EFPGA Acceleration in SoCs — Understanding the Speedcore IP Design Process

achronix
SEMICONDUCTOR CORPORATION

August 01, 2017 **WP008**

Introduction

Programmable logic has a thirty-year history of enhancing functionality and adding value to electronic systems. Just as memories, analog and mixed-signal functions, standards-based protocol processing circuits and microprocessors did years ago, programmable logic has now reached a juncture in its maturity and utility as a technology where embedding FPGA functionality in SoCs is now a feasible solution — one that is changing the semiconductor landscape.

Speedcore™ IP is embedded FPGA (eFPGA) that can be integrated into an ASIC or SoC. Customers specify their logic, RAM and DSP resource needs, then Achronix configures the Speedcore IP to meet their individual requirements. Speedcore look-up-tables (LUTs), RAM blocks and DSP64 blocks can be assembled like Lego blocks to create the optimal programmable function for any given application. A personalized version of the ACE design tools to program the Speedcore IP is included with the Speedcore IP delivery. For a complete overview of Speedcore IP, see the *Speedcore eFPGA Product Brief* (PB028).

Achronix began shipping its Speedcore eFPGA IP to multiple customers in 2016. This paper describes the silicon-proven methodology for successfully integrating and deploying Speedcore eFPGA IP in an SoC design. The discussion covers how Speedcore IP is sized for a given application as well as the methodology details and tools flows, along with illustrations and references to specific Achronix documentation for further reference.

Overview

While the Speedcore design and integration methodology is highly flexible and can accommodate significant variations in SoC/ASIC development flows, there are some general principles and guidelines which, if followed, will make the development and integration effort simpler, smoother and more efficient:

- The Speedcore eFPGA requirements must be defined early in the ASIC development process, including the mix of Speedcore programmable logic blocks needed for the target design(s), the physical dimensions of the desired Speedcore eFPGA and the metal stack of the target process technology. Defining these aspects early will streamline the Speedcore integration effort, as it will allow for ASIC design details such as the interface between hardwired ASIC circuitry and the Speedcore I/O boundary, routing in the metal stack, power management and ASIC circuit block layout to be worked out quickly and efficiently.
- ASIC designs typically operate above 1 GHz, whereas FPGA designs typically operate between 300 MHz and 500 MHz. As a result, the functionality in the Speedcore eFPGA will typically need to operate on separate clock domains and require interface circuitry for crossing clock domain. It is important to define the interface between the host ASIC and the Speedcore instance early on and verify that the implementation meets performance both in the ASIC and Speedcore instance with real or example designs that will target the Speedcore instance once the ASIC is completed.

With the above in mind, a quick summary of the design and integration methodology is as follows:

1. **Run benchmarks:** The most important decision for a Speedcore implementation is determining how many programmable resources (LUTs, embedded memories, DSP blocks, etc.) are needed for the application. The best methodology to determine the optimal resource count is to run benchmarks through the Achronix CAD Environment (ACE) design tools using typical designs that will later be hosted in the final Speedcore instance. These designs should include any interface functions required to transfer data between the host ASIC and the Speedcore instance.
2. **Sizing the Speedcore instance:** Based on the benchmark results, the process technology and metal stack of the host ASIC, Achronix can provide the resulting die size and aspect ratio for the Speedcore instance. Speedcore IP is a fully modular architecture which can be scaled from small implementations with less than 10,000 LUTs up to very large implementations with over 1 million LUTs. There is also great flexibility in determining the right aspect ratio.
3. **Finalize business terms:** Achronix has a standard contract to license Speedcore IP and a specification template for the customer to define requirements for operating conditions, physical dimensions, resource counts, configuration modes and features, pin counts and interface details, clocks, size, performance, power, testing details and quality specifications. This contract forms a complete framework for defining a Speedcore implementation.
4. **Speedcore development:** There are three phases of delivery from Achronix during the development of a Speedcore instance. The first phase includes preliminary physical pin information and timing. The next phase includes all deliverables for the Speedcore instance along with preliminary timing. The final delivery includes all deliverables for the Speedcore instance with final timing. A version of ACE is included with each delivery phase.
5. **DFT:** Shortly after delivering the Speedcore IP, Achronix delivers DFT simulation and ATE vector files for testing.

