

Additions- und Subtraktionsschaltungen

www.geocities.com/hackingcrew2000

Versuchsziel:

1. Kennenlernen des Aufbaus und der Funktionsweise von Additions- und Subtraktionsschaltungen sowie von digitalen Komparatoren
2. Festigung der Kenntnisse und Fertigkeiten zum Schaltungsentwurf und zur Schaltungsvereinfachung

Studienkontrollfragen für die Versuchsvorbereitung:

1. Wiederholen Sie den Aufbau und die Funktionsweise von Additionsschaltungen!
2. Charakterisieren Sie die Vor- und Nachteile serieller und paralleler Additionsschaltungen!
3. Welche Möglichkeiten gibt es zur Subtraktion dualer Zahlen?
4. Nennen sie die Vor- und Nachteile beider Arten der Subtraktion!
5. Was unterscheidet Übertrag und Überlauf bzw. Carry und Overflow?
6. Erläutern Sie den Aufbau einer Vollsubtrahierschaltung!
7. Welche Funktionen realisiert eine Arithmetisch-Logische-Einheit (ALU)?
8. Beschreiben Sie Funktion und Anwendungsbeispiele für digitale Komparatoren!
9. Entwerfen Sie die Schaltungen für die Versuchsaufgaben!

1. Additionsschaltungen

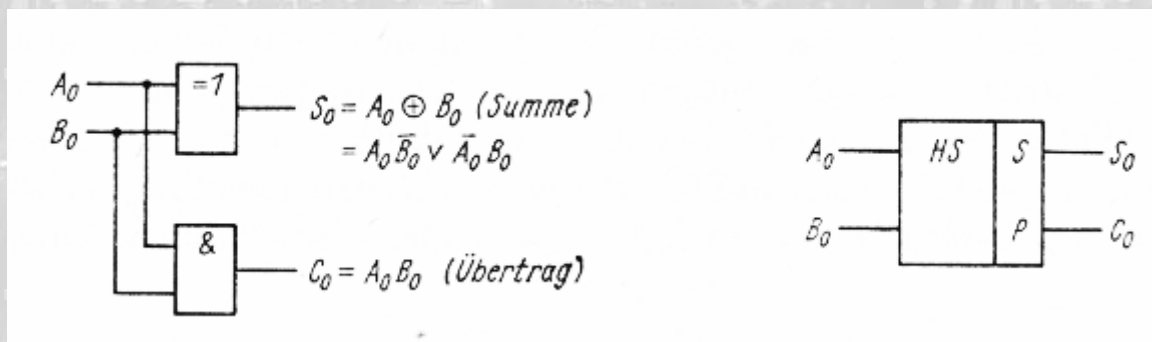
Halbaddierer

- für duale Addition zweier einstelliger Dualzahlen
- 2 Ausgänge: Summe (S_0), Übertrag (C_0)

$$0+0=0 \quad 0+1=1 \quad 1+0=1 \quad 1+1=10$$

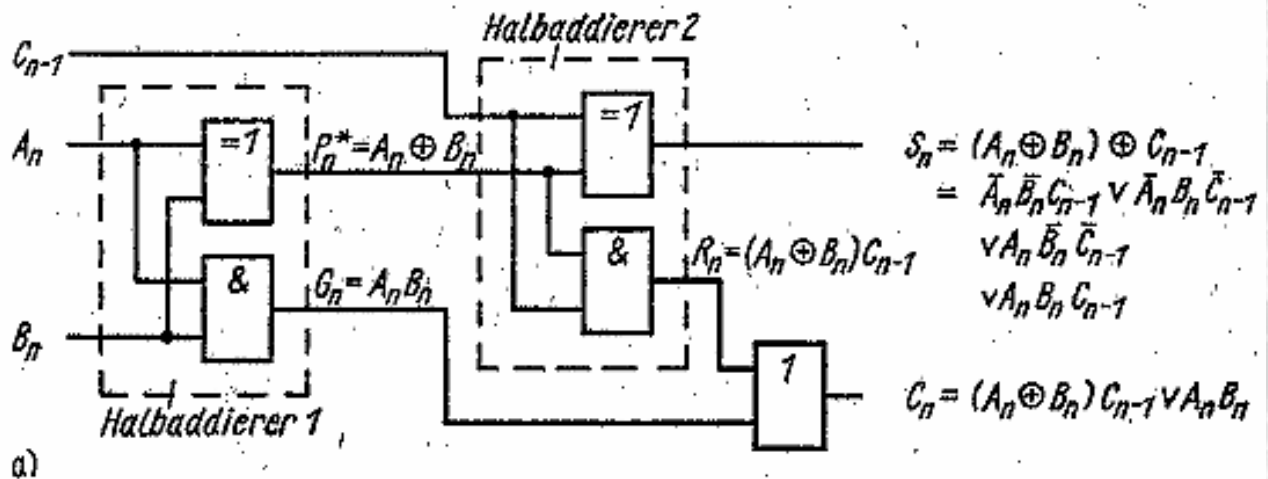
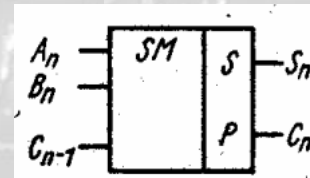
$$C_0 = A_0 \cdot B_0 \quad S_0 = \overline{A_0} B_0 \vee A_0 \overline{B_0} = A_0 \sim B_0$$

