

# 一种非写回整数除法器的并行结构设计

刘 冀

(同济大学微电子中心,上海市 200092)

**摘 要:**嵌入式微处理器是近年来国内研究的热点之一,如何以合适的成本实现高效的硬件除法单元是其中的一个技术难点。针对嵌入式微处理器设计的要求,介绍一种基于标准部件的整数除法器。电路用 1 个标准 64 位加法器、3 个 64 位寄存器和 3 个 64 位多路选择器为主体实现非写回除法算法,在 0.09  $\mu\text{m}$  工艺下以全定制方法实现的数据通道部分仿真时延为 0.92 ns。另外,针对多周期数字系统的基本结构之一硬件循环结构介绍一种逻辑优化方法。

**关键词:**除法器;非写回算法;并行结构;硬件循环结构;循环展开

**中图分类号:** TN402

## 0 引 言

MPU(微处理器)设计以高效性和通用性为考虑,一般在内部配备独立执行器件实现基本算术运算。其中除法实现的技术难度最大,软件中应用频度又比较低<sup>[1]</sup>,对注重低成本、高性价比的嵌入式 MPU 来说,结构简洁的基本除法器就成为众多架构的广泛选择。本文以低成本简约化设计为着眼点,实现一种基于非写回算法的 64 位除法器结构。采用系统设计中的标准逻辑部件,不论用全定制或用自动布线方法都能快速完成后端设计,节约设计成本。另外,可以加入数据对准逻辑,根据操作数大小用不同周期数完成运算,可以大幅提高性能。

并行是同步数字系统前端设计的核心概念,本文结合设计实例对其进行了一般性的讨论。基本除法器是硬件循环结构的典型代表,针对它的设计介绍了一种逻辑优化方法。方法的概念来自软件优化理论。在本文中可以看到,一种称为展开并压缩(unroll-and-jam)<sup>[2]</sup>的软件并行度优化方法,只要对其概念稍加改动,也可应用于硬件系统。

### 1 算法分析

基本除法器的概念来源于基本除法算法(basic division schemes)<sup>[3]</sup>。基本除法算法大体上分写回算法(restoring algorithm)和非写回算法(non-restoring algorithm)两种。平时笔算用的是写回算法,非写回算法因为逻辑结构更简单而多被计算机硬件采用<sup>[3]</sup>。对于 k 位有符号数来说,补码格式是用  $2^k - M$  来表示  $-M(M > 0, k$  为数据的位宽)。被除数如果为负,需要

先求补变为正数进行运算,再对结果进行调整得到商和余数,否则计算的是  $(2^k - M) \div N$  而不是  $(-M) \div N$ 。对非写回算法来说,运算过程可以分为 3 部分:被除数符号调整;循环加/减;结果调整。结果调整又分余数调整、余数符号调整和商调整 3 部分。

除法器状态如表 1 所示。

表 1 除法器状态

周期	描述	操作
IN IT	初始化(被除数调整)	如果被除数 $< 0$ , 执行以下步骤 a) 被除数求补, 结果存入商寄存器 $R_q$ b) 除数存入除数寄存器 $R_d$ c) 部分余数寄存器 $R_p$ (含符号位) 清零, 即首轮部分余数 $P_{64}$ 为 0 d) $\{R_p, R_q\}$ 整体左移 1 位, 原符号位舍弃 如果被除数 $> 0$ , 将被除数直接存入 $R_q$ , 然后执行步骤 b、c、d
ITER	循环 (63 次)	a) 若上一轮部分余数 $P_{n+1}$ 与除数同号则减去除数; 否则加上除数; 符号位不再产生进位(舍弃), 结果为本轮部分余数 $P_n$ b) 若 $P_n > 0$ 本轮商 1, 否则商 0; 商 $Q_n$ 填入上周期 $R_q$ 左移留下的空位 c) $\{R_p, R_q\}$ 整体左移 1 位(符号位舍弃)
FINL	最后一轮循环	$P_0$ 不再移位, 其他同 ITER
RA	余数调整	如果需要, $P_0$ 加上除数的绝对值
RSA	余数符号调整	如果需要, RA 结果求补得到余数
QA	商调整	如果需要, 把 $\{Q_{63}, Q_{62}, \dots, Q_0\}$ 求补得到商

