American Journal of Engineering Research (AJER)
e-ISSN: 2320-0847 p-ISSN : 2320-0936
Volume-5, Issue-3, pp-77-85
www.ajer.org
Research Paper Open Access

Cost Efficient Design Approach for Reversible Programmable Logic Arrays

Md. RiazurRahman
(DepartmentofComputerScienceandEngineering, DaffodilInternationalUniversity, Bangladesh)

ABSTRACT :*Reversible programmable logicarrays (PLA) are at the heart of designing of efficient low power computers. This paper presents anefficient approach to design Reversible PLAs that maximizes the usability of garbage outputs and also reduces the number of ancilla inputs generated. The designfor proposed essentialcomponents and the architecture of reversible grid network for designing AND and EX-OR planes are also presented. Several algorithms have been proposed and presented to describe the programming interfaces in context of Reversible PLAs construction. Lastly, recent result on the trade-off between cost factors of standard benchmarkcircuits shows that the proposed design clearly outperforms the existing ones in terms various cost factors.*

Keywords :*Reversible Logic Circuits, Quantum Computing, Low Power Computing.*

I. INTRODUCTION

During execution of every single instruction, stuff wastes $klln(2)$ joule of energy that converted into heat due to per bit erase and reload where l is the operating temperature and k is the Boltzmann constant [1]. Solution to such input energy loss mechanism after publishing the tremendous approach called Reversible Computationwas introduced by Bennett [2] in1973. It opens the tunnel of designing robust architecture of low power consumption where total input energy loss is zero, supports the behaviour of optical computing, quantum computing, etc. Generic prototype of designing low power programmable devices [3] has obtained popularity in recent days. So, the development of reversible PLAs would be anotherapplication that enhances capability of low power computing. Proposed idea presents the novel architectureof PLAs in reversible computing by attaining 100% use of every logical units/gates that propagate all primary inputs to outputs. Proposed architecture reflects the following ideology:

- Maximize the usability of primary input signals
- Avoiding any type of EX-OR operations in AND plane
- Reduce number of garbage outputs and ancilla inputs

Rest of the paper has been organized as: section II has described reversible logic and the standards of measuring the performance of reversible circuits. Section III has presented the details of proposed gates and demonstrate organizational placement of logical units (gates) in Reversible PLAs grid. Section IV has illustrated the corresponding algorithms forconstructing AND and EX-OR planes using reversible {UMGand UNG} and CNOT gates, respectively. Comparative performance analysis based on benchmark standard circuits has been showed in section V. Finally, section VI has concluded this paper with the summary and future directions.The contents of each section may be provided to understand easily about the paper.

II. LITERATURE REVIEW

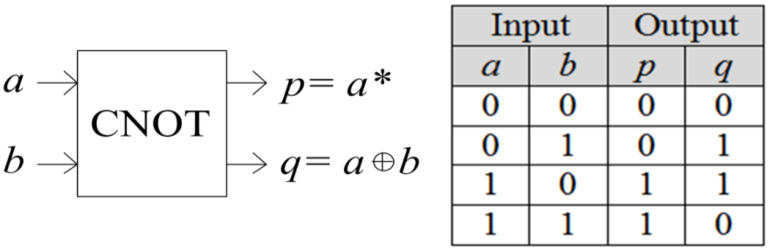
2.1Reversible Gates

Bidirectional orreversiblecircuitpreventsinputlossdueto uniquemappingbetweeninputandoutputstates.Likeclassicalcomputing,anyreversibleoperationalunitentityiscalled n×n, i.e. anyreversiblegatecontains:

- n-inputlinesandn-outputlines
- Uniquemappingbetweeninputandoutputstates

For example, controlled NOT (CNOT), widely known as Feynman gate [4] is reversible has two inputs (a, b) and two outputs (p, q) is shown in Fig. 1. Total number of input and output states are same (i.e. 4) and the mapping between input and output states is unique or vice versa.

There are many reversible gates have been populated based on conservativelogic [5], universalityof reversible circuit [6], fault tolerant mechanism [7], online testability [8], programmable devices [9], etc. Several reversible gates are self-reflexive, backs primary input signals by attaching self-copy and other gates stuck signals need extra circuitry to return in its initial state. In this paper, two new 3x3 reversible gates calledUniversal MUX (UMG) and Universal NOR (UNG) are used to design AND plane of reversible PLAs where UNG is self-reflexive reversible gate performs basic OR (or universal NOR) operation and returns primary inputs to output. On the other hand, UMG also performs AND operation and returns primary inputs as like UNG but not self-reflexive.



