August 2019
PhD Thesis

Input Vulnerability-Aware ATMR and Fault Tolerant Voter for Higher Fault Coverage and Reduced Overheads

Graduate School of Chosun University

Department of Computer Engineering

Tooba Arifeen

Input Vulnerability-Aware ATMR and Fault Tolerant Voter for Higher Fault Coverage and Reduced Overheads

데이터 분포 특성을 이용한 효율적인 어림 삼중장치와 다수결 회로 구조 연구

August 23, 2019

Graduate School of Chosun University

Department of Computer Engineering

Tooba Arifeen

Input Vulnerability-Aware ATMR and Fault Tolerant Voter for Higher Fault Coverage and Reduced Overheads

Advisor: Prof. Jeong A. lee

A thesis submitted in partial fulfillment of the requirements for a PhD degree

April, 2019

Graduate School of Chosun University

Department of Computer Engineering

Tooba Arifeen

아리핀 투바 박사학위논문을 인준함

위원장	조선대학교 교수	**모상만**	
위 원	조선대학교 교수	**신석주**	
위 원	조선대학교 교수	**강문수**	
위 원	전남대학교 교수	**김철홍**	
위 원	조선대학교 교수	**이정아**	

2019 년 06 월

조선대학교 대학원

TABLE OF CONTENTS

LIST OF ABBREVIATIONS AND ACRONYMS

TMR	Triple Modular Redundancy
ATMR	Approximate Triple Modular Redundancy
ATPG	Automatic Test Pattern Generation
PBS	Pre-Blocking Stage
AMES	Approximate Modules Extraction and Selection Stage
FAN	Fan-Out-Oriented
PI	Prime Implicants
MEROP	Minterms/Maxterms for Expansion/Reduction of Original PI
QoC	Quality of Circuit
FMR	Fault Masking Ratio
IC	Integrated Circuit
SEUs	Single Event Upsets
COTs	Commercial-Off-the-Shelf
R_M	Reliability of Module
R_{sys}	Reliability of System
R_T	Reliability of TMR
R_v	Reliability of Voter
R_{TMR}	Reliability of TMR with voter
R_{AT}	Reliability of ATMR
R_{ATMR}	Reliability of ATMR with voter
AM	Approximate Module
PTL	Pass Transistor Logic
s-a-1	Stuck at 1
s-a-0	Stuck at 0
SET	Single Event Transient
VIV	Vulnerable Input Vectors
IIV	Invulnerable Input Vectors
PDP	Power-Delay-Product
PDAP	Power-Delay-Area-Product

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Input Vulnerability-Aware ATMR and Fault Tolerant Voter for Higher Fault Coverage and Reduced Overheads

Tooba Arifeen
Advisor: Prof. Lee, Jeong-A, Ph.D.
Department of Computer Engineering
Graduate School of Chosun University

In Integrated Circuit design industry, the cons of aggregated complexity and probability of greater number of faults is inevitable. Fault masking can be provided through Triple Modular Redundancy (TMR) but TMR suffers from a 200% area overhead issue. Approximate Triple Modular Redundancy (ATMR) was introduced as a solution to this issue. In ATMR, approximate circuits for TMR modules are used and only one of the modules diverges from original circuit at every input vector, permitting the majority voter to still choose two match outputs out of three for any input vector.

Contradicting from TMR, ATMR is vulnerable to errors and approximations at critical inputs must be avoided. Automatic test pattern generation (ATPG) can identify vulnerable input vectors. In this thesis, we present an ATMR using ATPG that aims for lesser area overheard and greater fault coverage, with the benefit of protecting input space that is highly vulnerable to errors and lessening the search space for ATMR candidates. The work focuses on identifying critical input space through ATPG and making it unavailable for the technique of approximating modules of TMR, which involves a prime implicant reduction expansion. The

technique sets well-defined criteria for managing the problem of formulation of best possible combination of approximates modules for ATMR. The work contributes towards a input- vulnerability aware heuristic method for successive generation of ATMR modules. The proposed method provides 75% to 98% fault coverage, which amounts up to 43.8% improvement over that achieved previously. The input vulnerability-aware approach enables drastic reduction in search space, ranging from 41.5% to 95.5%, for selection of candidate ATMR modules and no compromise on area overhead reduction is noticed.

In TMR , if a fault arises which flips one of the inputs of voter, the voter output will still be accurate since the remaining inputs will coincide with each other. As ATMR authorities one of the approximate modules to disagree from original circuit at each input vector, there will inevitably be instances in which two of the voter inputs will be identical, and one input will be distinctive from them. If a fault arises at voter input it could be unfavorable for the ATMR technique. Prevailing research on fault tolerant voters have concentrated upon TMR as ATMR is a recent notion. We highlight the reliability of ATMR, in comparison with TMR and then, present a novel compact, low-power, high-speed, fault-tolerant voter for ATMR. We present a transistor-level analysis of fault-tolerant voters. For prior insight of a voter circuits, we present a metric, called Quality of Circuit (QoC) which acknowledges the inherent ability of a digital circuit to mask all probable internal faults for a given input vector. The proposed voter delivered upto 45.1%, 62.5%, 26.6%, 50% and 56% improvement, as compared to previous works, in Fault Masking Ratio (FMR), QoC, and reliability, transistor count and power delay product, respectively.

한 글 요 약

데이터 분포 특성을 이용한 효율적인 어림 삼중장치와 다수결 회로 구조 연구

아리핀 투바
지도 교수: 이정아
컴퓨터공학과
대학원, 조선대학교

집적 회로의 밀도와 복잡도가 증가함에 따라, 회로에서 오류가 발생할 확률은 더욱 증가하고 있다. 이러한 오류에 안정적으로 대처하기 위하여, 삼중화 시스템을 활용할 수는 있으나, 이 경우 하드웨어 오버헤드가 200%가 증가하는 문제가 있다. 최근, 이러한 오버헤드를 해결하는 방안으로, 모든 입력에 대하여, 적어도 두개의 장치가 동일한 출력을 생성할 수 있는 장치의 간소화를 허용하는 어림 삼중 장치(Approximate Triple Modular Redundancy)가 제안되었다. 어림 삼중 장치의 경우에도 다수결 회로를 통하여, 모든 입력에 대하여 삼중 장치와 동일한 출력을 생성할 수는 있다.

적어도 두개의 장치가 동일한 출력을 갖도록 간소화를 허용한 어림 삼중 장치에서는 출력 오류를 발생시킬 수 있는, 중요한 입력의 간소화가 일어나지 않도록, 장치의 간소화를 설계하는 일이 중요하다. 자동 테스트 패턴 생성(Automatic Test Pattern Generation) 기법을 활용하면, 회로의 취약성을 나타내는 입력 벡터를 식별할 수 있다. 본 논문에서는, 자동 테스트 패턴 생성(ATPG)을 이용하여 어림 삼중 장치를 설계하는데, ATPG는 오류에 매우 취약한 입력 공간을 보호할 수 있게 하고, 어림삼중장치(ATMR)후보에 대한 검색 공간을 줄이는 이점을 가지고 있다. ATPG를 통해 중요한 입력 공간을 식별하고, 어림

삼중 장치 설계에서 본 입력들이 근사화에 활용되지 않도록 하는데 초점을 맞추고 있다. 이 기법은 어림삼중장치를 구성할 근사 모듈 조합 선택의 기준으로, 연속적인 ATMR 모듈 생성을 위한 입력-취약성 인식 휴리스틱 방법에 기여한다. 본 논문에서 제안된 방법은 75 98%의 결함 커버리지를 달성하여 기존의 방식에 비하여 최대 43.8% 향상됨을 보였다. 본 논문에서 제안한 방식은 어림삼중 장치 후보 모듈 선정 관련 검색 공간을 기존에 비하여 41.5% 95.5% 대폭 줄일 수 있으며, 축소된 검색공간에서 찾은 어림 장치의 하드웨어 오버헤드 감소는 기존 방식에 비하여 다름이 없었다.

삼중 장치의 경우에는 한 장치에서 오류가 발생하더라도 다른 두 장치의 출력이 동일하기 때문에, 다수결 회로 (Voter)를 통과한 출력에는 오류가 없다. 반면에 적어도 두개의 장치가 동일한 출력을 생성할 수 있는 장치의 간소화를 허용하는 어림삼중장치의 경우에는, 한 장치에서 결함이 발생하면, 다수결 회로를 통한 출력물의 신뢰성은 항상 유지되지 못한다. 본 논문에서는, 어림삼중장치의 신뢰성 유지에 필요한 새로운 다수결회로를 제안한다. 내부 회로의 결함이 출력의 오류로 바로 연결되지 않는 경우를 측정하는 Quality of Circuit (QoC) 지표를 제안하고, 트랜지스터 결함에 대한 분석을 통하여 제안하는 다수결 회로의 우수성을 보인다. 어림 삼중 장치의 신뢰성 유지를 위하여 제안하는 다수결 회로는 각각 최대 45.1%, 62.5%, 26.6% 향상된 오류 결함 은폐 비율 (Fault Masking Ratio), QoC 및 신뢰성을 달성하였고, 트랜지스터 카운트와 전력-지연시간 곱 측면에서 각각 최대 50%와 56 %의 개선을 달성하였다.

I. INTRODUCTION

A. An Overview of Approximate TMR

The Integrated Circuit (IC) industry is on the verge of one of its most crucial eras. Despite the pros of extra density and potential performance of ICs, the cons of increasing complexity and likelihood of higher number of faults are also inevitable [1], [2]. The quest for reliability, the probability of not incurring a failure within the given operation period, has become a vital aspect for circuit designers and manufacturers [3]–[5]. The International Technology Roadmap for Semiconductors has reported that reliability is a challenge for nano-electronics owing to the technological issues posed by the ever-shrinking semiconductor devices [6]. Thus, the next generation of very deep sub-micron systems will be defined by their reliability and performance [1]. For a realistic analysis of the reliability, fault modeling must be conducted at the transistor level [7].

Primarily, fault prevention and fault tolerance are sought to yield reliable digital systems. Fault prevention aims at a priori elimination of all faults in which the system practically targets an acceptably low chance of failure [1]. Fault tolerance is the ability of a system to mitigate faults that arise during an operation. Fault tolerance is the assurance of a correct operation despite a fault occurrence, and thus signifies a higher reliability. When it is ensured that the occurrence of an internal or external fault does not affect the actual output of a function module, the fault is known to be masked or efficaciously hidden from observation by the outside world. A fault that does not cause an error is said to be masked. This phenomenon is referred to as fault masking. If a fault is not masked, it will lead to an erroneous output of the function module. In brief, the manifestation of a fault is construed to be an error [8], [9]. Fault tolerance is augmented into the system

for the purpose of fault masking through the inclusion of redundancy techniques such as Triple Modular Redundancy (TMR) [10].

TMR guarantees an accurate output through a majority voter, even when a single module is faulty. TMR is one of the most general and effective hardening techniques. However, TMR suffers from a 200% area overhead problem [5], [11]. Previous works on fault tolerance techniques have proposed the use of selective hardening methods such as Partial TMR [12], Selective TMR [13] and Approximate TMR (ATMR) [11], [14], [15] to overcome this area overhead issue of TMR without significantly compromising the fault masking. In the concept of selective hardening, the word 'Selective' implies that the protection method is providing full protection to selected part of the circuit/system. The simple idea behind selective hardening is that if hardening an entire circuit is too expensive, we can harden only those circuit parts that are particularly exposed to failure or that are critical for correct system functioning [13]. Selective hardening, Selective TMR, and ATMR have the same goal that is full protection of critical parts of the system. Selective TMR, partial TMR, and ATMR benefits from the insertion of TMR in the nodes that present a greater vulnerability to single event upsets (SEUs) [16]. In the case of ATMR, TMR uses approximated logic circuits to generate redundant modules that are optimized for the area, as compared to the original module. This idea consists of generating approximate logic circuits with reduced resource use while maintaining full protection against the most critical parts of the system and no approximations are made in that region. ATMR provides an additional level of selective hardening as depicted by Figure 1.

