Spacecraft Computer Systems

Colonel John E. Keesee

Overview

- Spacecraft data processing requires microcomputers and interfaces that are functionally similar to desktop systems
- However, space systems require:
  - Low power, volume, and mass
  - High reliability and fault tolerance

Outline

- Definitions
- Computer system specification
- Estimating throughput and processor speed requirements
- Computer selection
- Memory
- Mass storage
- Input/Output
- Radiation hardness
- Fault tolerance
- Error detection and correction
- Integration and test

Definitions

- Embedded system – A built-in processor, providing real-time control as a component of a larger system, often with no direct user interface
- Real-time processing – Handling or processing information at the time events occur or when the information is first created

Definitions (continued)

- Hard real-time – Requires precise timing to achieve correct results, where missing at time boundary has severe consequences
- Soft real-time – Requires that the tasks be performed in a timely manner, where missing a time boundary results in degraded but continuous performance

Definitions (continued)

- Operating system software – Manages the computer’s resources, such as input/output devices, memory, and scheduling of application software
- Application software – Mission-specific software which does work required by the user or the mission rather than in support of the computer

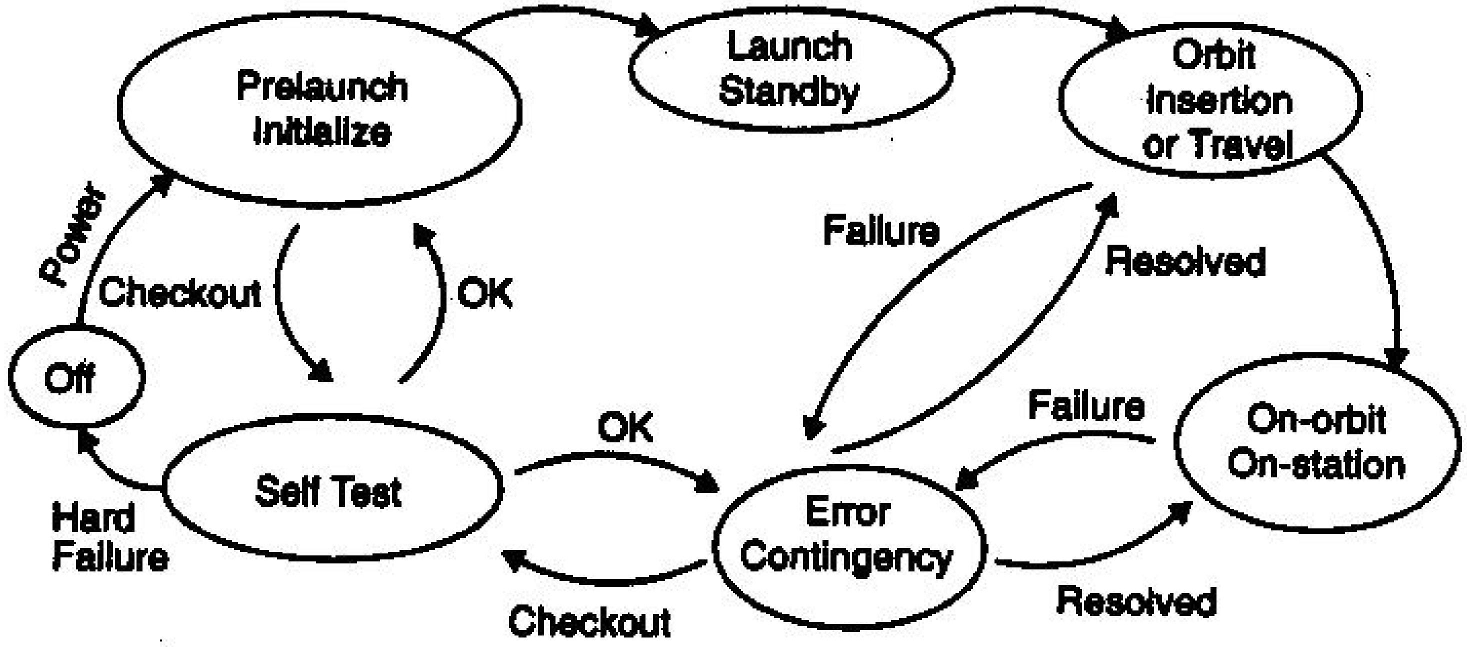
Computer System Specification

1. Allocate mission and system requirements to computer systems
2. Define the computer system’s operational modes and states
3. Functionally partition and allocate the computational requirements to space or ground, payload or spacecraft, individual systems, and to hardware or software

