

Systemorientierte Informatik 1



- 2. Grundlagen Digitaler Schaltungen
 - 2.15 Fan-In und Fan-Out
 - 2.16 Standard-Schaltnetze

2.15 Fan-In und Fan-Out:

Fan-In: Die Anzahl der Eingänge in ein Gatter. Bestimmt die Anzahl der Transistoren in Reihe, die durchlaufen werden müssen, um den Ausgang zu treiben.

Fan-Out: Die Anzahl der Eingänge gleicher Größe, die vom Ausgang eines Gatters auf- oder entladen werden müssen. Bestimmt die Kapazität, die das Gatter beim Schaltvorgang umladen muss.

Der Widerstand und die Kapazität bilden einen Tiefpass, der das Schaltverhalten des Gatters beschreibt. Der Umschaltvorgang lässt sich durch die Exponentialkurve des Tiefpasses beschreiben. $R \cdot C$ sind die Zeit, die der Umladevorgang benötigt. Also ist diese Zeit sowohl proportional zu R wie auch zu C . Daher müssen Fan-In und Fan-Out in realen Schaltungen begrenzt werden.

Aufladevorgang (Wechsel von 0 auf 1)

$$U_a = U_e - U_e \cdot e^{-\frac{t}{RC}} = U_e \cdot (1 - e^{-\frac{t}{RC}})$$

$$\frac{U_a}{U_e} = 1 - e^{-\frac{t}{RC}}$$

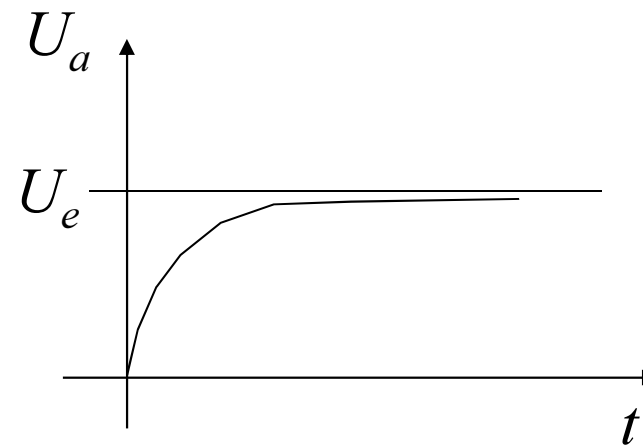
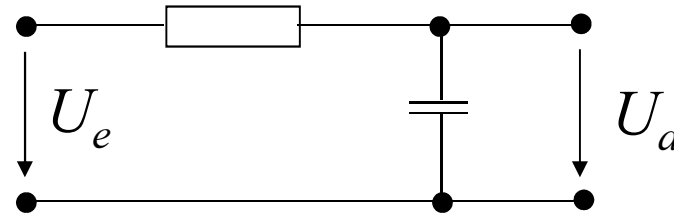
$$1 - \frac{U_a}{U_e} = e^{-\frac{t}{RC}}$$

$$\text{Annahme: } \frac{U_a}{U_e} = 0,9$$

$$\ln(1 - 0,9) = -\frac{t}{RC}$$

$$-2,3RC \approx -t$$

$$t \approx 2,3RC$$



Entladevorgang (Wechsel von 1 auf 0)

$$U_a = U_e \cdot e^{-\frac{t}{RC}}$$

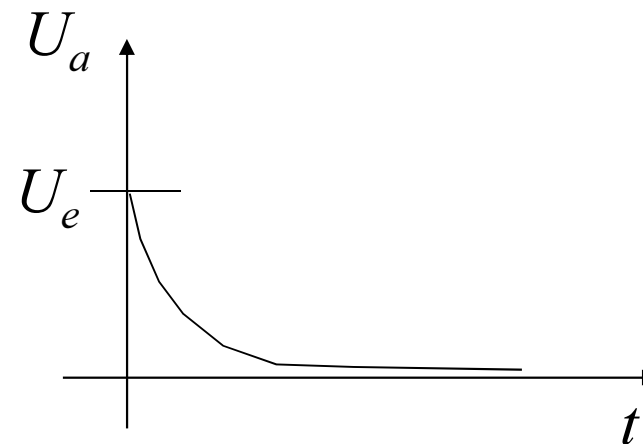
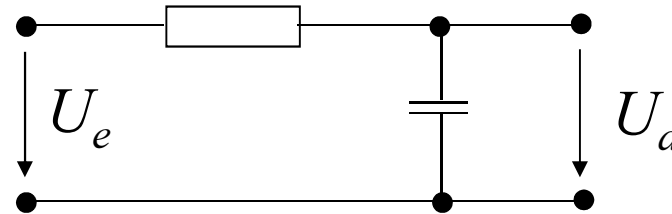
$$\frac{U_a}{U_e} = e^{-\frac{t}{RC}}$$

$$\text{Annahme: } \frac{U_a}{U_e} = 0,1$$

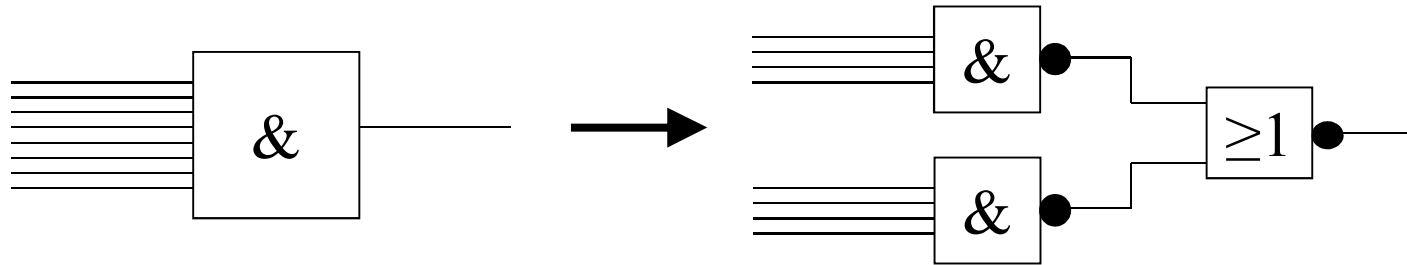
$$\ln 0,1 = -\frac{t}{RC}$$

$$-2,3RC \approx -t$$

$$t \approx 2,3RC$$

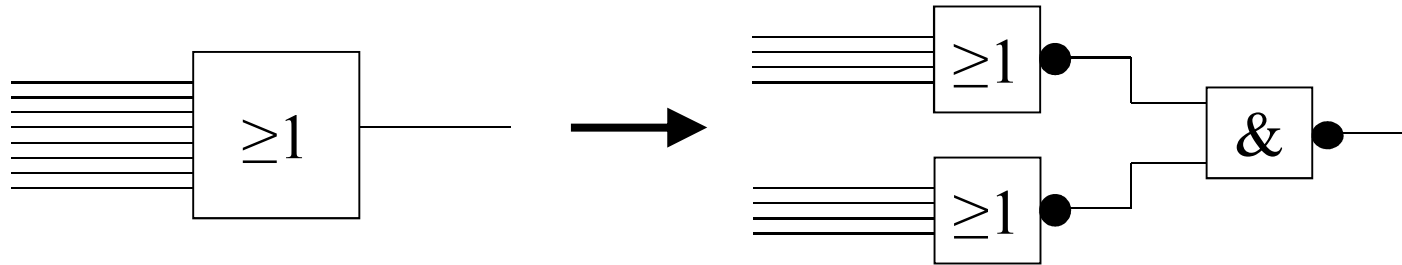


Was machen wir, wenn wir Gatter mit zu großem Fan-In haben?



