Universidad de Alcalá

ELECTRONICS DEPARTMENT

ADVANCED DIGITAL ELECTRONIC SYSTEMS

SoC DESIGN ON FPGAs

Raúl Mateos Gil

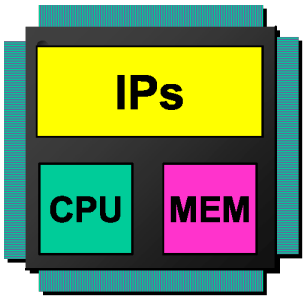
OFFICIAL MASTER IN ADVANCED ELECTRONIC SYSTEMS.
INTELLIGENT SYSTEMS

SoC DEFINITION

Some of these slides have been taken or adapted from:
"Fundamentals of SoC Design for FPGAs"
created by Sergio Lopez-Buedo

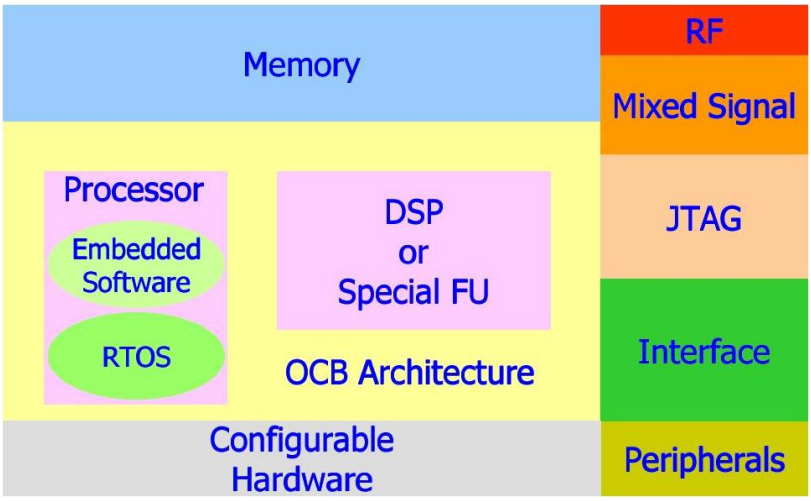
SoC DEFINITION

- A System on Chip or SoC is a complex IC that integrates the major functional elements of a complete end-product into a single chip or chipset. For FPGA devices:
  - CSoC = Configurable SoC
  - PSoC = Programmable SoC
  - SoPC = System on Programmable Chip
- The SoC design typically incorporates
  - Programmable processor
  - On-chip memory
  - HW accelerating function units (DSP)
  - Peripheral interfaces (GPIO and AMS blocks)
  - Embedded software

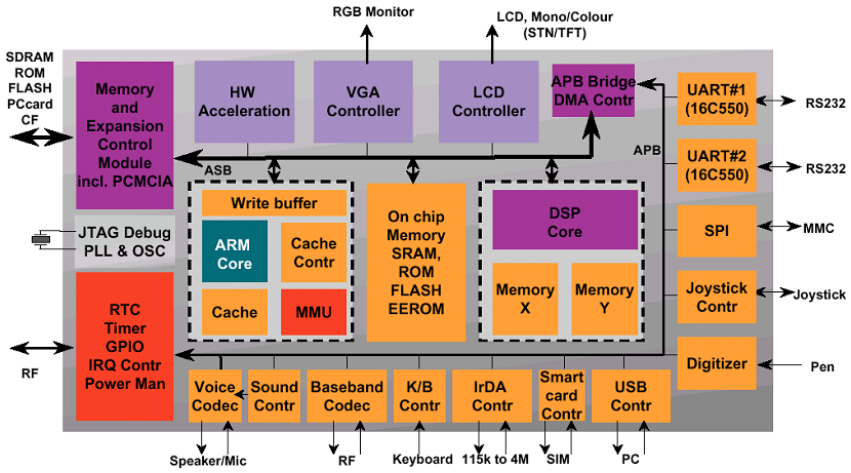


Source: "*Surviving the SoC revolution – A Guide to Platform-based Design*," Henry Chang et al, Kluwer Academic Publishers, 1999

