

一种基于简单移位的二——十进制相互转换算法

王迎春, 吉利久

(北京大学微电子学研究所, 北京 100871)

摘 要: 十进制码(BCD)与二进制代码相互转换的问题的研究, 主要偏重于软件实现. 本文基于数制变换的基本原理, 提出了移位为基础的、适合硬件实现的转换算法. 并根据该算法, 构造了63位二进制与十进制代码的转换器. 同时, 对该算法又进行了扩充, 提出基 2^r 移位的算法, 进一步提高性能. 从性能的比较可以看出, 该算法速度快, 逻辑简单, 非常适合实时性要求较强的嵌入式领域应用.

关键词: 十进制码; 二进制; 转换; N值逻辑; M值逻辑; 进制

中图分类号: TP333 **文献标识码:** A **文章编号:** 0372-2112 (2003) 02-0221-04

A Shift-based Hardware Converter Algorithm Between Binary & Binary-Coded Decimal (BCD)

WANG Ying-chun, JILi-jiu

(Institute of Microelectronics, Peking University, Beijing 100871, China)

Abstract: It appears many real-time applications with conversion delay between binary and decimal algorithms is a bottleneck of system performance. This paper introduces a kind of shift-based algorithm in order to shorten this delay with minimum cost. Based on the principle of data representation of binary and decimal algorithms, both binary-to-decimal and decimal-to-binary algorithms are deduced mathematically in equation. They are featured as converting data value through shift operation. Double of BCD, constructed to be Logic N, is atomic operation of binary-to-decimal bit-shifting conversion and half of BCD, constructed to be Logic M, is one of reverse conversion. Without carrying propagation from Logic M N to Logic N or from Logic M M to Logic M, the conversion speed only depends on the longest combinatorial path of Logic N or M. These radix- 2^r algorithms, i. e. by one-bit shift per cycle, is expanded to radix- 2^r algorithms, i. e. by multi-bit shift per cycle, to accelerate the conversion speed further. Finally, it shows that the shift-based algorithms are much faster than other algorithms by comparisons of these conversion algorithms.

Key words: BCD; binary; conversion; logic N; logic M; radix

1 引言

微处理器的操作是大量数据的输入——运算——输出的过程, 其中相当数量的数据使用十进制形式表达. 使用者希望微处理器的输入数据和输出结果能使用十进制形式表达, 而在微处理器采用二进制表示和处理数据更方便, 所以在二者之间的数制转换是非常必要的.

随着工业社会的飞速发展, 单片机在工业控制领域中的地位日益重要, 而数码转换是单片机中的一个重要环节. 此外, 数控机床、智能仪表、电子秤、多仪器仪表和资料头头的微机数据采集系统、以及数码管等显示设备也要频繁地用到BCD码与二进制(BIN)码的相互转换^[1].

BCD码与BIN码的相互转换问题, 很早就受到了研究者的重视, 产生了许多算法. 比较传统的方法是采用DAA调节指令来实现, 需要逐位调节, 比较费时. 文[2]提出一种新的转换方法, 利用模除, 每次移动半字节, 使转换效率大为提高. 但

这些也仅仅是软件算法, 无法满足数制转换高效率 and 实时性的要求, 限制了整个系统性能的提高.

文[3]提出了采用分段查找 EPROM 的硬件方法, 解决高速摄影测量仪中 BCD 码到 BIN 码的转换问题, 满足点阵记录实时性的要求. 但是, 该算法要求大容量的 EPROM, 且只适合带宽比较小的码制转换. 文[5]主要利用 5 是 $2^8 - 1$ 和 $2^{16} - 1$ 的因子, 通过严密的数学推导, 提出了用少量的移位和加减操作, 简化实现数制转换中的求余和整除运算. 缺点是需要占用大量的主处理器资源, 不利于整个系统性能的提高. 另外, 在很多高性能微处理器中, 除和求余一般有专门的硬件高效实现, 该算法也不适用. 文[6]利用 BCD 码与二进制码的逻辑对应关系直接转换, 需要经过多级复杂的变换逻辑, 面积和时延较差, 限于低带宽转换.

