)
- Scheduled by time (in the future)
- Some tasks only schedule a “job”
- Waitlist data structure to manage
  - List of tasks sorted by time to run

“JOBS”

- Priority Scheduled
- 12 words of state (4 regs + MPAC)
- 43 words for vector accumulator
  - only for Interpreted jobs
- Jobs adjust their own priority up/down
- New Job checked every 20 ms
  - Two basic instructions (CCS/TC)
  - At end of every interpreted instruction
WAYPOINTS AND RESTART

• Critical routines were restart protected

• Restart phase tables maintained in fixed memory

• Waypoints (phase table pointers) periodically updated in erasable memory

• Consumed 4% of fixed memory, additional coding and testing complexity
A form of compression to tradeoff memory space for time
I/O PROCESSING (MEMORY MAPPED)

“CHANNELS”
- Very low update rate
- Keystrokes & ELDs on DSKY
- Caution & Warning lights
- RCS Thruster firing
- Switch Statuses
- Managed via interrupt routines

“COUNTERS”
- Pulses from fine-grained state devices
  - IMU gimbals
  - Main engine gimbals
  - Optics & Radar gimbals
- PINC/MINC “instructions”
- Not managed by software
- Cycle stealing
FAULT TOLERANT COMPUTING WAS CRITICAL

- Hardware level power checks
- Parity check every memory ref
- NEWJOB word night watchman
- Program Alarms (e.g. radar turned off)
- P00DOO (program aborts)
- System Restarts (< 7 seconds)
  - Key data downlinked to Huston

- Extreme Reliability Achieved
  - 42 Units
  - 11,000 hours of vibration and heat/cold
  - 32,000 hours normal operation
  - Only 4 faults observed
  - MTBF $\rightarrow$ 40,000 hours
OUTLINE

• Background
• Hardware Architecture
• Guidance Software
• Brief Detour
• Mission Applications
EXTREME CO-DESIGN

• None of the guidance system components pictured here were known when MIT was awarded the PGNCS contract.
• Everything was being developed essentially simultaneously.
MIT received a contract based on only a short, very general requirements statement.

Requirements started changing immediately and continued to change throughout the program.

Midway through, the AGC was redesigned:
- 8x memory, 2x faster numerics
- Block I (original) used in uncrewed test flights
- Block II (re-designed) in crewed flights

Effectively doubled software development effort.

ON TODAY ANNOUNCED THAT MIT’S INSTRUMENTATION LABORATORY HAS BEEN SELECTED TO DEVELOP THE GUIDANCE NAVIGATION SYSTEM OF THE
PROJECT APOLLO SPACECRAFT. APOLLO IS CAPABLE OF CARRYING THREE
MEN TO THE MOON AND BACK. MIT IS THE FIRST MEMBER OF THE APOLLO
TEAM TO BE CHOSEN. BIDS ARE NOW UNDERWAY FOR THE PRIME CONTRACTOR’S
JOB. IN ADDITION TO APOLLO THE INSTRUMENTATION LABORATORY WILL
ALSO DEVELOP THE GROUND SUPPORT AND CHECKOUT EQUIPMENT. CONTRACT
COVERING THE FIRST YEAR IS AN ESTIMATED $4 MILLION.
THE ESSENTIAL STEP
MIT SOFTWARE ENGINEERS NEEDED TO PERFORM
• Assemble a “flight program” & release it to Raytheon for rope core weaving
  • 2 months to weave the ropes; 2 months to install, test, run crew rehearsals, etc.
  • Lead engineer for an assembled flight program was called a “rope mother”
• For ~30 flights (uncrewed and crewed) each with unique guidance requirements
THE AGC HAD AN “APP” FOR THAT

• A lunar mission involved ~11 “burns” of the main engines

• For each unique maneuver, there was a major mode program to handle it
LUNAR LANDING MAJOR MODES

Lunar Module Descent Profile

Braking Phase: Program 63

Approach Phase:
Program 64

Terminal Descent: Program 66
EXAMPLE OF GUIDANCE ROUTINE SOFTWARE DEVELPMENT WORKFLOW – EPHEMERIS ROUTINES

• Knowing the position of the moon at any moment
  • Accurately (within a fraction of a mile)
  • Over a sufficiently long time period (2 weeks)
  • Minimizing time and space resource usage