Computer System Specification

4. Evaluate internal and external interfaces
5. Select the baseline architecture
6. Form the baseline system specification

State Transition Diagram



Functional Partitioning

Perform processing in space	Perform processing on ground
Perform processing in hardware	Perform processing in software
Allocate processing between spacecraft bus and payloads	Do not allocate processing between spacecraft bus and payloads
Allocate processing along organizational lines	Do not allocate processing along organizational lines

Estimating Throughput and Processor Speed Requirements

- SMAD example
  - Operator types 100 words (600 characters)/min
  - 256 keyboard states implies an 8-bit data word
    - Input data stream = 4800 bits/min
  - Each character requires 10 instructions to process
  - Each word requires 100 instructions to process
    - 16,000 instructions/min = 267 instructions/sec

Estimating Throughput and Processor Speed Requirements

- SMAD example continued
  - Each instruction takes five clock cycles to read and one to execute
    - CPU must operate at 96,000 cycles/min = 1600 cycles/sec = 1.6 kHz
    - To store input data and transfer 8-bit instructions to the CPU requires 80 bits/sec for data plus 2136 bits/sec for instruction fetches = 2216 bits/sec transfer rate
    - In 24 hours the system will store 6.9 Mbits of data in addition to its 880 bits of stored instructions

Estimating Software Size and Throughput Requirements

- Control system tasks
- System management tasks
- Mission data software
- Operating system software

