element14 Essentials: FPGA I

element14 Learning Center

element14 ESSENTIALS

FPGA I: Getting Started with FPGAs

Sponsored by XILINX ALL PROGRAMMABLE.

1. Introduction

Field Programmable Gate Arrays (FPGAs) are considered an ideal platform for implementing complex digital systems in application areas as varied as aerospace, food & beverage processing, industrial automation, automotive, biomedicine, defense, logistics, robotics, and many more. This learning module introduces the types, concepts, architecture, and some examples of FPGAs.

2. Objectives

Upon completion of this module, you will be able to:

- Define what an FPGA is
- Describe hardware description languages (HDL) such as Verilog and VHDL
- Explain the difference between an FPGA and other digital systems such as PLA, SPLD, CPLD, and Programmable SoCs
- Discuss the characteristics and benefits of different types of FPGAs
- Understand how FPGAs are used in various applications

3. Overview

An FPGA is a reprogrammable logic device that implements multi-level logic. It can be considered a generic platform for the implementation of a digital system. One of the most important benefits of an FPGA is its reconfigurability, which gives a digital designer the ability to tailor an FPGA for a specific application. As the cost of a standard FPGA chip has become affordable recently, the popularity of evaluation boards (such as the Digilent Basys3 and Arty boards) using cost-optimized chips has increased. Hence, a maker, student, DIY project builder, or prototyping engineer can realize his or her design on such a platform quite easily.

- 3.1 Types

The FPGA belongs to the general class of programmable logic devices (PLD). The aim of this class of devices is eliminating the need to use discrete logic gate chips in digital system implementation. Besides FPGAs, there are other PLD members, including:

- Programmable Logic Array (PLA) or Programmable Array Logic (PAL)
- Small/simple programmable logic device (SPLD)
- Complex programmable logic device (CPLD)
- Programmable System on Chip (SoC)

PLAs or PALs are more suitable for implementing specific logic function representations such as *minterm* and *maxterm*. SPLD can be taken as the initial and more limited version of CPLD. They are most often used in implementing canonical logic functions. Recent CPLD structures share similar properties with FPGAs in implementing logic functions. FPGAs can be used for more general digital system implementations ranging from logic functions to soft core microprocessors. The next level of PLD is the Programmable SoC, which contains both programmable logic and hard core processors (such as the ARM Cortex A9) on a single chip.

- 3.2 Comparison

We can categorize digital system design and implementation resources into four groups: discrete element, application specific integrated circuit (ASIC), FPGA (as a PLD representative), and microcontroller-based. The standard question arises: when should we use an FPGA instead of other design options? Or, what are the advantages of using an FPGA over other design options? Let's try to answer this question by comparing the FPGA with other design options.

Discrete Elements: A digital system can be implemented using discrete elements, which has been a design strategy for a long time. The advantage here is that the designer uses only the logic gates or discrete elements he/she needs to build the system. Moreover, using discrete elements does not require any expertise beyond basic logic knowledge. On the other hand, using discrete elements in logic design is not feasible in most cases. First, the physical space needed to implement them may be limited. Second, wire connections between discrete elements may become prohibitive in implementation. Third, the design will be static once implemented. An FPGA provides a neat solution to these problems.

The size of an FPGA chip is fixed and independent of the logic elements inside it. Moreover, interconnection of these elements is implicit in the FPGA. Therefore, the wiring of logic elements is not an issue. The most important advantage of FPGAs comes when the design needs to be reconfigured. Using an FPGA simplifies life for the designer because the design can be reconfigured by simply altering the corresponding HDL section. The only issue here is the need of expertise in HDL.

Application Specific Integrated Circuit (ASIC): ASICs provide a good alternative to discrete implementation. They overcome the space and wiring problems. When mass produced, an ASIC chip becomes less expensive. Moreover, the ASIC chip will be specific to the design; therefore, it will only use the required number of digital logic elements. It is worth noting that an FPGA chip can also be used as an ASIC. Hence, we specifically call a digital circuit an ASIC when it is designed for a specific purpose. Therefore, once it is designed, the topology will be fixed, which is a drawback of ASIC design. The biggest problem in using an ASIC is its fabrication time; FPGAs provide a clear advantage here. In fact, most ASIC designs are prototyped and verified on an FPGA before being mass produced.

