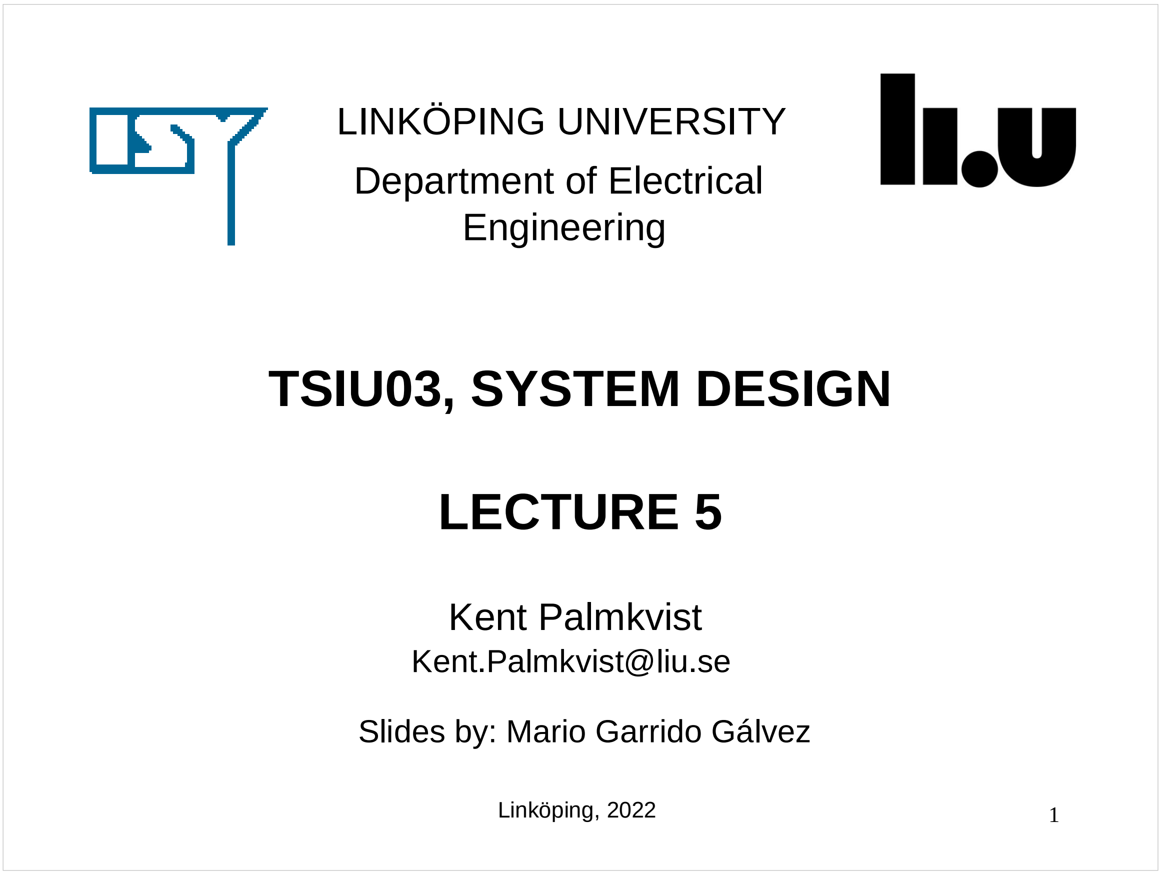


LINKÖPING UNIVERSITY

Department of Electrical Engineering

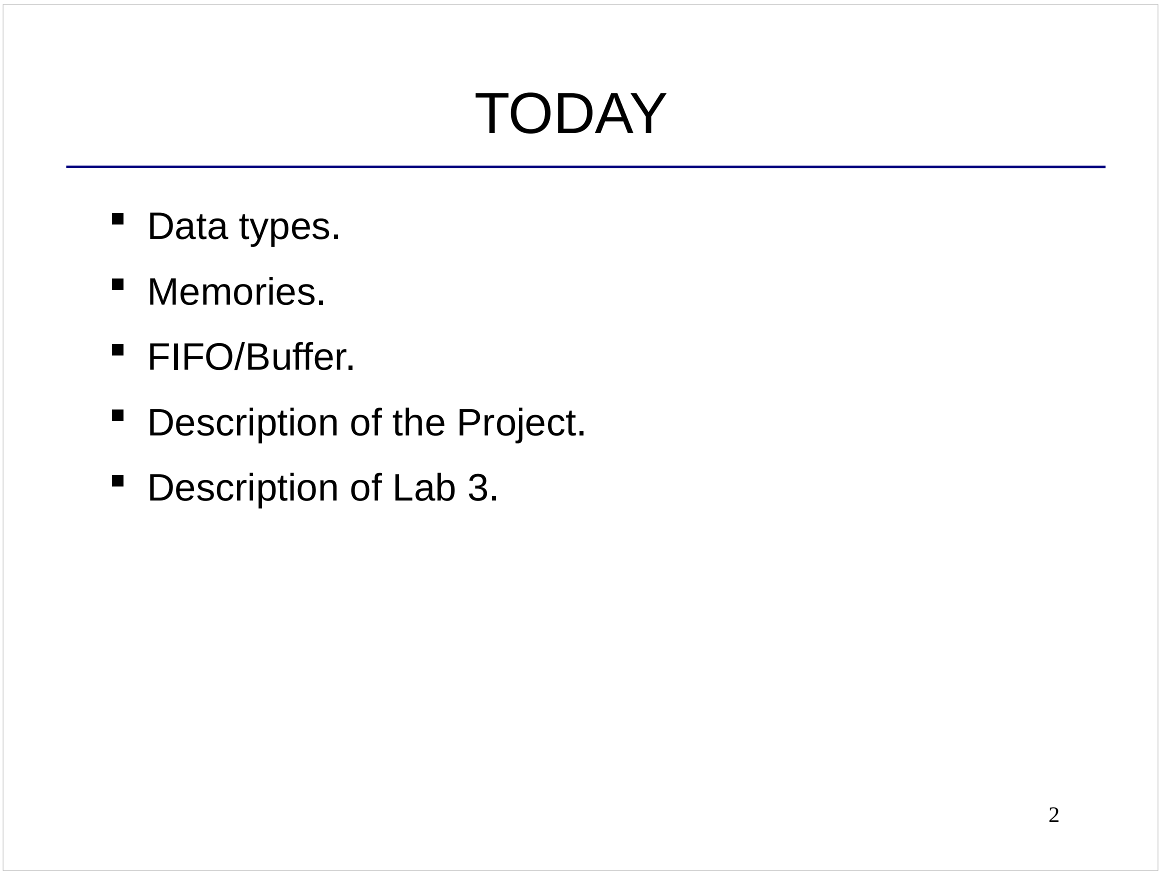
TSIU03, SYSTEM DESIGN

LECTURE 5

Kent Palmkvist
Kent.Palmkvist@liu.se

Slides by: Mario Garrido Gálvez

Linköping, 2022



TODAY

- Data types.
- Memories.
- FIFO/Buffer.
- Description of the Project.
- Description of Lab 3.

DATA TYPES

- VHDL is strongly typed: The type of each signal must always be clear.
- Operators depend on the type. Example: the operator < gives a different result for signed and unsigned, and cannot be used with std_logic_vector.
- Some standard types (predefined in VHDL):

```
type boolean is (true, false);
type integer is range -2147483648 to +2147483647;
