

基于 LDPC 译码的容错设计方法研究^①

王登超^② 班 恬^③ 薛 文

(南京理工大学电子工程与光电技术学院 南京 210094)

摘 要 半导体工艺尺寸的急剧缩小使得提升数字电路容错能力变得尤为必要。传统的三模冗余会带来面积和功耗的巨大开销。本文提出一种信息冗余的容错设计方法,采用低密度奇偶校验(LDPC)码实现。由于数字电路中概率门模型的出错的概率 ε 与二进制对称信道(BSC)的噪声相似,因此将通信系统中的信道编译码方法应用到数字电路容错设计中。该设计在由逻辑门组成模块的输入输出端添加 LDPC 编译码。本文的 LDPC 码以 1/2 码率、816 码长为例,并采用加权比特翻转(WBF)算法进行译码器设计。在 $\varepsilon \in (0.0001, 0.1)$ 范围内,对硬件冗余及信息冗余的容错结构进行了仿真和性能比较。误比特率(BER)曲线表明,当 ε 大于 0.05 时,硬件冗余比信息冗余的 BER 低;反之,信息冗余的 BER 低于硬件冗余。

关键词 数字电路,可靠性,低密度奇偶校验(LDPC),硬件冗余,信息冗余

0 引 言

半导体集成电路凭借其体积小、功耗低、功能强等优势,已经被广泛应用于航空航天、移动通信、计算机系统、仪器仪表等领域。随着半导体制造技术的不断发展,数字电路中晶体管的尺寸不断缩小,供电电压急剧下降,节点的关键电荷随之减少,这些均导致软错误率(soft error rate, SER)不断升高。软错误会导致数字电路中存储单元发生位翻转^[1],这严重影响着数字电路可靠性。软错误一般是由高能粒子碰撞和电磁干扰引起的瞬态故障。作为软错误的主要贡献者,单粒子翻转(single event upsets, SEUs)不仅存在于空间环境中,随着工艺不断进步半导体器件对辐射粒子的敏感性急剧上升,其在低空和地面环境中也逐渐成为焦点^[2]。因此,数字电路可靠性面临更严峻的挑战,使得数字电路容错设计尤为必要。

基于冗余概念的各种容错结构设计可以提高数

字电路的可靠性^[3],这些冗余结构已经被应用于晶体管级、门级、系统级等不同的数字电路级别^[4-6]。基于冗余概念的容错结构大致分为 4 类:硬件冗余、信息冗余、时间冗余和软件冗余容错结构。本文主要讨论硬件冗余和信息冗余容错结构的应用场合,其中硬件冗余包括三模冗余(triple modular redundancy, TMR)和五模冗余(five modular redundancy, FMR)等。TMR 因提高数字电路可靠性的效率较高而被普遍采用^[7,8],但是,通过对数字电路功能模块添加额外的冗余功能模块不可避免地降低了纳米级集成电路中器件的有效密度。而本文提出的冗余容错结构设计方法属于信息冗余,使用低密度奇偶校验码(low density parity check, LDPC)编码器产生冗余信息,与此同时,使用 LDPC 译码器纠正错误信息,从而达到提高数字电路可靠性的目的。这种信息冗余的方法与硬件冗余相比不需要添加冗余模块和表决器。

受噪声影响的逻辑门电路可以看作是一个以 ε

① 国家自然科学基金(61401205)和江苏高校“青蓝工程”资助项目。
② 男,1991 年生,硕士生;研究方向:高可靠性的数字设计方法;E-mail: 365553158@qq.com
③ 通信作者 E-mail: tian.ban@njust.edu.cn
(收稿日期:2018-07-01)

概率进行翻转的二进制对称信道(binary symmetric channel,BSC),概率门电路的输出以 ε 概率进行翻转。本文将大量概率门器件组成的概率模块看作是 BSC,在 BSC 两端分别加入 LDPC 编码器和译码器。编码器的作用是增加冗余信息,译码器的作用是纠正错误信息。该概率模型中噪声服从(0 ~ 1)二项分布,其中 0 表示信息位以 $1 - \varepsilon$ 的概率不发生错误,1 表示信息位以 ε 的概率发生错误。利用马尔可夫随机场(Markov random field, MRF)以及吉布斯函数(Gibbs function)可以算出输出概率,利用信息熵来表示模块的可靠性相对大小^[9]。若模块可靠性记为 R ,那么模块输出发生错误的概率为 $\varepsilon = 1 - R$ 。