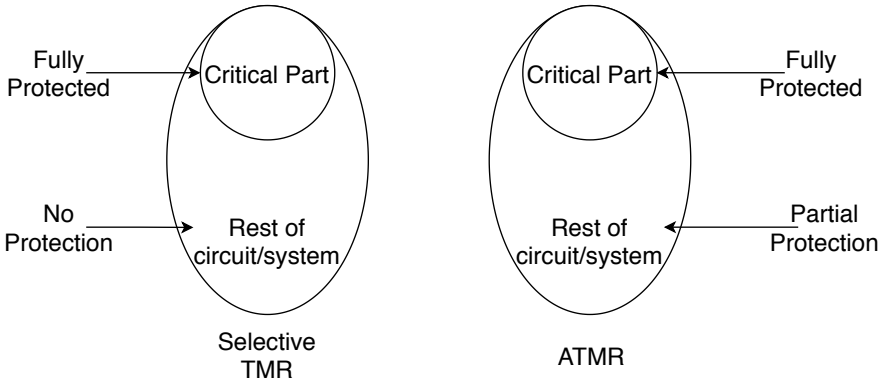


Figure 1: Selective Hardening: Selective TMR & ATMR.

For hardening purposes, the criticality of an application defines the choice of trade-off between a reliability enhancement and the cost allied with a fault tolerant design. Certainly, maintaining the balance between cost-budget and acceptable error rate constraint is imperative [17]. The experiments carried out in space also present a case of selective hardening. For instance, radiation hardened modules such as the LEON-FT microprocessor are used in spacecraft's mission-control computer whereas the scientific gadgets are mostly developed over commercial-off-the-shelf (COTS) parts with slight or no fault protection. This owes to the facts that radiation-hardened equivalents of the COTS parts, if accessible at all, tend to have a lower performance and consume more power, which is a limited resource in a spacecraft, than the COTS parts. The threat of an individual experiment failing might be tolerable, but a critical failure in an on-board mission control system could condense all experiments and complete mission useless [13].

B. Motivation

Triple modular redundancy (TMR) uses the majority voter to mask soft errors where three copies of the circuit are used, but it incurs a price of 200% area overhead. The area overhead problem can be handled through approximate computing using approximate circuits for TMR modules (ATMR) such that only one of the modules differs from the original circuit at each input vector scenario, allowing the majority voter to still select two match outputs out of three for any input vector. Table 1 shows an ATMR, where f1, f2, and f3 are approximate functions, G is the original function, and V is the voter output [18].

X	Y	Z	G	f1	f2	f3	V
0	0	0	0	0	0	**1**	0
0	0	1	0	0	**1**	0	0
0	1	0	1	1	1	1	1
0	1	1	0	0	0	**1**	0
1	0	0	0	**1**	0	0	0
1	0	1	1	1	**0**	1	1
1	1	0	0	0	**1**	0	0
1	1	1	1	1	**0**	1	1

Table 1: Truth table for original and approximate circuits

Let us assume that for a given example in Table 1, a TMR is constructed as $f1 = f2 = f3 = G$ and an error occurs at f2 for input vector 000. In this case, f1 and f3 will ensure that the output of V stays same as that of G. However, if the f2 of ATMR is flipped due to an error, then an erroneous output will be produced as two of the modules (f2 and f3) will vote for the same output value. Thus, unlike TMR, ATMR is vulnerable to errors and is of immense significance to avoid approximations at critical inputs. Note that identifying the most critical

or persistent bits is a difficult task [12].

Please note that a voter is integral to both TMR and ATMR. Because the final output is produced by a voter, a faulty voter poses a direct threat to the system reliability. Research on fault tolerant voters [1], [2], [9], [19] have mainly focused on TMR. In [1], [9], [19] the fault tolerance ability of TMR voters was evaluated at the gate level, whereas [2] assessed fault-tolerant voters at the transistor level while assuming that each module of TMR produces an accurate output.

Since ATMR is a recent notion, voter design for ATMR has not been proposed in literature.

C. Terminologies

1. ATMR and Full ATMR

For a given input vector, ATMR conditionally allows for the outcome of two or more modules to be contradictory. This is achieved by developing a TMR with one module as the original circuit and two modules as approximate circuits, as shown in Table 2. The example in Table 2 represents an ATMR composed of G, f1, and f2, where G is the output of the original circuit, f1 is the output of one approximate circuit for G, and f2 is the output of another approximate circuit for G.

A fuller version of ATMR can be constructed through approximate modules for all three modules of TMR. This is known to as full ATMR (FATMR), a term introduced by [5]. Table 3 presents an example for FATMR made by f1, f2, and f3, where f3 is the output of the third approximate circuit for G. It is ensured that the working principle of ATMR stays valid. Generally, FATMR leads to minimum

Table 2: Approximate TMR-An Example

Input Vectors			Original Circuit	Modules Forming ATMR			Voter Output
				Original	Approximate	Approximate	
X	Y	Z	Output (G)	[G]	Circuit 1 [f1]	Circuit 2 [f2]	[ATMR output]
0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0
0	1	0	1	1	1	1	1
0	1	1	0	0	1	0	0
1	0	0	1	1	0	1	1
1	0	1	1	1	1	1	1
1	1	0	1	1	1	1	1
1	1	1	0	0	1	0	0

area overhead but it is not always guaranteed, as the decrease relies on the original function. Consider a four variable input function $G = \bar{a}cd$. For one unprotected vector, the approximate functions of G are $f1 = ab\bar{c}d$, $f2 = a\bar{b}\bar{c}d$, $f3 = ad(\bar{c}+b)$. For two unprotected vectors, the approximate functions of G are $f4 = ad$, $f5 = a\bar{c}$ To construct an Approximate-TMR for 4 unprotected vectors, the options f1/f5/f2 and f1/f4/f3 along with other possibilities. These FATMR versions have an area overhead of 12 literals which is obviously even worse than TMR (9-literals). In contrast, an ATMR of G/f4/f5 will have an area overhead of 7 literals only. As the approximate function with 1 unprotected vector is even bigger than G itself, good enough composition of FATMR for four unprotected vectors is not available.

Table 3: Full-Approximate TMR-An Example

Input Vectors			Original Circuit	Modules Forming FATMR			Voter Output
				Approximate	Approximate	Approximate	
X	Y	Z	Output (G)	Circuit 1 [f1]	Circuit 2 [f2]	Circuit 3 [f3]	[FATMR output]
0	0	0	1	1	1	0	1
0	0	1	0	0	0	0	0
0	1	0	0	1	0	0	0
0	1	1	0	0	0	0	0
1	0	0	1	1	1	1	1
1	0	1	1	1	0	1	1
1	1	0	1	1	1	1	1
1	1	1	0	0	1	0	0

2. (Un)Protected Input Vectors

The set of all possible input combinations for a given circuit is known as Input vectors. In association with ATMR, protected input vectors are those input vectors of G for which none of the approximate module's output diverges from G. In contrast, the unprotected input vectors are those input vectors of G for which the output of one of the approximate modules diverges from G. For example, in Table 2, 001, 011, 100, and 111 are unprotected vectors. In Table 3, 000,010, 101 and 111 are unprotected vectors, and the remaining input vectors are protected. The existance of an ATMR voter aids in generation of same output as G in the case of unprotected vectors.

With respect to ATMR, the percentage of unprotected input vectors (UIV) can be stated as:

$$UIV = \frac{\text{Number of unprotected input vectors}}{\text{Number of total input vectors}} * 100$$

Consider a three-variable, eight-input vector Boolean circuit with exact output G, as shown in Table 2. The approximations for G are f1 and f2 with three and one unprotected vectors, respectively. The voter generates the ATMR output, where only one of the approximate modules can differ from G at each input vector scenario. Thus, the output of the ATMR method is always going to be correct. The UIV for the ATMR scheme relates to the unprotected vectors of the approximate modules, and it does not relate to the final output of the ATMR. In the example of Table 2, f1 diverges from G at three input vectors, i.e., 011, 100, and 111, while f2 diverges from G at one input vector, i.e., 001. The UIV of G/f1/f2 is given as:

$$UIV = 4/8 * 100 = 50\%$$

The UIV tolerance capability of applications can be quantified as an acceptable UIV threshold. This acceptable UIV threshold can be exploited to evaluate approximate circuits. The value of the UIV is preset [20].

3. Prime Implicant, PI Expansion, and PI Reduction

An implicant is a covering of one or more minterms in a sum of products of a boolean function. A prime implicant is the implicant that cannot be covered by a more general implicant. PI expansion is the process of expanding an original PI cover through 0 to its 1 complement. PI reduction is the process of reducing an original PI cover through 1 to its 0 complement.

D. Contributions

In this thesis, we present an ATMR that targets reduced area overheard and higher fault coverage , with the benefit of protecting the portion of the input space that is most vulnerable to errors and decreasing the search space for ATMR candidates. To the best of our knowledge, this is the first study that combats the critical input scenario for ATMR. The proposed technique sets clear criteria for handling the problem of excavating best possible combination of approximates modules for ATMR and contributes towrds a input-vulnerability aware heuristic approach for successive generation of ATMR modules .

Further on, we present a novel fault-tolerant voter for ATMR. Like conventional majority voters, the proposed voter produces an exact output. Hence, it can also be used in a TMR system.

The contributions of our work are as follows:

- We present a novel ATMR generation methodology where vulnerability of

is considered which benefits interms of search space reduction and higher fault coverage.

- We highlight the reliability of ATMR, in contrast with TMR.
- A compact, low-power, high-speed, and fault-tolerant voter for ATMR is designed.
- A new metric, Quality of Circuit (QoC), is proposed. It holds immense significance to have prior insight of a circuit's intrinsic aptitude towards assurance of fault free output for certain input vectors. Hence, we present a metric that acknowledges the inherent capability of a digital circuit to mask all possible internal faults for a given input vector.
- We present a transistor-level analysis of fault-tolerant voters, considering all possible states of the voter inputs and evaluate them through a comparison with the proposed voter.

E. Thesis Layout

The thesis is organized as follows. In Chapter II, we present the input vulnerability aware approach for generation of ATMR modules and formation of ATMR. Then in chapter III, we discuss the need of fault tolerant voter for ATMR and propose a voter design for ATMR. Next in Chapter IV, we analyze the performance of proposed ATMR generation technique and fault tolerant voter for ATMR. And finally, we conclude the thesis in Chapter V.

II. GENERATION METHODOLOGY FOR ATMR

A. Related Works

The concept of ATMR in [5], [21] utilizes a Boolean factoring algorithm for formulating candidates of approximate circuit modules. Our initially proposed ATMR method is an application of approximate logic synthesis founded upon minterm complements [22] to create a design approach improvement over [5], [21]. Similar to [22], because of error tolerance, numerous minterms of a given logic function can be complemented and a significantly simplified approximate function can be realized. For the selection of minterms, the emphasis is on PI transformations, expansion or reduction of the given logic expression. The authors of [22] presented only expansion of PIs in their algorithm. As the author of [23] pointed out, [22] did not explain the particulars of the algorithm, and though they referred to the effect of PI reduction for abridging logic circuits, their heuristic algorithm did not embrace PI reduction. The work in [22] complemented every minterm individually for two benchmark circuits and settled that PI expansion was more effectual than PI reduction for logic simplification. As in [23], we also deliberate this preliminary experiment inadequate to desert PI reduction. Here, it is essential to declare that though we synthesize approximate logic established upon minterm complements [22]; developing ATMR over minterm complements is not naive and straightforward.

B. ATMR Generation

Let us assume that with the objective of attaining best possible approximate functions and best probable combinations of the approximate functions for TMR, G is the original function. With assumed UIV threshold, consider f1/f2/f3 is the best possible arrangement of approximate functions to develop TMR for G with minimum area overhead. We need verification of the following to assist the proper working of ATMR:

- The minterm complements for generation of f1 hold its original value as G while generation of f2 and f3
- The minterm complements for generation of f2 hold its original value as G while generation of f1 and f3.
- The minterm complements which have been done for generation of f3 hold its original value as G while generation of f1 and f2.