A more detailed description of the methodology and its individual steps is presented below.

Speedcore Design and Integration Methodology Walk-through

This section describes the phases in a Speedcore engagement in further detail.

Phase Zero (Pre-Contract)

Customers interested in Speedcore eFPGAs will want to evaluate both Speedcore capabilities and the ACE design tools. While this evaluation process varies among customers, the common thread is that customers want to benchmark designs to determine performance, compile times and overall tool quality. Once the designs are selected, they can be compiled in the Achronix ACE design tools to determine resource requirements (i.e., the number of LUTs, memories, DSP blocks, etc.) and maximum performance. Typically, customers compile the same designs in competitive FPGA products to ensure Speedcore capabilities rival the quality-of-results (QoR) for leading-edge standalone FPGA products.

After completing the benchmark analysis, the customer will need to determine the size of the Speedcore instance needed for their application. This analysis is typically an interactive process where the customer specifies the resource requirements for LUTs, memories and DSP64 blocks for their Speedcore instance and then Achronix delivers die size, aspect ratio and other details about the proposed Speedcore instance. When the customer finalizes their requirements, Achronix delivers a complete specification that defines the size, performance, power consumption and other details about the custom Speedcore instance.

Phase One

Achronix delivers a preliminary version of the customer's specific Speedcore instance, including:

- A preliminary `.LEF` file that defines the Speedcore dimensions.
- A `.v` file that defines the Speedcore pin connections and includes fixed pin locations for configuration and debug pins along with a list of signal pin locations that can be changed by the customer before the next phase.

These files can then be employed in the ASIC flow as illustrated in the figure below.

