Case Study on Achieving First-Silicon Success for CDMA Handset ASIC Through Reconfigurable System Prototyping

Luis Aldaz
Philips Semiconductors

Greg Lara
Aptix Corporation

2000 Wireless and Broadband Design Conference
ABSTRACT

Engineers drastically reduced the development cycle of a CDMA handset ASIC and achieved first-silicon success through reconfigurable system prototyping. Expanding beyond traditional ASIC emulation, a unique programmable hardware platform enabled rapid assembly and debugging of a complete System-on-Chip (SoC) prototype. A block-based approach to design implementation resulted in smooth build-up and rapid bring-up of the initial prototype. Team-based development, debugging and software integration was enabled by constructing multiple prototype systems. The design flow, including algorithm development, hardware mapping and real-time evaluation, is reviewed. Hardware implementation considerations and tradeoffs are discussed. Though a wireless communications example is presented, the methodology discussed is applicable to a wide variety of applications.

BIOGRAPHIES

Luis Aldaz  luis.aldaz@philips.com

Luis Aldaz is a program manager at Philips Semiconductors. His professional experience and research interests are in areas of digital communications, digital signal processing, ASIC design methodology and wireless system design. He has led the Philips wireless systems group concentrating on Spread Spectrum CDMA systems and he is currently involved in the development of the Third Generation products.

Luis received a B.Sc. degree in Mathematics from the University of Lausanne in Switzerland and holds a M.Sc. degree in Communication Systems from the Swiss Federal Polytechnic Institute of Technology and the Eurecom Institute of Technology in France. He joined the Wireless Products Division at VLSI Technology in January 1996, which merged with Philips Semiconductors in June 1999

Greg Lara  greg@aptix.com

Greg Lara is Product Marketing Manager at Aptix Corporation. Greg joined Aptix in April 1993. Prior to that he worked at VLSI Technology, Inc., International CMOS Technology, and National Semiconductor Corporation where he held positions in product marketing and international sales management. Greg received a BSE from San Francisco State University.
Real-time performance makes all the difference when it comes to verifying a million-gate ASIC that includes an embedded processor and a DSP unit. By prototyping such a chip for a wireless handset at its rated clock speed, the system team of Business Line–Cellular Americas at Philips Semiconductor (formerly the Communications Products Group, Wireless Products Division at VLSI Technology, Inc.) shortened our design cycle and achieved first-time silicon success—a welcome change from the multiple re-spins we suffered through on a previous project.
The new wireless handset ASIC is the most highly integrated CDMA chip available, incorporating all the necessary baseband circuitry to implement a cellular handset. Philips offers this CDMA technology as an application-specific standard product and as a set of cores designed around the company’s Standard Communications Platform (SCP). This highly integrated platform includes the ARM processor and OAK DSP cores.

Verifying a design of this size and complexity demands thorough testing. Although we used RTL simulation, gate-level static timing analysis, and gate-level simulation with back-annotated interconnect delays, we knew that these methods were unlikely to uncover all of the design’s functional problems. Thus, we also used the Aptix System Explorer™ SoC verification system.
Verification Challenges

- Algorithm evaluation
- Integration of IP
  - ARM CPU
  - Oak DSP
  - Memory interface
  - Other peripherals
- RF/digital interface
- Integration and debugging of DSP firmware and CPU software
- System-level verification requires at-speed testing of complete prototype

We anticipated a number of challenges to verification of the handset design. These included: integration of the ARM CPU and Oak DSP cores with the other IP blocks and verification of the interaction between all of the blocks; integration and debugging of the DSP firmware and CPU software; and integration of the RF module with the completed CDMA+ processor and verification of the RF/digital interface.

We concluded that the only way to achieve all of these goals would be to build a prototype of the complete system. Aptix’s System Explorer™ products would enable us to build a flexible prototype in reconfigurable hardware. These tools would enable us to implement custom logic and legacy IP in FPGAs and integrate the ARM CPU, Oak DSP and RF module on a programmable hardware platform. Moving far beyond the limitations of simulation test benches, we would be able to verify the interaction of the IP, integrate and debug the firmware and software, and validate the design in a real-world environment.

