PYNQ™

Overlay Design Methodology labs

XILINX®

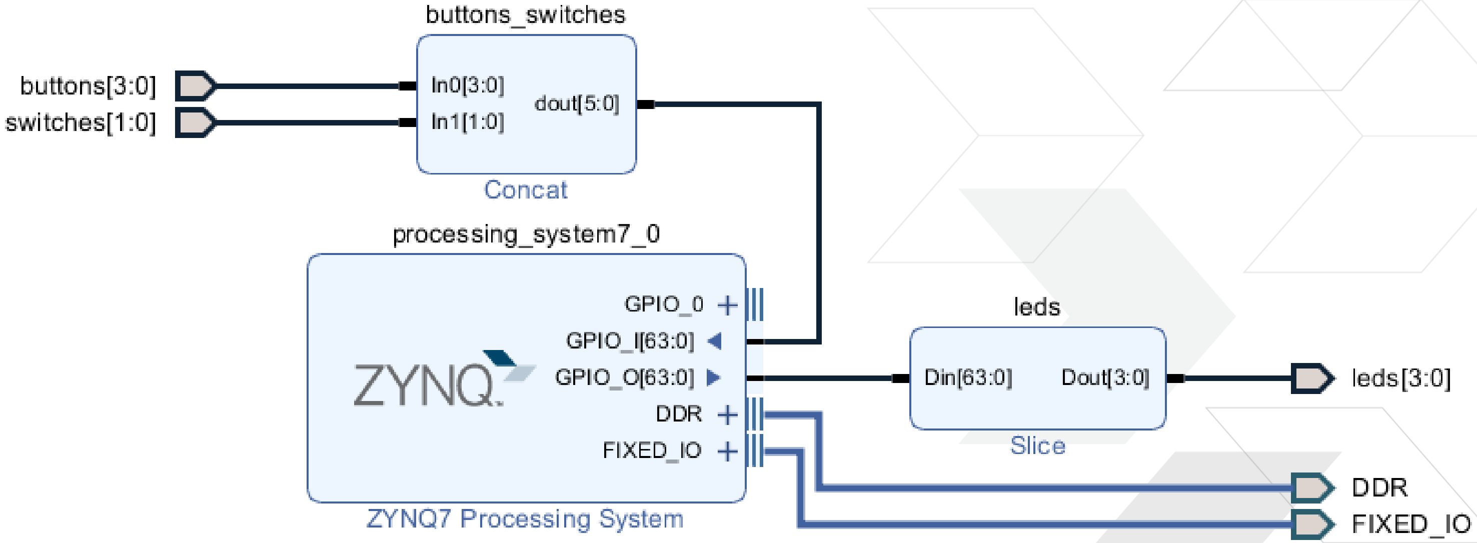
Labs

- PS GPIO
- AXI GPIO
- MMIO (GPIO)
- Memory allocation example using Xlnk
- Using memory from PL master
- DMA
- Image resizer

PS GPIO

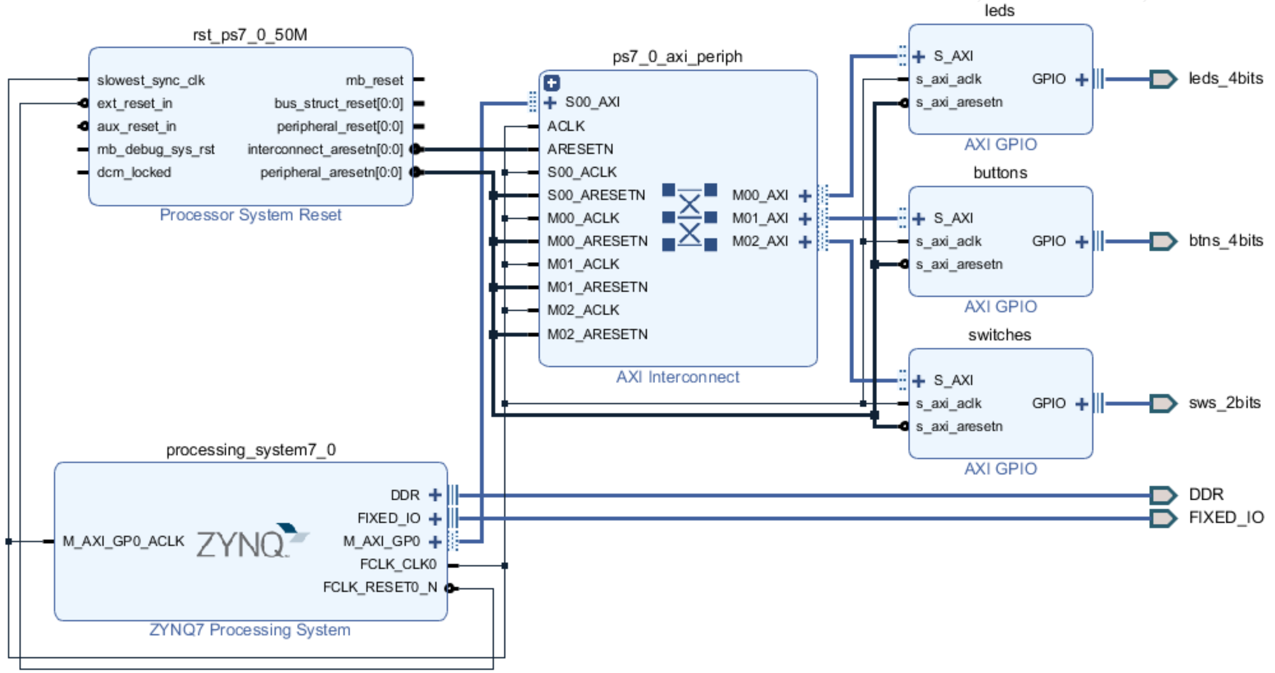
- **64 PS GPIO available**
  - Useful for debug/prototyping
- **Use PS GPIO class**
  - Instantiate GPIO pin using get_gpio_pin(n) method and direction
  - Use .write and .read methods to interact with the instantiated pin

