Reinventing Space Conference
RS-2011-2001

Cell-based Architecture for Adaptive Wiring Panels: A First Approach

Victor Murray, Daniel Llamocca,
Yuebing Jiang and Marios Pattichis
University of New Mexico

James Lyke, Stephen Achramowicz
and Keith Avery
Air Force Research Labs

Reinventing Space Conference
March 2–5, 2011
Los Angeles, CA

AIAA-RS-2011-2001

Cell-based Architecture for Adaptive Wiring Panels: A First Approach

Victor Murray, Daniel Llamocca, Yuebing Jiang, Marios Pattichis
Department of Electrical and Computer Engineering, University of New Mexico, Albuquerque, NM, 87131
Emails: {vmurray, dllamocca}@ieee.org, {yuebing, pattichis}@ece.unm.edu

James Lyke, Stephen Achramowicz, Keith Avery
Air Force Research Labs, Albuquerque, NM, 87112
{james.lyke, stephen.achramowicz, Keith.Avery}@kirtland.af.mil

ABSTRACT

We present a first approach for developing the concept of a manifold of adaptive wiring cells connected as a single overall Adaptive Wiring Panel (AWP). The main use of the AWP is related to affordable plug-and-play space applications but the concept can be used for different applications. A reconfigurable switch fabric enables dynamic routing of signals and power; thus power, digital, and analog signals can be routed for space systems. This concept can also be applied to terrestrial applications such as aircraft wiring and ground-based systems, for example dynamic routing of media such as light or fluids is also possible using the same fundamental switch architecture.

The AWP is a manifold of adaptive wiring cells cast as a single overall panel. The panel is a pegboard-like structure, which does not articulate specific sockets, but rather provides a continuous grid of contact pads and mechanical mounting holes. Implementation is based on three basic elements: (i) cell units (CU), (ii) a cell management unit (CMU), and (iii) modules. CUs are the minimum independent units of the AWP, each with interconnections and links with other cells to form the switch fabric by which we wire components to each other. The CMU talks independently with all CUs and manages the wiring path and panel switch connections. Inter-Integrated Circuit (I²C) is the protocol used for all the communications. Finally, modules are the "widgets" that make up components to be wired (e.g. power supplies, gyros, thermistors, resistors, LEDs, etc.). The modules can be plugged at any orientation, which is detected by the CUs.

We present the results related to the current compact version of the AWP based on 5x5cm cell units. Some of the advantages of this version are the elimination of internal cables and the inclusion of I²C repeaters.

KEYWORDS: adaptive wiring panels, adaptive manifolds, reconfigurable satellites

1. INTRODUCTION

In vehicular platforms, the network of wires that connect sensors and actuators to other electrical boxes is referred to as a wiring assembly or wiring harness. The wiring harness plays a critical role in distributing signals and power throughout the platform. These wiring harnesses are intensively custom, often expensive, and can take a long time to build (i.e. months). As such, the wiring harness can be a limiting factor in the time necessary to build a new system from scratch, even if all components and software are available for the new design.

Conventional spacecraft wiring harnesses are built with architectures that are fixed at manufacture. By implementing reversible (meaning they can be changed repetitively) and dynamically programmable software wires, we can form an "adaptive wiring manifold". Adaptive wiring systems would have many useful properties. They can, for example, be customized quickly, within seconds if the wiring configuration is known. They have tremendous flexibility in that they can be changed up to the last moment of a system's development without removing components and performing painstaking rework. They also have the potential of self-healing and enhanced diagnostics through soft-definable probe signals.

In this manuscript, we explore the concept of an adaptive wiring manifold (AWM), a wiring harness that is reconfigurable and scalable for general applications. In principle, it is like a field programmable gate array (FPGA), which is a chip that can be programmed to implement wiring patterns, except that our adaptive harness allows for the routing of continuously variable analog, power, and microwave signals (FPGAs, in general, can only manipulate Boolean signals). Since an AWM is pre-built and soft-configured as needed, it supports the rapid development of platforms (such as spacecraft). We envision that this might be achieved by assembling a number of tile-like panels, each a "smart substrate" containing a portion of an AWM, to form the overall wiring harness of an entire platform. Components can then be mechanically and electrically attached to panels in the simplest cases completing a platform build cycle. In this manner, we can reduce the time from building some custom systems from months to minutes.

This paper is organized as follows. In the next section, we present a background on the representation of wiring networks in typical systems. Then, in Section 3, we present the concepts for adaptive wiring architectures. Section 4 describes the architecture for cellular adaptive wiring manifolds. Next, in Section 5, we present the hardware and software details for the implementation of the current prototype built. We finish with the conclusions in Section 6. We will use the words "manifold", "harness", and "assembly" interchangeably, though the latter term will also be used to refer otherwise to aggregations of components. The meanings should be clear by context in discussion.

2. BACKGROUND

Wiring is "a necessary evil" in modern systems, and wiring is present hierarchically at every level in a platform, as depicted in Figure 1a. Wires connect the transistors together within an integrated circuit (IC). An IC is itself placed within an electronic package, and the package is a wiring structure. Many integrated and discrete circuit components are united onto printed circuit boards (obviously also wiring structures), and a number of these are put into electronic boxes (usually involving a wiring backplane). Finally, the platform itself, at the highest level of this hierarchy, is formed by connecting boxes together through a wiring harness. Hence, most of the bulk of the electronics in systems is wiring, and many "objects" that we view as components are themselves aggregations of components within its outer boundary, all connected with wiring. For the purposes of this paper, we focus only on the highest levels of the packaging hierarchy, the boxes (of a primary platform such as a spacecraft) and how they are connected through harness assemblies (Levels 3 and 4 in Figure 1a).

Wiring harnesses, because of the diversity of electronic boxes they connect together, can be very complicated in appearance, as demonstrated in the example of the TacSat 2 (an AFRL experimental satellite) wiring harness, depicted in Figure 1b. The formation of a wiring harness requires painstaking planning to identify the physical location, quantity, and qualities of the wiring network. The many individual connections of a wiring assembly are often created manually, resulting in considerable time and expense. Only limited changes in the system design can be made once a wiring assembly is produced, due to the possibility of complicated rework operations in which components may need to be removed to access a wiring network for modifications. Defects in either the wiring harness or the components in a platform can be difficult to isolate, and repairs are necessarily involved undertakings.

2.1. Representation

We introduce a few basic concepts that will facilitate understanding the schematic representation of wiring harnesses. We start with the schematic representation of a typical wiring harness shown in Figure 2. This schematic depicts a wiring manifold (a Level 4