Benefits of SoC Emulation

- Reduce risk
  - Verify SoC function in system environment
  - Verify digital/analog interfaces
  - Verify interaction of H/W with S/W, F/W
  - Increased visibility into internal IC logic
  - Facilitate post-silicon debugging
- Accelerate hardware/software integration
  - Hardware prototype available early in design cycle
- Confirm compliance to standards
- Rapid iteration of algorithms and logic
  - Optimize system performance to surpass standards
  - Adapt prototype to changing standards

Emulation of SoC ASIC designs provides significant benefits. Foremost is the reduction of risk. The competitive consumer electronics marketplace rewards manufacturers who deliver high quality products to market quickly. Philips could not afford the delays that would be imposed by a silicon re-spin; first silicon success was essential.

Simulation tools are valuable for verifying that a design matches the specifications. Emulation of the complete system design is required to validate the specification. With reconfigurable prototyping, engineers can evaluate the SoC function in a real-world environment. By building a prototype, they can verify the interaction of IP blocks, the interfaces between digital and analog subsystems, and the interaction of hardware, firmware, and software. Any problems identified can be addressed while the design is still flexible. The reconfigurable hardware provides the ability to observe the behavior of nodes deep within the circuit, an ability not available with ASIC silicon.

The availability of hardware prototypes early in the design cycle accelerates development and integration of system software. Software and hardware are developed in parallel. Instead of waiting for a pre-product prototype to be assembled, software developers can begin exercising and debugging their code even before the hardware design is complete. Unanticipated problems discovered while the hardware designs is still flexible can be addressed in the most optimal manner.

System Prototyping Tools

- System Explorer reconfigurable system prototyping platform
- Prototyping area accepts:
  - Interchangeable FPGA prototyping modules for custom logic
  - Other system components such as CPU, DSP, memory, etc.
  - Mounted on adapter cards or cabled into prototyping board
- Aptix proprietary field-programmable interconnect component (FPIC™) routes connections between all components
- Hardware configured via Aptix software
  - Remote configuration through Ethernet
  - Stand-alone operation with configuration data in EEPROM
- Automated interface to Agilent (HP) logic analyzers for hardware debugging

A Block-based Approach to Prototyping

The team selected a methodology based on reconfigurable system prototyping tools from Aptix Corporation. The System Explorer product line consists of software that maps the ASIC design to a programmable prototyping platform, along with automated interfaces to hardware debugging tools (and, more recently, simulation platforms).
At the heart of Aptix’s system is a programmable circuit board built around the company’s proprietary Field Programmable Interconnect Component (FPIC®). The boards provide prototyping areas populated with an array of pin sockets in a 100-mil-pitch pattern. The pin sockets accept FPGA prototyping modules and other system components mounted on pin-conversion adapters. The pin sockets are individually routed to an area of the board that holds Aptix’s proprietary 1,024-pin FPIC switch matrix arrays.

The FPIC architecture enables bi-directional nets to be routed between any of its 936 I/O pins under software control. The connections between the pins of the prototyping components are completed within the SRAM-based FPIC devices.
Visibility into virtually any point of the logic netlist implemented in the prototype is a key benefit of reconfigurable system prototyping. The Aptix System Explorer provides an automated method of probing with Agilent (formerly Hewlett Packard) logic analyzers. Diagnostic components—a special version of the FPIC device acting as a programmable probe—are mounted on the back of the prototyping board. Users select nets to be probed from the Aptix GUI. The Aptix software then creates and downloads a configuration file to the Agilent logic analyzer to set up channel assignment, signal labeling, and bus grouping. Finally, the Aptix system routes the desired signal through the FPIC diagnostic component to the Agilent logic analyzer.

