How to Meet Power, Performance and Cost for Autonomous Vehicle Systems using Speedcore eFPGAs (WP015)

Achronix
Data Acceleration

January 18, 2019 **White Paper**

When DARPA announced its first Grand Challenge for driverless vehicles back in 2002, the race to develop self-driving vehicles started in earnest. Although engineers, scientists, enthusiasts, and science-fiction writers have dreamt of autonomous land vehicles for decades, DARPA's Grand Challenges helped make these dreams reality. No vehicle crossed the finish line when DARPA ran its first Grand Challenge in 2004; it was not even close, but things steadily improved. Five vehicles completed the 212 km race the following year. As DARPA hoped, the Grand Challenge competitions sparked a commercial race to develop commercially viable self-driving cars, trucks, and buses — a race that continues today.

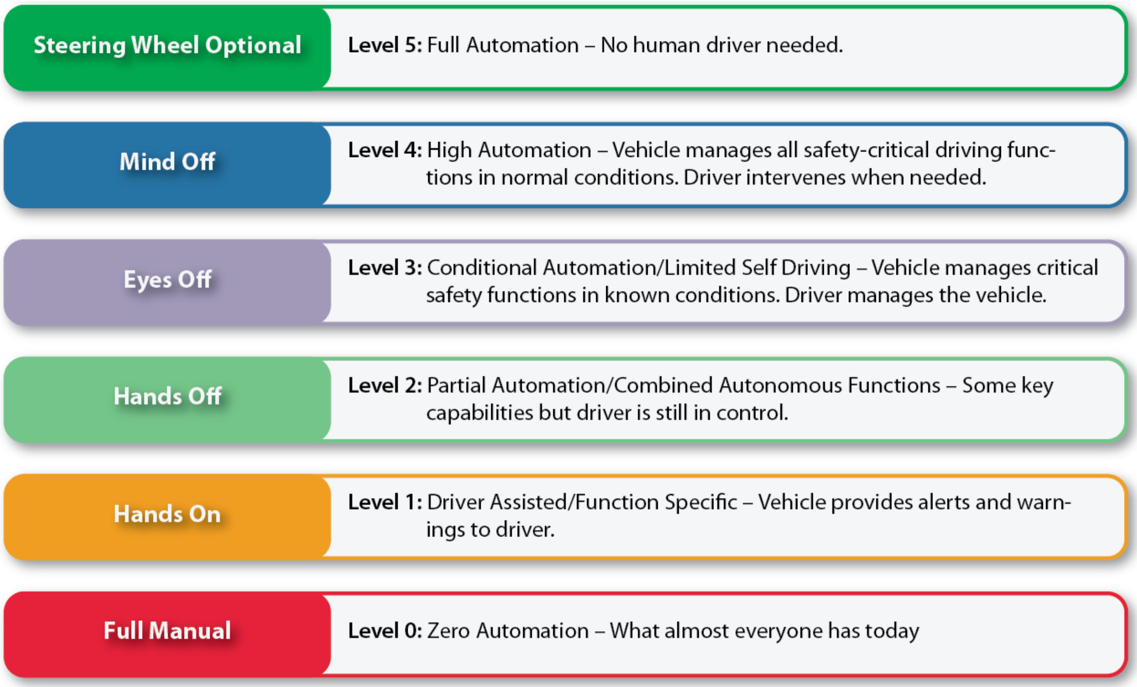
Automated vehicles are predicted to overcome a wide swath of society's challenges by improving traffic flow; increasing fleet fuel efficiency; providing enhanced mobility for non-drivers including children, the elderly, the disabled, and frequent users of public transportation; relieving tired and distracted travelers from driving and navigation chores; significantly reducing the need for parking spaces; and even reducing crime. At the same time, automated vehicles are already facilitating new business models for transportation as a service, especially via ride sharing.

AI and Self-Driving Architectures

On the technology side, development of self-driving vehicles made a major capability leap with the introduction of artificial intelligence (AI) in the form of trainable neural networks, deep learning, and sensor fusion. Trainable neural networks turned recognition algorithms upside down. Instead of hard-coding algorithms to recognize specific objects (other vehicles, pedestrians, lane markers, street signs, buildings, curbs, train and light-rail tracks, etc.), neural networks are trained to recognize individual objects through inference using object classes. Real-time neural networks require trillions of operations per second to make these inferences.

