Zusatzaufgabe: Umrechnung der Zahlencodierungen

Geben Sie die Dezimalzahl 2580 in folgenden Zahlen-Kodierungen/Systeme an

a) BCD-Kodierung

Lösung

BCD steht für „binary coded digit“ , also für binär codierte Ziffern.

Die binäre Darstellung jeder einzelnen Ziffer erfolgt mittels Zweierpotenzen:

$$2^0 = 1$$
$$2^1 = 2$$
$$2^2 = 4$$
$$2^3 = 8$$

Beim BCD wird jetzt jede Ziffer der 2580 einzeln nach folgendem Prinzip dargestellt?

2: [0010]

$$2^0 \rightarrow 1 \cdot 0 = 0$$
$$2^1 \rightarrow 2 \cdot 1 = 2$$
$$2^2 \rightarrow 4 \cdot 0 = 0$$
$$2^3 \rightarrow 8 \cdot 0 = 0$$
$$\Sigma = 2$$

5: [0101]

8: [1000]

0: [0000]

⇨ 2580 = [0010 0101 1000 0000]

b) Dual-System

Dual-Kodierung kann wie folgt umgerechnet werden:

2580	: 2 =	1290	*Rest*	0
1290	: 2 =	645	*Rest*	0
645	: 2 =	322	*Rest*	1
322	: 2 =	161	*Rest*	0
161	: 2 =	80	*Rest*	1
80	: 2 =	40	*Rest*	0
40	: 2 =	20	*Rest*	0
20	: 2 =	10	*Rest*	0
10	: 2 =	5	*Rest*	0
5	: 2 =	2	*Rest*	1
2	: 2 =	1	*Rest*	0
1	: 2 =	0	*Rest*	1

Die Spalte der Rest-Werte muss nun von unten nach oben gelesen werden:

⇨ $2580_{(10)} = 101000010100_{(2)}$

Umgekehrt ergibt sich dann:

$$2580 = 1 \cdot 2^{11} + 0 \cdot 2^{10} + 1 \cdot 2^9 + 0 \cdot 2^8 + 0 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 \cdot 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0$$

Wer gut im Kopfrechnen ist und die 2er-Potenzreihe im Kopf hat (1,2,4,8,16,32,64,128,256,512,1024,2048,4096,8192,16384,...), kann natürlich ohne große Rechnung sehen, dass
$2048 = 2^{11}$ die größte Potenz ist, die in 2580 passt,
$512 = 2^9$ die größte Potenz, die in den Rest (532) passt
$16 = 2^4$ noch in den Rest hiervon (20) passt und
$4 = 2^2$ der übrige Rest ist,
also 101000010100 die Dualdarstellung ist!

c) Hexadezimal-System

Im HEX-System werden 16 Ziffern verwendet, für Ziffernwerte größer 9 werden Buchstaben verwendet:

$$\left[0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F\right]$$

$A \triangleq 10$

$B \triangleq 11$

$C \triangleq 12$

$D \triangleq 13$

$E \triangleq 14$

$F \triangleq 15$

Damit folgt für unsere Zahl:

$$\begin{array}{llll} 2580 & :16 = & 161 & Rest \quad 4 \\ 161 & :16 = & 10 & Rest \quad 1 \\ 10 & :16 = & 0 & Rest \quad 10 = A \end{array}$$

Damit ergibt sich in Hexadezimaler Darstellung:

$$\underline{\underline{2580_{(10)} = A14_{(16)} = 0 \times A14}}$$

d) Gray-Kodierung (Fleißaufgabe!)

Bei der Gray-Codierung handelt es sich um einen so genannten einschrittigen Code. Das bedeutet, beim Übergang zum nächsthöheren (bzw. nächstniedrigeren) Zustand wird nur eine Stelle in der Codierung geändert.

Bei den Dualzahlen ändern sich von einer Ziffer auf die nächste unter Umständen mehrere Stellen (Bits) gleichzeitig. Wenn man in einem Register die Bits einer Zifferngruppe ausliest, kann man zufällig auf Zwischenzustände treffen, wo sich noch nicht alle Bits des Registers vollständig geändert haben. Somit würde man falsche Zahlen und eine falsche Ausgabe erhalten. Daher wird oft intern im Gerät bzw. der Codier-Einheit der sog. Gray-Code verwendet, bei dem sich immer nur eine Stelle beim Übergang zwischen zwei aufeinanderfolgenden Zahlenwerten ändert. (siehe auch Vorlesung)

Die Umrechnung von einer Dezimalzahl zur Gray-Codierung funktioniert wie folgt.

Zunächst wird die Dezimalzahl in eine Dualzahl umgerechnet:

$$2580_{(10)} \rightarrow X_1 = 1010\,0001\,0100_{(2)}$$

Anschließend erfolgt eine Linksverschiebung der Stellen, es wird also am Ende einfach eine 0 angehängt:

$$X_2 = 1010\,0010\,1000$$

Dann berechnet man den Rest beim Teilen der Summe von X_1 und X_2 durch 2:

X_1		1	0	1	0	0	0	0	1	0	1	0	0
X_2	1	0	1	0	0	0	0	1	0	1	0	0	0
$Modulo_2(X_1 + X_2)$	1	1	1	1	0	0	0	1	1	1	1	0	$[0]$

Zum Schluss erfolgt noch eine Rechtsverschiebung. Dabei wird die letzte Stelle einfach abgeschnitten (egal ob 1 oder 0):

⇨ $\underline{\underline{111100011110_{(Gray)}}}$

Weitere Beispiele:

X_1		1
X_2	1	0
$X_1 + X_2$	1	1
$Modulo_2(X_1 + X_2)$	1	1
$Gray - Code$	1	

X_1		1	0
X_2	1	0	0
$X_1 + X_2$	1	1	0
$Modulo_2(X_1 + X_2)$	1	1	0
$Gray - Code$	1	1	

X_1		1	1
X_2	1	1	0
$X_1 + X_2$	1	2	1
$Modulo_2(X_1 + X_2)$	1	0	1
$Gray - Code$	1	0	

X_1		1	0	0
X_2	1	0	0	0
$X_1 + X_2$	1	1	0	0
$Modulo_2(X_1 + X_2)$	1	1	0	0
$Gray - Code$	1	1	0	

X_1		1	0	1
X_2	1	0	1	0
$X_1 + X_2$	1	1	1	1
$Modulo_2(X_1 + X_2)$	1	1	1	1
$Gray - Code$	1	1	1	

X_1		1	1	0
X_2	1	1	0	0
$X_1 + X_2$	1	2	1	0
$Modulo_2(X_1 + X_2)$	1	0	1	0
$Gray - Code$	1	0	1	

X_1		1	1	1
X_2	1	1	1	0
$X_1 + X_2$	1	2	2	1
$Modulo_2(X_1 + X_2)$	1	0	0	1
$Gray - Code$	1	0	0	

X_1		1	0	0	0
X_2	1	0	0	0	0
$X_1 + X_2$	1	1	0	0	0
$Modulo_2(X_1 + X_2)$	1	1	0	0	0
$Gray - Code$	1	1	0	0	

Mann erkennt, dass von einem Wert zum anderen jeweils nur 1 Bit sich ändert!