Microcontroller: A microcontroller can be used instead of an FPGA in most cases. They share similar characteristics, such as reconfigurability, compactness, and cost-optimization. The first difference between them is that a microcontroller has a unique set of commands (instruction set) to perform an action. Therefore, the user will need to adjust his or her design accordingly. This is not an issue to an FPGA user. An FPGA can be considered as a free design environment within limits. The FPGA is more flexible than a microcontroller. The second difference between a microcontroller and an FPGA is power consumption, in which an FPGA has a clear advantage. The third difference between the microcontroller and an FPGA is in the inherent parallel implementation capacity of an FPGA. A microcontroller is a sequential device such that commands are performed step by step. However, an FPGA can be reconfigured as a parallel device. Hence, desired operations can be performed faster in orders of magnitude in an FPGA.

Programmable System on Chip (SoC): Recently, programmable SoC structures have become popular. These systems incorporate hard core processors and programmable logic on a single chip. Hence, they benefit from the desirable properties of both the FPGA and microcontroller. As an example, Xilinx offers an SoC platform, the Zynq®-7000 All Programmable SoC family, which has ARM dual core Cortex A9 processors along with a Xilinx 7 Series FPGA fabric. This SoC family deserves special consideration since it can be used in advanced applications.

4. Architecture

The architecture of an FPGA should be known by the reader to appreciate its working principles. Although the reader will not directly interact with the architecture, knowledge of it will lead to better usage of the FPGA. Plus, the design principles to be applied in implementing a digital system on an FPGA will make more sense with knowledge of its architecture.

- 4.1 Artix-7 XC7A35T FPGA

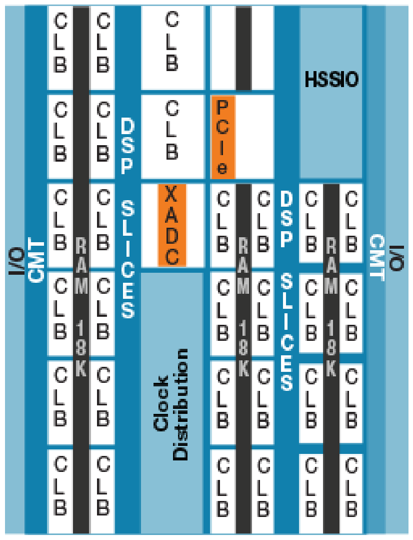


Figure 1: The Xilinx Artix-7 XC7A35T FPGA building blocks

The Xilinx Artix-7 XC7A35T FPGA is basically composed of nine different components: input/output (I/O) blocks, configurable logic blocks (CLB), interconnect resources, block RAM, DSP slices, clock management (clock distribution) block, analog-to-digital converter (XADC) block, high speed serial I/O (HSSIO) transceivers, and a PCIe interface. The layout of these blocks is shown in Figure 1.

Input/output (I/O) blocks: The architecture of an FPGA offers the flexibility of using each I/O pin as input, output, or input/output. I/O blocks are configured by the software (such as Vivado® Design Suite, to be covered in section 5.2) depending on the synthesized Hardware Description Language (HDL) code.

Configurable Logic Block (CLB): The basic building element of an FPGA. Complex logic functions are implemented in the FPGA by using CLBs and programmable interconnects. A CLB is composed of three components: look-up tables (LUTs), flip-flops, and multiplexers. LUTs, flip-flops, and multiplexers are grouped as slices in the CLB.
A look-up table or LUT can be thought of as a collection of flip-flops connected to input pins of a multiplexer. Select pins of the multiplexer will be taken as address bits of the flip-flop to be reached. This architecture can be used to implement any combinational logic function, which has a total number of variables as select pins. The important point here is that, as the entry of flip-flops change, the implemented logic function will also change. This will lead to reconfigurability of the FPGA.
In the Xilinx Artix-7 FPGA, each slice has four six-input LUTs, eight flip-flops, multiplexers, and other support circuitry. There are two slice types: SLICEM and SLICEL. Both can be used to implement combinational logic functions. SLICEM can also be used as a distributed memory element. As an example, the Xilinx Artix-7 XC7A35T FPGA has a total of 5200 slices, of which 3600 are SLICEL and 1600 are SLICEM. Each SLICEM can also be used as a 32-bit shift register.

