DOI:10.11918/201910089

二维非递归的低成本FIR滤波器设计方法

钟燕清[1,2],阎跃鹏[1,2],孟 真[1],田 易[1,2],刘 谋[1],李继秀[1]

(1. 中国科学院 微电子研究所,北京 100029; 2. 中国科学院大学,北京 100049)

摘 要: 为降低有限冲激响应(Finite impulse response, FIR)数字滤波器的成本,提升可综合性,提出了一种基于系数矩阵的二维非递归优化算法,并进行了仿真. 首先,对现有的数字滤波器优化算法进行了调研,比较了各优化算法的优势和不足;然后,对现有的一维非递归算法进行优化,提取一维非递归算法优化后的冗余项,得到了二维非递归优化算法,并分析了算法的复杂度;最后,生成多组滤波器分别对本算法与一维非递归算法,以及本算法和现有递归算法进行仿真和对比. 仿真结果表明:提出的二维非递归FIR滤波器设计方法充分利用了系数矩阵的冗余信息,保留了现有算法的最小逻辑深度特性,同时可以进一步节省中间加法器个数;相比于现有的一维非递归算法,本算法可节省10.05%(12 bit量化)和7.21%(16 bit量化)的加法器个数;在低阶滤波器的设计中,加法器使用量降低到了传统CSD表示法的30%左右,从逻辑深度和加法器个数两方面都超越了已发表的递归和非递归滤波器设计方法.

关键词: 低成本FIR滤波器;非递归;冗余项;系数矩阵;逻辑深度;加法器个数

中图分类号: TP510 **文献标志码:** A **文章编号:** 0367-6234(2021)06-0171-06

Two-dimensional non-recursive design method of low-cost FIR filter

ZHONG Yanqing[1,2], YAN Yuepeng[1,2], MENG Zhen[1], TIAN Yi[1,2], LIU Mou[1], LI Jixiu[1]

(1. Institute of Microelectronics, Chinese Academy of Sciences, Beijing 100029, China;
2. University of Chinese Academy of Sciences, Beijing 100049, China)

Abstract: To reduce the overall hardware cost as well as improve the synthesizability of finite impulse response (FIR) filter, a two-dimensional non-recursive optimization algorithm based on coefficient matrix was proposed and simulated. Firstly, the existing digital filter optimization algorithms were investigated, and the advantages and shortcomings of these algorithms were compared and analyzed. Then, efforts were made to optimize the existing one-dimensional non-recursive algorithm. By extracting the redundant terms of one-dimensional non-recursive optimization algorithm, a novel two-dimensional non-recursive FIR optimization method with low computing complexity was obtained. Finally, multiple groups of filters were generated to simulate and compare the performance between the proposed algorithm and one-dimensional non-recursive algorithm as well as the existing recursive algorithms. Simulation results show that the two-dimensional non-recursive FIR filter design method proposed in this paper makes full use of the redundant information of the coefficient matrix, keeps the character of minimum logic depth, and meanwhile reduces logic adder number. Compared with the existing one-dimensional non-recursive algorithm, this algorithm saved logic adder by 10.05% (12 bit quantization) and 7.21% (16 bit quantization). In the design of low-order filter, the adder cost was reduced to 30% of the conventional CSD representation, which outperforms the existing recursive and non-recursive filter design methods in terms of logic depth and adder number.

Keywords: low-cost FIR filter; non-recursive; redundant terms; coefficient matrix; logic depth; logic adder number

FIR数字滤波器为对称型结构,阶数和量化位宽是决定数字滤波器硬件资源的两个重要因素,通常情况下,阶数越高、量化位宽越大,硬件资源消耗越多. 考虑到FIR滤波器的参数是固定的,人们引入了固定乘数优化算法进行优化处理[1],即将乘法分解成加法和移位计算,降低运算复杂度. 在已有的数字滤波器的固定乘数优化算法中,加法深度LD(logic depth)和加法器个数LA(logic adder)是衡量算法优劣性的两个重要指标. 降低加法器个数需要尽可能复用系数中的公共项,从而带来加法深度的增加;降低加法深度则意味着降低公共项的复杂度,带来加法器LA的增加. LD和LA的结果不仅取决于系数的量化位宽、阶数,也取决于用户的优化方式,是一个综合性的优化问题. 考虑到常系数乘法的加法器个数与系数非零项直接相关,Park等[2-3]提出采用CSD、MSD表示法表示滤波器系数,在后续的算法中

