Aufgabe 1

Eine Zahl a ist mit 8 Bits vorzeichenlos (8 bit unsigned) dargestellt. Die Zahl y soll die Zahl a multipliziert mit 4 sein ($y = a \cdot 4_D$).

a) Wie viele Bits benötigen Sie für die Darstellung von y?

b) Geben Sie die vollständige Schaltung an, in der die einzelnen Bits von a und y dargestellt sind.

Lösung

a) Für die Darstellung des Ergebnisses werden zwei zusätzliche Bits benötigt. Das Ergebnis ist dann insgesamt 10 Bits breit. Die größte darstellbare Zahl mit 8 Bits ist 255_D. $4 \cdot 255$ ist 1020. Die größte darstellbare Zahl mit 10 Bits ist $2^{10} - 1 = 1023$.

b) Eine Multiplikation mit 2 kann durch Schieben nach Links um eine Position erreicht werden. Dies ist analog zu einer Multiplikation mit 10 im Dezimalsystem. Die neuen niedrigstwertigsten Stellen werden mit 0 aufgefüllt. Die Schaltung ist in Abbildung 1 dargestellt. Es ist eine einfache Umverdrahtung.

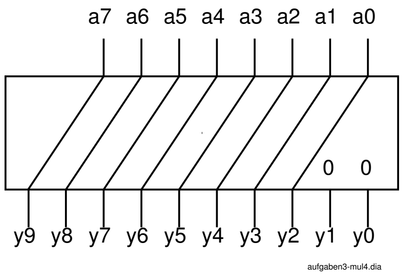

Abbildung 1: Schaltung für die Multiplikation mit 4

Aufgabe 2

Multiplizieren Sie die beiden Zahlen 14_H und 37_H im Binärsystem und geben Sie das Ergebnis in Binär- und Hexadezimaldarstellung an.

Lösung

Die Zahl 14_H in Binärdarstellung ist 10100_B. Die Zahl 37_H in Binärdarstellung ist 110111_B. In Tabelle 1 ist das Schema für die Multiplikation dargestellt.

Das Ergebnis der Multiplikation ist $44C_H$.

Tabelle 1: Schema für die Multiplikation im Binärsystem

	1	0	1	0	0	·	1	1	0	1	1	1
								1	0	1	0	0
+							1	0	1	0	0	
+						1	0	1	0	0		
+					0	0	0	0	0			
+				1	0	1	0	0				
+			1	0	1	0	0					
Übertrag			1	1	1	1	1					
		1	0	0	0	1	0	0	1	1	0	0
	4				4				C			

Aufgabe 3

Ein Multiplizierer kann zwei 8 Bit Zahlen kombinatorisch multiplizieren. Dieser Multiplizierer benötigt für diese Funktion 500 Gatter. Der Multiplizierer soll jetzt erweitert werden und zwei 16 Bit Zahlen multiplizieren. Schätzen Sie die Anzahl der benötigten Gatter für den neuen Multiplizierer ab.

Lösung

Die Komplexität eines Multiplizierers hängt in etwa quadratisch von der Bitbreite ab. Die Bitbreite des neuen Multiplizierers ist doppelt so groß wie die des alten Multiplizierers.

$$C_{alt} \approx A \cdot n^2$$
$$C_{neu} \approx A \cdot (2n)^2 = 4 \cdot A \cdot n^2 = 4 \cdot C_{alt}$$

Aufgabe 4

Ein 5 Bit Addierer addiert zwei Zahlen a und b, die in einer vorzeichenbehafteten 2er Komplement Darstellung gegeben sind. Das Ergebnis der Addition y hat wieder eine Bitbreite von 5 Bit. Geben Sie das Ergebnis der Addition als Dezimalzahl an.

a) $y = 8_D + 3_D$

b) $y = 7_D + 9_D$

c) $y = -4_D + 9_D$

d) $y = -4_H + -A_H$

e) $y = -9_H + -8_H$

f) $y = -11_D + -13_D$

Lösung

Der darstellbare Zahlenbereich ist -2^4 bis $+2^4-1$. Für die Lösung muss man überprüfen, ob es zu einem Überlauf kommt und dann den korrekten Wert berechnen.

a) $y = 8_D + 3_D = 11_D$

b) $y = 7_D + 9_D = -16_D$

Hier kommt es zu einem Überlauf. Das Ergebnis ist deshalb $-32 + 7 + 9 = -16_D$

c) $y = -4_D + 9_D = 5_D$

Bei der Addition einer negativen und einer positiven Zahl ist kein Überlauf möglich.

d) $y = -4_H + -A_H = -14_D$

Bei dieser Addition kommt es zu keinem Überlauf.

e) $y = -9_H + -B_H = 12_D$

Hier kommt es zu einem Überlauf. Das Ergebnis ist $32 - (9 + 11) = 12_D$

f) $y = -11_D + -13_D = 8_D$

Hier kommt es zu einem Überlauf. Ergebnis ist $32 - 11 - 13 = 8_D$

Aufgabe 5

Beweisen Sie, dass das Assoziativgesetz für die xor Verknüpfung gilt.