Interconnect resources: A collection of wires and programmable switches. These are responsible for connecting CLBs and other building blocks within the FPGA. These are also called routing channels.

Block RAM: Recent FPGAs have block RAM modules within them. These can be used to store data. Moreover, they can form buffers, large LUTs, or shift registers. Usage of these block RAMs will become mandatory when soft core microcontrollers are considered.

For example, the block RAM in the Artix-7 XC7A35T FPGA can be used to store one block of 36 Kb or two blocks of 18 Kb data. There are 50 such blocks within the FPGA. Therefore, the total block RAM capacity of the FPGA is 50×36 = 1800 Kb. Each 36 Kbit block RAM can have a 64-bit data width. Moreover, the extra eight bits can be used for single-bit error correction or double-bit error detection during the data read process.

DSP slices: Digital signal processing (DSP) slices are dedicated blocks for arithmetic and logic operations in recent FPGAs. Each DSP slice can perform several arithmetic and logic operations, such as multiplying two binary numbers of length 25 and 18 bits; adding, subtracting, and accumulating two 48-bit numbers; or applying logic operations on two 48-bit numbers. These operations would require complex algorithms for implementation when a DSP slice was not available. Therefore, DSP slices are very effective in implementing complex arithmetic and logic operations. DSP slices are specifically referred to as DSP48E1 in the Artix-7 FPGA, which has a total of 90 such slices.

Clock management block: The clock is a periodic square wave signal such that it stays at logic level zero and one for certain time intervals. Most digital systems need a clock signal to operate in synchronous manner. In such a setting, logic operations are done at the rising edge (from logic zero to one transition) or the falling edge (from logic one to zero transition) of the clock signal. Hence, the period of the clock signal indicates the operational speed of the digital system. The Artix-7 FPGA does not have an internal clock generating circuitry. Therefore, the user must feed a clock signal to the FPGA. Some input/output pins are capable of receiving such clock signals, along with a PLL clock synthesis block. As the clock signal is fed to the FPGA, it can be processed by the clock management tile (CMT) and distributed through the FPGA. The Basys3 board has an external clock source to be used by the Artix-7 FPGA.

The XADC block: An analog signal can be processed by a digital system after being sampled and quantized. Modules performing these operations are called analog-to-digital converters (ADC). Since recent advances in digital systems require the processing of analog signals, the Artix-7 FPGA has a dedicated ADC block called XADC.
The Artix-7 XC7A35T FPGA has one XADC block which consists of two ADC modules. Each module can acquire one million samples per second (MSPS). Each sample can be represented by 12 bits. Therefore, a sample can be represented by a binary number in the range 0 to $2^{12} - 1$. The two ADC modules in the XADC block can process two different analog signals simultaneously.

High Speed Serial I/O Transceivers (HSSIO): Specialized circuitry that transfers and receives serial data. These transceivers are necessary to transfer data at speeds around gigabits per second (Gb/s).

Peripheral Component Interconnect Express Interface: Peripheral component interconnect express (PCIe) is a high-speed serial connection bus standard. The Artix-7 XC7A35T FPGA has one integrated block for PCIe interfacing.
As mentioned previously, recent FPGAs are armed with many useful features. In addition to the conventional specifications like the maximum number of differential and single-ended I/O pins and the number of logic cells and flip-flops, recent FPGA chips have extra blocks such as block RAM, DSP slices, ADC blocks, transceivers, and PCIe connections. The designer has to choose the optimum FPGA type for his or her application by considering the size of the design, I/O pins used, and additional blocks.