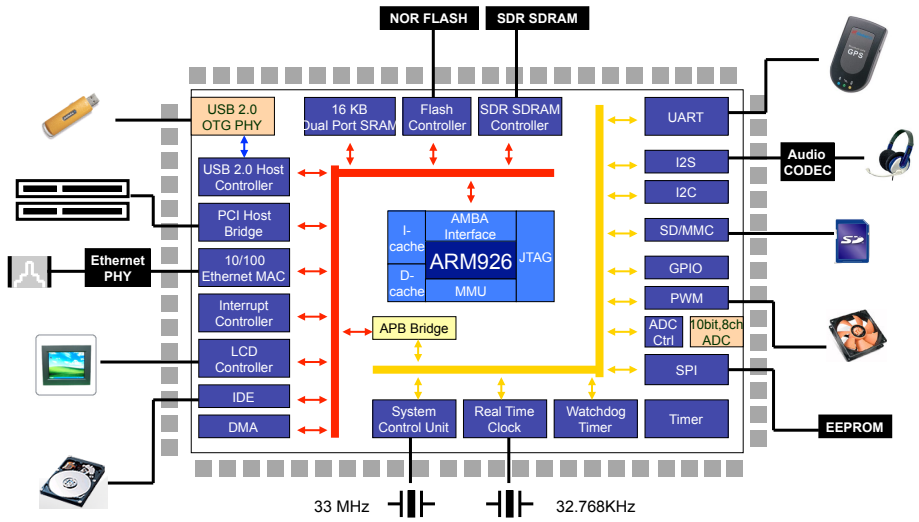
SoC BLOCK DIAGRAM



SoC EXAMPLE



SoC EXAMPLE



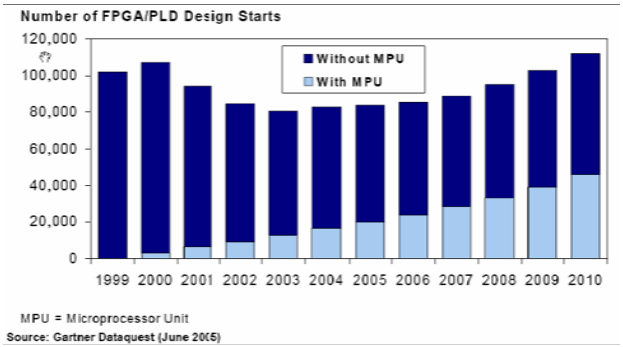
FPGA-BASED SoCs: FEATURES & RESTRICTIONS

- Complex, all-digital embedded systems based on 32-bit processors:
  - Several millions of equivalent gates
  - About 100-200 MIPS with soft processors, higher than 400 MIPS with hard processors
- Include enough memory for the boot ROM and communication buffers
  - Glueless interface to external DDR memory
- Include custom logic in your design
  - Greatly increase the performance with HW acceleration
  - Implement custom DSP algorithms

Electronics Department

FPGA-BASED SoCs: FEATURES & RESTRICTIONS

Universidad de Alcalá



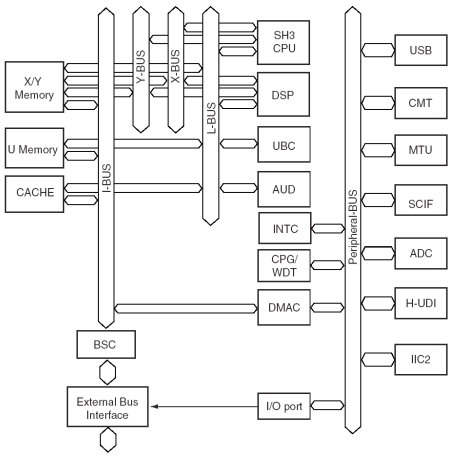
- To be real SoCs, they need also:
  - An external configuration memory, usually with a compact packaging
  - External program memory (Flash + DDR)
- Implement analog sections
  - The analog processing must be done outside, digitizing the signal as soon as possible (e.g. direct-conversion in RF)
- Become a low-power solution
  - FPGAs do consume a lot of power. You can have lower-power alternatives, but not a low-power solution (e.g. long battery life)

- Microcontroller
  - Find an off-the-shelf SoC that has all the components needed in your system
- ASIC
  - Create your custom silicon, usually using IPs (intellectual property) from various vendors
- Tandem systems: FPGA+MCU
  - Use a general-purpose programmable device to build your system. The chips are off-the-shelf (like MCUs) but you construct your own circuit using IPs (like ASICs)

