Section I
Administrative

Technical Topic Area:
Power Aware Computing/Communication

Lead Organization:
University of California at Irvine

Other Participating Organizations:
NASA Jet Propulsion Lab
University of Southern California

Type of Business:
OTHER EDUCATIONAL

Proposal Title:

IMPAC²T: Integrated Management of Power Aware Computation and Communication Technologies

Project Duration: 7/1/00 – 6/30/02

YEAR PROJECT COSTS Year 1 Year 2 Option Total

$800,456 $767,231 $786,940 $2,354,627

Technical Point of Contact
Nader Bagherzadeh, Ph.D.
Professor
Department of Electrical and Computer Engineering
University of California
Irvine, CA 92697-2625

Tel:  (949) 824-5689
Fax:  (949) 824-3779
Email: nader@ece.uci.edu

Administrative Point of Contact
Jeri Muniz
Office of Research Administration
Sponsored Projects Administration
160 Administration Building
University of California
Irvine, CA 92697-1875

Tel:  (949) 824-1558
Fax:  (949) 824-2094
Email: jlmuniz@uci.edu
Section II - Detailed Proposal Information

A. INNOVATIVE CLAIMS

Electronic and mechanical miniaturization gives rise to a new class of low-cost, low-power vehicles with wireless networking capabilities. One such class are Micro Air Vehicles. They form an aerial surveillance network with ground stations and a constellation of satellites, and they are critical to the success of combat missions. Other classes of aerial, terrestrial, aquatic, and space vehicles also help advance scientific discovery by enabling exploration missions and many mainstream civilian applications. We propose to contribute to this new paradigm by developing the technologies needed to overcome one of the greatest obstacles: power management. Our proposed project, called IMPAC2T, explores the system architectures and communication strategies required to manage power most effectively in the presence of dynamic mission planning and replanning. This project is the synergy of four areas: control composition, component libraries, power estimation, and cost/benefits power management. To support the development, experimentation, and validation of new power management technologies beyond state-of-the-art, the IMPAC2T project plans to develop a hardware/software codesign framework that integrates the latest ideas in power aware designs at the component, system, and network levels.

Our proposed framework empowers system-level design in ways analogous to compilers for high-level programming languages. We envision that the designer use our tools to instantiate and compose components, simulate and analyze the system with different power management policies, and synthesize and evaluate the power management mechanisms, which consist of a mix of hardware controllers and software schedulers. We capture system-level power management policies in terms of formal relationships between different power modes in high-level components. The low-level components leverage our prior experiences in parameterizable component library (PCL) in systematically tailoring the components to the chosen power management policy. Component hierarchy is well supported by our static and dynamic power prediction techniques, which account for circuit and layout effects prior to doing physical/circuit design. The synthesized system includes a power manager that coordinates power usage in the components and the communication module at run-time. The communication module controls data rate as a function of mode, available power, path loss to gateway, and interference.

Our structured framework enables and integrates key innovations in several areas. Just as compilers use language-level control flow in optimization, system-level power optimizations will require higher-order coordination structures, which are sorely missing in today's codesign tools. A second benefit is that designers are not limited to using our components; they can take advantage of today's rich collection of commercial off-the-shelf components, provided the power coordination interfaces can be defined. This is made possible by the decoupling of coordination interface and component functionality in our approach. Our components offer significant flexibility through parameterization, hierarchy, and much more accurate power estimation. To dramatically speed up power mode changes in response to dynamic mission planning and replanning, we propose to incorporate several versions of a given timing-critical and power-critical functionality at different power consumption levels and performance points. These techniques will make the most effective use of silicon for system-on-a-chip implementations. Finally, our system power manager and communication modules coordinate their power usage and data rate by means of an efficient cost/benefit distributed optimization scheme.
B. TECHNICAL RATIONALE

1. Introduction
This proposal centers on the development of an integrated CAD framework to study the overall architecture, interconnect requirements, and fault-tolerant properties of a large-scale, widely distributed, data acquisition system subject to hostile conditions. These systems must make intelligent use of available power not only as stand-alone units, but also coordinate their power usage. Power usage is intricately related to design problems at all levels of abstraction. Point tools exist today, but what is sorely missing is an integrated framework for the global characterization, evaluation, and management of power. Our efforts will focus on systems that are created as composition of reusable components from different DARPA projects as well as commodity intellectual properties (IPs) that play a key role in the targeted application. We review our application class and provide a roadmap for our approach.

1.1 Application
The key applications that will drive our work are a new emerging class of intelligent, mobile systems. They carry with them their own source of power, supplemented by solar energy when available, and power usage is particularly critical. They include the NASA X2000 and micro air vehicles.

The X2000 is a NASA program to develop the avionics for an aircraft for deep-space missions. It must be designed to be low cost, high performance, low power, low mass, low volume, and high reliability. Moreover, it must be configurable and scalable to the requirements of different missions. The X2000 must balance power consumption with subsystems that need to maintain very high data rates. The multi-mission requirement has driven the X2000 toward a symmetric, scalable and distributed architecture.

Micro air vehicles (MAVs) are hummingbird-sized flying machines [60]. They swarm to form a wireless network with air dropped ground-based sensors, fly-bys, and remote electromagnetic sensors. Because of the physical size and weight limitations on the battery, power is a central driving force in many decision decisions at design-time and run-time. The MAVs must coordinate their computation and communication among themselves as well as with ground stations and satellites in deciding where and when to process and analyze large streams of data.

To make intelligent use of power, the system must be designed to be power aware at all levels. The target platform must provide all necessary mechanisms at the low level to support the power management policies at the high level. We review elements of a power-aware architecture.

1.2 Power-aware system architecture
A system architecture defined for multiple missions must be configurable to meet the different computation demands. The proposed system architecture will leverage our prior experience with the NASA X2000 multi-mission spacecraft, which is targeting a 10-20x performance increase using an order of magnitude less power than its predecessor, the Mars Pathfinder. The system architecture, shown in Figure 1, is a loosely coupled, symmetric multiprocessor with multi-tier busses. The components include the processors, custom and programmable hardware components, memory, sensors and actuators, and a communication module. We propose to build on the proven design principles and introduce architectural innovations to take the next revolutionary leap in performance and power improvement.
1.2.1 System organization

A symmetric multiprocessor architecture is most scalable because processor cards can be added or removed to match the specific mission's demands. Having identical processor cards not only reduces cost but also simplifies the programming model, since it allows processes to be easily migrated at run-time. The system architecture is designed to address the largest power drains, namely RAM and I/O. RAMs and caches draw power during writes as the address and access lines are driven, and during reads due to pre-charging. A large RAM array can create a substantial power draw due to the larger the parasitic resistance and capacitance. Power dissipation can be reduced by organizing the RAM into multiple banks and only writing or pre-charging the required bank. When no memory instruction is active, the RAM banks should not draw power. I/O also draws substantial amounts of power. This can be especially true if the I/O uses a larger supply voltage, or if it has a large load. Both the core/output supply voltage and I/O loading are input parameters used to determine power dissipation. Another method to reducing power is asynchronous logic, which avoids the power dissipation of a large clocking network. Our proposed work will enable the evaluation of tradeoffs between locally synchronous systems and globally asynchronous designs. The hierarchical nature of the proposed system architectural organization lends itself to a scalable design where the communication links and clocking strategies can be explored to meet the design objectives.

1.2.2 Interconnect

The system will have multiple tiers of industry standard serial busses. The high-speed bus is used for streaming data, while a separate low-speed bus is mainly used for control. Serial busses can save wiring cost while operating at very high data rates, and busses like FireWire (IEEE 1394) also support deterministic real-time scheduling. By coordinating with task scheduling, we expect to be able to stream data through the processors as a systolic array without having to go through the global memory, one of the most power consuming parts of any system. Also, we propose to use controllers with multiple bus protocols on the same chip, and even reuse the same physical wires for different protocols at different times. With segmented busses, it will be possible to not only shut down parts of the bus for further power saving, but also to simultaneously use different protocols in different parts of the system. We propose to enable the evaluation of all these techniques that benefit from this class of architecture model.

1.2.3 Components

The components consist of a mix of commercial off-the-shelf (COTS) low-power components and our own parameterizable, dynamically reconfigurable hardware components. The use of COTS components significantly reduces design time for many standard features including bus controllers and communication. Also, many commercial embedded microcontrollers such as StrongARM and PowerPC are designed to be very power efficient and high performance for
many of the mission planning and control requirements. We assume that the processors has integrated bus controllers, which are also available as synthesizable IP that can be readily integrated with many other peripherals on the same processor die. Certain classes of data-regular computations, including video compression and signal processing, can be done more efficiently and with much less power using custom chips, though at the expense of longer design time and lack of programmability. We plan to address this problem by leveraging our experience in reconfigurable hardware components with the MorphoSys [63] architecture, which was designed for media and signal processing applications. Section 2.2 describes the proposed extensions.

1.3 High-level power tradeoffs

Having a loosely-coupled, symmetric multiprocessing, state-of-the-art architecture that supports flexible power management is important but it is not sufficient; it is how the architecture is used that ultimately determines the power usage. The overall mission goals must take precedence over individual systems. Energy is consumed by RF, computation, sensor, and mechanical components, and all tradeoffs must determine which of these components to use power under the specific circumstances. These techniques include relaying, dynamic reconfiguration, and migration.

In the first scenario, suppose a system’s power is too low to adequately transmit a critical message back to the ground control. As an example of coordination, the weak transmission can be detected by another nearby system, which may be able to relay the message at an adequate level, but after relaying it might not have sufficient power to complete the rest of its mission. This tradeoff may be desirable for overall mission goals.