- 4.2 FPGA Packaging

Recent FPGA chips typically have more than 300 pins, which makes them impossible to use in through-hole packages for housing. Surface-mount smaller pitch packages are more suitable for FPGAs. Two of the most common packages are called Quad Flat Package (QFP) and Plastic and Ball Grid Array (BGA). QFPs may be divided into two groups: thin QFP (TQFP) and plastic QFP (PQFP). TQFPs offer shorter height for cases where a thin end product or portability is desired. The typical thickness of TQFP changes from 1 mm to 1.4 mm where the pitch changes accordingly. PQFPs are also thinner than regular QFPs and their thickness ranges from 2.0 mm to 3.8 mm. The Artix-7 on the Digilent Basys3 evaluation board comes in a BGA package with a pitch of 0.5 mm and thickness of 1.28 mm. You can find the available packaging of the Xilinx 7-series FPGAs here and package datasheets here.

5. Programming

In order to program an FPGA, the digital logic function or circuit must first be described by a Hardware Description Language (HDL). In addition, an environment is necessary for checking the syntax errors of the description and preparing it for programming the FPGA. For Xilinx FPGAs, this environment is called Vivado Design Suite.

- 5.1 VHDL and Verilog

There are two popular Hardware Description Languages (HDLs) available to program an FPGA: VHDL and Verilog. VHDL was created in 1981 by the U.S. Department of Defense, while Verilog was invented by Gateway Design Automation in 1985. They became an IEEE Standard in 1987 and 1995, respectively. The designer is free to choose between these two HDLs to describe his or her hardware and program the FPGA. As a simple comparison, VHDL is a strongly typed language. On the other hand, Verilog is plain and can be considered a C-like language.

In VHDL, logic circuits are described in entities as a black box that has inputs, outputs, and inouts. Inside each entity, the circuit behavior is described under the architecture section. Designers have to define each data type carefully, because they have to be converted from one data-type to another when assigned. Synchronized events are defined under the "process" keyword, where the sensitivity list can contain clock signal or a pre-defined input(s) as well.

In Verilog, circuits are defined as modules having inputs, outputs, and inouts. Inside the module, the logic function to be implemented can be described with the "assign" keyword. As opposed to VHDL, Verilog can handle most data-type conversions automatically. However, this situation may lead to unwanted problems. Synchronized events are described under the "always" keyword, and the sensitivity list may have clock signals or pre-defined inputs.

- 5.2 Vivado Design Suite

The Vivado Design Suite is software designed by Xilinx for synthesis, place and route, and simulation and implementation of HDL designs. It also offers an IP library where users can find solutions for communication, memory and controllers, DSP and math, embedded processors, and more. It has a free-license version for those who want to start programming an FPGA. It also supports the Windows and Linux operating systems.

The overall FPGA programming process has three main steps: synthesizing, mapping and routing, and programming. The description provided by the user is synthesized by the software. Then, it is translated into an electrical and technology schematic. The electrical schematic is synthesized by using global logic circuit components, including multiplexers, adders, and encoders. On the other hand, the technology schematic is generated by using the available components of the target FPGA that are selected at the beginning of the project. Afterwards, depending on the technology schematic, the software starts to map the design blocks on the available resources of the target FPGA such as CLBs, DSP slices, and RAM blocks. Then, the routes between these blocks are formed by using programmable interconnects. At this step, the software also needs the pin definition of the design. This is a file showing which I/O pin of the design is associated with which I/O pin of the FPGA. The last step, programming, converts the whole design to a file that can program the FPGA over a cable. The outcome can also be selected as a flash memory compatible file, which can be loaded into a memory device that can reprogram the FPGA after every reboot.

The Vivado Design Suite communicates with FPGAs using the Test Access Port and Boundary-Scan Architecture, commonly referred to as JTAG. During programming, a .bit file is transferred from the host PC to the FPGA. There might be other programming options depending on the evaluation board or application. For example, the Digilent Basys3 evaluation board offers programming through SPI Flash, USB port on the board, or USB-JTAG interface.

6. Applications

Although FPGAs can be used in a wide variety of application areas, we have included four representative ones below. These applications are taken from the book, Digital System Design with FPGA: Implementation using Verilog and VHDL. Detailed design specifications, explanations, as well as Verilog and VHDL descriptions for these applications are available in the book and its accompanying web site.

