SIDDAGANGA INSTITUTE OF TECHNOLOGY, TUMKURU

MICROCONTROLLER AND EMEBEDED SYSTEM

Report on

FPGA -MICROCONTROLLER

Submitted by

A N SWATHI	1SI22CS001	ADITHYA D V	1SI22CS010
ABHILASH H T	1SI22CS003	ADYA JHA	1SI22CS011
ABHISHEK A	1SI22CS004	AKRITI KUMARI	1SI22CS012
ABHISHEK GOWDA BS	1SI22CS005	AKSHATHA R GOWDA	1SI22CS013
ABHISHEK TG	1SI22CS006	AMAN KUMAR	1SI22CS014
ADARSH RAJ	1SI22CS007	AMAN KUMAR CHAUHAN	1SI22CS015
ADITI PRIYA	1SI22CS008	AMOGH S	1SI22CS016
ADITYA AGARWAL	1SI22CS009		

Submmited to

MANJUSHREE MAM

Siddaganga Institute of Technology, Tumkur

(An Autonomous Institute, Affiliated to Visvesvaraya Technological University Belagavi,

Approved by AICTE, New Delhi, Accredited by NAAC and ISO 9001:2015 certtified)

2023-2024

TABLE OF CONTENT

FPGA -Field Programmable Gate Arrays

INTRODUCTION

Field Programmable Gate Array (FPGA) is an integrated circuit that consists of internal hardware blocks with user-programmable interconnects to customize operation for a specific application. The interconnects can readily be reprogrammed, allowing an FPGA to accommodate changes to a design or even support a new application during the lifetime of the part.

The FPGA has its roots in earlier devices such as programmable read-only memories (PROMs) and programmable logic devices (PLDs). These devices could be programmed either at the factory or in the field, but they used fuse technology (hence, the expression "burning a PROM") and could not be changed once programmed. In contrast, FPGA stores its configuration information in a re-programmable medium such as static RAM (SRAM) or flash memory. FPGA manufacturers include **Intel**, **Lattice Semiconductor**, **Microchip Technology** and **Microsemi**.

PIN DETAILS

FPGAs tend to have lots of pins... So to make it a little simpler, let's put them into two bins: "user pins" and "dedicated pins".

User pins

The user pins are called "IOs", or "I/Os", or "user I/Os", or "user IOs", or "IO pins", or ... you get the idea.
IO stands for "input-output".

- You usually have total control over user IOs. They can be programmed to be inputs, outputs, or bi-directional (i.e. with tri-statable buffers).
- Each IO pin is connected to an "IO cell" inside the FPGA. The "IO cells" are powered by the VCCIO pins (IO power pins) - more details below.

Dedicated pins

The "dedicated pins" are hard-coded to a specific function. They fall into the three following sub-categories.