A typical situation requiring reconfiguration occurs during a mission when a MAV or a drone aircraft is performing target search. Initially, the on-board system is looking for Regions Of Interest (ROI) where a possible target match occurs. This task can be performed on a binary image and may also be done on a low-resolution mode. Once an ROI is identified, the system will switch to full-resolution, full-color mode and look for the target in the ROI. In order to maximize mission lifetime, it is important to dynamically modify both the logic structure of the hardware as well as the recognition algorithms used in each phase. The former performs bit-level operations on a large image while the latter performs word-level calculations on a small image.

Power tradeoffs often require the ability to migrate functionality among different systems. As an example, during an airborne mission, a video camera is sending image data in real time. To minimize transmission power, video compression might need to be done on the aircraft. However, if compression consumes too much power, it may be more advantageous to transmit the raw image to the base station and have it compressed there. Another solution would be to reduce the image size (coarser resolution), or to reduce the compression quality (e.g. binary-search instead of full-search motion estimation in MPEG-2). Each one of these strategies may be feasible during a different phase of the mission, and would certainly tradeoff power, performance, and quality. Keeping an optimum balance between these metrics necessitates a careful coordination among the system in the network.

1.4 The IMPAC2T Approach to power management

To achieve the next order of magnitude improvement in power/performance, it will be imperative to explore the synergy of power management techniques at all levels. Many design parameters are all intricately related, and relatively small changes in power at one level could have a global impact on all levels. The key to enabling exploration is the separation of concerns...
at these levels, and the use of design tools to reconcile these concerns. Figure 2 depicts an overview of our proposed approach. At the component level, the library captures the essential attributes for power usage. At the level of behavioral composition of these components, designers explicitly capture system-level knowledge to enable many global tradeoffs, which would otherwise be difficult or impossible to extract or recover if embedded in implementation specific details, as is the case today. Additional design constraints imposed by the system architecture, the specific mission goals, and the operating environment will all need to be considered and reconciled using a combination of static and dynamic power management techniques.

![Table of Design Stage Support provided](image)

**Figure 2: Overview of the IMPAC³T power-aware design methodology**

### 2. System Design Framework

To enable the rapid exploration of different power management policies and implementation options, we propose a hardware/software codesign framework to bring together many innovations into one integrated design environment. The framework is intended for the development of a multi-mission platform, and for pre-mission analysis and configuration of the platform. We support the exploration at the component, system architecture, and network levels. Our proposed codesign framework is depicted in Figure 3 and will feature design tools for specification, synthesis, simulation, and library at each of these levels.

We envision that the designer capture design requirements with the specification tool, use the synthesis tools to automatically generate the power management configurations that satisfy these requirements, validate the effectiveness of the design by simulation, and build up a library by incorporating their own subsystems or newly available components. Designers can determine the optimal number of processors to employ in each system and other architecture-level configurations before the specific mission.
2.1 High-level specification

The key to enabling design space exploration is to model the system at a high level of abstraction. A high-level model of the system is constructed separately from the architecture, and it captures high-level knowledge to enable synthesis and optimization. This model consists of high-level components and composition operators that integrate the components by coordination. We propose a high-level, loosely-coupled, modal component model [7].

By high-level, we mean that the specification-level component is known to the user mainly by its functionality, interface, and power/performance characterizations. They are to be distinguished from implementation-level components (Section 2.2), which may actually be wrapped inside a higher-level component.

By loosely coupled, we mean that the component is designed to interact strictly through its interface and have little or no knowledge about the identities of those other components with which it is composed. This loose-coupling property is not only important for modularity and reusability, but it also best matches the underlying architecture.

By modal, we mean that the modes of operation of the component are explicitly stated as part of its composition interface, rather than being embedded implicitly in the component. A component
has modes for functionality, power consumption, and implementation selection. The reason we put modes on the interface is to enable the synthesis and optimization of system-wide power managers that can coordinate the power usage of different components. The mode knowledge will be necessary also for communication power management in Section 4. The implementation-specific modes refer to a new technique that encapsulates different implementations into the same high-level component for even more power/performance tradeoffs. For example, a specification-level component can encapsulate multiple hardware and software implementations, resulting in a new mode for activating each encapsulated implementation.

Our innovation at the specification level is the use of higher-order operators that compose components by defining explicit coordination protocols, which define the policy on how components interact with each other in a coherent manner [8]. These higher-order operators establish temporal correlations between the modes of different components, rather than relying on posting low-level events as done with today's approaches. They allow a local mode change in one component to be propagated to other components in the system and cause a whole other set of mode changes. Many coordination protocols can be defined and composed in terms of a small set of primitive constraints. On the small, they define policies on synchronization (priority based, round robin, etc) and resolution of intercomponent data/control dependencies. On the large, they define system-level coordination by specifying what modes need to be active at the same time or mutually exclusive, and a priority scheme for automatically changing modes in order to reach a coherent system configuration at run-time. The operators are composable with each other, and power modes can be added and refined without affecting the functional composition. These constraints will be solved by a combination of pre-run-time synthesis and run-time solver.

### 2.2 Parameterizable Component Library

The proposed system design framework relies on the characterization of a library of key components that establish the underlying basic elements used to develop the integrated architecture of a processor/communication system. The proposed Parameterizable Component Library (PCL) encompasses power consumption, performance, scalability, and even potential cost. Four distinctive methods of compiling the information are needed to generate PCL:

- **Analyzable** components are black-box models created entirely based on publicly available information, such as databooks. It captures the component interfaces, names of high-level functionalities, and their power and timing characterizations.
- **Simulatable** components are those modeled at the behavioral level for simulation of functionality, power, and timing; they are not necessarily synthesizable to an implementation.
- **Synthesizable** components are those that can be synthesized all the way down to silicon, whether they are behavioral or RTL models. These are also known as soft macro IP’s.
- **Synthesized** components, also called hard macros, correspond to layouts of utilized components.

To build the PCL, we will characterize popular off-the-shelf components and image processing components from MorphoSys[63], a DARPA funded UCI subsystem. Our detailed knowledge of the MorphoSys chip will enhance our understanding of how parameterization relates to power consumption, scalability, and fault tolerance. Additional components related to the communication fabric will be included in the initial collection of PCL elements. Our experience from the initial characterization of PCL elements will be used to formalize a methodology to include additional components. The major challenge in systematic library development is the extraction of essential characterization parameters from published sources.
For the PCL to provide the detailed power and performance parameters necessary for rapid design and analysis of system-level power optimization, we plan to leverage our previous experience with the design of highly advanced micro-architectures, including superscalar, multithreaded processor, and VLIW. At this level, the standard technique for dramatic power reduction is to turn off unused functional units by either holding the input value or clock gating. In the first method, holding the input value eliminates switching and therefore power draw by the combinational logic. However, since the registers that hold the input values are still clocked, the power draw can still be quite significant in large designs with numerous registers. The second method, clock gating, turns off functional units by holding the clock signal steady. This eliminates the power drawn by the registers and clock lines in addition to the combinatorial logic. However, clock gating can add risk to an ASIC design by introducing clock skews or glitches on the clock line. We intend to use both methods to enrich our library by combining existing basic components to develop the micro level components necessary for the IMPAC2T hardware/software codesign tool. Although IMPAC2T interacts at the interface level, our power estimation portion of IMPAC2T requires a hierarchical description of the components used throughout the design. Depending on the depth of analysis, PCL characterization (i.e. power, layout, wire length, etc.) will be used by the power estimator so that IMPAC2T meets the desired performance/power requirement.

2.3 Functional partitioning

Functional partitioning determines how the behavioral description can be mapped onto the elements in the architecture. This has been done in hardware and software designs for performance, real-time scheduling, communication minimization, and to some extent power as well. Codesign tools perform partitioning at compile time, while middleware and distributed operating systems and the system architecture support process migration at run-time. We propose a suite of novel partitioning methods for exploring power management policies and mechanisms.

Static partitioning is commonly done during scheduling or resource allocation stage in the synthesis flow. Many of these problems are NP-hard and thus should be computed statically. Another reason is that hardware is inherently statically allocated and cannot be spawned like software processes. Static schemes work well for applications whose operating environment are well under control. However, they are too limiting for our application, which must use power intelligently under very different circumstances, and they must rely on different specialized implementations and algorithms to accomplish the next order of power/performance improvement. We overcome this static limitation with our high-level component model and the corresponding partitioning scheme. To recall, our proposed specification-level component can encapsulate multiple implementations inside, each with different power/performance tradeoffs. Several static partitionings can be computed for coordinating alternative power/performance implementations of a given composition of functionality. Instead of computing one mapping, we compute multiple mappings, including multiple hardware and software ones. These statically computed partitionings are dynamically activated at different times to support the most effective power usage. They effectively enable dynamic process migration between not only processors but also between all hardware/software combinations.

For dynamically spawned software processes as well as dynamically changing network topology, it is not always possible to statically partition them, and dynamic schemes are needed. The symmetric multiprocessor architecture provides the mechanism for processors to be selectively shut down while the load is light. Optimal policies however are still NP hard to compute, and
solving the partitioning / migration problem exactly at run-time is impractical: not only could it take minutes to compute, it could also consume significant amounts of power. To support the widest possible design trade space on the system architecture, our solutions include suboptimal but efficient heuristics for dynamic power/performance optimization, as well as on-line analysis schemes described in Section 3.

2.4 System integration

Given the system specification as a composition of components, synthesis entails the realization of communication and coordination mechanisms for system integration. Coordination is a form of a protocol, and a protocol needs to be realized in some form of a controller. Controllers may be synthesized either in a distributed organization (one controller per component) or collectively in the form of a real-time/power scheduler. Given a partitioning, we propose the synthesis of system integration mechanisms for managing the coordination of power and timing.