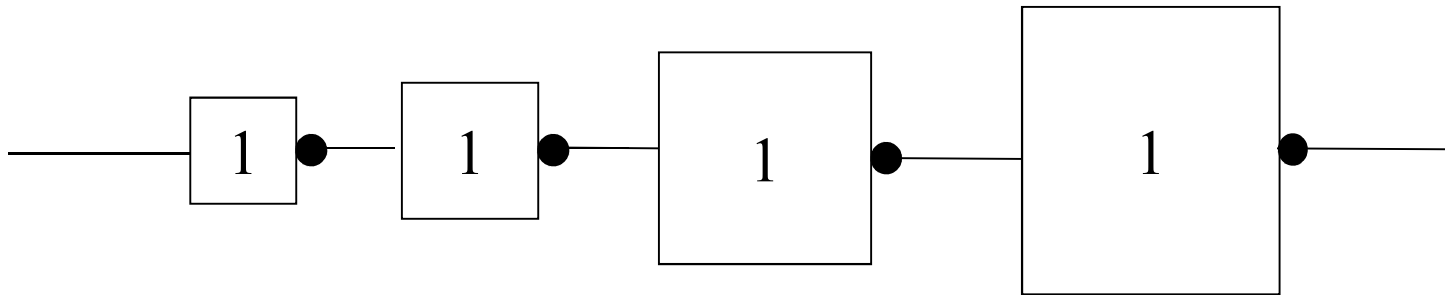
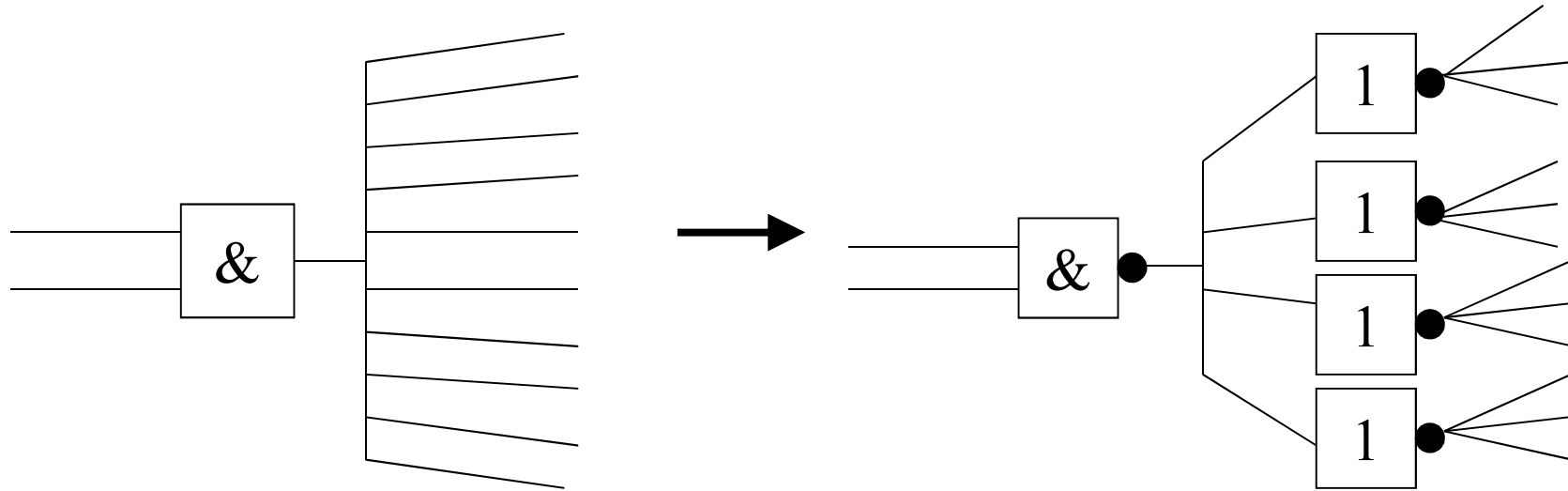
$$a \cdot b \cdot c \cdot d = \overline{\overline{a \cdot b \cdot c \cdot d}} = \overline{\overline{a \cdot b} + \overline{c \cdot d}}$$

2. Beispiel: Oder Gatter mit zu großem Fan-In

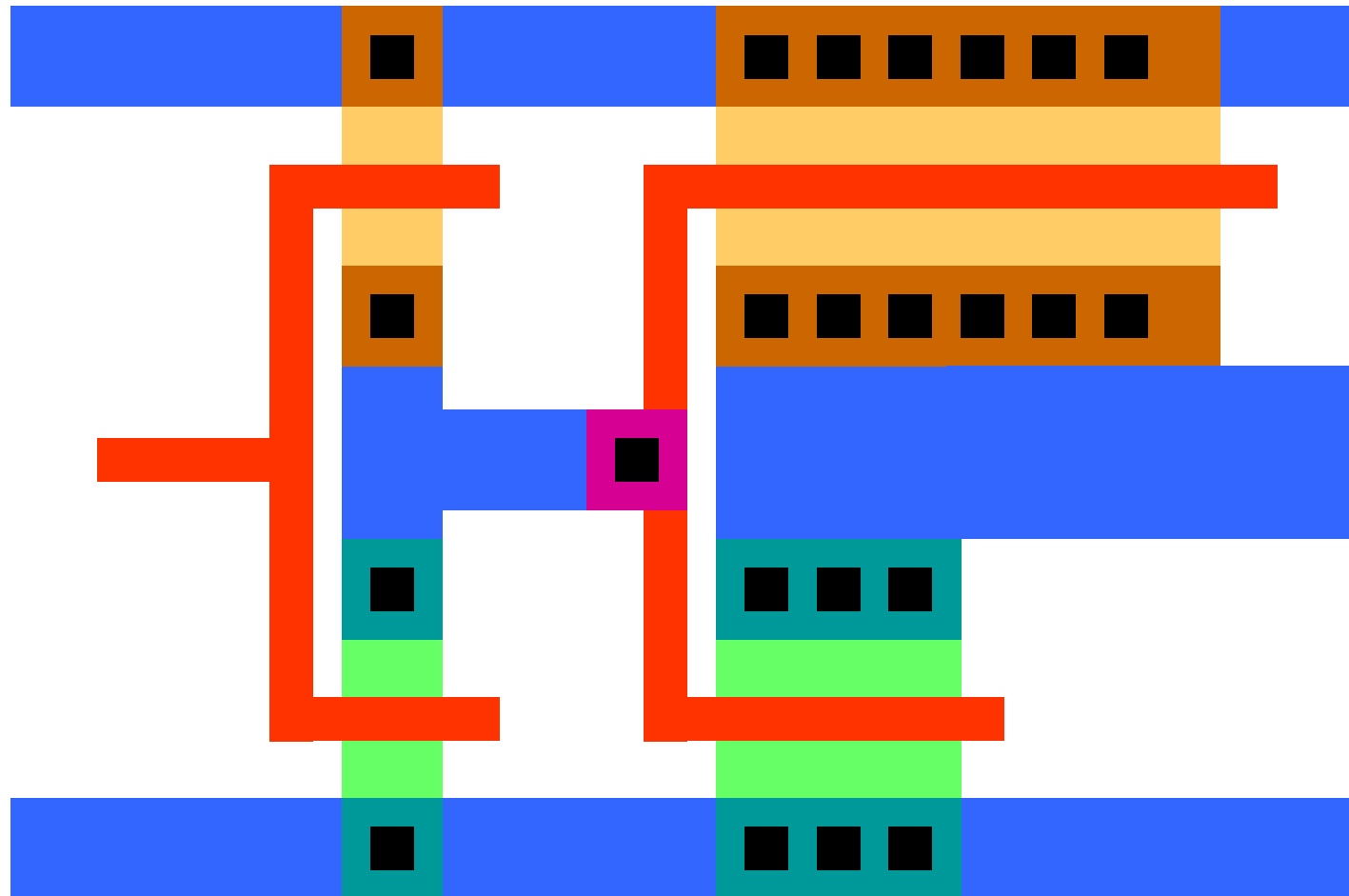


$$a + b + c + d = \overline{\overline{a + b + c + d}} = \overline{\overline{a + b} \cdot \overline{c + d}}$$

Was machen wir, wenn wir Gatter mit zu großem Fan-Out haben?



Inverterpaar als Treiber auf dem Chip von oben



2.16 Standard-Schaltnetze

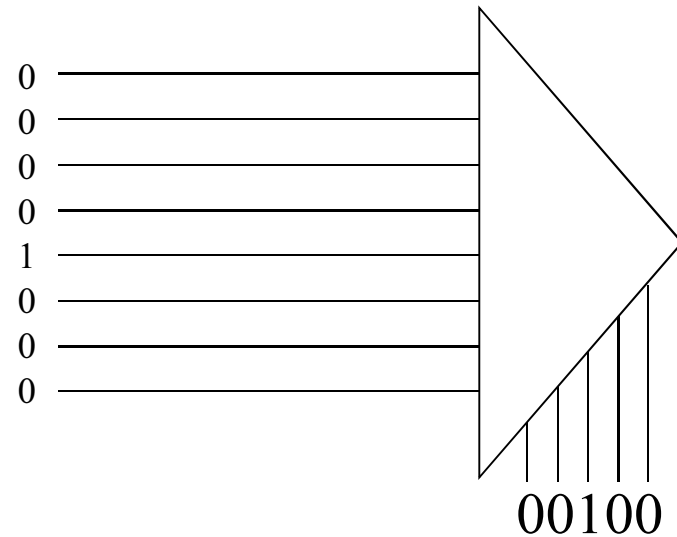
Bisher haben wir uns einen Werkzeugkasten zum Umsetzen von Funktionsbeschreibungen in Schaltungen zugelegt. Das ist quasi unser 1x1. Wie in der Schule folgen auf das 1x1 jetzt viele weitere Operationen: Dividieren, Wurzel ziehen, Potenzieren, Logarithmieren, Integrale.... Also Operationen, deren Bearbeitung das 1x1 erfordert. In diesem Stadium sind wir jetzt.

Wir lernen jetzt Grundschaltungen kennen, denen wir in Hardwaresystemen immer wieder begegnen werden, und die mit unserem Werkzeugkasten zu verstehen sind. Ziel ist, dass Sie am Ende ein Gefühl dafür haben, was ein Addierer macht, was ein Multiplexer, ein Zähler usw.

Der Codierer

Ein Codierer ist ein Schaltnetz, das eine Nachricht in eine definierten Code als Input bekommt, und diese in einen anderen Code wandelt und ausgibt. Sie erinnern sich an den Binär-Aiken-Codierer für die Dezimalziffern.

Häufig möchte man, dass einer der Codes ein 1-aus-N-Code ist. Dies ist eine Codierung der Zahlen von 0 bis N-1, bei der genau ein Bit 1 ist und alle anderen 0. Ein 1-aus-N-Code hat genau N verschieden Codeworte. Wie baut man einen Codierer, der einen 1-aus-N-Code in einen Binärcode übersetzt?

