Pipelined Two-Operand Modular Adders

Maciej CZYŻAK, Jacek HORISZNY, Robert SMYK

Faculty of Electrical and Control Engineering, Gdansk University of Technology, G Narutowicza 11/12, 80-233

mczyzak@ely.pg.gda.pl, jhor@ely.pg.gda.pl, rsmyk@ely.pg.gda.pl

Abstract. *Pipelined two-operand modular adder* (*TOMA*) *is one of basic components used in digital signal processing* (*DSP*) *systems that use the residue number system* (*RNS*). *Such modular adders are used in binary/residue and residue/binary converters, residue multipliers and scalers as well as within residue processing channels. The structure of pipelined TOMAs is usually obtained by inserting an appropriate number of pipeline register layers within a nonpipelined TOMA structure. Hence the area of pipelined TOMAs is determined by the nonpipelined TOMA structure and by the total number of pipeline registers. In this paper we propose a new pipelined TOMA, that has a considerably smaller area and the attainable pipelining frequency comparable with other known pipelined TOMA structures. We perform comparisons of the area and pipelining frequency with TOMAs based on ripple carry adder (RCA), Hiasat TOMA and parallel-prefix adder (PPA) using the data from the very large scale of integration (VLSI) standard cell library.*

Keywords

Carry-lookahead adder, FPGA, modular adder, parallel-prefix adder, residue number system (RNS), ripple-carry adder, VLSI design

1. Introduction

Modular addition plays an important role in the implementation of digital signal processing systems that use the residue number system [1–4] as well as its derivatives like the quadratic residue number system (QRNS) [5] and modified quadratic residue number system (MQRNS) [6] for processing of complex signals. The RNS is a non-weighted integer number system that is determined by its base $\boldsymbol{B}=\{m_1, m_2, ..., m_n\}$ being the set of positive pairwise prime integers m_i , $i = 1, 2,.., n$. Each integer $X \in \boldsymbol{Z}_M$, $M = \prod_{i=1}^{n} m_i$ and can be represented as $X \leftrightarrow (x_1,x_2,...,x_n) = = (|X|_{m_1},|X|_{m_2},...,|X|_{m_n})$ with $x_i \in \boldsymbol{Z}_{m_i}$. This mapping is the bijection and for $X, Y \in \boldsymbol{Z}_M$ and for $x_i, y_i \in \boldsymbol{Z}_{m_i}$, we have $z_i = |x_i \otimes y_i|_{m_i}$,where $\otimes$ denotes addition, subtraction or multiplication.

The reverse conversion from the RNS to a weighted system can be performed using the Chinese remainder theorem (CRT) [1], [2] or the mixed-radix system (MRS) [1], [2]. The main advantage of the RNS comes from the fact that addition, subtraction and multiplication are carry-free and can be performed without carries between individual positions of the number. The principal advantage of the RNS with respect to the high-speed DSP is due to the replacement of large multipliers that limit the pipelining frequency, by small multipliers modulo m_i. If their binary size $l = \lceil (\log_2 m_i) \rceil$, where $\lceil \bullet \rceil$ denotes rounding off to an integer, does not exceed six bits, multiplications by a constant can be performed by look-up with small ROMs or using combinatorial networks. General multiplications are also easier to perform because their standard realizations are small or segmentation of operands can be used for the combinatorial realization. It is worth mentioning that moduli with $l < 7$ may provide for the dynamic ranges over 90 bits [7]. The additional advantage of the RNS is the possibility of reducing power dissipation in CMOS circuits which is due to the lower switching activity and reduction of supply voltages [9]. The RNS has found numerous applications in the DSP, for example, in FIR filters [8–11], FFT processors [12], digital downconversion [13] and image processing [14], [15].

Generally TOMAs can be divided into two main categories determined by the type of the modulus. TOMAs for moduli akin to 2^n represent the first category and those for generic moduli the other. There are several works in the literature that consider the TOMA design.

Banerji [16] presented a look-up approach, Agrawal and Rao [17] proposed a TOMA for moduli of the form $(2^n + 1)$ based on binary adders. Soderstrand [18] introduced a hybrid approach based on look-up table along with the binary adder. Bayoumi and Jullien [19] described TOMAs using the table approach and binary adders approach. Dugdale [20] demonstrated an implementation of TOMAs that used binary adders, Piestrak [21] proposed a TOMA based on the carry-save adder (CSA) and two binary adders. Zimmermann [22] introduced modulo $(2^n \pm 1)$ adders based on parallel prefix-architecture (PPA). Hiasat [23] proposed a TOMA with the reduced area based on the carry-look-ahead (CLA) adder. Also a novel delay-power-area-efficient approach to the TOMA design was given by Patel et al. [24]. Their TOMA structure was based on the cascaded connection of the modified carry-save adder

(CSA) and reduced carry-propagate adder (CPA). The used CPA designs included ELM [25], Kogge-Stone [26] and Ladner Fischer [27] PPA.

