Chiplet Based Approach for Heterogeneous Processing and Packaging Architectures

Gabriel Mounce, Jim Lyke
Air Force Research Laboratory
3550 Aberdeen Ave
Kirtland AFB, NM 87112
{gabriel.mounce.3, james.lyke.2}@us.af.mil

Stephen Horan
NASA Langley Research Center
stephen.j.horan@nasa.gov

Wes Powell
NASA Goddard Space Flight Center
wesley.a.powell@nasa.gov

Rich Doyle, Rafi Some
Jet Propulsion Laboratory
California Institute of Technology
{richard.j.doyle, rafi.some}@jpl.nasa.gov

Abstract— **Creating integrated systems on-chip (SoCs) for aerospace platforms is becoming increasingly intractable in advanced semiconductor nodes (< 90 nm) due to: (1) the expense of semiconductor processing and fabrication, (2) sheer complexity in terms of number of circuit elements for a large die, and (3) limited quantities of systems over which development costs can be amortized. To overcome some of these barriers, a modular "chiplet" motif is proposed around which a scalable and heterogeneous architecture multi-generational roadmap for microelectronics can be based that preserves many of the benefits of a SoC approach. A chiplet is defined as a small, high-performance nodal architecture that can be connected to other chiplets using a number of universal links for high-speed communications. The links can be either parallel or serial, each conveying the same information. Parallel links are used in multichip module / 2.5D packaging, in which a number of chiplets may be a packaged into a tightly coupled configuration (having in theory thousands of interconnects). Serial links are used in simpler forms of packaging to connect nodes across boards, backplanes, and boxes. The universality is important for two reasons. First, by establishing an equivalence between parallel and serial links, the same grouping of chips can be packaged in several different ways that result in functionally equivalent implementations (except that the inter-nodal latency will vary between parallel and serial connections). The performance of the links can be evolved over time to take advantage of the fastest available transport (including optical) or the widest parallel embodiments (for aggressive 3-D through-silicon via connections). Second, since the links only pass information, it is conceivable that node designs can be substantially different, allowing heterogeneous mixtures of chiplets, to include not only different embodiments of the same processor, but also wholly different classes of node types, to include ultradense memory "servers" (capable of managing multiple high-speed streams through the same link mechanisms), field programmable gate array (FPGA) clusters, and even extended to include complex, configurable analog and radiofrequency functional blocks in the future. By establishing standard messaging protocols, node arrangements can self-organize as more copies of different node types are added, creating a natural approach for building systems flexibly based on the best of breed semiconductor and packaging technologies. This paper will introduce the basic form of the chiplet concept inspired from joint AFRL/NASA work on next-generation space processing and previous work on scaled reconfigurable processing architectures, and describe some of the features we believe necessary to support scalability and heterogeneity with multi-domain, hybrid architectures involving a mixture of semiconductor technologies, transport concepts, and advanced packaging approaches.**

TABLE OF CONTENTS

1. INTRODUCTION

Since the inception of the integrated circuit (IC), it has been the aspiration of electronics developers to form an entire system as a tightly compressed monolithic block of circuitry containing all requisite functionality. Were it possible, boards, boxes, cabling and the other contrivances of packaging necessary to bring these components together would be eliminated. The earliest attempts to do this in the 1960s were stymied by yield limitations, promoting the familiar pattern of sectioning semiconductor wafers into dice, and most attempts to form electronics from entire wafers (even wafer stacks) were never considered practical. For better or worse, Moore's law has largely short-circuited the need to chase the erstwhile holy grail of monolithic wafer scale integration (WSI). At the time of this writing, the semiconductor industry has advanced from transistors having minimum feature sizes measured in tens of microns to around ten nanometers, and individual ICs having more than 6 billion transistors.

It is safe to assume that a sliver of today's silicon contains more functionality than most would have imagined possible in the earliest attempts to achieve WSI. Nevertheless, the demand for more functionality seems insatiable, and it appears we can never pack enough performance and density

into emerging system designs, including those being developed for the space environment. Meanwhile, the cost of integrated circuits (ICs) has skyrocketed due to the increased cost of fabrication, the growing complexity of intellectual property (IP) cores, and the complexity in system design and verification. As such, creating the highest performance monolithic ICs, the so-called systems-on-chip (SoC), has been inaccessible to all but the most well-funded development groups, those capable of launching products on the scale wide enough to recover the considerable development costs.

In this paper, we introduce the concept of a chiplet to suggest a natural (but different) direction from massively monolithic integration. The "chiplet" is largely driven by a simple motivation: cost. Small chips are less expensive than large chips to develop. The cost factor is especially important for niche markets such as space electronics, where it is not generally possible to underwrite a large SoC development.

Motivation

The National Aeronautics and Space Administration (NASA) and the Air Force Research Lab (AFRL) recently evaluated future spacecraft computing performance requirements in a joint study to determine onboard computing need for multiple mission applications associated with robotic science investigations, other remote sensing, and human space exploration. The project study was referred to as the "Next Generation Space Processor", and the work was commissioned in response to the aforementioned concerns.

Traditionally, spacecraft onboard computing systems are single processor-core systems based on existing commercial or military computers that have been radiation hardened for use in space. The systems are implemented and operated at the maximum required mission performance point, where the term "performance point" includes throughput, power levels and fault tolerance. As NASA and AFRL consider future missions that require both an increase in throughput and greater variations in these operating points, considerations for a new, more flexible processor is warranted. Both agencies envision computing capability to provide orders of magnitude improvement in performance and performance-to-power ratio, as well as the ability to dynamically set the throughput, power, and fault tolerance operating points based on circumstantial priorities.

This paper is organized as follows. We discuss some relevant background giving rise to a joint USAF/NASA study to search for better processing solutions for space systems, which are not able to take advantage of the best commercial products. We then describe a reference homogeneous SoC and introduce the chiplet approach as an alternate embodiment, and the preferred features we might expect to support multiple packaging strategies. We then discuss extensions of the chiplet concept beyond homogeneous processing arrays, conventional packaging and traditional

2. BACKGROUND

AFRL and NASA regularly implement missions that require highly-capable, space-rated, computational processors. In 2012, anticipating the need to incorporate new flight computer architecture(s), NASA's Game Changing Development technology program conducted a study to identify the specific functions that may be required of a next-generation flight computer and, therefore, would drive the architecture of this new computer. The study, known as the High Performance Spaceflight Computing formulation study, used multiple mission scenarios and desired capabilities to identify key performance parameters for the new computer architecture. The study concluded that NASA should invest in developing a radiation-hardened, general purpose, multi-core flight computer that is additionally fault-tolerant, power scalable, active core-scalable, and extendable to co-processor interoperability. In parallel to NASA's study, AFRL's Space Vehicles Directorate performed analysis that extrapolated anticipated computer architecture capabilities and contrasted them with planned mission needs. This analysis indicated the need for a more capable flight computer.

