The Abdus Salam International Centre for Theoretical Physics

2499-23

International Training Workshop on FPGA Design for Scientific Instrumentation and Computing

11 - 22 November 2013

Embedded System Design using FPGA

Krishna Mohan KHARE
Laser Bio-Medical Applications and Instrumentation Division, Raja Ramana Centre for Advanced Technology, Indore 452013
India

EMBEDDED SYSTEM DESIGN USING FPGA

K. M. Khare
Senior Scientist
kmkhare@rrcat.gov.in

Raja Ramanna Centre for Advanced Technology
Department of Atomic Energy
Government of India
Indore - INDIA

ICTP, Trieste

November 21, 2013

Introduction

Our centre is know for LIGHT sources

- **LASERS**
- **SYNCHROTRON RADIATION SOURCE**

We are Involved:

- **Basic Research**
- **Design and Development**
- **Applications**

Laser Instrumentation

Various laser based instruments has been developed such as

- Uranium analyzer using N_2 laser.
- Land leveler,
- Surgical CO_2 laser system
- Density measurement system
- Micrometer
- Laser Marker
- Laser fluorescence spectroscopy of tissues etc.

Objective

- Basic concept of embedded systems
- Importance of FPGA based embedded systems
- Embedded system design flow and available tools
- Example design of FPGA based embedded system
- Physical Aspect of Hardware Design

Embedded Systems

- Embedded system is nearly any computing system
  - **Single function**
    - Typically designed to perform a predefined function
  - **Tightly constrained**
    - Tuned for low cost
    - Single-to-fewer components
    - Performs functions fast enough
    - Consumes minimum power
  - **Reactive and real-time**
    - Must continually monitor the desired environment and react to changes
  - **Hardware and software coexistence**

Embedded Systems…

- Examples:
  - **Mobile phone systems**
    - Customer handsets and base stations
  - **Automotive applications**
    - Braking systems, traction control, airbag release systems, and cruise-control applications
  - **Aerospace applications**
    - Flight-control systems, engine controllers, auto-piloting systems, and passenger in-flight entertainment systems
  - **Defense systems**
    - Radar systems, fighter aircraft flight-control systems, radio systems, and missile guidance systems

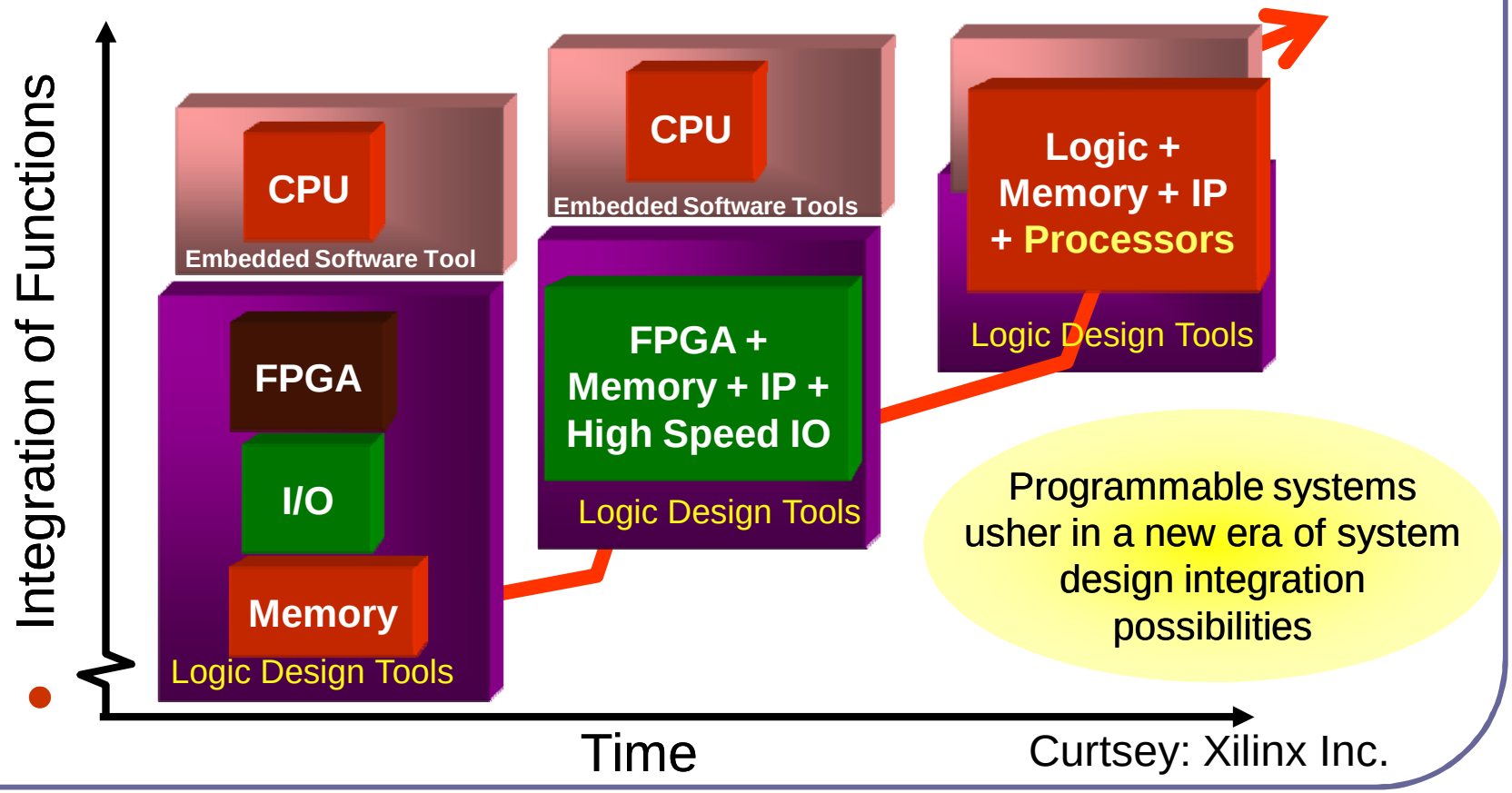
Embedded Systems…

Almost every embedded systems design includes

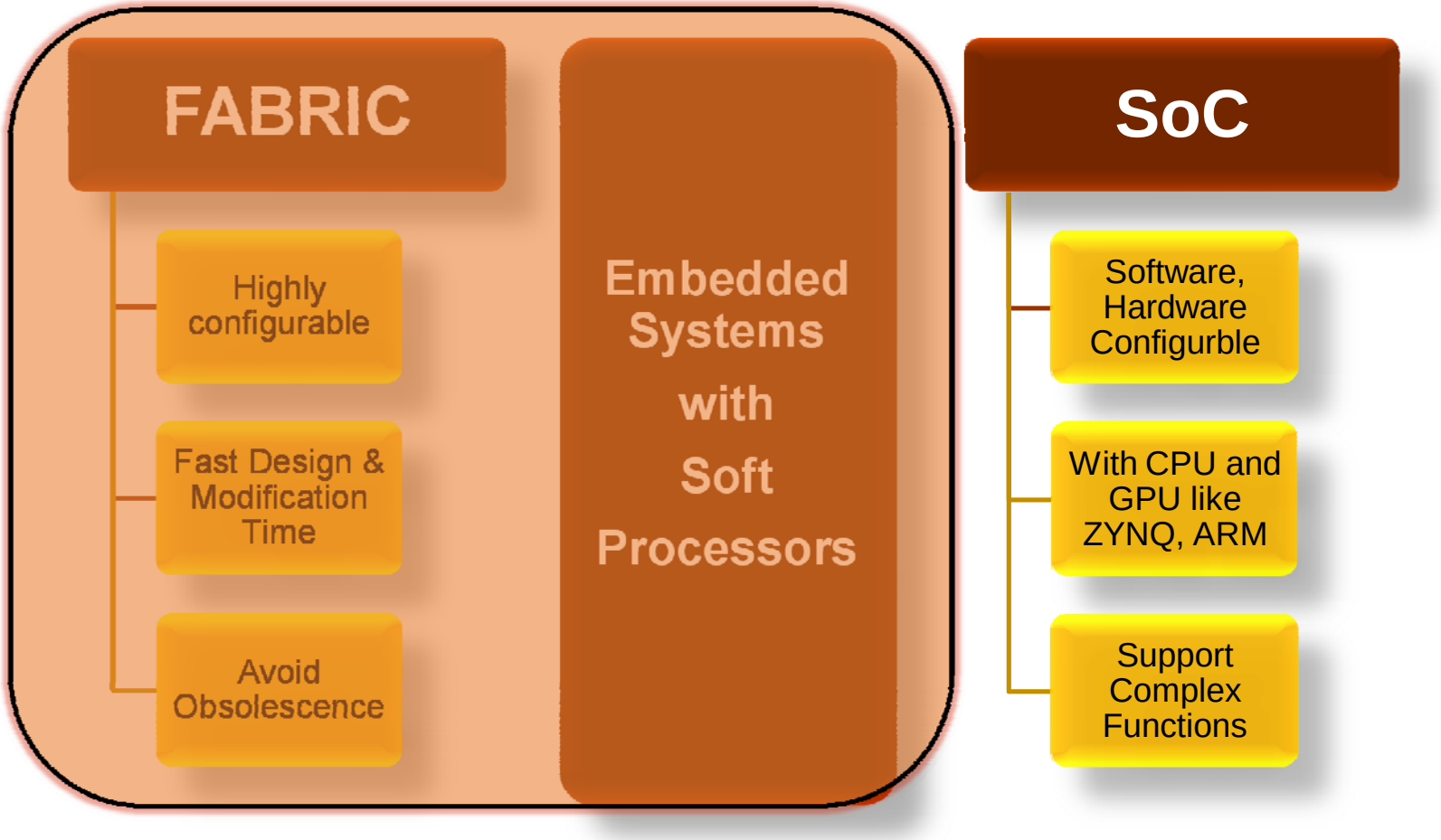
- Logic circuit design
- Processor-based hardware development
- Memory
- Other input output peripherals Interface

In a single or few chips solution.

