INTRODUCTION

Piloted deep space missions, such as a human mission to Mars, present new challenges in the design of spacecraft computer systems. Because of the human element involved, the demands of the operational mission, and the constraints imposed by weight, electrical power, reliability, environmental (especially radiation), and data processing performance considerations, general-purpose consumer or business oriented computer solutions are not acceptable [1]. That does not imply that every circuit and every microchip for the computer system needs to be designed and constructed from scratch. Some may indeed be, but most will be developed by utilizing, modifying, or tweaking existing designs and products. But it’s not as simple as just selecting top-rated top-value components from a computer catalog or store shelf, hooking them up, and then fine-tuning the result. That system won’t do the job. The solution for this application must be engineered. What this really means is that it is not sufficient to simply state that the system must be “user-friendly, long life, low weight, low power, high reliability, robust quality, and high performance”. Like many other complex designs, “the devil is in the details” [2] and that is especially true in this application. While these design characteristics are valid, exactly how best to achieve them is a very difficult task. There are many questions to answer. How reliable does it need to be? How much power should it consume? How small does it need to be? How much processing throughput should it have? These are just some of the major and more obvious questions. A myriad of questions will present themselves as each area is investigated. The engineering challenge is to determine the optimal or appropriate levels and tradeoffs (quantitative where possible) among these characteristics such that all the operational, system design, and programmatic requirements (cost and schedule being just another two parametric constraints) are satisfied and the system contribution to overall mission success can be assured to a level that is acceptable to all the stakeholders. The best way to systematically go about doing this is to utilize standard Systems Engineering procedures and techniques to flush out the driving design requirements and eventually synthesize a working design [3]. This two-part paper represents a combination of a presentation made at the 2002 Mars Society Convention (Part 1) and one made at the 2003 Mars Society Convention (Part 2)

APPROACH

To determine the design drivers for a system definition, four different perspectives can be combined in an analytical framework. First, we must look backward to the experience gained and lessons learned from earlier human spaceflight endeavors, such as Apollo, Skylab, Space Shuttle, and International Space Station (ISS) [although Space Stations have a different design
perspective because they are more akin to habitats than to transportation vehicles] as well as from unmanned deep space missions. This can provide valuable insight into the best way to move forward by not making the same mistakes that were previously made over again, and by keeping certain design approaches that previously were found to be successful as precepts. Second, we must understand the vision, goals, objectives, and operational concepts of reputable entities associated with Mars exploration, such as the Mars Direct Mission, NASA Mars Reference Mission, or ISTC Project 1172. This is required since the design must be tailored to fit the mission and not be a general-purpose or one-size fits-all type of design. Thirdly, we must understand the current commercial technological state-of-the-art by examining the computer products of industry leaders in both the manufacturing (hardware and software) and system integration areas (especially in the telecommunications industry). This will likely strongly affect the actual design synthesis because it is usually more expeditious to utilize or adapt an existing design than to initiate a design from scratch. Finally, we must review active computer technology R&D programs currently being pursued by government, university, and industrial R&D Labs that have applications to spacecraft design, in order to ascertain the directions, thrust, and focus of these endeavors. This will insure that our design perspective is not “out-of-kilter” with other efforts and that we have a project level information and technology base in which to leverage off.

Therefore, the process can be simplified to the answering of four questions:

1) What are the design requirement drivers based on the raison d’etre and ‘lessons-learned’ from both past and present human spaceflight and robotic spaceflight program designs?
2) What are the design requirement drivers as dictated by the most generally accepted mission concept?
3) What are the design requirement drivers that result from an understanding of today’s commercial state-of-the-art industrial computer product technology and projected future trends?
4) What are the design requirement drivers resulting from analyses of active ongoing R&D programs concerned with computer technology development applicable to space vehicle design?

In the process of investigating questions 1, 3, and 4 above, certain guiding principles of system design will emerge. These principles may be driven by performance, cost, or programmatic concerns, but they will represent the technological bounds within which the system will have to be designed.