Our prior experience with modal processes resulted in the synthesis of efficient "mode managers" for distributed architectures. A mode manager is a synthesized process that coordinates a set of processes that are mapped to the same processor; in addition, it also serves as a proxy for interprocessor coordination on behalf of the processes on its processor. The synthesis algorithm can exploit don't-care conditions in the coordination protocol when solving the constraint system statically in generating the mode manager. This has been shown to effectively minimize interprocessor communication for coordination purposes, and this translates into significant power savings. For mission-critical applications, we propose to extend the mode manager to handling dynamic constraints. This is because some goals are not known a priori with dynamic mission planning and replanning. For practical considerations, the mode manager may need to treat some goals as preferable but optional for the sake of performance or power. This will be based on designer directives without having to modify the system design.

Interprocessor communication at the high-level must be mapped down to actual bus messages that are routed and scheduled for the specific bus topology. Communication may be direct node-to-node and shared memory communications. Memory accesses are expensive in terms of power and are minimized in favor of node-to-node communications. For example, video or image processing will most likely be synthesized as having the camera stream the data directly to either a hardware block or a pipeline of processors without going through memory. The synthesis algorithm will schedule bus communication to ensure throughput and latency constraints can be satisfied with the least power. Of course, it is even more desirable to minimize bus accesses; but since each node has a limited memory capacity, the amount of work to be scheduled should have a working set that fits in the node’s memory capacity, or else it must access memory. These factors are all considered and fed back to the partitioning and synthesis loop.

2.5 Simulation

As with most codesign frameworks, we plan to develop a simulator for system-level design validation and analysis. The designer should be able to simulate individual components, the coordination protocols, a system in isolation, and a network of cooperating systems. The purpose of simulation is to validate an aspect of a design by executing a model. While it is possible to perform detailed simulation using any of the commercially available simulator, we plan to construct an efficient simulator to take full advantage of the high-level knowledge about the coordination protocols, which are defined in terms of our proposed higher-order operators. The proposed simulator will enable the development of reusable coordination policies without embedding them in specific components. Prior to system partitioning and synthesis, designers
can quickly validate the composed functionality and power independently of communication or partition-specific details. After synthesis, the coordination behavior has been refined down to low-level communication primitives that can also be simulated in more detail. Back-annotation techniques will provide designers a high-level view of their execution even when simulating at the low level.

The use of high-level abstraction will prove even more valuable when simulating a large-scale network of these coordinating systems. The designer may wish to simulate not just a system in isolation, but also its interactions with its peers over the wireless network. The high-level coordination operators can be applied to not only components on the same system but also across different systems. This will enable inter-system coordination policies to be validated and analyzed efficiently. To support detailed simulation of a system's interactions with its peers in the network, we propose a novel mixed-abstraction simulator: the system of interest is simulated at the detailed level, while its peers are simulated at the high-level to provide the workload needed to drive the simulation.

2.6 Power/Performance Analysis
This codesign framework includes several analysis tools to support the exploration of the design space. These tools analyze system quality metrics for both the computation and communication domains. These quality metrics include cost, area, performance, power, and communication quality of service (QoS). Our methodology is to simulate system benchmarks to evaluate performance and to acquire functional unit utilization metrics to be used for power estimation. The functional unit utilization values are used as parameters with a power calculation. The power of a functional block depends on several factors. First, the functional block layout will be very different depending on the function. RAMs are dense layouts which draw power when read or written. Combinatorial logic layouts are more sparsely laid out, depending on the logic function. Datapaths have some amount of regularity, while control functions are highly random. Datapaths are found in arithmetic pipelines and are also more amenable to power management. Control functions must often be enabled whenever the processor is active. A spreadsheet is generated for every functional layout block within the processor. The area and power are calculated for the processor as variable parameters add functional resources. The power for each block depends on several factors. A power equation is developed for each of the functional block types, such as RAM, datapath, and random control. RAMs have parameters for size, associativity, and banked multi-porting. The number of functional units selected affects datapath logic blocks. The number of functional units and complexity of the processor affect random control logic blocks. The sections that follow describe these methods in more detail.

3. Estimation of System Quality Metrics
A critical feature of the codesign framework is the analysis of alternate system configurations. Analysis will provide estimates of system quality metrics such as cost, area, performance, and power. Previous work in this domain [62] has concentrated on the first three metrics. As part of this proposal, we will extend the scope of the analysis tools to handle power estimation. Estimation of power consumption must be carried out both at the component as well as the overall system levels. This estimate must be accurate enough to allow for effective system-wide power management.
3.1 **Horizontal and Vertical Integration**

We propose the development of hardware models integrated in an estimation environment that is entrusted with providing feedback to the synthesis tasks. Such an estimation environment will be both *vertically* and *horizontally* integrated.

Vertical integration means that estimations must extend from the architecture to the physical level. By linking register-transfer (RT)-level estimation to accurate physical level estimators at both component level and chip level, it is possible to account for layout effect on both components (e.g. variations in area and delay with component shape and aspect ratio), and complete RT level designs (e.g. wiring effects, unused area). Using this approach, users can obtain quick and realistic feedback on design decisions made during scheduling, allocation, and binding, and get proper guidance to produce high-quality layout.

Horizontal integration with resources is key to high flexibility. Consider the dependencies between the different types of resources as well as the ordering in which the resources are allocated. In doing so, we introduce the concept of *conditional lower bounds*, which predict requirements of one type of resources subject to constraints on another type. An added degree of flexibility is that designers can arbitrarily constrain one type of resource and predict the remaining resources. This allows the designer to further explore the design space, analyze resource allocation tradeoffs, and perform a true "what-if" analysis before committing to a particular strategy and a set of resources.

3.2 **Accuracy and Efficiency**

One way to account for layout information is to actually go through a physical design procedure every time a candidate solution is generated. This is obviously infeasible, because one must explore a large number of such solutions, and evaluating each such solution takes too long. On the other hand, accurate modeling will be essential in system design. It must employ area and timing models with a finer granularity that permits the modeling of Functional Unit (FU), register, and interconnect area and delays. Timing information is based on data paths as bus-to-register delay, register-to-bus delay, FU-to-bus delay, bus-to-FU delay, FU-to-FU (chaining) delay, and control delay. To keep the design model not too costly to evaluate without sacrificing accuracy, our approach is to exploit hierarchical evaluation with controlled accuracy. The model should allow for varying degrees of accuracy in evaluation which would start in coarse mode during the initial design phases and increase in accuracy as more and more details are being examined and optimized.

As part of our prior work in the area, we have developed efficient algorithms and heuristics to support this model. The initial estimation environment we developed was targeted towards FPGAs, specifically LUT-based. According to our experimental results [23], it is clear that simple cost models do not perform well in assessing the overall design cost, and comprehensive models such as the proposed model are necessary to accurately explore the design space. In order to achieve the goal of synthesizing reconfigurable hardware while considering layout information, we must enhance the current model with accurate timing prediction capabilities. The area prediction model is a combination of constructive and analytical predictive models. We will use the same scheme for timing prediction. In addition, we need to retarget the current estimation environment to ASIC design style.

Over the past ten years, we have developed a provably accurate, multi-platform, multi-level suite of estimation tools. These tools are designed to be part of system-level design methodologies for
SOC. By multi-platform, we mean that these tools can target custom designs as well as flexible logic such as Xilinx. By multi-level, we mean that these tools can provide estimates of area and timing starting from different levels of abstraction, ranging from behavioral level to gate level. Of course, the less abstract the level, the more accurate the estimation. However, the layout effects are captured at all levels since we layer the estimates so that a gate level estimate is used by the data flow graph level estimate and so on. Not only do these tools provide area/timing estimates but also an approximate topology of the design. Such estimates can be generated in milliseconds to seconds.

The power modeling and estimation tools will use the existing area/timing estimators to obtain an approximate layout. This will in turn be used to provide estimates of layout effects that will be incorporated into power evaluation models. The challenging issue will be to develop models that capture the input data variation. Indeed, several methods have been proposed, most notably the Dual Bit Type method (DBT) case.

We propose to improve the accuracy of power estimation by taking advantage of higher level structures in the component specification. Such structures are represented at algorithmic or data flow graph levels, and can be used to narrow down the scope of the data input. Furthermore, regularity can be exploited to reduce the modeling complexity.

### 3.3 Power Estimation for Programmable Components

For programmable components, such as microprocessors, a somewhat different approach is required. In this case, power consumption will depend on the specific data values, the type of machine instruction, and most importantly, the memory access. Another main difference is that in addition to off-line power estimation, which is done at or before design time, there is also an on-line power estimation that provides estimates (or refinements of off-line estimates) during runtime.

Machine instructions can be characterized a priori by profiling, which can yield accurate estimates for instructions that do not access memory off-chip. Memory-related power consumption becomes a significant factor in deep submicron (DSM) technologies. This factor will depend on the memory architecture, the physical location where information is accessed, the bus architecture, and the access pattern itself. The main difficulty in this area is that computation is specified at a high level of abstraction (e.g. C code). In this case, one must estimate memory requirements for that given computation. For a given memory architecture, one must estimate the sizes of the different hierarchies (cache, RAM, etc..) as well as characteristics of the bus architecture (e.g. bus capacitance).

Once the memory architecture is characterized, we need to consider the memory access pattern in order to estimate the transition density for each bus. Bounds on such values can be established either through high level characterization of library functions (using statistical sampling), or by trace simulation and profiling. Techniques such as Gray coding can also be employed to minimize address line transitions. Thus, for a given configuration, upper and lower bounds can be computed and propagated to the system level synthesis tool which will use these values to make design decisions. Additionally, such bounds can be used at runtime for adapting the system to various power levels in real time. As an application is run, additional profiling data can be accumulated by the system (assuming that this accumulation does not consume too much power) and used to refine the bounds into more accurate estimates. A global history database can thus be...
updated and used for future design endeavors. This is essentially the basis of our on-line power estimation strategy.

