CHAPTER 2

FPGA Technologies

Before we delve into utilizing FPGAs for accelerating robotic workloads, in this chapter we first provide the background of FPGA technologies so that readers without prior knowledge can grasp the basic understanding of what an FPGA is and how an FPGA works. We also introduce partial reconfiguration, a technique that exploits the flexibility of FPGAs and one that is extremely useful for various robotic workloads to time-share an FPGA so as to minimize energy consumption and resource utilization. In addition, we explore existing techniques that enable the robot operating system (ROS), an essential infrastructure for robotic computing, to run directly on FPGAs.

2.1 AN INTRODUCTION TO FPGA TECHNOLOGIES

In the 1980s, FPGAs emerged as a result of increasing integration in electronics. Before the use of FPGAs, glue-logic designs were based on individual boards with fixed components interconnected via a shared standard bus, which has various drawbacks, such as hindrance of high volume data processing and higher susceptibility to radiation-induced errors, in addition to inflexibility.

In detail, FPGAs are semiconductor devices that are based around a matrix of configurable logic blocks (CLBs) connected via programmable interconnects. FPGAs can be reprogrammed to desired application or functionality requirements after manufacturing. This feature distinguishes FPGAs from Application-Specific Integrated Circuits (ASICs), which are custom manufactured for specific design tasks.

Note that ASICs and FPGAs have different value propositions, and they must be carefully evaluated before choosing any one over the other. While FPGAs used to be selected for lower-speed/complexity/volume designs in the past, today's FPGAs easily push the 500 MHz performance barrier. With unprecedented logic density increases and a host of other features, such as embedded processors, DSP blocks, clocking, and high-speed serial at ever lower price points, FPGAs are a compelling proposition for almost any type of design.

Modern FPGAs are with massive reconfigurable logic and memory, which let engineers build dedicated hardware with superb power and performance efficiency. Especially, FPGAs are attracting attention from the robotic community and becoming an energy-efficient platform for robotic computing. Unlike ASIC counterparts, FPGA technology provides the flexibility of on-site programming and re-programming without going through re-fabrication with a modified design, due to its underlying reconfigurable fabrics.

S. Liu et al., *Robotic Computing on FPGAs*

2.1.1 TYPES OF FPGAS

FPGAs can be categorized by the type of their programmable interconnection switches: antifuse, SRAM, and Flash. Each of the three technologies comes with trade-offs.

- **Antifuse FPGAs** are non-volatile and have a minimal delay due to routing, resulting in a faster speed and lower power consumption. The drawback is evident as they have a relatively more complicated fabrication process and are only one-time programmable.

- **SRAM-based FPGAs** are field reprogrammable and use the standard fabrication process that foundries put in significant effort in optimizing, resulting in a faster rate of performance increase. However, based on SRAM, these FPGAs are volatile and may not hold configuration if a power glitch occurs. Also, they have more substantial routing delays, require more power, and have a higher susceptibility to bit errors. Note that SRAM-based FPGAs are the most popular compute substrates in space applications.

- **Flash-based FPGAs** are non-volatile and reprogrammable, and also have low power consumption and route delay. The major drawback is that runtime reconfiguration is not recommended for flash-based FPGAs due to the potentially destructive results if radiation effects occur during the reconfiguration process [52]. Also, the stability of stored charge on the floating gate is of concern: it is a function including factors such as operating temperature, the electric fields that might disturb the charge. As a result, flash-based FPGAs are not as frequently used in space missions [53].

2.1.2 FPGA ARCHITECTURE

In this subsection, we introduce the basic components in FPGA architecture in the hope of providing basic background knowledge to readers with limited prior knowledge on FPGA technologies. For a detailed and thorough explanation, interested authors can refer to [54].

As shown in Fig. 2.1, a basic FPGA design usually contains the following components.

- **Configurable Logic Blocks (CLBs)** are the basic repeating logic resources on an FPGA. When linked together by the programmable routing blocks, CLBs can execute complex logic functions, implement memory functions, and synchronize code on the FPGA. CLBs contain smaller components, including flip-flops (FFs), look-up tables (LUTs), and multiplexers (MUX). An FF is the smallest storage resource on the FPGA. Each FF in a CLB is a binary register used to save logic states between clock cycles on an FPGA circuit. An LUT stores a predefined list of outputs for every combination of inputs. LUTs provide a fast way to retrieve the output of a logic operation because possible results are stored and then referenced rather than calculated. A MUX is a circuit that selects between two or more inputs and then returns the selected input. Any logic can be implemented using the combination of FFs, LUTs, and MUX.