

基于硬件可编程逻辑 (FPGA) 的SAT算法的综述

中山大学逻辑与认知研究所 周 进 赵希顺

【摘要】可满足性问题 (简称SAT问题) 作为第一个被证明的NP完全问题, 是计算机科学的核心问题之一。本文系统总结了基于硬件可编程逻辑 (FPGA--- Field Programmable Gate Array) 的SAT算法研究。将基于FPGA的SAT算法研究分为了实例型 (instance-specified solver) 和应用型 (application-specified solver) 两种类型。通过对各种方法的深入分析, 指出了它们的优点和缺陷, 进而提出未来研究的思路。

【关键词】可编程逻辑; SAT算法; 可满足性; FPGA

逻辑器件可分类两大类: 固定逻辑器件和可编程逻辑器件。固定逻辑器件中的电路是永久性的, 其硬件逻辑结构和连接方式固定不变。而可编程逻辑器件 (FPGA, CPLD等) 内部连接模式和逻辑结构可以根据不同的需要改变, 从而可以实现多种不同的功能。所以, 此类器件能够为客户提供范围广泛的具有多种逻辑能力、速度和电压特性的标准成品部件。

可编程硬件算法研究的发展是和硬件水平的发展密不可分的。可编程硬件是指最早的硬件可编程算法的概念是G. Estrin在1960年左右提出的^[1], 但由于主要以分立器件为硬件平台, 而集成电路也尚在初期, 因此并未得到更多研究和应用的支持。从1990年开始, 超过10K门级的可编程逻辑器件FPGA (其典型结构如图一所示) 的出现, 使得硬件可编程逻辑算法的研究进入了蓬勃发展的时期^[2], 其应用主要分为三种: 1、硬件设计模拟; 2、快速硬件成样; 3、硬件算法研究。而其中硬件算法研究是基于FPGA硬件并行处理能力, 已经在密码学, 图像和信号处理领域取得了很多成果。自1997年后, 随着FPGA进入到了百万门级的水平, 硬件可编程逻辑能力上升到了一个新的高度, 基于硬件可编程逻辑的高难度算法的实现与研究也受到越来越多的关注。其中, 基于FPGA的可满足性问题 (SAT问题) 的算法研究逐渐成为研究的热点^[7-14]。

SAT问题是逻辑学的一个基本问题, 也是当今计算机科学和人工智能研究的核心问题。工程技术、军事、工商管理、交通运输及自然科学研究中的许多重要问题, 如程控电话的自动交换、大型数据库的维护、大规模集成电路的自动布线、软件自动开发、机器人动作规划等, 都可转化成SAT问题。可满足性SAT问题同时也是第一个被证明的NP-Complete问题, 目前解决SAT问题已有的完全算法的运行时间呈指数增长。因此, 基于FPGA (如图一) 对SAT算法加速的研究, 具有很大的理论和应用价值, 同时也有助于寻找优化其他复杂计算的方法。

一、问题定义

对于命题公式Y, 如果存在对Y中变量的赋值, 使得Y在该赋值下的真值为

1, 则称Y是可满足的。可满足性问题 (SAT问题) 是判定一个任意给定的命题公式是否可满足的问题。众所周知, 任何命题逻辑公式都逻辑等价于一个合取范式 (CNF-Conjunctive Normal Form)。合取范式即是子句的合取, 而子句是文字 (变元或其否定) 的析取。

例如, $\psi = (a \vee \bar{b} \vee c)(c \vee \bar{a})$ 就是一个合取范式。当a=1, b=0, c=1时, Y=1。故Y是可满足的。

