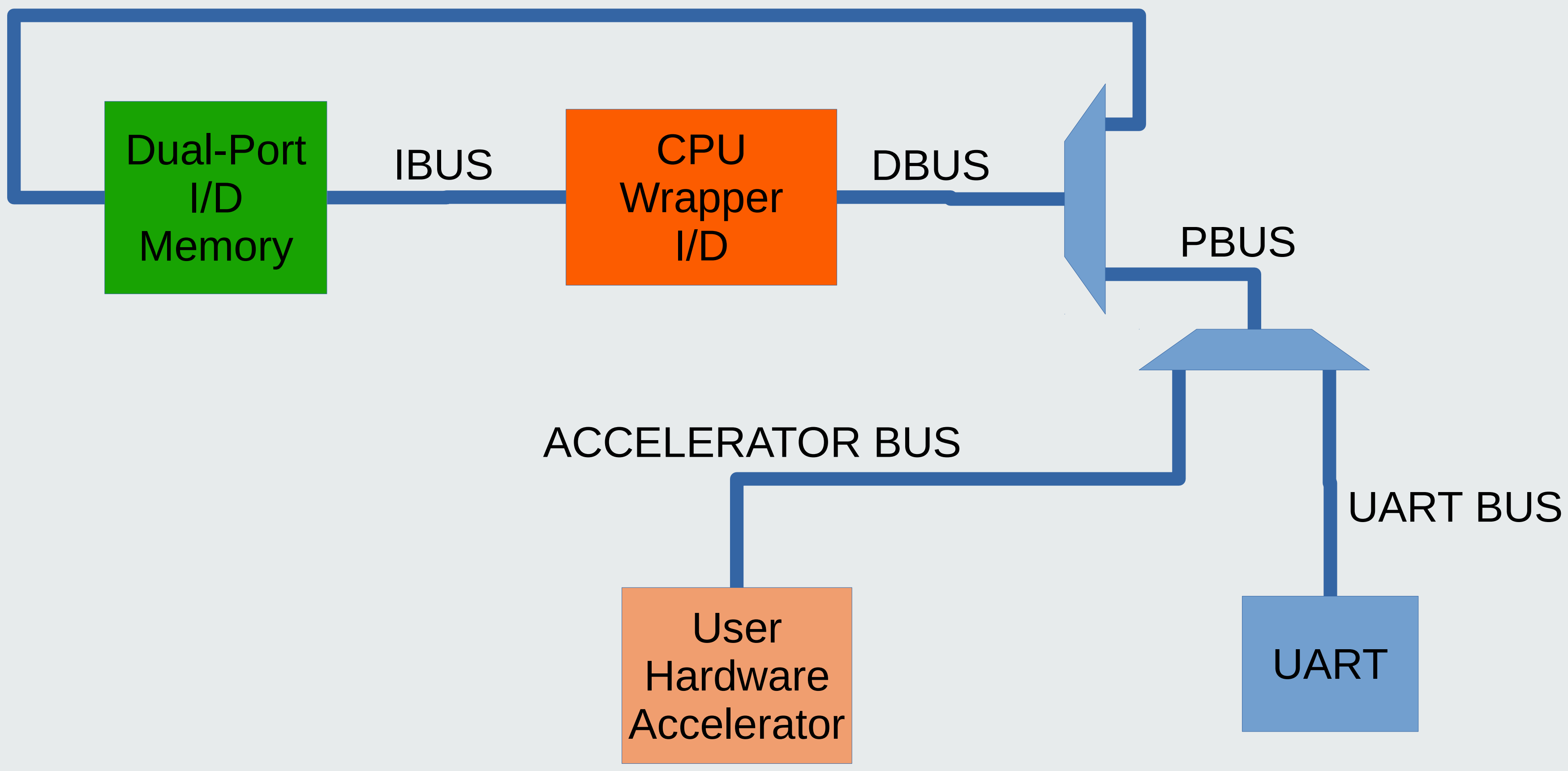
Computer Electronics

Lecture 4: IOb-SoC Tour (continued) and first Verilog notions

TÉCNICO LISBOA

IOb-SoC: recap of last lesson & more



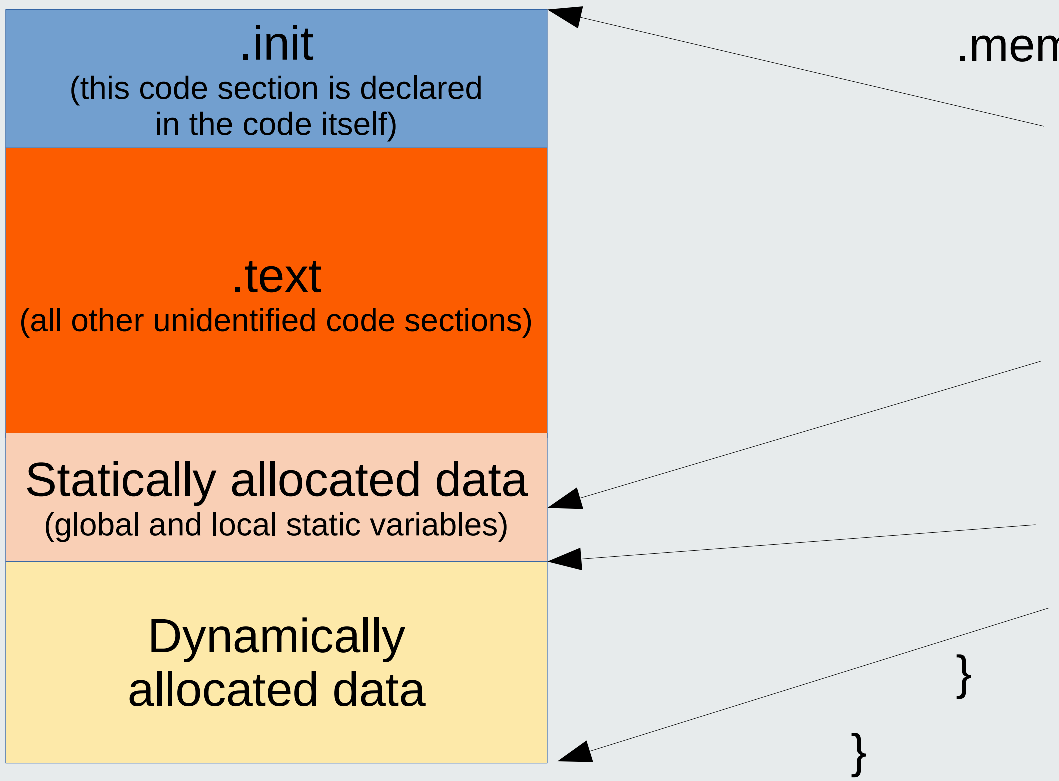
IOb-SoC: directories

- Software: C/C++ software programs + python scripts for automation
- Hardware: verilog hardware descriptions
- Document: latex documents
- Submodules: other repositories used by Iob-SoC, uses *git submodule* technology

IOb-SoC: the software directory

- Directories
  - firmware: the program to run on IOb-SoC (modify to create your own program)
  - Do not edit these:
    - bootloader: the resident bootloader program
    - console: the PC program that communicates with the board
    - python: directory of build automation scripts
- Files: (not intended to be edited)
  - software.mk: makefile segment for software makefiles
  - system.h: system parameters extracted from configurations entered in .mk files
  - template.lds: linker script

Iob-SoC: linker script


.init
(this code section is declared in the code itself)
.text
(all other unidentified code sections)
Statically allocated data
(global and local static variables)
Dynamically allocated data

```
SECTIONS {
          .memory : {
                    . = 0x000000;
                    *(.init);
                    *(.text);
                    *(*);
                    . = ALIGN(4);
                    _heap_start = .;
                    end = .;
          }
}
```

Also the initial Stack Pointer Value

IOb-SoC: the firmware directory

- Files (visit & understand)
  - Makefile: firmware top build script
    - Includes software.mk
  - firmware.S: the assembly start file
    - Includes system.h
  - firmware.c: the firmware C program
    - Includes system.h

Iob-SoC: bootloader, console, python

- Bootloader
  - Same structure as firmware
    - it is a normal program but used to boot the system
- Console
  - Compiled and run on the PC using <u>normal</u> gcc
  - Communicates with the target board where Iob-SoC is running via USB(TX)→RS232 (UART) → USB(RX)
  - (Sends the firmware to target and) commands it to run
  - May be run repeatedly during debug
- Python:
  - Contains scripts that are too cumbersome to do with Make commands or bash commands in the Make recipes

