结论与展望

## 结论

随着集成电路的集成度和复杂度不断提高，导致集成电路的测试工作面临诸多挑战。芯片日益增加的集成度和复杂度直接提高了电路产生故障数的概率。为了确保被测芯片的故障覆盖率达标，测试向量需要成倍的增加。庞大的测试数据增加了硬件代价和测试应用时间，一种卓有成效的方法就是对测试数据进行压缩，通过压缩测试数据不仅能大大降低被测时间更可以节约硬件存储开销。近年来有关学者提出了一种拆分压缩技术，此技术将原测试集拆分成为主分量集（在本文中由基向量生成）以及残分量集，主分量的选取直接影响最终的压缩率，本文在拆分压缩技术的基础上，对如何生成基向量展开了研究。

第一章绪论，主要介绍研究背景及研究意义、集成电路测试的相关先修知识、国内外研究现状、本文的主要工作以及文章的组织架构。第二章首先讲解测试的相关概念基础，接着介绍了随机测试技术以及常用的编码压缩技术，第一种是游程编码，第二种的字典编码，第三种是统计编码。随后花了少量篇幅介绍了两种非编码压缩方法，最后介绍了拆分压缩技术，基本思想是将原测试集，映射到另一个更易于压缩的表示空间，来达到提高压缩率的目的。本文就是在拆分压缩技术的基础上通过挑选出合适的基向量进行数据压缩。

本文基于测试集拆分压缩技术提出了三点创新性工作，主要有：

1. 采用预填充的策略对测试集进行处理。此方法首先对测试数据进行预填充，填充的方式有直接填充和策略填充两种，填充的目的是使测试集在当前编码规则下的压缩率更高，填充完毕后的测试集不存在无关位，然后以向量间距离最大最原则选取需要的基向量。由于本文使用的是拆分压缩技术，当原测试及中包含的码字0较多会有助于压缩率的提升，建议直接将无关位填充为码字0。实验结果表明，使用预填充的方式，RL-Huff编码的压缩率可达74.32%**，**相比与对测试集进行直接编码，压缩率提高了11.75%，并且硬件开销也是可接受的。通过对ISCAS’89部分基准大电路的实验结果表明，本方法取得了较理想的效果。
2. 提出了一种基于kmeans++聚类算法结合测试集生成基向量的方法。该方法的基本思想是对已填充的测试向量进行聚类，即相似的列向量我们将其归为一类，然后取当前聚类列向量每一位的均值作为聚类中心向量。然后以每一个聚类中心向量为基准，将测试集中的每一个列向量与以获得的聚类中心向量进行对比，若欧几里得距离最小则归为一类，如此反复迭代，当聚类基本不再发生变化时，最终确定的聚类中心向量即为我们所求的基向量。实验结果表明，通过使用kmeans++算法生成的基向量来进行压缩， RL-Huff编码的平均压缩率可达76.30%，与对测试集进行直接编码压缩相比平均压缩率提高了13.73%，与哈达玛相比，压缩率提高了4.45%。同时本人使用此方法对大电路进行了测试，在FDR编码编码方式下，比对测试集直接压缩所获取的缩率高 6.06%。由于基向量的选取的个数会直接影响最终的压缩率，为了更好的反映两者的对应关系，本人通过选取不同个数的基向量，计算出相应的压缩率，并绘画出其相应的折线图。
3. 将kmeans++聚类算法结合单轮位翻转算法进一步提高压缩率。此方法的主要思想是，先对原测试集进行预填充，利用kmeans++聚类算法找出所需的基向量并生成主分量集，然后使用主分量集进行故障模拟检测出一部分故障，原测试集只需检测出剩余故障即可，对于原测试集而言，可以将主分量能检测出故障相应确定位转化成为无关位，生成新测试集。最后对新测试集使用向量分解的方式进行压缩，本人基于多轮位翻转算法提出一种保留原主分量集合的单轮位翻转算法，使得在进行故障模拟时，所需要的硬件代价更小。结果表明使用位翻转算法结合kmeans++算法可以将平均压缩率在（2）的基础上提高7%。

## 展望

本文所研究的三个基于拆分压缩中基向量的生成方法虽然在研究上取得了一定的成功，但仍然有缺陷和待改进的地方，总结如下：

1. 本文在第三章使用采用预填充的策略对测试集进行处理，并且其取得了较好的效果，但是主要因为是因为基向量选取方式所导致的，在预填充的测试集中，如果选举基向量的策略为随机选取，而不是使用向量间距离最大原则为依据选取会对压缩率产生较大的影响，实验表明使用随机选取方式所达到的压缩率，与使用哈达码变换所达到的压缩率十分接近。虽然本文使用了据间距最大原则，但是第一列基向量的选取依旧是随机的，初始基向量的选取对实验结果的影响较为明显，因此预填充之后，对于基向量的选择还可以继续优化，减少随机选取带来的误差。
2. 本文第四章提出了一种基于kmeans++聚类算法结合测试集生成基向量的方法。该方法的基本思想是对已填充的测试向量进行聚类，以每一个聚类中心向量为基准，将测试集中的每一个列向量与以获得的聚类中心向量进行对比，若欧几里得距离最小则归为一类，如此反复迭代，当聚类基本不再发生变化时，基向量获取结束。由实验结果可知此方法能获取较高的压缩增益，但也有缺陷，第一kmeans++算法是一个以距离为基准的算法，简单易用，其缺点是无法确定聚类数目，需要事先设定，其次在聚类时初始基向量的选取也是随机的对实验结果也会存在一定的影响。综上所述，在聚类算法的选择上可能还有比kmeans++更合适的算法，能解决上述问题，进一步提高压缩率。
3. 本文第五章将kmeans++聚类算法结合位翻转算法进一步提高压缩率，此方法基本思想是对在不影响原测试集故障覆盖率的前提下，将测试集中部分确定位翻转为无关位。虽说针对ISCAS’99基准电路，此方法提升了较高的压缩率，但是本文中提及的翻转算法只翻转了一轮，并且翻转时使用的是贪婪算法，可能无法到达全局最优的效果，因此优化翻转算法进一步增加可翻转的确定位，比如可以根据对比测试集检测的故障集合以及相似主分量集检测的故障集合来设计翻转算法。