Input Vectors			Output
X	Y	Z	G
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Original Circuit

→

Input Vectors			Output
X	Y	Z	Approximate G (f1)
0	0	0	c
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	c
1	1	0	1
1	1	1	0

Approximate Circuit

Figure 2: Transformation Process: original circuit to ATMR module

The objective of the complements is to recognize minterm or maxterm complements that help in logic minimization as compared to the original module. An illustration of Boolean minimization is given in Fig. 2 where

an original circuit, $G = \bar{y}x + \bar{z}y$, is converted into an approximate circuit, $f1 = \bar{z}$ by introduction of complement ,c, at output for certain input vectors. Analysis by Boolean function, truth table, Karnaugh map, or transistor/gate-level implementation of circuits with a small number of inputs can be carried out manually to identify minterm and/or maxterm candidates that should be complemented. At preliminary level of the work, the fundamental idea is to generate approximate logic circuits successively for modular redundancy by enhancing methodology of [22], built an algorithm which follows the principles of ATMR, and provisions improved design trade-offs. We choose PI expansion as well as reduction to acquire higher logic simplification instead of only PI expansion. PI expansion always decreases the number of literals, and if an expanded PI covers other PIs, these covered PIs can be removed. The smaller the cover, the higher the number of literals required to represent it. If a cover is decreased, the number of literals necessary to present it will always increase as compared to the original cover. Hence, PI reduction is useful, if the cover is entirely removed. PI reduction also aids if the reduced PI is included by another expanded PI. Thus, PI expansion is always effectual in logic optimization, whilst PI reduction is only occasionally effective.

Minterms/maxterms that contribute in the expansion/reduction of an original PI are identified as the minterm/maxterm to expand/reduce the original PI (MEROP). In the primary stage of our preliminary method, all MEROPs that can expand/reduce PIs in minimum original cover are enumerated. Subsequently, identifying PIs that can be expanded/reduced through complementing the consequent single MEROPs, information is stored using procedure Generate MEROP List shown in Algorithm. 1.

Algorithm 1 Generate MEROP List

Input: *G*
// G is the original function

Output: L_{MEROP}
//L_{MEROP} is list of MEROPs

Add all minterms in original function to $\boldsymbol{L_{MEROP}}$
for each *PI in the original minimum cover*
    for each *possible expansion of PI*
        add info about PI to L_{MEROP}

Algorithm 1. contemplates upon the MEROPs that can expand/reduce at the minimum one PI in original minimum cover. The authors of [22] embraced a scheme similar to the Generate MEROP List, however it was restricted to only Minterms Set to Expand the Original PI (MSEOP).

We can consider the complementation of a union of multiple MEROPs so that the cardinality of union of multiple MEROPs is fewer than or equal to allowed number of complements. Through complementing multiple MEROPs, we can find different approximate functions from G, original function as specified by pseudocode in Algorithm 1.

Algorithm 2 represents the approximate function generation pseudocode. The function receives the original function (G), list of MEROPs (LMEROP), and the maximum allowed number of complements (n) as inputs then generates lists of approximate functions with 1, 2, . . . , n complemented vectors from G. The function, too, computes the number of literals for every generated approximate function.

Following acquirement of several candidates for approximate functions, three best approximate functions with minimum number of literals are chosen.

Algorithm 2 Approximate Function Generation

Input: *G, L_{MEROP}, n*
// G is the original function; L_{MEROP} is list of MEROPs; n is the allowed number of complements calculated from acceptable UIV (stated in the preliminaries section)

Output: *L^1, L^2, ..., L^n*
// L^1, L^2, ..., L^n are list of complemented original function and corresponding number of literals for each function with 1, 2, ..., n unprotected vectors respectively

for i = 1 to n
L^i = considering all combinations select i MEROP form L_{MEROP} list and complement them in G
Calculate number of literals for all generated functions in L^i

Definition of Block State: A block state denotes those input vectors whose output can no more be complemented and should essentially hold the similar value as original function.

To satisfy the working principle of ATMR, amongst three functions, any particular MEROP can be complemented just one time. The principle of ATMR [5], [21] is valid whilst only one of the approximate modules disagrees from the original function of each input vector situation. If two or three of same MEROPs are complemented in two or three approximate function, voter will vote for those two or three complemented outputs, then this breaches the principle of ATMR. Thus, to select three approximate functions, it is essential to contemplate upon the blocking properties.

The pseudocode for picking the three best approximate functions is specified in Algorithm 3. This program receives the original function, G, maximum allowed number of the complement vectors, n, lists of generated approximate functions with 1, 2, . . . , n number of unprotected (complemented) vectors, and L1, L2, . . . , Ln as inputs, and outputs three best approximate functions with total

number of literals. In an iterative routine, the generated approximate functions are utilized in a nested fashion to attain the three best approximate functions. The three nested for-loops generate all possibilities of selecting three functions with the different numbers of unprotected vectors. The total number of the literals of each of three selected functions are computed for each possibility, then the functions having minimum number of literals are chosen with respect to blocking property. If total number of the literals in f1/f2/f3 are fewer than total number of the literals in G/G/G (# Lit), then f1min, f2min, and f3min are revised as the newly generated f1, f2, and f3 respectively, and # Lit is the set of total number of literals in f1/f2/f3 for subsequent iteration.

Algorithm 3 Pseudocode: Approximate Function Selection

Input: *G, n, L^1, L^2, ..., L^n*
// G: Original function; n: Allowed complements calculated from acceptable UIV (stated in the preliminaries section); L^i: List of complemented original functions with i unprotected vectors

Output: f_{1min} ,f_{2min} ,f_{3min} , *#Lit*
// f_{1min} ,f_{2min} ,f_{3min} are selected approximate functions to form ATMR; #Lit is total number of literals in f_{1min} ,f_{2min} ,f_{3min}

Initialize: $f_{1min}=f_{2min}=f_{3min}=G$
*#Lit = 3 * number of literals in G*
Select *n1 = 1 to n*
Select *n2 = 0 to n_1*
if *(n1 + n2) <= n*
Select *n3 = 0 to n2*
if *(n1 + n2 + n3) <=n*
Select f1, f2, f3 *from L^{n1}, L^{n2}, L^{n3} with minimum literals considering blocking property (Intersection of all locations to be complemented in f1, f2, and f3 should be ∅)*
If *(total number of literals in f_1 /f_2/ f_3) < #Lit*
$f_{1\,min} = f_1$
$f_{2\,min} = f_2$
$f_{3\,min} = f_3$
#Lit = total number of literals in f_1 / f_2 /.f_3

The initial technique includes the succeeding phases:

- MEROP Listing Phase: The MEROP list is generated by using Generate-list (G)
- Approximate Function Generation: The MEROPs are selected from MEROP list relating to allowed number of complements for every approximate function. Once selection, the MEROPs are complemented to obtain the approximate functions.
- Approximate Function Selection: When the approximate function is generated, three best (minimum literals) approximate functions are selected considering the maximum number of the allowed unprotected vectors and blocking property to satisfy the working principle of ATMR.

To exhibit the flow of initial procedure, consider the following example: Example: Consider a four-variable Boolean circuit represented through the K-map in Fig. 3(a):

$$G = a\bar{c}d + \bar{a}\bar{b}(\bar{d} + \bar{c})$$

G comprises of seven literals. The total number of the input vectors is 16. Here, area overhead for TMR can be assessed through number of literals in G. Consequently, for TMR, G/G/G, number of literals is 21. Now, consider an ATMR design having acceptable UIV threshold as 18.75%. The total number of allowed complements or unprotected vectors can be estimated as: Here, it is imperative to state that it is not essential to obtain the best approximate functions by realizing the exact UIV threshold value or with exact total number of the allowed complements. With our technique, depending upon application, one can

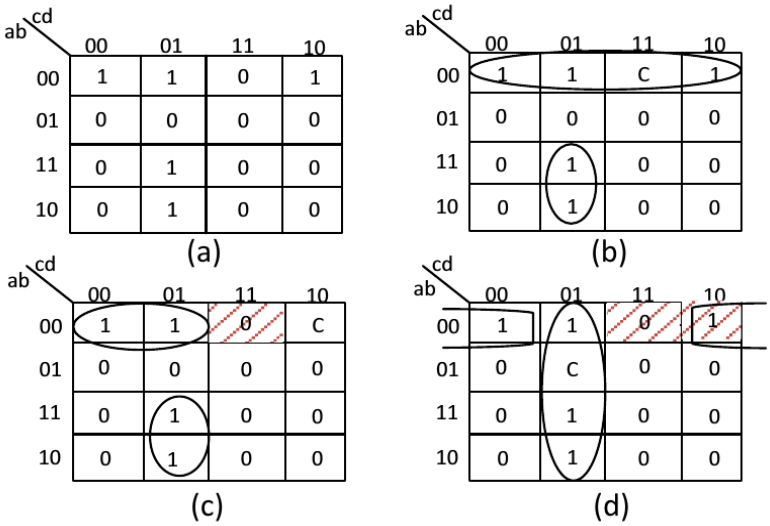


Figure 3: Example: (a) Original function, (b) Approximate function 1, (c) Approximate function 2, (d) Approximate function 3

find the best approximate functions by remaining within bounds of the threshold. Hence, ATMR fault coverage is improved, and the area overhead is abridged.

Approximation 1: First approximate module for ATMR can be attained by complementing $\bar{a}\bar{b}cd$ of G to attain largest possible cover. The approximate function f1 has five literals and is shown in the Fig. 3 (b):

$$f1 = a\bar{c}d + \bar{a}\bar{b}$$

To preserve the validity of blocking principle, $\bar{a}\bar{b}cd$ of G will be in a blocked state for computation of second approximate module.

Approximation 2: For second approximate module $\bar{a}bcd$, of G is complemented to produce the following as shown in Fig. 3 (c):

$$f2 = \bar{c}(ad + \bar{a}\bar{b})$$

Approximation 3: Now, $\bar{a}\bar{b}cd$ and $\bar{a}bcd$ of G will be in blocked state for obtaining the third approximate module as shown in Fig. 3(d). The extraction is carried out by complementing $\bar{a}b\bar{c}d$ of G:

$$f3 = \bar{a}\bar{b}\bar{d} + \bar{c}d$$

ATMR: Ultimately, our ATMR will be comprised of f1/f2/f3, and number of literals in f1/f2/f3 is: f1/f2/f3 = 5+5+5 = 15 literals

Therefore, the literals (area overhead) reduced from 21 to 15. As indicated before, "ATMR-An Exhaustive Approach" method warranties finding best candidates for the approximate functions to develop the Approximate TMR; however, by accumulating the number of MEROPs and unprotected vectors, search space surges exponentially. The relationship between number of MEROPs and number of unprotected vectors within the search space can be expressed as a binomial coefficient. If number of MEROPs is designated as n and number of unprotected vectors is k, search space can be computed by:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

Fig. 4 shows exhaustive approach considering unprotected input vectors in this primarily proposed method. The search space shows exponential growth as number of unprotected input vectors escalates. As shown in Fig. 4, search space is not big for up to three unprotected vectors. Though, for functions with more input vectors (more MEROPs) it cultivates exponentially. To overcome this inadequacy, we introduce our heuristic technique.

When we deal with circuits higher numbers of inputs and essentially work with numbers of unprotected input vectors greater than two, our heuristic

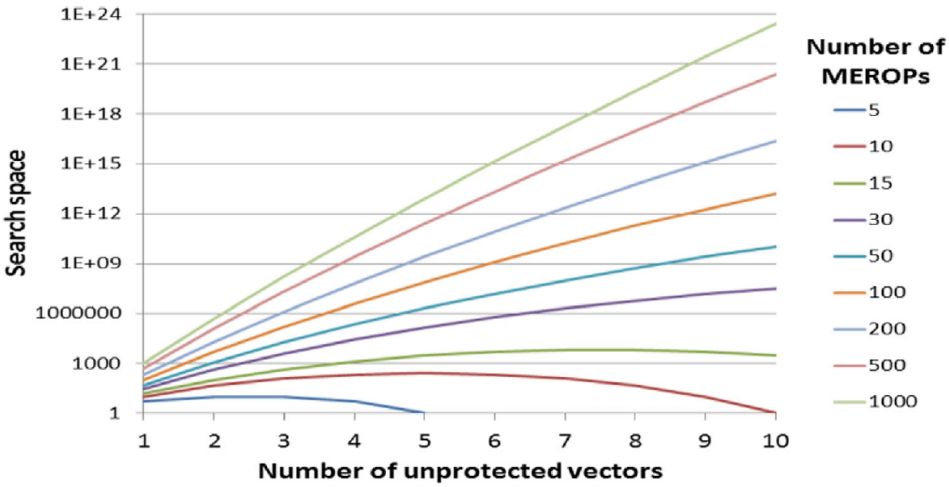


Figure 4: ATMR - An Exhaustive Approach: an exponential growth trend reflected by the number of unprotected input vectors versus the search space

technique uses only portion of the MEROPs.