A_0 x_1	B_0 x_0	S_0	C_0
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



Volladdierer

- Addition mehrerer Bits
- Übertrag geht mit in Summe der nächsthöheren Stelle ein
- 3 Eingänge (A_n , B_n , C_{n-1} = Übertrag), 2 Ausgänge (S_n , C_n)



A_n	B_n	C_{n-1}	S_n	C_n
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1

2. Eigenschaften von Addierern

serieller Addierer:

- minimaler Schaltungsaufwand
- zu addierende Eingangszahlen müssen so lange anliegen, bis Übertrag des vorletzten Volladdierers am letzten Volladdierer anliegt
- Ergebnis gelangt seriell in Ergebnisregister E
- Schiebetaktimpuls leitet niederstes Bit des Registers A,B in den Volladdierer, gleichzeitig gelangt Additionsergebnis der vorausgegangenen Addition in die linke Zelle des Ergebnisregisters E
- das Übertragsbit wird in D-FlipFlop zwischengespeichert und beim nächsten Schiebetakt wird dem Übertragungseingang (C_{n-1}) des Volladdierers zugeführt

paralleler Addierer:

- Übertrag durchläuft alle Stufen seriell
- zu addierende Eingangszahlen müssen so lange anliegen bis der Übertrag des letzten Volladdierers

3. Subtraktion dualer Zahlen

- a)
- $$\begin{aligned} 0-0 &= 0 \\ 1-0 &= 1 \\ 1-1 &= 0 \\ 0-1 &= 1 \quad \text{Entleihung von 1 von nächster Stelle} \\ 0-1-1 &= 0 \quad \text{Entleihung von 1 von nächster Stelle} \end{aligned}$$

- b) Addition des Zweierkomplements vom Subtrahenden zum Minuend, dabei gilt die höchste Bitstelle als Vorzeichen ($1 \hat{=} +$, $0 \hat{=} -$). Ist das Vorzeichen „negativ“ (Subtrahent > Minuend), dann wird vom Ergebnis nochmals das Zweierkomplement gebildet.

4. Vor- und Nachteile

- a)
- im Falle negativer (dezimaler) Ergebnisse tritt ein rekursiver Überlauf nach links auf
 - höherer Schaltungsaufwand zur Überprüfung der Zahlenwerte von Minuend und Subtrahent
- b)
- Addition von größeren Subtrahenden von Minuenden möglich, da im höchstwertigen Bit das Vorzeichen gespeichert ist (im Falle negativer Ergebnisse Bildung des Zweierkomplementes)

