Telemetry, Command, Data Processing & Handling

Space System Design, MAE 342, Princeton University
Robert Stengel

- System definition
- Computer architecture
- Components
- Data coding
- Fault tolerance and reliability
- Hardware and software testing

A Typical Space/Ground Information System

Wertz and Larson
Defining the System

- Identify the spacecraft bus and payload operational modes
- Allocate top-level requirements for the computer system
- Define sub-system interfaces
- Specify baseline computer system
  - Define computer system’s operational modes and states
  - Functionally partition and allocate computational requirements to
    - spacecraft sub-systems, hardware, or software
    - ground station
  - Analyze data flow
  - Evaluate candidate architectures
  - Select basic architecture
  - Develop baseline system configuration
- Do we need a new computing system, or can we use an old system that is already certified?
Defining the System

Requirements Definition

- What must the system do?
- Why must it be done?
- How do we achieve the design goal?
- What are the alternatives?
- What sub-systems perform specified functions?
- Are all functions technically feasible?
- How can the system be tested to show that it satisfies requirements?
Telecommand Waveforms

Pulse Code Modulation (PCM)

- **NRZ-L (Non-Return-to-Zero, Level)**
  - A signifies '1'
  - B signifies '0'
- **SP-L (Split-Phase, Level)**
  - '1' signified by A during 1st half, B during the 2nd half
  - '0' signified by B during 1st half, A during 2nd half
- **NRZ-M (Non-Return-to-Zero, Mark)**
  - Level change from A to B or B to A signifies '1'
  - No level change signifies '0'

Classification of Telemetry Data

- **Housekeeping data**
  - Temperatures, pressures, voltages, currents, ...
- **Attitude and acceleration data**
  - Sun sensors, star sensors, gyros, accelerometers, ...
- **Payload data**
  - Mission dependent
  - Wide range of data rates, bandwidth, criticality, ...

https://en.wikipedia.org/wiki/Spectrogram
Digital vs. Analog Modulation

- **Analog**
  - Amplitude modulation conserves bandwidth
  - Frequency modulation spreads information bandwidth over larger RF bandwidth

- **Digital**
  - Pulse-code modulation (particularly phase-shift keying) uses RF power most efficiently

**Link Budget for a Digital Data Link**

\[
\frac{E_b}{N_o} = \frac{P_t L_i G_t L_s L_a G_r}{kT_s R} 
\]

- **Link budget design goal** is to achieve satisfactory \(E_b/N_o\) by choice of link parameters

\[
E_b = \frac{S}{N} BW \\
N_o = \frac{S}{N} R 
\]

- \(P_t = \text{transmitter power}\)
- \(L_i = \text{transmitter-to-antenna line loss}\)
- \(G_t = \text{transmit antenna gain}\)
- \(L_s = \text{space loss}\)
- \(L_a = \text{transmission path loss}\)
- \(G_r = \text{receiver antenna gain}\)
- \(k = \text{Boltzmann's constant}\)
- \(T_s = \text{system noise temperature}\)
**Bit Error Rate vs. $E_b/N_0$**

- Goal is to achieve lowest bit error rate (BER) with lowest $E_b/N_0$
- Implementation losses increase required $E_b/N_0$
- Link margin is the difference between the minimum and actual $E_b/N_0$
- BER can be reduced by error-correcting codes
  - Number of bits transmitted is increased
  - Additional check bits allow errors to be detected and corrected

