2.1 Modellierung von Signalwerten

a)

0 >= W >= L >= Z

Nach der Aussagendiagram gilt:

$$0 \geq W \; und \; W \geq H \rightarrow 0 \geq H \; (Transitivität)$$
$$0 \geq H \; und \; H \geq Z \rightarrow 0 \geq Z \; (Transitivität)$$

Damit gilt

Cond (0,X) $\geq$ $sup\ (0, Z) \geq 0$

b)

1. **'U' Zustand steht für 'uninitialized'**. Kann durch einen Fehler in Design erzeugt werden, sodass der Entwickler den Fehler vor der Integration zu Kreis korrigieren kann. 'U' bezeichnet auch alle uninitialisierte gespeicherte Werte und Kreiseingabe bevor die Simuliation überprüft die Typ von notwendiger Eingabe.

2. "**Don't care -** Zustand ist ein Placeholder für solche Eingangskombinationen, bei denen Ausgangssignal nicht definiert ist oder wird keine Auswirkung auf gesamtes Ziel des Kreises haben.

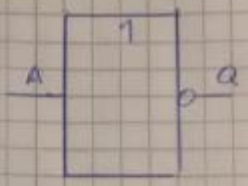
3. ghdl/libraries/ieee/std_logic_1164.vhdl: "This package defines a standard for designers to use in describing the interconnection data types used in vhdl modeling."

c) siehe signal_levels im Ordner 1_c.

2.2

a)

NOT

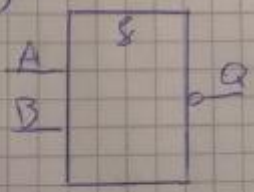


A	Q
0	1
1	0

AND

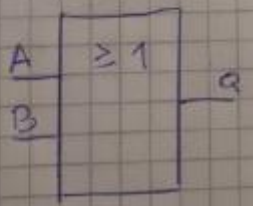
A	B	Q
0	0	0
0	1	0
1	0	0
1	1	1

NAND



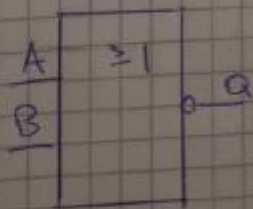
A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0

OR

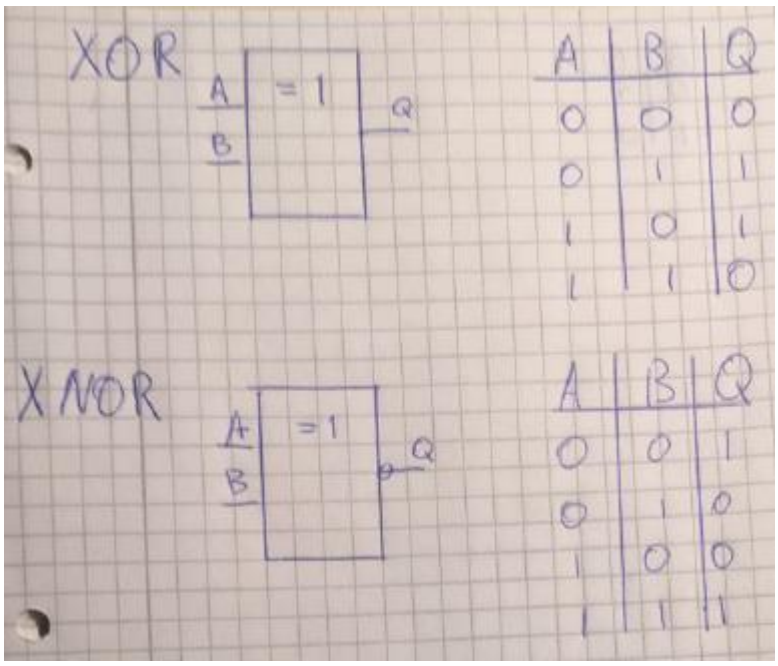


A	B	Q
0	0	0
0	1	1
1	0	1
1	1	1

NOR



A	B	Q
0	0	1
0	1	0
1	0	0
0	0	0



b) Alle Implementierungen befinden sich im Ordner 2_b.

c) Delay erweiterte XOR Gatter befindet sich im Ordner 2_c. Rolle von Delays: Signale der digitalen Schaltungen laufen mit einer Geschwindigkeit von ca 8cm/ns. FETs, die werden weit benutzt, brauchen bis zu paar hundert ps für Ein- Ausschaltung. Diese beide Trägheitsverzögerung (intertial delay) und Transportdelay müssen modelliert werden können.

Transportdelay hat nichts mit Wellbreite zu tun, modelliert nur eine Verzögerung zwischen Bausteine. Ganz im Gegenteil, Trägheitsverzögerung kann zu kurze Signale ablehnen.

d) Alle Implementierungen befinden sich im Ordner 2_d.

e) **Entity** - beschreibt alle I/O Kontakte von einem Modul. Analog zu ’public interface’ aus OOP.

Architecture - eine Funktionseinheit von bestimmtem Entity, die ein Verhalten bereitstellt. Es kann mehrere Architectures geben, wie ’extends’ in OOP.
Component declaration funktioniert als import aus OOP. Entity wird lokal instanziert, muss übereinstimmen mit referenziertem Entity, um den Component benutzen zu können.
Port map verbindet expressions benutzt in diesem Architecture mit denen, die in deklariertem Entity vorhanden sind. Beim process begin wird eine sequenzielle Teil von Architecture gelaufen mit leerem sensitivity list. Die Anweisungen werden nacheinander durchgeführt bis Ende des
Prozesses.