收稿日期: 2019-10-15
作者简介: 钟燕清(1984—),女,博士研究生;
阎跃鹏(1963—),男,教授,博士生导师
通信作者: 阎跃鹏,yanyuepeng@ime.ac.cn

得到了广泛应用.在此基础上,人们提出了采用递归式算法和非递归式算法的不同公共项提取思路来降低电路中的加法器消耗.前者[4-9]以BHM[4]、RAG-n[5-6]、HARTLEY[7]和HCUB[8]为代表,采用迭代运算穷举固定系数的所有公共项,可以达到最优的降低MCM加法器的效果.RAG-n、BHM和HCUB算法均采取图形启发式方式进行优化,即先对系数进行排序,以最小的系数为公因子,穷举各个系数的公因子组合方式,选取代价最小的作为最优解.其中Bull等[4]最早提出采用图形化的方法来从小到大组合滤波器系数,取得了较好的效果;Dempster等[5]在此基础上进行了改良,扩展了系数范围,提出了BHM算法;而RAG-n则更进一步,采用查表法遍历了所有可能,所以加法器个数LO最少,但是代价是需要预存分解表,算法复杂度最高,运算时间最长,加法深度最高.HCUB算法则是在此基础上,引入了中间变量算子,降低了算法对分解表的要求,是目前最优的启发式优化算法.然而,上述图形启发式优化方法均存在算法复杂度高、求解时间随量化位数增加急剧上升的问题,同时由于采用公共项的多次复用,逻辑深度居高不下,引入大量的路径延时,对后期电路的设计和综合非常不友好.非递归式[10-16]则以直接提取公共项进行有限次数迭代,具有算法复杂度低、可综合性好等优点,代表性的有NR-SCSE、HSSE、VSSE算法等.其中Peiro等[14]提出的非递归有符号公共项消去算法(Non-recursive signed common sub-expression elimination, NRSCSE),采用一维搜索频次最高的公共项的算法,算法复杂度低,逻辑深度低,取得了非常好的同时降低LO和LD的效果.为了达到降低逻辑深度的目的,该算法只启用包含2个非零项的公共项,具有最低逻辑深度特性.然而,上述非递归式算法均只进行了一维的共同项(单独的系数矩阵的行公共项或者单独的列公共项)的提取,并没有考虑到系数矩阵的二维特性,在加法器个数的指标上整体落后于递归算法.

为了达到逻辑深度和加法器个数的双重优化效果,本文提出了一种新的二维非递归优化算法.其在一维非递归式算法的基础上,对系数矩阵的第2个维度进行公共项提取,从而达到进一步降低加法器个数的效果.仿真结果表明,相比于一维的NRSCSE,本文设计的滤波器优化算法(后文简称ONRSCSE)可以降低10%左右的系数加法器数量,具有良好的推广价值.

1 一维非递归算法

数字滤波器的滤波效果是通过输入信号与滤波系数的卷积运算来实现的,具体来说,输入信号x进入一个N阶的数字FIR滤波器后,需要与$N-1$个系数进行乘法运算,之后进行累加,得到输出值.假定N阶FIR滤波器的系数由低阶到高阶分别为:$h(0),h(1),\cdots,h(N-1)$,则有该滤波器的响应为

$$y(n) = \sum_{i=0}^{N-1} h(i) \times x(n-i) = \sum_{i=0}^{N-1} h(i) \times x(n) \times z^{-i}. \tag{1}$$

式中:$x(n)$、$y(n)$分别为滤波器的输入和输出.假定滤波器的系数矩阵为H,$H=[h_0,h_1,\cdots,h_{N-1}]$,当前输入为$x(n)$,则输出$\boldsymbol{Y}=\boldsymbol{H}\times\boldsymbol{X}$,$\boldsymbol{X}=x(n)\times[z^0,z^{-1},z^{-2},z^{-3}]$为输入矢量.

