

13 Digital-Analog-Umsetzer, Analog-Digital-Umsetzer

13.1 Digital-Analog-Umsetzer

Digital-Analog-Umsetzer, auch DA-Wandler genannt, haben die Aufgabe, digitale Informationen in entsprechende analoge Informationen umzuwandeln.

13.1.1 Prinzip der Digital-Analog-Umsetzung

Betrachten wir eine Sinustabelle. Sie enthält die Sinusfunktionswerte, also die Informationen in digitaler Form. Nach der Sinustabelle kann eine Sinuskurve gezeichnet werden. Diese enthält die Informationen in analoger Form. Die Umwandlung der Tabelle in die Kurve ist eine Digital-Analog-Umsetzung.

In der digitalen Steuerungstechnik liegen die Informationen meist als binäre Informationen vor, die nach einem bestimmten Kode verschlüsselt sind. Für diesen Kode muß der Digital-Analog-Umsetzer geeignet sein.

Ein Digital-Analog-Umsetzer kann nur Signale eines bestimmten binären Kodens in analoge Signale wandeln.

Verschiedene binäre Kodens eignen sich nicht für eine Digital-Analog-Umsetzung. Es sind dies die *unbewerteten Kodens*. Unbewertet nennt man einen Kode, dessen Elementen keine bestimmten Zahlenwerte zugeordnet sind. Der Dualkode ist z.B. ein bewerteter Kode. Jedem Element, also jeder Stelle, ist eine Zweierpotenz zugeordnet. Ebenfalls ist der BCD-Kode ein bewerteter Kode. Der GRAY-Kode dagegen ist ein unbewerteter Kode. Seinen Elementen sind keine Zahlenwerte zugeordnet (s. Kapitel 8).

Unbewertete Kodens müssen vor einer Digital-Analog-Umsetzung in einen bewerteten Kode umgewandelt werden.

Die Umwandlung bereitet mit entsprechenden Kodewandlern keine Schwierigkeiten. Bei fehlererkennenden und fehlerkorrigierenden Kodens ergeben sich mit den Redundanzstellen Probleme. Wenn der Kode sonst ein bewerteter Kode ist (z.B. der Hamming-Kode), müssen die Redundanzstellen von der Digital-Analog-Umsetzung ausgeschlossen werden. Ist der Kode nicht bewertet, muß er vor der Digital-Analog-Umsetzung insgesamt in einen bewerteten Kode umgesetzt werden.

Das Prinzip der Digital-Analog-Umsetzung zeigt Bild 13.1. Mit 4-Bit-Einheiten lassen sich 16 Zahlenwerte bilden. Als analoges Signal ergibt sich eine Treppenspannung. Mit

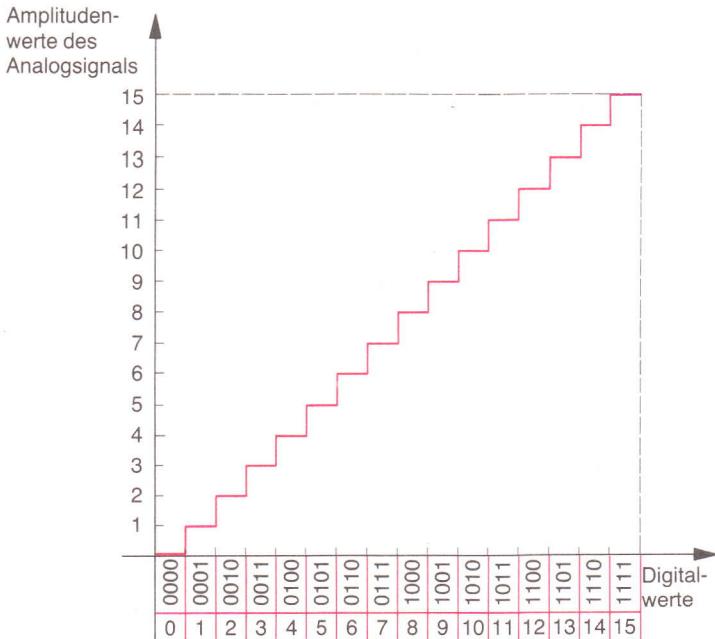


Bild 13.1 Prinzip der Digital-Analog-Umsetzung

Anzahl der Bits	Anzahl der Amplitudenwerte
4	16
5	32
6	64
7	128
8	256
9	512
10	1024
11	2048
12	4096
13	8192
14	16384
15	32768

Bild 13.2 Zusammenhang zwischen Bitzahl und Amplitudenwerten

4 Bit werden also 16 verschiedene Amplitudenwerte gebildet. Entsprechend sind mit 5 Bit 32 Amplitudenstufen möglich, mit 6 Bit 64 Amplitudenstufen usw. (s. Bild 13.2).

Das sich aus der Digital-Analog-Umsetzung ergebende Analogsignal ist ein gestuftes Signal mit einer bestimmten Anzahl von möglichen Amplitudenwerten.

Die Stufung kann beliebig fein gemacht werden. Sie wird um so feiner, je größer die Anzahl der Bit des digitalen Signals ist.

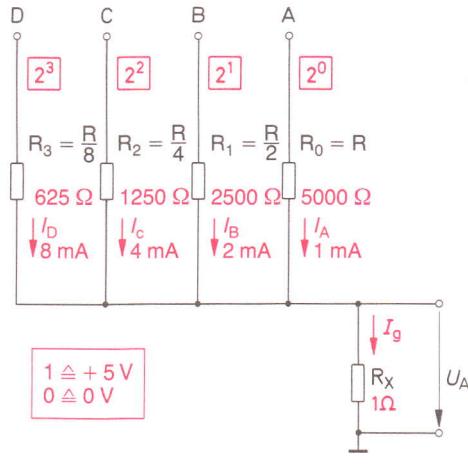
Die Stufen werden durch Siebglieder geglättet, so daß ein stetig verlaufendes Analogsignal entsteht.

Die Umwandlung digitaler Signale in analoge Signale ist mit verschiedenartigen Verfahren möglich. Die wichtigsten Verfahren sollen kurz vorgestellt werden.

13.1.2 DA-Umsetzer mit gestuften Widerständen

Die Prinzipschaltung eines DA-Umsetzers mit gestuften Widerständen ist in Bild 13.3 dargestellt. An die vier Eingänge A, B, C und D wird ein 4-Bit-Digitalsignal angelegt. Die Widerstände R_0 bis R_3 sind nach der Wertigkeit der Bit im Dualkode bemessen. Es gilt die Gleichung:

Bild 13.3 Prinzipschaltung eines DA-Umsetzers mit gestuften Widerständen für den Dualkode



Der Wert von R kann in Grenzen frei gewählt werden. Hier wurden 5000Ω gewählt. Für R_1 ergibt sich 2500Ω , für R_2 1250Ω . Jeder weitere Widerstand ist immer halb so groß wie der vorhergehende.

Die Tabelle Bild 13.3a zeigt die sich ergebenden Ströme, wenn das 1-Signal $+5 \text{ V}$ und das 0-Signal 0 V entspricht. Es ergibt sich eine Ausgangsspannung, die in 1-mV-Schritten gestuft ist. Sie hat stets soviel Millivolt, wie der Zahlenwert des 4-Bit-Dualsignals beträgt, ist also ein Analogsignal.

Bild 13.3a Tabelle der Teilströme I_A , I_b , I_C , I_D der Gesamtströme I_g und der Ausgangsspannungen U_A des DA-Umsetzers nach Bild 13.3 für duale Eingangssignale von 0000 bis 1111

Dezimalzahlenwert	D 2^3	C 2^2	B 2^1	A 2^0	$\frac{I_D}{\text{mA}}$	$\frac{I_C}{\text{mA}}$	$\frac{I_B}{\text{mA}}$	$\frac{I_A}{\text{mA}}$	$\frac{I_g}{\text{mA}}$	U_A mV
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1	1	1
2	0	0	1	0	0	0	2	0	2	2
3	0	0	1	1	0	0	2	1	3	3
4	0	1	0	0	0	4	0	0	4	4
5	0	1	0	1	0	4	0	1	5	5
6	0	1	1	0	0	4	2	0	6	6
7	0	1	1	1	0	4	2	1	7	7
8	1	0	0	0	8	0	0	0	8	8
9	1	0	0	1	8	0	0	1	9	9
10	1	0	1	0	8	0	2	0	10	10
11	1	0	1	1	8	0	2	1	11	11
12	1	1	0	0	8	4	0	0	12	12
13	1	1	0	1	8	4	0	1	13	13
14	1	1	1	0	8	4	2	0	14	14
15	1	1	1	1	8	4	2	1	15	15

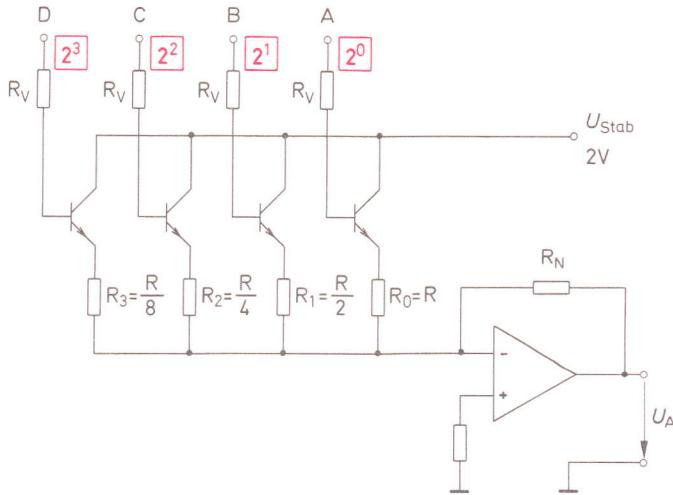


Bild 13.4 DA-Umsetzer mit Widerstandsnetzwerk für den Dualkode

Ein Digital-Analog-Umsetzer nach Bild 13.3 arbeitet nicht sehr genau. Die 1-Signale einer Digitalschaltung haben ja in den seltensten Fällen genau + 5 V. Abweichungen im Rahmen der zulässigen Toleranzen führen zu fehlerhaften Analogsignalen. Wesentlich genauer arbeitet die Schaltung Bild 13.4. Es wird eine stabilisierte Gleichspannung U_{Stab} verwendet. Diese wird über Transistor-Schalterstufen auf die Widerstände R_0 bis R_3 geschaltet. Der Operationsverstärker arbeitet als Summierverstärker. Die Widerstände R_V dienen der Basisstrombegrenzung. Die Transistoren arbeiten im inversen Betrieb. Bei dieser Betriebsart ist die Kollektor-Emitter-Sättigungsspannung U_{CESat} eines durchgeschalteten Transistors sehr klein (ca. 20 mV). An die Eingänge A bis D müssen 1-Signale angelegt werden, die größer als etwa 2,6 V sind.

DA-Umsetzer für andere bewertete Binärkodes sind entsprechend aufgebaut. Je nach der Wertigkeit der einzelnen Bit sind die Widerstände R_0 bis R_n zu bemessen.

13.1.3 R/2R-DA-Umsetzer

Ein DA-Wandler lässt sich mit Hilfe eines Kettenleiters aufbauen. Für einen solchen Kettenleiter sind nur zwei verschiedene Widerstandsgrößen erforderlich, eine Widerstandsgröße R und eine doppelt so große Widerstandsgröße $2R$. Diese Widerstandsgrößen geben dem DA-Wandler den Namen. Die Schaltung eines R/2R-DA-Umsetzers zeigt Bild 13.5.

Die Arbeitsweise der Schaltung Bild 13.5 ist schwierig zu übersehen. Zur Erläuterung der Arbeitsweise soll daher die nur mit zwei Schaltern arbeitende Schaltung Bild 13.6 dienen. Die Widerstandswerte betragen 1 kΩ und 2 kΩ, die Versorgungsspannung U_{Stab} beträgt 12 V.

Die Schaltung wandelt 2-Bit-Dualsignale in Analogsignale um. Bei $S_0 = 0$ ist der Schalter S_0 mit Masse verbunden. Bei $S_0 = 1$ ist der Schalter S_0 mit U_{Stab} verbunden. Entsprechendes gilt für den Schalter S_1 .

Bild 13.5 Prinzipschaltung eines R/2R-DA-Umsetzers

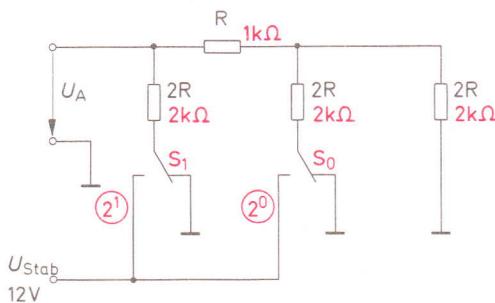
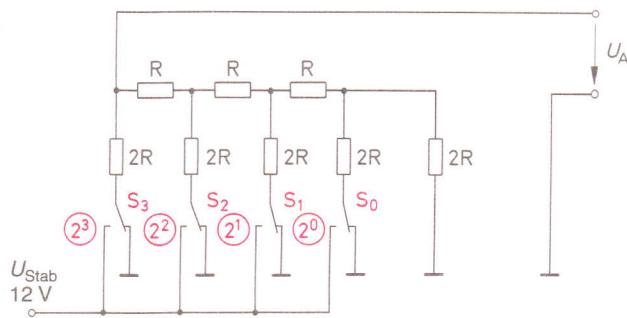


Bild 13.6 R/2R-DA-Umsetzer für 2-Bit-Dualsignale

Wenn beide Schalter geöffnet sind (Masseanschluß), liegt am Ausgang die Spannung $U_A = 0 \text{ V}$. Ist nur der Schalter S_0 geschlossen, ergibt sich die Schaltung Bild 13.8 mit einem Gesamtwiderstand von $3,2 \text{ k}\Omega$ und einem Gesamtstrom $I_g = 3,75 \text{ mA}$. Zwischen Punkt P und Masse liegt ein Widerstand von $1,2 \text{ k}\Omega$, an dem eine Spannung von $4,5 \text{ V}$ abfällt. Diese wird durch R_3 und R_4 aufgeteilt. An R_4 liegt eine Spannung von 3 V .

Ist nur der Schalter S_1 geschlossen, wird die Spannung U_{Stab} halbiert. Sind beide Schalter geschlossen, ergibt sich eine Ausgangsspannung von 9 V (Bild 13.7). Die Stufung der Spannung U_A beträgt also 3 V .

Die Stufung ΔU_A ergibt sich aus der Anzahl der Schalter bzw. aus der Anzahl der Bit. Sie wird mit der Gleichung

$$\Delta U_A = \frac{U_{\text{Stab}}}{2^n}$$

berechnet. Für die Schaltung Bild 13.5 beträgt sie

$$\Delta U_A = \frac{U_{\text{Stab}}}{2^n} = \frac{12 \text{ V}}{2^4} = \frac{12 \text{ V}}{16} = 0,75 \text{ V}$$

Dezimalwert	S_1 (2^1)	S_0 (2^0)	U_A
0	0	0	0V
1	0	1	3V
2	1	0	6V
3	1	1	9V

Bild 13.7 Tabelle zur Schaltung Bild 13.6

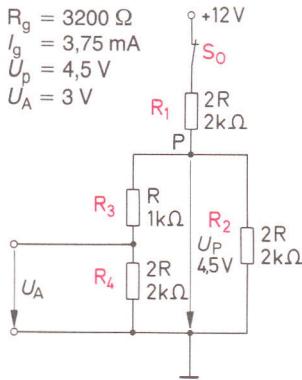


Bild 13.8 Widerstandsschaltung

Die Schalter S_0 bis S_n werden in der Praxis durch Transistor-Schalterstufen ersetzt. Das $R/2R$ -Verfahren eignet sich besonders gut für integrierte Schaltungen. Es müssen nur zwei verschiedene Widerstandswerte hergestellt werden.

DA-Umsetzer werden überwiegend als integrierte Schaltungen hergestellt. Wegen des geringen Leistungsbedarfs werden DA-Umsetzer in CMOS-Technik besonders häufig verwendet.

13.2 Analog-Digital-Umsetzer

Analog-Digital-Umsetzer, auch AD-Wandler genannt, wandeln analoge Signale in entsprechende digitale Signale um.

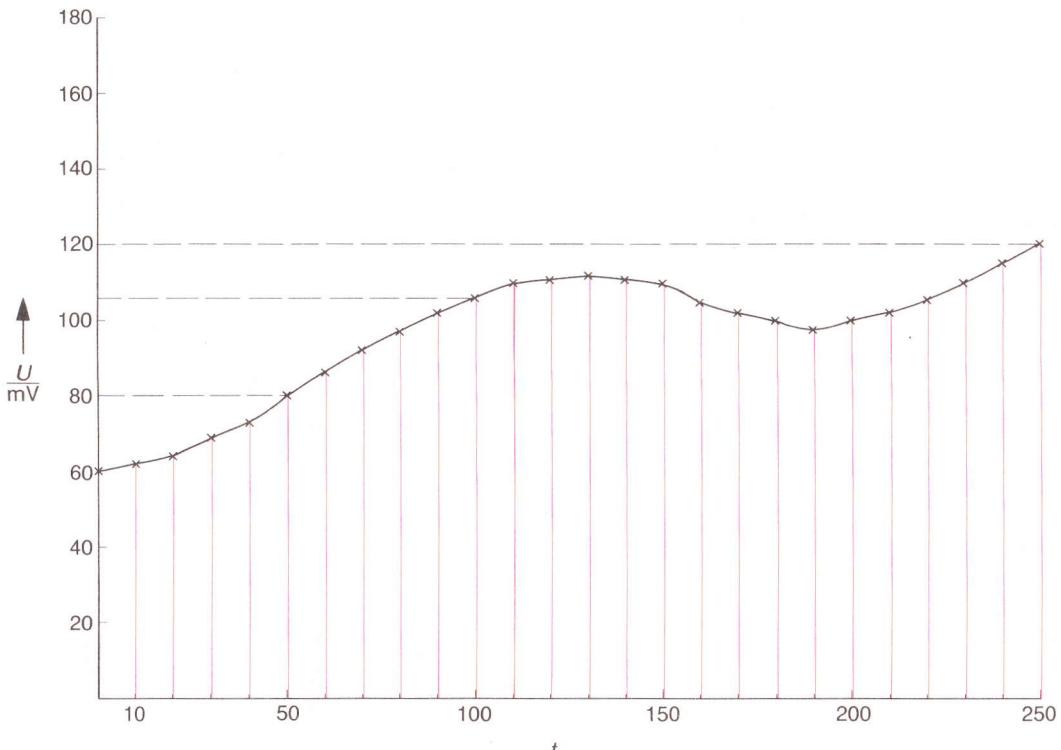
13.2.1 Prinzip der Analog-Digital-Umsetzung

Ein analoges Signal, z.B. ein Signal nach Bild 13.9, kann durch eine bestimmte Anzahl von Amplitudenwerten dargestellt werden. Die Amplitude des Signals wird z.B. alle $10 \mu\text{s}$ gemessen. Die gemessenen Zahlenwerte werden nacheinander in der richtigen Reihenfolge gespeichert. Alle Zahlenwerte zusammen bilden das digitale Signal.

Das digitale Signal einer zeitlich sich ändernden Größe besteht aus einer Anzahl von Zahlenwerten.

Die Zahlenwerte können in einem beliebigen Zahlensystem oder Kode dargestellt werden. In Bild 13.9 sind sie dezimal und dual dargestellt.

Analog-Digital-Umsetzer geben die Zahlenwerte meist im dualen Zahlensystem oder im BCD-Kode aus.



t μs	U mV	dezimal	dual
0	60	0 1 1 1 1	
10	62	0 1 1 1 1	
20	64	1 0 0 0 0	
30	69	1 0 0 0 1	
40	73	1 0 0 1 0	
50	80	1 0 1 0 0	
60	86	1 0 1 0 1	
70	92	1 0 1 1 1	
80	97	1 1 0 0 0	
90	102	1 1 0 0 1	
100	106	1 1 0 1 0	
110	110	1 1 0 1 1	
120	111	1 1 0 1 1	
130	112	1 1 1 0 0	
140	111	1 1 1 0 1	
150	110	1 1 0 1 1	
160	105	1 1 0 1 0	
170	102	1 1 0 0 1	
180	100	1 1 0 0 1	
usw.	:		

Die Amplitudenwerte werden in einem bestimmten Maßstab dargestellt, z.B. in Millivolt. Sollen Spannungswerte bis 4 V auf 1 mV genau gewandelt werden, sind 4000 Amplitudenstufen erforderlich. Zur Darstellung dieser 4000 Amplitudenstufen werden 12stellige Dualzahlen benötigt. Jeder Amplitudenwert wird dann durch 12 Bit dargestellt. Die Feinheit der Zahlendarstellung hängt von der Anzahl der Bit ab. Sie wird Auflösungsvermögen genannt.

Ein AD-Umsetzer hat ein um so größeres Auflösungsvermögen, je mehr Bit für die Darstellung der Zahlenwerte zur Verfügung stehen.

Das Auflösungsvermögen darf nicht mit der Genauigkeit des AD-Umsetzers verwechselt werden. Die Genauigkeit hängt von der Richtigkeit der ausgegebenen Zahlenwerte ab. Auch fein unterteilte Zahlenwerte, also Zahlenwerte mit vielen Bit, können ungenau sein.

Jeder Analog-Digital-Umsetzer arbeitet mit einer bestimmten Genauigkeit.

Die Genauigkeit gibt an, um welchen Bruchteil des richtigen Wertes das Umsetzungsergebnis höchstens nach oben und unten abweichen darf. Bei einer Genauigkeit von 10^{-3} dürfen die Ergebnisse um $1/1000$ größer oder kleiner als der richtige Wert sein, also um $\pm 1\%$ abweichen. Bei großem Aufwand sind zur Zeit Genauigkeiten von 10^{-5} erreichbar.

Ein zeitlich sich änderndes Analogsignal muß mit einer bestimmten Häufigkeit abgetastet werden. Das heißt, die Amplitudenwerte müssen z.B. jede μs oder alle 10 μs oder jede ms gemessen und gespeichert werden. Die Häufigkeit der Feststellung der Amplitudenwerte muß um so größer sein, je schneller sich das Analogsignal ändert. Allgemein gilt:

Die Abtasthäufigkeit eines Analogsignals muß mindestens doppelt so groß sein wie die höchste zu wandelnde Frequenz, die im Analogsignal enthalten ist.

Soll ein analoges Tonfrequenzsignal mit einer Bandbreite von 50 Hz bis 20 kHz in ein entsprechendes digitales Signal umgewandelt werden, sind mindestens 40 000 Abtastvorgänge je Sekunde erforderlich. Die sogenannte Abtastrate beträgt dann 40 kHz. Sie darf sehr wohl größer, aber nicht kleiner sein. Ist sie dennoch kleiner, wird das Frequenzband beschnitten.

Analog-Digital-Umsetzer werden überwiegend als integrierte Schaltungen hergestellt. Ein Aufbau mit diskreten Bauelementen wäre sehr aufwendig. Bei integrierten Schaltungen überwiegt z.Z. die CMOS-Technik. Schaltungen in TTL-Technik und ECL-

Technik werden wegen des verhältnismäßig großen Leistungsbedarfs nur dort eingesetzt, wo besondere Schnelligkeit erforderlich ist.

AD-Umsetzer unterscheiden sich im wesentlichen durch folgende Eigenschaften:

Auflösungsvermögen	(Anzahl der Bit)
Genauigkeit	(Fehler in % des Ergebnisses oder in % des Höchstwerts)
Schnelligkeit	(Dauer eines Umsetzervorgangs, Anzahl der höchstmöglichen Umsetzervorgänge je Zeiteinheit)
Spannungsbereich	(Bereich von der kleinsten bis zur größten umsetzbaren Spannung)

Eine Vielzahl verschiedener Umsetzerverfahren und Schaltungen ist gebräuchlich. Sie sollen im folgenden erläutert werden.