type real -- real value
type time –- integer number plus unit (ns, ms,...)
```

- Some types are synthesizable, such as boolean or integer and some types are not, such as real or time.
- Good description and further information on data types: [R4].

TYPES OF TYPES

- Integer types:

```
type integer is range -2147483648 to +2147483647;
type short is range -128 to 127;
```

- Subtypes:

```
subtype natural is integer range 0 to integer'high;
```

- Enumeration:

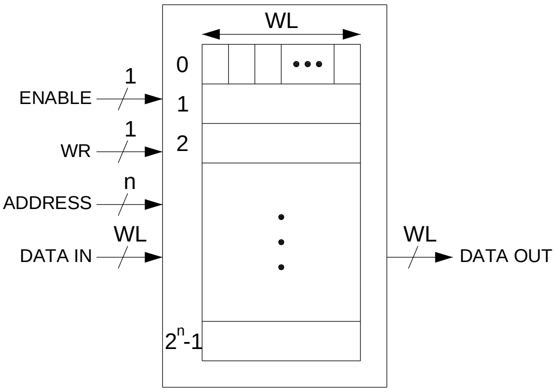
```
type std_logic is ('U','X','0','1','Z','W','L','H','-');
type bit is ('0','1')
type boolean is (false, true);
```

- Array:

```
type memory is array (0 to 2**bitsAddr -1) of
std_logic_vector (WL -1 downto 0);
```