Input		Output	
a	*b*	*p*	*q*
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

Fig. 1: Reversible CNOT and unique I/O states mapping

2.2PerformanceMeasurementStandards

Operational Competency of any circuit is always related to its technical design encroachment. In any particular technology, greater number of logical units slows down the strength of signal hampers net processing speed of circuit. But interestingly, logical minimization provides better opportunity to reduce the number of operational units and total cost.

2.2.3Total Number ofGates:

Inreversiblecircuit,the input lossiszeroinidealstatebutbending inputsignalstooutput linesabsorbsenergyanddeclinesthestrengthofinternalsignal duetohaving unavoidable resistance.Thetotalnumberofgates usedincircuitisconsidered oneoftheworthycostfactor that controlsperformanceofdigitalcircuits [10].

2.2.4 QuantumCost:

Every reversible circuit point’s unique singular unitary matrix which can be accomplished with one or more 2x2and 4x4 unitary matrices whose are also compatible to 1x1and2x2 basic primitives in QuantumComputing.Alternatively, the n-dimensional quantum primitive is identically formed of 2n x2n dimensional unitary matrix. The total numbers of 2x2 quantum primitives are used to realize any reversible circuit is called Quantum Cost [11].

2.2.5 Garbage Output and Ancilla Input:

Unlike classical computing, reversible circuit requires extra output lines to map all input the states uniquely, called garbage output [12]. On the other hand, one or more input line(s) get saturated in constant level (i.e. 0 or 1) to perform specific operations is called Ancilla Input [13].According to above definitions, the realization of 2-input EX-OR operation requires only one 2x2 reversible Feynman gate and the quantum cost of Feynman gate is 1 (single 2x2quantum XOR gate is able to realize CNOT operation), the number of garbage output is 1 and finally, the number of ancilla input is zero.

2.3 Reviewon Reversible PLAs

In 2006, author of [14] has proposed the Reversible architecture of PLAs that was similar to classical PLA design [3] where AND plane consists of vertical complement and non-complement input lines and horizontal products lines spread over EX-OR plane. Toffoli gates were used to perform AND operation in AND plane whereas Feynman performed EX-OR operation in EX-OR plane. Additionally, Feynman gates were also used in AND plane for avoiding fan-out(s). The improved design of [14] was proposed in [15] brought prominent modification in the basic architecture of classical reversible PLA circuits by using only single line for each input literal in AND plane. Ref. [15] used MUX and Feynman gates to realize improved design of reversible PLAs where AND plane also performed copy operation by using Feynman gates as the similar way in [14]. Both papers had used multiple output functions F (i.e. Eqn. 1) as a sample to represent their proposed designs and minimization methodologies.

$$\boldsymbol{F} = \left\{ \begin{matrix} f_1 = ab' \oplus ab'c \\ f_2 = ac \oplus a'b'c \\ f_3 = ab' \oplus ab'c \oplus bc' \\ f_4 = ac \\ f_5 = ab' \oplus ac \oplus bc' \end{matrix} \right\} (1)$$

2.4 Motivation of this Research Work

Fundamentally, classical architecture of PLAs [3] was implemented by placing configurable switches at cross-point. These switches copy input signal multiple times increases fan-outs (which is restricted in Reversible Computing). The simplicity of AND, OR and NOT logic has been promoted by novel researchers to design such architecture of Programmable Logic Devices (PLD) in classical digital circuit [3]. But in reversible computing, the operability of basic bidirectional components is unavoidable when designing logic circuit such as Reversible Programmable Logic Arrays. In both [14] and [15], reversible PLAs focused the ideal zero energy dissipation due to use of large number of CNOT gates to recover fan-out(s) increases number of ancilla bits and garbage outputs. The idea of proposed research work comes through the reusability of garbage outputs as the input to next operational unit(s) that reduces the number of ancilla input at the same time.

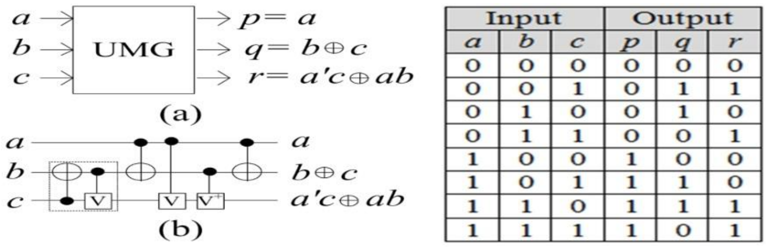
III. PROPOSEDREVERSIBLE GATESANDPLAGRID

In this section, two new reversible primitives (Universal MUX and Universal NOR gates) have been introduced followed by the demonstration of logical units placement in AND plane as well as the ordering principle of products generation.