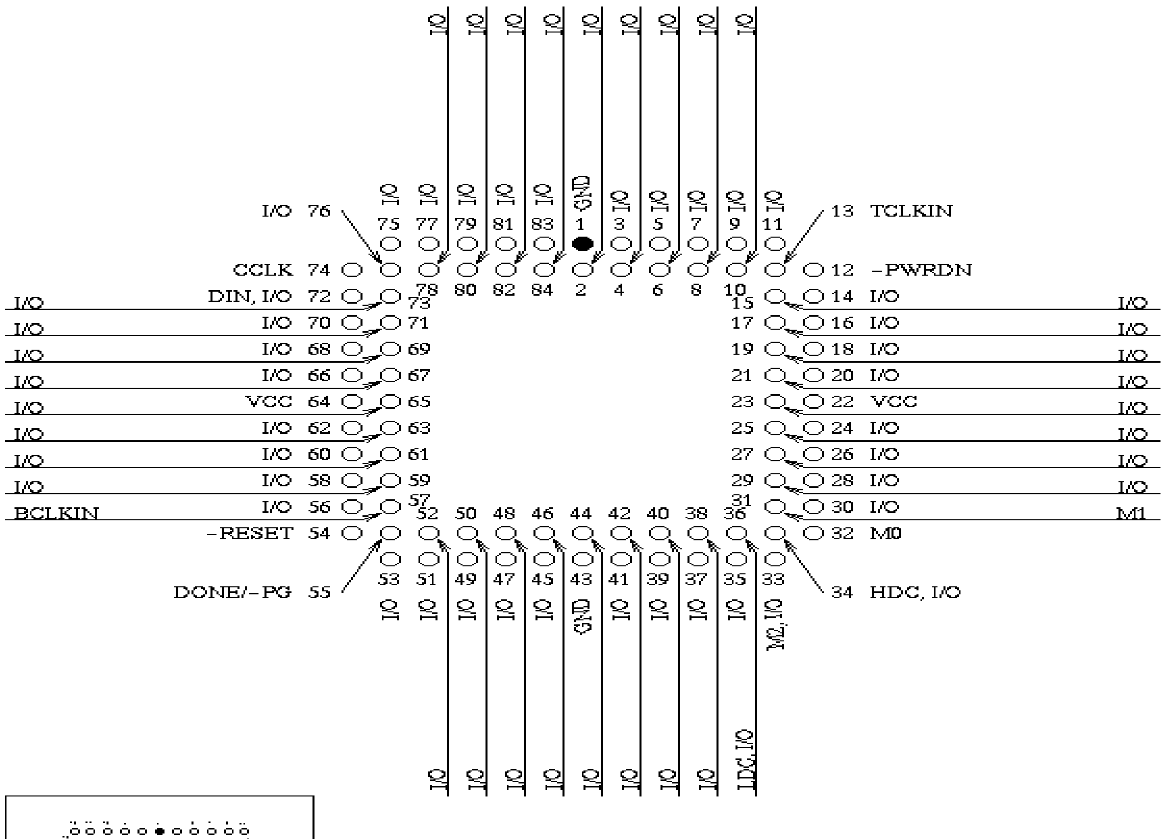
- Power pins.
- Configuration pins: used to "download" the FPGA.
- Dedicated inputs, or clock pins: these are able to drive large nets inside the FPGA, suitable for clocks or signals with large fan-outs.

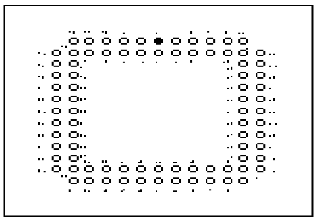
The power pins fall into two categories: "core voltage" and "IO voltage".

- The core voltage is named "VCC" for Xilinx and "VCCINT" for Altera. It is fixed (set by the model of FPGA that you are using). It is used to power the logic gates and flip-flops inside the FPGA. The voltage was 5V for older FPGA generations, and is coming down as new generations come (3.3V, 2.5V, 1.8V, 1.5V, 1.2V and even lower for the latest devices).
- The IO voltage is named "VCCO" for Xilinx and "VCCIO" for Altera. It is used to power the I/O blocks (= pins) of the FPGA. That voltage should match what the other devices connected to the FPGA expect.

An FPGA has many VCCIO pins that may be all powered by the same voltage. But new generations of FPGAs have a concept of "user IO banks": the IOs are split into groups, each having its own VCCIO pins. That allows using the FPGA as a voltage translator device, useful for example if one part of your board works with 3.3V logic, and another with 2.5V.

PIN DIAGRAM :





ARCHITECTURE OF FPGA

A basic FPGA architecture (***Figure 1***) consists of thousands of fundamental elements called configurable logic blocks (CLBs) surrounded by a system of programmable interconnects, called a fabric, that routes signals between CLBs. Input/output (I/O) blocks interface between the FPGA and external devices.

Depending on the manufacturer, the CLB may also be referred to as a logic block (LB), a logic element (LE) or a logic cell (LC).

