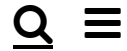


Advertisement

EE|Times



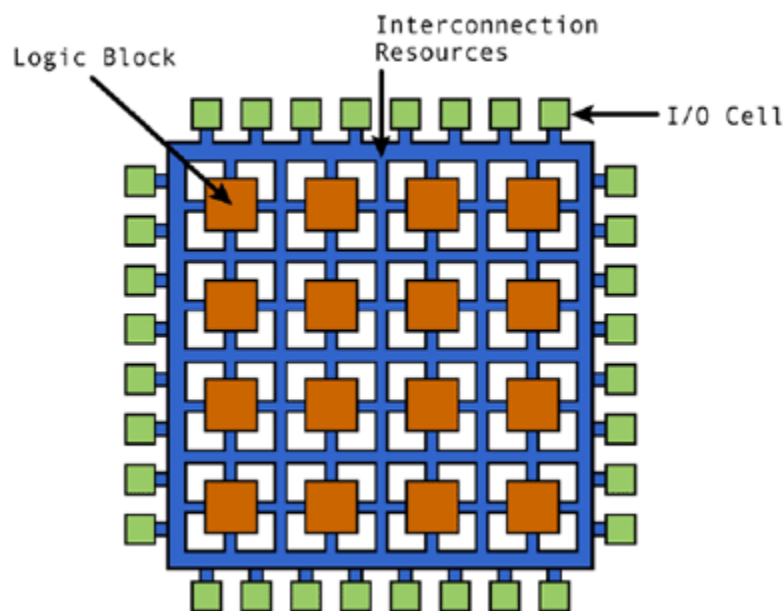
DESIGNLINES | **PROGRAMMABLE
LOGIC DESIGNLINE**

≤
<https://www.eetimes.com/designline/programmable-logic-designline/>

Introduction to Programmable Systems on a Chip

By Bob Zeidman, Zeidman Technologies 07.27.2005 0

By way of a short refresher, each FPGA vendor has its unique FPGA architecture. All are, however, in general terms, a variation of that shown in Figure 1. The architecture consists of configurable logic blocks, configurable I/O blocks, and programmable interconnect. Clock circuitry drives clock signals to each logic block and such additional logic resources as ALUs, memory and decoders may be available. The two basic types of programmable elements for an FPFA are static RAM and anti-fuses.



1. FPGA Architecture

Configurable logic blocks (CLBs) house the logic for the FPGA. Typically, these CLBs contain enough logic to create a small state machine. A CLB normally contains RAM to create arbitrary combinatorial logic functions also known as lookup tables (LUTs). Flip-flops for clocked storage elements and multiplexers to route the logic within the block and to and from external resources are also contained. Muxes allow polarity selection for outputs and reset and clear input selection for flip-flops.

A configurable I/O block is used to bring signals onto a chip and send them off again. It consists of an input buffer, an output buffer, and two flip-flops. One flip-flop is used to clock the output signal to shorten the clock-to-output delay for signals going off chip, while the other is used to register chip input, decreasing the device hold time requirement.

PARTNER CONTENT

VIEW
ALL

≤
» [HTTPS://WWW.EETIMES.COM/CATEGORY/SPONSORED-CONTENT/](https://www.eetimes.com/category/sponsored-content/)>



Solomon Systech Driving the Next-Generation Display Technologies <

<https://www.eetimes.com/solomon-systech-driving-the-next-generation-display-technologies/>>

≤

By Raymond Las Marias 12.13.2023

[w.eetimes](https://www.eetimes.com/solomon-systech-driving-the-next-generation-display-technologies/)



Embarking on a New Overseas Journey: G Camp Startup Team Showcases Taiwan's Technological Innovation Strength at Web Summit <

[https://www.eetimes.com/embarking-on-a-new-overseas-journey-g-camp-](https://www.eetimes.com/embarking-on-a-new-overseas-journey-g-camp-taiwan/)

[driving](https://www.eetimes.com/embarking-on-a-new-overseas-journey-g-camp-taiwan/)

By EE Times Taiwan 12.13.2023

[w.eetimes](https://www.eetimes.com/embarking-on-a-new-overseas-journey-g-camp-taiwan/)



Practical Tips for Accurate Bode Plot Measurements <

[https://www.techonline.com/tech-papers/eguide-practical-tips-for-accurate-bode-plot-measurements/?](https://www.techonline.com/tech-papers/eguide-practical-tips-for-accurate-bode-plot-measurements/)

