# FPGA Programming

## Class 1: Intro – FPGA Device Description

September 11, 2017

Louis W. Giokas

DesignNews
CEC CONTINUING EDUCATION CENTER
Presented by:
Digi-Key ELECTRONICS
ROHDE&SCHWARZ

# This Week's Agenda

| | |
|---|---|
| Monday | FPGA Device Description |
| Tuesday | Design Flow |
| Wednesday | HDL |
| Thursday | Synthesis and Layout |
| Friday | Programming the Chip |

# Course Description

We start with an introduction to the class of devices called Field Programmable Gate Arrays (FPGAs). The layout and design of several types and critical parameters will be described and discussed. It is important to understand the way the device is constructed to develop effective algorithms.

The device we will be using this week will be the Microsemi IGLOO2. We will also discuss other devices and their structure.

We will introduce two common Hardware Description Languages (HDL), but give examples in one (Verilog).

# Today's Agenda

- A word on "Programming"
- Device Overview
- Basic Elements
- Vendors
- Conclusion/Next Class

# A word on “Programming”

- The word “Programming” is somewhat overloaded in the FPGA world
  - The official meaning is to apply the bitstream to the device that controls the connections and data pathways
  - Coming from a computer science background I tend to use the word programming to mean writing the software, using Hardware Description Language (HDL) that determines what the chip will do

# A word on “Programming”

- We will see programming used in both ways in this class
  - In the HDL session (Wednesday) we will use programming in the later sense exclusively
  - In the Programming the chip session (Friday) we will be talking about the former exclusively
- Vendor documentation uses the term in the former sense
- I will try to make it very clear in what sense I am using it

# Device Overview

- FPGA stands for *field-programmable gate array*
  - Can be programmed in the "field" (e.g., at your desk) as opposed to the factory
  - Internal structure is not really a gate array; this is a historical reference to a predecessor device
  - Less efficient in power, size and performance than full custom, but very flexible and much less costly
    - Sometimes ASIC designs are developed in FPGAs the converted to ASIC implementation

# Device Overview

- History
  - Grew out of the programmable ROM (PROM) and programmable logic device (PLD)
  - First vendor of FPGAs was Xilinx
    - First device in 1985
  - Followed by Altera which had started with EPROMs
    - Now Intel FPGA Solutions
  - Third major player is Actel
    - Now Microsemi
  - Together they have about 75% or more of the market

# Device Overview

- Modern FPGAs use elements/abstractions that are higher level than gates
  - Typical elements are look up tables, programmable logic blocks and memories
  - We develop algorithms using a HDL
    - These are then processed through several steps
      - Compilation
      - Layout
      - Program (bit pattern) generation and loading

# Device Overview

- FPGAs can be programmed in the field, even during operation
  - Called *in-application programming*
  - Systems with storage and a control processor can be loaded quickly to implement new functionality
- Some chips have “hard” processors built-in
  - Usually called System on Chip (SoC)
    - We will not address these
- “Soft” processors can be implemented on standard FPGAs