基于 BCD 码与 BIN 码的数学表述, 本文推导出基于移位的转换算法, 快速准确地实现 BCD 码与 BIN 码的相互转换.

该算法逻辑规整,模块化好,且易向更高进制扩展,有很高的转换速度.不仅数学上验证了该算法的正确性,同时,经过了具体系统实现的功能仿真.

2 基 2 转换算法

2.1 基 2 转换原理

假设 n 是自然数, 2 是某数制系统基数,且

$$x = (x_{n-1}, \dots, x_0), \text{ 其中 } x_i \in \{0, \dots, 2-1\}$$

则基数为 2 的数据 x 为 $[x] = \sum_{i=0}^{n-1} x_i 2^i$

也就是

$$[x] = (\dots((x_{n-1} + x_{n-2}) * 2 + x_{n-3}) + \dots + x_1) * 2 + x_0$$

写成循环的形式:

```

X ← xn-1
for i = n-2 downto 0 loop
    X ← X * 2 + xi
end loop
    
```

其中, $a * b$ 是 $[a] * [b]$ 的缩写, $a + b$ 是 $[a] + [b]$ 的缩写, $a - b$ 是 $[a] - [b]$ 的缩写,即二进制时, $2 = 2$, x_i 表示每个位上的值;BCD 码时, $2 = 10$, x_i 表示 4 位 8421 码.

BIN 码向 BCD 码转换时,由式(1)对数据重新进行构造,形成 4 位组的 BCD 码,即

```

X ← xn-1
for i = n-2 downto 0 loop
    X ← X * 22 + 22 xi ⇔ X ← X * 102 + 10 xi
endloop
    
```

实际上,式(2)中的循环式仅仅是将二进制的表示形式替换为十进制的表示形式,从数值上讲,并没有改变,故利用式(2)就可以完成 BIN 码与 BCD 码的等值转换.这里,核心的问题是 $X * 10_2$ 的运算,下面着重进行讨论.

2.2 BCD 码的倍乘

在进制 2 , 数位由 2 个二进制位构成的数制系统,如 $c = \text{carry}$, (x, y) 表示 x, y 相加产生的进位序列 (c_{n-1}, \dots, c_0) , s 表示产生的和值序列 (s_{n-1}, \dots, s_0) , 有

$$\begin{cases} c_{-1} = 0 \\ c_i = \begin{cases} 0, & \text{if } x_i + y_i + c_{i-1} < 2 \\ 1, & \text{if } x_i + y_i + c_{i-1} \geq 2 \end{cases} \\ s_i = (x_i + y_i + c_{i-1}) \bmod 2 \end{cases} \quad (3)$$

其中, $i \in \{0, \dots, n-1\}$, x_i, y_i, s_i 都是由 2 个二进制位构成.

由文[7]知,如果 a, b 是 BCD 码,那么

$$\begin{aligned} a +_{10} b &= a +_{16} b +_{16} 6 * \text{carry}_{4,10}(a, b) \\ X *_{10} 2 &= [X +_{16} 3 *_{16} \text{carry}_{4,5}(X, 0)] *_{16} 2 = N *_{16} 2 \quad (4) \end{aligned}$$

再由式(3)得,

$$X *_{10} 2 = [X +_{16} 3 *_{16} \text{carry}_{4,5}(X, 0)] *_{16} 2 = N *_{16} 2 \quad (4)$$

由式(2)和式(4),得 $X \leftarrow N *_{16} 2 +_{10} x_i$ (X 是 N 值左移一位后,最末位置 x_i)

2.3 基 2 转换电路

转换电路如图 1 所示, BIN_MSB 是当前剩余二进制位的

最高位; $N(I)$ 在时钟沿有效的时候, 锁存为 $X(I)$, $I \in \{m-1, \dots, 1\}$. 每个时钟周期, 产生一个二进制位, n 位 BIN 码的转换需要 n 个时钟周期.