At the same time, these neural networks must process immense amounts of sensor data from visible-light and IR cameras, radars, lidars, ultrasonic sensors, and the like. This sensor data must be combined with map data to produce an accurate, real-time model of the vehicle's surrounding environment. Processing the data from all of these sensors in real-time is another huge computing challenge. Sensor-fusion and vision-processing systems help to reduce the computing load on neural-network inference engines.

The following figure illustrates the SAE's J3016 "Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles." As a vehicle's taxonomy level rises from level 0 (no automation) to level 5 (full automation), its computational needs rise exponentially.



Steering Wheel Optional — **Level 5:** Full Automation – No human driver needed.

Mind Off — **Level 4:** High Automation – Vehicle manages all safety-critical driving functions in normal conditions. Driver intervenes when needed.

Eyes Off — **Level 3:** Conditional Automation/Limited Self Driving – Vehicle manages critical safety functions in known conditions. Driver manages the vehicle.

Hands Off — **Level 2:** Partial Automation/Combined Autonomous Functions – Some key capabilities but driver is still in control.

Hands On — **Level 1:** Driver Assisted/Function Specific – Vehicle provides alerts and warnings to driver.

Full Manual — **Level 0:** Zero Automation – What almost everyone has today

Figure 1: ***The SAE's J3016 "Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles"***

Development of automated and autonomous vehicles represents a massive opportunity for every segment of the electronics supply chain including semiconductor and sensor vendors, IP suppliers, and systems companies. As EETimes' editor Junko Yoshida wrote back in 2016: *"From leading chip vendors to little known startups, it appears that nobody wants to miss the seemingly once-in-a-life-time opportunity to pile onto the self-driven bandwagon."* Consequently, the list of autonomous-vehicle developers is quite long, with hundreds of competitors.

In an article titled "Who's Winning the Self-Driving Car Race?", Bloomberg's David Welch and Elisabeth Behrmann published a developer scorecard in 2018. Their list of front-runners and other notable entrants included Waymo, GM, Daimler, Aptiv, Zoox, Renault-Nissan, Volkswagen/Audi, BMW, Toyota, Ford, Volvo, Hyundai, Fiat Chrysler, Uber, Tesla, and Baidu. This list is hardly exhaustive and, like the autonomous vehicular entries in DARPA's first Grand Challenge, not all of these entrants can be expected to cross the finish line.

The race to find the "right" architecture for a self-driving platform is ongoing. Just a few years ago, advanced driver-assist systems (ADAS) developers favored automotive intelligence models that assumed a centralized processing network in which multicore RISC CPUs with enhanced DSP capabilities managed a suite of specialized subnetworks. Today, the industry's direction is rapidly shifting towards decentralized automotive intelligence with sufficient distributed processing power to handle increasingly autonomous driving tasks at the higher SAE driving automation levels. These distributed systems employ complex cameras with associated vision systems, additional sensors (radar, lidar, etc.), sensor hubs, and other subnetworks to inform, manage, and integrate power, steering, braking, and drive systems; ADAS; navigation systems; and even in-vehicle infotainment systems. All of these diverse systems and subsystems must collaborate as an integrated whole in real time to implement safe and reliable automated/autonomous vehicles.

Meeting Self-Driving Performance Goals

Achronix anticipates that the favored self-driving architecture of the future will be increasingly decentralized, but both the centralized and decentralized architectural design approaches will require hardware acceleration in the form of far more lookaside co-processing than is currently realized. Whether centralized or decentralized, the anticipated computing architectures for automated and autonomous driving systems will clearly be heterogeneous and will require a mix of processing resources used for tasks ranging in complexity from local-area-network control, translation, and bridging to parallel object recognition based on deep-learning algorithms running on neural networks. As a result, the current level of more than 100 CPUs found in luxury piloted vehicles could easily swell to several hundred CPUs and other processing elements for more advanced, autonomous vehicles.