13.2.2 AD-Umsetzer nach dem Sägezahnverfahren

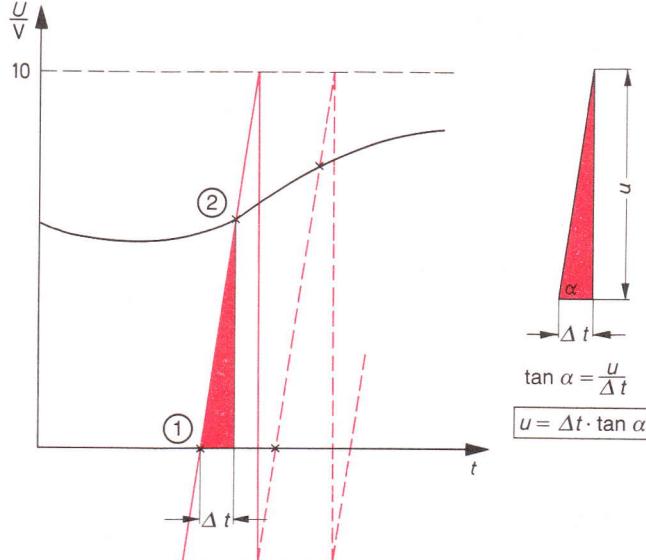
Ein AD-Umsetzer nach dem Sägezahnverfahren tastet das Analogsignal mit Sägezahnspannungen ab. Die Sägezahnspannungsflanke beginnt im negativen Spannungsbereich (Bild 13.10). Zum Zeitpunkt ①, wenn die Sägezahnspannung die Null-Linie überschreitet, wird ein Dualzähler gestartet. Dieser zählt Impulse eines Generators. Hat die Sägezahnspannung die Spannung des Analogsignals erreicht (Punkt ②), wird der Zähler gestoppt.

Der Zähler hat also während der Zeit Δt gezählt. Er hat genaugenommen die Zeit gemessen. Da die Sägezahnflanke aber in einem genau festgelegten Winkel α ansteigt, kann aus Δt und α die Spannungsamplitude u errechnet werden:

$$\tan \alpha = \frac{u}{\Delta t}$$

$$u = \Delta t \cdot \tan \alpha$$

Bild 13.10 Abtastung des Analogsystems mit Sägezahnspannung



Die Dualzahl, die der Zähler anzeigt, ist also ein Maß für den Spannungswert u . Die Frequenz des Impulsgenerators lässt sich nun so wählen, daß der Zähler die Spannung in der gewünschten Einheit, z.B. in Millivolt, angibt.

Durch den Höchstwert der Sägezahnspannung ist die größte abtastbare Spannung des Analogsignals gegeben. Steigt die Sägezahnspannung z.B. bis 10 V an (Bild 13.10), können nur Analogspannungen bis maximal 10 V gewandelt werden.

Bild 13.11 zeigt den prinzipiellen Schaltungsaufbau eines AD-Umsetzers nach dem Sägezahnverfahren. Die Sägezahnspannung wird auf zwei Komparatorschaltungen gegeben. Am Ausgang einer Komparatorschaltung liegt nur dann das binäre Signal 1, wenn beide Eingangsspannungen gleich groß sind (analoger Komparator bzw. Komparator mit Analogeingängen).

Hat die Sägezahnspannung einen negativen Wert, liegt am Ausgang des Komparators 2 0-Signal ($K_2 = 0$).

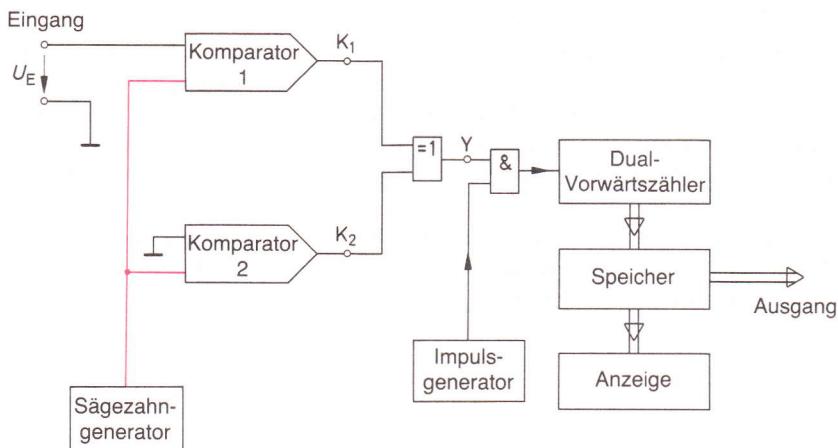


Bild 13.11 Schaltungsaufbau eines AD-Umsetzers nach dem Sägezahnverfahren

Überschreitet die Sägezahnspannung die Null-Linie (Punkt ①), wird $K_2 = 1$, denn beide Eingangsspannungen sind jetzt gleich. Am Ausgang Y des EXKLUSIV-ODER-Gliedes erscheint jetzt 1-Signal. Das UND-Glied gibt die Generatorimpulse zum Zählen frei. Erreicht die Sägezahnspannung die Analogspannung, gibt der Komparator 1 ebenfalls 1-Signal ($K_1 = 1$). Jetzt wird $Y = 0$. Das UND-Glied sperrt die Generatorimpulse. Das Digitalsignal des Zählers wird in einen Speicher gegeben und weiter verarbeitet. Es kann angezeigt werden. Eine Anzeige erfolgt z.B. bei einem digitalen Spannungsmesser.

Die Genauigkeit der Schaltung hängt sehr wesentlich von der Linearität der Sägezahnanstiegsflanke ab.

Der AD-Umsetzer nach Bild 13.11 ist nur für positive Meßspannungen geeignet. Man kann eine zu wandelnde Wechselspannung durch Addition einer bekannten Gleich-

spannung so anheben, daß alle Spannungswerte in den positiven Spannungsbereich fallen. Der Wert der bekannten Gleichspannung wird nach der AD-Umsetzung vom digitalen Ergebnis abgezogen.

Die Schaltung Bild 13.11 kann aber auch erweitert werden, so daß sie für positive und negative Meßspannungen geeignet ist.

13.2.3 AD-Umsetzer nach dem Dual-Slope-Verfahren

Das Dual-Slope-Verfahren arbeitet mit zwei Schritten. Es wird daher auch Zweischrittverfahren genannt. Dual-Slope ist ein englischer Ausdruck. Er bedeutet «zwei unterschiedlich steigende Flanken». Kernstück des AD-Umsetzers ist ein integrierter Verstärker, ein sogenannter Integrator (Bild 13.12). Die Ausgangsspannung des Integrators hat zwei unterschiedlich steigende Flanken.

Im 1. Schritt wird die positive Analogspannung während einer fest vorgegebenen Zeit t_1 integriert. Der Kondensator C des Integrators wird aufgeladen.

Im 2. Schritt wird eine negative Festspannung (Referenzspannung) an den Eingang des Integrators gelegt. Der Kondensator C wird entladen, bis die Ausgangsspannung des Integrators 0 ist. Die Zeit, die vom Anlegen der Referenzspannung bis zum Nullwerden der Integrator-Ausgangsspannung vergeht, ist ein Maß für die Größe der Analogspannung. Diese Zeit wird t_2 genannt.

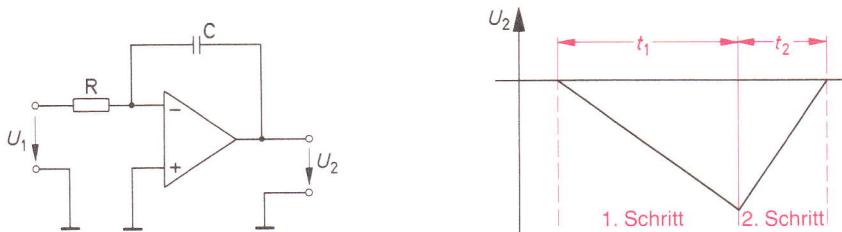


Bild 13.12 Integrator mit Angabe des Spannungsverlaufs während des 1. und 2. Schrittes

Während der Zeit t_1 läuft ein Vorwärtzähler. Er erhält die Zählimpulse von einem Impulsgenerator. Nach Ablauf der Zeit t_1 zeigt der Zähler eine binäre Größe an. Diese ist der digital ausgedrückte Amplitudenwert der Analogspannung. Wird die Analogspannung in den Dualkode gewandelt, ist der Zähler ein Dual-Vorwärtzähler. Der Zähler muß stets in dem Kode zählen, in den umgesetzt werden soll.

Wegen der zwei Arbeitsschritte des Integrators wird das Dual-Slope-Verfahren auch *Doppelintegrationsverfahren* genannt.

Der Schaltungsaufbau eines AD-Umsetzers nach dem Dual-Slope-Verfahren ist in Bild 13.13 dargestellt. Eine Steuerschaltung betätigt einen elektronischen Umschalter S. Zu Beginn des Umsetzungsvorganges, also zu Beginn der Zeit t_1 , wird der Umschalter in

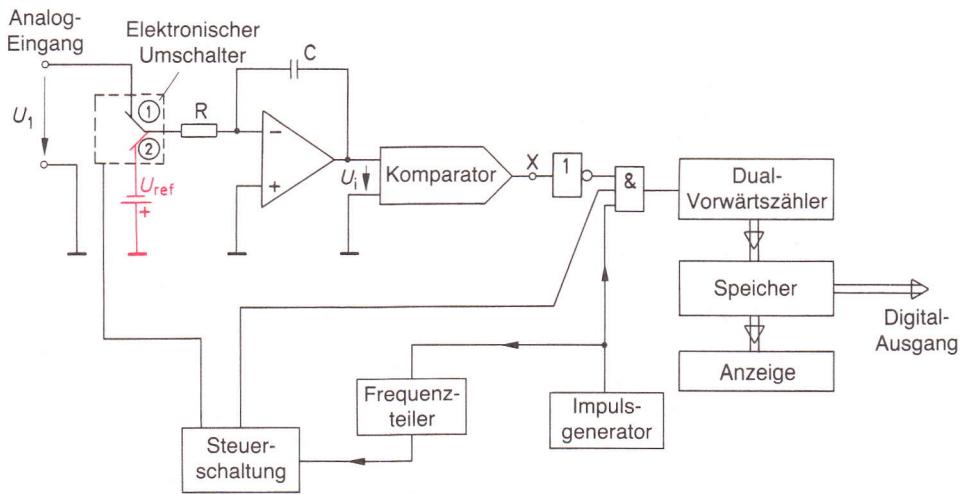


Bild 13.13 Schaltungsaufbau eines AD-Umsetzers nach dem Dual-Slope-Verfahren

Stellung 1 gebracht. In dieser Stellung verbleibt er während der Zeit t_1 . Die Zeit t_1 wird aus dem Impulsgenerator über einen Frequenzteiler gewonnen. Sie ist eine Festzeit – z.B. 100 µs. Während der Zeit t_1 wird C über R geladen.

Nach Ablauf der Zeit t_1 wird der elektronische Umschalter auf Stellung 2 geschaltet. Die Referenzspannung wird an den Eingang des Integrators gelegt. Gleichzeitig werden die Impulse des Impulsgenerators über das UND-Glied freigegeben. Sie werden vom Zähler gezählt. Die Zeit t_2 läuft. Aus dem Integrator wird Strom gezogen, C wird entladen. Während der Zeit t_2 sind die beiden Eingangsspannungen des Komparators ungleich. An seinem Ausgang X liegt 0-Signal. Dieses wird negiert und erscheint als 1-Signal am Eingang des UND-Gliedes.

Wenn die Ausgangsspannung U_2 des Integrators Null wird, ist die Zeit t_2 abgelaufen. Die beiden Eingangsspannungen des Komparators sind jetzt 0 V. Am Ausgang X erscheint 1-Signal. Dieses wird negiert in ein 0-Signal, das das UND-Glied sperrt. Die Impulszählung ist beendet. Das digitale Ergebnis liegt vor.

AD-Umsetzer nach dem Dual-Slope-Verfahren arbeiten langsamer als AD-Umsetzer nach dem Sägezahnverfahren. Sie erreichen jedoch bei gleichem Aufwand eine höhere Genauigkeit. Die Werte von R und C beeinflussen das digitale Ergebnis nicht. Dieses ist vor allem vom Verhältnis t_1/t_2 und von der Referenzspannung U_{ref} abhängig:

$$\frac{U_1}{U_{\text{ref}}} = \frac{t_2}{t_1}$$

Da die Zeiten t_1 und t_2 gleichermaßen von der Frequenz des Impulsgenerators abhängig sind, beeinflusst auch diese Frequenz das Ergebnis nicht. Sie muß allerdings während der Zeit $t_1 + t_2$ konstant sein.

13.2.4 AD-Umsetzer nach dem Kompensationsverfahren

Beim AD-Umsetzer nach dem Kompensationsverfahren wird die Analogspannung U_1 mit einer Kompensationsspannung U_K verglichen. Die Kompensationsspannung ergibt sich als Ausgangsspannung eines Digital-Analog-Umsetzers und wird daher auch Umsetzerspannung genannt. Dem DA-Umsetzer werden solange digital-kodierte Zahlen eingegeben, bis die Umsetzerspannung U_K die Spannung U_1 erreicht. Den prinzipiellen Schaltungsaufbau zeigt Bild 13.14.

Zu Beginn eines Umsetzungsvorganges wird der Zähler durch die Steuerschaltung auf Null gesetzt. Die Spannung U_K ist Null. Am Ausgang des Komparators liegt $X = 0$. Dieses Signal wird negiert auf den Eingang eines UND-Gliedes gegeben. Das UND-Glied lässt die Impulse des Impulsgenerators zum Zähler durch. Der Zähler zählt von 0 an vorwärts.

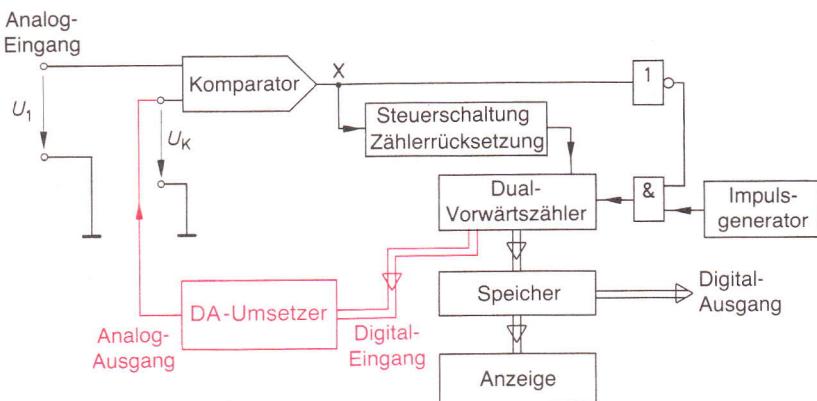


Bild 13.14 Schaltungsaufbau eines AD-Umsetzers nach dem Kompensationsverfahren

Das Digitalsignal des Zählers wird durch den DA-Umsetzer in ein entsprechendes Analogsignal gewandelt. Durch die Vorwärtzählung entsteht ein treppenförmig ansteigendes Ausgangssignal U_K . Erreicht die Spannung U_K die Spannung U_1 (Bild 13.15), so erscheint am Ausgang des Komparators 1-Signal. Dieses Signal wird einmal negiert. Das 0-Signal sperrt das UND-Glied. Weitere Impulse gelangen nicht zum Zähler. Der Zähler stoppt.

Das Signal $X = 1$ wird zum anderen der Steuerschaltung zugeführt. Diese veranlaßt die Eingabe des Zählerstandes in den Speicher. Der Zählerstand ist das Ergebnis der Analog-Digital-Umsetzung. Das Ergebnis steht zur Weiterverarbeitung zur Verfügung. Es kann angezeigt werden.

Die Steuerschaltung stellt den Zähler auf Null. Der nächste Umsetzervorgang beginnt. AD-Umsetzer nach diesem Verfahren sind recht langsam, da jeder Umsetzervorgang mit dem Zählen von Null ab beginnt.

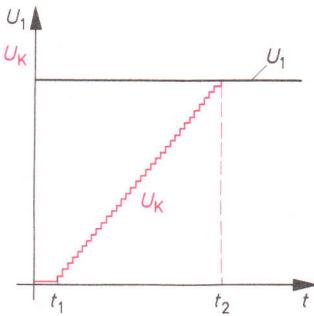


Bild 13.15 Verlauf der Spannung U_K während der Zeit eines Umsetzungsvorganges (t_1 = Beginn der Umsetzung, t_2 = Ende der Umsetzung, U_1 = Analogspannung)

Die Zahl der möglichen Umsetzungsvorgänge je Zeiteinheit kann durch Einsatz eines Vorwärts-Rückwärts-Zählers wesentlich erhöht werden. Man verwendet eine Schaltung nach Bild 13.16. Der Komparator ist durch einen Differenzverstärker ersetzt worden. Der Differenzverstärker liefert eine positive Ausgangsspannung, solange U_1 kleiner ist als U_K . Die positive Ausgangsspannung veranlaßt den Zählrichtungsumschalter, den Zähler auf «Vorwärtzählen» zu schalten. Außerdem beeinflußt die positive Ausgangsspannung den spannungsgesteuerten Oszillator, schneller zu schwingen. Die Impulse kommen schneller. Der Zähler zählt schneller. Die Spannung U_K steigt schneller an. Je mehr sich die Spannung U_K der Spannung U_1 nähert, desto geringer wird die positive Ausgangsspannung des Differenzverstärkers, desto langsamer schwingt der Oszillator, desto langsamer zählt der Zähler. Die Spannung U_K nähert sich langsam dem Wert der Spannung U_1 . Bei $U_K = U_1$ wird der Zähler durch eine Steuerschaltung stillgesetzt. Das

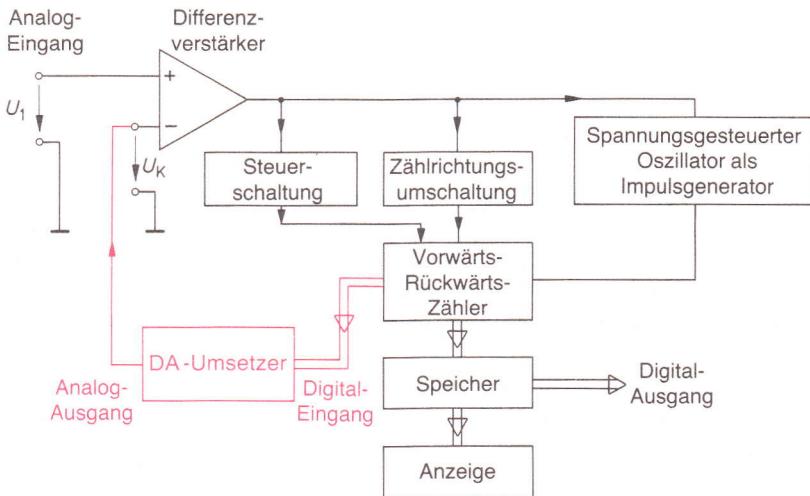


Bild 13.16 Schaltungsaufbau eines AD-Umsetzers nach dem Kompensationsverfahren mit kontinuierlichem Abgleich

Zählergebnis wird in den Speicher übernommen und steht als Wandlungsergebnis zur Weiterverarbeitung zur Verfügung.

Ändert sich die Analogspannung U_1 , so «läuft der Zähler nach». Wird U_1 z.B. größer als U_K , erscheint am Ausgang des Differenzverstärkers eine positive Spannung. Der Zähler zählt weiter in Vorwärtsrichtung, bis $U_K = U_1$ ist.

Wird U_1 kleiner als U_K , so erscheint am Ausgang des Differenzverstärkers eine negative Spannung. Diese veranlaßt den Zählrichtungsumschalter, den Zähler auf «Rückwärtszählen» zu schalten. Der Zähler zählt rückwärts, U_K wird kleiner, bis U_K gleich U_1 ist. Dann wird der Zähler wieder stillgesetzt und der Zählerstand gespeichert.

Der Zählerstand bleibt stets in der Nähe des Wertes von U_1 . Bei Änderung von U_1 sind verhältnismäßig wenige Zählschritte in Vorwärts- oder Rückwärtsrichtung erforderlich. Die Analog-Digital-Wandlung erfolgt daher sehr schnell.

Der spannungsgesteuerte Oszillator spricht auf Spannungsbeträge an, also auf Spannungen ohne Berücksichtigung des Vorzeichens. Er schwingt bei großen positiven oder negativen Spannungen schnell und bei kleinen positiven oder negativen Spannungen langsam.

13.2.5 AD-Umsetzer nach dem Spannungs-Frequenz-Verfahren

Der zu wandelnde Analog-Spannungswert wird in einem Spannungs-Frequenz-Umsetzer in eine Wechselspannung bestimmter Frequenz umgeformt. Hierfür gibt es verschiedene mögliche Schaltungen. Eine einfache Schaltung wäre ein LC-Oszillator, zu dessen Schwingkreiskondensator eine Kapazitätsdiode parallelgeschaltet ist. Die Kapazität der Kapazitätsdiode wird durch den Analog-Spannungswert gesteuert. Dadurch wird die Resonanzfrequenz des Schwingkreises und die Ausgangsfrequenz des LC-Oszillators geändert (s. «Elektronik 3»). Eine solche Schaltung wäre im Prinzip brauchbar, sie ist nur etwas zu ungenau.

Ein Spannungs-Frequenz-Umsetzer muß einen sehr linearen Zusammenhang zwischen Spannung und Frequenz haben. Nimmt der Spannungswert z.B. um 10% zu, muß auch die Frequenz um 10% zunehmen. Doppelte Spannung muß zu doppelter Frequenz führen. Die Frequenz wird digital gemessen. Der zu der gemessenen Frequenz gehörende Spannungswert wird als Digitalsignal gespeichert. Er kann angezeigt werden. Das Digitalsignal steht zur Weiterverarbeitung zur Verfügung.

In Bild 13.17 ist der prinzipielle Schaltungsaufbau eines AD-Umsetzers nach dem Spannungs-Frequenz-Verfahren angegeben. Eine Steuerelektronik startet den Wandlungsvorgang. Der Zeitgeber legt 1-Signal an das UND-Glied. Vom Spannungs-Frequenz-Umsetzer kommen Impulse, die aus der Frequenz abgeleitet sind, z.B. die positiven Halbwellen der Schwingung. Sie durchlaufen das UND-Glied und werden vom Zähler gezählt.

Nach Ablauf einer fest vorgegebenen Zeit stoppt der Zeitgeber den Zählvorgang. Er legt an das UND-Glied 0-Signal. Die Steuerelektronik veranlaßt, daß das Zählergebnis in den Speicher übernommen wird. Danach kann ein weiterer Wandlungsvorgang beginnen.

Die Genauigkeit des Wandlungsergebnisses ist vor allem von der Linearität des Spannungs-Frequenz-Umsetzers und von der Genauigkeit des Zeitgebers abhängig.