- Nowadays, there are zillions of microcontroller options
  - Basically, they are SoCs composed by a processor and a lot of peripherals
- There are general purpose MCUs, but the most common are the MCUs for a specific application:
  - Motor control: TMS320C2812
  - Video or audio processing: TMS320DM335 Digital Media SoC
  - Automotive: ST Cartesio
- They are a very good option when they fit your needs:
  - Very good performance, like ASICs
  - Low unit cost, and the lowest NRE

- The have the lowest NRE because:
  - There are no expenses to design or manufacture the chip
  - They usually provide drivers to handle the peripherals
- Another advantage of MCUs is that you do not need to care about IPs
  - All royalties have been paid when you buy the chip
- But unfortunately, you always need something more:
  - A certain peripheral
  - Glue logic
- So they never end up being true SoCs

- **Example: Renesas SH7641**



ADC: A/D converter
AUD: Advanced user debugger
BSC: Bus state controller
CACHE: Cache memory
CMT: Compare match timer
CPG/WDT: Clock Pulse generator/Watch dog Timer
CPU: Central processing unit
DMAC: Direct memory access controller

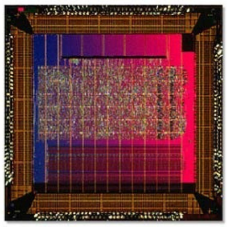
DSP: Digital signal processor
H-UDI: User debugging interface
INTC: Interrupt controller
SCIF: Serial communication interface
UBC: User break controller
MTU: Multi-Function Timer Pulse unit
USB : USB function module
IIC2: I^2C bus interface

STRUCTURED ASICs

- Though the unit cost is very low, ASICs are only worthwhile when they are produced in millions because of NREs
- An alternative with lower NREs are structured ASICs
  - Only 10,000 to 100,000 € NRE cost
  - Like gate arrays, but containing other components as large memory blocks
  - Only the upper metal layers are custom
  - Faster implementation times (few weeks)
- Some vendors are NEC, AMI, ChipX and LSI Logic

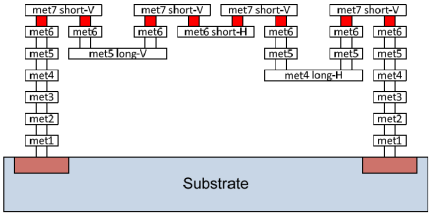
STRUCTURED ASICs

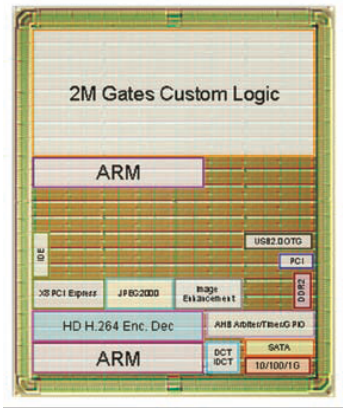
- Example: ChipX CX5000

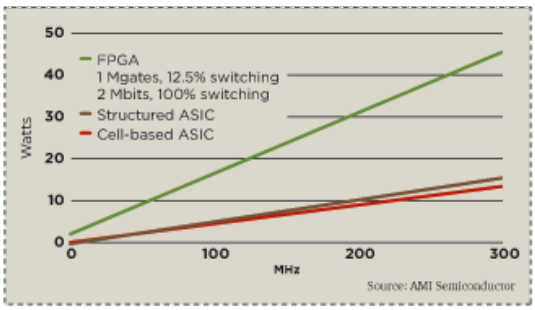


- 30K to 1.2M usable ASIC gates
- Up to 2.6M bits of fast block memory
- 2ns access time single-port SRAM, dual-port SRAM and ROM
- Low power consumption (0.06uW/MHz/Gate)
- 200MHz general core logic operation, up to 650MHz
- PCI, PCI-X, SSTL, HSTL, USB1.1, LVPECL and LVDS up to 622Mbps
- 1.5V or 1.8V or mixed supply voltage operation
- Up to 1100 total pads
- Low-jitter analog PLL macros with internal loop filter

Nextreme Zero Mask-Charge from eASIC





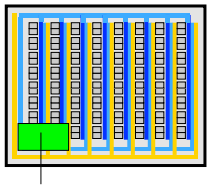


Considering all of the power-saving options, it is common for a structured ASIC design to use 20 to 50 percent less power than the same design implemented in an FPGA.