Because many of the important features of a new flight computer architecture were shared between NASA and AFRL, the two agencies conducted a joint evaluation to consider the needed onboard computing technology for the future. An agency-level joint partnership emerged and formulated the Next Generation Space Processor (NGSP) / High Performance Space Computing (HPSC) Trade Study.

In 2013, the AFRL and NASA issued a Broad Agency Announcement (BAA) (BAA-RVKV-2013-02) to solicit responses that included the development of a new flight computer architecture capable of meeting the identified objectives and key performance parameters. Contracts were awarded to British Aerospace (BAE), Boeing, and Honeywell following a competitive source selection process. These three companies independently determined a requirement set against which advanced computing architectures could be analyzed. A common set of themes emerged including considerations to leverage the power/performance operation, scalability, and extendability afforded by the latest in mobile computing architectures. We next discuss a traditional SoC approach for high-performance processing and discuss how the chiplet concept would work as a partitioning strategy.

3. SCALING GENERIC MONOLITHIC SOCS

In this section we introduce two simple reference multicore architectures. The first example, shown in Figure 1, represents a quad core architecture as a SoC. Each processor core is bound to a unique level one (L1) cache. We denote that the L2 cache may optionally be private to a core or in a common L2 cache is shared amongst the processor group, along with a shared L3 cache. Access to main memory (DDR3 or DDR4, etc.) is facilitated through a high-performance dedicated bus. Connections to any number of

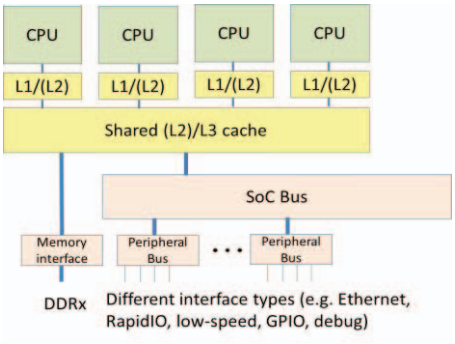


Figure 1. Quad core SoC reference architecture.

external peripheral devices and interfaces are brokered through a high-performance SoC bus (e.g., such as advanced extensible interface (AXI)).

The second example, shown in Figure 2, represents an *n*-core extended multicore system (where *n* can be a multiple of four). In this case the multicore processors remain grouped into clusters of four, primarily to maintain high bandwidth to the shared (L2 and/or L3) cache. Each cluster has a direct interface to a main memory store which serves a particular cluster. Each cluster connects to a common (global) SoC bus, which provides some possibility of brokering transactions on chip between clusters. Any number of variations can be considered from this diagram, to include the insertion of yet another caching level, replication of the SoC bus (i.e., multiple independent internal SoC busses), etc. One can envision more complex, heterogeneous variants, which involve loading up the monolithic floor plan with a variety of other computation engines that can connect through an SoC bus.

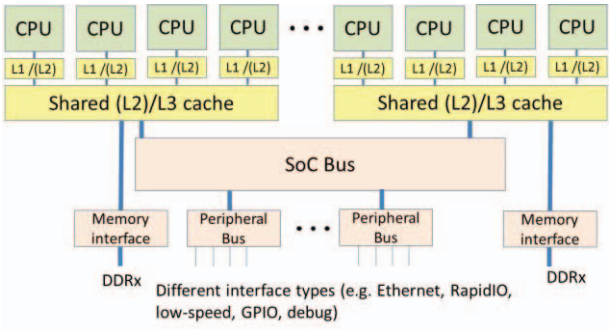


Figure 2. Extension of reference architecture.

Besides extreme expense, there are some drawbacks to the Figure 2 "mega-SoC", not the least of which include the finite scalability even within a monolithic IC of bandwidth between clusters, and the need to manage the integration test in the case where the pallete of disparate IP cores becomes excessive. The most striking limitation of course is the primary limitation to enough banks of main memory to maintain performance as a number of cores grows beyond about a dozen. The number of DDRx - equivalent memory channels can drive pin counts into thousands leading to an excessively complicated system design, even if the development cost of the IC can be managed.

3. A Chiplet Concept

An evaluation by the AFRL/NASA team of the three company's findings resulted in the synthesis of an idea and the creation of a new concept for more quickly leveraging and inserting modern computing architecure features into space rated component development and procurement processes. The variation in modern computing architectures has exploded in recent years, thanks in large part to the exponential rise of mobile computing platforms. The AFRL/NASA team sought to take advantage of this trend. As such, the team first decomposed complex computing architecture (embellished examples of the generic concepts shown in Figures 1-2) into more fundamental computational elements. It was realized that these more fundamental elements of a full architecture, such as the use of a few clusters of computing cores with corresponding interconnect fabric, memory and I/O, might be enough to accomplish the performance requirements set forth by the team within the cost constraints defined by the higher level government budget. After further investigation, what emerged was a concept to realize an extensible architecture that could be customized by the end user.

The Chiplet Approach

We next provide a brief generic sketch of how one might modularly decompose architectures of the form previously shown. As an initial example, we consider the extraction of a single core slice from Figure 2, shown in Figure 3. Figure 3a represents an example partition, resulting in the simplified Figure 3b SoC. As a degenerate "chiplet" no shared cache is necessary, and this concept of a partition would not itself remarkable when viewed against contemporary single core architectures, except for the external hooks for an external SoC bus. It is in fact the notion of this bus (and the different embodiments we shall discuss) that distinguishes chiplets from other integrated circuit systems. In particular, we intend that chiplets can be connected easily to other chiplets, composably and modularly, so as to construct systems at larger scales of integration with less effort than taking other (non-chiplets) and creating custom glue circuitry. Correspondingly, we show a reconstituted form of the Figure 3a SoC based on an interconnection of chiplets (Figure 3c). The notions of the SoC network are a very important consideration in creating such a system, which we address in a later section.