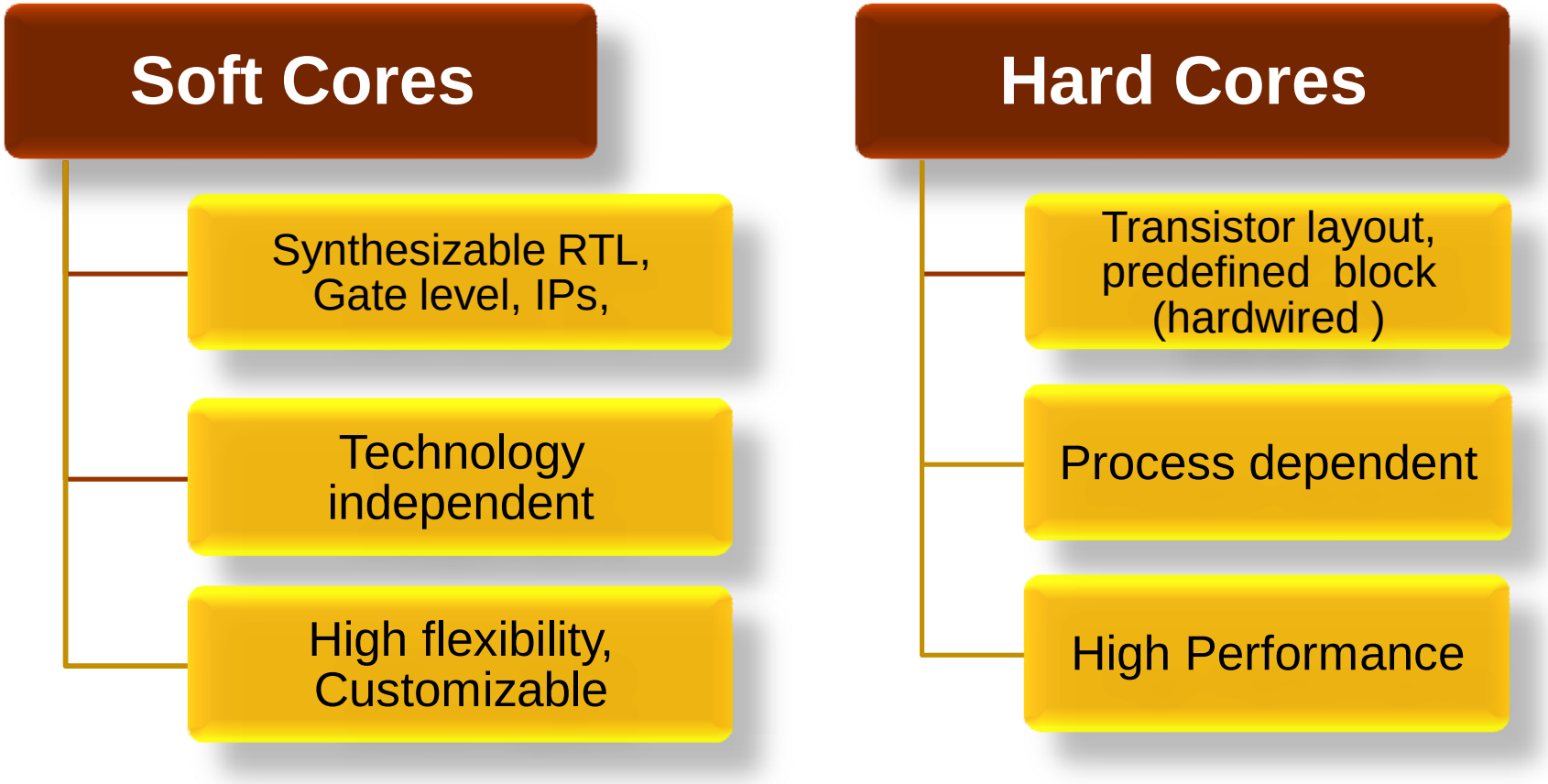
Integration in Embedded System Design



Embedded System Design…



Hardcore verses Soft-core Processors



Advantages of Softcore Processors

- Configurability to trade between price and performance,
- Faster time to market,
- Easy integration with the FPGA fabric,
- Avoids obsolesces.

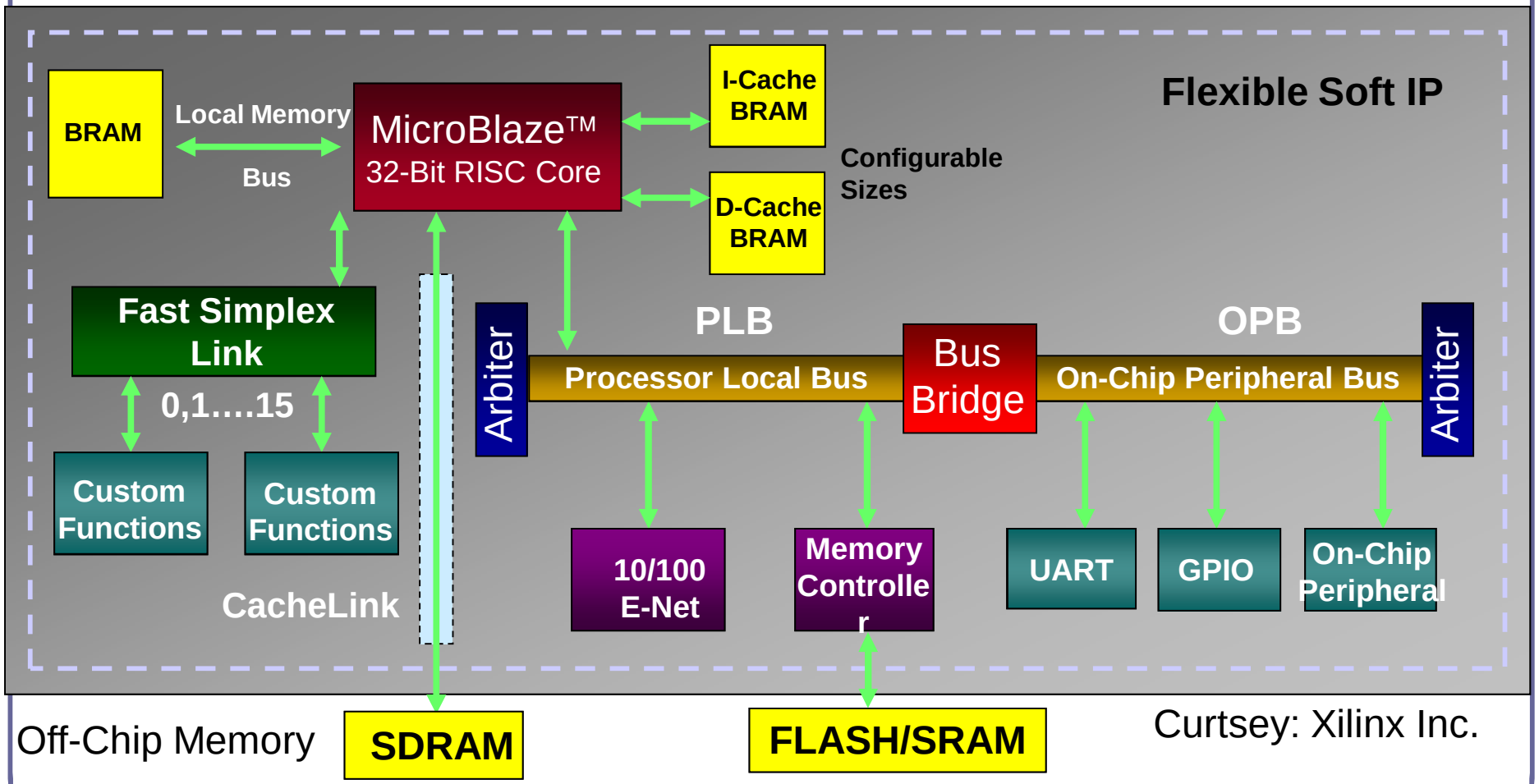
Softcores Processors…

- Different soft processor available from different FPGA manufacturers.
  - Xilinx : Picoblaze and Microblaze
  - Altera : Nios-II
  - Actel : Cortex-M3(From ARM)
  - Lattice : Lattice Micro32(open-source)
  - ARM : Cortex-M3(open-source)

FPGA based Embedded Design

- Embedded design in an FPGA consists of the following:
  - Develop FPGA hardware design
    - Customization of soft core processors and Custom IPs
  - Create the software application
    - Software routines
    - Interrupt service routines etc

MicroBlaze™ Processor Based-Embedded System Design



Curtsey: Xilinx Inc.