Vince Hopkin, FPGA-to-ASIC conversion a crucial concern, *EE Times 13 Sept. 2004*

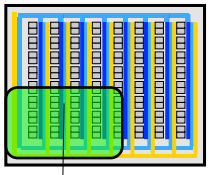
Two approaches:

Hardware Cores

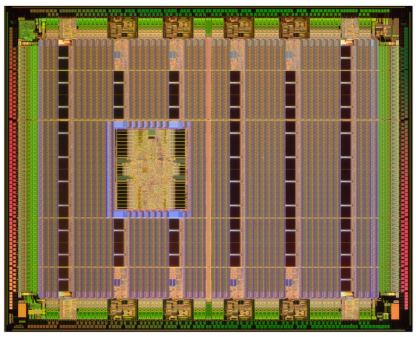


Replace FPGA region by uP hard core

Software Cores

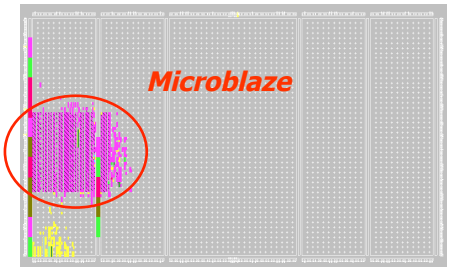


Use FPGA region to implement microprocessor core



- Hard processors are physically embedded in the FPGA silicon, as any other logic resource on it like LUTs, memories, etc…
  - No longer parameterizable designs
  - ASIC performance (speed and power)

SOFT PROCESSORS



- Soft processors come as parameterizable netlists that are implemented using the regular logic resources of the FPGA
  - Either concealed netlists implementing relatively placed macros or other highly optimized designs
  - Or plain synthesizable HDL code

INDUSTRY TRENDS

- Only Xilinx is betting hard on hard processors
  - PowerPC 405 on Virtex-II Pro and Virtex-4
  - Improved PowerPC 440 on Virtex-5
- Altera stopped supporting hard processors
  - Excalibur devices containing MIPS or ARM designs
- Few months ago Actel started with soft processors
  - 25 MHz ARM7 – That's a blazing speed!

WHEN A HW PROCESSOR MUST BE USED?

- High performance applications
  - Faster clock (2-3x) than soft processors
  - Better architectures (improved performance per MHz metric)
  - Bigger and more sophisticated cache memories
- Complex software and operating systems
  - Mature processor families, better toolchains
  - Memory management unit
- High added value applications
  - Because of higher price per FPGA unit
- Hard processors consume less power, although the overall power will depend on the rest of the FPGA

WHEN A HW PROCESSOR MUST BE USED?

- Usually the processor is not the only hard block available in the FPGA
  - Ethernet: For example, Xilinx Virtex-4 FX and Quicklogic QuickMIPS devices have both processor and Ethernet MAC cores
  - High Speed Serial I/O: Virtex-II Pro and Virtex-4 FX devices have high speed SERDES along with PowerPC processors
- Conclusion: Not only the existence of the hard processor must be considered, but also its synergy with other hard blocks

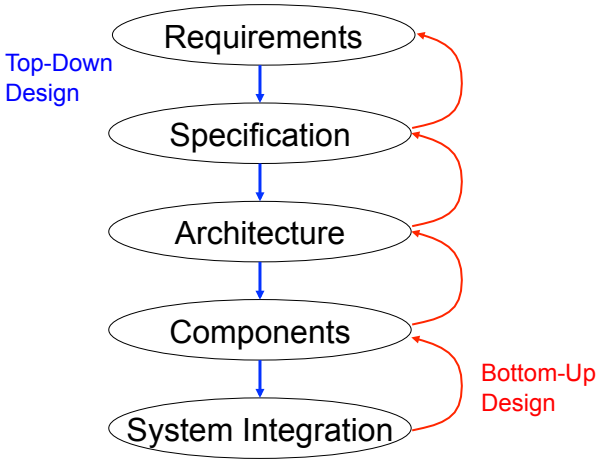
WHEN A SW PROCESSOR MUST BE USED?