•SMAD Tables 16-13, 16-14, and 16-15

•Analogy

•Bottom up analysis

•Parametric

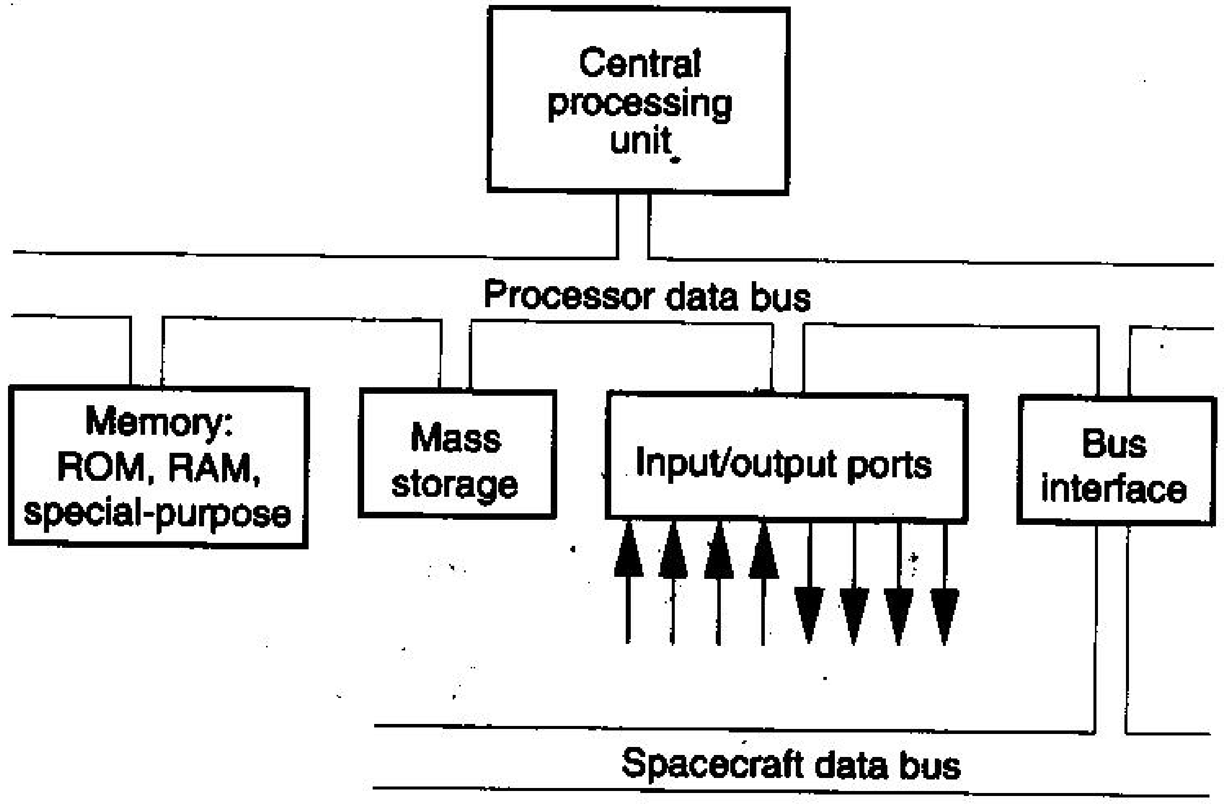
Computer Selection

- Computation Rate
  - For Reduced Instruction Set Computer the rate is about 1.5 times the clock rate
- Address space
  - 16-bit computers usually address 64 K words
  - 32-bit computers usually address 4 G words
- Built-in hardware functions
  - Floating point and transcendental functions
  - Direct Memory Address
- See SMAD Table 16-17

Processing Architectures

- Central Unit
  - Single processor or one of the processors is designated the master that coordinates all the others
- Distributed processing system
  - Multiprocessor units where any one can assume the role of master, or where executive tasks are shared by all processors
    - Difficult to design
    - Tolerates faults well

Typical Spacecraft Data Processing System



Memory

- Read Only Memory
  - Programs and start-up instructions
  - Slow, not volatile
- Random Access Memory
  - Volatile but fast
- Special purpose memory
  - Cache, multi-port, fast multiply accumulate, others

Mass Storage

- Disks
  - Mechanical systems -> reliability concerns
  - Angular momentum and vibration
- Digital tape
  - Highly reliable
  - Momentum, start/stop torques, vibration are concerns
  - Slow data access

Mass Storage (continued)

- Bubble memory
  - Solid state, non-volatile
  - Small cylindrical region of magnetization in a planar substrate (garnet)
  - Persists in a magnetic bias field
  - Patterns in magnetic permalloy and rotating magnetic field cause bubbles to move under read/write head
  - High mass, power dissipation, unwanted magnetic fields, but no moving parts

Mass Storage (continued)

- Integrated Circuits (RAM)
  - High power, cost
  - High speed and density
- Magneto-Optical disks
  - When the magnetization vector is pointed toward the light source reflected light is blocked by a polarization filter
  - Not blocked when vector is pointed away from light source

Mass Storage (continued)

- Magneto-Optical disks (continued)
  - To erase or write a laser heats the spot and a magnetic field changes the direction of the vector
  - High storage capacity but not used in space yet
  - Disadvantages similar to disks

Input/Output

- Ports
  - Serial I/O ports
  - Parallel I/O ports
  - I/O mapped ports
  - Memory mapped ports
- Direct Memory Access (cycle stealing)
- Multi-port memory

Input/Output

- Interrupts
  - Priorities
  - Context switching
- Timers
- Bus interface

Radiation Hardness

- Effects usually based on total dose of radiation on semiconductor chips
  - Digital logic slows down
  - Op Amp offset voltages change
  - Current drive capability is reduced
  - Power dissipation increases
- Natural and man-made sources

Radiation Effects

- Gamma rays leave a trail of charged particles in their wake
  - Measured in Rad (Si)
- High energy neutrons cause structural damage in solid state materials
- High energy charged particles generate clouds of electrical charges
  - Charge on gates of metal-oxide semiconductors can change its state – single event upset
  - CMOS can latch up across the power supply

Error Detection

- Parity checking
  - Add an extra bit to the word so that the number of logic ones is always even
  - When a word is read out of memory the logic ones are counted
- Checksum of many words
  - Logic ones are summed modulo 2^n to form n-bit memory checksum
  - Checksum is calculated before transmission and sent with the data
  - Checksum is calculated upon receipt

Fault Tolerance

- Redundancy
  - Duplicate equipment
  - Backup with different approaches
  - Backup with ground systems
  - Data bus delivery systems
  - Cross strap equipment

Fault Tolerance (continued)

- Distributed processing
  - Allocates software processing to any one of several processors depending on
    - Mission phase
    - Hardware availability
    - Subsystem failures

Fault Tolerance

- Multiple execution
  - Master/slave CPUs
  - Separate algorithms
  - Certification trails
- Fault roll-back – return to a saved context saved before the fault occurred
- Watchdog timers
- Improper sequence detectors