考虑到FIR滤波器的系数对称性,通常采用优化的转置式实现,假定FIR为I型滤波器,N为奇数,则实现的滤波器结构如图1所示.由于其系数对称性,即$h_0=h_{N-1}$,$h_1=h_{N-2}$,对系数的优化只需要对其前1/2系数$h_0,h_1,\cdots,h_{(N-1)/2}$进行即可.

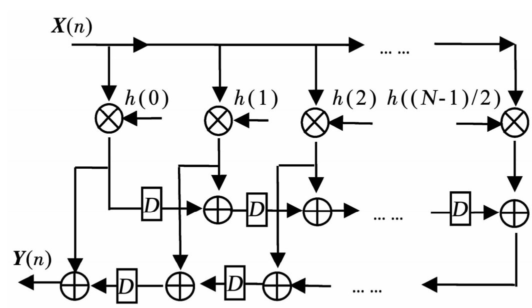

图1 FIR滤波器的优化转置结构

Fig.1 Optimized transposed structure of FIR filter

传统的非递归优化算法在将系数h_i采用CSD法表示后,对各个系数的非零项进行统计,依次提取出现频率最高的间距相同、符号相同/相反的非零项作为公共项,对系数进行分解,直到系数无法再提取公共项为止,通常为一维搜索算法.以NRSCSE[14]算法为例,其只搜索系数内的任意间距2个非零位公共项,在量化位数为b时,单个系数最多存在$2\times(b-1)$个公共项,逻辑深度最多为$\lceil \log_2 b/2 \rceil$.由于只需要搜索和消去运算,NRSCSE算法复杂度非常低,非常适合于普通滤波器的优化.然而,由于只用到了系数内的冗余项,在系数非零项为单数时必然会出现单个的1/-1不能复用,而导致$2n$个非零项和$2n-1$个非零项的优化效果一样,增加加法器的消耗.以4个系数为例说明此问题:

$$h(0)=1\,288, h(1)=776, h(2)=1\,077, h(3)=1\,189. \tag{2}$$

将其表示为12 bit的CSD数后,系数如下:

$\boldsymbol{h}(0)=0\text{x}508=(010100001000)_{\text{CSD}}$,

$$h(1)=0\text{x}308=(010\text{-}100001000)_{\mathrm{CSD}},$$
$$h(2)=0\text{x}435=(0100010\text{-}10101)_{\mathrm{CSD}},$$
$$h(3)=0\text{x}4\text{a}5=(010010100101)_{\mathrm{CSD}}.\qquad(3)$$

采用 NRSCSE 提取公共项 101/10 -1 后，得到剩余矩阵(后文称为残余矩阵)为

$$\boldsymbol{H}_{\mathrm{remain}}=\begin{bmatrix}1\ll 3\\1\ll 3\\1\ll 11\\1\ll 11\end{bmatrix},\qquad(4)$$

则整体表示法为

$$y_0=\boldsymbol{H}\times\boldsymbol{X}=\begin{bmatrix}S_0\ll 8+x_0\ll 3\\S_1\ll 8+x_0\ll 3\\S_1\ll 4+S_0+x_0\ll 10\\S_0\ll 5+S_0+x_0\ll 10\end{bmatrix}\times\begin{bmatrix}z^0\\z^{-1}\\z^{-2}\\z^{-3}\end{bmatrix}^{\mathrm{T}}.\qquad(5)$$

式中：公共项 $\boldsymbol{S}_0=\boldsymbol{x}_0\ll 2+\boldsymbol{x}_0$，$\boldsymbol{S}_1=\boldsymbol{x}_0\ll 2-\boldsymbol{x}_0$，如图 2 中红色方框内所示. 共需要加法器 8 个，逻辑深度(系数中最大加法器个数 +1)为 2 层.

$h(0)$	0	1	0	1	0	0	0	0	1	0	0	0
$h(1)$	0	1	0	-1	0	0	0	0	1	0	0	0
$h(2)$	0	1	0	0	0	1	0	-1	0	1	0	1
$h(3)$	0	1	0	0	1	0	1	0	0	1	0	1

图 2　滤波器系数的 NRSCSE 优化示例

Fig. 2　Example of NRSCSE coefficient optimization

考虑到经过系数内的公共项提取后，系数残余矩阵内还存在冗余信息，如图 2 中蓝色圆框所示，可以进一步进行优化.

