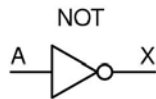
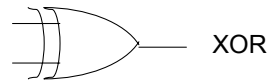

Grundlagen der Informatik

Beispiele:

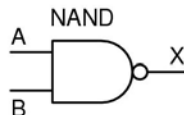
- 1-Bit-Arithmetisch-Logische-Einheit
 - 8-Bit-ALU
 - Speicher für vier 3-Bit-Worte
-

Amerikanische Notation



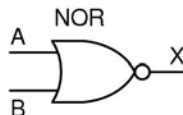
A	X
0	1
1	0

(a)



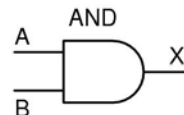
A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

(b)



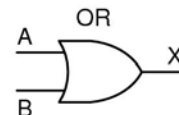
A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

(c)



A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

(d)

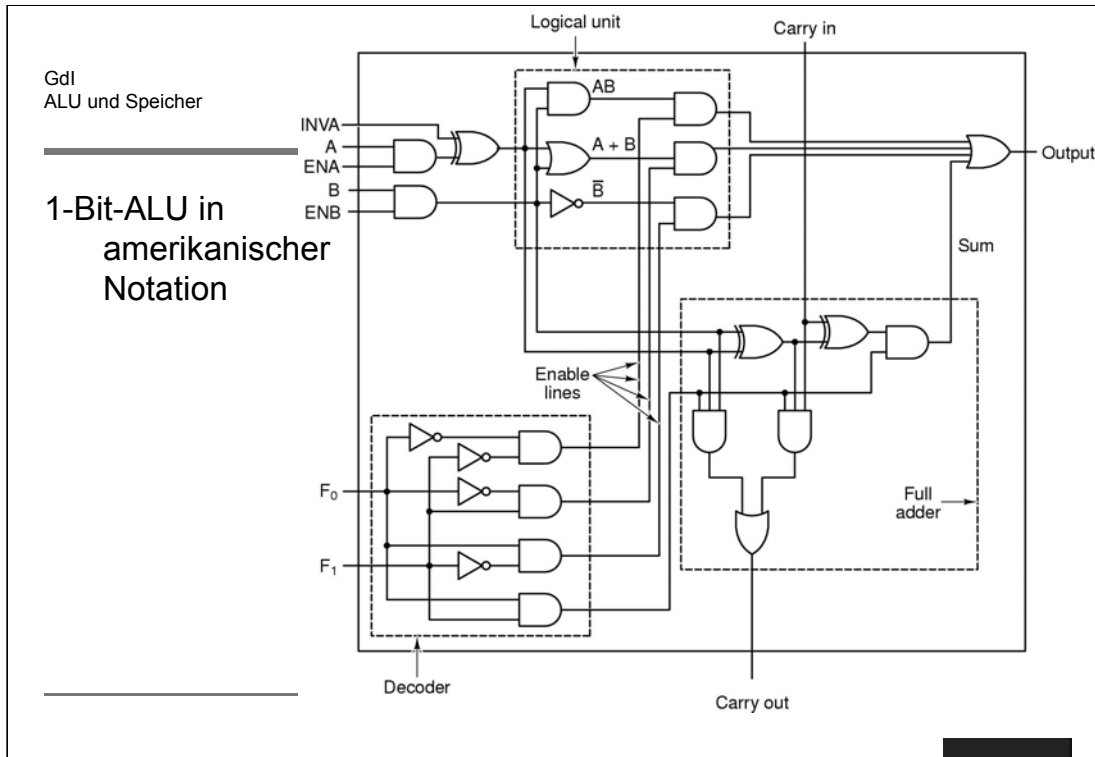


A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

(e)

Die Beispiele sind dem Buch von A. S. Tanenbaum „Structured Computer Organization“ entnommen. Dieses Buch verwendet die amerikanische Notation für Schaltsymbole, die sich von der Notation des GDI-Skripts sehr und von der ISO-Notation etwas unetrscheidet.

Diese Folie stellt diese Notation kurz vor.



Die abgebildete 1-Bit-ALU berechnet vier Werte gleichzeitig:

- $A \wedge B$, $A \vee B$, $\neg B$ in der logischen Einheit („Logical Unit“)
- Summe und Übertrag von $(A+B+\text{Carry_In})$ im Volladdierer

Der Dekodierer links unten wählt genau eines dieser Ergebnisse (bei der Addition die Summe) aus und schaltet sie auf die Ausgabe:

- $(F_0, F_1) = (0, 0)$: $A \wedge B$
- $(F_0, F_1) = (0, 1)$: $A \vee B$
- $(F_0, F_1) = (1, 0)$: $\neg B$
- $(F_0, F_1) = (1, 1)$: $A + B$

Die ALU kann somit diese vier „Maschinenbefehle“ ausführen.