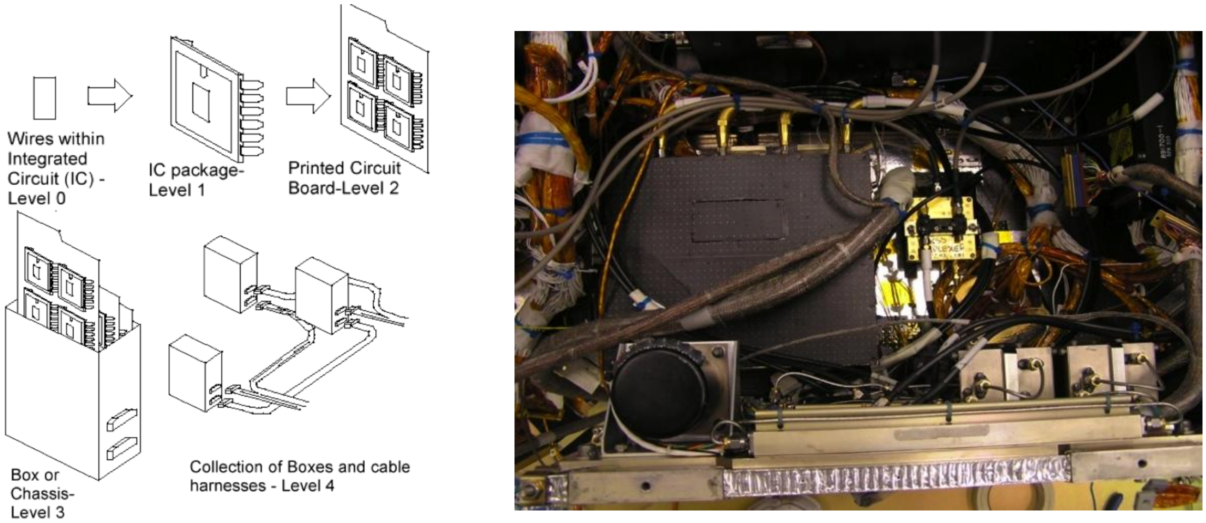

Wires within Integrated Circuit (IC) - Level 0
IC package- Level 1
Printed Circuit Board-Level 2
Box or Chassis- Level 3
Collection of Boxes and cable harnesses - Level 4

Figure 1. (a) Packaging hierarchy. (b) Wiring harness example (TacSat 2 experimental spacecraft).

assembly) containing nine electronic boxes (also referred to as "components" or "modules"), which are Level 3 assemblies (B1-B9). The modules in this diagram will have one or two ports, which are physically expressed in actual boxes as electrical connectors. In this sense, module B2 is a "single-port" module, whereas modules B1, B4, B5, and B6 are "two-port" modules. Wiring counts are indicated in red text next to slash marks, indicating bundled collections of wires. We see in Figure 2 three actual wiring assemblies: (1) a primary complex wiring harness that connects to all nine modules; (2) a cable from module B4 to an antenna having two wires; and (3) a cable connecting modules B5 and B6 having 17 wires.

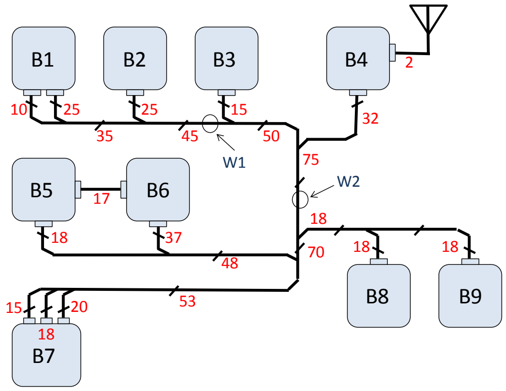


Figure 2. Schematic depiction of wiring harness

Wiring counts in this high level depiction do not appear to follow "simple math" rules. For example, one might expect 60 wires to be contained in the bundle at slice defined by W1 in Figure 2, instead of 45 wires. To understand why the wiring counts do not simply add, a more detailed breakout of the wiring is shown in Figure 3. The original bundle right of module B2 contains 45 wires, as shown in Figure 3a. The wiring bundle is in fact a composition of wires comprising four possible cases involving two modules (Figure 3b). The first case (α) is that subset of wires from module B1 that appear "distinctly" from the bundle, meaning that they do not combine (or bus together) with other signals from module B2. The second case (β) is the "distinct subset" from module B2. The third case (δ) is that subset of wires that combine between modules B1 and B2 but do not propagate past slice W1. The fourth case (γ) is that subset of combined wires between modules B1 and B2 that propagate past slice W1. This last subset represents bussed connections needed to form connections with other modules in the overall harnessing assembly. Hence the number of conductors in the actual wiring bundle at slice W1 is simply $\alpha + \beta + \gamma$.

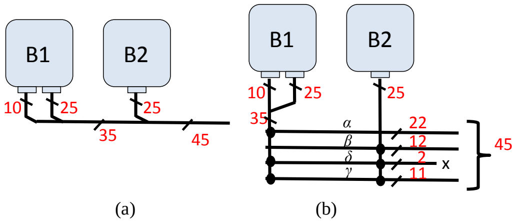


Figure 3. Decomposition of wiring harness at slice W1 (see Figure 2). (a) Original depiction of bundle. (b) Detailed "accounting" of wiring bundle, indicating wire bussing.

2.1.1. Abstract representation

In many circumstances, it may be convenient to represent the Figure 2 wiring harness in the much simpler form shown in Figure 4. This representation reduces the complicated depiction of the manifold to a simple component-like form. We can choose to collapse other parts of the wiring system using simple "transforms". For example, to capture the dedicated cable between modules B5 and B6, we can use the "transform" suggested in Figure 4b, in which we simply increase the wiring counts of each module to account for the extra wires of the dedicated cable and abstract away the second port. Similarly for the B4 module, we can either represent the antenna as another module, increasing the wiring counts through the same "transform" concept. Alternately, if the antenna is in close proximity to B4, we may choose to ignore it altogether from the wiring manifold. These notions are captured in Figure 4c.

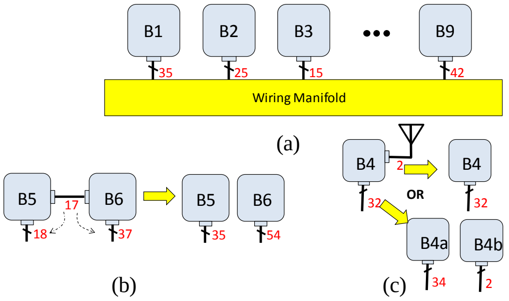


Figure 4. Abstract manifold representation. (a) Simplified representation. (b) Possible "transform" for module pair B5-B6. (c) Possible "transforms" for module B4.

Another notion in wiring harnesses that is much simpler to depict abstractly (than implementing in real life) is that of a "modular" or segmented wiring harness. For the Figure 2 case at slice "W2", we depict such a segmented manifold in Figure 5. This depiction can be interpreted as a two-piece harness, each piece supporting connections to several modules and one to the other harness segment. The segmented wiring harness is a useful concept for cases when wiring is recessed into a number of structures that when assembled form a system.

2.1.2. Connector/cabling options

There are three basic options for cables and connectors in the harnessing systems that we discuss in this paper, shown in Figure 6. The first of these involves the mechanical attachment of modules to a structure, and attaching the harness directly to the connector ports on the module (Figure 6a). The second option involves recessed (embedded) cabling in which a module is mechanically attached to a structure and connected electrically through a direct cable (Figure 6b). A variation of this approach, which is motivated by the elimination of this cable is the blind mating approach (Figure 6c), in which a module is attached to a structure with embedded harness mechanically and electrically. While this latter method has the advantage of eliminating cables, it has the strong disadvantage of positional dependence, requiring the module to align in position and orientation. (We will later discuss an approach with adaptive wiring in which this condition is relaxed.)

2.2. Nature of Wire in Systems

Understanding the qualities of wiring in systems is fundamental to any research in reducing it or creating adaptive approaches. In general, wires are not all "equal". Some carry power at high amperage, some carry impedance-controlled microwave waveforms, some carry sensitive analog signals. The number of wires and patterns are also not random in nature, and we present some results that suggest they may follow a scale-free model.

