ChatGPT

FPGA Protocols in Embedded Systems: A Beginner's Guide

This guide explains the various protocols used with FPGAs, from how they are configured to how they communicate with peripherals and processors. We'll cover configuration protocols (how the FPGA is programmed), peripheral interfaces (connecting sensors, memory, and networks), on-chip buses (system-on-chip interconnects), and high-speed interfaces (for video, storage, etc.). Each section includes real-world use cases. Citations are provided throughout for reference.

FPGA Configuration Protocols

FPGAs must be programmed (configured) with a bitstream. Many FPGAs are *volatile* (SRAM-based) and lose their configuration when powered off, so they require loading a bitstream at each boot [1]. In contrast, *non-volatile* FPGAs (flash or antifuse-based) retain their configuration without external memory [1]. Common methods for programming FPGAs include:

- **JTAG (IEEE 1149.1)** – A standardized four-wire interface (TDI, TDO, TMS, TCK) originally designed for board testing [2]. For FPGAs, JTAG is often used during development for direct programming and debugging. It connects a programmer (or a PC) to the FPGA via a header. All JTAG-capable chips share the TCK and TMS lines in parallel, while the TDI (data-in) and TDO (data-out) pins are daisy-chained through each device [2]. The figure below shows a JTAG chain: the clock and mode lines (TCK/TMS) go to every device in parallel, and the TDI/TDO pins are chained from one IC to the next:

Figure: JTAG chain of devices. TMS/TCK run in parallel to all chips; TDI→TDO form a serial chain [2]. When using JTAG to program an FPGA, a tool like Xilinx's Vivado or Intel's Quartus detects the chip and loads the bitstream directly into the FPGA's configuration memory [3] [4]. **Use case:** JTAG is ideal for initial bring-up, debugging, and in-system programming. It does not require external flash memory, but needs a dedicated JTAG port on the board [2].

- **SPI Flash / Passive Serial (PS)** – In this mode, the FPGA's configuration bitstream is stored in an external SPI Flash memory. On power-up, the FPGA reads its bitstream from the flash through a serial interface (one data pin plus clock). For example, in *Passive Serial* mode (used by Intel FPGAs), a flash or microcontroller feeds bits to the FPGA's DATA0 pin on each clock edge [5]. Xilinx calls this *Master Serial* or *Slave Serial*, using a similar method with an external ROM or controller. Because the bitstream resides in flash, the FPGA configures itself automatically after reset [6]. This enables *non-volatile* configuration without a host: after loading once, the FPGA will reprogram itself every time it powers up [7] [8]. **Use case:** Production systems often use SPI flash so that the FPGA boots itself with the application logic on each power-up, without extra programming tools [6] [8].

- **Other Modes** – Some FPGAs also support parallel buses (e.g. Xilinx's SelectMAP or Intel's byte-parallel modes) for very fast configuration. These use multiple data lines to load the bitstream more quickly. They are less common for low-speed boards but useful when very fast boot or simultaneous configuration of multiple FPGAs is needed.

Volatile vs. Non-volatile: Most Xilinx and Altera FPGAs are SRAM-based (volatile), requiring a bitstream load at every power-up. They usually rely on an external flash memory to store that bitstream [1]. Non-volatile FPGAs (e.g. certain Microsemi or Lattice devices) have flash or anti-fuse so that no external memory is needed [1].

Peripheral Communication Protocols

FPGAs commonly interface with peripherals (sensors, memories, microcontrollers, etc.) over standard serial buses. We categorize these by complexity:

Basic Protocols: UART, SPI, I2C

- **UART (Universal Asynchronous Receiver/Transmitter):** A simple asynchronous serial link using two lines (Tx and Rx). UART communication is frameless: data is sent as a start bit, data bits, optional parity, and stop bits [9]. No clock line is shared; instead both devices agree on a baud rate. Typical embedded baud rates range from a few kbps to 1–2 Mbps. UART is point-to-point (one transmitter, one receiver). **Use cases:** Console/debug output (e.g., a serial terminal on a PC), short-distance comms (GPS modules, Bluetooth modules), sensor modules. *“UART is often used as a form of device-to-device communication in computer and microcontroller applications”* [10].