- **GPIO 0:3 = push-buttons**
- **GPIO 4:5 = dip-switches**
- **GPIO 6:9 = LEDs**



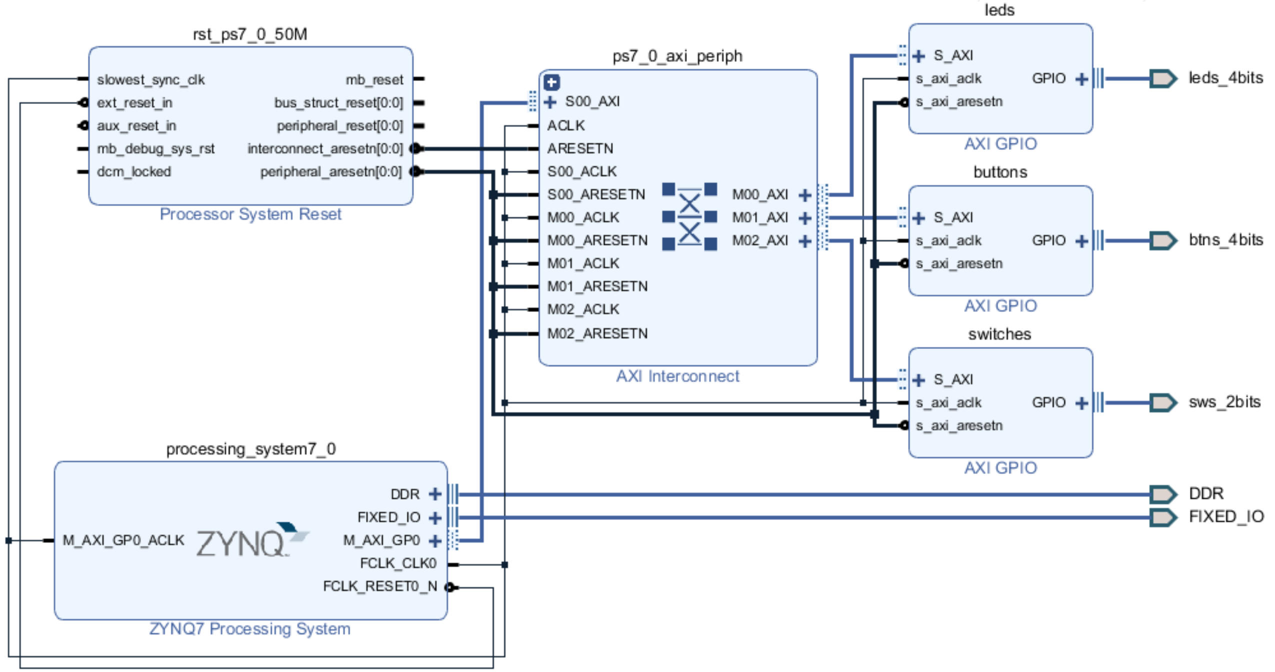
AXI GPIO

- **Similar to previous example**
  - Switches, buttons, LEDs
- **3x AXI GPIO controller**
- **Uses PYNQ AxiGPIO driver**
  - Address peripheral by slice
  - LED[0:3]