收稿日期: 2006-07-17; 修回日期: 2006-09-28。

在运算开始前,两个操作数都需要补充一位符号位(符号扩展),使它们成为 65 位数。这是为了兼容有符号格式与无符号格式。符号位的 0 表示正数,包括有符号正数与无符号数,1 表示负数。按照补码格式,0 与正数进行同样处理。在运算过程中,符号位的值与其他位一样需要用硬件保存。所有的处理周期可分为两类:一类是运算周期,包括表 1 中的 ITER 与 FINL;另一类是调整周期。运算周期的计算结果  $P_n$  是部分余数 (partial remainder)。运算周期总是需要的,周期数等于操作数(被除数、除数)的位数。调整周期视情况而定,可以不需要。调整周期见表 2。

表 2 调整周期

case	sa	sb	sRO	RA	RSA	QA
c1	+	+	+	n	n	n
c2	+	+	-	y	n	n
c3	-	+	+	n	y	y
c4	-	+	-	y	y	y
c5	+	-	+	n	n	y
c6	+	-	-	y	n	y
c7	-	-	+	n	y	n
c8	-	-	-	y	y	n

注: sa 为被除数的符号; sb 为除数的符号; sRO 为最后一轮部分余数 RO 的符号; y 为需要; n 为不需要。

这些结论可以用二进制数学的方法加以证明<sup>[3]</sup>。本文只做如下简要说明:

a) 余数调整是为了保证余数与被除数同号。因为循环开始时部分余数为正,最后一轮部分余数也必须为正。否则需要加上除数的绝对值。

b) 如果  $R_0$  需要做余数调整(加上或减去除数),商也需要相应加 1 或减 1。商调整(QA)实际上是这个操作与商符号调整合二为一。这样做的合理性同样可以用数学原理来证明。

2 电路设计

2.1 主要部件

同步数字系统电路结构的设计一般按照“运算/核心部件→存储部件→附加逻辑”的顺序。先由数据出发,以静态观点确定主要功能部件的数目,再以动态观点考虑附加逻辑的结构。考查表 1 中所列,主要操作(加、减、求补)都可以用一个标准加法器完成。由于前后数据的相关性,大部分操作只能串行。根据补码的特性,减一个数等于加上它的补码,实际操作是“取反加 1”,硬件结构见图 1。可以看出,除数不需单独进行符号调整,可以在运算过程中动态处理。

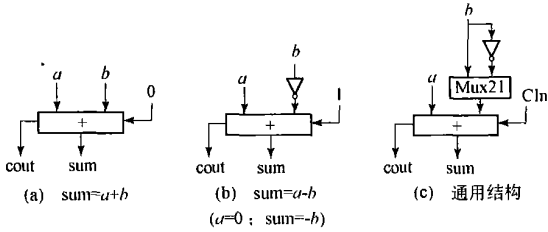


图 1 加法器及辅助逻辑

再看存储部件寄存器。由表 1 可以知道,除去符号位之外,数据通道中任一时刻有效数据的总位宽最多为  $64 \times 3$  位。循环中产生的商可以填入部分余数左移留下的空位,所以共需要 3 组 64 位寄存器,即部分余数寄存器  $R_p$ 、除数寄存器  $R_d$  和商寄存器  $R_q$ 。

2.2 单周期并行设计

余下的附加逻辑部分是并行设计的关键。并行可按时序观点分为一周期内的并行和跨周期的并行两种,前者体现在商选择逻辑。

基本除法算法的结构决定了商逻辑必然是关键路径的一部分。由表 1 可见,商由本轮部分余数的符号位取反得到,所以商逻辑可以看做是一个没有进位输出的全加器,实际可以用两个串行的异或门实现。图 2 是两种实现方式的区别。同样的器件,图 2(b)利用了算法中包含的并行度,获得了更短的时延。

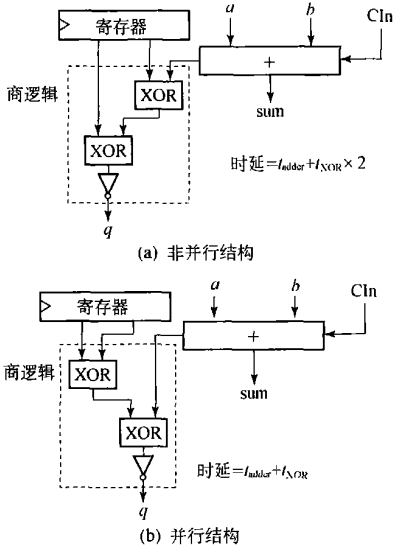


图 2 商逻辑的结构

需要指出的是,并行是建立在不相关性基础上的,或简单地说不相关才可以并行,这是传输门级并行与指令级并行相似的地方。对前者来说是指输入数据不相关。图 2(b)中商逻辑来自寄存器的两个输入与加法器的两个输入之间没有逻辑联系,所以加法器与第 1 个异或门之间存在并行度。这段逻辑的时延是两者