The combination of programmable logic and programmable interconnect provide a flexible platform for rapidly constructing complete system and SoC prototypes. The prototype can be built up block-by-block, modified easily to correct bugs or enhance function, and exercised in a real-world environment to provide a thorough validation of the system design.

Building the Handset Prototype

Philips acquired eight Aptix System Explorer MP3A prototyping systems so that we could build multiple copies of our prototype. These replicates enabled us to support our design team as well as our software and firmware developers at multiple sites. This approach was made possible by the ability to purchase replicate prototyping hardware to supplement the initial development system (software and hardware) configuration. Each prototype can be set up to operate in a stand-alone mode, automatically configuring itself using data stored in on-board, non-volatile memory.

We employed a block-based approach to developing and verifying the system design. We carried this approach over to the construction of the SoC ASIC prototype.
We began our prototyping effort early in the design cycle. We developed the first block (a DSP bus interface unit) using Cadence’s SPW environment, then used Synopsys’ FPGA Compiler for synthesis. We targeted the logic to a Lucent ORCA 2C40 FPGA (the largest, fastest FPGAs available at the time of our project) using Lucent’s ORCA Foundry partition-place-and-route (PPR) software. To ensure that the prototype implementation would meet the critical timing requirements of our circuit, we performed a thorough analysis of the post-PPR netlist. We verified the logic in Verilog-XL, ran static timing analysis with ORCA Foundry, then ran a back-annotated simulation in Verilog-XL. These extra steps might not be required for all prototyping applications, but they were key to producing a solid design that yielded full-speed operation in the completed prototype. When simulation was completed, we implemented the block in an FPGA and brought it up on the MP3A. We found that we could implement the DSP interrupt controller block in the same FPGA.

The concept of mapping each design block to the prototype is fairly simple, but some special techniques are required to ensure the most effective mapping. For example, our ASIC design includes a clock generator block that derives multiple clocks from a single clock source. To build the prototype, we added logic to each block to derive the needed clock internally rather than implementing the clock generator block in a single FPGA and routing the various clock signals to other blocks on the board. This approach ensured better timing by avoiding delays associated with part placement and interconnect routing.

This flow involved significant effort and a considerable learning curve, but eventually we learned all the techniques needed to smoothly map the blocks of our design to the prototype hardware with the desired performance.

A key factor in enabling us to prototype our entire million-gate ASIC is the ability of the Aptix system to accommodate off-the-shelf devices in addition to FPGA-based logic. We incorporated the Oak DSP into the prototype by cabling a connection from the OD-Kit development board to the System Explorer board. Development software running on a host computer allowed us to control and observe the OD-Kit board. Functional tests showed that the DSP bus interface unit was not performing as expected—a problem that simulation had not revealed. We had to add wait states to the interface unit so that it could support the full CDMA clock speed.

When the third block (CPU/DSP interface) was ready, we added it to the prototype in FPGA form. We incorporated the ARM CPU into the prototype by cabling a connection from the PID development board to the System Explorer board. The PID board contains an ARM CPU, memory and peripheral circuits. We were able to control and observe the operation of the ARM chip through development software running on the same computer that hosted the DSP development tools.
The next step was to integrate and verify the Viterbi controller, followed by the pulse density modulators and modulator hardware block. To incorporate the RAM for the Viterbi controller into the prototype, we could have mounted the memory in the prototyping area of the System Explorer board with a pin-conversion adapter. Instead, we created a simple daughter card for mounting the memory directly on top of the FPGA module that held the Viterbi block. This approach saved space on the prototyping board and provided a higher-speed connection between the logic and memory. The FPGA modules provided by Aptix have high-density connectors routed to FPGA I/O that make it easy to mount such daughter cards or attach cables to connect to off-board subsystems.

When it came to adding the RF module, we first connected an RF simulator board. As our evaluation progressed, we replaced the simulator with real RF circuitry. We used a variety of Agilent (Hewlett-Packard) lab tools to check out the function of our growing prototype, including logic analyzer, pattern generator, and oscilloscope.