IOb-SoC: the hardware directory

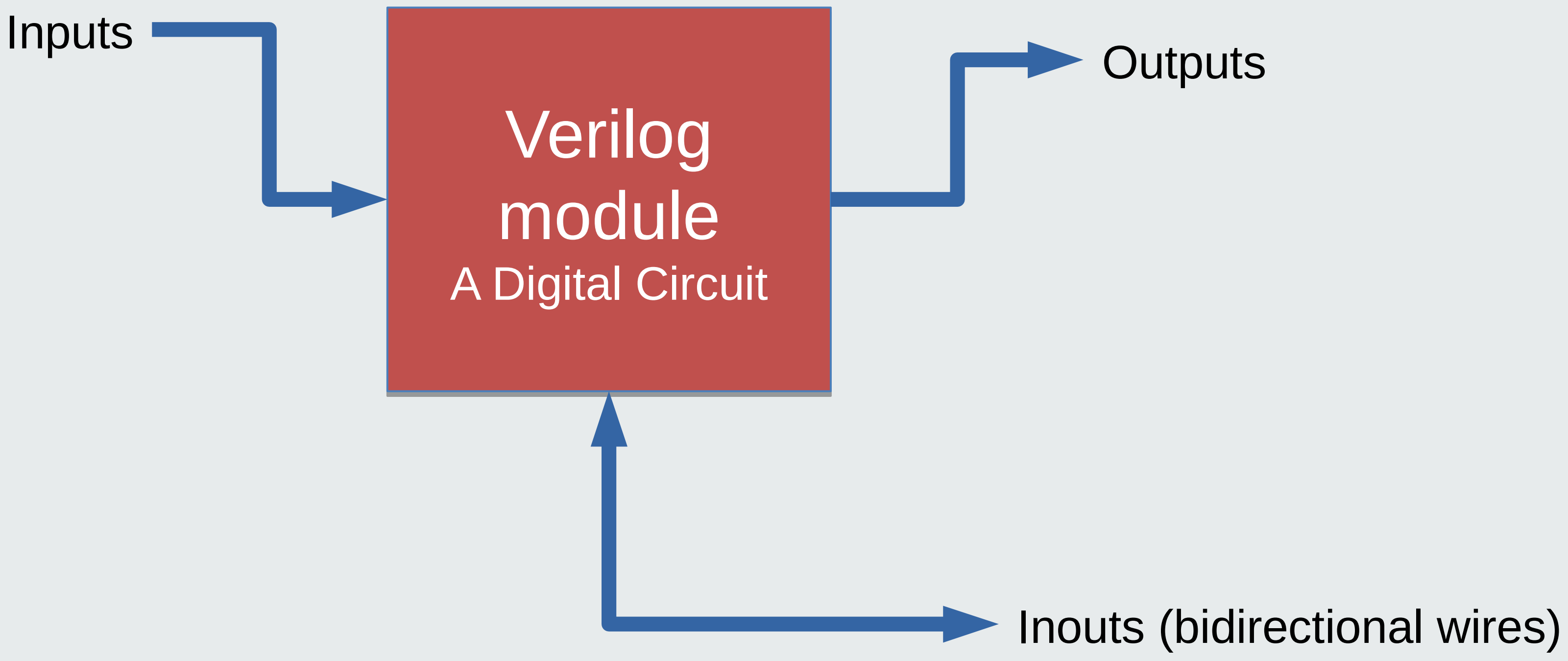
- Directories (not intended to be edited)
  - src: the basic system verilog files
  - testbench: Verilog testbench
  - include: the Verilog header files (shared by many files)
  - simulation: build and run directories
  - fpga: build and run directories
  - asic: build and run directories
- Files: (not intended to be edited)
  - hardware.mk: makefile segment for hardware makefiles (simulation, fpga and asic)

IOb-SoC: the hardware/src directory

- Contains Verilog modules
- Verilog is a hardware DESCRIPTION language
- Verilog is not a programming language
- We are describing circuits not procedures
- We use procedures to describe circuits

TÉCNICO LISBOA

Verilog module (a digital circuit)



Verilog module (a circuit)

