

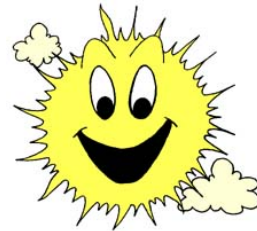
Bitte beachten:

Das Skript wurde korrigiert und aktualisiert. Einige Themen aus der DT sind nach RG gewandert, stehen aber noch im ursprünglichen Skript.

Beachten Sie die Hinweise aus den Vorlesungen!

<http://ww.fbi.fh-darmstadt.de/~jwietzke/DT1.pdf>

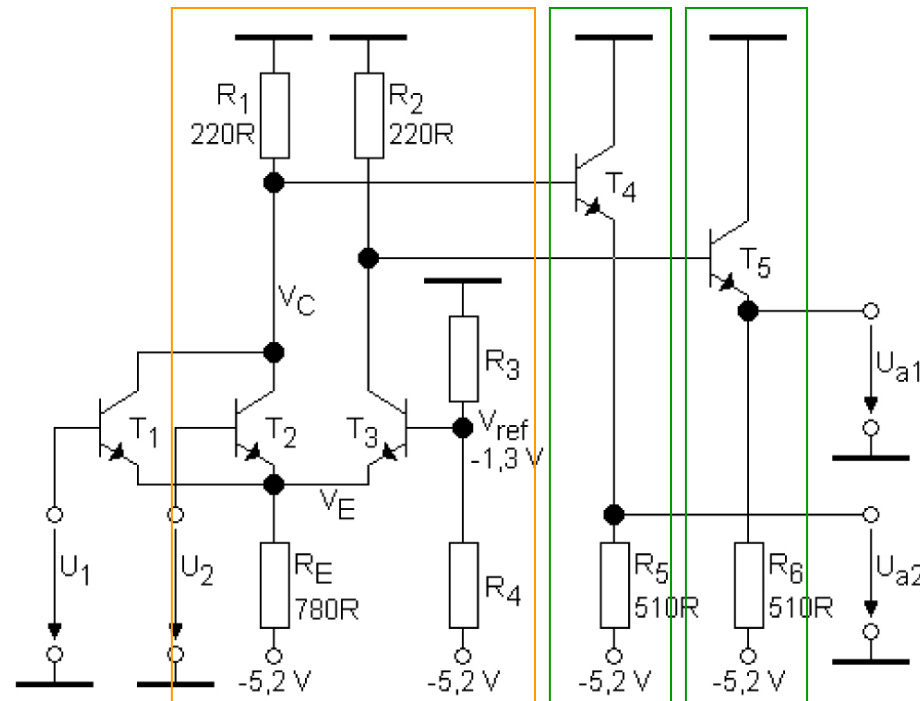
Ende der Wiederholung



7.5 BICMOS

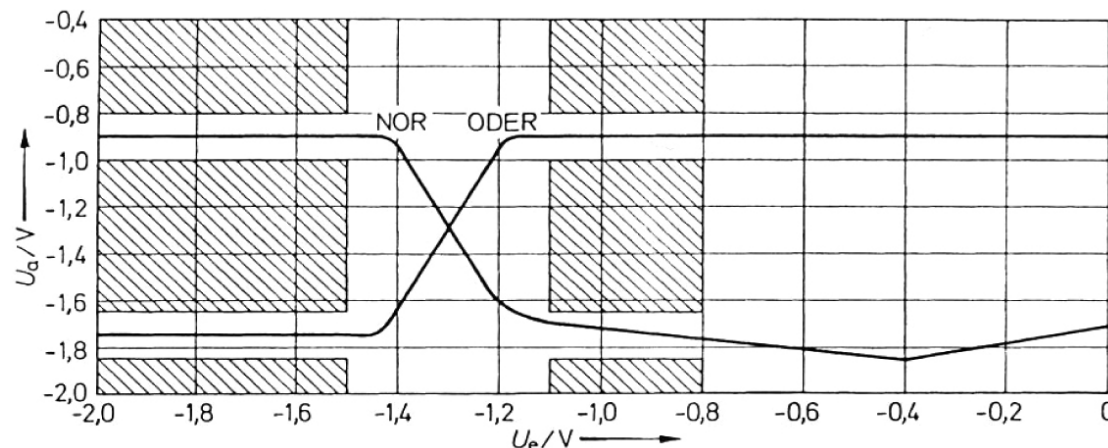
BICMOS-Bausteine verwenden eine **Kombination** aus **TTL-Ausgangsschaltung** und **CMOS-Eingangstechnik**. Dies vereinigt hohe **Arbeitsfrequenz** mit **robusten Ausgängen** und ermöglicht weiterhin leichten **Übergang zwischen Technologien und unterschiedlichen Betriebsspannungen**. Die Schaltzeiten von BICMOS-Gattern liegen bei wenigen Nanosekunden.

7.6 ECL (Funktionsweise)



Wesentliche Bestandteile der Schaltung sind als Eingangsteil ein **Differenzverstärker** und als (komplementäre) Ausgangsstufen zwei **Emitterfolger**

Bei einem Emitterfolger ist die Ausgangsspannung U_E stets um -0.7 V gegenüber der Spannung U_B an der Basis versetzt.



Pegel bei ECL in positiver Logik:

■ H: -0.9 V

■ L: -1.7 V

➔ Transistoren geraten nie in Sättigung

➔ schnell!

➔ kurze Gatterlaufzeiten!

Will man den **Sub-Nanosekunden Bereich** erreichen, wird man **ECL**-Technik verwenden (**Emitter coupled Logic**). Durch **Vermeidung von Sättigungszuständen der Transistoren** lassen sich Schaltzeiten von einigen 100 Picosekunden erreichen. Allerdings sind **Integrationsdichte niedrig und Leistungsverbrauch hoch**. Man wird daher diese Technologie nur im ‚Front-end‘ (z.B. A/D-Wandler) einsetzen und nach Reduzierung der Datenrate auf andere Technologien übergehen.