• Where do you get the “ground truth” data to test validity?
  • Classically studied problem (Newton, Euler, Lagrange, Laplace, Delaunay…)
  • Observational data from astronomers (approx. distance in Earth radii)
  • Brown’s Lunar Theory (1897) and Tables of Motion of the Moon (1919)
  • Data from main-frame codes using a 1,600 term Fourier series approximation
POSSIBLE APPROACHES TO EPHEMERIS FOR AGC

• Store tabulated data and interpolate \(\rightarrow\) too much memory
• Use truncated Fourier series \(\rightarrow\) not accurate enough
• Solve 2-body problem (Earth – Moon system) \(\rightarrow\) not accurate enough
  • 3-body problem (Earth, Moon and Sun) is likely accurate enough \(\rightarrow\) not enough compute
• Polynomial fit to X, Y, Z positional data \(\rightarrow\) Accurate and memory efficient
POLYNOMIAL FIT OF EPHEMERIS DATA

• Accuracy:
  • Position to ~1 mile and velocity ~0.5 mph
  • Over a 2-week long period

• 8 double precision coefficients for each of X, Y and Z → 48 words
  • Did this go into fixed or erasable?
  • Raytheon manufactured contingency ropes for delays in launch

• How implemented…
  • Initially on Honeywell 1800 using MAC language
  • Accuracy, performance and coding confirmed
  • Re-coded in AGC Interpreter Language → 86 words
  • Tested on AGC all-digital simulator, then test-lab AGC unit

• Became a part of all assembled flight “ropes”
## Reading an AGC Program

<table>
<thead>
<tr>
<th>line</th>
<th>label</th>
<th>opcode</th>
<th>address</th>
<th>comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>0184</td>
<td>P63SPOT3</td>
<td>CA</td>
<td>BIT6</td>
<td>IS THE LR ANTENNA IN POSITION 1 YET</td>
</tr>
<tr>
<td>0185</td>
<td>EXTEND</td>
<td>EXTEND</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0186</td>
<td>RAND</td>
<td>CHAN33</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0187</td>
<td>EXTEND</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0188</td>
<td>BZF</td>
<td>P63SPOT4</td>
<td>BRANCH IF ANTENNA ALREADY IN POSITION 1</td>
<td></td>
</tr>
<tr>
<td>0189</td>
<td>CAF</td>
<td>CODE500</td>
<td>ASTRONAUT: PLEASE CRANK THE</td>
<td></td>
</tr>
<tr>
<td>0190</td>
<td>TC</td>
<td>BANKCALL</td>
<td>SILLY THING AROUND</td>
<td></td>
</tr>
<tr>
<td>0191</td>
<td>CADR</td>
<td>GOPERF1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0192</td>
<td>TCF</td>
<td>GOTOP00H</td>
<td>TERMINATE</td>
<td></td>
</tr>
<tr>
<td>0193</td>
<td>TCF</td>
<td>P63SPOT3</td>
<td>PROCEED</td>
<td>SEE IF HE'S LYING</td>
</tr>
<tr>
<td>0194</td>
<td>P63SPOT4</td>
<td>TC</td>
<td>BANKCALL</td>
<td>ENTER</td>
</tr>
<tr>
<td>0195</td>
<td>CADR</td>
<td>SETPOS1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0196</td>
<td>TC</td>
<td>POSTJUMP</td>
<td>OFF TO SEE THE WIZARD...</td>
<td></td>
</tr>
<tr>
<td>0197</td>
<td>CADR</td>
<td>BURNBABY</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## INFRASTRUCTURE SOFTWARE

<table>
<thead>
<tr>
<th>Program Name</th>
<th>Purpose</th>
<th>Size (AGC words)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Executive(^{25})</td>
<td>Priority-driven large/long-running process manager</td>
<td>~350</td>
</tr>
<tr>
<td>Waitlist(^{26})</td>
<td>Time-sequenced small/short-running process manager</td>
<td>~300</td>
</tr>
<tr>
<td>Down-Telemetry(^{29})</td>
<td>Transmit system data to ground</td>
<td>~200</td>
</tr>
<tr>
<td>Restart(^{30,31,32})</td>
<td>Error recovery and restart protection</td>
<td>~1225</td>
</tr>
<tr>
<td>Interpreter(^{27})</td>
<td>Space guidance domain-specific programming language interpreter</td>
<td>~2200</td>
</tr>
<tr>
<td>DSKY I/O(^{28})</td>
<td>Cockpit displays and keypad</td>
<td>~3500</td>
</tr>
<tr>
<td>Combined Total</td>
<td>22% of fixed memory</td>
<td>~7775</td>
</tr>
</tbody>
</table>
A PERFORMANCE PORTABILITY CHALLENGE