同时, 为了避免转换过程中出现溢出, 要求 l 位 BCD 码 ($m = 4 * l$) 能够表示的最大数不小于 n 位 BIN 码能够表示的最大数, 即 $10^l - 1 \geq 2^n - 1, l \lceil \lg 2^n \rceil (l, n \text{ 是自然数}) \quad (5)$

转换电路主要是由 BCD 码的倍乘调整逻辑, 即式(4)的 N 值逻辑构成, 特点为:

(1) 从式(4)知, BCD 码形式的 X 与 3 倍的 Carry 序列相加, 实际是 X 中各 8421 码组与 10011_2 或 10000_2 相加. 由于 8421 码的最大取值仅为 1001_2 , 故不会产生向更高位的进位传递. 于是, 图 1 中的 Logic N (N 值逻辑) 可以看成 4 位一组的独立单元, 没有组间进位传播, 逻辑简化, 时钟频率提高.

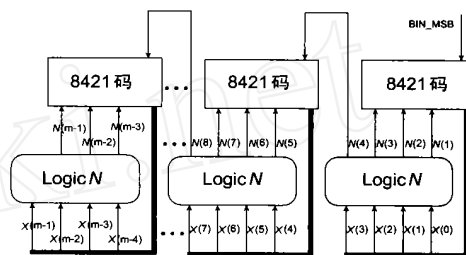


图 1 BIN 转 BCD 电路

(2) Logic N 如图 2 所示, l 个 Logic N 需要 l 个比较器, l 个二选一逻辑和 l 个四位加法器构成. 如果将比较器用 +3 加法器代替, 那么 Logic N 可以优化为图 3 的形式, 整个系统可以省掉 l 个比较器.

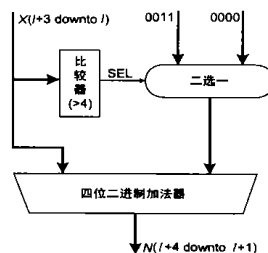


图 2 Logic N 逻辑图

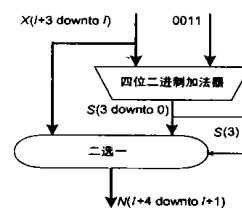


图 3 改进的 Logic N

3 基 2 逆转换算法

3.1 基 2 逆转换原理

如果 $X = (x_{l-1}, \dots, x_0)$, $x_i \in \{0000, \dots, 1001\}$, l 是自然数,

$$\text{则有 } [X]_{10} = \sum_{i=0}^{l-1} x_i 10^i.$$

式(6)表示 X 转换 BIN 码的过程, 最后, 产生的 $\{b_i | i \in [n-1, 0]\}$, 且 i 为整数 l 构成了相应的 BIN 码.

for $i = 0$ to $n-1$ loop

$$b_i \leftarrow [X]_{10} \bmod [10]_2$$

$$X \leftarrow X - b_i$$

(6)

$$X \leftarrow X *_{10} \left(\frac{1}{2}\right)$$

endloop

式(6)的迭代式中, 第一个运算是取 X 的最低位作为 b_i ,

第二个运算是置最低位为 0,第三个运算是关键的 BCD 码半乘操作。

3.2 BCD 码的半乘

式(4)中,如果令 $Y = X *_{10} 2$,那么

$$Y *_{16} (1/2) = Y *_{10} (1/2) + 16 *_{16} \text{carry}_{4,5}(X, 0) \quad (7)$$

依据式(3),知道 X 的数位 x_i 和 $Y *_{16} (1/2)$ 的数位 y_i 的关系 ($i \in \{0, \dots, n-1\}$):

$$y_i = \begin{cases} x_i + [0011]_2 + c_{i-1}, & x_i \geq 5 \\ x_i + c_{i-1}, & x_i < 5 \end{cases} \quad (8)$$

