

Grundlagen der Informatik

- Schaltnetze und Schaltwerke -

Prof. Dr. Klaus Volbert



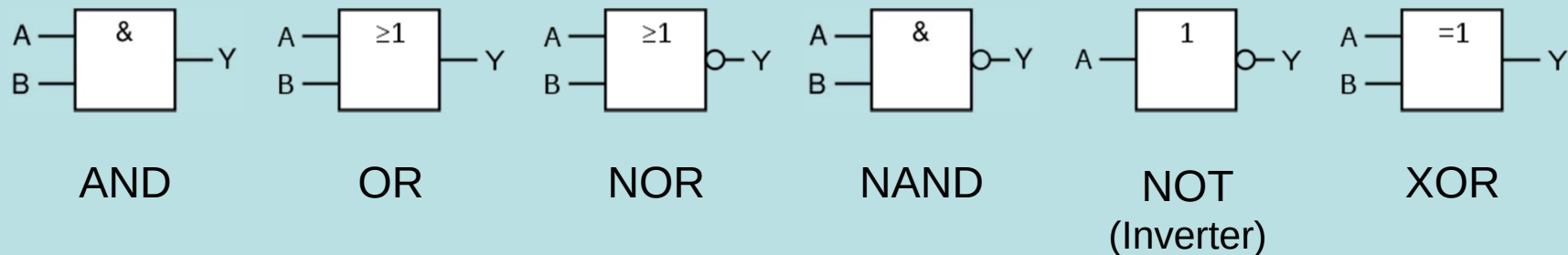
Hochschule für angewandte Wissenschaften
Fakultät Informatik und Mathematik

Wintersemester 2010/11
Regensburg, 01./02. Dezember 2010

- Abhängig von der Anzahl der Gatter auf einem Baustein:

Name	Anzahl Gatter pro Baustein	Jahr
Small Scale Integration (SSI)	≤ 10	1960
Medium Scale Integration (MSI)	10 bis 10^2	1965
Large Scale Integration (LSI)	10^3 bis 10^5	1970
Very Large Scale Integration (VLSI)	mehr als 10^5	1980
CPU-Chips / Speicherchips	mehr als 100 Mio. / 1 Mrd.	heute

- Wesentliche Gatter/Basisgatter (DIN)



- Beispiel:
 - XOR aus NAND-Gattern

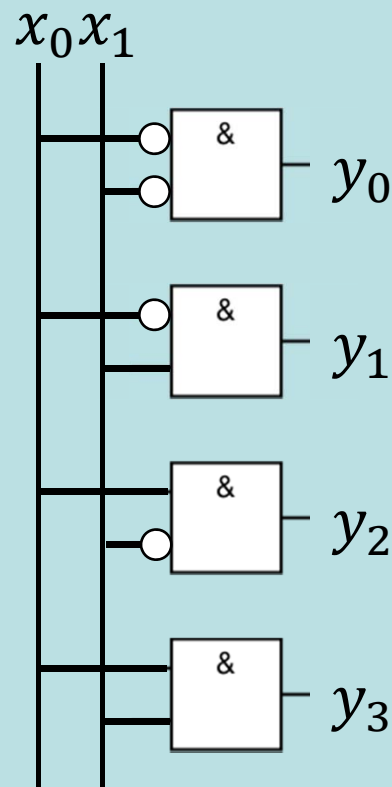
Minimierung boolescher Ausdrücke

- Je weniger Bauelemente, desto billiger das Produkt, und je kleiner die Anzahl der notwendigen Gatter, um so geringer sind Laufzeitprobleme bei zeitkritischen Schaltungen
- Minimierungsmethoden
 - Anwendung der Gesetze der booleschen Algebra („manuell“)
 - Karnaugh-Veitch-Diagramme („manuell“, nur bei wenigen Variablen)
 - Idee: KNF/DNF durch sinnvolles Zusammenfassen von Nullen oder Einsen und „günstiges“ Interpretieren der redundanten Felder möglichst kompakt angeben (schwierig zu programmieren)
 - Quine-McCluskey-Verfahren (einfacher zu programmieren)
 - Idee: Wiederholte Anwendung der folgenden Beziehung