- Processor configurability and upgradeability
- Cost-sensitive applications
  - Soft processors are the only possible solution in reduced cost devices such as Spartan-3 and Cyclone-II
- Applications where the processor is just a support
  - For example, a web-based management interface
- Regularity of FPGA fabric
  - Avoid big, heterogeneous blocks in the middle of the array
  - Useful, for example, in reconfiguration
- Benefits of open source designs
  - Modifications to the core, e.g. research or fault tolerance

DESIGN STYLES

- Design an embedded system is a challenging task:
  - Implement a lot of hardware components
  - Running complex software applications
- Three alternatives (design styles):
  - SoC design, when your competitive advantage is in the circuit design
  - Embedded software design, when your competitive advantage is in the software
  - Component based design, when you do not have the money to pay the design expenses, or the competitive advantage lies outside the electronics

CLASICAL DESIGN PROCESS



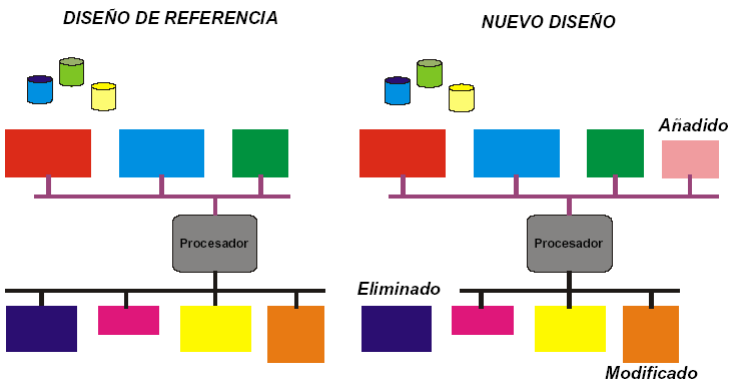
PLATFORM BASED DESIGN

- To ease the design of SoCs, the concept of platform-based design has recently appeared:
  - "A library of virtual components and an architectural framework, consisting of a set of integrated and pre-qualified software and hardware virtual components (VCs), models, EDA and software tools, libraries and methodology, to support rapid product development through architectural exploration, integration and verification." - *Virtual Socket Interface Alliance's (VSIA) platform-based design development working group (PBD DWG)*
- The idea is bring higher levels of abstraction to the design process, higher than IPs and drivers

PLATFORM BASED DESIGN

- The basic idea behind the platform-based design approach is to avoid designing a chip from scratch
- Some portion of the chip's architecture is predefined for a specific type of application:
  - There is a processor
  - A real-time operating system (RTOS)
  - Peripheral intellectual property (IP) blocks
  - Memory and a bus structure
- Users might customize it by adding hardware IP, programming FPGA logic or writing embedded software

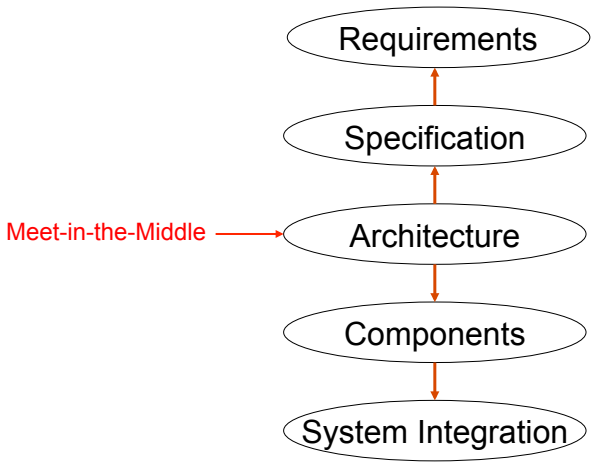
PLATFORM BASED DESIGN



MEET-IN-THE-MIDDLE DESIGN PROCESS

- It is neither a top-down nor a bottom-up methodology. It is a meet-in-the-middle design approach
- The starting point is a platform, that it is going to be customized:
  - Downwards when you add new peripherals or new HW
  - Upwards when you write a code that uses all the components of the platform to create your application
- It supports the embedded SW design style, because its final goal is to provide a SoC and OS that will the starting point for the SW development
  - It is not component based design, because this design style is just integration of components, also including SW components

