Microsystem and SoC Design with UMIPS

Michael S. McCorquodale, Eric D. Marsman, Robert M. Senger, Fadi H. Gebara,
Matthew R. Guthaus, Daniel J. Burke, and Richard B. Brown

Department of Electrical Engineering and Computer Science
Center for Wireless Integrated Microsystems
University of Michigan
Ann Arbor, Michigan 48109-2122
{mmccorq, emarsman, rsenger, fadi, mguthaus, djburke, brown}@eecs.umich.edu

Abstract

Several IP components have been developed and are presented here including analog, digital, mixed-signal, and MEMS circuitry. These components have been incorporated into a recently founded research IP repository, UMIPS, for design reuse. A complete microsystem has since been developed from this repository using an IP-based development framework and platform design approach. Here a discussion is presented regarding the development of these IP components, the research IP repository, use of the repository in the development of microsystems and SoC technology, and the importance of IP and design reuse for accelerating microsystems and SoC research.

1. Introduction

It has recently been estimated that semiconductor design productivity is increasing at a rate of 28% annually while semiconductor capacity is increasing at a rate of 58% annually [1]. This trend quantifies the now well-known productivity-capacity gap in semiconductor development. Many electronic design automation (EDA) tools have been developed to reduce this gap, but one particular solution that has received significant attention is design reuse through intellectual property (IP) [2]. For example, it has been reported that application specific integrated circuit (ASIC) design productivity increased by a factor of 2.5 from 1996 to 2000 while system-on-chip (SoC) development productivity increased 7 times within the same time frame. Most of the gains realized in SoC development have been attributed to IP design and reuse, clearly indicating the importance of IP to accelerating design time and increasing design productivity [3].

Although the benefits associated with IP design and reuse are clear, the challenges of its use are many. Reports in the field point to the importance of IP repositories and a standardized IP framework for IP component development [4]. Indeed a lack of these repositories and standards, both internal and external to organizations, is a significant bottleneck toward IP use and design productivity acceleration [5]. Industry has clearly identified this problem and is taking strides toward its solution where several commercial IP repositories now exist such as the *Virtual Component Exchange* (*VCX*) and *Design & Reuse*. However, research institutions have not yet pursued similar initiatives.

One of the first research IP repositories, the *University of Michigan Intellectual Property Source* (*UMIPS*), has been launched with a focus on microsystem and SoC design, which comprises a large portion of the semiconductor research pursued at Michigan. Microsystems technology has been defined as an intelligent miniaturized system comprising sensing, processing, and/or actuating functions where two or more of the following technologies are combined onto a single or multichip hybrid: electrical, magnetic, mechanical, optical, chemical, or biological [6]. Thus, although the vast majority of commercially available IP components are digital, the mission of *UMIPS* is to support digital, analog, mixed-signal, MEMS, CAD, and design methodology IP, including standardization of the deliverables for IP of these types. The current demand for IP of this nature far outweighs its availability [7] and thus *UMIPS* can accelerate microsystems and SoC research while preparing students and researchers with experience developing IP components and using these components in an IP design framework that will almost certainly become ubiquitous in future microsystems and SoC development.

2. IP Formats and Nomenclature

A cursory review of IP formats and nomenclature is presented here to familiarize the reader with the concepts that are presented subsequently within this paper.

IP can be divided in to three application formats: functional, infrastructure, and platform [8]. The corresponding IP can then be delivered as either hard, firm, soft, or design methodology IP. In the sections that follow, a review of the basic definitions of these formats is presented. In these descriptions, no effort has been made to promote one form of IP over another as each has specific benefits and shortcomings when utilized in a microsystem and SoC design framework. Illustrations of these IP formats are presented in Figure 1 and Figure 2.

2.1. Application Formats

2.1.1. Functional IP

IP components that provide, support, or implement specific operations are functional IP. Microprocessors, memory, logic, analog amplifiers, and MEMS are all examples of functional IP. This application format currently comprises the bulk of commercially available IP.