- Same code base operates optimally given variety of hardware configurations
Kalman Filtering and the Digital Auto Pilots (DAP)

- Prediction phase: Use idealized model for spacecraft motion.
- Comparison phase: Measured state from sensors compared with predicted.
- Control decisions based on the difference.
- Works well to smooth out noisy measurements from sensors.
- Does not require high sampling rate (2 second duty cycle).
- Performance Portability: Switch settings and pre-programmed parameters.

If the time required to drive $\dot{E}$ to the 5.625 deg./sec. rate be less than 0.020 sec., then no jets are fired.

The diagram illustrates the attitude error rate and the zones A, B, and C. It also shows the equations $-E - \frac{0.5}{a_{netneg}} \epsilon^2 + DB_1 = 0$ and $E - \frac{0.5}{a_{netpos}} \epsilon^2 + DB_2 = 0$.
• All-digital simulator: (AGC + Devices + Spacecraft "Environment")

• HW-800, 2 HW-1800, 2 IBM 360-75 (4,500 equiv. HW-1800 hours/month)

• Hybrid Simulator: Real AGC + Analog Computer (two story building)

• Test labs

• Flight simulators & Crew Rehearsals

• Actual flight testing in mission plans

Figure 8. Computer usage by the Draper Laboratory Apollo effort.
<table>
<thead>
<tr>
<th>Project</th>
<th>1965 ($M)</th>
<th>2019 ($M)</th>
<th>2019 $M/yr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apollo (10 yr)</td>
<td>25000</td>
<td>203000</td>
<td>2030</td>
</tr>
<tr>
<td>PGNCS (10 yr)</td>
<td>600</td>
<td>~5000</td>
<td>500</td>
</tr>
<tr>
<td>Software (5 yr)</td>
<td>60</td>
<td>~500</td>
<td>100</td>
</tr>
</tbody>
</table>
“The need for formal validation rose with the size of the software. Programs of 2,000 words took between 50 and 100 test runs to be fully debugged, and full-size flight program took from 1,000 to 1,200 runs.”
“In the early stages, there were no programmers. Instead, engineers and scientists learned the techniques of programming.

It was believed that competent engineers could learn programming more easily than programmers could learn engineering.”
“Throughout much of the Apollo effort, MIT experienced difficulty in estimating the time and effort required to design, test and verify successive mission programs.”
"SOFTWARE ENGINEERING"

• Margaret Hamilton, lead developer of Lunar Module flight program introduced this term...

"...to bring the software [effort] legitimacy so that it and those building it would be given due respect"
“No one doubted the quality of the software…It was the process used in development that caused great concern.

Five lessons were identified:

1. up-to-date documentation is crucial
2. verification must proceed through several levels
3. requirements must be clearly defined and carefully managed
4. good development plans should be created and executed
5. more programmers do not mean faster development
OUTLINE

• Background
• Hardware Architecture
• Guidance Software
• Brief Detour
• Mission Applications
A BRIEF DETOUR

HISTORICAL CONTEXT

OFTEN GLOSSED OVER

OR TOTALLY IGNORED
Valentina Tereshkova – 3 days in orbit, 1963
A BRIEF DETOUR: WOMEN AND COMPUTERS

• 1640-1950: “Computer”
• Tedious calculation was “women’s” work (“kilogirl”)
• 1950-1960: Computers were almost exclusively women

Harvard Observatory - 1890
Oak Ridge - 1942
Los Alamos - 1943
Bletchly Park - 1944
Jet Propulsion Lab - 1953
Langley, West Computing Group (1958)