In this paper we propose a new TOMA based on a modified CLA adder. This TOMA has the smaller area than other considered TOMAs and allows to derive a new pipelined TOMA that is better than other known pipelined TOMAs in terms of the area and the number of stages of pipeline registers. We shall show the structure of the new pipelined TOMA and, for comparison, TOMAs based on the RCA, PPA in the Brent-Kung form [28] and Hiasat TOMA [23]. Comparisons are made using the data from the VLSI standard cell library. We shall compare structures of individual TOMAs in terms of area, delay and pipelining frequency with the use of the additive method. The method uses summation of areas of individual components expressed in gate equivalents (GE), where 1 GE is the area of the NAND with the fan-out = 1 for the given standard cell library. The propagation delay of an individual element is taken as the worst case delay for all possible inputs. The analysis relies upon the established 130 nm Samsung standard cell library STDH150 [29]. Calculations of areas and delays of individual components are practically technology independent and they can be scaled down for VLSI technologies such as 28 nm or 22 nm. Therefore we may therefore suppose that for comparison of individual digital structures, the assumed technology will give sufficient and dependable information. The paper has the following structure: in Sec. 2 we review the basic TOMA structures, in Sec. 3 we consider the TOMA-RCA, and in Sec. 4 Hiasat TOMA, in Sec. 5 we present the TOMA based on the PPA adder and finally in Sec. 6 a new TOMA. In each section we analyze a nonpipelined and pipelined form.

2. Basic TOMA Structures Based on Binary Adders

In this section we shall shortly describe the basic known TOMA structures that use exclusively binary adders in series and which therefore may be the most suitable for transformation to the pipelined form and not those that use two parallel adders as in [21]. Two-operand modular addition for small m, $\lceil \log m \rceil \le 6$ can be implemented by using the ROM ($2^{2\cdot\lceil \log m \rceil} \times \lceil \log m \rceil$), but such approach remarkably reduces the attainable pipelining frequency.

The TOMA computes $r_m = |X+Y|_m$, where r_m is the least nonnegative remainder from the division $X+Y$ by the modulus m. Assuming $Z = 2^{\lceil \log m \rceil} - m$, the computation can be also expressed as

$$r = \begin{cases} |X+Y+Z|_{2^{\lceil \log m \rceil}} & \text{if } X+Y+Z \ge 2^{\lceil \log m \rceil} \\ X+Y & \text{otherwise} \end{cases} \quad (1a)$$

In Fig. 1 to 3 three basic TOMA structures are shown Bayoumi-Jullien, Hiasat, and Piestrak.

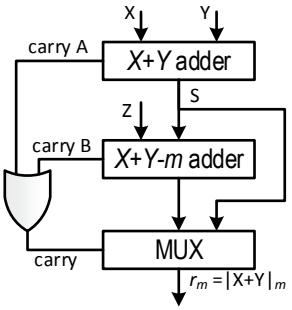

Fig. 1. Bayoumi-Jullien TOMA [19].

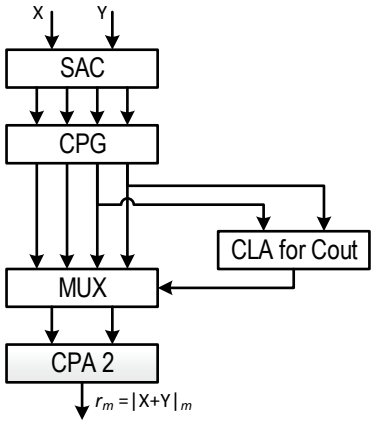

Fig. 2. Hiasat TOMA [23].

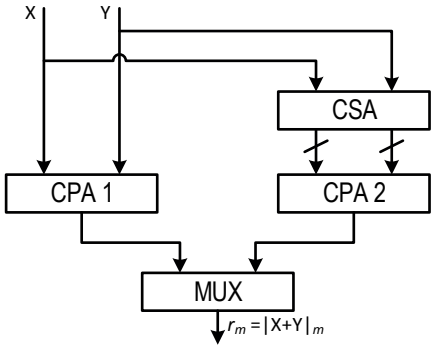

Fig. 3. Piestrak TOMA [21].

We shall shortly analyze the operation of the Bayoumi- Jullien TOMA (Fig. 1) because this structure will be the basis for the design of selected TOMAs. The binary adder in the first stage of this TOMA computes $X+Y$, whereas the second adder $X+Y-m$. The output of the TOMA is selected using $carry = carryA \vee carryB$. For $X+Y<m$, $carry = 0$ and $r_m = X+Y$, whereas for $X+Y \ge m$, $carry = 1$ and $r_m = X+Y-m$.

3. TOMA-RCA

By way of introduction we shall consider the realization of the Bayoumi-Jullien TOMA based on the RCA. In order to obtain a pipelined structure, layers of pipeline registers consisting of flip-flops (FFs) have to be inserted between individual adders as shown in Fig. 5.

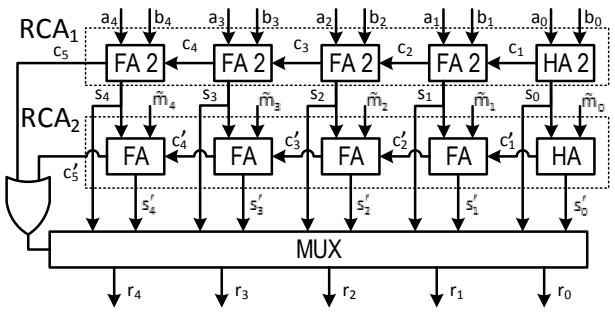

Fig. 4. Bayoumi-Jullien TOMA based on the RCA.