Lösung

Das Assoziativgesetz besagt: $(a \oplus b) \oplus c = a \oplus (b \oplus c)$

Der Beweis kann mit einer vollständigen Aufzählung der möglichen Eingangskombinationen von a, b, c in einer Wahrheitstabelle, wie in Tabelle 2 dargestellt, erfolgen.

In den beiden Spalten $(a \oplus b) \oplus c$ und $a \oplus (b \oplus c)$ stehen die gleichen Wahrheitswerte in jeder Zeile. Die beiden Funktionen sind also gleich.

Alternativ kann man das Assoziativgesetz mit den booleschen Theoremen beweisen. Die xor Funktion ist definiert als:

$$a \oplus b = a'b + ab'$$

Diese Definition wird jetzt jeweils unter Berücksichtigung der Klammerung auf die beiden Ausdrücke $(a \oplus b) \oplus c$ und $a \oplus (b \oplus c)$ angewandt.

Tabelle 2: Wahrheitstabelle für das Assoziativgesetz

a	b	c	$a \oplus b$	$(a \oplus b) \oplus c$	$b \oplus c$	$a \oplus (b \oplus c)$
0	0	0	0	0	0	0
0	0	1	0	1	1	1
0	1	0	1	1	1	1
0	1	1	1	0	0	0
1	0	0	1	1	0	1
1	0	1	1	0	1	0
1	1	0	0	0	1	0
1	1	1	0	1	0	1

$$
\begin{aligned}
(a \oplus b) \oplus c &= (a'b + ab')'c + (a'b + ab')c' \\
&= ((a'b)' \cdot (ab')')c + a'bc' + ab'c' && |DeMorgan \\
&= ((a + b') \cdot (a' + b))c + a'bc' + ab'c' && |DeMorgan \\
&= (aa' + ab + b'a' + b'b)c + a'bc' + ab'c' \\
&= abc + b'a'c + a'bc' + ab'c' \\
a \oplus (b \oplus c) &= a'(b'c + bc') + a(b'c + bc')' \\
&= a'b'c + a'bc' + a((b'c)'(bc')') \\
&= a'b'c + a'bc' + a((b + c')(b' + c)) \\
&= a'b'c + a'bc' + a(bb' + bc + c'b' + c'c) \\
&= a'b'c + a'bc' + abc + ac'b'
\end{aligned}
$$

Die beiden Gleichungen wurde jeweils mittels des Theorems von De Morgan und Anwendung des Distributivgesetzes in eine disjunktive Form überführt (hier sogar eine disjunktive Normalform, da alle Variablen a,b,c in jedem Term auftauchen). Die Terme der Gleichungen sind gleich, also sind die beiden Ausdrücke gleich.

Aufgabe 6

Bilden Sie die Funktion y = a + b nur mit NAND Gattern nach.

Lösung

Die NAND Funktion kann also ODER Funktion mit invertierten Eingängen aufgefasst werden.

$$\text{a nand b} = (ab)' = a' + b'$$

Um einen Inverter zu erhalten kann ein Eingang eines NAND Gatters auf 1 gelegt werden.

$$\text{a nand } 1 = (a1)' = a'$$

Die Schaltung ist in Abbildung 2 dargestellt.

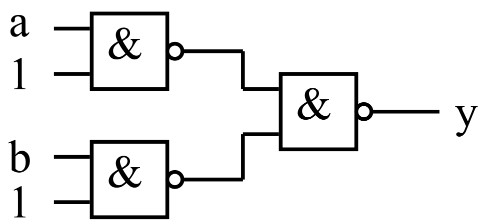

Abbildung 2: Die Funktion y = a + b mit NAND Gattern

Aufgabe 7

Entwerfen Sie eine Schaltung, die zwei 10 Bit Zahlen subtrahiert: y = a - b. Die Zahlen a und b sind im vorzeichenbehafteten 2er Komplement dargestellt. Das Ergebnis y soll auch als 10 Bit Zahl im 2er Komplement dargestellt werden.

Lösung

Die Subtraktion wird auf eine Addition zurückgeführt, bei der der Subtrahend mit -1 multipliziert wird. Der Addierer kann als Ripple-Carry-Adder mit Volladdierern ausgeführt werden. Der Vorzeichenwechsel beim Subtrahenden, also die Multiplikation mit -1, wird durch Bildung des 2er Komplements erreicht. Für die Bildung des 2er Komplements muss die Zahl invertiert und zu dem Ergebnis der Inversion eine 1_B addiert werden. Diese Addition kann durch die Nutzung des Übertragseingangs des ersten Volladdierers erreicht werden. Die Schaltung ist in Abbildung 3 dargestellt.