**Performance of Coding/Decoding Methods**

<table>
<thead>
<tr>
<th>Code</th>
<th>Decoding Method</th>
<th>Coding Gain at $P_e = 10^{-5}$</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block</td>
<td>Majority—hard decision</td>
<td>1.5–3.5 dB</td>
<td>Simple</td>
</tr>
<tr>
<td>Block BCH</td>
<td>Algebraic</td>
<td>1.5–4 dB</td>
<td>Complex</td>
</tr>
<tr>
<td>Convolutional</td>
<td>Threshold—hard decision</td>
<td>1.5–3 dB</td>
<td>Fairly simple</td>
</tr>
<tr>
<td>Convolutional</td>
<td>Viterbi—soft decision</td>
<td>4.5–5.5 dB</td>
<td>Fairly complex</td>
</tr>
<tr>
<td>Convolutional</td>
<td>Sequential—soft decision</td>
<td>5.7 dB</td>
<td>Fairly complex</td>
</tr>
<tr>
<td>Concatenated</td>
<td>Viterbi + algebraic</td>
<td>6.7–7.5 dB</td>
<td>Very complex</td>
</tr>
<tr>
<td>block-convolutional</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Turbo</td>
<td>Maximum à posteriori</td>
<td>8.5–9.4 dB</td>
<td>Fairly complex</td>
</tr>
</tbody>
</table>

Note: Theoretical BPSK requires $E_b/N_0 = 9.6$ dB for $P_e = 10^{-5}$. $P_e$ = Bit Error Rate (BER), BPSK = Binary phase-shift keying.

**Coding gain is net $S/N$ improvement provided by adding check bits**
Telemetry List and Data Format

- For each item
  - Signal ID, data type, required accuracy, sampling rate
- PCM message format
  - e.g. Eight frames, each with 64 8-bit words
  - Fixed synchronization code
  - Frame ID channel
- Specification of data channels
  - Housekeeping, “prime”, commutation

Telemetry Data Encoding

- Analog data
  - Filtering
  - A/D
  - Multiplexer, sub-multiplexer
- Digital bi-level data
  - On-off
- Digital serial data
  - Word length
  - PCM mode
Multiplexing

- Analog Modulation
  - AM, FM, PM, SSB (single sideband), …
- Circuit Mode (circuit mode)
  - TDM, FDM, Polarization, …
- Statistical Multiplexing (variable bandwidth)
  - Packet switching, Dynamic TDMA, Spread Spectrum, …

https://en.wikipedia.org/wiki/Multiplexing

Data Formatting

Packet Telemetry Data Flow

- Application Process Layer
- System Management Layer
- Packetization Layer
- Segmentation Layer
- Transfer Layer
- Coding Layer
- Physical Layer
Parity and Error Detection

- $n$-bit word = $(n - 1)$ bits of data plus a parity bit (e.g., ASCII 8-bit word for 7-bit code)
- Parity bit is computed (XOR gates) so that the number of "ones" in the word is even (or odd)
- Word is transmitted
- Error in one bit of the word is detected if the number of "ones" is not even (or odd)

<table>
<thead>
<tr>
<th>Action</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>A wants to transmit</td>
<td>1001</td>
</tr>
<tr>
<td>A computes parity bit</td>
<td>$1^0^0^1 = 0$</td>
</tr>
<tr>
<td>A adds parity bit and sends</td>
<td>10010</td>
</tr>
<tr>
<td>B receives</td>
<td>10010</td>
</tr>
<tr>
<td>B computes overall parity</td>
<td>$1^0^0^1^0 = 0$</td>
</tr>
<tr>
<td>B reports correct transmission</td>
<td>after observing expected even result.</td>
</tr>
</tbody>
</table>

- If error is detected, B requests re-transmission from A
- Error-correcting codes as in telemetry (convolution and block codes, memory refreshing, redundancy)

Error-Control Coding

- Typical BER: $1:10^5$
- Division by a polynomial
  - e.g., $x^{16} + x^{12} + x^5 + 1$
  - Send 16-bit remainder
- Ground station
  - Divide by same polynomial
  - If 16-bit remainder not the same, re-send
- Forward error correction
  - Various codes, e.g., ...

http://www.ccs.neu.edu/home/raj/Courses/6710/S10/Lectures/Coding.pdf
Telecommand User Interface

- Low-level on-off commands
- High-level on-off commands
- Proportional commands
- Telecommand standards

Memory Load Command Frame Structure
Communications Techniques & Protocols

- Ranging
- Advanced Orbiting Systems
- Proximity Links
- Protocols
  - Store-and-forward networking
  - Continuous file delivery
  - Negative automatic report queuing
  - Proxy transfer facilities
  - Graceful suspend/resume
  - Garbage clearance
  - File manipulation