C. Input Vulnerability Aware ATMR

It is critical to note that ATMR is vulnerable to errors, the approach for ATMR extraction must guarantee that vulnerability of inputs is taken care of. Vulnerability exists at diverse levels in design hierarchy. It is probability that a fault happening in that level demonstrates an error at its output. Assessment of which is tremendously expensive owing to the large number of inputs and the fault combinations [24]. Automatic test pattern generation is a method to algorithmically recognize input vectors that show the known inherent fault conditions in the circuit. ATPG algorithms intends to achieve maximum test quality at the minimum test cost, which comprises Of the test data volume, the time vital for testing, the design-for-testability hardware, and the power expended during test [25]. Algorithms for ATPG emphasis primarily on ways to produce

tests for the combinational circuits. Tests for these circuits include choices from many possibilities, consequently classical computer search methods are typically the basis for the algorithm. These search methods use information in network topology to specify inputs that form the test pattern. The methods have become more efficient over time. Three of the best-known algorithms for combinational ATPG are D algorithm; Podem, short for pathoriented decision making; and Fan. These algorithms diverge in the size of search space, the global search policy, and the heuristics that can be utilized to help guide that search. The size of search space is extremely significant, since it determines an upper bound on search time. The order of searching also momentously affects search time. Subsequently, both the global search approach used to build the search graph and the heuristics used to guide the search are just as vital as the size of search space [26].

Our proposed heuristic approach for input vulnerability aware ATMR includes two stages: 1) pre-blocking stage (PBS), it ensures that ATMR is aware of input vulnerability and as there are lots of candidates for blocking, ATPG aids to recognize the right candidate leading to decrease in design space and 2) approximate modules extraction and selection stage (AMES), it provides a good-enough combination of the approximate modules for ATMR having higher fault coverage and minimum area overhead.

Pre-Blocking Stage (PBS): Notice that the entire input space does not hold equal significance as some portion of the input space is more vulnerable to errors than the rest. ATMR formation is a sensitive case for the critical input scenarios. The aim of introducing TMR may be defeated, particularly for a critical input situation or safety-critical applications, which is not tolerable. Throughout the process of developing the approximate functions for ATMR, we will mainly focus on portions of the input space which are not vulnerable to errors and it

is warranted that most vulnerable to error input spaces are preserved. Thus, the concept of pre-blocking is initiated in this study. Pre-blocking denotes to those input vectors whose output is considered vulnerable to errors and it will be made unavailable for complement in the entire process of ATMR candidate generation. Pre-blocking ensures that process of ATMR formation is not oblivious towards critical input scenario. A critical input space can be acquired through ATPG tools. The primitive method of ATPG used D algorithm, which attempts to propagate stuck at fault D(1) or D'(0) to primary output. An improvement over D-algorithm came in form of path-oriented decision making (PODEM), that was further improved utilizing the fan-out-oriented algorithm (FAN). The FAN algorithm restricts atgp search space to decrease its computation time [26]. ATLANTA is a publicly accessible ATPG tool based on FAN. In our ATMR technique, modified Atalanta [27] is used. By means of the modified ATALANTA [27] test generation system, we produced all possible test patterns for every fault to recognize the most susceptible areas of circuit, as shown in Fig. 5.

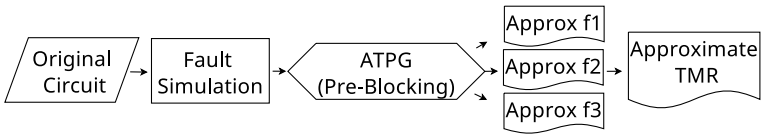


Figure 5: Flowchart of the proposed approximate TMR approach.

The tool produces test vectors/input space which are/is prone to error after doing fault simulations on all nodes in the circuit. We exploited these test vectors as a heuristic. To decrease the candidate (complement locations for ATMR) search space, output of produced test vectors is pre-blocked, as demonstrated in Fig. 5, and cannot be complemented for producing approximate modules of ATMR. By reduced candidate search space and vulnerable to error aware input

method, good-enough approximate functions are obtained to form an ATMR with higher fault coverage.

Approximate Modules Extraction and Selection Stage (AMES): Formation of ATMR with our method is explained in Fig. 6 through an example. Through ATLANTA, it is settled that input vectors 0000 and 0001 are vulnerable to errors, and therefore, are pre-blocked for approximation. Original function (G) is transformed to approximate modules (*fn*) by complementing min/maxterms [18]. A prime implicant (PI) is an implicant which cannot be covered by more general implicant. We choose for both PI expansion (0 to 1) and reduction (1 to 0) complements, as shown in Fig. 6. Here, $G = \bar{c}(\bar{b} + ad)$. *f1* is obtained by complementing $ab\bar{c}\bar{d}$ for largest possible cover. $f1 = \bar{c}(a + \bar{b})$ and has three literals. State $ab\bar{c}\bar{d}$ remains blocked (same as the original function) for further complements to develop other modules. This ensures that three modules provide correct output after voter circuitry. For *f2* and *f3*, $ab\bar{c}d$ and $\bar{a}b\bar{c}d$ are complemented, respectively, to develop $f2 = \bar{c}\bar{b}, f3 = \bar{c}(d + \bar{b})$, while $ab\bar{c}d$ and $ab\bar{c}\bar{d}$ stays unavailable to complement for f3 realization. Hence, area overhead decreases as the literals reduces from 12(TMR) to 8(ATMR). Input vectors corresponding to $ab\bar{c}\bar{d}$, $ab\bar{c}d, and\ \bar{a}b\bar{c}d$, i.e., 1100, 1101, and 0101, are called unprotected vectors.

Minterms/maxterms, which help in the expansion/reduction of an original PI, are called as minterm/maxterm to expand/reduce original PI (MEROP). In the case of larger circuits, after procedure of pre-blocking, primarily, the number of allowed complements is set to one, and every candidate in the MEROP list is complemented one at a time to produce a list of probable approximate functions with a single unprotected vector for given original function. After the list is attained, approximate functions are minimalized. Once an approximate

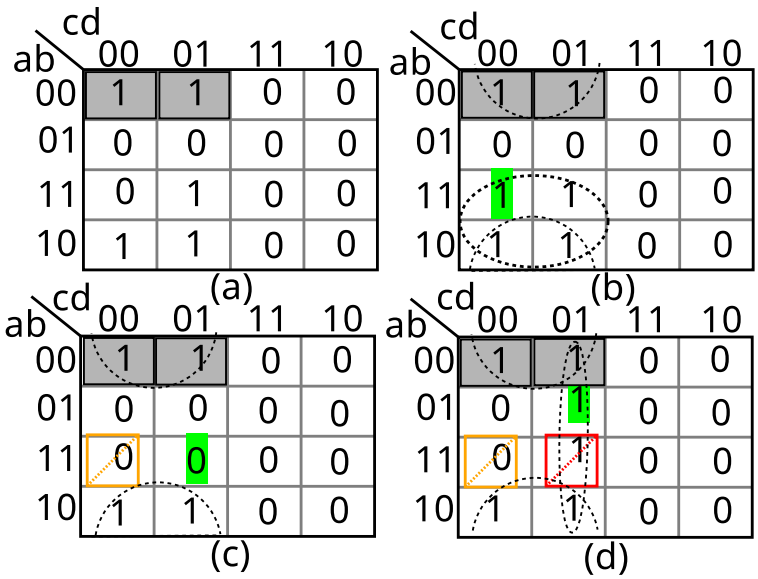


Figure 6: Methodology to form ATMR: (a) Original function, (b) Approximate f1, (c) Approximate f2, (d) Approximate f3.

function is presented in its minimized form, it is linked with its respective number of the literals and its complemented MEROP location. Afterwards, the list of probable approximate functions is sorted established on the number of literals. For a single complement, best approximate functions having least number of literals for ATMR are accessible at the top of the list. From sorted list of the approximate functions with single complements, portion of list is chosen such that there is no violation of capacity. The rest of the functions are discarded as they always have higher weights, i.e., a greater number of literals. Thenceforth, every member of this freshly generated list is again introduced with complement. The whole process is then repeated until we get list with the allowed number of complements. Afterwards the final list is created, the focal point is the location of the complemented MEROP and the number of literals, where the combination of three approximate functions to form ATMR is shortlisted, such that the

locations of complemented MEROPs of every approximate function do not match (blocking property) and total number of the literals is at a minimum. As [28] stated, two-level minimization tools, such as Espresso, are not scalable to circuits with more than 15−20 inputs. Hence, we utilized clustering algorithm in the ABC tool [29] for the circuits with more than 15 inputs.

III. FAULT TOLERANT VOTER FOR ATMR

A. TMR, Voter, and Reliability

In a TMR system, the original module is replicated three times, and the output of each module is fed into a voter. The voter produces the final output of the system based on the majority votes.

In case of a TMR system, if a single fault occurs that flips one of the voter inputs, the output of the voter will still be correct because the other inputs will agree with each other and produce the correct output. When two inputs of the voter are 0, a faulty input will be 1. However, since 0 holds the majority as input, the voter will output a 0 and mask the fault . Under such a scenario, the possible faulty input combinations are as follows: 001, 010, 100. Similarly, when two inputs of the voter are 1, a faulty input will be 0. In this case, the possible faulty input combinations are 011, 110, 101. Thus, the faults will be masked for faulty input conditions, i.e., 001, 010, 100, 011, 110, 101. Figure 7 illustrates the TMR phenomenon, where 000 and 111 are fault-free states. It would be safe to state that faults occurring within the voter circuit that propagates to the output are fatal to TMR.

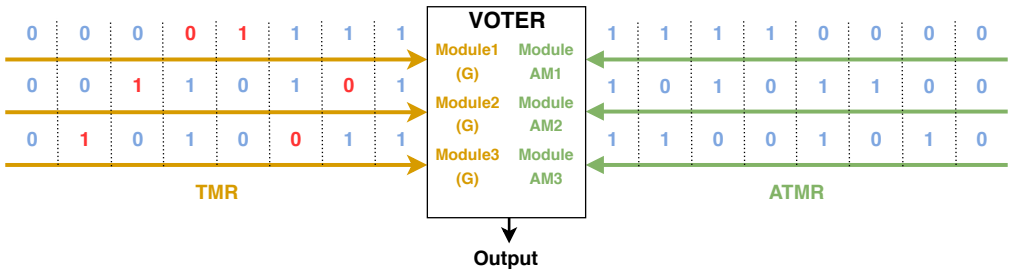


Figure 7: Input Vectors for Voter.

Reliability is the probability of no failure occurring within a given period

of operation. The non-faulty state of a function module is represented as R_M, where R_M is the probability that a module will generate a correct output and is an exponential function of time with a constant failure rate λ [9], [19]. Mathematically, R_M can be defined as $R_M = e^{-\lambda t}$. Consider the simplest system reliability model for a system with n modules. If the module reliability is R_M with a constant-failure rate λ, the system reliability, $Rsys$, is then given by [1]

$$Rsys(t) = [R_M(t)]^n = [e^{-\lambda t}]^n \tag{1}$$

As in case of TMR, all digital modules are independent and identical, and the reliability of TMR can be then determined as a function of the reliability R_M of a single module, assuming that the voting circuitry does not fail. Furthermore, the systems will function properly if any two modules operate appropriately. The reliability of TMR can be given as follows:

$$\begin{aligned} R_T = {} & \text{Probability of all three modules functioning +} \\ & \text{Probability of any two modules functioning} \end{aligned} \tag{2}$$

$$R_T = B(3:3) + B(2:3) \tag{3}$$

$$= \binom{3}{3} R_M^3 (1-R_M)^0 + \binom{3}{2} R_M^2 (1-R_M)^1 \tag{4}$$

$$= 3R_M^2 - 2R_M^3 \tag{5}$$

$$= 3e^{-2\lambda t} - 2e^{-3\lambda t} \tag{6}$$

Considering the reliability of a voter R_v, then

$$R_{TMR} = R_{voter} R_T \tag{7}$$

$$= R_{voter}(3e^{-2\lambda t} - 2e^{-3\lambda t}) \tag{8}$$

This shows that the failure of a voter will simply mean the failure of TMR even if all modules are functioning correctly.