**PERSPECTIVE**

In human spaceflight projects to date, custom designs, military designs, and carefully modified commercial designs have predominated. However, since 1994 the DOD has led an effort to utilize open system architectures and ruggedized industrial grade commercial off-the-shelf components to the greatest extent possible in the design of vehicular and shipborne computing platforms [4]. NASA JPL has embraced this philosophy in concert with robotic space projects developed under the “faster, better, cheaper” mantra [5]. Although there has been mixed success in the application of COTS/Open Systems to date, this has now become a mature guiding
principle of system design that is meant to counter the high expense, long lifetime inalterability, and undesired supplier dependencies associated with earlier custom Mil-Spec system designs [6]. Observance of this principle will allow future DOD and NASA programs to leverage off rapid growth of computer and information technologies now occurring in the commercial marketplace. There is no reason to believe that this overarching precept will not also be applicable to the design of the fleet of spacecraft that will carry humans to Mars (although some specific exclusions are probably inevitable). The benefits and payoffs of using COTS/Open Systems are just too great to ignore and it has become the modus operandi for implementing all but the most critical or the most sensitive functional requirements.

In addition, certain design integration approaches that have evolved over the past decade, such as distributed, integrated, modular, and miniaturized avionics [7], will be the cornerstones upon which the computer design of the Mars vehicle will be based. Fundamental design concepts, such as the de-centralization of functionality, the utility of embedded computer control of almost all spacecraft systems, the need for reliable and timely communications between all computing elements, and the ability to detect, isolate, and reconfigure around faulty elements, will dictate that certain design directions be followed. An example of such a direction is the use of network-centric system topologies with protocol driven information transactions instead of unique point-to-point user defined data transactions. This broad understanding of design directions will provide the high level guidelines from which the more detailed drivers to the final spacecraft computer system design will emerge.

Therefore, it is necessary to add a fifth question to the above four in order to complete the overall investigative arena.

5) How do the answers to the above four questions synthesize together with the high level guidelines that have evolved from the COTS/Open Systems paradigm and distributed/integrated/modular/miniaturized avionics concepts?

When answers to these five questions begin to materialize, we will start to accumulate a knowledge base of information regarding the desired attributes of the computer and data processing systems that would likely be onboard a piloted spacecraft on a mission to Mars. As the knowledge base matures, a set of preliminary system design requirements (including identified design drivers) will eventually be derived and formulated in appropriate specification type language. From these high-level system requirements, a preliminary architecture and notional system design, with performance metrics, will eventually emerge.

MANNED VS. UNMANNED SPACECRAFT

However, prior to delving into the analyses proper, it is necessary to understand and appreciate the key differences and similarities between manned and unmanned spacecraft, and in particular, between computer systems designed for manned spacecraft and computer systems designed for unmanned spacecraft. Although a human mission to Mars will obviously utilize a manned spacecraft, the operational mission design and logistics will be similar to that which has already driven the design of deep space unmanned robotic spacecraft. So there is a synergism that can be obtained by examining the attributes of each (Note that unmanned earth orbiting satellites are
Spacecraft computers designed for both manned and unmanned missions share a number of key attributes:

- May need to operate over long mission times
- Need to minimize power, weight, size requirements
- Need for autonomous operation under certain conditions
- Need for robust fault tolerance and reliability
- Need for automatic FDIR (fault detection, isolation, recovery)
- Need for efficient and reliable telemetry
- Need for robust environmental qualification
- Need for resource management
- Need for high computational and processing performance

Both designs must accommodate these concerns although the degree of concern and the level of specification is really determined on a case-by-case basis and is often driven by the greater vehicle design that provides the platform for the computer system or by the mission requirements. For example, because of the small physical size of unmanned probes, the need to minimize power, weight, and size is extremely important and is usually a fundamental driving requirement, whereas for manned spacecraft it may be a goal or negotiable requirement. The actual specified values in each case will be driven by the vehicle design constraints. Similarly, the need for autonomous operation and automatic FDIR is very important for unmanned probes because it is driven by mission requirements (e.g. long time out of communications, latency of command signal) but may be equally important for manned spacecraft because of different mission requirements (e.g. crew sleep cycle, small time window hazardous mission phases).

However, because of these differences in spacecraft vehicle design and in mission requirements, there are some fundamental differences in the associated computer system designs that have been installed and operated to date. Table 1 presents a comparison of vehicle and mission characteristics between manned and unmanned spacecraft related to computer system design and operation. A computer system designed for a manned vehicle will likely differ from a computer system designed for an unmanned vehicle because of the differences shown in this Table. It can be seen that some characteristics are unique to the manned system, some characteristics differ in relative value only (due to the vehicle design), and some characteristics differ because of mission requirements or perspective. The top five items form the crux of the comparison. The bottom line is that the primary objective of the computer system on a manned spacecraft is to operate the vehicle, including all the associated subsystems, and service the occupants in a safe and efficient manner, whereas the primary objective of the computer system on an unmanned spacecraft is to operate the vehicle and the sensor instruments in order to obtain the science data and transmit it back to earth. As can be seen from the Table, the presence of a human on the vehicle drives the design direction.

