

Malleable Signal Processor: A General-purpose Module for Sensor Integration

James C. Lyke, Gregory W. Donohoe
AFRL/VSSE

ABSTRACT

This paper presents the Malleable Signal Processor (MSP), a reconfigurable computing module being developed to simplify integration of a wide variety of sensors and actuators into an on-board spacecraft processing system. Interfacing to sensors and actuators requires a host of control signals with complex timing relationships. In addition, sensor data usually needs some signal conditioning, such as calibration, format conversion and feature extraction, before it is useable by the host system. The concurrent processing demanded by these activities often exceeds the computational capacity of a microcontroller, necessitating custom interface circuitry. The MSP offers an alternative approach, employing programmable logic devices with on-board memory to generate control signals and to condition the data in a reconfigurable module. The MSP's versatility will be demonstrated in a flight system, where it is used to interface to two very different kinds of sensors. The MSP represents a first step toward the more general application of configurable computing in Space.

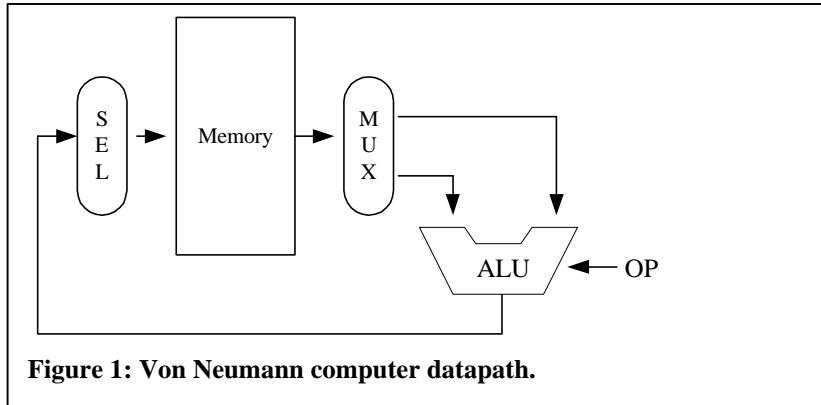
1. Introduction

Integrating complex sensors into flight systems is expensive, due to the high engineering costs required to generate control signals for the sensor, to read and condition the sensor data, and to interface to other computational elements. These signals are a blend of rudimentary, "time-intensive" signals, along with digital waveforms of more sophisticated structure, all real time. A focal plane imaging array is a typical complex sensor. Command and control signals include frame synchronization, a pixel clock, and region-of-interest extraction. Post-acquisition pixel processing includes nonuniformity correction (gain and offset), bad pixel replacement, pixel reordering, and frame buffering. Control for an active sensor, such as a focal plane array with pointing mirrors, requires a feedback control loop that reads mirror position and generates mirror control signals in hard real time. Higher-level, mission-specific functions might include simple pattern recognition, target detection, and tracking. Together, these make a very complex system. The set of concurrent processes, with hard-real-time deadlines and intricate timing relationships, quickly exceeds the computational capacity of a sequential microcontroller. To bring the computational burden within the scope of a microcontroller, custom peripheral processors can be designed, but this is expensive, time-consuming and inflexible. There is need for a programmable interface element that combines the real-time processing power and concurrency of hardware with the flexibility of software.

The Malleable Signal Processor (MSP) is being developed as a general-purpose, reconfigurable building block to facilitate the digital aspects of complex sensor integration. The principal design objective is to reduce the time and cost of integrating a sensor to a spacecraft system by providing a reconfigurable hardware module that incorporates programmable logic devices, data memory, and a predetermined set of programmable I/O lines on a multichip module. With the MSP hardware available off the shelf, the bulk of the system engineer's design effort is reduced to logic programming. Using commercial development tools, the designer can complete a design-implement-test cycle in hours or days, as opposed to weeks or months for custom hardware.