2.1.2. Infrastructure IP

Infrastructure IP supports development, characterization, configuration, debugging, and test of electronic systems. Some examples of infrastructure IP include embedded test modules, mask level interconnect for gate array definition, and design methodologies.

2.1.3. Platform IP

Platform IP is an emerging application format in which the IP is developed and delivered as a larger functional component that can be customized through the addition of newly developed peripherals or other IP components. Commercial entities such as *ARM Ltd.* deliver

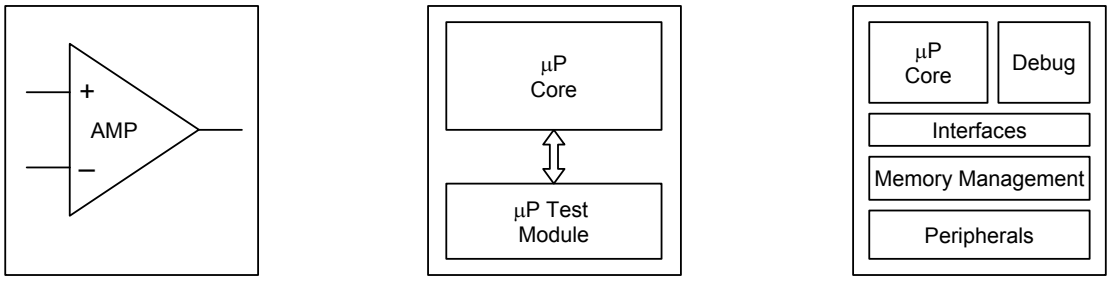

Figure 1. Application formats for intellectual property.

several microprocessor product lines as a platform where the CPU, operating system, and peripheral components are pre-defined. In essence, platform-based design is a development foundation, or baseline, for designing an SoC or microsystem for a particular application [9]. Plaform-based design can be of several topologies, where the *ARM* platform described previously is processor-centric. Other approaches include communications-centric, program-centric, and application-centric.

2.2. Delivery Formats

2.2.1. Hard IP

Hard IP is the physical design of a particular IP component and thus is process specific. It is delivered as GDS or some other physical design format. Hard IP is the least customizable, but the most reliable IP format because verification of hard IP includes specific timing and electrical information from fabricated designs. For this reason, hard IP is also the best option for closing the productivity-capacity gap as it can be used without additional internal component verification, although inter-module verification is still required.

2.2.2. Firm IP

Firm IP is a netlist or register transfer language (RTL) description of an IP component. Physical design for digital circuits can be compiled from firm IP with a place and route tool. As an optional prior step, the firm IP may be synthesized for any process technology without modification to the functionality of the original component. The advantage of firm IP to the user is that it is not process specific.

2.2.3. Soft IP

Soft IP is the most easily customized IP format as it is described by a hardware description language (HDL), such as *Verilog*. Soft IP can be synthesized to firm IP with a tool such as *Synopsys Design Compiler*. The firm IP can then be compiled to hard IP. However, considering that verification is at least 50% of the design cycle for IP components, soft IP might be an excellent option for customization, but it is not a good option for closing the productivity-capacity gap [10]. When many custom modifications are pursued, the component must be verified again.

2.2.4. Design Methodology IP

Design methodology IP is a format that has been identified as critical to microsystem and SoC development at Michigan. It is a type of infrastructure IP that is utilized within the design framework. Examples include design flow and framework documentation and scripts that modify data files for use across tool suites.

3. Overview of Previous Research

Four independent, but related, research projects have been under development and have recently been converted to IP components that are compliant with the *UMIPS* standards, described later in this paper. A new research project involving the convergence of these projects into a single microsystem motivated much of *UMIPS* development. Generation-0 of this microsystem was presented in [11] and the details and challenges of its design, simulation, and verification across digital, analog, and mechanical domains was described in [12]. In this work, the design of the next generation microsystem is presented along with a platform-centric IP design methodology which was employed in its development. This strategy is presented in addition to the techniques described in [12] and as a high-level design approach and methodology. In the sections that follow, a brief description of each individual project is presented.