Computer related attributes that are unique to manned spaceflight can be summarized by the following:

- Overarching requirement for Human Safety
  - Safety Driven Design & Development Process
• Safety Driven Design Architecture
• Importance of FDIR to operational safety
• Need for User Friendly Human Interface for Situational Awareness
  • Visual Display outputs
  • Tactile Keyboard, Edgekey, and Switch inputs
  • Voice Recognition input
  • Auditory Alarm outputs
• Greater allowable power, size, and weight constraints due to greater available resources driven by human needs
• High throughput and logical processing requirements to accommodate real time automatic control and monitoring of spacecraft systems and to accommodate flight crew needs for manual monitoring and commanding of spacecraft systems
• Allowance for onboard in-flight maintenance and repair
• Incorporates capability for manual pilot-in-the-loop control of certain flight maneuver and control functions (rendezvous, docking, landing)

Historically, when these attributes were considered during the development process, the resulting design usually contained certain characteristics that differentiated it from a computer system design for an unmanned robotic probe, such as:
• Greater throughput and processing power requirements
• Greater memory requirements
• Greater I/O requirements
• A greater number of larger and more complex programs
  • generally utilizes an Operating System and System Services software
  • more diverse application programs
• A sophisticated human interface
• More complex Redundancy Management and synchronization requirements
• Incorporate Central Processing Units (CPUs) and Microprocessors (uPs) instead of Digital Signal Processors (DSPs) and Microcontrollers (uCs)
• Greater power dissipation and cooling needs
• Greater concern for radiation induced Single Event Upsets (SEUs) and Single Event Effects (SEE)s
• Larger footprint
• Longer development time and higher cost

In fact, because of these demanding requirements, multiple computers or processors were often used for different functions. For example, the flight control computer, the command & control computer, the systems management computer, the display driver computer, the engine control computer, and the I/O computer may all utilize separate processors and be located on separate cards or even in separate Line Replaceable Units (LRUs). Mass memory, timing controllers, and backup systems may also be in separate cards or boxes. That is not to say that future manned spacecraft computer designs will follow this tendency. That’s just how it happens to be today. Although the attributes, characteristics, and requirements will be similar, the actual design implementation will change according to the high level guidelines and directions that are in effect at the time.

The key to the design of the computer system for the manned Mars mission will be to determine what combination of characteristics and attributes, conventionally associated with either manned or unmanned spacecraft computer systems, represents the best characteristics and attributes for the manned Mars mission.
SUMMARY

The analysis required to answer each of the five questions presented herein is non-trivial and will require careful research and investigation into areas normally affiliated with both manned and unmanned spacecraft design. But this will assure that only the best and most applicable aspects of commercial computer product technology and space vehicle related computer technology R&D will be properly integrated with mission requirements and experiential ‘lessons-learned’ type knowledge, in order to valuably contribute to the overall design of the spacecraft computer system for the human mission to Mars.

Table 1 Comparison of Spacecraft Computer System Characteristics

<table>
<thead>
<tr>
<th>Manned</th>
<th>Unmanned</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vehicle centric functionality</td>
<td>Payload centric functionality</td>
</tr>
<tr>
<td>High logical processing requirements needed for vehicle and crew applications</td>
<td>High computational processing requirements needed for sensor and instrument data</td>
</tr>
<tr>
<td>Safety directed design - destination driven</td>
<td>Performance directed design - science driven</td>
</tr>
<tr>
<td>Need for onboard user interface for command and display</td>
<td>N/A</td>
</tr>
<tr>
<td>Need for accessibility and maintainability</td>
<td>N/A</td>
</tr>
<tr>
<td>Importance of auto and manual FDIR (including manually switched backup)</td>
<td>Importance of auto FD and correction</td>
</tr>
<tr>
<td>Onboard power generators necessary for life support and thermal control allow for higher power consumption</td>
<td>Low onboard power generators dictate need for low power consumption</td>
</tr>
<tr>
<td>Optional reversion to manual operation when comm to MCC is unavailable</td>
<td>Need for autonomous operation when comm to MCC is unavailable</td>
</tr>
<tr>
<td>Need to work over short and long missions</td>
<td>Need to work over longer mission times</td>
</tr>
<tr>
<td>Greater memory and I/O requirements</td>
<td>Lesser memory and I/O requirements</td>
</tr>
<tr>
<td>Basic reliance on tried-and-true technology</td>
<td>More tendency toward innovation and less mature technology</td>
</tr>
</tbody>
</table>
INTRODUCTION