SAT算法可以分为完全算法 (Complete) 和不完全算法 (Incomplete) 算法两大类。通过完全算法总是可以判断公式是否可以满足, 尽管有可能计算时间是相当长的。如Davis-Putnam 的DP算法^[3]和广泛用于电子电路测试向量生成的PODEM (Path-Oriented Decision Making) 算法^[4]都是完全算法。不完全算法则是规定的时间内寻找SAT问题的解, 如果找到则说明可满足, 但找不到SAT的可满足解不能作为SAT是否可满足的依据。经典的DP算法如下所示, 通过遍历变量空间的方式搜索满足SAT的解。

```

Procedure DPLL (CNF formula  $\Phi$ )
If  $\Phi$  is empty return yes.
else if there is an empty clause
in  $\Phi$  return no.
else if there is a pure literal
l in  $\Phi$  return DPLL( $\Phi$ (l)).
else if there is a unit clause
{l} in  $\Phi$  return DPLL( $\Phi$ (l)).
else
select a variable v occurring in  $\Phi$ .
if DPLL( $\Phi$ (v))=yes
return yes.
else
return DPLL( $\Phi$ ( $\neg v$ )).
end
end
    
```

图二所示的是DP算法的例子。

随着研究的深入, 很多DP算法的优化算法和技术涌现出来, 其中有代表性的如GRASP (Generic search Algorithm for the Satisfiability Problem) 算法中的non-chronological回溯技术^[5]等。

二、研究进展现状和发展趋势

基于可编程逻辑的SAT的算法研究分为两个大的阶段, 前期 (1996-2002) 以实例型 (Instance-Specified Solver)

算法为标志; 后期 (2002-至今) 以应用型 (Application-Specified Solver) 算法为标志。

● 实例型 (Instance-Specified Solver) 算法: 硬件结构针对每一个实例进行编译配置然后计算的方式, 每一个不同的实例对应不同的硬件结构。

● 应用型 (Application-specified solver) 算法: 硬件结构针对应用进行一次编译和配置然后计算, 在同一个硬件机构上对应用中不同的实例进行计算。

(1) 实例型 (Instance-Specified Solver) 算法 (1996-2002)

i) 实例型硬件直接逻辑算法: Suyama et al (1996)

Suyama et al.^[7]在1996年首次提出了实例型硬件直接逻辑算法 (如图三所示)。他首先将SAT实例CNF先转换为3-SAT, 然后由SFL Generator转换为HDL语言, 将其编译下载后在FPGA固化为硬件逻辑。形成CNF的Clause逻辑计算模块。通过在2N-1的空间用穷举法或类似DP回溯算法搜索取值, 对实例的可满足型进行求解。

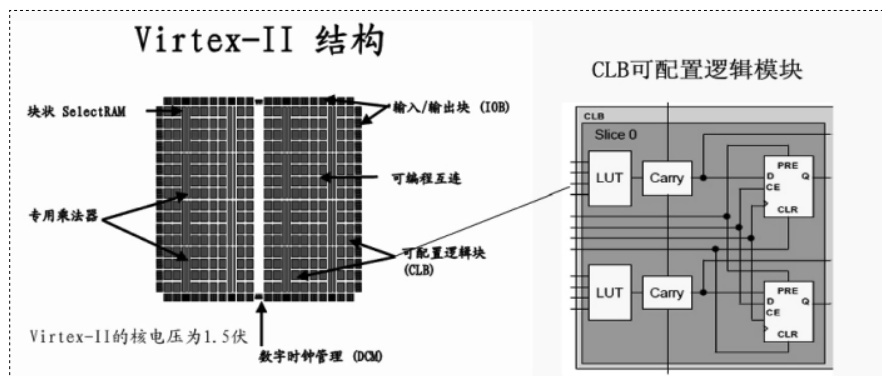
该研究采用DIMACS (Center for Discrete Mathematics & Theoretical Computer Science)^[15]的几个实例在 Altera FLEX10K250 FPGA时钟频率10Hz的平台上进行了验证。测试结果和在UltraSPARC-II/296MHz上的 POSIT (Propositional satisfiability testbed) 结果进行了比较, 速度可以达到1-10倍。但是其计算的速度的计算没有包含FPGA编译和配置的时间。

ii) 实例型硬件状态机 (FSM) DP算法: Zhong et al. (1999)