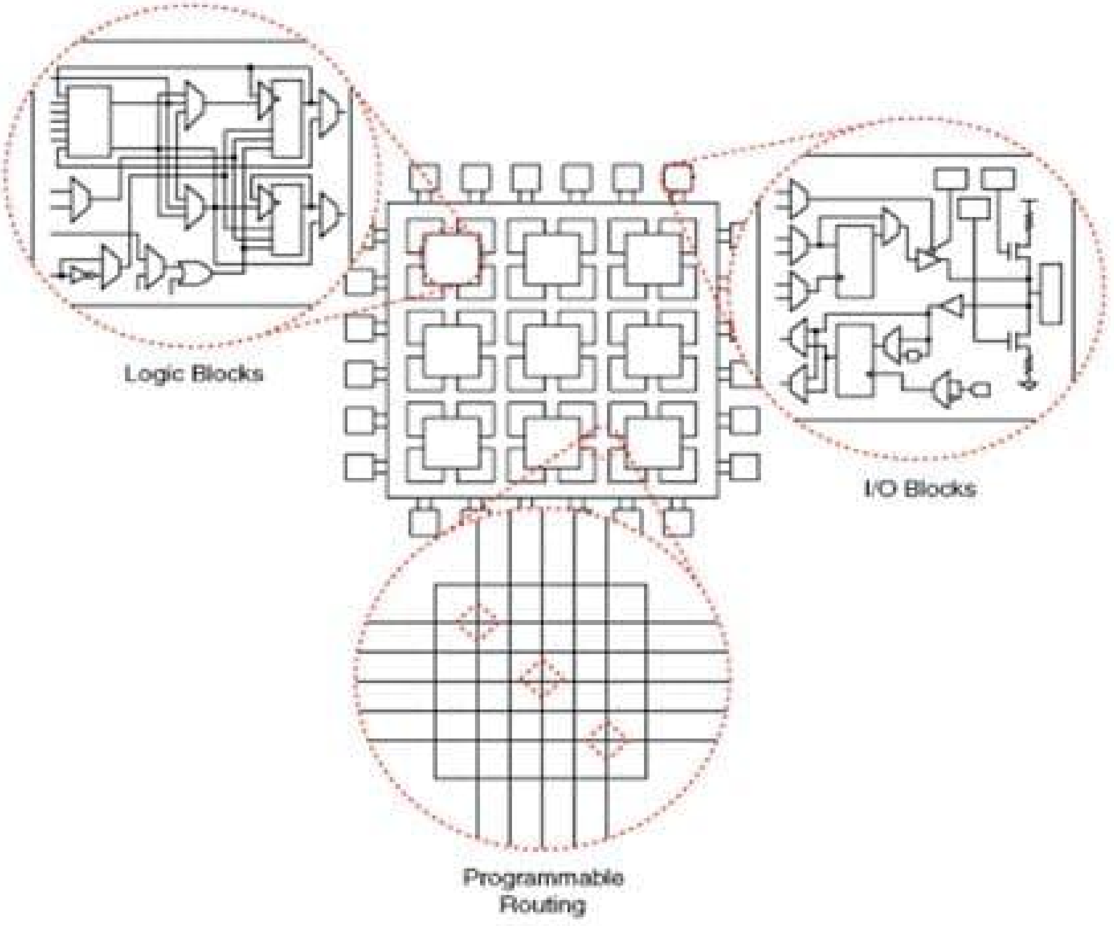


Figure 1: *The fundamental FPGA architecture (Image Source: National Instruments)*

An individual CLB (***Figure 2***) is made up of several logic blocks. A lookup table (LUT) is a characteristic feature of an FPGA. An LUT stores a predefined list of logic outputs for any combination of inputs: LUTs with four to six input bits are widely used. Standard logic functions such as multiplexers (mux), full adders (FAs) and flip-flops are also common.

.

