Exam preparation doc

Praktikum VHDL (Technische Universität München)

1. The family of the FPGA.

Spartan 3E

2. FPGA usage of the 3 cables: power cable, USB, UART.

Power cable: power up the device

Usb: connect to pc, program FPGA

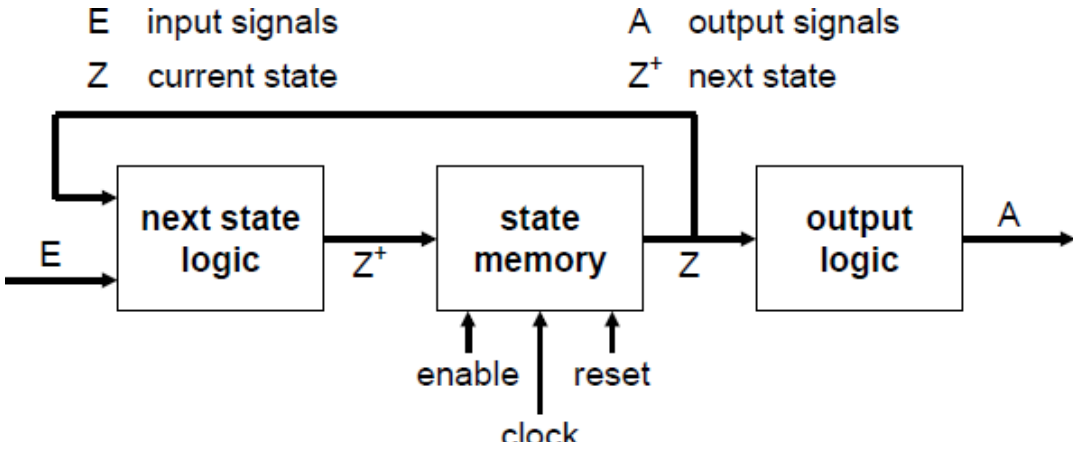
Uart: receive data from pc, start encryption, send data to pc

3. Explain the nouns (usage): entity, architecture (Lecture 1 project introduction slides 42-43; Lecture 2 vhdl1 slides 17), process (Lecture 2 vhdl1 slides 20), component.

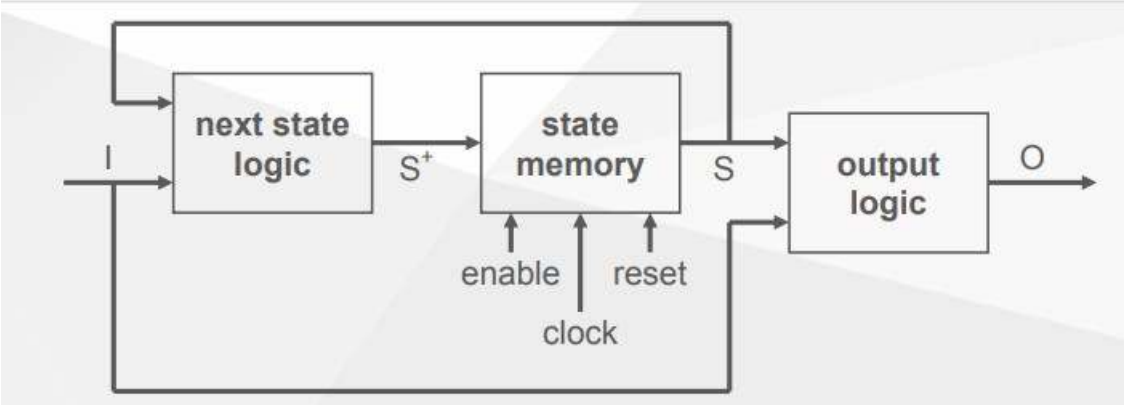
- An **entity** describes the external view of a circuit (component), i.e. the number and the type of its input and output ports.
- **Architecture**: provide an internal view of the circuit, i.e. they describe the function of the associated entity and how the circuit is realized.
- **Process**: sequential execution of statements
- **Component**: Each component instance is a copy of the entity representing the subsystem, using the corresponding basic architecture body. number of in and out ports along with their types
- **Port**: model data in/out

4. Draw the diagram of Moore FSM. Describe the parts. Lecture 3 vhdl2 slides 45\47.

The output signals of a Moore FSM are only a function of the current state of the FSM. The input signals can influence the output only via the state memory



Mealy type machine: output depends on both the current state and input values.



5. The difference between and Moore and Mealy FSM. (Book 150)

Moore: output is only function of the current state. Input can only affect the output through memory.

