UDC 004.4–004.9

INFERENCE OF LOW FAN-OUT IF-DECISION DIAGRAMS FOR LOGARITHMIC-DEPTH ADDERS

Prihozhy A. A.
Belarusian National Technical University, Minsk, Belarus
prihozhy@yahoo.com

The addition operation is critical in almost all modern processing units [1–6]. The adder parameters such as implementation area, latency and power dissipation decide the choice of adders for different applications. There is an extensive research attention towards designing higher speed and less complex adder architectures for electronic and quantum implementations. Decision diagram-based approaches [7–9] are a promising direction in the design of adders with required properties. The traditional binary decision diagrams have been extended to functional, biconditional, if-decision and other diagram types [10–18], which are more suitable for the adder design and optimization. The adder fan-out is crucial for most of electronic and quantum implementation technologies. This work proposes a formal method of inferring logarithmic-depth if-decision diagrams of low fan-out parallel adders. In the diagrams, long paths are split to shorter paths, which reduce time delays in the adder. Experimental results obtained in the work show that the parallel adder fan-out does not exceed four, which is much less than the fan-out of parallel adders generated using if-decision diagrams and described in [6]. The decrease of fan-out costs certain increase in adder size and depth.

The ripple-carry adder is constructed from the algorithm of adding two numbers represented in binary number system [1]. The look ahead adders such as Kogge-Stone [2], Brent-Kung [3] and others [4] are constructed on the concept of generation and propagation signals. The quantum adders [9–11] are constructed on the classical theory of adders from one side and on the theory of reversible functions from other side. Instead, this paper develops a method of formal inference of all kinds of adders that is based on the theory of incompletely specified functions and if-decision diagrams. The key mechanism of inferring fast parallel adders is the split of long paths in the if-decision diagram. Advantages of such an approach are an efficient exploration of the adder design space and the possibility of generating adders with required properties.

The author of [6] proposed a method of inference fast low-size parallel adders of any bit-size, which have many advantages and only one essential drawback, i. e., their fan-out grows rapidly depending on the adder bit-width. Many technologies cannot overcome the drawback and lead to adder implementations with worse parameters than expected. This paper proposes a technique, which is capable of generation parallel logarithmic-depth adders whose fan-out is restricted and small.

Let $f(x)$ and $d(x)$ be Boolean functions of vector argument $x = (x_1, \dots, x_n)$. Let $f^{on}(x)$ be an on-set of variable x values such that $f(x) = 1$. Variable g represents Boolean function $min\ (f \mid d)$ belonging to the slice of functions defined as follows [12–14]:

$$(f \wedge d)^{on} \subseteq g^{on} \subseteq (f \vee d)^{on} \qquad (1)$$

where $\wedge$ and $\vee$ are Boolean conjunction and disjunction respectively. Analogously, variable h represents Boolean function $min\ (f\mid\neg d)$. The following expansion of function f on function d holds:

$$f = d \wedge min(f|d) \vee \neg d \wedge min(f|\neg d) \tag{2}$$

where $\neg$ is Boolean negation. The expansion allows the construction of a nonterminal node (fig. 1a) of an if-decision diagram (IFD) proposed in [15–17]. Fig. 1b depicts a two-root IFD of 1-bit full adder. Fig. 2 depicts a many-root IFD of 8-bit ripple-carry adder. The IFD contains a nine-node long path which cause big time delays in adder implementation. To obtain faster adders, a technique proposed in [5] splits the long paths originated from the nodes s0…s7 and c7 of the IFD into shorter paths, which leads to a new IFD of the parallel many-bit adder of logarithmic-depth. The key drawback of the IFD is the exponential growth of its fan-out.