[Overseas-](https://www.techonline.com/tech-papers/eguide-practical-tips-for-accurate-bode-plot-measurements/)

By Ronde & Schwarz 12.12.2023

[https://ww](https://www.techonline.com/tech-papers/eguide-practical-tips-for-accurate-bode-plot-measurements/)

[camp-](https://www.techonline.com/tech-papers/eguide-practical-tips-for-accurate-bode-plot-measurements/)

[w.techonli](https://www.techonline.com/tech-papers/eguide-practical-tips-for-accurate-bode-plot-measurements/)

[startup-](https://www.techonline.com/tech-papers/eguide-practical-tips-for-accurate-bode-plot-measurements/)

[ne.com/te](https://www.techonline.com/tech-papers/eguide-practical-tips-for-accurate-bode-plot-measurements/)

[team-](https://www.techonline.com/tech-papers/eguide-practical-tips-for-accurate-bode-plot-measurements/)

[ch-](https://www.techonline.com/tech-papers/eguide-practical-tips-for-accurate-bode-plot-measurements/)

[showcase](https://www.techonline.com/tech-papers/eguide-practical-tips-for-accurate-bode-plot-measurements/)

[papers/eg](https://www.techonline.com/tech-papers/eguide-practical-tips-for-accurate-bode-plot-measurements/)

[uide-](https://www.techonline.com/tech-papers/eguide-practical-tips-for-accurate-bode-plot-measurements/)

[practical](https://www.techonline.com/tech-papers/eguide-practical-tips-for-accurate-bode-plot-measurements/)

[tips-for-](https://www.techonline.com/tech-papers/eguide-practical-tips-for-accurate-bode-plot-measurements/)

[innovatio](https://www.techonline.com/tech-papers/eguide-practical-tips-for-accurate-bode-plot-measurements/)

[n-](https://www.techonline.com/tech-papers/eguide-practical-tips-for-accurate-bode-plot-measurements/)

[bode-plot-](https://www.techonline.com/tech-papers/eguide-practical-tips-for-accurate-bode-plot-measurements/)

[strength-](https://www.techonline.com/tech-papers/eguide-practical-tips-for-accurate-bode-plot-measurements/)

[measure](https://www.techonline.com/tech-papers/eguide-practical-tips-for-accurate-bode-plot-measurements/)

[at-web](https://www.techonline.com/tech-papers/eguide-practical-tips-for-accurate-bode-plot-measurements/)

[summit/](https://www.techonline.com/tech-papers/eguide-practical-tips-for-accurate-bode-plot-measurements/)

[summit/](https://www.techonline.com/tech-papers/eguide-practical-tips-for-accurate-bode-plot-measurements/)

[summit/](https://www.techonline.com/tech-papers/eguide-practical-tips-for-accurate-bode-plot-measurements/)

The programmable interconnect within an FPGA consists of a hierarchy of interconnect resources. Long lines can be used to connect critical CLBs or as buses within the chip. Short lines are used to connect local CLBs. Often, one or several switch matrices connect long and short lines together. Programmable switches inside the chip allow the connection of CLBs to interconnect lines and interconnect lines to each other and to the switch matrix, enabling routing of the design.

I/O blocks with high-drive clock buffers known as clock drivers are distributed around the chip.

These buffers connect to clock input pads, and drive clock signals from the outside onto fast, low-skew global clock lines within the FPGA.

Programming technology

There are basically two competing methods of programming FPGAs. The first, SRAM programming, involves small static RAM bits for each programming element. The other method involves antifuses, which consist of microscopic structures that, unlike a regular fuse, normally do not make a connection. A certain amount of current during device programming causes the two sides of the anti-fuse to connect. Some FPGAs use a third technology – flash RAM bits as programming elements.

Site>

	SRAM	Antifuse	Flash
Theoretical Speed	Slow	Fast	Slow
Actual Speed	Fast	Fast	Slow
Density	Larger structures Less dense	Smaller structures More dense	In-between
Volatility	Volatile	Non-volatile	Non-volatile
Reprogrammable	Yes	No	Yes
Security	Low (bitstream accessible)	High (bitstream not accessible)	High (bitstream not accessible)
Power Consumption	High	Low	Low

An advantage of SRAM-based FPGAs is that by using a standard fabrication process, it is easier to improve the technology to make faster and lower power FPGAs. Since SRAMs are reprogrammable, these FPGAs can be reprogrammed a number of times, even while in the system and operating. SRAM-based devices easily use the internal SRAMs as small memories in the design. Disadvantages are that they are volatile, which means that they must be reprogrammed each time the system is powered up, and a power glitch can potentially change their state. SRAM-based devices also have large routing delays.