- **SPI (Serial Peripheral Interface):** A high-speed, synchronous master/slave bus. It uses at least four signals: SCLK (clock from master), MOSI (master-out), MISO (master-in), and SS/CS (slave select). The master drives SCLK, and data is exchanged on MOSI/MISO on each clock edge [11]. Because MOSI and MISO form a full-duplex link, SPI can transfer data in both directions simultaneously. There is no inherent protocol for packet framing; an SPI transaction simply clocks out bytes while the SS line is active. SPI has no hard upper speed limit – multi-MHz clock rates are common (speeds in excess of 100 MHz have been achieved [12]). Its multi-slave capability requires one SS line per slave, which can add wiring. **Use cases:** Fast sensors and memory. For example, ADCs, DACs, SD cards, EEPROMs, flash memories, and displays often use SPI. *“SPI is often used for...memory applications (SD cards, flash, etc.)”* [13]. An FPGA can act as SPI master or slave to talk to microcontrollers or peripherals.

- **I^{2}C (Inter-Integrated Circuit):** A two-wire synchronous bus (SDA data, SCL clock) with multiple masters and slaves. I2C signals are open-drain and require pull-up resistors; all devices share the same two lines. Each transaction includes a 7- or 10-bit address frame and read/write bit, followed by data and acknowledgments [14]. The master drives the clock and initiates transfers. Standard I2C runs at 100 kbps, though “fast mode” is 400 kbps and “ultra-fast” up to 5 Mbps [15]. I2C is designed for lower speeds and many devices on a single bus. **Use cases:** Short-range, low-speed peripherals like temperature sensors, EEPROMs, real-time clocks, PMICs, and other I/O chips. *“I2C is often a good choice for connecting...low-speed devices like sensors in an embedded system”* [16].

Figure: SPI (left) vs I^{2}C (right) bus. SPI uses separate MOSI/MISO lines and SS lines for each slave; I2C uses shared SDA/SCL with addressing. Both can connect multiple devices [11] [15].

Comparison (UART/SPI/I2C):

Protocol	Wires/Signals	Data Rate	Topology	Typical Use Cases
UART	2 (Tx, Rx, asynchronous)	Baud rate (e.g. 115 kbps common)	Point-to-point	Serial console, PC link, GPS, modems [10]
SPI	≥4 (SCLK, MOSI, MISO, SS)	High (MHz range) – full-duplex [17]	1 master, multiple slaves (indiv. SS)	High-speed sensors, ADCs/DACs, flash memory [13]
I^2C	2 (SDA, SCL, open-drain)	100 kbps standard (up to 5 Mbps) [15]	Multi-master/ slave bus	Low-speed sensors, EEPROM, RTC, PMIC [16]

Intermediate Protocols: CAN, LIN, USB

- **CAN (Controller Area Network):** A differential multi-master bus widely used in automotive and industrial systems. CAN transmits short frames (up to 8 data bytes, or more with CAN-FD) with built-in arbitration and error checking. It operates in half-duplex: any node can attempt to transmit, but priority-based arbitration (lower ID = higher priority) prevents collisions [18]. Typical speeds are 125 kbps to 1 Mbps (CAN 2.0). **Use cases:** Vehicle networks – e.g. engine control units, ABS, airbags – where reliability and fault handling are critical. *“The CAN bus is a common protocol that allows different ECUs in an automobile to communicate… Its reliable and efficient communication qualities are why it is widely employed in modern cars”* [19] [20]. Many FPGAs include or interface to CAN controllers for automotive applications.

- **LIN (Local Interconnect Network):** A simpler, single-wire serial bus (often one-wire + ground) used as a lower-cost complement to CAN in cars. LIN runs at lower speeds (up to ~19.2 kbps [21]). It uses a master/slave schedule: the master polls slaves by sending headers; only the addressed slave responds. LIN has no multi-master arbitration. **Use cases:** Non-critical automotive subnets, such as window/seat controllers, climate control actuators, and body electronics. The NI documentation notes LIN “provides cost-efficient communication in non-critical applications where the bandwidth of CAN is not required” and is often implemented using a standard UART in a microcontroller [22].

- **USB (Universal Serial Bus):** A host-controlled, tiered-star bus. A USB host (e.g. a PC or embedded host controller) polls devices connected via hubs. USB defines various speeds: Low Speed (1.5 Mbps), Full Speed (12 Mbps), Hi-Speed (480 Mbps, USB2.0), and SuperSpeed (5–20 Gbps, USB3.x). For an FPGA, USB is typically used to connect to a PC or microprocessor as a peripheral (or with a USB PHY/ IP for host mode). **Use cases:** Data acquisition, FPGAs acting as USB devices (e.g. USB-based logic analyzers), FPGA PCIe bridges, etc. (Although USB integration in FPGA is complex, many dev boards include an on-board USB-UART chip for simple PC connectivity.) According to Microchip, a single USB host can support up to 127 devices, and it controls all communications [23]. USB’s host-centric protocol and relatively high speed make it suitable for PC/FPGA interfaces (e.g. streaming data to/ from a PC).