7. Übersicht Technologien

Schaltkreis-familie	$\frac{U_B}{V}$	$\frac{\overline{P_v}}{mW}$	$\frac{\overline{T_p}}{ns}$	$\frac{\overline{U_s}}{V}$	fan out	$\frac{\overline{f_{max}}}{MHz}$
DTL	5	15	25	0,7	8	10
DTLZ (LSL)	12 15	16 27	175 167	5 6,5	10 10	0,5-2 0,5-2
TTL-Standard	5	10	10	0,4	10	15
TTL-Low-Power	5	1	33	0,4	20	2,5
TTL-High-Power	5	22	6	0,4	10	35
TTL-Shottky	5	19	3	0,4	10	75
TTL-Low-Power-Shottky	5	2	9,5	0,4	20	25
TTL-Fast	5	5	2,25	0,4	20	100
ECL	-5,2	60	1	0,2	15	250
I ² L	1-15	0,01-30	10-1000	0,3	5	5
PMOS	-12	6	100	3	20	2
NMOS	10	2	15	2	20	15
CMOS	5 10 15	} 10 ⁻⁵	100	1,5	} 50	2
	50		3	5		
	40		4,5	7		
LOC MOS	5 10 15	} 10 ⁻⁵	60	1,5	} 50	8
	30		3	16		
	20		4,5	24		
HCMOS	5	10 ⁻⁵	10	1,5	50	50

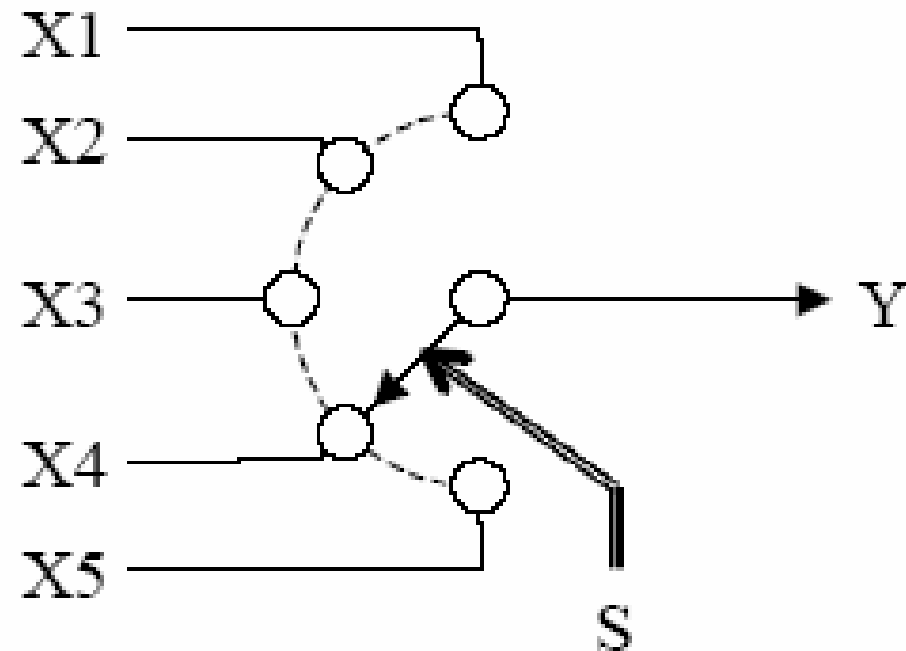
8 Kombinatorische Grundschaltungen

1. Multiplexer
2. Realisierung boolescher Funktionen mit Multiplexern
3. Demultiplexer
4. Realisierung boolescher Funktionen mit Demultiplexer
5. Code-Umsetzer
6. Rechenwerke → Rechnergrundlagen
7. Logische Operationswerke

8.1 Multiplexer

Multiplexer dienen dazu, aus vielen **Signalen** für die Weiterverarbeitung eines **auszuwählen**.

Anschaulich kann man sich die Funktion eines Multiplexers an einem Drehschalter verdeutlichen, der genau eine Eingangsleitung auf einen Ausgang schaltet:



8.1 Multiplexer

In der Digitaltechnik möchte man die **Auswahl**, welches Eingangssignal auf den Ausgang Y geschaltet wird, mit digitalen Signalen S_i vornehmen.

Somit erhält man eine **boolesche Funktion**, der als Parameter die **Eingangs-** und die **Auswahlsignale** (auch als **Selektionssignale** bezeichnet) übergeben werden.

Typischerweise wird das Auswahlsignal S als **binärer Signalvektor** realisiert und der selektierte Eingang binär kodiert.

8.1 2:1 Multiplexer

S	X2	X1	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

S	Y
0	X1
1	X2

Für einen 2:1 Multiplexer ergibt sich folgende Funktionstabelle (Tab. li).