3.1. Analog Front End

An analog front end (AFE), shown in Figure 3, has been developed in order to interface to a variety of off-chip sensors. It consists of three major components: buff-

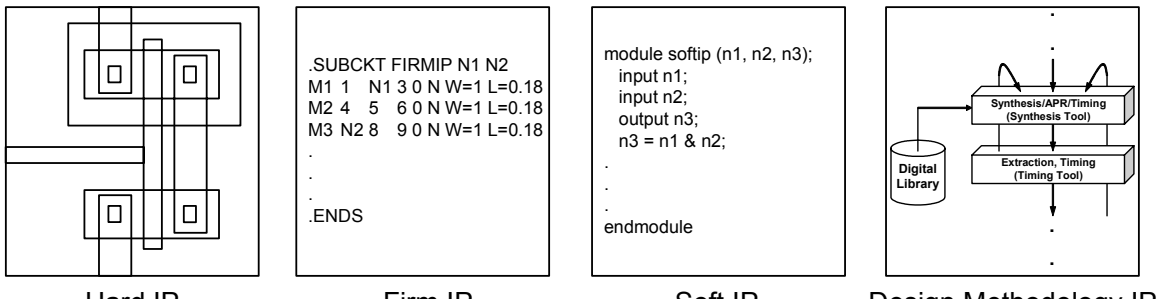

Figure 2. Delivery formats for intellectual property.

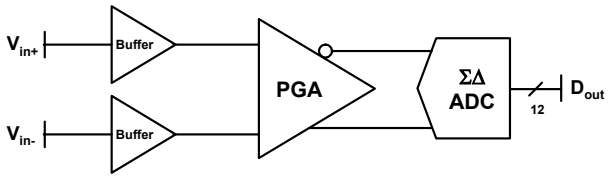

Figure 3. AFE architecture [11].

ers, a programmable gain amplifier (PGA), and a 12-bit 2^{nd} order ΣΔ modulator. The AFE is low power and low voltage, operating at supply voltages as low as 900mV through the use of weak inversion circuit design. The ADC supports a 100Hz Nyquist bandwidth and is ideal for low-bandwidth, high-precision sensor applications.

3.2. CMOS-MEMS Clock Reference

A monolithic MEMS-based clock reference has been developed in CMOS to eliminate the need for an off-chip crystal and an on-chip PLL for clock generation. This component synthesizes clock signals completely on-chip, and with low power consumption and low jitter compared to alternate approaches. The reference signal is synthesized by a low-jitter RF MEMS oscillator from which multiple clock frequencies are generated by frequency division through a series of flip-flops as shown in Figure 4. The processor taps the last output and divides this signal to several programmable frequencies that clock the core. This top-down approach to frequency synthesis possesses the systemic advantage that each divided frequency is more stable than the reference signal. In direct contrast, PLL-based systems synthesize signals from the bottom-up and thus the jitter is accumulated [13].

3.3. Low Power Microcontroller Core

A complete custom instruction set architecture (ISA) has been developed and implemented in a low-power processor core. A 16-bit load/store architecture with dual-operand register-to-register instructions was chosen to satisfy the power and performance requirements of the microsystem. The 16-bit datapath was selected to reduce the complexity and power consumption of the core while providing adequate precision in calculations. The pipeline consists of three stages: fetch, decode, and execute.

A 24-bit address space for unified data and instruction memory satisfies the potentially large storage requirements of remote sensor systems. The 16MB of supported memory is byte addressable and provides sufficient address space for program, data, and memory-mapped peripheral components. A loop cache interface is also included in order to reduce power consumed by commonly executed instructions.