Advanced Protocols: Ethernet, LVDS, Custom Serial

- **Ethernet (RMII/GMII/RGMII):** FPGAs often implement Ethernet MAC cores and connect to external PHY chips via standard interfaces. RMII (Reduced MII) is a 2-bit wide bus at 50 MHz for 10/100 Mb/s Ethernet, while GMII (Gigabit MII) is an 8-bit bus at 125 MHz for 1 Gb/s Ethernet [24]. (There are also RGMII, SGMII, etc.) Ethernet links (twisted-pair or fiber) carry 100Mb or 1Gb (or 10Gb+) of data. **Use cases:** Networked embedded systems – streaming video, IoT gateways, etc. For example, an FPGA might process data from cameras or sensors and send results over Ethernet. Many FPGAs include built-in Ethernet MACs.

- **LVDS (Low-Voltage Differential Signaling):** LVDS is a generic high-speed serial interface using pairs of wires with low-voltage differential signaling [25]. It is often used for FPGA I/O because it supports several hundred Mbps per lane. Unlike I2C or SPI, LVDS itself is just a physical layer standard (no defined packet protocol). *"LVDS is a high-speed… interface for serial communication (sending one bit at a time) over two copper wires… often used in SerDes configurations"* [25]. **Use cases:** Camera links, display links, ADC/DAC data lines. For example, FPGA video applications use LVDS TMDS pairs to drive HDMI, and some sensors use LVDS. High-speed ADC/DAC chips output multiple LVDS lanes.

- **Custom Serial / FPGA SerDes:** Many modern FPGAs include high-speed serial transceivers (SerDes) that can implement proprietary or standardized protocols. Examples include Xilinx's Aurora or SerialLite, or user-defined multi-gigabit links. These allow designers to create custom high-bandwidth links between FPGAs or to ASICs. Custom serial protocols might be used when off-the-shelf standards don't fit the application.

FPGA Interconnect and SoC Protocols

Inside an FPGA-based SoC (e.g. a hard/soft CPU plus peripherals), standard bus protocols connect masters (CPUs, DMA) to slaves (memories, peripherals). Key protocols include:

- **AMBA (ARM's bus family):** Widely used in industry.
- **APB (Advanced Peripheral Bus):** A simple, non-pipelined bus for low-speed peripherals [26]. It uses separate cycles for address and data, and supports simple read/write without bursts. APB is ideal for lightweight peripherals like UARTs, I²C controllers, timers.
- **AHB (Advanced High-performance Bus):** A shared bus with higher throughput (buses can be up to 64-bit) and burst transfers [27]. Multiple masters/slaves share the bus with arbitration. AHB supports pipelining and burst bursts (for DMA, memory). Used in AMBA2/3 designs.
- **AXI (Advanced eXtensible Interface):** A point-to-point, high-performance protocol introduced in AMBA3/4 [28]. AXI supports separate read/write channels, out-of-order transactions, and multiple outstanding bursts. It overcomes scalability limits of shared buses by using a configurable switch/crossbar interconnect. AXI-stream is a variant for unidirectional streaming data.

- **Comparison:** APB = simple reg access (no bursts) [26]; AHB = shared high-speed bus with bursts [27]; AXI = fully pipelined, burst-capable, multi-master interconnect [28].

- **Intel/Altera Avalon:** A proprietary bus standard used in Intel FPGAs (formerly Altera). Avalon defines several interface types (Avalon-MM for memory-mapped, Avalon-ST for streaming, etc.) and is functionally similar to AMBA [29]. Avalon-MM is the equivalent of a memory-mapped bus (like AXI or AHB), while Avalon-ST is for streaming data. These allow soft processors (like Nios II) and custom IP cores to interconnect easily on Intel FPGA platforms [29].

- **Wishbone:** An open-source bus specification for on-chip interconnect [30]. It defines a simple register-based interface to allow IP cores to communicate without licensing. Wishbone is flexible and used in many open-source FPGA cores. It is not tied to a specific company and is royalty-free [30]. **Use case:** hobbyist or academic FPGA projects and open-source SoC designs often adopt Wishbone to plug together IP from different sources.