2. Reconfigurable Computing

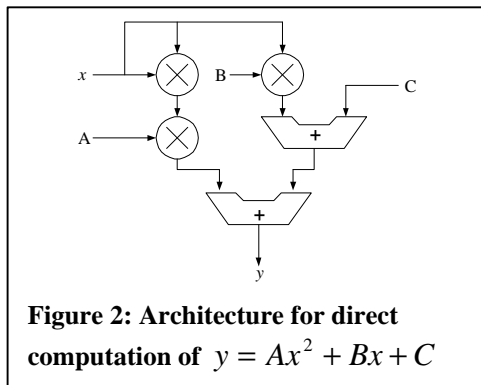
The advent of high-performance, high-density field-programmable logic devices has opened up the possibility of computers that can be reconfigured to optimize their architecture for the problem at hand [6,7]. Possibilities can now be discussed for configuring these devices at a fine granular level during system integration and after field deployment. Reconfigurable computers combine the speed of hardware with the flexibility of software. Conventional computers, including microprocessors and microcontrollers, are built on a Von Neumann model [5], shown in Figure 1. The datapath consists of memory,



communication busses, and a bit-transforming element such as an Arithmetic-Logic Unit (ALU). The hardware that defines the datapaths and instruction set architecture is fixed at design time. The machine is programmed by selecting a memory source and destination for the data, and by choosing operations for the ALU. Only one primitive operation can be

performed at a time. The enormous success of the Von Neumann model is due to its versatility. However, the fixed-instruction-set architecture is primarily oriented towards efficient utilization of the ALU, and is not optimal for all algorithms, particularly those that can take advantage of higher concurrency in hardware. In particular, embedded microcontrollers, which must handle many concurrent processes, quickly bog down in the “Von Neumann Bottleneck”. Modern high-performance processors, including Digital Signal Processors (DSP’s), achieve some instruction-level parallelism through pipelining, but they are still essentially sequential machines. These machines must handle concurrent tasks by multitasking or “context switching” between them to give the illusion of concurrency; this imposes hardware and software overhead. Increased performance requires increasing processor speed, which comes at a cost: faster devices require more power, and faster switching induces radio frequency and signal line noise. Often, a single processor with the necessary performance is very expensive, or in the case of Space applications, simply not available.

An alternative to multitasking is to perform these concurrent tasks in a parallel architecture that can rewire itself, generating data busses and computational elements like ALUs, under software control, and so permit better optimization for the problem at hand. Von Neumann architectures were originally intended to



maximally exploit limited numbers of computational resources, which were relatively expensive in the early days of computer science. To think that an ALU could be idle, even for a single cycle, was considered wasteful. Hence, early computers were completely structured around the proposition that such resources would be heavily utilized. Reconfigurable computers take advantage of more modern “silicon economics”, in which hardware is no more expensive (and sometimes far less expensive) than software. Tens or hundreds of thousands of inexpensive gates can be assigned to a specific computation task. Figure 2 shows an example of an architecture for the direct computation of an algebraic expression. It is convenient to define three different levels of reconfigurability. (1) In

design time reconfigurability, the connections are programmed once, after the system is manufactured, but before it is used. This model is the normal case for programmable logic devices, and can be an inexpensive

alternative to custom integrated circuits. (2) In *static reconfigurable computing*, computational hardware can be built to reconfigure itself at run time in a matter of milliseconds. In order to gain efficiency, it is necessary to exploit natural idle moments, if they exist, or amortize the dead time (milliseconds) between the computational “epochs” that delineate different configurations [2]. Each different configuration can solve a different piece of a problem. (3) In *context switching* or *dynamically reconfigurable computing*, a system can be reconfigured in the middle of a computation; for example, during a Fast Fourier Transform (FFT) calculation, part of the circuitry could be used for multiplication, then re-used for addition. This requires very fast configuration, on the order of microseconds or nanoseconds.