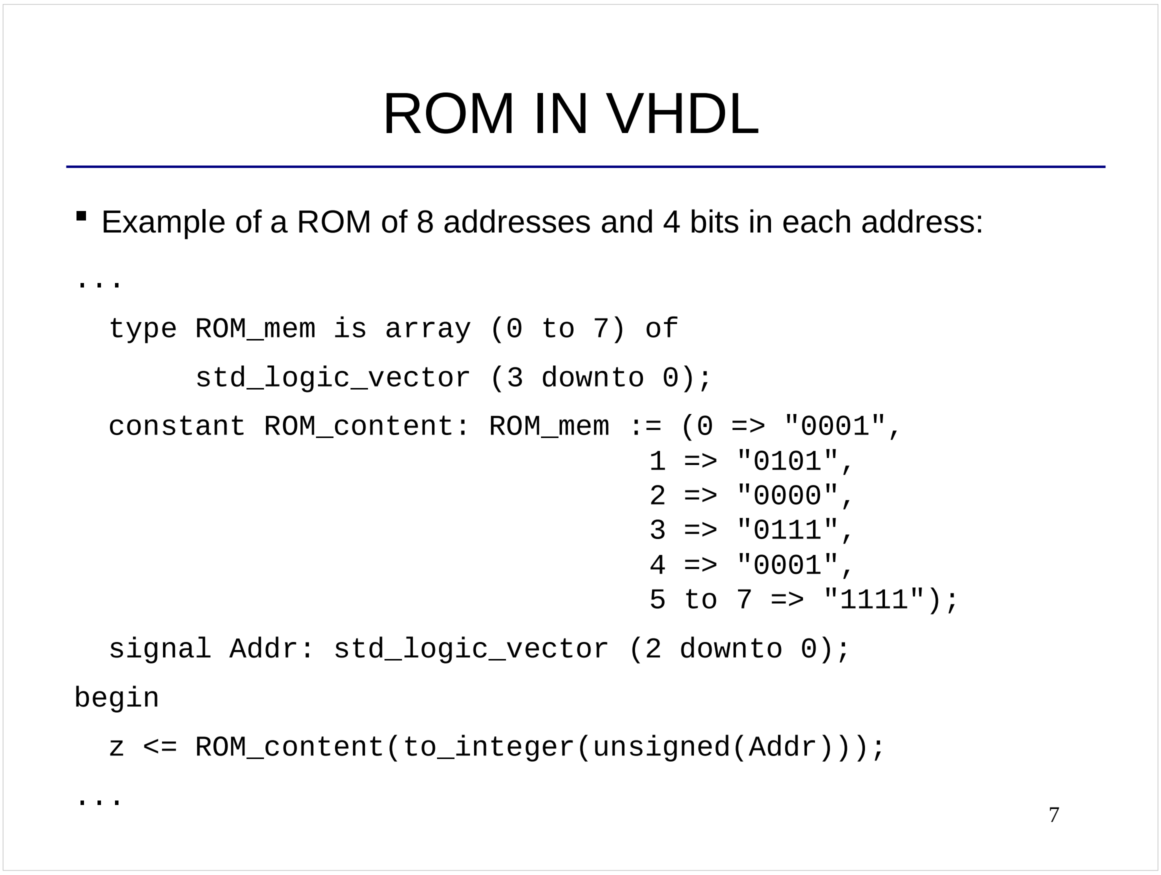
MEMORY

- Circuit used for storage.
- It is an array of 2^n locations, each of which stores *WL* bits of information.
- Each of the locations has one address that ranges from 0 to 2^n - 1. Therefore, the address of the memory is represented by *n* bits.
- We can read a memory location by selecting its address.
- In some memories we can also write in the memory locations.



TYPES OF MEMORY

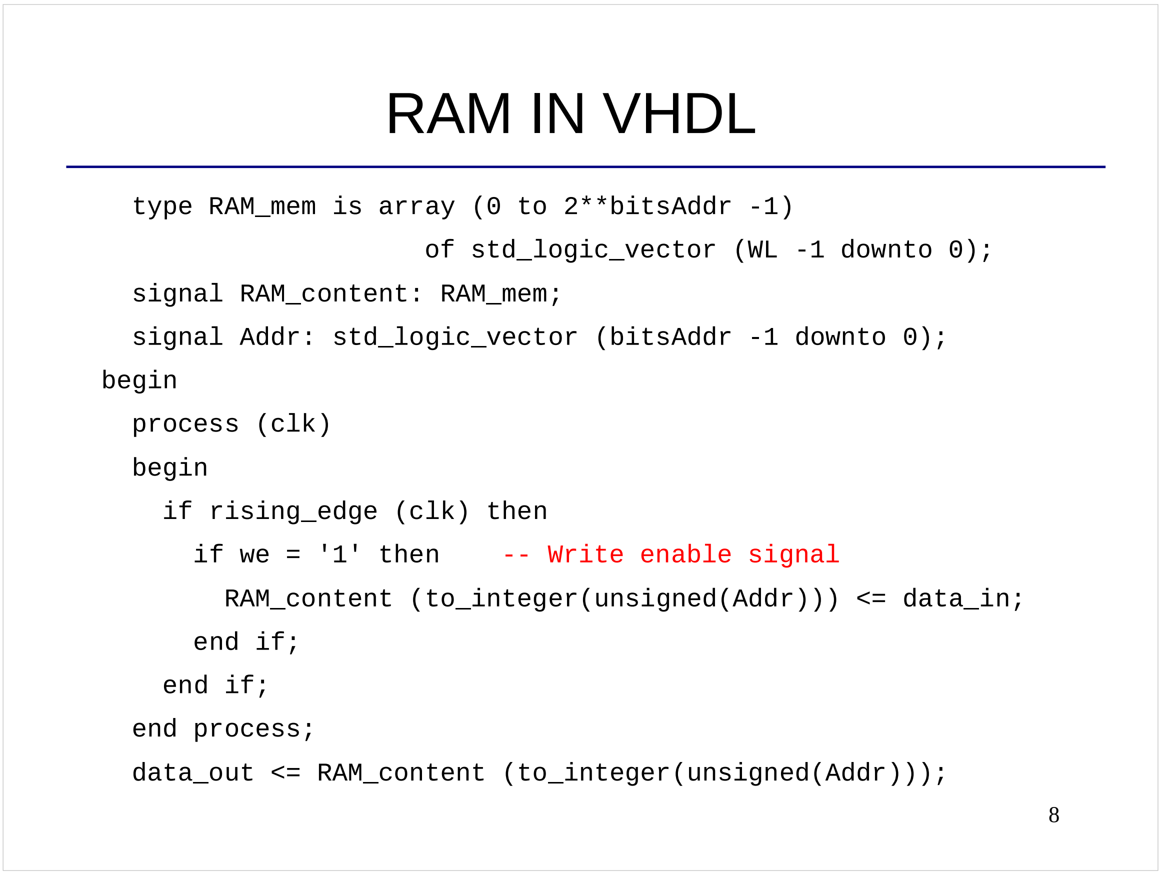
- Internal memory (we can create them in the FPGA):
  - RAM: Random Access Memory (we can read and write to it).
  - ROM: Read Only Memory (fixed values that we cannot modify).
- External memory (outside the FPGA):
  - RAM and ROM: They can be external as well.
  - SRAM: Static RAM (static = does not need refresh).
  - SDRAM: Synchronous Dynamic RAM (dynamic = needs refresh).
  - DDR SDRAM: Double Data Rate SDRAM (upper and lower edge).
  - FLASH (non volatile = the content is not erased without power).
- The DE2-115 board includes 2x 64MB SDRAM, a 2 MB SRAM and an 8 MB FLASH.