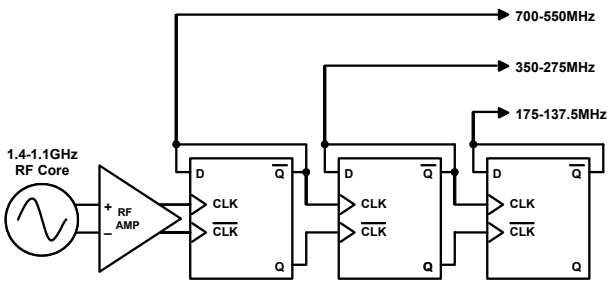

Figure 4. RF MEMS-based clock reference. The circuit utilizes a suspended inductor and tunable micromechanical varactor [11]. Frequencies are shown before (1.1GHz core) and after (1.4GHz core) release.

The core contains sixteen general-purpose data registers and four address registers, that are evenly split into two access windows. Three additional non-windowed address registers (a stack pointer, frame pointer, and link register) are available for use by the compiler for subroutine and stack support. The hardware supports maskable interrupts which are prioritized up to 64 different levels.

3.4. Microcontroller Peripherals

Several peripherals have been developed as communication interfaces to the microcontroller core. These peripherals include a Universal Synchronous Asynchronous Receive/Transmit (USART) unit, a Serial Peripheral Interface (SPI), and a multifunction programmable timer. A wrapper for the USART has been developed as a test interface for the core.

4. *UMIPS* as a Research IP Repository

4.1. Mission and Goals

UMIPS is a collaborative effort between Electrical Engineering and Computer Science students, faculty, and researchers at the University of Michigan to leverage each others' integrated circuit development in the pursuit of cutting-edge microsystems and SoC research. The motivation for *UMIPS* launch is design reuse and its benefits to research. The IP described previously marked the founding contributions to the *UMIPS* repository.

UMIPS was founded as a result of clearly identifying the lack of IP development and use within the research community. The value to research projects includes the elimination of repetitive basic design activities that are applicable to all projects, consequently enabling meaningful research productivity to be substantially increased. Second, an IP repository within the research community lends itself nicely to participation from a breadth of disciplines and contributing sources in semiconductor research including activities in analog, mixed-signal, digital, and MEMS development as well as design methodology and CAD. Third, it is now imperative that students gain experience with an IP-based design methodology as they will almost certainly encounter this type of development in industry. Similarly, design activities are almost never initiated from a blank slate, so students who are able to utilize IP components in their research gain valuable experience for any future development activity.

From its beginning, the most significant challenge faced has been standardization and management of the repository. In the sections that follow, the framework for the repository will be described as well as approaches to management and quality control.

4.2. Supported IP

UMIPS currently supports both functional and structural IP in hard, soft, and design methodology formats. Platform IP is not currently supported as the repository is in its infancy. However, current research activities will certainly lead to the development of several general purpose microsystem platforms, similar to the one presented subsequently within this paper. Firm IP is also not supported currently.

UMIPS IP is organized by function and by process technology. The majority of researchers at Michigan use the *MOSIS* multi-project service for IC fabrication, and thus *UMIPS* supports all process technologies offered by this service. In the near future, *UMIPS* will support IP specific to our own internal solid-state electronics laboratory as well as MEMS foundries.

4.3. *UMIPS* Interface and Access

UMIPS is an internal repository, but is currently accessible by all researchers in the academic community and is for noncommercial use exclusively. *UMIPS* has received and posted submissions from both the University of Michigan and other research institutions. The repository can be viewed via the main internet portal where nonproprietary descriptions of the IP components are listed and organized by function and process technology. Researchers wishing to utilize IP in their designs submit a request for IP to the management team and authentication information is provided to that researcher for the specific IP of interest.