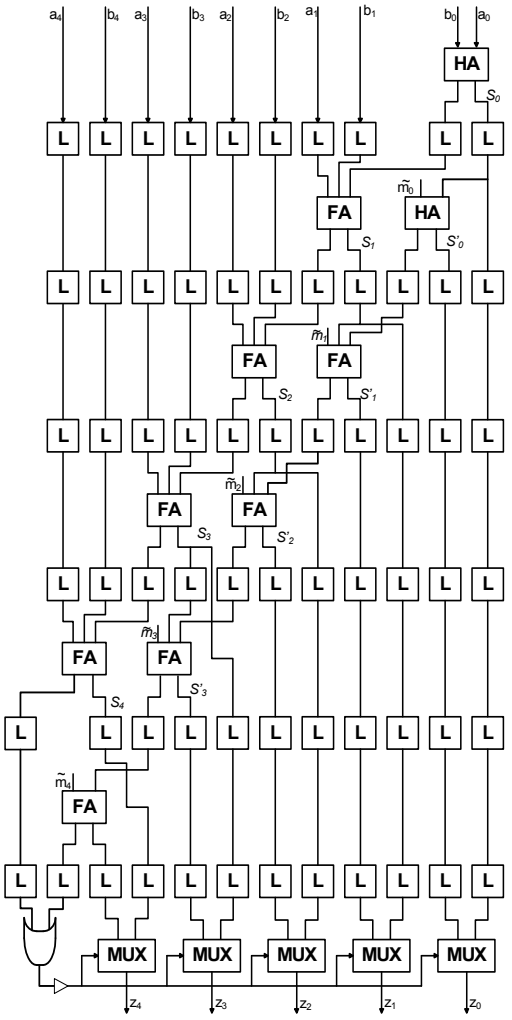

Fig. 5. Pipelined TOMA based on Bayoumi-Jullien TOMA and the RCAs.

In the following we shall analyze the area of the TOMA-RCA expressed in GE, the delay and the maximum attainable pipelining frequency. The area will be estimated using the areas of the individual components from STDH150, the delay for a nonpipelined structure will be evaluated by using the maximum delays for the individual components. In order to estimate the pipelining frequency a structure is divided into balanced layers with respect to the delay and the maximum pipelining frequency is obtained as the inverse of the sum of the delay of the slowest layer and the FF delay.

A. Nonpipelined 5-bit TOMA-RCA area

This area of 5-bit TOMA-RCA can be expressed in the following manner:

$$A_{TOMA\_RCA}=A_{HAd2}+4\cdot A_{FAd2}+A_{HAd1}+4\cdot A_{FAd1} +A_{OR2d1}+A_{NID6}+5\cdot A_{MX2d1}\,. \quad (1)$$

The indices of the individual components come from STDH150. The data of individual components is given in Appendix A. After inserting these data into (1) we obtain $A_{TOMA\ RCA}=98.68\text{GE}$. The area given by (1) does not depend upon the form the of the two's complement system (TCS) representation of $-m$, $\tilde{m}=(1,...,\tilde{m}_4,\tilde{m}_3,\tilde{m}_2,\tilde{m}_1,\tilde{m}_0)$. The particular form of this representation allows to reduce the area for the given modulus. For example, if $m_i=0$, the HA reduces to single connection and for $m_i=1$ to one connection and to one inverter. For the FA and $m_i=0$, we have one XOR gate and a single AND gate, and for $m_i=1$ one OR gate and exclusive NOR. For $m=29$ and $\tilde{m}=(1,...,0,0,0,1,1)$, we obtain $A_{TOMA\text{-}RCA}=81.68GE$.

B. Nonpipelined 5-bit TOMA-RCA delay

We shall estimate the delay of the structure of Fig. 4 taking into consideration individual delays of signals inside individual HAs and FAs.

The delay of the 5-bit TOMA-RCA can be expressed as

$$t_{TOMA-RCA}=\max(t_{s_4},t_{s_4'},\max(t_{c_5},t_{c_5'}))+t_{OR2d1}+t_{MX2d1}. \quad (2)$$

The delay for s_4 and c_5 bits can be calculated as

$$t_{s_4}=\max(t_{HAd2\_ACO},t_{HAd2\_BCO})+3\cdot t_{FAd2\_CICO}+t_{FAd2\_CIS},$$

$$t_{c_5}=\max(t_{HAd2\_ACO},t_{HA\_BCO})+4\cdot t_{FAd2\_CICO}.$$

In order to compute c_5', we shall first calculate t_{c_i} and $t_{c_i'}$, $i=1,\ 2,\ 3,\ 4$. We have

$$t_{c_1}=\max\left(t_{HAd2\_ACO},t_{HAd2\_BCO}\right), \quad (3a)$$

$$t_{c_i}=t_{c_{i-1}}+t_{FAd2\_CICO},\ i=2,3,4,5. \quad (3b)$$

Consequently

$$t_{c_1'}=\max\left(t_{HAd2\_AS},t_{HAd2\_BS}\right)+t_{HAd1\_BCO}, \quad (4a)$$

$$t_{c_i'}=\max\left(t_{c_{i\ 1}}+t_{FAd2\_CIS}+t_{FAd2\_BCO},t_{c_{i\ 1}'}+t_{FAd1\_CICO}\right), \quad i=2,3,4,5, \quad (4b)$$

and $t_{s_4'}=t_{c_4'}+t_{FAd1\_CIS}$. (4c)

Example 1. Computation of 5-bit TOMA-RCA delay for components from the STDH150.