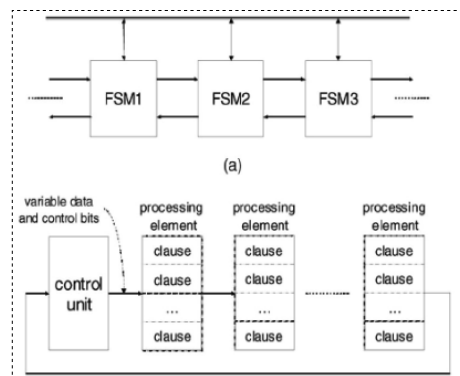
1999年Zhong et al. 提出了如图四所示DP算法的硬件可编程SAT算法^[8]。SAT Clause编译为FPGA硬件Deduction模块和Implication模块, 以采用流水线结构进行计算。算法模型采用了如图四所示的有限状态机 (FSM) 的方式。

实验通过多个FPGA互联形成Xilinx FPGA硬件阵列进行了验证。和著名的软件算法GRASP在Sun5/110MHz/64MB保护模式下比较, 提高了一个数量级的速度, 但该结果同样未将硬件编译和配置时间计入。

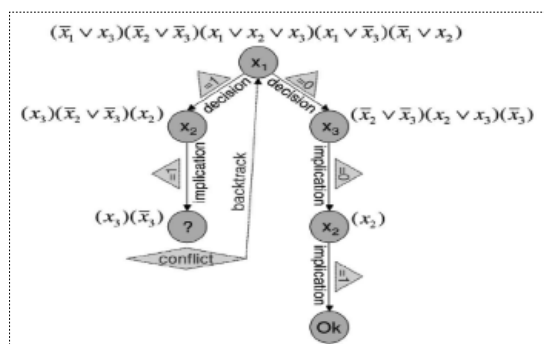
FSM的使用使得该算法可以有效利用DP的相关软件算法中的技术如non



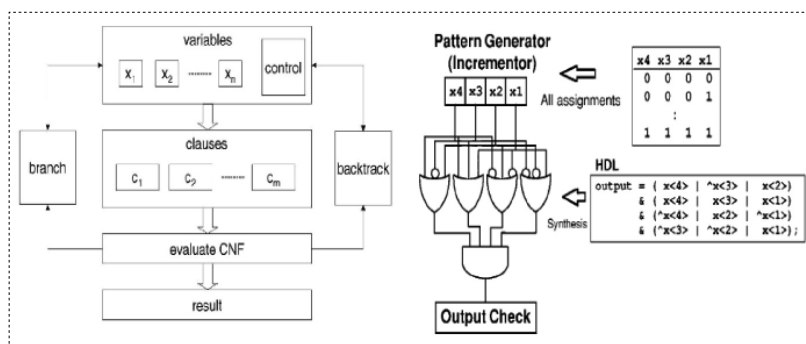
图一 FPGA 的典型结构



图四 实例型硬件算法-有限状态机



图二 DP算法举例



图三 实例型硬件算法-直接逻辑计算

chronological backtracking等。而且结构具有可扩展性,对于大的SAT实例,可以采用互联的FPGA阵列解决。该方法利用了FPGA的管道技术,提高了数据处理速度。但由于实例型算法编译和硬件配置时间漫长,实用性不够,同时运算中每个时钟周期只有一个FSM激活,周期利用率低。

iii) 实例型硬件并行PODEM算法: Abromavici et al. (2000)

Abromavici et al. (2000)利用了电子电路测试中测试向量生成的常用算法,提出了如图五所示的SAT实例型算法的新的思路^[9]。他将CNF编译成为对应的PODEM电路形式,通过向前推导输出模型(Forward model)和向后反推输入模型(Backward model),进行多路并行的推导,得到SAT问题的解。

实验通过DIMACS部分实例在XC6264 FPGA,主频3.5 MHz的平台上进行了验证。和著名的软件算法GRASP在Sun Ultra Sparc工作站1026Mb RAM和248MHz主频下比较,未将硬件编译和配置时间计入的情况下,提高了1.01~7000倍的速度。

iv) 实例型硬件混合算法: Dandalis et al. (2002)

Dandalis et al.首次提出了如图六所示的软硬件混合的算法,将算法在软件和硬件中进行功能划分,从而将算法中耗时最多的Deduction模块和Implication模块由FPGA硬件来实现^[10]。