Sensor hubs require lookaside image processing for warp and stitch effects. Ethernet networks require IP for packet filtering/monitoring and for special bridges to handle legacy CAN and FlexRay networks. The power-hungry CPUs and GPUs used in first-generation autonomous automotive computing architectures will give way to highly-specialized compute nodes using programmable acceleration because these design alternatives deliver more processing power with far less power consumption.

At the same time, the greatly increased computing capabilities needed for automated and autonomous driving systems will require a similar boost in memory performance. A self-driving car's AI system requires a continuous, uninterrupted stream of data and instructions in order to make real-time decisions based on complex data sets. According to Robert Bielby, Micron Technology's Senior Director responsible for automotive system architecture in the company's embedded business unit, there is a growing memory bottleneck in current automated and autonomous driving systems designs and is already seeing industry momentum towards the adoption of GDDR6 DRAM to address that bottleneck. Bielby predicts that by the time automated and autonomous driving systems need more than 200 GBpsc of memory bandwidth, that GDDR6 memories will provide the lowest cost DRAM (per bit), at power levels equivalent to LPDDR5 DRAM.

The growing need for far more computing power and more memory bandwidth strongly suggests that future designs for automated and autonomous driving systems will increasingly use ASIC and SoC technologies to achieve the specific power, performance, and cost objectives of these extremely demanding automotive system designs. Conventionally, ASICs and SoCs lack the hardware flexibility needed for dealing with situations where the most critical algorithms are changing rapidly, as with the case of self-driving automotive systems. The most direct path to incorporating flexible, programmable processing elements into ASICs and SoCs is through the addition of embedded FPGA (eFPGA) IP cores.

The configurable processing capabilities achieved by integrating Achronix's Speedcore™ eFPGA IP into ASICs and SoCs optimize real estate and power efficiency and represent a superior design choice for implementing co-processing in future automotive platforms when compared to fixed-function SoCs and traditional FPGAs. To learn more about this evolution in processing, see *EFPGA Acceleration in SoCs — Understanding the Speedcore IP Design Process* (WP008).

Speedcore eFPGA IP's Unique Role in Heterogeneous Processing for Automotive Applications

Speedcore eFPGA IP can be integrated into ASICs or SoCs targeted at self-driving automotive applications to provide a customized, programmable fabric that delivers hardware-level processing performance. Speedcore eFPGA IP provides the hardware programmability and reprogrammability needed to deal with the rapid evolutionary changes occurring in these autonomous vehicle systems.

To incorporate Speedcore eFPGA IP into their ASICs and SoCs, designers specify the required logic, memory and DSP resources and Achronix then configures a customized Speedcore IP block that meets these requirements. Speedcore look-up-tables (LUTs), RAM blocks, and machine learning processors (MLPs, new with the Speedcore Gen4 architecture) can be assembled like building blocks to create an optimal programmable fabric for any given application.

Incorporating Speedcore eFPGA IP to an ASIC or SoC adds unique design advantages including:

- Programmable hardware resources that permit rapid changes to algorithms running at hardware speeds.
- Resources in the eFPGA can be configured as a programmable offload engine, which can handle complex algorithms that have not yet become standardized but still must execute at hardware speeds.
- Elimination of ASIC/SoC spins for half-step or even full-step enhancements.
- Higher security in the form of security and cryptography engines that can be changed or upgraded as required.
- The ability to make significant, remote updates to the system design if required.
- Programmable hardware resources that can be used, for example, to implement the programmable BIST engines needed for enumerating the CAN bus. These resources can be reused to implement other functions when the BIST function is not needed.

Vision Processing and Sensor Fusion for Self-Driving Vehicles

Vision systems have played a key role in the development of ADAS. Fusion of multiple vision-processing systems will continue to play a central role in both ADAS-equipped and automated/autonomous vehicles, even as the multicore vision processor is displaced from its perceived role as manager-of-managers as these systems become more distributed. Real-time image processing was originally considered to be a matter of information extraction from still-camera or video images to determine object type, location, and speed. As designers prepare for automated/autonomous vehicles, the image-processing role must expand to include fusion of visual, infrared, ultrasonic, lidar, and radar images.

Conventionally, image pre-processing is executed separately from the CPU, interfacing with the CPU via one or more high-speed buses. There is a latency penalty when co-processors are implemented in separate devices. Combining eFPGA IP with a CPU in a unified sensor-fusion architecture can be the most efficient way to ensure rapid response to visual, IR, or radar alerts of rapidly-changing traffic conditions.