We have considered the notion of a chiplet based on modular decomposition of a complex SoC. We term this concept "chiplet" to denote a more primitive building block, one more economically viable than a larger SoC (smaller silicon area), yet capable of being combined to approximate a larger (more expensive) SoC. Aerospace applications seek high performance, but low volumes make it difficult to justify the expense of large custom ASICs optimized for even a single application use case, much less for the diversity found in the variety of aerospace projects. With the chiplet concept, it may be possible to underwrite the creation of a smaller

number of primitive blocks that can be combined in many arrangements to suit these disparate applications at a fraction of the price for many individual custom designs.

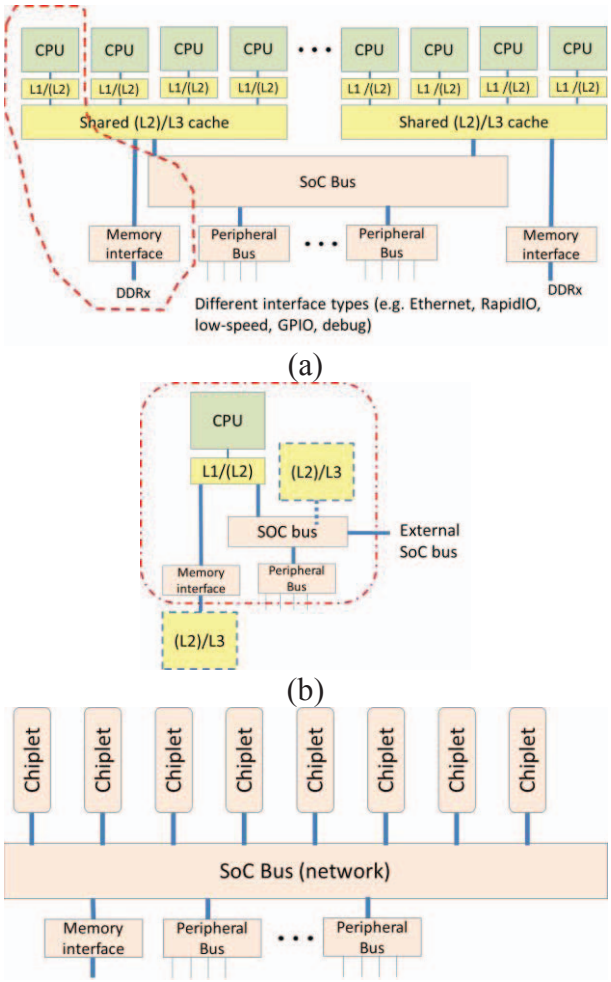


Figure 3. SoC decomposition. (a) Focal partition (in red). (b) Reduced complexity chiplet core (single core version) (c) Reformulation of SoC from chiplets.

Compared to contemporary radiation tolerant processors, which are built in trailing edge semiconductor technologies (e.g., 150 nm), even a single chiplet built in a more advanced node (e.g., 32nm) would represent substantial performance gains. The hope in pursuing chiplets as a modular strategy, however, is to do far better than creating an incrementally better implementation of a previous processor, but rather to provide a means to efficiently scale to much higher levels of performance and much greater power efficiencies.

Universal chiplet bus

We emphasize the notion of a universal bus structure for the transfer of information between chiplets in Figure 4. The chiplet "universal bus" is similar to a SoC bus, but it is intended for use in non-monolithic systems. The bus structure is present in two forms (Figure 4a): a serial form (UX, green) and a parallel form (UXP, red). These are analogous to traditional high-speed serial interfaces and high-performance SoC buses, respectively. The two important distinctions in the UX/UXP bus we propose for chiplets compared to traditional SoC buses are that (1) the UX and UXP buses are duals of each other and (2) the SoC bus is intentionally brought *off*-chip in the chiplet usage. By "duals", we mean that the detailed protocol structure of each bus is preserved in either, such that one can be automatically replaced for the other and transactions done on one can be directly mapped to the other in any compilation involving groupings of these chiplets. The obvious difference in the performance of the busses is speed, as notionally suggested in Figure 4b. Theoretically, the results of any computation will be identical, except for the performance difference due to the difference in the parallel (fast) compared to serial (slower). A certain amount of information (a bus transaction) can be done in a single bus cycle with the UXP bus, whereas the serial form must transfer the many parallel bits using a serial (UX) form. The time difference can be substantial for parallel busses containing hundreds or even thousands of parallel bits. We consider that the performance of the serial bus could be improved by channel bonding, in which several serial (UX) lanes (e.g., 2-10) can be combined in parallel to transfer the same bus transaction in parallel. Ideally, a chiplet would contain many copies of each bus type to enhance the ability to scale chiplets into large networks.

Enhanced chiplets

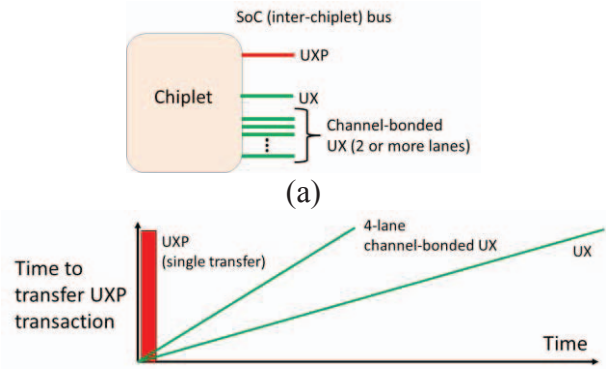


Figure 4. (a) Chiplet universal bus. (b) Symbolic depiction of notional transfer rates.

We now consider two important variations of the chiplet formulation of Figure 3b: higher-capability chiplets and heterogeneity.

Chiplets need not be limited to single processing cores. A example of such a chiplet is shown in Figure 5. This example involves a quad-core processor, in which we introduce some features to distinguish it from a traditional quad-core SoC (such as shown in Figure 1). This form of chiplet preserves the shared cache, which would usually exceed the performance of a cache connected through a UX/UXP bus for the core cluster within a chiplet. In recognition that the UX bus protocols may be distinct from traditional SoC buses, we depict the notion of intellectual property blocks that could prospectively launder SoC busses to the UX/UXP protocols. The DDRx interface is still present in this version of the

chiplet concept. It is depicted as a dashed line in Figure 5 to distinguish it from chiplet UX/UXP interfaces and to denote that it is considered captive to a particular node, as opposed to being a shared network resource.