ROM IN VHDL

- Example of a ROM of 8 addresses and 4 bits in each address:

```
...
  type ROM_mem is array (0 to 7) of
       std_logic_vector (3 downto 0);
  constant ROM_content: ROM_mem := (0 => "0001",
                                  1 => "0101",
                                  2 => "0000",
                                  3 => "0111",
                                  4 => "0001",
                                  5 to 7 => "1111");
  signal Addr: std_logic_vector (2 downto 0);
begin
  z <= ROM_content(to_integer(unsigned(Addr)));
...
```



RAM IN VHDL

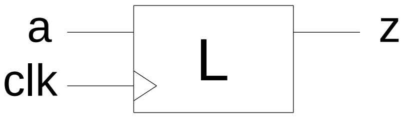
```
  type RAM_mem is array (0 to 2**bitsAddr -1)
                     of std_logic_vector (WL -1 downto 0);
  signal RAM_content: RAM_mem;
  signal Addr: std_logic_vector (bitsAddr -1 downto 0);
begin
  process (clk)
  begin
    if rising_edge (clk) then
      if we = '1' then    -- Write enable signal
        RAM_content (to_integer(unsigned(Addr))) <= data_in;
      end if;
    end if;
  end process;
  data_out <= RAM_content (to_integer(unsigned(Addr)));
```

DUAL-PORT RAM

- Dual-Port: memory in which it is possible to write on a memory location and read from another one. Obviously, it uses more resources (more area) than a single-port memory.
- wAddr: write address; rAddr: read address.

```
process (clk)
begin
  if rising_edge (clk) then
    if we = '1' then
      RAM_content (to_integer(unsigned(wAddr))) <= data_in;
    end if;
    data_out_reg <= data_out; -- If we want to register the output
  end if;
end process;
data_out <= RAM_content (to_integer(unsigned(rAddr)));
```

FIFO or BUFFER



$$z[n]=a[n-L]$$

- FIFO = First-In First-Out.
- It simply delays the inputs L clock cycles. The wordlength of the input and output is WL:

```
type FIFO is array (0 to L-1)
             of std_logic_vector (WL -1 downto 0);
```