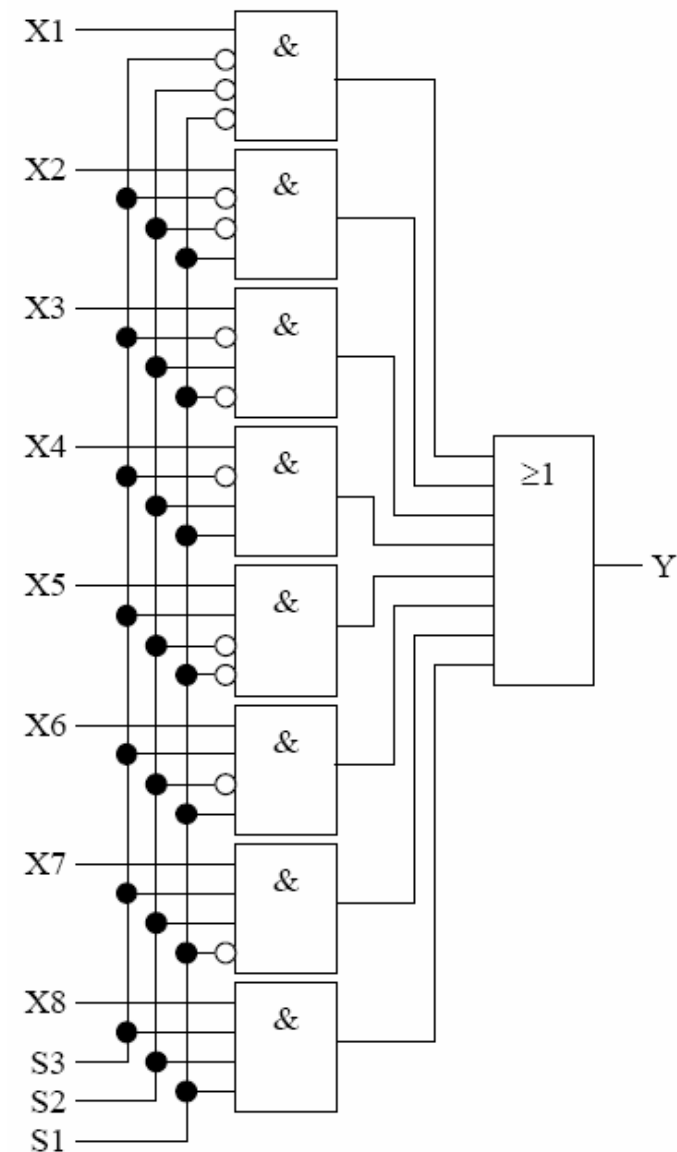
Im Fall $S=0$ wird der Wert $X1$ und im Fall $S=1$ der Wert $X2$ auf den Ausgang gegeben. Damit kann man die Funktionstabelle auch verkürzt schreiben (Tab. re).

8.1 8:1 Multiplexer

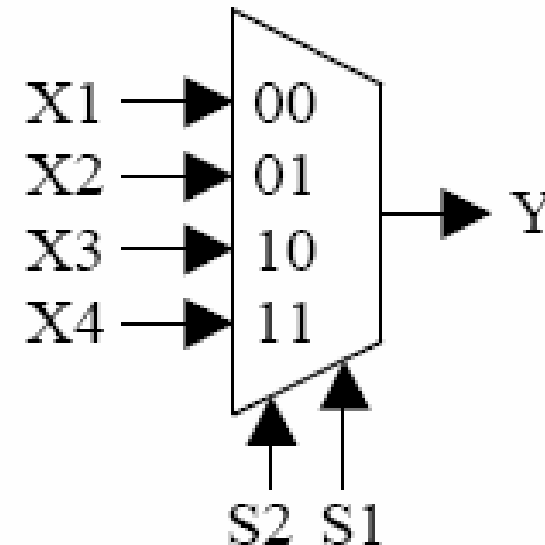
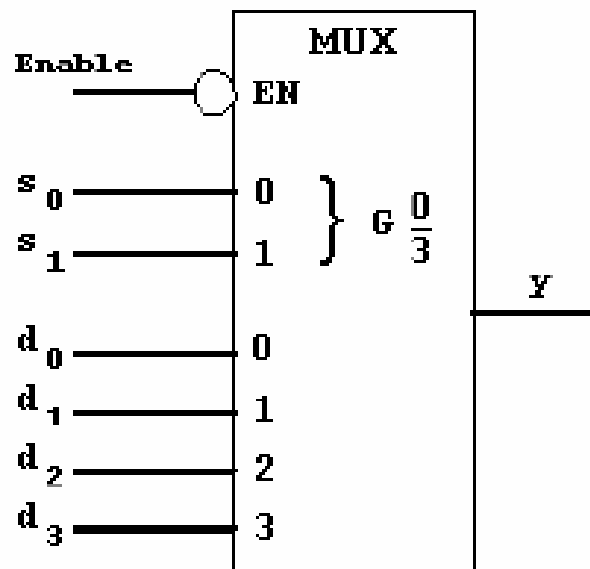
S3	S2	S1	Y
0	0	0	X1
0	0	1	X2
0	1	0	X3
0	1	1	X4
1	0	0	X5
1	0	1	X6
1	1	0	X7
1	1	1	X8

Für einen 8:1 Multiplexer ergibt sich obige Funktionstabelle in verkürzter Schreibweise

Digitale Schaltung, die 8:1 Multiplexer realisiert



8.1 Multiplexer DIN / ISO -Symbol



DIN 40900-Symbol für einen 4:1 Multiplexer

s0 und s1: Steuereingänge

d0-d3: Dateneingänge

G: Steuereingänge steuern Daten-Eingänge durch UND-Verknüpfung

ISO-Symbol für einen 4:1 Multiplexer

S1 und S2: Steuereingänge

X1-X4: Dateneingänge

Zahlen geben Indexbereich an

8.2 Realisierung von Funktionen mit MUX

Multiplexer sind gut für die Realisierung boolescher Funktionen geeignet:

Bei N Eingangsvariablen am einfachsten mit $2^N:1$ Multiplexern:

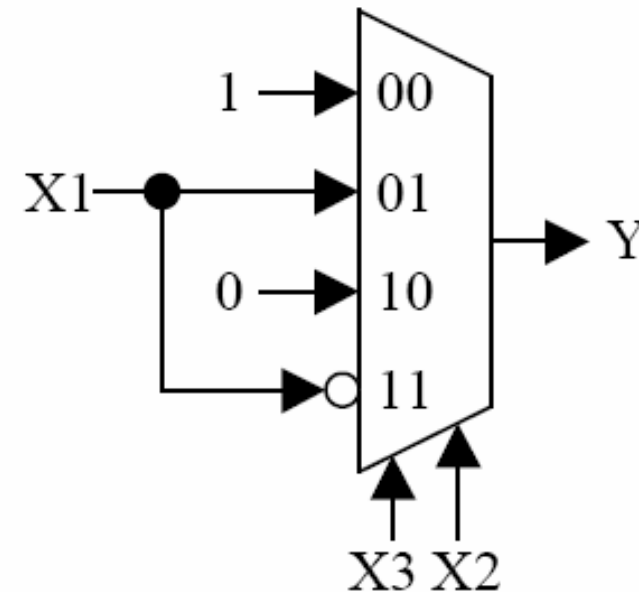
- Die Eingangsvariablen schaltet man an die Selektionseingänge
- An die Dateneingänge schaltet man, je nach Erfordernis der zu realisierenden Funktion eine 0 oder 1.