Gallager^[10]于 1962 年在博士论文中首次提出 LDPC,并发现 LDPC 具有逼近香农极限的良好性能。LDPC 译码算法有软判决译码算法和硬判决译码算法。软判决译码方面,Mackay 等^[11]于 1996 年重新发现 LDPC,并提出了概率测度的后验概率置信传播(belief propagation,BP)译码算法。随后 Fossorier 等^[12]相继提出 APP-Based 算法和 BP-Based 算法,以及一些为方便硬件实现的改进算法^[13]。硬判决译码方面,Gallager 提出比特翻转(bit flipping,BF)译码算法之后,Kou 等^[14]提出一种加权比特翻转(weighted bit flipping,WBF)算法。Zhang 等^[15]提出了改进的加权比特翻转(modified WBF,MWBF)算法。采用软判决和积算法,进行系统可靠性分析时,硬件实现复杂度较高、资源开销较大^[16],本文采用 WBF 算法进行 LDPC 译码。

1 信息冗余

对逻辑门电路进行可靠性分析时,可以通过概率门模型(probabilistic gate model, PGM)来分析其可靠性。以两输入的逻辑门电路为例,该门电路出错的过程可以用 PGM 表示。PGM 如图 1 所示,其中 A 和 B 为门电路的输入, $\theta = g(A,B)$ 为逻辑表达式, θ 为门电路输出的正确值, θ 在错误概率 ε 的影响下,输出不可靠的值 C 。

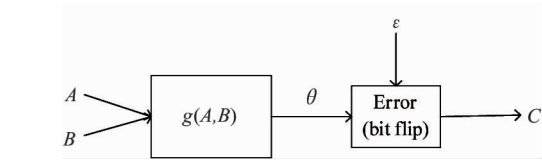


图 1 概率门模型

根据 PGM,本文推出由逻辑门所组成的逻辑单元的概率单元模型如图 2 所示。 $X(x \in X)$ 为概率单元模型的输入, $g(x)$ 为概率单元模型理想布尔函数, θ 为概率单元模型输出的理想值, ε 为模型出错概率, $Y(y \in Y)$ 为概率单元模型的输出。

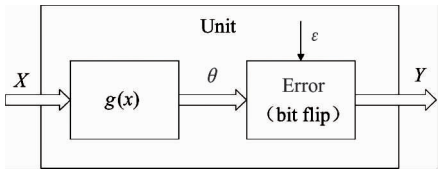


图 2 概率单元模型

为了提高数字电路可靠性,需要使用容错技术。下面介绍硬件冗余中的 TMR 来提高数字电路容错能力。TMR 容错模型如图 3 所示,将 X 分别输入 3 个相同的概率单元模型,相应地,3 个概率单元模型对应 3 个输出 $Y1$ 、 $Y2$ 、 $Y3$ 。 $Y1$ 、 $Y2$ 、 $Y3$ 按位通过 3 输入的多数表决器,输出纠正后的消息 Z 。五模冗余容错模型以及更高级别的冗余容错模型只需要添加额外的概率单元模型。

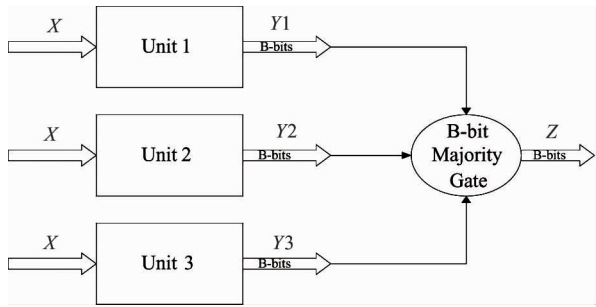


图 3 TMR 容错结构模型

将概率单元模型看作是一个数字通信系统中的通信信道,数字通信系统框图如图 4 所示。信源经过编码器编码以及调制器调制之后,进入传输信道,最后再由解调器解调以及译码器译码得到信宿。由

于信道中噪声源的影响,信道中传输的信息会发生错误,但编译码器可以一定程度上纠正信道传输产生的错误。

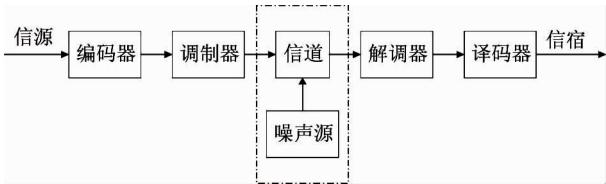


图 4 数字通信系统框图

概率单元模型中的错误比特翻转模型,如图 5 所示。该模型相当于数字通信系统中的信道,其出错概率 ε 相当于数字通信系统中的信道噪声, θ 相当于数字通信系统中的信道输入, Y 相当于数字通信系统中的信道输出。

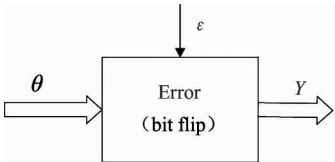


图 5 错误比特翻转模型

于是,在概率单元模型两端加入编码器及译码器,组成信息冗余容错结构模型如图 6 所示。该信息冗余容错结构模型输入为 in ,将 in 通过编码器进行编码产生码字 $[in\ k]$;然后通过逻辑功能函数 $h(x)$ 映射之后通过传输信道,即错误比特翻转模型;再将 $[in_e\ k_e]$ 通过逻辑功能函数 $h(x)$ 映射之后经过译码器译码,最后得到输出 out 。特别说明, $h(x)$ 映射关系对应了通信系统中的调制解调过程。本文选择码率为 $1/2$ 、码长为 816 的 LDPC 码进行信道编译码。