2.2.1. Wiring domains

We consider four broad domains for wiring:

a. Digital/discrete. Digital wires transmit discretized signals using physical voltages or currents that delineate values for each of two (Boolean) or more (for generalized, multi-value logics) defined states. Sometimes, the term "tri-state" is applied in circuit logic approaches in which the absence of an asserted value (as might occur in a high-impedance or unloaded condition) is not to be confused with valid digital information (i.e., it is actually "no information"). Digital wires are designed to satisfy the minimal noise margins necessary to resolve states and in a way consistent with the speed of transport of information. In very robust digital signaling approaches, signal isolation levels below 20 dB may be adequate, and wires can be very forgiving of ambient electrical noise. This is not always the case, as in high-speed low-voltage signaling standards that might require additional shielding and careful control of wire impedance.

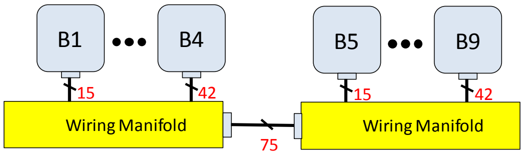


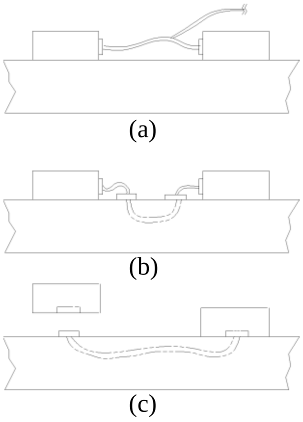
Figure 5. Modular wiring harness, segmented at slice W2 (see Figure 2).

b. Analog wires. Wires from simple sensors (such as thermocouples) can be analog in nature, in which continuous-time voltages and currents are impressed. While "everything is analog" at some level of consideration, we refer here to low-frequency, instrumentation-quality signal environments. Such wires may be extremely sensitive to coupling noise and attenuation, affecting the qualities of measurement.

c. Microwave wires. In radio-frequency signals, the characteristic impedance and electromagnetic environment affects operation, and wire length, shield configuration, and geometries of wire (e.g. discontinuities in wire and surrounding dielectric shape) can seriously degrade performance.

d. ***Power wires***. Wires involved with power generation and distribution are usually most concerned with series loss due to wiring resistance, which to first order is ameliorated with the larger cross-section of heavier conductors.

A typical platform will contain a mixture of wire from each of these domains. It is not generally possible to mix domains effectively. For example, wires good for digital signals may be unsuitable for power and microwave. Robust wires for power, even if suitable for


(a)

(b)

(c)

Figure 6. Connector/cabling options. (a) Standard. (b) Embedded wiring harness. (c) Blind connector.

other domains, will be unsatisfactory for broad use in a system due to an excessive mass penalty. As such, wiring harnesses are "tuned" to accommodate the qualities of each wire accordingly in the most efficient way possible.

2.2.2. Wiring distribution

The wiring in systems is not random. In printed wiring boards and integrated circuits (ICs), the pioneering work of Rent and Donath [7] established a power law relationship (also known as a scale-free model [8]) for the distribution of wirelengths. The distribution concept is shown notionally in the log-log plot in Figure 7 [9]. This relationship has been fundamental important in understanding the growth of wiring levels in ICs and in the planning of FPGA architectures [10].

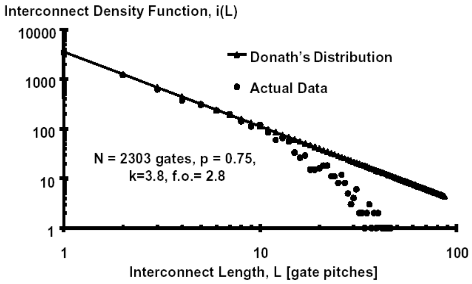


Figure 7. Interconnect density function for integrated circuit (from [9]).

Spacecraft may also respect the properties of scale-free distributions. An unpublished survey of wirelengths on the TacSat-2 spacecraft [11] (Figure 8) suggests a wiring distribution characteristic similar to those found in ICs, an observation clearly worth further study. Most of our assumptions for wiring architecture in this paper are informed from this preliminary insight.

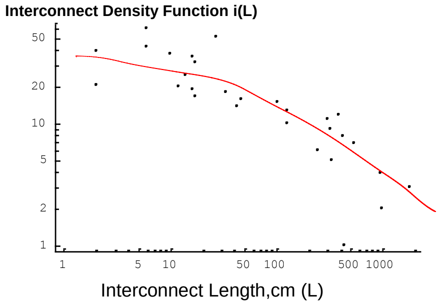


Figure 8. Interconnect density function for TacSat-2 spacecraft (from [11]).

2.3. Coping with Wiring Harness Complexity

With launch costs exceeding $30,000 per kg, reducing the mass of a spacecraft's wiring harness without compromising reliability is highly desirable [1]. One might propose a number of concepts to reduce the bulk of wiring harnesses, such as standardized interfaces and wireless interfaces.

2.3.1. Standard interfaces

In truly standard interfaces, the connectors and pin definitions of electrical boxes (Level 3 assemblies) follow a common, controlled definition. Ideally, in the disciplined application of standard interfaces, the bulk of an otherwise unconstrained wiring harness is diminished. This is in part because a number of custom, discrete wires might be eliminated by using protocol conventions that allow the functions of these wires to be subsumed within the standard interface (as a single-port connection). Unfortunately, disciplines can fail, due to poor control in systems engineering, complex legacy components (that may be difficult to re-engineer), or a lack of trust in moving separate signals into a standard interface. Otherwise, standard interfaces aid in reducing complexity, since when used, the patterns of wiring take on a more predictable format.

2.3.2. Plug-and-play interfaces

A "plug-and-play" interface can be thought of as a special case of standard interface, in which networks of self-describing components (with standard interfaces) can be freely arranged in a "topology-agnostic" network to form a system. The Air Force Research Laboratory (AFRL) created the Space Plug-and-play Architecture (SPA) [12] around the concept of single-point connections to Level 3 components (similar to those used in the mice and keyboards of personal computers), as suggested in the network diagram shown in Figure 9. The SPA concept is scalable and topology agnostic. The size of the network can be expanded by adding hubs, and links can be rearranged without affecting the system function. SPA-based systems have been developed in the laboratory, and simple SPA networks have been flown in experimental sub-orbital

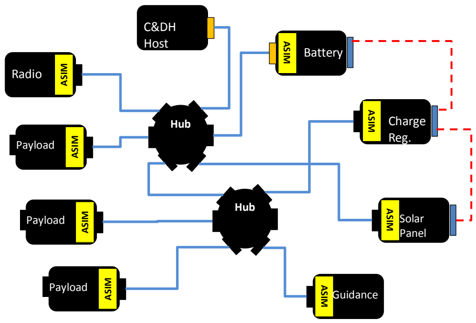


Figure 9. Network based on the space plug-and-play architecture (SPA).

and orbital space missions.