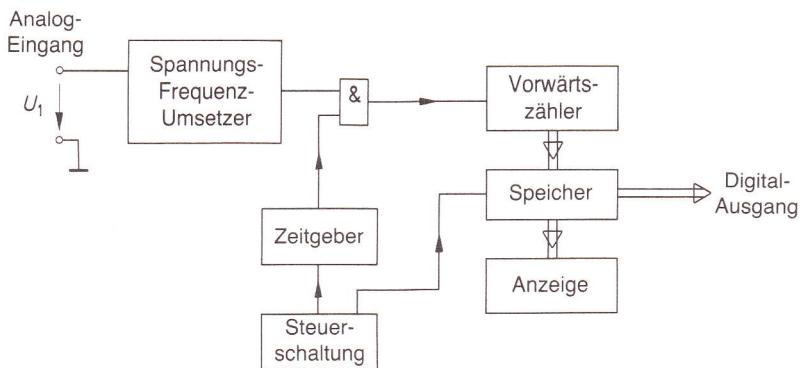


Bild 13.17 Schaltungsaufbau eines AD-Umsetzers nach dem Spannungs-Frequenz-Verfahren

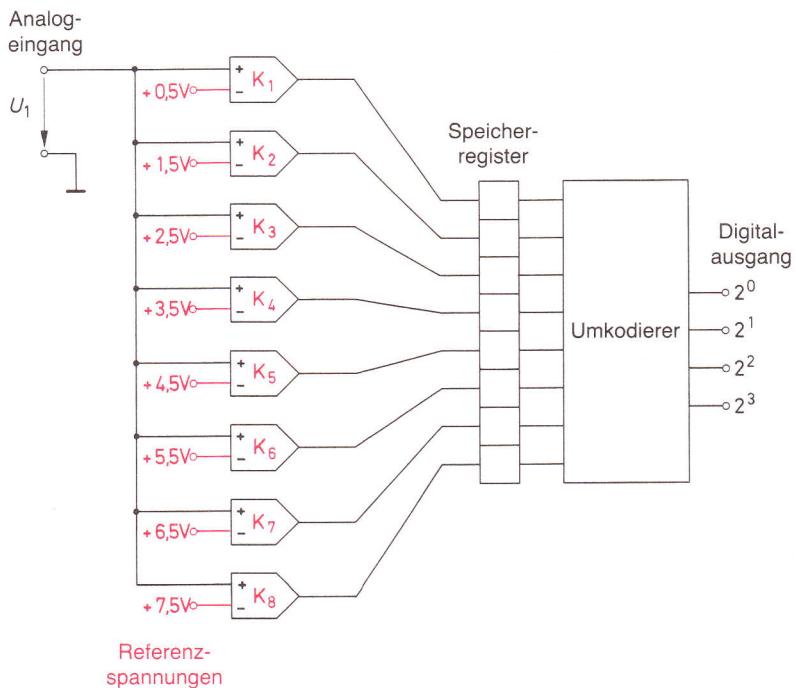


Bild 13.18 Schaltungsaufbau eines AD-Umsetzers nach dem Direktverfahren

13.2.6 AD-Umsetzer nach dem Direktverfahren

Ein AD-Umsetzer nach dem Direktverfahren arbeitet mit Komparatoren, die stets dann vom Ausgangszustand 0 in den Ausgangszustand 1 umschalten, wenn die am Plus-Eingang liegende Spannung gleich oder größer als die am Minus-Eingang liegende Spannung ist.

Je Ausgangs-Bit wird ein Komparator verwendet. Die Schaltung Bild 13.18 arbeitet mit 8 Komparatoren. Jeder Komparator erhält eine feste Referenzspannung, die an seinen Minus-Eingang gelegt wird. Die Referenzspannungen sind gestuft (z.B. 0,5 V, 1,5 V, 2,5 V, 3,5 V, 4,5 V, 5,5 V, 6,5 V, 7,5 V).

Steigt nun die zu wandelnde Spannung U_1 von Null aus an, so schaltet bei Erreichen der kleinsten Referenzspannung (z.B. 0,5 V) der Komparator K_1 das Signal 1 an seinen Ausgang. Bei Erreichen der nächsthöheren Referenzspannung (z.B. 1,5 V) schaltet zusätzlich der Komparator K_2 Signal 1 an seinen Ausgang – und so fort. Es ergeben sich die in der Arbeitstabelle Bild 13.19 aufgeführten Ausgangssignale. Mit 8 Komparatoren sind 8 Spannungsstufen möglich. Der Ausgangskode des AD-Umsetzers nach dem Direktverfahren wird zweckmäßigerverweise in einen anderen Kode, z.B. in den Dualkode, gewandelt.

Bild 13.19 Arbeitstabelle des AD-Umsetzers nach Bild 13.18

$\frac{U_1}{V}$	K_8	K_7	K_6	K_5	K_4	K_3	K_2	K_1	
0	0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	1	
2	0	0	0	0	0	0	1	1	
3	0	0	0	0	0	1	1	1	
4	0	0	0	0	1	1	1	1	
5	0	0	0	1	1	1	1	1	
6	0	0	1	1	1	1	1	1	
7	0	1	1	1	1	1	1	1	
8	1	1	1	1	1	1	1	1	
Schaltspannungen		7,5 V	6,5 V	5,5 V	4,5 V	3,5 V	2,5 V	1,5 V	0,5 V

AD-Umsetzer nach dem Direktverfahren arbeiten schneller als alle anderen AD-Umsetzer. Die Zeit für einen Umsetzungsvorgang wird bestimmt durch die Schaltzeit der Komparatoren. Diese liegt bei etwa 40 bis 50 ns. Das bedeutet, daß bei 100 ns je Umsetzungsvorgang in jeder Sekunde 10 Millionen Umsetzungsvorgänge möglich sind. Der Schaltungsaufwand ist allerdings sehr groß. Für einen AD-Umsetzer mit 128 Spannungsstufen sind 128 Komparatoren erforderlich. 256 Spannungsstufen hat ein 8-Bit-AD-Umsetzer. Für einen 10-Bit-AD-Umsetzer mit 1024 Spannungsstufen wären 1024 Komparatoren erforderlich.

Die hohe mögliche Packungsdichte integrierter Schaltungen erlaubt den Aufbau von AD-Umsetzern mit guter Auflösung nach dem Direktverfahren. Die Genauigkeit solcher Wandler hängt von der Genauigkeit der Referenzspannungen und von den Schalttoleranzen der Komparatoren ab.

13.3 Lernziel-Test

1. Wie arbeitet ein Digital-Analog-Umsetzer mit gestuften Widerständen?
2. Digitale Signale, die in einem unbewerteten Kode dargestellt sind, sollen in Analogsignale umgewandelt werden. Was ist zu tun?
3. Was versteht man unter dem Auflösungsvermögen eines AD-Umsetzers?
4. Welche wichtigen Eigenschaften sind für die Auswahl eines AD-Umsetzers von Bedeutung?
5. Zählen Sie die verschiedenen gebräuchlichen Umsetzerverfahren für die Analog-Digital-Umsetzung auf.
6. Erklären Sie die Arbeitsweise eines AD-Umsetzers nach dem Sägezahn-Verfahren.
7. Welche Vor- und Nachteile hat ein AD-Umsetzer nach dem Direktverfahren?

14 Rechenschaltungen

Mit Digitalschaltungen können Rechenvorgänge durchgeführt werden, z.B. Additionen und Subtraktionen. Man nennt derartige Schaltungen Rechenschaltungen.

Rechenschaltungen erzeugen zwischen ihren Eingangsvariablen logische Verknüpfungen, die einem Rechenvorgang entsprechen.

Die Eingangszahlen müssen in einem bestimmten binären Kode kodiert sein. Im gleichen Kode werden die Ergebniszahlen ausgegeben.

Jede Rechenschaltung ist nur für einen Kode oder ein entsprechendes Zahlensystem geeignet.

Häufig werden der Dualkode, also das duale Zahlensystem, und der BCD-Kode verwendet (s. Kapitel 8).

14.1 Halbaddierer

Die einfachste Rechenschaltung ist der Halbaddierer.

Ein Halbaddierer kann zwei Dualziffern addieren.

Es gelten folgende Regeln:

$$\begin{array}{rcl} 0 + 0 & = & 0 \\ 0 + 1 & = & 1 \\ 1 + 0 & = & 1 \\ 1 + 1 & = & 10 \end{array}$$

Die eine zu addierende Ziffer erhält den Variablennamen A. Die andere zu addierende Ziffer erhält den Variablennamen B. Die Schaltung muß zwei Ausgänge haben. Der Ausgang mit der Wertigkeit 2^0 soll Z heißen, der Ausgang mit der Wertigkeit 2^1 erhält den Namen Ü (Übertrag). Wird die Ziffer 0 dem binären Zustand 0 und die Ziffer 1 dem binären Zustand 1 zugeordnet, ergibt sich die Wahrheitstabelle nach Bild 14.1.

Aus der Wahrheitstabelle können über die ODER-Normalformen sehr leicht die Verknüpfungsgleichungen des Halbaddierers gefunden werden. Die Vollkonjunktionen sind in Bild 14.1 rot eingetragen. Der Halbaddierer wird durch das Summenzeichen Σ gekennzeichnet (DIN 40900 Teil 12).

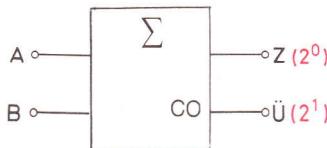


Bild 14.1 Halbaddierer mit Wahrheitstabelle CO kennzeichnet Übertragsausgang (engl.: Carry out output). Σ ist das Summenzeichen

Fall	B	A	\bar{U}	Z
1	0	0	0	0
2	0	1	0	1
3	1	0	0	1
4	1	1	1	0

$\rightarrow A \wedge \bar{B}$
 $\rightarrow \bar{A} \wedge B$
 $\rightarrow A \wedge B$

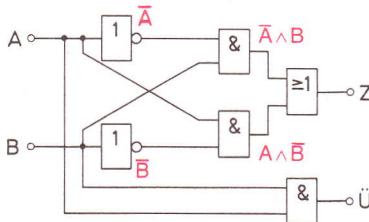


Bild 14.2 Schaltung eines Halbaddierers mit Grundgliedern

$$Z = (A \wedge \bar{B}) \vee (\bar{A} \wedge B)$$

$$\bar{U} = A \wedge B$$

Die sich aus den Gleichungen ergebende Schaltung zeigt Bild 14.2. Die Schaltung kann auf NAND-Verknüpfungen umgerechnet werden. Eine besonders einfache Schaltung ergibt sich durch mehrfache Verwendung von \bar{U} (Bild 14.3).

$$Z = (A \wedge \bar{B}) \vee (\bar{A} \wedge B) = (A \vee \bar{A}) \wedge (A \vee B) \wedge (\bar{B} \vee \bar{A}) \wedge (\bar{B} \vee B)$$

$$Z = (A \vee B) \wedge (\bar{A} \vee \bar{B}) = (A \vee B) \wedge \overline{\bar{A} \wedge B} = (A \vee B) \wedge \bar{U}$$

$$Z = (A \wedge \bar{U}) \vee (B \wedge \bar{U})$$

$$Z = \overline{(A \wedge \bar{U}) \vee (B \wedge \bar{U})} = \overline{\overline{A \wedge \bar{U}} \wedge \overline{B \wedge \bar{U}}}$$

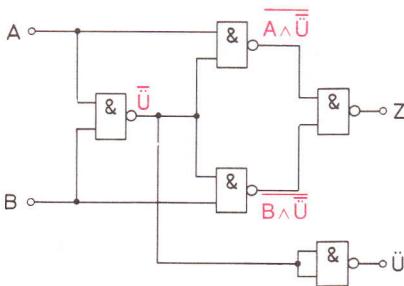


Bild 14.3 Schaltung eines Halbaddierers mit NAND-Gliedern

14.2 Volladdierer

Zum Aufbau von Addierwerken werden Schaltungen benötigt, die drei Dualziffern addieren können, da bei der Addition von zwei Dualzahlen die Überträge mit addiert werden müssen.

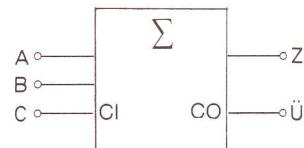
Beispiel:

$$\begin{array}{r}
 1 & 1 & 1 & | & 1 \\
 & 1 & 0 & 1 & 1 \\
 + & 0 & 1 & | & 1 \\
 \hline
 1 & 0 & 0 & 1 & 0
 \end{array}$$

Ein Volladdierer ist eine Schaltung, die drei Dualziffern addieren kann.

Die Schaltung eines Volladdierers kann nach den Regeln der Schaltungssynthese (s. Kapitel 5) entworfen werden. Der Volladdierer benötigt drei Eingänge – für jede zu addierende Dualziffer einen. Diese sollen A, B und C genannt werden. Die Ausgänge heißen wie beim Halbaddierer Z und Ü.

Bild 14.4 Volladdierer mit Wahrheitstabelle CO kennzeichnet Übertragsausgang, CI kennzeichnet Übertragseingang (engl.: Carry in input). Σ ist das Summenzeichen



Fall	C	B	A	Ü	Z
1	0	0	0	0	0
2	0	0	1	0	1
3	0	1	0	0	1
4	0	1	1	1	0
5	1	0	0	0	1
6	1	0	1	1	0
7	1	1	0	1	0
8	1	1	1	1	1

Die Wahrheitstabelle des Volladdierers ergibt sich aus den Rechenregeln für die Addition. Sie ist in Bild 14.4 dargestellt. Im Fall 1 sind Ü und Z Null, da alle Eingangsziffern Null sind. Im Fall 2 ergibt sich aus der Addition von $0 + 0 + 1$ $Z = 1$ und $\dot{U} = 0$. Im Fall 4 ist $0 + 1 + 1$ zu rechnen, was $Z = 0$ und $\dot{U} = 1$ ergibt. Betrachten wir noch den Fall 8. Die Rechnung $1 + 1 + 1$ führt zu $Z = 1$ und $\dot{U} = 1$.

Die ODER-Normalform für Z besteht aus vier Vollkonjunktionen. Sie lautet:

$$Z = (A \wedge \overline{B} \wedge \overline{C}) \vee (\overline{A} \wedge B \wedge \overline{C}) \vee (\overline{A} \wedge \overline{B} \wedge C) \vee (A \wedge B \wedge C)$$

Für Ü ergibt sich die ODER-Normalform:

$$\dot{U} = (A \wedge B \wedge \overline{C}) \vee (A \wedge \overline{B} \wedge C) \vee (\overline{A} \wedge B \wedge C) \vee (A \wedge B \wedge C)$$

Z	A		\bar{A}
B	1		1
\bar{B}	1		1
\bar{C}		C	\bar{C}

\bar{U}	A	\bar{A}	
B	1	1	1
\bar{B}			1
\bar{C}		C	\bar{C}

Bild 14.5 KV-Diagramme eines Volladdierers

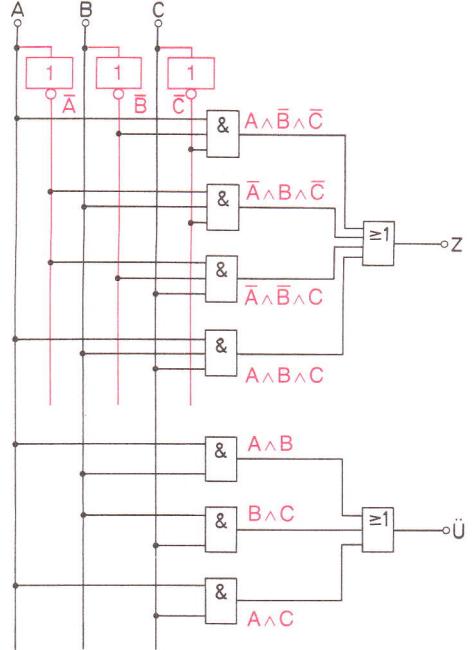


Bild 14.6 Schaltung eines Volladdierers

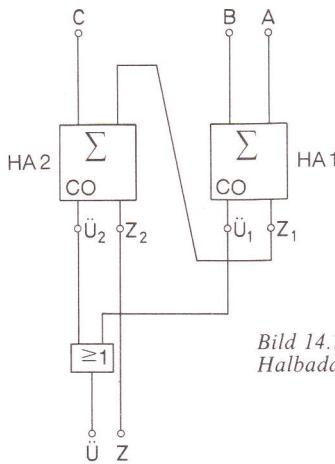


Bild 14.7 Volladdierer, aus zwei Halbaddierern aufgebaut

Die Gleichung für Z lässt sich nicht mehr vereinfachen (Bild 14.5). Für \bar{U} erhält man mit Hilfe des KV-Diagramms die vereinfachte Gleichung:

$$\bar{U} = (A \wedge B) \vee (B \wedge C) \vee (A \wedge C)$$

Diese Gleichungen führen zu der Schaltung Bild 14.6.

Ein Volladdierer kann auch aus zwei Halbaddierern und einem ODER-Glied aufgebaut werden.

Typ	Bestellbezeichnung
FLH 451-74H183	Q67000-H495
FLH 455-84H183	Q67000-H511

Der Baustein FLH 451/455 nimmt über die Dateneingänge A und B sowie C_n (Übertragsinformation der niedrigwertigeren Stelle) Daten auf und gibt das Additionsergebnis über den Σ -Ausgang sowie den Ausgang C_{n+1} (Übertragsausgang für höherwertigere Stelle) ab.

Statische Kenndaten
im Temperaturbereich 1 und 5

		Prüfbedingungen	untere Grenze B	typ.	obere Grenze A	Einheit
Speisespannung	U_S		4,74	5,0	5,25	V
H-Eingangsspannung	U_{IH}	$U_S = 4,75 \text{ V}$	2,0			V
L-Eingangsspannung	U_{IL}	$U_S = 4,75 \text{ V}$			0,8	V
Eingangsklemmspannung	$-U_I$	$U_S = 4,75 \text{ V}, -I_I = 8 \text{ mA}$			1,5	V
H-Ausgangsspannung	U_{QH}	$U_S = 4,75 \text{ V}, U_{IH} = 2,0 \text{ V}, -I_{QH} = 1 \text{ mA}$	2,4	3,5		V
L-Ausgangsspannung	U_{QL}	$U_S = 4,75 \text{ V}, U_{IL} = 0,8 \text{ V}, I_{QL} = 20 \text{ mA}$		0,2	0,4	V
Eingangsstrom je Eingang	I_I	$I_I = 5,5 \text{ V}$			1	mA
H-Eingangsstrom je Eingang	I_{IH}	$I_{IH} = 2,4 \text{ V} \quad \quad U_S = 5,25 \text{ V}$			150	μA
L-Eingangsstrom je Eingang	$-I_{IL}$	$U_S = 5,25 \text{ V}, U_{IL} = 0,4 \text{ V}$			6	mA
Kurzschlußausgangsstrom je Ausgang	$-I_Q$	$U_S = 5,25 \text{ V}$	40		100	mA
H-Speisestrom	I_{SH}	$U_S = 5,25 \text{ V}, U_{IQ} = 4,5 \text{ V}$		40		mA
L-Speisestrom	I_{SL}	$U_S = 5,25 \text{ V}, U_{IL} = 0 \text{ V}$		48	75	mA

Schaltzeiten bei $U_S = 5 \text{ V}, T_U = 25^\circ \text{C}$

Signallaufzeit	t_{PLH}	$C_L = 25 \text{ pF}, R_L = 280 \Omega$		10	15	ns
	t_{PHL}			12	18	ns

Logische Daten

H-Ausgangslastfaktor je Ausgang	F_{QH}			24	
L-Ausgangslastfaktor je Ausgang	F_{QL}			12	
Eingangslastfaktor je Eingang	F_I			3,75	

Logisches Verhalten			Ausgänge	
Eingänge			$\Sigma = T$	$C_{n+1} = \dot{U}$
C_n	B	A		
L	L	L	L	L
L	L	H	H	L
L	H	L	H	L
L	H	H	L	H
H	L	L	H	L
H	L	H	L	H
H	H	L	L	H
H	H	H	H	H

Anschlußanordnung
Ansicht von oben

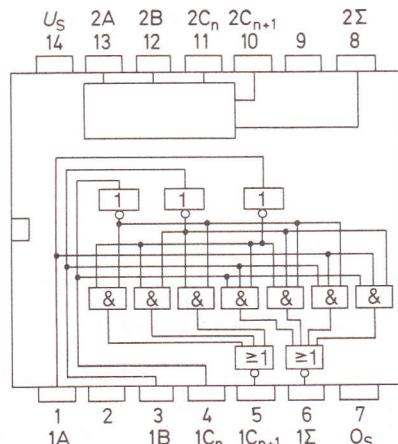


Bild 14.8 Datenblatt der integrierten Schaltung FLH451-74H183 (Siemens)

Die Schaltung zeigt Bild 14.7. Auf diese Schaltung geht auch der Name «Halbaddierer» zurück. Zwei Halbaddierer bilden den Volladdierer. Lediglich ein ODER-Glied wird noch zusätzlich benötigt.

Volladdierer werden überwiegend als integrierte Schaltungen hergestellt. Sie werden auch 1-Bit-Volladdierer genannt, da sie bei Additionen nach dem Beispiel zu Beginn des Abschnitts 14.2 nur eine Spalte der Zahlen (roter Kasten) addieren können.

Eine häufig verwendete integrierte Schaltung trägt die Bezeichnung FLH 451-74H183. Sie enthält zwei 1-Bit-Volladdierer in TTL-Technik. Das vollständige Datenblatt ist in Bild 14.8 wiedergegeben.

14.3 Paralleladdierschaltung

Will man zwei vierstellige Dualzahlen in einem Arbeitsschritt addieren, benötigt man einen Halbaddierer und drei Volladdierer. Die erste Spalte von rechts (Wertigkeit 2^0) kann mit einem Halbaddierer addiert werden, da in dieser Spalte nie ein Übertrag auftreten kann. In den anderen drei Spalten mit den Wertigkeiten 2^1 , 2^2 und 2^3 können Überträge auftreten. Für die Addition dieser Spalten werden Volladdierer benötigt (Bild 14.9).

Das Addieren in einem Arbeitsschritt wird *Paralleladdition* genannt. Eine 4-Bit-Paralleladdierschaltung zeigt Bild 14.10. Auf die Eingänge des Halbaddierers HA sind die ersten Ziffern von rechts der beiden zu addierenden Zahlen (Wertigkeit 2^0) geschaltet. Der Ausgang Z_0 führt zum Ergebnisregister. Der Übertragungsausgang \dot{U}_0 ist mit einem Eingang des Volladdierers VA1 für die zweite Spalte verbunden, denn in dieser Spalte muß ein entstehender Übertrag addiert werden.

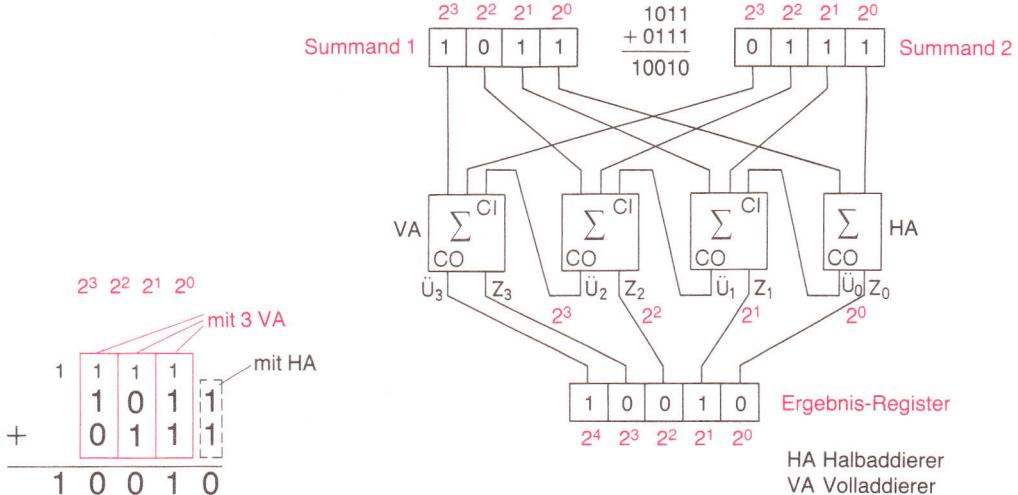


Bild 14.9 Addition von zwei viersteligen Dualzahlen

Bild 14.10 4-Bit-Parallel-Addierschaltung

Der Volladdierer VA1 für die zweite Spalte erhält außer dem Übertrag des Halbaddierers die zweiten Ziffern der zu addierenden Zahlen (Wertigkeit 2^1). Der Ausgang Z_1 dieses Volladdierers liefert eine Ergebnisziffer. Der Übertragsausgang \dot{U}_1 des Volladdierers führt auf einen Eingang des Volladdierers VA2 für die dritte Spalte (Wertigkeit 2^2). Dieser Volladdierer erhält außerdem die dritten Ziffern von rechts der zu addierenden Zahlen.