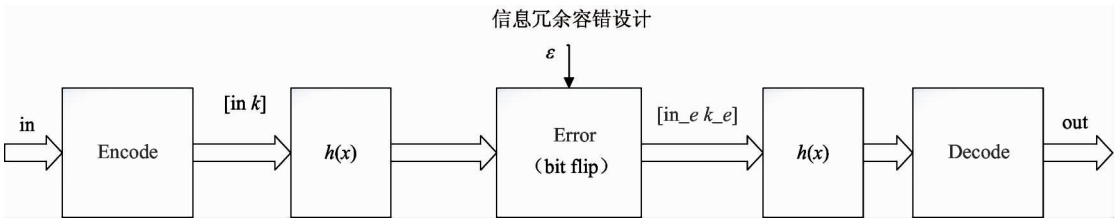


图 6 信息冗余容错结构模型

2 LDPC 码

在通信系统中,可以将信道编码分为分组码和卷积码两种,这两种码字在通信系统中都有着广泛的应用。本文在信息序列(长度为 k)后面,以一定的规则增加校验码元(长度为 $n - k$),组成码字(长度为 n)。如果校验码元的产生只与本组的信息位有关,与其他组的信息位无关,且译码时也仅仅利用本组的码元,则这种码被称为分组码,用 (n, k) 表示分组码集合。如果校验码元的产生不仅与本组信息位有关,还与之前输入编码器的信息位有关,并且

译码时同样也要利用前面码组的相关信息,这种码被称为卷积码,常用 (n, k, m) 来表示卷积码的集合, m 为约束长度。LDPC 就是线性分组码的一种。

$GF(2)$ 域上的 LDPC 码字 C 是一种 (n, k) 线性分组码,其校验矩阵是一种稀疏矩阵,即矩阵中非 0 元素的个数远小于 0 元素的个数。正是由于校验矩阵为低密度校验矩阵,所以构造出来的 LDPC 码,具有高性能、低复杂度等优点。

LDPC 码字 C 码长为 n , 信息序列长 k 。其校验矩阵 H 的维数是 $m \cdot n (m = n - k)$, H 的每一行对应一个校验方程,每一列对应码字中的一位。校验矩阵 H 中每一行非 0 元素的和称为该行的行重,每

—列非 0 元素的和称为该列的列重。式(1)是一个(8,4)线性分组码的校验矩阵对应的校验方程:

$$H = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}$$
$$\rightarrow \begin{cases} c_1 + c_3 + c_6 + c_7 = 0 \\ c_2 + c_3 + c_5 + c_8 = 0 \\ c_2 + c_4 + c_6 + c_8 = 0 \\ c_1 + c_4 + c_5 + c_7 = 0 \end{cases} \quad (1)$$

其码字 $\vec{c} = (c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8) \in C$, 满足式(2)。

$$H \cdot \vec{c}^T = 0 \quad (2)$$

在通信系统中,消息序列 $\vec{u} = (u_1, u_2, \dots, u_k)$ 根据生成矩阵 G ,通过式(3)进行信道编码,编码后的码字为 $\vec{c} = (c_1, c_2, \dots, c_n)$, $\vec{c}$ 经过信道调制后被映射为适合在信道中传输的传输序列 $\vec{x} = (x_1, x_2, \dots, x_n)$,传输序列 $\vec{x}$ 通过信道传输后,在接收端接收到接收序列为 $\vec{y} = (y_1, y_2, \dots, y_n)$ 的序列。根据 $\vec{y}$,通过译码算法进行译码得到序列 $\hat{c}$ 。

$$\vec{c} = \vec{u} \cdot G = \begin{bmatrix} u_1 & u_2 & \dots & u_k \end{bmatrix} \cdot \begin{bmatrix} g_{11} & g_{12} & \dots & g_{1n} \\ g_{21} & g_{22} & \dots & g_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ g_{k1} & g_{k2} & \dots & g_{kn} \end{bmatrix} \quad (3)$$

LDPC 译码方法大致可以分为两大类:一类是硬判决的译码,另一类则是基于软判决的译码。硬判决译码的运算量较小,硬件实现复杂度较低,较为实用。而软判决译码采用了后验概率信息,并通过迭代运算,使得 LDPC 的性能得以逼近香农限。近年来出现了各种软判决结果与硬判决算法相结合的译码方法,在保持较低硬件实现复杂度的情况下使译码性能进一步提高,从而进一步推动了 LDPC 的实用化。本文将采用加权比特翻转(WBF)算法,并将其用于硬件实现。