Wiring harness approaches for SPA systems can take on different embodiments. In the wiring distribution concepts pioneered in the plug-and-play satellite (PnPSat-1) [13], a segmented wiring harness approach is used to dramatically simplify the appearance of the system-wide manifold. The PnPSat-1 structure can be thought of as a box, arranged in six panels. Each panel contains a network hub, externally presented connectors for up to eight modules, and internally embedded wiring. The schematic for two of these panels is suggested in Figure 10a. A ten-port hub supports eight ports for modules that can be attached to panels, with two ports for inter-panel connection. Internally (Figure 10b), the hub wiring and components are integrated, including a data connection (based on spacewire), power distribution, and test bus functionality. While this internal harness is complicated in appearance, the complexity is hidden in the sealed panel (Figure 10c). We can consider the panel itself to be a Level 3 assembly, and a platform then is comprised of a set of Level 3 components, attached to each other with individual point-to-point cables. Following the approach shown in Figure 6(b), modules are fastened to panels using bolts in standard mounting holes (a 5cm x 5cm pegboard-like grid is used on PnPSat-1), and short cables are used to connect the module to the panel. The panels are then assembled together to form a spacecraft.

2.3.3. Wireless interfaces

Wireless approaches might also be used (radio frequency (“RF”) or free-space optical) to simplify the wiring harness by eliminating wires from modules inside a system. Of course, the reduction is limited when physical power must be delivered to modules, since that delivery is usually done with wires, meaning that the modules are not totally wireless. Critics of wireless approaches in spacecraft might attack them for: (1) enriching the electromagnetic environment, possibly introducing interference within the system, especially with telecommunications; (2) increasing the possibly undesired possibility for detecting the satellite due to its own emissions; (3) the possibility of “cyber” attack due to infiltration of the wireless network; and (4) the need to engineer the spacecraft to permit RF/optical routing (in some cases, the transit of these signals might be impeded by nature of the structural design).

Nevertheless, the use of wireless may be attractive for many very low-power devices that might be able to implement energy scavenging approaches (aided by equipping platforms with means to distribute power

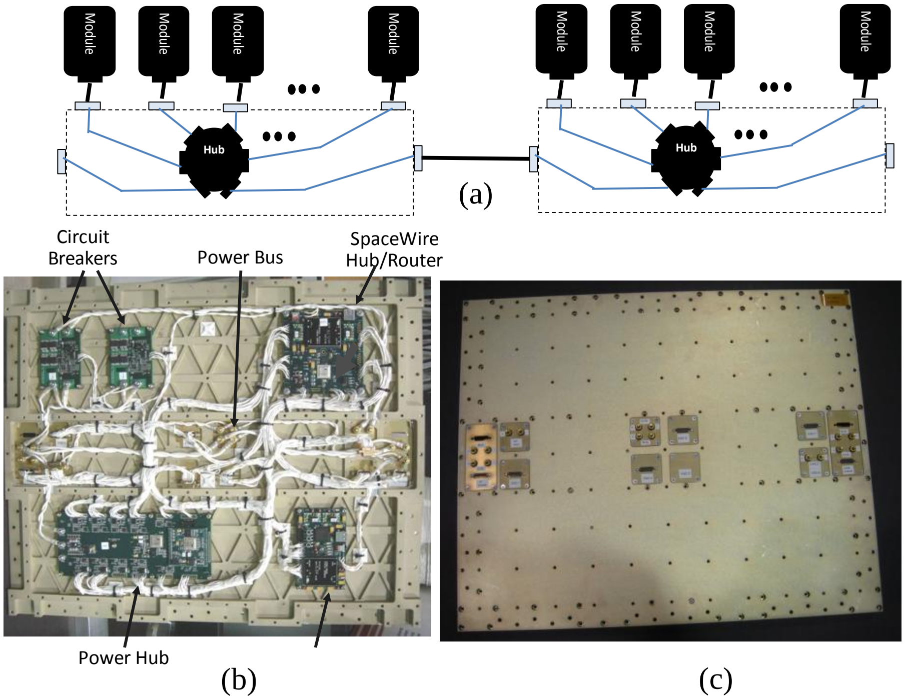


Figure 10. Wiring approach used in plug-and-play satellite (PnPSat-1). (a) Segmented wiring harness. (b) Internal detail of one of the PnPSat 1 panels. (c) External appearance after panel is sealed.

through broadcasting energy inductively or through ambient RF) to operate without the need for power delivery using physical wires. Even with the aggressive use of standardized and wireless interfaces, it is likely impossible to eliminate all wires from Level 3 interfaces.

3. ADAPTIVE WIRING ARCHITECTURE CONCEPTS

In the wiring concepts reviewed so far with the exception of the plug-and-play architecture, the harness configurations are fixed and cannot be changed easily. In the case of the plug-and-play architecture, we reviewed a limited form of adaptiveness, in which a number of modules could be freely commuted on panels, and the panels themselves can be composed to form a larger system. In this section, we present a far more powerful form of adaptive wiring architecture. In this architecture, we demonstrate a much greater flexibility in the types of modules, their termini count and arrangement, and the ability to reform the wiring to accommodate faults, testing, and repurposing to meet different needs.

3.1. Conceptual Architecture

This subsection describes a number of basic principles for adaptive wiring systems. To introduce the basic idea, an abstract adaptive wiring structure (Figure 12) that could be referred to as a panel or substrate contains a number of input/output (I/O) termini. These are shown in Figure 12a as connection points on the left (AI, BI, ..) and right (EO, DO,…) edges. In the case of adaptive wiring (as discussed further in [4]), we can form connections between the termini "on demand". We can supply a wiring "problem" that we wish to solve, a set of termini that we wish to connect together. Through some (not shown) control mechanism, we can convey commands into the panel that cause it to form "virtual" wire connections as desired. In Figure 12b, for example, we show the solution of virtual wires needed to connect termini together bearing the same pre-fix label input on the left edge to the corresponding label output on the right edge (e.g., "AI" connecting to "AO", etc.). In general, the wiring problem to be solved can be referred to as a *netlist* specification. In fixed wiring systems, netlists are implemented physically in the form of a wiring harness. In this example, the adaptive wiring system, represented as a "cloud" within the substrate, forms the wiring dynamically (under program control). At this point, the adaptive wiring concept is notional, and we have not suggested how the "cloud" is implemented.

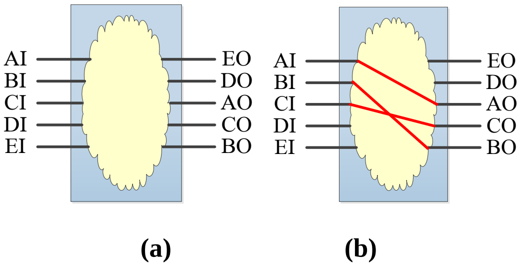


Figure 11. Basic concept of an adaptive wiring cell. (a) Substrate (yellow cloud) having a number of inputs and outputs defining a wiring "problem". (b) Depiction of a possible wiring solution.

Figure 12 depicts a notional implementation concept to provide some intuition about how an adaptive wiring system might actually be implemented. In this case, the substrate takes on the aspect of a physical panel featuring four sockets where components or "modules" can be mounted (Figure 12a). To implement the amorphous "cloud" of wiring resources in Figure 12, a deliberate configuration is depicted consisting of a matrix of wires in rows and columns, with circles shown at the intersection points [5]. The circles represent electrical switches that, when closed, short together the associated row and column. We are not immediately concerned over the specific medium for switches. They could be, for example, metallic relays, solid-state switches, microelectromechanical systems (MEMS) devices, or combinations of these and other switch types [2,3]. Using such a fabric, implementing a solution to a particular wiring netlist amounts to closing

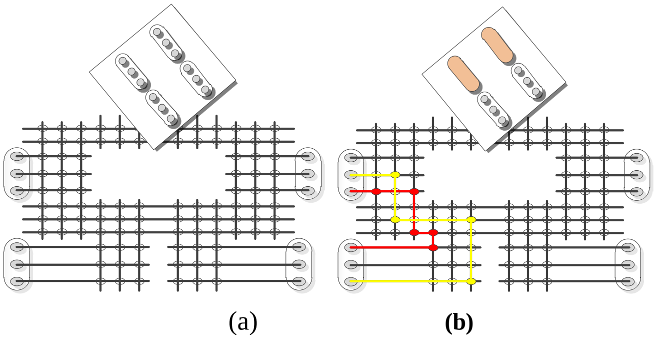


Figure 12. A physical wiring problem example. (a) Unprogrammed substrate, containing four sockets for components (modules). (b) Example placement of two modules and wiring of a two-net netlist.

a number of switches, as shown in Figure 12b, which shows how a netlist problem having two "virtual wires" or nets (involving connections between two placed modules) might be solved (through a total of eight switch closures).