In comparison, anti-fuse based FPGAs are non-volatile and the delays due to routing are minimal, making the devices faster in theory. In practice, because SRAM technology is mature and well understood, the speed of SRAM-based FPGAs and anti-fuse FPGAs are very close. Antifuse-based FPGAs tend to require lower power and are better at keeping design information secure as they do not require an external device to program them at power-up. Disadvantages are that they require a complex fabrication process, an external programmer to program them, and once they are programmed, they cannot be changed.

Flash programming offers the potential for combining the advantages of both SRAM and anti-fuses technologies. Flash-based devices have the advantage of SRAM devices in that they use a standard semiconductor process. They are non-volatile and therefore use less power and the intellectual property of the design is secure. Flash-based devices can be reprogrammed multiple times, even while operating in the system. A drawback is that they are slower than either SRAM or anti-fuse devices.

Programmable system on a chip

The definition of a programmable system on a chip (SoC) varies depending on its use. One definition is an FPGA that is so large and contains so many logic gates that it takes the place of an entire system that would have taken an entire board full of chips only a few years ago. Within the context of this article, a programmable SoC must include a microprocessor. Thus, any FPGA is a

programmable SoC if it includes a microprocessor so that it is both hardware programmable and software programmable.

An FPGA is a programmable SoC if it includes enough gates to allow the inclusion of a microprocessor design and provides support for such a design. These two definitions encompass the two types of processors in a programmable SoC, a hard processor or a soft processor. A third type of programmable SoC is one in which a chip contains various blocks of a microprocessor and peripherals that can be programmatically connected or disconnected. This differs from an F)GA-based programmable SoC in that the programmability is at a very high level of functionality and does not allow as much flexibility for low-level, user-defined functions to be designed on a chip.

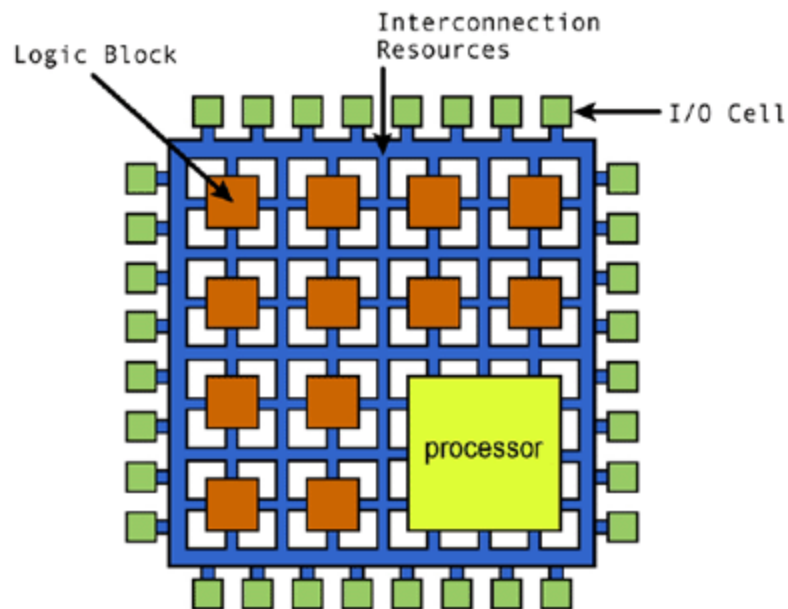
Hard Processor

A programmable SoC with a hard processor, or hard processor core) is an FPGA with circuitry for a microprocessor embedded in the chip, surrounded by programmable logic. As shown in Figure 2, the processor in the lower right corner is fixed circuitry that takes the place of some programmable logic in an ordinary FPGA.

An advantage of a hard processor is that the processor circuitry is optimized for timing and power consumption and can be characterized precisely by the vendor. A hard processor specified to run at 100 MHz will run at that speed, although there is no guarantee that the rest of the design will keep up. A hard processor specified to meet a specific amount of power will meet that criteria. The circuitry is fixed and the vendor has characterized the circuitry independently of the rest of the design.