In Part 1, the overall perspective and design approach was presented along with an analysis of the differences between computer systems for manned and unmanned spacecraft. It was shown that the computer system for the piloted Mars vehicle will leverage off certain key design technologies found on Space Shuttle and ISS, such as redundancy and fault tolerance, user friendly human interface, real-time processing and control, on-board maintenance and repair, and manual pilot-in-the-loop control; but will also benefit from evolving technologies found on the latest genre of unmanned robotic spacecraft, such as automated reasoning for autonomous system control, model based health monitoring, diagnosis and recovery, and automated science data evaluation and observation planning. However, the overarching guiding design principle will be the requirement for human safety and the safety directed design process that necessarily follows. In Part 2 it is shown that this will drive a system-of-systems engineering approach that will employ network based wideband communications between distributed embedded processing units and laptop applications program / user interface units for all but the most safety critical flight control or life support functions. These safety critical functions will be architected similar to Space Shuttle but with a drive towards low power, small size, high performance hardware devices, and advanced software techniques for improving fault detection, isolation, and identification, and for improving the situational awareness resulting from the user interface.

APPROACH

An understanding of the factors that influence the design of spaceborne computer systems illustrates how the overarching architectural and integration ground rules, that have evolved from contemporary NASA and DOD programs and studies, reveal the high-level design drivers. The following factors that influence the design of spaceborne computer systems include many of the same factors that influence the design of terrestrial computer systems, but must be optimized in a predictable manner for the space application:

- System Hardware Architecture
  - LRUs, Boards, Backplanes, Buses
- System Software Architecture
  - Operating System, System Services, Applications Programs
- Computational Capability
  - Precision, Speed, Throughput
  - Memory Capacity, Instruction Repertoire
- Software Language, Programming, & Code Size
- Input / Output Capability
- Communications Bus & Network Characteristics
  - Bandwidth
  - Availability & Determinism
- Reliability & Safety Related Items
  - Hardware Failure Rate
  - Software Integrity
Central to the overall design process is the role of fault tolerance. Although the importance of fault tolerance has not really changed over the years, the methodology by which it is obtained most certainly has. History can be projected forward to identify the key requirements in the design of the computer systems for a piloted Mars vehicle. However, in recognizing the importance of fault tolerance, it is necessary to analyze the man-machine system as a whole and not just the computer components in isolation. This was true for Space Shuttle, much more true for ISS, and will be of even greater significance for the Mars vehicle because of the plethora of complex subsystems involved, the criticality of certain operational phases, and the overall mission duration. Fault tolerance is not just about redundancy of parts but also about efficient and errorless human involvement in the mitigation or correction of abnormal situations. To achieve this human performance requires incorporation of key design elements and careful attention to the user interface.

EVOLUTION OF DESIGN PHILOSOPHY IN HUMAN SPACEFLIGHT

Early space projects relied on reliability of components (MIL-SPECs), process quality (QA Inspections), extensive component, assembly, unit, and system testing, and minimal use of software. Apollo was a turning point. A dual redundant computer system was rejected in the early stages as being too complex and too uncertain in reliability. But designing around component reliability constraints and failure analysis revelations (FMEA, Fault Tree) turned out to be even more expensive. It was hard to get good reliability data on components, and testing revealed disconnects between theoretical reliability and real-world reliability. Furthermore, FMEA/Fault Tree results came too late to meaningfully influence system design. Even without hardware faults, small operator errors often caused computer re-starts, with accompanying service interruptions and initialization problems. To overcome these shortcomings required a movement toward more redundancy and fault tolerance in design. Nowadays, the primary approach used in achieving safety of human spaceflight is the adherence to the “design to eliminate hazards” mantra and its resulting use of redundancy management and fault tolerance in design.