其中, c_{i-1} 是从低位来的进位,且 $c_{-1} = 0$ 。前文已经讨论过, x_i 的最大取值是 9,对应的加 3 操作不会形成进位,故 c_{i-1} 为 0。这样, y_i 只与对应的 x_i 相关。从式(8)知,下面的关系成立:

$$\left. \begin{aligned} x_i \geq 5 &\Rightarrow y_i \geq 8 \\ x_i < 5 &\Rightarrow y_i < 5 \Rightarrow y_i < 8 \end{aligned} \right\} \Rightarrow \text{carry}_{4,5}(X, 0) = \text{carry}_{4,8}(Y *_{16} (1/2), 0) \quad (9)$$

式(9)代入式(7)且 Y 换成 X ,得

$$\begin{aligned} X *_{10} (1/2) &= X *_{16} (1/2) - 16 *_{16} \text{carry}_{4,8}(X *_{16} (1/2), 0) \\ &= M \end{aligned} \quad (10)$$

从整个 BCD 向 BIN 码的转换过程来看,实际是 BIN 码向 BCD 码转换的逆过程,故称为逆转换。

3.3 基 2 逆转换电路

与基 2 转换类似,逆转换电路也是在 BCD 码侧进行变换,BIN 码侧仅仅是一个简单的移位寄存器,故没有画出,如图 4。 $X(0)$ 是 BIN 码寄存器中最高位的输入,在有效时钟沿的触发下,整个 BIN 码寄存器左移, $X(0)$ 移入最高有效位。 $M(I)$ 在时钟沿有效的时候,锁存为 $X(I)$, $I \in \{m-1, \dots, 0\}$ 。每个时钟周期转换一个二进制位,所以,需 n 个时钟周期产生 n 位 BIN 码结果。

同样,防止出现溢出, m 位宽 BCD 向 n 位宽 BIN 码转换,要保证式(11)关系成立。

$$n \lceil l \rceil \log_2 10 \quad (11)$$

其中, l, n 是自然数,且 $m = 4 * l$ 。

逆转换电路主要由 BCD 码半乘逻辑,即 M 值逻辑(Logic M)构成,特点为:

(1) M 值逻辑没有向高位的借位传递,可以构成 4 位一组的独立单元,如图 4 所示,提高了转换速度。

(2) Logic M 的结构有图 5 和图 6 两种选择。图 5 中的二选一逻辑是常数之间的二选一,二选一逻辑可以得到很大的

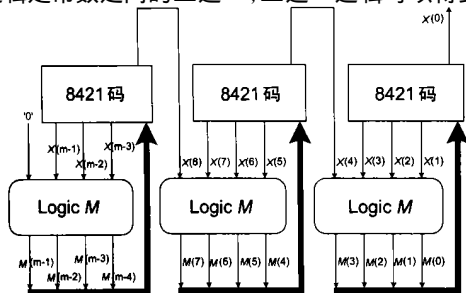


图 4 BCD 转 BIN 电路

简化;图 6 中是相对于常量的减法器,能够有非常优化的减法器设计。但减法器的开销比二选一逻辑大很多,故图 6 是较优化的设计。

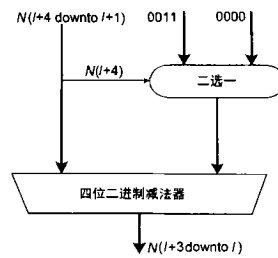


图 5 Logic M 结构图

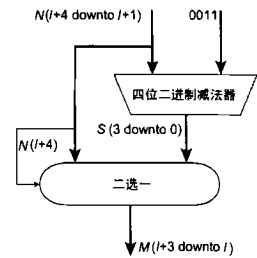


图 6 Logic M 优化结构图

4 基 2 算法扩展

从基 2 算法的实现可以看出,运行时间与转换的 BIN 码的位数成正比,一般 BIN 码的位数就是转换的时钟周期数。在位数很大的情况下,会需要很长的转换时间,往往无法满足一些应用环境的实时性要求。于是,对基 2 算法扩展,提出了基 2^r (r 为不小于 2 的自然数)算法。