8.2 Beispiel 2: Funktion mit MUX

Gegeben sei eine Funktion mit $N=3 \rightarrow 4:1$ MUX

X3	X2	X1	Y	
0	0	0	1	Y=1
0	0	1	1	
0	1	0	0	Y=X1
0	1	1	1	
1	0	0	0	Y=0
1	0	1	0	
1	1	0	1	
1	1	1	0	Y= \neg X1

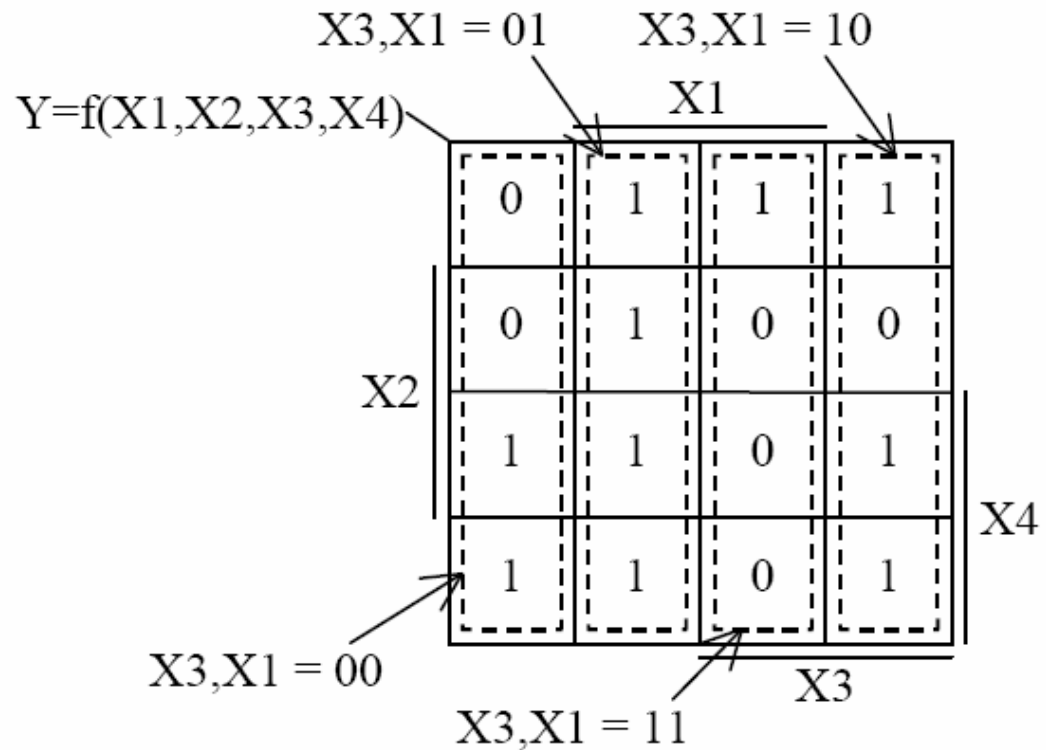
Funktion erlaubt
Verwendung eines
4:1 Multiplexer falls
X1 auch invertiert
zur Verfügung steht



Realisierung mit 4:1
Multiplexer mit X2
und X3 als Steuer-
eingang

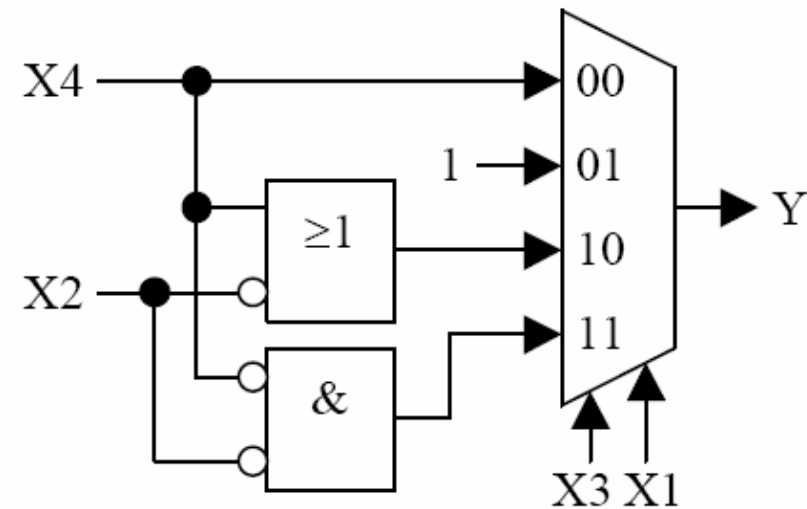
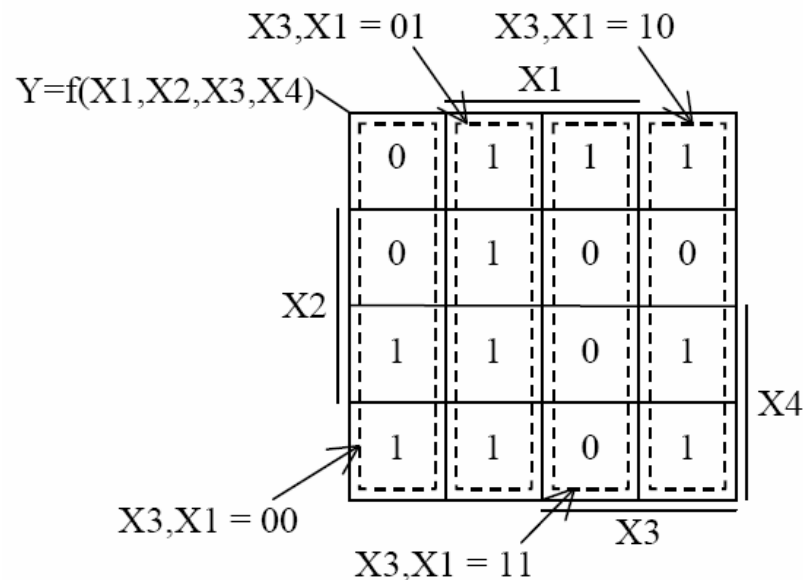
8.2 Beispiel 3: Funktion N=4 mit 4:1 MUX