在译码时,如果检测到式(2)不成立,就会有相应的校验方程不满足条件,即在错误图样 $\vec{s} = (s_1, s_2, \dots, s_j)$ 中的某些位 s_j 等于1。因为校验矩阵为稀疏矩阵,校验矩阵中大部分元素为0元素,已知0元

素与任何数相乘为0,所以校验矩阵中非0元素与其所对应的码字一起决定了式(2)是否成立。如不成立,即校验矩阵中非0元素位置对应的码字位中某些位等于1。比特翻转算法正是利用了接收序列某些比特翻转与相应的校验失败数目的变化进行译码。

译码器利用式(4)计算所有的校验和,如果与某些比特相关的失败的校验方程数目超过一定的阈值,则将该比特翻转。利用比特翻转后产生的新序列重新计算所有校验和,直到所有校验方程满足条件或者达到最大迭代次数。关于阈值的选择,要遵循两个条件:一是要是纠错性能最优化,二是要使译码过程中计算校验和的次数最小化。

$$s_j = \vec{z} \cdot h_j = \sum_{l=0}^{n-1} z_l h_{j,l} \quad (4)$$

WBF 算法实现步骤如下。

步骤1 计算错误图样,如果错误图样 $\vec{s}$ 为0,说明接收到的码字满足所有校验方程,结束译码。

步骤2 遍历码字中每个比特,统计该比特对应的不满足校验方程的个数,记作 $f_i, i = 0, 1, \dots, n - 1$ 。

步骤3 定义集合A,查找 f_i 中最大值对应的比特并将其存入该集合。

步骤4 翻转集合中的比特。

步骤5 重复步骤1~4直到所有校验方程满足或达到最大迭代次数。

3 仿真

假设在数字电路中逻辑门所组成逻辑模块的出错概率 ε 在 $10^{-4} \sim 10^{-1}$ 范围内^[17],进行仿真。信息冗余容错设计仿真算法如下。

算法1 对于输入长度为 N 的二进制序列 $Msg_{1 \times N}$,通过LDPC生成矩阵 G 对 $Msg_{1 \times N}$ 进行编码得到长度为 $2N$ 的二进制序列 $Encoder_{1 \times 2N}$;概率门电路的运算关系可以抽象为 $Y = H(X)$,概率门电路的出错概率为 ε ,产生长度为 $2N$ 的二进制错误序列 $Err_{1 \times 2N}$ (0表示无错误发生,1表示有错误发生);概率门电路对输入 $Encoder_{1 \times 2N}$ 的影响可以用表达式 $Y1 = H(Encoder_{1 \times 2N})$ 及 $Y2 = Y1 \wedge Err_{1 \times 2N}$ 表示;最

后 $Y2$ 通过译码器译码得 $Out_{1 \times 2N}$, 计算 $Out_{1 \times 2N}$ 前 N 位与 $Msg_{1 \times N}$ 对应位置上值不一致的个数, 记为 Sum , 则 $BER = Sum/N$ 。

证明 本文采用 $1/2$ 码率的规则 LDPC 码, 其生成矩阵 $G_{N \times 2N}$ 的形式为 $[E_{N \times N}; F_{N \times N}]$, 其中 $E_{N \times N}$ 为单位矩阵, 即 $Encoder_{1 \times 2N} = [Msg_{1 \times N}; C_{1 \times N}]$; 表达式 $Y1 = H(Encoder_{1 \times 2N})$ 对二进制矩阵 $Encoder_{1 \times 2N}$ 每

一位同时操作, $H \cdot Encoer_{1 \times 2N} = 0$ 恒成立。对 $Y1$ 加入随机噪声得到 $Y2$, $Y2$ 通过译码得到 $Out_{1 \times 2N}$, $Out_{1 \times 2N}$ 前 N 位即为纠错后的概率门电路输出。

信息冗余设计仿真算法伪代码如图 7 所示, 输入参数 N 、packnum 以及错误概率序列 ε , 输出 BER。

Algorithm1: pseudo-code for simulation of information redundancy design
Input: a list of ε , packnum, N
Output: A set of error: BER
1: Begin
2: Initialize: msg=randi([0,1],1,N)
3: Encode: encode=msg*G
4: for $i=1:\text{length}(\varepsilon)$
5: for $j=1:\text{packnum}$
6: encode= $h(\text{encode})$
7: noise=binornd(1, $\varepsilon(i)$, 1, 2N)
8: rec=noise^encode
9: Decode: msg_out=WBF(rec)
10: Pe=sum(abs(msg_out-encode))
11: end
12: BER(i)=pe/packnum/N
13: end
14: End