Digital Signal Processing via the XADC Block Usage

Digilent Artix-7 35T Basys3 Evaluation Kit

Processing an analog signal in a digital system requires analog-to-digital conversion (ADC) as the first step. The Artix®-7 FPGA has a specific XADC block for this purpose. This block is connected to the JXADC Pmod port of the Basys3 board. It is capable of converting four external differential signals to digital form since the port has four differential pins. Also, the XADC block has an internal temperature sensor which can be selected to read its output.

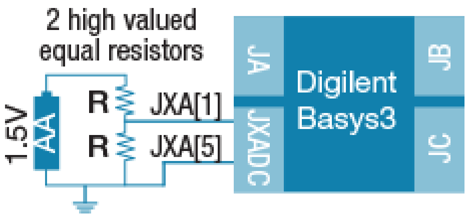


Figure 2: Schematic layout of the voltage level reading application via the XADC block

There are two applications on the usage of the XADC block in the book. The first application concerns reading the temperature value from the internal sensor on the FPGA chip. This application does not need any external wiring for the Basys3 board. The second application concerns measuring the voltage level of a battery connected to the ports of the Basys3 board. We provide the schematic layout of this application in Figure 2. These applications can be expanded to process any analog signal by the FPGA.

In addition to the Digilent Basys3 evaluation board, the parts needed for the second part of this application include One AA battery and Two high valued resistors (such as ≥ 510 kΩ). The Verilog and VHDL descriptions for this application can be obtained from the book web site.

Home Alarm System

We can design a basic home alarm system using sensors and an FPGA board (like the Basys3 or Arty boards). Let's first define the problem. Assume that the alarm system to be designed will be implemented on a house with three windows and a door. Each window and door has a proximity sensor working as follows: if someone steps in front of the sensor, it provides output of logic level 0. Otherwise, the output of the sensor is at logic level 1. Besides these sensors, we can also add a movement (PIR) sensor and sound detector to the home alarm system.

The output of the movement sensor is at logic level 0 when no movement is observed. If the sensor detects a movement, its output goes to logic level 1. If the sound detector detects a sound higher than its sensitivity value (threshold), then its output goes to logic level 0. Otherwise, its output stays at logic level 1. There should be an on/off switch for the alarm. If we want to activate the alarm, the switch will give logic level 1. Otherwise, it will give logic level 0. We can improve the home alarm system using a seven-segment display. When the system is active, the display will show the character 'A'. When it is deactivated, the display will show the character 'C'. To do so, we should add a seven-segment display decoder module to the system. This module converts the provided hexadecimal number to the corresponding seven-segment display pattern.

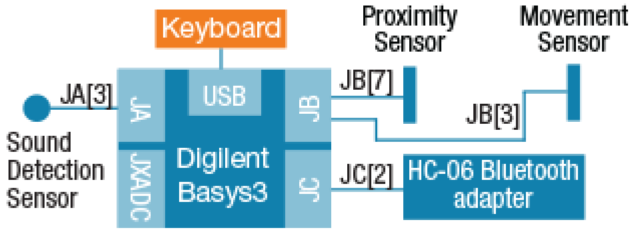


Figure 3: Schematic layout of the home alarm system

Using sequential circuits can improve the home alarm system. To do so, we can add password, buzzer, and LED blink modules to the system. The modified system works as follows: once the alarm is activated, the rightmost seven-segment display on the Basys3 board shows the character 'A'. This indicates that the system is active. If one of the windows is opened, then the alarm LED turns on to indicate that the alarm has turned on. If the user enters the correct password, then the alarm turns off. If the door is opened, then the user has 20 seconds to enter the correct password. If the correct password is entered within this time slot, then the alarm turns off. Otherwise, the alarm LED turns on and the buzzer starts working. "Counting 20 seconds" is displayed on the two seven-segment display digits. A Bluetooth transmitter module can also be added to the system such that when an intruder enters the house, a message can be sent to the router to communicate to the predefined police station phone number automatically. We provide the schematic layout of this application in Figure 3.

In addition to the Digilent Basys3 evaluation board, the parts needed for the home alarm application include an HC-06 bluetooth adapter, a passive buzzer module, a proximity sensor, a sound detector, and a movement sensor. The Verilog and VHDL descriptions of this application can be obtained from the book web site.

