

Approximate computing is a design paradigm that trades off accuracy for improvement in hardware cost and energy efficiency. Approximate logic synthesis (ALS) focuses on automatically synthesizing an optimized circuit that approximately implements the target function according to some error constraints. The objective is to identify an approximate SOP expression with the lowest literal count for a given optimized SOP and a threshold of number-of-errors (NoE).

There are two ways to approximate an SOP: (1) to insert a new cube that allow remove cubes in the SOP, and (2) to directly remove a cube from the SOP. The proposed method flow is presented in Algorithm 1.

Algorithm 1: Proposed Method

Input: A simplified SOP expression F and a NoE e

Output: an approximated SOP expression F'

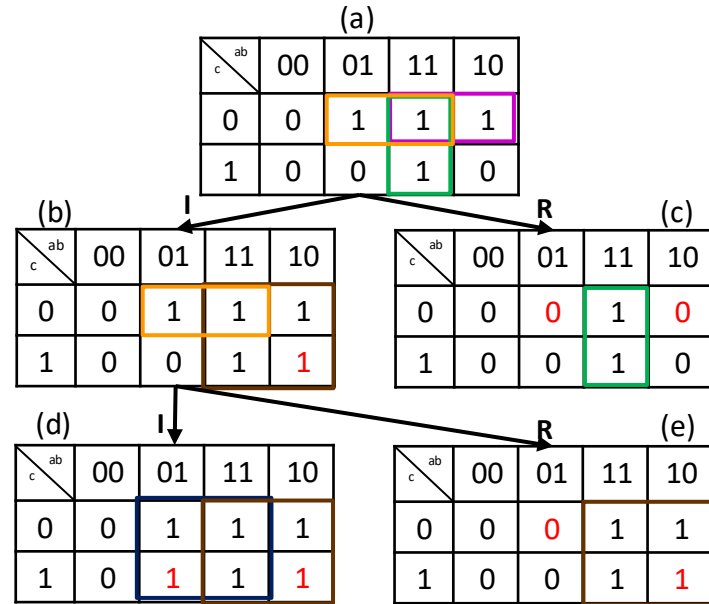
```

1  $S_0 \leftarrow F$ ,  $best \leftarrow F$ ;
2 for  $i \leftarrow 0$  to  $e-1$  do
3    $S_{i+1} \leftarrow \text{CubeInsert}(S_i, i+1)$ ;
4    $sr \leftarrow \text{CubeRemove}(S_i, e-i)$ ;
5   if  $\min(S_{i+1}, sr) < best$  then  $best \leftarrow \min(S_{i+1}, sr)$ ;
6 end
7 return  $\text{espresso}(best)$ ;

```

In Figure 1 is presented an execution example for NoE = 2 with Karnaugh map. Figure 1(a) presents the original SOP. The arrows marked with "I" are a *CubeInsert* execution and the ones with "R" a *CubeRemove* execution. The red colored minterms represents an error. *CubeInsert* execution can insert one error more than the last solution whereas *CubeRemove* execution can insert the difference between the last solution error and the NoE threshold.

Figure 1 – Example of execution with NoE = 2



In Table 1 is shown the comparison to the state-of-the-art with a NoE threshold of 16 [1]. In Table 2 is shown our results with NoE equal to $ER * 2^i$.

Table 1 – Comparison considering 16 errors.

Circuit	Literals			Time(s)	
	Orig	[1]	Ours	[1]	Ours
5xp1i:7;o:10	347	235	202	0,73	0,51
sqrt8i:8;o:4	188	98	84	0,59	0,35
clipi:9;o:5	793	588	584	1,99	1,59
sao2i:10;o:4	496	231	165	2,48	1,97
misex3ci:14;o:14	1561	1239	1115	252	26,89
table5i:17;o:15	2501	2410	2270	7868	27,44
Average	981	800	737	1354,3	9,79

Table 2 – Results considering percentage ER

Circuit	Literals	ER(%)	NoE	App. Lit.	Reduction	Time(s)
sao2i:10;o:4	496	1	10	281	43 %	1,25
		3	30	79	84 %	5,13
		5	51	37	93 %	7,76
misex3ci:14;o:14	1561	1	163	678	57 %	180,56
		3	491	572	63 %	301,77
		5	819	514	67 %	467,34
table5i:17;o:15	2501	1	1310	720	71 %	200,55
		3	3932	280	89 %	407,72
		5	6553	153	94 %	478,07

[1] Su, S. et al. "A Novel Heuristic Search Method for Two-Level Approximate Logic Synthesis", IEEE TCAD (2020), v.39, n.3, p.654–69.