ADVANTAGES OF PLATFORM-BASED DESIGN



ADVANTAGES OF PLATFORM BASED DESIGN

- The most important is the development time
  - It allows us to create a SoC in a few days
  - When designing with IPs, each component is solved but you still have to connect them
  - This is solved with platform-based design
- The implementation details of the SoC are abstracted, and you can center in the application development
  - All the technology-dependent details are hidden, so you only have to customize the reference platform
  - The development system will provide you the circuit and the libraries, OS and drivers needed to work with it

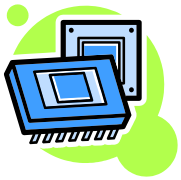
EXAMPLE: XILINX PLATFORM STUDIO

- Is a powerful tool to design FPGA based SoC platforms, based on MicroBlaze or PowerPC
- Graphical interface, the peripherals are automatically connected when you add them, and it has a push-button implementation interface
- Platform Studio also generates the SW libraries
- It also has an integrated compiler and a debugging system

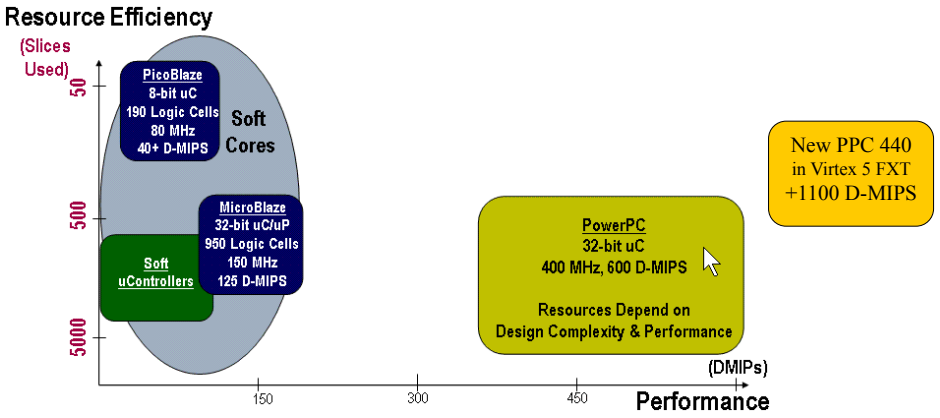
BUILDING BLOCKS OF A EMBEDDED SYSTEM

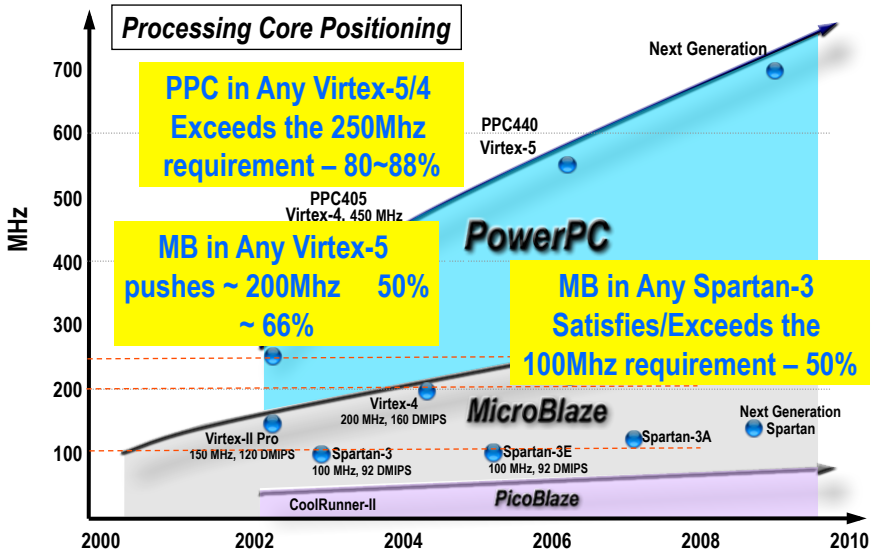
- PROCESSOR
- BUSES
- PERIPHERALS

PROCESSOR OPTIONS



- Commercial processors
  - Xilinx: MicroBlaze and PowerPC 405
  - Altera: Nios II
  - Actel: ARM7
  - Quicklogic: MIPS
  - Atmel: AVR
- Open-Source processors
  - LEON
  - OpenRISC
  - UltraSPARC T1
  - Many more…