3. The Malleable Signal Processor

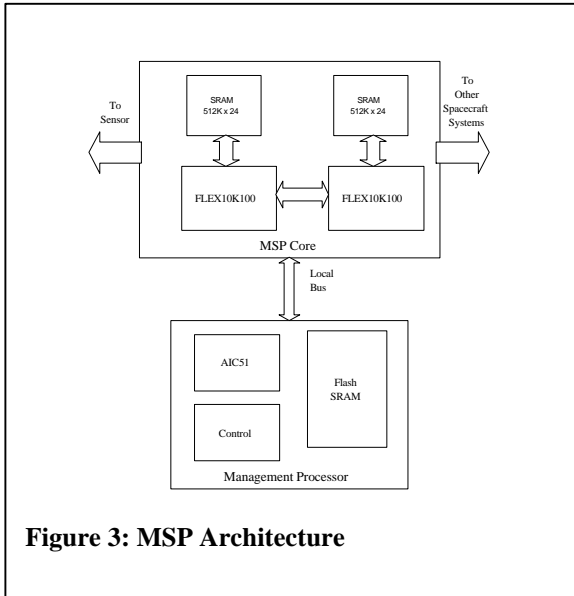


Figure 3: MSP Architecture

Despite a significant base of research in configurable computers [2], little work has been done to realize a practical reconfigurable computer for embedded systems. The Malleable Signal Processor (MSP) is a hardware platform consisting of a configurable computing element, memory, a generous supply of input and output (I/O) lines, and a sequential processor to manage configuration and reconfiguration. The MSP is designed to be reconfigured *in situ* during the course of a mission. For example, one configuration can be used for in-system testing prior to flight; a second would be used for in-flight sensor calibration; and a third, for operation. The commercial field programmable gate array devices employed in the MSP design require several hundred milliseconds of total reconfiguration time. As such, MSP is necessarily classified as a static reconfigurable computer. The basic MSP architecture consists of two functional

modules: the reconfigurable MSP Core, and the Configuration Management Processor. For high-performance interfaces, an optional embedded version of the Myrinet (gigabit/sec) link technology has been developed for direct use with the MSP core.

3.1 MSP Core

The MSP Core contains two Altera FLEX 10K100 Complex Programmable Logic Devices (CPLD), each with nominally 100,000 equivalent gates. Each CPLD is augmented with 512 K by 24 bits of Static RAM. The native 10K100 devices have 406 I/O pins. In the MSP core application, some of these are dedicated to memory access and inter-CPLD communication, and others are used for input and output from the MSP core. The CPLDs can be programmed individually; however, each CPLD can not be partially programmed. Reconfiguration requires hundreds of milliseconds, using either a JTAG serial or a byte-parallel programming mode.

3.2 Configuration Management Processor

The MSP Core is not self-booting; this function is handled by the Configuration Management Processor (CMP). CMP provides overall control of the MSP, by handling the sequencing of one or more configuration sets for the MSP core, and by generating the system clock. The CMP is also capable of preserving sensor calibration information, service history, and other state information in non-volatile memory data structures. The CMP is based on the Advanced Instrument Controller, a very low power, chip-scale multichip module processor that employs a specially modified 8051 microcontroller core and on-board memory and I/O, developed for space applications by the Air Force Research Laboratory [4]. Low performance processors underscore the “background” nature of a CMP. In general, CMPs are used

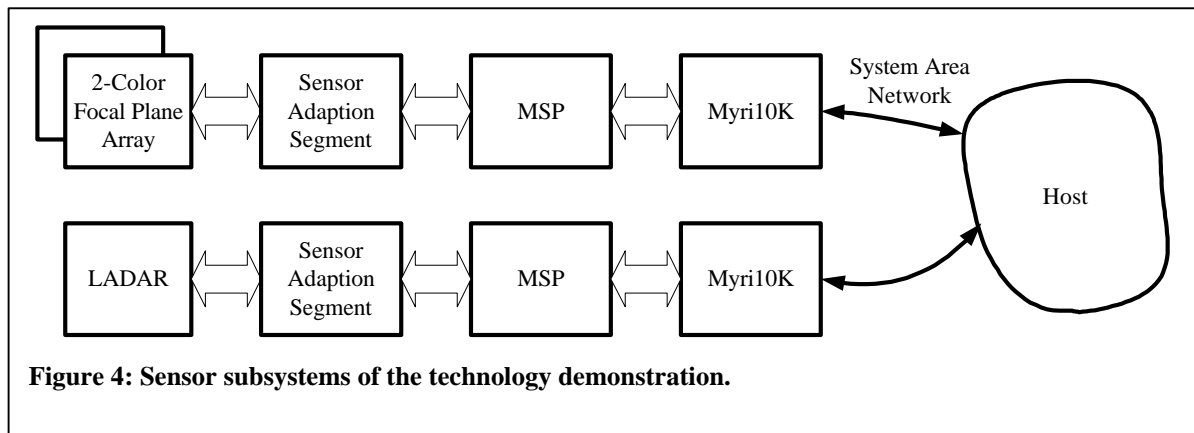
primarily to keep an MSP core in an appropriate operational mode. One megabyte of flash memory contains up to seven configuration files for the CPLDs in the MSP core. A local bus connects the MSP Core and the Management Processor. The local bus is designed to deal with the special issues of *in situ* reconfiguration, in which the logic and routing patterns of the 10K100 devices can be completely reconstituted without loss of memory contents. A “hold/freeze” signal on the local bus serves as a warning to applications coded into the MSP core fabric (as well as outside the core) that a reconfiguration is taking place.