B. ATMR, Voter, and Reliability

ATMR can be developed using approximate modules for all three modules of TMR . The output of the approximate modules differs from the original module for some of the input vectors, which helps in complexity reduction. The working principle for ATMR is as follows [5]:

"*Only one of the approximate modules can differ from the original circuit at each input vector scenario, allowing the majority voter to still select two match outputs out of three for any input vectors.*"

Consider an original module G with inputs I, J, and K. The truth table of G is given in Table 4. An approximate module (AM) of G, AM1, is produced by complementing the output of G for $\bar{I}J\bar{K}$. Another approximate module of G, AM2, is produced by complementing the output of G for $I\bar{J}K$ and IJK. The third approximate module of G, AM3, is produced by complementing the output of G for $\bar{I}\bar{J}\bar{K}$. It can be seen from Table 4 that AMA1 differs from G at $\bar{I}J\bar{K}$ and produces a value of 1. Similarly, AM2 differs from G at $I\bar{J}K$ and IJK, whereas AM3 differs from G at $\bar{I}\bar{J}\bar{K}$. An ATMR is formed by AM1, AM2, and AM3 where these approximate modules provide input to the voter that produces the final

output. It can be seen from the example that the voter output is the same as that of G because the validity of the working principle of ATMR is maintained despite approximate modules being used to form the TMR. The vectors corresponding to $\bar{I}\bar{J}\bar{K}$, $\bar{I}J\bar{K}$, $I\bar{J}K$, and IJK, i.e., $000, 010, 101$, and 111, are called unprotected vectors.

Table 4: Approximate TMR: An Example.

Input Vectors			Original Circuit	Approximate TMR			Voter Output
				Approximate	Approximate	Approximate	
I	J	K	Output (G)	Module 1 (AM1=A)	Module 2 (AM2=B)	Module 3 (AM3=C)	[ATMR output]
0	0	0	1	1	1	0	1
0	0	1	0	0	0	0	0
0	1	0	0	1	0	0	0
0	1	1	0	0	0	0	0
1	0	0	1	1	1	1	1
1	0	1	1	1	0	1	1
1	1	0	1	1	1	1	1
1	1	1	0	0	1	0	0

Assume that the three modules of ATMR are fault-free. Fault-free conditions for ATMR, as depicted in Figure 7, are $000, 001, 010, 100, 011, 101, 110, 111$. This is because ATMR permits one of the approximate modules to differ from the original circuit at each input vector scenario. Thus, there will necessarily be cases in which two inputs of the voter will be the same, and one input will be different from them. Under such a scenario, if a fault occurs at the voter input it could be unfavorable for the ATMR system. Please note that the suitability of ATMR is application dependent as the application or design engineer can distinguish between critical and non-critical parts of a system and sensibly adapt ATMR. A TMR voter does not face this situation because this condition primarily exists owing to the working principle of the ATMR. Previously proposed fault tolerant

voters [1], [9], [19] will still produce an erroneous output for unprotected vectors under such conditions because this is out of the scope for the TMR voter design. For example, in Table 4, when input vector $\bar{I}\bar{J}\bar{K}$ is considered, the original circuit should produce a value of 1 at the output. For TMR, if one of the inputs fed to voter is flipped, the remaining two inputs will cover for it to produce a fault-free output of 1. For the ATMR in Table 4, $AM1$ and $AM2$ vote to propagate a value of 1 to the output despite $AM3$ producing a value of 0. If a fault occurs at the input of the voter fed from AM1, 1 is flipped to 0, and AM1 and AM3 will vote a faulty output of 0.

For ATMR, all digital modules are independent. Furthermore, the systems will only function properly if all modules work appropriately. The reliability of the ATMR (R_{AT}) and ATMR with voter (R_{ATMR}) can be respectively given as follows:

$$R_{AT} = \text{Probability of all three modules functioning} \tag{9}$$

$$R_{AT} = \binom{3}{3} R_M^3 (1 - R_M)^0 \tag{10}$$

$$= R_M^3 = e^{-3\lambda t} \tag{11}$$

$$R_{ATMR} = R_{voter} R_{AT} \tag{12}$$

$$= R_{voter}(e^{-3\lambda t}) \tag{13}$$

It is quite evident from (8) and (13) that the reliability of ATMR is much lower compared to that of TMR. It is of immense significance to have reliable voters and reliable ATMR modules. Hence, it is imperative to design a fault tolerant voter for ATMR which should not only be able to mask faults that occur inside the voter circuit but also, the voter input needs to be fault free.

C. Proposed Voter for ATMR

ATMR is meant to overcome the area overhead issue of TMR. Hence, any application that opts to adopt ATMR must be sensitive to the area consumption. Thus, as a pre-requisite, a voter for ATMR should be designed such that it occupies the minimum area. This design aspect is one of our forte.

With respect to fault tolerance, as discussed earlier, a voter for ATMR should be tolerant toward (a) faults occurring within the voter circuit nodes (internal outputs) and (b) the faults flipping the inputs of the voter.

To limit faults occurring at the circuit nodes, a circuit with the minimum nodes is endorsed. According to [19], a voter with less transistors, and hence less nodes, has a lower probability of having an internal fault. The work concluded that a voter with less transistors has a higher probability that the TMR system will operate correctly. In a similar fashion, we propose a fault-tolerant voter for ATMR by using pass transistor logic (PTL) to keep the number of transistors to a minimum. In PTL, the primary inputs drive the gate and source/drain terminals, which leads to the use of a reduced number of transistors for logic implementation. The use of a pass transistor helps in decreasing the possible faulty nodes within the circuit. Please note that only single node faults are considered, which implies that only one internal node of the entire circuit is faulty at a given time.

Faults flipping the inputs of the voter behave such that an input stuck-at-1 (s-a-1) fault of the gate terminal of an NMOS transistor causes a closing of the path between the drain and its source of it. An input stuck-at-1 (s-a-1) fault to the gate terminal of a PMOS transistor causes in opening of the path between the drain and its source. An input stuck-at-0 (s-a-0) fault to an NMOS transistor

results in an opening of the path between the drain and source. An input stuck-at-0 (s-a-0) fault at a PMOS transistor results in a closed path between the drain and source. References [30], [31] proposed the use of Quadded Transistor redundancy and Triple Transistor redundancy, respectively, for the provisioning of input fault tolerance. For the proposed voter faults, flipping of the inputs of the voter is handled using the redundancy technique of [30]. As the number of transistors is kept to a minimum using pass transistors, unlike in previous studies, this provides us with the freedom to introduce transistor-level redundancy for the voter inputs. However, the use of pass transistors brings about certain design constraints and not every transistor can be made redundant. This is because NMOS produces a weak 1 and PMOS produces a weak 0. Hence, the outputs do not have a full voltage swing (low noise margin), and additional drivers may be needed.

For design simplicity, redundancy for all transistors with inputs A and B is introduced. It is ensured that any single transistor fault will not change the logic behavior and the fault will be tolerated. The redundancy is provided though quadded transistors, which has higher reliability than triple redundancy [31]. In a quadded redundancy technique, a transistor with gate input A is replaced with a four-transistor structure, as shown in Figure 8. Two transistors are connected in parallel $(A+A)$ with gate input A. Two such structures are then connected in series $(A+A)(A+A)$. This configuration of quad transistor is opted because each quad structure has an effective resistance equal to that of a single transistor [30]. Hence, a quadded structure implements the logic function $(A+A)(A+A)$ and $(B+B)(B+B)$ for A and B inputs, respectively, in the proposed voter, as shown in Figure 9e. No single transistor fault will change the logic behavior, and the fault will be tolerated. Furthermore, double s-a-0 (NMOS)/s-a-1 (PMOS) are tolerated if they do not occur in any two parallel transistors. Double s-a-

1 (NMOS)/s-a-0 (PMOS) are tolerated if they do not occur in any two series transistors. In addition, any triple fault that does not include two parallel s-a-0 (NMOS)/s-a-1 (PMOS) transistors or two series s-a-1 (NMOS)/s-a-0 (PMOS) transistors is tolerated [30].

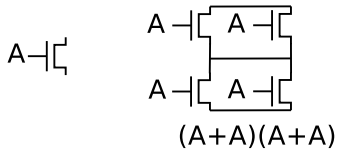
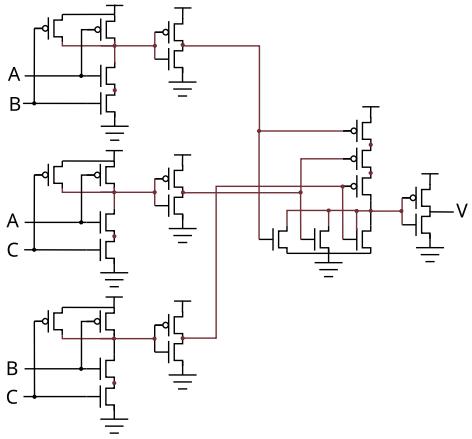
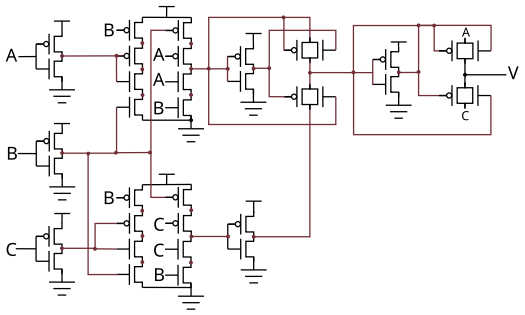


Figure 8: Quadded Redundancy: $(A+A)(A+A)$ [30].

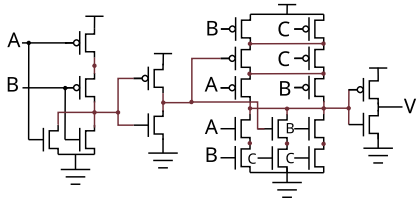
The proposed voter design is presented in Figure 9e. The main design transistors are Tp1, Tp5, Tn3, Tn7, Tp9, and Tn9. Tp1, gated by A, is made redundant through Tp2, Tp3, and Tp4. Similarly, Tp5 with B as an input has a redundancy provision through Tp6, Tp7, and Tp8. Tn3 has a gate input of B and source input of A. This transistor is made redundant through Tn1, Tn2, and Tn4. Similarly, Tn7 has redundancy through Tn5, Tn6, and Tn8. The output is produced through a mux. Because the gate inputs of mux are driven by 0/1 produced through the pass transistors, owing to the design constraints, redundancy is not provided here. All transistors, except for two at the output stage, are made redundant. This input level redundancy facilitates an ATMR structure because the voter inputs of ATMR can differ from each other under a fault-free condition, and an input fault will simply lead to a faulty output. For instance, consider the case of ATMR in Table 4, where the input vector is $\bar{I}J\bar{K} = 010$. Now, the voter inputs are $A = AM1 = 1$ and $B = C = AM2 = AM3 = 0$. In this case, the fault-free voter should output a 0. Assuming that B at Tp5 is stuck at 1 and that the transistor behaves like an open switch, in this case Tp6, Tp7, and Tp8 will work as a closed switch to ensure a fault-free final output.



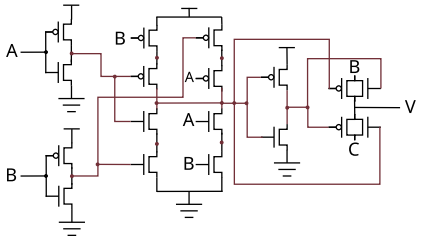
(a) Classical.