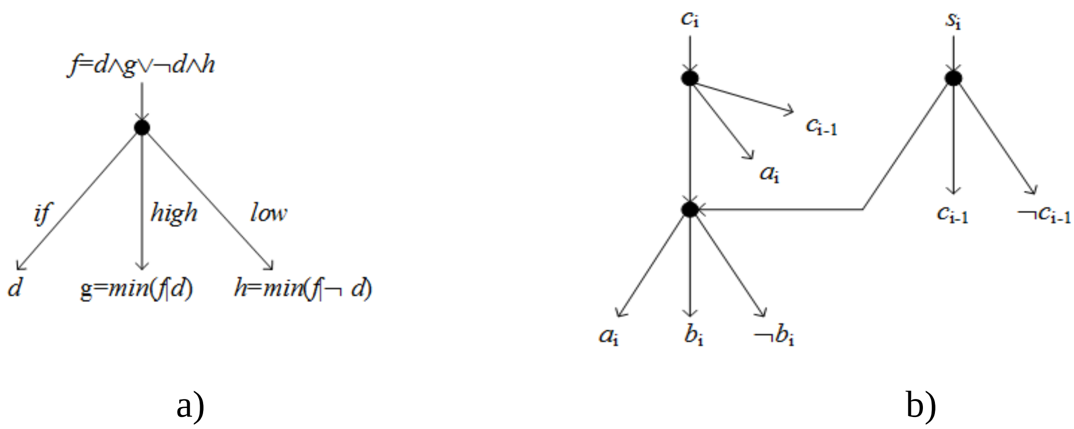


Figure 1 – If-decision diagram:
a – IFD's nonterminal node; b – IFD of 1-bit full adder

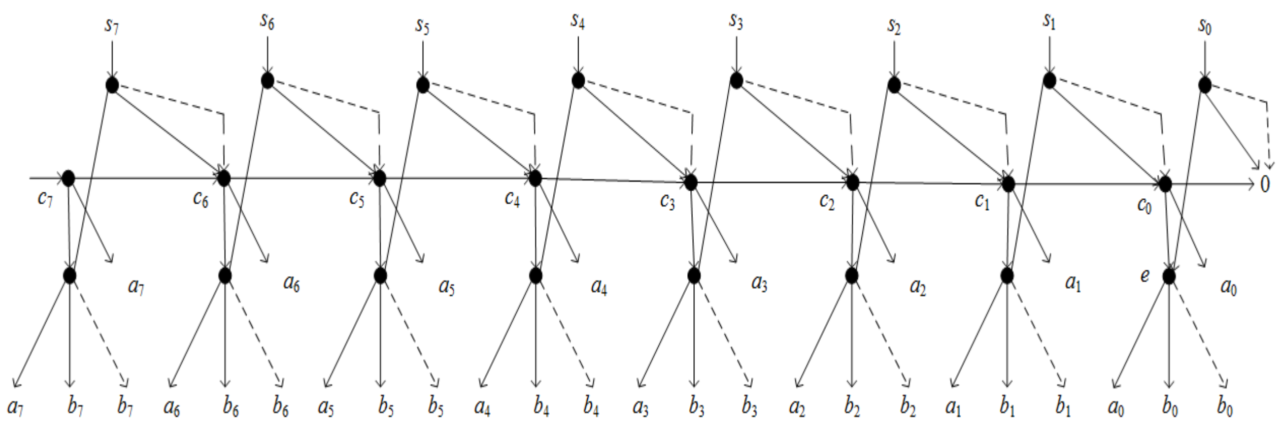
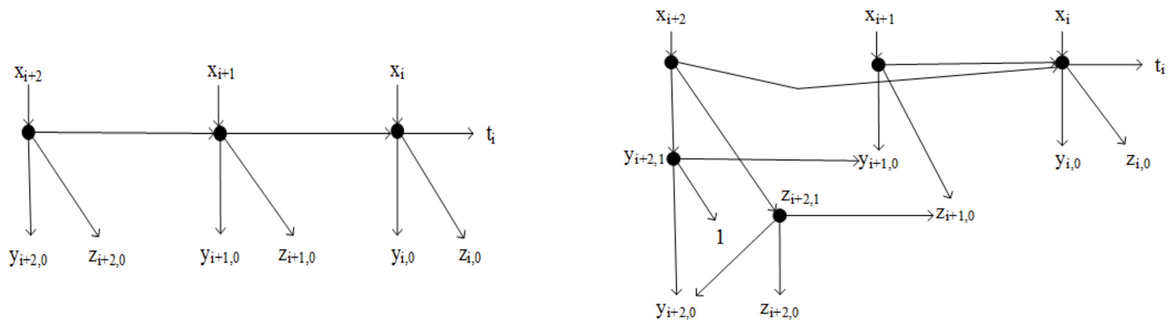