4.1 基 2^r (r 为不小于 2 的自然数)算法

将基 2 转换算法中相邻的 r 个迭代压缩为一个迭代,就产生了基 2^r 的算法。BIN 码向 BCD 码转换时,将相邻 r 个迭代压缩后,就变成式(12)所示的迭代形式。基 2^r 逆变换算法也是将相邻的 r 个 BCD 半乘迭代合并,从式(6)和(10)得到一个与式(12)形式类似的迭代过程,这里就不再赘述。

$X \leftarrow 0$

for $i \leftarrow \lceil \frac{n}{r} \rceil$ downto 1 loop

$$\left. \begin{aligned} T_0 &\leftarrow [X + 16^3 *_{16} \text{carry}_{4,5}(X, 0)] *_{16} 2 + 10 *_{10} x_i *_{r-1} \\ T_1 &\leftarrow [T_0 + 16^3 *_{16} \text{carry}_{4,5}(T_0, 0)] *_{16} 2 + 10 *_{10} x_i *_{r-2} \\ &\dots \\ &\dots \\ T_{r-2} &\leftarrow [T_{r-3} + 16^3 *_{16} \text{carry}_{4,5}(T_{r-3}, 0)] *_{16} 2 + 10 *_{10} x_{(i-1)} *_{r-1} \\ X &\leftarrow [X_{T-2} + 16^3 *_{16} \text{carry}_{4,5}(T_{r-2}, 0)] *_{16} 2 + 10 *_{10} x_{(i-1)} *_{r-1} \end{aligned} \right\} \begin{array}{l} r \text{ 个} \\ \text{变} \\ \text{换} \\ \text{成} \\ 1 \text{ 个} \end{array} \quad (12)$$

4.2 基 2^r 电路

电路实现上,基 2 算法是 N 值逻辑变换,而基 2^r 算法是连续的 r 个 N 值逻辑串联,定义为 N^r 值逻辑。相应的,基 2^r 逆变换算法也是连续的 r 个 M 值逻辑串联,称为 M^r 值逻辑。

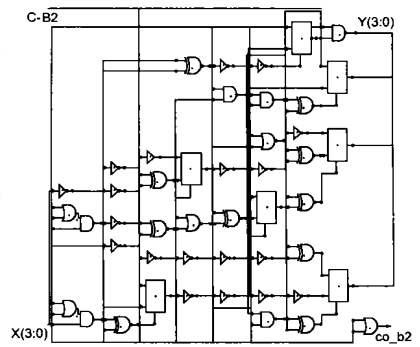


图 7 N^2 值逻辑

图 7 所示为 N^2 值逻辑,应用该逻辑的基 2^r 算法转换为 BIN 码只需要 $\lceil n/2 \rceil$ 个时钟周期,但最大组合路径延时也随之变大.总之,由式(12)可以看出,基 2^r 算法不论是 BCD 转 BIN 还是 BIN 转 BCD,所需迭代周期只有 $\lceil n/r \rceil$ (n 为 BIN 码的带宽). r 值的决定,最终是系统设计取舍的结果. (X 是输入 BCD 码, Y 是调整后的 BCD 码, C_{b2} 是第二级 N 变换的进位输入,也是上一个模块的第一级 N 变换的进位输出,即 CO_{b2} .)

5 性能比较

面向实际的系统,对基 2、基 4、基 8 的二—十进制相互转换电路进行了构造,在 alcatel 0.35 μ m 工艺下实现,性能指标如表 1 和表 2 所示.

表 1 是 63 位 BIN 向 76 位 BCD 转换系统的性能指标.该系统含有一个移位计数器,产生转换结束信号;这时,外部控制打开 BCD 码的三态驱动,输出结果.表 1 中基 8 算法功耗最低,转换速度最快;基 2 算法面积和主频最优;基 4 算法介于二者之间.

