Hello LED

Digital Systems M, Module 1

Stefano Mattoccia, Università di Bologna

Development phases of a Zynq project

Device creation (C/C++) and validation (C/C++)

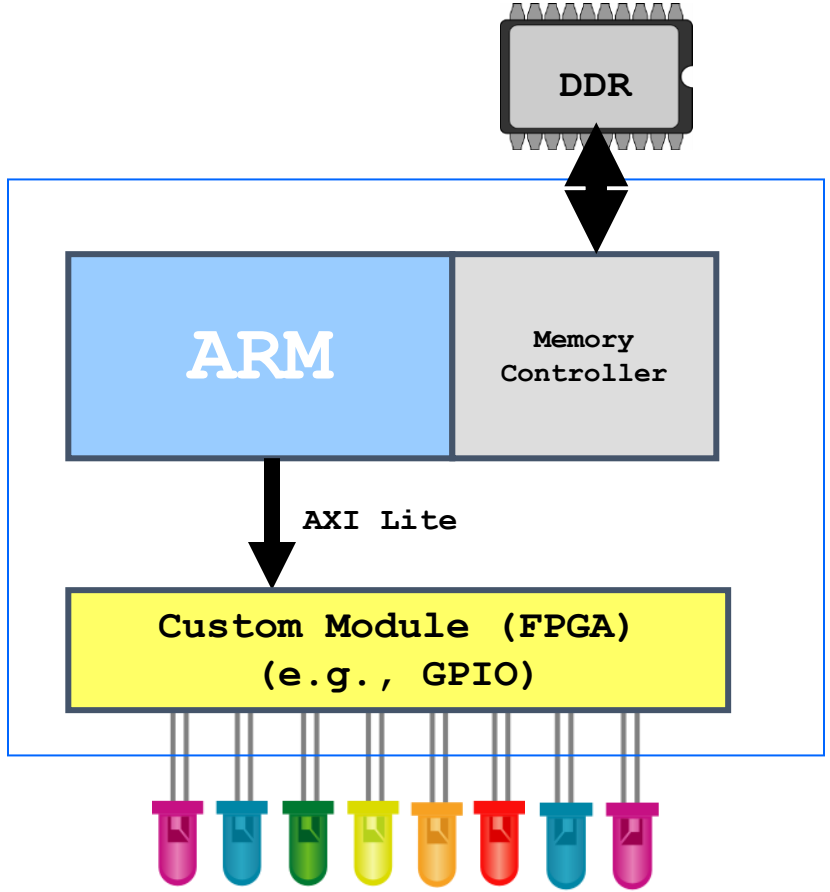
This part will take most of the following lectures

Graphical design entry

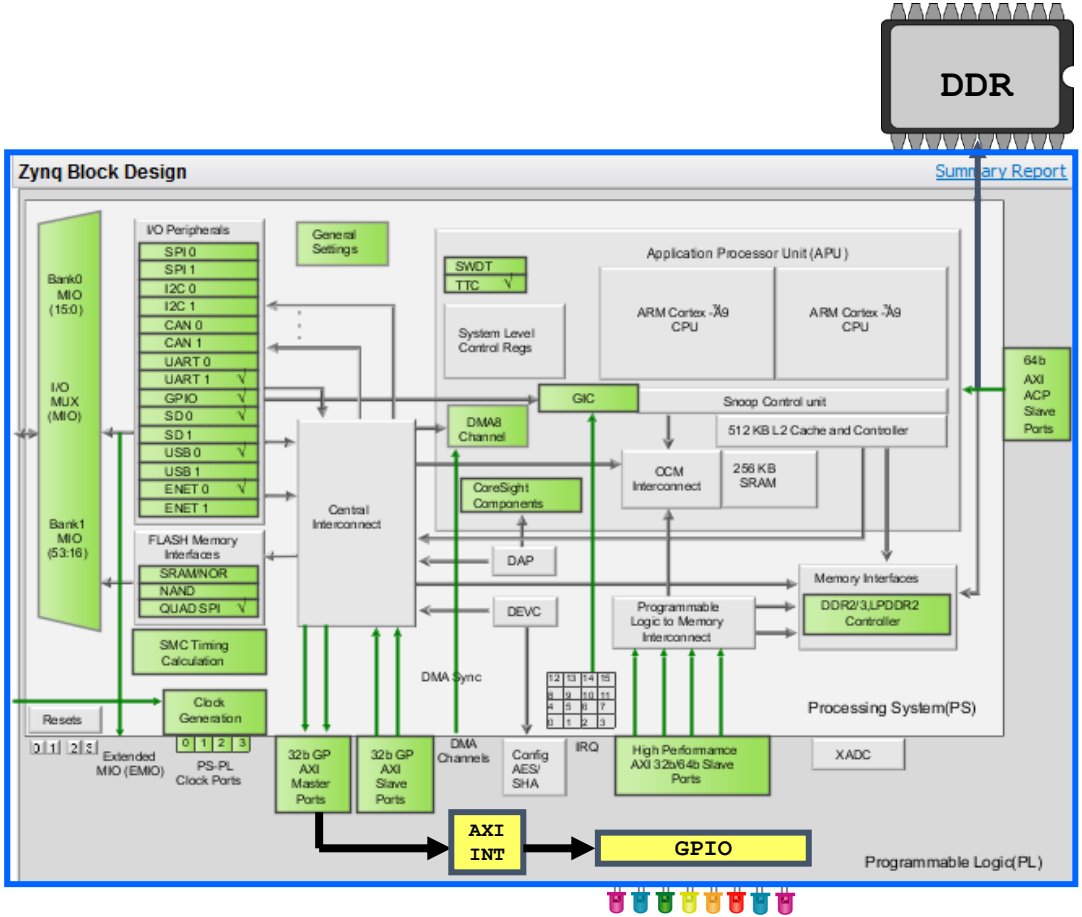
Software development C/C++ for specific OS