Another hard-processor advantage is that the architecture is usually standard with support in terms of existing code, libraries, and such tools as compilers, debuggers, and operating systems from multiple vendors. The processor may be high end such as an IBM PowerPC, MIPS, or ARM processor capable of high-speed processing of very complex algorithms. It may be a low-end processor such as an Intel 8051 that features low power, takes up a small area, and is adept at simple hardware-related control functions. Often each programmable SOC vendor offers only one hardware processor and system requirements may limit the vendors and SoC families that can be used.

Hard-processor designs are compact rather than spread among CLBs, so less space on the chip is used. This translates to lower cost since, as a customer, cost is based on size of the chip. There is, however, a tradeoff regarding die area since the complete processor functionality exists, needed or not. For instance, when using a hard processor with a memory manager, should the code not require a memory manager, the unused feature raises the cost of the FPGA and the unused logic increases power consumption.



2. Programmable SoC with Hard Processor

Soft processor

Any processor with sufficient logic resources can be an SoC by including a soft processor. A soft processor is a logic description of a processor that can be included with the rest of the design, compiled into a gate level description, and placed and routed onto the FPGA. Typically, a soft processor is described in Verilog or VHDL and then combined with the remainder of the Verilog or VHDL design.

A soft processor advantage is that it can be configured optimally for a system. Unnecessary functionality can often be removed, although significant changes may render the software compiler and other tools unusable. The memory manager previously discussed can be removed in this case, as long as it is done without disturbing the other functionality of the processor.

A disadvantage of a soft processor is that it is specific to the FPGA vendor. A few third-party soft processors exist that run on FPGAs from multiple vendors that are well supported with software tools. It is unlikely, though, that any major high-performance processor vendor such as Intel or IBM will offer a soft version of its processor since that gives away valuable intellectual property.

A comparison of advantages and disadvantages of soft and hard processors is shown in Table. Although at first glance hard processors win hands down, it is important to make a fair comparison. Given that a soft processor can be configured to remove unnecessary functionality, a soft processor can be reconfigured to beat out a hard processor in each category.

	Soft Processor	Hard Processor
Configurability	High	Low
Predictability	Low	High
Software Tools	Fewer	More
Speed	Low	High

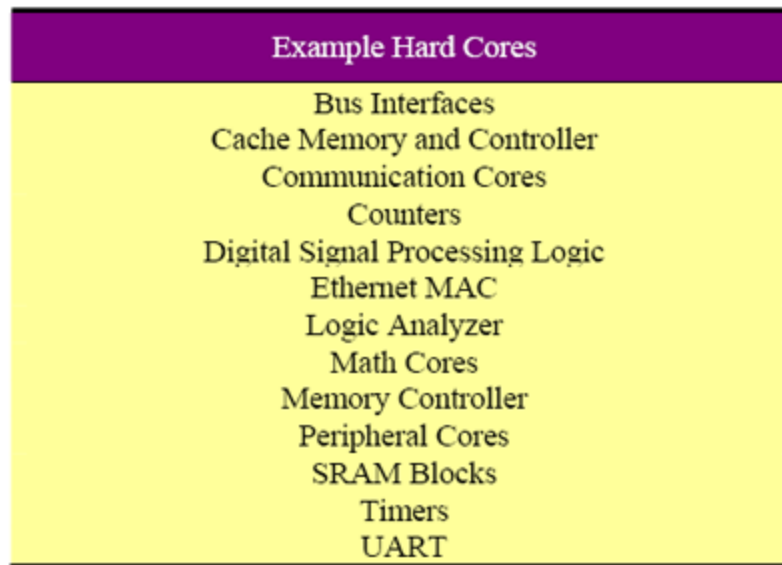
FPGA vendors and processors

A current list of FPGA vendors and the hard and soft processors they offer is shown in Table 3, while a list of third-party vendors that provide soft processors is shown in Table 4.

FPGA Vendor	Hard Processor	Soft Processor
Actel	none	third-party only
Altera	ARM	NIOS, NIOS II
Atmel	AVR	third-party only
Lattice	none	third-party only
Quicklogic	MIPS	third-party only
Xilinx	IBM PowerPC	MicroBlaze, PicoBlaze

Many programmable SoCs include hard cores to provide additional functionality. The advantage is that this functionality is compact, optimized, and predictable. The disadvantage becomes that hard cores use die space and consume power even if the system does not require the functionality the core provides. Table 5 contains examples of hard cores offered by FPGA vendors on their programmable SoCs.