Embedding a Speedcore eFPGA IP core with a CPU is an ideal way to create a sensor-fusion chip that integrates the continuous data streams from several sensor sources. Speedcore IP allows designers to embed a custom programmable fabric in a standardized SoC platform with specialized computing resources (see the figure below). In practice, this on-chip integration allows data aggregated from various imaging sources and other sensors to be written directly to on-chip SRAM or a CPU's cache rather than to standalone SDRAM. Reducing memory latencies in this manner improves real-time response to objects in a moving automobile's field of view.

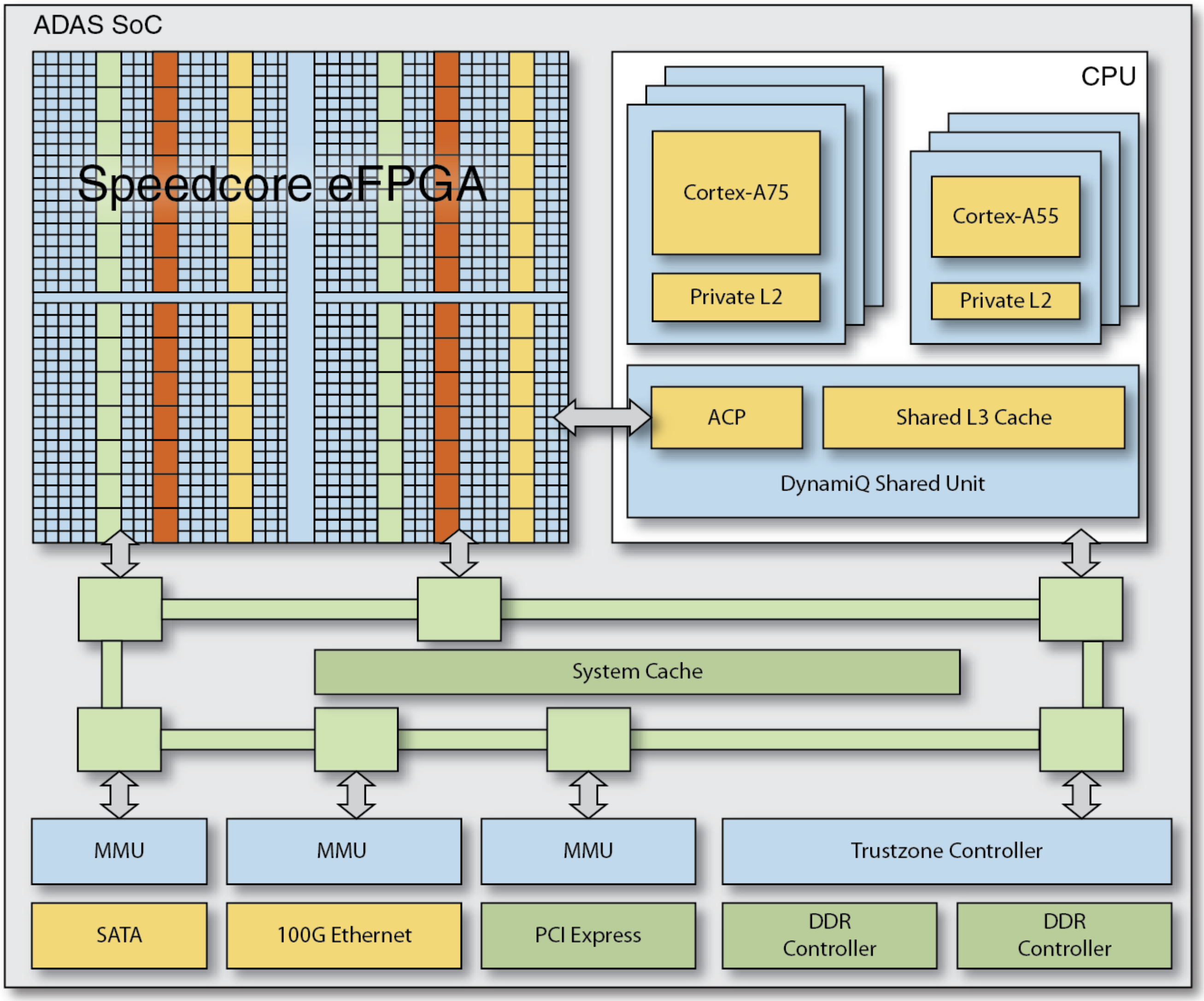


Figure 2: ***A Speedcore eFPGA Array (upper left) has Direct Access to a CPU's Cache and Memory Subsystem when these Elements are Combined on the Same SoC.***