该研究在硬件设计中采用了管道(Pipeline)技术,将CNF的Deduction分成了并行的N路M容量管道,并将其结

果在m次周期后合并(结果)吗,并且采用了FPGA动态加载的功能将Implication生成的learnt Clause构成新的管道。

实验通过DIMACS部分实例在Virtex XCV1000-6-FG680 FPGA进行了验证。验证结果:没有和其他算法进行比较,主要对比了不同的N值下,算法速度的影响。该算法使用了FPGA中的动态配置功能,同时使用Pipeline和immerge的功能对PFPGA并行能力的利用提出了新的思路。但其整体算法系统未完善,还未能进行实际的实验对比验证,通样存在编译载入时间过长的问题。

(2) 应用型(Application-specified solver)算法 (2002-至今)

i) 应用型SAT FPGA DP混合算法: Sousa et al. (2002)

Sousa et al.首次推出了SAT的DP应用型FPGA硬件算法,即一次编译下载,FPGA将可以解决不同的Instance问题,而不再需要再编译重载。

该算法以特殊3SAT为对象,采用寄存器阵列存储CNF的实例将算法在软件和硬件中分工^[11-12]。如图七所示,软件模块部分主要处理:冲突Conflict分析,回溯backtrack控制,语句的数据库管理;硬件模块部分:并行推导Deduction和变量选择Decide。

通过3SAT实例在RC1000 PCI板上集成XCV2000E Xilinx FPGA和4块SRAM(2M)的平台上进行了验证。但由于没有建立软件接口,因而没有进行对比和得到实际的结果。

该研究对于应用型的FPGA硬件算法

进行了创造性的尝试,用动态更新寄存器的方法解决FPGA对于不同的实例配置的问题。采用了虚拟内存的概念解决当FPGA容量时的问题,但仅限于对于3SAT的解决方案。

ii) 应用型SAT FPGA DP混合算法: Skliarova et al. (2004)

Skliarova et al.采用了如图八所示的算法,以三态存储阵列(0, 1, free)作为联系变量和Clause阵列的矩阵模块,从而实现了将SAT的实例存储到矩阵模块中的转换^[13]。算法为求解该矩阵每个Clause都正交向量V,如果该向量存在则SAT的实例可满足。

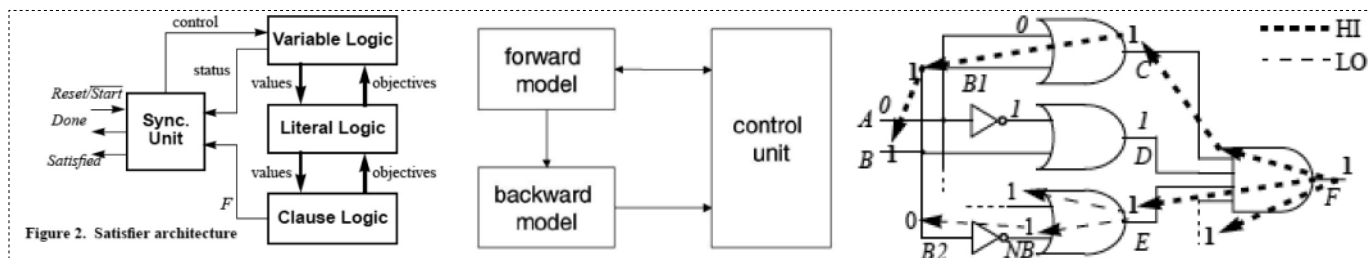
实验通过DIMACS benchmark中的几个实例在:ADM-XRC PCI板上集成XCV812E V EM FPGA 40MHZ平台上进行了验证。

验证结果:和采用GRASP算法在AMD/1GHZ/256M计算机上的计算提高了2个数量级(包含实例载入的时间)

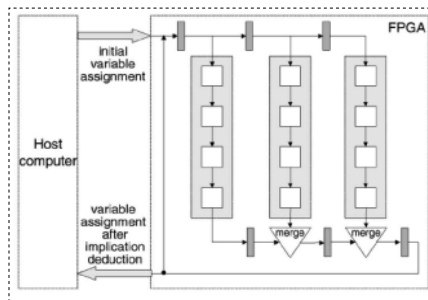
该研究采用了三态存储阵列和矩阵模型,将应用型SAT FPGA的算法的对象,扩展到了一般SAT实例。该算法实现了软硬件的系统 and 接口,形成较完整算法功能。但提出的矩阵模型在不同应用实例中效果和效率差异很大,并且该算法只是用硬件实现了SAT的算法,并未有效的利用硬件的并行处理能力。

iii) 应用型SAT FPGA DP混合算法: John D. Davis et al. (2008至今)

