

- Das zugrundeliegende Schaltnetz: 2-Bit-Komparator

Als Beispiel für einen Komparator wird der 2-Bit-Komparator im Kurstext 1608 wie folgt eingeführt:

Komparatoren sind Rechenelemente, die analoge oder binäre Signale vergleichen (DIN 40700 Blatt 18/34). In digitalen Rechenanlagen sind Komparatoren Schaltnetze, die zwei Binärzahlen miteinander vergleichen. Werden zwei Binärzahlen mit A und B bezeichnet, dann sind die Vergleichskriterien $A = B$, $A > B$ und $A < B$.

b_1	B		A		Y_1	Y_2	Y_3
	b_0	a_1	a_0		$A = B$	$A < B$	$A > B$
0	0	0	0	1	0	0	0
0	0	0	1	0	0	1	
0	0	1	0	0	0	1	
0	0	1	1	0	0	1	
0	1	0	0	0	1	0	
0	1	0	1	1	0	0	
0	1	1	0	0	0	1	
0	1	1	1	0	0	1	
1	0	0	0	0	1	0	
1	0	0	1	0	1	0	
1	0	1	0	1	0	0	
1	0	1	1	0	0	1	
1	1	0	0	0	1	0	
1	1	0	1	0	1	0	
1	1	1	0	0	1	0	
1	1	1	1	1	0	0	

Für zwei zweistellige Binärzahlen $A = a_1a_0$ und $B = b_1b_0$, wobei a_0, b_0 den Stellenwert 2^0 haben, und a_1, b_1 den Stellenwert 2^1 , ergibt sich die obige Wertetabelle.

Mithilfe der Wertetabelle ergeben sich nach Vereinfachung mit dem KV-Diagramm die Schaltfunktionen in der DF:

$$Y_2 = \bar{a}_1b_1 \vee \bar{a}_1\bar{a}_0b_0 \vee \bar{a}_0b_1b_0 \quad \text{für } A < B$$

$$Y_3 = a_1\bar{b}_1 \vee a_0\bar{b}_1\bar{b}_0 \vee a_1a_0\bar{b}_0 \quad \text{für } A > B$$

$$Y_1 = \bar{Y}_2 \cdot \bar{Y}_3 = \overline{(Y_2 \vee Y_3)} \quad \text{für } A = B$$

Für den Vergleich von mehrstelligen Binärzahlen wird ein Algorithmus angewandt, der schrittweise alle Bit-Stellen miteinander vergleicht. Der Vergleich kann mit der MSB-Stelle (wertehöchste) oder der LSB-Stelle (wertniedrigste) beginnen. Die Schaltnetze, die dann entstehen, sind mehrstufig.

- **Die Hades Simulation:**

Links zeigt der Screenshot die 4 Eingangssignale für die zu vergleichenden Eingangsbits

$$A = a_1 a_0 \text{ und } B = b_1 b_0, \text{ rechts die Ausgangssignale für } Y_1, Y_2 \text{ und } Y_3.$$

Initiale Vorbelegung ist $A = 3$ und $B = 0$, und wie zu erwarten ist damit $Y_3 = 1$ und

$Y_1 = 0 = Y_2$, denn $A > B$. Um nun A und B mit unterschiedlichen Werten zu belegen, klickt man auf die entsprechenden Schalter für die Eingangssignale, wobei wieder grau für 0 und rot für 1 steht. Entsprechend werden die Ausgangssignale 0 (grau) oder 1 (rot) anzeigen und das Schaltverhalten kann verfolgt werden.

Die Ausgangssignale realisieren die Schaltfunktionen für Y_1, Y_2 und Y_3 exakt. Beispiel:

Das OR, dessen Ausgang Y_2 signalisiert, erhält als Eingänge:

$$\text{Eingang } A : \bar{a}_1 \wedge b_1$$

$$\text{Eingang } B : \bar{a}_1 \wedge \bar{a}_0 \wedge b_0$$

$$\text{Eingang } C : \bar{a}_0 \wedge b_1 \wedge b_0$$

was also genau oben genannte Schaltfunktion umsetzt.

- **Die Simulation besteht aus folgenden Komponenten:**

- 4 Ipins (switch)
- 4 INV (small)
- 2 XNOR2
- 3 AND2
- 4 AND3
- 2 OR3
- 3 Opins (LED)