Vision processors, which typically operate on 2D camera images, are increasingly including 3D imagery. These vision-processing systems rely on years of graphic-processor research in edge extraction, format conversion, color-balancing, and resolution changes. Some vision-processing vendors have promoted the value of convolutional neural networks in object classification and recognition. CPU vendors with experience in both worlds have tried to balance more traditional CPU/GPU tasks with specific neural-network pattern-recognition engines.

As neural-network sub-architectures in automotive applications migrate from early trained architectures requiring high-precision, floating-point DSP computations to self-training inference engines capable of using lower-precision, fixed-point computations, the MLP blocks available in the Speedcore Gen4 eFPGA architecture can provide plenty of processing headroom for advanced, deep-learning architectures.

Years of vision-processing experience lead to a common realization — there will never be an optimal centralized vision processor for real-time automotive situational awareness (or at least, such a processor will not exist for quite a while). There always will be unanticipated co-processing and acceleration tasks to add to the solution. The programmable nature of a Speedcore eFPGA is essential in this situation.

Two adjunct duties inherent in any ADAS processor are sensor-fusion/hub integration and network translation. The former involves combining and correlating information from a wide range of sensors: CMOS imagers, IR, lidar, and the now-emerging miniaturized radars. In addition, automotive designs require network translation to interface Ethernet backbones with older-but-still-useful automotive networks including CSI-2, FlexRay, CAN, and even older protocols. Again, Speedcore eFPGAs provide the flexibility needed to interface with any of these older network protocols.

eFPGA Programmability and Functional Safety

The transition from pilot-assisted ADAS to fully-autonomous, self-driving vehicles has intensified the role of safety in new vehicles. As self-driving electronics take more control of the vehicle, drivers expect to be swaddled in multiple levels of autonomous safety to prevent accidents and collisions. This need for fault-tolerant safety drove the development of the ISO 26262 safety standard for autonomous vehicles, a derivative of the IEC 61508 generic functional safety standard for electrical and electronic systems.

Early work in EDA and SoC communities has standardized ISO 26262 development methodologies for ensuring functional safety in IP. The failure modes, effects, and diagnostics analysis (FMEDA) technique spells out standard specifications for functionality and failure modes of IP elements, the effect of a failure mode on product functionality, the ability of automatic diagnostics to detect failure, the design strength, and the operational profile, including environmental stress. A robust self-driving system must maximize the diagnostic coverage of IP elements and deliver extraordinary functional safety by handling safe, detected, and undetected faults appropriately.

Embedded FPGAs enhance the safety of the vehicle as a system due to their programmable nature. Besides hosting in-flight vehicle functions, an eFPGA in an SoC can also host extensive hardware diagnostic routines that can run orders of magnitude faster than software-based diagnostics. Greater diagnostic speeds can greatly increase fault coverage for in-vehicle built-in self-test (BIST). Moreover, the eFPGA's programmatically aids in implementing the ISO 26262 safety life cycle by allowing automotive manufacturers the ability to update systems that are already deployed. For example, if the root cause of an accident was an error in the object-detection algorithm hosted in hardware (for performance reasons), a fix can be pushed out to the entire vehicle fleet as soon as it is developed, bypassing the lengthy and costly hardware development and redeployment process.

Distributed Control Implies Distributed Intelligence

Automotive designers increasingly assume significant distributed intelligence within the vehicle's chassis because of camera placement and the need for localized sensor hubs to reduce the volume of data to be centrally processed. Early proponents of multicore, multithreaded processors foresaw much of the sensor processor being centralized in or near the dashboard, albeit for highly parallel operations such as object recognition. Today, the fuzzy borders between advanced ADAS in piloted autos and the full autonomy of level-3-and-higher, self-driving cars has steered attention to more distributed intelligence, where CPUs, GPUs, and neural-network processors provide multiple points of management and control within the chassis. This shift implies more opportunities for programmable architectures that exist outside of one, all-encompassing SoC design.