2　二维非递归优化算法

二维非递归优化算法(Optimized non-recursive signed common sub-expression elimination，ONRSCSE)是行公共项和列公共项提取的结合，是一种典型的非递归固定乘数优化方法. 其大致思路为：首先采用 CSD 法表示 FIR 滤波器的前 1/2 系数，将系数表示为{ -1,0,1}的初始系数矩阵；然后采用行公共项提取的方法依次分解系数矩阵的行系数，直到满足行分解的终止条件为止；之后对分解后剩余的非 0 系数矩阵(后文称为残余矩阵)进行列公共项分解，直到满足终止条件为止；最后将系数表示成分解出的行与列公共项的移位与加法组合，完成整个算法优化.

以一维优化为例，本文在一维非递归算法的基础上，对上述残余矩阵式 6 进行列公共项的提取，得到列公共项[1 1]$^{\mathrm{T}}$，记为 $S_2=x_0+x_0[-1]$，[-1]表示对系数进行一个单位的延时，则滤波器此部分的输出为

$$y_0=\begin{bmatrix}S_0\ll 8\\S_1\ll 8\\S_1\ll 4+x_0\ll 10\\S_0\ll 5+S_0\end{bmatrix}+\begin{bmatrix}0\\S_2\ll 4\\0\\S_2\ll 10\end{bmatrix}*\begin{bmatrix}z^0\\z^{-1}\\\cdots\\z^{-3}\end{bmatrix}^{\mathrm{T}}.\qquad(6)$$

整体实现一共需要加法器 7 个，逻辑深度 2 层. 因此，相比于一维的 NRSCSE 算法，本算法(ONRSCSE)增加了列系数的非零项搜索，节省了 1 个加法器资源.

不难推算，NRSCSE 算法优化的残余矩阵中非零项越多，列公共项出现次数越多，本算法的优化效果越显著.

2.1　名词定义

2.1.1　系数矩阵

由 CSD 表示后的滤波器系数组成的矩阵，矩阵元素为{ -1,0,1}，假定滤波器系数量化位宽为 b，阶数为 n，本算法通常取前 1/2 系数第 0 ~ m 阶组成系数矩阵(如果 n 为偶数，$m=n/2$；如果 n 为奇数则 $m=(n-1)/2$). 系数矩阵记为 $\boldsymbol{H}_i$，下标 i 代表第 i 次系数分解，$\boldsymbol{H}_0$为初始化系数矩阵.

2.1.2　残余矩阵

系数矩阵减去公共项元素后剩余的非 0 矩阵.

2.2　算法描述

在系数矩阵的基础上，对公共项进行二维搜索，遵循规则为——先迭代搜索行公共项，再搜索列公共项.

1)首先对 n 阶、系数量化为 b bit 的 FIR 滤波器进行 CSD 表示，产生 $n*b$ 的系数矩阵，取前 1/2 系数 0 ~ m 项 $C_{b/2}^2(m=(n-1)/2)$组成系数矩阵 $\boldsymbol{H}_0$；

2)对系数矩阵 $\boldsymbol{H}_i$进行行系数分解，方法为统计在整个系数矩阵中出现次数最多的行公共项，将其作为最优公共项；

3)在系数矩阵 $\boldsymbol{H}_i$中消去最优公共项，得到残余矩阵 $\boldsymbol{H}_{i+1}$，则 $\boldsymbol{H}_i$可以表示为 $\boldsymbol{H}_{i+1}$与最优公共项的移位和加法组合；

4)返回步骤 2)继续分解残余矩阵 $\boldsymbol{H}_{i+1}$，直到满足行分解的终止条件——公共项矩阵中最优公共项出现次数小于 2 为止，进入步骤 5). 假定此时已经进行了 j 次迭代，得到了残余矩阵 $\boldsymbol{H}_j$；

5）对残余系数矩阵 $\boldsymbol{H}_j$ 进行列分解，方法为统计在整个系数矩阵中出现次数最多的列公共项，将其作为最优公共项；

6）在系数矩阵 $\boldsymbol{H}_j$ 中消去最优公共项，得到残余矩阵 $\boldsymbol{H}_{j+1}$，则 $\boldsymbol{H}_j$ 可表示为 $\boldsymbol{H}_{j+1}$ 与最优列公共项的移位和加法组合；

7）返回步骤 5）继续分解残余矩阵 $\boldsymbol{H}_{j+1}$，直到满足列分解终止条件为止，进入步骤 8）；

8）将滤波器的输出 y 表示为所有行、列公共项的移位和累加 * 输入的组合.