图 7 信息冗余仿真设计伪代码

实例 随机产生长度为 408 的二进制随机序列 $Msg_{1 \times 408}$, 通过 LDPC 生成矩阵 G 对 $Msg_{1 \times 408}$ 进行编码得到长度为 816 的二进制序列 $Encoder_{1 \times 816}$; 概率门电路选择由与非门组成的非门(与非门其中一端输入置 1, 另一端连接 $Encoder_{1 \times 816}$ 序列的每一位), 产生长度为 816 的二进制错误序列 $Err_{1 \times 816}$ (0 表示无错误发生, 1 表示有错误发生); 对 $Encoder_{1 \times 816}$ 每一位取反 $Y1 = H(Encoder_{1 \times 816})$ 并加入随机噪声 $Y2 = Y1 \wedge Err_{1 \times 816}$; 最后, $Y2$ 通过译码器译码得 $Out_{1 \times 816}$, 计算 $Out_{1 \times 816}$ 前 408 位与 $Msg_{1 \times 408}$ 对应位置上值不一致的个数, 记为 Sum , $BER = Sum/N$ 。仿真程序框图如图 8 所示。

在相同出错概率 ε 的情况下, 4 000 包数据及 6 000 包数据时 TMR、FMR 以及信息冗余容错结构

的比特误码率 (bit error rate, BER) 曲线如图 9 及图 10 所示。随着逻辑模块可靠性的提高, 采用信息冗余容错结构的 BER 逐渐降低。并且当逻辑模块出错概率 $\varepsilon < 0.5$, 即逻辑模块可靠性 $R = (1 - \varepsilon) > 0.95$ 时, 采用 WBF 算法的信息冗余容错结构的纠错性能优于 TMR 容错结构、FMR 容错结构。

4 硬判决译码硬件实现

信息冗余容错结构设计中的关键部分为译码模块, 该模块采用 WBF 译码算法。其在 Vivado 2015.4 开发工具 RTL Analysis -> Schematic 中原理图如图 11 所示。为方便举例说明, 硬件实现部分采用码长为 96 的 LDPC 码为例。其余码长的实现方法与此类似, 只需改变相对应的位宽即可。