Die Eingänge ENA, ENB und INVA spielen hier noch keine Rolle

Addition zweier Bits

$$A + B = \text{Carry} * 2 + \text{Output}$$

Befehlsfolge

Lade Bit A

Lade Bit B

Lade den Befehl F_1F_0 (hier 11=„+“)

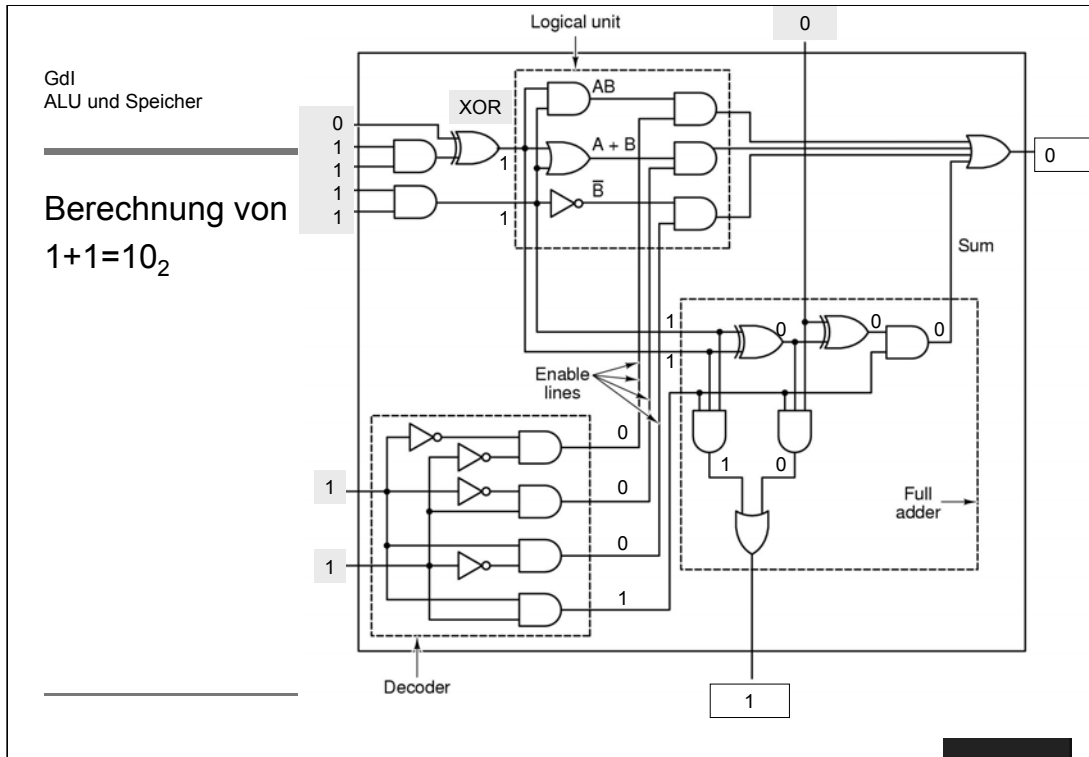
ALU rechnet

Speichere Output

Idee: Lade zunächst die beiden 1-Bit-Register \$A und \$B (Maschinenbefehle hier nicht darstellbar).

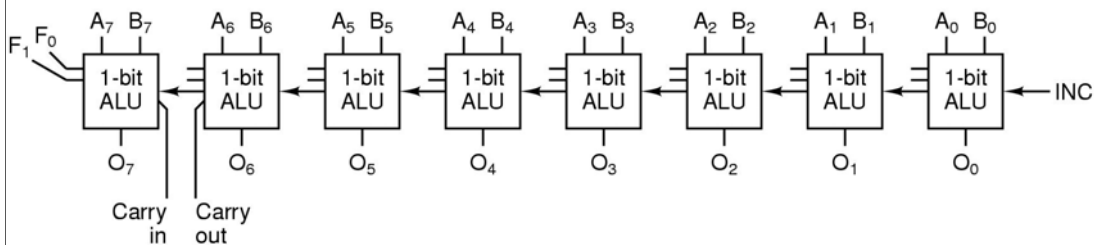
Dann lege einen der Befehle add, or, and oder not an die Befehlsleitungen der ALU.

