



Embedded Computing and Partial Reconfiguration

Marc Defossez

May 2005

Agenda

- Virtex-IIpro architectural overview.
- Virtex-4 architectural overview.
- Software flow.
- Configuration, frames and Bits.
- Partial re-configuration.
- Embedded partial re-configuration.
- Partial re-configuration flow.



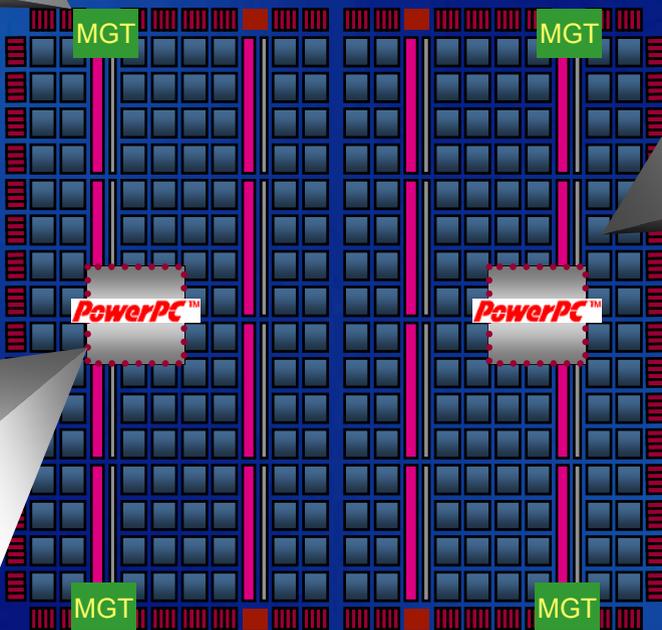
Virtex-II Pro Platform FPGA

RocketIO

3.125 Gbps Multi-Gigabit Transceivers (MGTs)
Supports 10 Gbps standards
Up to 24 per device

PowerPC™

- PowerPC 405 Core
- 300+ MHz / 450+ DMIPS Performance
- Up to 4 PPC405 per device



VIRTEX-II
Fabric

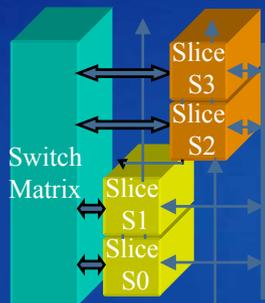
- IP-Immersion™ Fabric
- ActiveInterconnect™
- 18Kb Dual-Port RAM
- Xtreme™ Multipliers
- 16 Global Clock Domains

VIRTEX-II
PRO



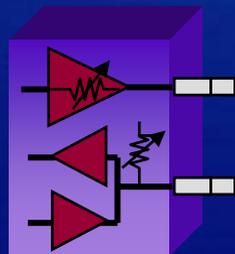
Common Virtex-II Series Fabric

Powerful CLB



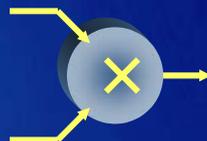
- 8 LUTs, distributed RAMs, or Shift Registers

XCITE technology™

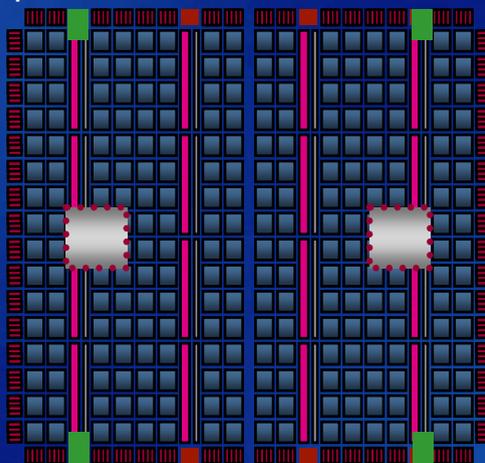


- Improves Signal Integrity plus Eliminates 100's of Resistors

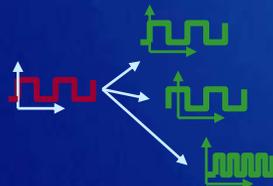
Xtreme DSP™ Multipliers



- Up to 200MHz 18b x 18b multiply

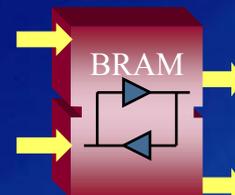


Digital Clock Managers



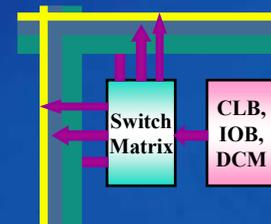
- Zero Delay Clock, Phase Shift, Frequency Synthesis

Block RAM



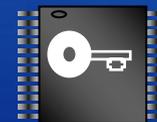
- 18KBit True Dual Port Blocks (Up to 10Mbits Total)

Active Interconnect™



- Fully Buffered, Fast, Predictable

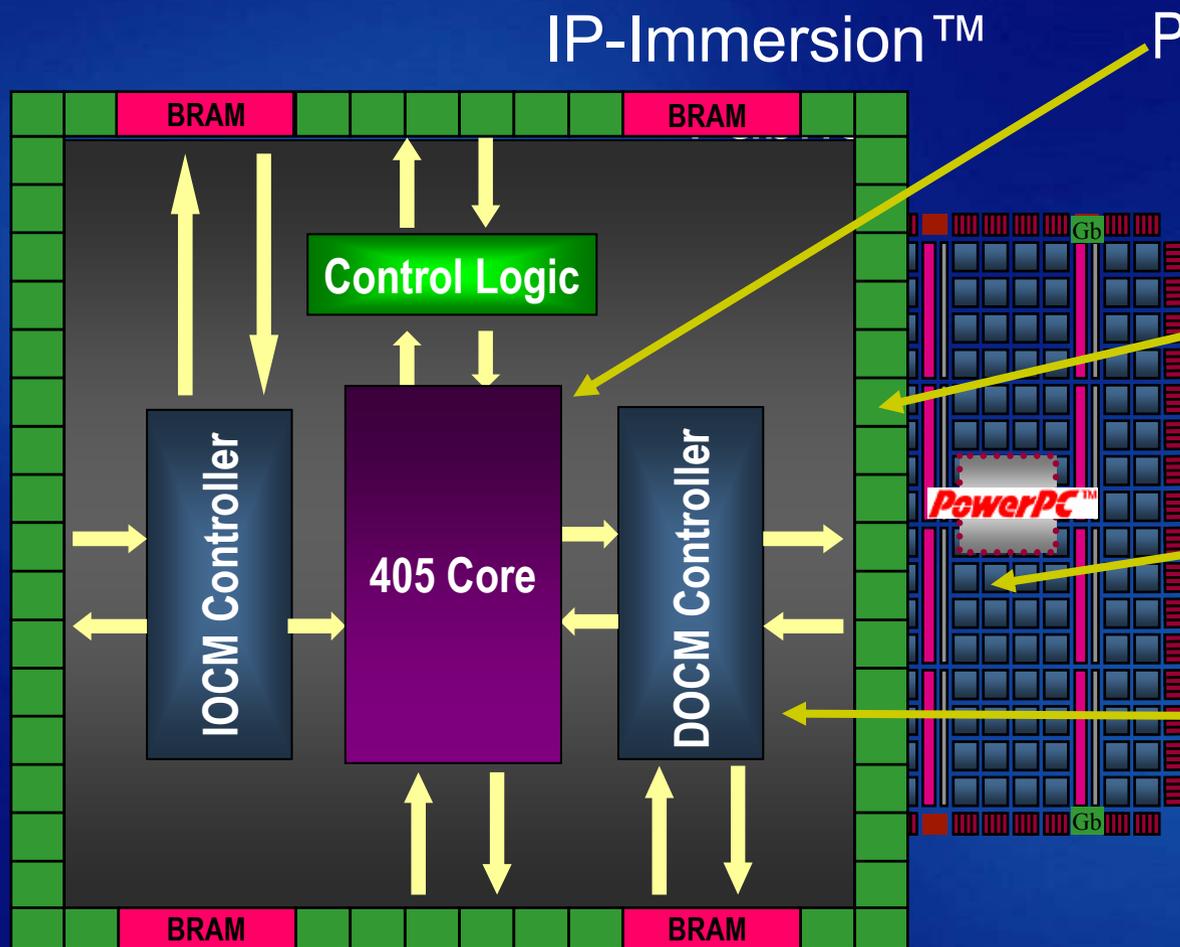
Bitstream Security



- Triple DES Encryption



Processor Integration Technology



IP-Immersion™

PowerPC 405 Core

- MMU Enable
- Operand Forwarding
- Deterministic Multiply
- N:1 PLB Clock Ratio

IP-Immersion Tiles

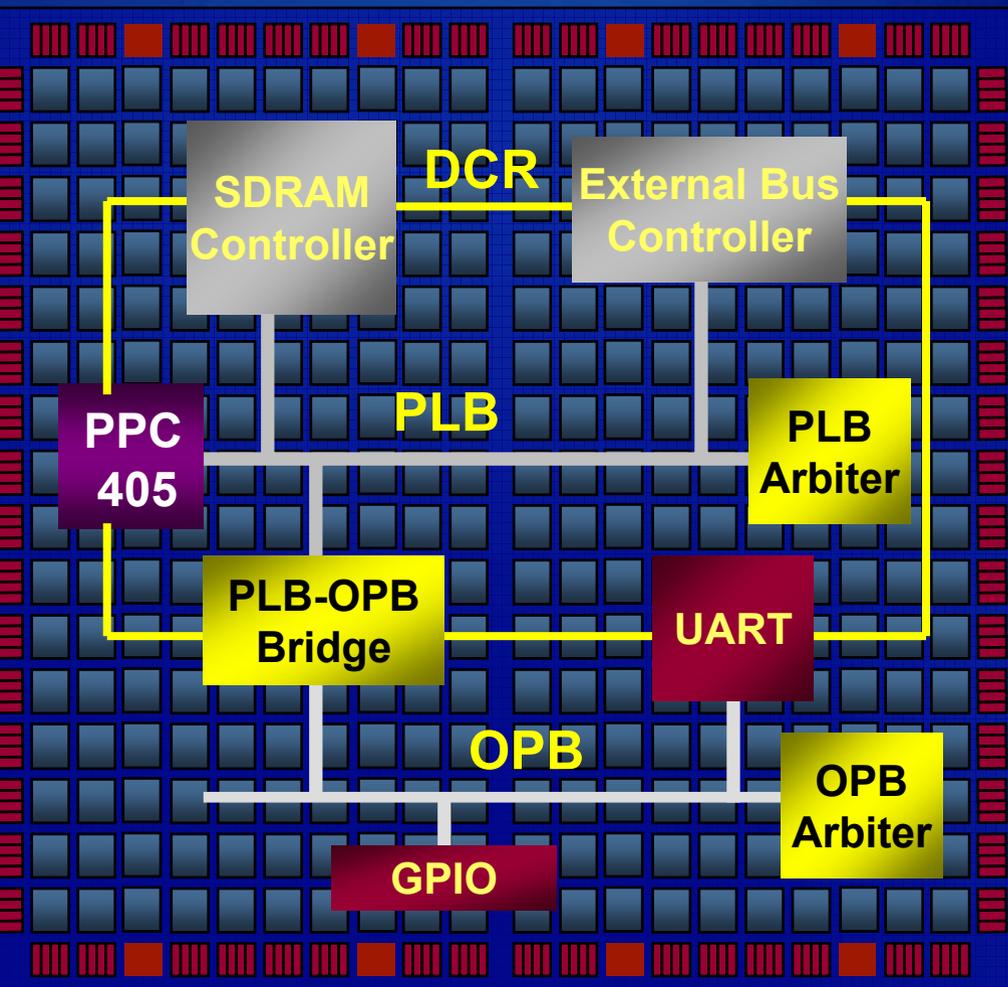
provide IP-to-Fabric connectivity

FPGA CLB Array

- Memory Map Location
- BRAM Size
- Operand Forwarding
- OCM Enable
- N:1 BRAM Clock Ratio

Interface Logic

CoreConnect™ Bus Architecture



- Processor Local Bus (PLB)
 - 32-bit address, 64-bit data
 - Separate read and write busses for overlapped transfers
 - 2.1 GB/s Max BW @ 133 MHz
- On-Chip Peripheral Bus (OPB)
 - 32-bit address, 32-bit data
 - Single cycle data transfers
- Device Control Register Bus (DCR)
 - 32-bit transfers to and from GPR

Agenda

- Virtex-IIpro architectural overview.
- **Virtex-4 architectural overview.**
- Software flow.
- Configuration, frames and Bits.
- Partial re-configuration.
- Embedded partial re-configuration.
- Partial re-configuration flow.