Third Party Vendor	Soft Processor
ARC	ARC 600, 700, A4
ARM	ARM 7, 9E, 10E, 11 SecurCore
Cast	MPCore AMD 2901, 2910A, 29116A Dallas 80530 Infineon R80515 Intel C8051, C8051, C80186tx Motorola C6805, C6811, C68000 Ocean Logic AES Cryptoprocessor PIC C165X, C1655X Texas Instruments C32025 DSP X_DES Cryptoprocessor Zilog CZ80CPU, CZ80PSC
Digital Core Design	DR8051 DP80390 DFPIC1655X DF6805, DF6808, DF6811
MIPS	MIPS32, MIPS64 Pro Series

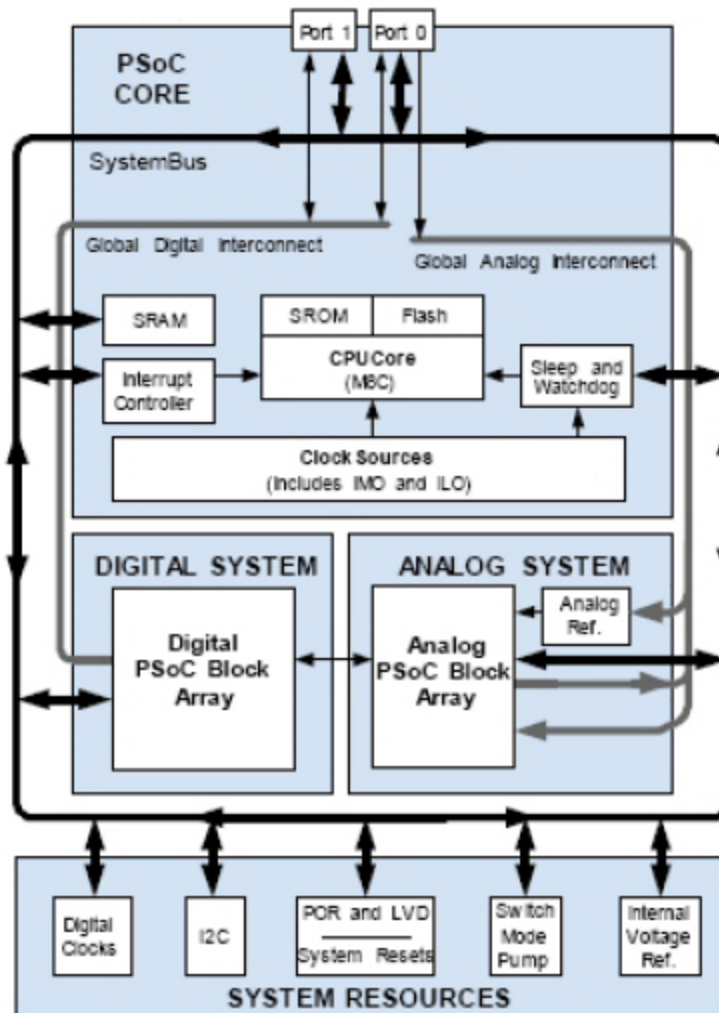


Configurable processor chips

Configurable processor chips come in many flavors. Unlike FPGA SoCs, there are no common architectures for these chips. Each vendor has a very different architecture and, in fact, a very different design philosophy for its SoCs. Because technology is continually changing, this is not an exhaustive list, but only a sampling of vendors at the time of writing.