We shall first compute t_{s_4} and t_{c_5} as

$$t_{s_4} = \max(0.092\,\text{ns}, 0.074\,\text{ns}) + 3 \cdot 0.089\ \text{ns} + 0.150\,\text{ns} = 0.509\,\text{ns},$$

$$t_{c_5} = \max(0.092\,\text{ns}, 0.074\,\text{ns}) + 4 \cdot 0.089\,\text{ns} = 0.448\,\text{ns}.$$

Before we can compute c_5', we have to determine c_i, $i = 1, 2, 3, 4$. We have

$$t_{c_1} = \max(0.092\ \text{ns}, 0.074\ \text{ns}) = 0.092\ \text{ns},$$

$$t_{c_2} = 0.181\ \text{ns},$$

$$t_{c_3} = 0.270\ \text{ns},$$

$$t_{c_4} = 0.359\ \text{ns}.$$

Subsequently we obtain

$$t_{c_1'} = \max(0.092\,\text{ns}, 0.074\,\text{ns}) + 0.055\,\text{ns} = 0.147\,\text{ns},$$

$$t_{c_2'} = \max(0.092\,\text{ns} + 0.102\,\text{ns} + 0.152\,\text{ns}, 0.147\,\text{ns} + 0.083\,\text{ns}) = \max(0.346\,\text{ns}, 0.230\,\text{ns}) = 0.346\,\text{ns},$$

$$t_{c_3'} = \max(0.181\,\text{ns} + 0.102\,\text{ns} + 0.143\,\text{ns}, 0.346\,\text{ns} + 0.083\,\text{ns}) = \max(0.426\,\text{ns}, 0.429\,\text{ns}) = 0.429\,\text{ns},$$

$$t_{c_4'} = \max(0.270\,\text{ns} + 0.102\,\text{ns} + 0.143\,\text{ns}, 0.429\,\text{ns} + 0.083\,\text{ns}) = \max(0.515\,\text{ns}, 0.512\,\text{ns}) = 0.429\,\text{ns},$$

$$t_{c_5'} = \max(0.359\,\text{ns} + 0.102\,\text{ns} + 0.143\,\text{ns}, 0.515\,\text{ns} + 0.083\,\text{ns}) = \max(0.604\,\text{ns}, 0.598\,\text{ns}) = 0.604\,\text{ns},$$

$$t_{s_4'} = 0.515\,\text{ns} + 0.095\,\text{ns} = 0.601\,\text{ns}.$$

Finally, we may determine the TOMA-RCA delay as

$$t_D^{TOMA-RCA} = \max(0.509\ \text{ns}, 0.601\ \text{ns}, 0.604\ \text{ns} + 0.065\ \text{ns}) + 0.078\ \text{ns} = 0.747\ \text{ns}.$$

C. The area of pipelined 5-bit TOMA-RCA

In Fig. 5 a pipelined form of the RCA-TOMA is presented. Six flip-flops stages are used with 66 flip-flops. The area is the sum of the nonpipelined 5-bit TOMA-RCA area and the area of pipeline registers. In this case these registers require $n_s = 66$ FFs. Thus the area can be expressed as

$$A_{TOMA\_RCA\_p} = A_{TOMA\_RCA} + n_s \cdot A_{FF}\ . \tag{5}$$

As A_{FF} we shall use the area of the flip-flop FD1Q, A_{FD1Q} from STDH150. For the structure from Fig. 5 we receive $A_{TOMA\text{-}RCAp} = 472.9$ GE.

D. Pipelined 5-bit RCA-TOMA pipelining rate

In order to design a pipelined structure of a TOMA, we have to decompose its nonpipelined structure into a certain number of layers and place pipeline registers between them. The decomposition is, to certain extent, arbitrary. The lower limit of the number of layers is two and the upper limit is determined by a delay of the component that we treat as indivisible. The minimum pipelining rate is approximately the sum of the delay of the layer with the maximum delay and the delay of the pipeline register. In this case we have assumed that after each FA or HA a register layer is placed and the OR gate and the MUXs are in the same layer. Hence we may evaluate the maximum delay of the layer as

$$t_{LD}^{TOMA\_RCA} = \max(t_{FAd1}, t_{OR2d1} + t_{MX2d1}) + t_{FD1Q} \tag{6}$$

where t_{FD1Q} is the maximum delay of the flip-flop.