We can extend the concept through an approach analogous to the segmenting previously described. A number of substrates could be tiled together, connected through some of the available external termini as suggested in Figure 13. In this case, two adaptive wiring substrates or "cells" form an extended system. Now, a netlist solution is *compound* in nature, involving a global specification (such as "connect AI to AO") and local specifications (the specific solutions of each "cloud"). The local specification involves allocating terminals between cells, and then defining subnet problems for each cell. It is then necessary to compute local solutions within each cell to implement the implied sub-netlist. It is obvious upon inspection that there are many non-unique solutions for a particular global netlist problem, both in terms of the allocation of nets between cells and implementations of the sub-netlists within each cell.

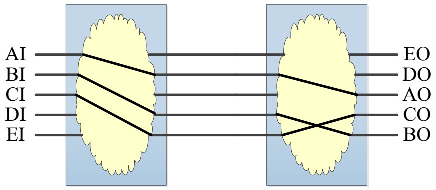


Figure 13. Extended adaptive wiring system by using two cells.

3.2 Benefits of Adaptive Wiring

As previously discussed, adaptive wiring manifolds offer a number of benefits in developing new systems. Since the adaptive wiring substrates (or panels) may be pre-built and inventoried until use, it is possible to retrieve them as needed and configure them on demand. Rather than wait for custom-defined wiring harnesses to be developed and delivered, a process that could take weeks or months, the adaptive versions can be configured very quickly. Unlike custom wiring harnesses, whose wiring pattern is permanently locked in, adaptive panels can be altered as needed to accommodate late-point changes.

Adaptive wiring systems furthermore, have two powerful benefits that are impossible in any other wiring technology. First, this architecture has the ability to adapt to faults that occur after a system is placed in the field. Since wiring patterns can be software-definable, defects could conceivably be fixed by computing an alternate configuration. In Figure 14, a faulty connection between B1-BO can be rectified by configuring other wiring resources that can achieve an equivalent connection without removing a system from the field (which is often impractical, as is the case for space systems).

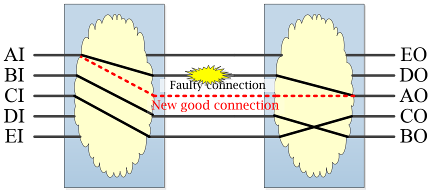


Figure 14. Example of the ability to adapt to faults.

The second unique advantage of adaptive wiring systems is the ability to form probe connections for diagnostic and maintenance purposes. Temporary probes can be inserted at normally inaccessible buried nodes within a wiring system and removed from the system software to be not longer used. This concept is depicted in Figure 15. In this case, we use the adaptive wiring system to set up a temporary connection to check a possible problem with terminal CO on the right panel.

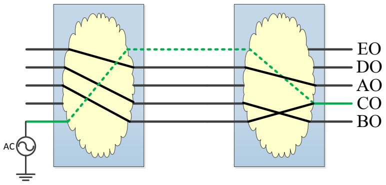


Figure 15. Example of manually diagnosis of connections.

The techniques demonstrated in Figure 14 and Figure 15 can be combined with algorithms to form a self-healing system. Self-healing, as an "active" concept, can be viewed as having two phases, the first being diagnostic, the second being restorative. Clearly, the use of temporary probes can serve to probe an adaptive wiring system, even in situ, to explore the continuity of wiring resources. Upon discovery of defects, an algorithm in the real-time system can compute an alternative wiring path. In earlier AFRL-sponsored work [1], we learned it was possible to achieve self-healing as a linear-time process in an active system.

4. ARCHITECTURE DESCRIPTION OF CELLULAR ADAPTIVE WIRING MANIFOLD

In this section, we make the concepts discussed more concrete through an example of a *cellular* adaptive

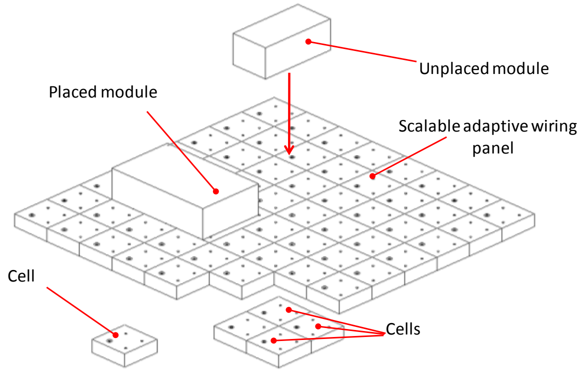


Figure 16. Cellular implementation of adaptive wiring panel.

wiring manifold. It extends the notions of the scalable adaptive wiring harness, drawing inspiration from the panelized construction of the PnPSat-1 platform.

4.1 Cellular Adaptive Wiring Architecture Overview

A simplified physical depiction of an adaptive wiring panel is shown in Figure 16. The Adaptive Wiring Panel (AWP) is a **panel**, a pegboard-like structure, which (unlike Figure 12 or [5]) does not articulate specific sockets, but rather provides a continuous grid of contact pads and mechanical mounting holes. It is a planar substrate composed by tiling together a number of **cells**. Each cell is conceptually similar to the abstracted cells shown in the two-cell concept in Figure 13, tileable to form arbitrary panels, such as the 8x8 cell panel shown. Each cell contains termini, both around the edges (for inter-cell connections) and on the top surface. These latter termini (**pins**) are the only ones that most users would experience in creating a system.

To "use" the panel, a number of **modules** can be arranged on the panel surface and attached. These modules connect to a number of the panel pins, and in this sense, the bottom surface of modules can be thought of as their electrical connector. As such, this architecture implements the Figure 6c notion of a surface-mounted, blind mated connector. However, this approach allows tremendous flexibility over a traditional blind-mated connector in that modules can be placed in many locations and any of the four "Manhattan" directions (orientations).

Once the modules are placed as desired, the **netlist** specification for their interconnections is fed to computer referred to as the **cell management unit** (not shown in Figure 16), which computes the configuration for the virtual wiring in the array of cells that implement the desired wiring solution.

Therefore, the AWP implementation is based on three basic elements: (i) cell units, (ii) a cell management unit and (iii) modules. Cell units are defined as the minimum independent unit of the AWP, all with interconnections and communications with other cells, forming (by iterative tiling) the switch fabric by which we wire modules to each other. The Cell Management Unit communicates independently with all cells and manages the wiring path and switch connections of the panel. Finally, the modules, being the "widgets" that make up components to be wired, contain some features to make it possible to integrate them efficiently (such as a small processor).