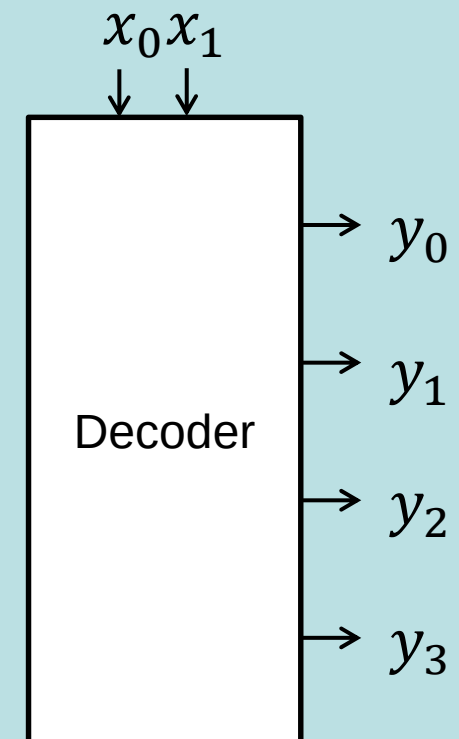
$$x_0 x_1 + x_0 \overline{x_1} = x_0 (x_1 + \overline{x_1}) = x_0$$

Dekodierer (engl. decoder)

- Ein Dekodierer hat n Eingänge und 2^n Ausgänge (für jede Eingabekombination genau einen Ausgang mit Wert 1)
- Beispiel: 2-zu-4 Dekodierer



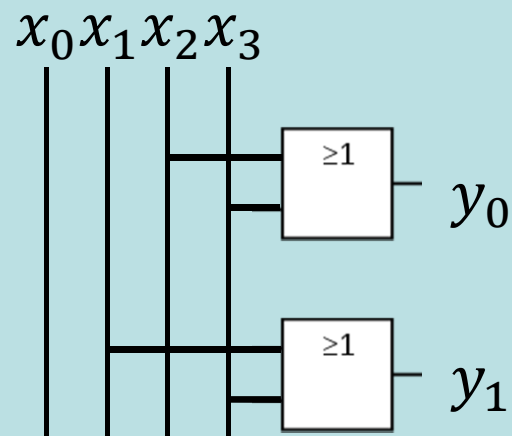
Eingänge		Ausgänge			
x_0	x_1	y_0	y_1	y_2	y_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1



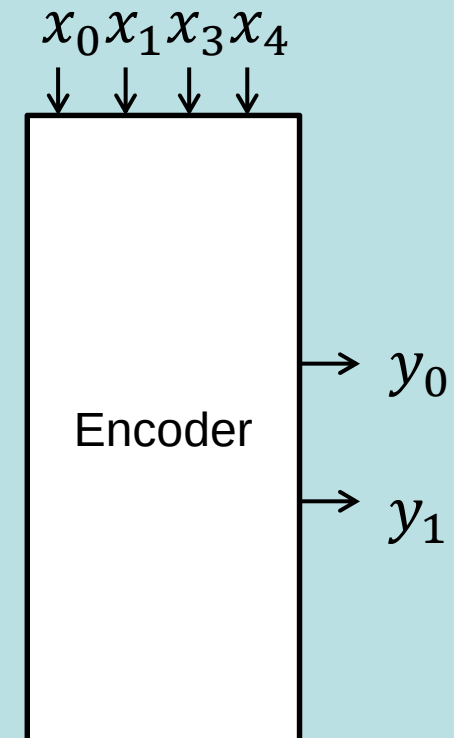
- Anwendung z.B. in ROM-Speichern (Befehls-/Adreßdekodierer)

Kodierer (engl. encoder)

- Ein Kodierer hat 2^n Eingänge und n Ausgänge (für jede Eingabekombination mit genau einem Wert 1 wird ein Ausgang produziert, Gegenstück des Dekodierers)
- Beispiel: 4-zu-2 Kodierer