New ASMBL™ Architecture

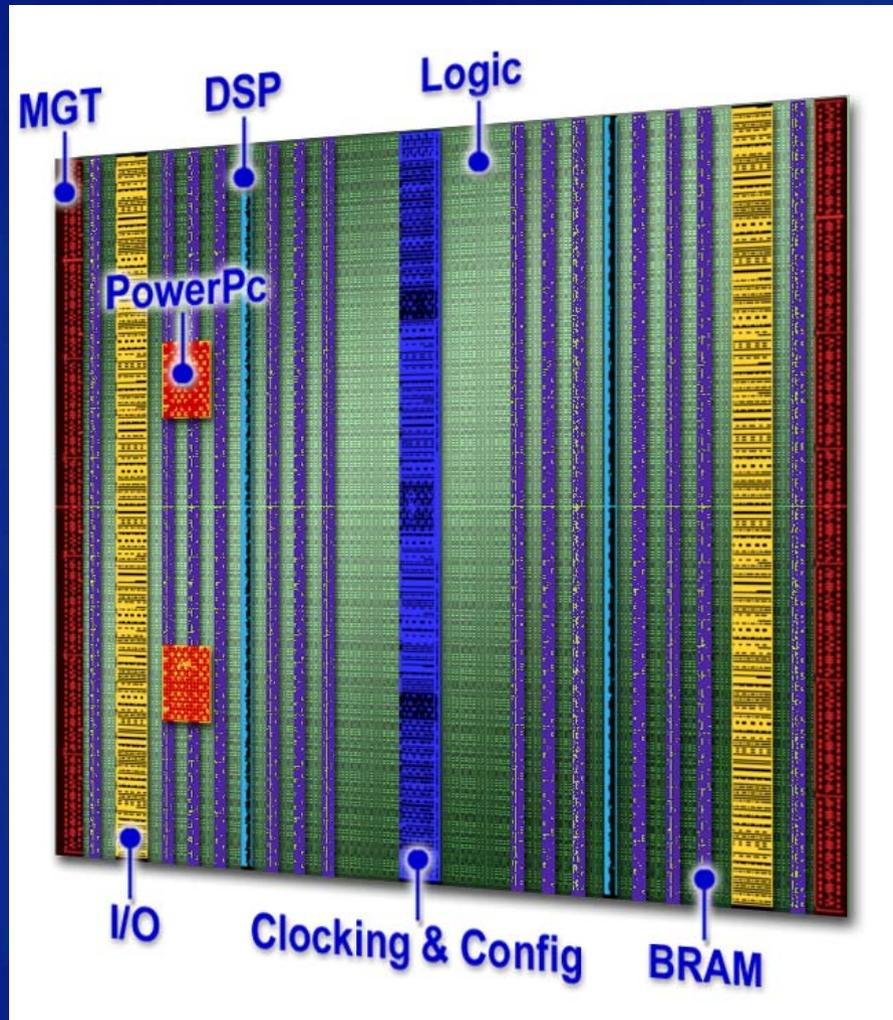
- **Functional Blocks**

 - Organized in Columns

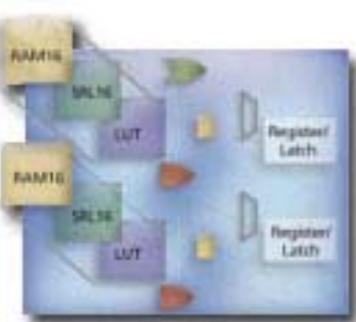
 - Logic, DSP, BRAM, I/O, MGT, DCM, PowerPC
 - Removes feature-to-feature interdependencies

- **Key Benefits**

 - Customer gets wider choice by Xilinx offering families with different feature mix (LX / FX / SX)
 - Distributed IO improves signal integrity and PWR / GND distribution
 - Distributed features give additional placement flexibility



Embedded Hard IP



200,000
Logic Cells



500 MHz Xesium™
Differential Clocking



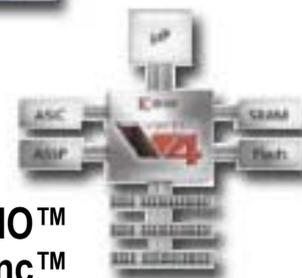
500 MHz BRAM
with FIFO & ECC



0.6-11.1 Gbps
RocketIO™
Transceivers

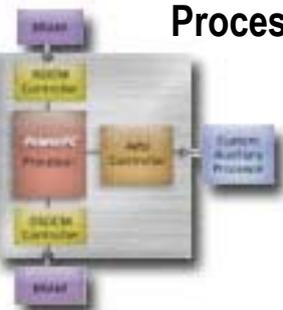


AES Secure Chip
Design Security



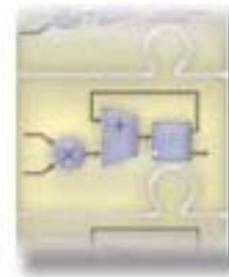
1 Gbps SelectIO™
with ChipSync™

PowerPC®
Processor with APU

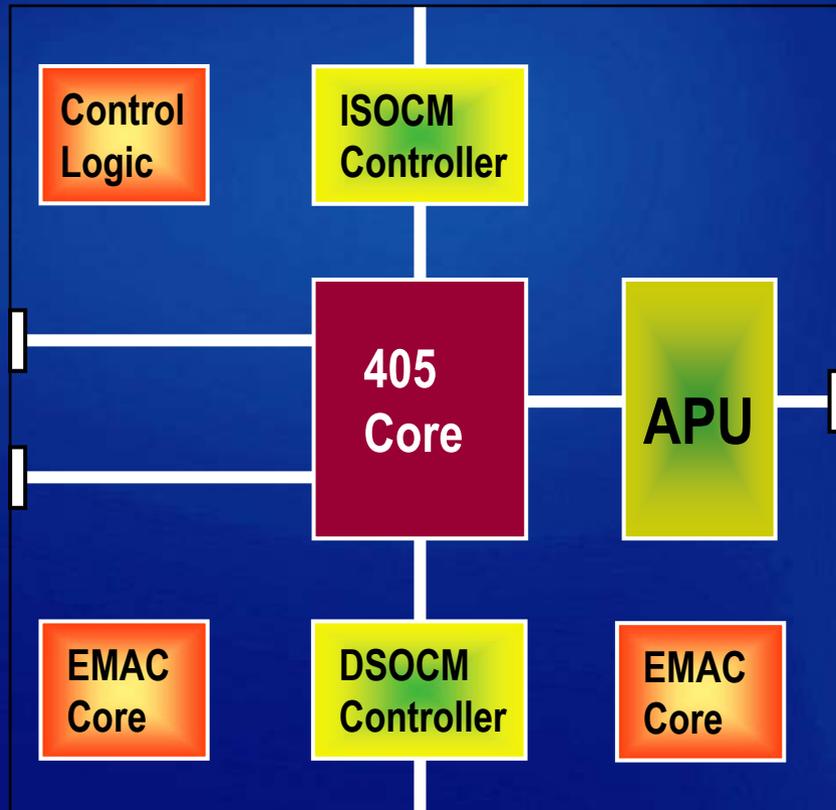


10/100/1000
Ethernet MAC

500 MHz
XtremeDSP™ Slice



Virtex-4 Processor Block



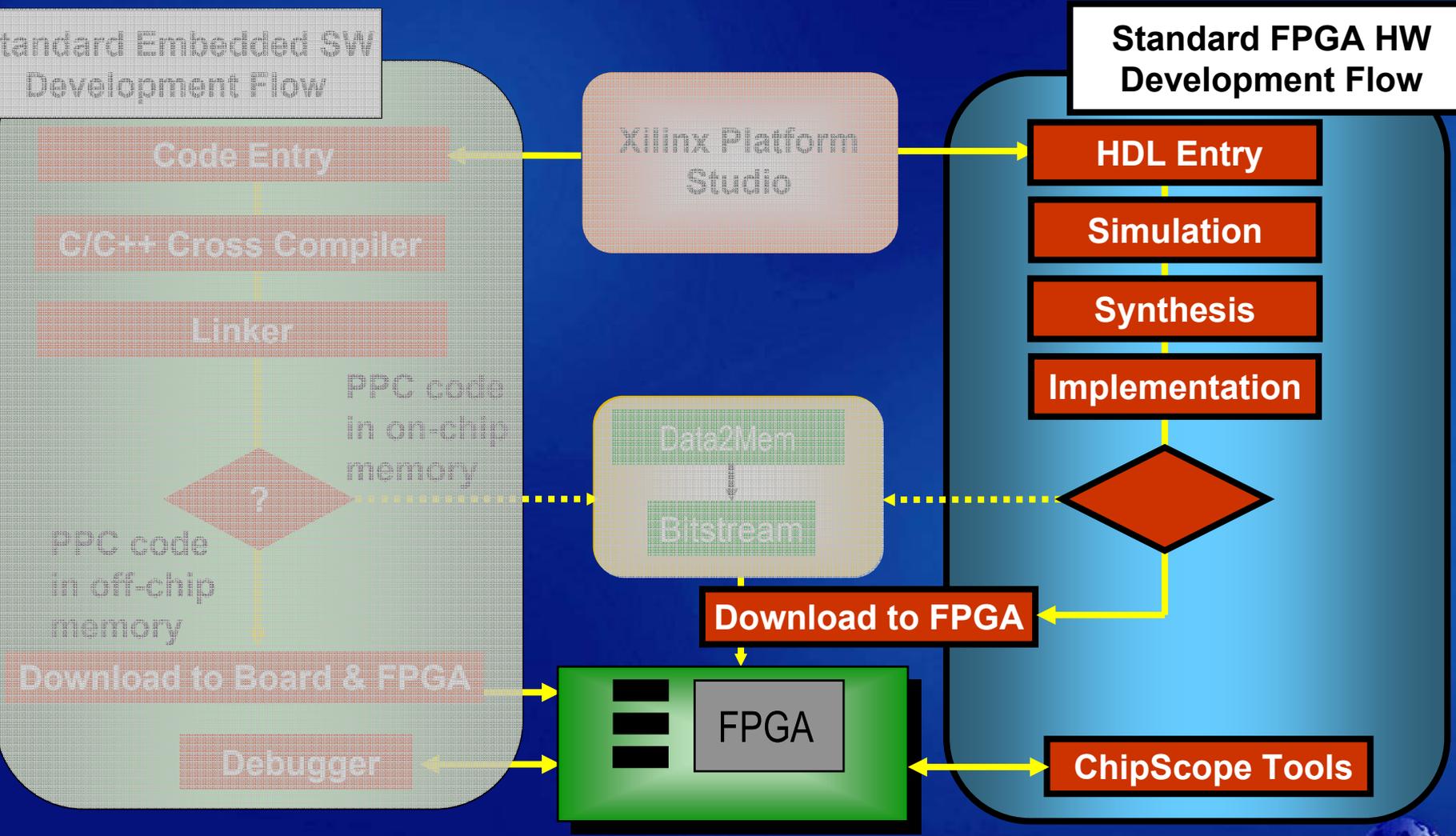
- New:
 - **APU module**
 - **Two EMAC cores per PPC405**
- Enhancements:
 - More flexible OCM controllers
 - Higher PLB performance
 - Reduced area
 - Improved performance
- Fully compatible with Virtex-II Pro processor

