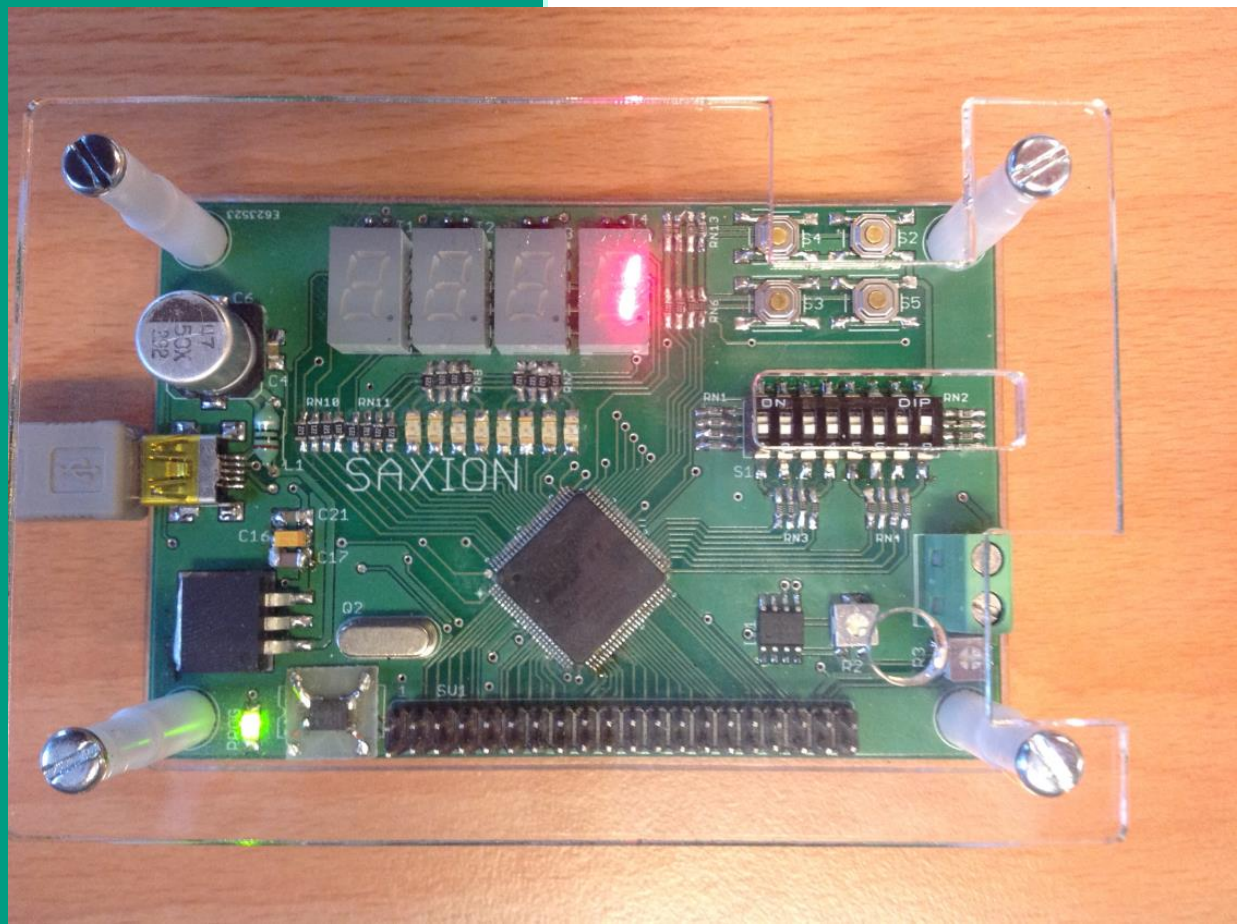
Very High Speed Integrated Circuits
Hardware Description Language

Lesson 1
VHDL



SAXION
UNIVERSITY OF
APPLIED SCIENCES

SAXION UNIVERSITY OF APPLIED SCIENCES

Lesson outline

- Goal
- Assessments
- Non-appearance
- Subjects
- Plan
- Questions.

SAXION UNIVERSITY OF APPLIED SCIENCES

Goal

- The students are able to describe hardware with VHDL and to implement it on a testboard .