For unmanned robotic spacecraft computers, the highest driving fault tolerance requirement is that no credible single point failure shall prevent attainment of the mission objectives or result in a significantly degraded mission. This is not good enough for manned missions. For computers on crewed spacecraft, the highest driving fault tolerance requirement is twofold:

- No single failure or operator error can result in a non-disabling injury, severe illness, loss
of life sustaining functionality, or loss of an emergency protection system, AND

No combination of two failures or operator errors can result in a disabling injury, loss of life, or loss of the spacecraft (excludes independent simultaneous failures)

This is known as the Fail Operational / Fail Safe (FO/FS) criteria. The human element adds complexity and presents significant design challenges. There is still today much controversy and intellectual analysis that surrounds this criteria. It was the result of an evolutionary design process on Space Shuttle. It is still true for ISS although slightly modified because of the longer time availability for human intervention and the lack of dynamic flight phases.

The original Shuttle design concept was FO/FO/FS utilizing 7 parallel processors in 7 strings with independent input data. This was rejected because of weight, power, and cost constraints. Every alternative between this and 3 parallel processors in 3 strings was evaluated in detail. The final design chosen was a FO/FS strategy utilizing 5 parallel processors in 5 strings, where 4 processors in 4 strings comprised the primary system and 1 processor in 1 string provided a hot switched backup system. This design incorporates replicated hardware and dissimilar software. A fault detection and isolation scheme is used to eliminate 1st and 2nd faults and it also protects against generic software faults. Redundancy management and computer synchronization are integral and necessary to the design but these functions come with a very high overhead cost in design analysis and verification testing. Voting is performed at the computer inputs and at the actuators / end effectors.

A significant limiting and complicating factor to a 4 string computer design is the fact that most spacecraft subsystems have evolved to a 3 string design. This is true for the Electrical Power, Engine Control, Navigation Sensors, and Display Generators that interface with, and are closely coupled to the computer system.

SHUTTLE LESSONS LEARNED

The design goal started as 2-fault-tolerance with failure coverage of 100%. In reality, many factors made achieving this nearly impossible. In striving to meet FO/FS requirements with a less than pure approach, every combination of failure, however improbable, had to be analyzed and verified as acceptable - magnified by reconfiguration actions which had to be verified - resulting in a design, testing, and training program of staggering proportions (> 255 exceptions to FO/FS requirements, > 700 pages of crew malfunction procedures, > 1000 pages of off-nominal crew procedures).

One alternate approach would be to draw a probability “line in the sand”. Unfortunately, there are significant drawbacks to this approach, such as the uncertainty in determining an acceptable P, the uncertainty in determining a method for calculating P, and the uncertainty in the actual calculations of P.

A different alternate approach, which is the path followed today and likely into the near future, is to use a robust software driven fault detection, isolation, and recovery scheme that incorporates advanced software error detecting & correcting algorithms along with advanced hardware design
of sensors, monitors, memory, and I/O (data bus) drivers. Improved accuracy of diagnostics and reduced false alarm rate are critical to an improved design for future applications.

HIGH AVAILABILITY TECHNOLOGY - TELECOMMUNICATIONS INDUSTRY

Features of High Availability (HA) technology that are attractive in spacecraft design include the following:

- Hot Swap
- Plug and Play
- I/O Fail-over, CPU Fail-over
- Robust Error Detection & Correction (EDAC)
- Dynamic Reconfiguration
- Hot Re-start

The question is whether these design features can be utilized as is, with modifications, or at all. Mission critical HA computing usually follows the “open systems” approach and demands continuous service with no loss of application state. It utilizes redundant resources of clustered processors and fault management software that typically recovers faults in 10s of seconds and requires application program re-start with loss of data (state) not already saved to mass memory. Can we utilize this technology for the computing needs on a piloted spacecraft to Mars? Unfortunately, in most cases the answer is ‘no’ for a number of reasons.

