Multiplier Input Decomposition Instances generated by ToughSAT

1st Shunyang Bi, 1st Zhang Qu, 5th Hailong You
School of Microelectronics
XiDian University
Xi'an China
shybi@stu.xidian.edu.cn, quzhang2019@stu.xidian.edu.cn, hlyou@mail.xidian.edu.cn

2nd Meihua Liu
School of Electronic and Computer Engineering
Peking University Shenzhen Graduate School
Shenzhen Guangdong China
liumh@pku.edu.cn

3rd Pengfei Li, 4th Yang Zhang
EDA Group
SMIT Holdings Limited
Shenzhen Guangdong China
379401663@qq.com, yanzhang@smit.com.cn

***Abstract*—this description introduce our instances to the SAT Competition 2021. We generated instances that would select proper input decomposition from multiplication of large numbers.**

I. DATA

In the circuit design of n-digit multiplication multiplier, using the ordinary multiplication algorithm needs n^2 times of multiplication, while $3 * n^{\log_2 3}$ ($3 * n^{1.585}$) times of multiplication in the fast multiplication algorithm(Karatsuba's algorithm). For example, let x and y be represented as n-bit strings in a cardinality b. For any positive integer less than n, two given numbers can be written as:

$$x = b^m * x_1 + x_0$$

$$y = b^m * y_1 + y_0$$

Where x and y are less than b^m, that is to say:

$$x * y = (b^m * x_1 + x_0) * (b^m * y_1 + y_0)$$

Let

$$z_0 = x_1 * y_1$$

$$z_1 = x_0 * y_1 + x_1 * y_0$$

$$z_2 = x_0 * y_0$$

Then,

$$x * y = b^{2m} * z_0 + b^m * z_1 + z_2$$

In this process, it takes 4 times multiplication operations to decompose the multiplication. But in fast multiplication algorithm, z_1 can be expressed as:

$$z_1 = (x_1 + x_0) * (y_1 + y_0) - x_1 * y_1 - x_0 * y_0$$

And we just need 3 times of multiplication. In the actual circuit, we need to verify whether this decomposition method is feasible.

II. SELECTION

Whether the input of a designed multiplier circuit can be decomposed into multiplication factor based on fast multiplication algorithm is very important for our circuit design. The multiplier constraint is defined as the multiplier inputs of the circuit we designed. These inputs have appeared in our circuit design. We define the input in the multiplier as f_1, f_2, and assign them according to the actual design circuit. TABLE Ⅰ shows the running time of 20 instances in Minisat.

TABLE I. RESULTS WITH MINISAT FOR 20 INSTANCES SUBMITTED FOR SAT COMPETITION-2021.

Instance name	*f1*	*f2*	*Minisat Time*	*Status*
Circuit_multiplier_18.cnf	71472475	35478902	5000	UNKNOWN
Circuit_multiplier_20.cnf	17783402	274475	206.98	SAT
Circuit_multiplier_22.cnf	47545134	8348021	1659.06	SAT
Circuit_multiplier_23.cnf	54513144	34802174	595.69	SAT
Circuit_multiplier_24.cnf	479613144	1802174	5000	UNKNOWN
Circuit_multiplier_25.cnf	96131440	802174	5000	UNKNOWN
Circuit_multiplier_26.cnf	61314404	2174734	5000	UNKNOWN
Circuit_multiplier_28.cnf	144024741	773457	1444.49	SAT
Circuit_multiplier_29.cnf	77340057	40247415	5000	UNKNOWN
Circuit_multiplier_33.cnf	979147121	175171	253.31	SAT
Circuit_multiplier_34.cnf	59147121	7325171	3073.5	SAT
Circuit_multiplier_35.cnf	98325171	1441721	4539.31	SAT
Circuit_multiplier_36.cnf	179325171	93411721	5000	UNKNOWN
Circuit_multiplier_37.cnf	9263325171	721721	181.99	SAT
Circuit_multiplier_47.cnf	977317491	7894567	5000	UNKNOWN
Circuit_multiplier_48.cnf	435678915	9647851	4307.78	SAT
Circuit_multiplier_45.cnf	169117141	16773165	703.21	SAT
Circuit_multiplier_17.cnf	8642475	6547892	500.02	SAT
Circuit_multiplier_53.cnf	92147042	13795646	5000	UNKNOWN
Circuit_multiplier_54.cnf	92776646	85247042	5000	UNKNOWN

III. TOOLS

We used ToughSAT [1] to assist in adding the constraints of multiplier and generating the CNF formulas.

REFERENCES

[1] Joseph Bebel, "Harder SA T Instances from Factoring with Karatsuba and Espresso,"in Proceedings of SAT Competition 2019. [Online]. Available: https://helda.helsinki.fi/handle/10138/306988