Převod Bin do BCD pomocí Hornerova schématu

Každé číslo ve dvojkové soustavě můžeme vyjádřit výrazem:

$$N = ((a_m*2+a_{n-1})*2+a_{n-2})*2+....+a_0$$

Pokud bychom neaplikovali dekadickou korekci, dostali bychom hodnotu binárního čísla.
Pokud při sčítání aplikujeme dekadickou korekci, je-li hodnota čtveřice bitů větší než 9, přičteme k ní 0110 (6), abychom dostali BCD kód.
BCD kód je desítkové číslo, jehož každá číslice je vyjádřena čtyřbitovým dvojkovým číslem.

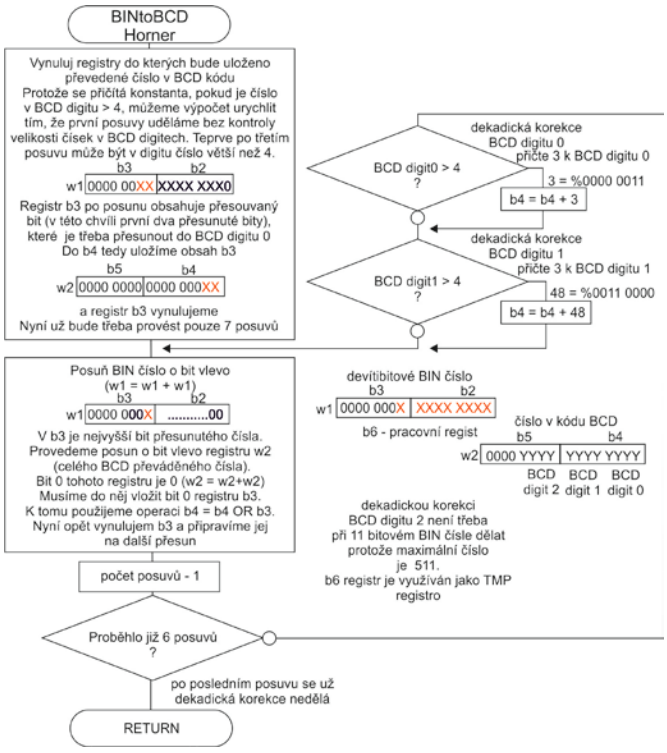
(číslo)10	(číslo)BCD	(číslo)BIN	BIN -->BCD
0	0000 0000	0000 0000	BIN
1	0000 0001	0000 0001	BIN
.....			
9	0000 1001	0000 1001	BIN
10	0001 0000	0000 1010	BIN + 0110 dekadická korekce
11	0001 0001	0000 1011	
.....			
15	0001 0101	0000 1111	
16	0001 0110	0001 0000	
17	0001 0111	0001 0001	
....			
31	0011 0001	0001 1111	
32	0011 0010	0010 0000	
33	0011 0011	0010 0001	
....			

Při převodu z BIN do BCD s využitím Hornerova schématu (http://forum.mcontrollers.com/download.php?id=316&sid=77b33a3c69768b158b104c1d0012d7a4) posouváme číslo vlevo a přitom kontrolujeme, zda čtveřice je větší než 9. Pokud ano, provedeme v ní dekadickou korekci. Posuv děláme tolikrát, kolikati bitové binární číslo převádíme. Postup převodu vidíme v následujících tabulkách.

operace	b100	b10	b1	BIN
				000110110100
shift 1			0	001 1011 0100
shift 4			0001	1011 0100
shift 5		0	0011	011 0100
shift 6		00	0110	11 0100
$+3_1$			+11	
		00	1001	11 0100
shift 7		001	0011	1 0100
shift 8		0010	0111	0100
$+3_1$			+11	
		0010	1010	0100
shift 9	0	0101	0100	100
$+3_{10}$		+11		
	0	1000	0100	100
shift 10	01	0000	1001	00
$+3_1$			+11	
	01	0000	1100	00
shift 11	010	0001	1000	0
$+3_1$			+11	
	010	0001	1011	0
shift 12	0100	0011	0110	
	4	3	6	

operace	b100	b10	b1	BIN
				000111111111
6xshift		00	0111	111111
			+11	
		00	1010	111111
7. shift		001	0101	11111
			+11	
		001	1000	11111
8. shift		0011	0001	1111
9. shift	0	0110	0011	111
		+11		
	0	1001	0011	111
10. shift	01	0010	0111	11
			+11	
	01	0010	1010	11
11. shift	010	0101	0101	1
		+11	+11	
	010	1000	1000	1
12. shift	0101	0001	0001	
	5	1	1	

Na základě tohoto algoritmu vytvoříme konkrétní postup, který bude převod devítibitové hodnoty do BCD kódu řešit.



