Embedded Security
SToESD1 FPGA - Accelaration

Lydie Stella Nguemo Sokeng 00098008

Selected Topics of
Embedded Software Development

Overview

1 What is a CPU ..... **Fehler! Textmarke nicht definiert.**

2 What is an FPGA? ..... **Fehler! Textmarke nicht definiert.**

3. What separate FPGA vs CPU? ..... **Fehler! Textmarke nicht definiert.**

4. Conclusion ..... **Fehler! Textmarke nicht definiert.**

5. References ..... **Fehler! Textmarke nicht definiert.**

1 What is a CPU?

CPUs or Central Processing Units have been around for decades. A CPU is a chip that sequentially executes a program based on a specified set of instructions. While CPUs are optimal for single-process systems that require code to be executed sequentially or linearly, they lack the implementation of parallelism.

The internal hardware structure of the CPU is defined by the CPU manufacturer and cannot be changed. CPUs are often general-purpose CPUs and so can perform any function based on the software program uploaded into them. As a general-purpose chip, they do not represent any particular hardware advances such as video compression that can be done quickly using dedicated hardware blocks.

CPUs come in a variety of sizes and prices, from an Intel CPU that powers your computer to a small CPU that runs in your computer mouse.

2 What is an FPGA?

An FPGA is a chip made up of a series of logic blocks that can be modified and configured by the user. As such, these chips give the user much more flexibility and adaptability in performing specific tasks that require timely results. FPGAs are ideal for parallel systems where multiple tasks must be performed at the same time because they are electronically wired in the form of discrete programmable logic blocks that can be configured according to the user's needs. FPGAs are programmable chips and their functionality can be updated multiple times.

FPGAs come in a variety of sizes and prices and are most likely used in low to medium size products. Due to their price, FPGAs are not suitable for mass products.

3 What Separate FPGA vs CPU?

➢ Flexibility

Both technologies offer engineers great flexibility. Although CPUs have a fixed set of instructions that must be followed by the programmer, the program itself can be modified several times from bug fixes to a complete change to the program. One of the main features and advantages of FPGAs is that all internal hardware can be reprogrammed and reconfigured as the user can control the logic of each block of the system. This makes them much more flexible in their programming and can be adapted to the needs of the programmer.

➢ Execution Speed

Nothing beats dedicated hardware designed for a single function. Therefore, a well-designed FPGA will always run faster than software code running on a general-purpose CPU chip.

➢ Portability of Designs

Since the FPGA code is written in VHDL or Verilog languages, it can be ported to other FPGA types with relative ease. CPU programs can also be easily ported to other CPUs if the user has used a high-level programming language such as C / C ++ or Java.

➢ Power Consumption

While FPGAs are more flexible, they use more power, which means you need more juice to keep them running, unlike CPUs, which aren't such a problem in terms of power consumption.

➢ Complexity

CPUs are usually more complex compared to FPGAs, mainly because they already have a fixed set of internal blocks and processes that are already developed by the manufacturer. An FPGA on the opposite side of the spectrum is like a blank sheet, it's configurable and modifiable, which means that the user can decide how complex the design should be. This makes them easier and less complex to understand and to modify if necessary.

- Time Critical Processing

CPUs are usually limited in this regard and are therefore not ideal for time-critical processing, especially if the requirements cannot be met within the scope of their possibilities. FPGAs are able to carry out complex and time-critical processing operations in parallel to other critical processing tasks.

- General Purpose Computing

CPUs offer the greatest versatility, making them best for general-purpose computing. FPGAs can be used to perform more specific and specialized tasks, but are not ideal for general computing purposes.

- Updates

If the development is finished and a change has to be made to the code of the CPU, it can be inserted into the CPU after compilation like a download and will probably work after a reset. This is also possible with FPGAs because you can upload a new code to the chip and from this point on the FPGA is configured with the new functionality.

4 Conclusion

CPUs are optimized for sequential processing. They do not have sufficient support for parallelism and are limited in their high-speed processing capabilities. FPGAs have their own set of difficulties, including programming, which is relatively more complex and difficult to learn. Unlike CPUs, FPGAs work well in parallel and tend to fall short in terms of price and power consumption.

5 References

- http://www.pynq.io/community.html
- https://www.xilinx.com/products/silicon-devices/fpga/what-is-an-fpga.html
- https://www.xilinx.com/support/documentation/application_notes/xapp1170-zynq-hls.pdf
- https://de.wikipedia.org/wiki/Field_Programmable_Gate_Array