Agenda

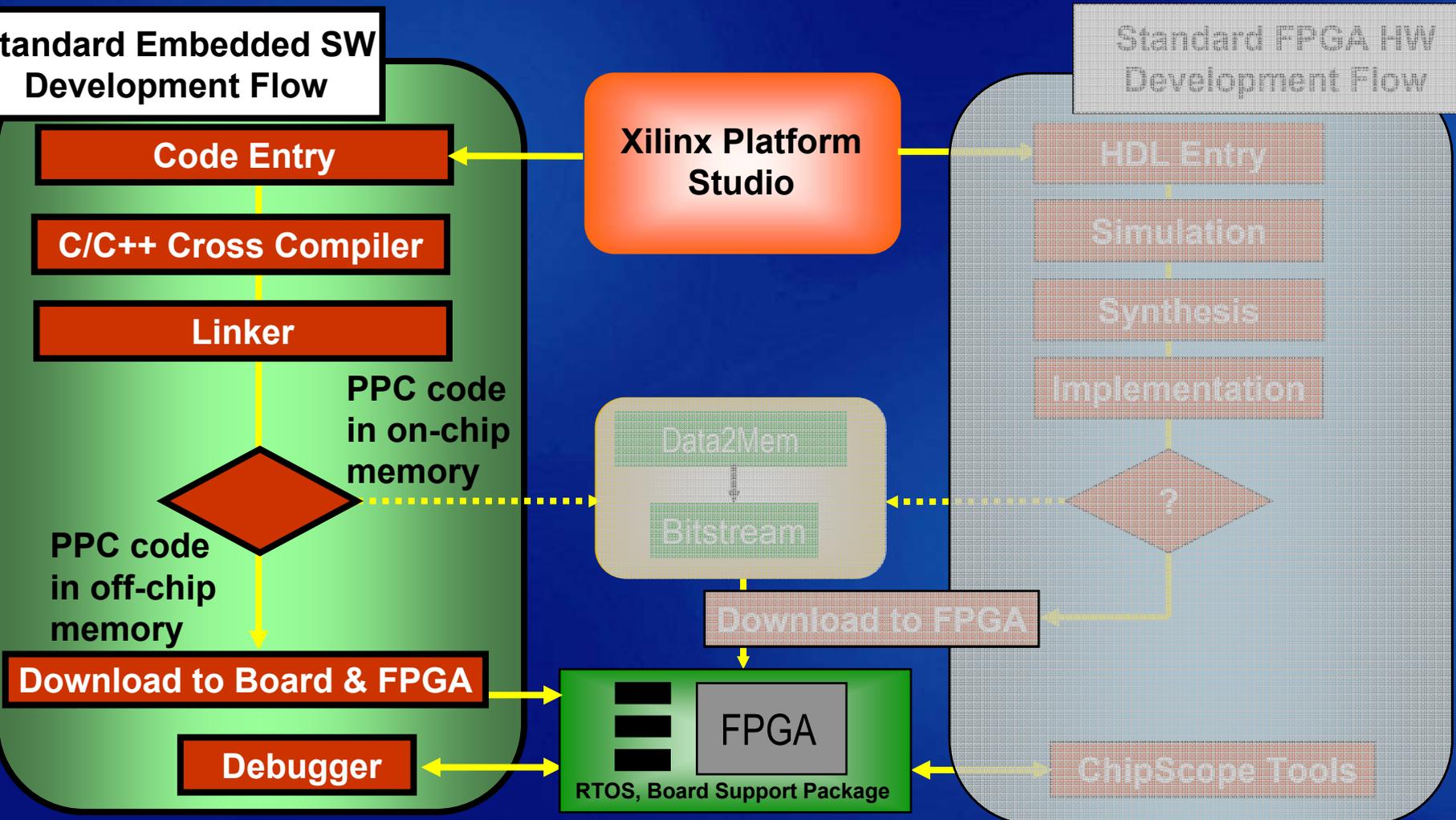
- Virtex-IIpro architectural overview.
- Virtex-4 architectural overview.
- **Software flow.**
- Configuration, frames and Bits.
- Partial re-configuration.
- Embedded partial re-configuration.
- Partial re-configuration flow.



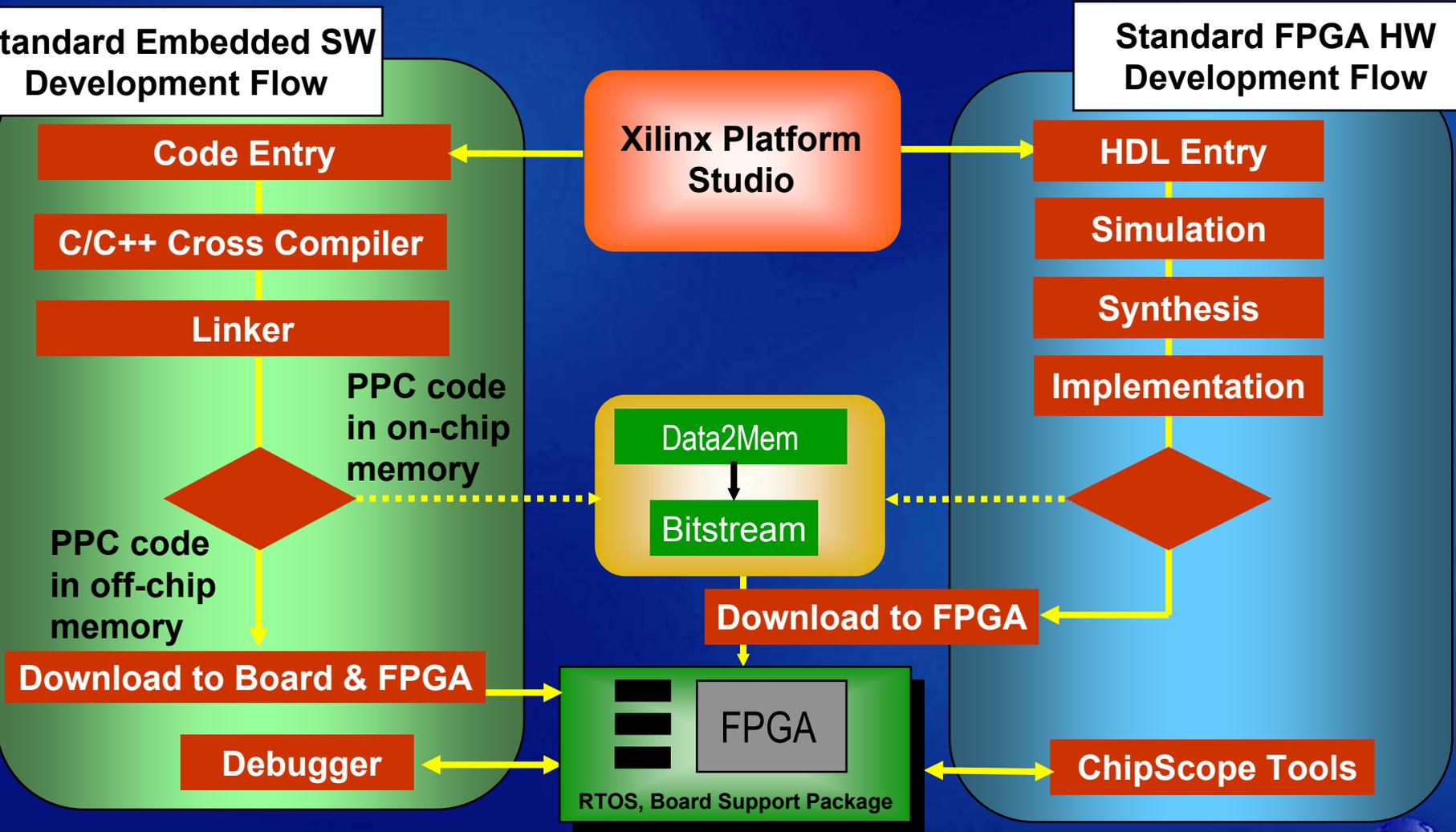
HW Development Tools



The Embedded SW Flow



SW & HW Developed Concurrently



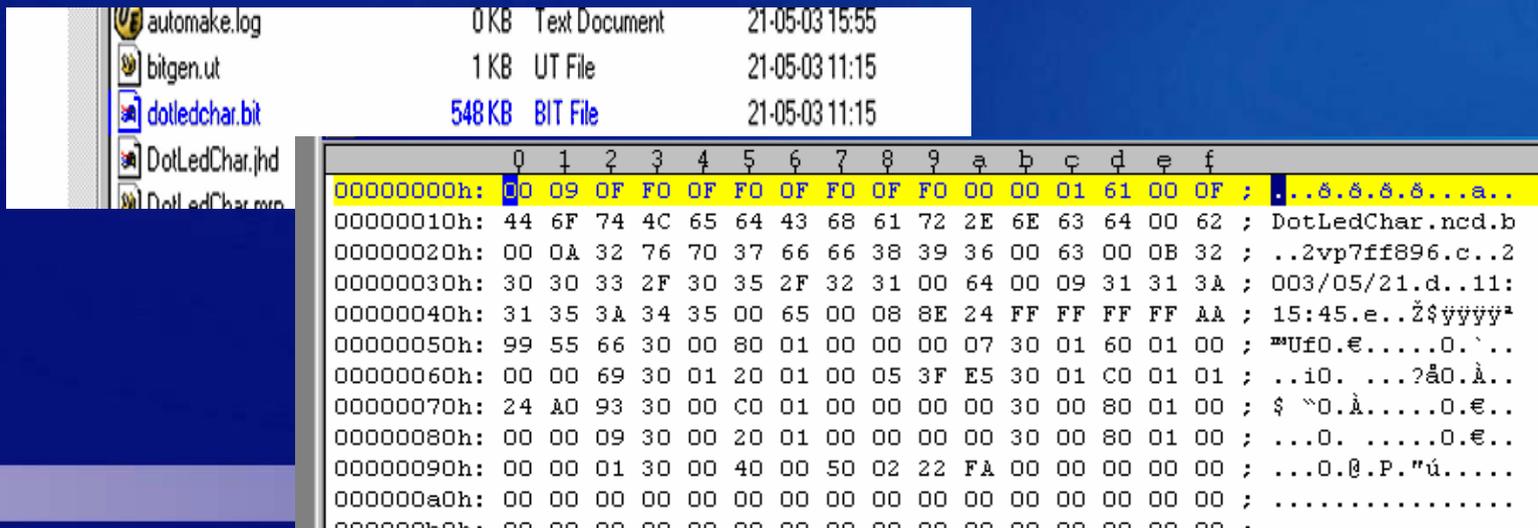
Agenda

- Virtex-IIpro architectural overview.
- Virtex-4 architectural overview.
- Software flow.
- Configuration, frames and Bits.
- Partial re-configuration.
- Embedded partial re-configuration.
- Partial re-configuration flow.



Configuration

- FPGA bits are stored in a file called: **Bitfile**.
- It's containing the info needed to program the transistors of the FPGA so that they obey the correct way and perform to the intended function.
- Generated by the ISE software.



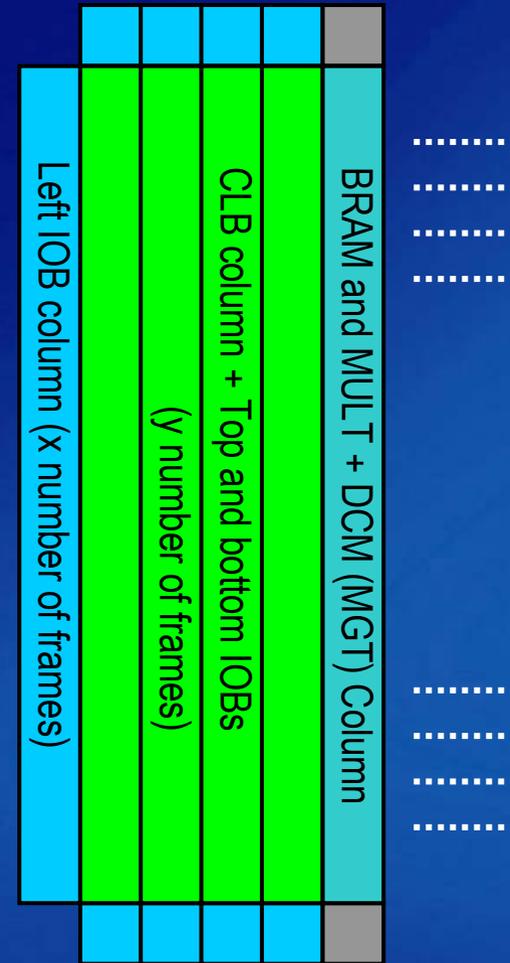
Frames and bits

- Bitstream is organized as a set of Frames.
 - The length and number of frames depends on the type of FPGA.
- A number of frames describes a column.
- A column performs a specific FPGA function.
 - There are columns for:
 - Block SelectRAM
 - CLB / IOB
 - DCM
 - RocketIO



Frames

- Virtex-IIpro frames are column reaching from bottom to top of the FPGA.
- Virtex-4 frames are columns that span “8 CLB + 1 HCLK + 8CLB” or equivalent logic.
 - 2 RAMB + 1 HCLK + 2RAMB
 - 16 IOB + 1 HCLK + 16 IOB
 - The granularity is greater than in V2pro.
- Columns do not take in account the special logic as JTAG, ICAP, etc.



Virtex-IIpro example

Agenda

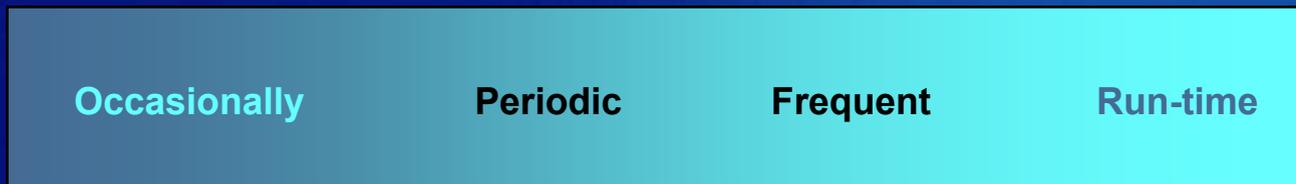
- Virtex-IIpro architectural overview.
- Virtex-4 architectural overview.
- Software flow.
- Frames and Bits.
- **Partial re-configuration.**
- Embedded partial re-configuration.
- Partial re-configuration flow.