Telecom HA systems are not in general “real-time” systems needed for critical flight phases such as take-off, landing, docking, and rendezvous. Furthermore, they suffer from the following undesirable attributes:

- Susceptible to undetected faults - bit flips, polarity reversals, level shifts, etc.
  - non-detection of failure in active unit
  - non-detected latent failure in standby unit
  - standby failure can affect active unit causing additional failure (pollution)
- Total loss of availability for failure to switch-over or fail-over
- Overall outage time too long
  - even “6-Nines” allows an outage of 31.5 sec./year
  - not good enough for life and death situations
- Active - Standby Mode switch-over and fail-over times not fast enough
  - requires power-off of bad unit and initialization of good unit (20 sec. - 1 minute)
- Fault coverage estimating methodology not that good
  - likelihood of successful switch-over/fail-over not precise
- Service group autonomous recovery techniques not good enough
  - added complexity and failure modes
- “Role-swap” accommodation not good enough
  - overlapping operation doesn’t help for many faults

SPACE QUALIFIED MICROPROCESSORS

Computer technology has come a long way in 30 years and will undoubtedly continue to advance during the next decade and beyond. The Shuttle computer runs about 500K software lines of code (SLOC) written in the HAL/S language. It is expensive to maintain, not easily expandable, and requires unique software tools. Figure 1 compares the Shuttle computer architecture to a modern space computer architecture.
Moore’s Law is often used to model the development of information technology. It states that the density of devices on a single chip grows exponentially over time. By implication, a corollary to the law states that processor performance will grow at the same rate. Mathematically, this can be expressed as:

\[ \text{Perf}(N) = 2 \exp \left( \frac{N - N_0}{1.5} \right) \times \text{Perf}(N_0) \]

Thus, a 10x improvement every 5 years is predicted, and has in fact occurred (more or less). However, while this may be good for the consumer electronics customer, it doesn’t directly apply to space qualified processors. Because of the design and test effort involved, the small market, and the low profit margins, space qualified processors lag conventional processors by about 3 semiconductor life-cycle generations (approx 12 years). This is a fact of life, so it would not be wise to lock-in the processor type today for the manned Mars mission if final design of the rest of the spacecraft is not going to occur until 2015, 2020, or later. Modularity of design is key. Being able to plug-in the latest technology processor at the latest time possible, with minimal programmatic effects, is the goal. So a review of yesterday’s or today’s processors is not that meaningful. It is more instructive to analyze the future trends.

Figure 2 compares the performance of some popular space qualified processors (Note that this does not include Digital Signal Processors [DSPs] which are tailored for scientific data crunching and not really efficient at meeting the logical and interactive needs of a control/display oriented computer). A throughput of 300 million instructions per second (MIPS) is now state-of-the-art for the PowerPC 750 but already efforts are well underway to qualify the PowerPC 7455 to 800 MIPS. Projections of growth to 1800 MIPS (the SCS750) are well founded.

**SPACE QUALIFIED COMPUTERS**

Unfortunately, a space qualified processor does not a space qualified computer make. The space qualified computer must incorporate all the hardness (against environments and radiation) and reliability of the space qualified processor in all of the components that make up the computer. That applies to memory (Flash and DRAM), controllers, timers, and I/O. Plus, it applies to both chips and boards. A space qualified Single Board Computer (SBC) may be just one board in and among memory boards, controller boards, and I/O boards.

In addition, the computer must be architected for fault tolerance. A flexible, low cost, state-of-the-art fault tolerant computer would have the following characteristics:

- **Fault Tolerant**
  - masks all failures automatically
  - discriminates between transient, intermittent, permanent faults
  - prevents fault propagation between channels/strings
  - assesses remaining degree of fault tolerance
  - re-integrates “lost” fault tolerance when possible
- **Low Cost**
  - mostly COTS architecture to leverage upgrading with newly developed products
  - redundancy management transparent to applications developers
- **Flexible design**
  - processors operate independently or as members of a fault containment region
  - all processors have access to all data regardless of connectivity
- **State-of-the-art technology**
  - small, low power, low weight
The cancelled X-38, originally intended as a Crew Rescue Vehicle for the ISS, incorporated a fault tolerant computer design that advertised a reliability > 0.99999 and fault coverage of 100%. However, recent advances intended for introduction on the Orbital Space Plane (OSP) have already improved upon this design. The new design has a throughput that is 5x faster while having weight, volume, and power specifications that are 50% less than the X-38 design. Figure 3 illustrates a Single Board Computer using advanced processor and fault mitigation technologies.

SPACE QUALIFIED COMPUTER AND AVIONICS SYSTEM

The overall computer system must accommodate a mix of safety critical, mission critical, and non-critical functions. To minimize the probability of a fault in a non-critical function from affecting the performance of a critical function (and also to minimize development and redesign work), functions of differing criticality can be segregated by the computer and data bus architecture.