This is a v7.1 architecture. Versions 6.0 or earlier do not support PLB bus off the processor. Instead they have OPB bus

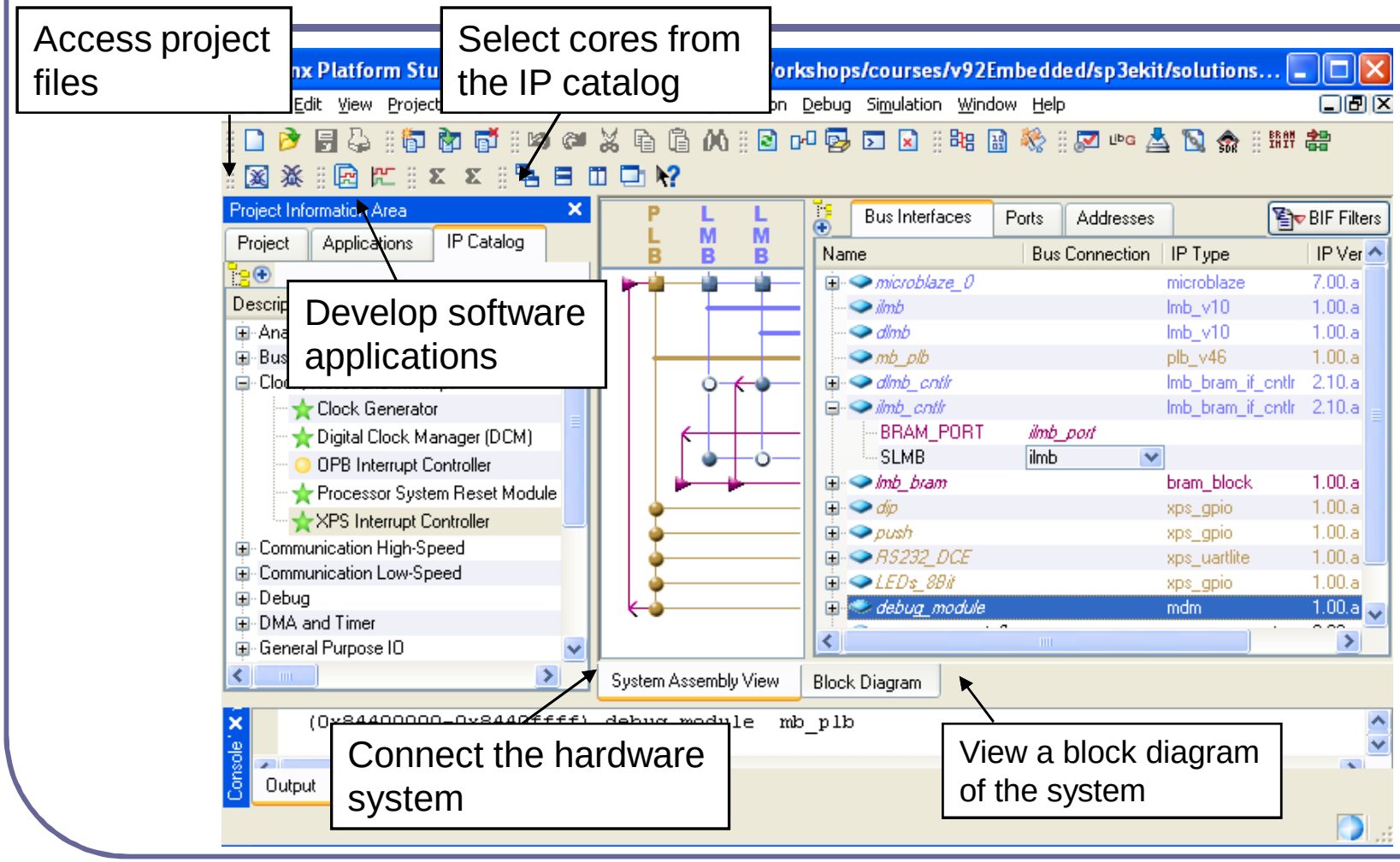
Partitioning the design

- Criteria for partitioning into hardware and software components
  - Picoseconds and nanosecond logic
  - To be implemented in hardware(fabric)
  - Microsecond logic
  - Can be implemented in hardware or software
  - Millisecond logic
  - Such as communication with slower peripherals can be mapped to software

Xilinx Embedded Development Kit (EDK)

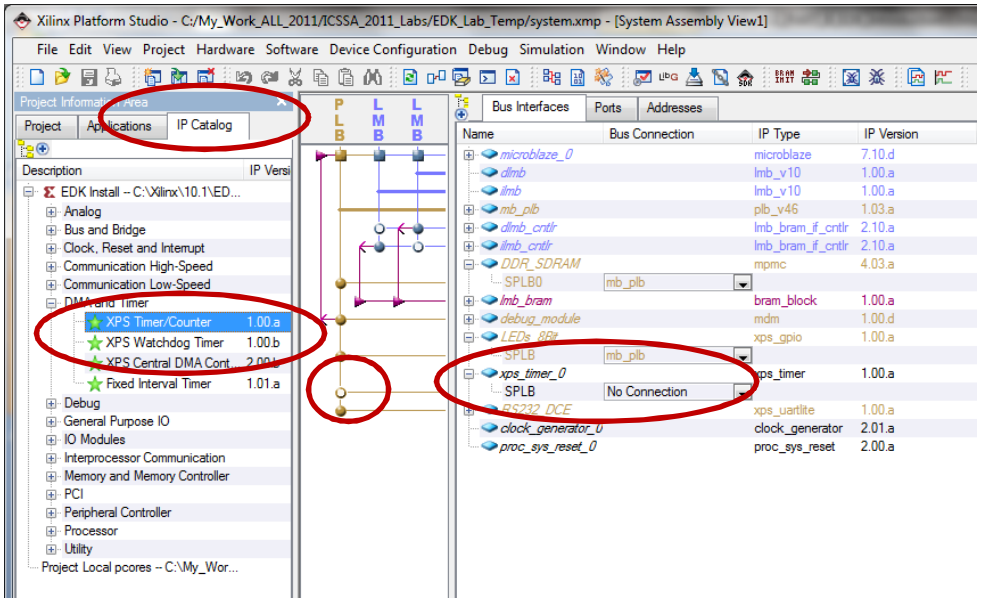
- Embedded Development Kit is the Xilinx software for
  - Designing complete embedded programmable systems.
  - Tools for customization and integration of soft processor cores.
  - Tools for integration of both hardware and software components of an embedded system.

Xilinx Platform Studio (XPS)

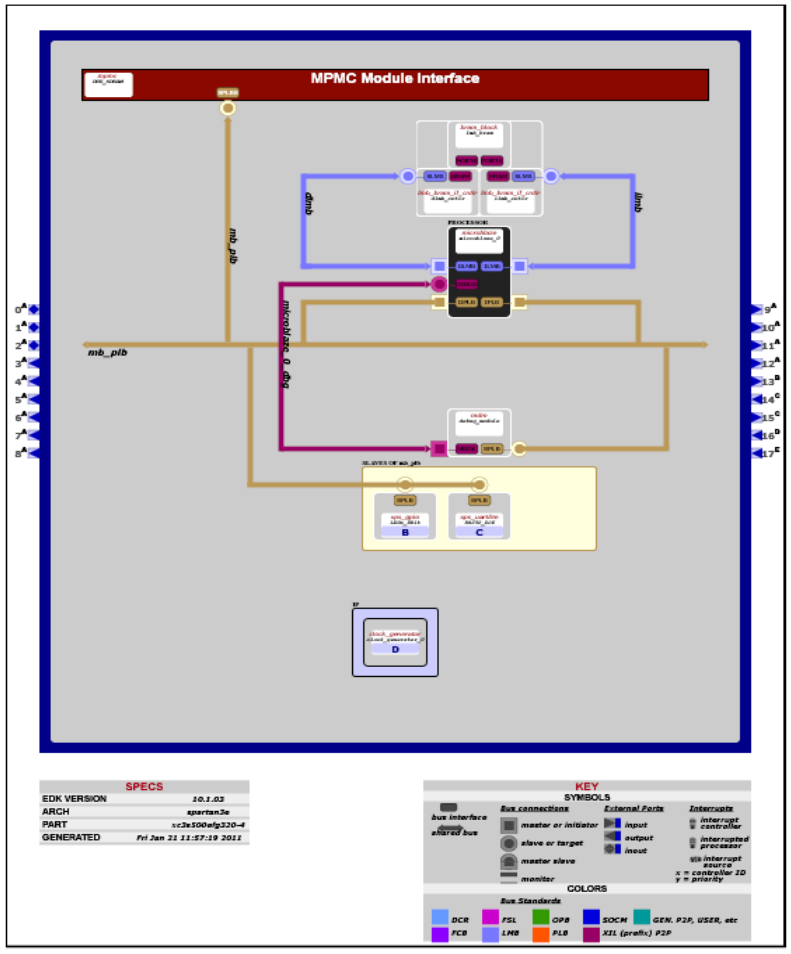


Adding IP and Bus Connection

- Add IP cores to an existing project, select the IP Catalog tab in XPS
- Select a core and drop it in the system view or double-click on it to add
- Select a bus instance to which it need to connect



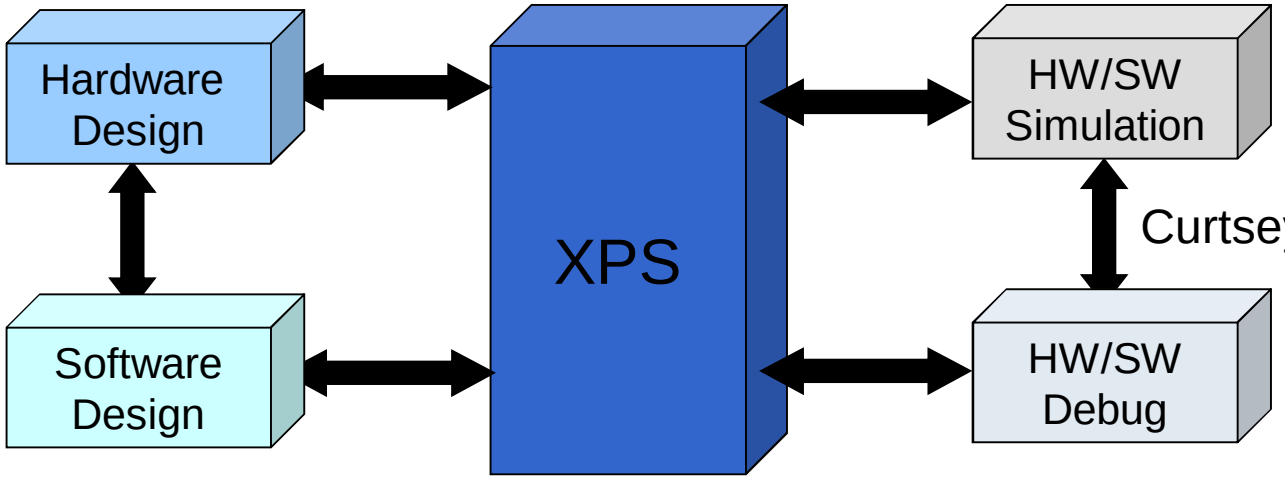
Created Hardware: Block Diagram



- Block diagram of created hardware shows the interconnection between the selected peripherals and processor
- In also shows the input and out ports of the system