Why use it?

- Opens a totally new set of design applications.
 - Adaptive hardware algorithms.
 - Reduced device count and cost.
 - New business models possible.
 - ASICs cannot compete in this arena.
 - Allows things that are impossible using non-reconfigurable silicon.

Spectrum of Reconfiguration



Field Upgrades

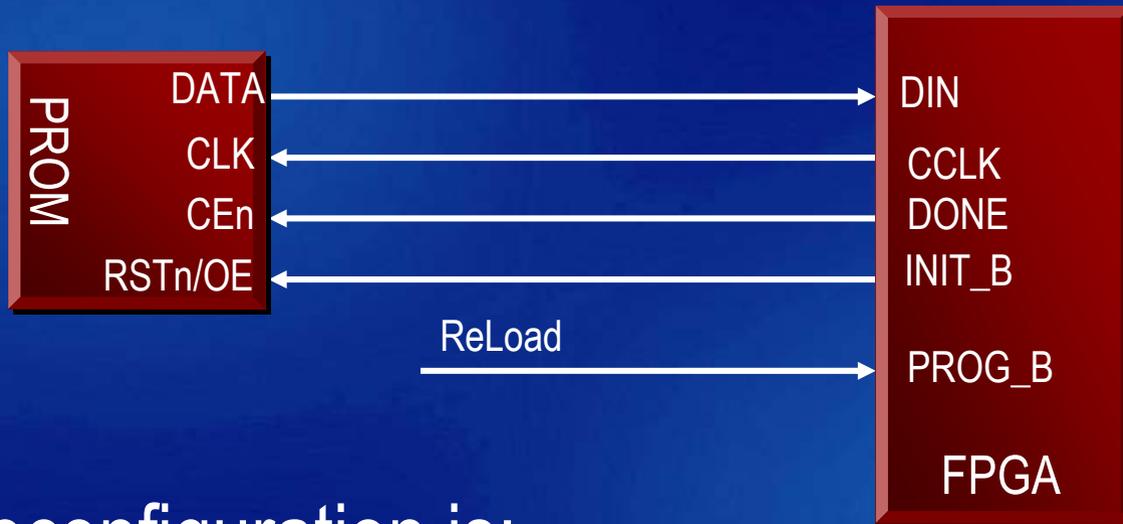
*Rapid Design/Data
Processing*

*Mobile Applications
Network Appliance*



(re)configuration

- Each FPGA design is using re-configuration.



- Partial reconfiguration is:
 - Modify parts of hardware while the rest keeps running.
 - Here some special techniques must be applied.

How can it be done?

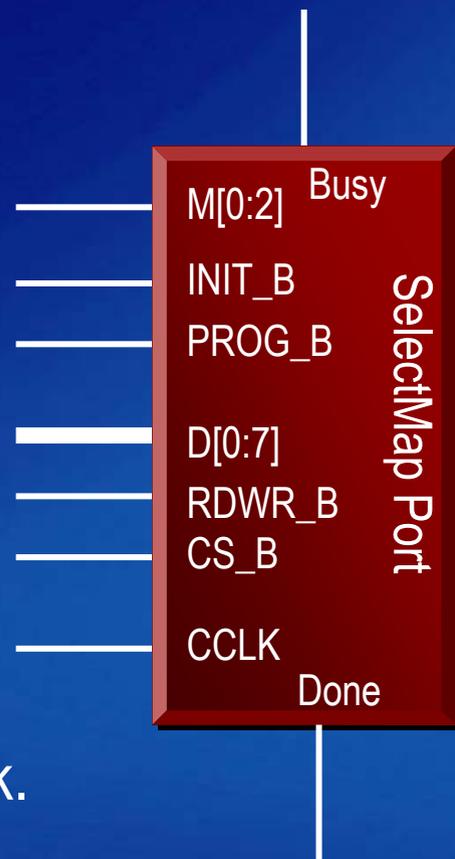
- Reconfigure the FPGA, complete or partial, per an external connected device as a processor.
 - A connected processor can use the JTAG or SelectMap port.
- Reconfigure the FPGA, partial, using an on board processor.
 - The PPC405 or Microblaze processor configures parts of the FPGA through a ICAP port.



SelectMap

- SelectMap

- Is an 8-bit bi-directional data bus interface.
- Can be used to configure the FPGA.
- When in “persist” mode can be used to re-configure the FPGA.
- Two modes exist:
 - Master SelectMap, FPGA generates a clock.
 - Slave SelectMap, FPGA accepts a clock.



“persist” is a BITGEN option

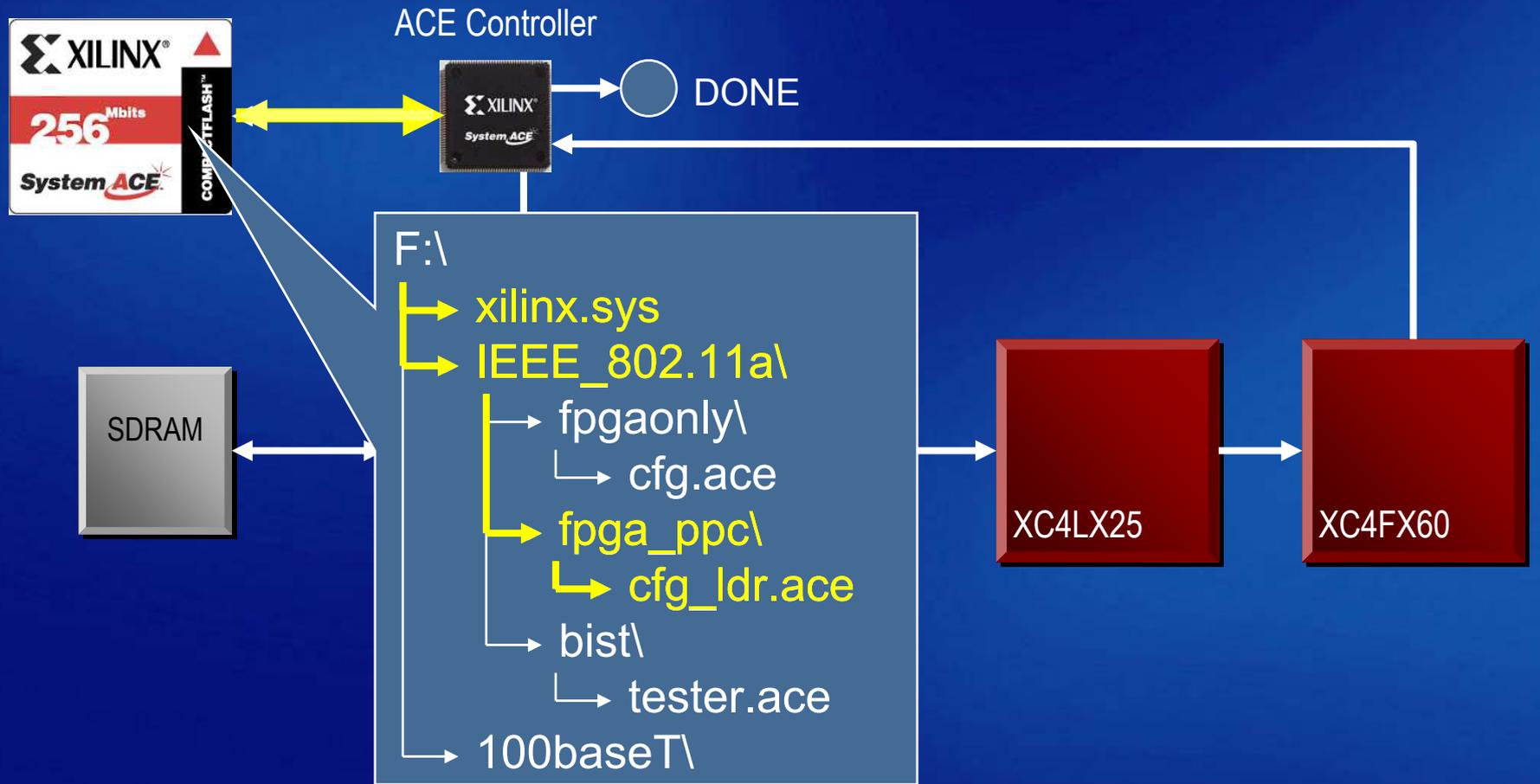
JTAG



- Fully compliant with IEEE-1149.1
- Commonly named: Test Access Port or Boundary-Scan.
- Data can be send out on I/O pins in order to test connections at board level.
- Can also be used to send signals internally to test the device-specific behavior.
- In addition to testing, the flexibility is offered for having a own set of user-defined instructions.
- The added common vendor-specific instructions, such as configure and verify, have increased the popularity of boundary-scan testing and functionality.

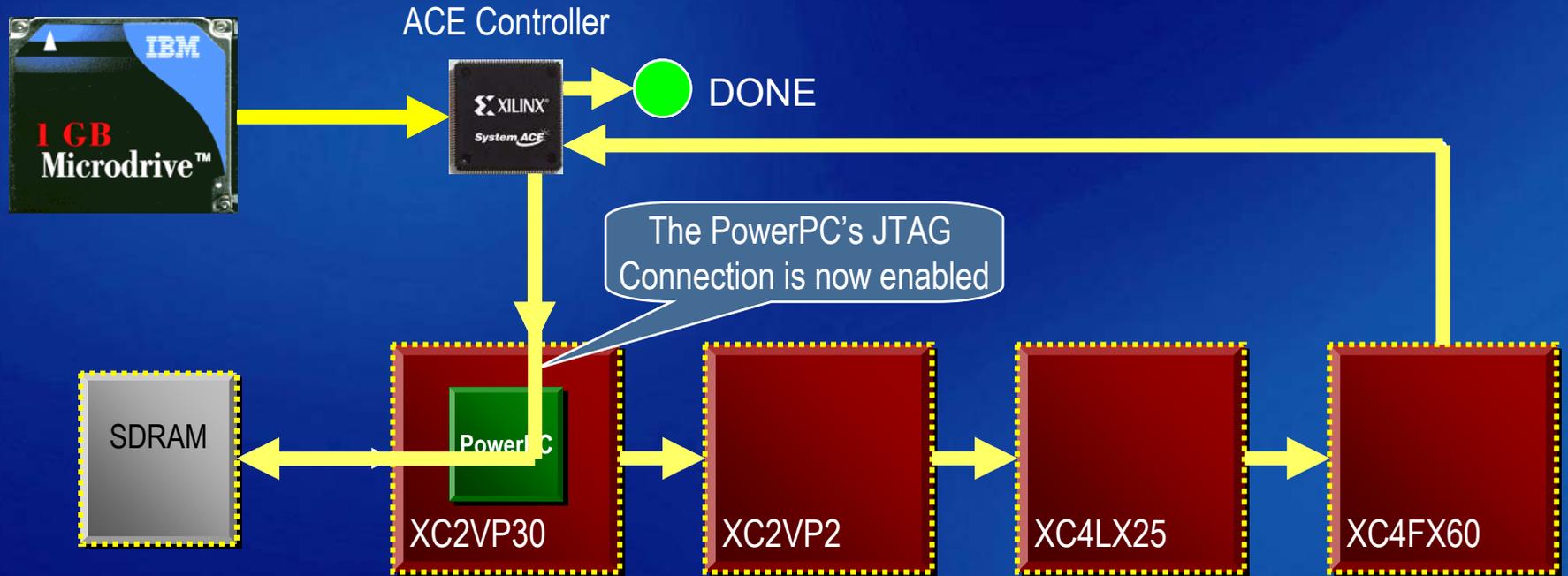


JTAG Example

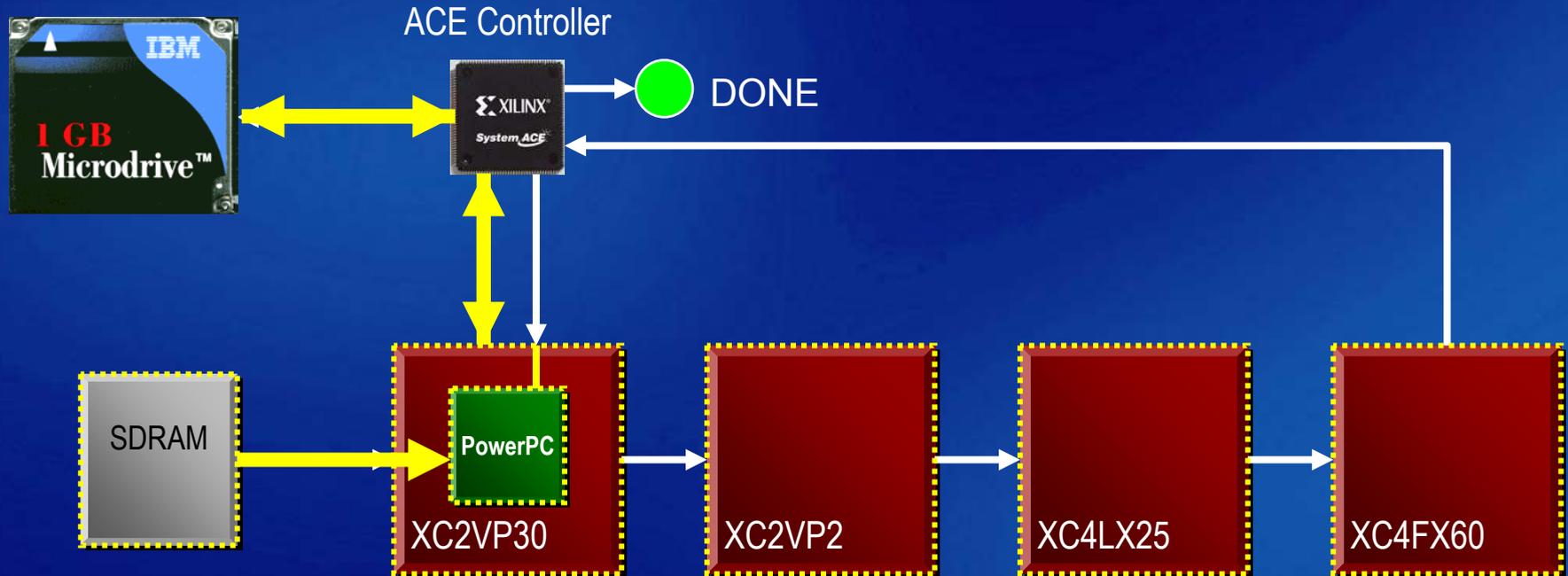


...Configure the FPGAs...