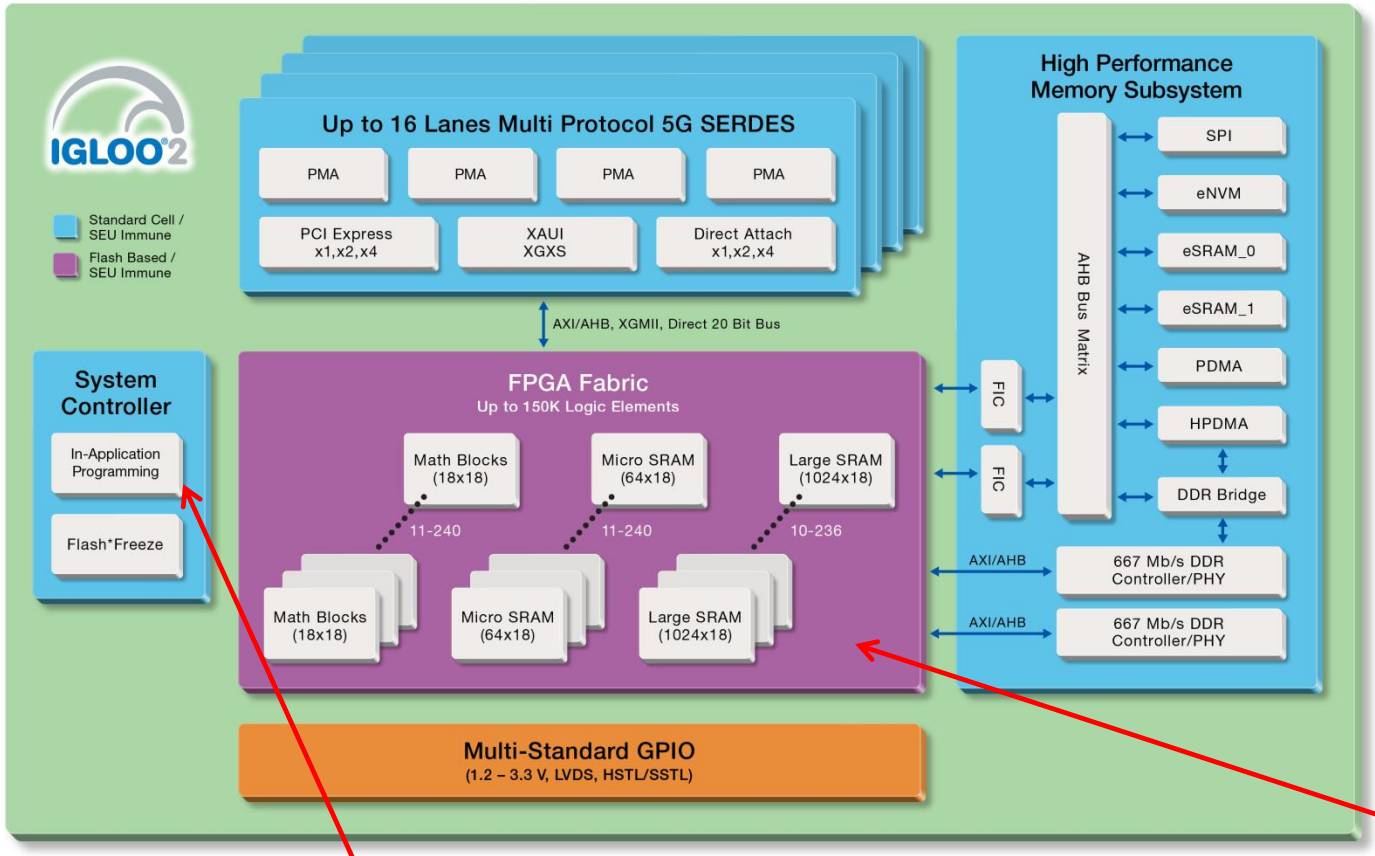
# Basic Elements

- Several types of basic elements will be present on each FPGA
  - The most common are the Configurable Logic Block (CLB) and the switch matrix
  - Math blocks are included for computation
  - Various types of memory are also included to store parameters, inputs and intermediate results
  - I/O elements are used to data into and out of the fabric

# Basic Elements

- Memory
  - Memories included are of various types
    - Volatile RAM
    - Non-volatile RAM
  - Interfaces to external RAM
    - These allow data to be ingested into the device memory and to be offloaded to external RAM or devices
  - Memory usually comes in two sizes, small and large