We depict both serial (UX) and parallel (UXP) versions of SOC bus. The chiplet can be used to retrofit architectures involving more primitive chiplets, replacing several chiplets with single instances of more powerful ones, as suggested in Figure 5b, which replaces four single-core chiplets in the Figure 3c architecture with a single quad-core chiplet.

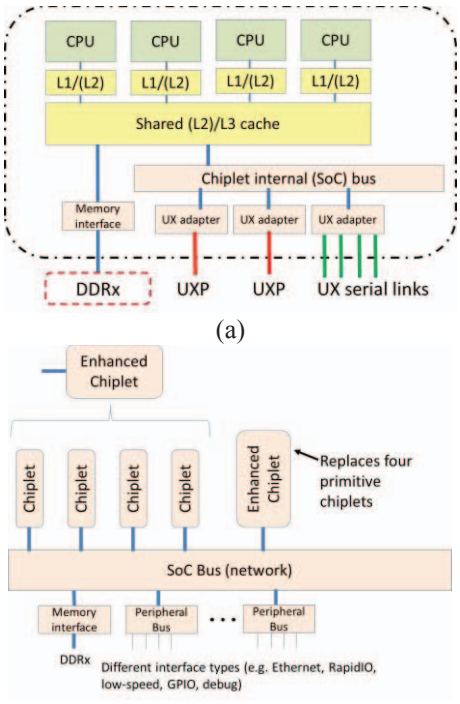


Figure 5. Multicore chiplet. (a) Quad-core embodiment. (b) Replacing more primitive chiplets.

Each chiplet would consist of some basic set of one or computing engines, which might be based for example on popular mobile computing cores. Of course, chiplets need not be based on a particular processing architecture or even be a traditional processor. The only fundamental requirement would be compatibility with the UX/UXP bus protocols, which allows for mixed (heterogeneous) architectures. We discuss more consequences of this heterogeneity in a later section.

Packaging effects on chiplet embodiment--- Figure 6 reinforces the concepts of how different packaging embodiments can increase performance in multi-chiplet configurations. Figure 6a depicts the interconnection of four chiplets. The chiplets are in single-chip packages (memory stores are shown, but they likely will require off chip connections), and connect to each other through the serial form of the universal bus (UX) lines. To enhance bandwidth/coupling, the optional diagonal links can be implemented. Since a chiplets are in single chip packages, the I/O limitations do not permit parallel bus connections.

Figure 5b depicts the same chiplet cluster integrated into a tightly coupled multichip module (MCM). In this case, we assume the existence of an extremely high density interconnect medium, since it will be necessary to support many thousands of pin connections between components. In this case, it is possible (in theory) to create a single tightly coupled MCM containing the four chiplets in their localized memory stores all within a common package. The chiplets use the parallel form of the universal bus (UXP) between chiplets within the same MCM, but use the lower pincount form for buses that extend out of the MCM package. Obviously, the MCM form can operate at a much higher bisection bandwidth than the version based on single chip packages. We might find on further analysis that even with higher pincounts, the reduction in capacitance for MCM-optimized die may result in more power-efficient implementation.

There are many 2-D/3-D packaging technologies that could be brought to bear on increased performance chiplet clusters.

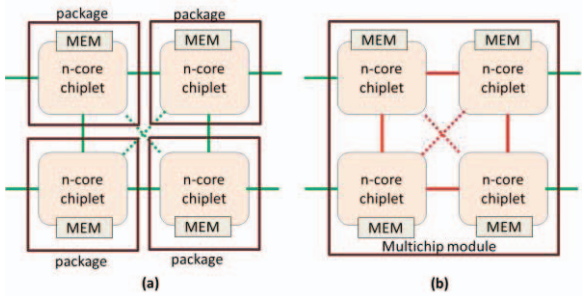


Figure 6. Alternate packaging embodiments. (a) Based on single-chip packages. (b) Based on tightly coupled multichip module packages.

One concept, shown in Figure 7, is simply provided as a reference example, a workable possibility that some of the authors demonstrated in previous research to stack high-performance (DDR2) memory components. The approach involves the use of patterned overlay (copper/Kapton) packaging. Modules are initially fabricated in planar (2-D) form (Figure 7a). In this particular approach, pioneered by General Electric (and used in the number of aerospace and medical applications), a high density interconnection manifold can be brought into intimate contact with dense arrangements of bond pads (Figure 7b). After fabrication, the modules can be singulated from their initial carriers (Figure 7c), and if desired thinned to permit dense stacking (Figure 7d). Hyper-thinning has been demonstrated down to ~75 um. The modules can then be stacked to an almost arbitrary number of layers (an eight layer stack is shown in Figure 6e).

Is important to understand that the Figure 7 approach is distinct from 3-D IC technologies, especially those employing through silicon vias (TSVs). Those technologies have far greater density capability. For example, the Figure 7 approach has a likely upper bound of about 5,000 input output connections per square cm, whereas TSV approaches can exceed that by three orders of magnitude. In fact, the techniques are complementary. TSVs work well in stacking

carefully choreographed individual IC layers, whereas the high-density interconnect (HDI) approach shown in Figure 6 can accommodate a wide diversity of component types and furthermore integrate discrete, passive, sensor, and other component types.

Heterogeneity—We observe furthermore that since chiplets only "see the world" through UX connections, these connections need not merely be made to identical carbon copies of the same chiplet type. Hence, chiplets can connect to completely different types of components having the same UX connection. UX is a transactional/message passing interface, and so long as any element follows the conventions of this interface, it can participate in a network with other chiplets.

For instance, a chiplet might be designed with a set of leading edge mobile device computing "cores" (each of which contains four individual computing cores and a streaming computing cluster) and several layers of cache all interconnected via a high-speed fabric to a high-speed memory controller and and to very high-speed serial and parallel I/O. With this construct as a basic building block module, end users would be able to create customized chips which contain one to multiple chiplets interconnected via the chiplets I/O within a multi-chip 2.5D module or 3D structure. The key innovation lies in the incorporation of high-speed serial and/or parallel I/O into each chiplet device to enable highbandwidth communications between tiles of chiplets or between chiplets and tiles composed of other types of computing engines such as digital signal processors (DSPs), graphics processing units (GPUs), field programmable gate arrays (FPGA), etc.