Car Park Occupied Slot Counting System
Our next real-life problem which can be solved by an FPGA is as follows: there is a car park with 16 slots and we would like to know how many of its slots are occupied at a given time. As in the home alarm system, for each slot we can put a proximity sensor to detect whether there is a car there or not. We can display the number of occupied slots on the two seven-segment displays of the Basys3 board. We can add a Bluetooth module such that the user can open the garage gate by using his or her cell phone. Here, a simple Android application developed under MIT App inventor will be sufficient. Also, we can add a proximity sensor to the garage gate. Hence, we can detect whether a car is passing through the gate. We can also add a stepper motor to open and close the garage gate. We provide the schematic layout of this application in Figure 4.

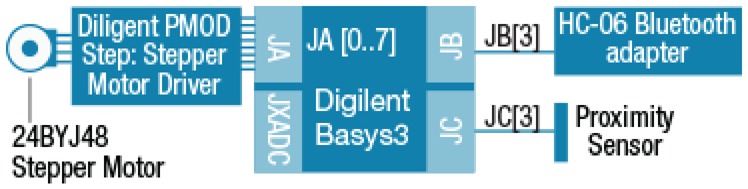


Figure 4: Schematic layout of the car park occupied slot counting system

In addition to the Digilent Basys3 evaluation board, the parts needed for car park occupied slot counting application include an HC-06 bluetooth adapter, Digilent PMOD Step: stepper motor driver, a 24BYJ48 stepper motor, and a proximity sensor. The Verilog and VHDL descriptions of this application can be obtained from the book web site.

Translator

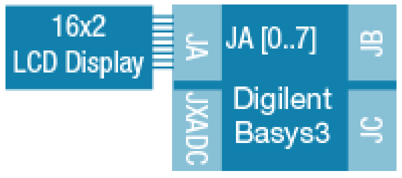


Figure 5: Schematic layout of the translator system

We can design a digital system to translate voice commands from English to Spanish (or another language) and show them on a 16x2 LCD. The system will have two parts. The first part will recognize the spelled English word. We can use the EasyVR shield for this purpose. This module has predefined speaker independent word sets. Also, the user can create his or her own speaker dependent word set. The LCD we used in our system is a WH1602N, featuring a built-in ST7066 LCD controller (or you can use an equivalent). The EasyVR shield communicates through a UART interface. Hence, we can use the UART transmitter and receiver blocks introduced in the book. After activating the EasyVR shield, it recognizes words in its predefined word set as default. This word set includes English words 'Action', 'Move', 'Turn', 'Run', 'Look', 'Attack', 'Stop', and 'Hello'. The Spanish translations of these words are 'Accion', 'Movimiento', 'Giro', 'Correr', 'Mirar', 'Ataque', 'Detener', and 'Hola', respectively. In the second part of the translator system, we can get the recognition result and form a state machine in the FPGA to provide the translated word corresponding to the recognized one. Then, this word is shown on the LCD. We provide the schematic layout of this application in Figure 5.

In addition to the Digilent Basys3 evaluation board, the parts needed for the translator application include a 16x2 LCD Display and the VeeaR Voice Recognition (VR) shield. The Verilog and VHDL descriptions to use the LCD display as well as an LCD driver module can be obtained from the book web site.

7. Product Examples

As indicated in the previous application examples, the application will influence the use of an FPGA. There are a variety of FPGAs available to the reader from which to choose for FPGA projects or designs. What follows is a summary of some Xilinx FPGAs that can be used with such projects.

- 7.1 FPGAs

Spartan®-6 FPGA for I/O optimization: Spartan®-6 devices offer connectivity features such as high logic-to-pin ratios, small form-factor packaging, MicroBlaze™ soft processor, 800Mb/s DDR3 support, and a diverse number of supported I/O protocols. Built on 45nm technology, the devices are suitable for advanced bridging applications found in automotive infotainment, consumer, and industrial automation. For more information, click here.

Spartan®-7 FPGAs for I/O optimization with the highest performance-per-watt: Spartan®-7 devices offer the highest performance-per-watt, along with small form factor packaging. These devices feature a MicroBlaze soft processor running over 200 DMIPs with 800Mb/s DDR3 support built on 28nm technology. Spartan-7 devices also offer an integrated ADC, dedicated security features, and Q-grade (-40 to +125°C) on all commercial devices. These devices are targeted for industrial, consumer, and automotive applications including any-to-any connectivity, sensor fusion, and embedded vision. For more information, click here.