XPS Functions

- Project management
  - Creation of MHS or MSS file
  - Xilinx Microprocessor Project (XMP) file
- Software application Management
- Platform management
  - Tool flow settings
  - Software platform settings
  - Tool invocation
  - Debug and simulation

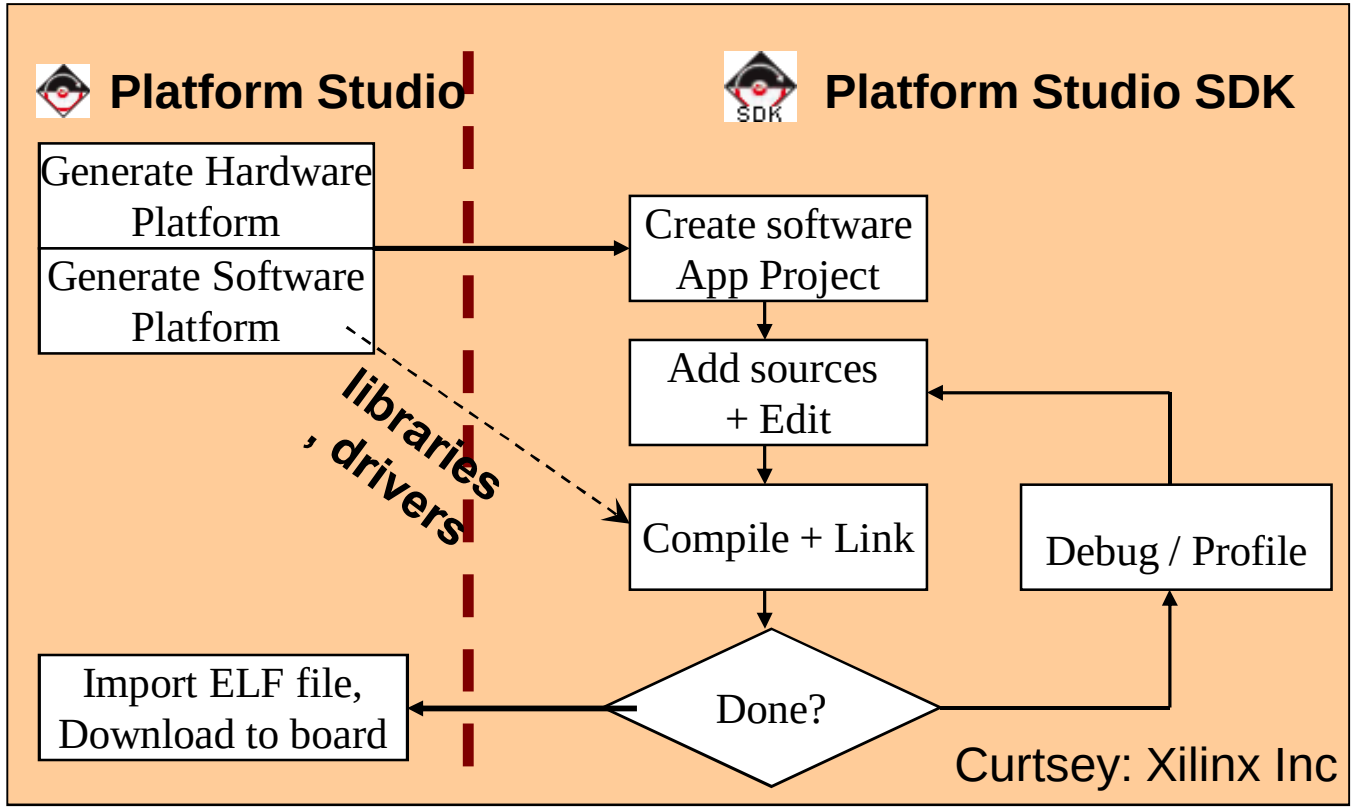


Curtsey: Xilinx Inc

Library Generation Flow

- Then Library Generator (**LibGen**) utility generates the necessary libraries, drivers and user project directories for the embedded software processors
- The LibGen takes Microprocessors Software Specification(MSS) file as input and produces an archive of object files libc.a, libxil.a and libm.a
- The MSS file, generated by XPS, defines the defines the drivers associated with peripherals, standard input/output devices, interrupt handler routines and other related software features

SDK Application Development Flow



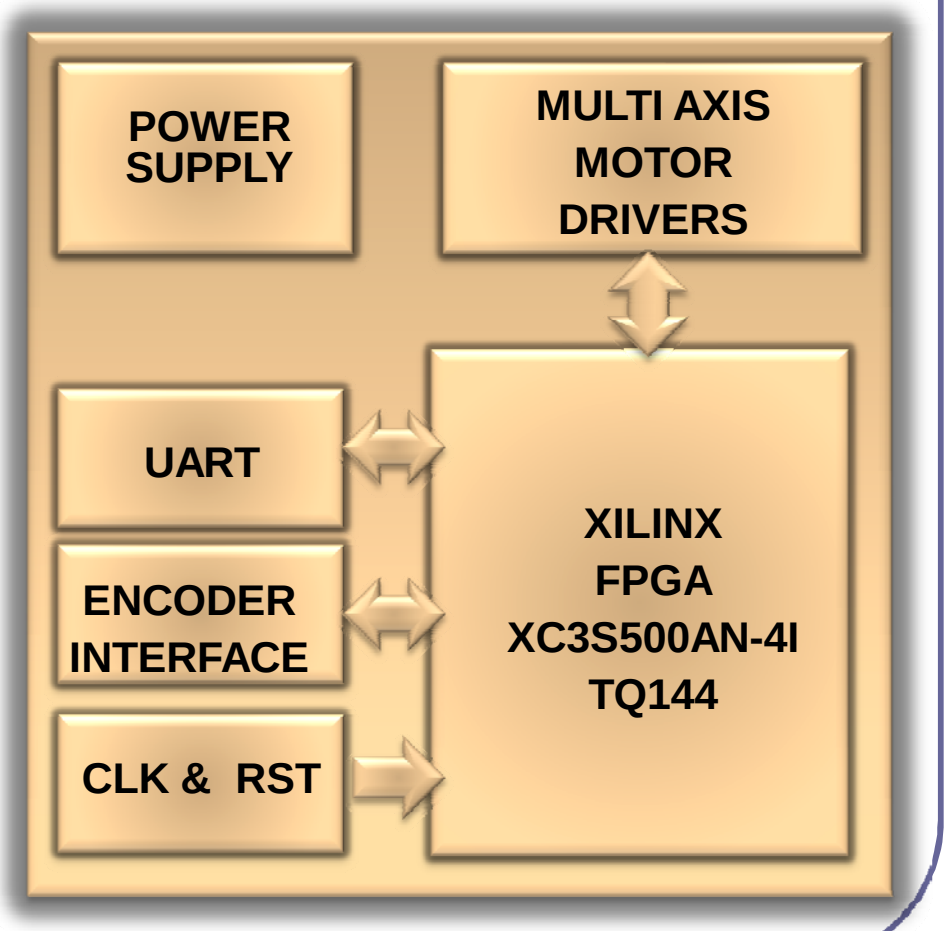
Libraries can be generate/updated from SDK

Merging Hardware and Software Flows

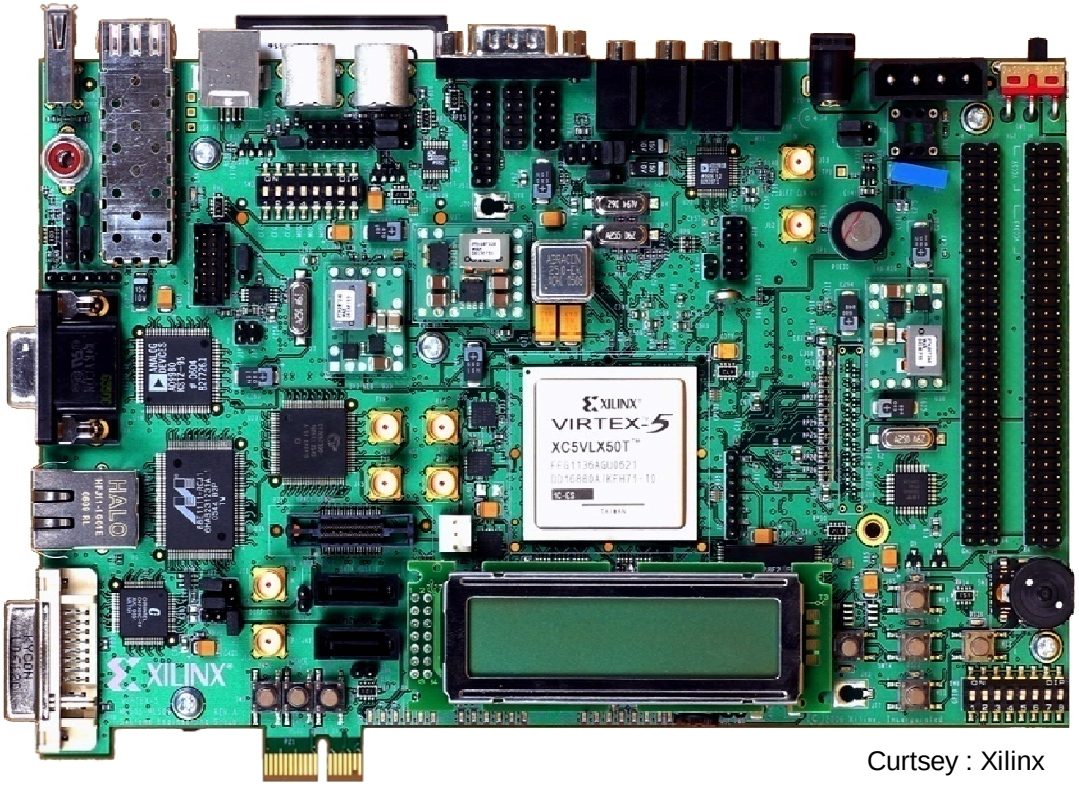
- Final download.bit file generated from the input files system_bd.bmm, system.bit and executable.elf files, which contains information regarding both the software and the hardware part of the design
- This invokes the data2MEM tool, which initializes the instruction memory of the processor
- This is the stage where hardware and software flows come together.
- Download the generated bitstream file

Example Design of Embedded System

- Designed system consists of softcore processor
- Custom IP for multi axis motor controllers, Encoder Interface.
- UART and general purpose I/Os Peripheral interface.
- Application Software for soft-core processor.