Figure 2 – IFD0 of ripple carry 8-bit adder (dash is complement)

In the paper we propose a formal method of stepwise transforming the IFD0 of a ripple-carry adder (fig. 2) to a parallel low fan-out IFD of logarithmic-depth adder. The ripple-carry adder is slow since it has the longest path of nine nodes. The transformation consists in multiple application of a transformation rule to long paths of the source or intermediate IFDs. The rule is a pair of IFDs: the IFD_{left} has three-node depth (fig. 3a), and the $\text{IFD}_{\text{right}}$ has a reduced two-node depth (fig. 3b). The rule splits the three-node path into four two-node paths. One application of the rule reduces the longest path length by one.



a) b)

Figure 3 – Diagram transformation rule:
a – IFD_{left} of one three-node path; b – $\text{IFD}_{\text{right}}$ of four two-node paths

Observing the IFD_{left} and $\text{IFD}_{\text{right}}$ of fig. 3, we can conclude that two nodes labeled by variable x_{i+1} are identical. The same concerns two nodes labeled by variable x_i. Two nodes which are labeled by x_{i+2}, are represented by different sub-diagrams. To prove their functional equivalence, we formulate an equation for each of them.

IFD1:

$$x_{i+2} = y_{i+2,0} \wedge z_{i+2,0} \vee \neg y_{i+2,0} \wedge \left(y_{i+1,0} \wedge z_{i+1,0} \vee \neg y_{i+1,0} \wedge x_i\right)$$

IFD2:

$y_{i+2,1} = y_{i+2,0} \vee y_{i+1,0}$ and

$$x_{i+2} = y_{i+2,1} \wedge \left(y_{i+2,0} \wedge z_{i+2,0} \vee \neg y_{i+2,0} \wedge z_{i+1,0}\right) \vee \neg y_{i+2,1} \wedge x_i =$$

$$= y_{i+2,0} \wedge z_{i+2,0} \vee \neg y_{i+2,0} \wedge y_{i+1,0} \wedge z_{i+1,0} \vee \neg y_{i+2,0} \wedge \neg y_{i+1,0} \wedge x_i =$$

$$= y_{i+2,0} \wedge z_{i+2,0} \vee \neg y_{i+2,0} \wedge \left(y_{i+1,0} \wedge z_{i+1,0} \vee \neg y_{i+1,0} \wedge x_i\right)$$

It is easy to see from the equations that the x_{i+2} node's semantics is the same in IFD_{left} and $\text{IFD}_{\text{right}}$. Therefore, the diagrams are functionally equivalent.

We can apply the transformation rule to the longest path of ripple carry adder IFD0 in different ways. Every application adds two nodes to the IFD. Different ways are possible for the rule application, which lead to different number of additional nodes in IFD. The longest path of IFD0 shown in fig. 2 consists of 9 nodes. Our first way of transformation applies the rule to the following node-sets:

$\{c_7, c_6, c_5\}$, $\{c_6, c_5, c_4\}$,$\{c_5, c_4, c_3\}$,$\{c_4, c_3, c_2\}$,$\{c_3, c_2, c_1\}$ and $\{c_2, c_1, c_0\}$.