Navržený algoritmus ověříme simulací v následující tabulce.

nShift	b6	w2		w1		činnost
		bcd3	bcd21	b3	b2	
	0000 0000	0000 0000	0000 0000	0001	11111111	bcd3 = 0
			0000 0000	0111	111111	2 x w1 = w1 + w1
6			0000 0111			bcd21 = b3
	0000 0111		+11			b6=bcd21&00001111
			0000 1010			b6 >4, bcd21+0011
	0000 0000					b6=bcd21&11110000
			0000 1010			b6 <64(0100 0000)
				0000 0000	111111	b3=0
				0001	11111	w1=w1+w1
		0000 0000	0001 0100			w2=w2+w2
		0000 0000	0001 0101	0000 0001	11111	b3>0, w2=w2+1
5		0000 0000	0001 0101			numPosun – 1, >0
	0000 0101		+11			b6=bcd21&00001111
		0000 0000	0001 1000			b6 >4, bcd21+0011
	0001 0000					b6=bcd21&11110000
		0000 0000	0001 1000			b6 <64
				0000 0000		b3=0
				0001	1111	w1=w1+w1
		0000 0000	0011 0000			w2=w2+w2
		0000 0000	0011 0001			b3>0, w2=w2+1
4		0000 0000	0011 0001			numPosun – 1, >0
	0011 0001					b6=bcd21&00001111
			0011 0001			b6 <4
	0011 0001					b6=bcd21&11110000
			0011 0001			b6 <64
				0000 0000		b3=0
				0001	111	w1=w1+w1
		0000 0000	0110 0010			w2=w2+w2
		0000 0000	0110 0011			b3>0, w2=w2+1
3		0000 0000	0110 0011			numPosun – 1, >0
	0000 0011					b6=bcd21&00001111
		0000 0000	0110 0011			b6 <4
	0110 0000		+0011 0000			b6=bcd21&11110000
		0000 0000	1001 0011			b6 >64, bcd21+0011 0000
				0000 0000		b3=0
				0001	11	w1=w1+w1
		0000 0001	0010 0110			w2=w2+w2

		0000 0001	0010 0111			b3>0, w2=w2+1
2		0000 0001	0010 0111			numPosun – 1, >0
	0000 0111		+11			b6=bcd21&00001111
		0000 0001	0010 1010			b6 >4, bcd21+0011
	0010 0000					b6=bcd21&11110000
						b6 <64
		0000 0001	0010 1010	0000 0000		b3=0
				0001	1	w1=w1+w1
		0000 0010	0101 0100			w2=w2+w2
		0000 0010	0101 0101			b3>0, w2=w2+1
1		0000 0010	0101 0101			numPosun – 1, >0
	0000 0101		+11			b6=bcd21&00001111
		0000 0010	0101 1000			b6 >4, bcd21+0011
	0101 0000		+0011 0000			b6=bcd21&11110000
		0000 0010	1000 1000			b6 >64, bcd21+0011 0000
				0000 0000		b3=0
				0001		w1=w1+w1
		0000 0101	0001 0000			w2=w2+w2
		0000 0101	0001 0001			b3>0, w2=w2+1
0		0000 0101	0001 0001		0000 0000	numPosun – 1, >0
		5	1 1			výsledek

Převod Bin do BCD s využitím softwarového dekadického čítače

Tento způsob převodu je jednoduchý ale může trvat relativně dlouho (velký počet instrukčních kroků). Devítibitové číslo může mít maximální hodnotu 511. Obecně je-li n počet bitů binárního čísla, největší hodnota je pak 2^n-1. Počet cyklů, potřebných pro převod je dán právě maximální hodnotou binárního čísla. Pro převod osmibitového čísla je třeba maximálně 511 cyklů. V každém cyklu je provedeno přičtení jedničky (inkrementace) do besítkového čítače. Desítkový čítač realizujeme snadno. Vždy po přičtení jedničky je třeba zjistit, zda je nejnižší digit větší než 9.
Dekadická korekce v prvním případě je následující:
Pokud je, přičteme k němu 6. Poté testujeme následující digit a opět provedeme dekadickou korekci. Je-li jeho hodnota větší než 9 (je 0Ah), přičteme k němu 6.
Dekadická korekce ve verzi 2 je následující:
Je-li číslo v bytu, kde je digit větší než 9 (je 0Ah), digit vynulujeme a do následujícího digitu přičteme 1. Poté provádíme tutéž korekci pro vyšší digit.