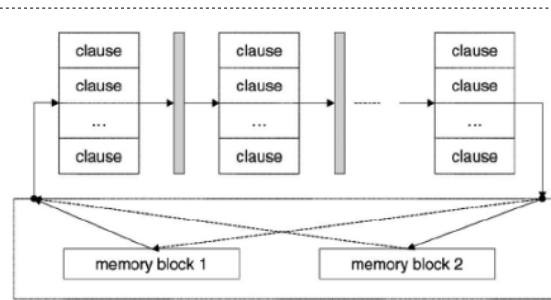
John D. Davis et al.将软硬件模块功能进行较合理划分,如图九所示:建立和实现了BCP在硬件FPGA中的运算和相关接口模型^[14]。算法使用了Index walk



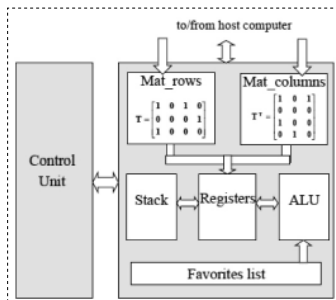
图五 实例型硬件算法-并行PODEM算法



图六 实例型硬件算法-软硬件混合算法



图七 应用型硬件算法1



图八 应用型硬件算法2

的技术更紧密的将Clause阵列存储在BRAM中，同时采用硬件并行方式：将BCP的功能通过Inference模块，Mux模块和Conflict detection模块以及输入输出模块完成。

实验通过Crypto benchmarks和k-SAT实例 ($3 \leq k \leq 6$) 在HyperTransport (HT) 和PCI-Express两种高速板上集成FPGA (200MHZ) 的平台进行的测试。

验证结果：和采用GRASP算法在Pentium 4/3.6 GHz/2 GB RAM系统上的计算提高了3.7和38.6倍 (包含实例载入的时间)。

该应用型SAT FPGA DP混合算法目前为止比较完整实现功能的算法，其创造性的Index walk算法将实例有效和紧凑的存储，便于在同等资源下更大的实例计算。该算法通过多个并行的模块对BRAM进行操作，有效的利用了FPGA的并行性。但由于高速计算下，软件和FPGA硬件的通讯和同步较困难，不同的实例的工作量协调成为需要解决的问题。

三、概括总结

从上面的研究情况中可以得出以下的概括总结：

(1) 算法类型：分为实例型和应用

型两种。早期的研究以实例型为主，由于对于每个实例都有漫长的编译下载时间，其实用性和发展都有限；近期的研究都以应用型为主，在对应用的实例的算法加速过程中取得了初步的进展。

(2) 执行模式：分为单纯的硬件算法和混合型算法。由于FPGA硬件的资源有限，研究中大部分采用了混合型的算法，将SAT算法功能在软件和硬件之间划分。划分分为两种方式：按照计算复杂性划分：将计算量大的，可以并行计算的功能划分给硬件系统；按照存储量或者逻辑容量来划分，FPGA不能容纳的部分由软件来管理和控制。

(3) 逻辑容量：硬件系统的计算能力受其硬件资源的限制，与应用SAT实例需求相比始终不足，而采用了多种扩展方式进行扩展：

- i) 多个硬件组成阵列的方式。
- ii) 将硬件中的功能采用并联或流水线的方式划分。
- iii) 按照硬件的存储量和逻辑容量来划分，其余部分软件管理，如采用软件系统中的虚拟内存概念在硬件中调配。