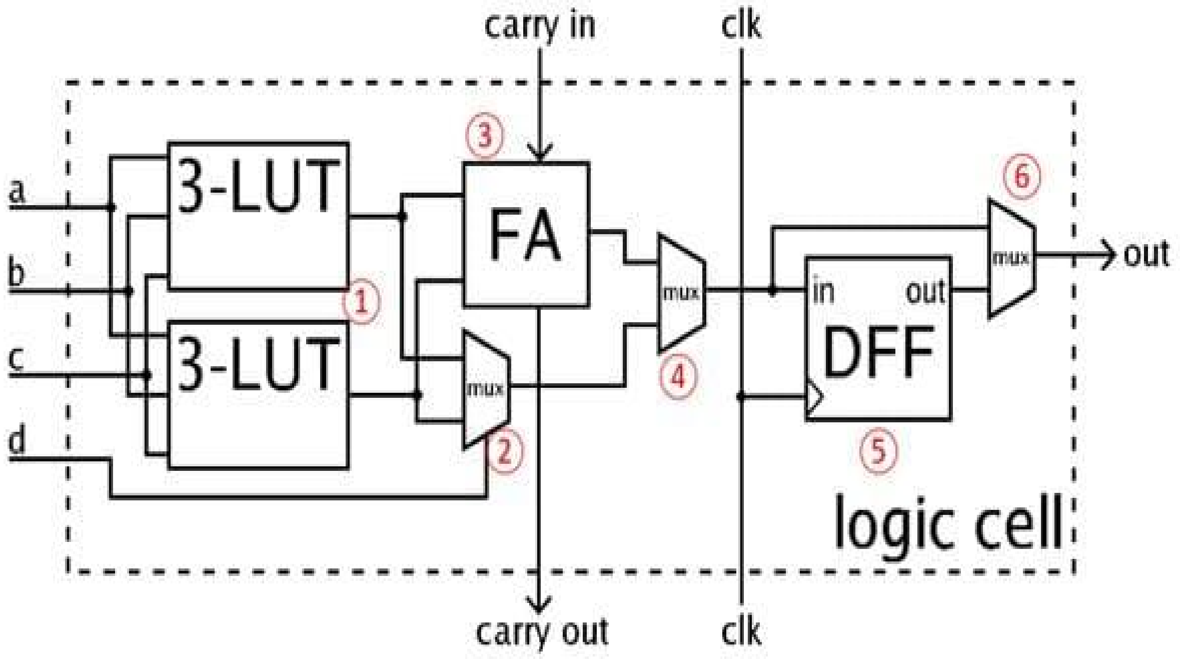


Figure 2: *A simplified CLB: The four-input LUT is formed from two three-input units. (Image source: Wikipedia)*

The number and arrangement of components in the CLB varies by device; the simplified example in Figure 2 contains two three-input LUTs (1), an FA (3) and a D-type flip-flop (5), plus a standard mux (2) and two muxes, (4) and (6), that are configured during FPGA programming.

This simplified CLB has two modes of operation. In normal mode, the LUTs are combined with Mux 2 to form a four-input LUT; in arithmetic mode, the LUT outputs are fed as inputs to the FA together with a carry input from another CLB. Mux 4 selects between the FA output or the LUT output. Mux 6 determines whether the operation is asynchronous or synchronized to the FPGA clock via the D flip-flop.

Current-generation FPGAs include more complex CLBs capable of multiple operations with a single block; CLBs can combine for more complex operations such as multipliers, registers, counters and even digital signal processing (DSP) functions.

INSTRUCTION SETS

Many popular and useful instruction sets are important and useful in computer science. These sets of instructions have their advantages as well as usages. Here are the types of instruction sets:

1. Reduced instruction set computer (RISC)

RISC has only a few cycles per instruction. It has a simpler form than a complex set of instructions. RISC is also used in many supercomputers. For example, it uses a summit, which is a supercomputer. It was the world's fastest supercomputer as per data in 2018.

2. Minimal instruction set computers (MISC)

A few codes and a set of instructions are basic for any processor. They also include sub-codes. As a result, they are smaller and faster. A disadvantage of MISC is that it has more sequential dependencies.

3. Complex instruction set computer (CISC)

CISC is a set of instructions with a few instructions per program. A CISC has fewer instructions than RISC.

4. Explicitly parallel instruction computing (EPIC)

This is an instruction set that permits microprocessors that help to execute instructions in parallel software. EPIC intends to give a simpler performance.

5. Very long instruction word (VLIW)

VLIW exploits parallelism at the instruction level. By this set of instructions, instructions are processed in sequence only in the CPU. This set of instructions improves the performance of the CPU.

6. Zero instruction set computer (ZISC)

The instructions that do not include microinstructions are known as ZISC. They are based on the pattern matching and can be compared to networks of synapses and neurons.

7. One instruction set computer (OISC)

The OISC set of instructions uses only one instruction for a machine language. This set of instructions is used to teach computer architecture and to compute structural computing research.

Stacks

A stack is a device used to store a set of instructions or information such that the item stored at last is the first item to pick. A stack is a memory unit in digital computers responsible for counting only. There are two types of stacks. These two types are register stack and memory stack.

Register stacks are built using registers. A memory stack forms a logical part of the memory kept under the stack. Stacks are used to evaluate as well as backtrack algorithms.

An advantage of stacks is that they are short and simpler instructions used to evaluate. However, they can not be accessed randomly. It is hard to access them, resulting in difficulty generating their code.