Mealy: output is function of current state and input signal.

General: Finite State Machines (FSM): Systematic method for describing and designing digital systems. Synchronous sequential circuit with three functional blocks: State memory, Next state logic, Output logic

6. The difference between a variable and a signal (2 aspects: the place, variable stores intermediate data, signal is the wire connecting hardware blocks, Lecture 1 project introduction slides 20).

Variable: is declared and used only inside a process and store intermediate data (No timing behavior).

Signal: is declared and used in the whole architecture (also used inside of processes) and use as wiring between hardware blocks.

- **Signals**: Connect components and processes, have a value over time
- **Variables**: Local to a process, used for intermediate calculations
- **Constants**: Fixed values, cannot be changed after declaration

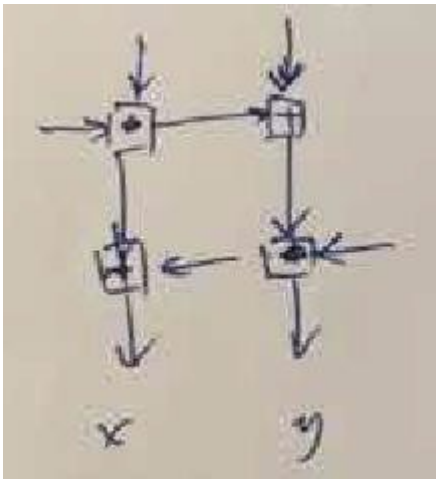
7. The difference between transaction and event (Lecture 3 slides 19).

Transaction: A transaction is a pair of values (new_signal_value, simulation_time). When executing a signal assignment statement, a transaction is scheduled for later execution.

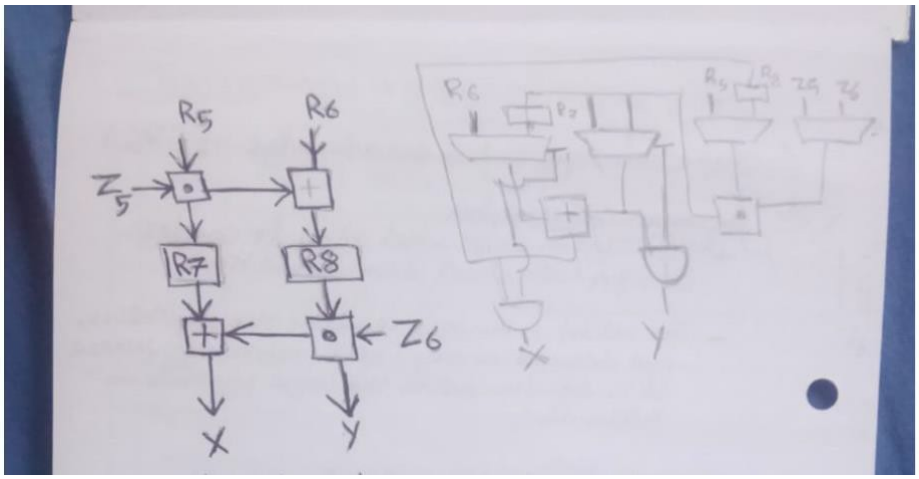
Event: When at simulation time of a transaction the new signal value differs from the previous value, we refer to this as the occurrence of an event

Processes react on events, not on transactions!

8. Correct errors in a testbench (It seems wrong to be too decent, so I didn't do it at first, but it was too late)



9. If using only one multiplier and one adder, draw the datapath (using multiplexers & registers).



10. Design flow of VHDL design (fill in the blanks). Lecture 1 project introduction slides 30-39.

Vhdl coding ☾ simulation ☾ synthesis ☾ implementation ☾ generating configuration file ☾ program FPGA

11. The difference between *after* and *wait for* statements in hardware synthesis.

Wait used to suspend a process cause synthesis failure / just for simulation purpose.

After for single signal assignment statement and ignored by synthesizer / Can be used to create stimuli in a testbench. Ignored by synthesis tool => No relevance to hardware.

Difference to the wait for statement: wait for creates an error during synthesis

12. One advantage and one disadvantage of FPGA in comparison of ASIC.
Advantages: semi-custom chips ☾ allow complex design and short development time / repeated modification / cost-efficient for small production numbers and brings the additional benefit of short production times.

Disadvantages: The power consumption is more, and programmers do not have any control on power optimization in FPGA.

13. Name 3 basic components of a SLICE.

- 2 functions generators F and G.

- Special carry logic.