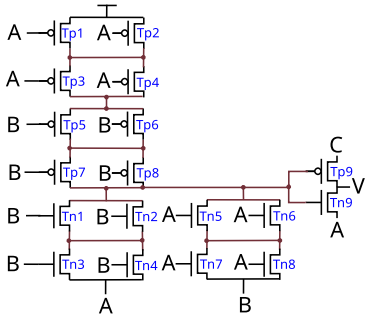
(b) Kshirsagar [1].



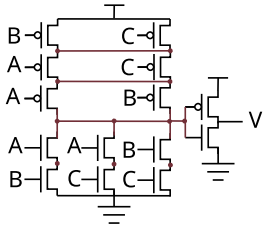
(c) Bala [9].



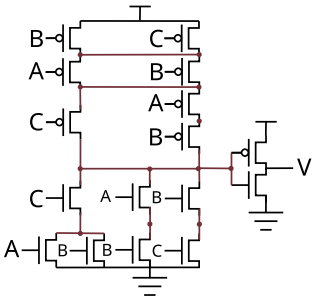
(d) Ban [19].



(e) Proposed Voter.



(f) Classical_CMOS.



(g) Bala_CMOS.

Figure 9: Voters Architectures.

IV. PERFORMANCE EVALUATION

A. Performance of Input-Vulnerability Aware ATMR

Fault coverage is generally defined by the number of faults detected by the number of potential faults. For evaluating the fault coverage ability of the proposed method, comparative analysis is carried out using a case study of [5]. HOPE [27] was employed for fault simulations where all input patterns were applied to the circuit with faults injected at each gate. The circuit was mapped using a state-of-the-art logic synthesis tool [29] for transistor count using an academic library. Fig. 10 shows comparative results on the basis of the number of transistors, (un)detected faults, and fault coverage for the case study of [5]. The proposed method provides better fault coverage and overhead with the same number of unprotected vectors.

For circuits with primary inputs greater than 14, the following two steps were followed: 1) clustering gates [29] in the given logic circuit into Boolean functions of $10-14$ inputs and 2) simplification of the technology-independent nodes to reduce overheads while maintaining a good protection ratio. Pre-blocking and MEROP are the starting points of AMES, permitting to make a locally optimal choice at each stage with the hope of finding a global optimum at the final stage.

Through four benchmarks, Fig. 11 shows the significance of pre-blocking by drawing comparison between search space with pre-blocking and without pre-blocking for the proposed ATMR method. The number of complements are represented on x-axis and y-axis represents the size of the candidate list and area overhead. As the pre-blocking stage is dependent on ATLANTA, it proves to be computationally efficient, and because pre-blocking restricts the number of available MEROPS for ATMR, it turns out to be an effective heuristic.

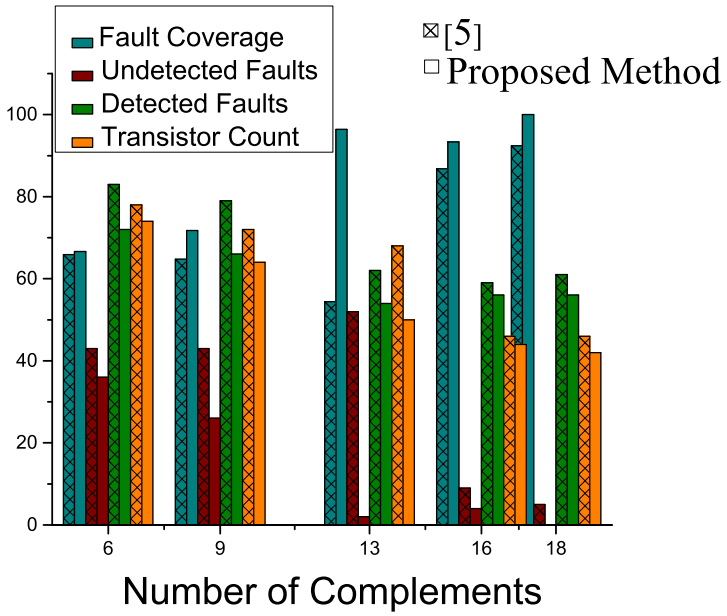


Figure 10: Comparative results

The strength of this methodology is that, usually, the same area overheads and protected vectors are available with less processing. For example, in the case of i1, only 4987 candidates need to be processed instead of 74052 while attaining similar overhead ATMRs (200 literals with 15 unprotected vectors).

B. Analysis of Fault-Tolerant Voters

1. Simulation Setup

The fault masking capability of proposed voter for ATMR is evaluated by testing the design for stuck at faults. The voter is simulated with single fault using ModelSim through Verilog. In Verilog, a Single Event Transient (SET) is emulated by forcing a change in the state of the node by means of the force/release command. This simulated SET pulse is only logical, and therefore

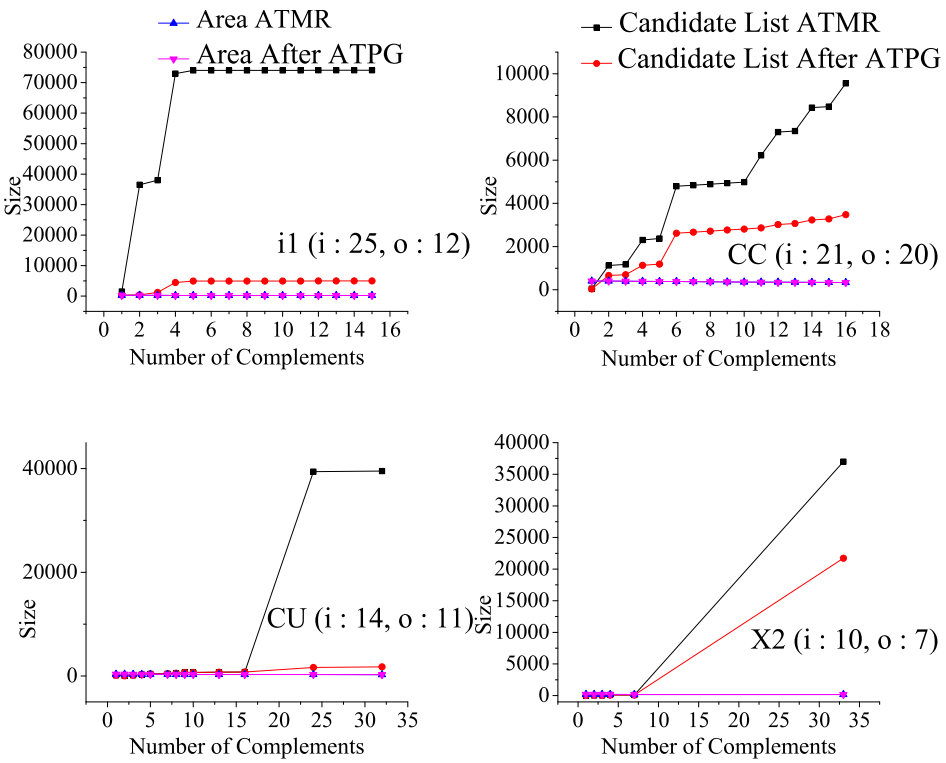


Figure 11: Candidate search space reduction

the sizing the electrical and temporal masking is not evaluated [5]. Thus, to test the voter circuit, force/release is exhaustively used at each node. The simulation setup, as shown in Figure 12, is similar to that of ATALANTA [27], which enumerates those input vectors where a fault propagates to the output at the gate-level. In our case, as shown in Figure 12, the design is assessed at the transistor level. Each node, one at a time, is introduced with a fault and all possible input vectors are applied. Finally, those input vectors for which a fault propagates to the output are generated. The set of these input vectors is referred to as vulnerable input vectors (VIV). The set of input vectors for which a fault occurring inside the circuit nodes will not propagate to the output is referred to as invulnerable

input vectors (IIV).

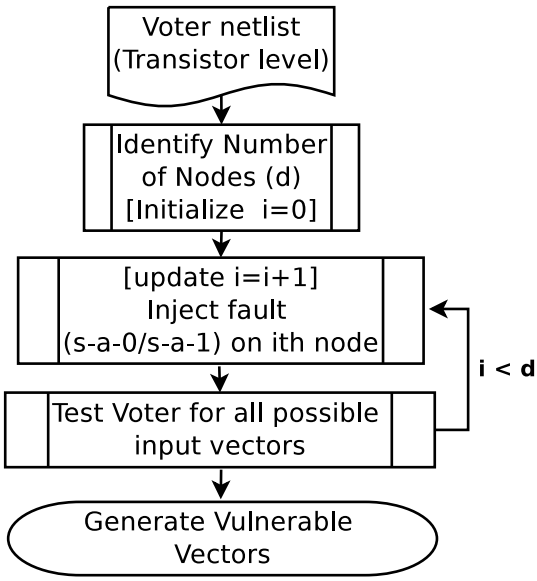


Figure 12: Flow Chart for acquiring vulnerable vectors.

For a comparative analysis, the following voters are considered: (a) Classical (b) Kshrisagar [1] (c) Bala [9] (d) Ban [19] (e) Proposed Voter (f) Classical_CMOS, and (g) Bala_CMOS. A Classical voter has a multistage architecture and is the most conventional voter design implementation using three two-input AND gates and one three-input OR gate , as shown in Figure 9a. The CMOS implementation of a classical voter using complex logic gates leads to a smaller number of transistors in the design, as presented in Figure 9f. The CMOS version of Classical voter is implemented and studied by references [2], [32]. The Kshirsagar voter is a fault-tolerant design with 36 transistors based on a priority encoder and multiplexer, as illustrated in Figure 9b. The Kshirsagar voter uses transmission gates. The Ban voter shown in Figure 9d is a simplification of a Kshirsagar voter through the exclusion of a priority encoder and a reduction of

the XOR logic gates to a single gate . It uses a total 18 transistors. The Bala voter, shown in Figure 9c, involves a 2-input OR gate and a complex gate to achieve a more robust design than the Classical, Ban, and Kshirsagar voters. Figure 9g shows the CMOS implementation of a Bala voter using single complex logic [2].

The voters are tested for stuck-at-faults and metrics such as Fault Masking Ratio (FMR) and Vulnerable Input Vectors (VIV). FMR quantifies the intrinsic fault tolerance of a logic circuit. As discussed earlier, some of the internal faults do not propagate to the output owing to the masking capability of the voter circuit. FMR is defined as the ratio of total number of correct voter output states in the presence of internal faults, which are masked, divided by the total number of potential internal fault occurrences [6], [9].

$$FMR = \frac{k}{2^{n+m} - 2^n} \tag{14}$$

where n, m, and k are the primary inputs, internal nodes, and internal faulty combinations for which a fault is masked at the output, respectively. Here, 2^n represents the non-faulty combinations of nodes, whereas 2^{n+m} specifies the total number of faulty and non-faulty combinations [6].

To investigate the electrical properties of voter circuits, the designs were mapped using a 45 nm library [33] in Cadence Virtuoso. All the designs were simulated under the same conditions. The following parameters were used for the purpose of simulation: W = 90 nm, L = 50 nm, and Voltage = 1 V. The designs were examined in terms of the number of transistors, delay, power, and power-delay-product (PDP). Please note that all the designs were tested using all possible test vectors to determine the worst-case delay.