4.4. *UMIPS* Standards and Quality Control

The most important aspect of repository management is standardization to maintain quality. *UMIPS* has been launched with a relatively simple standards framework which is described here. IP is added to the repository through an audit process where the founding management team reviews weekly submissions to the repository to ensure that the baseline deliverables are included for each IP component. This is, of course, quite time consuming, thus the process will be automated as the standards are formalized and the repository framework matures. Some techniques, such as those presented in [14][15], may be employed in the future.

4.4.1. Deliverables for All IP Types

The baseline deliverables for all *UMIPS* IP include the following items. These deliverables have been determined after review of many relevant concepts presented in [16][17][18].

1. Nonproprietary Description
A text file containing the description of the macro that does not divulge the specific details of its implementation. This description is available publicly.

2. Proprietary Description
A text file containing the full and detailed description of the macro that includes specific details of the implementation. This file is not available publicly.

3. Specification
A detailed specification of the part from either simulation, test, or both, including timing diagrams and a complete specification of the interface. This file is the main source of information for the IP user.

4. README
A text file describing every file in the repository for the given component.

4.4.2. Soft IP Deliverables

In addition to the baseline deliverables, the following are required for *UMIPS* soft IP submissions:

1. Synthesizable HDL
Synthesizable code describing the macro in HDL. The HDL code must be commented and maintainable.

2. Test Fixture and Vectors
An HDL test fixture including stimulus vectors for user test of the macro.

3. Synthesis Scripts
Synthesis scripts that define clocks, pin timing, load constraints, false paths, synthesis commands, and any other special requirements. *PrimeTime* scripts and constraints are also required.

4. Process Specific Details
These details include information on hard instantiated cells, “dont_use cells,” memory, or other macro components used in the design, and suggested wire-load models. If applicable, automatic place and route scripts are included.

4.4.3. Hard IP Deliverables

In addition to the baseline deliverables, the following are required for *UMIPS* hard IP submissions:

1. Netlist
Hierarchical netlist for the macro in SPICE format.

2. Physical Design
GDS stream for the physical design of the macro.

3. Map Table
The map table for the GDS stream in and stream out specific to the process technology.

4. Routing File
Library Exchange Format (LEF) file for automatic place and route of the macro.

5. Verification Reports
LVS and DRC reports. ERC, metal fill, and others are optional. Known errors must be clearly explained in the README file.

6. Verification Decks
The exact decks used for verification. Edits to the baseline deck for the process technology must be clearly explained in the README file.

7. Model Files
All process-specific device model files.

8. HDL Model
A *Verilog* or *Verilog-A* model for the component.

9. Test Vectors
A test stimulus file for use with the HDL model. Test vectors and stimulus of the HDL model with these vectors must be clearly described in the specification.

4.5. *UMIPS* Qualification Metrics

IP components are incorporated into *UMIPS* as either “pre-silicon” or “post-silicon” verified. Pre-silicon components must include simulation performance data and be detailed in the specification. Post-silicon verified components must include the same data from test. The specification must also include a test plan and set-up.

IP components are also qualified as either “instantiated” or “uninstantiated” components. This designation indicates whether the component has been developed into a larger design. Instantiated components have been through at least one design cycle and thus the deliverables have been utilized for development, which indicates that they are adequate for design.

Design methodologies must be utilized on at least one design cycle which has been fabricated and tested with favorable results. This ensures that no aspects of the design flow or methodology have been overlooked, such as antenna effects or timing approximation errors.

Users who take advantage of IP will have the opportunity to provide feedback and comments about the IP to other prospective users of *UMIPS*.

4.6. Challenges

A number of challenges are associated with *UMIPS* management, participation, and technical support. Extensive management is required in order for the repository to

operate successfully. Qualification and auditing are the most rigorous and time-consuming as well as the most important. Automation of portions of these activities can be accomplished in the future with software development. Participation is also difficult to stimulate. Many would-be contributors are deterred by the breadth of requirements for depositing IP into *UMIPS*. An incentive program is under development where the most highly utilized IP and the most novel IP submissions are granted awards. The goal is to stimulate participation from students in particular. However, the founders also anticipate that the repository will grow organically where participants that utilize *UMIPS* IP appreciate its value and are consequently compelled to contribute in order to ensure its sustainability.