3.3 Embedded Implementation and Malleability

The MSP system is unique among reconfigurable processor designs in that it was intended for real-time embedded applications. Furthermore, the MSP was designed to be miniaturized. In the ensuing “design for package-ability”, it was deemed necessary to minimize component types and quantities, and eliminate components that could not be obtain in die form. These constraints, along with the desire to apply the MSP across a range of applications, led to a design that seems less aggressive in some respects than today’s technology limits might suggest (it is possible, for example, to obtain programmable devices with 1,000,000 equivalent gates). The driving goal of the MSP was not to maximize the raw numbers of gates in a processor, but to establish a flexible framework for embedded reconfigurable processing. In principle, if the MSP were used in the front end of a complex system, it would be possible to rapidly interchange one complex sensor for another type without dismantling electronics boxes and spending months redesigning interface circuitry. The MSP, since it can carry a “palette” of configurations, permits the rapid assessment of several alternatives. A sequence of configurations can be queued from the palette, and obsolete configurations can be deleted or re-written.

4. Technology Demonstration

The Malleable Signal Processor will be demonstrated in the Discriminating Interceptor Technology Program (DITP), an autonomous and experimental missile system designed to track bright targets. To demonstrate the versatility of the architecture, the MSP will provide sensing and control functions for two very different complex sensors: a LADAR, and a two-color infrared focal plane array (FPA). Since the MSP is limited to digital signals, an intermediate module called the Sensor Adaption Segment (SAS) will be designed to handle the analog signal conversions. The sensor subsystems will communicate with the host through a System Area Network (SAN) adapted from the commercially available Myrinet. The overhead for a full-blown Myrinet system would render it too bulky for flight, so a scaled-down interface node was designed. Called the Myri10K, it contains an Altera FLEX 10K100 CPLD and a custom Myrinet interface chip, the FI32. A lightweight client-server protocol was developed to enable message-based communication between the MSP/sensor modules and the host multiprocessor. Significantly, the MSP and Myri10K units are physically identical for the two sensors; only the CPLD configuration software differs. The system will be packaged into 2-1/4 inch square multichip modules and stacked, using the High-Density Interconnect Packaging Program (HIPP) technology [3].



5. Discussion

The fact that the sensor interface and System Area Network operate without a microprocessor (except for the configuration processor) is a testament to the capacity of today's complex programmable logic devices. CPLD design environments provide off-the-shelf libraries for most signal processing functions, including digital filters, Fast Fourier Transforms, multipliers, adders, coders, etc [1]. Using these, the algorithm of Figure 2 can be programmed in a few minutes. With eight-bit inputs, it consumes about 7% of the logic resources, and none of the memory, of the Flex 10K100. Without optimization, it executes in about 120 ns. However, CPLDs are still designed with hardware design tools, requiring intimate knowledge of the device architecture. A design may require several hours to compile. The designer must pay attention to low-level concerns like place-and-route and use of I/O pins. A higher-level, more software-like design environment will be required to bring this technology into common use.