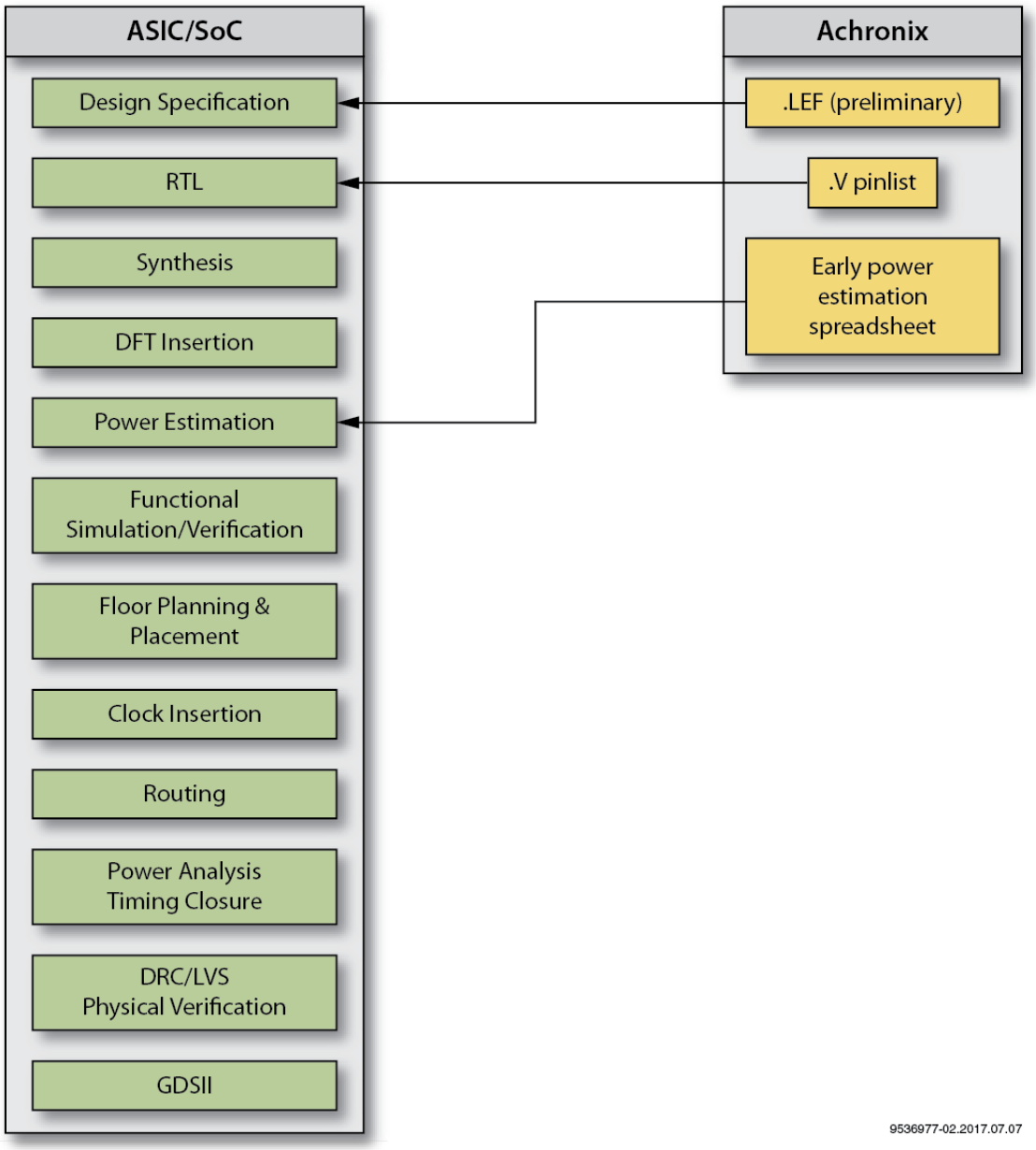

Figure 1: ***Phase One Deliverables***

The first version of the ACE design tools personalized for the specific Speedcore instance is also delivered to the customer. This version does not support full ASIC simulation or bitstream generation.

To estimate Speedcore power consumption, a power estimator tool is provided by Achronix. The power consumption of a given Speedcore instance is dependent on block mix, usage and environmental factors. Speedcore instance parameters are entered by the user and include:

- Device characteristics
- Thermal characteristics
- IP utilization for various block eFPGA types (BRAM, LRAM, LUT, DSP, clock and configuration circuitry), with separate entries for each block.

The user can select typical or maximum power settings and both static and dynamic power can be estimated.

The power estimator will help the ASIC development team's package designer account for factors such as maximum junction temperature limitations, heat sink and fan requirements, as well as board-level effects.

It is recommended that the power estimator be used early in a design to help in developing an overall estimate for ASIC and board power requirements. The results of the power estimator spreadsheet will also be useful as a reference point against the ACE post-implementation power report of the target Speedcore design. More information on power estimation can be found in the *Speedcore Power Estimator User Guide* (UG073).

Phase Two

Achronix delivers the final `.LEF` file with final Speedcore dimensions and preliminary models of the configuration circuitry of the Speedcore instance (shown below). An updated version of ACE is also delivered in this phase. This version offers timing models accurate to within 5% of final silicon.

Achronix provides an environment that allows for RTL simulation of the Speedcore instance with the full ASIC. More information on simulation can be found in the *Speedcore ASIC Integration and Timing User Guide* (UG064).