2.3 算法复杂度分析

基于上述算法描述，对 ONRSCSE 的复杂度进行分析. 假定待优化滤波器的阶数为 n，量化位宽为 b，则系数矩阵的行系数最多有 $b/2$ 个非零项，每个系数的行公共项搜索最多进行 $C_{b/2}^2$ 次运算，一次系数矩阵的行公共项搜索最多进行 $n * C_{b/2}^2$ 次计算. 由于行公共项最多有 $(b-2)$ 个，故行分解的运算次数为 $n*(b-2)*C_{b/2}^2$ 次. 完成行分解后，残余矩阵已经是稀疏矩阵，只有少量的非零项，因此列分解运算不影响算法复杂度. 故 ONRSCSE 的整体算法复杂度与一维非递归算法 NRSCSE 相当，均为 $o(nb^3)$. 相比于现有递归算法 HCUB[8] 的 $o(n^3b^5)$，算法复杂度得到了极大的降低.

3 结果及分析

3.1 实验数据集

本文采用通用的滤波器设计方法，一共设计了 19 个不同特性的滤波器，并对其进行不同量化位宽、不同阶数的例化，产生了共 82 个滤波器，此为算法优化的对象.

其中为对比本算法与一维非递归算法，采用 MATLAB 的 FDAtool 工具产生了 12 组数字 FIR 带通滤波器，阶数从 37 阶到 649 阶，并对每组滤波器系数进行了 12、16 bit 的量化，覆盖了常用的数字滤波器范围，此为数据集 1.

为对比本算法与递归型优化算法，采用 Parks-McClellan 法设计 7 种不同阶数的数字带通滤波器，每种阶数内有 10 组不同的滤波器系数，分别对应不同的带通特性，此为数据集 2. 采用 Voronenko 等[18] 提供的优化算法库对数据集 2 内的滤波器进行优化. Voronenko[18] 算法库为卡内基梅隆大学 SPIRAL 算法库的一部分，滤波器优化部分由 HCUB 算法的提出者 Voronenko[18] 提供，是一个集成多个主流递归式滤波器优化算法的经典算法实现库，已经被 Vinod 等[16-17] 多位学者引用，可以实现包括 BHM、RAG-n、HCUB 在内的 3 种递归式滤波器优化算法，量化位宽支持到 12 bit，滤波器阶数支持到 100 阶左右. 由于 Parks-McClellan 设计法和算法库的限制，只产生了 70 组 30 – 100 阶的滤波器，量化位宽均为 12 bit.

3.2 本算法与一维非递归算法及传统 CSD 表示法的对比

对数据集 1 内的 18 组滤波器进行了优化，统计采用 CSD 表示法、NRSCSE 算法和 ONRSCSE 算法时运算需要的加法器资源，得到结果见表 1、2.

表 1 CSD 表示法的加法器资源

Tab. 1 Logic adder number of CSD representation

阶数	12 bit	16 bit
FIR1（37 阶）	48	70
FIR2（74 阶）	87	143
FIR3（131 阶）	124	206
FIR4（163 阶）	164	293
FIR5（170 阶）	199	289
FIR6（254 阶）	216	374
FIR7（325 阶）	226	500
FIR8（401 阶）	287	446
FIR9（501 阶）	296	448
FIR10（507 阶）	282	717
FIR11（601 阶）	302	471
FIR12（649 阶）	286	868

表 2 NRSCSE 与 ONRSCSE 算法优化后的加法器资源

Tab. 2 Logic adder number after NRSCSE and ONRSCSE optimization

阶数	NRSCSE		ONRSCSE	
	12 bit	16 bit	12 bit	16 bit
FIR1（37 阶）	20	31	18	29
FIR2（74 阶）	36	66	32	62
FIR3（131 阶）	47	86	43	81
FIR4（163 阶）	60	130	55	121
FIR5（170 阶）	78	131	70	116
FIR6（254 阶）	83	164	75	149
FIR7（325 阶）	88	222	77	208
FIR8（401 阶）	114	200	101	185
FIR9（501 阶）	119	200	109	190
FIR10（507 阶）	109	310	97	284
FIR11（601 阶）	122	207	113	195
FIR12（649 阶）	108	374	95	348

可以看到，与 CSD 表示法相比，NRSCSE 与 ONRSCSE 优化效果都很显著. 在阶数低于 100 时，优化后的加法器资源均控制在传统 CSD 表示法的 30% 左右；阶数高于 100 时，优化后的加法器资源控制在传统 CSD 表示法的 50% 以内.