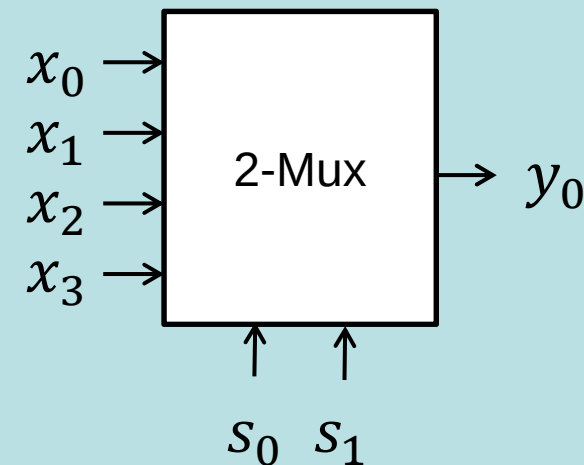
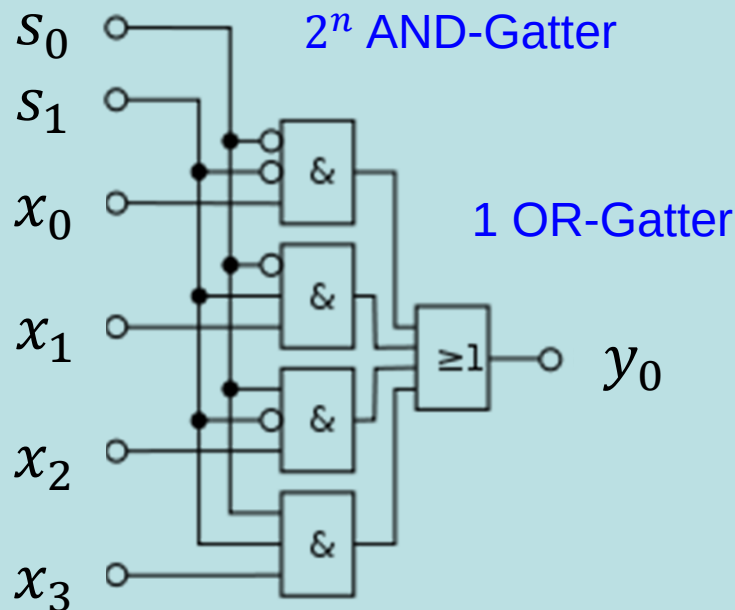


Eingänge				Ausgänge	
x_0	x_1	x_2	x_3	y_0	y_1
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1



Multiplexer (Selektor)

- Ein n -Multiplexer hat 2^n Eingänge, n Steuerleitungen und 1 Ausgang
- Beispiel: 2-Multiplexer (Bottom-Up Entwurf)



- Anmerkung:
 - 2-Multiplexer kann auch durch die Kombination von 1-Multiplexern realisiert werden (Top-Down Entwurf)
- Durch die Steuerleitungen wird ein Eingang ausgewählt und auf den Ausgang geleitet (z.B. Auswahl von Operanden im Rechenwerk)

Demultiplexer

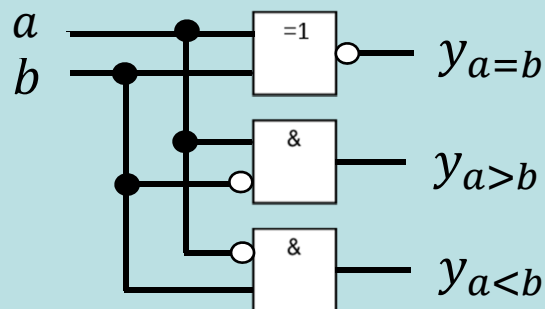
- Ein n -Demultiplexer hat 1 Eingang, n Steuerleitungen und 2^n Ausgänge
- Beispiel: 2-Demultiplexer, Eingang x

s_0	s_1	Ausgang	Boolescher Ausdruck (DNF)
0	0	y_0	$x \overline{s_0} \overline{s_1}$
0	1	y_1	$x \overline{s_0} s_1$
1	0	y_2	$x s_0 \overline{s_1}$
1	1	y_3	$x s_0 s_1$

- Durch die Steuerleitungen wird der Eingang auf den ausgewählten Ausgang geleitet (z.B. Auswahl von Operationen)