3.1 New Reversible Gates and Operational Templates

3.1.1 Reversible Universal MUX Gate (UMG):

The input and output vectors of Universal MUX gate can be written as (a, b, c) and $(p = a, q = b \oplus c, r = ab \oplus a'c$;respectively. The equivalent quantum representation of UMG is shown in Fig. 2.

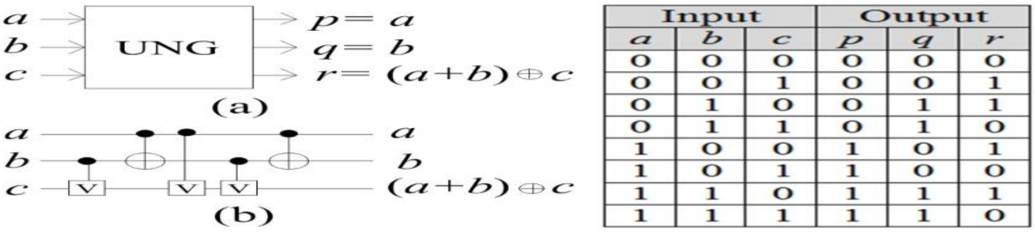


Input			Output		
a	b	c	p	q	r
0	0	0	0	0	0
0	0	1	0	1	1
0	1	0	0	1	0
0	1	1	0	0	1
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	1	0	1

Fig. 2: Reversible Universal Multiplexer gate (UMG): (a) Block diagram of UMG and (b) Quantum realization of UMG; Truth table of UMG maps uniquely all the input states to output states

3.1.2Reversible Universal NOR Gate (UNG):

In Boolean logic, NOR gate is a universal primitives can interpret the functionalities of all basic gates (AND, OR, NOT). Similarly,the input and output vectors of proposed 3x3 Universal NORgate which perform NOR operation can be written as $(a, b, c)$and$(p = a, q = b, r = (a + b) \oplus c)$, respectively (shown in Fig. 3).



Input			Output		
a	b	c	p	q	r
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	1
0	1	1	0	1	0
1	0	0	1	0	1
1	0	1	1	0	0
1	1	0	1	1	1
1	1	1	1	1	0

Fig. 3: Reversible Universal NOR gate (UNG): (a) Block diagram of UNG and (b) Quantum cost is 5; UNG maps input and output states uniquely

3.1.3 Operational Procedures of Proposed Gates:

Proposed UMG and UNG gates are used to perform AND operation of two literals (or a literal and a product). The forms of logical unit(s) which are used in proposed reversible PLAs have been selected based on following facts:

- Best orientation of Input and/or Output line(s)
- Projected output(s)(product/sum) of plane(AND/OR)

UMG performs MUX operation by setting input a as selectionlineandothers(bandc)asdata.ProposedUMG isabletogeneratethreemin termsoftwoinputs(ab, ab' and $a'b$) arerepresentedthroughtemplates α , β and γ (swaping orientationsare α' , β' and γ')asshowninFig.4.UMGdoesn't erasetheinputvalueofanyoperational unitwhileperforming ANDoperation andthoseunusedoutputscanbeusedasthe primaryinputstoanotherreversible gate.UNGrecoversthe limitationofUMGandtheoperationaltemplateissymbolized using$\pi(\pi')$(showninFig.5).

Algorithm 1 shows the methodology of selecting template (oriented form of logical unit) to perform AND operation of inputs p and$q$ depending on the value of swapFlag. The statement, swapFlag = 0 indicates to perform AND operation by usingα', β' or γ' (otherwiseα, β, γ or π).