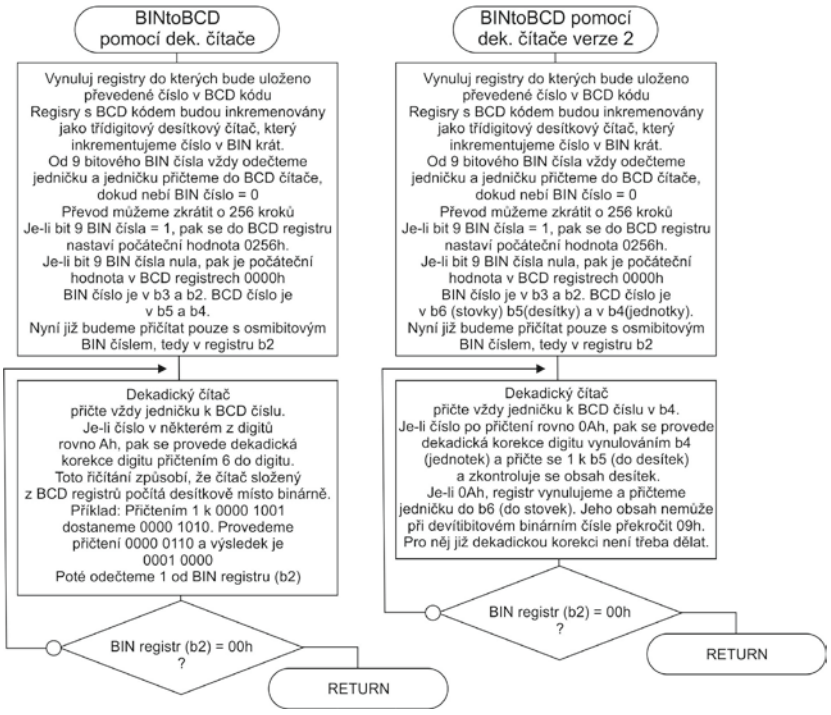
Ukažme si výše uvedené algoritmy na příkladech. Na levo je BCD číslo ve dvou bytech (třech půlbytech), na pravé straně pak ve třech bytech.

cyklus	b100	b10	b1	BIN	operace
	0000	0000	0000	111111111	
1.	0000	0000	0001	111111110	
2.			0010	111111101	
....					
9.		0000	1001	111110110	
10.		0000	1010	111110101	větší než 9
			+0110		přičteme 6
		0001	0000		
11.		0001	0001	111110100	
....					
19.		0001	1001	111101100	
20.		0001	1010	111101011	větší než 9
			+0110		přičteme 6
		0010	0000		
....					
99.		1001	1001	110011100	
100.		1001	1010	110011011	větší než 9
			+0110		přičteme 6
	0000	1010	0000		větší než 9
		+0110			přičteme 6
	0001	0000	0000		
101.	0001	0000	0001	110011010	
....					
511.	0101	0001	0001	000000000	
	2	5	5		

cyklus	b100	b10	b1	BIN	operace
	00	00	00	111111111	
1.			01	111111110	
2.			02	111111101	
....					
9.			09	111110110	
10.			0A	111110101	větší než 9
		+1	00		korekce
		01	00		
11.		01	01	111110100	
....					
19.		01	09	111101100	
20.		01	0A	111101011	větší než 9
		+1	00		korekce
		02	00		
....					
99.		09	09	110011100	
100.		09	0A	110011011	větší než 9
		+1	00		korekce
		0A	00		větší než 9
	+1	00			korekce
	01	00	00		
101.			01	110011010	
....					
511.	05	01	01	000000000	
	2	5	5		

Algoritmus převodu však můžeme zkrátit až o 256 kroků následujícím postupem. Je-li devátý bit v jedničce, bude převáděné číslo 256 a větší. Výchozí hodnotu BCD čísla tedy nastavíme na 256h (0010 0101 0110). Bude-li v nule, pak převáděné číslo bude menší než 255. Výchozí hodnotu BCD čísla tedy nastavíme na 000h (0000 0000 0000). Nyní můžeme inkrementovat výsledek pouze podle dolních 8 bitů čísla.