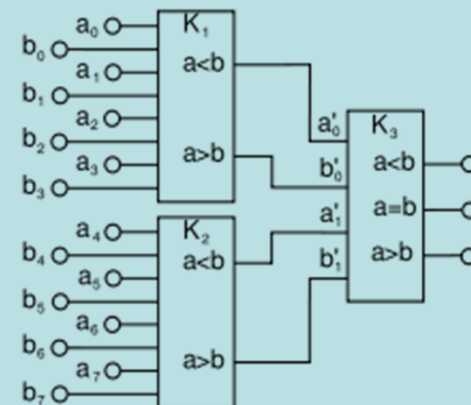
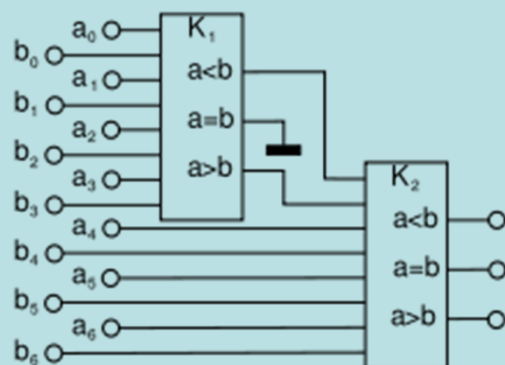
Komparator

- Ein Komparator vergleicht zwei Eingaben Bit-weise miteinander und gibt an den Ausgängen das Ergebnis aus
- Beispiel: 1-Bit-Komparator



a	b	$y_{a<b}$	$y_{a=b}$	$y_{a>b}$
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0

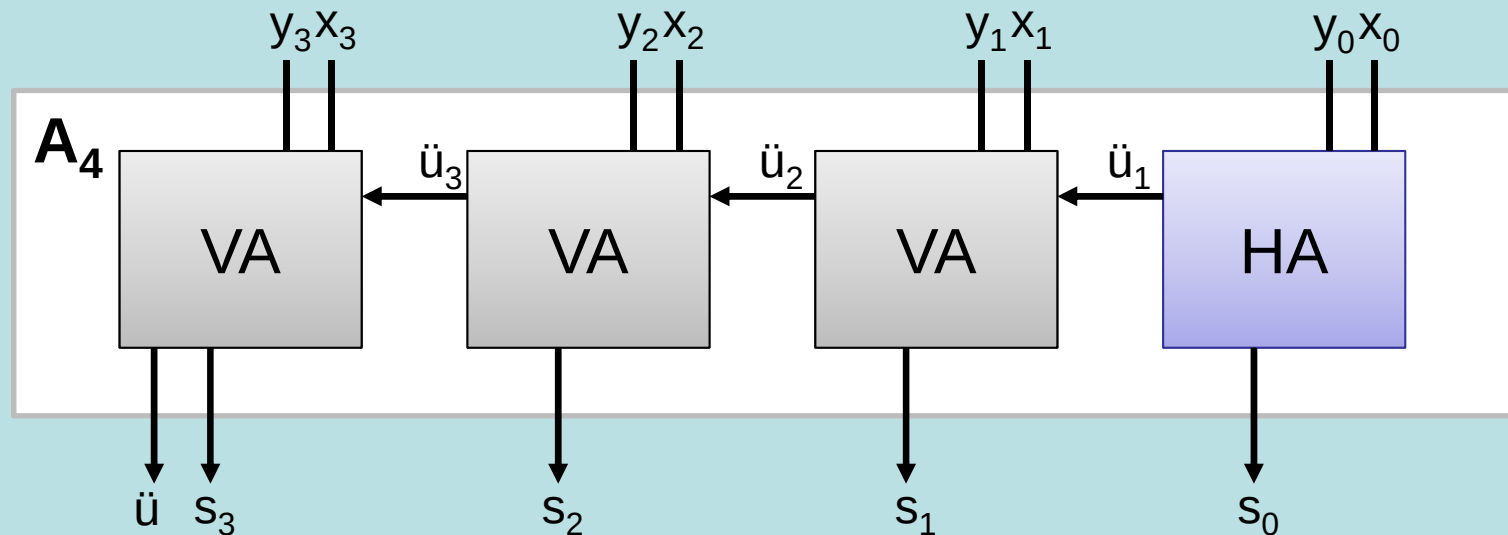
- Serieller 7-Bit und paralleler 8-Bit Komparator



