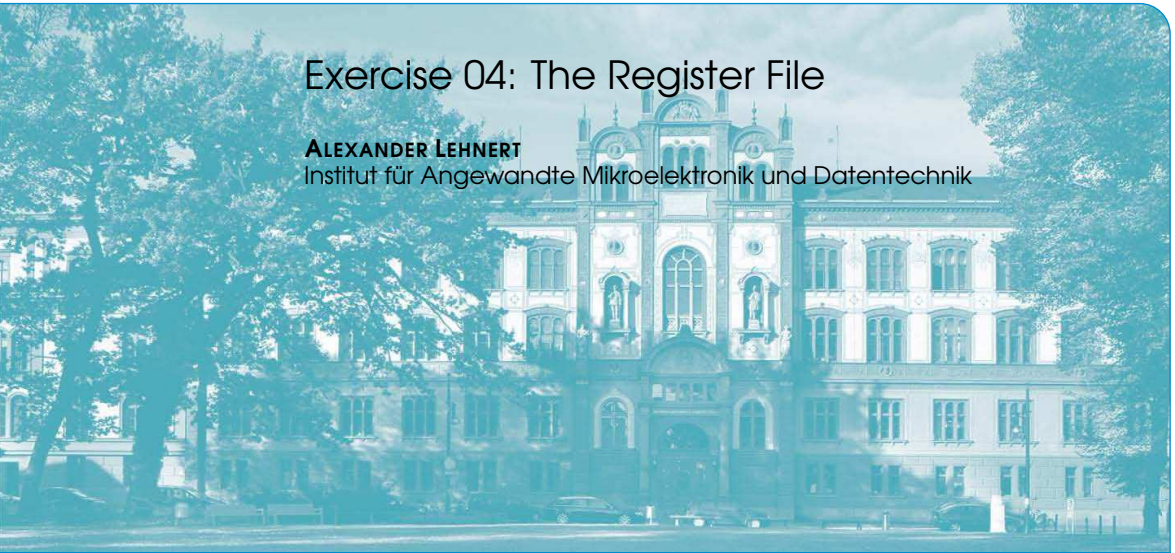
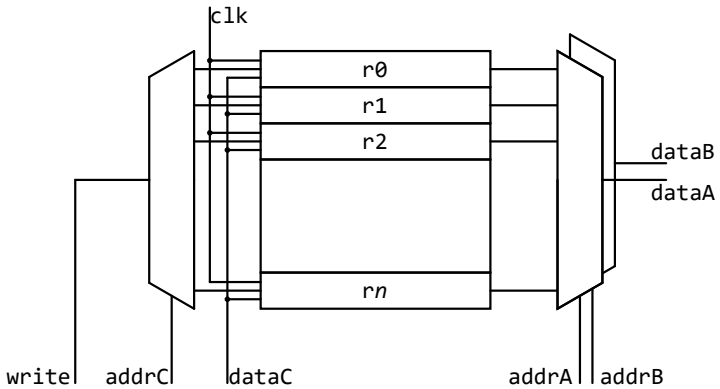


Exercise 04: The Register File

Alexander Lehnert
Institut für Angewandte Mikroelektronik und Datentechnik

Table of Contents

Block Diagram



Implementation

Three components:

- Registers
  - Synchronous process
  - Register instantiation
- Multiplexor
  - Combinational process: `if-then-else`
  - Select statement
  - MUX instantiation
- Decoder
  - Combinational process
  - `for-generate`
  - Decoder instantiation

Easy way: Integer and Type Cast

VHDL Tips and Tricks: Integer Type Cast

Integer Type Cast

```
data_a <= regs(to_integer(unsigned(addr_a)));

regs(to_integer(unsigned(addr_c))) <= data_c;
```

- Registers are a vector of `std_logic_vectors`.
- We want to access them at the index defined by the address.
- Solution: Simply cast the address from `std_logic_vector` to `integer`

Evaluation: Critical Path

- Critical Path: longest combinational path between two synchronous elements
- Determines the maximum frequency
- Timing analysis: measure the delay all combinational paths to find the critical one
- Requires all combinational paths to be terminated by registers

Type definition

Vectors

```
type <NAME> is array (<RANGE>) of <SUBTYPE>;

type mem is array(natural range <>)
        of std_logic_vector(word_size-1 downto 0);
```

Logarithm

- Bit-widths are the logarithm of the maximal representable number
- How to calculate bit-widths in VHDL?

Logarithm

```
log2 (int x) {
    int i = 0;
    while (x > 1) {
        x = x/2;
    }
}
```

```
library ieee;
use ieee.std_logic_1164.all;

package example_pkg is
    function log2 (a: integer) return integer;
end example_pkg;

package body example_pkg is
    function log2 ( x: integer ) return integer is
        variable i : natural;
    begin
        i := 0;
        while 2** i < x loop
            i := i+1;
        end loop;
    end function;
end example_pkg;
```

Task Overview

1. Implement log_2 function in VHDL
2. Implement register file
3. Analysis: Mapping of the register file
4. Analysis: Critical path
5. Bonus: Program the FPGA