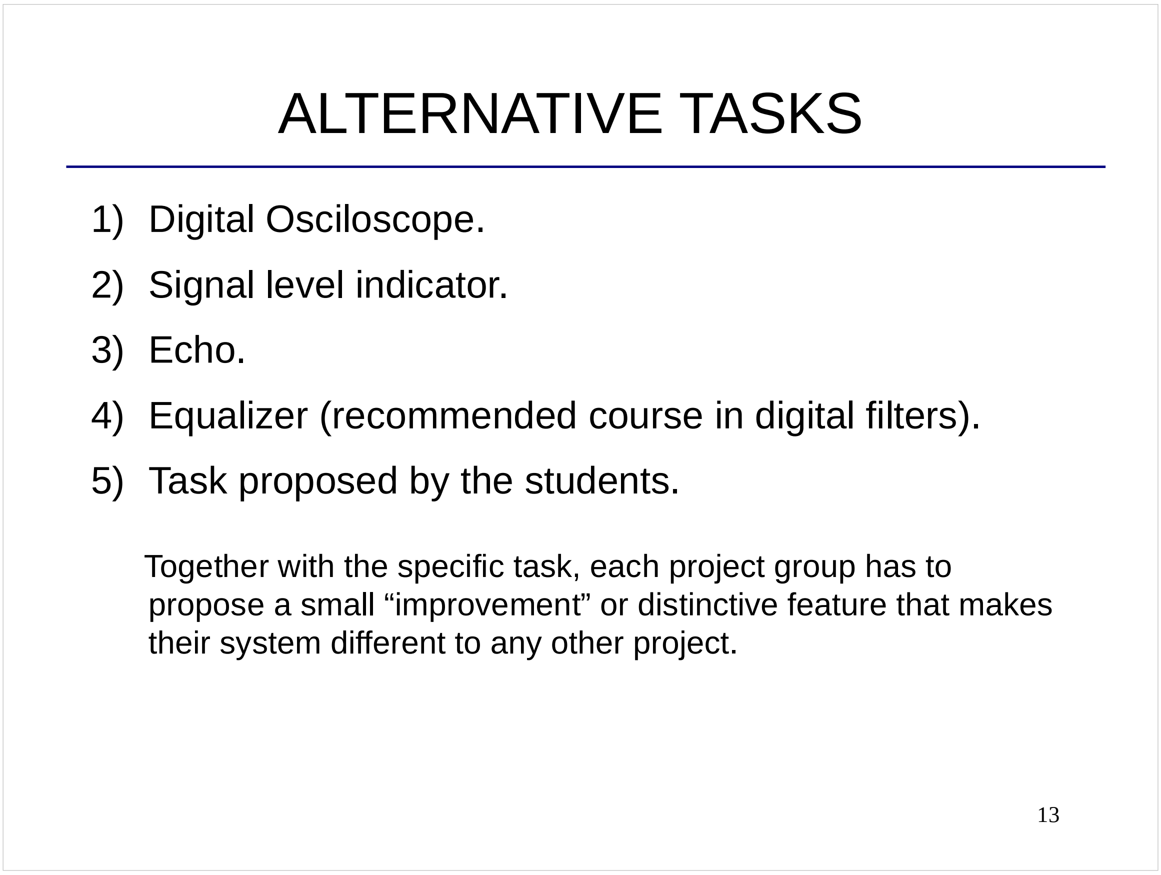
- The buffer can be implemented in a similar way as a shift register, i.e., moving each word to the next position every clock cycle (preferable for small L) or by using a memory (preferable for large L).

PROJECT

- Aim: Develop a complete functional system using the DE2-115 board and the FPGA included in it.
- Task: Use the DE2-115 board to modify/analyze a sound signal (in stereo) that is generated by a phone. The input sound signal shall be converted into digital form and modified on the FPGA. The output sound shall be sent to the loudspeakers. Information related to the signal will be displayed graphically on a VGA attached to the board. The red and green LEDs and 7-segment displays on the DE2-115 board can also be used for this purpose. Everything can be controlled using the push buttons and switches on the board, or from an attached keyboard.

GENERAL REQUIREMENTS

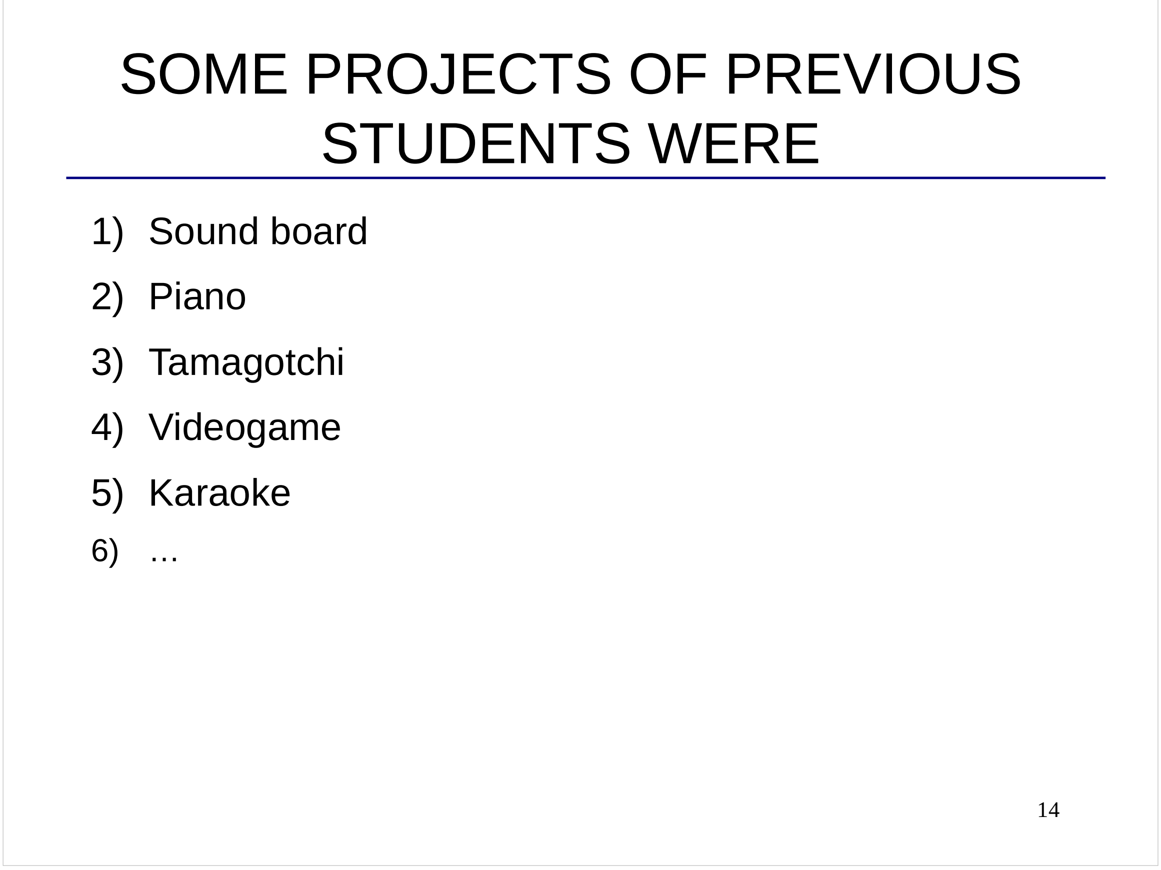
- It has to fit in the DE2-115 board and in the FPGA.
- All the hardware on the board can be used: push buttons, displays, switches, SRAM memory, infrared port, etc.
- All groups must implement a digital volume control and balance for the incoming sound signal. The volume shall have at least 10 levels of resolution, and the balance shall also have at least 10 levels.
- Apart from these general requirements, each project group has to choose an alternative task.