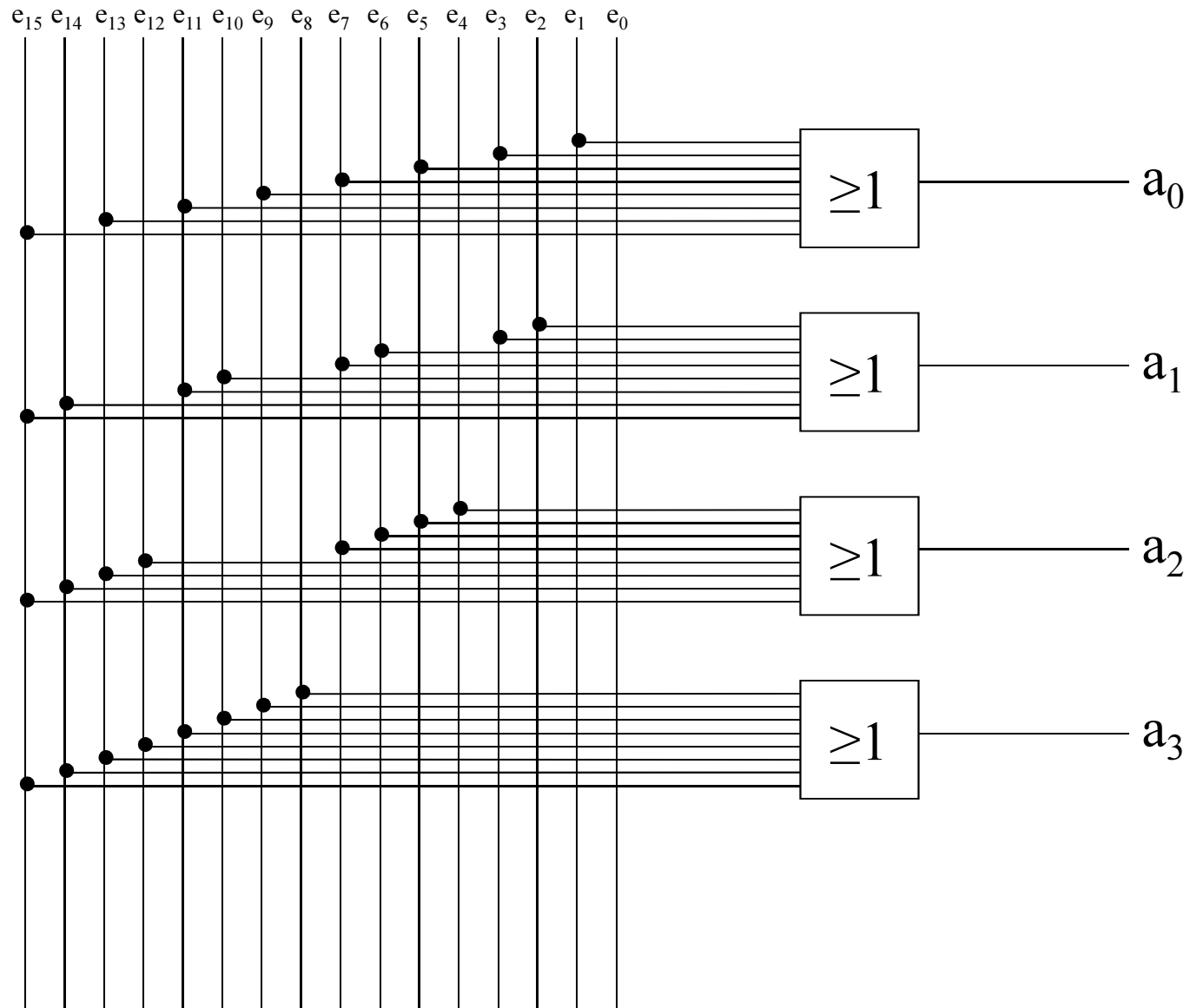


Relevanter Teil der Wertetabelle für 1-aus-16-in-Binär-Codierer

e ₁₅	e ₁₄	e ₁₃	e ₁₂	e ₁₁	e ₁₀	e ₉	e ₈	e ₇	e ₆	e ₅	e ₄	e ₃	e ₂	e ₁	e ₀	a ₃	a ₂	a ₁	a ₀
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

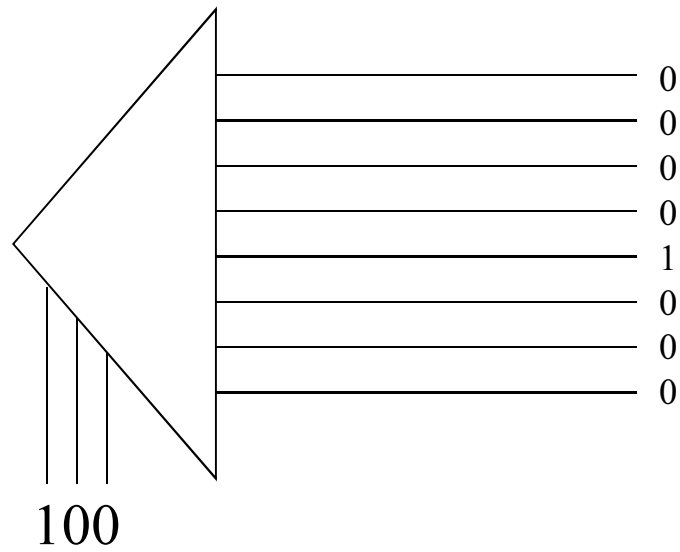
Alle anderen Werte sind X (don't care)

1-aus-16-in-Binär-Codierer



Der Decodierer

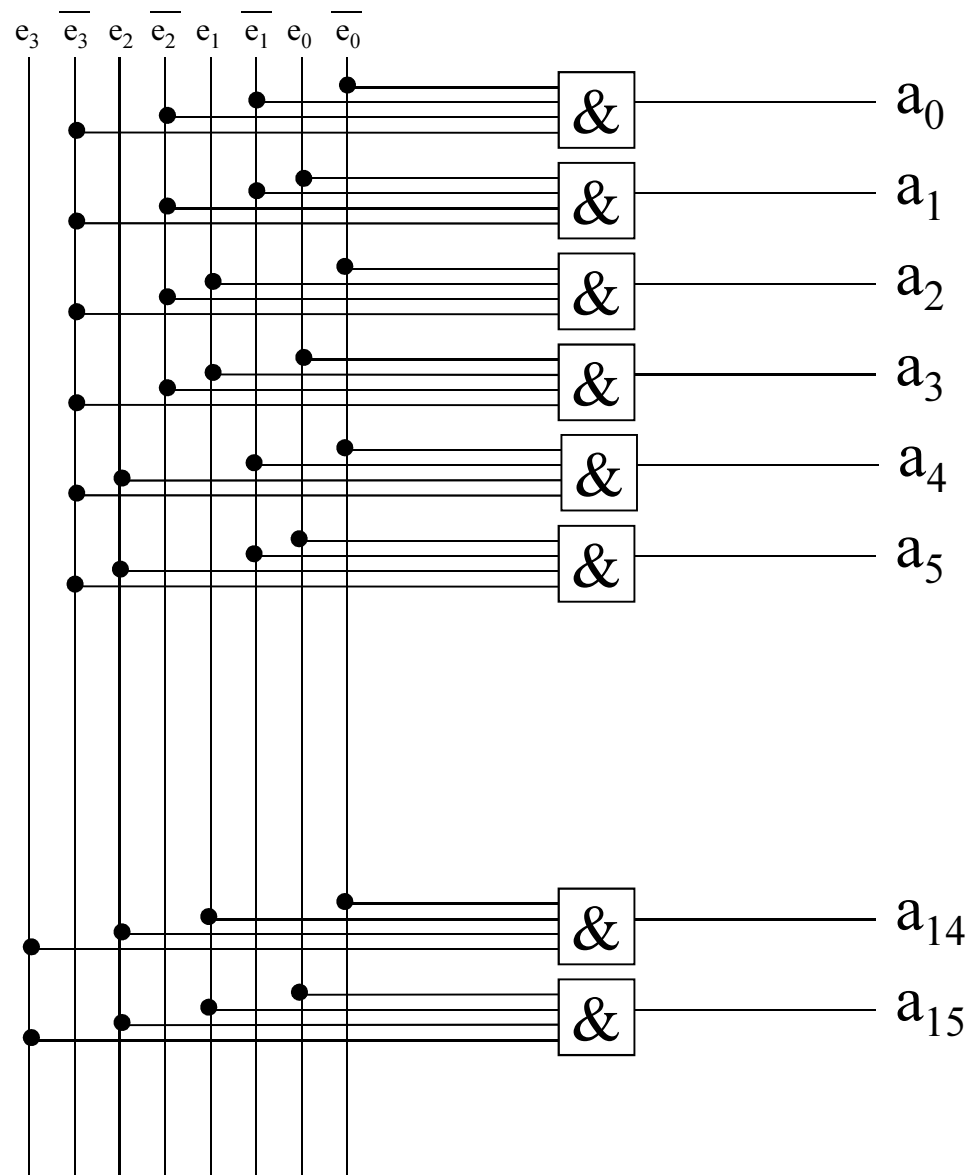
Jetzt ist umgekehrt der Eingang ein Binärcode und der Ausgang ein 1-aus-N-Code ist.