4. Communication Module Power Management

Sections 2. and 3. addressed the design stages from component evaluation through behavioral composition, architectural exploration, and pre-mission configuration. This section continues by addressing communication module power management in both pre-mission configuration and the dynamic power management design stage (Figure 2).

4.1 The Problem

Our goal in this portion of the project is to dynamically control the transmission rate of a system. Our approach uses a novel economic framework in which this distributed optimization is performed through exchange of benefit and cost information between the system power managers and communication modules.

The transmission rate should reflect:
- the current mode of operation,
- the available power in the system,
- the current attenuation of the communication signal, and
- the impact upon other systems in the neighborhood.

Different modes of operation (e.g. detection, identification, ranging) will result in different priorities on communication at the current time. If communication is currently a low priority, then the communication module should respond either by lowering the bit rate or by delaying the transmission until a later time. Information about the mode comes from the mission configuration.

The available power in the system plays a crucial role in determining transmission rate. If the current system power is low, then the communication module should respond either by lowering the bit rate or by transmitting only at times at which a reasonable bit rate can be obtained at a low power level. Information about these factors come from the system power manager and task scheduler, as discussed above.

The current attenuation of the communication signal changes with distance between the system and the destination of the transmission, and with fading and shadowing. If the mode of operation indicates that communication is currently a low priority, then the communication module should respond by choosing to transmit when the attenuation is low.

Finally, the impact upon other systems in the neighborhood must be considered. All systems within some reuse distance share the same communication capacity, and thus a high usage of these resources by one system will decrease the maximum transmission rates of other systems. For instance, if CDMA (Code Division Multiple Access) is used, then a system's transmission power impacts other systems by producing interference. When communication resources are constrained, the communication module should respond by lowering the transmission rate if the required transmission power has a strong impact upon other systems.

Of course, all four of these factors must be considered together. The communication module, therefore, can not make this decision alone. It must obtain information from other modules.
within the system regarding the current mode and the available power. It must also coordinate transmission power with other systems within its neighborhood. Coordination must therefore be implemented not only within a system, but also within the network.

4.2 Our Approach

We pose this system- and network- coordinated choice of transmission rates as a distributed optimization problem. The optimization metric is a function of the transmission rate of each system. Since these systems form a cooperative network, it is reasonable to assume that we can define a numerical benefit to each system as a function of its transmission rate. We model the network optimization metric as the maximization of the total benefit of all systems within the network, i.e. we are trying to coordinate transmission rates so as to maximize the benefit to the entire network. As a result, this tool should translate an implementation-level system model and a network economic model into a system model with optimized communication power management (Figure 3).

The solution to this problem must be obtained dynamically, since each of the four factors that affect the transmission rate vary in time. Furthermore, the solution must be obtained in a distributed fashion, since a full exchange of information among all systems in the neighborhood would likely entail too much overhead. This work must therefore occur during the dynamic power management design stage (Figure 2).

Our approach is to distribute the optimization among all the communication modules in the network. Modules exchange information between each other and with their system power manager. The result of this information exchange therefore affects both system-level power management and network-level power management. We propose a minimal exchange between these modules, in order to keep the overhead low. This minimal exchange is accomplished through the use of benefit and cost metrics. In the following subsections, we describe the network topology, the benefit and cost functions, and the resulting distributed optimization.

4.3 Network Topology

The information exchange, for purposes of optimization, should mirror the physical topology of the network. We do not address in this proposal how the network topology is chosen. We assume for purposes of discussion that systems self-organize into a hierarchical topology, here assumed to be a tree. We are primarily concerned with the lowest two levels of the tree: each system transmits up to a higher level system called a gateway.

The topology is assumed to be dynamic: each system and gateway may be mobile. We assume that systems are assigned to gateways on the basis of current path attenuation, so that a system's assigned gateway may change with movement. At a particular time, the neighborhood of a gateway is defined as those systems that share a gateway. Other topologies may easily apply to our approach, as long as there are distinct neighborhoods that share communication resources.

4.4 Benefit

The benefit to a system due to transmission at a certain rate is modeled as a family of nondecreasing functions of that rate. This family of functions is defined as part of our concept of modal interfaces (introduced in section 2.1), and includes one function for each application mode, e.g. detection, identification, and ranging. The benefit is a numerical measure of the
current importance of communication to the system tasks. Two characteristics of these functions affect transmission rate: the shape of a particular function, and the relative scale of different functions within the family.

If a mode is insensitive to the delay required to transmit information, then the function for this mode would be fairly flat, indicating that a low bit rate is appropriate if system or network resources are tightly constrained. On the other hand, if a mode is sensitive to transmission delay, because fast communication is integral to the mission, then the function for this mode would rise quickly at those bit rates that are desirable.

Within the family of benefit functions, if the benefit function for one mode dominates the benefit function for another mode over all transmission rates, then this mode places a higher importance on communication. The ratio of these benefits will be used to give more network communication resources to the systems that are in such communication-intensive modes.

Such functions have been used in both wireline and wireless networks to distinguish between the performance requirements of non-real-time (e.g. data) and real-time (e.g. voice and video) services. Non-real-time services typically have a benefit that is steadily increasing, indicating that the perceived application performance is continuously increasing with throughput. In contrast, real-time services typically have a benefit function that takes a jump at a certain threshold, indicating that the perceived performance is unacceptable below this threshold, but not substantially improving above this threshold.

4.5 Costs

The two costs associated with transmission are power drain and interference. These two costs are assumed to be additive, and are measured on the same scale as the benefit.

The power drain cost is calculated by the system power manager, which is a dynamic constraint solver. This cost is a function of the current battery level and of competing demands for power from other system components. By setting the ratio of the power drain cost to the benefit, the system power manager can set a rough target for desired transmission rate. By lowering this ratio as the battery drains, the power manager can slow down the transmission rate and thereby lower the power drain by the communication module.

Similarly, the functional dependence of the communication power drain cost on the power demands from other system components gives the system power manager control over the tradeoff between computation and communication. If the current mode is computationally intensive, then the communication power drain cost can be raised, resulting in a lowering of transmission rate.

The interference costs are calculated by each gateway. We describe here the calculation for transmission from a system to a gateway, called the uplink, within a single neighborhood; extensions to downlink and to multiple neighborhoods will also be considered in the project as options. Each such cost is a function of the negative effect that transmission from one system has upon the other systems' throughputs in that neighborhood. This negative effect, called an externality, is equal to the total decrease in benefit to other systems caused by transmission by the nominal system.
At light loads, this decrease may be negligible, resulting in a small interference cost. At high loads, this decrease is substantial, resulting in a significant interference cost. A higher interference cost will result in a system's communication module choosing a lower transmission rate. This exchange of information results in coordination between systems within a neighborhood to optimally share scarce communication resources.

For example, if CDMA is used for transmission, then the uplink throughput for a particular system is a function of the received signal to interference plus noise level at the gateway. Since the interference is simply the sum of other systems' received signal power, the interference cost can be simply calculated as a function of each system's received power. This cost is therefore determined by the system's transmission power and the attenuation from the system to the gateway.

### 4.6 Distributed Optimization

The distributed optimization will be performed individually by each system communication module after an information exchange, as pictured in Figure 4. The module obtains the benefit and power drain cost from the system power manager, as a function of transmission rate and power usage respectively. The module maps transmission rate into power requirements based on the modulation and multiple access used. The module also obtains interference cost from its gateway, as a function of received power. Using knowledge of the current attenuation, the system also translates this into power requirements in the communication module.

![Figure 4: Communication module information exchange within a neighborhood](image)

This information exchange can be minimized by exchanging only the values at the operating point, rather than the entire function. At a proposed transmission rate and associated power level, the system power manager tells the communication module the current benefit and power drain cost. In CDMA, the interference cost can be easily and efficiently communicated by the gateway by broadcasting a current price per unit received power to all systems within its neighborhood.

This information exchange must be updated on a time scale that is fast enough to capture the dynamics of the system. Updates occur at three locations: the system power manager, the communication module, and the gateway. The system power manager updates the benefit and power drain cost based on changes in the battery life, competing power demands from other
components, and the operational mode. The gateway updates the interference cost based on changes in the total demand for communication resources in the neighborhood. For instance, if CDMA is used, the gateway can obtain information about marginal benefit from each system and use a gradient algorithm to set the price per unit received power in order to maximize total benefit. Finally, the communication module updates its transmission rate, and corresponding power usage, using an iterative algorithm to maximize its surplus, defined as benefit minus cost.

4.7 Potential Accomplishments

This proposed scheme has the following attributes:

- It can coordinate communication module power usage with other system components, through specification of the power drain cost.
- It can coordinate use of scarce communication resources throughout the network, through use of interference costs.
- It can accommodate a wide range of application modes, since each mode can be specified using a benefit function.
- It can be used to implement a tradeoff between communication and computation, by properly scaling each mode's benefit.
- Transmission rate, and communication module power expenditure, will vary with path loss from that system to the gateway and will adapt to handoffs to a new gateway.

5. Benchmarking and Evaluation Testbed Suite

The Jet Propulsion Laboratory (JPL) has a state-of-the-art integrated testbed that is used to prototype avionics systems for NASA’s unmanned space exploration. The IMPAC\textsuperscript{2}T tools will be integrated into this testbed so as to enable prototyping testing and validation of the proposed power aware technologies. The testbed, located at JPL, will incorporate a set of key applications for existing and emerging space missions, e.g. NASA’s New Millennium, X2000 Program or Planetary Distributed Rovers program. This testbed, and the associated applications, will also be made available to other researchers in the DARPA community.