- Multiplexer

14. Lecture 4 hardware slides 6-7, draw the RT schematics according to the VHDL codes.

There is one more assignment statement *S*<= *Z and C* after the end of *if* modulo, but still in the process. (Pay attention to the variable and signal => **need D-flipflop or not**.)

15. Multiple driver problem. What? When will it occur? How to avoid? Lecture 4 hardware slides 12-13.

It occurs when one signal is driven/changed in more than one processes(drivers)

- Only one driver for each signal / There is no resolution mechanism (beside the resolution function)

- Avoid it ☾ Use the same edge of clock signal

16. What is the goal of RCS in the lab? How does RCS1 and RCS2 do in general to reach this goal?

The goal is to be able to implement the idea algorithm in the FPGA. Since the direct implementation require more resources than FPGA can offer. The design must be changed so that less CLB and multipliers are needed. In RCS1 the same algorithm is designed with only one round module and output transformation (result of every round is saved in register and served to next round as input). In RCS2 the round module is adapted to use only one multiplier, one adder and 5 Xors by splitting the round into 4 partial steps thus using less resources.

17. What's Gated Clock? What problem may it have?

Gated Clock: Combinational logic between the clock source and a clock sink (e.g. register)

May cause timing problem. And it might exist by accident, during optimization the skew in clock distribution or switching off certain circuit for energy saving

18. The difference between *behavioral models* and *structural models*. Write some codes as examples, no need to be complete. Lecture 2 vhdl1 slides 17.

- **Behavioral** models describe the behavior (the function) of a circuit (component), i.e. the values of the output signals as a function of the values of its input signals. A pure behavioral description contains only process statements and signal assignments.

-**Structural** models describe a circuit (component) as a hierarchical structure of subcomponents interconnected by signals (hierarchically structured netlists). A pure structural description contains only component instantiation statements.

- **Behavioral modeling (Sequential code):** describing the behavior (the function) of a circuit (component), i.e. the values of the output signals as a function of the values of its input signals
- **Data flow modeling (Concurrent code):** also behavioral modeling, where the modeled function of the circuit (component) is a simple transformation of the input into the output signals

- **Structural modeling:** describing a circuit (component) as a hierarchical structure of subcomponents interconnected by signals (hierarchically structured netlists)

19. Delta delay. Fill in the chart. Lecture 3 vhdl2 slides 27-33. (Book 128)

Even with a delay of 0 fs a new simulation cycle is performed, although the simulation time does not change; this is called a **delta delay (an infinitesimally small-time step).**

The concept of delta delays guaranties the causality of the statements. Thus, a simulation is possible, even if no delays are known.

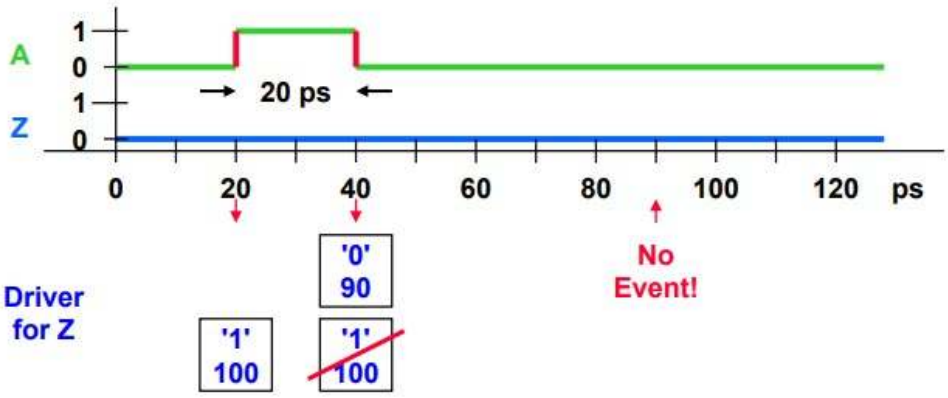
20. Draw the different waveforms of *transport delay* and *inertial delay*? Lecture 3 vhdl2 slides 34-39.

Transport: Rules for inserting a new transaction (Vn,Tn):

- All transactions (Vi,Ti) with Tn <= Ti are deleted.
- The new transaction (Vn,Tn) is appended to the end of the list.