Using the data from the STDH150, we may evaluate a theoretical maximum pipelining frequency as

$$f_{PF\_\max}^{TOMA\_RCA} = 1/(\max(0.143\,\text{ns}, 0.065\,\text{ns} + 0.078\,\text{ns}) + 0.094\,\text{ns}) = 1/0.237\,\text{ns} = 4.22\ \text{GHz}.$$

4. Hiasat TOMA

In the following we shall examine the results of transforming the Hiasat TOMA which requires the smallest hardware amount among known TOMAs. This TOMA consists of the serial connection of five units: the sum-and-carry (SAC), the carry propagate and generate (CPG), CLA for c_{OUT}, multiplexer (MUX), CLA and Summation (CLAS). The SAC is composed of HAs and HALs (the modified HAs in [23]). The SAC performs

$$s_i = x_i \oplus y_i \oplus z_i\,, \tag{7}$$

$$c_{i+1} = x_i \cdot y_i + x_i \cdot z_i + y_i \cdot z_i\,, \tag{8}$$

for the individual bits of $X + Y$, and $X + Y - m$, with the assumption that TCS representation of $-m$ without the sign bit is $(z_{n-1},...,z_0)$ with $n = 5$. Regarding that $z_i = 0$ or $z_i = 1$, the HAL is obtained that implements

$$A_i = x_i \oplus y_i\,, \tag{9a}$$

$$\hat{A}_i = \overline{x_i \oplus y_i}\,, \tag{9b}$$

$$B_{i+1} = x_i \cdot y_i\,, \tag{9c}$$

$$\hat{B}_{i+1} = x_i + y_i\,. \tag{9d}$$

As $(z_{n-1},...,z_0)$ may have w bits for which $z_i = 0$ and $n - w$ bits for which $z_i = 1$. Hence the SAC has w HAs and $n - w$ HAL cells. The CFG computes the carry generate and carry propagate vectors as in the standard CLA $P_i = A_i \oplus B_i$, $G_i = A_i \quad B_i$ and $p_i = \hat{A}_i \oplus \hat{B}_i$, $g_i = \hat{A}_i \cdot \hat{B}_i$. This unit has at most $2k - 2$ HAs. In the CLAS p_i and g_i are used to compute c_{OUT}, that controls the selection of $X + Y$ or $X + Y - m$. Regarding that $c_0 = 0$, $g_0 = 0$, c_{OUT} can

be computed for the five-bit Hiasat adder as

$$c_{OUT} = B_5 + g_4 + g_3 \cdot p_4 + g_2 \cdot p_3 \cdot p_4 + g_1 \cdot p_2 \cdot P_3 \cdot P_4 . \quad (10)$$

The following stage, MUX selects using c_{OUT} the carry's and generate's $p_i^{'} = p_i$ or $p_i^{'} = P_i$ and $g_i^{'} = g_i$ or $g_i^{'} = G_i$, $i = 0, 1,..., 4$.

The final stage, the five-bit CLA adder computes the carries

$$c_1 = g_0^{'}, \quad (11)$$

$$c_2 = g_1^{'} + g_0^{'} \cdot p_1^{'}, \quad (12)$$

$$c_3 = g_2^{'} + g_1^{'} \cdot p_2^{'} + g_0^{'} \cdot p_1^{'} \cdot p_2^{'}, \quad (13)$$

$$c_4 = g_3^{'} + g_2^{'} \cdot p_3^{'} + g_1^{'} \cdot p_2^{'} \cdot p_3^{'} + g_0^{'} \cdot p_1^{'} \cdot p_2^{'} \cdot p_3^{'} . \quad (14)$$

In the next step the sum bits are calculated as

$$s_i^{'} = c_i \oplus p_i^{'}, \; i = 0, 1, 2, 3, 4. \quad (15)$$

First we shall determine the area for components of the Hiasat five-bit TOMA and then the area for $m = 29$.

A. 5-bit Hiasat TOMA area

The area of the five-bit Hiasat TOMA can be computed as follows

$$A_{TOMA\_Hiasat} = A_{SAC\_5} + A_{CFG\_5} + A_{CLA\_Cout\_5} + A_{MUX\_5} + A_{CLAS\_5}. \quad (16)$$

The areas of the individual blocks from (16) can be expressed as:

$$A_{SAC\_5} = 2 \cdot A_{HAd1} + A_{HAd2} + A_{HAL}, \quad (17)$$

with

$$A_{HAL} = A_{OR2d1} + A_{AND2d1} + A_{XOR2d1} + A_{IVd1}. \quad (18)$$

In general, the area of the CFG_5 can be expressed as

$$A_{CFG\_5} = 5 \cdot A_{HAd2} + A_{HAd1}, \quad (19)$$

$$A_{CLA\_out\_5} = A_{AND2d1} + A_{AND3d1} + A_{AND4d1} + A_{OR5d1} + A_{NID6}, \quad (20)$$

$$A_{MUX\_5} = A_{MX2d1} + A_{MX2d2} + 3 \cdot A_{MX4d1}. \quad (21)$$

The CLAS block consists of the five-bit Propagate-Generate Unit (PGU_5), Carry-Generate Unit (CGU_5) and Summation Unit (SU_5). Its hardware amount can be estimated as

$$A_{CLAS\_5} = A_{CGU\_5} + A_{SU\_5}, \quad (22)$$

with the fan-outs 1, 3, 3, 4, 2. We get

$$A_{CGU\_5} = A_{AND2d1} + A_{OR2d1} + A_{AND2d2} + A_{AND3d1} + A_{OR3d1}, \quad (23)$$

and

$$A_{SU\_5} = 5A_{XOR2d1} = 15.0 \text{ GE}. \quad (24)$$

Example 2. Area of the five-bit Hiasat TOMA for $m = 29$.