比较 NRSCSE 与 ONRSCSE 的优化效果发现，同等情况下，ONRSCSE 均优于 NRSCSE，优化效率见表 3. 可以看到，ONRSCSE 对 NRSCSE 的优化效率不随着滤波器阶数和量化位宽线性变化，而是主要取决于滤波器经过一维非递归优化后的加法器资

源. 普遍来说，同样量化位宽的条件下，ONRSCSE 对 NRSCSE 的优化效果呈现一定的比例关系，NRSCSE 优化后的加法器个数越多，ONRSCSE 节省的加法器个数就相对更大. 然而由于经过 NRSCSE 优化后的残余系数矩阵的非零位在不同滤波器系数矩阵中分布各异，所以优化效果会体现出个体差异，如 FIR10 与 FIR11，12 bit 量化时经过 NRSCSE 优化后加法器个数分别为 109/122 个，但是 ONRSCSE 的节省加法器个数却是 12/9 个，FIR11 的 ONRSCSE 优化效果略差于 FIR10. 这是因为 FIR11 经过行系数公共项提取后，残余稀疏矩阵非零位分布相对发散，可以提取的列公共项个数少于 FIR10.

表 3　ONRSCSE 对 NRSCSE 的提升效果

Tab. 3　Optimization effect of ONRSCSE versus NRSCSE

阶数	节省加法器个数		优化率/%	
	12 bit	16 bit	12 bit	16 bit
FIR1（37 阶）	2	2	10.00	6.45
FIR2（74 阶）	4	4	11.11	6.06
FIR3（131 阶）	4	5	8.51	5.81
FIR4（163 阶）	5	9	8.33	6.92
FIR5（170 阶）	8	15	10.26	11.45
FIR6（254 阶）	8	15	9.64	9.15
FIR7（325 阶）	11	14	12.50	6.31
FIR8（401 阶）	13	15	11.40	7.50
FIR9（501 阶）	10	10	8.40	5.00
FIR10（507 阶）	12	26	11.01	8.39
FIR11（601 阶）	9	12	7.38	5.80
FIR12（649 阶）	13	26	12.04	6.95
平均优化比例	/	/	10.05	7.21

对上述优化结果的加法器资源进行绘图，得到 ONRSCSE 与 NRSCSE 加法器个数对比，如图 3 所示.

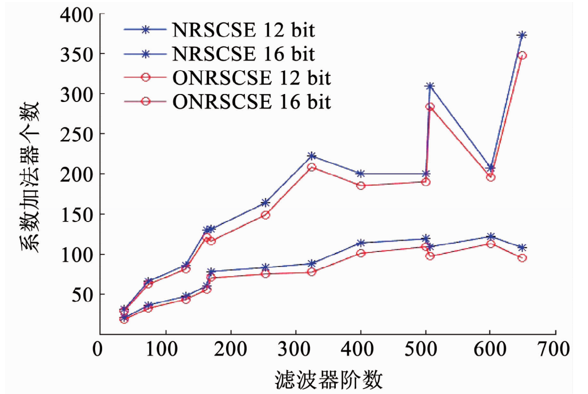

图 3　ONRSCSE 与 NRSCSE 加法器个数对比

Fig. 3　Comparison of adder number between ONRSCSE and NRSCSE

与传统一维非递归算法 NRSCSE 算法相比，采用 ONRSCSE 算法的优化效果更高，降低加法器资源的比率平均为 10.05%（12 bit 量化）和 7.21%（16 bit 量化）. 同时，优化的效果受阶数的变化影响较小，呈现非常稳定的特性. 由于 ONRSCSE 对相邻系数的共同项提取结果直接通过 D 触发器进入下一级运算，对滤波器的逻辑深度没有影响，因此 ONRSCSE 和一维非递归算法一样，仍然具有最低逻辑深度特性.