Only the **delay influences** the result, the **signal value** does **not**.

```
begin
  if A='1' then
    Z <= transport A after 80 ps;
  else -- A='0'
    Z <= transport A after 50 ps;
  end if;
end process AsymDelay;
```



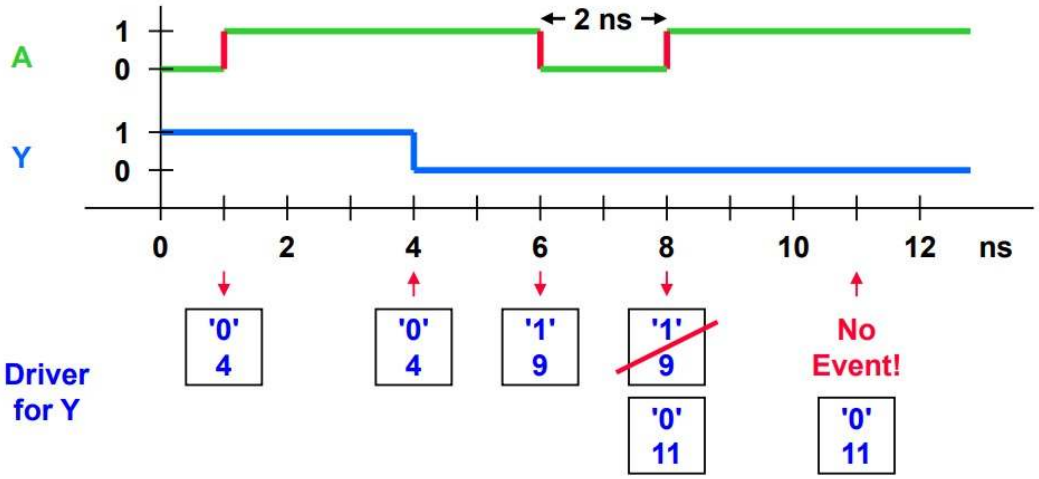
Inertial:

Rules for inserting a new transaction (Vn,Tn):

- All transactions (V_i,T_i) with T_n <= T_i are deleted.
- The new transaction (V_n,T_n) is appended to the end of the list.
- In the interval T_0 <=T_i <= T_n an uninterrupted sequence of transactions (V_i,T_i), ..., (V_n,T_n) with V_i = V_n is maintained, all other transactions are erased.

Delay and **signal value influence** the behavior.

```
signal A: bit := '0';
signal Y: bit := '1';
...
INV: process(A)
begin
  Y <= inertial not A after 3 ns;
end process INV;
```



21. Sensitivity list -> *wait on* statements.

- Process
  - Sequential execution of statements (like a conventional programming language)
  - Sensitivity list or „wait“-statements

```
Proc_label: process(X,Y)
begin
    Z <= X and Y;
end process proc_label;
```

```
Proc_label: process
begin
    Z <= X and Y;
wait on X,Y;
end process proc_label;
```

Not synthesisable

  - „wait“-statements just for simulation purpose (more information later on)

Wait on Signal Sensitivity Clause:

```
wait on CLK_SIGNAL, ENABLE;
```

- The process resumes execution when at least one of the named signals changes its value.

Wait Until Condition Clause:

```
wait until CLOCK = '1';
```

- The Boolean expression must be true for the process to resume.

Wait for Time-out Clause:

```
wait for 1 ms;
wait for 2 * ClockPeriod;
```

- The process resumes execution after the given delay time.

Wait during Initialization:

```
wait;
```

- The process executes exactly once during initialization and remains inactive afterwards.

Extra:

VHDL: V VHSIC (Very High-Speed Integrated Circuit) or VLSI / H Hardware / D Description or Design / L Language

- Common programming language to describe hardware.

- Concurrent & Hierarchical Modeling

- Support of different design styles (Behavioral ↔ Structural) Model:

– idealized description (an abstraction) of a real object

– expresses important properties and neglects minor properties The major objectives for using HDLs:

– avoidance of design errors requiring redesigns

– minimization of design cost and time

Design flow of VHDL design:

VHDL Coding:

Describes our model (Behaviour, Structure, Dataflow)

Simulation:

Check syntax / Testbench including the stimuli Output: Waveforms / Logical verification of the model Synthesis:

Input: working structural or behavioral VHDL model

Output: Gate - RTL Netlist / No information about the later Hardware Implementation:

Translation: Synthesized netlist and constraint file is translated into a Xilinx file.

Map: Devices of the netlist are mapped to the type of HW resources Place & Route: Exact placement and wiring is defined

Generation of the Configuration File & Programming:

Based on the Place & Route step the configuration file (here: *.bit) is generated.

Testbench: a module that is used for testing the functionality of a design module by simulation. Entity declarations of testbench modules have no input and output ports.