Der Volladdierer VA3 für die vierte Spalte (Wertigkeit 2^3) wird entsprechend beschaltet. Das Übertragssignal dieses Volladdierers wird dem Ergebnisregister zugeführt.

Eine Paralleladdierstufe zur Addition von zwei 8-Bit-Dualzahlen besteht aus einem Halbaddierer und sieben Volladdierern.

14.4 Serielle Addierschaltung

Bei einer seriellen Addierschaltung werden die Spalten der zu addierenden Dualzahlen zeitlich nacheinander addiert. Zuerst erfolgt die Addition in der Spalte mit der niedrigsten Wertigkeit (Spalte ganz rechts). Dann erfolgt die Addition in der Spalte mit der nächsthöheren Wertigkeit. Danach wird die Spalte mit der dann nächsthöheren Wertigkeit addiert, und so fort, bis alle Spalten addiert sind. Ein Übertrag aus der Addition der vorhergehenden Spalte wird in die gerade durchgeführte Addition mit übernommen. Der Ablauf der Addition ähnelt einer handschriftlich vorgenommenen Addition von zwei Dualzahlen.

Der prinzipielle Aufbau einer seriellen Addierschaltung ist in Bild 14.11 angegeben. Die erste zu addierende Zahl, also der erste Summand, ist im Schieberegister A gespeichert.

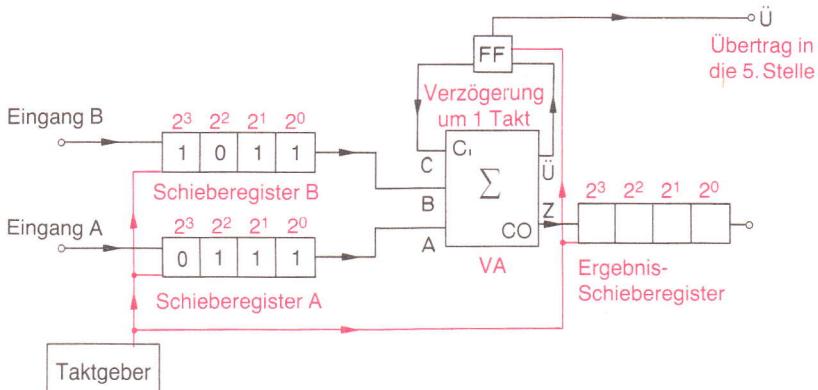


Bild 14.11 Prinzipieller Aufbau einer seriellen Addierschaltung

Die zweite zu addierende Zahl, also der zweite Summand, ist im Schieberegister B enthalten. Die seriellen Ausgänge der Schieberegister sind auf einen Volladdierer geführt. Der Z-Ausgang des Volladdierers liefert Ergebnissignale, die in das Ergebnis-Schieberegister aufgenommen werden. Das Signal, das am Übertragsausgang des Volladdierers liegt, wird um einen Takt verzögert und dann bei der nächsten Spaltenaddition mitaddiert. Die Verzögerung um einen Takt wird durch ein Master-Slave-Flipflop FF erreicht.

Mit dem 1. Takt werden z.B. die beiden 1-Signale der Wertigkeit 2^0 nach Bild 14.11 dem Volladdierer zugeführt. Am Ausgang Z erscheint 0, am Ausgang Ü erscheint 1. Das Z-Signal wird in das Ergebnis-Schieberegister übernommen. Das Ü-Signal wird vom Master-Slave-Flipflop FF gespeichert. Mit dem zweiten Takt werden die beiden Signale mit der Wertigkeit 2^1 und das Übertragssignal auf die Eingänge des Volladdierers gegeben (z.B. A = 1, B = 1, C = 1). Das Z-Signal (z.B. Z = 1) wird in das Ergebnis-Schieberegister eingespeichert. Das Ü-Signal (z.B. Ü = 1) wird vom Flipflop FF aufgenommen.

Mit dem 3. Takt wird die dritte Spalte mit der Wertigkeit 2^2 addiert. Dann folgt die Addition der vierten Spalte mit dem 4. Takt. Die serielle Addition ist dann beendet. Die vier Ergebnis-Bit mit den Wertigkeiten $2^0, 2^1, 2^2$ und 2^3 sind im Ergebnis-Schieberegister enthalten. Das 5. Ergebnis-Bit mit der Wertigkeit 2^4 befindet sich im Flipflop FF und kann dort abgerufen werden. Weitere Takte dürfen nicht wirksam werden.

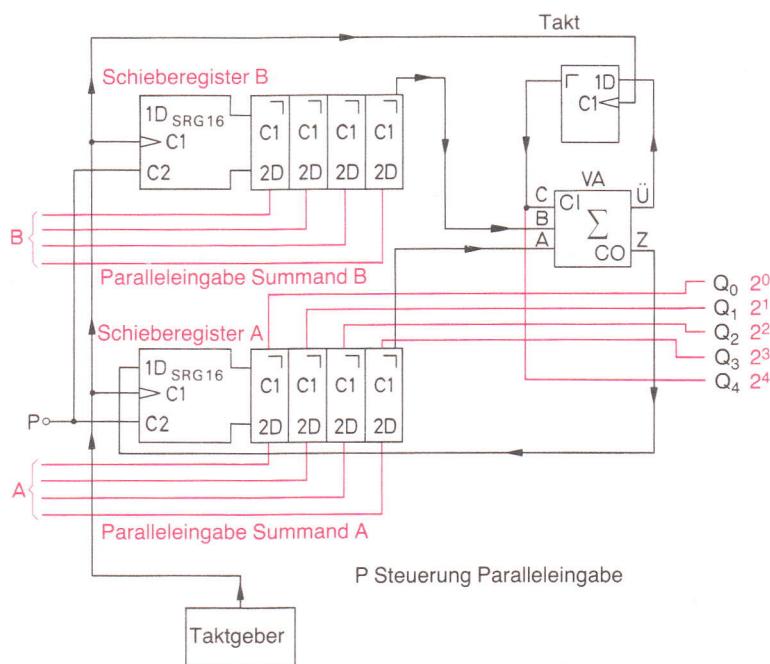
Die serielle Addition erfordert mehr Zeit als die Paralleladdition.

Das Ergebnis-Schieberegister kann eingespart werden. Das Schieberegister A (oder auch das Schieberegister B) kann die Aufgabe des Ergebnis-Schieberegisters mit übernehmen. Das Schieberegister A wird während der spaltenweisen Addition leergetaktet. Die Ergebnissignale des Ausgangs Z des Volladdierers können auf den Eingang des Schieberegis-

sters A gegeben und dort eingespeichert werden. Sie sind nach Ende des Additionsver-
ganges dort verfügbar.

Die sich aus diesen Überlegungen ergebende serielle Addierschaltung zeigt Bild 14.12. Die beiden Schieberegister sind mit D-Flipflops aufgebaut. Sie haben parallele Dateneingabe, d.h. die Summanden A und B werden parallel eingegeben. Die Paralleleingabe erfolgt, wenn 1-Signal am Eingang P anliegt. Dann werden über die Eingänge C2 der Schieberegister die 2D-Eingänge der Schieberegister-Flipflops freigegeben. Bei P = 0 ist die Paralleleingabe gesperrt. Das Weitertakten der eingespeicherten Information erfolgt über die C1-Eingänge. Das Schieberegister A arbeitet gleichzeitig als Ergebnis-Schieberegister. Das Ergebnis der Addition liegt nach Ende des Additionsvorganges an den Ausgängen Q₀ bis Q₄.

Bild 14.12 Serielle
4-Bit-Addierschal-
tung



14.5 Subtrahierschaltungen

Subtrahierschaltungen können nach den gleichen Gesetzmäßigkeiten aufgebaut werden wie Addierschaltungen. In Wahrheitstabellen werden die möglichen Eingangssignale und die dazu gewünschten Ausgangssignale zusammengestellt. Nach den Regeln der Schaltungssynthese wird dann die gesuchte Schaltung entwickelt.

Subtraktionen lassen sich auf Additionen zurückführen (s. Kapitel 8, Abschnitt 8.2.6.2). Eine aus Volladdierern bestehende Additionsschaltung kann durch kleine Änderungen in eine Subtraktionsschaltung umgewandelt werden.

14.5.1 Halbsubtrahierer

Eine sehr einfache Subtrahierschaltung ist der Halbsubtrahierer.

Ein Halbsubtrahierer kann eine Dualziffer von einer anderen Dualziffer abziehen.

Es gelten folgende Rechenregeln:

$$0 - 0 = 0$$

$$0 - 1 = -1$$

$$1 - 0 = 1$$

$$1 - 1 = 0$$

Die Dualziffer, von der abgezogen werden soll (Minuend), enthält den Variablennamen A. Die abzuziehende Dualziffer (Subtrahend) soll B genannt werden. Die Ziffer 0 wird dem binären Zustand 0 zugeordnet, die Ziffer 1 dem binären Zustand 1.

Der Halbsubtrahierer hat also die beiden Eingänge A und B und einen Ergebnisausgang D für die Differenz. Die Darstellung von -1 bereitet jedoch Schwierigkeiten. Es wird ein zweiter Ausgang vorgesehen. Das Ergebnis -1 erzeugt an diesem zweiten Ausgang zusätzlich ein 1-Signal. Der zweite Ausgang erhält die Bezeichnung E. Er wird auch Entleihungsausgang genannt.

Führt die Subtraktion also zu -1 , so soll $D = 1$ und $E = 1$ sein. Die sich aus diesen Überlegungen ergebende Wahrheitstabelle zeigt Bild 14.13.

Nach der Wahrheitstabelle Bild 14.13 lassen sich folgende Gleichungen aufstellen:

$$D = (\overline{A} \wedge B) \vee (A \wedge \overline{B})$$

$$E = \overline{A} \wedge B$$

Die Gleichungen führen zu der Schaltung Bild 14.14. Das 1-Signal am Ausgang E ist außer zur Kennzeichnung von -1 vor allem für mehrspaltige Subtraktionen, also für Subtraktionen von mehrstelligen Dualzahlen, für die sogenannte Entleihung erforderlich.

14.5.2 Vollsubtrahierer

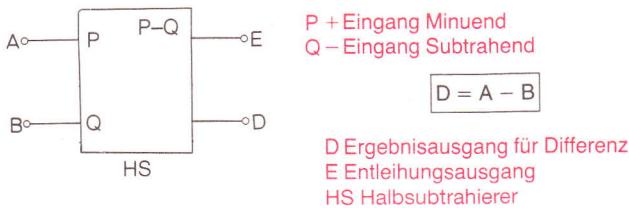
Ein Vollsubtrahierer wird für mehrspaltiges Subtrahieren benötigt.

Beispiel:

$$\begin{array}{r} & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\ & \boxed{1} & \boxed{1} & \boxed{0} & 1 & 1 \\ - & 1 & 0 & 1 & 1 & 0 \\ \text{Entleihung} \rightarrow & \underline{\boxed{1}} & & & & \\ & 0 & 0 & 1 & 0 & 1 \\ & & & & & \hline & & & & & 5 \end{array}$$

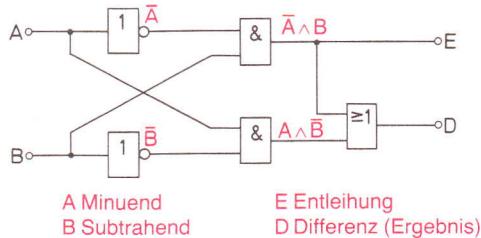
Die Subtraktion in der 3. Spalte (Wertigkeit 2^2) fordert eine Entleihung. Ein Halbsubtrahierer würde hier $D = 1$ und $E = 1$ ausgeben. Das 1-Signal des E-Ausgangs muß in

Bild 14.13 Halbsubtrahierer mit Wahrheitstabelle



Fall	A	B	E	D
1	0	0	0	0
2	0	1	1	1
3	1	0	0	1
4	1	1	0	0

Bild 14.14 Schaltung eines Halbsubtrahierers



der nächsten Spalte (Wertigkeit 2^3) zum abzuziehenden Ziffernwert hinzugeaddiert werden. Hierfür ist ein Vollsubtrahierer erforderlich.

Ein Vollsubtrahierer ist eine Schaltung, die zum Wert der abzuziehenden Ziffer (Subtrahend) eine Entleihung (1-Signal) hinzugeaddieren kann und den so vergrößerten Subtrahend vom Minuend abzieht.

Der Vollsubtrahierer muß drei Eingänge haben (Bild 14.15). An den Eingang A wird die Ziffer gelegt, von der abgezogen werden soll, der sogenannte Minuend. An den Eingang B wird die abzuziehende Ziffer, der Subtrahend, gelegt. An den Eingang E_x kommt die Entleihung von der vorhergehenden Spalte. Die Ziffern an E_x und B werden addiert. Die Summe wird von A abgezogen. Die Differenz ergibt D.

$$D = A - (B + E_x)$$

Ist eine neue Entleihung erforderlich, erscheint an E 1-Signal.

Ein Vollsubtrahierer kann aus einem Halbaddierer und aus einem Halbsubtrahierer aufgebaut werden. Der Halbaddierer HA in Bild 14.16 addiert die Ziffern B und E_x zum Gesamtsubtrahenden Z. Ergibt sich ein Übertrag, wird dieser an den Ausgang E gegeben.

Der Halbsubtrahierer HS subtrahiert Z von A. Er rechnet also $A - Z$. Wird eine Entleihung erforderlich, erscheint am Ausgang E_1 1-Signal. Dieses wird über ein ODER-Glied an den E-Ausgang gegeben (Bild 14.16).

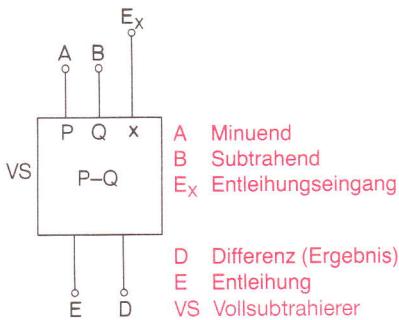


Bild 14.15 Vollsubtrahierer

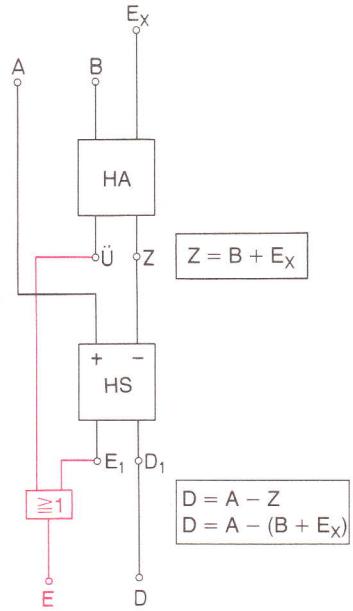


Bild 14.16 Vollsubtrahierer, aufgebaut aus einem Halbaddierer und aus einem Halb-subtrahierer

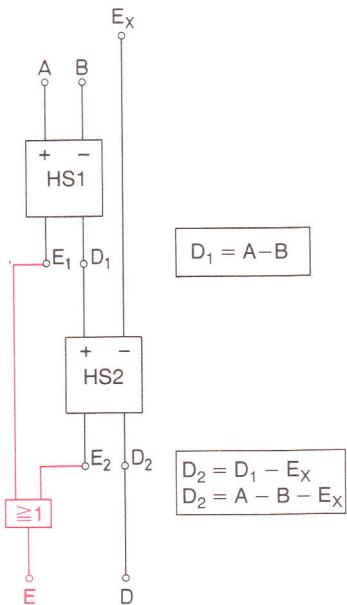


Bild 14.17 Vollsubtrahierer, aufgebaut aus zwei Halbsubtrahierern

Ein Vollsubtrahierer kann auch aus zwei Halbsubtrahierern gebildet werden (Bild 14.17). Im Halbsubtrahierer HS1 wird zuerst die Differenz $A - B$ gebildet. Von diesem Ergebnis wird eine Entleihung in der Spalte vorher durch den Halbsubtrahierer HS2 abgezogen. Bei erforderlichen Entleihungen sowohl bei der Subtraktion im HS1 als auch bei der Subtraktion im HS2 erscheint am Ausgang E das 1-Signal.

14.5.3 4-Bit-Subtrahier-Schaltung

Eine 4-Bit-Subtrahierschaltung kann von einer vierstelligen Dualzahl eine maximal vierstellige Dualzahl abziehen.

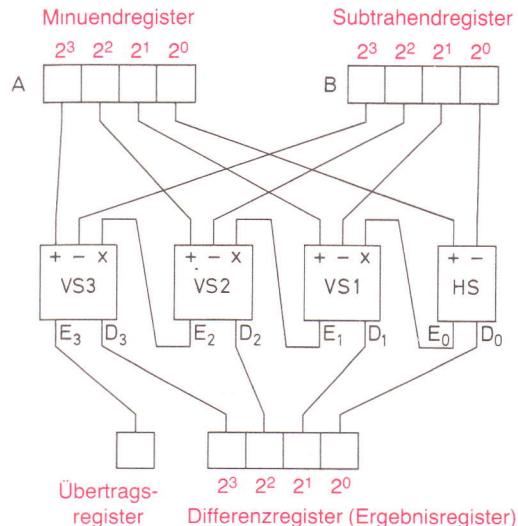
Für den Aufbau der Schaltung werden drei Vollsubtrahierer und ein Halbsubtrahierer benötigt (Bild 14.18).

Der Halbsubtrahierer HS subtrahiert die wertniedrigste Ziffer des Subtrahenden von der wertniedrigsten Ziffer des Minuenden. Wird eine Entleihung erforderlich, wird $E = 1$. Diese Entleihung wird bei der Subtraktion der Ziffern mit der Wertigkeit 2^1 berücksichtigt. Es wird in diesem Fall eine 1 mehr abgezogen. Wird wieder eine Entleihung erforderlich, erscheint an E_1 1-Signal. Bei der Subtraktion der Ziffern mit der Wertigkeit 2^2 wird diese erneute Entleihung berücksichtigt, indem wieder eine 1 mehr abgezogen wird. Entsprechend wird bei der Subtraktion der Ziffern mit der Wertigkeit 2^3 verfahren.

Wird bei der letzten Ziffernsubtraktion eine Entleihung erforderlich, erscheint im Übertragsregister eine 1. Das bedeutet, daß die abzuziehende Zahl (Subtrahend) größer ist als die Zahl, von der abgezogen wurde (Minuend). Das Ergebnis ist eine negative Zahl. Die negative Zahl wird aber nicht richtig dargestellt.

Ist der Subtrahend größer als der Minuend, wird die entstehende negative Zahl im Differenzregister falsch angegeben.

Bild 14.18 4-Bit-Subtrahiererschaltung (Parallel-Subtrahierschaltung)



Der Inhalt des Differenzregisters muß komplementiert werden. Die Komplementbildung ist in Kapitel 8 näher erläutert.

Den Betrag der negativen Ergebniszahl erhält man, indem man den Inhalt des Differenzregisters negiert und 1 dazuzählt.

14.5.4 Subtrahierschaltung mit Volladdierern

In Kapitel 8, Abschnitt 8.2.6.2, wurde gezeigt, daß die Subtraktion von Dualzahlen auf eine Addition des Komplements der abzuziehenden Dualzahl zurückgeführt werden kann. Eine 4-Bit-Subtrahierschaltung läßt sich daher auch aus einer 4-Bit-Addierschaltung entwickeln. Das Prinzip einer solchen Subtrahierschaltung zeigt Bild 14.19.

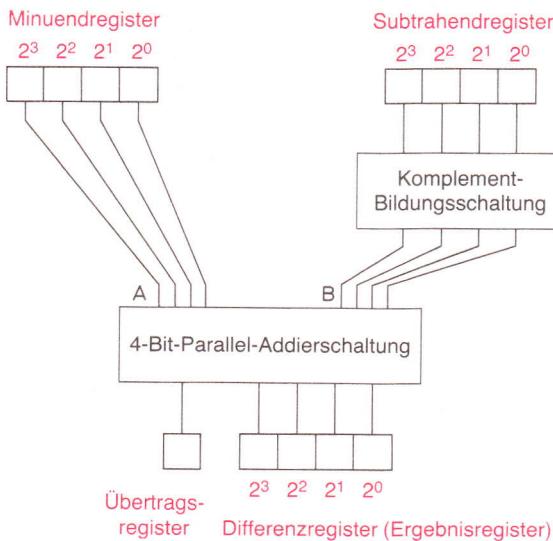
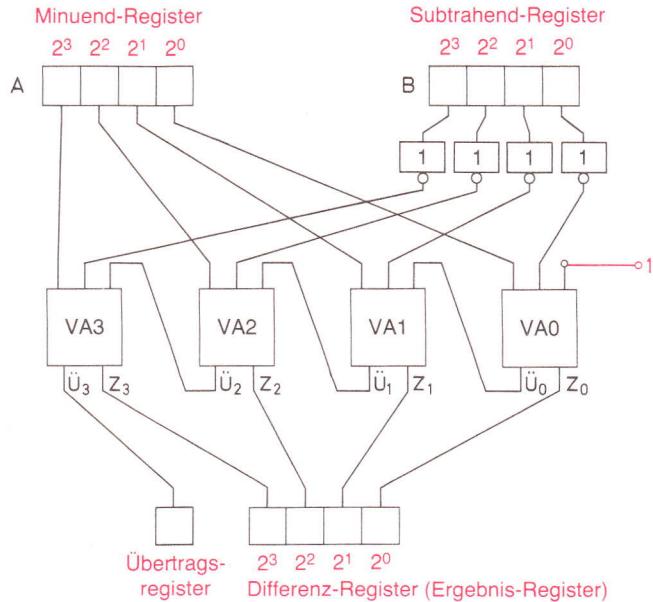


Bild 14.19 Prinzip einer Subtrahierschaltung

Bild 14.20 4-Bit-Subtrahierschaltung mit Volladdierern



14.6 Addier-Subtrahier-Werk

Die im vorstehenden Abschnitt betrachtete 4-Bit-Subtrahierschaltung mit Volladdierern kann leicht so abgewandelt werden, daß sie sich wahlweise zum Addieren und zum Subtrahieren eignet. Bei der Verwendung zur Addition sind nur zwei Maßnahmen zu treffen:

1. Die Negation des Inhalts des Subtrahendregisters muß unterbleiben.
2. Die Addition von 1 über den Eingang C des Volladdierers VA1 darf nicht erfolgen.

Die Negationsglieder werden durch EXKLUSIV-ODER-Glieder ersetzt (Bild 14.21). Der B-Eingang wird zur Steuerung verwendet. Bei $B = 0$ erfolgt keine Negation, bei $B = 1$ wird negiert. Das so entstehende Addier-Subtrahier-Werk ist in Bild 14.22 dargestellt. Wird an den Steuereingang S 0-Signal gelegt, wird die Addition $Z = A + B$ durchgeführt. Wird an den Steuereingang 1-Signal gelegt, arbeitet die Schaltung als Subtrahierschaltung. Es wird die Differenz $Z = A - B$ gebildet.

Das 4-Bit-Addier-Subtrahier-Werk kann noch universell verwendbar gemacht werden. Schaltet man den Ausgängen des A-Registers ebenfalls EXKLUSIV-ODER-Glieder nach, kann bei entsprechender Steuerung auch $B - A$ gerechnet werden. Werden außerdem die Ausgänge des A-Registers und die Ausgänge des B-Registers durch UND-Glieder wahlweise sperrbar gemacht, ergeben sich noch weit mehr Möglichkeiten. Man kann dann z.B. auch A in $-A$ umwandeln.