X4	X3	X2	X1	Y
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0



mit X3 und X1 erhält man
Steuerfunktion über die
Variablen X2 und X4

8.2 Beispiel 3: Funktion N=4 mit 4:1 MUX (2)

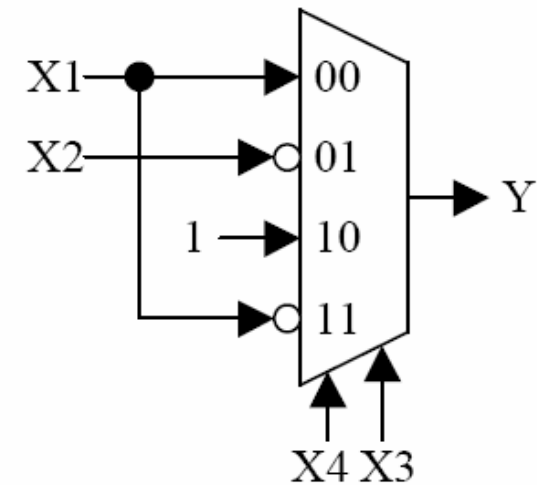
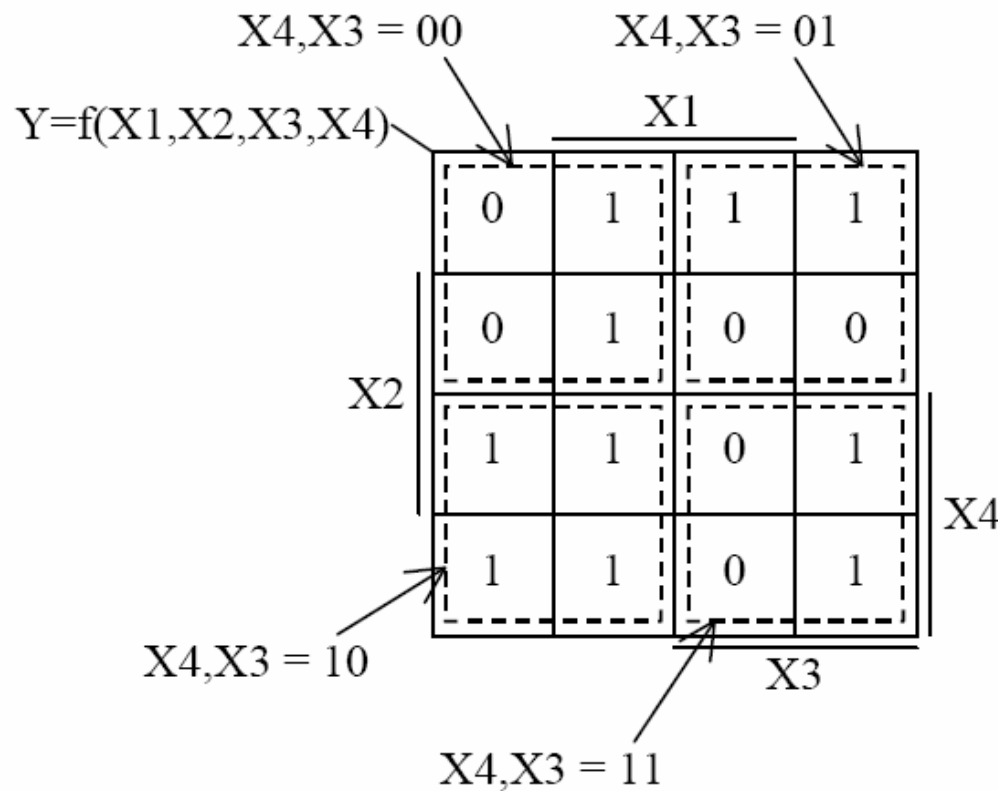


Steuerkombination $X3, X1$	Ausgangsfunktion
00	$Y = X4$
01	$Y = 1$
10	$Y = X4 \vee (\neg X2)$
11	$Y = (\neg X4) \wedge (\neg X2)$

Die gefundene Lösung ist noch nicht optimal.

➔ Suche andere Kombination der Ansteuersignale ...

8.2 Beispiel 3: Funktion $N=4$ mit 4:1 MUX (3)



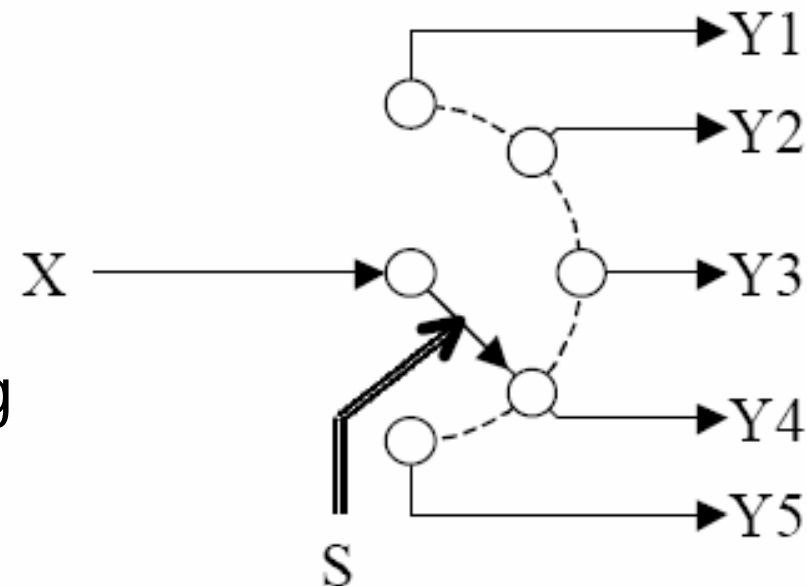
Aus dem KV-Diagramm und andere Wahl der Unterdiagramme ergibt sich einfache Lösung

8.3 Demultiplexer

Demultiplexer realisieren die umgekehrte Funktion des Multiplexers.