Wertetabelle für Binär-in-1-aus-16-Decodierer

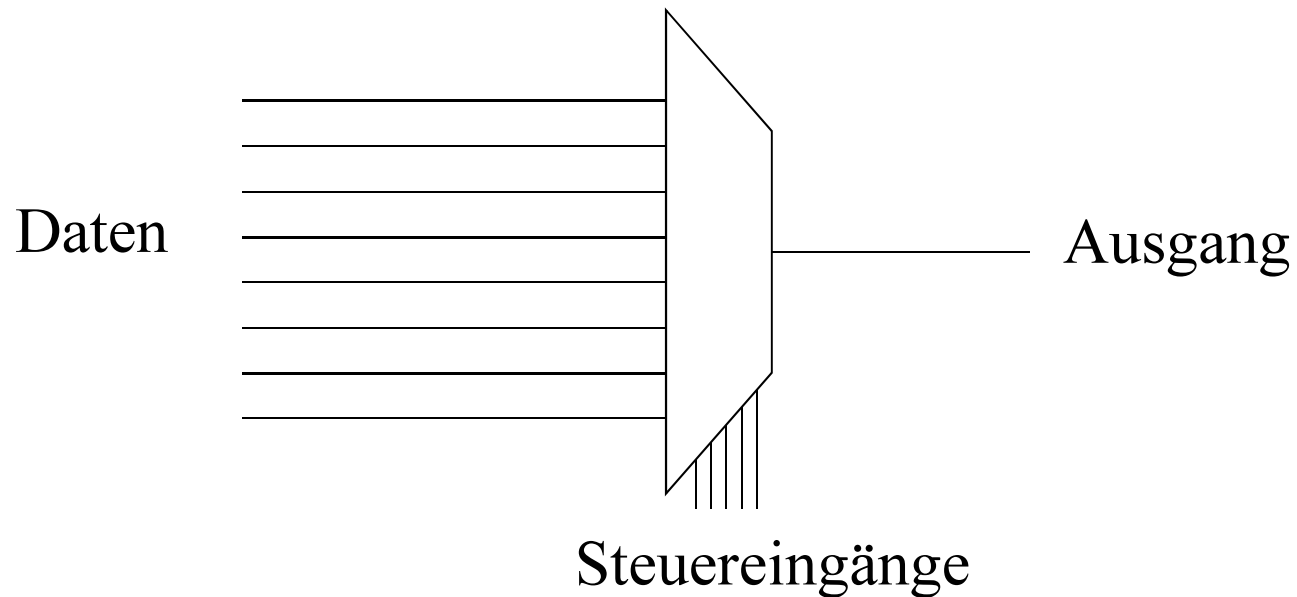
e ₃	e ₂	e ₁	e ₀	a ₁₅	a ₁₄	a ₁₃	a ₁₂	a ₁₁	a ₁₀	a ₉	a ₈	a ₇	a ₆	a ₅	a ₄	a ₃	a ₂	a ₁	a ₀	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
0	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Binär-in-1-aus-16-Decodierer



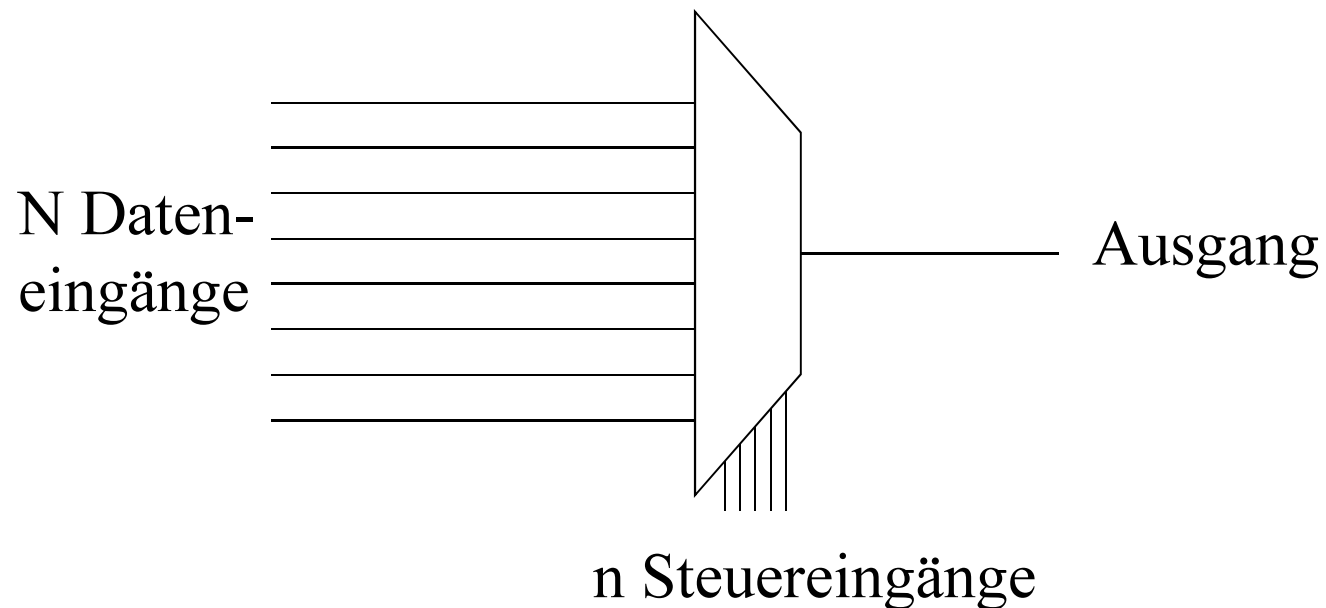
Der Multiplexer

Ein Multiplexer ist ein Schaltnetz, das einen von mehreren Eingängen wählt, und diesen unverändert auf den Ausgang legt. Man kann sich einen Multiplexer wie eine Verzeigung (oder Weiche) vorstellen. Multiplexer haben Dateneingänge und Steuereingänge. Abhängig von den Signalen der Steuereingänge wird der richtige Dateneingang ausgewählt.

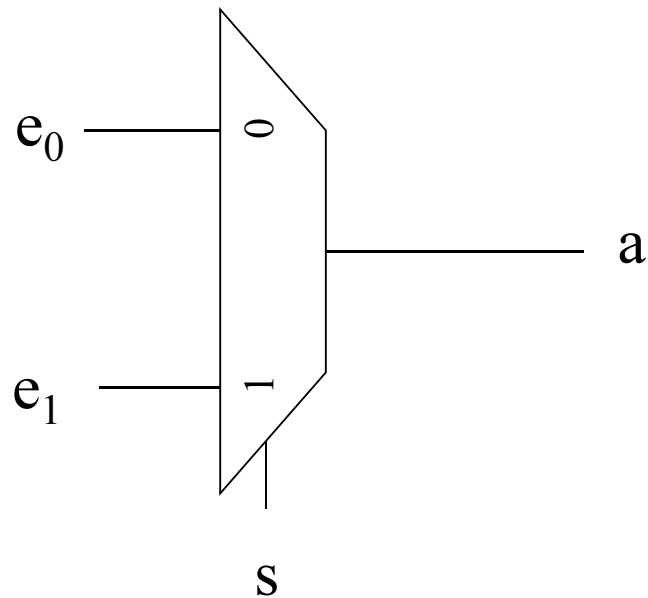


Seien die Dateneingänge mit e_i bezeichnet, $i=0, \dots, N-1$, die Steuereingänge mit s_i , $i=0, \dots, n-1$, der Ausgang mit a . Dann muss $N \leq 2^n$ sein. Die Funktion des Multiplexers wird beschrieben durch

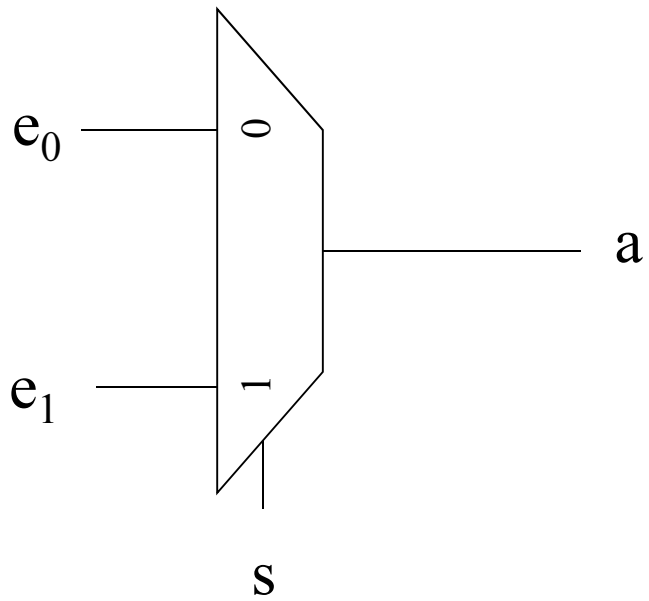
$$a = e_i, \text{ falls } (i)_{10} = (s_{n-1} s_{n-2} \dots s_0)_2$$



Beispiel: Ein 2-auf-1-Multiplexer. Dieser hat 2 Dateneingänge e_0 und e_1 und einen Steuereingang s . Wenn die Steuerleitung $s=0$ ist, so ist $a=e_0$ und wenn $s=1$ ist, so ist $a=e_1$.