By proving the functionality at each step, we gradually refined the system from the earliest stages of the design cycle. Eventually, our prototyping environment included the functional elements shown in the figure below. The ability to include the processors and other devices such as memories in the prototype eliminated the need to emulate these functions in FPGAs and helped us reach real-time system speeds.
Real-time Testing

SoC Designs Require Real-time Testing

- Go beyond simulation for complete system verification
  - Subjective evaluation of voice, image, video
  - Thorough testing of systems that process massive amounts of data
- Simulation run times increasing
  - Fixed-point simulation requires six to eight weeks
- How much simulation is enough?
  - Found first modulator bug in frame number 1,001

Real-time verification is essential when developing a complex SoC design. The speed of simulation limits the number of vectors that can be reasonably processed and this limitation is becoming more severe as designs increase in size. For example, a thorough fixed-point simulation of our circuit would require up to four weeks of run time. And you can never be sure if you have performed enough simulation. Our experience verifying the modulator block illustrates this risk. Verilog simulation showed that the block functioned properly. The SPW simulation was also successful. We detected a bug in the modulator circuit only after we had implemented the prototype. As we began running frame data through the prototype, 1,000 frames were processed properly. However, frame number 1,001 produced bad results. This shows that you need to take verification further than simulation allows.

Achieving Real-time Operation

- Verified prototype at 20 MHz
- Techniques for maximizing performance
  - Hand modifications to FPGA routing
  - Set timing constraints in PPR tool
  - Design modifications such as additional pipelining
- Must understand system-level impact of design changes
- Debugged hardware with automated probing through Agilent (HP) logic analyzer

Our primary goal in the prototyping effort was to test our CDMA logic at its standard 19.6608-MHz clock speed. Reaching this speed required considerable effort, which could have been reduced if we had taken advantage of the System Explorer’s dedicated address and data buses. As it was, we routed the buses through the System Explorer’s FPIC routing switches. These and some other signals required careful routing through the FPIC devices to minimize propagation delays.

In an effort to meet timing goals we also modified some of our Verilog code, but found that better results could be obtained more quickly by modifying the routing of the logic in the FPGA. Even though learning to deal with the FPGA’s routing was itself time consuming, as we became proficient at it we got faster turnaround times on changes and good timing results.

The real-time payoff was worth the extra effort. By running our design at its full rated speed, we were able to debug and optimize the software and firmware. Our challenge was to see how much better than the standard specification we could get because the results translate into performance that the cell phone user will experience every day.

Hardware Debugging

We took advantage of the Agilent logic analyzer support to observe and debug the system logic. To increase our visibility into the operation of the logic blocks, we routed connections from nodes within the FPGA to unused pins on the device for probing with the HP16500C logic analyzer. We used an HP16522A pattern generator and HP54540C oscilloscope to further exercise and observe the function of the circuit.

Software Integration

A key element of the reconfigurable prototyping methodology is the ability to deliver copies of the prototypes to the software and firmware development teams. We were able to deliver prototypes very quickly, thus giving the developers a significant head start on integrating and testing their code. Because the prototype replicates operate in stand-alone mode, the software developers did not need to know any details of the prototypes’ implementation. They merely turned on the power and the prototypes configured themselves with bit streams stored in onboard FLASH memory. As the hardware design evolved, we gave the developers new PROMs with updated data for configuring the FPGAs and interconnect. This enabled us to instantly update all of the prototypes to the latest design configuration, even those being used in remote locations.
Making a Phone Call

• Exercised forward and reverse links with voice data
  – CDMA Development Tool (CDT) from Philips
  – Base station simulator (HP8924C)
  – Fading simulator
  – Confirmed compliance to standards
  – Modified reconfigurable prototype to enhance performance

Making the Call

The ultimate test was to actually use the prototype to make a call. We used a combination of tools to exercise and analyze both the forward and reverse links. The CDMA Development Tool (CDT) is a software package developed by Philips that contains a variety of tools for the development and debug of CDMA designs. We used CDT to setup the call and send voice data to the DSP. An HP8924C base station simulator received the modulated data transmitted by the RF card through a fading simulator. The base station simulator looped the data back to the prototype which demodulated the data and sent it back through the DSP to the CDT where we could analyze the data. With the combination of the prototype and these tools we further refined the design’s search rate and acquisition time. Simulation offers little help in these kinds of optimizations, but they are crucial to developing a good product.