ALTERNATIVE TASKS

1) Digital Osciloscope.
2) Signal level indicator.
3) Echo.
4) Equalizer (recommended course in digital filters).
5) Task proposed by the students.

Together with the specific task, each project group has to propose a small “improvement” or distinctive feature that makes their system different to any other project.



SOME PROJECTS OF PREVIOUS STUDENTS WERE

1) Sound board
2) Piano
3) Tamagotchi
4) Videogame
5) Karaoke
6) …

PHASES OF THE PROJECT

- Create the project groups.
- Define the requirements of the system:
  - Write the Requirement Specification document.
- Hardware design:
  - Do the hardware design of the system.
  - Write the Design Specification and the Project Plan documents.
  - First Presentation.
- Description in VHDL:
  - Do the VHDL description in the lab.
  - Write the Project Report document.
  - Final Presentation and Demo.

CREATING THE GROUPS

- Groups of approx. 6 people.
- Decide your group and sign up between Sep. 8th and 9th.
  - Use an existing excel sheet in Teams
- Group members must:
  - Pass labs 1 and 2 to be allowed to do the project (deadline Sep. 12th).
- We will publish the list of project groups on Sep. 9th
- Lab deadline on Sep. 12th may force changes to project groups

DEFINITION OF REQUIREMENTS

- As soon as your group is confirmed (on Sep 10th), book the first meeting with the supervisor.
- To book the meeting, the supervisor will have a list on the board outside the MUX:en lab, where you can write your group number.
- First meeting (Sep. 12th or 13th):
  - Discuss the project tasks that you have thought about. Ask questions that you have. Decide the project manager.
  - After the meeting, write the Requirement Specification. Submit it 24h before the second meeting.
- Second meeting (Sep. 15th or 16th):
  - Discuss the proposed Requirement Specification with the supervisor.