Viewed as an individual chiplet or as a system of multiple chiplets, a key requirement for the NASA/AFRL HPSC is the flexibility to dynamically trade between processing throughput, power consumption, and fault tolerance to meet varying demands and priorities across NASA and AF missions and within the a particular mission workload profile.

Computing cores—It is envisioned that multiple-chiplet systems will satisfy high performance computing for increased processing bandwidth or increased fault tolerance. The chiplet approach allows system designers to vary processing bandwidth based on the "tiling" of chiplets within MCM schemes which affords designs to segment multiple chiplets as separate fault containment regions.

The chiplet approach also takes advantage of mobile computing state-of-the practice capabilities to dynamically power on/off cores via software control to include processors, memory, I/O and debug unit and the capability to reduce the chiplet system clock rate to reduce the power consumption of each chiplet and the overall tiled system.

Memory—The memory hierarchy built into the chiplet concept relies, more or less, on a typical Harvard architecture approach. Each computing core included in the chiplet will include up to level 2 cache. At the next level removed, the cores will connect to the high speed interconnection fabric (likely derived from IP associated with the computing core)

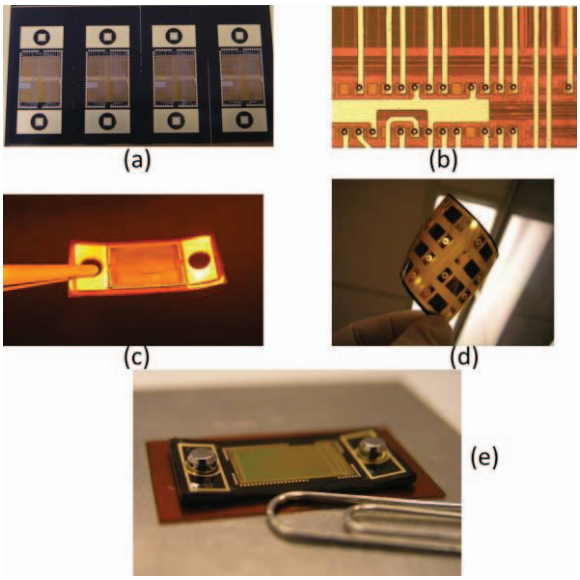


Figure 7. Aggressive 3-D packaging concept. (a) Initial 2-D fabrication. (b) Detail of bond pad connection. (c) Individual module. (d) Hyper-thinned demonstration. (e) Demonstration (8-layer) 3-D assembly.

which allows access to a high-speed memory controller (likely based on DDRx equivalent protocols). At the next level up, the flexibility of the chiplet architecture becomes evident as memory can be placed adjacent to the chiplet in one or several "tiles" within the MCM scheme or as stacks within a 3-D scheme connected using through silicon vias (TSVs). As with the Chiplets, this allows for a wide range of memory schemes everything from typical level 3 cache to full blown flight data recorders.

Chiplet interconnect fabric—A key consideration for the interconnect fabric is the ability to move high-bandwidth information between computing core clusters, to the memory hierarchy, and the I/O for transference from chiplet to chiplet to chiplet to other MCM tiles. As such the ability to provide cache coherency across processor cores within the chiplet while also providing the capability for the coherency to be dynamically and selectively disabled is critical to the operation of the device to support many different computing domains and operate on a variety of computing classes. The interconnect fabric will be based on a crossbar interconnect arrangement and will incorporate the requisite memory management unit (MMU) to control the memory hierarchy and coherency as well as direct memory access (DMA) and interrupt controller.

Software Infrastructure—The intended software infrastructure for the HPSC chiplet approach will support both symmetric and asymmetric processing, and support both real-time operating systems and Unix/Linux based parallel processing. This infrastructure will also support hierarchical fault tolerance, ranging from single chiplet small mission to multi-chiplet highly redundant human missions. The infrastructure includes an operating system that explicitly supports parallel processing and a real-time operating system that supports dynamic resource allocation.

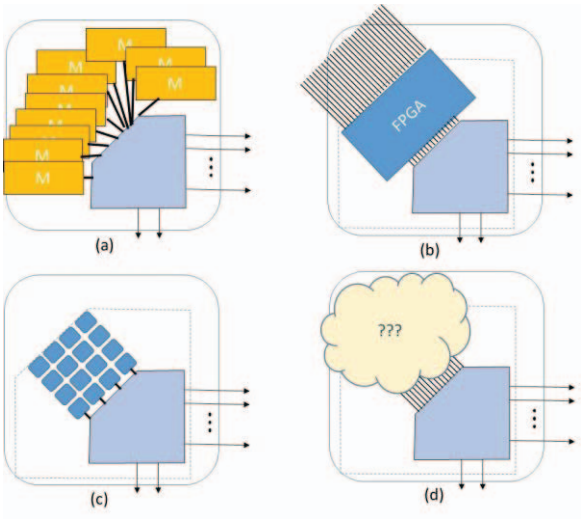


Figure 8. Alternative chiplet functionalites. (a) Mass memory (possibly implemented as 3-D stacks). (b) FPGA (with expansion user I/O). (c) GPU array. (d) Placeholder for alternate analog/RF chiplets.

Other types of tiles—The innovative nature of the Chiplet approach allows the capability of a tiled chiplet system to be tailored to the mission of interest, depending on the overall computational and capability requirements of the space mission platform. As such, a variety of system architectures can be achieved and may be composed of multiple chiplets tiled together to create a large many-core system, or arranged in heterogeneous/hybrid schemes for more specialized computing such as pairing with computing engines for streaming (such as DSPs or GPU), reconfigurable engines for multi-mission modes using FPGA fabrics, or providing increased security posture and fault tolerance via specialty engines such a encryption or security computing cores.

The appeal of the chiplet architecture as an architecture strategy is several-fold. First, the chiplets, being modular, allow architectures of almost any size to be formed by tiling (there are limitations, based on mostly I/O considerations, but discussing them goes beyond the scope of a simple strategy discussion). Second, the modularity allows for chiplets to be potentially upgraded by replacing them with chiplets made in more advanced nodes. For example, a 65 nm chiplet can in principle be mixed with chiplets made at 28 nm or even 7 nm. Finally, since chiplets in this scheme interface with other chiplets through I/O only, it is possible to create other types of chiplets, such as memory-only chiplets, FPGA chiplets, GPU chiplets, and even analog/RF chiplets, all fitting into the uniform system. These possibilities are shown in Figure 8.