Hardware Reliability Techniques

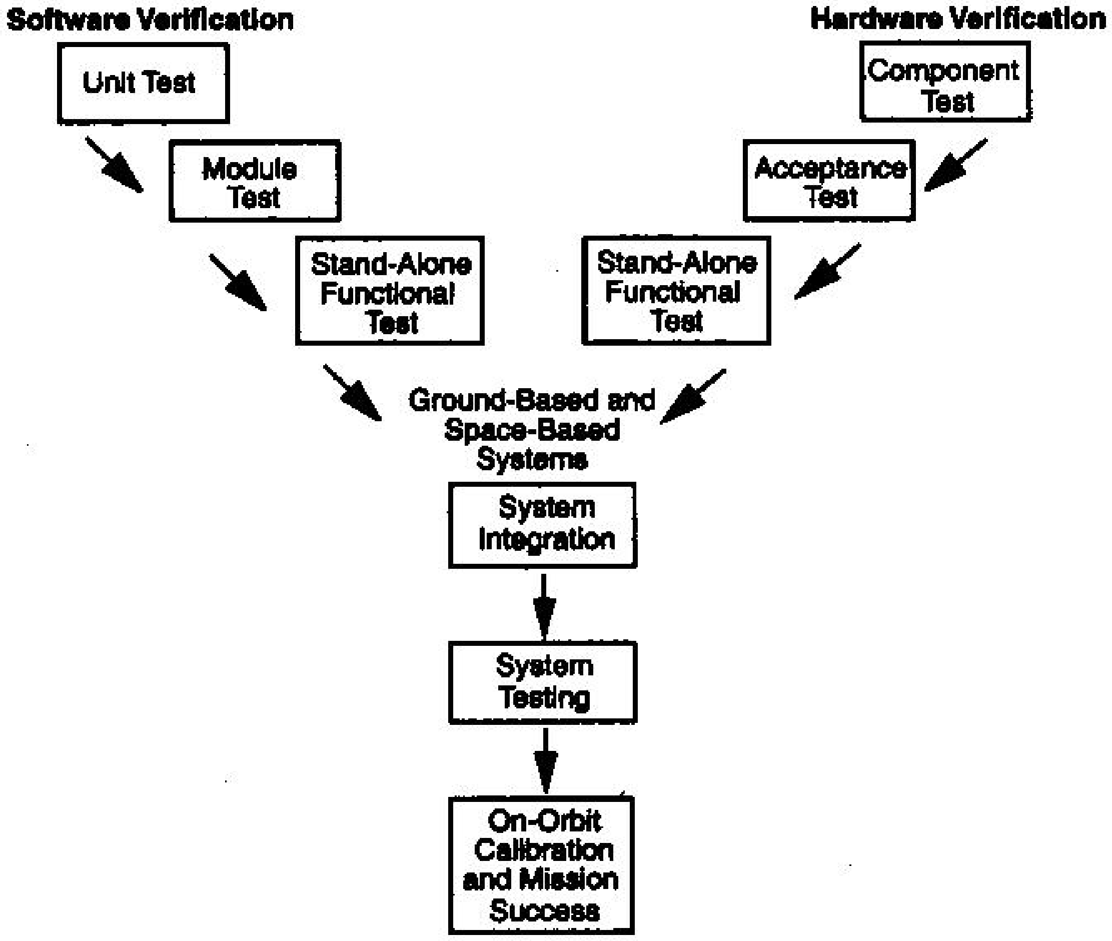
- Memory Blocks limit span of failure effects
- Replicated Start-Up ROMs
- Error Detection and Correction

Error Detection and Correction

- Hamming codes put parity bits at special locations
  - Each data bit is checked by a unique set of the parity bits
  - Permits error detection and correction
  - Requires k parity bits for an n-bit data word

n	**k**
8	4
16	5
32	6
64	7

Integration and Test



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

Software Safety

- It is impossible to test all the software states of even simple programs
- Software often gives no indication of impending failure
- Analyze the hazards and design the software to prevent unsafe conditions

References

- Pisacane, Vincent L. and Robert C. Moore, Fundamentals of Space Systems, Oxford University Press, New York, 1994
- Wertz, James R. and Wiley J. Larson, Space Mission Analysis and Design, Third edition, Microcosm Press, Torrance Ca, 1999