- Umsetzung des Befehls cmp

Addierer

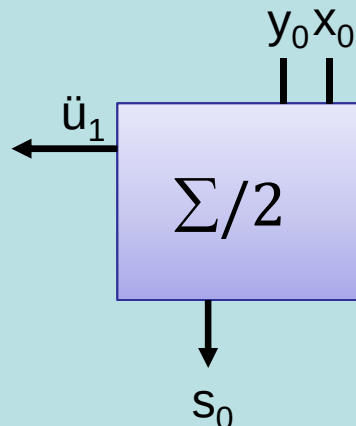
- Schaltnetz-Definitionen
 - Ein **Paralleladdierer** addiert zwei n -stellige Dualzahlen
 - Ein **Halbaddierer** liefert zu je zwei Dualziffern Summe und Übertrag
 - Ein **Volladdierer** liefert zu drei Dualziffern Summe und Übertrag
- Beispiel (4-Bit-Paralleladdierer):



- Halbaddierer:

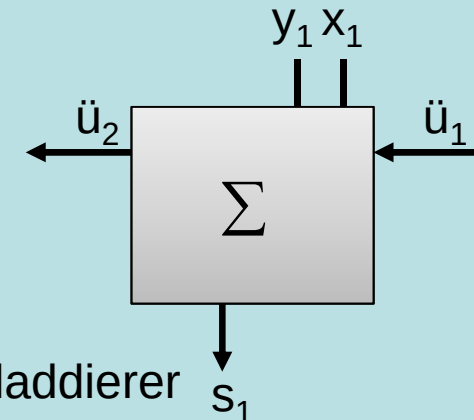
x_0	y_0	s	u
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

- $s = (x_0 + y_0)(\overline{x_0} + \overline{y_0}) = x_0 \text{ XOR } y_0$
- $u = (x_0 + y_0)(x_0 + \overline{y_0})(\overline{x_0} + y_0) = x_0 y_0$



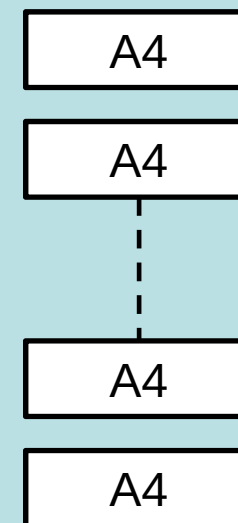
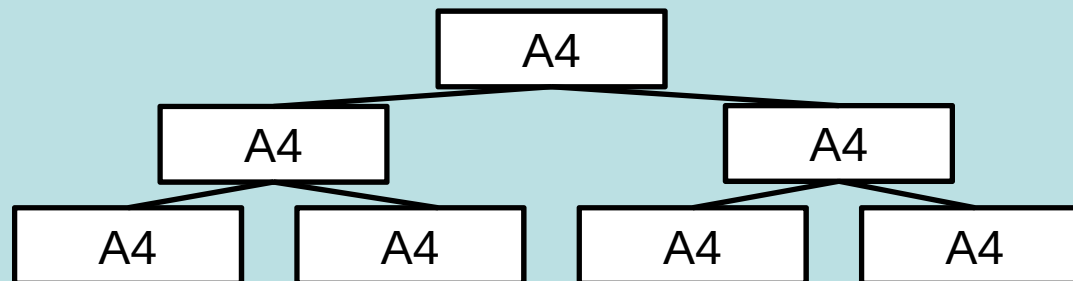
- Volladdierer:

x_0	y_0	u	s	u'
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



Übung: Schaltnetz für Volladdierer

- Die Multiplikation ganzer Zahlen lässt sich mit Hilfe wiederholter Addition durchführen (schriftl. Multiplizieren)
- Realisierung durch
 - Schiebeoperationen (richtige Stellen untereinander setzen)
 - Hintereinanderschaltung von Addierern
- Schaltungstiefe bei Multiplikation von zwei n -Bit Zahlen
 - Seriell: $(n - 1)$ Additionen erforderlich
 - Parallel: $(\log_2 n)$ Additionen erforderlich



- Beispiel: 16-Bit Zahlen
 - 15 Additionen ggü. 4 Additionen