...Load PPC code in SDRAM...



Boot the PowerPC from SDRAM. Use the "Microdrive" as Hard Disk.



Agenda

- Virtex-IIpro architectural overview.
- Virtex-4 architectural overview.
- Normal Software flow.
- Frames and Bits.
- Partial re-configuration.
- **Embedded partial re-configuration.**
- Partial re-configuration flow.

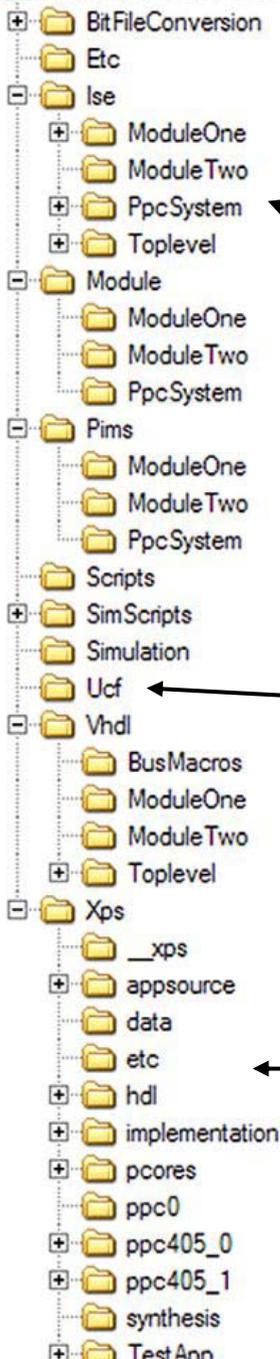


Initial Start

- Determine what the partial reconfiguration design will look like:
 - Separate modules, not talking to each other.
 - A design where modules swap data.
 - A combination of both.
- Determine the IO and IO location of each module.
- Determine the size of each module.
- Determine the PPC design.

- These steps are very important!

Project setup



ISE projects

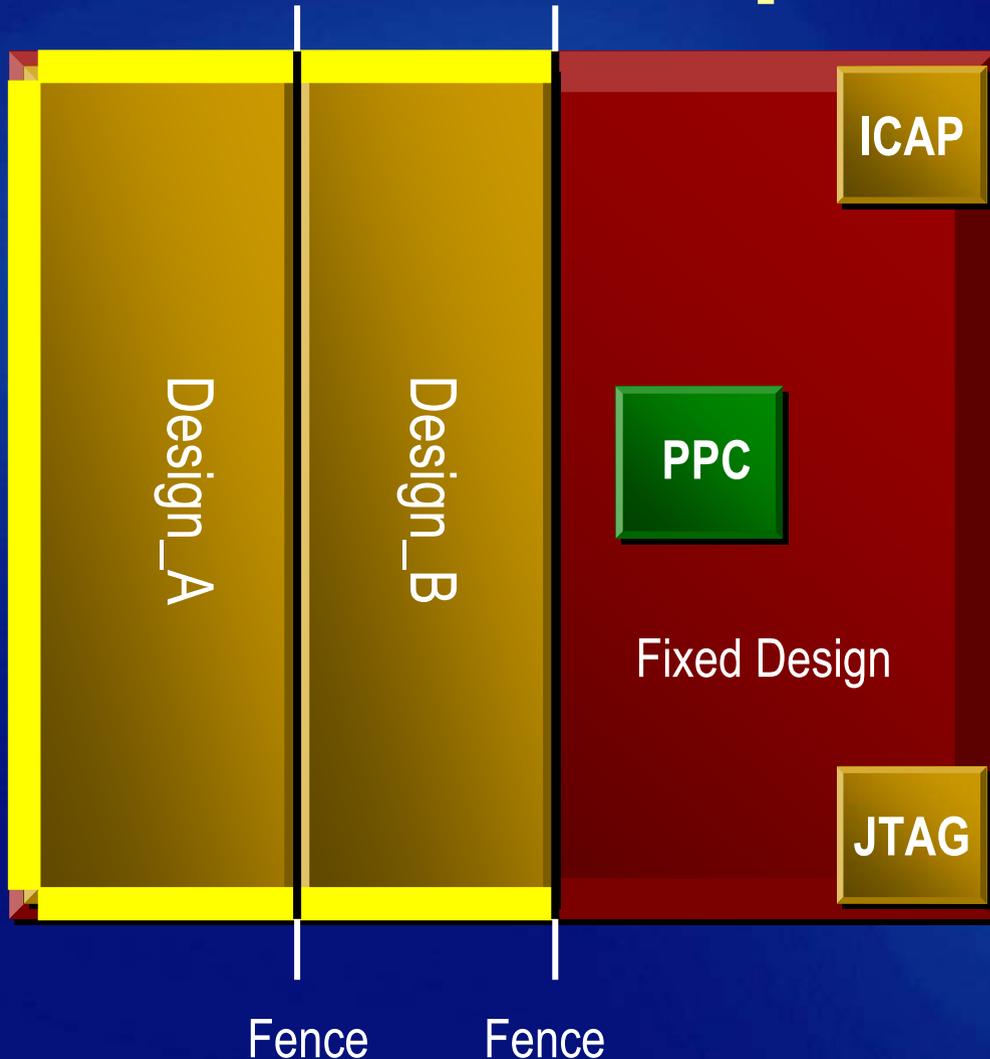
User Constraint Files

HDL source code

Xilinx Platform Studio
project for the processor
design.

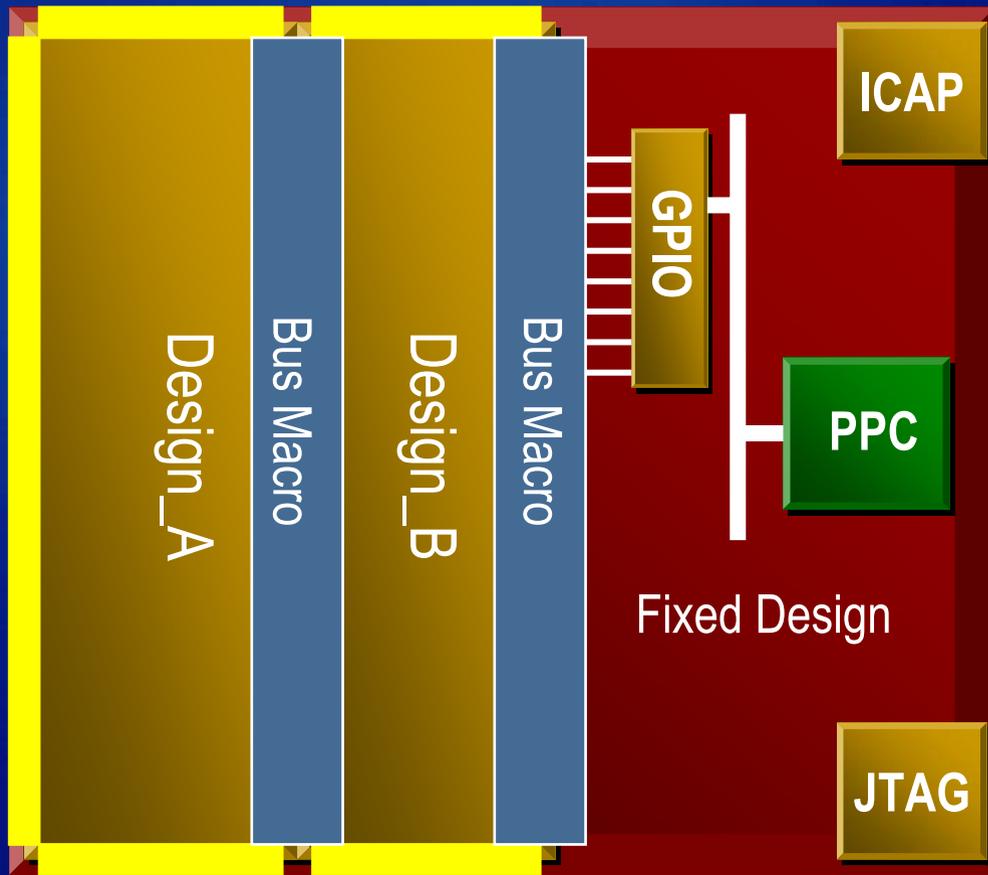
- A good start comes with a setup of a directory structure to use for the project running partial reconfiguration.

Initial planning 1



- Design Modules don't talk to each other.
- Design_A and Design_B are partial reconfigurable areas.
- Complete initial design gets loaded through JTAG.
- When needed PPC gets a bitstream from memory and re-configures area A or / and area B.

Initial Planning 2



- Modules talk to each other and to the PPC.
- PPC is in command of reconfiguration.
- Enables and disables the bus-macro's to allow reconfiguration.
- Bus-macro is a set of tri-state buffers locked in a fixed place.

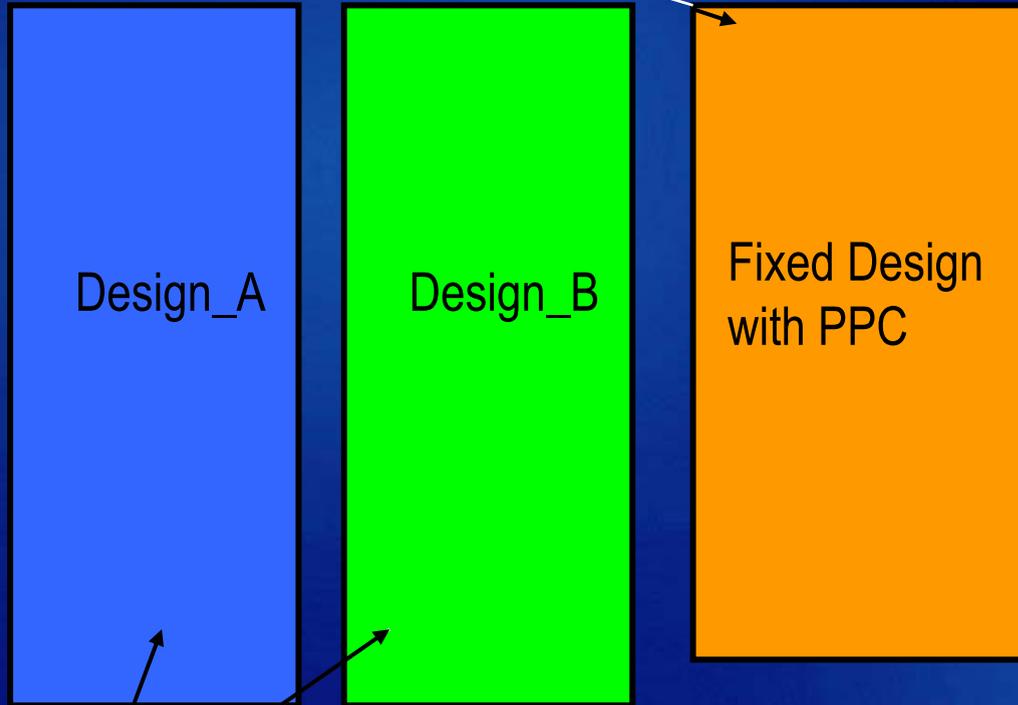
Agenda

- Virtex-IIpro architectural overview.
- Virtex-4 architectural overview.
- Software flow.
- Frames and Bits.
- Partial re-configuration.
- Embedded partial re-configuration.
- Partial re-configuration flow.