SAXION UNIVERSITY OF APPLIED SCIENCES

Practical

- Attendance at pratical lessons is mandatory.
- If you are not able to attend please contact the teacher.
- If you miss more than two practical lessons: no credits

Assessment

- Theoretical/ practical examination

SAXION UNIVERSITY OF APPLIED SCIENCES

Responsibilities

- The student is responsible for the hard and software, which must be used by the next group.
- Defects must be reported immediately

Resources

- Reader on BB.
- The VHDL Cookbook on BB
- Tutorial Quartus on BB.
- YouTube

SAXION UNIVERSITY OF APPLIED SCIENCES

Course subjects.

- History VHDL
- VHDL Data Types
- VHDL Operators
- VHDL Based Synthesis of Digital Hardware
- Synthesis Models of Gate Networks
- Seven-segment LED Decoder
- Multiplexer
- Tri-State Output
- Flip-flops and Registers

SAXION UNIVERSITY OF APPLIED SCIENCES

Course subjects.

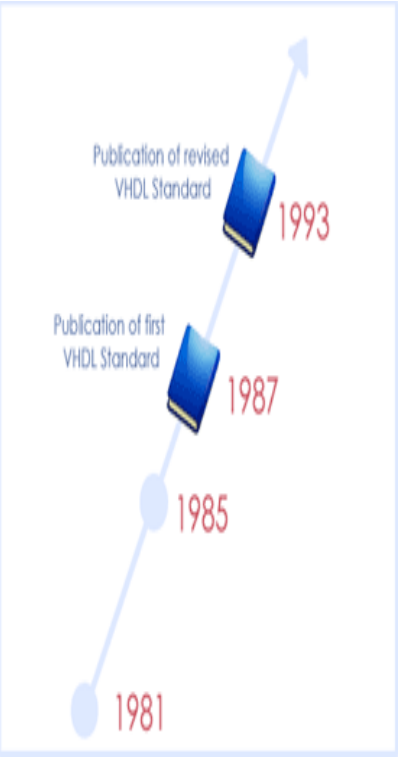
- Inferred Latches
- Counter
- State Machine
- ALU
- Multiply and Divide
- Memory
- Hierarchy

SAXION UNIVERSITY OF APPLIED SCIENCES

Today ……

- History
- Data types
- VHDL operators
- Synthesis
- Gate Networks
- Exercise 1 ‘Adder’
- Introduction Quartus

History



1981: initiation: USA Department of Defence
1983-85: baseline language Intermetrics, IBM, IT
1986: rights to IEEE
1987: first publication of IEEE Standard
1994: Revised standard VHDL(1076-1993)

A new version approximately every three years
2006: new version (more analogue description possible)

The VHDL standard IEEE 1076-2008 was published in January 2009.

Alternatives to VHDL

Verilog: biggest next to VHDL

AHDL: Specific to ALTERA company

SystemC: supported by XILINX

SAXION UNIVERSITY OF APPLIED SCIENCES

Why a HDL instead of CPU

- CPU and GPU are sub-optimal for a specific problem because of their general purpose structure
- Only FPGA can guarantee fixed low-latency when the problem has real-time constraints
- FPGA can be one step to ASIC design.

For example: Historically, the video decoder world was all software. Now all media processor have specific computing structure for video-decoding that where once prototyped in FPGA

Variables and Signal types

- Boolean integer real etc
- BIT (IN OUT) (Do not use them in QUARTUS!)
- STD_LOGIC (U, X,0,1,Z,W,L,H,-)
- ‘1’ ‘0’ single quotes 1 bit
- “1001” bitvector

STD_LOGIC (U, X,0,1,Z,W,L,H,-)

- 'U': uninitialized. This signal hasn't been set yet.
- 'X': unknown. Impossible to determine this value/result.
- '0': logic 0
- '1': logic 1
- 'Z': High Impedance
- 'W': Weak signal, can't tell if it should be 0 or 1.
- 'L': Weak signal that should probably go to 0
- 'H': Weak signal that should probably go to 1
- '-': Don’t care

Std_logic_vector

- signal t1 : std_logic_vector(7 downto 0); --7th bit is MSB and 0th bit is LSB

- signal t2 : std_logic_vector(0 to 7); --0th bit is MSB and 7th bit is LSB here.

- You are free to use both types of representations, but make sure that other parts of the design are written accordingly.