s	e_1	e_0	a
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



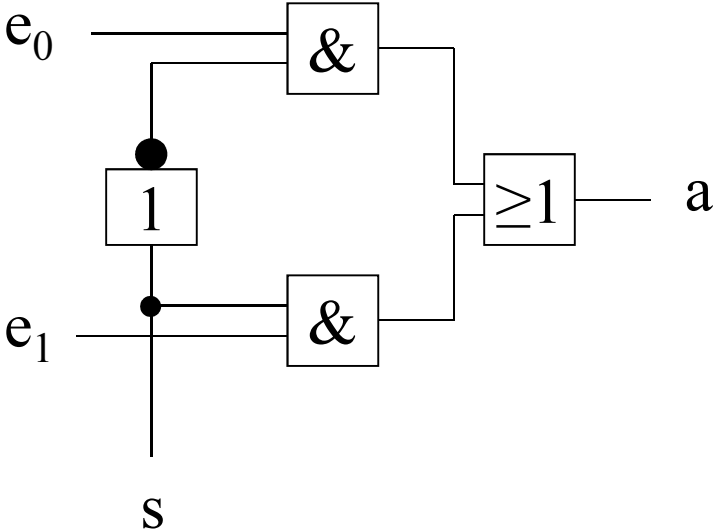
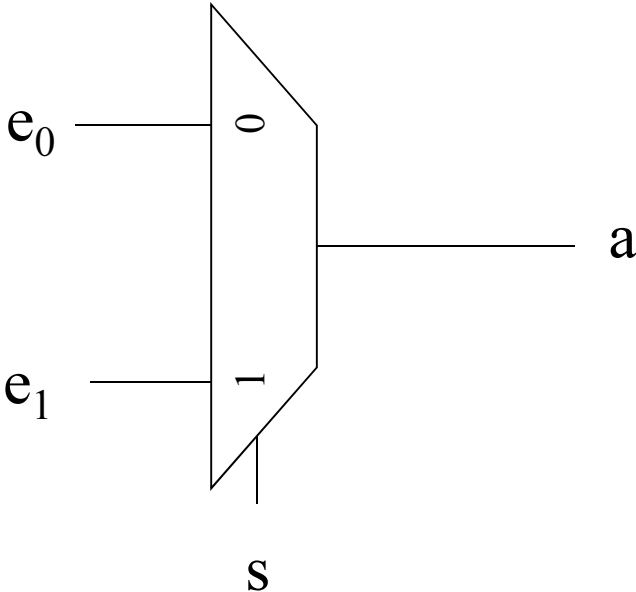
s	e ₁	e ₀	a
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

		e ₀			
s	a	0	1	1	0
		1	1	0	0
		e ₁			

$$a = \bar{s}e_0 + se_1$$

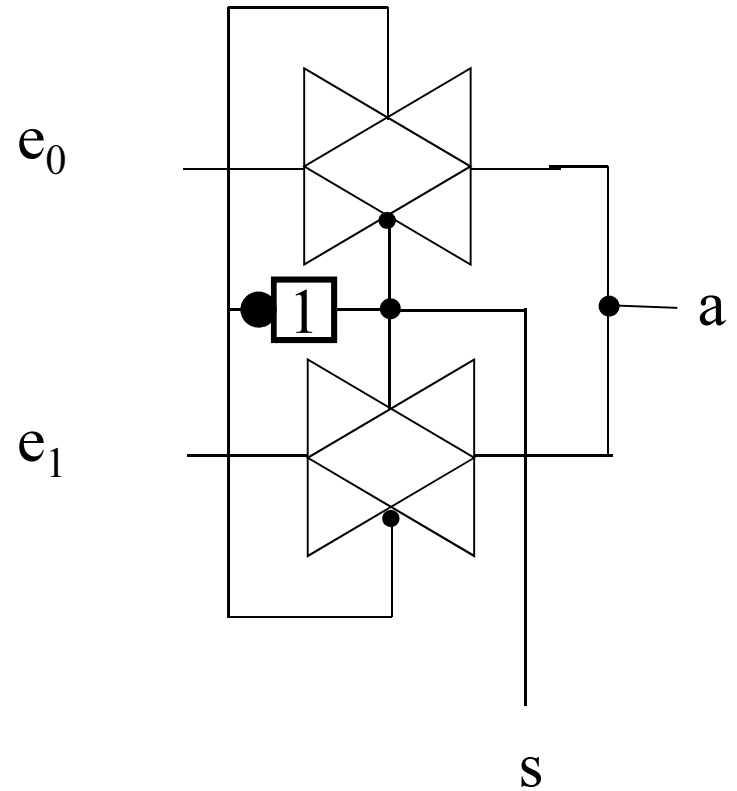
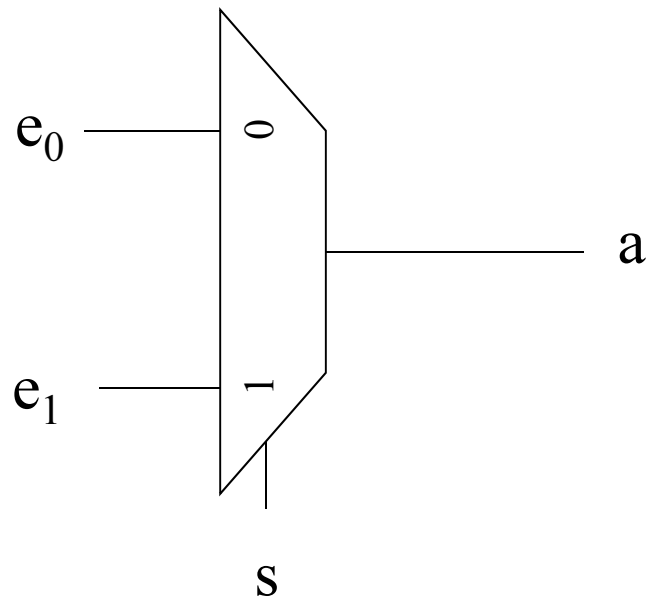
Realisierung:

$$a = \bar{s}e_0 + se_1$$



2. Möglichkeit der Realisierung:

$$a = \bar{s}e_0 + se_1$$

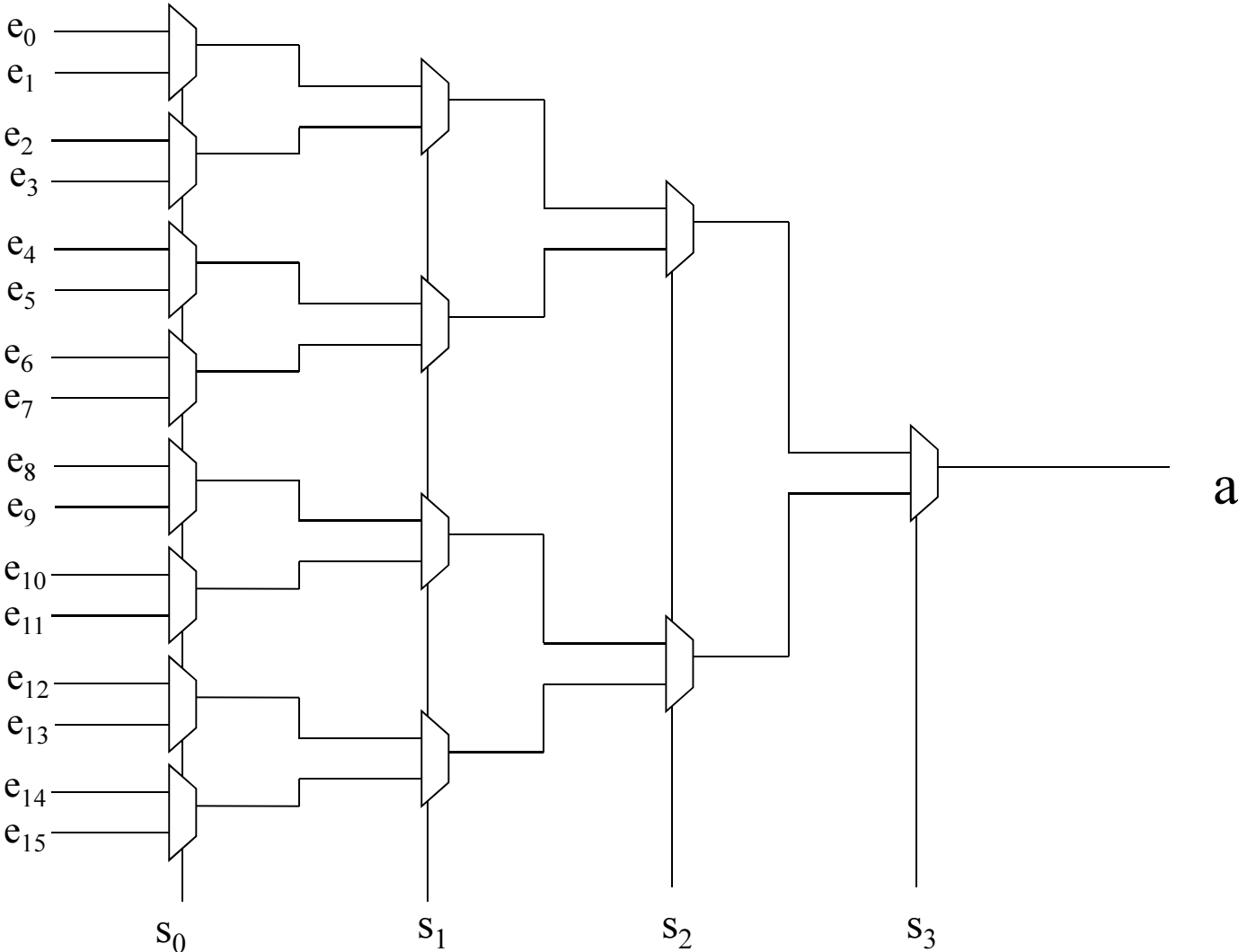


Wertetabelle für 16-auf-1-Multiplexer

s ₃	s ₂	s ₁	s ₀	a
0	0	0	0	e ₀
0	0	0	1	e ₁
0	0	1	0	e ₂
0	0	1	1	e ₃
0	1	0	0	e ₄
0	1	0	1	e ₅
0	1	1	0	e ₆
0	1	1	1	e ₇
1	0	0	0	e ₈
1	0	0	1	e ₉
1	0	1	0	e ₁₀
1	0	1	1	e ₁₁
1	1	0	0	e ₁₂
1	1	0	1	e ₁₃
1	1	1	0	e ₁₄
1	1	1	1	e ₁₅

stark verkürzte Darstellung.
Vollständige Wertetabelle
hätte $2^{20} \approx 1.000.000$ Zeilen.