On-Board Data Handling and Processing
Instruments, Experiments, and Sensors

Compression and Storage

- **On-line data compression**
  - Block-adaptive quantization
  - first-in/first-out buffering
- **Off-line data compression**
  - Lossless or lossy compression
- **Data storage**
  - Blocks and files
Data Downlinks

- Data-handling function RF transmit chain
  - Data routing
  - Buffering
  - Formatting
  - Carrier modulation
  - Amplification
  - Transmission

- Modulation techniques
  - QPSK
  - Amplification
  - Link layer
  - Link availability

Electronics Technology

- Radiation hardness
- Single-event upsets
- CMOS latch-up
- Parity
- Error detection and correction
- Triple modular redundancy
- Multiple execution
- Fault roll-back
  - repeat the function if error is sensed

- Fault roll-forward
  - correct the error and move on

- Watchdog timers
  - detect unusual execution time for program function
  - force a restart if fault is detected

- Improper sequence detection

- Hardware vs. software errors
Radiation Hardness and Single-Event Upsets

- Radiation degrades semiconductor devices
- Ionization due to Gamma rays may trap charges in devices, altering their function
  - Can produce a single-event upset
- Random and age-related failures must be anticipated
  - Shielding
  - Radiation-hardened dielectrics
- Single-event upset (SEU)
  - Radiation flips a bit in data or instruction
- CMOS latch-up
  - Large transient current flow may destroy the device
  - Build in a circuit breaker that shuts off current before damage is done

Triple Modular Redundancy: Hardware

- Parallel hardware implementation for fault tolerance
  - Each sensor, computer, or actuator is replicated three times
  - Multiple execution
  - Voting logic compares the three versions of each output and chooses the version
    - transmitted by two (or all three),
    - middle value, or
    - average value
  - Cost and maintenance implications
Triple Modular Redundancy: Software

- Software implementation for serial data transmission
  - Each word is transmitted three times
  - Voting logic compares the three versions and chooses the version transmitted by two (or all three)
  - Serial data transfer rate is slowed by a factor of three

Reliability

Probability of Success during Period of Operation

\[ R(t) : \text{Probability of success} \]
\[ P(t) : \text{Probability of failure} \]

\[ R(t) = 1 - P(t) \]
Reliability Assessment

- Tools for reliability assessment: Testing
  - Levels of test: development, qualification, acceptance, function
  - Destructive physical analysis
- Tools for reliability assessment: Analysis
  - Statistical distributions
  - Statistics, regression, and inference
  - Fault trees and reliability prediction
  - Confidence level or interval

reliability of a single string

Reliability of a single string = product of individual reliabilities

\[ R_{1-n}(t) = R_1 R_2 \ldots R_n \]
Reliability of Parallel (Redundant) Components

Probability of failure of parallel components = product of individual probabilities

\[ P_{13}(t) = P_1(t)P_2(t)P_3(t) \]

\[ R(t) = 1 - P(t) \]

\[ R_{13}(t) = 1 - P_{13}(t) = 1 - P_1(t)P_2(t)P_3(t) \]

Reliability of a Switched Redundant System

Reliability of the switch must be considered

\[ R_{\text{system}}(t) = R_1(t) \{ 1 - [1 - R_2(t)][1 - R_3(t)R_2'(t)] \} R_3(t) \]

\[ = R_1(t) \{ 1 - P_2(t)P_{S2'}(t) \} R_3(t) \]
Reliability of a String of Parallel Components

\[ R_{\text{system}}(t) = \sum_{x=r}^{n} \binom{n}{x} R^x (1 - R)^{n-x} \]

Binomial coefficient

\[ \binom{n}{x} = \frac{n!}{x!(n-x)!} \]

\( r = \# \text{ of elements in a parallel component that must survive for operation} \)