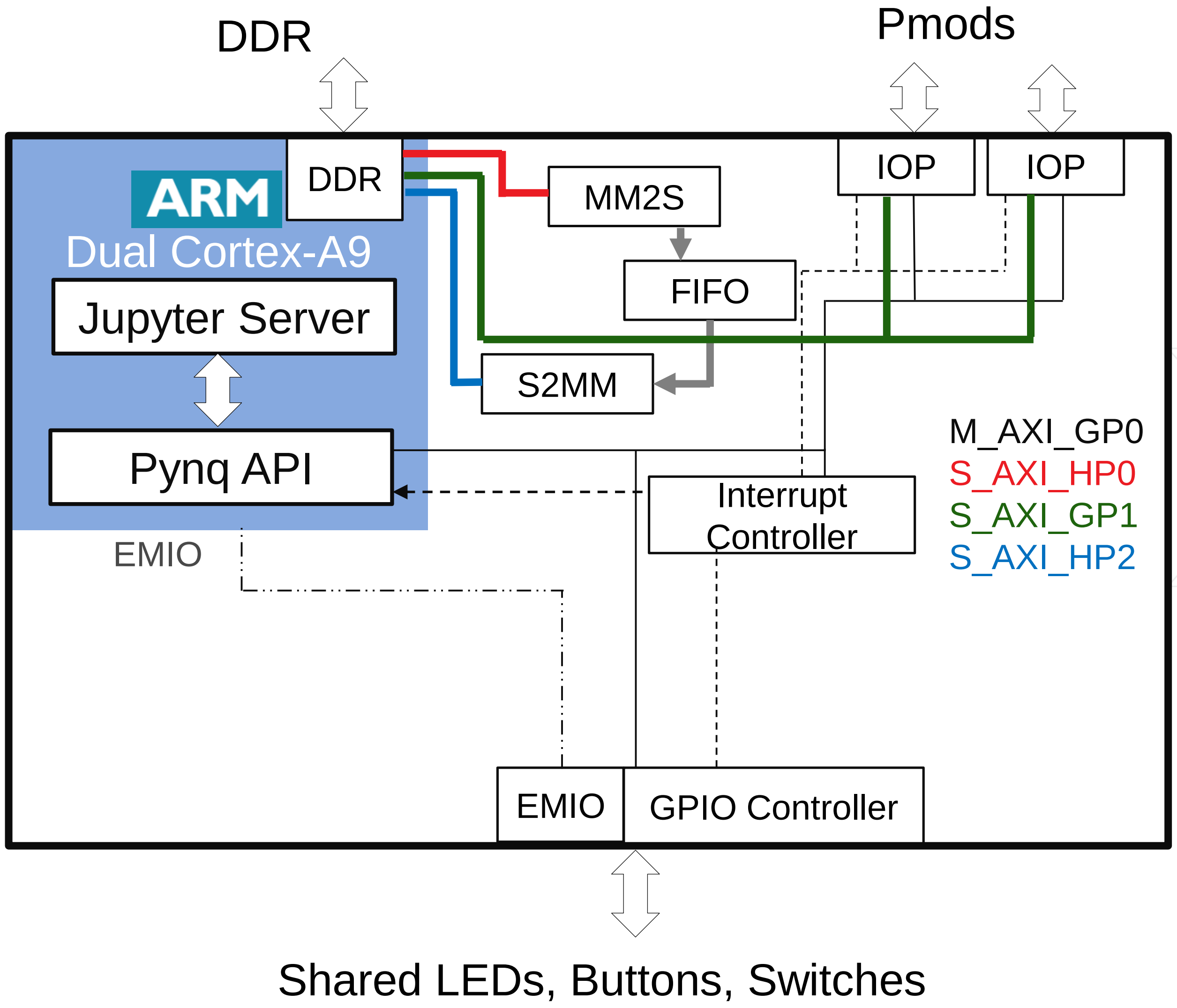


PYNQ MMIO

- > **Uses previous design**
- > **AXI GPIO controller**
- > **Read and write Register 0x0 directly**
- > **Write tri-state register (0x4) to configure as inputs or outputs**
- > **Build your own Python functions to control IP**



pynqtutorial Overlay (pynqtutorial.bit)



pynqtutorial Overlay – Interface View

- IP included
  - IOP1 and IOP2
  - DMA -> FIFO -> DMA
- Shared LEDs and buttons between EMIO and MIO ports of PS
- Allows testing of DMA interfaces to AXI streams
- Replace FIFO with custom stream IP