Artix®-7 FPGAs for transceiver optimization and highest DSP bandwidth: Artix®-7 devices provide the highest performance-per-watt fabric, transceiver line rates, DSP processing, and AMS integration. Featuring the MicroBlaze soft processor and 1,066Mb/s DDR3 support, the family is suited to a variety of cost and power-sensitive applications, including software-defined radio, machine vision cameras, and low-end wireless backhaul. For more information, click here.

- 7.2 Programmable SoCs and MPSoCs

Zynq®-7000 All Programmable SoCs for system optimization with scalable processor integration: The Zynq®-7000 All Programmable SoC (AP SoC) family integrates the software programmability of an ARM®-based processor with the hardware programmability of an FPGA, enabling key analytics and hardware acceleration while integrating CPU, DSP, ASSP, and mixed signal functionality on a single device. Consisting of single-core Zynq-7000S and dual-core Zynq-7000 devices, the Zynq-7000 family is a fully scalable SoC platform. For more information, click here.

Zynq UltraScale+™ MPSoCs heterogeneous multiprocessor SoC: Zynq UltraScale+ MPSoC devices provide 64-bit processor scalability while combining real-time control with soft and hard engines for graphics, video, waveform, and packet processing. Built on a common real-time processor and programmable logic equipped platform, three distinct variants include dual application processor (CG) devices, quad application processor and GPU (EG) devices, and video codec (EV) devices are available for creating applications such as 5G Wireless, next generation ADAS, and Industrial Internet-of-Things. For more information, click here.

8. Evaluation and Dev Tools

Development boards and evaluation kits are simple way to get started learning and experimenting with FPGA projects and designs. Below, we summarize some of the currently available FGPA and Programmable SoC development kits.

Avnet Artix-7 50T FPGA Evaluation Kit

Avnet Artix-7 50T FPGA Evaluation Kit: This is a customizable development kit for those looking for a flexible, low-power platform. It contains all the necessary functions and interfaces needed for an embedded processor system in a small footprint. This board is also for designers interested in exploring the MicroBlaze soft processor or Artix-7 FPGAs in general. Experienced FPGA users will find this board useful for general purpose prototyping and testing. Its peripherals and expansion interfaces make the kit well suited for a wide variety of applications, including the evaluation of Industrial Ethernet protocols to the integration of multiple sensors to running a Linux-based web server.

Digilent Artix-7 35T Arty FPGA Evaluation Kit

Digilent Artix-7 35T Arty FPGA Evaluation Kit: The Arty evaluation kit is great way to get started with embedded applications ranging from compute-intensive Linux-based systems to light-weight microcontroller applications. Designed around the Artix®-7 35T FPGA from Xilinx, it features the MicroBlaze™ Processor customizable for nearly any processor use case.

Digilent ARTY S7 Dev Board

Digilent ARTY S7 Dev Board: The Arty S7 board features the Xilinx Spartan-7 FPGA, offering size, performance, and cost-conscious design benefits. Putting this FPGA in the Arty form factor provides users with a wide variety of I/O and expansion options. Use the dual row Arduino® connectors to mount one of the hundreds of hardware compatible shields available, or use the Pmod ports with Digilent's pre-made Pmod IP blocks for a more streamlined design experience. It's compatible with the Vivado Design Suite. The Arty S7 was designed to be MicroBlaze ready and comes ready to use with Xilinx's WebPACK licensing.

Avnet Spartan-6 FPGA LX9 Microboard

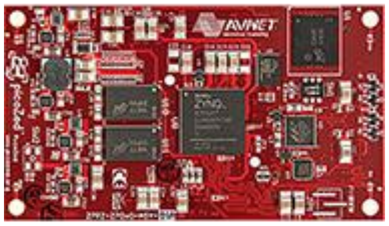
Avnet Spartan-6 Lx9 FPGA Microboard: The MicroBoard is for designers interested in exploring the MicroBlaze soft processor or Spartan-6 FPGAs in general. The kit comes with several pre-built MicroBlaze "systems" allowing users to start software development just like any standard off-the-shelf microprocessor. The Embedded Development Kit provides both an embedded hardware development tool (Xilinx Platform Studio - XPS) as well as an Eclipse-based environment for writing and debugging code (Software Development Kit - SDK). Experienced FPGA users will find the MicroBoard a useful tool for general purpose prototyping and testing.