表 1 BIN 转 BCD 电路性能

| 算法类型 | 最长组合路径(ns) | 转换周期数 | 面积(μ m ²) | 平均动态功耗(毫瓦) |
|--------|------------|-------|----------------------------|------------|
| 基 2 算法 | 2.35 | 63 | 2350.520020 | 59.6013 |
| 基 4 算法 | 3.89 | 32 | 2931.340088 | 54.7119 |
| 基 8 算法 | 5.30 | 21 | 3626.820312 | 50.6854 |

表 2 BCD 转 BIN 电路性能

| 算法类型 | 最长组合路径(ns) | 转换周期数 | 面积(μ m ²) | 平均动态功耗(毫瓦) |
|--------|------------|-------|----------------------------|------------|
| 基 2 算法 | 2.09 | 60 | 2145.189941 | 194.0875 |
| 基 4 算法 | 2.91 | 30 | 2138.040039 | 64.7211 |
| 基 8 算法 | 3.79 | 20 | 2700.829834 | 76.0473 |

表 3 移位转换算法与其它算法转换速度比较

| BIN 转 BCD | | BCD 转 BIN | |
|--------------------------|----------------|-------------------------|----------------|
| 算法类型 | 二进制位转换率(位转换时间) | 算法类型 | 二进制位转换率(位转换时间) |
| 常用算法(单片机) ^[4] | 119 μ s | EPROM 分段 ^[3] | 5.5 μ s 以内 |
| 改进算法(单片机) ^[4] | 96.4 μ s | 基 2 算法 | 2.09 ns |
| 基 2 算法 | 2.35ns | 基 4 算法 | 1.46ns |
| 基 4 算法 | 1.95ns | 基 8 算法 | 1.26ns |
| 基 8 算法 | 1.77ns | | |

表 2 是 72 位 BCD 向 60 位 BIN 转换系统的性能指标.该系统中没有移位计数器,而是通过初始化 BIN 码寄存器为“10……00”,然后用最低位作为转换结束信号.这时,外部控制打开 BIN 码的三态驱动,输出结果.从表 2 可以看出,面积和功耗方面,基 4 算法有比较好的性能;基 2 算法主频最高;基 8 算法转换时间最短.

与软件实现的单片机常用算法和改进算法,以及硬件的 EPROM 分段算法比较,即使本文提出的基 2 算法(最慢的一种)也比它们的转换速度高很多,如表 3 所示.这里,由于移位基础转换算法的关键路径就是相应的 N 、 M 值或 N' 、 M' 值逻辑的最长组合路径,故位转换率(转换每个二进制位需要的时间)与转换的位数无关,而由基值(每次迭代移动的位数)决定.

6 总结

本文提出了基 2 算法以及扩展的基 2^r 算法,来解决微机系统输入输出时经常面临的进制转换问题.该算法实现上逻辑规整,模块化比较好,易于扩展;同时没有长进位链,转换速度快,非常适合二—十进制转换实时性要求较高的应用.

参考文献:

- [1] 周启明,等.多个仪器仪表和数字表头的微机数据采集系统[J].自动化与仪表,2000,15(3):51-57.
- [2] 潘昊,等.单片机十六进制数与 BCD 码转换新探讨[J].微机发展,1997,6:35-36.
- [3] 刘波,吴圣雄.高速摄影测量仪点阵码制的快速转换[J].光子学报,1998,27(3):270-275.
- [4] 金应哲.剖析多字节二进制整数转换为 BCD 码的子程序[J].测控技术,1997,16(2):53-54.
- [5] Arazi, B, Naccache, D. Binary-to-decimal conversion based on the divisibility of $2^8 - 1$ by 5[J]. Electronics Letters, 1992, 28(23):2151-2152.
- [6] Guy, B M, III. VLSI design of a custom ASIC using VHDL for converting 12-bit binary to BCD[A]. Proc. of the 27th Southeastern Symp on System Theory[C], 1995.
- [7] Andreas Doring, et al. Decimal adjustment of long numbers in constant time[J]. Information Processing Letters, 1997, 62:161-163.

作者简介:

王迎春 男,1973 年 2 月生于陕西韩城,2000 年获西北工业大学计算机应用专业博士学位,现在北京大学微电子所从事博士后研究,研究领域包括:计算机体系结构、处理器微体系结构、IP 建库和 SoC 设计等.