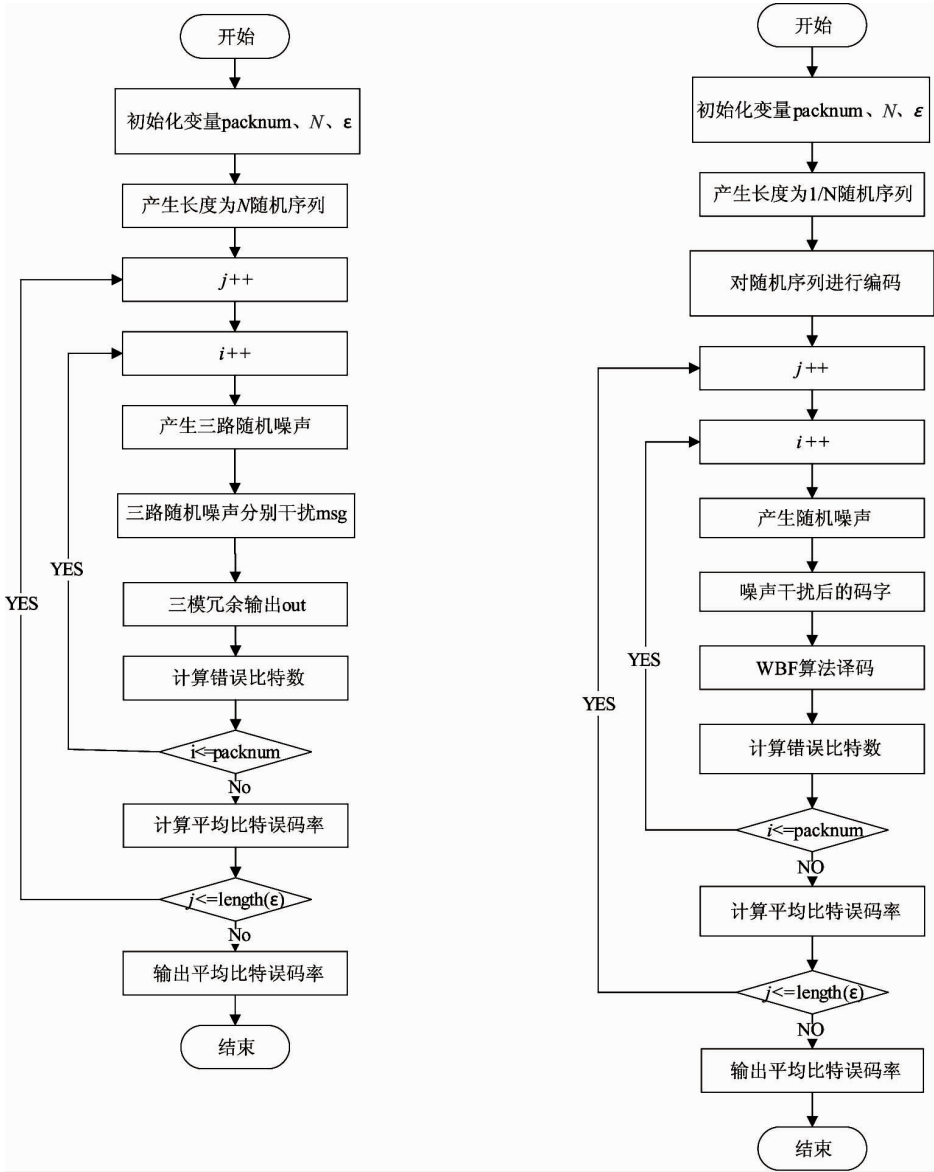


图 8 三模冗余、信息冗余容错结构仿真程序流程图

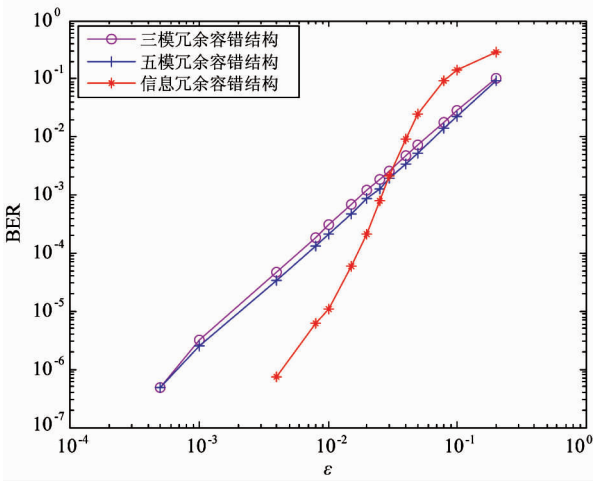


图 9 4000 包数据 TMR、FMR、信息冗余容错结构 BER 曲线

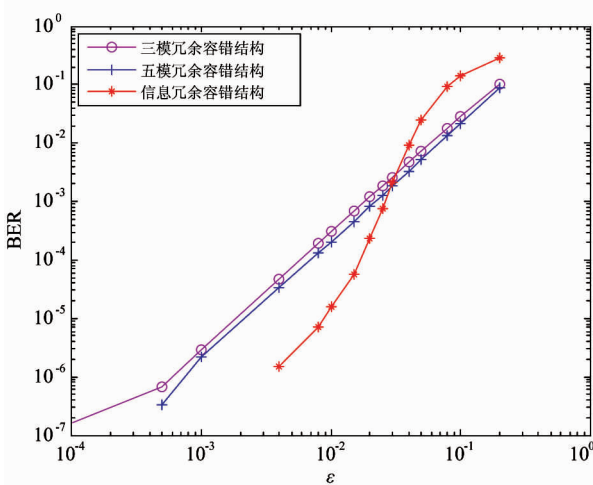


图 10 6000 包数据 TMR、FMR、信息冗余容错结构 BER 曲线

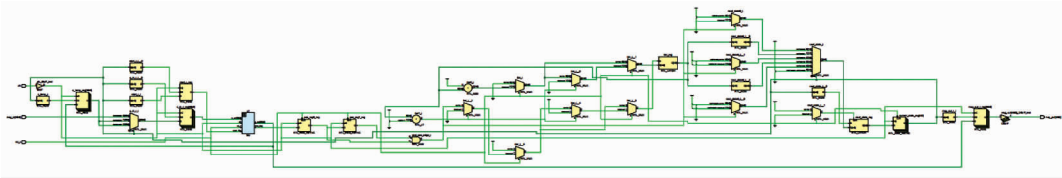


图 11 译码模块原理图

译码模块的局部原理图如图 12 所示。外部输入分为时钟 clk , 复位 rst_n 和消息输入 $msg_in[95:0]$ 。 s 为状态机状态, $s = 00000$ 和 $s = 00010$ 分别控制 $u1$ 模块输入来自外部输入还是迭代后的更新数据, $s = 01000$ 为状态结束标志。