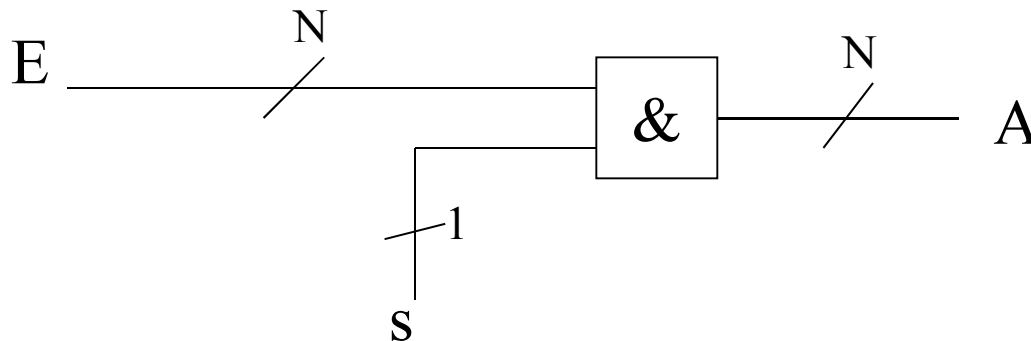
Mögliche Realisierung:



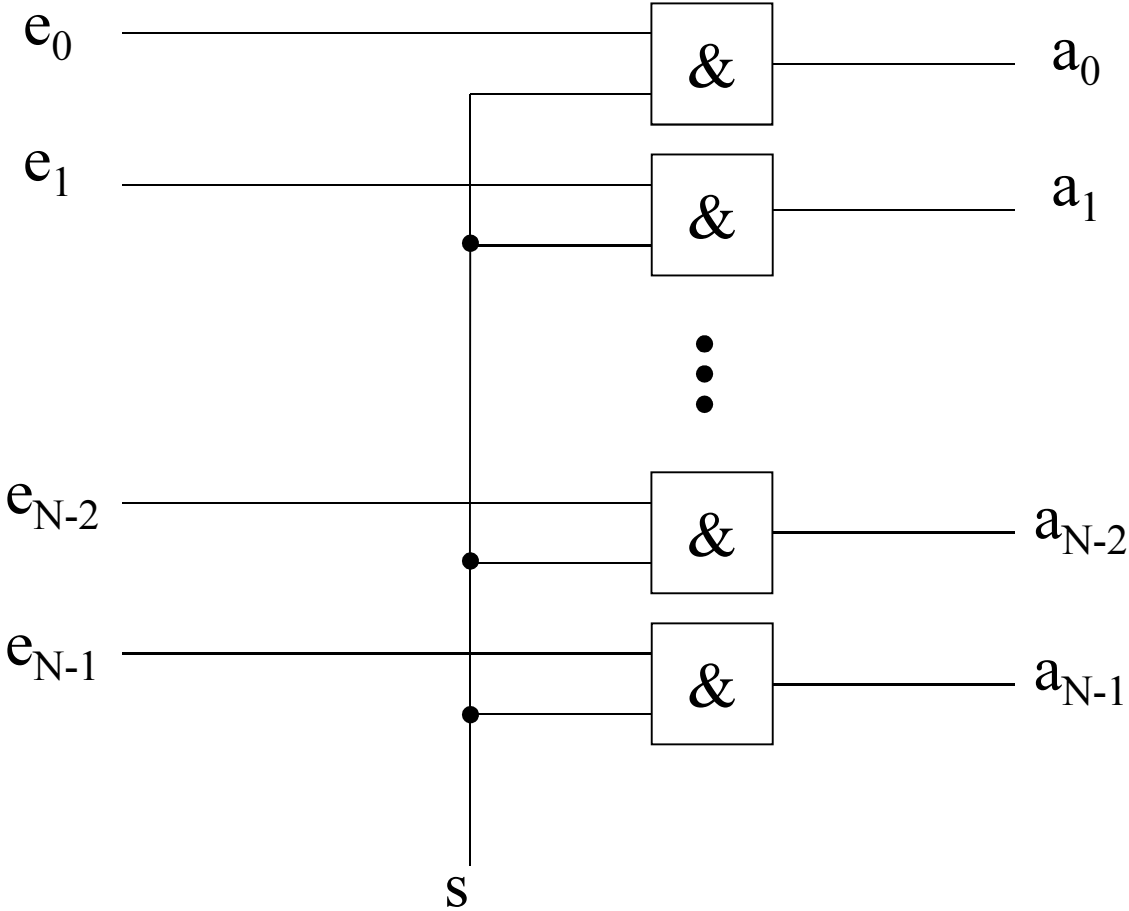
Der Datenwegschalter

Ein Datenwegschalter ist ein Schaltnetz, das eine Reihe von Eingängen e_i unverändert an eine entsprechende Reihe von Ausgängen a_i weiterleitet, wenn ein Steuereingang auf 1 gesetzt ist ($i=0,\dots,N-1$). Wenn der Steuereingang auf 0 ist, wird an alle Ausgänge eine 0 gegeben.

$$a_i = \begin{cases} e_i, & \text{falls } s=1 \\ 0, & \text{falls } s=0 \end{cases}$$

