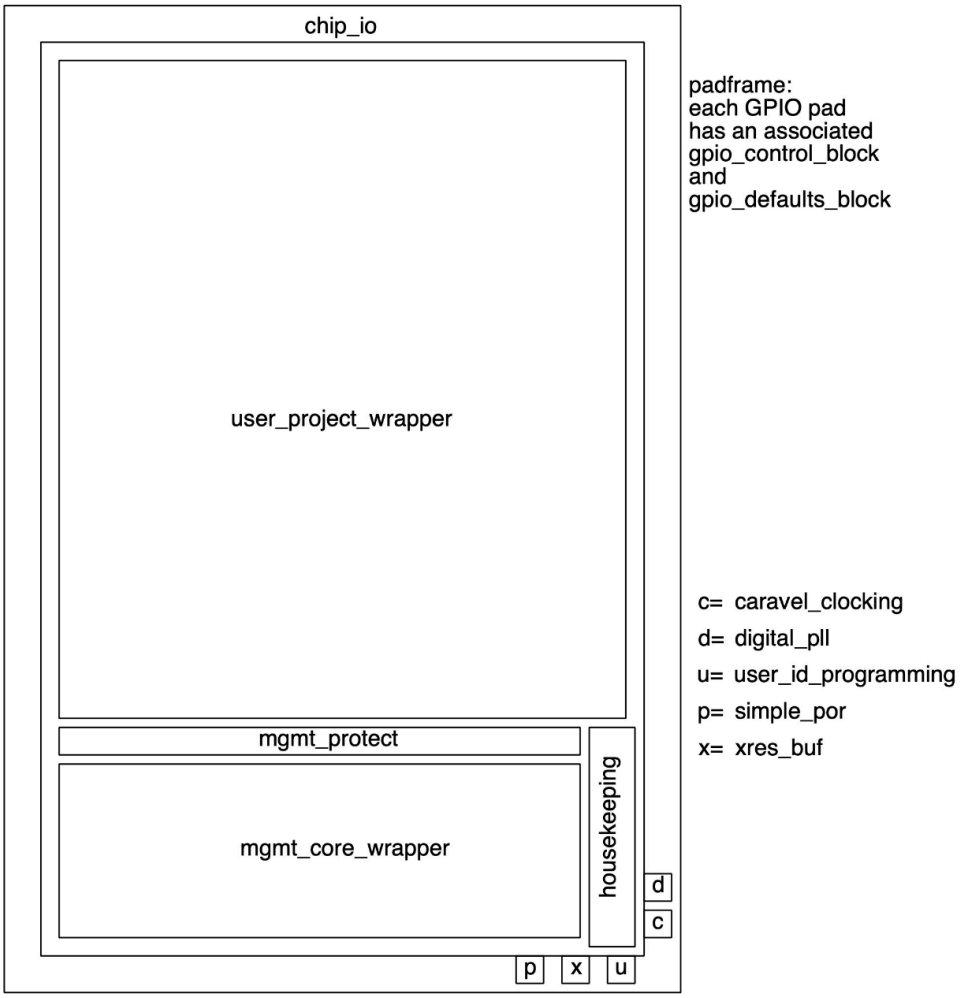
Audience

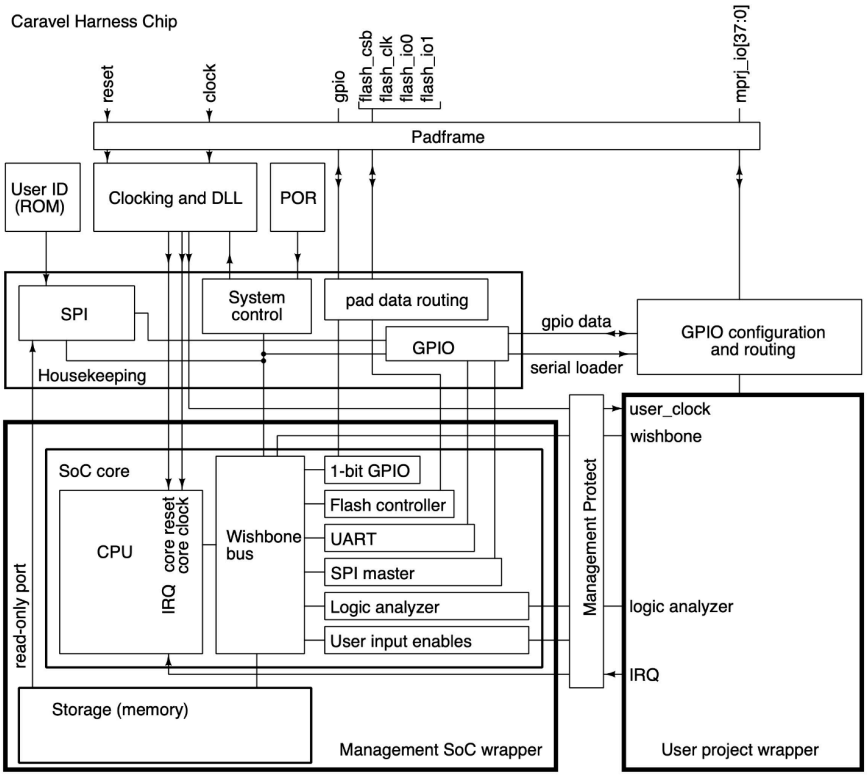
- Anyone who has a design on MPW2,3, 4 and needs to write firmware.
- Anyone who has taped out on MPW1, 2, 3,4 and wants to go on MPW5, what do they need to do differently this time.

Overview

After the timing issues on MPW1, the management SoC was updated. The old PicoRV32 CPU was replaced with a litex generated SoC with VexRiscv as the CPU. The physical layout has stayed very similar:



But the functional diagram has changed:



The GPIO configuration is now outside the management SoC wrapper, so in the event of something like MPW1 issues happening again, we could program the IOs externally using SPI.

The aim was to make the replacement totally transparent, so that the user doesn't need to do anything differently.

However, there are a number of discrepancies that got through, and independently, the way that the project is setup, simulated and documented has changed.

Documentation

The documentation is split into 3 parts:

1. User_project_wrapper - how to setup the build system and get your design hardened and turned into an ASIC: https://github.com/efabless/caravel_user_project/blob/main/docs/source/quickstart.rst
2. Caravel - the padframe, gpios, SPI - everything outside of the user project and the management SoC wrapper in the image above.

https://caravel-harness.readthedocs.io/en/latest/ WARNING! This documentation is out of date. Ignore anything to do with the CPU or peripherals.

3. Management SoC wrapper - this aims to document the peripherals, CPU, memory maps etc. This documentation is not yet hosted on a read the docs, but you can see the source files here: https://github.com/efabless/caravel_mgmt_soc_litex/tree/main/docs/generated

Major differences

- There is a bug that reduces the wishbone address width to 0x10_000. This will not be fixed for MPW5. Make sure your design simulates correctly to be guaranteed it will work after tapeout.
- Logic analyser enables oen_b are now 1 to enable. On MPW1 it was 0 to enable.