Benefits Obtained

• Shortened development cycle by seven months
• Achieved first-silicon success
  – Avoiding ASIC re-spin saved two to three months
  – Caught bugs missed by simulation
  – Verified complete system
• Accelerated software development
  – Pulled in schedule at least four months
  – Used prototype replicate copies used for concurrent development and integration of firmware and software
• Early demos increased customer confidence
• Optimized system performance
  – Enhance F/W and S/W through testing on prototype

Lessons Learned

In addition to helping us optimize the performance of our CDMA design, the Aptix prototype helped ensure first-pass ASIC success. Despite the complexity of the design, our first ASIC implementation worked as expected. This accomplishment saved us the eight-to-twelve weeks typically required to debug a faulty ASIC, modify the design, verify the corrections, and re-spin the silicon. Providing our software and firmware developers with replicate prototypes to develop code before the ASIC became available pulled in the schedule at least four months. In fact, the code was so solid that we had the ASIC sample up and running on our development board just two days after receiving the chip.

Overall, the prototype we created with Aptix’s System Explorer tool enabled us to improve our design while shortening our development cycle by seven months. We are now evaluating the latest-generation System Explorer products for future projects like 3G CDMA.

Limitations of SoC Emulation

• Cost
  – Aptix system
  – Other software tools: FPGA PPR, synthesis, simulation
  – Logic analyzer
• Additional up-front effort
  – Learning curve can affect schedule
• Speed
  – User knowledge and effort required to obtain desired performance
• Functional verification only
  – No verification of IC timing, power consumption, etc.
  – Complements simulation tools, does not replace them

The benefits of reconfigurable system prototyping do not come without a cost. First is the expense of the tools. We see, though, that the System Explorer is more cost effective than other approaches we have evaluated. Another consideration is the learning curve associated with the tools. It is best to plan on prototyping at the beginning of the project and budget resources accordingly. Adopting this methodology late in the design cycle in an attempt to catch up on a slipping schedule will put prototyping in a critical path and is sure to cause frustration.

Additional effort is required to map the design to the prototype and achieve the desired performance. Although we took many steps to ensure the prototype would run at the desired speed, we see that designers can avoid much of this timing-related work by creating logic specifically for FPGA implementation in the prototype. After building and verifying the prototype, designers can retarget the netlist to ASIC technology with no modifications to the design. The
ASIC design would then be even more solid because obtaining the desired performance and functionality in the FPGA implementation virtually guarantees that these targets will be met in the ASIC silicon.

Finally, reconfigurable system prototyping provides functional verification only. While essential for validating the system design, it does not provide verification of issues critical to the final ASIC design such as timing, power consumption, etc. As such, prototyping is an invaluable complement to simulation tools, but it does not replace them.

**Recent Enhancements to System Explorer**

Working closely with customers, Aptix has continuously enhanced the capabilities of the System Explorer product line.

The approach of using plug-in FPGA prototyping modules has enabled the system’s emulation capacity to scale with Moore’s law. By leveraging the stunning advances in FPGA technology, Aptix has increased the nominal capacity of the System Explorer to nearly 2 million ASIC gates on a single board. As an added benefit, larger FPGAs further simplify the process of mapping design blocks to the prototype. System capacity will continue to increase as Aptix offers prototyping modules built with even larger FPGAs in the future.