JPL shall investigate the feasibility and advantages of using IMPAC\textsuperscript{2}T tools to support complex real-time control tasks and computation-intensive applications for space missions. We plan to demonstrate with a prototype design that IMPAC\textsuperscript{2}T tools will synthesize significantly lower power and higher performance electronics as compared to existing designs and systems while offering a greater degree of flexibility.

JPL’s overall role is to integrate, test and validate a breadboard design that will demonstrate the IMPAC\textsuperscript{2}T tools for X2000 missions. This will be accomplished in the following three stages:

- Integrate, test and validate any single component within the Center for Integrated Space Microsystems avionics testbed.
- Integrate, test and validate a breadboard within the Flight System Testbed.
- Integrate, test and validate the breadboard design in a real spacecraft environment, such as the X2000 Spacecraft Testbed.

In performance of this task plan, we will also explore:
1. Low power high performance issues to maintain high bandwidth with low power and include power management features such as sleep mode to conserve power.

2. System total power management issues such as in a real spacecraft system that includes not only the core avionics system but also the complete I/O system and a relevant workload.

3. Power management for networking and communication issues such as in the case of the Planetary Cooperating Rovers.
C. Deliverables

1. System-Level Tools
   - **Design Authoring.** The authoring tool lets a designer instantiate reusable components from a component library, define coordination protocols, and use them as higher-order operators for composing the high-level behavior of the system. Also captured are the design constraints on power, performance, and resources (e.g. number of processors allocated).
   - **System-Level Simulation.** The simulation tool lets a designer experiment with a virtual prototype of the system for the validation of functional correctness, timing, power usage, and resource utilization such as CPU or bus activities.
   - **Functional Partitioning.** This tool determines how the functionality can be partitioned statically and migrated dynamically in order to satisfy the power goals on the given target architecture configuration.
   - **System Integration.** The system integration tool is responsible for realizing the mechanisms needed for integrating the components into a complete system. These include the synthesis of coordination controllers from the higher-order composition operators. These controllers are realized in the form of additional software processes, configuration bits for hardware controllers, and combined real-time/power scheduler.
   - **Power Management Design Techniques.** In order to have the most effective design from the point of view of power consumption, efficient techniques must be introduced at all levels. At the architecture level, such ideas as logic block isolation or memory banking can be envisioned.
   - **Partitioning of the Power Savings Components.** Obviously, the larger the blocks bounded between two sets of registers, the greater the potential for power savings. Conversely, however, with larger protected blocks, comes a reduced chance of finding idle logic. One important part of the work will be to identify suitable candidates for protection and optimization of the size of those blocks.
   - **Component Simulator.** Design of a detailed simulator with appropriate parameters in order to analyze the power consumption of the various components. It will allow the designer to evaluate power consumption in a dynamic manner, performance, as well as resources utilized.
   - **Complete Integration with System Level Tools, Estimation Tools, and Network Level Tools (option).** This will include a complete interfacing with the other levels, including synthesis of the behaviors.
   - **Parameterizable Component Library (PCL).** PCL provides the component-level input needed to perform the system-level simulation, analysis, synthesis, and general exploration of the design space. The parameters are power, timing, functional behavior, area, etc.

2. Estimation Tools
   - **Pre-synthesis estimation of area and delay - Version 1.** Prior to synthesis, place and route: this tool will provide best and worst case estimates of circuit area and delay. It can therefore provide the user with immediate feedback on the hardware costs and expected performance of the selected programs to be synthesized.
   - **Power Estimator - Version 1.** A first version of a power estimation tool is developed. This version will estimate datapath and sequential logic, without memories.
Component Rater. This tool will use the estimation tools developed above to characterize a given component. This can be coupled with PCL as well as the system level tools to generate tradeoff points for each component reflecting area-cost/performance/power tradeoffs.

Pre-synthesis estimation of area and delay - Version 2. Will incorporate memory and interconnect estimates from behavioral specs.


System Integration (option). The developed tools will be integrated within the IMPAC\textsuperscript{2}T framework. Spreadsheet-like tools will be developed to allow interactive as well as automatic tradeoff evaluation of metrics.

On-line power estimation (option). Will investigate on-line refinement of power estimates. Tradeoffs will be evaluated and design techniques for augmenting the system with such capabilities will be researched. Depending on the outcome, appropriate prototype tools will be developed and integrated within the IMPACT framework.

3. Network-Level Tools

Interface Characterization. Characterize the benefit and cost information provided to the communication module, and the form of the information provided by the communication module. Develop understanding of the applications and of the network topology. Propose initial benefit and cost functions.

System-Coordinated Communication Module Power Management. Develop the communication module power management optimization procedure within a single system, given static benefit and cost information. Verify system-wide optimization.

Network-Coordinated Communication Module Power Management. Develop procedures for gateways to measure competition for communication resources on the uplink and broadcast interference costs. Verify network-wide optimization.

Downlink Analysis (option). Develop procedures for gateways to measure competition for communication resources on the downlink and broadcast interference costs. Verify network-wide optimization.

Analysis of Mobility (option). Simulate the variation of power with changes in attenuation and gateway handoff. Analyze the dynamics of communication module power in response to system mobility. Verify convergence of power assignment.

Multiple Neighborhood Coordination (option). Develop procedures for exchange of interference cost information between gateways. Develop coordination mechanism between gateways to instigate handoff on the basis of local communication congestion. Implement in mobility simulation to verify effectiveness.

4. Benchmarking and Evaluation Testbed Suite

X2000 System Characterization. Characterize energy versus performance profiles for a baseline family of subsystems from existing X2000 prototypes. Identify different scenarios for X2000 subsystems where power aware technologies can be the enabling technology. Benchmark IMPAC\textsuperscript{2}T tools against these existing X2000 subsystems in terms of performance versus power consumption and power management.

X2000 System Prototyping & Integration. Incorporate IMPAC\textsuperscript{2}T tools into new X2000 prototypes. Investigate insertion into reconfigurable systems such as JPL’s evolvable hardware program. Demonstrate an X2000 subsystem prototype designed with IMPAC\textsuperscript{2}T tools at JPL’s integrated testbed.
♦ **Distributed Rovers Characterization and Testbed.** Study the distributed rovers program computational environment and identify its architectures and power requirements. Incorporate IMPAC²T tools in the program from the early hardware design stages. Develop, using IMPAC²T tools simulations for power estimation and management versus performance and integrate with existing functional software simulations. Simulate the fully integrated virtual rovers fleet.

♦ **X2000 and Distributed Rovers Integration (option).** Integration of IMPAC²T tools into X2000 future deliveries. A full demonstration of X2000 subsystem prototype designed with IMPAC²T tools in real spacecraft environment. A demonstration of distributed rovers prototypes at JPL’s avionics testbed.
D. Statement of Work
These tasks are divided into the same categories as the deliverables in section C.

1. TASKS in YEAR 1

1.1 System-Level Tools
   ♦ Task 1.1.1 - High-Level System Modeling and Coordination Synthesis techniques
   ♦ Task 1.1.2 - Architecture Definition
   ♦ Task 1.1.3 - Static Functional Partitioning
   ♦ Task 1.1.4 - Power management design techniques
   ♦ Task 1.1.5 - Partitioning of the power savings components (1st version)
   ♦ Task 1.1.6 - Compilation of simulatable and analyzable macros
   ♦ Task 1.1.7 - Methodology development for the organization and interfaces to PCL

1.2 Estimation
   ♦ Task 1.2.1 - Pre-synthesis estimation of area and delay (1st version)
   ♦ Task 1.2.2 - Power Estimator (1st version)
   ♦ Task 1.2.3 - Component Rater.

1.3 Communication Module Power Management
   ♦ Task 1.3.1 - Interface Characterization
   ♦ Task 1.3.2 - System-Coordinated Communication Module Power Management (1st version)

1.4 Benchmarking and Evaluation Testbed Suite
   ♦ Task 1.4.1 - X2000 System Characterization

2. TASKS in YEAR 2

2.1 System-Level Tools
   ♦ Task 2.1.1 - Authoring Tool (1st version)
   ♦ Task 2.1.2 - Dynamic Functional Partitioning
   ♦ Task 2.1.3 - High-Level Simulation (1st version)
   ♦ Task 2.1.4 - Partitioning of the power savings components (2nd version)
   ♦ Task 2.1.5 - Component Simulator
   ♦ Task 2.1.6 - Compilation of synthesizable macros
   ♦ Task 2.1.7 - PCL benchmarking and system integration

2.2 Estimation
   ♦ Task 2.2.1 - Pre-synthesis estimation of area and delay (1st version)
   ♦ Task 2.2.2 - Power Estimator (2nd version)

2.3 Communication Module Power Management
   ♦ Task 2.3.1 - System-Coordinated Communication Module Power Management (2nd version)
   ♦ Task 2.3.2 - Network-Coordinated Communication Module Power Management
2.4 Benchmarking and Evaluation Testbed Suite

- Task 2.4.1 - X2000 System Prototyping & Integration
- Task 2.4.2 - Distributed Rovers Characterization and Testbed

3. TASKS in YEAR 3 (Options)

3.1 System-Level Tools

- Task 3.1.1 – Authoring Tool (2nd version)
- Task 3.1.2 – System-Integration
- Task 3.1.3 – High-Level simulation (2nd version)
- Task 3.1.4 - Complete integration with system level tools, estimation tools, and network level tools
- Task 3.1.5 - Addition of synthesized components
- Task 3.1.6 - Prototype design evaluation using PCL components

3.2 Estimation

- Task 3.2.1 - System Integration.
- Task 3.2.2 - On-line power estimation.