Accumulator

A kind of register included in any CPU is known as an accumulator. It serves as a location for temporary storage for intermediate values. These intermediate values are a part of calculations done in mathematical and logical calculations. The set of instructions is short in accumulators. It is just temporary storage, which makes traffic memory high for its approach.

General-purpose register

General-purpose registers are used as temporary data storage. They are stored temporarily in microprocessors. They are used to store 8-bit data and are represented by six figures. These figures are B, C, D, H, and L. They can also form 16-bit operations. This can be done by combining the pairs of general pair registers such as HL, DE, and BC. They are short instructions. A disadvantage of the general-purpose register is that all its operands should be named. This results in a longer set of instructions. They are easy to use with computer processors.

Real-World Applications of the Field Programmable Gate Array

Field Programmable Gate Array find applications across a wide array of industries due to their flexibility, speed, and reprogrammability. Here are some real-world use cases:

1. **Digital Signal Processing (DSP):** FPGA are widely used for high-performance DSP applications like image and video processing, audio processing, and wireless communication. Their parallel processing capabilities and reconfigurability make them ideal for real-time signal processing.
2. **Software-Defined Networking (SDN):**In the field of networking, FPGA are employed to accelerate data packet processing. They can be reconfigured to meet specific networking requirements, making them invaluable in SDN solutions and routers.

3. **Embedded Systems:**FPGA are used in embedded systems, especially in industries where real-time processing is critical, such as automotive, aerospace, and robotics. They can be customized for specific control and data processing tasks, enhancing system performance and flexibility.
4. **High-Performance Computing (HPC):** In HPC, FPGA are used for accelerating computationally intensive tasks, like scientific simulations and cryptography. Their ability to perform complex operations in parallel can significantly boost processing power.
5. **Cryptocurrency Mining:** Cryptocurrency miners utilize FPGA to optimize the mining process. These devices can be reconfigured to efficiently handle cryptographic hash functions, offering better energy efficiency compared to traditional GPUs.
6. **IoT and Edge Computing:** FPGA are increasingly being integrated into Internet of Things (IoT) devices and edge computing platforms. They enable efficient data preprocessing and analytics at the edge, reducing the need for sending vast amounts of raw data to the cloud.
7. **Aerospace and Defense:** In aerospace and defense, FPGA are used for radar and communication systems. Their ability to withstand harsh environmental conditions and adapt to changing requirements makes them invaluable in this sector.
8. **Scientific Research:** FPGA play a crucial role in scientific research, especially in fields like particle physics and astronomy. They are used for data acquisition and real-time analysis in large experiments

Code-Traffic light controller

```
module traffic_light_controller(
  input wire clk,        // Clock signal
  input wire reset,       // Reset signal
  output reg red,         // Red light
  output reg yellow,      // Yellow light
  output reg green        // Green light
);

  // Define state encoding
  typedef enum reg [1:0] {
    S_RED = 2'b00,      // State for Red light
    S_GREEN = 2'b01,    // State for Green light
    S_YELLOW = 2'b10     // State for Yellow light
} state_t;

state_t state, next_state;

// State transition and output logic
always @(posedge clk or posedge reset) begin
    if (reset) begin
        state <= S_RED;
    end else begin
        state <= next_state;
    end
end

// Next state logic
always @(*) begin
    case (state)
        S_RED: begin
            next_state = S_GREEN;
            red = 1;
            yellow = 0;
            green = 0;
        end
        S_GREEN: begin
            next_state = S_YELLOW;
            red = 0;
            yellow = 0;
            green = 1;
        end
        S_YELLOW: begin
            next_state = S_RED;
            red = 0;
            yellow = 1;
            green = 0;
        end
        default: begin
            next_state = S_RED;
            red = 1;
            yellow = 0;
            green = 0;
        end
    endcase
```

Features of FPGA

1. **Reconfigurability**: FPGAs can be reprogrammed to perform different tasks, making them highly versatile.

2. **Parallel Processing**: FPGAs can execute many operations simultaneously, providing significant performance advantages for parallelizable tasks.

3. **Customizable Logic**: Designers can implement custom digital circuits tailored to specific applications.

4. **Low Latency**: FPGAs can achieve very low latencies compared to software running on general-purpose processors.