Konkrétní postup ukazují následující vývojové diagramy. První způsob umožňuje snadné zobrazení, jsou-li výstupy připojeny na dekodéry 7 segmentovýcgh zobrazovacích jednotek, nebo přímo na LED. Verze 2, kdy je každý digit v jednom bytu je vhodný, potřebujeme-li číslo zobrazit na LCD inteligentním display, kdy potřebujeme BCD hodnotu vyjádřit v ASCII kódu.

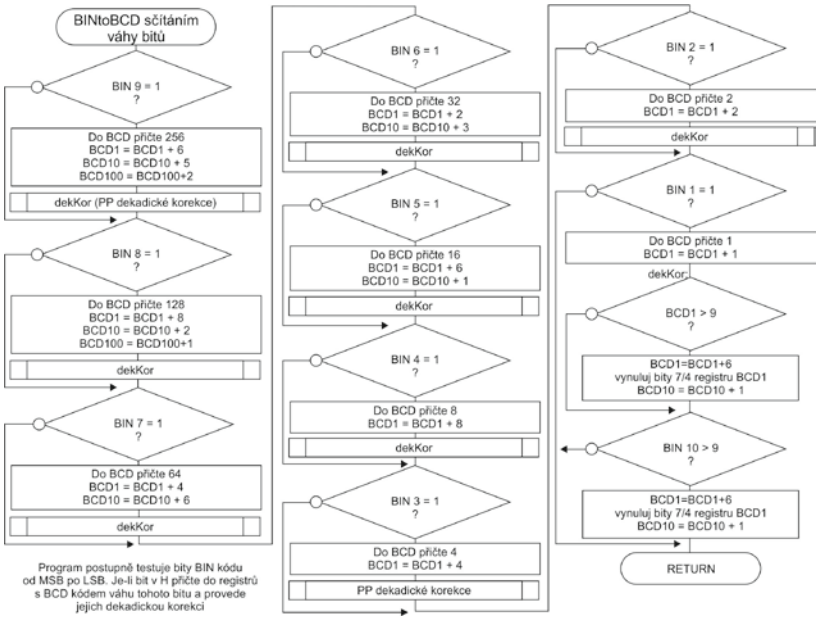


Převod Bin do BCD sčítáním vah bitů s jedničkou

Tento způsob vychází z klasického převodu binárního čísla na dekadické.
Potřebujeme-li vyjádřit desítkovou hodnotu binárního čísla, postupně sčítáme hodnotu bitu s jeho váhou bity číslujeme od 0 po 8 (devítibitové číslo). Bit 0 je LSB a bit 8 MSB.

$$\text{bit } 8*2^8+\text{bit } 7*2^7+\text{bit } 6*2^6+\text{bit } 5*2^5+\text{bit } 4*2^4+\text{bit } 3*2^3+\text{bit } 2*2^2+\text{bit } 1*2^1+\text{bit } 0*2^0 = \text{digit } 2*10^2+\text{digit } 1*10^1+\text{digit } 0*10^0$$

Jednoduše řečeno při převodu desítkově sčítáme váhy bitů binárního čísla, ve kterých je jednička.



cyklus	b100	b10	b1	BIN	operace
	00	00	00	111111111	
1.	+2	+5	+6	111111111	
	02	05	06		součet
2.	+1	+2	+8	111111111	
	03	07	0E		součet
		+1	+6		(b1+6) &0F, b10+1
	03	08	04		
3.		+6	+4	111111111	
	03	0E	08		součet
	+1	+6			(b10+6) &0F, b100+1
	04	04	08		
4.		+3	+2	111111111	
	04	07	0A		součet
		+1	+6		(b1+6) &0F, b10+1
	04	08	00		součet
5.		+1	+6	111111111	
	04	09	06		součet
6.			+8	11111111	
	04	09	0E		součet
		+1	+6		(b1+6) &0F, b10+1
	04	0A	04		
	+1	+6			(b10+6) &0F, b100+1
	05	00	04		
7.			+4	111111111	
	05	00	08		součet
8.			+2	111111111	
	05	00	0A		(b1+6) &0F, b10+1
		+1	+6		součet
	05	01	00		
9.			+1	111111111	
	05	01	01		finální výsledek