3.3 Communication Module Power Management

- Task 3.3.1 - Downlink Analysis
- Task 3.3.2 - Analysis of Mobility
- Task 3.3.3 - Multiple Neighborhood Coordination

3.4 Benchmarking and Evaluation Testbed Suite

- Task 3.4.1 - Distributed Rovers Testbed & Integration.
### E. Schedule of Milestones

<table>
<thead>
<tr>
<th>Category</th>
<th>Tasks</th>
<th>Year One</th>
<th>Year Two</th>
<th>Year Three</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Q1</td>
<td>Q2</td>
<td>Q3</td>
</tr>
<tr>
<td>System-Level</td>
<td>1.1.1</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.1.2</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.1.3</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.1.4</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.1.5</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.1.6</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.1.7</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.1.1</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.1.2</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.1.3</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.1.4</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.1.5</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.1.6</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.1.7</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3.1.1</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3.1.2</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3.1.3</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3.1.4</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3.1.5</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3.1.6</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Estimation</td>
<td>1.2.1</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.2.2</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.2.3</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.2.1</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.2.2</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3.2.1</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3.2.2</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Network-Level</td>
<td>1.3.1</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.3.2</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.3.1</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.3.2</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3.3.1</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3.3.2</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3.3.3</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Benchmarking</td>
<td>1.4.1</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.4.1</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.4.2</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3.4.1</td>
<td>A</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A: Task is active in that time period.
C: Task is completed at end of that time period.


F. Technology Transfer

It is the expressed goal of this project to develop technology that will have both military and industrial impact, and to transfer this technology out of the laboratory and into real-world applications as quickly as possible. The primary technology transition path is embodied in the partnership between University of California Irvine (UCI), and the NASA Jet Propulsion Laboratory (JPL). Beyond the tasks specifically described in this proposal, IMPAC$^2$T will have potential transfer to space technology in the following manners:

- Development of a prototype design automation tool integrated in a JPL spacecraft testbed for emerging space missions in the next century, which involves highly capable, autonomous, and highly miniaturized robotic spacecrafts.

- First-time demonstration of power aware technologies in a space mission on a key set of benchmark applications to show orders of magnitude improvement in terms of power, performance, and flexibility.

- A combined metric of performance that compares the performance of the target computing design to conventional spacecraft designs, from the point of view of both power consumption and performance.

In addition, the procedures developed for communication module power management can be easily adapted for use in commercial 3$^{rd}$ generation wireless networks. These networks are expected to be based on wideband CDMA technology, and to integrate a wide variety of service types. Resource allocation is expected to be a key issue. If processing gain is chosen in accordance with the service type, then the Quality of Service is determined by the transmitted power.

We would expect new mappings would need to be developed to transfer this work to 3$^{rd}$ generation wireless networks. First, whereas the application mode here depends on mission objectives, in commercial networks the mode will depend on the traffic type, e.g. voice, data, or video. Second, whereas the network topology here is expected to be a mobile ad-hoc network, commercial networks will be either ad-hoc or have stationary base-stations. Finally, whereas the network objective here is assumed to be to maximize total user benefit, commercial networks might have revenue maximization as an objective. With these mappings, most of the results of this project should transfer to such commercial networks.
G. Related Work

Previous work on system-level dynamic power management attempted to minimize power by slowing down the system enough so as not to violate performance constraints. This is also complicated by potential penalties associated with changing power modes. Stochastic models derived from profiling have been proposed to consider system-wide effects in policy optimization [19]. However, if the actual workload has different characteristics from the model, the power manager may fail to save power due to poor adaptability. On the other hand, more adaptive techniques [9] are not as deterministic. Rate-monotonic scheduling (RMS) algorithm [13], which is supported on virtually all real-time operating systems, has the attractive property that feasibility can be determined statically, and has been extended to consider power. However, RMS is applicable only to preemptively scheduled tasks on uniprocessors; it is not applicable to multiple processors or mixed hardware/software systems.

More general power management techniques are structured as part of on-line configuration planners. While they are can be adaptive by solving general constraints, they require significant amounts of computation on the order of minutes on modern CPUs, even if solved heuristically [20],[18]. This may be acceptable for space missions but not for other event-rich missions where reaction time must happen in seconds or less.

Hardware/software codesign tools to date have focused on low-level system integration [10][17]and simulation [61]. They can help with vertical system integration by generating protocol adapters and device drivers. However, designers are responsible for horizontal integration by ensuring that all data communications and control coordination obey the same protocols. Control is especially problematic due to the lack of appropriate high-level abstraction, and power management is a control task. So far, codesign tools help generate centralized control from a hierarchical, state-based description [12][6], but it is not reusable and not distributable to multiple processors, unless specified in a distributed way in the first place [14]. Automated distributed control generation is done for coordination languages, though these are for mainly software systems. The only codesign system that generates and optimizes distributed control is our previous work with modal processes.

Coordination protocols can be represented several ways. Today's codesign frameworks take two approaches to coordination protocols. One way is to limit the coordination to very specific classes of data-regular coordination protocols, such as synchronous dataflow (SDF), where processes coordinate their execution based on data dependencies. While such simplifying assumptions enable many optimization techniques, specifically static scheduling of data graphs on a variety of architectures, the need to handle more dynamic or conditional behavior eventually leads to extensions to which most optimizations no longer apply. On the other extreme, models proposed for control-dominated applications [12][6] invariably push the burden of authoring the coordination behavior to the designers, who have no choice but to manifest the protocol in terms of scattered, unstructured low-level signaling primitives. As a result, such systems are difficult to reason, have no modularity, and do not lend themselves to automatic synthesis and optimization. UML [14] lets designers define it imperatively in terms of finite state machine controllers, each with a different role, and each component in a composition plays a role. Coordination languages
in general specify the sequencing of events and synchronizations required at the transaction level. Higher-order composition operators are used by [2], but it needs to be extended to handling power constraints and dynamic reconfigurability.

The VSI (Virtual Socket Interface) effort by industry and academia in the past couple of years [15] is tangential to our PCL approach. The main objective for VSI is to develop a set of IPs with standard interfaces for the development of next generation Systems-On-Chip (SOC). In order to meet a widely accepted industry standard, many high level design issues that are the focus of IMPAC²T are not addressed. Our PCL approach has several advantages over previous techniques. Because of the utilization of high level operators, the interface include component definitions from the highest level (analyzable) to the lowest levels (synthesizable, simulatable and synthesized). This will allow our tool to work at the low level and in conjunction with VSI as well as more abstract models of systems level design.

Minimizing power consumption has become an area of increasing concern for a number of applications centered around embedded processors. Initial effort have born on the low-level devices, namely by designing components which incur limited power drains. Such efforts include the work on adiabatic device design [47], [48], [49], the main thrust of it being the reuse of energy stored and otherwise dissipated during a transition. Power consumption occurs when a voltage transition takes place and the power expended is a linear function of the capacity of the wire on which the charge is carried. By conserving the charge which has been expended during a transition, one can limit the amount of charge that must be brought from outside the devices.

Alternatively, one may reduce power consumption with higher-level techniques. First, at the architecture level, a number of approaches can be envisioned which avoid large capacitance wires, wires driving many loads and large subcomponents which are logically idle and yet may see man transitions. Consequently, obvious sources of large capacitance such as clocking wires and addressing lines in memory should be minimized as much as possible.

Special purpose designs and ASICs have been invented for such algorithms such as FFT [50]. Also, many research efforts [44],[45], [46] have demonstrated the feasibility of constructing DSP in low power configurations. It has also been envisioned to design complete subsystem-level components such as the ARM IP [58]. Finally, efforts at the compilation level or even the algorithmic level are promising in that they offer a high-level view of the applications [53].

Current research in power estimation is done either at the gate level or architectural level [59]. Gate-level power analysis and estimation form the basis for most efforts, since they provide accurate estimates by capturing low-level design parameters, such as wiring, routing, and crosstalk. However, such tools are too low-level to incorporate into a system level design framework such as the one envisioned for IMPAC²T. High-level, architectural power estimations are needed for system-level design. Previous work at this level has not kept up with changes in technologies and the dominant contributing factors to area, performance and power. For example, in deep-submicron designs, low-level design parameters, such as wiring, routing, and crosstalk, are now primary contributors at the high level. The issues are further compounded for flexible systems such as reconfigurable logic where programmable interconnect constructs can amplify the wiring effects even further.
At the network level, the communication module power management discussed here is related to the literature on dynamic resource allocation for wireless networks. Whereas the application mode discussed here depends on mission objectives, in commercial networks a similar distinction is made based on the traffic type, e.g. voice, data, or video. Much of the earlier work in this area has focused on a single traffic class -- see [22] and [25] for surveys of dynamic channel allocation in narrowband TDMA, and [26]-[28] for examples of power control in CDMA. In our context, these works would be interpreted as only addressing a single application mode.

Multiple application modes would roughly correspond to commercial multimedia networks. Initial work on the design and performance of a multimedia CDMA network was reported in [29]-[30]. The idea of using power control in CDMA to satisfy QoS constraints associated with heterogeneous traffic flows has been addressed in [31]-[34]. Dynamic assignment of processing gains for a time-slotted packet CDMA model is studied in [35]. Such multimedia CDMA research, however, typically sets power levels to achieve target QoS of each media type. In contrast, IMPAC^2T will produce a range of power levels which not only differ among modes, but which can be also traded off versus power usage in other system modules.