Hardware Design

Processor design
done in XPS



Ex: Pure HDL design

- Toplevel design contains hierarchical levels of sub-level designs as black boxes.
- Synthesize and translate the Toplevel.
- Floorplan the Toplevel.
- Write a UCF file.

- Design each hierarchical block separately.
- Use the UCF file for each block and add specific constraints.



Floorplan Guidelines

- A module must always be the full height of the device.
- Horizontal placement is on a four slice boundary.
- Clocks are not part of reconfigurable modules.
- IOB's directly above or below a module are part of that module.
- For left and right oriented modules the same applies.
- Minimize the number of reconfigurable modules.
- Modules communicate through a Xilinx Bus-Macro.
- Make sure that the fixed design does NOT relate on a state of the reconfigurable module.

Virtex-IIpro example



Reference slide 1

- Provide an empty Toplevel.ucf file in /Ucf.
- Setup a ISE project in the /Ise directory.
 - Call it “Toplevel” (creates the project in a directory Toplevel).

Translate Properties	
Property Name	Value
Use LOC Constraints	<input checked="" type="checkbox"/>
Netlist Translation Type	Timestamp
Macro Search Path	
Create I/O Pads from Ports	<input type="checkbox"/>
Allow Unexpanded Blocks	<input type="checkbox"/>
User Rules File for Netlister Launcher	
Allow Unmatched LOC Constraints	<input type="checkbox"/>
Preserve Hierarchy on Sub Module	<input checked="" type="checkbox"/>
Other Ngnbuild Command Line Options	-modular initial

!

The screenshot shows the ISE Project Navigator with the following structure:

- Sources in Project:
 - Toplevel
 - xc2vp20-6ff896
 - toplevel-toplevel_arch (..\..\Whd\Toplevel\toplevel.vhd)
 - ..\..\Ucf\Toplevel.ucf
 - moduleone
 - ppcsystem

- Processes for Source: "toplevel-toplevel_arch"
- Create Area Constraints
- Edit Constraints (Text)
- Synthesize - XST
 - View Synthesis Report
 - View RTL Schematic
 - Check Syntax
- Implement Design
 - Translate
 - Translation Report
 - Floorplan Design (highlighted with a red arrow)
 - Generate Post-Translate Simulation Model
 - Post-Translate Simulation Model Report
 - Assign Package Pins Post-Translate
 - Map
 - Map Report

Reference slide 2

- Toplevel UCF File

```
INST "RecoModule" AREA_GROUP = "AG_RecoModule" ;  
INST "PpcSystem" AREA_GROUP = "AG_PpcSystem" ;  
.....
```

```
AREA_GROUP "AG_RecoModule" RANGE = SLICE_X0Y111:SLICE_X7Y0,  
TBUF_X0Y111:TBUF_X6Y0,RAMB16_X0Y0:RAMB16_X0Y13,MULT18X18_X0Y0:  
MULT18X18_X0Y13 ;
```

```
.....  
NET "Reset" LOC = "AL6";  
NET "Led3" LOC = "AK28";  
NET "Led2" LOC = "AK29";  
.....
```

Floorplanner writes for each sort of device a separate
`AREA_GROUP` line in the UCF file.

Put everything in one `AREA_GROUP` text line as
shown here else problems during PAR will pop-up.

```
.....  
INST "SysBufg" LOC = "BUFGMUX7P";  
INST "SysDcm" LOC = "DCM_X1Y0";  
INST "SysDcm" CLKIN_PERIOD=10;  
INST "SysBufg" LOC = "BUFGMUX3S";
```



Reference slide 3

Module UCF file.

- Renamed copy of the Toplevel.UCF file.
- Timing constraints added for the module in cause.
- Module needs some extra constraints for Partial Reconfiguration:
 - INST "Module_Name" AREA_GROUP = "Grouped_Name";
 - AREA_GROUP "Grouped_Name" RANGE = SLICE_X0Y111:SLICE_X7Y0,
.....
 - AREA_GROUP "Grouped_Name" MODE = RECONFIG;
 - AREA_GROUP "Grouped_Name" GROUP = CLOSED;
 - AREA_GROUP "Grouped_Name" PLACE = CLOSED;
- These constraints push logic into boundaries set by the Area_Group ... Range.

Constraints from Toplevel design, set by the floorplanning stage.



Reference slide 4

- Synthesis of each module with XST tool.
- ISE used on each module for XST run.

Property Name	Value
Optimization Goal	Speed
Optimization Effort	Normal
Synthesis Constraints File	
Use Synthesis Constraints File	<input checked="" type="checkbox"/>
Library Search Order	
Keep Hierarchy	Yes
Global Optimization Goal	AllClockNets
Generate RTL Schematic	Yes
Read Cores	<input checked="" type="checkbox"/>
Cores Search Directories	E:\Projects\Modular_Partial\PartialPpc\PartialPpcNoBusMacro
Write Timing Constraints	<input type="checkbox"/>
Cross Clock Analysis	<input type="checkbox"/>
Hierarchy Separator	/
Bus Delimiter	<>
Slice Utilization Ratio	100
Case	Maintain
Work Directory	./xst
HDL INI File	
Verilog 2001	<input checked="" type="checkbox"/>
Verilog Include Directories	
Custom Compile File List	
Other XST Command Line Options	

Property Name	Value
Add I/O Buffers	<input type="checkbox"/>
Max Fanout	500
Number of Clock Buffers	16
Register Duplication	<input checked="" type="checkbox"/>
Equivalent Register Removal	<input checked="" type="checkbox"/>
Register Balancing	No
Move First Flip-Flop Stage	N/A
Move Last Flip-Flop Stage	N/A
Pack I/O Registers into IOBs	Auto
Slice Packing	<input checked="" type="checkbox"/>
Convert Tristates To Logic	No
Optimize Instantiated Primitives	<input type="checkbox"/>

Keep Hierarchy, YES

Add I/O Buffers, OFF

• XPS:

- Set to write the .npl file to the \lse\PpcSystem directory when exporting the design.
- Run XST from ISE tools.

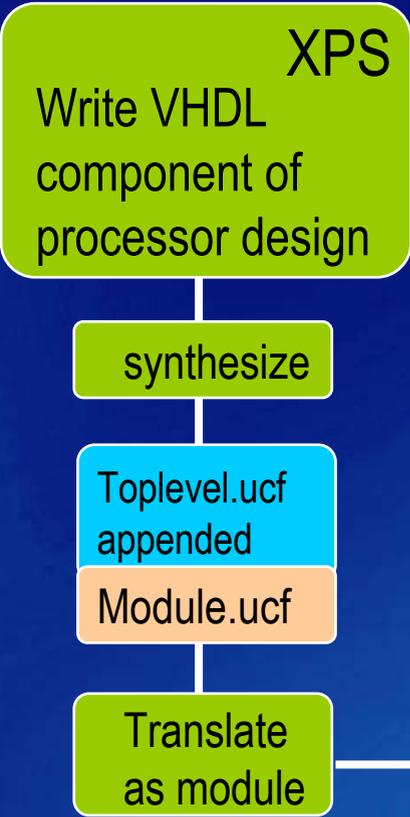
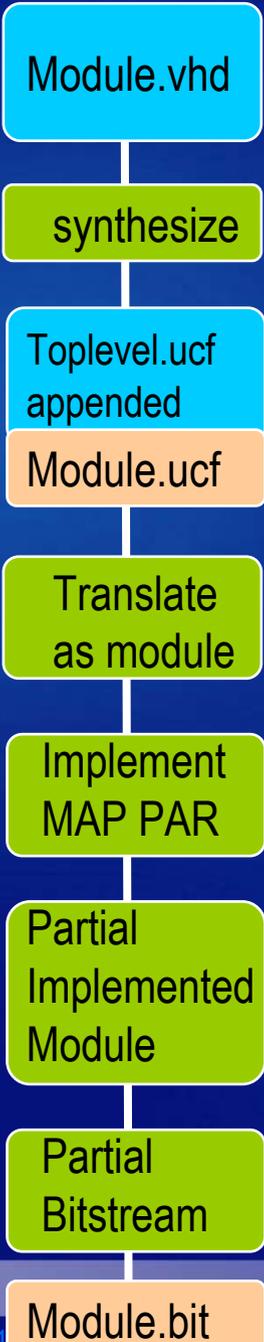
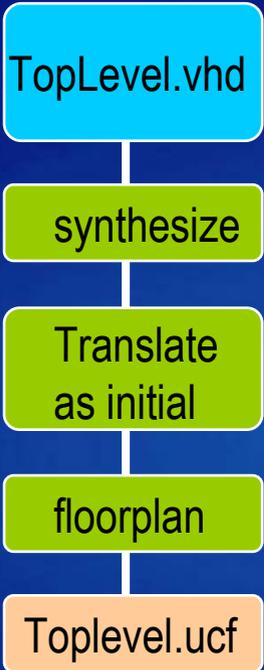


Module Design

- Design each module as if it was a stand alone design.
 - This is the easiest for debug and simulation .
 - Apply constraint to meet the requested timing.
- As a module can be seen as a standalone design it can have hierarchical levels.
- Make sure no connections exist with other modules.
 - If connections are needed us a Bus-Macro.
- Use XPS to generate the processor design.
 - XPS must be set to generate a hierarchical level of a top level design.
- Generate for each module a partial bitstream.
- Convert the bitstream to a “C” source code “array”.

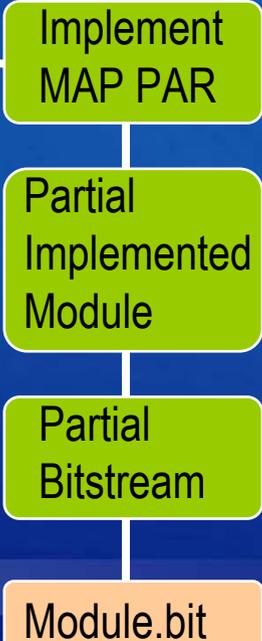
XPS = Xilinx Platform Studio





Flow

In most case a partial bitstream for the processor design is not needed because it's going to be the fixed element of the design.

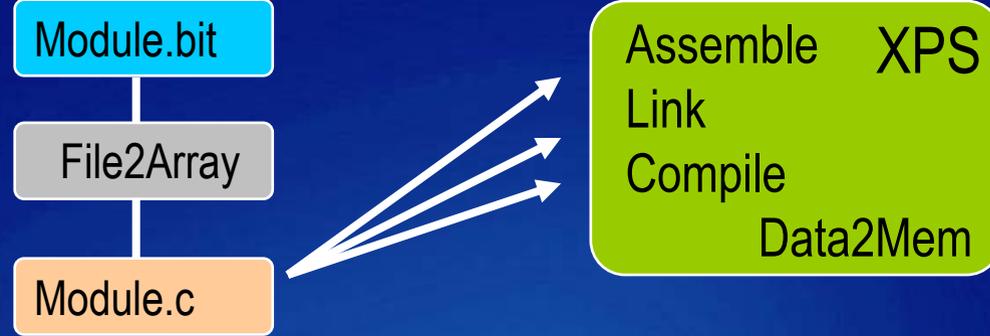


Run this flow for every module in the design creating 'x' number of partial bitstreams.

- Input file(s)
- Output file(s)
- Xilinx Programs
- Cust. programs

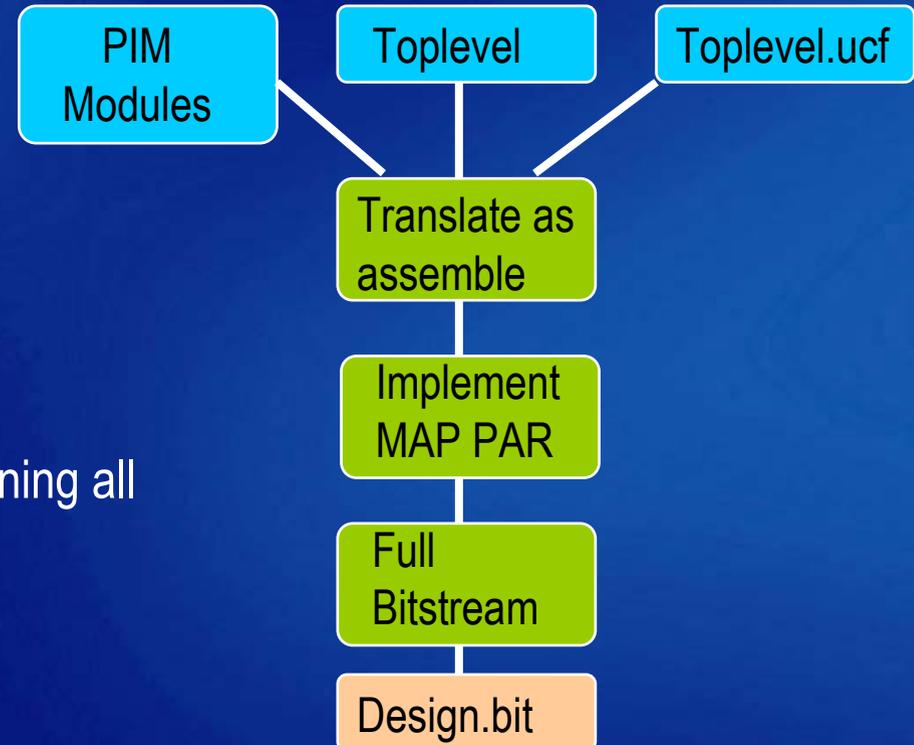


Flow



Run this flow for every module in the design creating 'x' number of partial bitstreams.