Ein einzelner Eingang X wird auf einen von vielen Ausgängen geschaltet. Beispiel Drehschalter.

Bei der digitalen Realisierung eines Demultiplexers wird der Ausgang Y_i , der mit dem der Eingang X verbunden werden soll, durch Steuereingänge S_i ausgewählt.



8.3 1:4 Demultiplexer

S2	S1	X	Y1	Y2	Y3	Y4
0	0	0	0	*	*	*
0	0	1	1	*	*	*
0	1	0	*	0	*	*
0	1	1	*	1	*	*
1	0	0	*	*	0	*
1	0	1	*	*	1	*
1	1	0	*	*	*	0
1	1	1	*	*	*	1

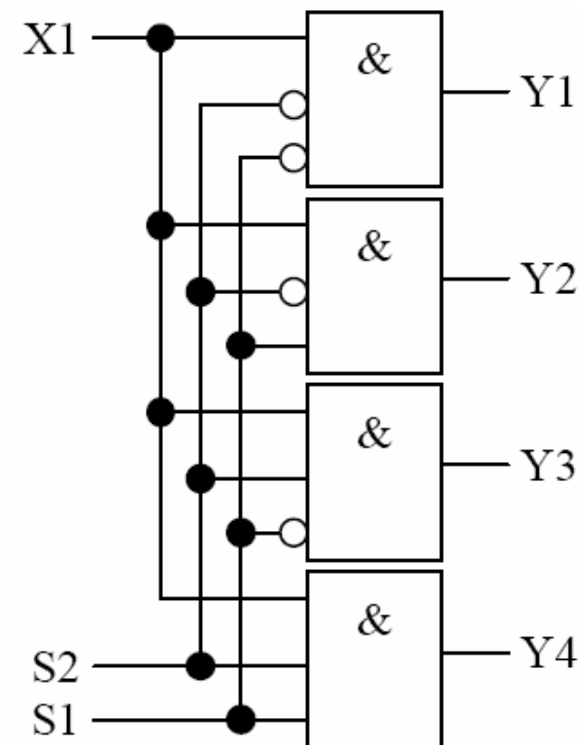
$$Y1 = X \wedge (\neg S2) \wedge (\neg S1)$$

$$Y2 = X \wedge (\neg S2) \wedge S1$$

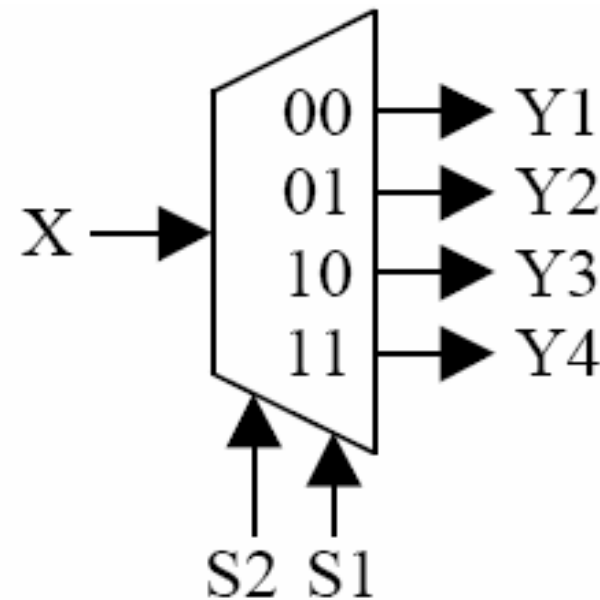
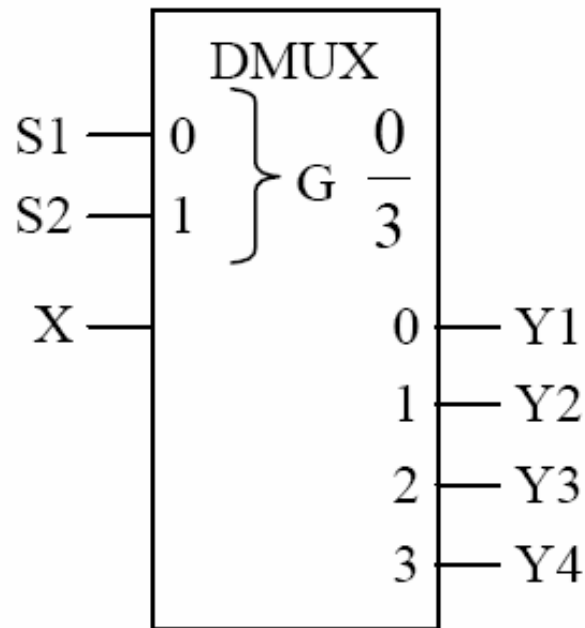
$$Y3 = X \wedge S2 \wedge (\neg S1)$$