3.3　本算法与传统算法的比较

对数据集 2 的 7 种 12 bit 量化的滤波器进行优化，分别采用 BHM、RAG-n、HCUB、NRSCSE、ONRSCSE 算法进行设计，每种滤波器的 10 组系数优化结果 LA 与 LD 进行平均，得到相同阶数内滤波器优化的平均优化结果. 记录最终得到的平均加法器个数（Mean logic adder，MLA）和平均逻辑深度（Mean logic depth，MLD），进行优化效果对比. 其中 BHM、RAG-n、HCUB 的滤波器优化算法由 Voronenko[18] 提供，优化结果对比见表 4.

表 4　ONRSCSE 与 NRSCSE、BHM、RAG-n、HCUB 的优化效果对比

Tab. 4　Comparison of optimization effect among ONRSCSE, NRSCSE, BHM, RAG-n, and HCUB

滤波器	NRSCSE		ONRSCSE		BHM		RAG-n		HCUB	
	MLA	MLD	MLA	MLD	MLA	MLD	MLA	MLD	MLA	MLD
FIR1（31 阶）	14.6	1.9	13.3	1.9	16.2	/	15.3	5.1	15.4	5.0
FIR2（42 阶）	18.8	1.9	16.3	1.9	19.3	/	18.3	4.3	18.3	4.4
FIR3（51 阶）	20.6	1.9	18.7	1.9	19.2	/	19.0	2.9	20.6	2.8
FIR4（61 阶）	20.6	1.7	18.1	1.7	21.1	/	21.1	2.7	19.8	2.5
FIR5（71 阶）	25.1	3.6	21.5	3.6	22.6	/	22.6	2.8	21.2	2.7
FIR6（81 阶）	29.1	2.0	25.2	2.0	29.9	/	29.5	3.0	29.5	3.2
FIR7（101 阶）	33.7	2.0	29.0	2.0	30.6	/	30.6	2.7	32.0	2.8

对优化结果的加法器资源和逻辑深度进行绘图,得到算法性能对比如图 4 所示。可以看到,在 31 ~ 101 阶的一共 70 组 12 bit 系数量化滤波器中,非递归的滤波器优化算法 NRSCSE 算法相比于递归式算法 HCUB、BHM、RAG-n,逻辑深度最低,加法器资源最高,优化后的 ONRSCSE 算法逻辑深度与 NRSCSE 算法相同,均保持为最低,但是加法器资源已经明显优于 HCUB、BHM、RAG-n,取得了最好的效果. 由于 ONRSCSE 只依靠系数内的公共项,不利用系数值进行下一步的系数优化,系数间不存在依赖关系,不会对后续的综合产生影响,非常有利于算法的综合与实现.