Female Share of Bachelor’s Degrees in Computer Science, 1970-2016

Source: US Department of Education
A BRIEF DETOUR
WOMEN
IN THE AGC PROJECT

Margaret Hamilton, Phyllis Rye,
Saydean Zeldin, Elain Denniston
A BRIEF DETOUR
PEOPLE OF COLOR IN THE AGC PROJECT

William Mallory
Ramon Alonso
Robert Pinckney
A BRIEF DETOUR:
WERNHER VON BRAUN

• Creator of V2 Rocket
• Member of NAZI Party; arrested for suspicion
• Captured and brought to US with ~1,600 others in 1945
• Led development of F1 engine and Saturn booster
• Championed racial integration in Wallace’s Alabama
OUTLINE

• Background
• Hardware Architecture
• Guidance Software
• Brief Detour
• Mission Applications
**USER INTERFACE**

- **VERB – NOUN**
- **3, 5 char line display**
- **Indicator Lights**
- **Two in CM, one in LM one in Huston**
APOLLO 1: FIRE DURING CREW REHEARSAL

• 100% oxygen + 500 cm² of Velcro + a spark
• Grounded all crewed Apollo flights for 21 months
• Did not involve the AGC
• Had a profound impact on AGC software development

TO: See list
FROM: FM/Deputy Chief, Mission Planning
TO: Analysis Division
SUBJECT: Altitude and velocity limits imposed by the spacecraft computer program on the AS-503 mission

As you know, we are currently figuring on using the AS-278 spacecraft computer programs for AS-503. Ed Copps called me the other day to state that the orbital integration routines in the AS-278 program are scaled such that they will only work for altitudes less than about 5,400 nautical miles above the surface of the earth and velocities no greater than about 32,700 feet per second. (I am told the maximum values to be encountered in a nominal mission are about 3,900 nautical miles and 29,500 feet per second). He was looking for reassurance that this scaling would not present a constraint on the AS-503 mission, and I told him that I didn't think it would but I would check here at MSC. In the meantime, MIT is proceeding, assuming that these limits are not unacceptably restrictive for the AS-503 mission. If anyone knows a reason why this is not satisfactory, please let me know immediately.

Howard W. Tindall, Jr.
APOLLO 11

• Russian Luna 15
• Bad communications
• Program alarms & restarts
• Boulder Field
• Ascent engine arm CB
• Gimbal lock at rendezvous and switch to AGS
APOLLO 14

• Abort switch hack
APOLLO 8

• Entering a pre-launch program, P01 while in flight can crash the AGC

• Lovell practices using the space sextant, accidentally enters P01 instead of star 01.

• Corrupts some guidance parameters in AGC erasable memory.
APOLLO 9

• First use of Erasable Memory Program (EMP) in crewed flight

• LM descent engine test configuration
APOLLO 10

- Barbeque mode troubles
- Full-up lunar descent abort test
- AGS in "AUTO" not "ATT-HOLD"

(video)
APOLLO 12

• Lightning strike
• Landing accuracy
APOLLO 13

- What-if thinking
- Three burns to get home
APOLLO 15

• Landing over lunar mountain range

• Added a simple terrain model for landing radar
Computing was an essential tool in all aspects of Apollo project:
- Simulation and modeling used in all major vehicle components.
- Digital and Analog computers for Training simulators.
- Real time computing complex (RTCC) for missing planning, tracking.
- Computing deficiencies are a key reason Russia was unable to match US.
- Apollo both drove innovations in computing and benefited from them.
ARGON-11C: RUSSIAN GUIDANCE COMPUTER

- Hybrid Integrated Circuits
- 14 bit data words, 17 bit op-code
- 128 words RAM / 4 k-words ROM (9 kb)
- 5.2 K-Flops
- Triple redundant logic w/voting
- 34 kg / 75 watts

**Flops/x Computing Metrics Comparison**

- flop/s/KB
- flop/s/watt
- flop/s/kg
- flop/s/m3

Legend:
- AGC
- IBM 360-20
- Argon-11C
RESOURCES LINKS

- bssw.io blog post
- Mercury 13 (Netflix doc)
- AGC Restoration
- AGC Source Code on GitHub
- Virtual AGC Project
- Ultimate AGC Talk
- Spaceflight Computing History

- AGC Software Cost Model
- Hidden Figures (the book)