Within the computer unit itself, advanced technologies such as Triple Modular Redundant (TMR) processing, single-event-upset (SEU) immune controllers and “glue” chips residing on radiation tolerant Field Programmable Gate Arrays (FPGA), and error-protected memory accessing, will be utilized (see Figure 3). In the software arena, functions of varying criticality will be isolated from one another in time and space using advanced partitioning techniques, as shown in Figure 4. Faults occurring in one partition will be contained and not propagate to functions in another partition.

The overall avionics system will be architected such that processors and buses are partitioned in order to segregate functions. For flight critical functions, highly reliable serial data buses (e.g. 1553, 1394) with deterministic data bus software protocols, will remain the design approach of choice. For non-critical mission and payload functions, high-speed switched fabric network technologies will transfer data between diverse and distributed processors, nodes, and workstations throughout the spacecraft. This will include display processing for cockpit Multifunction Display Unit (MDU) screens, camera-driven virtual windows, and laptop/notebook computers. The network medium will include wireless, wired, and optical communications components. Figure 5 illustrates a notional avionics system architecture.

The characteristics of modularity, integration, and distribution will be strongly emphasized and supported as follows:

- **Modular**
  - building block approach
  - standardized interfaces and components
- **Integrated**
  - some components shared by different functions
  - processor hardware runs software functions of differing criticality levels - partitioning
- **Distributed**
  - processing embedded within various subsystems
  - need for communication networks, bridges, and small remote interface units
  - need for coordination and management of distributed components and processes
replication in mass and time
pollution protection

FUTURE VIEW OF COMPUTER SYSTEM FOR A CREW CARRYING SPACECRAFT

Using the design principles, methodical design approach, and analysis of technology previously outlined, certain design features and concepts become apparent. These will be the guiding principles for future designs. Each of these features can be elaborated upon in more detail in separate and more specific design documents, but a summary listing is included here:

- Networks of diverse, programmable information handling components/units, from chip level upwards
- Information may be digital or analog
- Components may be electronic, photonic, or electromechanical
- Communications and Availability are the key concepts
- Wireless, fiber-optic, and copper cable connectivity used together where appropriate
- Most user command and control from portable devices
- Capability to adapt to different operating modes and environments
- Capability to concurrently execute multiple applications
- Embedded autonomous vehicle health monitoring - Integrated Vehicle Health Management (IVHM)
- “State” centered approach to data and redundancy management
- Real-time system design accommodation
- Goal oriented automation
- Reconfigurable computing (use of FPGAs)
- System-On-Chip design (CPU, DRAM, SRAM, NVRAM, mass memory, power management, communication control, built-in self test [BIST], watchdog timer, error correcting)
- Plug-and-Play Common Modular Avionics
- Fault Tolerant Software

To better understand how general hardware and software design characteristics and features will become important in future Mars missions, it is useful to identify current conditions and project them into the future. In Figure 6, a selected set of hardware and software design characteristics are statused today and extrapolated into the future. These future characteristics will be integral to the design (and in many cases will become design drivers) of the computer system for the flagship spacecraft of the human mission to Mars.

CONCLUSION

There are no showstoppers and no technological leaps are required. The basic concepts, trade-offs, and approaches have been presented in this Paper and the design process is well established in industry. We are better prepared to go to Mars today than we were to go to the Moon in 1961. Only sound engineering judgment and practical engineering planning are needed (plus the requisite funding and management direction). That, together with some old fashioned hard work, and the spacecraft computer system will be ready to support the first human mission to explore Mars.

REFERENCES

http://www.hq.nasa.gov/office/pao/History/computers/Compspace.html
A common quotation often applied to computer programming and design meaning that if you do not pay enough attention to the details, you can get burned. See Martin, W. Mike, 1999, *The Devil is in the Details - The Difference Between Programmatic Concepts and Design Concepts*, Lecture, http://bucharest.stanford.edu/KitTemp/1999/Lecture-2-12-99/MikeMartin/Sld001.htm


Computer Technology Has Come a Long Way in 30 Years

Current: ~500 KSLOC HAL/S
- Unique Language
- Unique Tools
- Expensive to Maintain
- H/W Dependent
- Not easily Expandable

New: ~10 MSLOC C++
- Commercial High Order Language & Tools
- Robust Partitioning – Changes Isolated
- Portable and Maintainable
- Scalable/Expandable