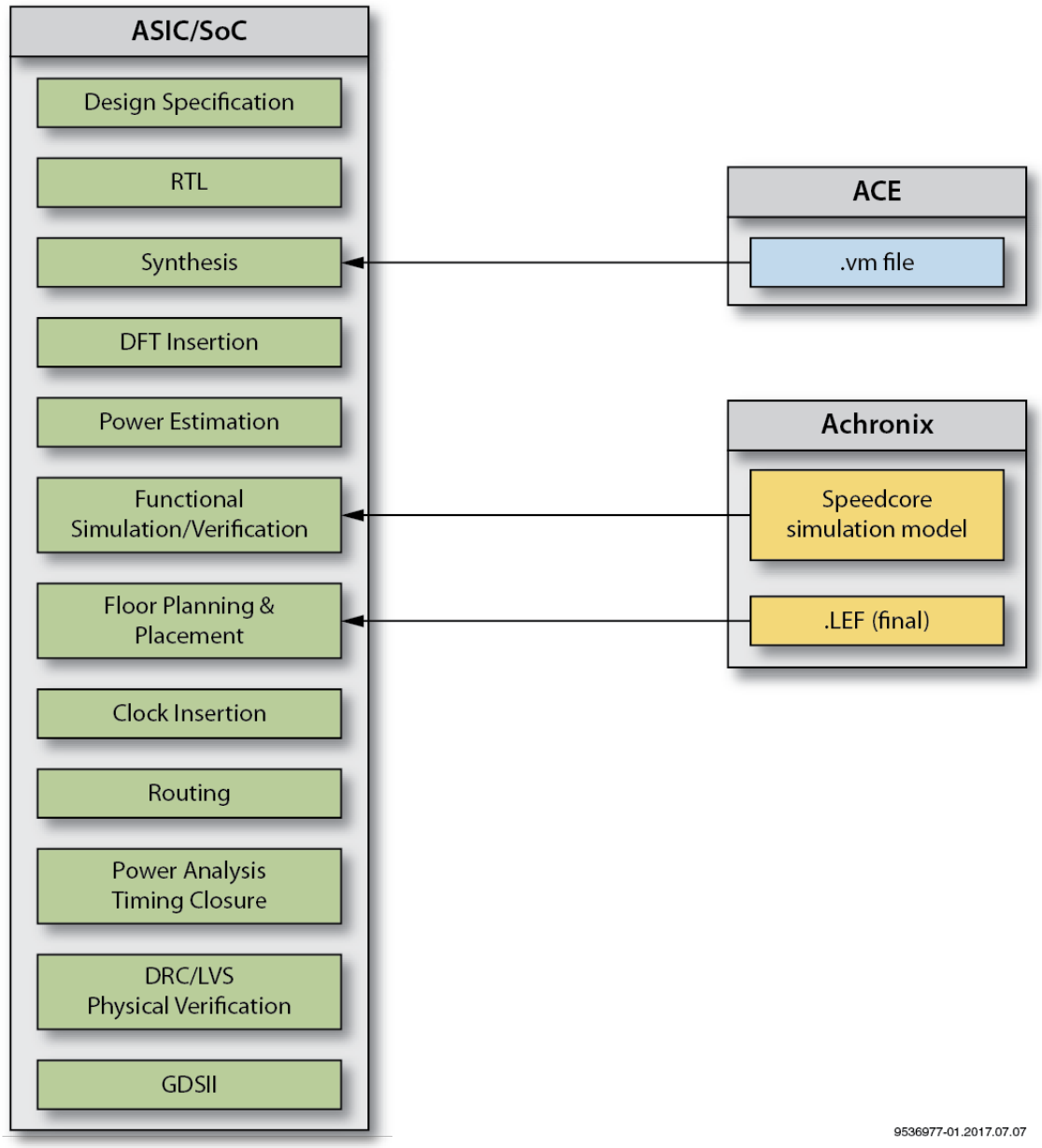

Figure 2: ***Phase Two Deliverables***

Phase Three

Several deliverables are provided in final form in phase three, including the `.v` file for simulating the full programming sequence, a DFT simulation netlist and a variety of items for production-level test. Files for ASIC-level timing closure are also delivered by Achronix or generated by the customer using ACE at this juncture as shown below.