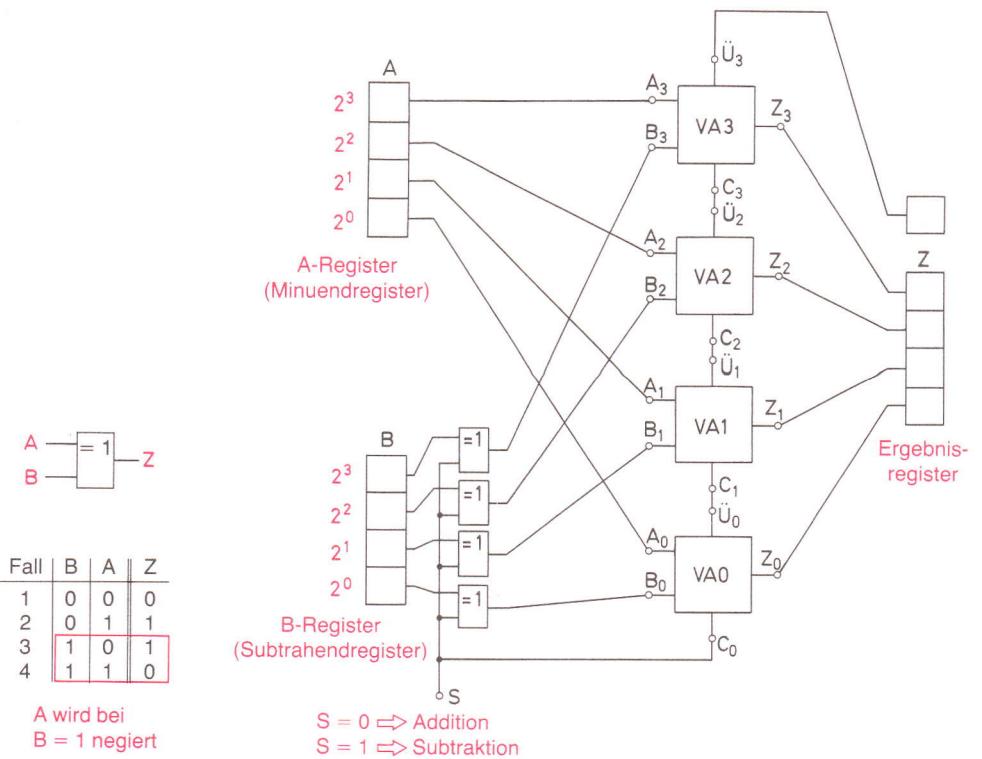


Bild 14.21 Schaltzeichen und Wahrheitstabelle eines Exklusiv-ODER-Gliedes (Antivalenzglied)

Bild 14.22 4-Bit-Addier-Subtrahier-Werk

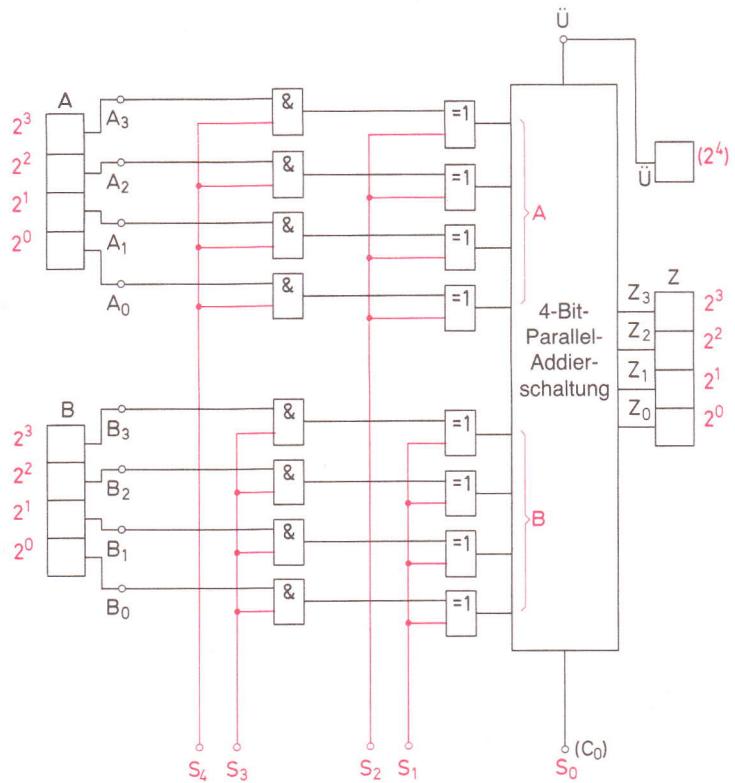
Das erweiterte 4-Bit-Addier-Subtrahier-Werk zeigt Bild 14.23. Die vier Volladdierer sind in einem Block zu einer 4-Bit-Paralleladdierschaltung zusammengefaßt worden. Die Eingänge S_0 bis S_4 sind Steuereingänge.

Fünf Steuereingänge ergeben 32 verschiedene Steuermöglichkeiten. Diese sind in Bild 14.24 aufgeführt. Jede 5-Bit-Einheit kann als ein 5-Bit-Befehl aufgefaßt werden. Der Befehl, den Inhalt des Registers A mit dem Inhalt des Registers B zu addieren, lautet somit 11000.

Ist $S_4 = 1$, wird der Inhalt des A-Registers durchgelassen. Bei $S_2 = 1$ wird der Inhalt des A-Registers negiert. Der Befehl 10100 führt also zu $Z = \bar{A}$.

Soll das Komplement von A gebildet werden, so muß der Befehl 10101 lauten. A wird durchgelassen, A wird negiert, 1 wird dazugezählt. Die Schaltung erzeugt dann $-A$, da das Komplement von A gleich $-A$ ist. Negative Dualzahlen sind in Kapitel 8, Abschnitt 8.2.7, erläutert.

Bild 14.23 Erweitertes 4-Bit-Addier-Subtrahier-Werk



Der Befehl 00010 soll nach Bild 14.24 zur Ausgabe von -1 führen. Das ist nicht leicht einsehbar. Bei diesem Befehl sind die Register A und B gesperrt. Die Ausgänge aller UND-Glieder führen 0-Signal. Da $S_1 = 1$ ist, werden die vier 0-Signale der UND-Glieder von B negiert. Die B-Eingänge der 4-Bit-Parallel-Addierschaltung erhalten also 1111, die A-Eingänge 0000. Es wird folgende Addition durchgeführt:

$$\begin{array}{r} A \rightarrow 0 \ 0 \ 0 \ 0 \\ B \rightarrow +1 \ 1 \ 1 \ 1 \\ \hline Z \rightarrow 1 \ 1 \ 1 \ 1 \end{array}$$

Dieser Wert soll nicht als 15, sondern als -1 angesehen werden, denn er ist ebenfalls das Komplement von 0001. (Definitionsbereiche positiver und negativer Dualzahlen, siehe Abschnitt 8.2.7.)

Wie muß nun der Befehl für $B - A$ lauten? Die Inhalte des A-Registers und des B-Registers müssen durchgelassen werden ($S_4 = 1, S_3 = 1$). Der Inhalt des A-Registers muß negiert werden ($S_2 = 1$). Eine 1 muß hinzugeaddiert werden ($S_0 = 1$). Der Befehl lautet also 11101.

Das erweiterte 4-Bit-Addier-Subtrahier-Werk kann somit über das Addieren und Subtrahieren hinaus für weitere Zwecke verwendet werden.

Fall Nr.	Befehle					Funktionen
	S ₄	S ₃	S ₂	S ₁	S ₀	
1	0	0	0	0	0	0
2	0	0	0	0	1	1
3	0	0	0	1	0	-1
4	0	0	0	1	1	0
5	0	0	1	0	0	-1
6	0	0	1	0	1	0
7	0	0	1	1	0	-2
8	0	0	1	1	1	-1
9	0	1	0	0	0	B
10	0	1	0	0	1	B + 1
11	0	1	0	1	0	-B - 1 = \bar{B}
12	0	1	0	1	1	-B
13	0	1	1	0	0	B - 1
14	0	1	1	0	1	B
15	0	1	1	1	0	-B - 2
16	0	1	1	1	1	-B - 1 = \bar{B}
17	1	0	0	0	0	A
18	1	0	0	0	1	A + 1
19	1	0	0	1	0	A - 1
20	1	0	0	1	1	A
21	1	0	1	0	0	-A - 1 = \bar{A}
22	1	0	1	0	1	-A
23	1	0	1	1	0	-A - 2
24	1	0	1	1	1	-A - 1 = \bar{A}
25	1	1	0	0	0	A + B
26	1	1	0	0	1	A + B + 1
27	1	1	0	1	0	A - B - 1
28	1	1	0	1	1	A - B
29	1	1	1	0	0	B - A - 1
30	1	1	1	0	1	B - A
31	1	1	1	1	0	-A - B - 2
32	1	1	1	1	1	-A - B - 1

Bild 14.24 Steuermöglichkeiten des erweiterten 4-Bit-Addier-Subtrahier-Werkes Bild 14.23

14.7 Multiplikationsschaltungen

Für die Multiplikation von Dualzahlen gelten folgende Rechenregeln:

$$0 \cdot 0 = 0$$

$$0 \cdot 1 = 0$$

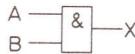
$$1 \cdot 0 = 0$$

$$1 \cdot 1 = 1$$

Ordnet man der Ziffer 0 den binären Zustand 0 und der Ziffer 1 den binären Zustand 1 zu, ergibt sich die Wahrheitstabelle nach Bild 14.25. Es ist die Wahrheitstabelle eines UND-Gliedes. Grundelement der Multiplikationsschaltungen ist das UND-Glied.

Das UND-Glied ist ein 1-Bit-Multiplizierer.

Bild 14.25 Wahrheitstabelle eines 1-Bit-Multiplizierers (UND-Glied)



Fall	B	A	X
1	0	0	0
2	0	1	0
3	1	0	0
4	1	1	1

Multiplikationen können in einem Arbeitsschritt, also parallel, durchgeführt werden. Hierzu dienen Parallel-Multiplikationsschaltungen.

Ein Multiplizieren Bit nach Bit, also eine serielle Multiplikation, ist ebenfalls möglich. Sie benötigt bei vielsteligen Zahlen weniger Schaltungsaufwand als die Parallelmultiplikation, erfordert aber mehr Zeit.

14.7.1 Parallel-Multiplikationsschaltung

Bei der Multiplikation werden die Begriffe Multiplikand und Multiplikator verwendet. Der Multiplikand ist die Grundzahl von der ausgegangen wird. Der Multiplikator ist die Zahl, mit der vervielfacht wird. Das Ergebnis nennt man Produkt.

Multiplikand Multiplikator Produkt

$$2 \quad \cdot \quad 3 = 6$$

Die Parallel-Multplikation soll zunächst mit zweistelligen Dualzahlen durchgeführt werden:

$$\boxed{2 \cdot 3 = 6}$$

$$\begin{array}{r} 10 \cdot 11 \\ \hline 10 \rightarrow 1. \text{ Summand} \\ 10 \rightarrow 2. \text{ Summand} \\ \hline 110 \rightarrow \text{Ergebnis} \end{array}$$

Der erste Summand ergibt sich aus zwei 1-Bit-Multiplikationen ($1 \cdot 0, 1 \cdot 1$). Hierfür sind zwei UND-Glieder erforderlich. Der zweite Summand entsteht ebenfalls durch zwei 1-Bit-Multiplikationen ($1 \cdot 0, 1 \cdot 1$), die auch durch zwei UND-Glieder erfolgen. Multiplikand und Multiplikator befinden sich in zwei Registern (Bild 14.26). An den Ausgängen der UND-Glieder sind die Summanden verfügbar.

Die beiden Summanden müssen jetzt stellenrichtig addiert werden. Die Addition erfolgt

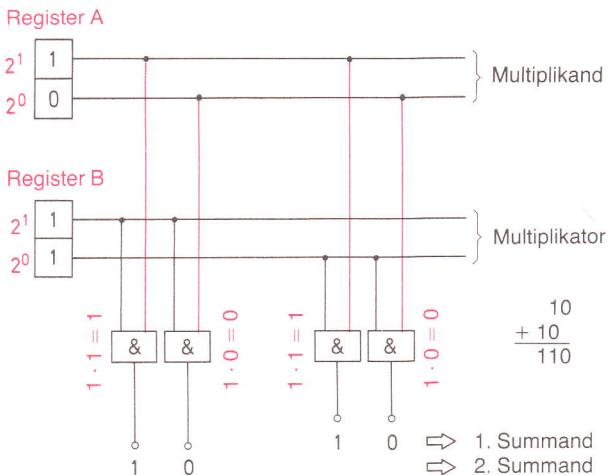


Bild 14.26 Multiplikationsschaltung zum Erzeugen der Summanden

in einer 2-Bit-Parallel-Addierschaltung. Die UND-Glieder müssen so angeschlossen werden, daß der zweite Summand um eine Stelle nach links verschoben zum ersten Summanden hinzuaddiert wird (Bild 14.27).

Für die Multiplikation von mehr als 2stelligen Dualzahlen ist die Schaltung Bild 14.27 entsprechend zu erweitern. Sollen zwei 4stellige Dualzahlen miteinander multipliziert werden, sind 16 UND-Glieder erforderlich, denn 16 1-Bit-Multiplikationen müssen ausführbar sein. Es entstehen vier 4stellige Summanden, die stellenrichtig zu addieren sind.

$$9 \cdot 11 = 99$$

Beispiel:

$$\begin{array}{r}
 1\ 0\ 0\ 1 \cdot 1\ 0\ 1\ 1 \\
 \hline
 1\ 0\ 0\ 1 \\
 1\ 0\ 0\ 1 \\
 0\ 0\ 0\ 0 \\
 1\ 0\ 0\ 1 \\
 \hline
 1\ 1\ 0\ 0\ 0\ 1\ 1
 \end{array}
 \quad \begin{array}{l}
 \rightarrow 1. \text{ Summand} \\
 \rightarrow 2. \text{ Summand} \\
 \rightarrow 3. \text{ Summand} \\
 \rightarrow 4. \text{ Summand} \\
 \rightarrow \text{Ergebnis}
 \end{array}$$

Eine 4-Bit-Parallel-Multiplikationsschaltung ist in Bild 14.28 dargestellt. Der Schaltungsaufwand ist verhältnismäßig groß. Er steigt mit größer werdender Stellenzahl der Dualzahlen sehr stark an. Für die Multiplikation von zwei 8-Bit-Dualzahlen sind 64 UND-Glieder und acht 8-Bit-Parallel-Addierschaltungen erforderlich.

Bild 14.27 2-Bit-Parallel-Multiplikationschaltung

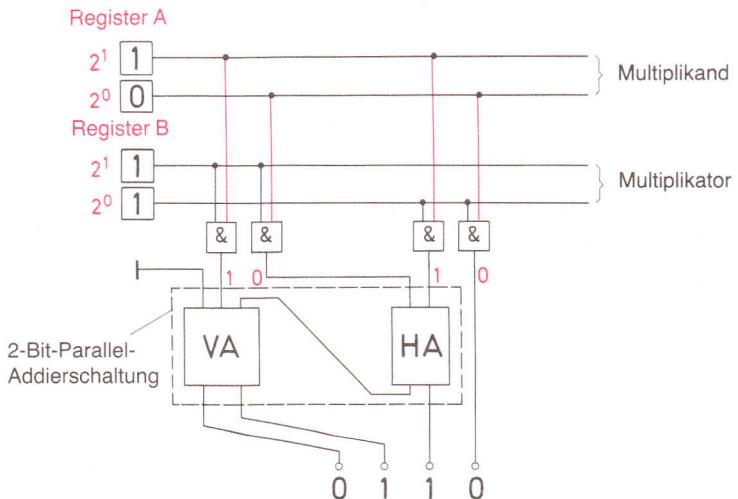
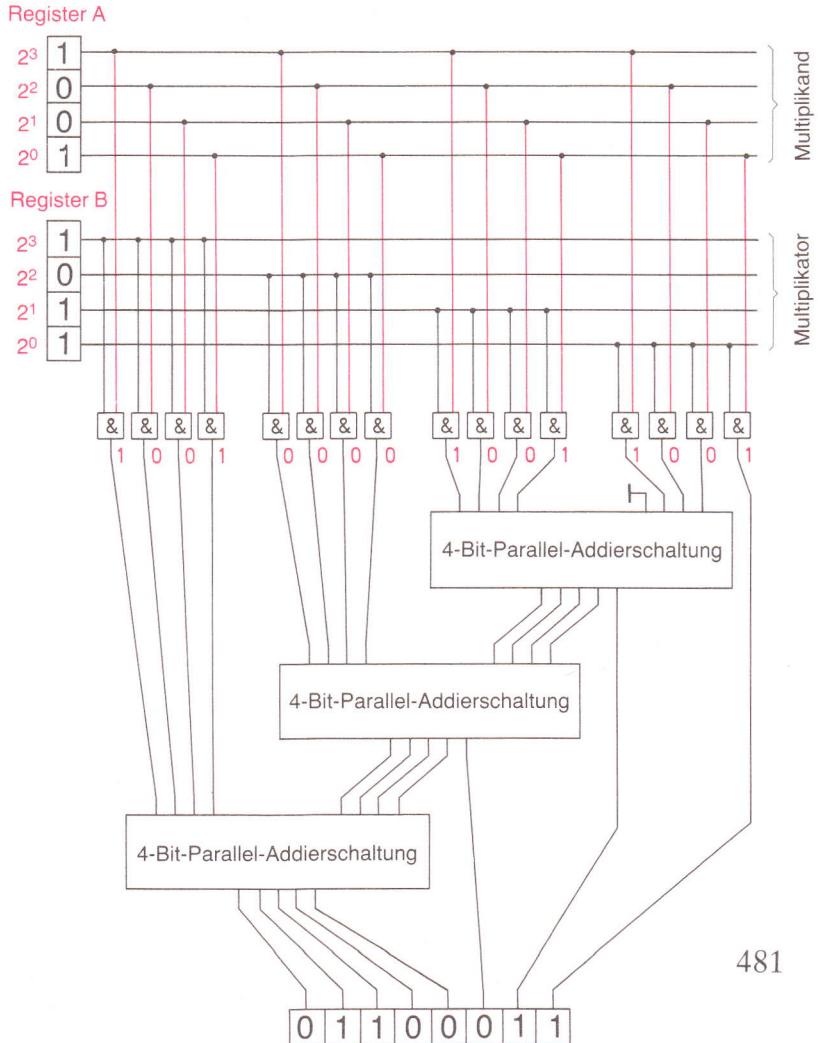


Bild 14.28
4-Bit-Parallel-Multiplikations-
schaltung



14.7.2 Serielle Multiplikationsschaltung

Der Aufbau einer seriellen Multiplikationsschaltung ist in Bild 14.29 dargestellt. Die Schaltung ist für die Multiplikation 4stelliger Dualzahlen geeignet. Multiplikand und Multiplikator werden in je ein Register eingegeben. Die eigentliche Multiplikation erfolgt durch das rot eingezzeichnete UND-Glied, das als 1-Bit-Multiplizierer arbeitet. Der 1-Bit-Volladdierer addiert das Multiplikationsergebnis stellenrichtig zu einem bereits vorliegenden Ergebnis, das sich im Ergebnisregister befindet.

Vor Beginn des Multiplikationszyklus ist das Ergebnisregister E leer. Die wertniedrigste Stelle (2^0) des Multiplikators wird an den unteren Eingang des UND-Gliedes gelegt. Diese Stelle enthält in Bild 14.29 eine 1. Mit dieser 1 werden jetzt die Stellen des Multiplikanden nacheinander multipliziert, beginnend mit der Stelle 2^0 . Der Inhalt des Multiplikandenregisters (A) wird durch das Taktsignal T_1 weitergetaktet. Die einzelnen Ziffern kommen nacheinander an den oberen Eingang des roten UND-Gliedes und werden multipliziert. Das Multiplikandenregister ist als Ringregister geschaltet. Nach vier Takten ist die ursprüngliche Stellung der Ziffern im Multiplikandenregister wiederhergestellt. Der erste Summand (1001) wurde gebildet und befindet sich im Ergebnisregister.

Jetzt wird ein Takt T_2 gegeben. Der Inhalt des Multiplikatorregisters (B) wird um eine Stelle nach rechts verschoben. Nun erfolgt die Multiplikation des Multiplikanden mit der 2. Stelle des Multiplikators (2^1) in 4 Takten. Gleichzeitig wird ein Takt T_3 auf das Ergebnisregister E gegeben. Der Inhalt des Ergebnisregisters wird um eine Stelle nach rechts verschoben. Die wertniedrigste Ziffer des Ergebnisregisters wird in das B-Register übernommen.

Nun erfolgt die Multiplikation des Multiplikanden mit der 2. Stelle des Multiplikators (2^1) in 4 Takten. Das entstehende Multiplikationsergebnis wird zum Inhalt des Ergebnisregisters addiert. Der neu entstehende 2. Summand wird zum bereits vorhandenen 1. Summanden addiert. Dabei wird die wertniedrigste Stelle des 1. Summanden, die sich ja im B-Register befindet, ausgespart.

Beispiel:

$$\begin{array}{r}
 1\ 0\ 0\ 1 \cdot 1\ 0\ 1\ 1 \\
 \hline
 1\ 0\ 0 \boxed{1} \\
 1\ 0\ 0\ 1 \\
 \hline
 1\ 1\ 0\ 1
 \end{array}$$

1. Summand
2. Summand

Neuer Inhalt des E-Registers → Ziffer befindet sich im B-Register

Durch einen weiteren Takt T_2 und einen weiteren Takt T_3 wird einmal der Multiplikator um eine weitere Stelle nach rechts verschoben, zum anderen wird wiederum der Inhalt der wertniedrigsten Stelle des E-Registers an das B-Register abgegeben.

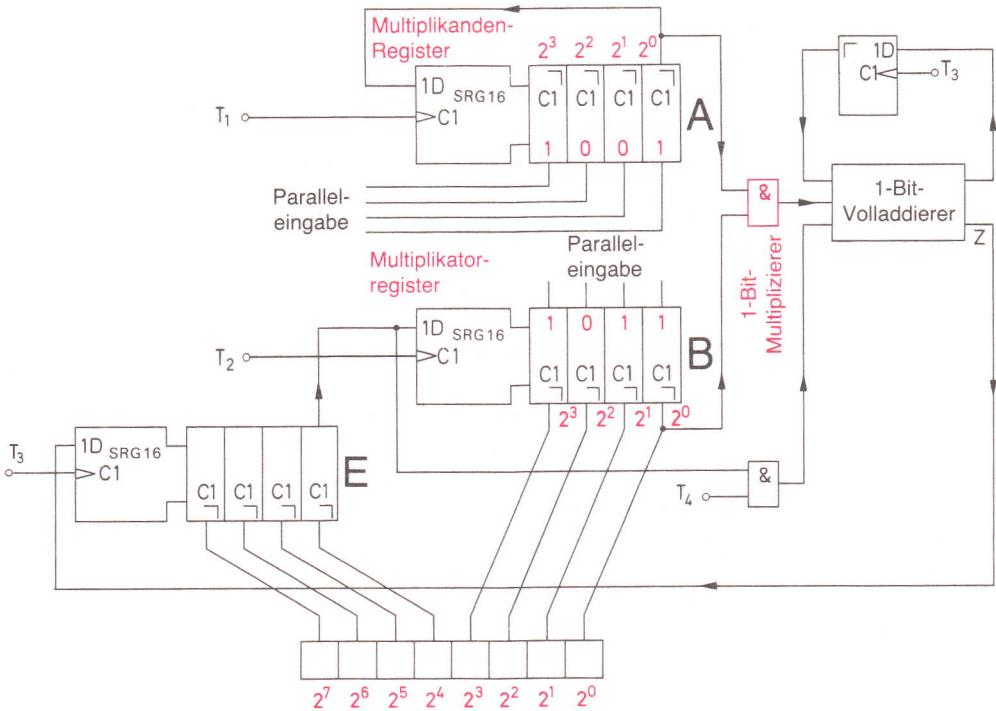


Bild 14.29 Aufbau einer seriellen 4-Bit-Multiplikationsschaltung

Danach erfolgt die Multiplikation des Multiplikanden mit der 3. Stelle des Multiplikators (2^2) in 4 Takten. Das entstehende Multiplikationsergebnis wird zum Inhalt des Ergebnisregisters addiert.