Today, the ADAS processor market is growing by more than 25% per year. The migration of ADAS features – including automatic emergency braking, lane-changing assist, and adaptive cruise control functions – from luxury vehicles to midrange and even entry-level vehicles, is driving this growth. These ADAS features which will be almost universal by the middle of the next decade. In 2018, several manufacturers offered vehicles with self-driving platforms that delivered better-than-level-2 autonomy including GM's Super Cruise, Mercedes-Benz's Distronic Plus, Nissan's ProPilot Assist, and Tesla's Autopilot.

Meanwhile, level-3 autonomous platforms such as Audi's AI Traffic Jam Pilot system are not far behind with fully autonomous, level-5 vehicles possibly being available for commercial sale as early as 2022. As autonomous platforms evolve from level 3 to levels 4 and 5, sensor hubs, cameras, and lidar/radar devices will proliferate. Each of these subsystems requires local processing control.

Large semiconductor vendors that seek to steer development ecosystems toward particular domains of expertise will dominate the market for self-driving system processors. IP developers will attempt to disrupt these closed models by offering specialized processor cores while highlighting the differentiation that these cores enable.

This market somewhat resembles the days when enterprise networks were evolving into data centers. Processor-centric semiconductor suppliers sought to define the entire system architecture, while the design community displayed a diverse, Wild-West attitude, sampling different architectures to avoid giving one component supplier (and hence an OEM or automotive manufacturer) a proprietary edge. In such an environment, programmable logic configured as IP, including Achronix's multiple generations of Speedcore eFPGA, will play an important role both in near-term piloted and autonomous vehicle development and for many years to come as distributed architectural development for all types of vehicle subsystems evolves (learn more about Speedcore eFPGA IP here).

Speedcore eFPGA IP offers many design advantages in addition to raw processing power. These design advantages include minimizing memory-transfer overhead and CPU interrupts by allowing fused sensor data to be written directly to a CPU's cache or on-chip memory instead of off-chip memory. In addition, dedicated BIST circuits, which are required in CAN designs and often consume as much as 10% or 15% of an ASIC's or SoC's logic, can in many cases be eliminated by instantiating the BIST circuitry within the eFPGA – but only when needed. When the BIST circuitry is not required, the resources in the eFPGA can be reallocated using partial reconfiguration. Similarly, an eFPGA can implement on-chip probing functions for diagnostics, but only when required.

An eFPGA permits new algorithms to be programmed into fielded systems. This ability extends the life of deployed ASICs and SoCs in the field, if suitably equipped with an eFPGA. Coincidentally, the existing use of Speedcore IP in 5G cellular designs will enable yet-to-be-designed vehicle-to-anything (V2X) communication interfaces to be implemented in existing equipment.

In the advanced, fully autonomous, self-driving vehicles of the future, the existence of dozens and even hundreds of distributed CPUs and numerous other processing elements is assured. Peripheral sensor-fusion and other processing tasks can be served by ASICs, SoCs, or traditional FPGAs. But the introduction of embedded FPGA blocks such as Achronix's Speedcore eFPGA IP provides numerous system-design advantages in terms of shorter latency, more security, greater bandwidth, and better reliability that are simply not possible when using CPUs, GPUs, or even standalone FPGAs.

Achronix®

Data Acceleration

Achronix Semiconductor Corporation

2903 Bunker Hill Lane
Santa Clara, CA 95054
USA

Website: www.achronix.com
E-mail : info@achronix.com

NOTICE of DISCLAIMER: The information given in this document is believed to be accurate and reliable. However, Achronix Semiconductor Corporation does not give any representations or warranties as to the completeness or accuracy of such information and shall have no liability for the use of the information contained herein. Achronix Semiconductor Corporation reserves the right to make changes to this document and the information contained herein at any time and without notice. All Achronix trademarks, registered trademarks, disclaimers and patents are listed at http://www.achronix.com/legal.