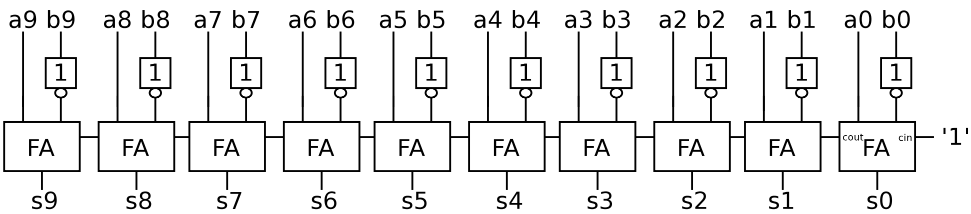

Abbildung 3: Schaltung zur Subtraktion

Aufgabe 8

Ein gegebener Algorithmus zur digitalen Signalverarbeitung führt bei einer gewählten Bitbreite zur Zahlendarstellung bei einer Addition y = a + b regelmäßig zu Überläufen. Die gewählte Bitbreite ist 10 Bit und die Zahlendarstellung vorzeichenbehaftet im 2er Komplement. Es treten im Algorithmus beispielsweise Zahlenkombinationen $a = +318_D$ und $b = +219_D$ auf. Das Ergebnis einer Standardaddition wäre in diesem Fall $-1024_D + 318_D + 219_D = -487_D$ statt des erwarteten Wertes von $+537_D$.

a) Entwerfen Sie eine Additionsschaltung, bei der bei einem Überlauf das Ergebnis der Addition auf den maximal darstellbaren Wert gekappt wird. In diesem Beispiel sollte das Ergebnis der Addition $+318_D+219_D = +511_D$ ergeben. Bei einem Überlauf im negativen Zahlenbereich soll das Ergebnis auch analog auf -512_D gekappt werden.

b) Vergleichen Sie Ihre Lösung mit der Alternative die Bitbreite von 10 auf 11 Bit zu erhöhen im Hinblick auf die Anzahl der erforderlichen Gatter.

Lösung

a) Ein Überlauf bei einer Addition im 2er Komplement kann nur auftreten, wenn beide Summanden positiv oder beide Summanden negativ sind. Bei einem Überlauf tritt dann ein Vorzeichenwechsel beim Ergebnis auf. Die Schaltung muss diesen Überlauf erkennen und in diesem Fall das Ergebnis des Standardaddierers verändern. Wenn die Summanden positiv sind, muss das neue Ergebnis $+511_D = 0111111111_B$ sein und wenn die Summanden negativ sind, dann muss das neue Ergebnis $-512_D = 1000000000_B$ sein. In Abbildung 4 ist die Schaltung zur Begrenzung dargestellt.

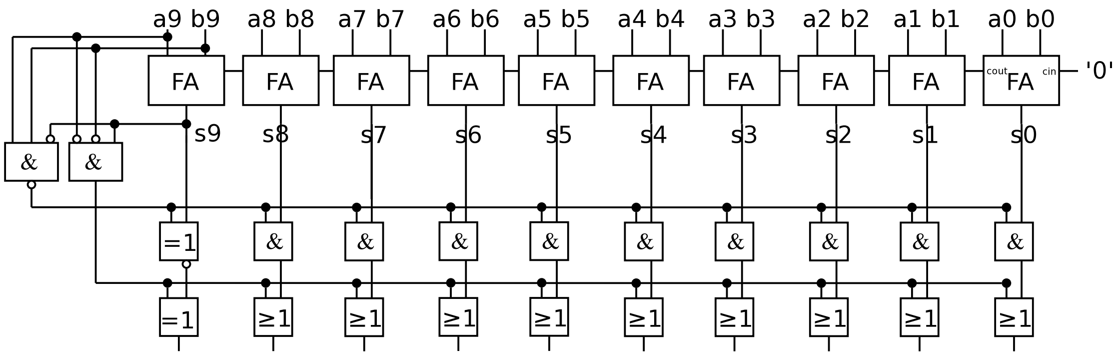

Abbildung 4: Addierer mit Begrenzung bei Überlauf

b) Um die Bitbreite des Addierers von 10 auf 11 Bit zu erhöhen, benötigt man einen zusätzlichen Volladdierer. Ein Volladdierer aufgebaut aus UND und ODER Gattern, benötigt etwa 9 Gatter. Für die obige Limitierungsschaltung sind 22 Gatter notwendig. Eine Erhöhung der Bitbreite von 10 auf 11 Bit benötigt demnach weniger Gatter.