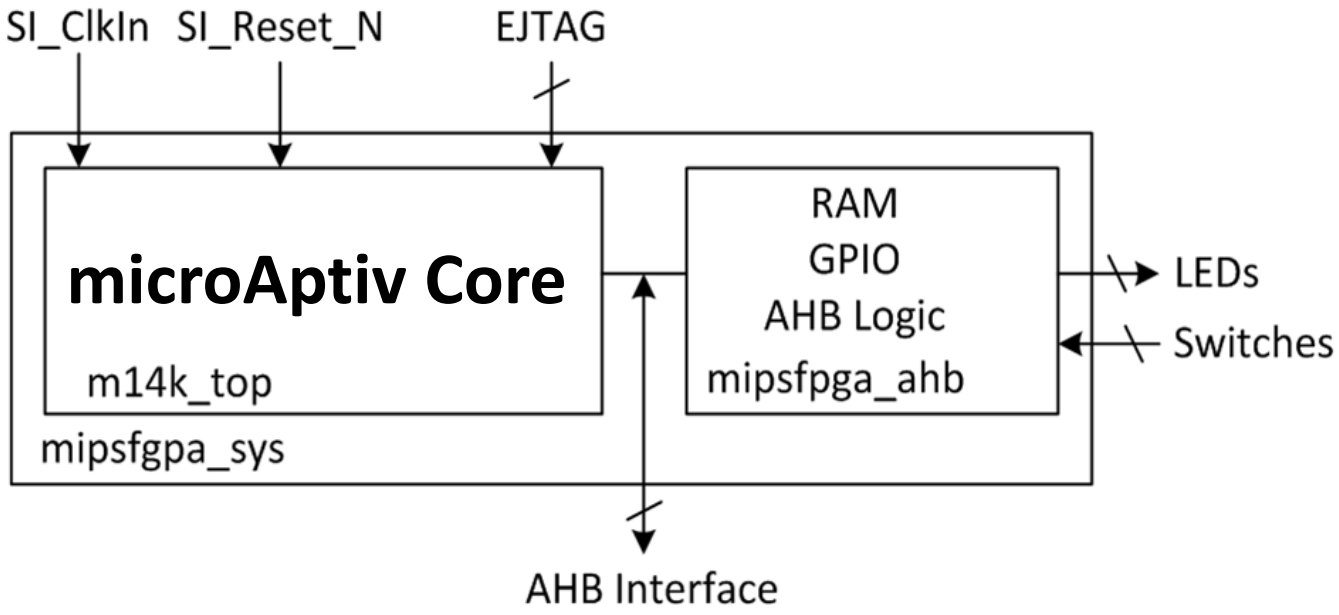
MIPSfpga

These materials produced in association with Imagination.
Join our University community for more resources.
community.imgtec.com/university

Getting Started Guide v2.0

1. MicroAptiv soft-core plus MIPSfpga system
2. Guide (more than 130 pgs)
3. Multiple scripts and documentation



Fundamentals v2.0 – Part 1 (Intro)

Part	Lab	Name	Description
1: Intro	1	**Vivado or Quartus II Project**	Create a project for the MIPSfpga system using Vivado (for the Nexys4 DDR board) or Quartus II (for the DE2-115 board). The RTL (Verilog) for MIPSfpga is available as part of the MIPSfpga Getting Started Guide package.
	2	**C Programming**	Learn how to use Imagination's Codescape Software Development Kit (SDK) and the Bus Blaster probe to write, compile, debug, and run C programs on the MIPSfpga system. Start by running example C code and then write a C program to calculate Fibonacci numbers.
	3	**MIPS Assembly Programming**	Learn how to write, compile, debug, and run MIPS assembly programs on the MIPSfpga system. Start by running example MIPS assembly code and then write an assembly program to calculate Fibonacci numbers.
	4	**More Programming Practice**	Write two more C programs to practice using the MIPSfpga system: a pocket hypnotizer that uses the Nexys4 DDR or DE2-115 FPGA board's LEDs, and a memory game that uses the board's LEDs and pushbuttons.