5. Übertrag / Überlauf

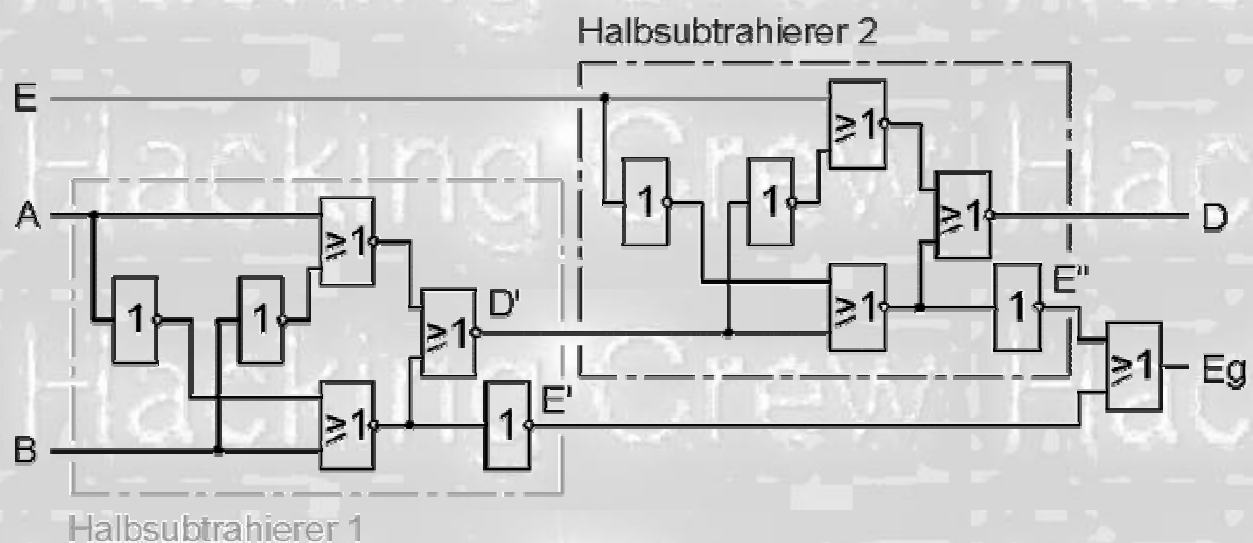
Übertrag („Carry“)

- entsteht ggf. bei Addition ($1 + 1$) durch Übergabe von 1 auf das nächsthöhere Bit

Überlauf („Overflow“)

- entsteht ggf. bei Subtraktion ($0 - 1$) durch „Entleihen“ von 1 von der nächsthöheren Bitstelle