4.2. Cellular wiring grid conventions

In the adaptive wiring panel, it is essential to identify conventions relating to mounting and electrical distribution, such as those shown in Figure 17. We define the adaptive panel as a series of intercalated grids, namely "mechanical", "power", and "signal". Of these, the **mechanical grid** is the coarsest grid. The mechanical grid defines the attachment locations for physically mounting modules. We show the points on this grid as occurring at a density equivalent to the pitch of a "unit cell" (in our definition, we specify this to be 5 cm, although there is nothing special in this choice). The **power grid** is defined by superimposing a 2.5cm grid onto the mechanical grid (by convention, the point belonging to the coarsest grid "wins", and the coincident points of finer superimposed grids are suppressed). The intended purpose of these grid points is the support of higher current wiring, consistent with those associated with the delivery of power. The **signal grid** is defined at four times the density of the mechanical grid (1.25cm pitch), which is intended to be the most common case in general-purpose wiring.

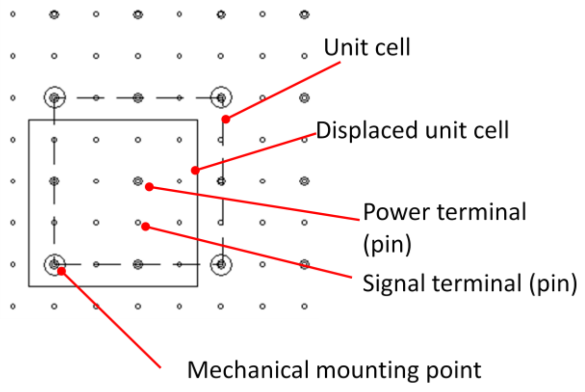


Figure 17. Conventions within adaptive wiring panel.

For cellular implementations of the adaptive wiring panel, it is convenient to render cells as integer multiples of the unit cell dimension. The physical tile boundaries are offset (as shown) to avoid cutting through grid points.

5. EXAMPLE ADAPTIVE PANEL DESIGN AND IMPLEMENTATION

In this section, we will describe the implementation of a demonstration adaptive wiring panel (AWP) system shown in Figure 18. At the time of this writing, the demonstration system (right) contains six (of 64 planned) cell units, a few simple modules (for plugging into the adaptive panel), and a laptop as the controlling cell management unit (CMU).

As a programmable fabric, the AWP requires tools to generate specifications for implementing particular wiring solutions to interconnect “modules” (shown lower left). A simple graphical user interface (GUI), shown upper left, was created for this purpose.

The next subsections will describe the cell unit, the modules, and the CMU that manages the configuration of the overall system.

5.1. The AWP cell unit

Each cell unit (CU) is an “atomic” element, a minimum independent unit required to create the AWP. Its logical architecture is shown in Figure 20. A rectilinear tile that connects to its nearest neighbors in all four directions is referred to as a “NEWS” (north-east-west-south) network. Each edge (detailed only for the “east” port) contains a local communications port (for inter-tile information sharing), as well as pins for routing wiring connections between other edges and the primary surface array of contact pins. The surface array is the primary set of termini that are user-accessible. These are intended to support connections to matching pins present on “modules”, which are to be surface mounted onto an AWP. “Modules”, as will be discussed, are intelligent assemblies, and as such, require communications. The cell-module I2C port provides support for this purpose. Finally, a single “cell common I2C port” is provided to support communications to the cell management unit. Unlike the other I2C ports, which are implemented as point-to-point interfaces, the common I2C port is connected to all cells in an AWP.

The cell functions are managed by a “cell local processing unit” (fully implemented in hardware using FPGAs), including the six communications ports, cell status functions (such as maintaining a globally unique identification code), and configuring the switches connecting the wiring resources in the cell. (additional details are described in [4]). The functions of each CU are:

1) Control the programmable connections of the AWP.
2) Communication with (up to) four neighbors: each CU needs to communicate with its physical neighbors to recognize spatial orientations.
3) Read “electronic datasheet” information from modules (each module has a probe pin which sends

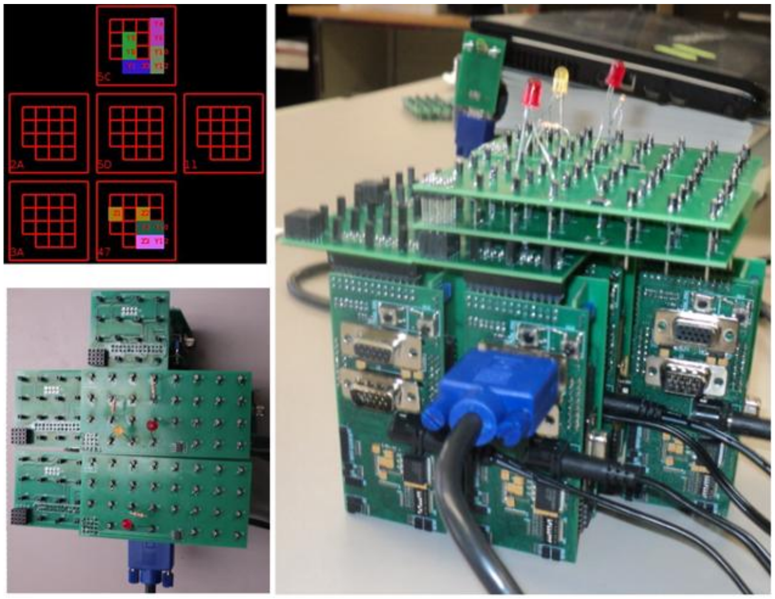


Figure 18. Partial adaptive wiring panel (AWP) system, containing six cells and two modules. (top left) Graphical user interface (GUI) snapshot used to set-up the netlist. (bottom left) Top view of the AWP. (right) Side view of demonstration system revealing the circuitry of cell units below the panel and module boards.

module specifications to the cell it is plugged into).
4) A low current power supply will be sent to power the modules to enable transmission of electronic data sheet information through predefined probe pins.
5) Communication with the cell management unit: each cell unit transmits and receives information to/from the cell management unit (i.e., cell units aside from neighbor recognition cannot communicate directly with each other). Each cell block, upon system power up, will send identification information such as ID of the CU, the IDs of its neighbors and relative orientations, and module Electronic Data Sheets, if connected to that CU.

Figure 21 illustrates the physical embodiment of an AWP cell unit in our prototype. Each AWP cell is made of 5 boards:

1) Top board is where the modules are placed.
2) South board is the main board where the main hardware is placed: a FPGA with all the logic control, the relays to close the connections, and extra hardware (for example, to reconfigure the FPGA for updates of the system). This board controls the rest of the boards. It includes a connector that connects to the North board of a neighboring AWP cell.
3) East board is connected to the West board of a neighboring AWP cell.
4) North board is connected to the South side of a neighboring AWP cell.
5) West board is connected to the East side of a neighboring AWP cell.

5.2 Cell Management Unit

The cell management unit (CMU) manages global communications and routing configurations of all cell units on the AWP. An architectural diagram depicting the connection to a typical cell unit is shown in Figure 19, which indicates the chain between the cell management unit (global), the cell unit (local), and particular relays (switches) controlled by particular cells.

