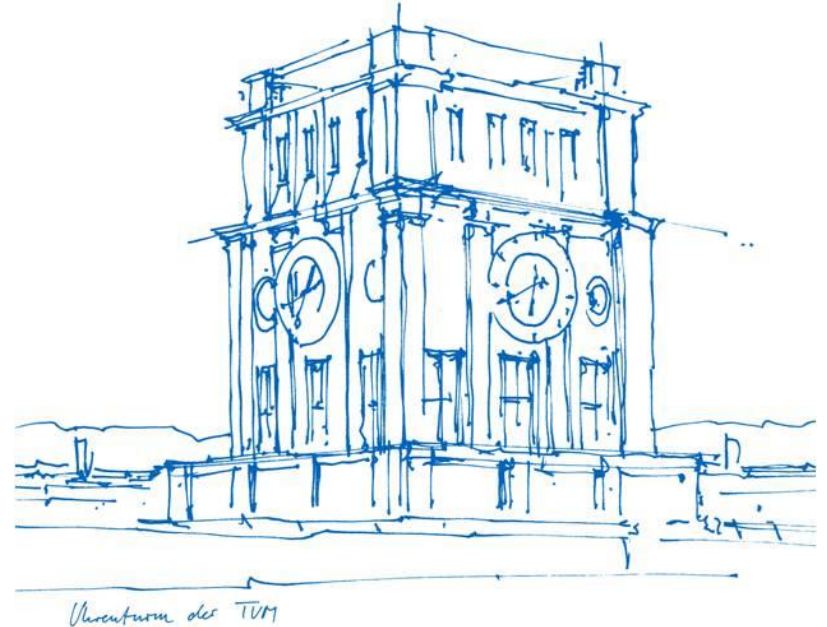


Grundlagenpraktikum: Rechnerarchitektur

SoSe 2024

~ *Danial Arbabi*

danial.arbabi@tum.de



Zulip-Gruppen

Gruppe 29

FR 12:00



<https://zulip.in.tum.de/#narrow/stream/2276-GRA-Tutorium---Gruppe-29>

Gruppe 32

FR 15:00



<https://zulip.in.tum.de/#narrow/stream/2279-GRA-Tutorium---Gruppe-32>

Tutoriums-Website



<https://home.in.tum.de/~arb/>

Disclaimer:

*Dies sind keine offiziellen
Materialien, somit besteht keine
Garantie auf Korrektheit und
Vollständigkeit.*

*Falls euch Fehler auffallen, bitte
gerne melden.*

Module in SystemC

Stichworte: Konstruktor, Sensitivity-Lists, behaviour

```
1 SC_MODULE(M1) {  
2     sc_signal<bool> x;  
3     sc_signal<bool> y;  
4     sc_signal<bool> output;  
5  
6     SC_CTOR(M1) {  
7         SC_THREAD(bhaviour);  
8         sensitive << x << y;  
9     }  
10  
11     void behaviour() {  
12         while (true) {  
13             output = x.read() | (!x.read() & y.read());  
14             wait();  
15         }  
16     }  
17 };
```

Module in SystemC

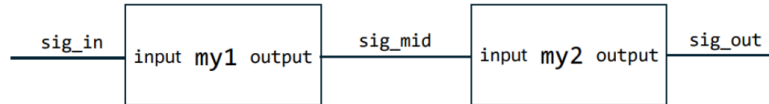
Stichworte: Ports (input, output)

```
1 SC_MODULE(MyModule) {  
2     sc_in<bool> input;  
3     sc_out<bool> output;  
4  
5     SC_CTOR(MyModule) {  
6         SC_THREAD(behaviour);  
7     }  
8  
9     void behaviour() {  
10         while (true) {  
11             output->write(!input->read());  
12             wait();  
13         }  
14     }  
15 };
```

- Lesen vom Input
- Schreiben auf den Output

Module in SystemC

Kommunikation zwischen Modulen (Kabel verbinden)



```
1 MyModule my1("my1");
2 MyModule my2("my2");
3 sc_signal<bool> sig_in, sig_mid, sig_out;
4 // Die Tatsächlichen Signale werden außerhalb der Module
  erstellt.
5
6 my1.input.bind(sig_in);
7 my1.output.bind(sig_mid);
8 // port.bind(signal) weist einem Port ein Signal zu.
9
10 my2.input(sig_mid);
11 my2.output(sig_out);
12 // Alternative Schreibweise: port(signal).
```

- Schreiben / Lesen von Signalen
- Verbinden dieser Signale mit den Ports

Aufgabe 1

Wahrheitstabellen

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

Aufgabe 2

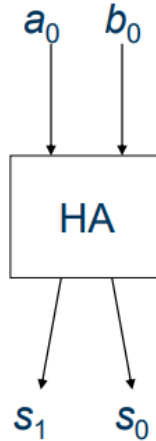
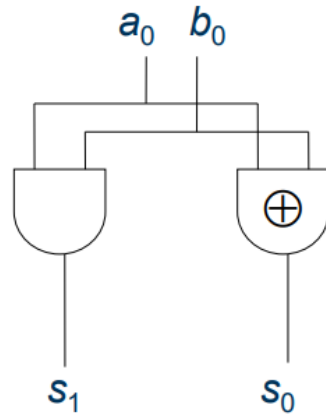
XOR-Gatter

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

- 1, wenn genau ein Input 1 ist
- Bzw. 1 wenn beide Inputs unterschiedlich sind

Aufgabe 3

Halbaddierer

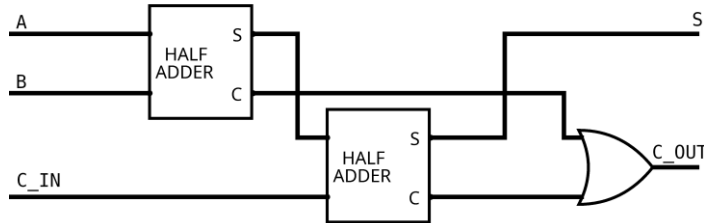


Teilaufgabe:

Implementiere den
Halbaddierer in SystemC

Aufgabe 3

Volladdierer



Teilaufgabe:

Implementiere den Volladdierer in SystemC