It yields the IFD2 depicted in fig. 4. The 9-node path is split into two shorter 6-node paths: 1) s_7, c_6, c_4, c_2, c_0 and e; 2) c_7, c_5, c_3, c_1, c_0 and e. The depth reduction costs of the increase in the node count by 12. Nodes across the diagram bottom row have the highest fan-out of 4.

Our next transformation step is to split each of the 6-node paths into shorter paths without increasing the fan-out. It applies the rule to node-sets $\{c_7, c_5, c_3\}$ and $\{c_6, c_4, c_2\}$. Fig. 5 depicts the resulting IFD2. The attempt is not successful since the obtained diagram still has a path of 6 nodes.

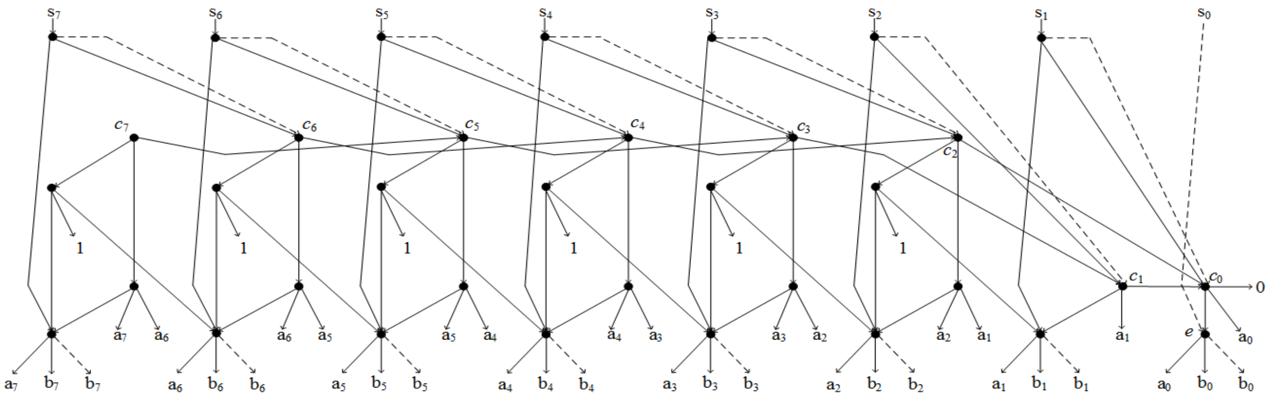


Figure 4 – Split of nine-node path of IFD0 to two six-node paths of IFD2

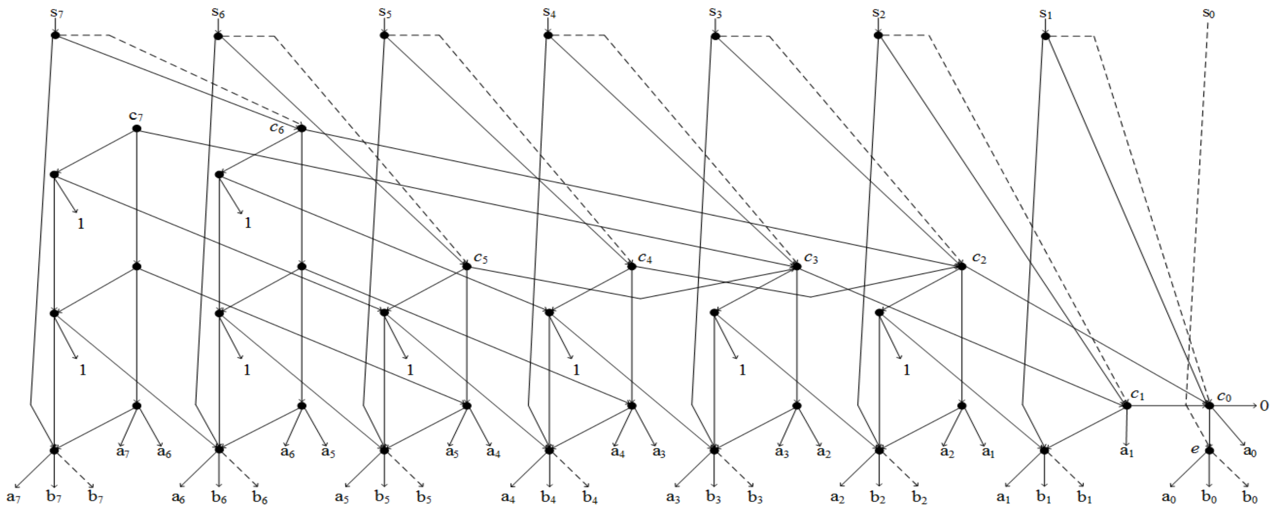


Figure 5 – Low-fan-out IFD2 of six-depth for look ahead eight-bit parallel adder

Tab. 1 reports experimental results for two methods of transforming the IFD of ripple-carry adder to logarithmic-depth IFDs of the parallel adder. The first method was proposed in [5–6], and the second one is based on the above-described transformation rule. The IFD depth, size, maximum and average fan-out depend on the adder bit-width, which varies in the range from 8 to 1024 bit. The key difference between the methods is the range of fan-out values. The fan-out grows exponentially from 6 to 514 for the first method. It keeps the constant fan-out value of 4 for all bit-width for the second method. The cost of such an advance of the second method is the increase in the IFD depth and size. Tab. 2 shows that the first method has a gain in smaller depth by up to 66.7 % and a gain in smaller size by up to 54.9 %. The second method has a gain in smaller fan-out by up to 128.5 times. It is very important when the circuit implementation technology dictates strict constraints on the fan-out. Tab. 3 reports results for the advanced second method, which impressively reduces the IFD3 size but yields a logarithmic growth of the fan-out depending on the adder bit-width.