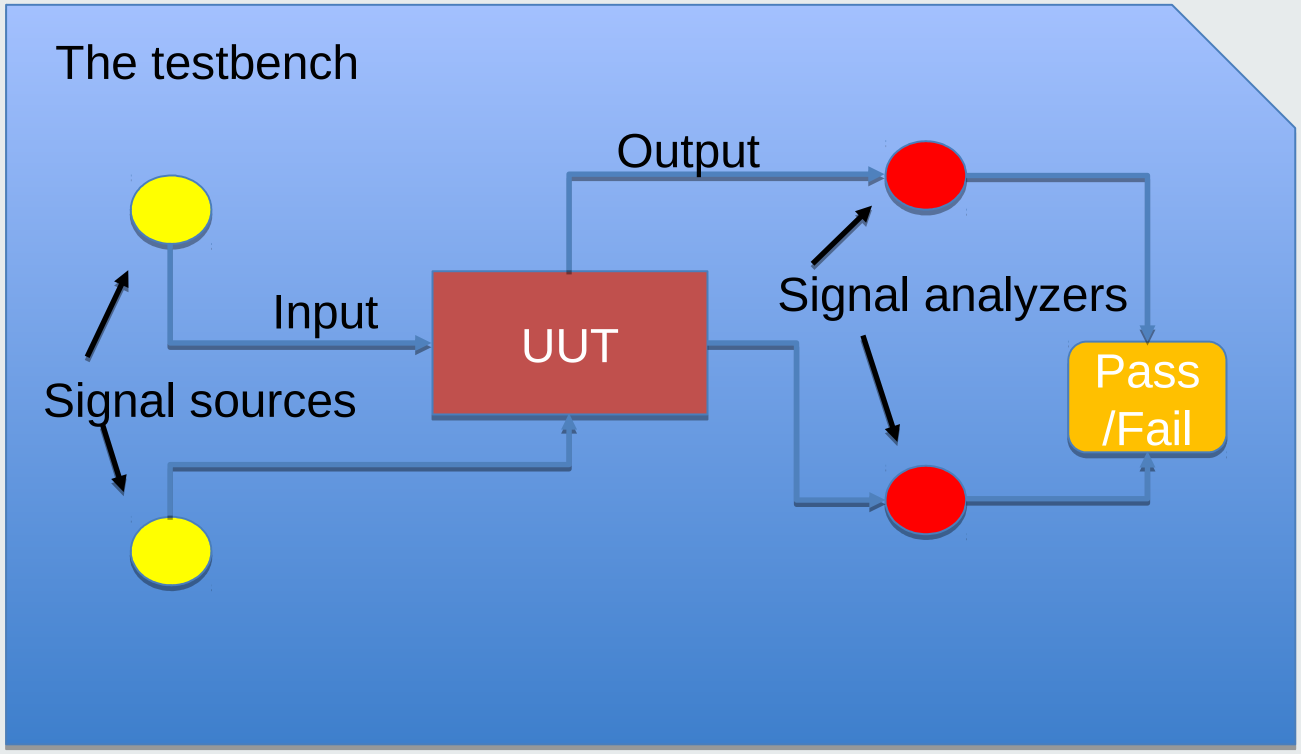
```
module my_unimaginable_peripheral
  (
   input                    clk,
   input                    rst,

   // native bus
   input                                 valid,
   input    [`SRAM_ADDR_W-3:0] addr,
   input    [`DATA_W-1:0]               wdata, //used for booting
   input    [`DATA_W/8-1:0]             wstrb, //used for booting
   output  [`DATA_W-1:0]                rdata,
   output                                ready
);

//description of module goes here

endmodule
```

The testbench directory



Verilog testbench (not a real circuit!)

Used in Verilog simulation!

```
module my_megapower_testbench;

//description of test hardware

//instance of the Unit Under Test

endmodule
```

Verilog header files

```
`include "my_humble_header_file.v"
// this like the C language #include

module my_megapower_testbench;

//description of test hardware

//instance of the Unit Under Test

endmodule
```

IOb-SoC: the simulation directory

- Contains simulation build and run directories, one directory per simulator
  - Only the icarus simulator will be used
    - The icarus directory only has a simulation makefile (visit)
- Files:
  - the simulation.mk makefile segment (visit)
  - The waves.gtkw input file for the gtkwave program

IOb-SoC: the <u>document</u> directory

- Contains document build directories, one directory per document type
  - pb: product brief
    - Makefile
    - .tex files
    - .pdf file for easy viewing
    - Private <u>figures</u> directory
  - presentation: slide presentation
- Contains the <u>figures</u> directory
  - Figures that are shared among different documents

IOb-SoC: the submodules directory

- System submodules
  - CPU
  - CACHE
  - UART
- A git submodule is a pointer to another git repository
  - Allows for modular and distributed development :-)
  - Easy to add with the git submodule add <url> path/to/the/submodule :-)
  - Not so easy to remove: git submodule remove does not exist :-(
- The way the system collects hardware and software components from the submodules and assembles them in the system is the topic of the next class.