Reconfigurability must be deliberately designed into the hardware. Care must be taken to avoid the disruption of data memory contents and the introduction of faults in peripherals. The reconfiguration process cannot detract from the real-time nature of a system if *in situ* changes are required. The introduction of a "hold/freeze" signal serves as one mechanism, which warns other parts of the system that a reconfiguration is about to occur. Under those conditions, data transmissions into the MSP core are suspended, and MSP core outputs are ignored (assumed invalid). In systems with multiple reconfigurable devices, it may be necessary to establish a protocol of "hold/freeze" signals, particularly if reconfiguration of only particular devices, and not the entire network, is desired.

It is important to consider how such technologies and architectures would benefit aerospace systems. In the first place, it is evident that reconfigurable computing allows one to evolve and refine concepts of on-board processing in systems even *after* they have been fielded. In space systems, the reconfiguration can be done while the platform is in orbit. This is important in ideas involving a distribution of several satellites of a homogeneous constellation, but perhaps even more so in a heterogeneous collection of satellites, which allows us to continue to extract new mission roles and capabilities from space assets. In some cases, these new roles and capabilities might not have been foreseen at the time of initial deployment. Second, it is useful to be able to rapidly reconfigure assets to achieve a sort of tailor-able "processing-on-demand". If information is power, then dynamic re-constitution and delivery of that information yields even more power. Roles and missions of various classes of embedded systems can be re-focussed rapidly, at will, through reconfigurable technologies. Finally, reconfigurable capabilities can be channeled to improve the robustness of systems. New concepts in fault-tolerance become possible.

6. Conclusion

The Malleable Signal Processor is perhaps the first instance of reconfigurable computing to be demonstrated in space. The fact that the hardware for the MSP Core, the Configuration Processor, and the Myri10K is identical for both the focal plane array and LADAR subsystems is a demonstration of cost savings through hardware reuse, enable by reconfigurability. The MSP is capable of in-system reconfiguration for testing on the ground, and for multiple in-flight configurations: calibration and operational modes, for example.

With current technology, CPLD-based configurable computing is limited to design-time and static reconfiguration. The tens or hundreds of milliseconds required to perform a context switch are simply too long for many applications: the vehicle is essentially blind during this time. The ability to perform partial reconfiguration would allow the vehicle to function while changing its program, and high-speed reconfiguration would enable run-time adaptation. New, innovative architectures will be required to allow high-speed and partial configuration.

Currently, the benefits of reconfigurability are limited to the digital portion of the system. The analog portions of the sensor interface must still be custom designed. The ability to extend reconfigurability to analog would reap great rewards in reduced design and integration time, and in in-flight versatility. New software tools and integrated development environments will have to be developed to facilitate the adoption of reconfigurable computing by the design community.

The Malleable Signal Processor is a first step toward reconfigurable systems for space applications. Experience with it will point the way for future research.

7. References

- [1] Altera Megafuction Partners Program (AMPP) Catalog, 1997. www.altera.com.
- [2] DeHon, Andre. "Reconfigurable Architectures for General-Purpose Computing", *Massachusetts Institute of Technology Artificial Intelligence Laboratory Report No. 1586*, sponsored under contract F30602-94-C-0252, October 1996.
- [3] Linderman, R.W., J.C. Lyke, and Y. Kinashi, "DITP: A Flexible Miniaturized Sensor Fusion Processor for Next Generation Interceptor Seekers and Surveillance Sensors", *Proc. 7th Annual AIAA/BMDO Technology Readiness Conference, Colorado Springs, CO, August 1988*.
- [4] Lyke, J., "Design of a Power-Optimized Micro Miniature Advanced Instrument Controller for Sensor Craft Applications", *Proc. 15th Digital Avionics System Conference, Atlanta, GA, October, 1998*.
- [5] Patterson, D., and J.L. Hennessey, *Computer Organization & Design*, 2nd Ed., Morgan Kauffman, 1997.
- [6] Villasenor, J., and W.H. Mangione-Smith, "Configurable Computing", *Scientific American*, June, 1997.
- [7] Wilson, Andrew, "Gate-Array-Based Adaptive Computing Speeds Image Processing", *Vision Systems Design*, September 1998, pp. 66-76.