Reliability of Parallel Strings

\[ P_{1n}(t) = P_1(t)P_2(t)...P_n(t) \]

\[ R(t) = 1 - P(t) \]

\[ R_{1n}(t) = 1 - P_{1n}(t) = 1 - P_1(t)P_2(t)...P_n(t) \]
Spacecraft Computers

- Spacecraft computing hardware; analogous to Macs and PCs, but
  - Must be ultra-reliable
  - A few generations behind the state-of-the-art
- Memory
- Input/output
- Fault tolerance
- Special-purpose peripherals

Hardware, Software, and Documentation

- Hardware
  - Hardware Configuration Item (HWCI)
  - Computer Board
  - Computer Chip Set/Analog Devices/Logic Components/Discrete Components

- Software
  - Computer Software Configuration Items (CSCIs)
  - Computer Software Component (CSCs)
  - Computer Software Unit

- Documentation
  - Requirements Specification
  - Design Documents
  - Detailed Design Documents
  - Interface Control Documents (ICDs)
System states must be consistent with allocated requirements and with spacecraft’s and ground station’s concepts of operation (“conops”)

**Computer System State Diagram**

1. Initialize
2. Normal Operations
3. Contingency Operations
4. Planned Operations
5. Planned or Emergency Shutdown
6. Off
7. Power On
8. Failure
9. Another Failure

**Computer System Functional Partitioning**

- **Group functions**
  - Similarity
  - Complexity
  - Processing type
  - Urgency
  - Timing and throughput
  - External interface
  - Data storage req’t
  - Human participation
  - Flight safety

- **Space/ground tradeoffs**
  - Autonomy
  - Time criticality
  - Downlink bandwidth
  - Uplink bandwidth

- **Hardware/software tradeoffs**
  - Special-purpose h/w
  - Algorithmic complexity
Computer Architecture

- **Central processor**
  - Point-to-point interfaces, central processor and devices
  - Dedicated wiring and software
- **Bus**
  - Processors and devices communicate via a bus
  - Protocol software for transmission control
  - Standard interfaces
- **Ring**
  - Established arbitration (e.g., token-passing) for bus control
- **Instruction set**
  - Assembly language
  - Higher-order language

Computer Resource Estimation

- **Defining processing tasks**
  - Software requirements specification
  - Interface requirements specification
  - Principal classes
    - Control systems
    - System management
    - Mission data management
    - Operating system
      - Utilities
      - Built-in test
- **Estimating software size and throughput**
  - Processor instruction sets
  - Processor clock speeds
  - Historical data for similar processing tasks
  - Preliminary coding of example tasks
Development Phase Issues

- **Hardware selection**
  - Performance, cost, availability, vendor competition
- **Developmental environment**
  - Software languages, tools for coding, compiling, and testing
  - Host/target machines
- **Development costs**
  - Mission life cycle
- **Development tools and methodologies**
  - Specification and analysis aids
  - Design aids
  - Traceability analysis
  - Documentation aids

---

**Typical Life-Cycle Cost Distribution**

- Detailed Implementation: 35%
- Design, Code, Test: 45%
- Analysis and Early Design: 20%
- System Maintenance: 10%

---

**Computer System Integration and Test**

- **Software Verification**
  - Unit Test
  - Module Test
  - Stand-alone Functional Test
- **Hardware Verification**
  - Component Test
  - Acceptance Test
  - Ground-Based and Space-Based Systems
  - System Integration
  - System Testing
  - On-orbit Calibration and Mission Success

---

*Wertz and Larson*
Computer Memory

• Read-only memory (ROM)
  – Non-volatile
  – Non-alterable
  – Store critical programs
  – EAROM, EEROM, EEPROM

• Flash memory (special EEPROM)

• Random-access memory
  – Volatile

• Special-purpose memory
  – Multi-port
  – Cache
  – Multiply-accumulate

• Disk
  – Magnetic
  – CD, DVD

Computer Input/Output

• Ports
  – Data transfer between processor and bus
    • Serial I/O ports
    • Parallel I/O ports
    • I/O-mapped ports
    • Memory-mapped ports

• Direct memory access
  – Sub-systems access memory without going through the processor for large blocks of data or high data rate