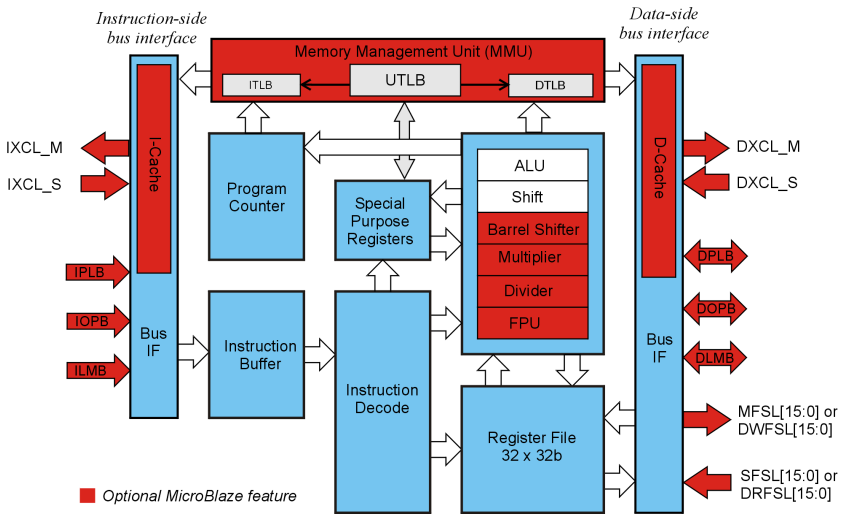

Resource Efficiency
(Slices Used)
50
500
5000
PicoBlaze
8-bit uC
190 Logic Cells
80 MHz
40+ D-MIPS
Soft Cores
MicroBlaze
32-bit uC/uP
950 Logic Cells
150 MHz
125 D-MIPS
Soft uControllers
PowerPC
32-bit uC
400 MHz, 600 D-MIPS
Resources Depend on Design Complexity & Performance
New PPC 440 in Virtex 5 FXT +1100 D-MIPS
150
300
450
(DMIPs)
Performance


Processing Core Positioning
PPC in Any Virtex-5/4 Exceeds the 250Mhz requirement – 80~88%
MB in Any Virtex-5 pushes ~ 200Mhz 50% ~ 66%
MB in Any Spartan-3 Satisfies/Exceeds the 100Mhz requirement – 50%
MHz
700
600
500
400
300
200
100
2000
2002
2004
2006
2008
2010
Next Generation
PPC440
Virtex-5
PPC405
Virtex-4, 450 MHz
PowerPC
Virtex-4
200 MHz, 160 DMIPS
MicroBlaze
Virtex-II Pro
150 MHz, 120 DMIPS
Spartan-3
100 MHz, 92 DMIPS
Spartan-3E
100 MHz, 92 DMIPS
Spartan-3A
Next Generation Spartan
CoolRunner-II
PicoBlaze

MICROBLAZE

- 32-bit RISC soft-core processor
  - 3/5-stage pipeline, single issue (MicroBlaze 7)
  - Runs at up to 100 (Spartan-3) or 200 (Virtex-4) MHz
- Configurable core
  - Multiplication, division, FPU, barrel shifter, etc… units can be added or not depending on the area/performance tradeoff
  - Different bus configurations, including coprocessing interface
  - Level-1 cache memory can be configured with different sizes
- Thrifty implementation
  - Around 1000 slices, less than $1 in Spartan-3
  - Optimized RPM for Xilinx FPGAs

MICROBLAZE BLOCK DIAGRAM



PowerPC 405

- Advanced 32-bit RISC hard-core processor
  - 32 GPRs plus many other registers and MAC instructions
- High performance
  - 5-stage pipeline, 450 MHz on Virtex-4 FX (400 in V2-Pro)
  - Better performance per MHz, up to 700 dhrystone MIPS
- Improved architectural features
  - 16 KB data and instruction cache memories
  - Simple, TLB-based memory management unit (MMU)
  - Support for user and privileged modes
- Includes 3 timers, one working as watchdog
- Coprocessing interface on Virtex-4 (APU)

