

Digitaltechnik

Wintersemester 2017/2018

2. Übung



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Andreas Engel, Raad Bahmani

LÖSUNGSVORSCHLAG

KW44

Die Präsenzübungen werden in Kleingruppen während der wöchentlichen Übungsstunde bearbeitet. Bei Fragen hilft Ihnen Ihr Tutor gerne weiter.

Die mit „Zusatzaufgabe“ gekennzeichneten Aufgaben sind zur zusätzlichen Vertiefung für interessierte Studierende gedacht und daher nicht im Zeitumfang von 90 Minuten einkalkuliert.

Übung 2.1 Wertebereich binärer Zahlendarstellungen

Der Wertebereich einer Funktion $f : A \rightarrow B$ auf der Eingabemenge A wird durch die Menge $f(A) = \{f(a) : a \in A\} \subseteq B$ charakterisiert. Diese Schreibweise wird in den folgenden beiden Teilaufgaben verwendet, um den Wertebereich der verschiedenen binären Zahlendarstellungen zu beschreiben. Die Angabe einer konkreten Binärdarstellung ist für beide Teilaufgaben nicht notwendig.

Übung 2.1.1 Minimal und maximale Werte

Tragen Sie in folgender Tabelle die minimal und maximal darstellbare Dezimalzahl ein.

k	Vorzeichenlos: $u_{2,k}(\mathbb{B}^k)$		Betrag und Vorzeichen: $bv_{2,k}(\mathbb{B}^k)$		Zweierkomplement: $s_k(\mathbb{B}^k)$	
	min	max	min	max	min	max
4	0	$2^4 - 1 = 15$	$-(2^{4-1} - 1) = -7$	$+(2^{4-1} - 1) = 7$	$-(2^{4-1}) = -8$	$2^{4-1} - 1 = 7$
9	0	$2^9 - 1 = 511$	$-(2^{9-1} - 1) = -255$	$+(2^{9-1} - 1) = 255$	$-(2^{9-1}) = -256$	$2^{9-1} - 1 = 255$
12	0	$2^{12} - 1 = 4095$	$-(2^{12-1} - 1) = -2047$	$+(2^{12-1} - 1) = 2047$	$-(2^{12-1}) = -2048$	$2^{12-1} - 1 = 2047$

Übung 2.1.2 Notwendige Bitbreite

Tragen Sie in folgender Tabelle die minimal notwendige Bitbreite k zur binären Darstellung der Dezimalzahlen n ein. Die beide schwarz hinterlegten Felder bleiben frei.

Dezimal n	Vorzeichenlos $\min k \in \mathbb{N} : n \in u_{2,k}(\mathbb{B}^k)$	Betrag und Vorzeichen $\min k \in \mathbb{N} : n \in bv_{2,k}(\mathbb{B}^k)$	Zweierkomplement $\min k \in \mathbb{N} : n \in s_k(\mathbb{B}^k)$
205_{10}	8	9	9
5_{10}	3	4	4
1024_{10}	11	12	12
31_{10}	5	6	6
-50_{10}		7	7
-128_{10}		9	8

Übung 2.2 Bitbreitenerweiterung

Erweitern Sie die angegebenen Bitfolgen von 4 bit auf 8 bit.

$a \in \mathbb{B}^4$	Vorzeichenlos $\hat{a} \in \mathbb{B}^8 : u_{2,4}(a) = u_{2,8}(\hat{a})$	Betrag und Vorzeichen $\hat{a} \in \mathbb{B}^8 : bv_{2,4}(a) = bv_{2,8}(\hat{a})$	Zweierkomplement $\hat{a} \in \mathbb{B}^8 : s_4(a) = s_8(\hat{a})$
0000_2	$0000\ 0000_2$	$0000\ 0000_2$	$0000\ 0000_2$
0110_2	$0000\ 0110_2$	$0000\ 0110_2$	$0000\ 0110_2$
1100_2	$0000\ 1100_2$	$1000\ 0100_2$	$1111\ 1100_2$
1111_2	$0000\ 1111_2$	$1000\ 0111_2$	$1111\ 1111_2$

Übung 2.3 Konvertierung zwischen Zahlendarstellungen

Vervollständigen Sie die folgenden Tabellen. Nutzen Sie für die Konvertierung von der Dezimaldarstellung in die Binärdarstellung jeweils beide in der Vorlesung vorgestellten Konvertierungsverfahren. Geben Sie alle Ziffernfolgen jeweils mit der minimal möglichen Länge an.