• Multi-port memory
  – Simultaneous data access by two or more devices

• Interrupts
  – May be generated by a timer or an event, changing processor function
  – Synchronize activity of multiple processors
  – Context switching and storage

• Timers
• Bus interface
Special-Purpose Peripherals (Signal-Processing Hardware)

Data acquisition

- Logarithmic and data compression
  - Rounding, filtering, coding, channel capacity, probability, incremental values, ...
- Frequency domain transformation
  - Time domain -> frequency domain
  - Fourier transform, inverse transform, wavelets
- Power/energy spectrum accumulation
- Image processing
- Digital/analog conversion
Apollo GNC Software
Testing and Verification

• Major areas of testing
  – Computational accuracy
  – Proper logical sequences
• Testing program
  – Comprehensive test plans
  – Specific initial conditions and operating sequences
  – Performance of tests
  – Comparison with prior simulations, evaluation, and re-testing
• Levels of testing
  – 1: Specifications coded in higher-order language for non-flight hardware (e.g., PCs)
  – 2: Digital simulation of flight code
  – 3: Verification of complete programs or routines on laboratory flight hardware
  – 4: Verification of program compatibility in mission scenarios
  – 5: Repeat 3 and 4 with flight hardware to be used for actual mission
  – 6: Prediction of mission performance using non-flight computers and laboratory flight hardware

Apollo GNC Software
Specification Control

• Guidance System Operations Plan (GSOP)
  – NASA-approved specifications document for mission software
  – Changes must be approved by NASA Software Control Board
• Change control procedures
  – Program Change Request (NASA) or Notice (MIT)
  – Anomaly reports
  – Program and operational notes
• Software control meetings
  – Biweekly internal meetings
  – Joint development plan meetings
  – First Article Configuration Inspection
  – Customer Acceptance Readiness Review
  – Flight Software Readiness Review
Apollo GNC Software Documentation and Mission Support

- **Documentation generation and review**
  - Functional description document: H/W-S/W interfaces, flowcharts of procedures
  - Computer listing of flight code
  - Independently generated program flowchart
  - Users’ Guide to AGC

- **Mission support**
  - Pre-flight briefings to the crew
  - Personnel in Mission Control and at MIT during mission

Apollo Guidance Computer

- Parallel processor
- 16-bit word length (14 bits + sign + parity)
- Memory cycle time: 11.7 $\mu$sec
- Add time: 23.4 $\mu$sec
- Multiply time: 46.8 $\mu$sec
- Divide time: 81.9 $\mu$sec

- Memory (ceramic magnetic cores)
  - 36,864 words (ROM)
  - 2,048 words (RAM)
- 34 normal instructions
- Identical computers in CSM and LM
- Different software (with many identical subroutines)
- 70 lb, 55 w

- There were NO computer hardware failures during Apollo flights
Some Flight Computer Variations

### Technical Specifications

<table>
<thead>
<tr>
<th>On-Board Computer</th>
<th>VME Compatible On-Board Computer</th>
<th>Single-String Controller</th>
<th>C&amp;DN Processor Controller</th>
<th>Mechanism Controller</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instruction Set</td>
<td>PowerPC</td>
<td>PowerPC</td>
<td>PowerPC</td>
<td>PowerPC</td>
</tr>
<tr>
<td>Peak Throughput</td>
<td>480 MIPS</td>
<td>480 MIPS</td>
<td>480 MIPS</td>
<td>480 MIPS</td>
</tr>
<tr>
<td>Processor RAM</td>
<td>256 MB</td>
<td>64 MB</td>
<td>256 MB</td>
<td>256 MB</td>
</tr>
<tr>
<td>Non-Volatile Program Storage</td>
<td>144 MB</td>
<td>64 MB</td>
<td>144 MB</td>
<td>144 MB</td>
</tr>
<tr>
<td>DMA Channels</td>
<td>5</td>
<td>5</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>System Bus</td>
<td>2.1553</td>
<td>1.1553</td>
<td>2.1553</td>
<td>2.1553</td>
</tr>
<tr>
<td>Other Interfaces</td>
<td>3-High Speed Serial, VME Bus</td>
<td>3-High Speed Serial, VME Bus</td>
<td>Analog and Digital Interfaces</td>
<td>Tolerates and Telemetry, Analog and Digital Interfaces</td>
</tr>
<tr>
<td>Typical Size (inches)</td>
<td>9 x 6.6 x 3.0 (3 Modules)</td>
<td>Single-Width 6U-60 I/O E1591.2</td>
<td>9 x 6.6 x 3.0 (3 Modules)</td>
<td>9 x 6.6 x 6.3 (15 Modules)</td>
</tr>
<tr>
<td>Typical Power (Watts)</td>
<td>15 - 30</td>
<td>10 - 25</td>
<td>10 - 35</td>
<td>Application dependent</td>
</tr>
<tr>
<td>Weight (Kg)</td>
<td>3.5 Kg</td>
<td>1.2 Kg</td>
<td>5 Kg</td>
<td>9 Kg</td>
</tr>
</tbody>
</table>