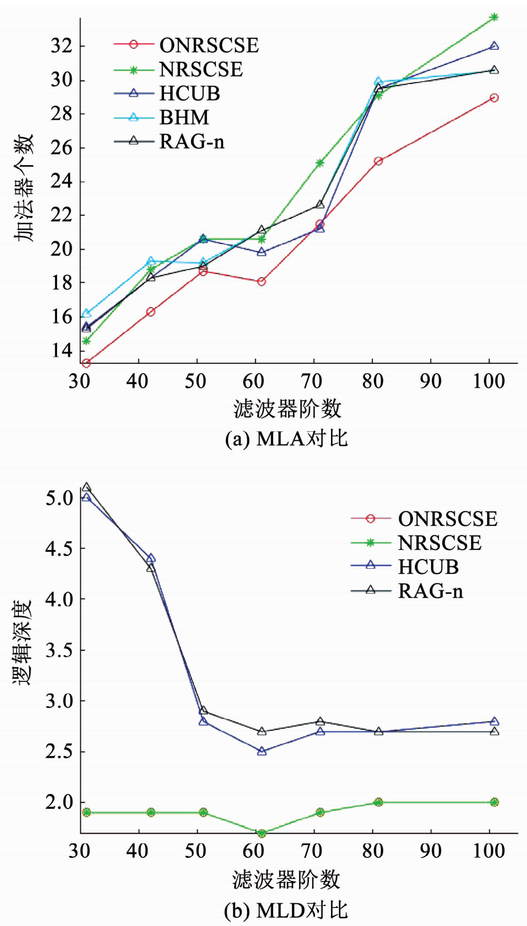

图 4　ONRSCSE/NRSCSE/HCUB/BHM/RAG-n 算法的加法器资源与逻辑深度对比

Fig. 4　Mean logic adder and mean logic depth of ONRSCSE/NRSCSE/HCUB/BHM/RAG-n

4　结　论

1)本文首次提出了结合系数行公共项优化和列公共项优化的系数矩阵二维优化方法,并将其运用到了非递归的 FIR 滤波器优化算法中,取得了较好的效果. 首先将系数用 CSD 法表示,降低了系数中的非 0 值个数;然后对系数矩阵进行行系数分解,降低行系数中的加法器个数;之后对残余矩阵进行公共项提取,进一步降低整体加法器个数.

2)相比于现有的一维非递归算法,本算法可节省 10.05% (12 bit 量化) 和 7.21% (16 bit 量化) 加法器个数. 在低阶滤波器的优化中,加法器使用量降低到了传统的 CSD 表示法的 30% 左右. 仿真结果表明,在阶数低于 100、量化位宽为 12 bit 时平均加法器个数和深度指标均优于已发表的 NRSCSE、BHM、RAG-n、HCUB 算法.

3)由于只采用 2 个非零项作为公共项,逻辑深度保持在$\log_2 \dfrac{b}{2}$(b 为系数的量化位宽),算法复杂度在 $o(nb^3)$(n 为滤波器阶数),是目前逻辑深度最优、算法复杂度最低、最有利于硬件综合的低成本 FIR 滤波器设计算法,具有良好的推广意义.

参考文献

[1] CHANDRA A, CHATTOPADHYAY S. Design of hardware efficient FIR filter: A review of the state-of-the-art approaches [J]. Engineering Science and Technology, an International Journal, 2016, 19(1): 212. DOI: 10.1016/j.jestch.2015.06.006

[2] PARK I C, KANG H J. Digital filter synthesis based on minimal signed digit representation [C]//Proceedings of the 38th Design Automation Conference. Las Vegas, NV: IEEE, 2001: 468. DOI: 10.1109/DAC.2001.156185

[3] HARTLEY R. Optimization of canonic signed digit multipliers for filter design[C]//Proceedings of the IEEE International Sympoisum on Circuits and Systems. Piscataway, NJ: IEEE, 1991: 1992. DOI: 10.1109/ISCAS.1991.176054

[4] BULL D R, HORROCKS D H. Primitive operator digital filters[J]. IEE Proceedings-Circuits, Devices and Systems, 1991, 138(3): 401. DOI: 10.1049/ip-g-2.1991.0066

[5] DEMPSTER A G, MACLEOD M D. Constant integer multiplication using minimum adders[J]. IEE Proceedings-Circuits, Devices and Systems, 1994, 141(5): 407. DOI: 10.1049/ip-cds:19941191

[6] DEMPSTER A G, MACLEOD M D. Use of minimum-adder multiplier blocks in FIR digital filters[J]. IEEE Transactions in Circuits and Systems II: Analog and Digital Signal Processing, 1995, 42(9): 569. DOI: 10.1109/82.466647

[7] HARTLEY R I. Subexpression sharing in filters using canonic signed digit multipliers[J]. IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, 1996, 43(10): 677. DOI: 10.1109/82.539000

[8] BOUDJELABA K, CHIKOUCHE D, ROS F. Evolutionary techniques for the synthesis of 2-D FIR filters [C]// Proceedings of the IEEE Statistical Signal Processing Workshop. Nice, France: IEEE, 2011: 601. DOI: 10.1109/SSP.2011.5967771

[9] VORONENKO Y, PUSCHEL M. Multiplierless multiple constant multiplication[J]. ACM Transactions on Algorithms, 2007, 3(2): 1. DOI: 10.1145/1240233.1240234

[10] PASKO R, SCHAUMONT P, DERUDDER V, et al. A new algorithm for elimination of common subexpressions [J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 1999, 18(1): 58. DOI: 10.1109/43.739059

(下转第 191 页)