其中, $u1$ 模块为译码核心运算模块, 其结构框

图如图 13 所示。其中包括错误图样计算、错误统计、查找最大值、比特位翻转等功能。

经过功能仿真, 整个译码模块如图 14 所示, 经过 3 次迭代后达到阈值 $msg_out[95:0]$ 输出译码消息。

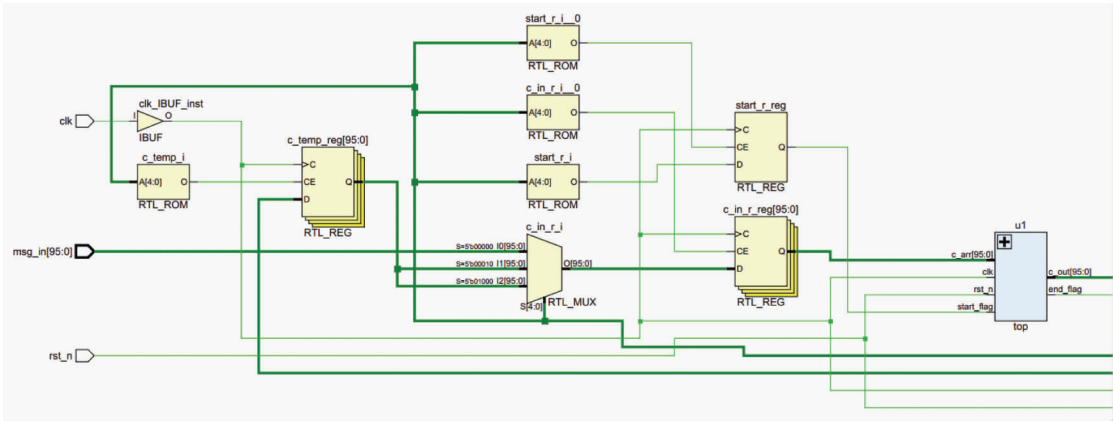


图 12 译码模块局部原理图

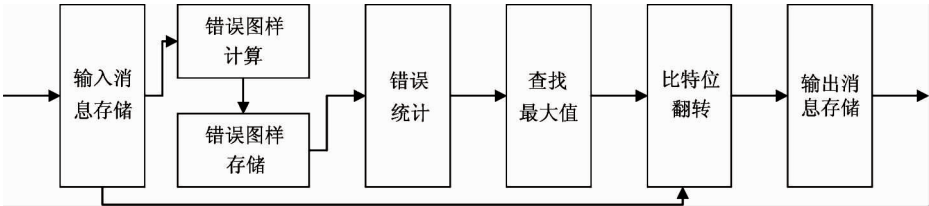


图 13 u1 模块结构框图

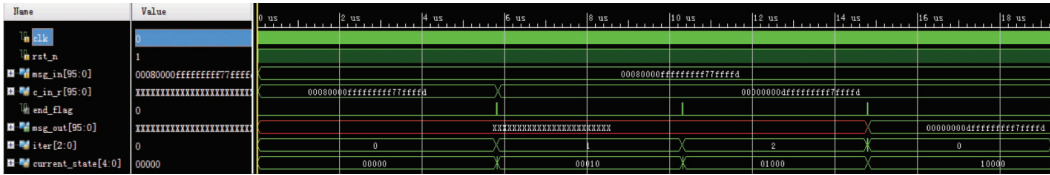


图 14 译码模块功能仿真图

$u1$ 模块功能仿真图如图 15 所示, t_in 为输入错误图样, 遍历 t_in 数组将数组值为 1 的索引值赋值给 h_addr_out 。如图中 h_addr_out 依次为 0,

3, 5, 13, $\dots$, 45 表示对应行的校验方程, 同时 h_val_in 将输出该校验方程对应的消息比特位, 即 $num0 \sim num5$, 并对相应位进行加 1 操作。之后统计信息

位对应的错误校验方程个数,选择错误个数最多的信息位进行比特翻转。

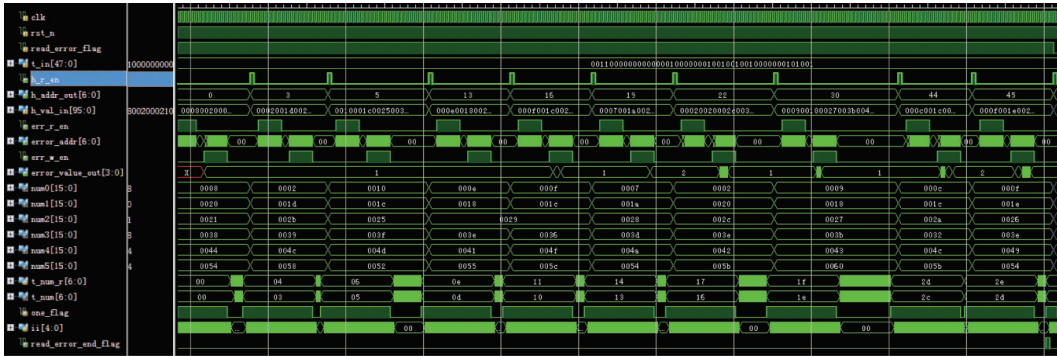


图 15 u1 模块功能仿真图

5 结 论

当数字电路中逻辑门组成模块的出错概率大于 0.05 时,硬件冗余容错结构比信息冗余容错结构 BER 低;反之,信息冗余容错结构 BER 低于硬件冗余容错结构。由此可见,当数字电路中逻辑模块出错概率低于 0.05 时,基于 LDPC 信道编码的信息冗余容错结构比硬件冗余的容错结构更有利于提高数字电路容错能力。特别地,采用信息冗余的容错结构对资源的消耗仅来自于编码器和译码器,并且不会随着数字电路的复杂度增加而增加。当数字电路的资源开销远高于编码器和译码器开销时,采用信息冗余容错结构会比硬件冗余容错结构节省资源。下一步的工作是两种容错设计方法的硬件实现,并给出资源消耗的具体比较。