Example: turning a LED on

Let's take, for now, an existing device in form of IP Core, named **GPIO**



Hello LED


DDR
Zynq Block Design
Summary Report
I/O Peripherals
SPI 0
SPI 1
I2C 0
I2C 1
CAN 0
CAN 1
UART 0
UART 1
GPIO
SD 0
SD 1
USB 0
USB 1
ENET 0
ENET 1
Bank0 MIO (15:0)
I/O MUX (MIO)
Bank1 MIO (53:16)
FLASH Memory Interfaces
SRAM/NOR
NAND
QUAD SPI
SMC Timing Calculation
General Settings
SWDT
TTC
System Level Control Regs
Application Processor Unit (APU)
ARM Cortex-A9 CPU
ARM Cortex-A9 CPU
GIC
Snoop Control unit
512 KB L2 Cache and Controller
DMA8 Channel
CoreSight Components
Central Interconnect
OCM Interconnect
256 KB SRAM
DAP
DEVC
Memory Interfaces
DDR2/3,LPDDR2 Controller
Programmable Logic to Memory Interconnect
64b AXI ACP Slave Ports
DMA Sync
Resets
Clock Generation
Extended MIO (EMIO)
PS-PL Clock Ports
32b GP AXI Master Ports
32b GP AXI Slave Ports
DMA Channels
Config AES/ SHA
IRQ
High Performance AXI 32b/64b Slave Ports
XADC
Processing System(PS)
AXI INT
GPIO
Programmable Logic(PL)

Zynq: Addressing space

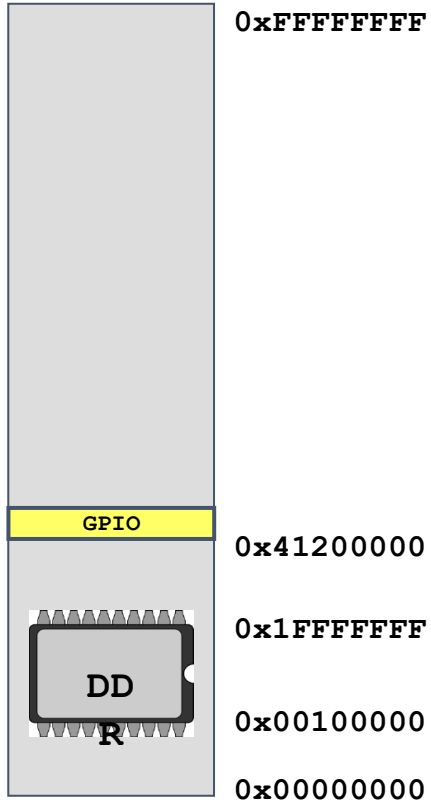
- ✓ (External) memory and addressing space (32 bit) are shared among:
  - ✓ ARM Cortex A9_0
  - ✓ ARM Cortex A9_1
  - ✓ FPGA
- ✓ Single memory controller (inside the PS) and single addressing space for memory and devices (both ARM and FPGA), that are memory-mapped in the range