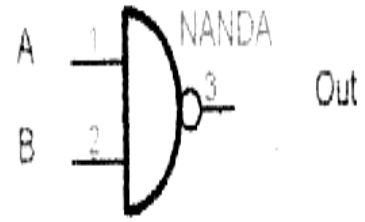
SAXION UNIVERSITY OF APPLIED SCIENCES

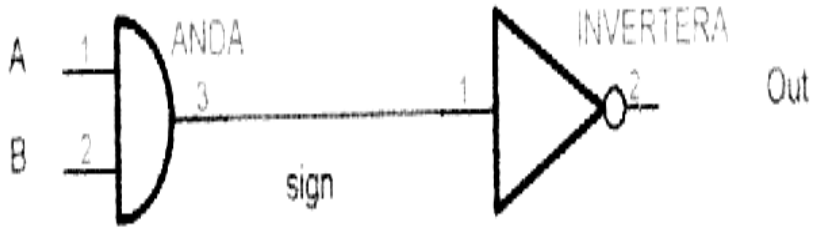
VHDL Operators

Table 1.1 VHDL Operators.

VHDL operator	Operation
+	Addition
-	Subtraction
*	Multiplication
/	Division
MOD	Modulus
REM	Remainder
&	Concatenation-used to combine bits
SLL	Logical shift left
SRL	Logical shift right
SLA	Arithmetic shift left
SRA	Arithmetic shift right
ROL	Rotate left
ROR	Rotate right
=	Equality
/=	Inequality
<	Less than
<=	Less than or equal
>	Greater than
>=	Greater than or equal
NOT	Logical NOT
AND	Logical AND
OR	Logical OR
NAND	Logical NAND
NOR	Logical NOR
XOR	Logical XOR
XNOR*	Logical XNOR

Gate networks





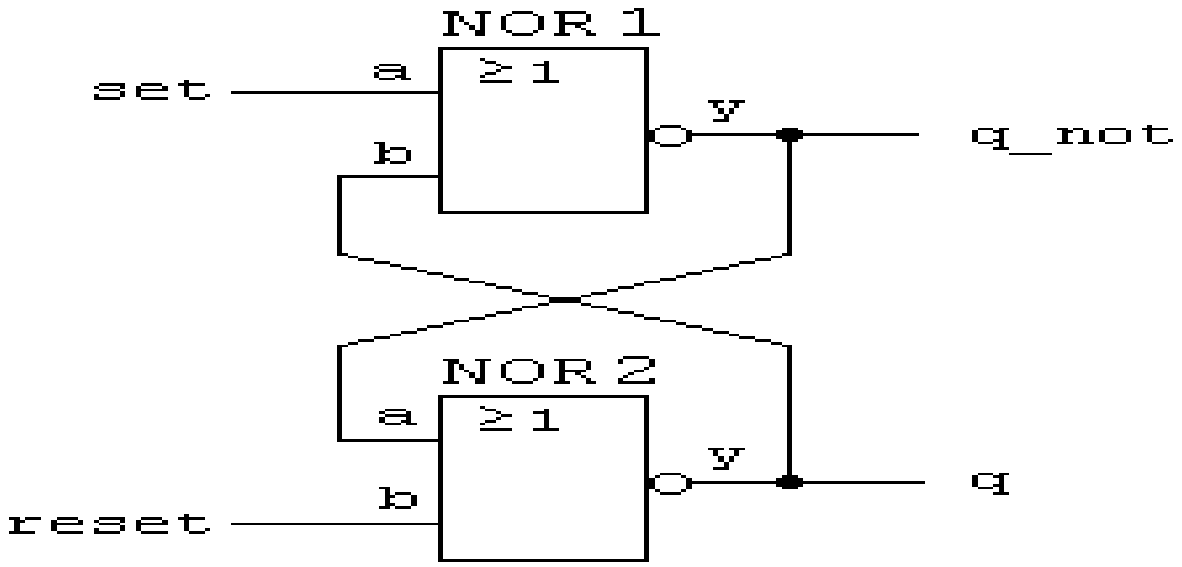

```
ENTITY nand IS
PORT (A,B : IN STD_LOGIC;
        Out: OUT STD_LOGIC);
END nand;

ARCHITECTURE behav1 OF nand is
BEGIN
 Out <= A NAND B;
END behav1;
ARCHITECTURE behav2 OF nand is
SIGNAL  sign : STD_LOGIC;
BEGIN
 sign <= A AND B;
 Out <=  NOT sign;
END behav2;
```

Exercise 1

Design a full adder in VHDL

Exercise 2

Design a S-R element



Introduction Quartus

- Use Web edition
- Big program (takes time to install)
- (Too)Many possibilities
- This link is NOT exact how we use Quartus. Use our tutorial)

https://www.youtube.com/watch?v=TML35NuuSdk

Assignment week 1

-Install Quartus 14 (WEBedition)
-Do the Tutorial
-Do the self-test chapter 2

Submit your work on Blackboard before week 3
1 .pdf or word file, so no RAR or ZIP or .vhd files.