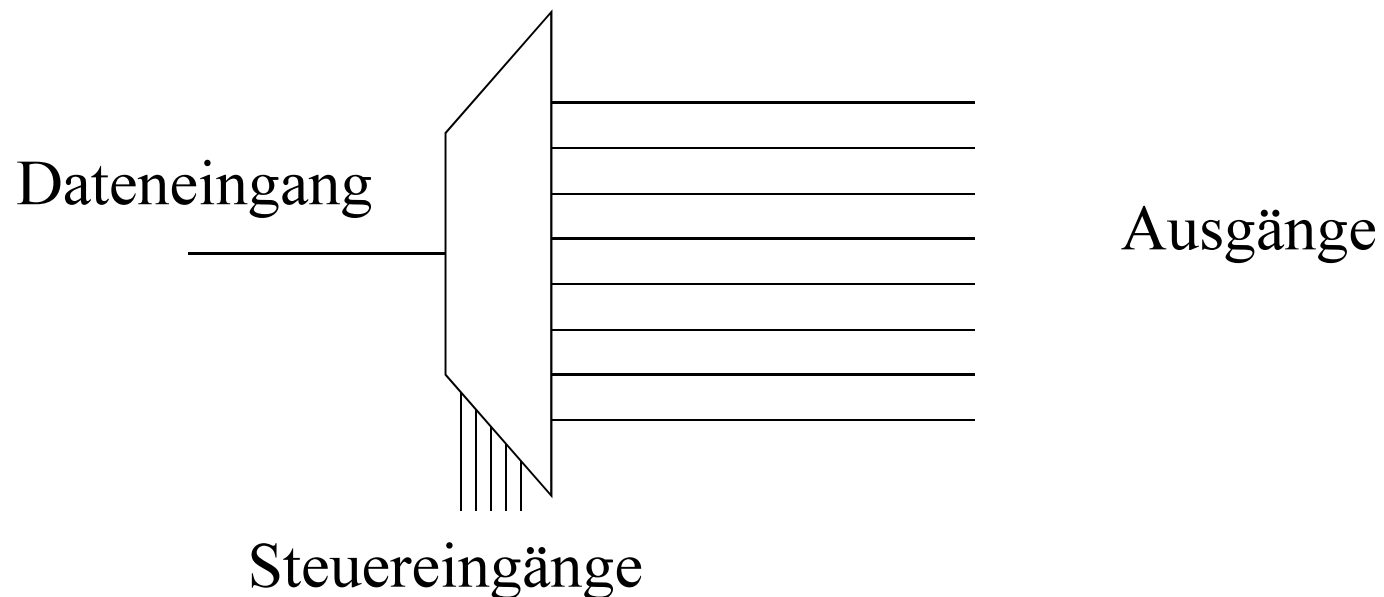


Realisierung des Datenwegschalters



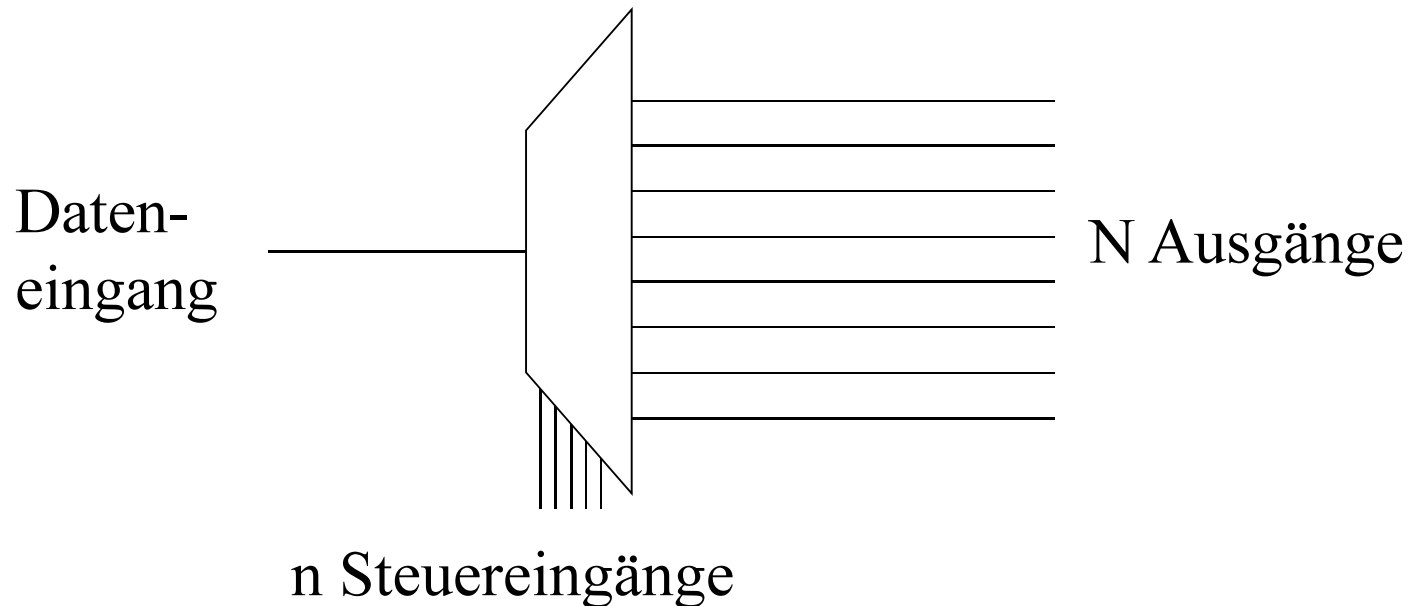
Der Demultiplexer

Ein Demultiplexer ist ein Schaltnetz, das seinen einzigen Dateneingang an einen von N Ausgängen übertragen kann. Die Auswahl, welcher Ausgang gewählt wird, wird durch die Steuereingänge bestimmt.



Seien die Datenausgänge mit a_i bezeichnet, $i=0, \dots, N-1$, die Steuereingänge mit s_i , $i=0, \dots, n-1$, der Eingang mit e . Dann muss $N \leq 2^n$ sein. Die Funktion des Demultiplexers wird beschrieben durch

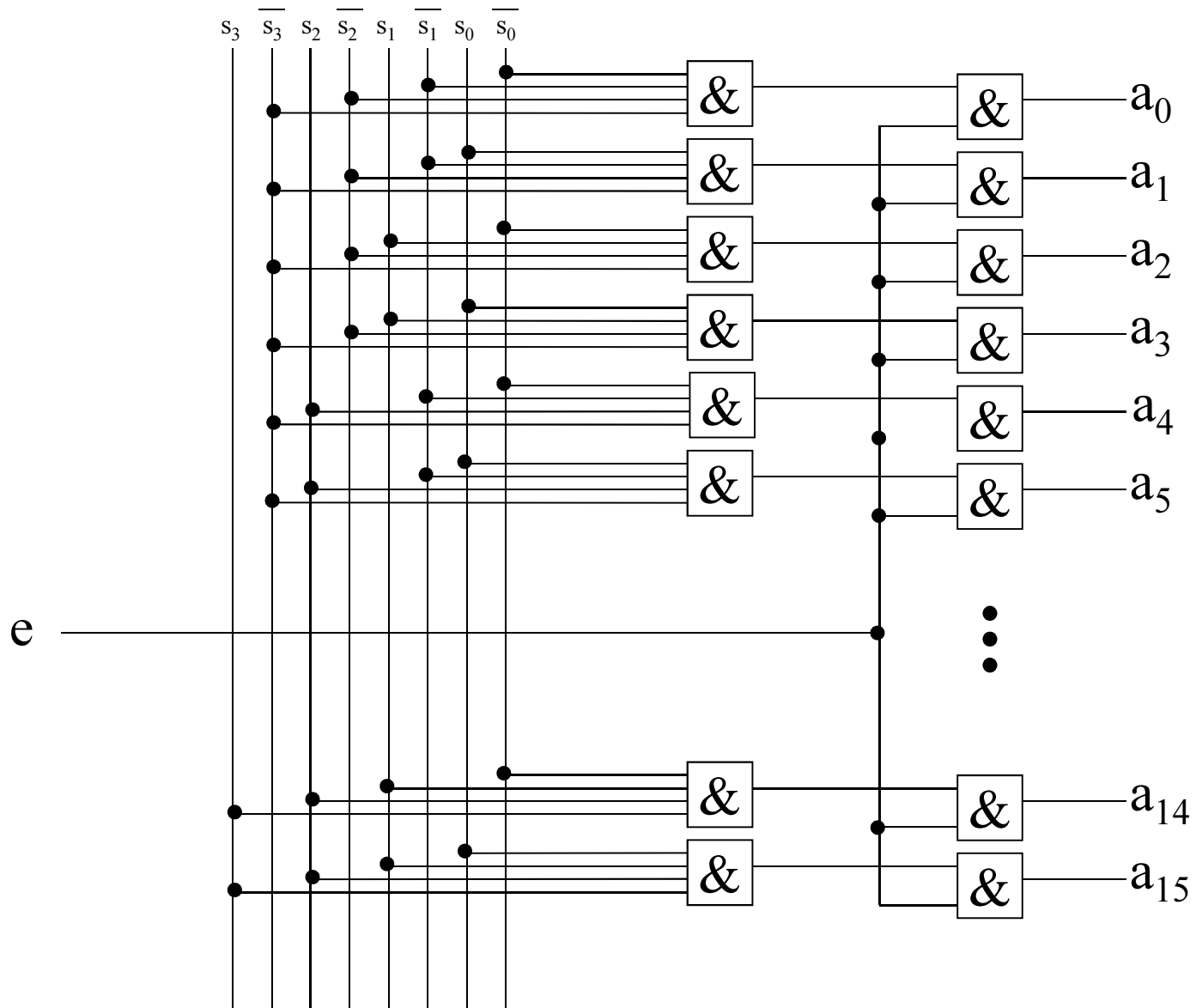
$$a_i = e, \text{ falls } (i)_{10} = (s_{n-1}s_{n-2}\dots s_0)_2, \text{ sonst } a_i = 0$$



Wertetabelle für 1-aus-16-Demultiplexer

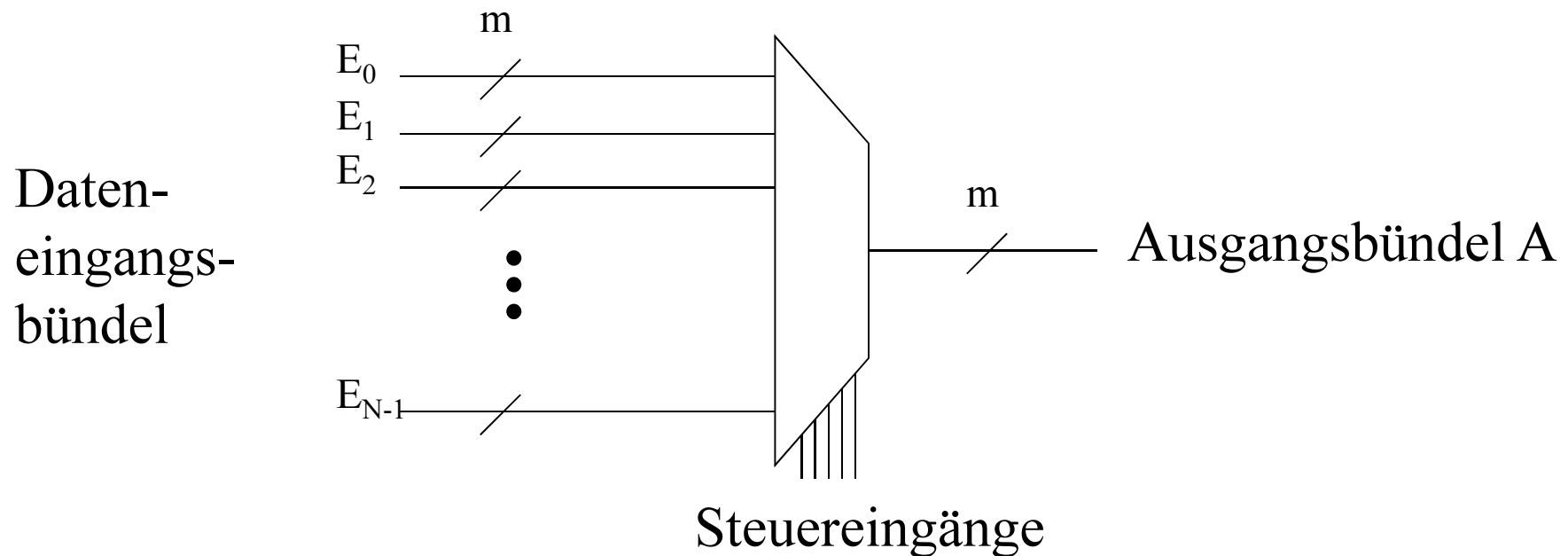
e	s ₃	s ₂	s ₁	s ₀	a ₁₅	a ₁₄	a ₁₃	a ₁₂	a ₁₁	a ₁₀	a ₉	a ₈	a ₇	a ₆	a ₅	a ₄	a ₃	a ₂	a ₁	a ₀	
0	X	X	X	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
1	0	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
1	0	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
1	1	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Realisierung eines 1-aus-16-Demultiplexers