  0x00000000 -> 0xFFFFFFFF
- ✓ The Programmable Logic (FPGA) is connected to the DDR memory controller through the High Performance ports HP 0, 1, 2, 3.
  High transfer rate (GB/s) both at reading and writing

Hello LED

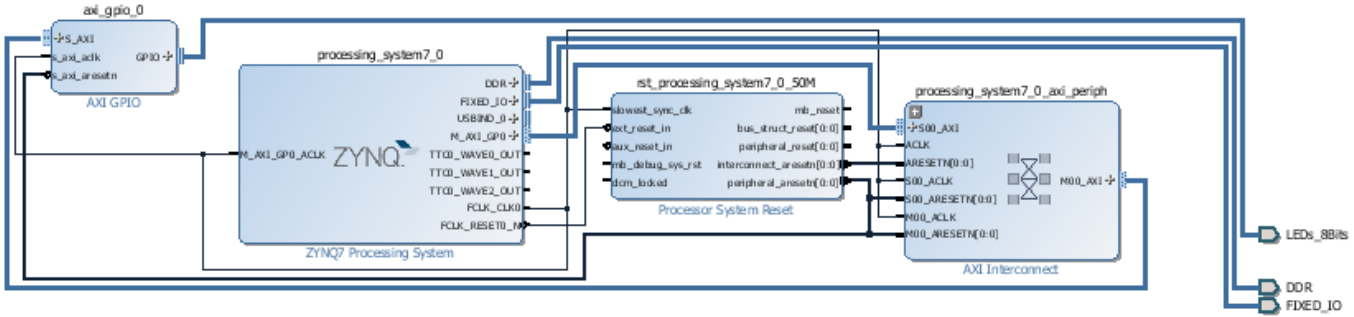
Address Map for processor ps7_cortexa9_0

Cell	Base Addr	High Addr	Slave I/f	Mem/Reg
axi_gpio_0	0x41200000	0x4120ffff	S_AXI	REGISTER
ps7_afi_0	0xf8008000	0xf8008fff		REGISTER
ps7_afi_1	0xf8009000	0xf8009fff		REGISTER
ps7_afi_2	0xf800a000	0xf800afff		REGISTER
ps7_afi_3	0xf800b000	0xf800bfff		REGISTER
ps7_coresight_comp_0	0xf8800000	0xf88fffff		REGISTER
ps7_ddr_0	0x00100000	0x1fffffff		MEMORY
ps7_ddrc_0	0xf8006000	0xf8006fff		REGISTER
ps7_dev_cfg_0	0xf8007000	0xf80070ff		REGISTER
ps7_dma_ns	0xf8004000	0xf8004fff		REGISTER
ps7_dma_s	0xf8003000	0xf8003fff		REGISTER
ps7_ethernet_0	0xe000b000	0xe000bfff		REGISTER
ps7_globaltimer_0	0xf8f00200	0xf8f002ff		REGISTER
ps7_gpio_0	0xe000a000	0xe000afff		REGISTER
ps7_gpv_0	0xf8900000	0xf89fffff		REGISTER
ps7_intc_dist_0	0xf8f01000	0xf8f01fff		REGISTER
ps7_iop_bus_config_0	0xe0200000	0xe0200fff		REGISTER
ps7_l2cachec_0	0xf8f02000	0xf8f02fff		REGISTER
ps7_ocmc_0	0xf800c000	0xf800cfff		REGISTER



Vivado project

- ✓ Resulting Vivado project to control the 8 LEDs through an IP core (GPIO) AXI *lite* interface:



- ✓ An AXI interconnected hub has been inserted (automatically) for future insertion of additional AXI lite devices
- ✓ Moreover, a reset controller module has been inserted (automatically) for system reset (Processor System Reset)

ARM/SDK code (Baremetal OS)

```
#include <stdio.h>
#include "platform.h"
#include "xil_io.h"
#include <unistd.h>

int main()
{
    init_platform();

    int c;
    useconds_t  sleeping_time_us = 50000;

    printf("Hello LED\n");

    for (c=0;c<=255;c++)
    {
    printf("Xil_Out start %d\n",c);
    Xil_Out8(0x41200000, c);
    printf("Xil_Out done, sleeping for %d us\n",sleeping_time_us);
    usleep(sleeping_time_us);
    }

    for (c=255;c>=0;c--)
     {
     printf("Xil_Out start %d\n",c);
     Xil_Out8(0x41200000, c);
     printf("Xil_Out done, sleeping for %d us\n",sleeping_time_us);
     usleep(sleeping_time_us);
     }

    cleanup_platform();
    return 0;
}
```

Hello LED