- Wishbone must be enabled by writing to special register: reg_wb_enable = 1

Other differences

Setup

- Make setup does everything, no need to install caravel or management wrapper
- New management litex SoC is installed to mgmt_core_wrapper
- No need for magic installation, pdk installed using a magic container
- PDK installation with sram by default
- Simenv target not needed as make target for simulation sets up the environment automatically

Simulation

- Makefiles have been moved about and the simulation makefiles include 4 others.
- C files now need to include #include <defs.h> and #include <stub.c>
- Testbench.v files don't need to include any other files.
- To simulate the example you can run make verify-<test name>-rtl of verify-<test name>-gl for gate level. For example: make verify-io_ports-rtl to run the simulation of the io ports.

Adding a new design for simulation

To simulate a new design, copy one of the example test directories and adapt:

- The _tb file
- The .c file

- Add your RTL source files to verilog/includes/includes.rtl.caravel_user_project
- Add your GL source file to verilog/includes/includes.gl.caravel_user_project

Spare logic

Spare logic has been added to Caravel. For example, the spare logic block is designed to give some limited capacity to fix an error with a metal mask correction, meaning that you could run the chip manufacture a second time but avoid re-creating all the masks, which are the most expensive part of the manufacturing process. The spare logic is designed so that you can add or rework a small amount of logic using only a few metal layers. It depends somewhat on just how deeply buried the error is, whether it is really fixable with only a small amount of logic, and whether it is worth doing the manufacturing run over. Something that affected all projects and made everything dead on arrival would be a good candidate for a metal mask fix (although MPW one's problems went far beyond what a simple metal mask fix could do).