Finally, the economic approach proposed here for distributed optimization of transmission rates (and transmission power levels) bears some resemblance to economic approaches used in wireline networks. A number of papers have observed that the overall utility of the network can be greatly increased by matching QoS to individual user requests and priorities (c.f. [36]-[39]). A few have proposed schemes for dynamically adjusting price to allocate networks resources, e.g. buffer and bandwidth, to individual traffic flows (c.f. [40]-[43]). We aim to modify and extend these schemes to both system-wide and network-wide power control for the applications considered here.
H. List of Key Personnel

<table>
<thead>
<tr>
<th>UCI</th>
<th>JPL</th>
<th>USC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nader Bagherzadeh</td>
<td>Nazeeh Aranki</td>
<td>Jean-Luc Gaudiot</td>
</tr>
<tr>
<td>Pai H. Chou</td>
<td>Nikzad &quot;Benny&quot; Toomarian</td>
<td></td>
</tr>
<tr>
<td>Scott Jordan</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fadi J. Kurdahi</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1. University of California Irvine

Nader Bagherzadeh has been involved in the design of advanced microprocessor architectures for the past ten years. He was lead designer of the first 4-issue VLIW microprocessor called VIPER. The VIPER architecture demonstrated pioneering work in the areas of processor optimization for routing operand data through bypass circuitry. Moreover, it utilized a unique pipeline architecture with a refined addressing mode that mitigated the development of future VLIW processors by industry. Later he worked on an out-of-order issue superscalar microprocessor called SDSP targeted for multimedia computation intensive applications. This study led to several new discoveries in the areas of instruction fetching and register renaming hardware optimization. He is the co-PI on the Darpa-funded MorphoSys Project (www.eng.uci.edu/morphosys).

Pai H. Chou has been doing research in hardware/software codesign of embedded systems for the past nine years. He was a contributor to the Chinook cosynthesis project (www.cs.washington.edu), where his key contributions include interface synthesis, hard real-time scheduling, and most recently the high-level component model for control composition and synthesis of distributed embedded systems.

Scott Jordan is concerned with the integration of voice, data, and video on computer and telecommunication networks. In particular, he focuses on modeling and analysis of resource allocation and quality of service. In one project, he is concerned with the efficient allocation of the key network resources, buffer and bandwidth, among heterogeneous services, in a manner that guarantees performance to each in accordance with its desires. This project involves application of economic theory to determine a pricing mechanism for network services and resources. In a second project, he is concerned with allocation of power and codes in integrated service wireless networks. This project involves distributed optimization to shift capacity toward those users who will most benefit or who will generate the highest revenue.

Fadi J. Kurdahi has been doing research in Design Automation and VLSI design for over thirteen years. Specifically, he has researched and developed CAD tools for architectural synthesis and estimation and has published numerous papers on the subject of linking layout information to synthesis. This becomes particularly crucial for the projected fabrication technologies that are likely to implement the proposed DRA. Recently, his work has focused on FPGAs as target implementations. His work on estimation for FPGA resulted in a suite of prediction tools that are highly accurate yet runtime efficient. Architectural synthesis techniques such as scheduling and binding are being developed to utilize these estimator and generate "first-time-correct" silicon. He is the PI of the DARPA/ITO/ACS MorphoSys Project.
2. JPL

Nazeeh Aranki is a Research Engineer at the Jet Propulsion Laboratory. He is a member of the adaptive and evolvable hardware team at the Center for Integrated Space Microsystems (CISM). Since joining JPL his research interests have included reconfigurable hardware, evolvable hardware, neural networks, parallel processing, digital signal and image processing and compression. He was a key contributor in the development of an FPGA-based neuroprocessor for automotive control and diagnostics.

Nikzad "Benny" Toomarian is a senior member of the technical staff in JPL Distributed Computing Architectures within the Avionics Equipment Section. He is currently leading the Revolutionary Computing Technologies (RCT) element of the Center for Integrated Space Microsystems. The main purpose of the RCT task is to explore research into advance computing concepts that will potentially mature in the 10-20 year time frame and may fundamentally change the way computing is performed. Although RCT is in general a long-term research task, some elements of this research are targeted for near term, 2-5 years, such as reconfigure and evolvable computing. The mid term elements, 5-10 years, include quantum functional devices and biological computing. Dr. Toomarian has a multi-disciplinary background in sensitivity analysis of nonlinear dynamical systems, neural networks theory and applications, distributed and massively parallel computing. He is also investigator and Task Manager on a number of NASA, DOD and DOE programs related to neural networks, distributed computing, predictive intelligence and interactive science. Currently, he is involved in several tech-transfer initiatives related to neural networks technology.

3. University of Southern California

Jean-Luc Gaudiot has been researching advanced architectures for over 15 years. More specifically, he has investigated the issues of programming and implementing large-scale multiprocessor systems and has published numerous papers on the issues of large structure representations in multiprocessor systems. This work has led to applications in designing efficient models for parallel architectures including networks of workstations. In his recent work, he has worked on integrating multiple processors in single chip architectures. For his efforts on solving the programmability problems of multiprocessors, he is now a Fellow of the IEEE.
I. DESCRIPTION OF FACILITIES

1. UCI

The available facilities at UCI include a large number of UNIX and PC workstations supported by several high-powered SUN servers. These computers are networked via a campus-wide gigabit Ethernet backbone. In addition, the PIs have access to a wide variety of CAD tools for both hardware and software design. Such tools include: Mentor Graphics, Compass, Vantage, Cadence, Synopsys and Xilinx. A fully-equipped FPGA design lab is available for usage to prototype hardware using the Xilinx XC4000 FPGA evaluation boards. This is supported by a suite of the latest Xilinx XACT tools running on both PCs and SUNs. Campus-wide computing resources include a CONVEX supercomputer and access to Crays in the San Diego Supercomputer center.

2. JPL

The Jet Propulsion Laboratory (JPL) will support UCI in its proposal "IMPAC²T: Integrated Management of Power Aware Computation and Communication Technologies" to DARPA with the following institutional facilities, as well as on-going programs:

Center for Integrated Space Microsystems. The Center for Integrated Microelectronics Systems (CISM) is a new program chartered with developing innovative computing technologies that will lead to "thinking spacecraft" of the future. CISM will support the integration and test of the proposed technologies into its microelectronics testbed, where it will be integrated with other related avionics technologies.

Flight System Testbed (FST). Whereas CISM provides an avionics laboratory for integration and test, the JPL Flight System Testbed is an important institutional facility that allows new technologies to be integrated and demonstrated at the spacecraft system level. Many of the component spacecraft subsystems are simulated either in software or hardware.

X2000 Spacecraft A series of spacecraft engineering models (testbeds) are currently under development at JPL, funded by NASA Office of Space Science. Three spacecraft are currently under development, referred to as X2000, X2003, and X2006, with each spacecraft being more technology driven than the previous one. An important point is that the above described engineering model spacecraft are intended to be fully functional spacecraft that can quickly be made into real flight systems.

New Millennium Program flight validation opportunities. The New Millennium Program is NASA's primary flight validation program, which currently consists of three Earth Orbiting and three Deep-Space missions. The sole purpose of the NMP is to enable future missions in the 21 century to use proven, and yet highly advanced, enabling technologies.

3. USC

USC's PDPC (Parallel and Distributed Processing Center) currently uses the following equipment:

- One Sun SPARCstation 5 S5 (110MHz), 32MB, 3.5Gb
- Two IBM RS6000/43P PowerPC 604(133MHz), 128MB, 4GB
- Four IBM RS6000 7043/140 PowerPC604e(200MHz), 64MB, 4.5GB
- One IBM 8260 Switching Hub with 4 155 Mbps ATM ports
### J. Budget Summary

#### J.1 Summary Costs per Year

<table>
<thead>
<tr>
<th></th>
<th>Year 1</th>
<th>Year 2</th>
<th>Year 3 (Option)</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Personnel</td>
<td>$340,731</td>
<td>$355,312</td>
<td>$368,126</td>
<td>$1,064,169</td>
</tr>
<tr>
<td>Travel (Domestic)</td>
<td>$33,500</td>
<td>$32,350</td>
<td>$32,400</td>
<td>$98,250</td>
</tr>
<tr>
<td>Supplies</td>
<td>$20,100</td>
<td>$23,464</td>
<td>$13,130</td>
<td>$56,694</td>
</tr>
<tr>
<td>Other Costs</td>
<td>$142,815</td>
<td>$144,522</td>
<td>$144,543</td>
<td>$431,881</td>
</tr>
<tr>
<td>Equipment</td>
<td>$36,044</td>
<td>$0</td>
<td>$16,000</td>
<td>$52,044</td>
</tr>
<tr>
<td><strong>Total Direct</strong></td>
<td>$573,190</td>
<td>$555,648</td>
<td>$574,199</td>
<td>$1,703,037</td>
</tr>
<tr>
<td>Indirect Cost</td>
<td>$227,266</td>
<td>$211,582</td>
<td>$212,742</td>
<td>$651,590</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td>$800,456</td>
<td>$767,231</td>
<td>$786,940</td>
<td>$2,354,626</td>
</tr>
</tbody>
</table>