The chiplet approach offers a potential unification to advanced microelectronics architecture, even for components that are non-digital, reconfigurable, and those based on multiple process technologies. As physical performance (in process technologies) scales, then the chiplet concept can evolve in a way that is backwards compatible.

4. FURTHER WORK

A notional roadmap for the chiplet pursuit is suggested in Figure 9.

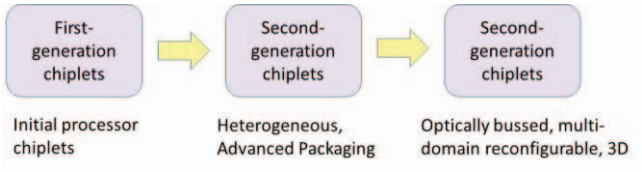


Figure 9. Intellectual property (IP) strategy/roadmap for performance-driven architecture.

In this architecture strategy, the "devil is in the details". In keeping with the discussion of this document, we are trying to identify broad-brush strategy principles. We, in a nutshell, seek to advance from ad hoc architecture constructs (varieties of components support ad hoc needs) to ubiquitous approaches that become more powerful and flexible over time. The chiplet does *not* replace the need for individual components and supporting technologies. Rather, it provides them a context by which they can be related to a broad strategy.

First-generation Chiplets. A first-generation chiplet explores basic intellectual property development for an important building block, namely a processing chiplet. It is not expected that a solution to the UX/UXP bus may be present in the first-generation designs, though we expect an emphasis on creating an approach for connecting chiplets together using high-speed serial links. NASA and AFRL have initiated a procurement program to realize near-term concepts to explore the HSPC Chiplet concept for use in future space missions. Through this program, the two agencies are further detailing the concept, in partnership with the commercial space electronics industry, to design and manufacture an individual chiplet proof-of-concept device culminating in a single die mated to a prototyping board. Future considerations to mature the concept focus on evolving the I/O scheme to take advantage of higher bandwidth data movement schemes, fault tolerance, security, and higher density memory.

Second-generation Chiplets. Second-generation chiplet concepts will emphasize more sophisticated monolithic chiplets, heterogeneity, universal buses (in both serial UX with channel bonding support and parallel UXP form), and advanced packaging concepts that are consistent with these embodiments. Significant work will be necessary to fully

understand suitable networking concepts, laying groundwork for later developments in software defined networking and photonic buses. Protocol advancements will also be necessary to allow for the automatic detection and recognition of chiplet types to permit more efficient mapping software. Tool concepts to manage chiplet arrangements will allow users to take better advantage of modular and scalable chiplet design.

An example of a chiplet-inspired advanced packaging framework is suggested in Figure 10. This concept allows for the coexistence of a contemporary packaging construction (e.g. traditional 3U cards that mount within boxes through wedge locks and backplanes) with a concept we refer to as "plex zones". In this concept, a traditional 3U card allocates a significant part of its floor plan to surface-mounted plex zones. There are eight plex zones possible, four on each side. For the purposes of discussion, as the concept is preliminary (no standards have been defined), the plex zones are nominally 45 mm x 45 mm. Each zone is capable of accommodating one or more chiplet-based functional blocks. As depicted in Figure 10, the chiplets may be based on single-chip packaging, multichip modules, stacked multichip modules (e.g., 2.5D / 3D), and advanced formulations

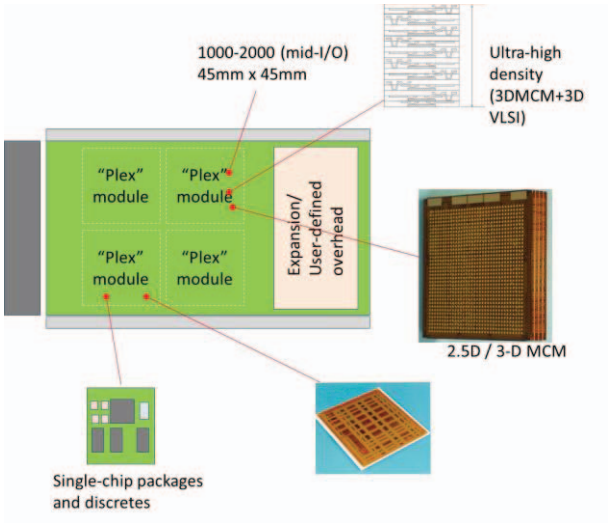


Figure 10. Plex zone concept, based on traditional 3U card packaging.

involving combinations of 3-D (TSV-connected) ICs and multichip modules.

Ostensibly, the plex zones are connected within a 3U card using one or more UX busses in one of several board mounting configurations (e.g. peripherally-leaded or area array). Standardizing the I/O land patterns will greatly simplify forward compatibility (as new chiplet module concepts are created) as well as test and verification. Cards supporting the plex concept can take advantage of recent work in backplane protocols (such as Space VPX), and they can support the placement of custom circuitry for sensors, radio-frequency components, and other needs. It is expected that the plex cards can take advantage of existing (and future) power converter cards. Obviously, more work may be needed to substantiate the ability of this infrastructure to accommodate power and thermal management.

One advantage of the conceptual plex zone approach is the ability to accommodate increased integration scale. This is depicted by the notion of n-plex modules (Figure 11). It should be possible to accommodate aggressive miniaturization and 3-D packaging advancements by providing the possibility of *n*-plex (4-plex, 9-plex, 16-plex) integration. For example, an early version of a scaled processing system might consist of four 3U cards, each supporting eight single core processes (single plex configuration). A more advanced version could upgrade each plex zone with four cores, either by using a monolithic four-core chiplet (simplex) or with a 4-plex of single cores. In either case, four 3U cards could be replaced with one card.

Third-generation Chiplets. Third-generation chiplet concepts will build upon the work in reconfigurable systems

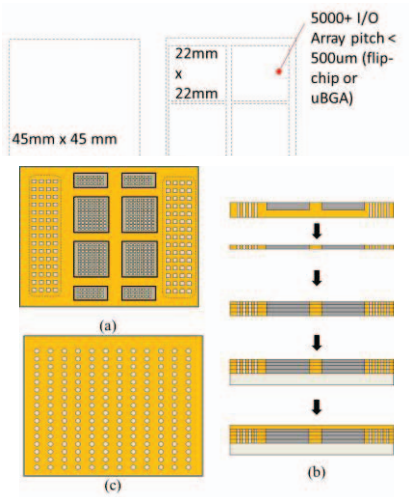


Figure 11. Example 16-plex 3-D MCM. (a) Planar 4-plex module. (b) Thinning, stacking, and thermal spreader mounting. (c) Top-layer area array grid.