6. Aufbau von Vollsubtrahierschaltungen

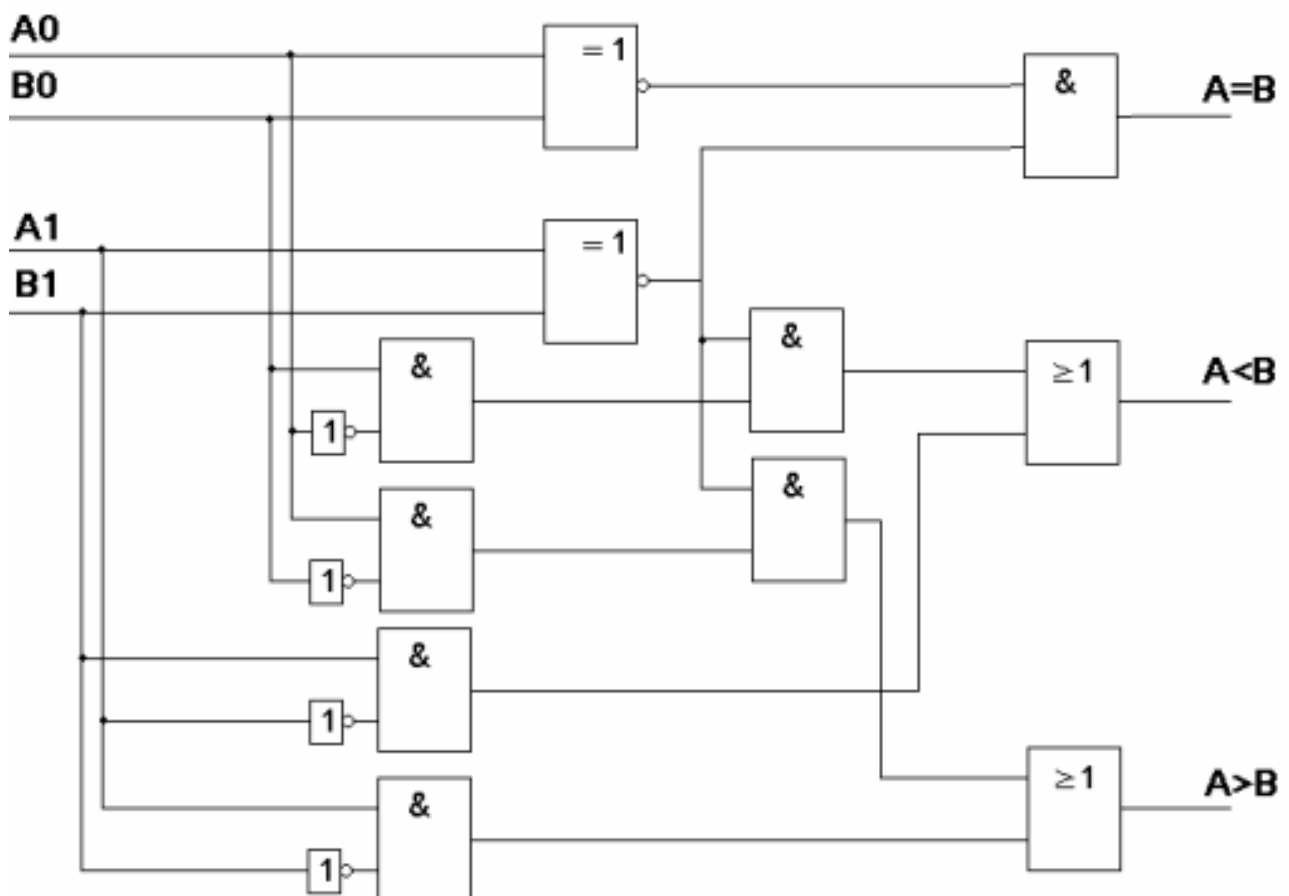


Halbsubtrahierer			
A	B	D'	E'
0	0	1	0
0	1	0	0
1	0	0	1
1	1	1	0

Vollsubtrahierer				
x	y	U_n	U_{n+1}	D
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

7. Funktionsweise einer ALU

- besteht aus Kombinationsschaltungen, die eine bestimmte Anzahl verschiedener arithmetischer und logischer Operationen mit zwei digitalen Eingangsworten ausführen kann
- z.B. Vergleichen ($>$, $<$, $=$), Addition, Subtraktion



$$\text{für } A=B: (A_0 \sim B_0) \cdot (A_1 \sim B_1)$$

$$\text{für } A>B: A_1 \cdot \overline{B_1} + \left((A_1 \sim B_1) \cdot A_0 \overline{B_0} \right)$$

$$\text{für } A<B: \overline{A_1} \cdot B_1 + \left((A_1 \sim B_1) \cdot \overline{A_0} B_0 \right)$$

A_0	B_0	A_1	B_1	$A=B$	$A<B$	$A>B$
0	0	0	0	1	0	0
0	0	0	1	0	1	0
0	0	1	0	0	0	1
0	0	1	1	1	0	0
0	1	0	0	0	1	0
0	1	0	1	0	1	0
0	1	1	0	0	0	1
0	1	1	1	0	1	0
1	0	0	0	0	0	1
1	0	0	1	0	1	0
1	0	1	0	0	0	1
1	0	1	1	0	0	1
1	1	0	0	1	0	0
1	1	0	1	0	1	0
1	1	1	0	0	0	1
1	1	1	1	1	0	0

8. Funktion und Anwendungsbeispiele von digitalen Komparatoren

Gleichheitskomparator:

- Äquivalenzschaltung, d.h. eine Antivalenzschaltung mit anschließender Negation (exklusives NOR), realisiert die Ausgangsfunktion

$$\overline{y} = A_0 B_0 \vee \overline{A_0} \overline{B_0}$$

- Ausgang nimmt Signal 1 an, wenn beide Eingänge gleich sind ($A_0 = B_0$)

Komparator mit Größenvergleich:

- Größenvergleich zweier einstelliger Dualzahlen lässt sich mit einem einfachen UND-Gatter, dem das eine Eingangssignal negiert zugeführt wird, vornehmen

$$y = A_0 \overline{B_0}$$

- wenn $A_0 > B_0$ und somit $A_0 = 1$ und $B_0 = 0$, dann $y = 1$, ansonsten $y = 0$

Anwendungsbeispiele:

- programmierbare Adressdekodierer (in Mikroprozessorsystemen Vergleich mit auf Adressbus anliegendem Wort)
- in seriellen Rechenwerken und Mikroprozessorschaltkreisen zum Größenvergleich zweier mehrstelliger Zahlen