Der Ablauf setzt sich in gleicher Weise wie bereits beschrieben fort. Weitere Verschiebung des Multiplikators und des Inhalts des Ergebnisregisters um eine Stelle. Aufnahme der wertniedrigsten Ziffer des E-Registers in das B-Register. Multiplikation des Multiplikanden mit der 4. Stelle des Multiplikators (2^3) in 4 Takten. Addition des neuen Multiplikationsergebnisses zum Inhalt des Ergebnisregisters.

Durch einen weiteren Takt T_2 und einen weiteren Takt T_3 werden die Inhalte der Register B und E um eine weitere Stelle nach rechts verschoben und gleichzeitig die wertniedrigste Stelle des E-Registers in das B-Register übernommen.

Nun ist der Multiplikationszyklus beendet. Das Endergebnis steht in den Registern E und B und kann an die Ausgänge abgegeben werden. Die Wertigkeit ist an den Ausgängen in Bild 14.29 angegeben.

14.8 Lernziel-Test

1. Geben Sie die Wahrheitstabelle eines Halbaddierers an, und entwickeln Sie aus der Wahrheitstabelle die Schaltung. Die Schaltung ist unter Verwendung von Grundgliedern zu skizzieren.
2. Wodurch unterscheidet sich ein Volladdierer von einem Halbaddierer?
3. Aus zwei Halbaddierern und einem ODER-Glied soll ein Volladdierer aufgebaut werden. Wie sind die Bausteine zusammenzuschalten? Geben Sie das Schaltbild an.
4. Erklären Sie den Begriff «8-Bit-Parallel-Addierschaltung».
5. Wie ist eine serielle Addierschaltung im Prinzip aufgebaut?
6. Skizzieren Sie die Schaltung eines Halbsubtrahierers, und erläutern Sie die Arbeitsweise dieser Schaltung.
7. Mit drei Volladdierern und beliebigen Verknüpfungsgliedern soll ein 3-Bit-Addier-Subtrahier-Werk hergestellt werden. Das Schaltbild ist zu zeichnen.
8. Welche Verknüpfungen muß ein 1-Bit-Multiplizierer erzeugen können? Geben Sie die Wahrheitstabelle des 1-Bit-Multiplizierers an.
9. Wie arbeitet eine 3-Bit-Parallel-Multiplikationsschaltung? Das Prinzip ist zu erläutern. Wieviel 1-Bit-Multiplizierer und wieviel Additionsschaltungen sind erforderlich? Von welcher Art müssen die Additionsschaltungen sein?
10. Erklären Sie das Prinzip einer seriellen Multiplikationsschaltung.

15 Mikroprozessoren und Mikrocomputer

15.1 Der Mikroprozessor als Universalschaltung

Könnte man eine Schaltung bauen, die addieren, subtrahieren und multiplizieren kann und die darüber hinaus alle nur möglichen logischen Verknüpfungen von binären Signalen auszuführen in der Lage ist? Die eingegebenen Signale – auch Daten genannt – müßten zeitlich nacheinander bestimmten gewünschten Bearbeitungen unterzogen werden können. Die zeitliche Folge der Bearbeitungen, also z.B. die Folge der durchzuführenden logischen Verknüpfungen, wäre vor Arbeitsbeginn der Schaltung in einem Programm festzulegen.

Eine solche Schaltung wäre universell verwendbar. Sie könnte logische Schaltungen aller Art ersetzen. Eine benötigte Verknüpfungsschaltung müßte nicht mehr aus verschiedenen Verknüpfungsgliedern «zusammengelötet» werden. Man könnte die Universalschaltung nehmen und sie so programmieren, daß sie die gewünschte Verknüpfung erzeugt.

Der Aufbau dieser Universalschaltung wäre sicherlich verhältnismäßig kompliziert, die Herstellung der Schaltung also vermutlich teuer. Die moderne Technik integrierter Schaltungen gibt jedoch die Möglichkeit, auch komplizierte Schaltungen preisgünstig herzustellen.

Überlegungen dieser Art standen am Anfang der Entwicklung solcher Universalschaltungen, die heute *Mikrocomputer* genannt werden. Hauptteil eines Mikrocomputers ist der *Mikroprozessor*. Mikroprozessoren verschiedener Typen werden zur Zeit als integrierte Schaltungen verhältnismäßig preisgünstig angeboten.

Komplizierte Steuerschaltungen, deren Aufbau aus Verknüpfungsgliedern und Flipflops außerordentlich teuer wäre, lassen sich mit Mikrocomputern sehr kostengünstig aufbauen.

15.2 Arithmetisch-logische Einheit (ALU)

Bei der Entwicklung einer Universalschaltung ist es zweckmäßig, von dem erweiterten 4-Bit-Addier-Subtrahier-Werk Bild 14.23 auszugehen, das im vorhergehenden Kapitel näher erläutert wurde. Mit dieser Schaltung können die Eingangssignale A und B wahlweise addiert und subtrahiert werden.

Zusätzlich ist es erforderlich, daß die Signale A und B
einer UND-Verknüpfung,
einer ODER-Verknüpfung,
einer EXKLUSIV-ODER-Verknüpfung

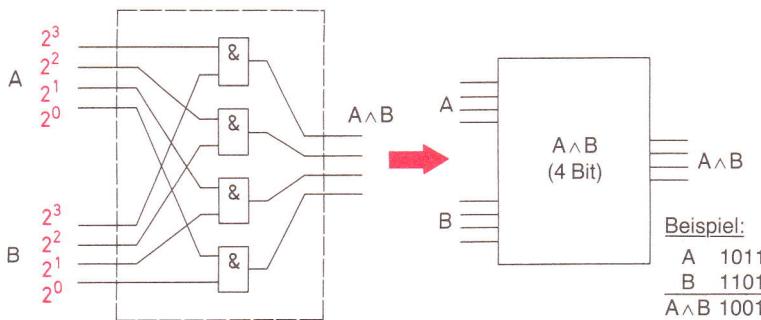


Bild 15.1 Schaltung zur Erzeugung einer UND-Verknüpfung von zwei 4-Bit-Wörtern

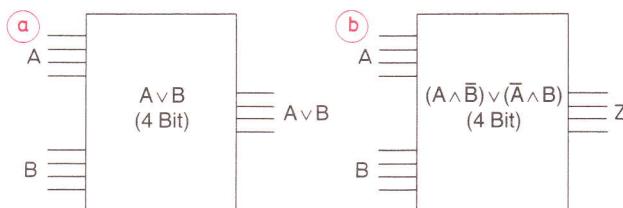


Bild 15.2 Schaltungen zur Erzeugung einer 4-Bit-ODER-Verknüpfung (a) und einer 4-Bit-EXKLUSIV-ODER-Verknüpfung (b)

unterzogen werden können. Die Schaltung zur Erzeugung einer 4-Bit-UND-Verknüpfung ist in Bild 15.1 angegeben. Entsprechend aufgebaut sind die Schaltungen zur Erzeugung einer 4-Bit-ODER-Verknüpfung und einer 4-Bit-EXKLUSIV-ODER-Verknüpfung (Bild 15.2).

Eine Schaltung, die zwei n -Bit-Wörter wahlweise addieren, subtrahieren, UND-verknüpfen, ODER-verknüpfen und EXKLUSIV-ODER-verknüpfen kann, wird Arithmetisch-logische Einheit – abgekürzt ALU – genannt.

Eine ALU für 4-Bit-Wörter besteht also aus einem erweiterten Addier-Subtrahier-Werk gemäß Bild 14.23, einer Schaltung zur Erzeugung einer 4-Bit-UND-Verknüpfung, einer Schaltung zur Erzeugung einer 4-Bit-ODER-Verknüpfung und aus einer Schaltung zur Erzeugung einer 4-Bit-EXKLUSIV-ODER-Verknüpfung. Die vier 4-Bit-Ausgänge werden über vier Multiplexer (s. Kapitel 11) wahlweise auf den 4-Bit-Z-Ausgang gegeben. Das Addier-Subtrahier-Werk hat außerdem noch einen Übertragsausgang \bar{U} , der herausgeführt wird (Bild 15.3).

Arithmetisch-logische Einheiten werden als integrierte Schaltungen für 4 Bit, 6 Bit, 8 Bit und 16 Bit hergestellt. Am häufigsten werden 8-Bit-ALU verwendet. Die Darstellung als Block (Bild 15.4) ist üblich. Da eine 8-Bit-ALU grundsätzlich 8 A-Eingänge, 8 B-Eingänge und 8 Z-Ausgänge hat, können jeweils 8 Leitungen durch einen Leitungsstrich dargestellt werden. Die Schaltbilder werden dadurch übersichtlicher (Bild 15.4).

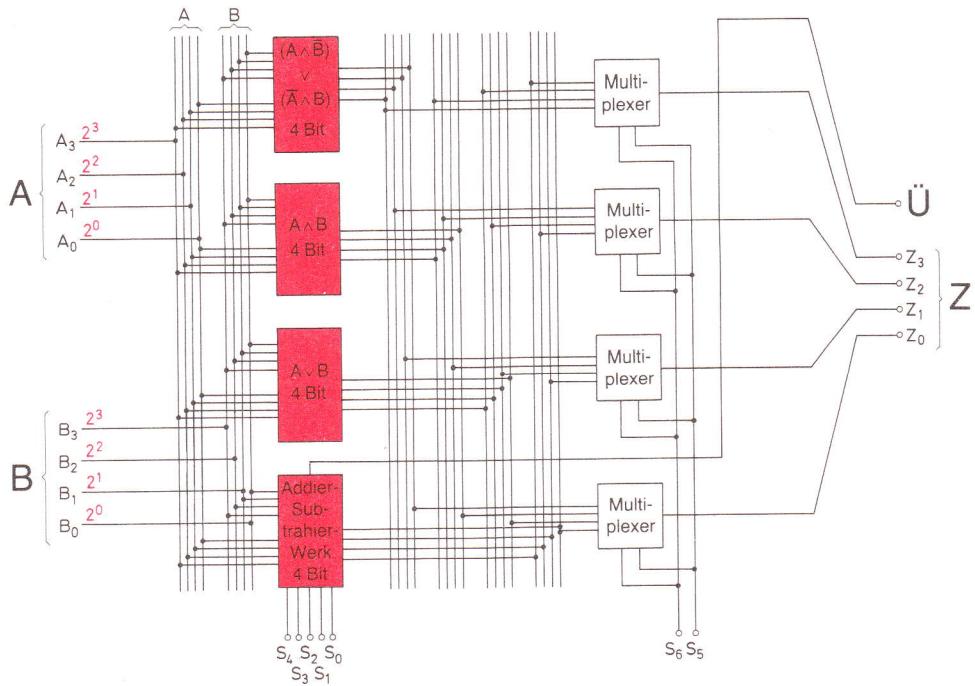
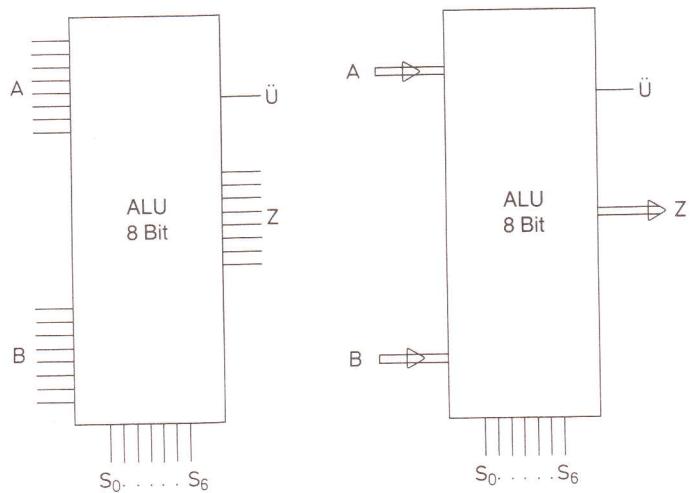


Bild 15.3 Schaltbild einer Arithmetisch-logischen Einheit für 4 Bit

Bild 15.4 Block-darstellungen einer 8-Bit-ALU



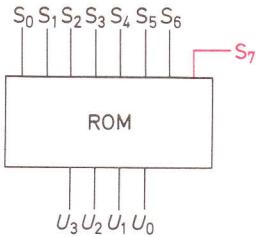


Bild 15.5 Umkodierschaltung mit ROM zur Umkodierung von 7 auf 4 Steuereingänge

Befehl				Funktion
U_3	U_2	U_1	U_0	
1	0	0	0	A
2	0	0	1	$\bar{1}$
3	0	0	1	\bar{A}
4	0	0	1	B
5	0	1	0	0
6	0	1	0	$A + 1$
7	0	1	1	$A - 1$
8	0	1	1	$A + B$
9	1	0	0	$A - B$
10	1	0	1	$A \wedge B$
11	1	0	1	$A \vee B$
12	1	0	1	$(A \wedge \bar{B}) \vee (\bar{A} \wedge B)$
13	1	1	0	-1
14	1	1	0	
15	1	1	1	
16	1	1	1	

Bild 15.6 Befehle einer ALU

Über die sieben Steuerleitungen S_0 bis S_6 können insgesamt $2^7 = 128$ verschiedene Steuerbefehle gegeben werden. Von diesen Steuerbefehlen werden nur 13 benötigt. Es ist also sinnvoll, eine Umkodierung vorzunehmen. Diese erfolgt mit Hilfe eines ROM (s. Kapitel 12, Abschnitt 12.4). Man verwendet 4 Steuereingänge (Bild 15.5). Mit diesen lassen sich 16 verschiedene Befehle darstellen. 3 mögliche Befehle bleiben ungenutzt. Die Befehle sind in Bild 15.6 aufgeführt. Einige Befehle erfordern die Unterdrückung des Übertrages \bar{U} . Zu diesem Zweck hat das ROM einen Ausgang S_7 , der immer dann 0-

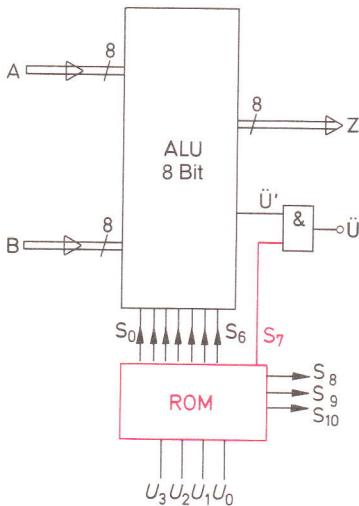


Bild 15.7 ALU mit ROM zur Umkodierung und UND-Glied zur Übertragsunterdrückung

Signal führt, wenn ein Übertrag nicht am Ausgang \dot{U} erscheinen soll. Die Ausgänge S_8 und S_9 werden für Zusatzsteuerungen benötigt (s. Abschnitt 15.3 und 15.4).

Das Blockschaltbild einer 8-Bit-ALU mit Umkodierungs-ROM und Übertragsunterdrückung zeigt Bild 15.7.

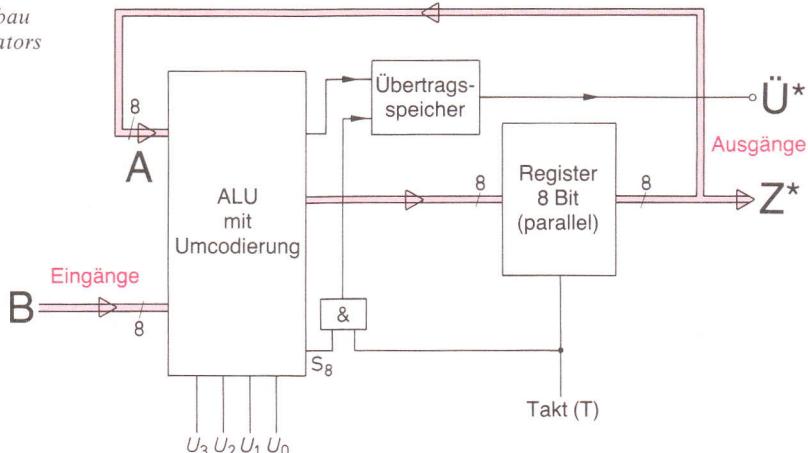
15.3 Akkumulator

Ein Akkumulator besteht aus einer ALU mit Umkodierung, aus einem Register und aus einem 1-Bit-Speicher für den Übertrag. Die Dateneingabe erfolgt nur über die B-Eingänge. Die A-Eingänge sind mit den Ausgängen des Registers verbunden (Bild 15.8). Register und Übertragsspeicher sind taktgesteuert. Die Taktsteuerung des Übertragsspeichers kann über ein UND-Glied gesperrt werden. Das S_8 -Signal wird dem Umkodierungs-ROM entnommen.

Sollen zwei 8-Bit-Wörter addiert werden, wird das erste 8-Bit-Wort auf die B-Eingänge gegeben. Es wird über die ALU dem Register zugeführt und mit dem nächsten Takt übernommen. Das Register ist mit 8 Flipflops aufgebaut, in denen das 8-Bit-Wort parallel gespeichert wird. Es steht an den 8 Ausgängen des Registers zur Verfügung und liegt gleichzeitig an den 8 A-Eingängen.

Nach der Einspeicherung des ersten 8-Bit-Worts wird das zweite 8-Bit-Wort auf die B-Eingänge gegeben. Die beiden zu addierenden 8-Bit-Worte liegen jetzt an den A- und den B-Eingängen der ALU. Die ALU addiert auf Befehl beide 8-Bit-Worte und bietet das Ergebnis dem Register an. Das Register übernimmt das Ergebnis mit dem nächsten Takt. Ein eventuell entstehender Übertrag wird ebenfalls taktgesteuert in den Übertragsspeicher übernommen. Der Übertragsspeicher wird auch Übertrags-Flag genannt (Flag, engl.: Flagge, Kennzeichen). Das Ergebnis der Addition steht nun an den Z^* -Ausgängen und am Ausgang \dot{U}^* zur Verfügung.

Bild 15.8 Aufbau eines Akkumulators



Eine UND-Verknüpfung von zwei 8-Bit-Wörtern erfolgt auf ähnliche Weise. Die ALU erhält lediglich statt des Additionsbefehls den Befehl, eine UND-Verknüpfung durchzuführen.

Die 13 Befehle einer ALU gelten entsprechend auch für den Akkumulator. Statt der A-Signale ist jedoch stets der Akkuinhalt zu berücksichtigen. Den Befehlen wird eine leicht merkbare Abkürzung zugeordnet, die auf die englische Befehlsbezeichnung hindeutet. Diese Abkürzung ist der symbolische Befehlsname. Alle Befehle eines Akkumulators sind in einer sogenannten Befehlsliste zusammengefaßt (Bild 15.9).

Befehl-Nr. Dez.	Befehl-Nr. Hexa.	Befehl				Befehls- name	Befehlsbeschreibung	Übertrags- speicher
		U_3	U_2	U_1	U_0			
0	0	0	0	0	0	NOP	Keine Operation	T
1	1	0	0	0	1	SP1	Im Akku ist der Inhalt 1 einzuspeichern	T
2	2	0	0	1	0	CMA	Der Akkuinhalt ist zu negieren	-
3	3	0	0	1	1	LDA	B-Signal soll in den Akku geladen werden	-
4	4	0	1	0	0	CLA	Der Akkuinhalt ist zu löschen	-
5	5	0	1	0	1	INC	Der Akkuinhalt ist um 1 zu erhöhen	T
6	6	0	1	1	0	DEC	Der Akkuinhalt ist um 1 zu vermindern	T
7	7	0	1	1	1	ADD	Addiere B-Signal zum Akkuinhalt	T
8	8	1	0	0	0	SUB	Subtrahiere B-Signal vom Akkuinhalt	T
9	9	1	0	0	1	AND	UND-Verknüpfung von Akkuinhalt und B-Signal	T
10	A	1	0	1	0	IOR	ODER-Verknüpfung von Akkuinhalt und B-Signal	T
11	B	1	0	1	1	XOR	Exklusiv-ODER-Verknüpfung von Akkuinhalt und B-Signal	T
12	C	1	1	0	0	SM1	Im Akku ist der Inhalt -1 zu speichern	T
13	D	1	1	0	1		-	-
14	E	1	1	1	0		-	-
15	F	1	1	1	1		-	-

T: Übertragsspeicher wird getaktet

-: Übertragsspeicher wird nicht getaktet

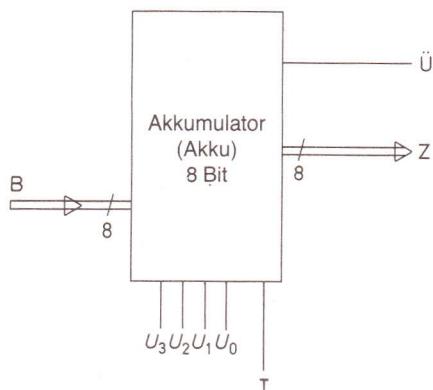
Bild 15.9 Befehlsliste eines Akkumulators

Der Übertragsspeicher wird bei einigen Befehlen nicht getaktet. Er behält also den vorher vorhandenen Informationsinhalt bei. Das bringt einige Vorteile. Wird der Übertragsspeicher getaktet, entsteht aber bei der Ausführung des Befehls kein Übertrag, ist der Übertragsspeicher nach dem Takt gelöscht.

Betrachten wir den Ablauf der Subtraktion $X - Y$. Die Zahl X wird an die B-Eingänge gelegt. Der Befehl LDA (0011) hat zur Folge, daß die Zahl X in das Register geladen wird. Jetzt wird die Zahl Y an die B-Eingänge gelegt. Zur Subtraktion ist der Befehl SUB (1000) erforderlich. Die Subtraktion wird ausgeführt, das Ergebnis wird in das Register gespeichert und kann an den Ausgängen Z^* abgenommen werden. War Y größer als X, ist die Ergebniszahl negativ. \dot{U}^* führt dann 1-Signal. Bei positiver Ergebniszahl führt \dot{U}^* 0-Signal.

Die Blockdarstellung eines Akkumulators zeigt Bild 15.10.

Bild 15.10 Blockdarstellung eines 8-Bit-Akkumulators



15.4 Akkumulator mit Datenspeicher

Der nächste Schritt auf dem Wege zu der gesuchten Universalsschaltung, zum sogenannten Mikroprozessor, ist der Akkumulator mit Datenspeicher. Als Datenspeicher wird ein statischer Schreib-Lese-Speicher (RAM) verwendet (s. Kapitel 12, Abschnitt 12.3.1).

Ein Akkumulator mit Datenspeicher kann Zwischenergebnisse im Datenspeicher ablegen und sie bei Bedarf wieder herausholen.

Den Aufbau eines Akkumulators mit Datenspeicher zeigt Bild 15.11. Die acht Ausgänge des Akkumulators führen nach außen (Z^*) und sind außerdem mit den Eingängen des RAM verbunden. Ausgangsdaten des Akkumulators können in das RAM übernommen werden. Die Einspeicherung in das RAM erfolgt taktgesteuert. Das im Akkumulator enthaltene Umkoppler-ROM liefert über die Steuerleitung S_9 1-Signal und gibt damit den Takt über das UND-Glied frei, so daß die Einspeicherung erfolgen kann.