Create a full design that will be loaded as initial design.



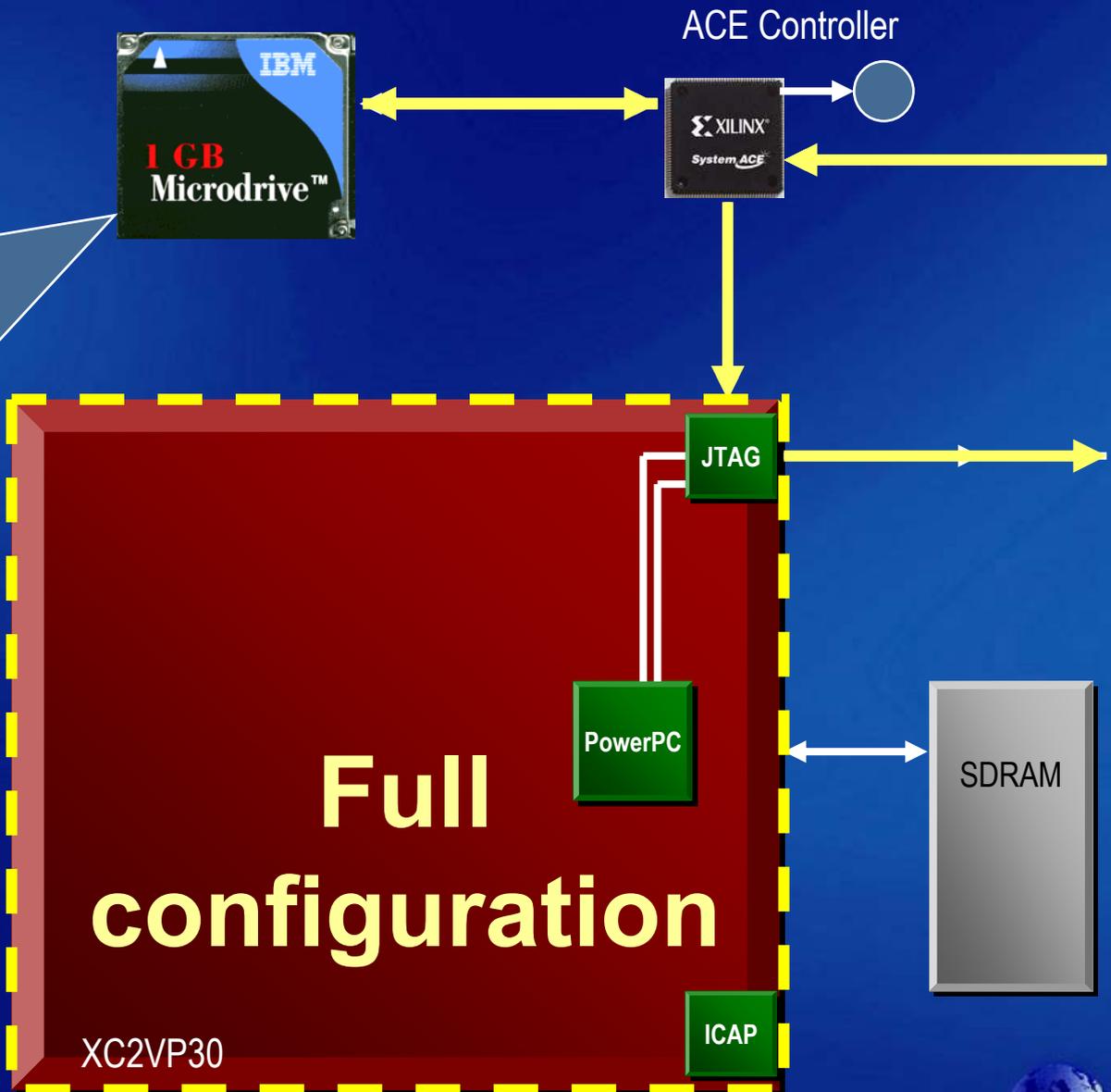
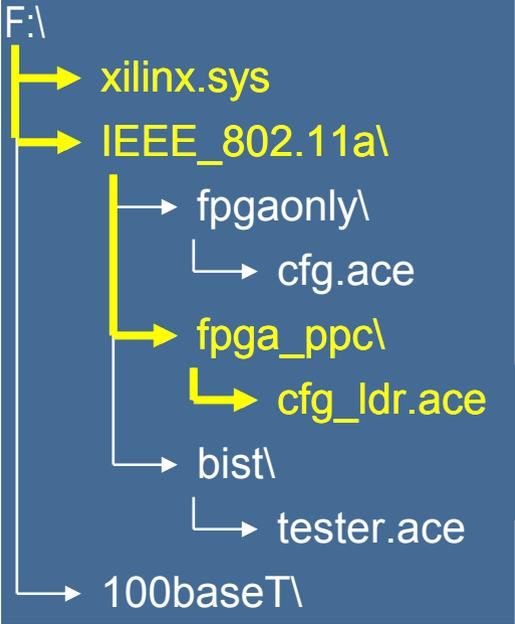
What do we have?

A "design.bit" for initial download containing all "Module.bit" files.

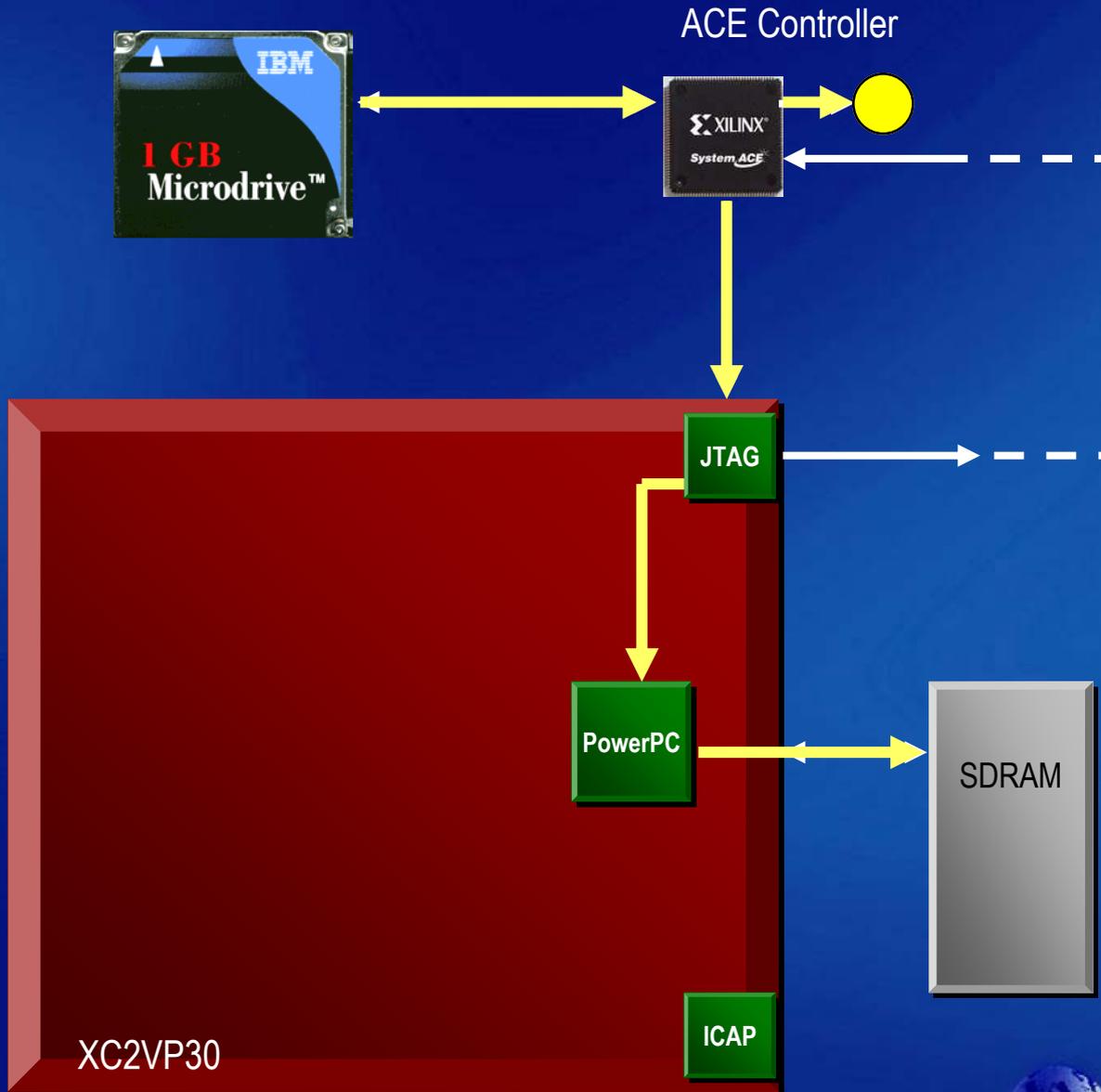
They are part of the "C" source code.

What do we do now?