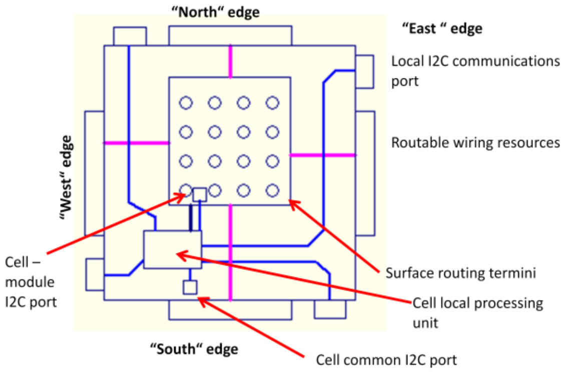


Figure 20. Cell unit logical architecture.

In the current demonstration system, the CMU is implemented as software running on a Linux machine. The basic setup for a GUI is shown in Figure 18 (top left) used to debug the mechanical connections and other design issues.

The functions of the CMU are:

1) Manage the communication bus (I^2C protocol).
2) Read and organize initial location/orientation information from the CUs.
3) On a periodic basis, scan for any modules connected to a particular CU, and read embedded module "electronic data sheets".
4) Compute the global and local routing / path connections necessary to implement a desired netlist. The CMU has a priori knowledge of the layout of relays

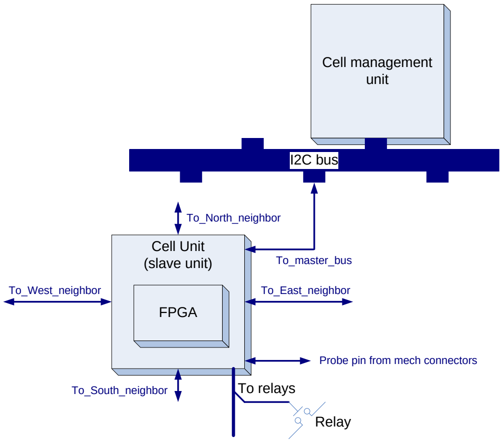


Figure 19. Cell Management Unit (CMU) with a typical connection to a Cell Unit.

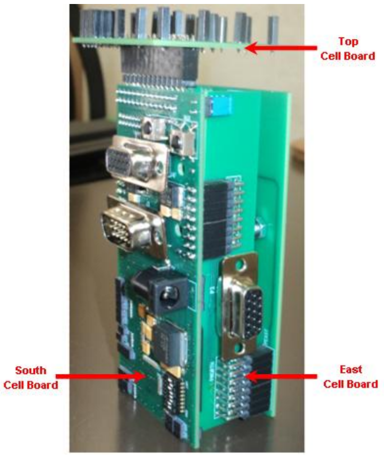


Figure 21. Cell unit of the AWP.

in each cell unit (they are identical in our demonstration system, but in principle the relay distribution can vary amongst cells). When the geometry of the AWP is formed (by assembling a tiled arrangement of cells), a complete graph related to the AWP and the connections is built by merging the subgraphs of each individual cell. After reading the module "data sheets" describing module contents and terminal information, the CMU attempts to compute connection paths specified between the corresponding terminals on the various modules plugged into the AWP.

5) Command individual cell units to open or close specific relays.

5.3. Module

Modules refer to components that are plugged into AWP assemblies. Before we describe them, it is insightful to consider how an AWP might be used in a simple design example shown in Figure 23. A prospective AWP is shown with three modules in Figure 23a (a light bulb, a switch, and a battery) placed on the panel. The placed modules cover a number of pin locations and mechanical attachment points (revealed in Figure 23b). These modules can be placed in any Manhattan direction and in any linear position so long as the mechanical attachment grids of the module align to those on the AWP. At this point, the AWP does not "know" what to do with these modules. Rather, the user placing the modules must supply this information in the form of a netlist. When this is done, the AWP can connect the modules by forming virtual wires, as shown in Figure 23c. If a second copy of a module (e.g., an extra light bulb) is placed on the AWP, it intrinsically has the ability to connect to this second copy when the failure is detected (Figure 23d).

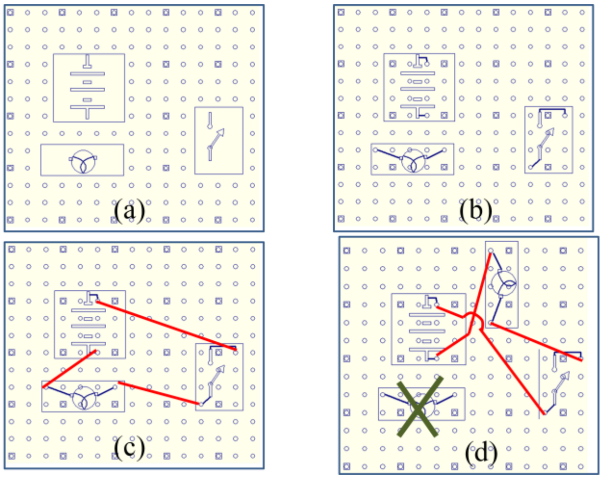


Figure 23. AWP in use. (a) Modules are placed. (b) Connection details revealed. (c) Virtual wire formation. (d) Connection to redundant module.

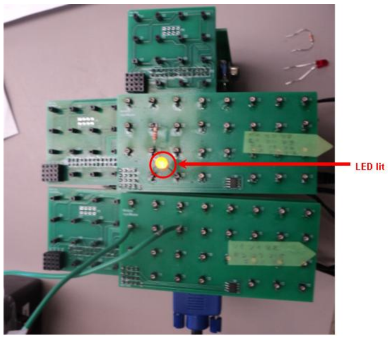


Figure 22. AWP with the circuit connected using two modules. The bottom module has a battery and the top module with a resistor and a LED.

We now move from this abstract description of an AWP with modules to describe our demonstration implementation. The modules we built were 5cm x 10cm (or 2 cell units) with 24 signals connectors, 6 power connectors and 2 mechanical connectors (most of which need not be connected for a specific module type).

The modules employ an electronic datasheet on a small processor resident on the module (but not otherwise a part of the component(s) described). The datasheet of each module employs the SPICE language for the netlist descriptive format. It is downloaded over one of the I2C ports, interfacing to one of the cell processors contained in the panel, eventually being routed to the cell management unit. The CMU manages the database of modules and netlist connections, forming virtual wires on demand as needed (when it is possible to do so).

In Figure 22, we show an example of 2 modules connected to the AWP: the bottom module with a battery and the top module with a resistor and a LED (fulfilling the role of the light bulb in Figure 23).

5.4. Routing algorithms

As described in [2] (for example), the wiring configurations of the AWP can be described as graphs, and manipulated with graph algorithms to find satisfying assignments for solving routing problems, as done routinely in FPGA synthesis algorithms. The basic heuristic algorithm used in the demonstration system can be summarized in the following pseudo-code:

```
Generate the graph based on the current cell configuration.
Generate a Johnson-Trotter [6] ordering of the required connections.
```

Apply ***shortest-path algorithm*** for each connection.
Update graph.

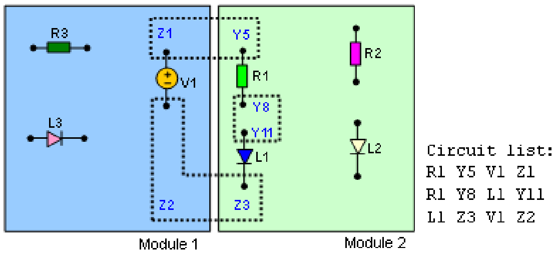
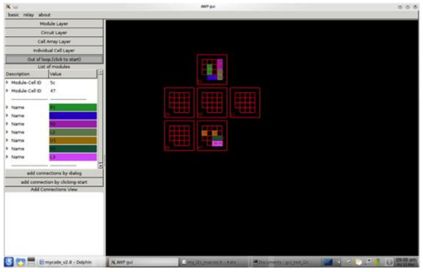


Figure 24. Example of a circuit to be connected: module 1 has a battery (V1), a resistor (R3) and a LED (L3), and module 2 has two resistors and two LEDs.