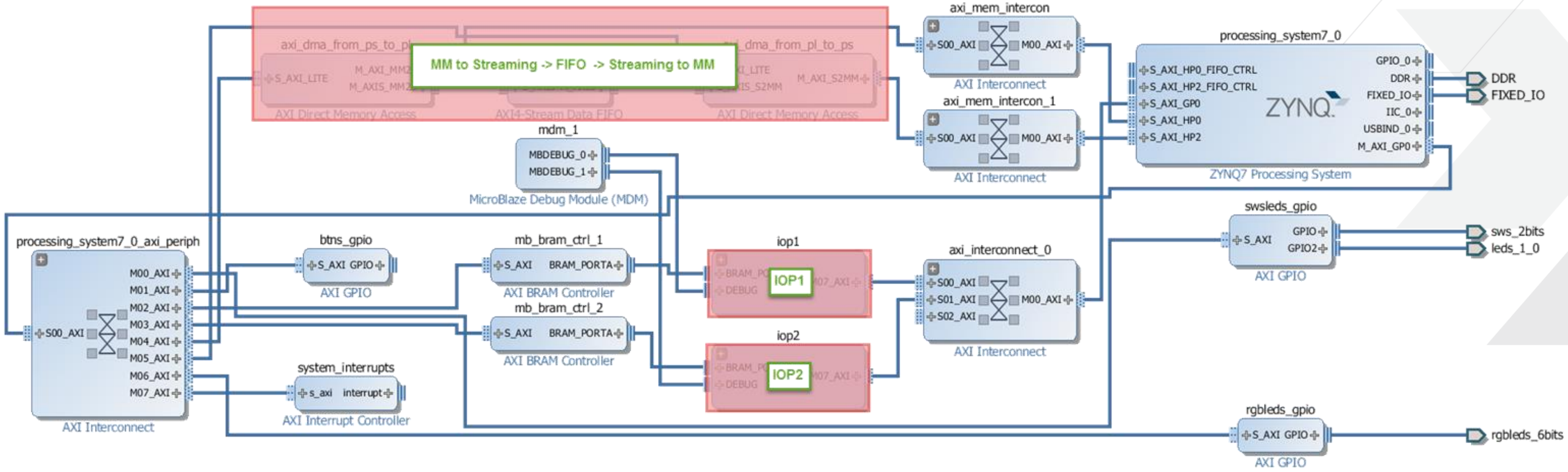
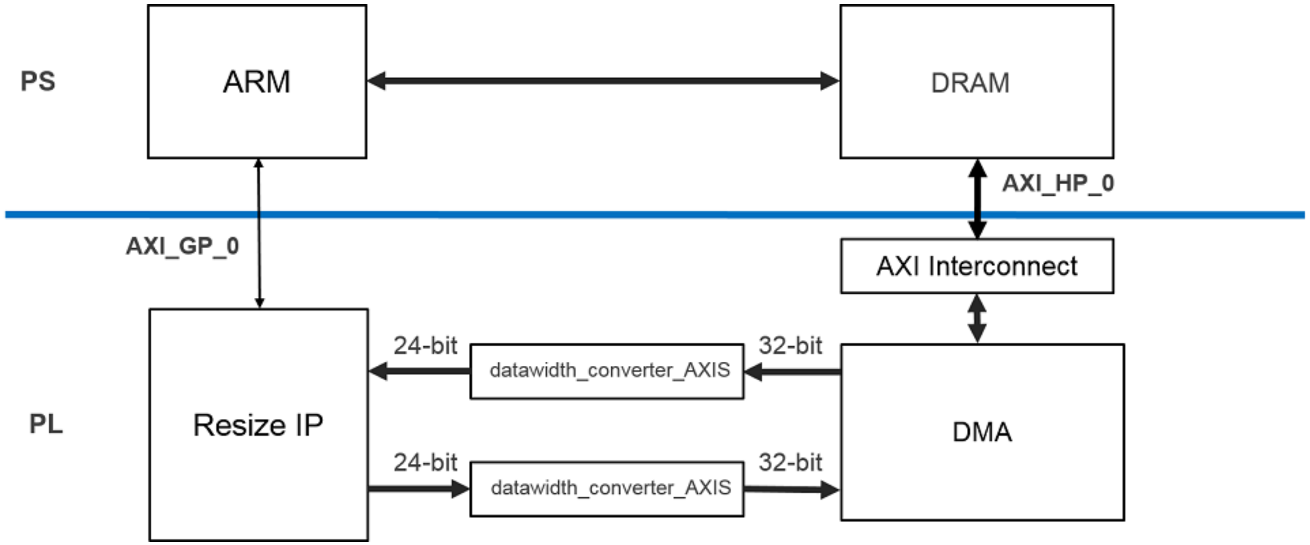


Image resize

- Resizer IP in PL
- Uses DMA, and MMIO
- Resize an image in software
- Resize image in PL and compare



Adaptable.
Intelligent.

XILINX

PYNQ