Nach Berechnung der ALU liegt das Ergebnis im 1-Bit-Register \$Output.



Hier ist als Beispiel die Addition $1+1$ mit $\text{Carry_In}=0$ dargestellt.

8-Bit-ALU

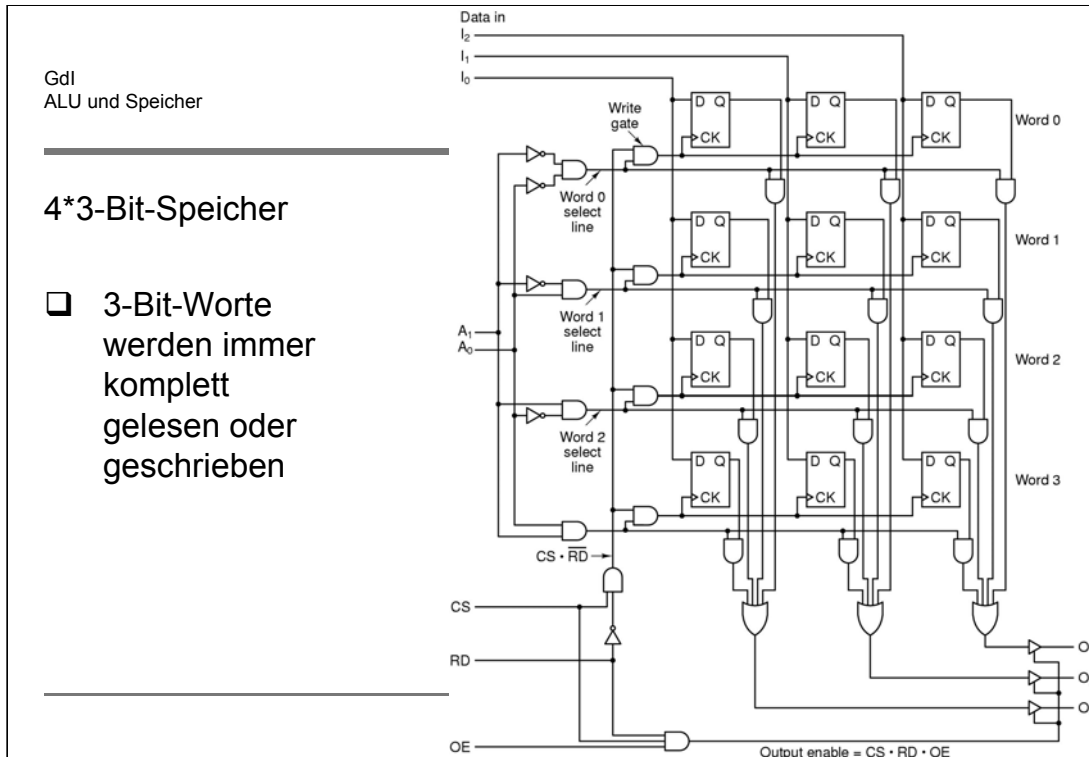


Diese 8-Bit-ALU ist aus identischen 1-Bit-ALUS aufgebaut.

Die Steuerleitungen F1, F0 sind für alle 1-Bit-ALUs identisch belegt.

Der Pfeil von ALU i zu ALU i+1 ist der Übertrag („Carry“), der so weitergeleitet wird.

Mit dem INC-Eingang kann man auf einfache Art Werte wie $A+1$ oder $A+B+1$ berechnen (Verweis auf Einer- und Zweierkomplement).



I2, I1, I0: Daten-Input

O2, O1, O0: Daten-Output

A1, A0: Adressleitungen; mit Hilfe dieser Adressleitungen und dem nachgeschalteten Dekoder (die vier „Word Select“-UND-Gatter) wird eines der vier Speicherwörter zum Lesen oder Schreiben ausgewählt.

CS: Chip Select; CS=0 blockiert diesen Speicherchip, CS=1 aktiviert ihn.

RD=1: Read, RD=0: Write

OE Output Enable: Nur bei OE=1 wird ein Output ausgegeben.

Beispiel 1: Um 101 in Speicherwort 3 zu schreiben muss gelten:

I2=1, I1=0, I0=1, A1=1, A0=1, CS=1, RD=0 (OE=0)

Beispiel 2: Um Speicherwort 1 auszulesen muss gelten:

A1=0, A0=1, CS=1, RD=1, OE=1; Speicherwort 1 liegt dann an O2, O1, O0 an.