# Basic Elements

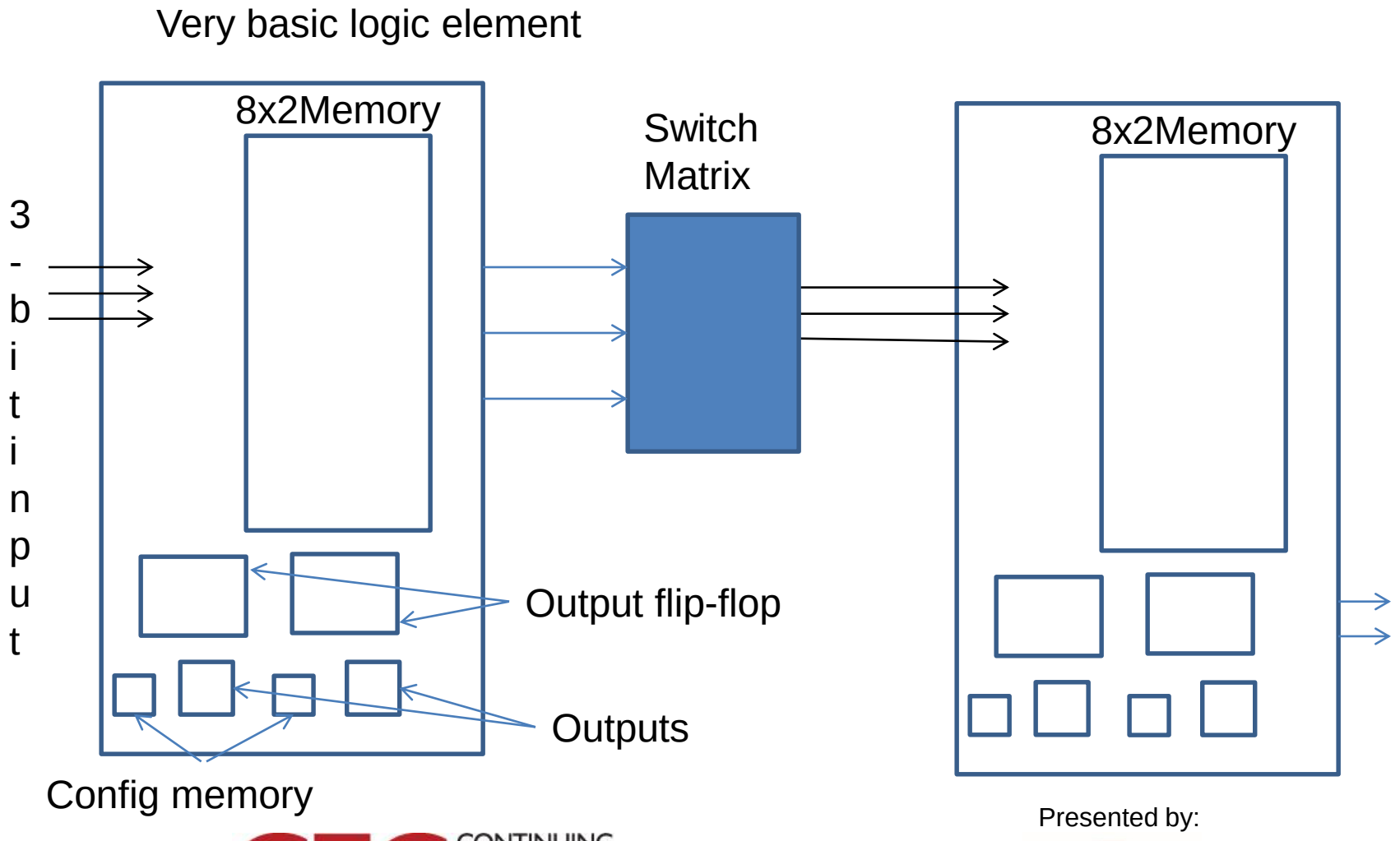



The Microsemi IGLOO2 is the sample device we will use

Not shown are the basic logic elements and the switch matrix

Note the In-application programming circuitry

# Basic Elements



# Basic Elements

- From these simple elements, any logic circuit can be synthesized
- Setting the contents of the loop-up table determines the logic function realized
  - This is essentially the truth table of a logic circuit
  - Configuration memory determines if the output comes from the flip-flop or the lookup table directly
- These values, along with the switch matrix setting are the “program” of the FPGA

# Basic Elements

- Other elements required to make the system work include:
  - On-chip oscillators
  - Phase Locked Loop (PLL)
  - Clock Conditioning Circuit (CCC)
- Math blocks are typically used for implementing DSP type operations (precision arithmetic)

# Vendors

- Intel FPGA (Altera)
  - One of the earliest vendors with probably the second largest market share
  - Devices include (from highest to lowest perf.):
    - STRATIX (includes SoC Versions)
    - ARRIA (includes SoC Versions)
    - MAX
    - CYCLONE (includes SoC Versions)

# Vendors

- Xilinx
  - One of the earliest vendors with probably the largest market share
  - Devices include:
    - SPARTAN (45nm & 28nm)
    - VIRTEX (28nm, 20nm & 16nm)
    - KINETX (28nm, 20nm & 16nm)
    - ARTIX (28nm)
  - ZYNQ is their SoC line combining a hard processor with some of the above FPGA fabrics

# Vendors

- Microsemi (Actel)
  - PolarFire (low cost device)
  - IGLOO2
  - RTG4 (radiation tolerant)
  - SmartFusion2 (SoC, combining a IGLOO2 with and ARM processor subsystem)
  - Target military, low power and high reliability applications
    - Typically not as large as the other vendors devices in terms of logic elements

# Conclusion/Next Class

- Today we introduced the topic and reviewed some of the terminology
- We reviewed the concept of a logic element and switch fabric
- We introduced the main vendors
- Tomorrow we will talk about the design flow necessary to “program” our FPGA