$$Y4 = X \wedge S2 \wedge S1$$

S2	S1	Y1	Y2	Y3	Y4
0	0	X	0	0	0
0	1	0	X	0	0
1	0	0	0	X	0
1	1	0	0	0	X



8.3 Demultiplexer DIN / ISO -Symbol



DIN 40900-Symbol für einen 1:4 Demultiplexer

s0 und s1: Steuereingänge

X: Dateneingang

Y1-Y4: Datenausgang

G: Steuereingänge steuern Datenausgänge durch UND-Verknüpfung

ISO-Symbol für einen 1:4 Demultiplexer

S1 und S2: Steuereingänge

Y1-Y4: Datenausgänge

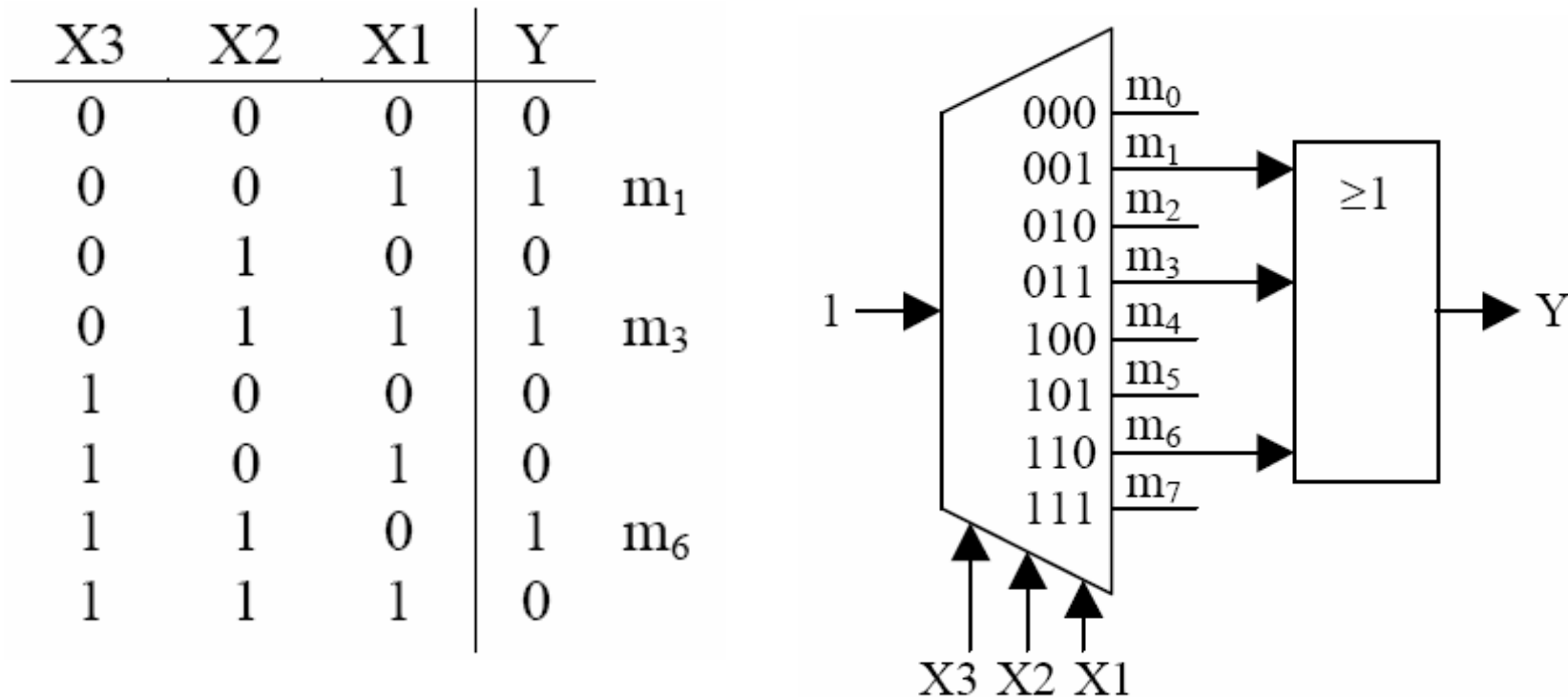
Zahlen geben Indexbereich an

8.4 Realisierung von Funktionen mit DMUX

Legt man bei einem Demultiplexer den Eingang X auf 1, liefert jeder Ausgang eine **Minterm-Funktion m_i der Steuereingänge**. Die gewünschte Funktion erhält man durch die disjunktive Verknüpfung der gewünschten Minterme:

Mit einem **$1:2^N$ Demultiplexer** und einem **N Bit breiten ODER-Gatter** lässt sich **jede Funktion** aus N Eingangsvariablen realisieren.