HARDWARE DESIGN PART

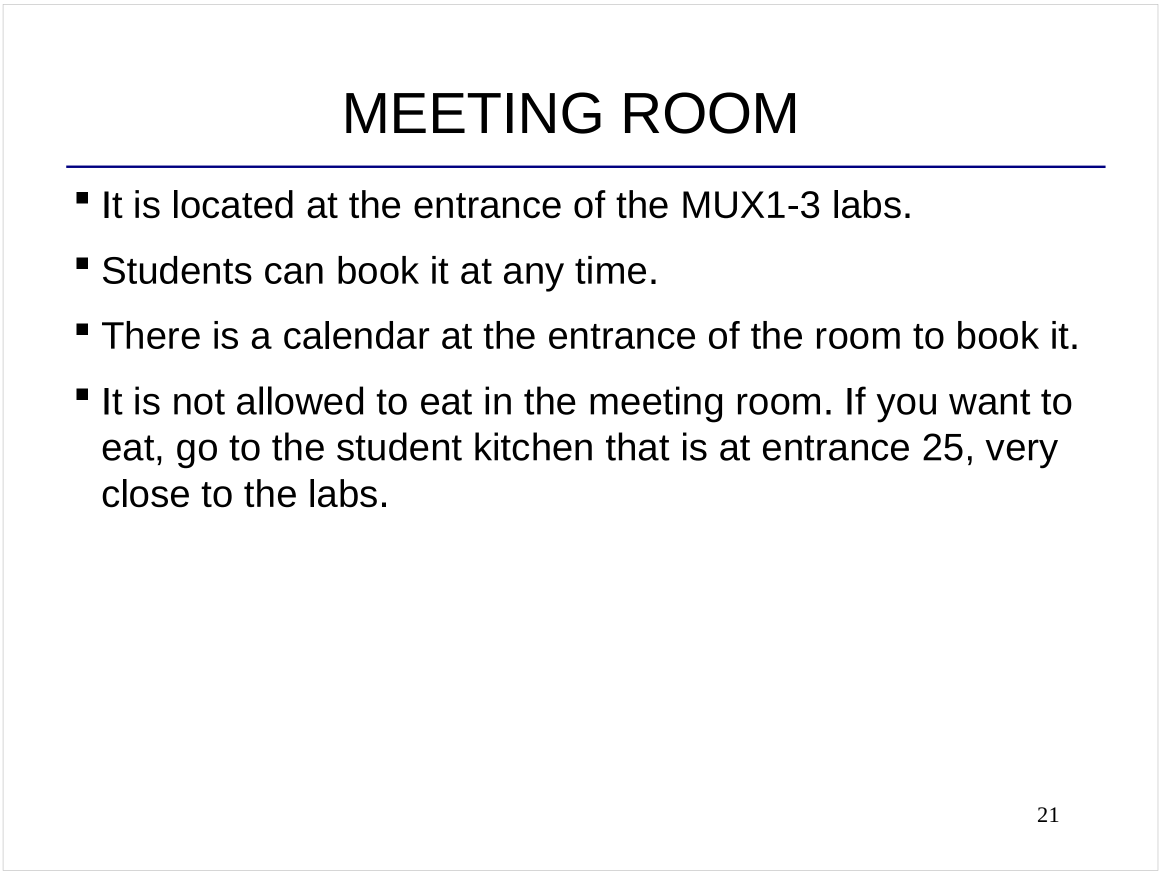
- In this phase you have to do the hardware design part of your system (no VHDL implementation).
- The result of this phase will be the Design Specification and Project Plan documents, and the First Presentation.
- The supervisor and the other teachers will be available for discussions. You will also have a third meeting with the supervisor, where you discuss the first version of the Design Specification.
- It is important that you plan this phase well. Doing the hardware design and explaining it in the document takes time and cannot be done in a few days.

DESCRIPTION IN VHDL PART

- In this phase, you have to describe your design in VHDL.
- The result of this phase will be the Project Report and the Final Presentation.
- The lab rooms will be available and you are welcome at any time.
- Your supervisor and the other teachers will be available to help you.
- If you did a good design, the description in VHDL should be easy to do. For debugging, prepare a good test bench.