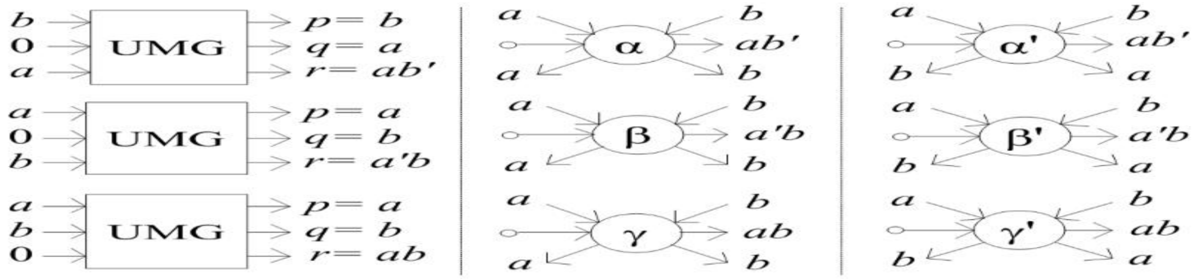


Fig. 4: Proposed templates of Universal MUX gate (UMG) that are used in proposed Reversible PLAs design

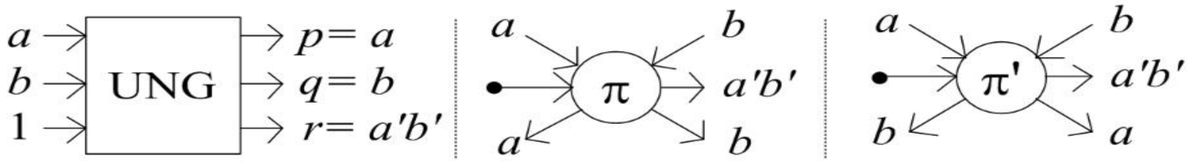


Fig. 5: Proposed templates of UniversalNOR Gate (UNG) those are used in proposed Reversible PLAs design

Algorithm 1:OpAND(p,q,swapFlag)

Templates $\{\alpha,\beta,\gamma,\pi\}$ (for swapping $\{\alpha',\beta',\gamma',\pi'\}$) are used to AND $\{p,q\}$ based on the value of *swapFlag*.

```
Start
If$p$isaLiteralincomplementedformThen
    If$q$isincomplementedformThen
        IfswapFlag=0Thenuse$\pi'$Elseuse$\pi$
        End If
    Else
        If$swapFlag=0$Thenuse$\beta'$Elseuse$\beta$
        End If
    End If
Else
    If$q$isincomplementedformThen
        If$swapFlag=0$Thenuse$\alpha'$Elseuse$\alpha$
        End If
    Else
        If$swapFlag=0$Thenuse$\gamma'$Elseuse$\gamma$
        End If
```

```
16.         End If
17.End If
18.Return$p, q$
END
```

Ontheotherhand,proposedEX-ORplaneconsistsofonly FeynmangatesareusedtoperformXORingproducts. Threetemplates of Feynman gates have been used in proposed design are symbolized through symbols Δ,$\lambda$and ∇perform NOT$\{a, a'\}$, EX-OR $\{a, (a \oplus b)\}$and COPY operation $\{a, a\}$,respectively (shown in Fig. 6).

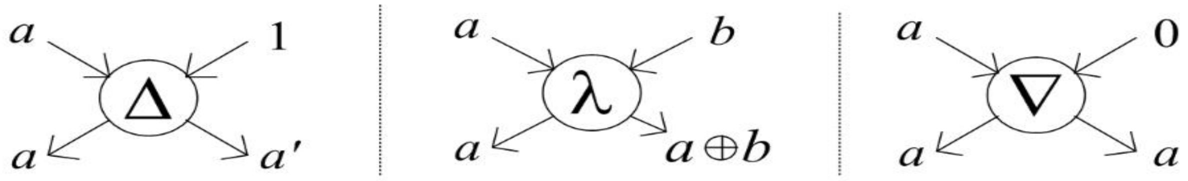


Fig. 6: Templates of CNOT gate are used to design EX-OR plane

3.2 Reversible PLAs Grid and Primitives Placement

Reversible gatesaremorepowerfulperformsmultiplelogic operations insinglecycle[10].Theorientation ofinputand productlinesofproposed ANDplaneispointedthroughsolid lines(shown inFig.7a)wheredottedlinesindicate another pathwaytoswapinputsignals(showninFig.7b)according tofollowingalgorithm(Algorithm2).

```
Algorithm 2: SwapLiterals($L_i, L_j$ )
Exchanging input signals $\{I_i, I_j\}$ in lines, $\{L_i, L_j\}$
Start
1. Set $a$:= signal at input line, $L_i$
2. Set $b$:= signal at input line, $L_j$
3. $L_i$:= $b$ and $L_j$ := $a$
End
```

Basically, swap operation of two literals be performed when the uses of any literal got ended for doing AND operation in AND plane. SwapLiterals (L_i, L_j) moves unused literals from left to right vertical tracks of AND Plane. Performing AND operation at any cross-point of two vertical lines binds single horizontal line to generate cumulative product and again, connecting another literals to cumulative product (if needed) to generate final product of AND plane.