The TCS representation of $(-m)$ is equal to 100011, hence $w = 3$, and $k - w = 2$ (the sign bit is excluded). Thus we obtain

$$A_{SAC\_5} = 2A_{HAd1} + A_{HAd2} + 2A_{HAL} = 2 \;\; 4.67 \text{ GE} + 5.67 \text{ GE} + 7.34 \text{ GE} = 22.350 \text{ GE},$$

$$A_{CFG\_5} = 5 \cdot A_{HAd2} + A_{HAd2} = 5 \cdot 5.67\,\text{GE} + 4.67\,\text{GE} = 33.02\,\text{GE},$$

$$A_{CLA\_Cout\_5} = 1.67 \text{ GE} + 2 \text{ GE} + 2.33 \text{ GE} + 3.33 \text{ GE} + 3.67 \text{ GE} = 13 \text{ GE},$$

$$A_{MUX\_5} = 3 \text{ GE} + 3.33 \text{ GE} + 3 \cdot 6.33 \text{ GE} = 25.32 \text{ GE},$$

$$A_{CGU\_5} = 1.67\text{GE} + 1.67\,\text{GE} + 2\,\text{GE} + 2\,\text{GE} + 2\,\text{GE} + 1.67\,\text{GE} + 2\,\text{GE} + 2.33\,\text{GE} + 3\,\text{GE} = 18.34\,\text{GE},$$

$$A_{SU\_5} = 5 \cdot 3\,\text{GE} = 15\,\text{GE},$$

$$A_{CLAS\_5} = 18.34\,\text{GE} + 15\,\text{GE} = 33.34\,\text{GE}.$$

In effect we obtain the area of the five-bit Hiasat TOMA as

$$A_{TOMA\_Hiasat\_5} = 22.35\,\text{GE} + 33.02\,\text{GE} + 13\,\text{GE} + 25.32\,\text{GE} + 33.34\,\text{GE} = 127.03\,\text{GE}.$$

B. 5-bit Hiasat TOMA delay

The Hiasat five-bit TOMA delay, t_H can be expressed as

$$t_D^{Hiasat-TOMA} = t_{HAL} + t_{HAd2} + t_{AND4d1} + t_{OR5d1} + t_{NID6} + t_{MX2d4} + t_{AND4d1} + t_{OR2d1} + t_{OR4d1} + t_{XORd1} = 0.119\,\text{ns} + 0.092\,\text{ns} + 0.082\,\text{ns} + 0.094\,\text{ns} + 0.054\,\text{ns} + 0.092\,\text{ns} + 0.082\,\text{ns} + 0.090\,\text{ns} + 0.076\,\text{ns} + 0.090\,\text{ns} = 0.871\,\text{ns},$$

with

$$t_{HAL} = t_{XOR2d1} + t_{IVd1} = 0.090 \text{ ns} + 0.029 \text{ ns} = 0.119 \text{ ns}.$$

C. Pipelined 5-bit Hiasat TOMA area

The area of the Hiasat pipelined 5-bit TOMA can be expressed as

$$A_{TOMA\_Hiasat\_p} = A_{TOMA\_Hiasat} + n_h A_{FF} \quad (25)$$

where n_h is a number of flip-flops used in pipeline registers. For example, for the structure from Fig. 6 we obtain

$$A_{TOMA\_Hiasat\_p} = 127.03\,\text{GE} + 64 \cdot 5.67\,\text{GE} = 489.91\,\text{GE}.$$

D. Pipelining frequency of pipelined 5-bit Hiasat TOMA

In Fig. 6, a pipelined form of the Hiasat TOMA is presented. Five pipeline register stages are used with 58 flip-flops.

In this case we have adopted a decomposition into six layers that leads to a balanced structure. In order to evaluate the maximum pipelining frequency we shall calculate delays of the adopted individual layers. The maximum pipelining frequency will depend on the delay of the layer with the maximum delay and the delay of the assumed pipeline register. These layers have the following delays:

layer 1 $t_D^{L1,H}: t_{HAL} = 0.119\,\text{ns}$,

layer 2 $t_D^{L2,H}: t_{HAd1} = 0.088\,\text{ns}$,

layer 3 $t_D^{L3,H}: t_{AND4d1} + t_{OR5d1} = 0.176\,\text{ns}$,

layer 4 $t_D^{4,H}: t_{MX2d1} + t_{NID6} = 0.132\,\text{ns}$,

layer 5 $t_D^{L5,H}: t_{AND4d1} + t_{XOR2d1} = 0.172\,\text{ns}$,

layer 6 $t_D^{L6,H}: t_{XOR2d1} + t_{OR4d1} = 0.166\,\text{ns}$.

Using $t_D^{L3,H}$ as the maximum layer delay, we may evaluate the maximum pipelining frequency as

$$f_{PF\_\max}^{TOMA\_Hiasat} = 1/(0.176\,\text{ns} + 0.094\,\text{ns}) = 1/0.27\,\text{ns} = 3.7\,\text{GHz}.$$

5. PPA-based TOMA

As the next structure we shall consider the TOMA based on a PPA. As the PPA the Brent-Kung (BK) [28] adder has been selected. The Brent-Kung TOMA can be relatively easy transformed to the pipelined form, moreover the use of the Brent-Kung PPA allows one to simplify the adder used in the second stage when one of addends is a constant. The prefix operator ϕ is defined as

$$(g, p) = (g', p')\,\phi\,(g'', p''), \tag{26}$$

where

$$g = g'' + g' \cdot p'', \tag{27a}$$

$$p = p' \cdot p''. \tag{27b}$$

The block that implements (27a-b) will be denoted as BK_i. Subsequently we shall analyze the area and delay of the TOMA based on two BK adders.