参考文献

[1] 汪俊,师谦,邓文基. 浅析 SEU 原理研究及其应用 [C]. 见:第十二届全国可靠性物理学术讨论会,中国,都江堰,2007.51-55

[2] Raine M, Gaillardin M, Lagutere T, et al. Estimation of the single-event upset sensitivity of advanced SOI SRAMs [J]. *IEEE Transactions on Nuclear Science*, 2018, 65 (1):339-345

[3] Von Neumann J. Probabilistic logic and the synthesis of reliable organisms from unreliable components[J]. *Automata Studies*, 1956:43-98

[4] Sreeba S Z, Jose J, Mini M G. Energy-efficient fault tol-

erant technique for deflection routers in two-dimensional mesh Network-on-Chips [J]. *Iet Computers & Digital Techniques*, 2018, 12(3):69-79

[5] Ban T, Naviner L. Progressive module redundancy for fault-tolerant designs in Nano electronics[J]. *Microelectronics Reliability*, 2011, 51(9-11):1489-1492

[6] Martorell F, Cotozana S D, Rubio A. Fault tolerant structures for nanoscale gates[C]. In: *Proceedings of the 7th IEEE Conference on Nanotechnology*, Hong Kong, China, 2007.605-610

[7] Soltani H, Dolatshahi M, Sadeghi M. Comparing the reliability in systems with triple and five modular redundancy [C]. In: *Proceedings of the International Conference on Computer Science and Network Technology*, Changchun, China, 2016.437-442

[8] 张程程,班恬. SRAM 型 FPGA 的可重构容错结构研究[J]. *电子测量技术*, 2016, 39(11):41-45

[9] Bhaduri D, Shukla S. NANOLAB-a tool for evaluating reliability of defect-tolerant nanoarchitectures [J]. *IEEE Transactions on Naneotechnology*, 2005,4(4):381-394

[10] Gallager R G. Low-density parity-check codes[J]. *IRE Transactions on Information Theory*, 1962, 8(1):21-28

[11] Mackay D J C, Neal R M. Near Shannon limit performance of low density parity check codes[J]. *Electronics Letters*, 1996, 32(18):1645-1646

[12] Fossorier M P C, Mihaljevic M, Imai H. Reduced complexity iterative decoding of low-density parity check codes based on belief propagation[J]. *IEEE Transactions on Communications*, 2002, 47(5):673-680

[13] Xue W, Ban T, Wang J X. A modified normalized min-

- sum algorithm for LDPC decoding using order statistics [J]. *International Journal of Satellite Communication and Networks*, 2017, 35:163-175
- [14] Zhang J, Fosserier M P C. A modified weighted bit-flipping decoding of low-density parity-check codes [J]. *IEEE Communications Letters*, 2004, 8(3):165-167
- [15] Guo F, Hanzo L. Reliability-ratio based weighted bit-flipping decoding for low-density parity check codes[J]. *Electronics Letters*, 2004, 40(21):1356-1358
- [16] Winstead C, Howard S. A probabilistic LDPC-coded fault compensation technique for reliable nanoscale computing [J]. *IEEE Transactions on Circuits & Systems II Express Briefs*, 2015, 56(6):484-488
- [17] Tang Y, Winstead C, Boutillon E, et al. An LDPC decoding method for fault-tolerant digital logic[C]. In: *Proceedings of the IEEE International Symposium on Circuits and Systems*, Seoul, Korea, 2012. 3025-3028

Fault-tolerant design method based on LDPC decoding

Wang Dengchao, Ban Tian, Xue Wen

(School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing 210094)

Abstract

The miniaturization of semiconductor processes has rendered fault tolerance of digital circuits essential. The traditional method such as triple modular redundancy (TMR) results in excessive consumption of area and power. In this paper, a fault-tolerant design method based on information redundancy is proposed for logic circuits, which is implemented by low density parity check (LDPC) code. Since the probability of error in the probability gate model in digital circuits is similar to that of the binary symmetric channel (BSC), the channel coding method in the communication system is applied to fault-tolerant design of digital circuits. The proposed design adds an LDPC encoder and decoder at the input and output of logic gates. The LDPC in this paper takes 816 code length and 1/2 code rate as example, and uses weighted bit flip (WBF) algorithm to design the decoder. Within the range of 0.0001 to 0.1, the performance of hardware-redundant and information-redundant fault-tolerance is simulated. The bit error rate (BER) curve shows that when ε is higher than 0.05, the hardware redundancy is lower than the information redundancy BER; conversely, BER of the information redundancy is lower than that of the hardware redundancy.

Key words: digital circuit, reliability, low density party check, hardware redundancy, information redundancy