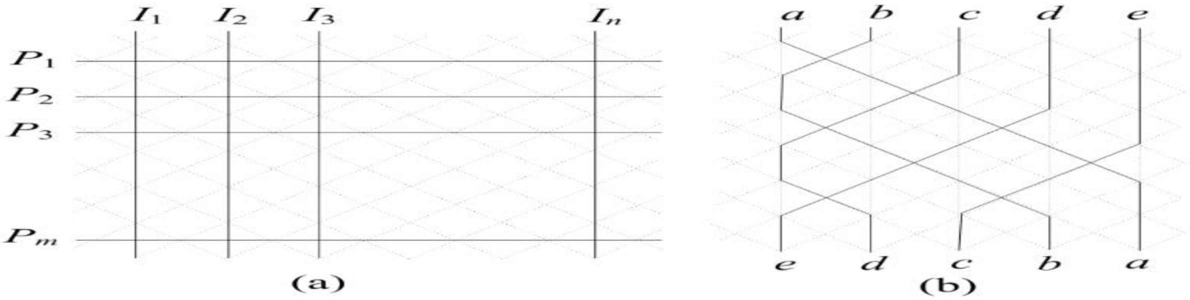


Fig. 7: Reversible PLAs Grid Architecture of AND plane: (a) Primary inputs and generated products passed through vertical and horizontal lines, respectively and (b) Swapping inputs

Ordering products takescrucialrolereducesthecostof productsgeneration byusinggarbageoutput(s).Also,the usabilityofdifferenttemplates providesminingopportunities optimizing thecostinphysicallayer.Resultantproductscontainsamenumberofliteralsplacedaccordingtotheorderof inputliterals.For example, products (P_v) consist of literals $\{a, b$ *and* $c\}$ be produced in order, product(s) start with a followedby start with $b$followed by start with c.Algorithm 3 describes the methodology of placing products based on their usability. For example, product *abcd*be generated before*abc, abd or ab* which are consisted of less number of literals.

Algorithm 3:OrderingProducts(I_v,P_v)

Products,$P_v$beorderedaccordingtoinputs,I_v.

```
Start
SetP_Q:=∅[P_Q isusedtostoreproducts]
Sort P_v basedonSizeOf(P_i)indescendingorder
For i=1tototalLiterals
    For j=1tototalProducts
        If I_i ∈ P_j and P_j ∉ P_Q then Add P_j to P_Q
        End If
    End Loop
End Loop
Set P_v:=P_Q
End
```

According to above algorithm (ALG. 3), the order of products consists of inputs *a*, *b*, *c*, *d* and *e* (only the non-complemented forms) can be graphed as shown in Fig. 8.

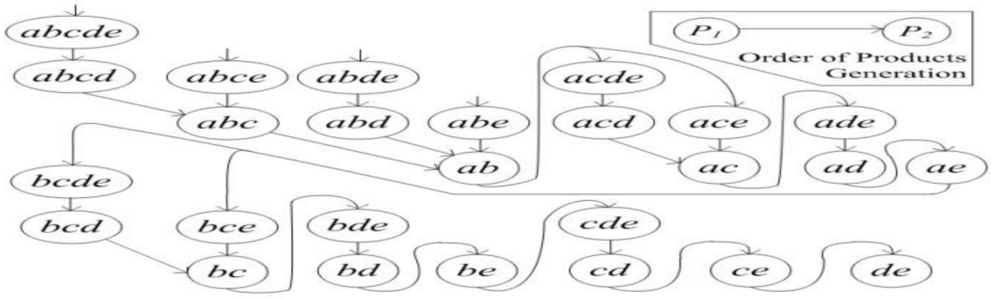


Fig. 8: The order of products consist of literals (a, b, c, d, e) is: {[abcde, abce, abde, abe], [abcd],[abc, abd], [ab], [acde, ace], [acd], [ac], [ade], [ad], [ae], [bcde, bce], [bcd], [bc], [bde], [bd], [be], [cde], [cd], [ce], [de]}. Products in the same group (for example [abcde, abce, abde, abe]) are independent can be generated in any order.