(When designing a complete FPGA-based system with a CPU, designers often use one of these bus protocols to connect all the blocks. For example, Xilinx Zynq SoCs use AXI internally, and Intel SoC FPGAs use Avalon with bridges to AXI for ARM cores. Tables comparing AMBA protocols are given above.)

High-Speed and Advanced Interfaces

Modern FPGAs can implement very high-speed protocols, often with hard IP blocks for the physical/link layers. Key examples:

- **PCI Express (PCIe):** A high-bandwidth serial bus used to connect FPGA accelerators to a host PC or CPU. PCIe v3.0 (8 Gb/s per lane) and v4.0 (16 Gb/s) are common. PCIe is scalable (x1, x4, x16 lanes). An FPGA with PCIe can appear as a high-speed peripheral to a CPU, enabling applications like data acquisition cards or GPU-like accelerators. Intel describes PCIe as a “high-performance, scalable, serial protocol with data rates from 2.5 GT/s to 32 GT/s” [31]. **Use case:** FPGA boards with PCIe plug into PCs for fast I/O, e.g. RAID controllers, network cards, accelerators.

- **SATA (Serial ATA):** A serial bus for connecting storage drives (up to 6 Gb/s in SATA III). FPGAs with SATA MAC/PHY IP can interface to SSDs or HDDs. **Use case:** Custom storage controllers, test equipment for drives.

- **HDMI (High-Definition Multimedia Interface):** A digital video/audio interface using four TMDS differential pairs [32]. FPGAs can drive displays by outputting video data on these pairs. HDMI carries high-definition video (e.g. 1080p, 4K) and audio. **Use case:** FPGA video generators and image processing; for example, an FPGA can encode video and send it to a monitor via HDMI. The fpga4fun tutorial notes “A standard HDMI connector has 19 pins… 8 of [them] form 4 TMDS differential pairs to transport the high-speed video info” [32].

- **MIPI CSI/DSI:** Camera and display serial interfaces used in mobile and embedded devices.

- *CSI-2* (Camera Serial Interface) is a widely used high-speed protocol for image sensors [33]. It uses D-PHY lanes (typically 1–4 differential pairs) to carry raw or YUV camera data up to multiple Gbps per lane.
- *DSI* (Display Serial Interface) is a high-speed bus for displays [34]. It sends graphics/video frames from a host to a panel. FPGAs (especially SoC FPGAs in mobile/embedded boards) often have MIPI IP or transceivers for CSI-2/DSI.

- **Use cases:** Smartphone cameras and screens, automotive cameras, vision systems. For example, an FPGA may interface directly to a camera module via CSI-2 to capture video for processing.

- **Others:** FPGAs also support protocols like **HD-SDI** (for broadcast video), **DisplayPort**, **Thunderbolt**, **10GbE/40GbE Ethernet**, etc., via transceivers. Emerging interfaces like **PCIe Gen4/5** and **10Gb/s Ethernet** are commonplace in the latest FPGA families.

Use Cases by Protocol Family

- **Configuration protocols:** *JTAG* is used in development and debugging (e.g. programming an FPGA on a dev board via USB-JTAG). *SPI flash* is used in end products to auto-boot the FPGA. For instance, a video processing board might store its FPGA bitstream in SPI flash so it powers up configured.

- **UART/SPI/I2C:** A microcontroller might talk to an FPGA over SPI to offload computations, or sensors might feed data into an FPGA via I2C (e.g. temperature sensors, accelerometers on an FPGA-based robotics board). UART is ubiquitous for debug consoles (many FPGA dev boards include a USB-UART chip to allow printf-style debugging).

- **CAN/LIN:** Automotive FPGAs (e.g. in advanced ADAS or vehicle body systems) connect to the vehicle network via CAN or LIN controllers. An FPGA handling engine control or sensor fusion would use CAN for main ECU comms.

- **USB/Ethernet:** FPGAs on PCIe cards often interface to the system via Ethernet or USB. For example, a data-acquisition FPGA board might send data to a PC over Gigabit Ethernet.

- **AMBA/Avalon/Wishbone:** In a complex FPGA SoC design, on-chip buses connect CPU(s) to memory and peripherals. For example, a Zynq (ARM+FPGA) design uses AXI to connect a CPU to FPGA accelerators; an Intel FPGA might use Avalon to hook a Nios II CPU to SPI, UART, and other cores; an open-source RISC-V on FPGA might use Wishbone.