*Exclusive of mounting flange

Pervasive use of VMEbus in spacecraft computers

### RAD750 Single Board Computer

- Mars Curiosity Rover, Mars/Lunar Reconnaissance Orbiters, Deep Impact, ...

- Produced: From 2001 to Present
- Designed by: IBM
- Manufacturer: BAE
- Max. CPU clock rate: 110 MHz to 200 MHz
- Min. feature size: 250 nm to 150 nm
- Instruction set: PowerPC v.1.1
- Microarchitecture: PowerPC 750
- Cores: 1
- Application: Radiation hardened
RAD5545 Single Board Computer

- **Designed by**: IBM, Freescale
- **Manufacturer**: BAE
- **Speeds**: 5200 MIPS, 3700MFLOPS
- **Min. feature size**: 45 nm
- **Instruction set**: Power ISA, v 2.06
- **Microarchitecture**: PowerPC e5500, VPX backplane
- **Cores**: 4
- **Application**: Radiation hardened

Fault Tolerance Requirements for Overall System

- Failure at a single point should not cause failure of entire system
- It should be possible to isolate the effects of a single component failure
- It should be possible to contain individual failures to prevent failure propagation
- Reversionary modes should be available ("fail-safe" design)
  - backup software
  - backup hardware
Next Time:
Ground Segment

Supplemental Material
Astronaut Interface With the AGC

- Computer Display Unit or Display/Keyboard
- Sentence
  - Subject and predicate
  - Subject is implied
    - Astronaut, or
    - GNC system
  - Sentence describes action to be taken employing or involving the object
- Predicate
  - Verb = Action
  - Noun = Variable or Program

See http://apollo.spaceborn.dk/dsky-sim.html
And http://www.ibiblio.org/apollo/ for simulation

Verbs and Nouns in Apollo Guidance Computer Program

- Verbs (Actions)
  - Display
  - Enter
  - Monitor
  - Write
  - Terminate
  - Start
  - Change
  - Align
  - Lock
  - Set
  - Return
  - Test
  - Calculate
  - Update

- Selected Nouns (Variables)
  - Checklist
  - Self-test ON/OFF
  - Star number
  - Failure register code
  - Event time
  - Inertial velocity
  - Altitude
  - Latitude
  - Miss distance
  - Delta time of burn
  - Velocity to be gained

- Selected Programs (CM)
  - AGC Idling
  - Gyro Compassing
  - LET Abort
  - Landmark Tracking
  - Ground Track Determination
  - Return to Earth
  - SPS Minimum Impulse
  - CSM/IMU Align
  - Final Phase
  - First Abort Burn
A Little AGC Digital Autopilot Code

Space Shuttle Quintuply Redundant Flight Control Computers

- Five identical IBM AP-101 computers
  - Magnetic core memory later upgraded to semiconductor memory
  - Primary system: 4 parallel computers with identical coding and complex redundancy management software
  - Backup system: 5th computer with independent coding of the same functions
  - Concern for generic software failures
  - HAL/S programming language
Space Shuttle Quintuply Redundant Flight Control Computers

IBM AP-101 Input/Output Processor and Central Processing Unit