IV. PROPOSED DESIGN OF AND AND OR PLANES

Inthissection,proposed designofANDplanebasedhas beendescribedfollowedbytherealizationofEX-ORplane.

4.1 Designing Reversible AND and EX-OR Planes

ANDplanedominatestheperformance andcostfactorsof reversible PLAswhereeveryANDoperation risesallcost factorscomparedtosimpleofEX-ORoperation. Algorithm 4 presentstheconstructionofproposedANDplaneaswell astheminimizationofgarbage outputs. Theconstructionof ANDplaneincludes,theorderingofproducts(Algorithm 3) followedbycounting*swapFlag*andtheninvokes*OpAND(p, q,swapFlag)*(Algorithm 1).Inputlinesexchangesignals(as Algorithm2)afterfinishingthegenerationofallmutualproduct(s).Finally,Queue(*P_{QG}*)storesunusedgarbageproducts whichareusedinafterwardasresultantproductswhenthey getsimilartounexploredproducts.

Algorithm 4:ConstuctANDPlane(I_v , P_v)

Thisfunctionconstructs ANDplanebytakingsetof input literals(I_v)andgeneratesproducts(P_v)connectingmultiple inputlines(L_v)byusingUMGand UNG gates.

```
Start
OrderingProducts(I_v,P_v)
Set P_QG:=∅ and ndot:= 0 [P_QG stores garbage]
For g=1tototalLiterals-1
  For h=g+1tototalLiterals
    SetswapFlag :=0
    For i=1tototalProducts
      If I_g I_h ∈ P_i Then swapFlag :=swapFlag + 1
      End If
```

```
        End Loop
    If $swapFlag$>0 Then
      For $i$=1 to $totalProducts$
        If SizeOf($P_i$)>1 Then
          If $P_i \in P_{OG}$ Then Remove $P_i$ from $P_{OG}$
          Else
            If $I_g I_h \in P_i$ Then $swapFlag := swapFlag - 1$
              Set $pivotP$:=OpAND($I_g$,$I_h$,$swapFlag$)
              If SizeOf($P_i$)>2 Then
                For $k = h + 1$ to $totalLiterals$
                  If $I_k \in P_i$ Then
                    Set $P_G$:= $pivotP$
                    $pivotP$:=OpAND($pivotP$,$I_k$, $false$)
                  End If
                End Loop
                Add $P_G$ to $P_{OG}$ [Add new garbage to $P_{OG}$]
              End If
            End If
          End If
        Else $ndot$:=$ndot$ + 1 [ Use via (.)]
        End If
      End Loop
    Else SwapLiterals($L_G$ , $L_h$) [No mutual products]
    End If
  End Loop
End Loop
End
```

Theorem 1. Let,$n$bethenumberofANDoperations of$m$ outputfunctionsand$t$bethenumberofAND operation of garbage outputs,(P_{OG})whichare identicaltoanyproducts thentheminimumnumberofreversiblegatestorealizeAND planeis(n-t),thequantumcost is5(n-t),thenumberofancilla inputis(n-t).

Proof: As performing every reversible AND operation needs single UMG or UNG gate, results total number of gates to realize AND plane is n. But reusability of garbage reduces the number of acting AND operations to (n-t). Similarly, the quantum cost of UMG or UNG is 5 sums-up the total quantum cost of circuit is 5(n-t) and every reversible AND operationrequiresan ancilla bitsummarizestotalnumberof ancilla inputsto(n-t).

Formulti-outputfunction$F$ inEqn. (1),totalnumberof ANDoperations(n)is7,thenumberofANDoperation(s) ingarbage whicharesimilartoany product (t)is1.So,the numberofgates=(n-t) =7-1=6,quantum cost=30andtotal ancillainput=6(showninFig.9).

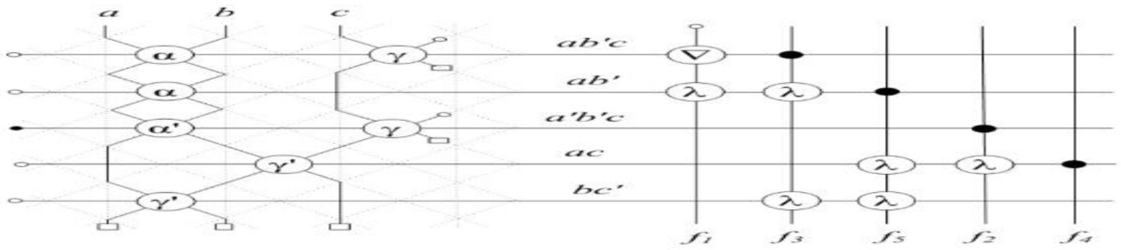


Fig. 9: OptimizedversionofreversiblePLAsofmulti-outputfunction$F$in Equ. (1)

Theorem 2. Let,$p$bethenumberofproducts(consistofmore thantwoliterals)of$m$outputfunctions, $q$bethenumberof garbageoutputswhichareidenticaltoanyproductsand$ndot$ bethenumberofcross-pointthenthenumberofgarbage is$p$ + $totalLiterals$– $ndot$– q.