Cypress Semiconductor

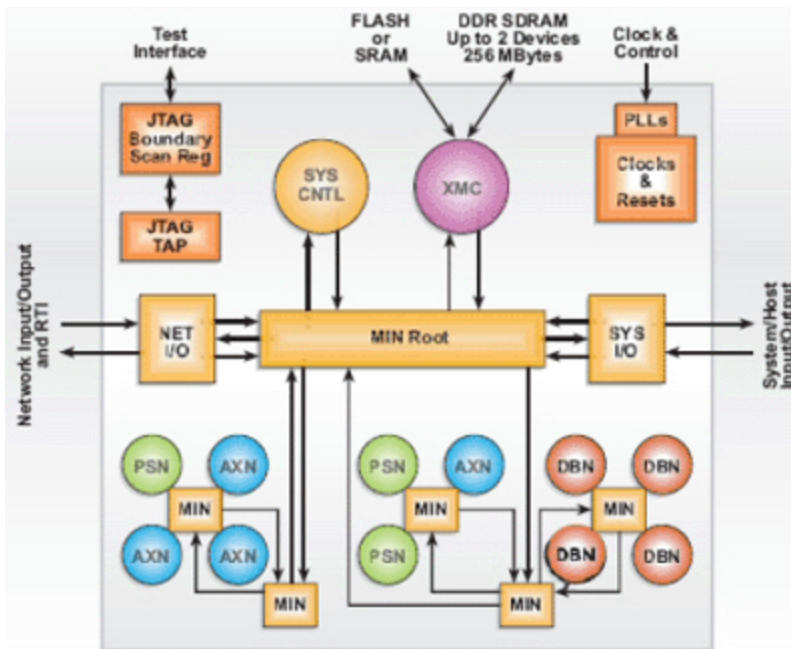
The Cypress PSoC family of chips has an architecture that includes an 8-bit processor and various peripherals that can be connected or left unconnected using SRAM-based programming elements. A unique aspect of these chips is that in addition to digital peripherals, analog peripherals are included as well. A typical PSoC architecture is shown in Figure 3. The analog units of a PSoC that can be configured in the chip include comparators and analog-to-digital converters (ADCs). The digital units include timers, counters, pulse width modulators (PWMs), Cyclic Redundancy Check (CRC) modules, a full-duplex UART, and a Serial Peripheral Interface (SPI) module. It also includes RAM and Flash memory.



3. Cypress PSoC Block Diagram

Quicksilver Technology

Quicksilver Technology has a programmable SoC that consists of multiple tiny processors that can each be programmed to execute low-level logic functions. The architecture of one of these devices, called an Advanced Computing Machine (ACM) is shown in Figure 4. At the bottom of the figure are the elements that make up the most unique aspects of the SoC. These are three different types of processing units, Adaptive Execution Nodes (AXNs), Domain Bit Manipulation Nodes (DBNs), and Programmable Scalar Nodes (PSNs). The nodes are connected via Matrix Interconnect Networks (MINs). PSNs are 32-bit RISC processors. AXNs can implement ALUs, MACs, multipliers, and address generators as well as other functions. DBNs perform bit manipulation that is useful for encryption, decryption, and error-checking functions.



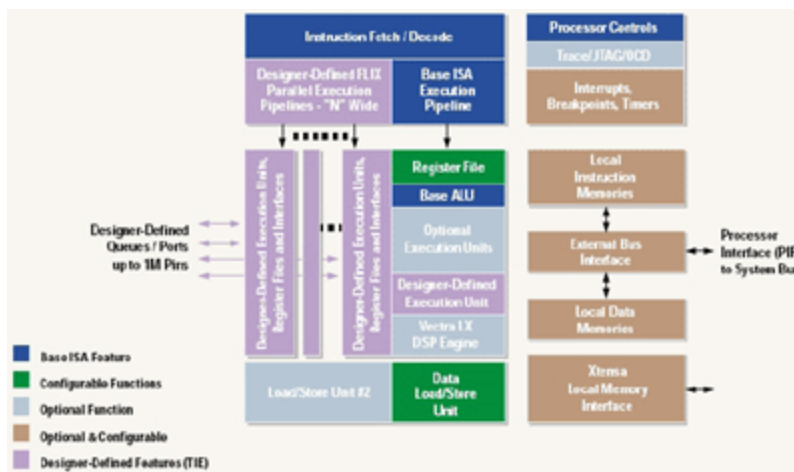
4. Quicksilver Adaptive Computing Machine

Sections of the device that communicate with the outside world are shown at the top of Figure 4. These include network I/O devices, system I/O devices, a JTAG interface for testing purposes, and a memory controller. The ACM architecture also includes a system controller that distributes tasks to each processor and allocates time for them. This system controller acts as a real-time operating system (RTOS) in hardware.

The ACM architecture is reminiscent of the Multiple Instruction Multiple Data (MIMD) parallel processors of the 1980s. These machines failed because of the difficulty of compiling standard programming languages to make use of the processors efficiently. New programming languages were devised for these architectures, but they were too difficult to use to write programs for all but very specialized applications. The MIMD approach is successful in grid computing where large chunks of programs can be allocated to entire processors on a network.