Das RAM kann verschiedene große Speicherkapazität haben. In Bild 15.11 ist ein RAM mit 16 Speicherzellen zu je 8 Bit eingezeichnet. Die 16 Speicherzellen müssen Adressen erhalten (s. Kapitel 12). Mit 4-Bit-Einheiten lassen sich 16 verschiedene Adressen herstellen (0000 bis 1111). Da die Adressen vierstellig sind, werden 4 Adresseingänge benötigt. Diese erhalten die Bezeichnungen A_0 bis A_3 (s. Bild 15.11).

Im RAM gespeicherte Daten können nach Wunsch wieder ausgespeichert und dem Akkumulator zugeführt werden. Die B-Eingänge des Akkumulators werden über einen Datenselektor (Multiplexer) angesteuert. Erhält der Datenselektor 1-Signal über die Steuerleitung S_{10} des im Akkumulator enthaltenen Umkopplers-ROM, schaltet er die B^* -Eingänge auf die B-Eingänge des Akkumulators. Liegt auf der Steuerleitung S_{10} 0-Signal, sind die Ausgänge des RAM mit den B-Eingängen des Akkumulators verbunden.

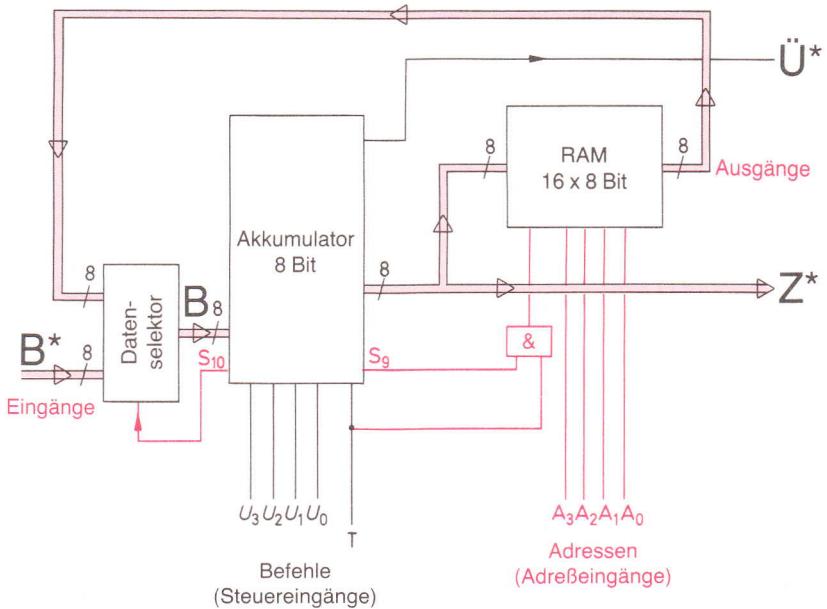


Bild 15.11 Aufbau eines Akkumulators mit Datenspeicher

Zu den 13 Befehlen des Akkumulators ohne Datenspeicher (Bild 15.9) müssen zwei weitere Befehle hinzukommen. Benötigt wird einmal ein Befehl, durch den der Daten selektor auf die Eingänge B^* umgeschaltet wird. Dieser Befehl muß das Anlegen des 1-Signals an die Steuerleitung S_{10} auslösen. Ihm wird die Bit-Kombination Nr. 13 (1101) zugeordnet (Bild 15.12). Sein Befehlsname ist INP.

Benötigt wird weiterhin ein Befehl, der das Einspeichern des Akkuinhalts in eine Speicherzelle des RAM auslöst. Dieser Befehl bekommt die Bit-Kombination Nr. 14 (1110) und den Namen STA. Er muß von dem im Akkumulator enthaltenen Umkodierungs-ROM so verarbeitet werden, daß 1-Signal auf der Steuerleitung S_9 auftritt und der Takt zum RAM dadurch freigegeben wird (Bild 15.11). Dieser Befehl muß stets mit einer Adresse verbunden sein.

Beim Akkumulator mit Datenspeicher bestehen die einzelnen Befehle aus 8-Bit-Wörtern. Vier Bit werden für das Steuersignal (U_0 bis U_3) und vier Bit werden für die Adresse (A_0 bis A_3) benötigt.

Befehl Nr. Dez.	Befehl Hexa.	Befehl				Befehls- name	Befehlsbeschreibung	Übertrags- speicher
		U_3	U_2	U_1	U_0			
13	D	1	1	0	1	INP	Eingangssignal B* ist in den Akku zu laden	–
14	E	1	1	1	0	STA	Der Akkuinhalt ist in die Speicherzelle mit der Adresse A ₃ A ₂ A ₁ A ₀ zu speichern	–

Bild 15.12 Zusätzliche Befehle des Akkumulators mit Datenspeicher

Die Steuer-Bit bilden den Operationsteil, die Adreß-Bit bilden den Adress- teil eines Befehls.

Vor Beginn eines Arbeitsablaufs, also vor Beginn einer Rechenoperation oder einer Steuerung, müssen die einzelnen auszuführenden Befehle und ihre zeitliche Reihenfolge genau festgelegt werden.

Eine festgelegte Befehlsfolge wird Programm genannt.

Man muß dem Akkumulator mit Datenspeicher also schrittweise mitteilen, was er im einzelnen zu tun hat. Zuerst wird der 1. Befehl eingegeben, dann der 2., dann der 3. usw., bis alle Befehle ausgeführt sind. Dann muß das gewünschte Ergebnis in richtiger Form vorliegen. Liegt es nicht vor, war das Programm falsch.

Bei jedem erneuten Arbeitsablauf müssen die Befehle wieder neu eingegeben werden. Das ist sehr mühsam und zeitraubend und in der Praxis kaum durchführbar. Man stelle sich eine Werkzeugmaschinensteuerung vor, bei der während eines jeden Drehvorganges etwa 40 oder mehr 8-Bit-Befehle über Tasten eingegeben werden müßten! Niemand würde eine solche Maschine kaufen.

15.5 Programmgesteuerter vereinfachter Rechner

Der Akkumulator mit Datenspeicher läßt sich ganz wesentlich durch eine Programmsteuerung verbessern. Die Programmsteuerung besteht aus einem Programmspeicher, einer Ladeeinrichtung und einem Befehlszähler. Durch Einsatz dieser zusätzlichen Bausteine wird aus dem Akkumulator mit Datenspeicher ein programmgesteuerter vereinfachter Rechner (Bild 15.13).

Vor Beginn eines Arbeitsvorganges wird das Programm geladen. Der 1. Befehl wird an die Eingänge B^* gelegt. Durch einen Impuls auf den Anschluß T_2 wird ein Ladesignal ausgelöst. Der Befehl wird in den Programmspeicher eingespeichert. Dann wird der 2. Befehl an die Eingänge B^* gelegt und auf die gleiche Weise eingespeichert. Nach dem 2. Befehl folgt der 3. – und so fort. Die Befehle werden also in der Reihenfolge geladen, in der sie später auszuführen sind. Hierbei darf es keine Verwechslungen geben.

Sind alle Befehle des Programms in den Programmspeicher geladen, kann mit der Ausführung des Programms begonnen werden. Das Startsignal wird durch einen Impuls auf den Eingang S des Befehlszählers gegeben. Jetzt veranlaßt der Befehlszähler die Ausgabe des ersten Befehls aus dem Programmspeicher. Der Befehl wird an die Eingänge U_0 bis U_3 und A_0 bis A_3 gelegt und ausgeführt. Danach wird durch den Befehlszähler die Ausgabe des zweiten Befehls veranlaßt. Nach Ausführung des zweiten Befehls wird der dritte Befehl ausgegeben und ausgeführt.

Das gesamte Programm wird Befehl nach Befehl abgearbeitet. Nach Ausführung aller

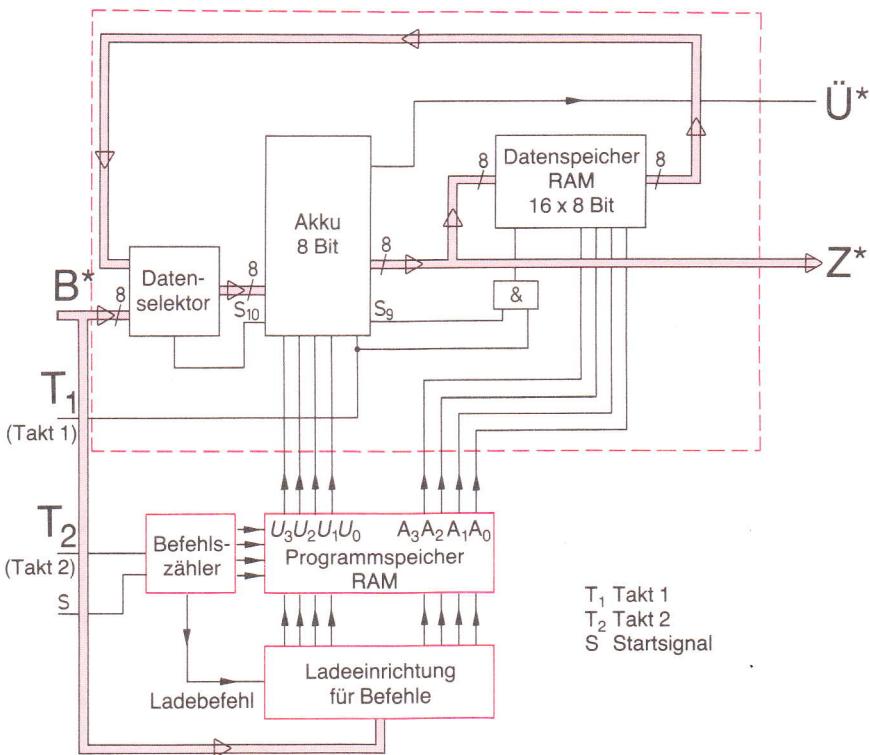


Bild 15.13 Aufbau eines programmgesteuerten einfachen Rechners

Befehle muß der Befehlszähler stillgesetzt werden. Hierzu ist ein weiterer Befehl erforderlich, der sogenannte HALT-Befehl (HLT). Die 4-Bit-Kombination Nr. 15 (1111) ist noch frei (siehe Bild 15.9 und Bild 15.12). Diese Bit-Kombination wird dem Befehl HLT zugeteilt.

Der programmgesteuerte vereinfachte Rechner verfügt nun über 15 Befehle, die in Bild 15.14 ausgeführt sind.

Der programmgesteuerte vereinfachte Rechner erlaubt die Durchführung komplizierter Rechen- und Steuervorgänge. Die Befehle werden streng in der Reihenfolge der Eingabe ausgeführt. Das ist nicht immer erwünscht. Häufig möchte man einen Befehlssprung haben, d.h., man möchte z.B. nach dem 35. Befehl wieder zum 10. Befehl zurückspringen und die folgenden Befehle erneut abarbeiten. Oder man möchte nach dem 20. Befehl zum 45. Befehl springen. Diese Möglichkeiten bietet der programmgesteuerte vereinfachte Rechner nicht.

Befehl Nr. Dez.	Befehl Hexa.	Befehl								Befehlsname	Befehlsbeschreibung
		Operationsteil				Adresse Teil					
		U_3	U_2	U_1	U_0	A_3	A_2	A_1	A_0		
0	0	0	0	0	0	–	–	–	–	NOP	Keine Operation
1	1	0	0	0	1	–	–	–	–	SP1	In den Akku ist der Inhalt 1 zu speichern
2	2	0	0	1	0	–	–	–	–	CMA	Der Inhalt des Akkus ist zu negieren
3	3	0	0	1	1	[A]	[A]	[A]	[A]	LDA	Der Inhalt der Datenspeicherzelle mit der Adresse AAAA soll in den Akku geladen werden
4	4	0	1	0	0	–	–	–	–	CLA	Der Akkuinhalt ist zu löschen
5	5	0	1	0	1	–	–	–	–	INC	Der Akkuinhalt ist um 1 zu erhöhen
6	6	0	1	1	0	–	–	–	–	DEC	Der Akkuinhalt ist um 1 zu vermindern
7	7	0	1	1	1	[A]	[A]	[A]	[A]	ADD	Der Inhalt der Datenspeicherzelle mit der Adresse AAAA soll zum Akkuinhalt addiert werden
8	8	1	0	0	0	[A]	[A]	[A]	[A]	SUB	Der Inhalt der Datenspeicherzelle mit der Adresse AAAA ist vom Akkuinhalt zu subtrahieren
9	9	1	0	0	1	[A]	[A]	[A]	[A]	AND	UND-Verknüpfung zwischen dem Inhalt der Speicherzelle mit der Adresse AAAA und dem Akkuinhalt
10	A	1	0	1	0	[A]	[A]	[A]	[A]	IOR	ODER-Verknüpfung zwischen dem Inhalt der Speicherzelle mit der Adresse AAAA und dem Akkuinhalt
11	B	1	0	1	1	[A]	[A]	[A]	[A]	XOR	Exklusiv-ODER-Verknüpfung zwischen dem Inhalt der Speicherzelle mit der Adresse AAAA und dem Akkuinhalt
12	C	1	1	0	0	–	–	–	–	SM1	Im Akku ist der Inhalt –1 zu speichern
13	D	1	1	0	1	–	–	–	–	INP	Eingangssignal B* ist in den Akku zu laden
14	E	1	1	1	0	[A]	[A]	[A]	[A]	STA	Der Akkuinhalt ist in die Speicherzelle mit der Adresse AAAA zu speichern
15	F	1	1	1	1	–	–	–	–	HLT	Der Befehlszähler ist anzuhalten

– keine Adresse erforderlich

[A] Platzhalter für ein Adressen-Bit

Bild 15.14 Befehlsliste eines programmgesteuerten vereinfachten Rechners (die Befehle gelten auch für den Akkumulator mit Datenspeicher mit Ausnahme des Befehls Nr. 15)

15.6 Mikroprozessorbausteine

Entwickelt man den programmgesteuerten Rechner nach Bild 15.13 weiter, so daß er in der Lage ist, auch Programmsprünge durchzuführen, erhält man eine mögliche Mikrocomputerschaltung, die als Universalschaltung für alle Steuerungs- und Rechnungsaufgaben geeignet ist. Der Kernteil dieser Schaltung mit ALU, Datenspeicher, Befehlszähler, Registern und Steuereinrichtungen wird im sogenannten Mikroprozessor zusammengefaßt.

Ein Mikroprozessor ist eine unvollständige programmgesteuerte Rechnerg schaltung, die besonders für Steuerungszwecke geeignet ist.

Das Steuern steht im Vordergrund, nicht das Rechnen. Doch zur Durchführung von Steuerungen sind auch Rechnungen erforderlich.

15.6.1 Mikroprozessortypen

Bei der Mikroprozessorentwicklung kann man unterschiedliche Wege gehen und zu unterschiedlichen Ergebnissen kommen. Es gibt daher viele verschiedene Mikroprozessortypen mit teilweise stark voneinander abweichenden Eigenschaften. Zur Zeit sind etwa 60 Mikroprozessortypen auf dem Markt. Sie werden alle als 1-Chip-Mikroprozessoren hergestellt, d.h., sie bestehen aus einer einzigen integrierten Schaltung.

Mikroprozessoren werden ausschließlich als integrierte Schaltungen her gestellt.

Verwendet wird üblicherweise ein Dual-In-Line-Gehäuse mit bis zu 40 Anschlüssen. Mikroprozessoren unterscheiden sich vor allem durch folgende Eigenschaften:

1. Wortlänge

Die Wortlänge gibt an, wie viele Bit parallel verarbeitet werden können, also wie viele Bit die Eingangs- und Ausgangsgrößen haben. Es gibt 4-Bit-, 8-Bit-, 16-Bit- und 32-Bit-Mikroprozessoren.

2. Rechengeschwindigkeit

Bei der Rechengeschwindigkeit vergleicht man die sogenannten Zykluszeiten miteinander. Unter der Zykluszeit versteht man meist die Zeit, die für die Paralleladdition von zwei Dualzahlen und für das Ein- und Ausspeichern dieser Zahlen erforderlich ist. Üblich sind Zykluszahlen von 10 µs bis zu 0,1 µs.

3. Technologie (Schaltkreisfamilie)

Mikroprozessoren werden überwiegend in MOS-Technik hergestellt. In dieser Technologie ist die höchste Integrationsdichte erreichbar. Die Schaltungen können verhältnis-

mäßig kompliziert aufgebaut werden. Es gibt aber auch einige wenige bipolare Mikroprozessoren, die zur Schaltkreisfamilie Schottky-TTL gehören. Sie arbeiten sehr schnell, sind aber verhältnismäßig einfach aufgebaut.

In der MOS-Schaltkreisfamilie sind die NMOS-Typen in der Mehrzahl. Sie benötigen etwa 0,5 bis 1,5 W Leistung. CMOS-Typen gibt es weniger. Ihr besonderer Vorteil ist der äußerst geringe Leistungsbedarf von 1 bis 5 mW. Man unterscheidet ferner statische Mikroprozessoren und dynamische Mikroprozessoren. Statische Mikroprozessoren haben statische Schreib-Lese-Speicher, die keine Auffrischung benötigen. Dynamische Mikroprozessoren haben dynamische Schreib-Lese-Speicher in ihrem Inneren. Diese benötigen einen Auffrischtakt.

4. Befehlsvorrat

Die Größe des Befehlsvorrats ist ein Maß für die Leistungsfähigkeit eines Mikroprozessors – aber nicht das einzige. Es kommt auch auf die Art der Befehle an. Viele geschickt gewählte Befehle ergeben eine große Leistungsfähigkeit. Nach Herstellerangaben schwankt die Zahl der Befehle zwischen 46 und 158.

In der Tabelle Bild 15.15 sind einige häufig verwendete Mikroprozessoren und ihre wichtigsten Eigenschaften aufgeführt.

<i>Bild 15.15 Zusammenstellung einiger wichtiger Mikroprozessoren</i>	Typ	Hersteller	Wortlänge in Bit	Befehls- vorrat	Zykluszeit in µs	Logik-Familie (Technologie)
	4040	Intel	4	60	10,0	NMOS
	8080 A	Intel/Siemens	8	78	2,0	NMOS
	8085	Intel/Siemens	8	80	1,3	NMOS
	IM 6100	Intersil	8	87	2,5	CMOS
	M 6800	Motorola	8	72	2,0	NMOS
	SCMP	Nat. Semic.	8	46	2,0	PMOS/NMOS
	Z 80	Zilog	8	158	1,0	NMOS
	8086	Intel	16	135	0,5	CMOS
	68000	Motorola	16	56	0,6	CMOS/NMOS
	TMS9900	Texas Instr.	16	69	7,5	NMOS
	80386	Intel	32	150	0,3	CMOS
	68020	Motorola	32	100	0,4	CMOS/NMOS
	32332	Nat. Semic.	32	130	0,3	NMOS

15.6.2 Mikroprozessor SAB 8080A

Der Mikroprozessor 8080A wird zur Zeit in sehr großen Stückzahlen eingesetzt und hat sich zu einer Art Standardmikroprozessor entwickelt. Er wird von Intel, Siemens und anderen hergestellt. Die Siemens-Bezeichnung ist SAB 8080A.

Der Mikroprozessor SAB 8080A ist ein 8-Bit-Mikroprozessor in NMOS-Technik mit einer Zykluszeit von 2 µs. Sein Befehlsvorrat umfaßt 78 Befehle (s. Tabelle Bild 15.15). Der Mikroprozessor ist TTL-kompatibel und in Tri-State-Technik ausgeführt, das heißt, die Dateneingänge und -ausgänge und die Adreßausgänge können außer den Pegeln L und H noch einen hochohmigen (abgeschalteten) Zustand annehmen.

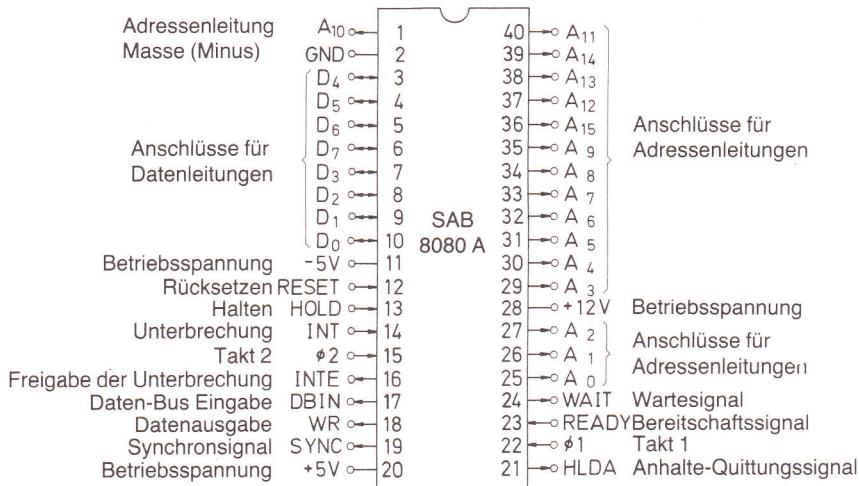


Bild 15.16 Gehäuseanschlüsse des Mikroprozessors SAB 8080A (Siemens)

Die im Mikroprozessor SAB 8080A enthaltenen Schreib-Lese-Speicher sind dynamische Speicher, die der Auffrischung bedürfen. Die Auffrischung erfolgt mit Takt signal. Der Mikroprozessor SAB 8080A wird in einem 40poligen Dual-Inline-Gehäuse geliefert. Die Bedeutung der Gehäuseanschlüsse zeigt Bild 15.16.

Die Anschlüsse 3 bis 10 sind Datenanschlüsse. Sie sind Ausgänge und Eingänge des 8-Bit-Daten-BUS. (BUS-Schaltungen siehe Kapitel 11, Abschnitt 11.4.) Dieser Daten-BUS ist ein Zweiweg-Daten-BUS. Er kann Daten zuführen und abführen.

Der Mikroprozessor verfügt über 16 Adresseausgänge. Es können $2^{16} = 65\,536$ verschiedene Adressen damit angewählt werden. Der Mikroprozessor kann also mit einem externen Speicherbaustein von 64 kBytes zusammenarbeiten. Mit jeder Adresse kann 1 Byte = 8 Bit abgerufen werden. Zusätzlich stehen noch Adressen für die Ansteuerung von Eingabe- und Ausgabebausteinen zur Verfügung. Statt eines 64-kByte-Speichers können auch mehrere kleinere Speicher angeschlossen werden. Dabei ist es gleichgültig, ob es sich um RAM, ROM oder um PROM handelt. Als Adressenausgänge dienen die Gehäuseanschlüsse 1, 25, 26, 27, 29 bis 40 (Bild 15.16). Sie sind mit dem Adressen-BUS zu verbinden.

Die weiteren Anschlüsse dienen der Stromversorgung und der Steuerung des Mikroprozessors. Benötigt werden die Spannungen +12 V (Pin 28), +5 V (Pin 20), -5 V (Pin 11). Masse ist an Pin 2 zu legen. Der Mikroprozessor benötigt einen externen Taktgenerator, der zwei verschiedene Takte erzeugt, Takt 1 und 2 (Takt 1 an Pin 22, Takt 2 an Pin 15). Über den Eingang RESET (Pin 12) wird der Befehlszähler auf 0 gesetzt. Mit einem 0-Signal am Anschluß HOLD (Pin 13) kann der Mikroprozessor angehalten werden. Es können in dieser Zeit Daten eingegeben oder ausgegeben werden. Die Gehäuseanschlüsse INT (Pin 14) und INTE (Pin 16) dienen der Programmunterbrechung und der Freigabe der Programmunterbrechung.

Am Gehäuseanschluß DBIN (Data Bus In) zeigt ein Signal an, daß sich der Daten-BUS in einem Eingabezustand befindet. Es können Daten in den Mikroprozessor übernommen werden. Der Anschluß WR führt 0-Signal, solange Daten an einen äußeren Speicher abgegeben werden. Der SYNC-Anschluß liefert ein Synchronsignal, wenn ein neuer Operationszyklus beginnt.