(extensions of FPGA concepts into other functional domains) and dynamically scheduled optical buses to dramatically enhance the potential bandwidth of chiplets. Advanced packaging formulations, such as the notional stacked multichip module concept depicted in Figure 12, might be reduced to practice, allowing dramatic improvements in integration scale through the use of chiplet concepts. Such approaches represent a level of interconnection demand even within a single box that would seem to practically mandate and aggressive software defined network based on multiple spatial channels of wavelength-division multiplexing. Additional analysis is needed to make competent projections about the levels of bisection bandwidth that might be required within such systems, and it may be necessary that both card and backplane the provisioned with optical bux support. Extensions of the UX/UXP concepts to even higher levels of performance will likely be necessary but should be straightforward.

5. CONCLUSIONS

Fortunately, the chiplet idea can take advantage of two essential trends in embedded computing. The first of these is the advent of multicore computing. For a variety of reasons, the progression of ever more capable monolithic (single-core) computing stalled over the last 15 years (marked by a leveling off of clock speeds), driving the industry to partition monolithic floor plans into several processor elements (cores) to maintain performance through parallelization. Were this not the case, it would be difficult to consider how to cleanly partition a very large tightly-coupled monolithic region, and the chiplet concept would be impractical.

The second essential trend in embedded computing is the trend toward heterogeneous computing, in which several different styles of computation are present within the same complex system. In this case, by "style" we referred to the notion that some types of computing tend to be driven by regular structure (e.g., predictable, stream-based processing that can take advantage of pipelining, "circuitize-able" forms that lend towards implementation in field programmable gate arrays), while others tend to have less predictable structure (more complex threads and randomized branching). No single processing architecture can do all very well, leading to systems that contain a mix of processing types. A modern cellular telephone for example will typically have a multicore processor (suited for general-purpose computing, especially thread-intensive processing), graphics processing unit (for intensive stream-based processing), and several digital signal processing units (more efficiently pipelined for audio and radio-frequency functions). Here again, the chiplet concept can take advantage of the implicit modularity represented by the need for these different types of computing to coexist in the same overall system.

To be sure, a chiplet approach will have compromises. For intensive, thread-based computing with stringent latency bounds, the shorter proximity and more intimate connection within a monolithic component will give a performance edge to a chiplet approach (which would implement cores in separate components). The need to drive off-chip, off-board, etc. will give a disadvantage in power utilization. In this paper, we outline a strategy to reduce the penalty based on a duality of serial and parallel inter-chiplet links, the latter being conducive to simpler conventional forms of packaging, while the latter is more optimal for tightly coupled multichip (2.5D and 3D) approaches. We believe that a chiplet formulation providing dual options allows designers more flexibility to manage latency and power penalties based on application criticality.

AFRL and NASA recently concluded the Next Generation Space Processor (NGSP)/High Performance Space Computing (HPSC) evaluation. The results of the evaluation indicate that an advanced on-board computing architecture paradigm that is affordable, resilient, and extensible is required to achieve the capabilities necessary to meet future AF and NASA mission demands. The evaluation resulted in a clear trend for utilizing mobile computing architectures, based on multi-core paradigms is necessary to provide the orders of magnitude improvement in performance and performance:power ratio as well as the ability to dynamically set the throughput, power, and fault tolerance operating points. Using this information, the government team distilled a set of computing architecture design preferences that instantiate a new paradigm in embedded space computing. This paradigm takes advantages of a high throughput interconnect fabric and I/O to provide an extensible architecture based on multi-core "Chiplets" that can be tiled together using 2.5D MCM techniques to create a modular and composeable architecture for low power and power:performance scalable operations. Additionally, the new architecture will be made space worthy by incorporating standard radiation hardened by design techniques, and will provide a new level of fidelity in fault tolerance control and security measures. The design concept is envisioned to be made further capable by taking advantage of advancements in high density TSV technology to allow the formation of 3D stacked architectures composed of many chiplet devices, while also affording the opportunity to incorporate and arrange a mix of computing engines to accommodate a multitude of classes of computation.

We do not believe the ideas we present this paper will be the last word on modularization complex monolithic design. Ultimately, a detailed comparative analysis of relevant benchmarks will be necessary to quantify the benefits for a given use case. Often, cost concerns eventually dominate most large-scale system developments, and in those cases, the chiplet approach may give a better result than either building a custom set of application specific integrated circuits (ASICs) or agglomerating large quantities of older components. We are particular concerned in the case of radiation-tolerant applications of finding alternatives to using commercial components prone to frequent disruption due to latchup or single event effects, which is a significant temptation since these components offers attractive performance and low-cost.

6. SUMMARY

This paper outlined a new space computing architecture concept referred to as the High Performance Space Computing (HPSC) "Chiplet" approach. This concept was a result of an evaluation performed by NASA and AFRL to understand the future mission computing requirements and to develop an on-board space computing architecture to satisfy those requirements. The AFRL/NASA team developed the "Chiplet" approach as an affordable, scalable and extensible solution to the meet future mission needs of government space agencies. The approach leverages technology common to the mobile computing domain to create an innovative new construct for realizing computing platforms for space. NASA, along with AFRL, is currently undertaking a procurement effort to realize this concept and approach to enable multiple spacecraft mission applications associated

with robotic science investigations, other remote sensing, and human space exploration.

REFERENCES

[1] Richard Doyle, Raphael Some, Wesley Powell, Gabriel Mounce, Montgomery Goforth, Stephen Horan, and Michael Lowry, "High Performance Spaceflight Computing; Next-Generation Space Processor: A Joint Investment of NASA and AFRL," International Symposium on Artificial Intelligence, Robotics, and Automation in Space (iSAIRAS 2014), Montreal, Canada, June 2014.

[2] Raphael Some, Richard Doyle, Larry Bergman, William Whitaker, Wesley Powell, Michael Johnson, Montgomery Goforth, and Michael Lowry, "Human and Robotic Mission Use Cases for High-Performance Spaceflight Computing," AIAA Infotech, Boston, MA, August 2013.