The most difficult technical challenge associated with *UMIPS* is managing IP deliverables across various tool suites. Most analog and RF designers at Michigan pursue design work within the *Cadence Design Framework II* and therefore utilize the associated application specific verification decks. However, digital work at Michigan is typically pursued with the *Synopsys* and *Mentor Graphics* tool suites. For several process technologies, incompatibility between the verification decks for these tools has been discovered.

Extraction decks written for a tool suite that is typically used for a digital design flow will often not support MiM capacitor structures, inductors, and other specialized analog components. Likewise, verification decks designed for analog and MEMS components may not be appropriate for handling the much larger digital designs. This creates verification challenges when IP is withdrawn from the repository and incorporated into a user's design flow that does not support these devices. This is being addressed by inclusion of all verification decks for all IP components, as described previously. At a minimum, the user that withdraws the IP can use the included deck to modify their own in order to verify the IP in their flow. This is still a significant burden. Several additional technical challenges similar to this circumstance exist and will be addressed as *UMIPS* matures.

5. A Microsystem Design with *UMIPS*

5.1. System Overview

Using the IP components described above, a microsystem was developed with a focus on sensor control applications, although the design retains enough flexibility for a variety of other general-purpose uses. As such, the system was developed with a platform approach. The system has been developed in the 0.18µm mixed-mode process available from *Taiwan Semiconductor Manufacturing Company* (*TSMC*). This is the process in which the hard IP components, the AFE and the clock reference, were developed.

The microsystem, shown in Figure 5, is comprised of five major subsystems: a processor core, peripherals, memory, an analog front end, and an on-chip clock reference. As described previously, this particular microsystem is a second generation of the microsystem presented in [11] and includes additional functionality.

5.2. IP-Based Platform Design Methodology

The microsystem was developed using a processor-centric platform-based design approach. Specifically, the processor core and associated peripherals were developed and synthesized together from soft IP. Synthesis was achieved with the standard cell library for the *TSMC* 0.18µm process available from *Artisan Components Inc*. These components comprise the IP-platform, or baseline processor. The 64KB on-chip memory banks and loop cache were generated from *Artisan*'s memory compilers. The AFE and clock reference components were then instantiated and routed as hard IP components, as shown in Figure 6.

Once each IP component had been created and verified individually, the system was assembled at the top level. A *Verilog* description of the top-level connections, top-level buffering, an abstract of the IP in the standard LEF, and the standard process and routing layer information were used by *Silicon Ensemble* to automatically place and route the top-level design. RC extraction was performed for final simulation verification purposes.

The ghost views of the hard IP components, standard cells, and memory components were replaced with GDS using the *Cadence Design Framework II*. *Mentor Graphics Calibre* was then used for DRC, ERC, and LVS. The details of the complete design methodology and verification for this mircrosystem have been presented in [12].

5.3. Challenges

While IP reuse has several advantages over an original design effort, it is not without difficulties. Developing a design methodology that allows the EDA tools to understand the required attributes of each IP component

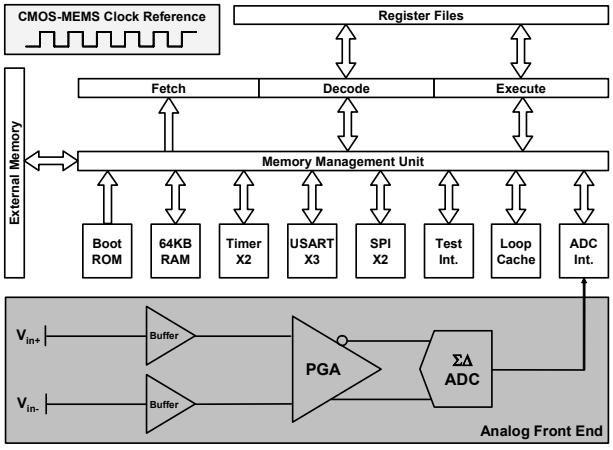

Figure 5. Microsystem functional architecture.