Besonders wichtig sind die Anschlüsse WAIT (Warten, Pin 24) und READY (Bereit, Pin 23). An WAIT liegt 1-Signal, wenn sich der Mikroprozessor in Wartestellung befindet. 1-Signal am Anschluß READY zeigt an, daß auf dem Daten-BUS Daten zur Übernahme bereitstehen. Es läßt den Mikroprozessor kurzzeitig halten, damit die Daten übernommen werden können.

Über den Anschluß HLDA (Pin 21) liefert der Mikroprozessor eine sogenannte «Anhaltequittung» als Antwort auf ein HOLD-Signal. Sie zeigt an, daß Daten- und Adreß-BUS in den hochohmigen Zustand schalten.

Der innere Aufbau des SAB 8080A ist etwas verwirrend. Er soll deshalb in zwei Stufen erklärt werden. In der vereinfachten Darstellung des Innenaufbaus (Bild 15.17) ist die ALU leicht zu erkennen. Die Rückführung der A-Eingänge erfolgt über den internen Daten-BUS. Über den internen Daten-BUS laufen auch die Daten bei der Einspeicherung von Daten in den Datenspeicher und bei der Ausspeicherung.

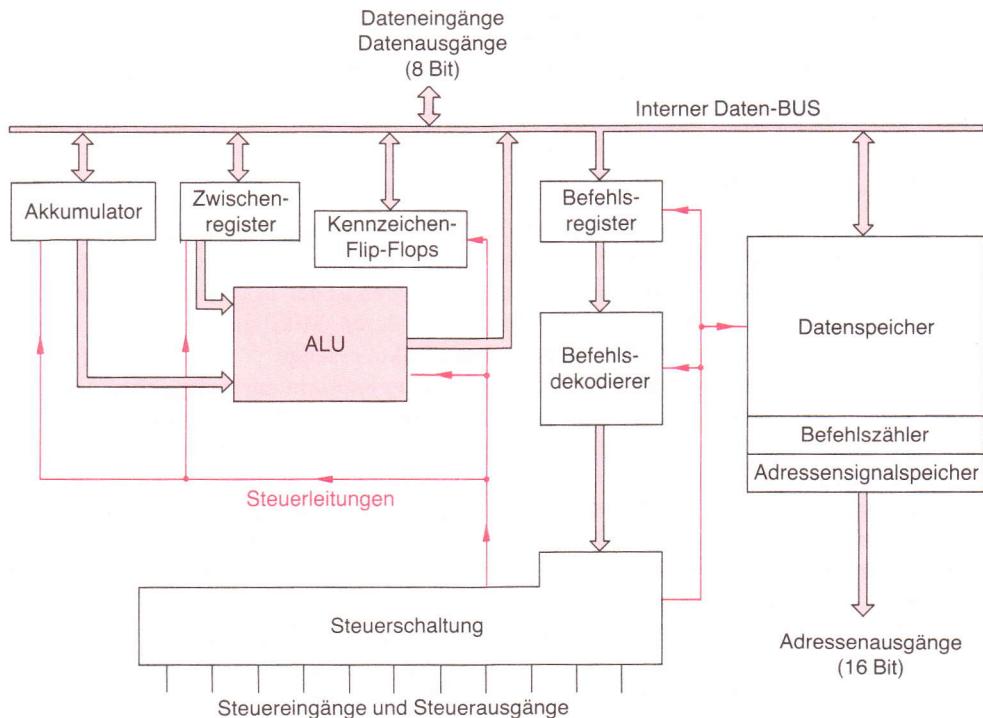


Bild 15.17 Vereinfachter Innenaufbau des Mikroprozessors SAB 8080A

Befehle werden über die Dateneingänge eingegeben. Sie können im Datenspeicher zwischengespeichert werden. Einen besonderen Programmspeicher gibt es nicht. Ein Programmspeicher kann aber als externer Speicher zusätzlich angeschlossen werden. Die Befehle werden über das Befehlsregister dem Befehlsdekodierer zugeführt und als Steuersignale auf die Steuerschaltung gegeben. Die Steuerschaltung verfügt über eine Anzahl von Steuersignalen eingängen und -ausgängen, die bereits besprochen wurden.

Mit Hilfe des Befehlszählers und des Adressensignalspeichers werden die Adressen für die Ansteuerung externer Bausteine erzeugt, z.B. Adressen für RAM und ROM. Die Adressen bestehen aus 16-Bit-Wörtern. Sie werden dem Adreß-BUS zugeführt.

Die Hersteller geben in ihren Datenbüchern den vollständigen Innenaufbau des Mikroprozessors SAB 8080A an (Bild 15.18).

Hier kommen noch einige weitere Einheiten hinzu, z.B. Puffer für Daten und Adressen. Puffer sind Zwischenspeicher. Der Datenspeicher wird in viele Register aufgeteilt, die über Multiplexer angewählt werden.

Es ist kaum vorstellbar, daß diese sehr komplizierte Schaltung in einem Silizium-Chip von nur etwa $5 \text{ mm} \times 5 \text{ mm}$ Größe untergebracht werden kann. Erstaunlich ist auch der verhältnismäßig geringe Preis, der zur Zeit für Mikroprozessoren allgemein gefordert wird.

Wie wird nun eine Steuerschaltung mit dem Mikroprozessor SAB 8080A aufgebaut? Mit dem Mikroprozessor allein kann keine Steuerschaltung aufgebaut werden. Man benötigt Zusatzbausteine – z.B. einen Taktgenerator, Eingabe-Ausgabe-Bausteine und einen oder mehrere Speicher vom Typ ROM, PROM, EPROM oder RAM für Programm und Daten. Vor allem benötigt man ein Stromversorgungsteil.

15.6.3 Zusatzbausteine für Mikroprozessoren

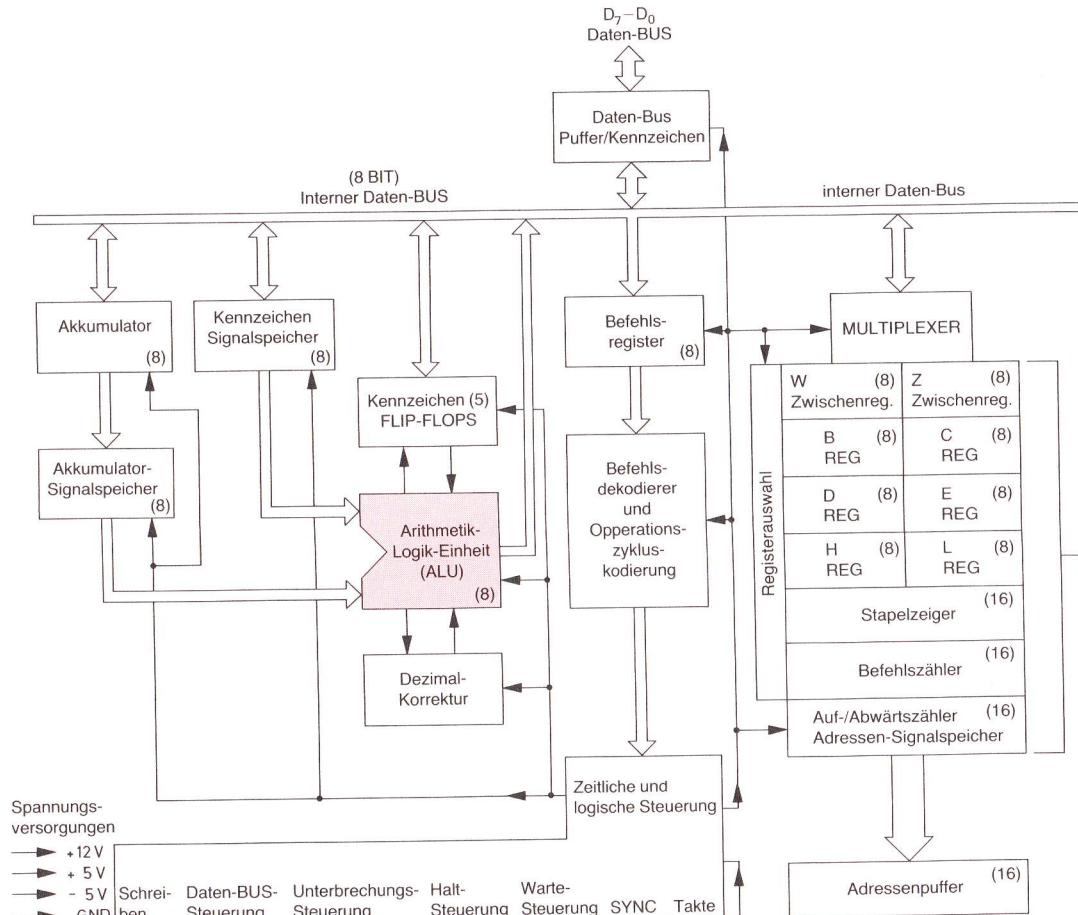
Mikroprozessoren arbeiten taktgesteuert. Erforderlich ist mindestens ein Takt, oft werden zwei verschiedene Takte benötigt. Nur wenige Mikroprozessortypen besitzen einen «inneren Taktgenerator». Bei diesen Mikroprozessoren ist der Taktgenerator bereits auf dem Mikroprozessor-Chip enthalten. Bei den anderen Mikroprozessortypen ist ein Zusatzbaustein erforderlich, der die benötigten Takte erzeugt.

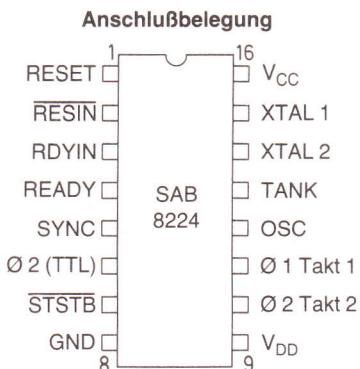
Für den Mikroprozessor SAB 8080A wird als Zusatzbaustein der Taktgeber SAB 8224 empfohlen. Dieser Baustein wird in einem 16poligen Dual-In-Line-Gehäuse geliefert. Er enthält einen Quarzoszillator, dessen Frequenz durch einen außen anzuschließenden Quarz bestimmt wird. Die Eigenfrequenz des Quarzes kann in weiten Grenzen gewählt werden. Sie beeinflußt die Arbeitsgeschwindigkeit des Mikroprozessors. Üblich sind Quarzfrequenzen von etwa 18 MHz. Die Schwingung des Quarzoszillators wird in eine Rechteckschwingung umgewandelt, die dann um den Faktor 9 heruntergeteilt wird. So ergibt sich die Arbeitsfrequenz von 2 MHz.

Das Anschlußschema des Taktbausteins SAB 8224 und die Form der erzeugten Takte sind in Bild 15.19 dargestellt. Zusätzlich zu den Takten erzeugt der Baustein noch Steuersignale, z.B. ein Rücksetzsignal nach Einschalten der Speisespannung.

Für die Dateneingabe und die Datenausgabe sind ebenfalls Zusatzbausteine erforderlich. Diese werden E/A-Bausteine (engl. I/O-Bausteine, Input/Output-Bausteine) genannt.

Bild 15.18 Vollständiger Innenaufbau
Mikroprozessors SAB 8080A (Siemens)

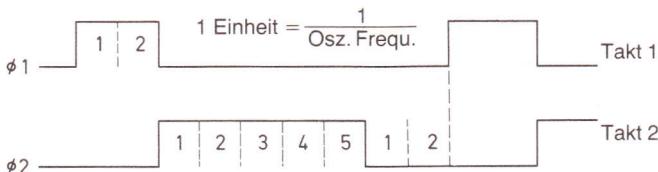




Anschlußbezeichnungen

RESIN	Rücksetzeingang
RESET	Rücksetzausgang
RDYIN	Bereiteingang
READY	Bereitausgang
SYNC	Synchr. Eingang
STSTB	Zustandsübernahme ("Low" aktiv)
Ø 1	Taktgeber
Ø 2	für SAB 8080
XTAL 1	Anschlüsse
XTAL 2	für externen Quarz
TANK	Eingang für Oberwellen Quarz
OSC	Oszillatoren-Ausgang
Ø 2 (TTL)	Taktgeber (TTL-Pegel)
V _{CC}	Versorgungsspannung (+ 5 V)
V _{DD}	Versorgungsspannung (+ 12 V)
GND	Masse (0 V)

Impulsdiagramm



Beispiel: SAB 8080 t_{CY} (Taktperiode) = 500 ns

$$\begin{aligned} \text{OSC} &= \text{mHz}/55 \text{ ns} \\ \text{Ø}_1 &= 110 \text{ ns} (2 \times 55 \text{ ns}) \\ \text{Ø}_2 &= 275 \text{ ns} (5 \times 55 \text{ ns}) \\ \text{Ø}_2 - \text{Ø}_1 &= 110 \text{ ns} (2 \times 55 \text{ ns}) \end{aligned}$$

Bild 15.19 Anschlußschema des Taktbausteins SAB 8224 und zeitlicher Verlauf der Takte (Siemens)

Sie nehmen die Daten in einen Zwischenspeicher (Puffer) auf. Bei der Dateneingabe erzeugen sie Steuersignale, mit deren Hilfe der Mikroprozessor solange angehalten wird, bis die Daten eingegeben sind. Die E/A-Bausteine enthalten oft auch eine Baustein-Auswahlschaltung. Sollen Informationen dem Mikroprozessor entnommen werden, veranlaßt der E/A-Baustein die gewünschte Datenausgabe. Für den Mikroprozessor SAB 8080A sind verschiedene E/A-Bausteine verfügbar. Häufig verwendet wird der Baustein SAB 8212.

Sehr wichtige Zusatzbausteine sind die Speicherbausteine. Hier können RAM-, ROM- und PROM-Bausteine und selbstverständlich auch lösrbare PROM-Bausteine in verschiedenen Kombinationen bis zu einer maximal zulässigen Gesamtspeicherkapazität eingesetzt werden. Die zulässige Gesamtspeicherkapazität beträgt beim Mikroprozessor SAB 8080A 64 kByte, da der Mikroprozessor insgesamt 65 636 verschiedene Adressen erzeugen kann.

Die Speicherbausteine sind deshalb so wichtig, weil die interne Speicherkapazität der Mikroprozessoren verhältnismäßig gering ist, so daß Programmdaten und Verarbeitungsdaten extern gespeichert werden müssen.

Als externe Speicher können auch Magnetbandgeräte (z.B. Digital-Kassettenrecorder) und Magnetplattengeräte verwendet werden. Zum Anschluß derartiger Geräte werden Schnittstellen-Bausteine benötigt, die die Daten in bestimmter Weise umformen und Steuersignale verarbeiten und erzeugen. Sie müssen z.B. parallel ausgegebene 8-Bit-Daten eines Mikroprozessors in die serielle Datenform umsetzen, die ein Magnetplattenspeicher benötigt.

Der Einsatz von E/A-Bausteinen, von Speicherbausteinen und von Schnittstellenbausteinen erfordert Steuersignale, die der Mikroprozessor nicht oder nur unvollkommen liefern kann. Aus diesem Grund ist in vielen Fällen ein System-Steuerbaustein als weiterer Zusatzbaustein erforderlich. Ein solcher Baustein erzeugt alle Signale, die zur direkten Kopplung von Zusatzbausteinen benötigt werden. Er enthält oft auch einen sogenannten BUS-Treiber. Ein BUS-Treiber ist eine Verstärkerschaltung für Signale, die an einen BUS abgegeben werden. Verstärkungsbedürftig sind oft die Signale für einen Daten-BUS.

Der System-Steuerbaustein SAB 8228 wurde für den Mikroprozessor SAB 8080A entwickelt. Er enthält einen 8-Bit-Zweiweg-BUS-Treiber für den Daten-Bus. Das Steuersystem liefert alle erforderlichen Steuersignale und darüber hinaus noch zusätzliche Steuersignale für eine einfache Gestaltung von Programmunterbrechungen und für die Verwendung von Mehr-Byte-Befehlen. Mehr-Byte-Befehle sind Befehle, die eine Wortlänge von zwei oder mehr Byte haben.

15.7 Mikrocomputer

Schaltet man einen Mikroprozessor mit den erforderlichen Zusatzbausteinen zusammen, erhält man einen Mikrocomputer. Einige Zusatzbausteine sind unbedingt notwendig – wie Taktgeber und Speicher. Andere Zusatzbausteine werden je nach der zu erfüllenden Aufgabe ausgewählt.

Ein Mikrocomputer ist eine funktionsfähige Steuereinheit aus Mikroprozessor und Zusatzbausteinen.

Mikrocomputer werden meist auf einer Platine aufgebaut. Eine solche Platine kann z.B. folgende Bausteine enthalten:

Mikroprozessor	SAB 8080A
Taktgeber	SAB 8224
REPROM	SAB 8708
RAM	SAB 8111-2
E/A-Baustein	SAB 8212
System-Steuerbaustein	SAB 8228
Quarz für den Taktgeber	

Für einen solchen Mikrocomputer ergibt sich die Schaltung Bild 15.20. Der vom SAB 8080A ausgehende Daten-BUS durchläuft den System-Steuerbaustein SAB 8228. Hier werden die ankommenden und die abgehenden Daten verstärkt. Der Daten-BUS ist ein 8-Bit-Zweiweg-BUS.

Der Adreß-BUS ist ein 16-Bit-Einweg-BUS. Die Adressen kommen stets vom Mikroprozessor. An den Daten-BUS und an den Adreß-BUS sind die E/A-Einheiten und die Speicherbausteine angeschlossen. Die Steuerung erfolgt über den SAB-8228-Baustein. Im REPROM sollen die einzelnen Befehle gespeichert sein, die nacheinander auszuführen sind – also das Programm. Die benötigten Daten werden von außen in das RAM eingegeben. Jetzt kann der Steuer- oder Rechenvorgang ablaufen. Ergebnisse werden wieder im RAM gespeichert. Sie können nach Wunsch nach außen abgegeben werden.

Das Programmieren eines Mikrocomputers muß recht mühsam erlernt werden. Leider hat jeder Mikroprozessortyp etwas andere Befehle. Es ist daher zu empfehlen, sich auf einen Mikroprozessortyp einzuarbeiten und am Anfang nur mit diesem Mikroprozessortyp zu arbeiten. Wenn man den Befehlsvorrat eines Mikroprozessortyps voll beherrscht, ist eine Umstellung auf einen anderen Mikroprozessortyp verhältnismäßig

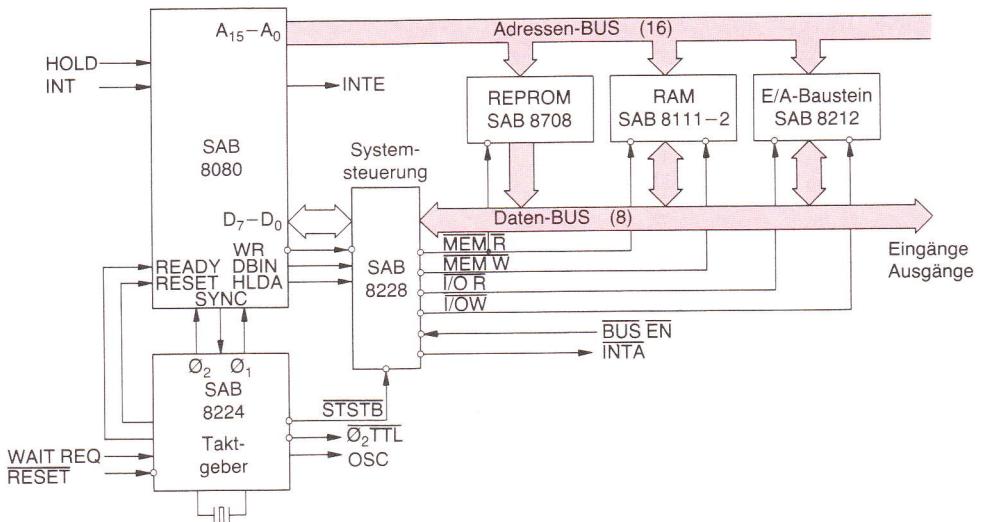


Bild 15.20 Aufbau eines Mikrocomputers

leicht. Die Hersteller geben Hilfen beim Erlernen der Programmierung. Auch gibt es eine größere Zahl von Übungs-Mikrocomputern, mit denen man sich das Programmieren schrittweise selbst erarbeiten kann. Die Teilnahme an einem Lehrgang ist zu empfehlen.

Mikrocomputer können aus Mikroprozessoren und Zusatzbausteinen auf vielfältige Art zusammengestellt werden. Man kann verschiedenartige Zusatzbausteine auswählen und kombinieren und Speicherarten und Speicherkapazitäten variieren, um eine optimale Lösung des gestellten Problems zu erreichen. Hierfür bieten die Hersteller *Entwicklungsgeräte* an, durch die die Entwicklungsarbeit wesentlich vereinfacht wird.

Eine interessante Lösung des Entwicklungsproblems stellen *1-Chip-Mikrocomputer* dar. Hier ist auf einem Chip ein vollständiger Mikrocomputer integriert. Über den Systemaufbau, über Zusatzbausteine wie in Bild 15.20 braucht man sich keine Gedanken zu machen. Das System ist fertig. Der 1-Chip-Mikrocomputer soll alles Erforderliche enthalten.

Doch was enthält ein solcher Mikrocomputer nun tatsächlich? Interessant ist vor allem, welche Speicher er hat und wie groß deren Speicherkapazität ist.

Untersucht man die auf dem Markt befindlichen 1-Chip-Mikrocomputer etwas genauer – z.B. den TMS 1000 von Texas Instruments –, so stellt man fest, daß die Speicherkapazitäten doch recht klein sind. Die zur Zeit erreichbare maximale Integrationsdichte läßt größere Speicherkapazitäten nicht zu. Auch ist der Mikroprozessor nur als 4-Bit-Mikroprozessor ausgelegt. Daraus folgt:

1-Chip-Mikrocomputer sind zur Zeit nur für einfache Steuerungsaufgaben geeignet.

Sie sind wenig flexibel, d.h., sie lassen eine Anpassung an besondere Aufgabenstellungen nicht zu.

Die erreichbare Integrationsdichte wird jedoch größer werden. Das bedeutet, daß 1-Chip-Mikrocomputer in Zukunft komplexer aufgebaut werden können und auch größere Speicherkapazitäten haben werden. Sie werden dann eine größere Bedeutung erlangen.

15.8 Lernziel-Test

1. Erklären Sie den Aufbau einer ALU.
2. Welchen Vorteil bringt es, die 6 Steuereingänge einer ALU auf nur 4 Steuereingänge umzukodieren?
3. Was versteht man unter einem Akkumulator? Beschreiben Sie die Arbeitsweise.
4. Skizzieren Sie die Schaltung eines Akkumulators mit ALU, Register und Übertragspeicher.
5. Wieviel verschiedene Befehle lassen sich mit 4-Bit-Einheiten darstellen?

6. Erklären Sie die Arbeitsweise eines Akkumulators mit Datenspeicher anhand der Schaltung Bild 15.11.
7. Wodurch unterscheidet sich ein Mikroprozessor von einem Mikrocomputer?
8. Was bedeuten die Bezeichnungen: 4-Bit-Mikroprozessor, 8-Bit-Mikroprozessor, 16-Bit-Mikroprozessor?
9. In welchen Technologien werden Mikroprozessoren hergestellt bzw. zu welchen Schaltkreisfamilien gehören sie?
10. Was versteht man unter einem Daten-BUS, was unter einem Adreß-BUS?
11. Ein Mikroprozessor benötigt Zusatzbausteine. Nennen Sie die Namen von vier möglichen Zusatzbausteinen.
12. Was ist ein 1-Chip-Mikrocomputer, was ist ein 1-Platinen-Mikrocomputer?