[3] Richard Doyle, Raphael Some, Wesley Powell, Montgomery Goforth, David Guibeau and Michael Lowry, "High Performance Spaceflight Computing, An Avionics Formulation Task," Study Report Executive Summary, NASA Game Changing Development Program, October 2012.

[4] James Alexander, Bradley J. Clement, Kim P. Gostelow, John Y. Lai, "Fault Mitigation Schemes for Future Spaceflight Multicore Processors, AIAA Infotech, Garden Grove, CA, June 2012.

[5] Peter M. Kogge, Benjamin J. Bornstein and Tara A. Estlin, "Energy Usage in an Embedded Space Vision Application on a Tiled Architecture," AIAA Infotech, St. Louis, MO, March 2011.

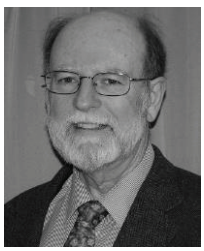
[6] Wesley A. Powell, Michael A. Johnson, Jonathan Wilmot, Raphael Some, Kim P. Gostelow, Glenn Reeves and Richard J. Doyle, "Enabling Future Robotic Missions with Multicore Processors," AIAA Infotech, St. Louis, MO, March 2011.

[7] Carlos Y. Villalpando, David Rennels, Raphael Some, and Manuel Cabanas-Holmen, "Reliable Multicore Processors for NASA Space Missions," IEEE Aerospace, Big Sky, MT, March 2011. High Performance Spaceflight

BIOGRAPHY

Gabriel Mounce *is a Senior Electronics Engineer for the Space Electronic Technology Program of the Air Force Research Laboratory's Space Vehicles Directorate. As such, Mr. Mounce directs research activities focused on increasing the reliability, survivability, and performance of space electronics used in the U.S. Air Force and other federal agency space systems. Mr. Mounce received his B.S. in Electrical Engineering from New Mexico State University and his M.S. in Electrical Engineering from the Air Force Institute of Technology.*

James C. Lyke *(Senior Member, IEEE) received the B.S. degree in electrical engineering at the University of Tennessee, Knoxville, TN, USA in 1984, the M.S. degree in electrical engineering at the Air Force Institute of Technology, Wright-Patterson Air Force Base, OH, USA in 1989, and the Ph.D. degree in electrical engineering from University of New Mexico, Albuquerque, NM, USA in 2004. He was in active duty military service with the U.S. Air Force from 1984 through 1995. Since 1990, he has supported the Air Force Research Laboratory (AFRL), Space Vehicles Directorate (AFRL/RV), Kirtland Air Force Base, NM, USA, including its precursor organizations (Weapons Laboratory, 1990-1991, and Phillips Laboratory, 1991-1998), in a number of capacities. He is currently technical advisor to the AFRL Space Electronics Branch (Space Vehicles Directorate) and an AFRL Fellow since 2008. He has authored over 100 publications (journal and conference papers, book chapters, and technical reports), and 11 U.S. patents. Dr. Lyke is an Associate Fellow of the American Institute of Aeronautics and Astronautics (AIAA) and serves on the AIAA Computer Systems Technical Committee. He was selected as recipient of the Federal Laboratory Consortium award for Excellence in Technology Transfer in 1992 and twice for the U.S. Air Force Science and Engineering Award in Exploratory and Advanced Technology Development (1997 and 2000).*



Stephen Horan *(Senior Member, IEEE) received an A.B. degree in physics from Franklin and Marshall College in 1976, an M.S. degree in astronomy in 1979, the M.S.E.E degree in 1981, and the Ph.D. degree in electrical engineering in 1984 all from New Mexico State University. From 1984 through 1986, he was a Software Engineer and Systems Engineer with Space Communications Company at the NASA White Sands Ground Terminal where he was*

involved with the software maintenance and system specification for satellite command and telemetry systems and operator interfaces. From 1986 through 2009 he was a faculty member in the Klipsch School of Electrical and Computer Engineering at New Mexico State University until retiring as a Professor and Department Head and holder of the Frank Carden Chair in Telemetering and Telecommunications. In 2009, he joined NASA's Langley Research Center working on satellite and ground systems communications. Presently, he is working on technology development activities with NASA's Space Technology Mission Directorate as Principal Technologist for Avionics. His research and teaching interests are in space communications and telemetry systems, especially for small satellite systems. Dr. Horan is a Senior Member of both the IEEE and AIAA, and a member of Eta Kappa Nu. He is the author of Introduction to PCM Telemetering Systems published by CRC Press.

***Wesley. Powell** is the Assistant Chief for Technology of the Electrical Engineering Division at the NASA Goddard Space Flight Center. He has been employed at Goddard Space Flight Center for 27 years in positions ranging from ground systems development, flight systems development, technology development, and management. Wes holds a B.S. degree in electrical engineering from the University of Maryland and M.S. degrees in electrical engineering and systems engineering from the Johns Hopkins University. Interests include onboard computing technology and applications, and low-power radiation-tolerant microelectronics.*

***Richard J. Doyle** is the Program Manager for Information and Data Science at the Jet Propulsion Laboratory in Pasadena, California. The scope of the office spans data science, autonomous systems, computing systems, software engineering, space asset protection and related topics that apply computer science principles and capabilities to space missions. Dr. Doyle is an Associate Fellow of the American Institute of Aeronautics and Astronautics (AIAA) and a member of the AIAA Intelligent Systems Technical Committee. He holds the Ph.D. in Computer Science / Artificial Intelligence from the Massachusetts Institute of Technology. He is past Executive Council member of the Association for the Advancement of Artificial Intelligence (AAAI). He is a member of the Advisory Board for IEEE Intelligent Systems. He was General Chair for the International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS), held at University City, Los Angeles in 2007, and he was Local Arrangements Chair for the International Joint Conference on Artificial Intelligence (IJCAI-09) held in Pasadena in 2009.*

***Raphael Some** (Senior Member, IEEE) is the Chief Technologist of the Autonomous Systems Division at Caltech JPL. He is a JPL Principal Engineer and the NASA Technical Authority for the joint NASA/AFRL High Performance Spaceflight Computing project. He is coming upon 40 years' experience in the development of embedded computing systems and related technologiesand is looking forward to finally getting it right!*