Table 1 – Depth, size and fan-out of two IFDs of *n*-bit parallel adder

Adder width *n*, bit	High fan-out IFD1 (depth is 2 + log *n*)				Low constant fan-out IFD2 (depth is 2 log *n*)			
	Depth	Size	Maximum fan-out	Average fan-out	Depth	Size	Maximum fan-out	Average fan-out
8	5	33	6	1.69	6	35	4	1.69
16	6	81	10	1.88	8	95	4	1.91
32	7	193	18	1.99	10	243	4	2.04
64	8	449	34	2.07	12	599	4	2.12
128	9	1025	66	2.12	14	1435	4	2.19
256	10	2305	130	2.17	16	3359	4	2.23
512	11	5121	258	2.20	18	7715	4	2.27
1024	12	11265	514	2.23	20	17447	4	2.29

Table 2 – Comparison of high fan-out IFD1 to low fan-out IFD2 of *n*-bit parallel adder

Adder width *n*, bit	High fan-out IFD1			Low fan-out IFD2
	Depth, %	Size, %	Average fan-out, %	Maximum fan-out, times
8	20.0	6.1	0.00	1.5
16	33.3	17.3	1.60	2.5
32	42.9	25.9	2.51	4.5
64	50.0	33.4	2.42	8.5
128	55.6	40.0	3.30	16.5
256	60.0	45.7	2.76	32.5
512	63.6	50.7	3.18	64.5
1024	66.7	54.9	2.69	128.5

Table 3 – Depth and size of a n-bit parallel adder IFD3 with logarithmic fan-out

Adder width n, bit	Logarithmic fan-out IFD3 (depth is 1 + 2 log n)			
	Depth	Size	Maximum fan-out	Average fan-out
8	7	29	4	1.69
16	9	69	5	1.80
32	11	149	6	1.85
64	13	309	7	1.88
128	15	629	8	1.89
256	17	1269	9	1.89
512	19	2549	10	1.90
1024	21	5109	11	1.90

Conclusion. The paper has proposed a new method of the inference of parallel adder IFDs of logarithmic depth and low fan-out and has compared it to the alternative method proposed earlier. The main advantage of the new method is the obtaining of constant or logarithmic fan-out, which meets constraints raised by modern electronic, quantum and other implementation technologies. At 1024 bit-width, the constant fan-out ща 4 is 128.5 times less that the fan-out given by the known method. The cost of such an improvement is the increase of the IFD's depth (up to 66.7 %) and the increase of the IFD's size (up to 54.9 %).