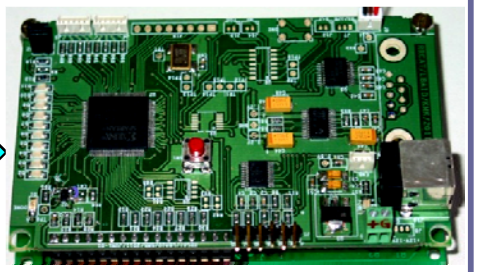


Development Board for Embedded Applications



Curtsey : Xilinx

Development Board



Custom Board

Hardware Design

- Why customization required?
- How to customize FPGA boards?
- Practical Example
- Advantages of customization
- Some Tricks and Techniques
- Tools Available
- Overview

Why Customization Required?

Customization technique used very widely:

- Fulfill specific requirements.
- Gain the competitive edge.
- Save time, money and space.
- Minimize board complexity.
- Reduce unused components
- Make it flexible.

How to customize FPGA Boards?

- Specify your product features
- Selection of FPGA
- Incorporating essential circuits
- Specify the constraints
- Implementation techniques
- Testing and debugging

How to customize FPGA Boards? (cont..)

Specification of Product Features

- UART/USB/Ethernet
- DISPLAY(LCD/OLED/Touch Screen/Matrix)
- Key Board/Buttons
- External Memory Interface
- ADCs/DACs
- Indication and Debug port

How to customize FPGA Boards? (cont..)

Selection of FPGA

An appropriate FPGA for specific application is determined by the following features.

- Density
- No. of I/Os
- Package
- Speed Grade
- Vendor
- Series
- Part no.

How to customize FPGA Boards? (cont..)

Incorporating Essential Circuits

- Power Circuit
- Clock Circuit
- Reset Circuit
- In system programming circuit (JTAG)
- Configuration Memory Interface
- Debug Port

How to customize FPGA Boards? (cont..)

Power Circuit Design

While designing power supply to fulfill the power requirements of FPGA and other peripherals on board, we keep following points in mind.

- Voltage and current requirements
- Voltage tolerances
- Power distribution
- Sequencing
- Monotonicity
- Power up ramp time.

How to customize FPGA Boards? (cont..)

Power Circuit Design ….

- **Voltage Requirement:** Most of FPGAs require multiple power supply.
  - Internal core logic power supply (VCCINT).
  - Input Output drivers power supply (VCCO).
  - Auxiliary power supply (VCCAUX).

- **Current Requirement:** Depends on
  - Logic utilization,
  - Frequency of operation
  - Other on board peripherals.

  Hence estimate power requirement before designing the power circuit.

How to customize FPGA Boards? (cont..)

Power Circuit Design….

- **Voltages Tolerances:** Check the voltage tolerances of FPGA and peripherals. Generally voltage tolerances of FPGA in the order of 5% to 10% of voltage requirement.
- **Power Distribution**: Power distribution should be such that to maintain power around the device during peaks and drops.
- **Power Sequencing:** It is good design practice to switch on power in sequence (core and then I/O) to avoid initial power on surges. Most of the FPGAs do not require power sequencing,
- **Monotonicity:** Ramp the voltages without any dips in the power ramp up to respective threshold.
- **Power up Ramp Time:** It should neither be fast nor be slow. Ex. Minimum ramp time is 200 micro second and maximum is 100 mili seconds.

How to customize FPGA Boards? (cont..)

Switching vs. Linear Regulators

Linear Regulators

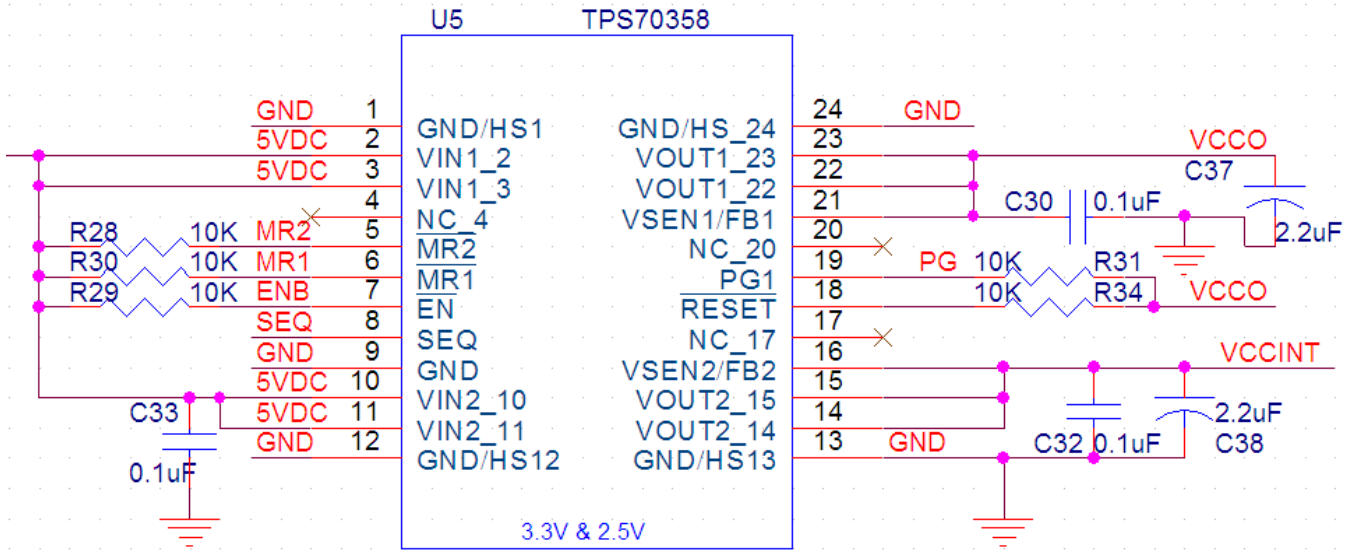
- Advantages
  - Good for low power applications
  - Few external components
  - Low output noise
  - Fast response to output disturbances
- Disadvantages
  - Lower efficiency
  - Higher power consumption
  - Limited range for Vin/Vout

Switching Regulators

- Advantages
  - Higher efficiency
  - Lower power consumption
  - Large Vin/Vout range, largely independent of load current
  - Ability to step-up and step-down.
- Disadvantages
  - More external components if modules are not used

How to customize FPGA Boards? (cont..)

Power Circuit Design….



Example: Power supply design for Spartan-II FPGA

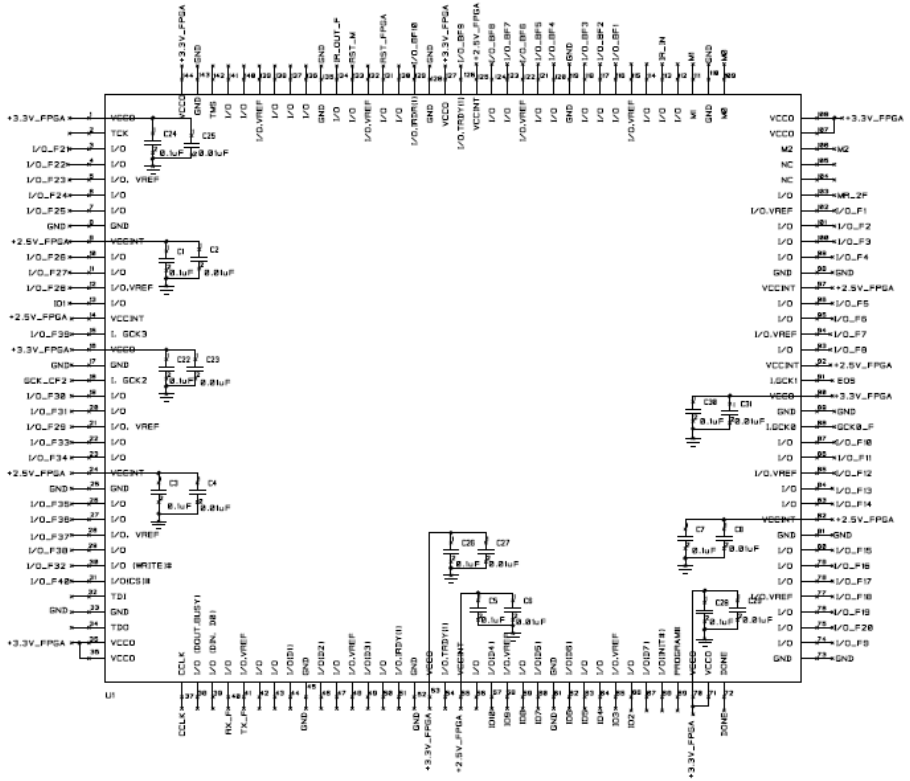
How to customize FPGA Boards? (cont..)