#### J.2 Costs per Task and Year

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>% effort</strong></td>
<td>45%</td>
<td>15%</td>
<td>15%</td>
<td>25%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Year 1</th>
<th>Year 2</th>
<th>Year 3</th>
<th>Year 4</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Tasks</strong></td>
<td>Personnel</td>
<td>Personnel</td>
<td>Personnel</td>
<td>Personnel</td>
</tr>
<tr>
<td></td>
<td>$153,329</td>
<td>$51,110</td>
<td>$51,110</td>
<td>$85,183</td>
</tr>
<tr>
<td><strong>Travel (Domestic)</strong></td>
<td>$15,075</td>
<td>$5,025</td>
<td>$5,025</td>
<td>$8,088</td>
</tr>
<tr>
<td><strong>Supplies</strong></td>
<td>$9,045</td>
<td>$3,015</td>
<td>$3,015</td>
<td>$3,282</td>
</tr>
<tr>
<td><strong>Other Costs</strong></td>
<td>$64,267</td>
<td>$21,422</td>
<td>$21,422</td>
<td>$36,131</td>
</tr>
<tr>
<td><strong>Equipment</strong></td>
<td>$16,220</td>
<td>$5,407</td>
<td>$5,407</td>
<td>$9,011</td>
</tr>
<tr>
<td><strong>Total Direct</strong></td>
<td>$257,936</td>
<td>$85,979</td>
<td>$85,979</td>
<td>$143,298</td>
</tr>
<tr>
<td><strong>Indirect Cost</strong></td>
<td>$102,269</td>
<td>$34,090</td>
<td>$34,090</td>
<td>$52,896</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td>$360,205</td>
<td>$120,068</td>
<td>$120,068</td>
<td>$200,114</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Year 2</th>
<th>Year 3</th>
<th>Year 4</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Tasks</strong></td>
<td>Personnel</td>
<td>Personnel</td>
<td>Personnel</td>
</tr>
<tr>
<td></td>
<td>$159,890</td>
<td>$53,297</td>
<td>$53,297</td>
</tr>
<tr>
<td><strong>Travel (Domestic)</strong></td>
<td>$14,558</td>
<td>$4,852.50</td>
<td>$4,852.50</td>
</tr>
<tr>
<td><strong>Supplies</strong></td>
<td>$10,559</td>
<td>$3,520</td>
<td>$3,519.60</td>
</tr>
<tr>
<td><strong>Other Costs</strong></td>
<td>$65,035</td>
<td>$21,678</td>
<td>$21,678</td>
</tr>
<tr>
<td><strong>Equipment</strong></td>
<td>$0</td>
<td>$0</td>
<td>$0</td>
</tr>
<tr>
<td><strong>Total Direct</strong></td>
<td>$250,042</td>
<td>$83,347</td>
<td>$83,347</td>
</tr>
<tr>
<td><strong>Indirect Cost</strong></td>
<td>$95,212</td>
<td>$31,737</td>
<td>$31,737</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td>$345,254</td>
<td>$115,085</td>
<td>$115,085</td>
</tr>
<tr>
<td>Tasks</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>------------</td>
<td>----</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>% effort</td>
<td>45%</td>
<td>15%</td>
<td>15%</td>
</tr>
<tr>
<td>Personnel</td>
<td>$165,657</td>
<td>$55,219</td>
<td>$55,219</td>
</tr>
<tr>
<td>Travel (Domestic)</td>
<td>$14,580</td>
<td>$4,860</td>
<td>$4,860</td>
</tr>
<tr>
<td>Supplies</td>
<td>$5,908</td>
<td>$1,969</td>
<td>$1,969.43</td>
</tr>
<tr>
<td>Other Costs</td>
<td>$65,044</td>
<td>$21,681</td>
<td>$21,681</td>
</tr>
<tr>
<td>Equipment</td>
<td>$7,200</td>
<td>$2,400</td>
<td>$2,400</td>
</tr>
<tr>
<td>Total Direct</td>
<td>$258,389</td>
<td>$86,130</td>
<td>$86,130</td>
</tr>
<tr>
<td>Indirect Cost</td>
<td>$95,734</td>
<td>$31,911</td>
<td>$31,911</td>
</tr>
<tr>
<td>TOTAL</td>
<td>$354,123</td>
<td>$118,041</td>
<td>$118,041</td>
</tr>
</tbody>
</table>

### J.3 Costs per Contract Site

#### UCI

<table>
<thead>
<tr>
<th>Year 1</th>
<th>Year 2</th>
<th>Year 3 (Option)</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Personnel</td>
<td>$178,400</td>
<td>$188,660</td>
<td>$193,084</td>
</tr>
<tr>
<td>Travel (Domestic)</td>
<td>$28,000</td>
<td>$26,600</td>
<td>$26,600</td>
</tr>
<tr>
<td>Supplies</td>
<td>$8,000</td>
<td>$10,000</td>
<td>$6,000</td>
</tr>
<tr>
<td>Other Costs</td>
<td>$90,290</td>
<td>$94,580</td>
<td>$96,559</td>
</tr>
<tr>
<td>Equipment</td>
<td>$30,000</td>
<td>$94,580</td>
<td>$96,559</td>
</tr>
<tr>
<td>Total Direct</td>
<td>$334,690</td>
<td>$319,840</td>
<td>$338,243</td>
</tr>
<tr>
<td>Indirect Cost</td>
<td>$140,566</td>
<td>$121,091</td>
<td>$120,297</td>
</tr>
<tr>
<td>TOTAL</td>
<td>$475,256</td>
<td>$440,931</td>
<td>$458,540</td>
</tr>
</tbody>
</table>

#### JPL

<table>
<thead>
<tr>
<th>Year 1</th>
<th>Year 2</th>
<th>Year 3 (Option)</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Personnel</td>
<td>$119,300</td>
<td>$121,900</td>
<td>$128,500</td>
</tr>
<tr>
<td>Travel (Domestic)</td>
<td>$4,000</td>
<td>$4,000</td>
<td>$4,000</td>
</tr>
<tr>
<td>Supplies/Procurement</td>
<td>$10,600</td>
<td>$10,600</td>
<td>$6,600</td>
</tr>
<tr>
<td>Eng. Services</td>
<td>$10,000</td>
<td>$6,000</td>
<td>$2,000</td>
</tr>
<tr>
<td>Multiple Program Support</td>
<td>$25,100</td>
<td>$25,600</td>
<td>$26,900</td>
</tr>
<tr>
<td>Total Direct</td>
<td>$169,000</td>
<td>$168,100</td>
<td>$168,000</td>
</tr>
<tr>
<td>Indirect Cost</td>
<td>$56,200</td>
<td>$58,200</td>
<td>$60,400</td>
</tr>
<tr>
<td>TOTAL</td>
<td>$225,200</td>
<td>$226,300</td>
<td>$228,400</td>
</tr>
</tbody>
</table>

#### USC

<table>
<thead>
<tr>
<th>Year 1</th>
<th>Year 2</th>
<th>Year 3 (Option)</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Personnel</td>
<td>$43,031</td>
<td>$44,752</td>
<td>$46,542</td>
</tr>
<tr>
<td>Travel (Domestic)</td>
<td>$1,500</td>
<td>$1,750</td>
<td>$1,800</td>
</tr>
<tr>
<td>Supplies</td>
<td>$1,500</td>
<td>$2,864</td>
<td>$530</td>
</tr>
<tr>
<td>Other Costs</td>
<td>$17,425</td>
<td>$18,342</td>
<td>$19,084</td>
</tr>
<tr>
<td>Equipment</td>
<td>$6,044</td>
<td>$0</td>
<td>$0</td>
</tr>
</tbody>
</table>
J.4 Justification of Budget Items

Personnel:

Funding for personnel participating on the project is requested as follows.

   Principal Investigator and Co-Investigators: Will be responsible for the research work for development of IMPACCT System

   Research Assistants (Graduate students) will assist with program development. 49% academic year and 100% summer months

Personnel costs are estimated at published UCI academic and staff salary scales. Anticipated cost of living increases for personnel are calculated at UCI published escalation rates. Where appropriate merit increases are estimated and included for subsequent years.

Employee Benefits:

   Principal Investigator and Co-Investigator @ 9.2%
   Graduate students @ 1.3% and 3.0%

Employee benefits are estimated using published composite rates agreed upon by UCOP and DHHS.

UC policy requires tuition/fee remission for graduate students participating on the project. These costs are based on published University rates.

Supplies and Materials:

Supplies include expendable supplies for computer modeling and simulations, ie: Computer supplies, printer supplies, and other expendable supplies. These costs are based on historical data and escalated by inflationary factor.

Other Direct Costs:

Publication and reproduction costs for copying research materials, disseminating the results of the research, publishing technical papers, etc.

Telephone Costs:

Telephones calls made on behalf of this research project. Consulting with individuals from areas all over the world and the United States about the research procedures and techniques performed.

These costs are based on historical data and published recharge rates. In accordance with guidance issued by UCI Sponsored Projects Administration, when appropriate costs are escalated by applying an inflationary factor.
Facilities and Administrative Costs:

Facilities and administrative costs are calculated at 50.4%. This rate is in agreement with DARPA.

Travel (Only Domestic Travel is Requested)

Trip of 2 meeting days and 3 nights:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Airfare</td>
<td>420</td>
</tr>
<tr>
<td>Surface Transportation</td>
<td>120</td>
</tr>
<tr>
<td>Hotel ($125/night)</td>
<td>375</td>
</tr>
<tr>
<td>Per diem ($45/day)</td>
<td>135</td>
</tr>
<tr>
<td>Registration</td>
<td>250</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>1,300</strong></td>
</tr>
</tbody>
</table>

This is an estimate for an average trip.

<table>
<thead>
<tr>
<th>Meeting</th>
<th>No./year</th>
<th>No. Persons</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>PI meeting</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Annual Review</td>
<td>1</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Conference</td>
<td>2</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>Research</td>
<td>2</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>22</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

22 trips @ $1,300/trip at approximately $28,000 per year

1.1 Equipment needs

Equipment –

**Year 1:**

- 6 Unix Workstations @ $5,000 ea. $30,000
- Software licences $12,000

**Year 2:**

- Software licences $12,000

**Year 3:**

- 4 NT Workstations @ $4,000 ea. $16,000
- Software licences $12,000

Software licences include synthesis, simulation and CAD tools tools (e.g. Synopsys, Synplicity, Mentor Graphics, ModelSim, etc.) in addition to software development tools (Khoros, Visual C/C++ etc.).
BIBLIOGRAPHY


[57] K. Harper, Motorola M-CORE Technology Center, Austin TX, See www.mot.com/SPS/MCORE/