中较长的一个,即加法器的时延。图 2 体现的原则适用于所有同步数字系统设计。

2.3 跨周期并行设计

跨周期并行是一种简单的说法。对多周期同步数字系统来说,有时分属不同周期的逻辑之间可以并行。在一定条件下,可以通过改变系统结构,调整逻辑的时序划分来达到并行的目的。图 3 的设计过程应用了一种对硬件循环结构进行逻辑优化的重要方法——硬件时序展开法。

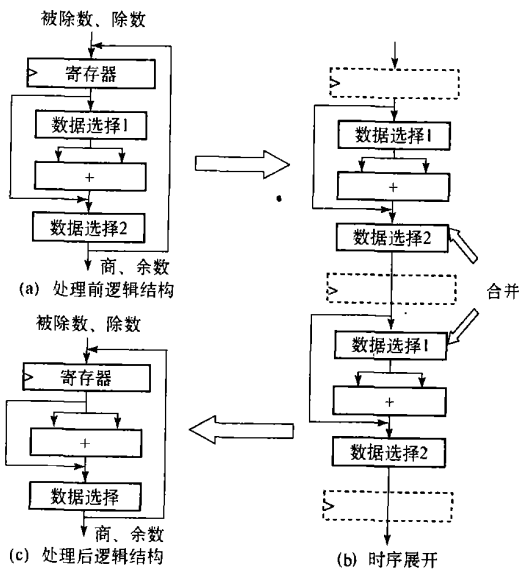


图 3 整体结构的时序优化

以加法器为核心并沿着用一套硬件完成所有操作的思路,自然可以得到图 3(a)的数据通道结构。第 1 级数据选择逻辑为后面的加法器准备合适的操作数,例如将除数取反。后面一级数据选择逻辑对加的结果进行处理,例如补商移位。数据选择逻辑一般由传输门构成,串连的多级可以合并,具体如图 4 所示。

图 4 演示了组合逻辑结构优化的基本概念,把两级串行的二选一结构简化为一级三选一结构。把图 3(a)的结构在时序上展开,抽掉时序部件(寄存器),对组合逻辑进行并行处理后在适当位置重新加入,处理后的结构比处理前少了一级数据选择逻辑。

需要补充说明的是,对于图 2 所演示的组合逻辑优化过程,一般可以借助综合工具进行;而图 3 演示的时序优化过程目前必须手工完成,因为目前各种综合软件都是在系统结构基础上进行优化,而不会改变系统结构自身,这显示了时序优化方法的重要性。时序展开法的基本过程可以总结为“展开→压缩→重新整合”。这种方法同样适用于相同类型的其他系统。

最后经过验证的数据通道电路整体结构符合图 3(c)。在运算周期中,寄存器  $R_p$  输出经过移位的部分

余数,  $R_q$  存放被除数低位和商,  $R_d$  输出动态调整后的除数。调整周期中,  $R_p$  输出需要调整的数据,  $R_d$  输出调整后的除数或者 0。

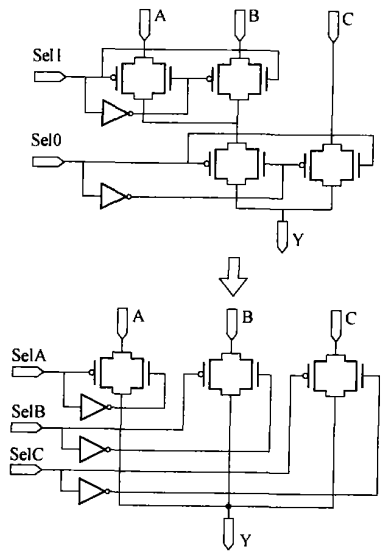


图 4 数据选择逻辑的简化

为了提高除法器性能,还可以在前述结构的基础上加入数据对准逻辑(Aligner)。数据对准需要单独用一周完成,这个周期位于  $INIT$  与  $ITER$  之间。对准器将操作数的实际宽度分为 8 段。当检测到调整后被除数的高 8 位(或高 16 位, 24 位, ...)全为 0 时就左移相应的位数,并通知控制逻辑在后面的运算中省去相应的周期,整个除法器就会根据操作数的实际大小以  $8k+5$  周期完成运算。对准器的设计也很好体现了组合逻辑的并行原则,具体可以用多级反相输入的与非门实现,所用器件都是后端设计的标准器件。