Übung 2.3.1 Vorzeichenlose Zahlendarstellungen $u_{b,k}$

Dezimal	Binär	Hexadezimal
147_{10}	$1001\ 0011_2$	93_{16}
213_{10}	$1101\ 0101_2$	$D5_{16}$
999_{10}	$11\ 1110\ 0111_2$	$3E7_{16}$
2100_{10}	$1000\ 0011\ 0100_2$	834_{16}

Übung 2.3.2 Zweierkomplement Darstellung s_k

In der Vorlesung wurde die hexadezimale Darstellung nur für vorzeichenlose Zahlen behandelt. Die binäre Ziffernfolge einer Zweierkomplementdarstellung kann aber ebenfalls hexadezimal dargestellt werden. Solange die Bitbreite nicht festgelegt ist, muss man dabei aber eine Vorzeichenexpansion auf eine durch Vier teilbare Bitbreite durchführen.

Dezimal	Binär	Hexadezimal
7_{10}	0111_2	7_{16}
-45_{10}	$101\ 0011_2$	$D3_{16}$
127_{10}	$0111\ 1111_2$	$7F_{16}$
-210_{10}	$1\ 0010\ 1110_2$	$F2E_{16}$

Bei der binären Darstellung nicht-negativer Zahlen ist ein führendes Null-Bit erforderlich.

Übung 2.4 Addition von Zweierkomplement-Zahlen

Addieren Sie die folgenden Zweierkomplement-Zahlen. Geben Sie Summe und Summanden auch dezimal an. Tritt ein Überlauf auf?

	1	1		1				Übertrag
	0	0	1	1	0	0	1	(51_{10})
+	1	0	1	1	1	0	1	(-70_{10})
	1	1	1	0	1	1	0	(-19_{10})
								kein Überlauf

	1	1	1	1	1	1	1	Übertrag
	0	1	1	0	0	1	1	(103_{10})
+	0	0	0	1	1	1	0	(29_{10})
	1	0	0	0	0	1	0	(-124_{10})
								Überlauf

Übung 2.5 Subtraktion von Zweierkomplement-Zahlen

Wandeln Sie die folgenden Dezimalzahlen in 1 Byte breite Zweierkomplement-Zahlen um. Subtrahieren Sie die Binärdarstellungen voneinander, indem Sie den Minuend mit dem negierten Subtrahend addieren. Wandeln Sie das Ergebnis wieder ins Dezimalformat um. Hat die Subtraktion einen Überlauf verursacht?

a) $49_{10} - 74_{10}$

1. Umrechnung ins Zweierkomplement:

$$49_{10} = 0011\ 0001_2$$

$$74_{10} = 0100\ 1010_2$$

2. Subtrahend negieren:

$$-(0100\ 1010_2) = \sim 0100\ 1010_2 + 1 = 1011\ 0101_2 + 1 = 1011\ 0110_2$$

3. Addition:

$$0011\ 0001_2$$

$$+1011\ 0110_2$$

$$=1110\ 0111_2$$

4. Umrechnen ins Dezimalformat:

$$1110\ 0111_2 = -(\sim 1110\ 0111_2 + 1) = -(0001\ 1000_2 + 1) = -0001\ 1001_2 = -25_{10} = 49_{10} - 74_{10} \quad \checkmark$$

kein Überlauf

b) $-73_{10} - 60_{10}$

1. Umrechnung ins Zweierkomplement:

$$-73_{10} = \sim 0100\ 1001_2 + 1 = 1011\ 0110_2 + 1 = 1011\ 0111_2$$

$$60_{10} = 0011\ 1100_2$$

2. Subtrahend negieren:

$$-(0011\ 1100_2) = \sim 0011\ 1100_2 + 1 = 1100\ 0011_2 + 1 = 1100\ 0100_2$$

3. Addition:

$$1011\ 0111_2$$

$$+1100\ 0100_2$$

$$=0111\ 1011_2$$

4. Umrechnen ins Dezimalformat:

$$0111\ 1011_2 = 123_{10} \neq -73_{10} - 60_{10} \quad \text{! (Überlauf)}$$

Übung 2.5.1 Binary-Coded Decimal (BCD) - Zusatzaufgabe

BCD ist eine weitere vorzeichenlose binäre Zahlendarstellung. Dabei wird jede Ziffer der Dezimaldarstellung einer Zahl einzeln als vorzeichenlose 4 bit Binärzahl repräsentiert. Um also beispielsweise die Zahl 39_{10} im BCD-Format anzugeben, müssen die beiden Dezimalziffern 3 und 9 jeweils mit 4 bit dargestellt werden: $39_{10} = 0011\ 1001_{\text{bcd}}$.