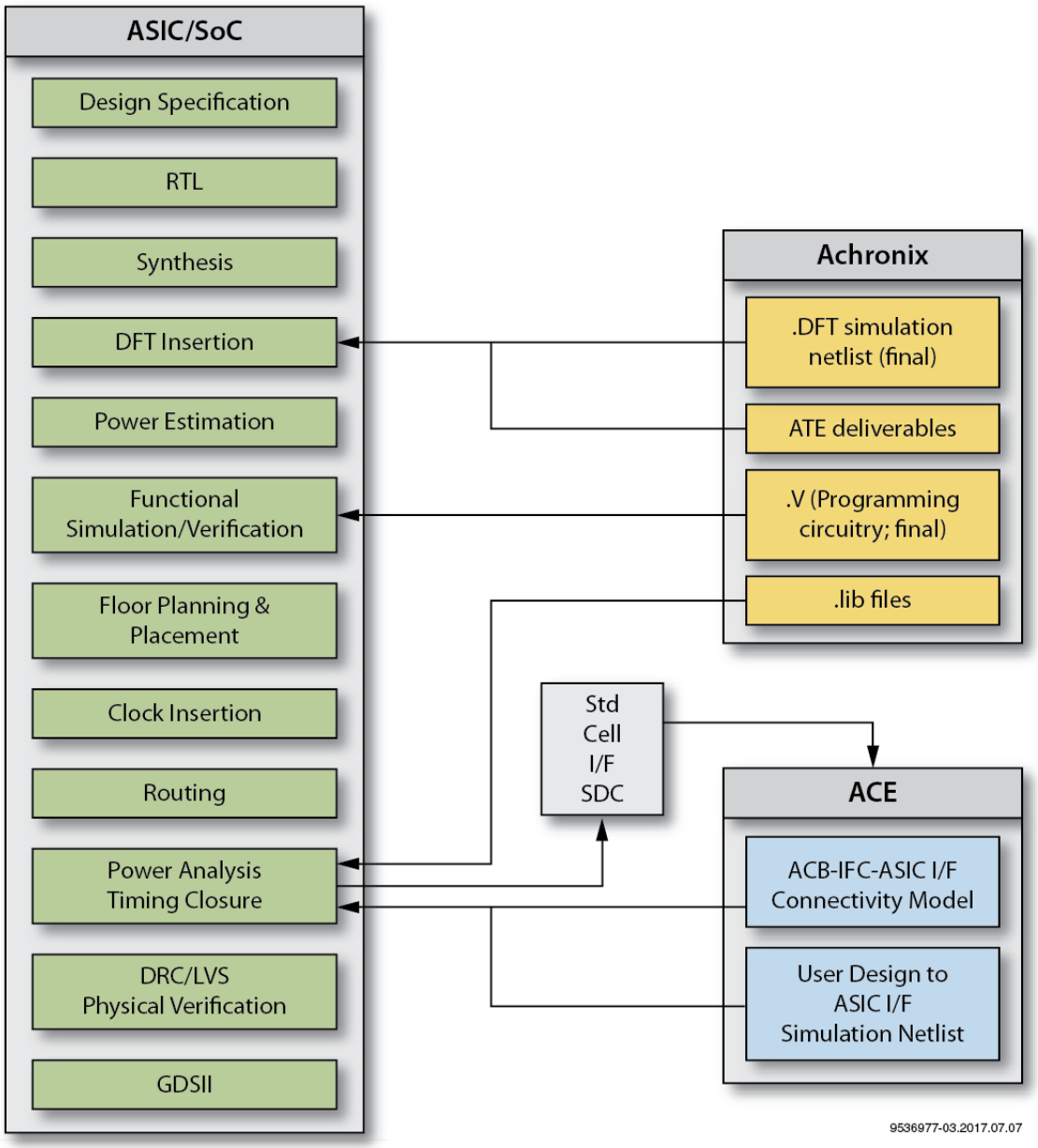

Figure 3: ***Phase Three Deliverables***

A third version of ACE is delivered in this phase, offering everything provided in the phase-two version along with full bitstream support.

In the Speedcore programmable fabric, the BRAM blocks, configuration RAM and the ACB can be scanned for stuck-at, transition delay and cell delay faults. The I/O boundary ring of the interface cluster (including the connections between the programmable fabric thru to the Speedcore I/O) is limited to testing for stuck-at defects.

Defect testing for the core interconnect, clocking scheme and all programmable blocks other than BRAM is supported by Achronix, which generates test vectors for these functions. Achronix delivers the following set of DFT/production test deliverables:

- A DFT simulation netlist describing scan connections
- A text file listing all SRAM locations

- BSDL file for JTAG registers (boundary scan is *not* supported)
- ATE deliverables, including the following:
    - Full production test suite in WGL format
    - ATE integration instructions
    - ATE test condition recommendations
    - Binning support (by request)

More details are available in the *Speedcore DFT and Test User Guide* (UG067).

At this point, the Speedcore instance can be completely integrated into the ASIC design, and timing can be closed for the ASIC in its entirety. At this stage of ASIC development, Achronix provides a simulation model for the entire Speedcore implementation all the way out to the core's I/O pins. This simulation netlist can be optionally timing-annotated at the gate level so that the customer can perform simulations to ensure functional and timing correctness.

Timing Closure

There are two approaches supported for ASIC-level timing closure with Speedcore IP: simple and advanced timing modes. Simple timing mode is used when the SoC connects to dedicated registers in the Speedcore peripheral I/O. Advanced timing mode is used when the SoC signals are registered inside the Speedcore FPGA fabric. The criteria for choosing the applicable timing mode depends on where system latency/clock cycle count is critical.