Algorithm5describestheconstruction ofEX-ORplaneby usingFeynmangateswhere(λ)connectsproductlines(P_i)to correspondingfunctionlines(F_j)producesoutputsignalsand anotheridenticalcopyofP_i.

Algorithm 5:ConstructXORplane(P_v,F_v)

EX-ORplanegeneratesthefinal outputsofmulti-output function,(F_v)consistsofproducts,P_v.

```
Start
Set$F_Q$:=∅ and $xdot$:= 0
For $i$=1to$totalProducts$
    For $j$ =1to$totalFunctions$
        If$P_i \in F_j$ Then
            If$F_i \notin F_Q$ Then
                IfFreqOf($P_i$) == 1 Then
                    $xdot$:=$xdot$+1
                Else
Use ∇ [ Keep a copy of $P_i$]
                    Set FreqOf($P_i$) :=FreqOf($P_i$) – 1
                End If
                Add$F_i$ to $F_Q$
            ElseUse$\lambda$[XORing $P_i$ to $F_j$ line]
End If
Else If$P_i'$ ∈ $F_j$ Then
UseΔ[ Keep a copy of $P_i$]
End If
    End Loop
End Loop
End
```

Theorem 3.Let,*n*bethenumberofEX-ORoperations of *m*outputfunctions and*xdot*bethenumberofcross-points, thentheminimumnumberofFeynmangatestorealizeEX-OR planeis*n*+*m*–*xdot*,thequantumcostis*n*+*m*–*xdot*,total numberofancillainputis*m*–*xdot*.

According toproposedalgorithms (ALG.4&5),theconstructionofmulti-outputfunction*F*inEquation(1)isshown inFig.9wheregarbageoutputsarerepresented byusingline ends with box. Table 1 summarizes that the proposeddesignof reversiblePLAsrequireslessnumberofgates,garbageoutputs andancillainputsaswellasminimum quantumcostcompared toexistingdesign[15].

Table 1. Comparison between the proposed and existing [15] designs of multi-output function *F* in equation (1)

RPLAs Design	Total Gates (GA)	Garbage (GB)	Ancilla Input (AI)	Quantum Cost (QC)
Proposed	13	5	7	37
Existing [15]	18	10	12	39

V. PERFORMANCEANALYSIS

The realizationof benchmarkcircuitsanalysisbased on proposed algorithms by using programminglanguage Java (jdk1.7)onNetbeansIDE(8.0)inWindow7Workstation is presentedinTable 2.Alltheexperiment resultsaretestedon Intel(R)Core(TM) i3 CPU @3.30GHz with 2GB RAM.Table 2 showstheexperimental resultsfordifferentbenchmark functionsandthecomparison withtheexistingmethod[15] wheretherequired numberofgates,garbage outputsand ancillainputsareminimized innotablemanner.Finally,the trade-off between quantumcostandotherfactorssummarizes thebetteroptimization ofproposeddesignofreversiblePLAs is presented.

Table2.Experimentalresultsusingdifferentbenchmarkfunctions

RPLAs	Total Gates (GA)		Garbage (GB)		Ancilla Input (AI)		Quantum Cost (QC)	
	Existing	Proposed	Existing	Proposed	Existing	Proposed	Existing	Proposed
5xp1	166	132	112	44	115	47	418	452
9sym	427	377	385	111	377	103	1405	1681
adr3	67	52	48	26	46	24	157	172
apex3	3998	3719	1799	520	1795	516	8654	9655
b12	159	127	132	56	167	50	453	495
bw	305	281	64	30	87	53	446	445
cordic	12162	11389	10640	1573	10661	1552	41694	50765
duke2	941	874	667	157	674	164	2735	3262
e64	2170	2074	2148	121	2148	121	8410	10282
inc4	16	11	10	4	11	5	34	35
inc5	23	16	16	6	17	7	53	56
misex1	88	74	51	25	50	24	193	206
misex2	199	165	176	79	149	52	122	673
pdc	3801	3451	3006	487	3030	511	12096	14115
rd53	56	44	42	23	40	21	134	141
rd73	187	150	141	68	137	64	487	550
rd84	328	267	265	114	261	110	928	1067
sasao	14	9	13	8	10	5	29	29
sao2	284	257	236	60	230	54	890	1065
t481	49	40	53	36	38	21	134	152
table3	2602	2439	1814	342	1814	342	7537	8995
table5	2539	2384	1819	319	1682	182	7516	8896
xor5	8	4	8	9	4	5	8	4
z5xp1	167	139	114	50	117	53	425	483
z9sym	427	377	385	111	377	103	1405	1681