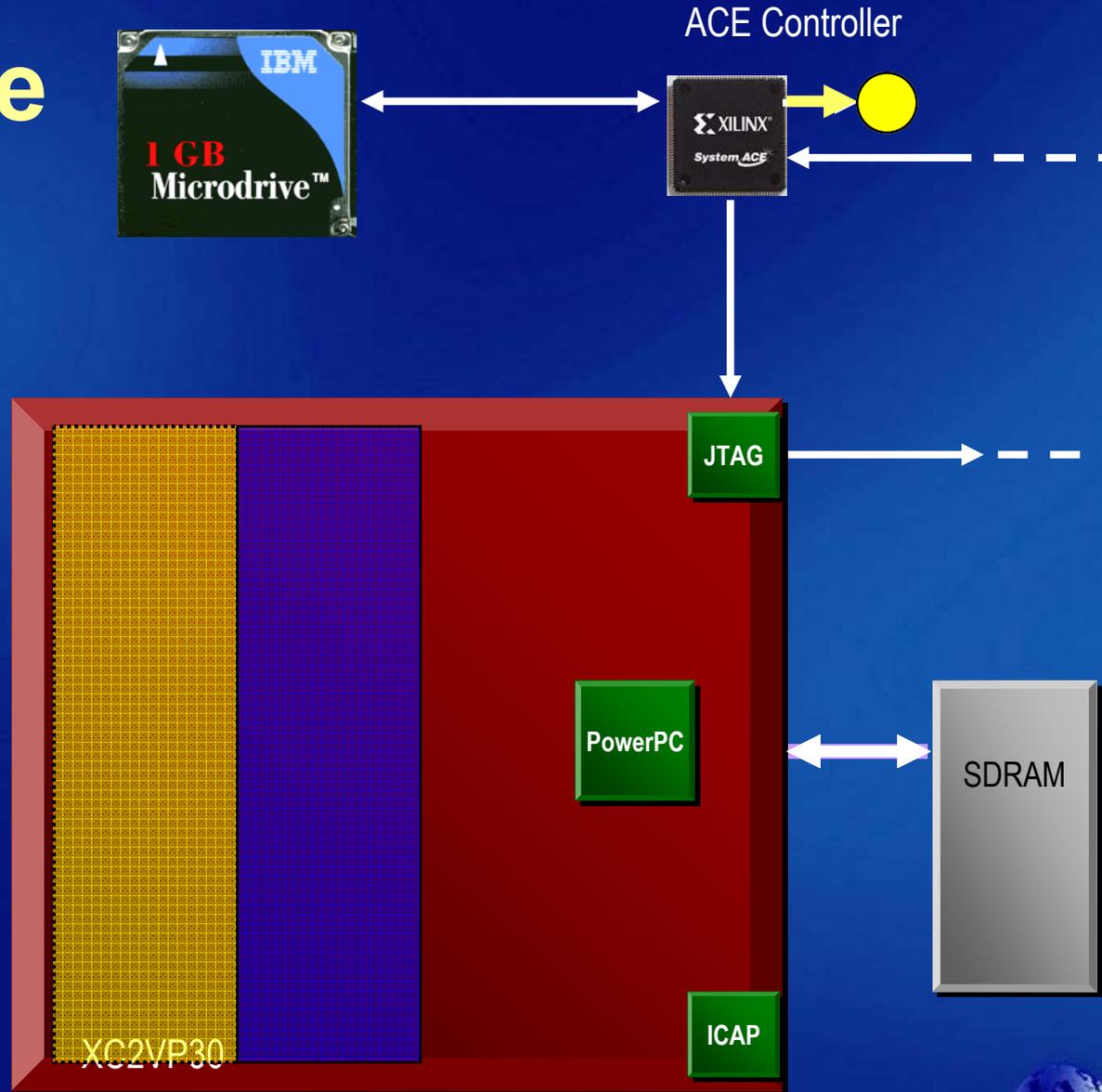


Download the processor program



Reconfigure on the fly

Configuration A
Configuration B
Configuration C

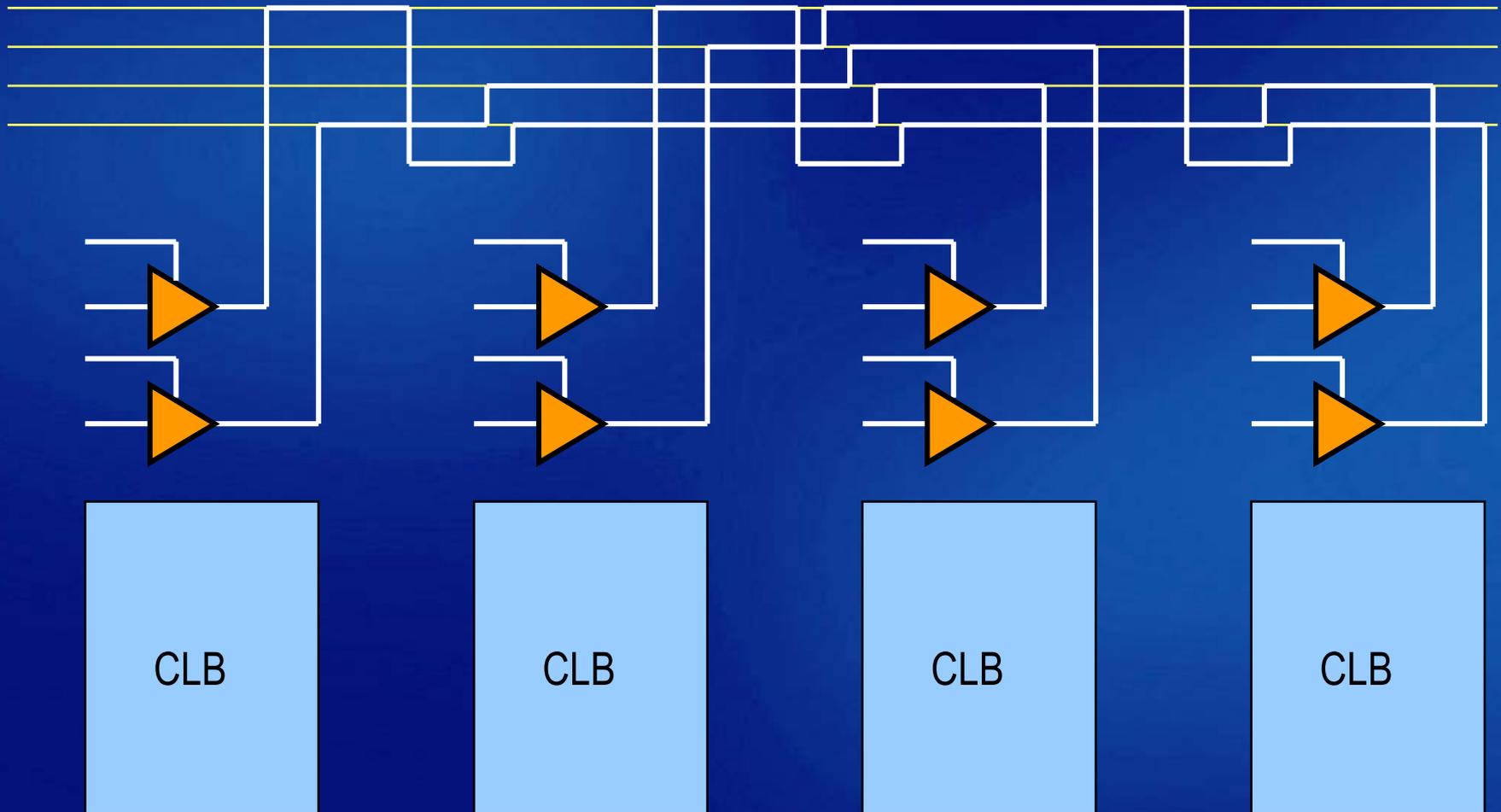


Reconfigurable Design with Bus-Macro

- Previous design switched on the fly partial FPGA designs that ran independently from each other.
- NO Bus-Macro was used
- When modules swap data, a Bus-Macro is used.
- What is a Bus-Macro?
 - A pre-routed, pre-placed set of tri-state buffers.
 - The Bus-Macro is built in FPGA_editor.
 - FPGA_Editor is the assembler of the FPGA, is you like.
 - Bus-Macro file extension is: .nmc

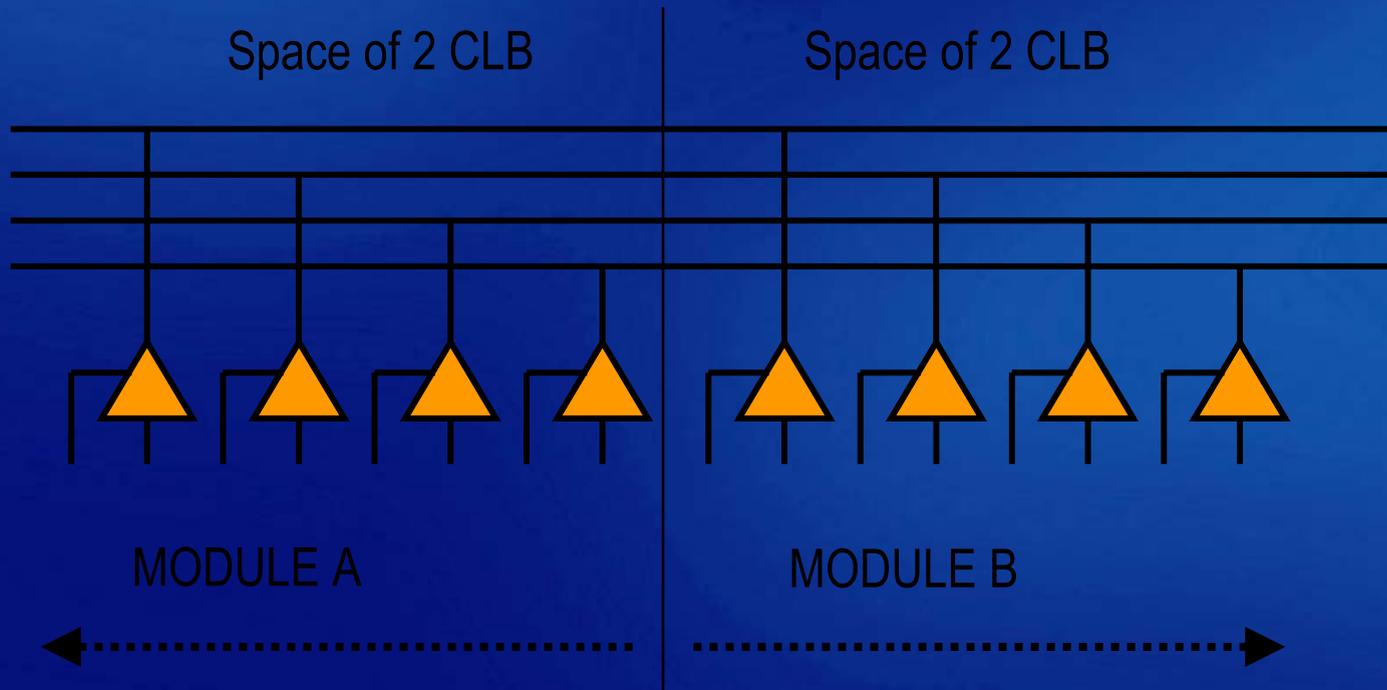


Basic Bus Macro



Use of Bus Macro

- Must be locked.
 - UCF syntax: `INST "busmacro" LOC = "TBUF_X8Y110";`
 - LOC one TBUF of the Bus-Macro to loc the whole macro.

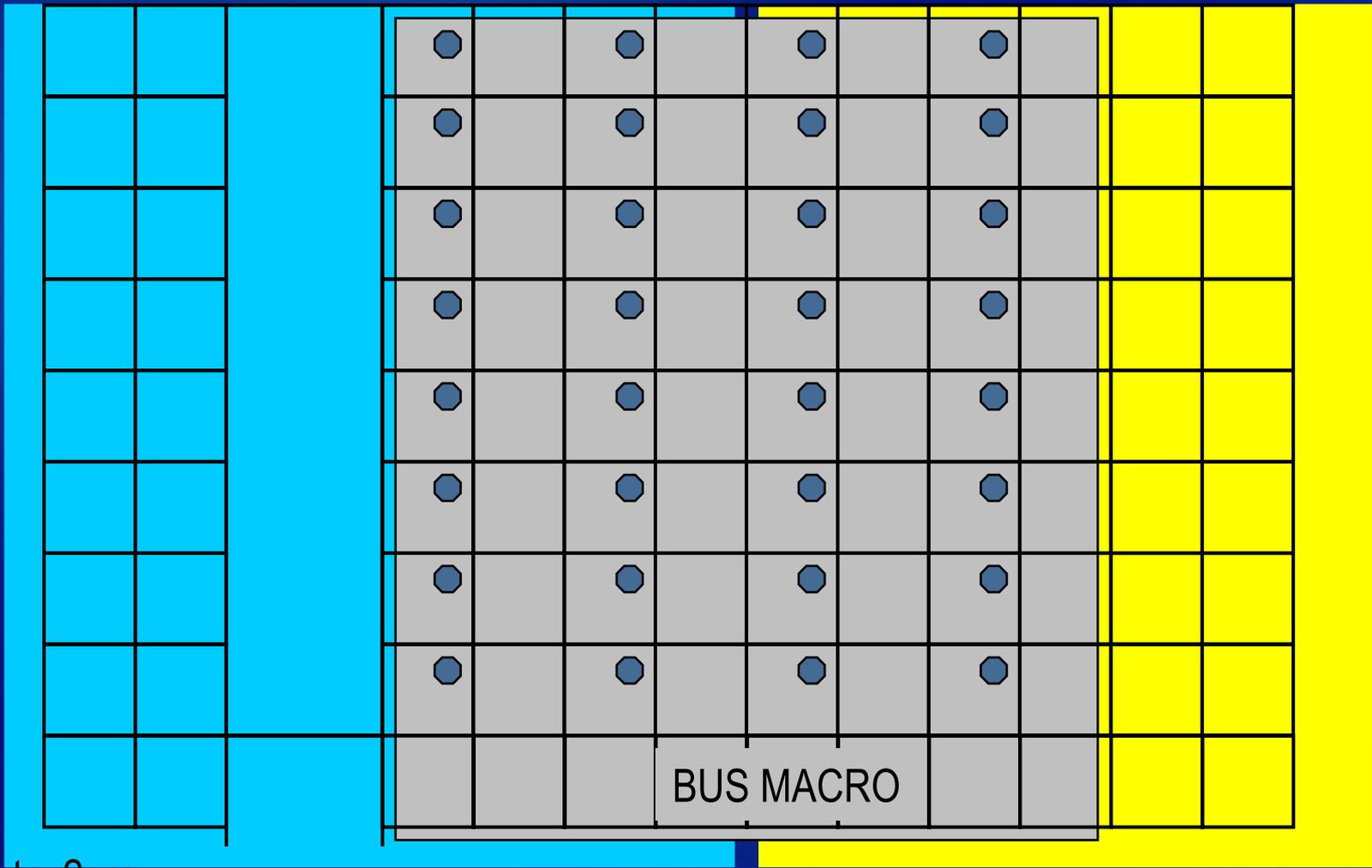


Use of Bus Macro (continued)

- Toplevel design must contain the Bus Macro(s).
- Modules have dedicated input, output and tri-state ports for the Bus Macro.
- Module area overlap the Bus Macro locked area.
- Bus Macro is controlled from the fixed area per GPIO processor peripheral.
- When reconfigurable module sizes are known, a special pass through Bus Macro can be made.



Floorplan for Bus-Macro



Virtex2pro



The End

- That's All Folks!