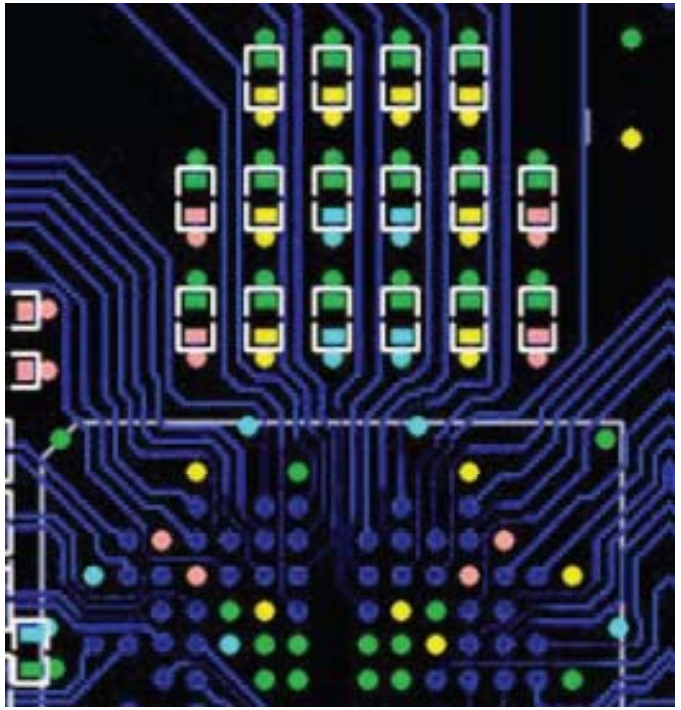
Decoupling Capacitors

- Power and Ground voltages are affected by logic transition and may cause the operational issues.
- The external package pins should be properly decoupled, which provides local energy storage, for stable power supply and ground.
- Proper decoupling improves the overall signal integrity .
- Decoupling capacitors should place as close as possible. Smaller the capacitor package, lower the Inductance and hence recommended.

How to customize FPGA Boards? (cont..)

Decoupling Capacitors…





Curtsey : Xilinx

How to customize FPGA Boards? (cont..)

Clock Circuit

- External clock should be connected to global clock inputs (GCLK) pin of the FPGA.
- GCLK pins are low-capacitance, low-skew interconnect lines well-suited to carrying high-frequency signals throughout the FPGA.

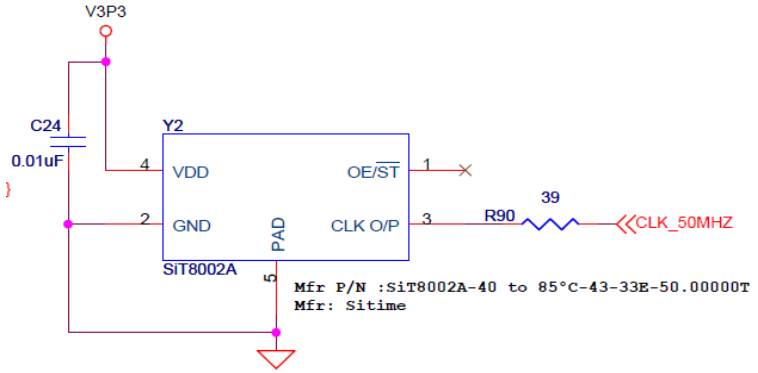


Fig.: Clock Circuit using oscillator

How to customize FPGA Boards? (cont..)

Reset Circuit

- Most of the FPGA has internal power on reset.
- It's good design practice to provide external power on reset
- RST should be connected to global Set/Reset (GSR) pin of the FPGA.

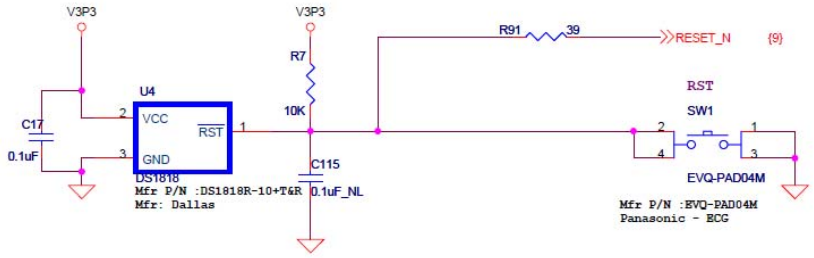


Fig.: Reset Circuit

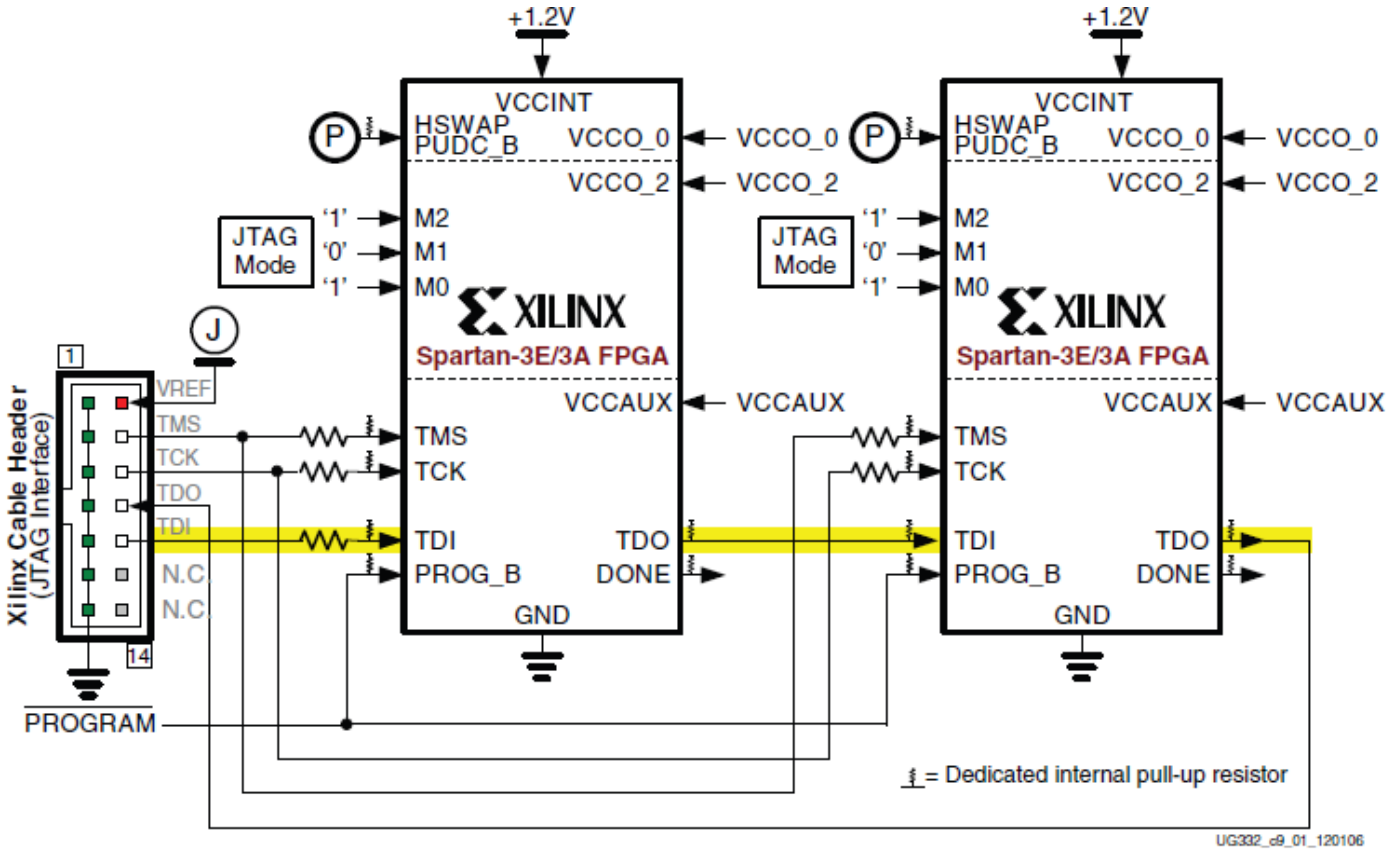
How to customize FPGA Boards? (cont..)

In System Programming (ISP)

- Dedicated JTAG port is available in all FPGA.
- Every package has four dedicated JTAG pins. Namely, TDI, TMS, TCK (Input) and TDO (Output).
- Internal charge pumps create high voltages for programming the memories powered by VCCAUX.
- The signal integrity of the TCK signal is critical because all JTAG operations are synchronous to the TCK clock.
- The JTAG interface is easily cascaded to any number of FPGAs by connecting the TDO output of one device to the TDI input of the next device in the chain. The TDO output of the last device in the chain loops back to the port connector.

How to customize FPGA Boards? (cont..)

In System Programming (ISP)



Curtsey : Xilinx

How to customize FPGA Boards? (cont..)