Simple Timing Mode

In simple timing mode, Speedcore timing is isolated from the ASIC by registering all signals in the Speedcore I/O boundary ring. This isolation allows the ASIC development team to, in essence, drive standard-cell timing per the requirements of the Speedcore design. In simple timing mode:

- Timing closure is owned by the customer using standard tools (e.g., PrimeTime)
- Achronix provides `.lib` files represent timing data to/from boundary flops (setup/hold/clock-to-q)
- Delays cannot be user-design dependent
- If ASIC-level delays are not passed to ACE, signals cannot cross cluster boundaries without using an asynchronous FIFO

Advanced Timing Mode

In advanced timing mode, both the ASIC interface to the Speedcore instance as well as the Speedcore I/O ring circuitry up to the programmable fabric are captured together. Advance timing mode is useful for where system latency/clock cycle count is critical.

Standard-cell timing and clock skew data at the interface between the host ASIC and Speedcore instance must be captured in an SDC file and fed into ACE. The SDC file also needs to include false path and multicycle path identification to allow ACE to understand the Speedcore boundary conditions with respect to the ASIC. The resulting constraints, along with all PVT and RC corner cases, need to be applied to every design intended for the Speedcore programmable fabric.

The basics of the advanced timing mode are as follows:

- Timing closure is shared between the customer (e.g., PrimeTime) and Achronix (ACE)

- Achronix provides `.lib` files represent timing data to/from all flip-flops in first clusters (setup/hold/clock-to-q)
- The customer first closes timing to the closest flip-flops at the target frequency
- Afterwards, delays for clock and data are extracted and fed to ACE (via constraint files)
- Finally, the set of sign-off designs are pushed through ACE to confirm performance. If desired performance is not met:
    - The flow is repeated with more aggressive ASIC timing (e.g., further flip-flops, higher frequency, etc.)
    - The user design is modified to run faster (e.g., fewer levels of logic)

When standard-cell timing has already been closed (for instance, when an ASIC has been shipped to end customers and is already deployed in the field), it becomes necessary to close Speedcore timing with respect to the ASIC in order to successfully implement any new configuration for the Speedcore programmable fabric. The ASIC development team will need to capture timing for the immediately surrounding standard-cell circuitry and feed this timing into ACE. In particular, the clock skew data between the last standard-cell register and the first Speedcore internal register will need to be captured.

Further information, including examples and various options, can be found in the *Speedcore ASIC Integration and Timing User Guide* (UG064) and *Speedcore Software Integration and Flow User Guide* (UG062).

Power Analysis

Final power analysis for the ASIC is also executed once timing has been closed. In order to properly account for the power profile of the Speedcore instance and integrate it properly into the power architecture of the ASIC design, the ASIC team must account for a variety of factors specific to the core. Every 16 nm Speedcore instance has three power rails:

- V_{DD} – for the FCU, boundary registers, JTAG TAP controller and ACB.
- V_{DDI} – for the programmable logic fabric, clock network and switch boxes.
- V_{CFG} – for the configuration RAM.

All three power rails can be shared on-die and treated like a single power rail, all sharing a common ground (V_{SS}). There are no slew rate requirements for the power rails. Each rail has its own robust ESD clamps, based on the IEEE HBM. Further details on power management, including power-up sequencing, can be found in the *Speedcore Power User Guide* (UG066).

Conclusion

The Speedcore design and integration methodology has been defined with intimate awareness of the difficulties ASIC engineering teams must contend with. All the necessary files and flows for capturing the functional, timing and power characteristics of a user-defined and programmed Speedcore instance, along with support for successfully reconfiguring an already field-deployed Speedcore IP embedded in an ASIC, are available to an ASIC development team either as products of the ACE design tools or as deliverables provided by Achronix. This methodology has already been proven in silicon and readily accommodates variations and preferences in company-specific ASIC development methodologies.

Achronix Semiconductor Corporation

2953 Bunker Hill Lane, Suite 101
Santa Clara, CA 95054
USA

Phone : 855.GHZ.FPGA (855.449.3742)
Fax : 408.286.3645
E-mail : info@achronix.com