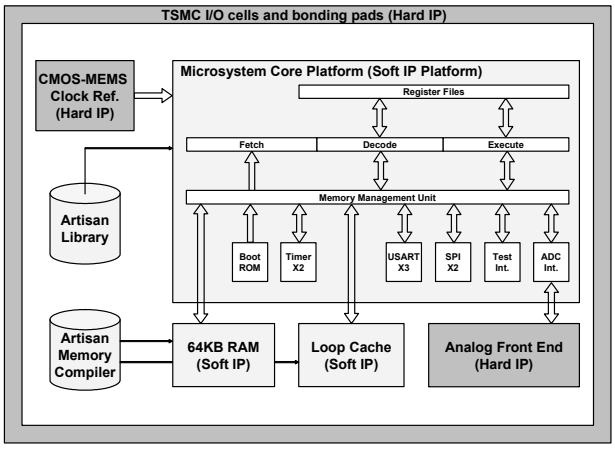

Figure 6. Processor-centric platform IP structure.

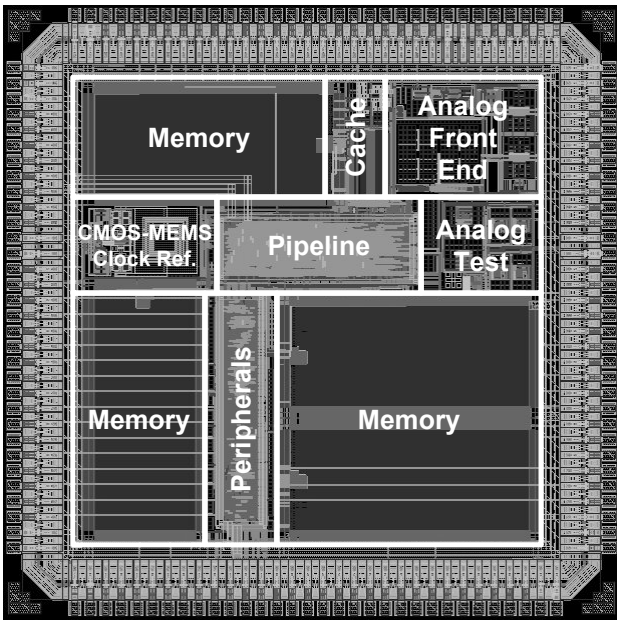

Figure 7. Physical design of the Gen-1 microsystem.

is the first challenge. This issue is related to the previously discussed technical difficulties with maintaining a repository across several tool suites. An intimate understanding of all of the data files and environment variables for the tools used is required in order to achieve the desired design performance and verification. However, once these required modifications are developed and documented as they are on *UMIPS*, the data files required to describe any type of IP component, examples of these components, and a design flow that uses them with a complete suite of tools are easily accessible and understandable. This makes reuse of the design flow another advantage of *UMIPS*.

5.4. Microsystem Physical Design

The physical implementation of the microsystem is shown in Figure 7. The die area is 12.8mm^2 and the transistor count is over 3.5 million. The microsystem includes two components of hard IP and one platform of soft IP, all from the *UMIPS* repository. Third party IP includes the cell library, memory, loop cache, input/output drivers, and bonding pads. The developed microsystem includes digital, analog, mixed-signal, and MEMS components. The project is currently in fabrication at *Taiwan Semiconductor Manufacturing Company* through the *MOSIS* service.