8.4 Beispiel: Funktionen N=3 und 1:8 DMUX

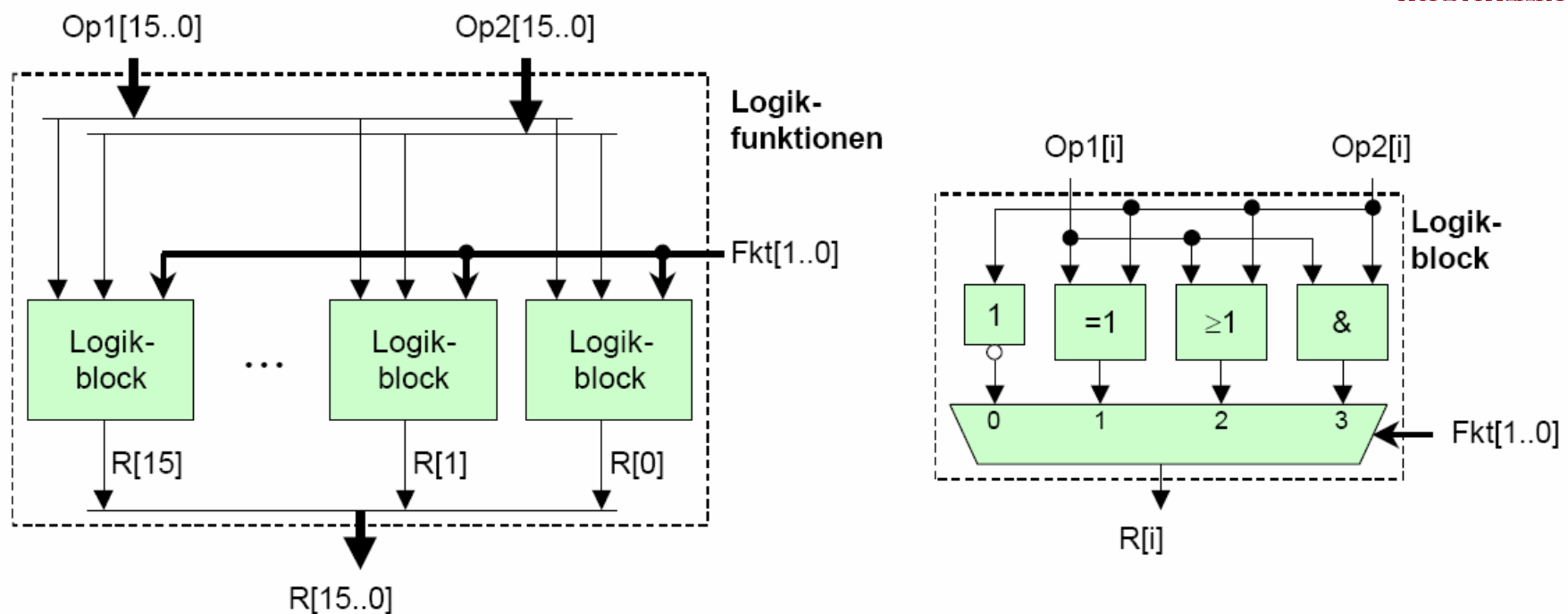


Demultiplexer liefert direkt alle Minterme m_i aus den N Eingangsvariablen.

Man verbindet die benötigten Minterme aus dem $1:2^N$ DMUX mit den Eingängen eines nachfolgenden ODER-Gatters (evtl. unbenutzte Eingänge auf 0 legen).

Damit ist die vorgegebene Funktion realisiert.

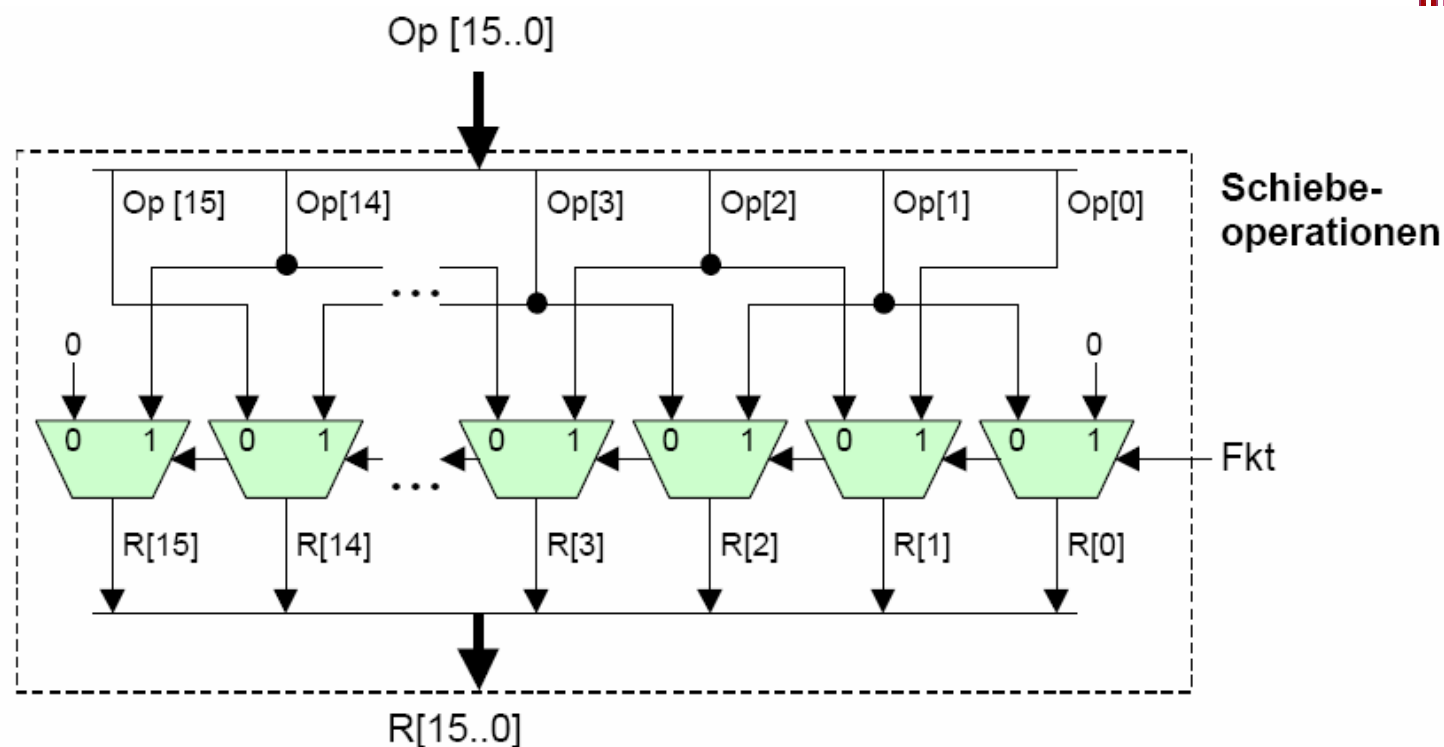
8.4.1 Bitweise Logische Verknüpfung



Zwei Operanden Op1 und Op2 mit beispielsweise 16 Bit sollen bitweise miteinander verknüpft werden.

Die Innenschaltung für die Verknüpfung zweier Einzelbits in einem Logik-Block ist im rechten Ausschnitt dargestellt.

8.4.2 Schiebe-Operation (s.a. RG)



Grundsätzlich muss zuerst entschieden werden, ob der Rechner Schiebeoperationen in einem oder mehrerer Schritten durchführen soll.

Oben: „**Barrel-Shifter**“, der um eine Stelle nach rechts oder links verschieben kann.