在 90 nm 工艺下使用仿真工具 eLdoD 对全定制方法实现的除法器结构进行仿真,在使用慢速模型、温度设置为 25℃ 时得到的时延结果为 0.92 ns(加法器时延 0.76 ns)。本文实现的除法器可以应用于最高工作频率为 1 GHz 左右的微处理器中。

3 结束语

在国内力求于嵌入式 MPU 领域有所突破之际,对其内部功能单元的结构进行研究有重要的意义。除法运算在软件中的使用频度小而硬件实现成本大,因此一些嵌入式 MPU 架构内部不配备独立的除法单元,如 ARM® 系列微处理器,而配备除法单元的大多采用结构简洁实用的基本除法器以控制成本,如 MIPS® 系列嵌入式微处理器。在 MIPS64 系列某些型号(如 MIPS 5k)中,除法单元的功能特性与本文所述相同。

(下转第 24 页)

tenuators[J]. IEEE Trans on Microwave Theory and Techniques 1994, 42(7): 1140-1148.

[2] FRANCO D P. Networks and devices using planar transmission lines[M]. Boca Raton, FL, USA: CRC Press, 2000.

[3] WADELL B C. Transmission line design handbook[M]. Boston, MA, USA: Artech House, 1991.

[4] HARPER C A. Electronic packaging and interconnection handbook[M]. New York, NY, USA: McGraw-Hill, 1997.

[5] GIPPRICH J W, DICKENS L E, HAYES R E, et al. A compact

LTCC multilayer multiport stripline coupler network for wideband low loss power combining/splitting[C]//Proceedings of IEEE 3rd Topical Meeting on Electrical Performance of Electronic Packaging, Nov 2-4, 1994, Monterey, CA, USA. Piscataway, NJ, USA: IEEE, 1994: 167-169.

符 鹏 (1974-), 男, 工程师, 主要从事微波器件与电路的设计与工艺工作。

The Directional Coupler in X-band T/R Modules

FU Peng

(Nanjing Research Institute of Electronics Technology, Nanjing 210013, China)

**Abstract:** This paper presents an approach to develop a directional coupler which picks up and provides the indication signal in X-band T/R modules. The coupler and an embedded 50 Ω resistor terminal are designed into LTCC (Low Temperature Co-fired Ceramic) multilayer substrate of the X-band high-integrated encapsulated module. The input, output and couple ports are coplanar waveguide style, for measured with probe station and microwave VNA.

**Keywords:** directional coupler; X-band; LTCC; resistor terminal

(上接第 17 页)

本文的设计过程强调并行的概念。在后端设计成本一定的情况下, 必须采用先进的算法和高度优化的系统结构以获得最快的运行速度。在设计过程中对这一概念和相关的设计方法进行初步讨论。设计方法是所有系统通用的, 它与除法器结构本身同样具有实用价值。

参 考 文 献

[1] PATTERSON D A, HENNESSY J L. Computer architecture: a quantitative approach[M]. 3rd ed. 北京: 机械工业出版社, 2002.

[2] CARR S, DING C, SWEANY P. Improving software pipelining

with unroll-and-jam[C]//Proceedings of 29th Annual Hawaii International Conference on System Sciences, Vol. 1, Jan 3-6, 1996, Maui, HI, USA. Los Alamitos, CA, USA: IEEE Computer Society, 1996: 183-192.

[3] PARHAM I B. Computer arithmetic: algorithms and hardware designs[M]. New York, NY, USA: Oxford University Press, 2000.

[4] DATTEYSN D A, HENNESSY J L. Computer organization and design: hardware/software interface[M]. 3rd ed. 北京: 机械工业出版社, 2006.

刘 冀 (1982-), 男, 硕士研究生, 主要研究方向为微处理器设计。

Parallel Architecture Design of an Integer Non-restoring Divider

LIU Ji

(Tongji University, Shanghai 200092, China)

**Abstract:** This paper presents a general description of how to form a 64-bit non-restoring integer divider mainly using 1 standard 64-bit adder, 3 64-bit flip-flops and 3 64-bit multiplexers. The full-custom-designed datapath using 0.09 μm technology has reached a time delay of 0.92 ns. Along with the architecture, an optimization method applied effectively to hardware loop structure is introduced as well.

**Keywords:** divider; non-restoring algorithm; parallel structure; hardware loop structure; loop unroll