(4) 计算能力。硬件算法的时间由

四个部分组成：硬件编译时间，硬件配置时间，数据通讯或下载时间，实际执行时间。实例型硬件算法受硬件编译时间限制较大，应用型硬件算法决定于通讯时间和实际执行时间。但由于各种研究的硬件平台各不相同，大都采用与类似DIMACS benchmark的实例用软件算法的计算结果比较，但由于软件运行的计算机系统的各不相同，其实际速度提升很难把握。在和软件进行对比中大部分采用了GRASP的SAT计算结果，但在2002年以来其计算能力就已经被zChaff和BerkMin等越来越多的算法超越，要提高硬件算法的实用性，需要找到更有效的评估方法。

四、研究思路

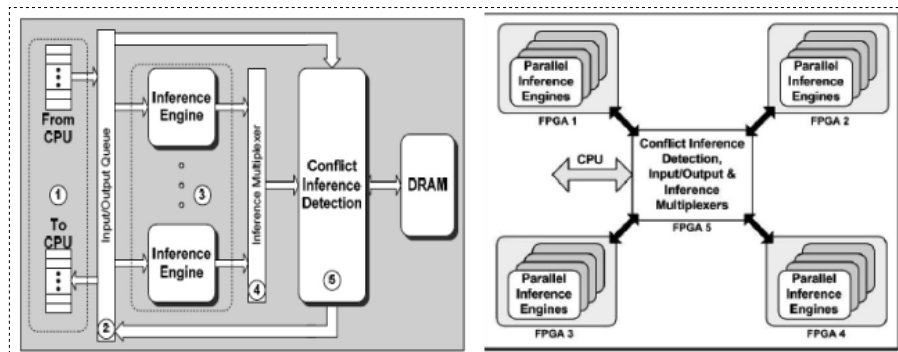
实例型 (Instance-Specify) 的编译配置过程本身就是一个SAT的可满足性问题求解的过程，时间过长，在研究中将着重在更具有实用价值应用型 (application-Specify) 硬件算法。硬件平台的资源与SAT应用的需求相比总是不足，很多SAT软件算法中的先进技术硬件还暂时很难实现，将功能在软件和硬件上合理分工的混合型算法更具合理性。

下一步研究将致力于：1、找到合理的应用型硬件算法，充分发挥硬件的并行处理能力，从而提高算法速度，降低同步过程中对速度的影响，达到提升整体的计算速度效果。2、建立不单纯使用计算时间来进行算法速度比较的硬件算法评估方式，使不同平台的硬件算法速度更具有可比性和参考价值。

参考文献

- [1] Estrin, G.: Reconfigurable Computer Origins: The UCLA Fixed-Plus-Variable (F+V) Structure Computer. IEEE Annals of the History of Computing, Oct.-Dec. (2002) 3-9.

(下转第69页)



图九 应用型硬件算法3

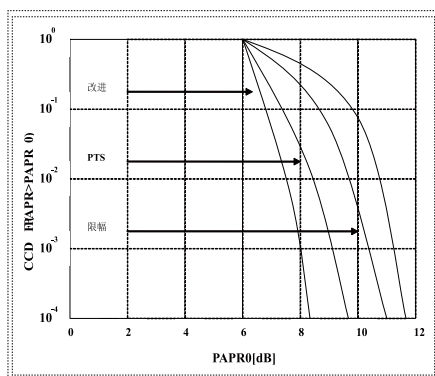


图3.1 限幅法、PTS法以及改进算法降低PAPR的CCDF曲线图

不会使信号产生畸变，它的主要方法有两种，一种是选择性映射法SLM，另一种是部分传输法PTS。

SLM方法的主要思想是利用多组统计独立的向量来表示同一组要传输的信息，然后选择其中具有最小PAPR的那一组序列进行传输，同时随机向量也要作为边带信息传输，这些边带信息是SLM的冗余，因此它牺牲了一些发送带宽。同时因为要进行多次IFFT，从而加大了系统的复杂度。为了弥补不足，可以对SLM方法进行改进，例如可以把SLM与ACE技术相结合在不降低比特误差率的基础上可以大大降低系统的复杂度。还可以让SLM与Turbo编码技术相结合，可以在降低峰均比的基础上进行纠错和改善比特误差率。