In System Programming (ISP)

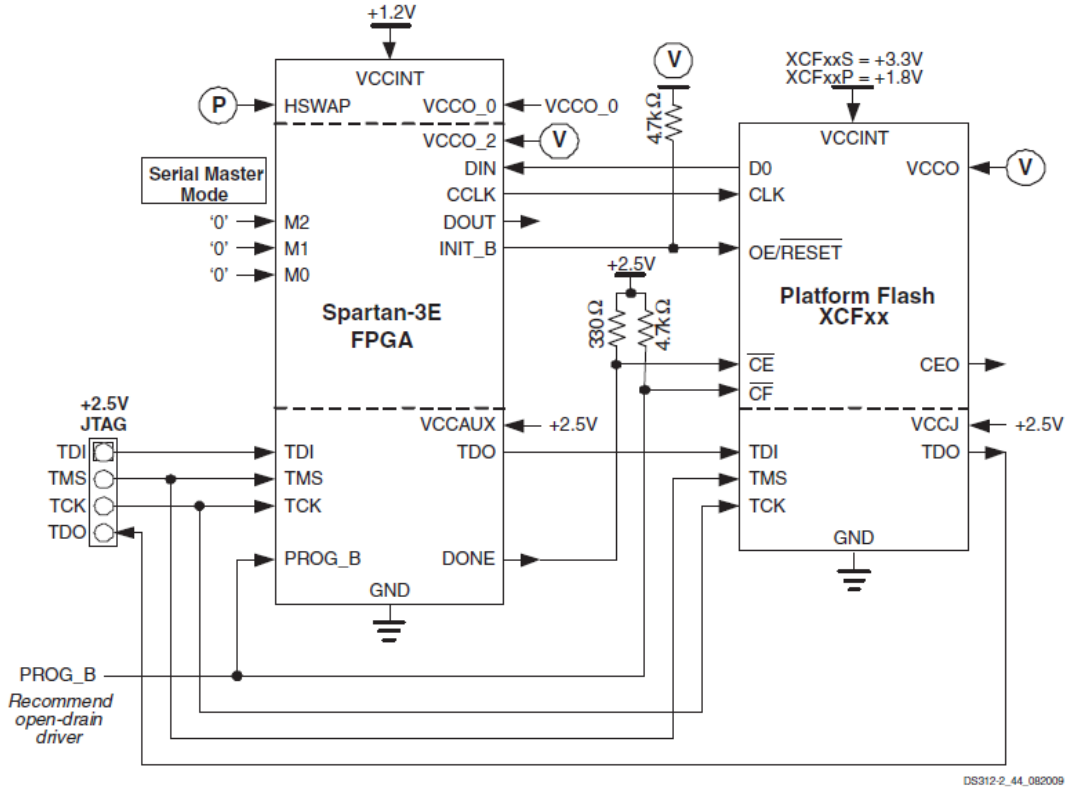


Figure 51: **Master Serial Mode using Platform Flash PROM**

Curtsey : Xilinx

Mode configuration Pin: M0. M1,M2.

How to customize FPGA Boards? (cont..)

Debug & Testing Port

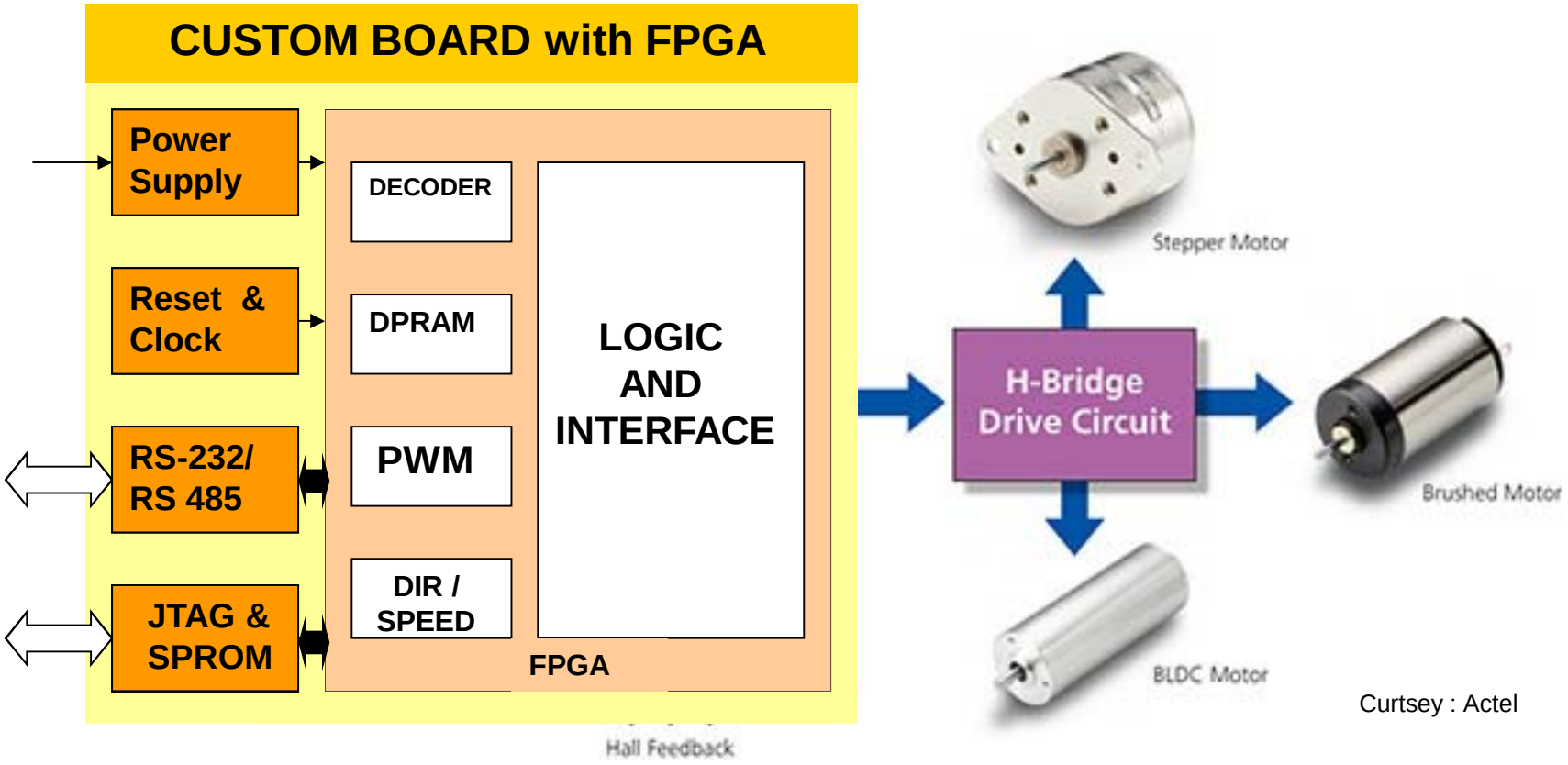
- Wherever it is necessary, implement the hardware debugging and testing point must tap and brought out of circuit.

  Ex. Power supply, clk, Rx and Tx etc

- Some of I/O of the FPGA may also be connected through switch/LED and brought out for debugging of hardware. This port may also be used for implementation debugging.

Practical Example:

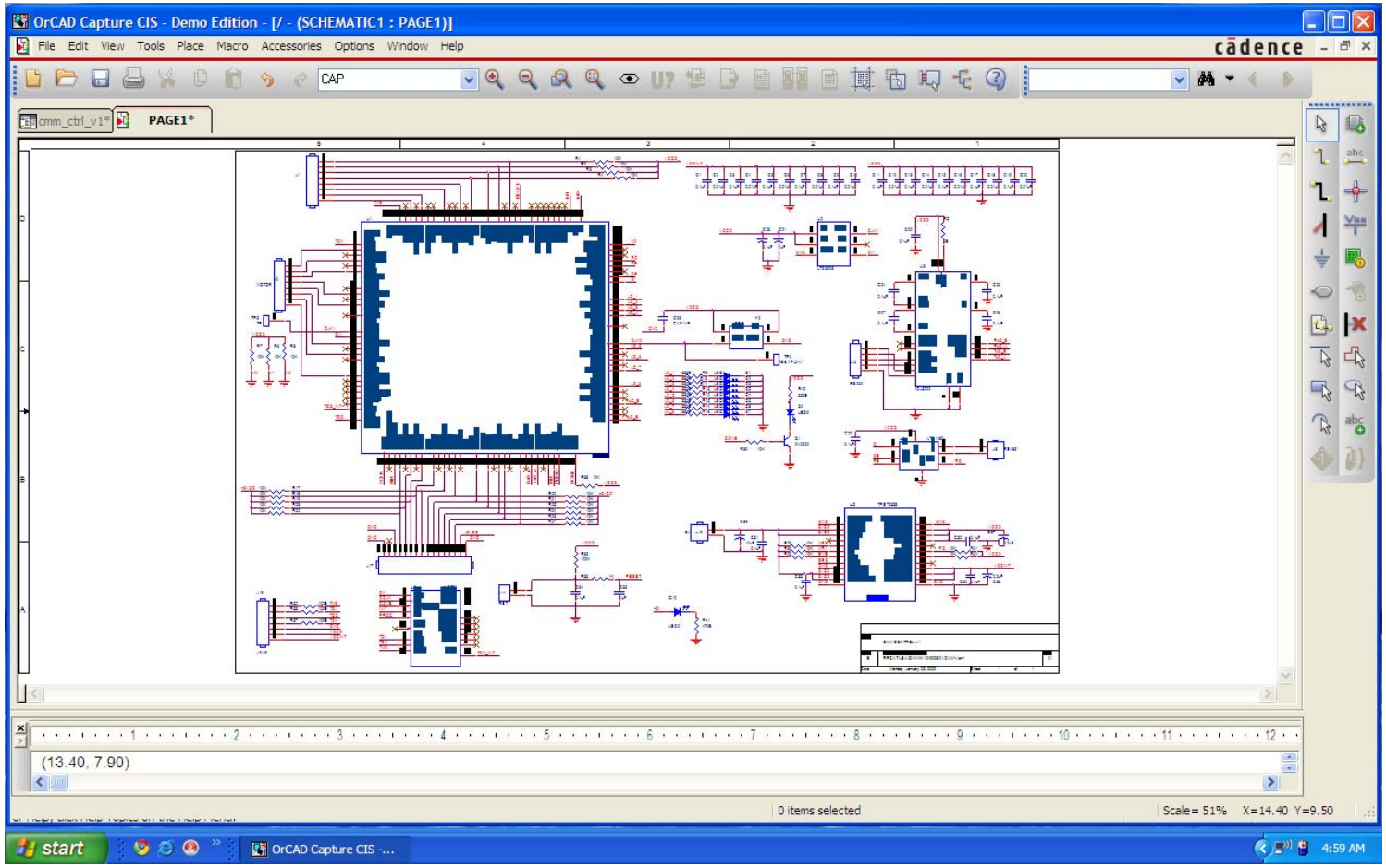
FPGA Based Motor Controller



Curtsey : Actel

Schematic Design:

For Motor Controller



Tools Available

- Cadence PCB design Tools:
  - OrCAD PCB design tools
  - Allegro PCB design tools
  - OrCAD/Allegro FPGA system planner
- ModelSim PCB design Tools
  - PADs
- Zuken PCB design tools

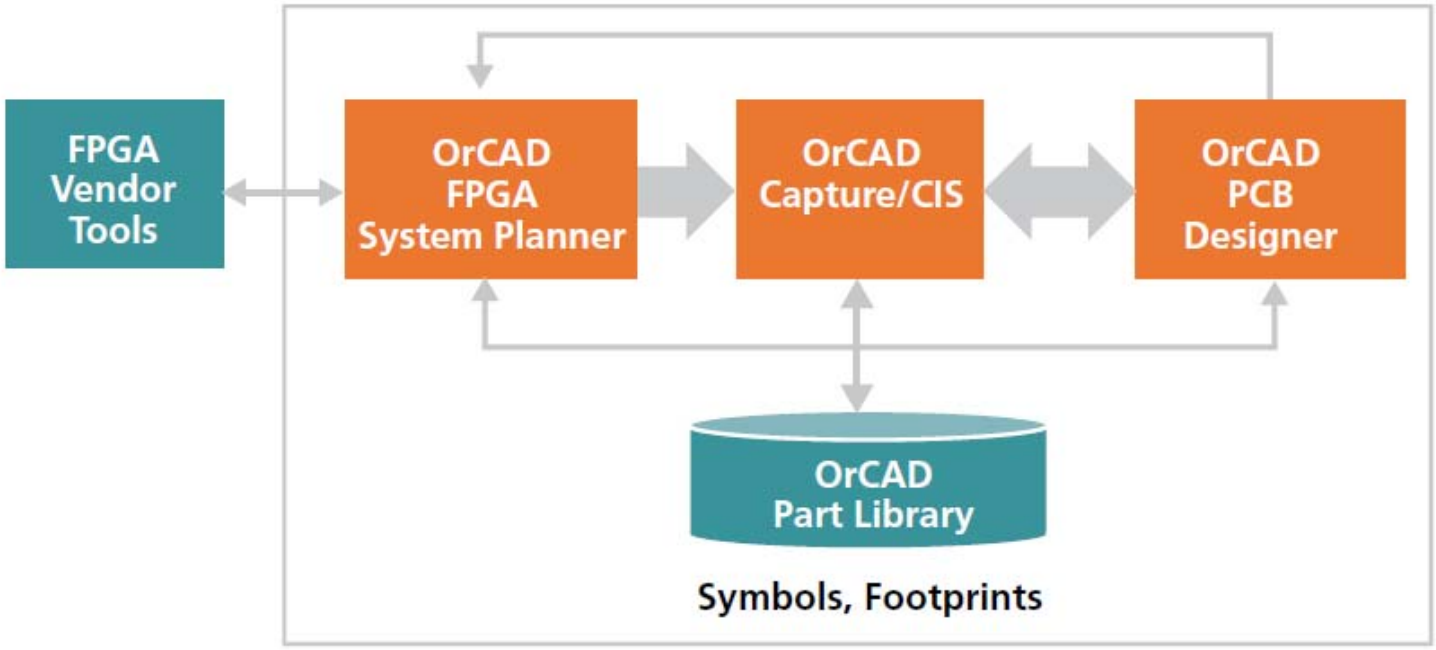
and many more…

PCB Designing Tools…

Almost all available PCB design tools are bundle of following modules used for different stages of the PCB design.

- Schematic design tool
- PCB design tool
- Foot print editor
- Thermal analysis tools
- EMI analysis tools
- Verification Tools
- Simulation tools

OrCAD FPGA System Planner…



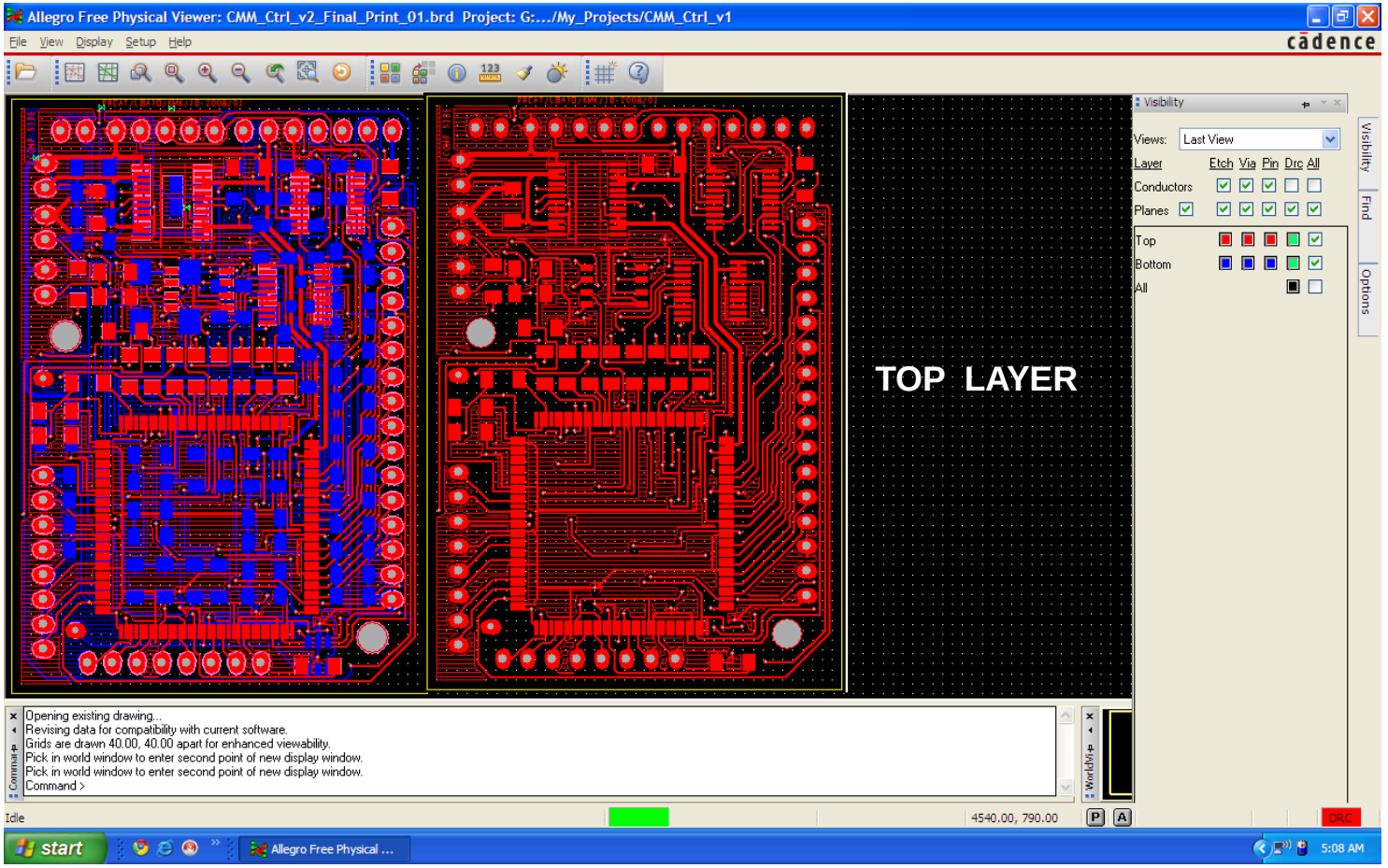
Curtsey : Cadence

Fig.: *The OrCAD FPGA System Planner Methodology*

PCB Design

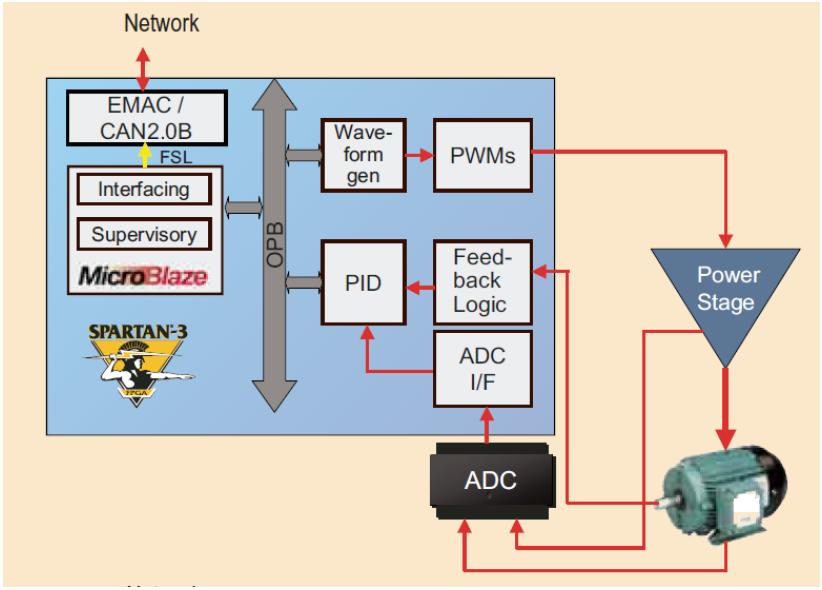
for Motor Controller

After Routing



Embedded Controller

FPGA: For sequential control module and digital PID etc



FPGA: For peripheral Interface and other digital I/Os etc

Thank You