Der Datenwegmultiplexer

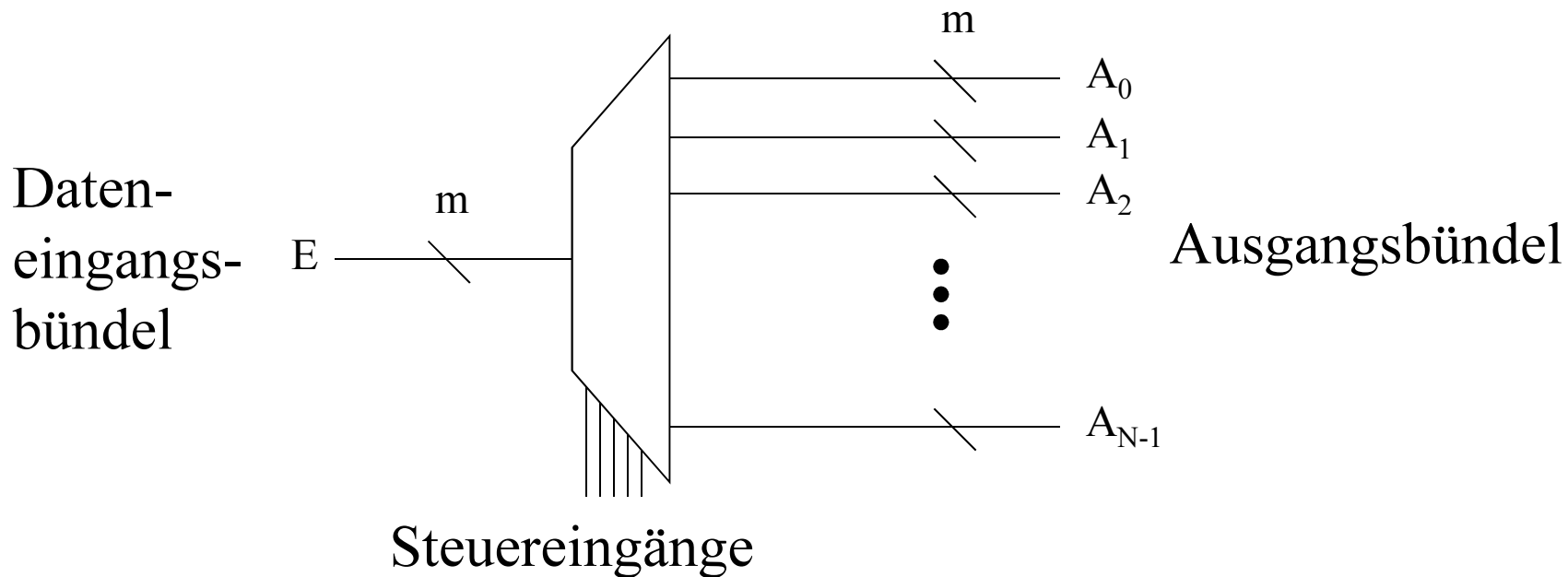
Ein Datenwegmultiplexer ist ein Schaltnetz, das aus m parallel geschalteten Multiplexern besteht. Mit ihm kann man Leitungsbündel der Stärke m schalten. Alle parallelen Leitungen werden durch die Steuereingänge auf das selbe Ausgangsbündel geschaltet.



Für alle $i=0, \dots, m-1$ gilt $a_i = e_{k,i}$ genau dann wenn $(k)_{10} = (s_{n-1} \dots s_0)_2$

Der Datenwegdemultiplexer

Ein Datenwegdemultiplexer ist ein Schaltnetz, das aus m parallel geschalteten Demultiplexern besteht. Mit ihm kann man ein Leitungsbündel der Stärke m auf eins von N Ausgangsleitungsbündeln schalten.



Für alle $i=0, \dots, m-1$ gilt $a_{k,i} = e_i$ genau dann wenn $(k)_{10} = (s_{n-1} \dots s_0)_2$
alle anderen $a_{k,i}$ sind 0.

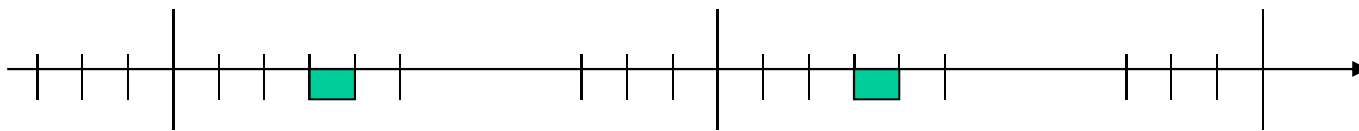
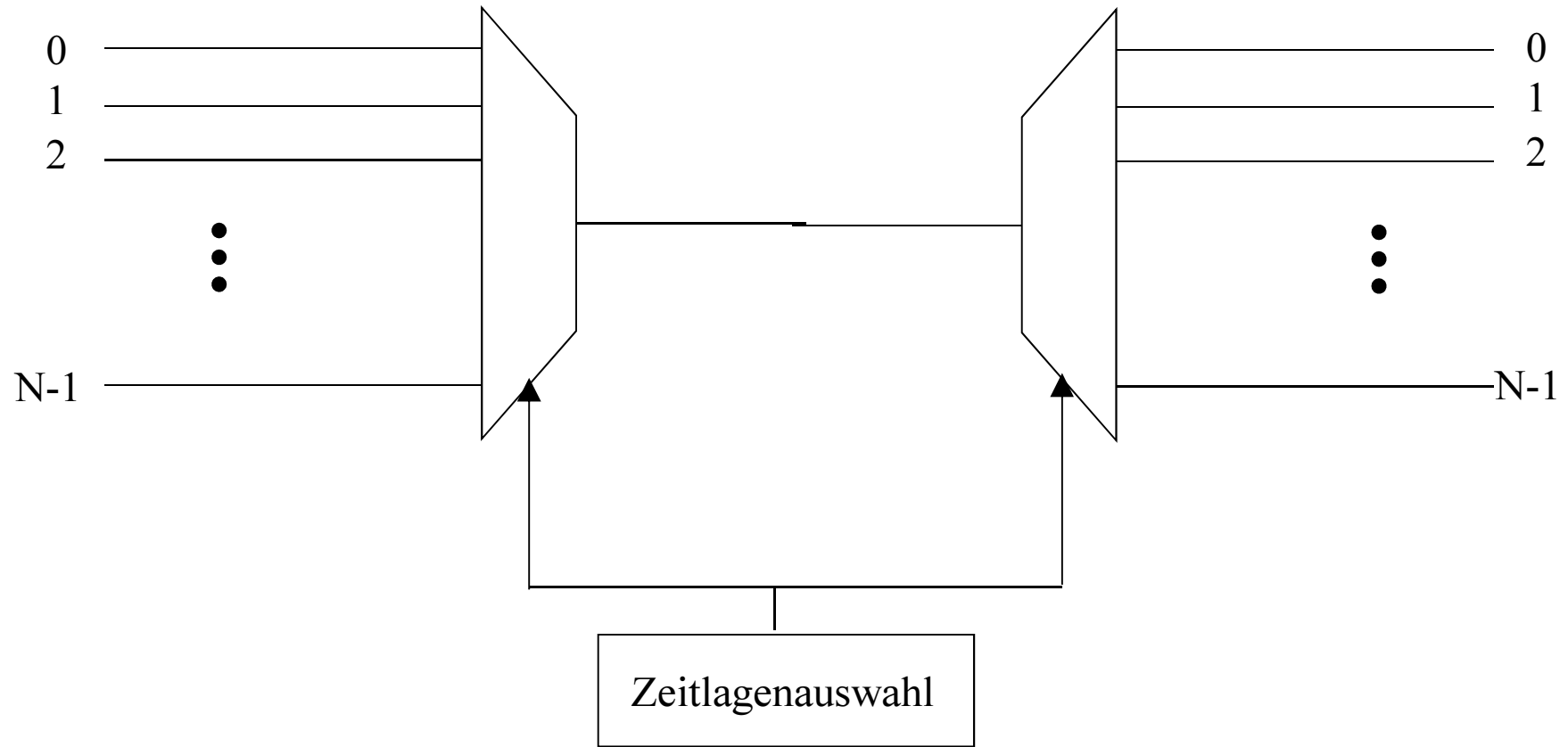
Vereinfachte Wertetabelle des Datenweg-Demultiplexers

s_{n-1}	s_{n-2}	s_2	s_1	s_0	Ausgangsbündel
0	0	0	0	0	$A_0 = E, A_i = 0$ für $i \neq 0$
0	0	0	0	1	$A_1 = E, A_i = 0$ für $i \neq 1$
0	0	0	1	0	$A_2 = E, A_i = 0$ für $i \neq 2$
0	0	0	1	1	$A_3 = E, A_i = 0$ für $i \neq 3$
0	0	1	0	0	$A_4 = E, A_i = 0$ für $i \neq 4$
\vdots						\vdots
1	1	1	1	0	$A_{N-2} = E, A_i = 0$ für $i \neq N-2$
1	1	1	1	1	$A_{N-1} = E, A_i = 0$ für $i \neq N-1$

Zeitmultiplexübertragung

Häufig ist es notwendig, dass über dieselbe Leitung mehrere verschiedene Kommunikationsvorgänge durchgeführt werden müssen. Ein einfaches Beispiel dafür ist Ihr Telefonanschluss zu Hause, über den (z.B. bei ISDN) zwei Telefongespräche gleichzeitig und vielleicht zusätzlich (über DSL) Datenverkehr mit Ihrem Rechner abgewickelt werden. Dies gelingt, indem man die Leitung immer abwechselnd für sehr kurze Zeitintervalle jedem der beteiligten Kommunikationspaare zur Verfügung stellt. Diese Technik nennt man **Zeitmultiplex-Technik**. Sie kann durch ein Schaltnetz aus einem Multiplexer und einem Demultiplexer realisiert werden, die beide durch eine gemeinsame Steuereinheit, die sogenannte **Zeitlagenauswahl**, gesteuert werden.

Zeitmultiplextechnik (Time Division Multiplex Access, TDMA)



Beispiel für Zeit-Multiplex: GSM-Netz