Application Software Overlay
(Choice of GNC, Sys Mgmt, or Payload)
- Minimal Change Containment
- Differing Load Configurations (doesn’t all fit)

Open Systems Architecture
- Layered and Distributed
- Self-Contained – No Overlays

Figure 1
Near Term flight Processor
Performance

<table>
<thead>
<tr>
<th>Processor</th>
<th>closest commercial</th>
<th>clock speed</th>
<th>mips</th>
<th>mflops</th>
<th>SPECint95</th>
<th>SPECfp95</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rad750</td>
<td>PPC 750</td>
<td>150</td>
<td>300</td>
<td></td>
<td>7</td>
<td>4.7</td>
</tr>
<tr>
<td>Pentium</td>
<td>Pentium-I</td>
<td>166</td>
<td>178.4</td>
<td></td>
<td>4.5</td>
<td>3.73</td>
</tr>
<tr>
<td>RHPPC</td>
<td>PPC 603e</td>
<td>166</td>
<td>210</td>
<td></td>
<td>3.94</td>
<td>2.71</td>
</tr>
<tr>
<td>Rad6000</td>
<td>PPC</td>
<td>33</td>
<td>35</td>
<td></td>
<td>1.68</td>
<td>4.9</td>
</tr>
<tr>
<td>StrongARM</td>
<td>StrongARM</td>
<td>88</td>
<td>90</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>Thor</td>
<td>n/a</td>
<td>50</td>
<td>50</td>
<td>16</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>MongooseV</td>
<td>R3000</td>
<td>12</td>
<td>10.6</td>
<td></td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>1750A</td>
<td>n/a</td>
<td>1</td>
<td>3</td>
<td></td>
<td>n/a</td>
<td>n/a</td>
</tr>
</tbody>
</table>

Dhrystone measures integer performance.
Whetstone measures floating-point performance.

Performance of Selected Flight Architectures

Figure 2
Figure 3
Partitions Isolate Varying Software Functional Criticality

**Flight Critical Functions**
- Core System Software (CSS)
- Infrastructure & System Support Software
- Sensor Management
- Actuator Signal Management
- Guidance, Navigation, & Control (GNC)
- Engine Control
- Reaction Control System
- OMS Control
- Landing Gear/Brakes/Nose Wheel Steering (if applicable)
- Door/Hatch/Docking Port Control
- Payload Manipulator Arm Control
- Pyrotechnics Control / Master Events Control
- Power Distribution, Monitoring & Control
- Environmental control
- Integrated Health Management
- Flight Critical Network Management
  - Ground Data Management/Test Support
  - Display Generation
  - Display Management
  - Crew Vehicle Interface
  - Intelligent Flight Control

**Mission Essential Functions**
- Core System Software (CSS)
- Infrastructure & System Support Software
- Communications Management
- Mission Systems Health Management
- Mission Critical Network Management
- Crew Vehicle Interface
- Ground/Test Support & Instrumentation
- Comm Processing

**Payload Functions**
- Core System Software (CSS)
- System Support Software
- Payload Data Processing
- Payload IVHM
- Payload Network Management
- Crew Vehicle Interface
- Ground/Test Support

**Physically Isolated Subsystems**
Within Each Computer, Software Time/Space Partitions Between Functions

Figure 4
Figure 5
Looking to the Future

Current

Hardware

- High power, heavy, actively cooled
- Non-standard interfaces
- Custom or VME architecture
- Single sensor packaging
- Point isolation sensors, copper wiring
- Low data rate interfaces
- Centralized main computer
- Low density memory

Future

- Low power, light-weight, reduced cooling
- Standard interfaces
- Modular architectures
- Multiple/multi-parametric sensor packaging
- Fiber sensor networks
- High data rate optical communications
- Distributed computing
- High density memory

Software

- Limited Qualitative Model Based Reasoners
- Statistical Analysis
- Expert Systems
- Ground Systems: Real-Time and Post-Flight
- Limited Integration Across Vehicle Subsystem and Phases of Operation

- Systemic Qualitative & Quantitative Reasoners
- Simple On-Line & Off-Line Prognostics
- Adaptive / Self Learning software
- Real-Time On-Board Flight Systems
- Completely Integrated Across Vehicle Subsystems and Phases of Operation

Figure 6