- **PCIe/SATA/HDMI/MIPI:** High-speed links often define the whole application. For instance, an FPGA-based accelerator in a server uses PCIe to talk to the CPU and might capture data from disk over SATA or streams data to a GPU over 10Gb Ethernet. An FPGA video card outputs HDMI to a monitor. A drone's vision FPGA uses MIPI CSI to ingest camera feeds.

Tables: The tables above compare the key attributes of common protocol families (basic serial and AMBA bus protocols). Each has trade-offs (speed, complexity, topology) that suit particular embedded uses.

Summary: FPGAs support a rich set of protocols. Configuration protocols (JTAG, SPI, etc.) load bitstreams; basic peripherals (UART, SPI, I^2C) connect simple devices; automotive and USB/Ethernet cover mid-range needs; and advanced high-speed interfaces (PCIe, HDMI, MIPI) enable heavy data transfer. Internal bus standards (AXI, Avalon, Wishbone) organize on-chip communication. By choosing the right protocol, FPGA designs can interface effectively with nearly any sensor, actuator, memory, or network in embedded systems.

---

[1] Xilinx redefines the non-volatile FPGA landscape - EE Times
https://www.eetimes.com/xilinx-redefines-the-non-volatile-fpga-landscape/

[2] fpga4fun.com - JTAG 1 - What is JTAG?
https://www.fpga4fun.com/JTAG1.html

[3] [4] [6] [7] [8] Xilinx FPGA Programming Guide: JTAG, SPI Flash, and Vivado Tools for Spartan 6 & Zynq - RayPCB
https://www.raypcb.com/xilinx-fpga-programming/

[5] Passive Serial Configuration (PS) | Intel
https://www.intel.com/content/www/us/en/support/programmable/support-resources/configuration/cfg-ps.html

[9] [10] [11] [12] [13] [14] [15] [16] [17] I2C vs SPI vs UART – Introduction and Comparison of their Similarities and Differences - Total Phase
https://www.totalphase.com/blog/2021/12/i2c-vs-spi-vs-uart-introduction-and-comparison-similarities-differences/?srsltid=AfmBOorE4LYqP8eAx1fcSVWSqcTpy4yBA13vurakyFBD_YoNx4Fsl7s-

[18] [19] [20] Ultimate Guide to Controller Area Network (CAN) (2025)
https://www.logic-fruit.com/blog/can/controller-area-network-can-guide/?srsltid=AfmBOopEAwpQwiGMYhAOjQygDXHhv9XQMVQr5p3OxvHeDeXDUK16wL-1

[21] Local Interconnect Network - Wikipedia
https://en.wikipedia.org/wiki/Local_Interconnect_Network

[22] Introduction to the LIN Bus Protocol - NI
https://www.ni.com/en/shop/seamlessly-connect-to-third-party-devices-and-supervisory-system/introduction-to-the-local-interconnect-network-lin-bus.html?srsltid=AfmBOop36zhMNH9OMZ2Iwm_uj_nyhKuX0C5fhCs0VWqaNCGKE0ONu5Fx

[23] Basic USB Concepts and FAQs | Microchip Technology
https://www.microchip.com/en-us/products/interface-and-connectivity/usb/basic-usb-concepts-and-faqs

[24] Media-independent interface - Wikipedia
https://en.wikipedia.org/wiki/Media-independent_interface

[25] LVDS (low-voltage differential signaling) - Semiconductor Engineering
https://semiengineering.com/knowledge_centers/communications-io/off-chip-communications/i-o-enabling-technology/lvds-low-voltage-differential-signaling/

[26] [27] [28] AMBA Bus Protocols - AXI, AHB, APB - Understanding Architecture and References
http://verificationexcellence.in/amba-bus-architecture/

[29] 1. Introduction to the Avalon® Interface Specifications
https://www.intel.com/content/www/us/en/docs/programmable/683091/20-1/introduction-to-the-interface-specifications.html

[30] WISHBONE System-on-Chip (SoC) Interconnection Architecture for Portable IP Cores — WISHBONE B3
https://wishbone-interconnect.readthedocs.io/en/latest/

[31] FPGA IP for PCIe
https://www.intel.com/content/www/us/en/products/details/fpga/intellectual-property/interface-protocols/pci-express-protocol.html

[32] fpga4fun.com - HDMI
https://www.fpga4fun.com/HDMI.html

[33] Camera Serial Interface 2 (MIPI CSI-2) | MIPI
https://www.mipi.org/specifications/csi-2

[34] MIPI Display Serial Interface (MIPI DSI) | MIPI
https://www.mipi.org/specifications/dsi