Fundamentals v2.0 – Part 2 (I/O)

Part	Lab	Name	Description
2: I/O	5	**7-Segment Displays**	Learn how to add memory-mapped I/O to a system. Expand the MIPSfpga system to add access to the eight 7-segment displays available on the Nexys4 DDR or DE2-115 board. Then write two programs to exercise the new I/O.
	6	**Reaction Timer**	Add a memory-mapped millisecond counter to the MIPSfpga system. Write a reaction timer program to test it.
	7	**Audio**	Add a memory-mapped buzzer to the MIPSfpga system. Write a program that plays an example song using the buzzer. Then write a program that plays a song of your choosing.
	8	**SPI Light Sensor**	Add a memory-mapped serial peripheral interface (SPI) port to the MIPSfpga system to drive a light sensor
	9	**SPI LCD**	Add a memory-mapped serial peripheral interface (SPI) port to the MIPSfpga system. Use the SPI port to drive a liquid crystal display (LCD). Write a program that plays a number guessing game to test the functionality.
	10	**Direct-memory-access (DMA)**	Build a DMA engine to drive interactions between peripherals
	11	**DES Encryption**	Build a Data Encryption Standard (DES) encryption engine with DMA access.
	12	**Interrupt-driven I/O**	Interact with peripherals using interrupts.
	13	**Performance Counters**	Learn how to configure and use the Performance Counters in microAptiv and test the performance of various applications.

Imagination

XILINX®

Fundamentals v2.0 – Part 3 (Core)

Part	Lab	Name	Description
3: Core	14	**Instruction Flow: ADD**	Learn how an ADD instruction passes through all the stages of the core and experiment with related arithmetic instructions
	15	**Instruction Flow: AND**	Analyze an AND instruction and perform related exercises.
	16	**Instruction Flow: LW**	Analyze a LW instruction and perform related exercises.
	17	**Instruction Flow: BEQ**	Analyze a branch-if-equal (BEQ) instruction and perform related exercises.
	18	**Hazard Logic**	Analyze and experiment with the microAptiv hazard logic.
	19	**CorExtend**	Add new instructions to MIPSfpga: Learn how to use the CorExtend interface to add user defined instructions (UDIs) to the MIPS32 ISA.

Fundamentals v2.0 – Part 4 (Memory)

Part	Lab	Name	Description
4: Memory	20	Basic Caching	Introduction to the caches available in the MIPSfpga microprocessor (microAptiv UP).
	21	Cache Structure	Analyze the structure of the data cache in detail and implement and test new configurations.
	22	Cache Controller: Hit and Miss Management	Learn about the theory and practice behind hit and miss management within the cache controller via simulation and exercises.
	23	Cache Controller: Cache Content Management Policies	Learn about and test the allocation, write, and replacement policies available in microAptiv. Implement new policies.
	24	Cache Controller: Store Buffer and Fill Buffer	Learn about the buffers (Store Buffer and Fill Buffer) included in the cache controller.
	25	Scratchpad RAM	Implement an Instruction Scratchpad RAM in MIPSfpga (microAptiv).

MIPSfpga-SoC

- Build a System On Chip based on microAptiv core and Xilinx IP blocks + Run Linux and play with it
  - Part 1 (RTL focused): 5 tutorials in separate documents + solution projects + bare metal code. These sequentially build up the SOC design
    - 01_LED
    - 02_UART
    - 03_Custom_GPIO
    - 04_IIC_TempSensor
    - 05_INTController_Ethernet_DDR
  - Part 2 (Linux focused): 5 tutorials that sequentially build up the Linux side after Part 1
    - 01_LinuxKernelInitial
    - 02_Buildroot
    - 03_IntController
    - 04_IIC_Ethernet
    - 05_CustomGPIO