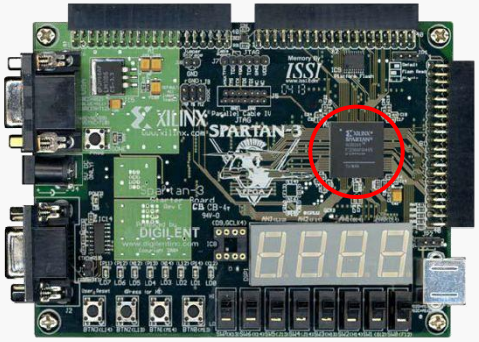
Performance Engineering of Real-Time and Embedded Systems

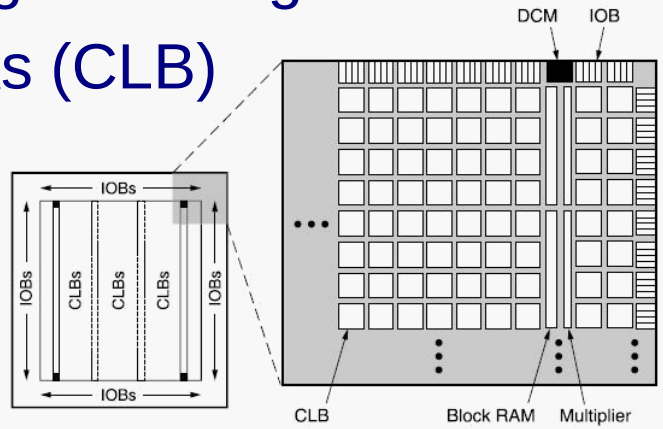


FPGA Devices

Field-programmable gate arrays bridge the gap between microprocessors and custom hardware.

- Hardware functions are programmable after manufacture.

Thousands of configurable logic blocks (CLB)



Lookup tables, registers, clock management; multipliers, counters

Spartan-3	XC 3S200
System Gates	200K
Logic Cells	4,320
18x18 Multipliers	12
Block RAM Bits	216K
Distributed RAM Bits	30K
DCMs	4
I/O Standards	24
Max Differential I/O Pairs†	76
Max Single Ended I/O†	173
Package	**User I/O**
VQ100	63
TQ144	97
PQ208	141
FT256	173

All FPGAs have some common characteristics.

- Small logic cells
- Lookup tables as input logic to a register
- Some memory resources
- Routing resources
  - ***Block to block***
  - ***Cross-chip long lines***
- Configurable input/output

The FPGA device must be configured.

- Configuration may be persistent in the device or loaded at start-up.
- Persistent device configuration may be modifiable or permanently configured.
- Latest frontier is dynamically reconfigurable computing elements.

Tools take the tediousness out of generating a configuration.

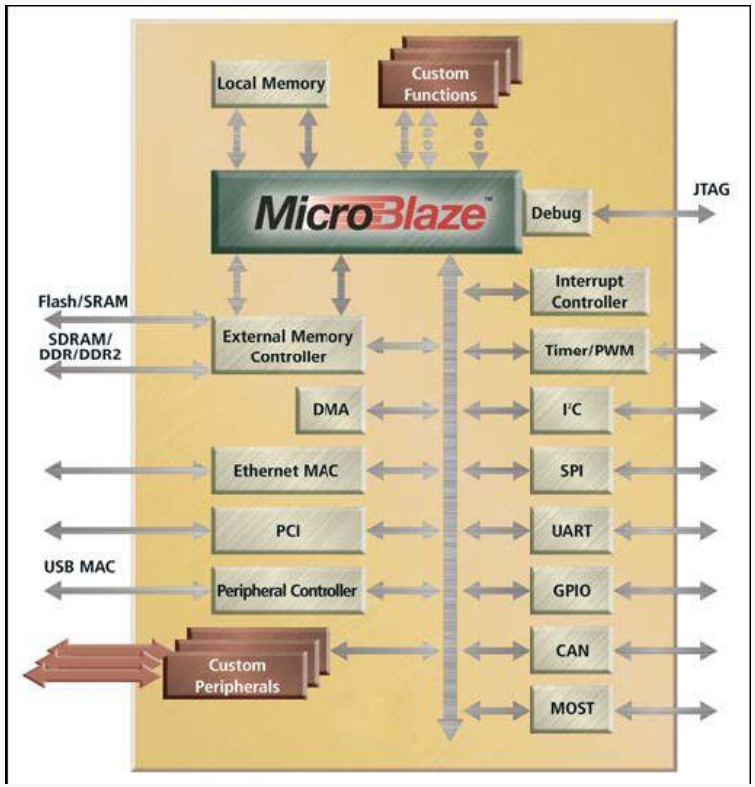
- Map from high-level logic to small configurable logic block
- Very high speed integrated circuit Hardware Description Language (VHDL) is the traditional way to describe FPGA behavior.

New advanced techniques
support variants of C, C++
and Java.

FPGA functionality can be built from larger pre-defined components.

Adding a “soft” processor core to the FPGA is common.

- ***Executes real “programs”***



Tiny OSs are available to manage resources.

RIT Software Engineering

As FPGAs improve they are being used in a larger number of application areas.

- Have the capability to replace dedicated digital signal processors in some applications
- No risks or up-front costs for custom designs (ASICs)
- Combine multiple devices (peripherals) and “glue” logic into one device.
- Capacity and features allow FPGAs to take on new applications.
- New design tools make application of FPGAs easier.

Tools play a very important role in the use of FPGA devices.

- Provide a good abstraction of the platform
- Provide mechanical conversion from high-level to lower-level description
- The optimal situation is that the tool “just works” without any operator intervention

The emerging tools focus on creating a software-oriented design experience.

- This approach is appropriate for several reasons
  - ***Software is a higher-level abstraction that will help manage growing complexity***
  - ***Many algorithms are specified and verified as software***

The technology does not eliminate the need for detailed hardware design skills.

- Substantial hardware design skills are still needed
- Optimizations are still needed at the very lowest levels but they are reduced
- Good tools can not save a poor algorithm

Applications with some specific characteristics lend themselves to software design methods.

- Pay careful attention to the hardware/software partitioning
  - ***Computational requirements***
  - ***Bandwidth requirements***
- Highly parallel applications work well
- Resist the temptation to off-load functions thinking like a remote procedure call