5**. High Throughput:** Due to parallel processing and pipelining, FPGAs can handle high data throughput.

6. **Real-Time Processing**: FPGAs are well-suited for real-time processing tasks due to their deterministic nature.

7. **Hardware Acceleration**: They can accelerate computationally intensive algorithms, such as those in cryptography, signal processing, and machine learning.

Pros and Cons of FPGA

Pros:

1. **Flexibility**: Ability to reconfigure and adapt to different applications.

2**. Performance**: High-speed processing and low latency due to parallel execution.

3. **Customization**: Tailored hardware solutions can be designed for specific tasks.

4. **Scalability**: FPGAs can be scaled to meet the needs of various applications, from small embedded systems to large-scale data centers.

5. **Power Efficiency**: For certain applications, FPGAs can be more power-efficient than CPUs or GPUs.

Cons:

1. **Complexity**: Designing FPGA applications requires knowledge of hardware description languages and digital design principles.

2. **Cost**: Higher initial costs for development and often higher per-unit costs compared to microcontrollers or CPUs.

3. **Development Time**: Longer design cycles due to the need for hardware validation and testing.

4**. Toolchain**: Requires specialized tools for synthesis, place and route, and programming, which can be complex to use.

Differences from Other Types of Microcontrollers

1. **Reconfigurability vs. Fixed Functionality**: FPGAs can be reconfigured to perform different tasks, whereas traditional microcontrollers have fixed functionality defined by their architecture.

2. **Parallel Processing**: FPGAs excel at parallel processing, unlike most microcontrollers, which execute instructions sequentially.

3. **Customization**: FPGAs allow for the design of custom digital circuits, while microcontrollers have a fixed set of peripherals and functionalities.

4. **Performance**: For certain tasks, FPGAs can provide significantly higher performance due to their ability to process multiple data streams concurrently.

5. **Design Complexity**: FPGA design is more complex and requires specialized knowledge and tools, while microcontroller programming is generally simpler and more accessible.

Applications of FPGA

1. **Telecommunications**: Signal processing, baseband processing, and network infrastructure.

2. **Aerospace and Defense**: Radar systems, avionics, and secure communications.

3. **Automotive**: Advanced driver-assistance systems (ADAS), in-car entertainment, and vehicle-to-everything (V2X) communication.

4. **Industrial Automation**: Robotics, control systems, and machine vision.

5. **Consumer Electronics**: High-definition video processing, gaming, and smart home devices.

6. **Data Centers**: Hardware acceleration for machine learning, data compression, and encryption.

7. **Medical Devices**: Imaging systems, diagnostic equipment, and patient monitoring.

8. **Finance**: High-frequency trading and real-time data analysis.

Example: Real-Time Video Processing

FPGAs can be used in video processing applications such as live video streaming, video encoding/decoding, and real-time image enhancement. For instance, an FPGA can implement a video codec that compresses video data efficiently for transmission over the internet. The parallel processing capabilities of FPGAs enable them to handle high-resolution video streams in real-time, providing low-latency and high-throughput performance that is crucial for live broadcasting and video conferencing applications.

CONCLUSION

FPGA microcontrollers provide a versatile and powerful solution for a wide range of applications, thanks to their ability to be reconfigured to meet specific performance and functionality needs. This adaptability makes them suitable for industries requiring real-time processing and high-speed data handling, such as telecommunications, automotive, aerospace, and consumer electronics. The capability to tailor hardware to particular tasks ensures optimal performance and efficiency, distinguishing FPGAs from traditional microcontrollers.

Despite their advantages, FPGA microcontrollers come with challenges, primarily due to their complexity and the steep learning curve associated with their development. Proficiency in hardware description languages (HDLs) and electronic design automation (EDA) tools is essential, which can be a significant barrier for many developers. However, advancements in development environments and educational efforts are gradually lowering these barriers, making FPGAs more accessible to a broader range of engineers and developers.

In summary, FPGA microcontrollers offer a unique blend of flexibility and performance that is increasingly valuable in today's fast-paced technological landscape. While the complexity of their design and programming poses challenges, the benefits they bring in terms of customization and efficiency are substantial. As development tools continue to improve and knowledge of FPGA technology becomes more widespread, their adoption is likely to grow, driving innovation and enhanced performance in numerous fields.