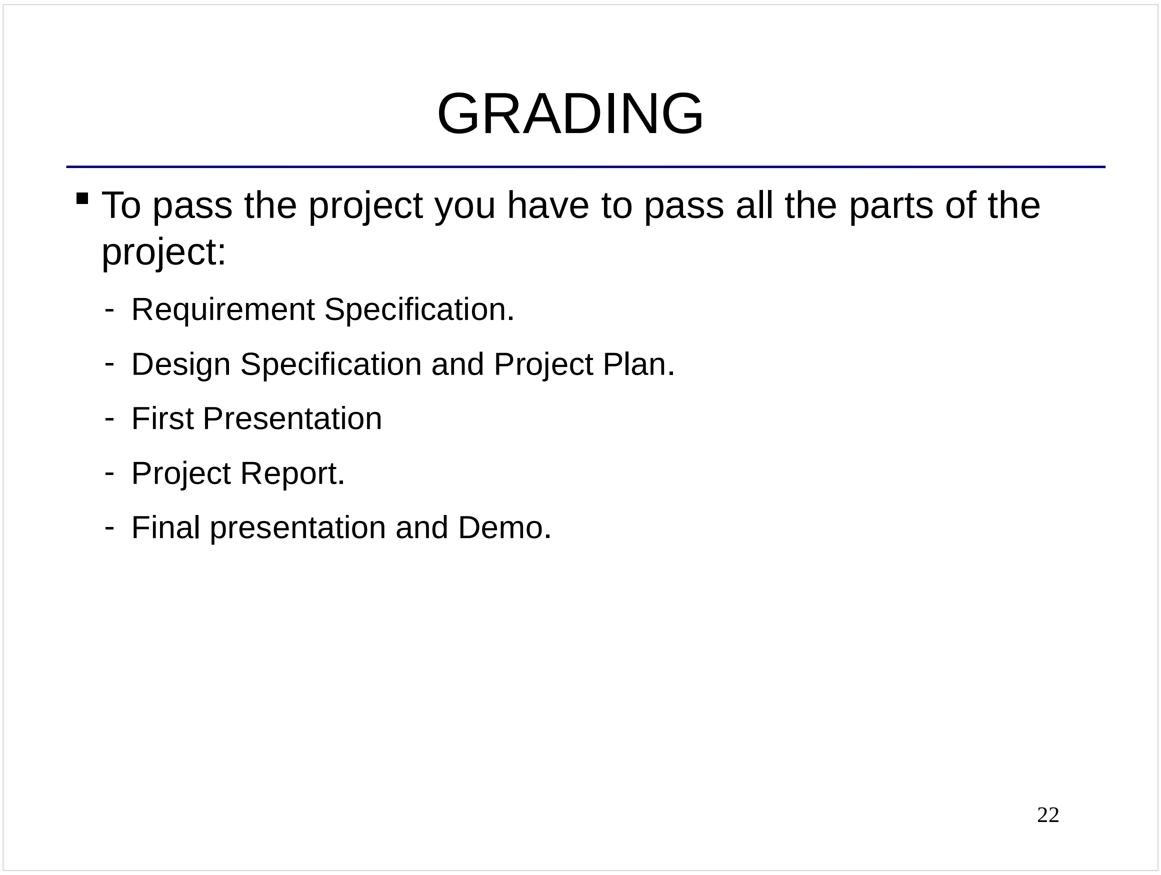
DOCUMENTS

- Information that you have to read and use:
  - Project Specification.
  - Guidelines to Write the Documents for the Project.
  - How to Design a HW System.
  - How to Describe a HW Circuit.
  - Template for the Time Reports.
- Document that you have to create (in **English**):
  - Requirement Specification.
  - Design Specification and Project Plan.
  - Project Report.
  - (Optional) Video.
  - First Presentation, Final presentation.



MEETING ROOM

- It is located at the entrance of the MUX1-3 labs.
- Students can book it at any time.
- There is a calendar at the entrance of the room to book it.
- It is not allowed to eat in the meeting room. If you want to eat, go to the student kitchen that is at entrance 25, very close to the labs.



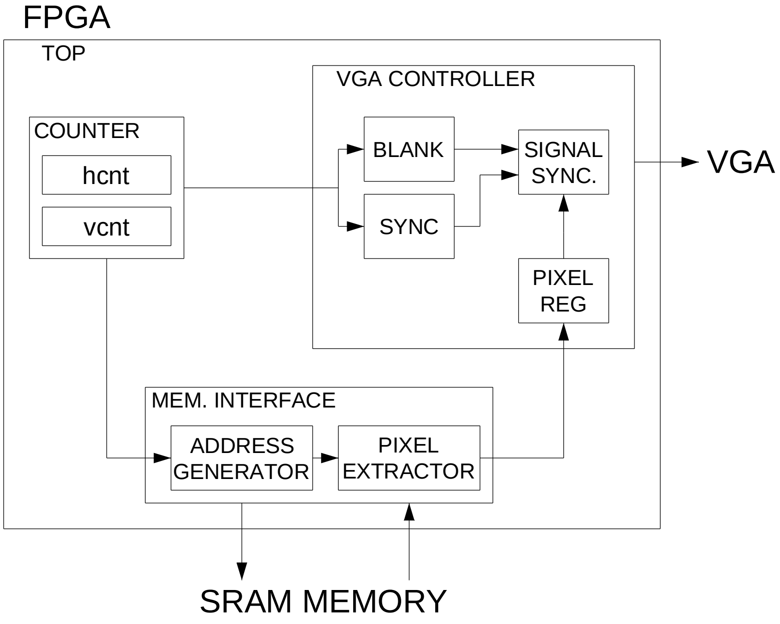
GRADING

- To pass the project you have to pass all the parts of the project:
  - Requirement Specification.
  - Design Specification and Project Plan.
  - First Presentation
  - Project Report.
  - Final presentation and Demo.

TO TAKE INTO ACCOUNT

- In today’s presentation we have seen a summary of the project. The complete information is in the **Project Specification** document. Read it carefully.
- The Project Specification is clearly divided in phases, so that you know what you have to do at each time.
- For writing documents, read also carefully the **Guidelines to Write the Documents** for the project.
- Deadlines are shown on the web page of the course.
- If you get stuck, do not waste time. Ask the teachers!

LAB 3



FPGA

TOP

COUNTER

hcnt

vcnt

VGA CONTROLLER

BLANK

SYNC

SIGNAL SYNC.

PIXEL REG

VGA

MEM. INTERFACE

ADDRESS GENERATOR

PIXEL EXTRACTOR

SRAM MEMORY

CHECKLIST FOR LECTURE 5

- Storage circuits: memory types, RAM, ROM, FIFO/Buffer.
- VHDL: type, boolean, real, time, enumeration, array, RAM, Dual-port RAM, ROM.
- Description of the project (development part).
- Description of Lab 3.

AT HOME

- Review the checklist for lecture 5 and check that you understand all the concepts and you know how to use them.
- Read the Project Specification document carefully.
- Find classmates to form a group together with them.
- Prepare Lab 3.