The algorithm begins with the graph representation of the current AWP configuration. Cells are subgraphs of an overall extended (multi-cell) AWP. If the cells are re-arranged or extended, the change is automatically detected and the graph is re-generated.

Once the graph has been constructed, we apply a very simple, greedy approach in an attempt to route all required connections of a particular netlist. Each connection is called simply a *net*. Our simple algorithm works through all possible nets in an arbitrary initial sequence. When one net is routed, the routing resources (wires and switches) consumed to form that route are no longer available for nets still to be routed.
As one might suspect, this approach may not lead to a solution due to congestion. It is, of course, well known that routing problems such as these (which are referred to as graph Steiner Forest routing problems) are NP hard, and require more sophisticated heuristics. As our priorities were on forming the elements of the basic system design, we did not extensively pursue more sophisticated algorithms. While we might suggest this as an "exercise for the reader", it will likely be re-examined as the scale of our demonstrations increases.



Description	Value
Module-Cell ID	5c
Module-Cell ID	47
Name	R1
Name	
Name	R2
Name	L2
Name	V1
Name	
Name	L3

Figure 25. GUI Snapshot of the software implemented for the cell management unit. We can see the cell units detected in the real array, the components in the cell units, and the list of components.

Fortunately, a rich base of research on these algorithms awaits us, and we will hope to have occasion to adapt them for the nuances of this novel architecture (such as "domain mapping" of nets into graph regions, analogous to graph coloring problems.).

5.5. Status of demonstration

Our recently built demonstration, a partial panel containing six cells and two modules, has demonstrated all of the elements of the AWP described in this section. Module 1 has been prototyped as a "compound" consisting of a battery source (V1), a resistor (R3) and a LED (L3). Module 2, also a "compound", has two resistors (R1 and R2) and two LEDs (L1 and L2). Simple circuits of the form shown in Figure 24 can be composed, in which subsets of modules can be connected.

Even these simple demonstrations have considerable underlying complexity, as each cell contains a dedicated processor, internal wiring and ~70 relays to implement local connections. The modules also have internal microprocessors to "explain" modules to the adaptive system. Our eventual demonstration will contain 64 cells, resulting in nearly 4500 relays. To manage this complexity, tools will be necessary. Figure 25 provides a snapshot of the graphical user interface (GUI) software developed for the cell management unit.

We have successfully created a GUI and negotiated it to compose circuits on the AWP. Once a circuit is set-up, the user can re-arrange either the cell units or the modules. The AWP will look for the new locations of the components defined before and adapt the routing to connect the circuit.

6. CONCLUSIONS

In this paper, we have described an unusual architecture to interconnect (in principle) arbitrary electronic components together using a programmable manifold. The adaptive wiring concept, in one way of thinking, is an extension of the ideas of FPGA routing. The ideas clearly demonstrate that a considerable investment in overhead is required, even to do simple things, like turn on a light bulb. Similarly, FPGAs also come with considerable overhead. For million gate systems, the overhead is often acceptable. The same level of overhead for a 10-gate FPGA, however, would be considered profligate. As such, the power and utility of AWM will likely become more appreciated with larger scale systems (DeHon described a similar phenomenon with Minnick's work on cut-point cellular arrays in the 1960s, with a briefcase-sized system required to demonstrate a few tens of gates equivalent in expressive capacity [14]).

The technology implications for fully adaptive wiring are potentially profound. Systems can be formed more quickly, resilient, and flexibly. The benefits come at a price, namely that of excess overhead, the need to make components (modules) "smart", and the need to have tools, such as a synthesis engine, to manage the complexity of the dynamic wiring.

Our initial work has progressed to a demonstration system which (at the time of this writing) implements only a few of the 64 cells in our proposed experiment. At this scale, we expect to find no technological surprises, but expect to uncover new insights of application potential and learn how to better cope with overhead. We expect to learn to balance global and local considerations (should tiles determine their own local routes?) and to optimize better the balance of switch and wire resources. Is there a benefit in extending these concepts to three dimensions, replacing the notions of smart tiles with "smart cubes"? Can we finally achieve breakthroughs in micro-electromechanical systems to allow us to economically implement a hundred thousand relays in integrated form instead of the painfully tedious discrete implementations of today? In the future, we may, instead of bringing modules into adaptive panels, find instead it is better to simply make all modules adaptive in their own bundle of configurable wires, forming perhaps the ultimate form of configurable system.

References

1. Thompson, Sarah, Mycroft, Ian, "Self-Healing Reconfigurable Manifolds," *Proc. DCC'06: Designing Correct Circuits*, Vienna, Austria, Mar. 25-26, 2006.

2. Lyke, James, Wilson, W., Contino, P. MEMS-based reconfigurable manifold. In *Proceedings of the 2005 NASA MAPLD Conference*, available at http://klabs.org/mapld/index.htm.

3. Lyke, James C., R., Wilson, Warren G., and Broyls, Ren H., Albuquerque, NM, U.S. Patent Application for a "Adaptive Manifold," Pub No. US 2002/0141130 A1, filed 3 Oct. 2002.

4. Murray, V., Feucht, G.A., Lyke, J.C., Pattichis, M. and Plusquellic, J., "Cell-Based Architecture for Reconfigurable Wiring Manifolds," *Proceedings of American Institute of Aeronautics and Astronautics 2010 Infotech Conference*, Atlanta, Georgia, 2010.

5. DeHon, A., Huang, R., and Wawrzynek, J., "Hardware-assisted fast routing," *Proceedings of Field-Programmable Custom Computing Machines*, 2002.

6. Levitin, *Introduction to The Design & Analysis of Algorithms*, Addison Wesley, 2003.

7. Donath, Wilm E. "Placement and Average Interconnection Lengths of Computer Logic", *IEEE Trans. Of Circuits and Systems*, **CAS-26**(4), April 1979.

8. R. Albert and A.-L. Barabasi, Statistical mechanics of complex networks, *Reviews of Modern Physics*, **74**, 47-97 (2002).

9. Davis, Jeffrey A. *A Hierarchy of Interconnect Limits and Opportunities for Gigascale Integration (GSI)*, PhD dissertation, Georgia Institute of Technology, March 1999.

10. Andre DeHon. Balancing interconnect and computation in a reconfigurable computing array (or, why you don't really want 100% LUT utilization). In *Proceedings of the International Symposium on Field Programmable Gate Arrays*, pages 125-134, February 1999.

11. Owczarzak, Laura and James Lyke, unpublished work, Air Force Research Laboratory, August 2005.

12. Lyke, James. "Plug-and-play as an Enabler for Future Systems", *Proceedings of the AIAA Space Conference*, September 2010.

13. Fronterhouse, Don; Lyke, James; Achramowicz, Steve.; "Plug-and-play Satellite", *Proceedings of the AIAA Infotech Conference*, 7-9 May 2007, Rohnert Park, CA.

14. DeHon, A. A. DeHon, "Reconfigurable architectures for general-purpose computing," Tech. Rep. AI Technical Report 1586, MIT Artificial Intelligence Laboratory, Cambridge, MA, October 1996.