Aptix has upgraded the System Explorer hardware to increase ease-of-use and support the latest IC components. The prototyping board is mounted in a chassis with integrated power supplies providing multiple supply voltages down to 1.5 volts. An integrated controller configures the prototypes FPGAs and FPIC devices. An Ethernet interface enables the system to be used as a network appliance for remote configuration and probing, while on-board, interchangeable flash EEPROM cards enable stand-alone configuration. Connections to Agilent logic analyzer pods are built into the chassis.

Aptix has enhanced the software environment to speed mapping and debugging of users’ designs. Logic AggreGATEr™, an interactive hierarchy management tool, simplifies logic partitioning. FPGA place-and-route function is encapsulated in the System Explorer software flow for increased ease-of-use. To increase visibility into large ASIC designs, System Explorer software automates the insertion of probes deep within the logic hierarchy through incremental routing of the FPGAs.

A key element of the block-based prototyping methodology is the Module Verification Platform (MVP) co-emulation interface. MVP is a hardware/software package that links the System Explorer to the user’s simulation environment. As users map each block of their design to the prototype, they use MVP to exercise the hardware-mapped blocks against a C-level system model running in their simulator. The co-emulation interface speeds block verification and integration. In addition, MVP dramatically accelerates regression testing. As users implement design changes and bug fixes in the prototype, they run regression test suites against the prototype and capture the results on their workstations for comparison with the expected results using third-party waveform viewers. By running regression tests at hardware speeds, MVP enables engineers to perform more verification cycles, thus producing higher-quality designs.

**Recent Enhancements**

- Tremendous advancements in FPGA technology greatly increase emulation capacity and ease partitioning
  - Single FPGA can emulate 250K ASIC gates, plus memory
  - Single prototyping board can emulate 2 million ASIC gates
- Enhanced software tools
  - Automated partitioning through hierarchical grouping
  - Encapsulated logic synthesis
  - Encapsulated FPGA place-and-route
- Simulator/emulator interface
  - Block-by-block verification of hardware-mapped design
  - Accelerated regression testing

**Recent Enhancements to System Explorer**

Working closely with customers, Aptix has continuously enhanced the capabilities of the System Explorer product line.

The approach of using plug-in FPGA prototyping modules has enabled the system’s emulation capacity to scale with Moore’s law. By leveraging the stunning advances in FPGA technology, Aptix has increased the nominal capacity of the System Explorer to nearly 2 million ASIC gates on a single board. As an added benefit, larger FPGAs further simplify the process of mapping design blocks to the prototype. System capacity will continue to increase as Aptix offers prototyping modules built with even larger FPGAs in the future.

Aptix has upgraded the System Explorer hardware to increase ease-of-use and support the latest IC components. The prototyping board is mounted in a chassis with integrated power supplies providing multiple supply voltages down to 1.5 volts. An integrated controller configures the prototypes FPGAs and FPIC devices. An Ethernet interface enables the system to be used as a network appliance for remote configuration and probing, while on-board, interchangeable flash EEPROM cards enable stand-alone configuration. Connections to Agilent logic analyzer pods are built into the chassis.

Aptix has enhanced the software environment to speed mapping and debugging of users’ designs. Logic AggreGATEr™, an interactive hierarchy management tool, simplifies logic partitioning. FPGA place-and-route function is encapsulated in the System Explorer software flow for increased ease-of-use. To increase visibility into large ASIC designs, System Explorer software automates the insertion of probes deep within the logic hierarchy through incremental routing of the FPGAs.

A key element of the block-based prototyping methodology is the Module Verification Platform (MVP) co-emulation interface. MVP is a hardware/software package that links the System Explorer to the user’s simulation environment. As users map each block of their design to the prototype, they use MVP to exercise the hardware-mapped blocks against a C-level system model running in their simulator. The co-emulation interface speeds block verification and integration. In addition, MVP dramatically accelerates regression testing. As users implement design changes and bug fixes in the prototype, they run regression test suites against the prototype and capture the results on their workstations for comparison with the expected results using third-party waveform viewers. By running regression tests at hardware speeds, MVP enables engineers to perform more verification cycles, thus producing higher-quality designs.