The area of the TOMA BK A_{TOMA_BK} can be expressed as

$$A_{TOMA\_BK} = A_{BK} + A_{BK-m} \tag{28}$$

where A_{BK}, $A_{BK\text{-}m}$ represent the area of the *BK* adder and the modified *BK-m* adder that subtracts *m*, respectively.

A. The area of BK adder

A_{BK} can be calculated as

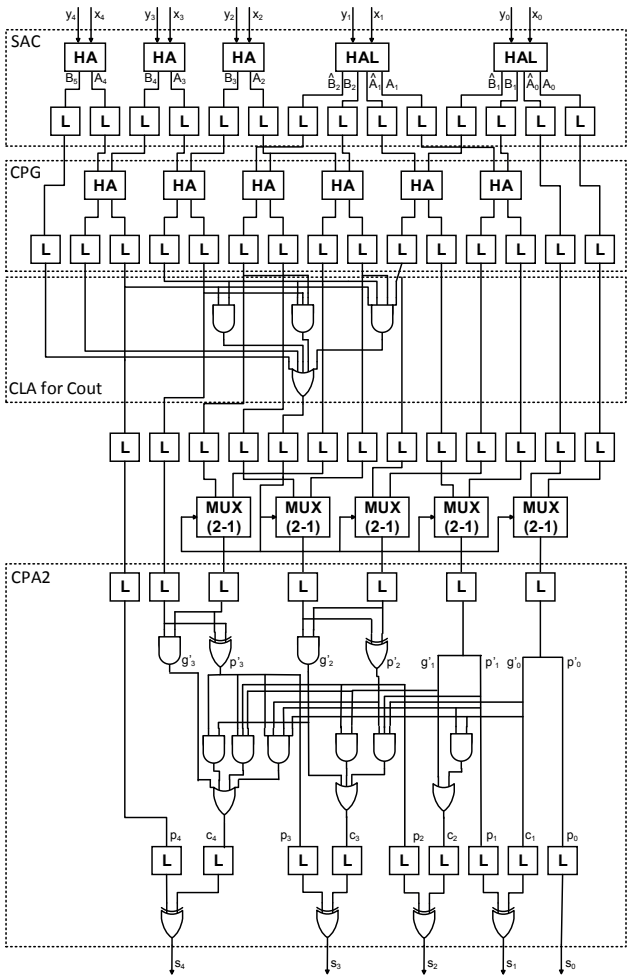

Fig. 6. Pipelined TOMA based on Hiasat TOMA.

$$A_{BK} = 4 \cdot A_{HAd2} + A_{HAd1} + A_{BK0} + A_{BK1} + A_{BK2} + A_{BK3} + \; + A_{BK4} + 4 \cdot A_{XOR2d1}. \tag{29}$$

The area of the first two terms is

$$4 \cdot A_{HAd2} + A_{HAd1} = 25.99\ \text{GE}. \tag{30}$$

After transforming the logic functions used for the realization of individual adders in (29), we receive the following areas

$$A_{BK_0} = A_{IVd1} + A_{NAND2d1} + A_{NAND2d2} + A_{AND2d2} = 6\ \text{GE}, \tag{31a}$$

$$A_{BK_1} = A_{IVd1} + A_{NAND2d1} + A_{NAND2d1} + A_{AND2d1} = 4.67\,\text{GE}, \tag{31b}$$

$$A_{BK_2} = A_{IVd1} + A_{NAND2d1} + A_{NAND2d1} = 3\ \text{GE}, \tag{31c}$$

$$A_{BK_4} = A_{IVd1} + A_{NAND2d1} + A_{NAND2d2} + A_{AND2d2} = 6\ \text{GE}, \tag{31d}$$

$$A_{BK_4} = A_{BK_2} = 3\ \text{GE}. \tag{31e}$$

Using (29), (30) and (31a-e) we obtain

$$A_{BK} = 62.02\ \text{GE}. \tag{31f}$$

B. The delay of BK adder

The BK adder delay can be expressed as

$$t_{BK} = t_{HA2} + \max(t_{BK_0}, t_{BK_1}) + \max(t_{BK_2}, t_{BK_3}) + t_{BK_4} + t_{XOR2d1}, \quad (32)$$

where

$t_{BK_0} = 2 \cdot t_{NAND2d1}$, $t_{BK_1} = t_{NAND2d1} + t_{NAND2d2}$, $t_{BK_2} = t_{BK_1}$, $t_{BK_3} = t_{BK_1}$, $t_{BK_4} = t_{BK_0}$.

Using the data from the STDH150, we have $t_{BK_0} = 0.074$ ns and $t_{BK_1} = 0.068$ ns.

Finally we obtain

$$t_{BK} = 0.092 \text{ ns} + 0.074 \text{ ns} + 0.068 \text{ ns} + 0.074 \text{ ns} + 0.09 \text{ ns} = 0.398 \text{ ns}.$$

C. The area of BK-m adder

The form of the first layer of the BK-*m* adder depends on the TCS representation of – m, $\tilde{m}$. We shall analyze the prefix operator computation for a pair of bits ($\tilde{m}_i$, $\tilde{m}_{i+1}$).