PTS方法的主要思想是把输入的OFDM符号 X ，分割成 n 组互不重叠的子序列，用 $X(n)$ 来表示，并分别对每一组子序列乘以不同的权值，然后通过比较得到的 n 组序列的PAPR，选择其中最小PAPR的那一组序列进行传输。

3. 一种降低峰均比的改进方法

限幅类，编码类以及概率类技术对降低峰均功率比的效果不同，它们各有自己的优缺点。限幅类技术的优点主要是它可以有效地降低信号的PAPR，操作并不复杂，其计算量不大，但是由于它对信号进行的是一种非线性操作，因而

会给系统带来噪声和干扰，增加系统的误码率。而PTS和SLM技术的优点是它们都是对信号进行线性操作，在接收端可以完全进行恢复，所以不会给系统引入噪声，造成干扰。但它们的计算量比较大，对峰均功率比的降低幅度不如限幅类技术那样明显。可以看出限幅类技术和PTS技术之间在性能上存在互补性。事实上在很多时候，只运用一种技术方法并不一定能够达到降低峰均功率比的理想的效果，有时为了取得较好的效果，只能在某些方面付出很大的代价，这些都不能令人满意。按照上面的分析，可以考虑将几种方法联合起来降低OFDM系统的PAPR。

将PTS法和限幅法联合起来，有两种顺序，一种是先进行限幅再进行PTS，另一种则是先进行PTS再进行限幅，这两种顺序对降低PAPR性能的效果不一样。因为PTS法是一种线性操作，不会造成信号的失真，而限幅法是一种非线性操作，进行限幅操作的点数越多，引起的限幅噪声就越多。如果先进行限幅操作再进行PTS方法操作，则限幅法降低PAPR的效果会在PTS阶段得到破坏，同时限幅法给系统带来的噪声和干扰得不到改善；而如果先进行PTS法操作再进行限幅法操作，那么PTS法降低PAPR的效果在限幅过程中不会受到影响，另一方面，限幅法的输入信号是已经通过PTS法降低过峰值的，这样这些信号超过限幅门限的点数比没有经过PTS法处理的时候要少，那么这个时候的限幅引入的噪声和干扰就会减少。通过上述的分析，显然，改进算法先进行PTS处理再进行限幅处理要合理有效。它的实现步骤为：首先将输入的信号数据分成 V 组，让每组的信号数据和相位因子相结合，组成 U 路备选信号。接着对这 U 路信号分别计算峰均功率比，然后选择峰均功率比最小的一路，最后对峰均功率比最小的那一组信号进行限幅、滤波处理并发送出去。

在子载波数 $N=512$ ，限幅因子 $CR=4$ ，并且采用4倍过采样，调制方式为QPSK的情况下，PTS采用 $V=4$ 的分组方式，对

10000个OFDM符号用联合算法进行MATLAB仿真，得到的CCDF曲线如图3.1所示。

从图3.1中可以看出，改进算法降低OFDM系统峰值功率比的效果要好于单独的限幅法和PTS法，具体来说，在CCDF为1/100的情况下，原始信号的峰值功率比约为10.8dB，而经过限幅后的峰值功率比约为9.8dB，经过PTS后的峰值功率比约为8.5dB，经过限幅和PTS联合后的峰值功率比约为7.5dB，可见，在CCDF为1/100的情况下，限幅法使原始信号的PAPR下降了1dB，PTS法使原始信号的PAPR下降了2.3dB，而改进算法使原始信号的PAPR下降了3.3dB。在CCDF为1/1000的情况下，原始信号的峰值功率比约为11.3dB，而经过限幅后的峰值功率比约为10.4dB，经过PTS后的峰值功率比约为9.2dB，经过限幅和PTS联合后的峰值功率比约为8dB，可见，在CCDF为1/1000的情况下，限幅法使原始信号的PAPR下降了0.9dB，PTS法使原始信号的PAPR下降了2.1dB，而改进算法使原始信号的PAPR下降了3.3dB。可见，改进算法和单一的限幅法和PTS法相比，对降低OFDM系统的PAPR效果确实要更好一些，这证明了其有效性和可行性。