Xilinx Zynq-7000 All Programmable SoC ZC702 Evaluation Kit

Xilinx Zynq-7000 All Programmable SoC ZC702 Evaluation Kit: This evaluation kit includes all the basic components of hardware, design tools, IP, and pre-verified reference designs, including a targeted design, enabling a complete embedded processing platform. The included pre-verified reference designs and FPGA Mezzanine Connectors (FMC) allow scaling and customization with daughter cards.

Avnet MicroZed™ Evaluation Kit

Avnet MicroZed™ Evaluation Kit: MicroZed™ is a development board based on the Xilinx Zynq®-7000 All Programmable SoC. It can be used as both a stand-alone evaluation board for basic SoC experimentation, or combined with a carrier card as an embeddable system-on-module (SOM). It contains two I/O headers that provide connections to two I/O banks on the programmable logic (PL) side of the Zynq-7000 AP SoC device. In stand-alone mode, these 100 PL I/O are inactive. When plugged into a carrier card, the I/Os are accessible in a manner defined by the carrier card design. Designers can start with MicroZed in stand-alone mode as a learning platform and then easily expand its functionality as a SOM through the addition of an off-the-shelf or custom designed carrier card.



Avnet PicoZed 7010 SOM - Zynq SoC based

Avnet PicoZed is a flexible System-On-Module (SOM), that is based on the Xilinx Zynq®-7000 All Programmable (AP) SoC. It offers designers the flexibility to migrate between the 7010, 7015, 7020, and 7030 Zynq-7000 AP SoC devices in a pin-compatible footprint. It contains the common functions required to support the core of most SoC designs, including memory, configuration, Ethernet, USB, and clocks. It provides easy access to over 100 user I/O pins through three I/O connectors on the backside of the module. These connectors also support access to dedicated interfaces for Ethernet, USB, JTAG, power and other control signals, as well as the GTP/GTX transceivers on the 7015/7030 models. The transceiver based 7015 and 7030 versions of PicoZed are a superset of the 7010/7020 version, adding four high-speed serial transceiver ports to the I/O connectors. Designers can simply design their own carrier card, plug-in PicoZed, and start their application development with the Zynq-7000 All Programmable SoC sub-system.

Avnet MiniZed SoC Development Board

Avnet MiniZed SoC Development Board: MiniZed™ is a single-core Zynq 7Z007S development board. This compact design features on-board connectivity through USB, Wi-Fi and Bluetooth. Peripherals can be plugged into dual Pmod-compatible connectors, the Arduino-compatible shield interface, or the USB 2.0 host interface. JTAG circuitry is incorporated onto the MiniZed, so with a single micro-USB cable to your laptop you are already up and running. User LEDs, a button, and a switch allow for a physical board interface. Micron memory solutions are presented for QSPI flash, DDR3L memory and on-board eMMC instead of an external SD card. The Murata Type 1DX wireless solution incorporates 802.11b/g/n Wi-Fi as well as Bluetooth 4.1, which provides both Bluetooth Classic and Low Energy (BLE). The integrated power supply generates all on-board voltages, while an auxiliary supply input can be used to power designs that require additional current. There is an on-board motion and temperature sensor, as well as a digital microphone.

This learning module is composed of material from Digital System Design with FPGA: Implementation using Verilog and VHDL by C. Ünsalan and B. Tar (McGraw-Hill, ISBN: 978-1259837906, 2017). Any mention of “the book” or “the book website” in this course is referring this title or its or its website.

Test Your Knowledge

Are you ready to demonstrate your FPGA Essentials knowledge? **Then take a quick 15-question multiple choice quiz to see how much you've learned from this Essentials of FPGA 1 Module.**

Take Quiz