Tensilica

The Tensilica Xtensa processor architecture falls somewhere between a microprocessor and a programmable SoC. It is not truly an SoC by itself since it is only a processor and some limited peripherals. Yet, it is not just a microprocessor since it is programmable, allowing the processor functionality to be optimized specifically for the target system. The concept is that a user defines the execution datapaths, I/O ports, and registers needed for the particular application using a language very much like Verilog. The user also defines extensions to the instruction set for the processor, which has a Very Long Instruction Word (VLIW) architecture. Alternately, the user can write C code and use Tensilica's XPRES compiler tool to analyze the code and suggest processor implementation and instruction set extensions. Within Figure 5, the diagram is color coded to illustrate the parts of the chip that are fixed and those that are reconfigurable.



5. Tensilica XTensa LX Architecture

Conclusion

A variety of options exist when considering programmable systems on a chip. Traditional FPGA vendors offer two kinds of programmable SoCs, those with hard processor cores and those with soft cores. Tradeoffs include speed, power consumption, configurability, cost, predictability, and software support.

There are also programmable SoCs based on configurable processors available from several vendors. Each of these devices has a unique architecture and philosophy and tools to create and optimize the devices.

About the Author

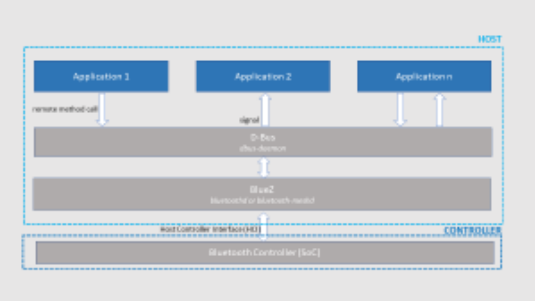
Bob Zeidman is the president of Zeidman Technologies, a company that develops tools for embedded system development, and of Zeidman Consulting, a contract research and development firm. Since 1983 he has designed ASICs, FPGAs, and PC boards for RISC-based parallel processor systems, laser printers, network switches and routers, and other real-time systems. Zeidman has authored three textbooks, *Designing with FPGAs and CPLDs*, *Verilog Designer's Library*, and *Introduction to Verilog*. He can be reached at bob@zeidman.biz.

Suggested Reading

- [Embedded in the IoT Era - EE Times](#)
- [What RF Wireless Needs Now - EE Times](#)
- [Sponsored content Archives - EE Times](#)
- [Nvidia Trains LLM on Chip Design - EE Times](#)
- [This Week: PowerUP Energy Virtual Event](#)

RELATED TOPICS: [DESIGN FOR MANUFACTURING < https://www.eetimes.com/tag/design-for-manufacturing/>](https://www.eetimes.com/tag/design-for-manufacturing/), [DESIGN MANAGEMENT < https://www.eetimes.com/tag/design-management/>](https://www.eetimes.com/tag/design-management/)

WIRELESS INDUSTRY RESOURCES



The diagram illustrates the Bluetooth architecture. At the top, within a dashed box labeled 'HOST', are three application blocks: 'Application 1', 'Application 2', and 'Application n'. Arrows labeled 'method call' and 'signal' point from these applications to a 'D-Bus (Linux) or Java RMI (Android)' layer. Below this is the 'BlueZ (Bluetooth or Bluetooth module)' layer. An arrow labeled 'Host Controller Interface (HCI)' connects the BlueZ layer to the 'Bluetooth Controller (SoC)' layer, which is located within a dashed box labeled 'CONTROLLER'.

Linux Study Guide

Bluetooth® Technology for Linux Developers – learn to create Bluetooth LE products using Python, D-Bus, and BlueZ.

GET THE GUIDE





Improve Vehicle Safety Using ADAS Technology

The Peripheral Touch Controller solution by Microchip simplifies the swift integration of low-power, highly sensitive and durable HoD systems.

[Learn More](#)



Radial Graphics Display IoT Gateway Application Reference Design

The PIC32MZ DA RGB MIWI protocol IoT gateway reference design builds a gateway for an IoT network using the MIWI protocol. The application sends and receives messages from the SAMR3018A MIWI protocol module, which is available on the Radial Graphics Board (RGB).

[Learn More](#)



Single Die

Hapti
HV56
Oper

RELATED ARTICLES



[Imec's Virtual Fab Goes Public to Speed IC](#)



[Manufacturing Decarbonization <](#)

<https://www.eetimes.eu/imec-virtual-fab-goes-eetimes.eu>

[PCB design system shortens development time <](#)

<https://www.edn.com/pcb-design-system-shortens-development-time/?>

edn.com