Vervollständigen Sie die folgende Tabelle

80_{10}	$1000\ 0000_{\text{bcd}}$
256_{10}	$0010\ 0101\ 0110_{\text{bcd}}$
734_{10}	$0111\ 0011\ 0100_{\text{bcd}}$
9107_{10}	$1001\ 0001\ 0000\ 0111_{\text{bcd}}$

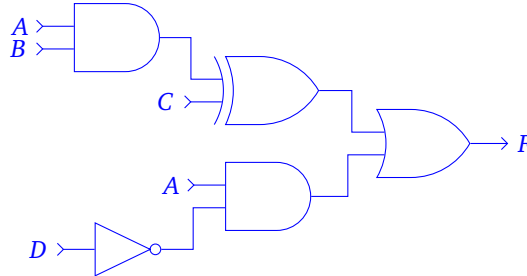
Um systematisch Binärzahlen in das BCD-Format zu konvertieren, wird der *Double Dabble Algorithmus* verwendet, der auch *shift-and-add-3-Algorithmus* genannt wird. Eine detaillierte Beschreibung des Algorithmus sowie Beispiele für seine Anwendung finden Sie unter https://en.wikipedia.org/wiki/Double_dabble. Machen Sie sich mit dem Algorithmus vertraut und wenden Sie ihn anschließend schrittweise auf die Binärzahl $1011011000_2 = 728_{10}$ an.

Hundert	Zehn	Eins	Binär	Kommentar
0000	0000	0000	1011011000	Initialisierung
0000	0000	0001	0110110000	1. Shift
0000	0000	0010	1101100000	2. Shift
0000	0000	0101	1011000000	3. Shift
0000	0000	1000	1011000000	3 zu Eins addieren, da es 9 war
0000	0001	0001	0110000000	4. Shift
0000	0010	0010	1100000000	5. Shift
0000	0100	0101	1000000000	6. Shift
0000	0100	1000	1000000000	3 zu Eins addieren, da es 5 war
0000	1001	0001	0000000000	7. Shift
0000	1100	0001	0000000000	3 zu Zehn addieren, da es 9 war
0001	1000	0010	0000000000	8. Shift
0001	1011	0010	0000000000	3 zu Zehn addieren, da es 8 war
0011	0110	0100	0000000000	9. Shift
0011	1001	0100	0000000000	3 zu Zehn addieren, da es 6 war
0111	0010	1000	0000000000	10. Shift
7	2	8		Endergebnis

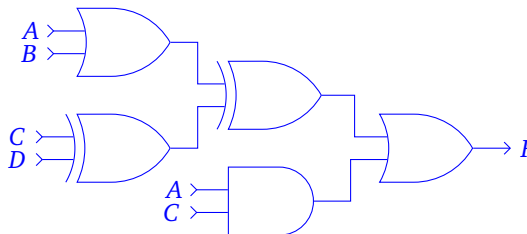
Übung 2.6 Logikgatter-Schaltungen

Implementieren Sie die folgenden Funktionen mit Logikgattern:

a) $F = ((AB) \oplus C) + (A\bar{D})$



b) $F = (AC) + ((A+B) \oplus (C \oplus D))$



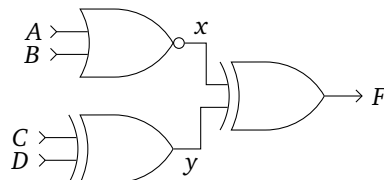
Übung 2.7 Logikgatter-Substitution

Zeichnen Sie eine Logikgatterschaltung, die ein Signal $A \in \mathbb{B}$ invertiert, und ausschließlich aus NAND-Gattern besteht.



Übung 2.8 Wahrheitstabellen

Stellen Sie die Wahrheitstabelle für die folgende Schaltungen auf. Geben Sie dabei auch die Zwischenwerte x und y an.



A	B	C	D	x	y	F
0	0	0	0	1	0	1
0	0	0	1	1	1	0
0	0	1	0	1	1	0
0	0	1	1	1	0	1
0	1	0	0	0	0	0
0	1	0	1	0	1	1
0	1	1	0	0	1	1
0	1	1	1	0	0	0
1	0	0	0	0	0	0
1	0	0	1	0	1	1
1	0	1	0	0	1	1
1	0	1	1	0	0	0
1	1	0	0	0	0	0
1	1	0	1	0	1	1
1	1	1	0	0	1	1
1	1	1	1	0	0	0

Übung 2.9 Multiplexer - Zusatzaufgabe

Ein Multiplexer $\text{MUX} : \mathbb{B}^3 \rightarrow \mathbb{B}$ wird verwendet, um über ein Steuersignal S einen der beiden Eingänge I_0 oder I_1 auszuwählen:

$$\text{MUX}(I_0, I_1, S) = I_S = S ? I_1 : I_0$$

Erstellen Sie eine Wahrheitstabelle für einen solchen Multiplexer und realisieren Sie diese Funktion ausschließlich mit AND, OR und NOT-Gattern.

I_0	I_1	S	F
0	0	0	0
0	1	0	0
1	0	0	1
1	1	0	1
0	0	1	0
0	1	1	1
1	0	1	0
1	1	1	1