6. Conclusion and Future Research

IP design and reuse will become indispensable aspects of microsystem and SoC research and development in the near future. Moreover, support for analog, mixed-signal, and MEMS IP components will become commonplace as development and standardization of IP for these components matures. Through the development and use of IP, design cycles can be substantially reduced, and accelerated design productivity will enable more meaningful research contributions to be achieved. Clearly the advantages of research IP design and reuse are many.

One of the first research IP repositories, *UMIPS*, has been launched at the University of Michigan. Here the infrastructure of the repository has been described along with the details of its implementation. Using the founding IP components, a complete microsystem has been developed from *UMIPS*. The processor-centric platform design approach used in the development of this microsystem was presented, and the research IP components of several students were put to use.

Several challenges have been addressed in the formal launch of *UMIPS* and more will certainly be encountered. Nevertheless, the founders aspire to foster the development of the repository and expand its use. Future work will include the incorporation of IP protection mechanisms, such as watermarking, and support for bus interface standards that will facilitate interconnection of IP components. A formal licensing framework for *UMIPS* components will also be developed in the near future. Lastly, *UMIPS* will also likely support research-based CAD tools and cell libraries for synthesis at some point in the future.

7. Acknowledgement

Fabrication of this work at *Taiwan Semiconductor Manufacturing Company* was supported by the *MOSIS* Educational Program. Digital cell libraries and SRAMs were supplied by *Artisan Components*, *Inc*. Input, output, and bonding pads were provided by *Taiwan Semiconductor Manufacturing Company*. This work was supported primarily by the *Engineering Research Centers Program* of the *National Science Foundation* under award number EEC-9986866.

8. References

[1] H. Qi, *et al.*, "IP-Based SOC Design Methodology," *International Conference on Chip Design Automation*, 2000.
[2] R. Abrishami, "Vendors must support IP reuse in SoC," *EETimes*, Dec. 19, 2002.
[3] T. Maniwa, "EDAC panel marks path to productivity," *EEdesign*, April 11, 2001.
[4] P. Clarke, "Repositories called linchpin of widespread design reuse," *EETimes*, March 21, 2000.
[5] P. Rose, "Intellectual Property Reuse – A New Business Model," *D&R Industry Article*, 2002.
[6] European Union 4th Framework, *ESPIRIT Workprogramme*, Sept. 1996.
[7] S. Ohr, "Interest in analog IP outpaces execution," *EEdesign*, Feb. 11, 2003.
[8] P. Santos, "Synthesizable analog IP key to embedded SoC platforms," *EETimes*, April 18, 2003.
[9] R. Wilson, "Design reuse expands across industry," *EETimes*, March 27, 2003.
[10] L. Rosenberg, "What 'IP' really means," *EDNaccess*, Aug. 5, 2000.
[11] R. Senger, *et al*., "A 16-Bit Mixed-Signal Microsystem with Integrated CMOS-MEMS Clock Reference," *Design Automation Conference*, 2003.
[12] M. S. McCorquodale, *et al*., "A Top-Down Microsystems Design Methodology and Associated Challenges," *Design Automation and Test Europe*, 2003.
[13] M. S. McCorquodale, *et al.*, "Top-Down and Bottom-Up Approaches to Stable Clock Synthesis," *International Conference on Electronic Circuits and Systems*, 2003.
[14] R. Seepold, "A Qualification Platform for Design Reuse," *International Symposium on Quality of Electronic Design*, 2002.
[15] J. Haase, *et al*., "Automated Qualification Flow for Soft IP," *Conference on Application-Oriented SoC Design*, 2001.
[16] N. Bray, "Designing for the IP Supermarket," *VHDL International Users Forum Workshop*, October, 1999.
[17] H. Qi, *et al*., "IP reusable design methodology," *4th International Conference on ASIC*, 2001.
[18] N. M. Madrid, "Analog/Mixed-Signal IP Modeling for Design Reuse," *Design Automation and Test Europe*, 2001.