Behavioral architectures can appear in two forms:

Algorithmic models: describe the function (the behavior) of a circuit (component) just as in a (conventional) programming language by means of a sequence of statements listed in a single process, especially used for the Boolean Equations level.

- Dataflow models: describe the circuit's function by way of the data flow between communicating processes (concurrent, parallel in time), well suited to model circuits on register-transfer level.

Process Statements: There are two (equivalent) forms.

Processes without a sensitivity list (containing wait statements): Statements in the process body are executed in sequence. After the last statement, the first statement is executed next without updating the simulation time.

- Wait statements are used to suspend processes.
- To avoid infinite loops at least one wait statement must exist.
- Not for synthesis => Error

Processes with a sensitivity list (containing no wait statements):

- The sensitivity list is a list of signal names. A process is activated when at least one of these signals changes its value.
- This process form is equivalent (just in the simulation) to a process with a wait-on-signal statement.
- This type of process does not need to contain wait statements.

Simulation time vs execution time:

- simulation time: The time, in which the circuit being modeled is deemed to operate. Simulation time is measured starting from zero at the start of execution and increasing in discrete steps as events occur in the model. That is why this is called discrete event simulation.
- execution time: The time needed for the VHDL model running on a host computer.

Signal Drivers:

A signal driver is a chronologically ordered list of all transactions scheduled for this signal, i.e. a list of value pairs (signal_value, simulation_time) = (Vi,Ti).

Hardware debugging:

- No wait statement is synthesizable! ☾ Use Sensitivity List for processes instead of „wait on".
- If you want to model a delay of a module use „after" instead of „wait for".
- Avoid latches in your design.
- Consider every signal/output in all cases and if/else statements.

How is Start generated?

The signal Start is generated by an external sequential module. External module shares the same Clock signal with the control unit. Start is high at second rising clock edge.

Post-Place & Route Simulation

- Synthesis: behavioral description → gate netlist

- Translation: merges gate netlist with additional constraints.

- Mapping: The gates are mapped to the resources on the FPGA.

- Place & Route: The resources gets connected by wires → Timing models of gates + wires.

Cryptography

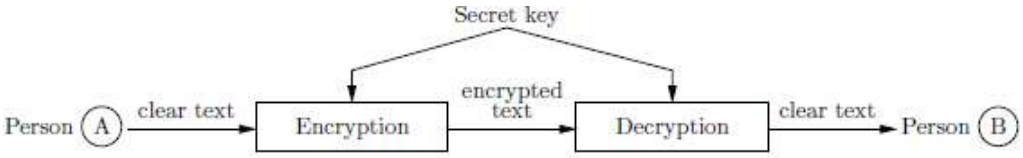


Figure 2: Symmetric encryption

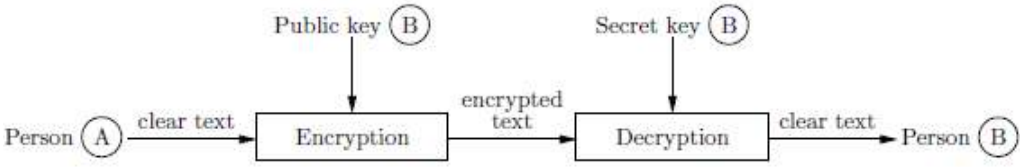


Figure 3: Asymmetric encryption

Delay Estimation: Register

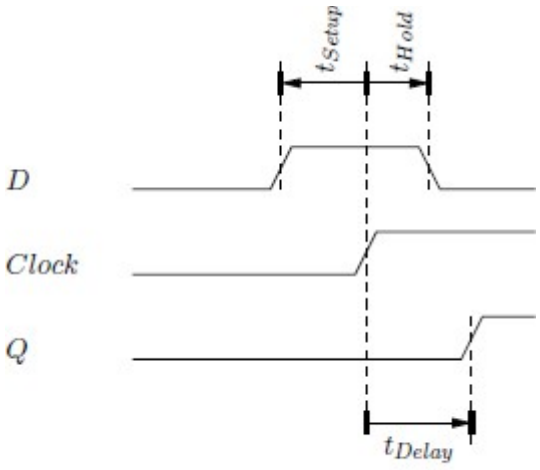


Figure 45: Propagation delay register

The propagation delay of registers are characterized by three times: The switching time

tdelay, the setup time tsetup and the hold time thold.

Event-Driven Circuit Simulation

Problem: Very large digital circuits & Simulation is time-consuming

- Solution: Event-Driven Simulation (Very efficient) –Digital inputs (stimuli)
  - –Simulation is done in time intervals
  - –Time intervals defined by input events