PowerPC 405 BLOCK DIAGRAM

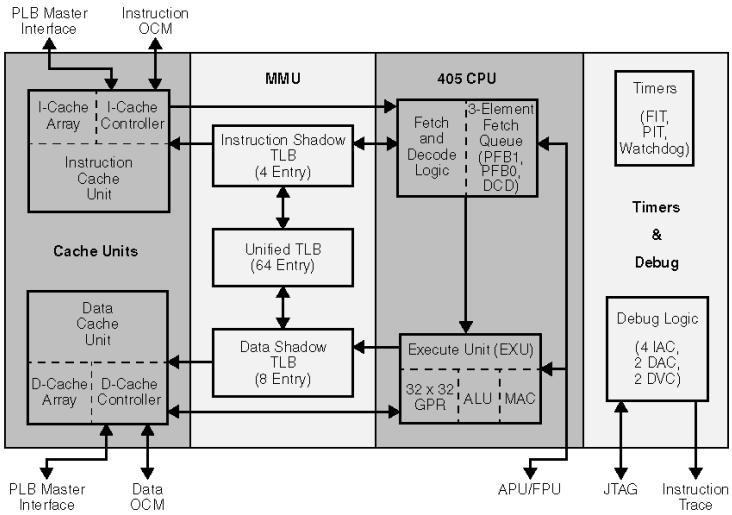
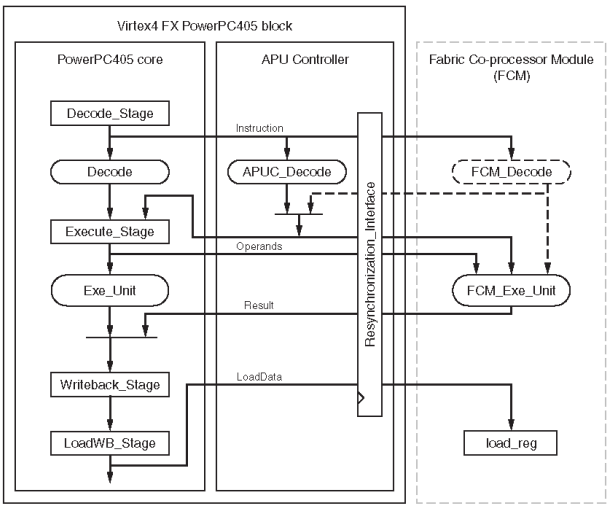


Figure 1-1. PPC405-S Block Diagram

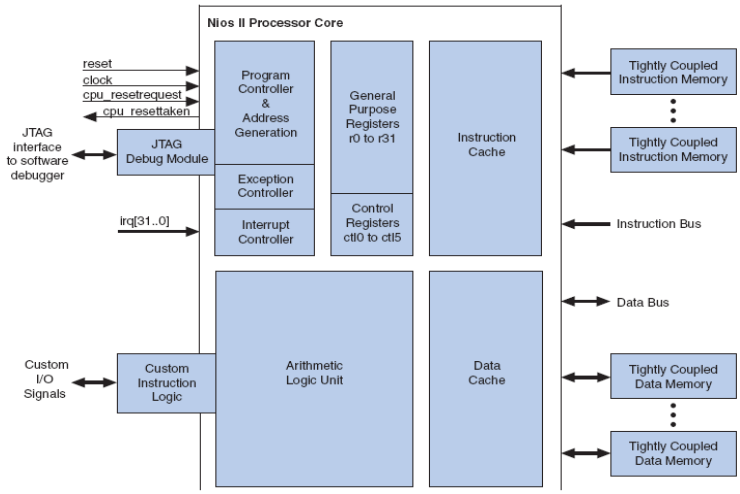
Auxiliary Processor Unit controller:

It extends the native PowerPC 405 instruction set with custom instructions that are executed by an FPGA Fabric Co-processor Module (FCM)



- 32-bit RISC soft-core processor
  - Similar alternative to MicroBlaze for Altera devices
  - But features a different, model-based approach to configurability
- Three models available
  - Fast, 6-stage pipeline, up to 185 MHz and 215 DMIPS, better performance per MHz than MicroBlaze, 1800 Les
  - Standard, 5-stage pipeline and 165 MHz, similar performance per MHz than MicroBlaze, 1600 Les
  - Economy, no pipeline, 700 LEs and 200 MHz, poor performance per MHz (30 DMIPS)
- •Support for up to 256 custom instructions

NIOS II BLOCK DIAGRAM



END