4. 结束语

总之，OFDM技术具有广泛的发展前景，是目前各国通信领域研究的热点，峰均功率比问题是OFDM系统中一个比较关键的问题，本文主要研究了如何降低OFDM系统的峰值平均功率比，实际运用时，不但要考虑到算法对降低系统PAPR性能的效果有多大，还要顾及到算法实现的复杂度，同时还需要权衡算法在实现的过程中对接收端解调时带来的影响，因此还要进一步研究和学习。

参考文献

- [1] 佟学俭, 罗涛. OFDM移动通信原理与应用[M]. 北京: 人民邮电出版社, 2003.
- [2] 张荣. OFDM系统降低峰均比技术研究[D]. 陕西: 西安电子科技大学, 2009.
- [3] 白玉. OFDM技术的分析与研究[D]. 山西: 太原理工大学, 2008.
- [4] 赵君洁, 张海林. OFDM系统原理及关键技术[J]. 现代交换技术, 2009.

(上接第63页)

- [2] S.A. Guccione, "Reconfigurable Computing at Xilinx," keynote talk, EUROMICRO Symp. Digital System Design, Sept. 2001.
- [3] M. Davis, G. Logemann, and D. Loveland, "A Machine Program for Theorem Proving," Comm. ACM, no. 5, pp. 394-397, 1962.
- [4] P. Goel, "An Implicit Enumeration Algorithm to Generate Tests for Combinatorial Logic Circuits," IEEE Trans. Computers, vol. 30, no. 3, pp. 215-222, Mar. 1981.
- [5] L.M. Silva and K.A. Sakallah, "GRASP: A Search Algorithm for Propositional Satisfiability," IEEE Trans. Computers, vol. 48, no. 5, pp. 506-521, May 1999.
- [6] B. Selman, H. Levesque, and D. Mitchell, "A New Method for Solving Hard Satisfiability Problems," Proc. Nat'l Conf. Am. Assoc. Artificial Intelligence (AAAI' 92), pp. 440-446, July 1992.
- [7] M. Yokoo, T. Suyama, and H. Sawada, "Solving Satisfiability Problems Using Field Programmable Gate Arrays: First Results," Proc. Second Int'l Conf. Principles and Practice of

Constraint Programming, pp. 497-509, 1996.

- [8] P. Zhong, M. Martonosi, P. Ashar, and S. Malik, "Using Configurable Computing to Accelerate Boolean Satisfiability," IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems, vol. 18, no. 6, pp. 861-868, 1999.
- [9] M. Abramovici and D. Saab, "Satisfiability on Reconfigurable Hardware," Proc. Seventh Int'l Workshop Field-Programmable Logic and Applications, pp. 448-456, 1997.
- [10] A. Dandalis and V.K. Prasanna, "Run-Time Performance Optimization of an FPGA-Based Deduction Engine for SAT Solvers," ACM Trans. Design Automation of Electronic Systems, vol. 7, no. 4, pp. 547-562, Oct. 2002.
- [11] J. de Sousa, J.P. Marques-Silva, and M. Abramovici, "A Configurable/Software Approach to SAT Solving," Proc. Ninth IEEE Int'l Symp. Field-Programmable Custom Computing Machines, 2001.
- [12] N.A. Reis and J.T. de Sousa, "On Implementing a Configurable/Software SAT Solver," Proc. 10th IEEE Int'l Symp. Field-Programmable Custom Computing

Machines, pp. 282-283, 2002.

- [13] I. Sklarova and A.B. Ferrari, "A Software/Reconfigurable Hardware SAT Solver," IEEE Trans. Very Large Scale Integration (VLSI) Systems, vol. 12, no. 4, pp. 408-419, Apr. 2004.
- [14] Kestur, S.; Davis, J.D.; Williams, O. "BLAS Comparison on FPGA, CPU and GPU" VLSI (ISVLSI), 2010 IEEE Computer Society Annual Symposium 2010.
- [15] DIMACS challenge benchmarks. [Online]. Available: <http://www.intellectik.informatik.tu-darmstadt.de/SATLIB/benchm.html>.

注：国家自然科学基金资助项目（项目编号：60970040）。

作者简介：周进（1973—），男，重庆人，博士研究生，现供职于中山大学逻辑与认知研究所。