2. Analysis of Simulation Results

Table 5 shows the fault making ability of the voters on basis of the FMR and VIV. A high FMR and minimum length of VIV is desired. High FMR values of 0.720, 0.645, and 0.625 are reflected by the Kshirsagar [1], proposed, and Ban [19] voters. The proposed voter delivered upto 45.1% improvement in FMR. For s-a-0 and s-a-1, the proposed, Classical_CMOS, and Bala_CMOS voters are vulnerable under fewer input vectors. In the case of s-a-0, out of eight possible input vectors, the proposed, Classical_CMOS, and Bala_CMOS voters have 3, 4, and 5 vulnerable input vectors, respectively. In the case of s-a-1, out of eight possible input vectors, the proposed, Classical_CMOS, and Bala_CMOS voters have 3, 4, and 4 vulnerable input vectors, respectively. It can be observed that the proposed design has the maximum number of invulnerable vectors for both stuck-at-faults. This is a very important aspect because it ensures that a stuck-at-fault will not propagate to the output for five out of the eight vectors, making the proposed voter highly reliable. Here, it can be noticed that FMR fails to highlight a circuit's capability of fully masking an input vector from a fault. The capability of fully masking an input vector reflects the guarantee of a circuit that, for the given input vector, the output will always be fault free. Hence, we propose a new metric, the QoC, based on the VIV. The QoC quantifies the aptitude of a circuit for fully masking an input vector, and can be given as follows:

$$QoC = 1 - \frac{VIV_T}{2^n} \tag{15}$$

where VIV_T is the total number of vectors in the VIV, and n is the number of circuit inputs.

Table 5: Internal Fault-Tolerance Capabilities of the Voters.

Design	Total Nodes	FMR	Vulnerable Input Vectors				QoC	
			Stuck-at-0		Stuck-at-1		Stuck-at-0	Stuck-at-1
Classical	12	0.427	$[0--,1-0,101]$	7/8	$[0---]$	8/8	0.125	0
Kshirsagar [1]	17	0.720	$[---]$	8/8	$[---]$	8/8	0	0
Bala [9]	9	0.472	$[00-,10-,010]$	5/8	$[11-,0-1,101]$	5/8	0.375	0.375
Ban [19]	8	0.625	$[-10,-01,011,100]$	6/8	$[-00,11-,011,001]$	6/8	0.25	0.25
Proposed Voter	6	0.645	$[001,110,111]$	3/8	$[011,100,101]$	3/8	0.625	0.625
Classical_CMOS	6	0.354	$[-00,001,010]$	4/8	$[11-,101,011]$	4/8	0.5	0.5
Bala_CMOS	7	0.410	$[00-,10-,010]$	5/8	$[11-,011,101]$	4/8	0.375	0.5

A fault-tolerant circuit design should target a high FMR and high QoC. Please note that either s-a-0 or s-a-1 can occur for a circuit at any given time. Hence, the QoC can be given as the minimum QoC for s-a-0 and s-a-1. The QoC of the different voter circuits is presented in Table 5. The Kshirsagar voter architecture shows a good FMR. However, it is unable to establish a high QoC. The proposed voter produces the highest QoC of 0.625, followed by Classical_CMOS and Bala_CMOS, which present a good offset between the FMR and QoC. The proposed voter delivered upto 62.5% improvement in QoC.

Table 6 compares all voters in terms of the electrical features. The proposed design, with 18 transistors, outperforms all other designs in terms of the delay, power, and power-delay-product (PDP). There are four quad structures present in the design where each quad structure has an effective resistance equal to that of a single transistor [30]. The proposed design basically involves six effective resistances which is minimum among all existing works and enables the design to have the least delay compared to all the other voters. Since lesser number of transistors are used and pass transistors need lower switching energy to charge up a node because of the reduced voltage swing, the proposed design has the lowest power consumption. It should be noted that the Classical_CMOS, Bala_CMOS,

Ban, and proposed voters have the least numbers of transistors, ranging from 14 to 18. However, 12 transistors exclusively serve for the provisioning of quadded redundancy for the proposed voter. The Classical voter shows the highest PDP, whereas the proposed voter has the lowest PDP. The proposed voter delivered upto 56% improvement in PDP.

Table 6: Design Metrics.

Design	**Number of Transistors**	**Delay (ps)**	**Power (nW)**	**PDP** (10^3)
Classical	24	30.92	3233	99.96
Kshirsagar [1]	36	3.20	4395	14.09
Bala [9]	20	15.90	1533	24.37
Ban [19]	18	3.16	1404	4.44
Proposed Voter	18	1.85	427	0.79
Classical_CMOS	14	13.80	1267	17.48
Bala_CMOS	16	13.05	1179	15.39

Figure 13 shows the product of the power, delay, and area (PDAP) of the evaluated circuits. It also shows the relation between the number of circuit nodes and invulnerable input vectors. Note that the IIV, here, of each design is an intersection of the IIV for stuck-at-0 and IIV for stuck-at-1. The Kshirsagar voter [1] has the highest number of nodes, invulnerable input vectors do not exist, and the PDAP is extremely high. However, the proposed design has two invulnerable input vectors, minimum nodes, and a low PDAP.

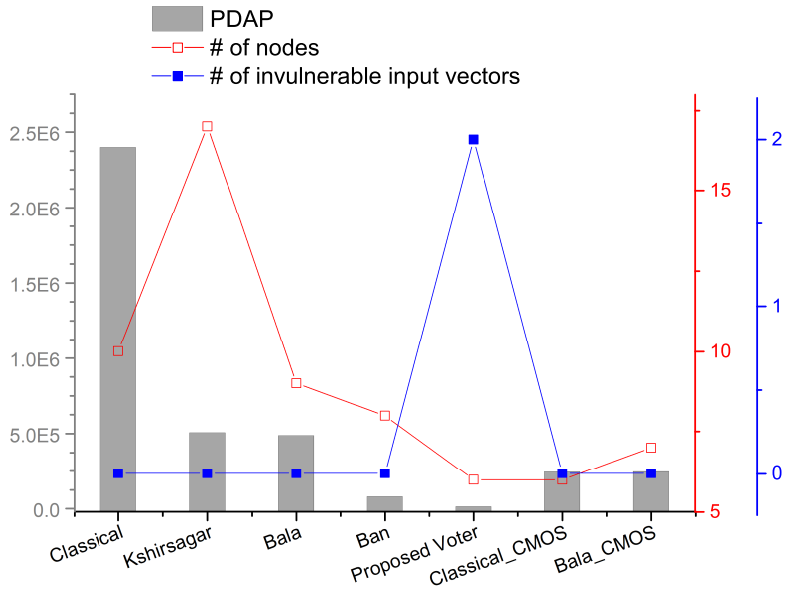


Figure 13: Relationship between nodes, invulnerable vectors, and PDAP.

3. Reliability Calculations

In this section the the reliability of the voters is analyzed, when a repair method is absent. Assume that the probability of the fault being masked is p. It relies on a ratio based on the number of simulations for which the fault propagates to the output and the total number of simulation iterations performed. It can be calculated using the simulation setup explained in Section 1. . Please note that exhaustively generated input vectors corresponding to a diverse sequencing of primary input patterns were applied to the voters for estimation of p.

$(1-p)$ represents the probability that the voter is unsuccessful in masking the fault . Consider a failure rate λ_1, then the probability for successful fault masking of voter is given as $\lambda_1 p$. Hence, the probability that the voter is affected and does not mask the fault can be represented by $(1-p)\lambda_1$.

The state transition diagram illustrated in Figure 14 is used to derive reliability of the voting circuit. The operational states of voters are given as : State-1 and state-2. State-2 represents the state where the voter experiences a fault but masks it. State-3 presents for the system failure. The transition rate from state-2 to state-3 is represented by λ_2. The reliability of the voter in the state transition diagram is then computed assuming $(\lambda_1 = \lambda_2)$ and is given by [34]:

$$R_V(t) = (1 + p\lambda t)e^{-\lambda t} \tag{16}$$

λ represents the failure rate of single node of the voter. In order to cover for all the nodes present in the circuit, $d\lambda$ is considered for the calculation of (16) where d is the total number of nodes in the circuit.

$$R_V(t) = (1 + pd\lambda t)e^{-d\lambda t} \tag{17}$$

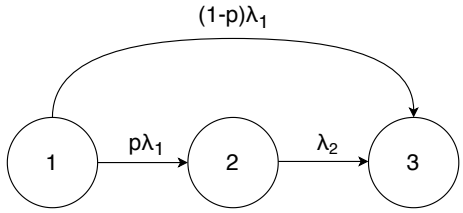


Figure 14: State transition diagram for reliability analysis.

The reliability curves for voters against normalized time (λt) are presented in Figure 15. In comparison with existing works [1], [9], [19], proposed voter is of most use for $\lambda t > 0.02$. It can be seen that the proposed voter is able to achieve upto 26.6% improvement in reliability for $\lambda t > 0.04$. The proposed voter shows improved reliability as compared with previous works, owing to its high FMR

and better QoC.

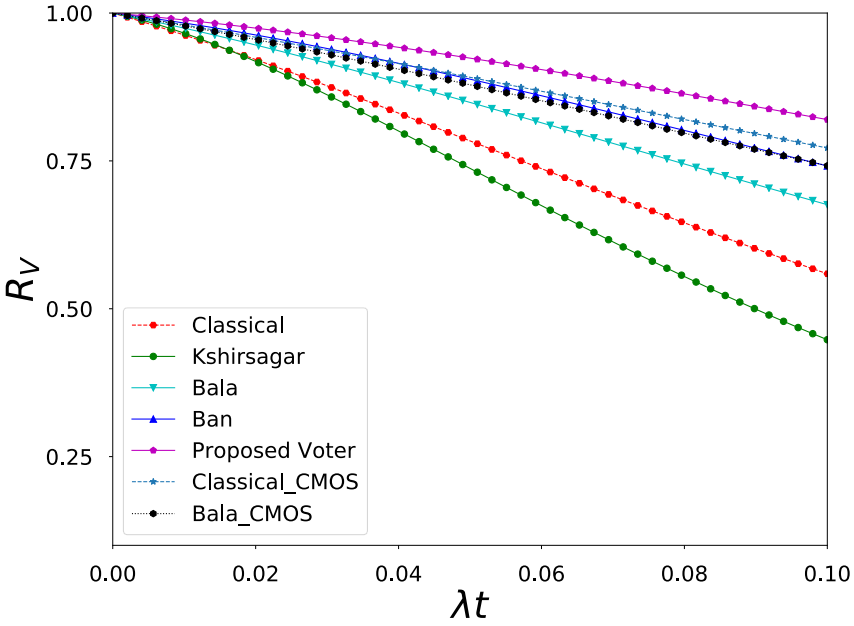


Figure 15: Reliability curves for voters.

V. CONCLUSION

In this thesis, an input-vulnerability aware approach for Approximate Tripple Modular redundancy is presented. Also, the work highlights the significance of a fault tolerant voter for ATMR and presents its design.

The simulation results for input vulnerability aware ATMR verified that pre-blocking of a vulnerable portion of the input space led to a drastic reduction in search space, yet fault coverage was not compromised. Thus, the reliability of ATMR was enhanced with the minimum area overhead. The presented structure is more suitable for larger circuits that target less processing time.

The significance of a voter for ATMR was also investigated. The work focused on fault tolerance at voter inputs and also, aimed for such voter design that minimum internal faults can propagate to output. A novel fault tolerant voter with superior electrical features designed using pass transistors and quadded transistor redundancy was proposed. The use of pass transistors aided in designing the voter with reduced number of transistors which led to minimum internal faults propagating to output. Since the number of transistors was less, conveniently embedded quadded transistor redundancy were conveniently used to mask faults at voter inputs. These design criteria especially target ATMR and the proposed voter is specialized design which is meant for ATMR system. Hence, the work proposed a novel fault tolerant voter for ATMR with superior electrical and masking features.

The merit of proposed voter design is illustrated through better QoC, a new proposed metric. A higher QoC ensure that greater number of input vectors are fully masked against internal faults. Finally, it is shown that the proposed voter has higher reliability than existing works. The proposed design can be effectively

used for TMR too because of its better electrical and fault tolerant characteristics. Please note that a fault tolerant TMR voter is not effective for ATMR.

PUBLICATIONS & PATENTS

A. Journals

1. T. Arifeen, A. S. Hassan, and J.-A. Lee, "A Fault Tolerant Voter for Approximate Triple Modular Redundancy", *Electronics*, vol. 8, no. 3, p. 332, 2019.

2. A. S. Hassan, T. Arifeen, H. Moradian, *et al.*, "Generation Methodology for Good-Enough Approximate Modules of ATMR", *Journal of Electronic Testing*, vol. 34, no. 6, pp. 651–665, 2018.