VI. CONCLUSION

Reversible computing is in high demands due to the increasing thirst for low power computation. The proposeddesign increases thereusability ofgarbageoutputs which in turn enhanceszeroenergydissipationwhichistheprimary concern forreversiblecomputing.This work has introduced a novel approach to design reversible PLAs by proposing reversible PLAs grid and algorithms to construct AND and EX-OR planes with minimized gates and other cost factors which lead towards the advancement of reversible PLAs which will promote the development of Reversible Field Programmable Logic Arrays in the near future[9].

References

[1] R.Landauer,Irreversibilityandheatgeneration inthecomputingprocess,*IBMJ.ofR&D,vol. 5(3)*,pp.183-191,1961.

[2] C.H.Bennett,LogicalReversibilityofComputation,*IBM J.ofResearch andDevelopment,vol.17(6)*,pp.525-532,1973.

[3] H. Fleisher and L.I. Maissel, An Introduction to Array Logic, *IBM J. of Res. and Development, vol. 19(2)*, pp. 98-109, 1975.

[4] R.P.Feynman,QuantumMechanicalComputers, *FoundationsofPhysics, vol.16(6)*,pp.507-531,1986.

[5] E. Fredkin and T. Toffoli, Conservative Logic, *International Journal of Theoretical Physics, vol. 21*, pp. 219-253, 1982.

[6] T.Toffoli,Reversible Computing,*LectureNotesinComp.Sci.onAuto., Lang.andProg., vol.85,pp.632*-644,1980.

[7] B. Parhami, Fault-tolerant Reversible Circuits, *Proc. of 40thAsilomar conference on sig., sys. and comp., ACSSC'06*, CA, pp. 1726-1729, 2006.

[8] S.N. Mahammad and K. Veezhinathan,Constructing Online Testable Circuits UsingReversibleLogic,IEEETrans. OnInstrumentationand Measurement,vol.59(1),pp.101-109,2010.

[9] A.S.M.SayemandS.K.Mitra,Fault-tolerantReversibleCircuits,*Proc. Of IEEE Int'l conf.oncomputersci.and automationengg.*,CSAE'11, Shanghai,China,pp.251-255,2011.

[10] A.K.Biswas,M.M.Hasan,A.R.ChowdhuryandH.M.H.Babu,Efficient Approaches for designing ReversibleBinary Coded Decimal Adders, *Micro.J.*,vol.39(12),pp.1693-1703,2008.

[11] M.Perkowskiandetal,Ahierarchical approach tocomputeraided designofquantum circuits,*Proc. of 6thInt'l sym.onR. & M.ofFuture Comp.Tech.*,Germany,pp.201-209,2003.

[12] M.R. Rahman, Cost Efficient Fault Tolerant Decoder in Reversible Logic Synthesis, *International Journal of Computer Applications*, vol. 108-2, pp. 7-12, 2014.

[13] H.ThapliyalandN.Ranganathan,Design,synthesisand testofreversiblecircuitsforemergingnanotechnologies,*Proc.of IEEEcomp. soc., annualsym.onVLSI*,USA,pp.5-6, 2012.

[14] A.R. Chowdhury and R. Nazmul and H.M.H. Babu, A new approach to synthesize multiple-output functions using reversible programmable logic arrays, *Proc. of 19th Int'l conf. on VLSI Design (VLSID'06)*, India, pp. 311–316, 2006.

[15] S.K. Mitra and L. Jamal and M. Kaneko and H.M.H. Babu, An efficient approach for designing and minimizing reversible programmable logic arrays, *Proc. of the Great Lakes sym. on VLSI*, ACM, New York, NY, USA, pp. 215–220, 2012.