References

1. Rosenberger, G. B. Simultaneous Carry Adder. U. S. Patent 2,966,305. – 1960 – 12–27 p.

2. P. M. Kogge, H. S. Stone. “A Parallel Algorithm for the Efficient Solution of a General Class of Recurrence Equations”. IEEE Transactions on Computers. 1973, C-22 (8): 786–793.

3. R. P. Brent, H. T. Kung A Regular Layout for Parallel Adders. IEEE Transactions on Computers. 1982, C-31, (3): 260–264.

4. N. Poornima, V. S. Kanchana Bhaaskaran. “Area Efficient Hybrid Parallel Prefix Adders”. Procedia Materials Science 10 (2015), 371 – 380 p.

5. Prihozhy, A. A. If-Decision Diagram Based Synthesis of Digital Circuits. Information Technologies for Education, Science and Business, Minsk, Belarus, 1999/ – 65–69 p.

6. Prihozhy, A. A. Synthesis of parallel adders from if-decision diagrams. System Analysis and Applied Information Science, 2020, No 2. – 61–70 p.

7. Lee, C. Y. Representation of Switching Circuits by Binary-Decision Programs / C.Y. Lee // Bell Systems Technical Journal, 1959, Vol. 38, No 4. – 985–999 p.

8. Bryant, R. Graph-based algorithms for Boolean function manipulation / R. Bryant // IEEE Trans. on Comp. 35, 1986. – 677–691 p.

9. Zulehner A., Niemann P., Drechsler R., Wille R. Accuracy and Compactness in Decision Diagrams for Quantum Computation. Design, Automation and Test in Europe Conference. – DATE, 2019. – 280–283 p.

10. Prihozhy, A. A. Synthesis of quantum circuits based on incompletely specified functions and if-decision diagrams, Journal of the Belarusian State University. Mathematics and Informatics, 2021, Volume 3. – 84–97 p.

11. Prihozhy, A. A. Modelling reversible circuits by if-decision diagrams. VIII Международная научно-техническая интернет-конференция «Информационные технологии в образовании, науке и производстве», 21–22 ноября 2020 года [Электронный ресурс]: Белорусский национальный технический университет. – С. 63–168.

12. Прихожий, А. А. Частично определенные логические системы и алгоритмы / А. А. Прихожий // Минск, БНТУ. – 2013. – 343 с.

13. Прихожий, А. А. Обобщение разложения Шеннона для частично определенных функций: теория и применение. Системный анализ и прикладная информатика. – 2013, № 1–2. – С. 6–11.

14. Прихожий, А. А. Новые разложения булевых функций по операции исключающее или в системах логического проектирования. Системный анализ и прикладная информатика. – 2014, № 1–3. – С. 9–16.

15. Prihozhy, A. A. If-Diagrams: Theory and Application. Proc. 7th Int. Workshop PATMOS’97. – UCL, Belgium, 1997. – 369 – 378 p.

16. Prihozhy A. A., Brancevich P. U. Parallel Computing with If-Decision-Diagrams Proc. Int. Conference PARELEC’98. – Poland, Technical University of Bialystok. – 1998. – 179–184 p.

17. Prihozhy A. A., Becker B. If-Decision Diagram Based Modeling and Synthesis of Incompletely Specified Digital Systems. Electronics and communications, Special Issue on Electronics Design, 2005. – 103–108 p.

18. Prihozhy, A. A. High-Level Synthesis through Transforming VHDL Models / A. A. Prihozhy // Chapter in Book “System-on-Chip Methodologies and Design Languages”. – Kluwer Academic Publishers. – 2001. – 135–146 p.