3. A. Hassan, U. Afzaal, T. Arifeen, *et al.*, "Input-Aware Implication Selection Scheme Utilizing ATPG for Efficient Concurrent Error Detection", *Electronics*, vol. 7, no. 10, p. 258, 2018.

4. T. Arifeen, A. S. Hassan, H. Moradian, *et al.*, "Input vulnerability-aware approximate triple modular redundancy: higher fault coverage, improved search space, and reduced area overhead", *Electronics Letters*, vol. 54, no. 15, pp. 934–936, 2018.

B. International Conferences

1. T. Arifeen, A. S. Hassan, and J. A. Lee, "Error Correctable Approximate Multiplier with Area/Power Efficient Design Through Mixed CMOS/PTL", in *2018 21st Euromicro Conference on Digital System Design (DSD)*, IEEE, 2018, pp. 190–195.

2. U. Afzaal, A. S. Hassan, T. Arifeen, *et al.*, "Effect of FPGA Circuit Implementation on Error Detection Using Logic Implication Checking", in

2018 21st Euromicro Conference on Digital System Design (DSD), IEEE, 2018, pp. 196–200.

3. T. Arifeen, A. S. Hassan, H. Moradian, *et al.*, "Probing approximate TMR in error resilient applications for better design tradeoffs", in *2016 Euromicro Conference on Digital System Design (DSD)*, IEEE, 2016, pp. 637–640.

C. Domestic Conferences

1. T. Arifeen, A. S. Hassan, H. Moradian, *et al.*, "Area and Power Efficient Adders for Digital Processing Systems", in *2017 27th Joint Conference on Information Communication (JCCI)*, 2017.

2. H. Moradian, T. Arifeen, A. S. Hassan, *et al.*, "Accuracy Analyzing of Approximate Adders Based on Input Data Range in IoT", in *2017 27th Joint Conference on Information Communication (JCCI)*, 2017.

3. A. S. Hassan, A. Hassan, T. Arifeen, *et al.*, "FPGA Based Low Power Design of An FIR Filter Using Polyphase Decomposition", in *2016 Korea Information Science Society (KISA)*, 2016.

D. Patents

1. J. A. Lee, T. Arifeen, A. S. Hassan, *et al.*, "Approximate adder consists of 4 transistors has TED of 4 and DSP integrated with the adder", pat., Korea Patent 1020160137439, 2019. [Online]. Available: `http://link.kipris.or.kr/link/main/sharePage_EN.jsp?reg_key=gaXejDaQeuOtyJld32oYgA==&APPLNO=1020160137439`.

2. J. A. Lee, T. Arifeen, A. S. Hassan, *et al.*, "Apprpximate adder consists of 18 transistors has TED of 2 and DSP integrated with the adder", pat., Korea Patent 1020160138097, 2019. [Online]. Available: http://link.kipris.or.kr/link/main/sharePage_EN.jsp?reg_key=gaXejDaQeuOtyJld32oYgA==&APPLNO=1020160138097.

3. J. A. Lee, T. Arifeen, A. S. Hassan, *et al.*, "Accurate adder consists of 14 transistors and DSP integrated with the adder", pat., Korea Patent 1020160135953, 2019. [Online]. Available: http://link.kipris.or.kr/link/main/sharePage_EN.jsp?reg_key=gaXejDaQeuOtyJld32oYgA==&APPLNO=1020160135953.

4. J. A. Lee, T. Arifeen, A. S. Hassan, *et al.*, "Apprpximate adder consists of 6 transistors has TED of 3 and DSP integrated with the adder", pat., Korea Patent 1020160137438, 2018. [Online]. Available: http://link.kipris.or.kr/link/main/sharePage_EN.jsp?reg_key=gaXejDaQeuOtyJld32oYgA==&APPLNO=1020160137438.

5. J. A. Lee, T. Arifeen, A. S. Hassan, *et al.*, "Systematic methodology for development of approximate circuits for triple modular redundancy modules and triple modular redundancy modules thereof", pat., Korea Patent 1020160107927, 2017. [Online]. Available: http://link.kipris.or.kr/link/main/sharePage_EN.jsp?reg_key=gaXejDaQeuOtyJld32oYgA==&APPLNO=1020160107927.

6. J. A. Lee, T. Arifeen, A. S. Hassan, *et al.*, "Accurate adder consists of 18 transistors and DSP integrated with the adder", pat., Korea Patent 1020160135758, 2018.

REFERENCES

[1] R. V. Kshirsagar and R. M. Patrikar, “Design of a novel fault-tolerant voter circuit for TMR implementation to improve reliability in digital circuits”, *Microelectronics Reliability*, vol. 49, no. 12, pp. 1573–1577, 2009.

[2] I. F. Oliveira, R. B. Schvittz, and P. F. Butzen, “Fault masking ratio analysis of majority voters topologies”, in *Test Symposium (LATS), 2018 IEEE 19th Latin-American*, IEEE, 2018, pp. 1–6.

[3] M. Blanke, M. Kinnaert, J. Lunze, M. Staroswiecki, and J Schröder, *Diagnosis and fault-tolerant control*. Springer, 2006, vol. 2.

[4] R. Isermann, *Fault-diagnosis systems: an introduction from fault detection to fault tolerance*. Springer Science & Business Media, 2006.

[5] I. A. Gomes, M. G. Martins, A. I. Reis, and F. L. Kastensmidt, “Exploring the use of approximate TMR to mask transient faults in logic with low area overhead”, *Microelectronics Reliability*, vol. 55, no. 9-10, pp. 2072–2076, 2015.

[6] P Balasubramanian and R. Naayagi, “Redundant logic insertion and fault tolerance improvement in combinational circuits”, in *Circuits, System and Simulation (ICCSS), 2017 International Conference on*, IEEE, 2017, pp. 6–13.

[7] P. Lala, F Bu-Saba, A Xie, and K. Yarlagadda, “Design of a fault-tolerant universal cell”, *International journal of electronics*, vol. 72, no. 3, pp. 467–470, 1992.

[9] P Balasubramanian and K Prasad, “A Fault Tolerance Improved Majority Voter for TMR System Architectures”, *WSEAS Transactions on Circuits and Systems*, vol. 15, no. 14, pp. 108–122, 2016.

[10] P. Balasubramanian, D. Maskell, and N. Mastorakis, “Majority and minority voted redundancy scheme for safety-critical applications with error/no-error signaling logic”, *Electronics*, vol. 7, no. 11, 2018, ISSN: 2079-9292. DOI: 10.3390/electronics7110272. [Online]. Available: http://www.mdpi.com/2079-9292/7/11/272.

[11] A. S. Hassan, T. Arifeen, H. Moradian, and J.-A. Lee, “Generation Methodology for Good-Enough Approximate Modules of ATMR”, *Journal of Electronic Testing*, vol. 34, no. 6, pp. 651–665, 2018.

[12] A. J. Sánchez-Clemente, L Entrena, and M. García-Valderas, “Partial tmr in fpgas using approximate logic circuits”, *IEEE Transactions on Nuclear Science*, vol. 63, no. 4, pp. 2233–2240, 2016.

[13] I. Polian and J. P. Hayes, “Selective hardening: Toward cost-effective error tolerance”, *IEEE Design & Test of Computers*, vol. 28, no. 3, pp. 54–63, 2011.

[14] I. Albandes, A. Serrano-Cases, M Martins, A. Martínez-Álvarez, S. Cuenca-Asensi, and F. L. Kastensmidt, “Design of approximate-tmr using approximate library and heuristic approaches”, *Microelectronics Reliability*, vol. 88, pp. 898–902, 2018.

[15] T. Arifeen, A. S. Hassan, H. Moradian, and J.-A. Lee, “Input Vulnerability-Aware Approximate Triple Modular Redundancy: Higher Fault Coverage, Improved Search Space, and Reduced Area Overhead”, *Electronics Letters*, 2018.

[16] O Ruano, J. Maestro, and P Reviriego, "A methodology for automatic insertion of selective tmr in digital circuits affected by seus", *IEEE Transactions on Nuclear Science*, vol. 56, no. 4, pp. 2091–2102, 2009.

[17] S. N. Pagliarini, L. A.d. B. Naviner, and J.-F. Naviner, "Selective hardening methodology for combinational logic", in *Test Workshop (LATW), 2012 13th Latin American*, IEEE, 2012, pp. 1–6.

[19] T. Ban and L. A. de Barros Naviner, "A simple fault-tolerant digital voter circuit in TMR nanoarchitectures", in *NEWCAS Conference (NEWCAS), 2010 8th IEEE International*, IEEE, 2010, pp. 269–272.

[20] K. Chen, J. Han, and F. Lombardi, "Two approximate voting schemes for reliable computing", *IEEE Transactions on Computers*, vol. 66, no. 7, pp. 1227–1239, 2017.

[21] I. A. Gomes, M. Martins, A. Reis, and F. L. Kastensmidt, "Using only redundant modules with approximate logic to reduce drastically area overhead in TMR", in *2015 16th Latin-American Test Symposium (LATS)*, IEEE, 2015, pp. 1–6.

[22] D. Shin and S. K. Gupta, "Approximate logic synthesis for error tolerant applications", in *2010 Design, Automation & Test in Europe Conference & Exhibition (DATE 2010)*, IEEE, 2010, pp. 957–960.

[23] H. Ichihara, T. Inaoka, T. Iwagaki, and T. Inoue, "Logic simplification by minterm complement for error tolerant application", in *2015 33rd IEEE International Conference on Computer Design (ICCD)*, IEEE, 2015, pp. 94–100.

[24] N. George and J. Lach, "Characterization of logical masking and error propagation in combinational circuits and effects on system vulnerability", in *2011 IEEE/IFIP 41st International Conference on Dependable Systems & Networks (DSN)*, IEEE, 2011, pp. 323–334.

[25] M. Yasin, O. Sinanoglu, and J. Rajendran, "Testing the trustworthiness of IC testing: An oracle-less attack on IC camouflaging", *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 11, pp. 2668–2682, 2017.

[26] T. Kirkland and M. R. Mercer, "Algorithms for automatic test-pattern generation", *IEEE Design & Test of Computers*, vol. 5, no. 3, pp. 43–55, 1988.

[27] H. Lee and D. Ha, "Atalanta: An efficient ATPG for combinational circuits", Technical Report, 93-12, Dep't of Electrical Eng., Virginia Polytechnic Institute and State University, Blacksburg, Virginia, Tech. Rep., 1993.

[28] M. R. Choudhury and K. Mohanram, "Low cost concurrent error masking using approximate logic circuits", *IEEE Transactions on computer-aided design of integrated circuits and systems*, vol. 32, no. 8, pp. 1163–1176, 2013.

[29] R. Brayton and A. Mishchenko, "ABC: An academic industrial-strength verification tool", in *International Conference on Computer Aided Verification*, Springer, 2010, pp. 24–40.

[30] A. H. El-Maleh, B. M. Al-Hashimi, A. Melouki, and A. Al-Yamani, "Transistor-level based defect-tolerance for reliable nanoelectronics", in *Robust Computing with Nano-scale Devices*, Springer, 2010, pp. 29–49.

[31] A. Mukherjee and A. S. Dhar, "Triple transistor based fault tolerance for resource constrained applications", *Microelectronics Journal*, vol. 68, pp. 1–6, 2017.

[32] I. A. Danilov, M. S. Gorbunov, and A. A. Antonov, "Set tolerance of 65 nm cmos majority voters: A comparative study", *IEEE Transactions on Nuclear Science*, vol. 61, no. 4, pp. 1597–1602, 2014.

[33] N. FreePDK45, *North Carolina State University*, `https://www.eda.ncsu.edu/wiki/FreePDK45:Contents`, Accessed: 2018-09-15, 2011.

[34] E. Dubrova, "Fault tolerant design: An introduction", *Department of Microelectronics and Information Technology, Royal Institute of Technology, Stockholm, Sweden*, 2008.

ACKNOWLEDGEMENTS

All praise is due to the Lord of the Worlds alone. This thesis made possible, my heartfelt appreciation for the constant reassurance and prayer of my family. The exceptional guidance and support from Professor Lee, my supervisor during the course of this degree.