(27a-b) can be expressed as

$$g_{\tilde{m}} = s_{i+1} \cdot \tilde{m}_{i+1} + s_i \cdot \tilde{m}_i \cdot (s_{i+1} \oplus \tilde{m}_{i+1}), \quad (33a)$$

$$p_{\tilde{m}} = (s_{i+1} \oplus \tilde{m}_{i+1}) \cdot (s_i \oplus \tilde{m}_{i1}). \quad (33b)$$

For individual combinations of $(\tilde{m}_i, \tilde{m}_{i+1})$ we get

$(\tilde{m}_i, \tilde{m}_{i+1}) = (0,0)$ $g_{\tilde{m}} = 0$ and $p_{\tilde{m}} = s_i \cdot s_{i+1}$,
$(\tilde{m}_i, \tilde{m}_{i+1}) = (0,1)$ $g_{\tilde{m}} = \bar{s}_i \cdot s_{i+1}$ and $p_{\tilde{m}} = s_i \cdot s_{i+1}$,
$(\tilde{m}_i, \tilde{m}_{i+1}) = (1,0)$ $g_{\tilde{m}} = s_i$ and $p_{\tilde{m}} = s_i \cdot \bar{s}_{i+1}$,
$(\tilde{m}_i, \tilde{m}_{i+1}) = (1,1)$ $g_{\tilde{m}} = s_{i+1} + s_i \cdot \bar{s}_{i+1}$ and $p_{\tilde{m}} = \bar{s}_0 \cdot \bar{s}_1$.

The HA's become reduced, for we have $g_i = 0$, and the XOR gate that computes p_i, is reduced to the direct connection, i.e. $p_i = s_i$. For $\tilde{m}_i = 1$, $g_i = s_i$, the XOR gate that computes p_i .becomes an inverter, i.e. $p_i = \bar{s}_i$. The form of $g_{\tilde{m}}$ and $p_{\tilde{m}}$ influences the form of BK_0 and BK_1.

Next we shall analyze the BK-*m* adder for $m = 29$ in order to have a comparison with the adder presented by Hiasat [23]. The TCS representation of $m = 29$ has the form 100011, then for HA_0, g_0 - connection, p_0 - inversion, for HA_1 g_1 - connection, p_1 - inversion, for HA_2 $g_2 = 0$, p_2- connection, for HA_3 $g_3 = 0$, p_3 - connection, for HA_4 $g_4 = 0$, p_4 - connection.

Moreover, regarding that $\tilde{m}_0 = 1$ and $\tilde{m}_1 = 1$, we may transform BK_0, to obtain $BK_{0\text{-}m}$ as

$$g_{\tilde{m},BK_0} = s_{i+1} + s_i \cdot \bar{s}_{i+1} \quad (34a)$$

and

$$p_{\tilde{m},BK_0} = \bar{s}_i \cdot \bar{s}_{i+1} = \overline{s_i + s_{i+1}}, \quad (34b)$$

and the $A_{BK_{0-m}}$ can be calculated as

$$A_{BK_{0-m}} = A_{IVd1} + A_{AND2d1} + A_{OR2d1} + A_{NOR2d2} = 6.34 \text{ GE} \quad (35)$$

and the delay

$$t_{BK_{0-m}} = t_{IVd1} + t_{ANDd1} + t_{ORd1} = 0.05 \text{ ns} + 0.105 \text{ ns} + 0.111 \text{ ns} = 0.266 \text{ ns}. \quad (36)$$

For BK_1 $\tilde{m}_2 = 0$, $\tilde{m}_3 = 0$, hence $g_{\tilde{m}} = 0$ and $p_{\tilde{m}} = s_i \cdot s_{i+1}$.

Assuming the direct realization we receive

$$A_{BK_{1-m}} = A_{AND2d1}, \quad (37)$$

$$t_{BK_{1-m}} = t_{AND2d1}. \quad (38)$$

For other blocks we have

$$A_{BK_{2-m}} = A_{BK_2}, \; A_{BK_{3-m}} = A_{BK_3}, \; A_{BK_{4-m}} = A_{BK_4}. \quad (39)$$

We finally receive for the BK-*m* adder

$$A_{BK-m} = 2A_{IV1d2} + A_{BK0m} + A_{BK1m} + A_{BK2} + A_{BK3} + A_{BK4} + 4A_{XOR2d1} = 32.34 \text{ GE}, \quad (40)$$

and for TOMA for $m = 29$ based on BK adders

$$A_{TOMA\ BK} = A_{BK} + A_{BK\ m} = 59.99 \text{ GE} + 31.68 \text{ GE} = 91.67 \text{ GE}. \quad (41)$$

The *BK-m* delay can be calculated as

$$t_{BK-m} = t_{IV1} + t_{BK0\text{-}m} + \max(t_{BK2\text{-}m}, t_{BK3\text{-}m}) + t_{BK4m} + t_{XOR2d1} + t_{MX2d1}. \quad (42)$$

Hence

$$t_{BK-m} = 0.03 \text{ ns} + 0.15 \text{ ns} + 0.07 \text{ ns} + 0.07 \text{ ns} + 0.09 \text{ ns} + 0.08 \text{ ns} = 0.49 \text{ ns}.$$

Finally we obtain

$$t_{TOMA\_BK} = t_{BK} + t_{BK\text{-}m} = 0.398 \text{ ns} + 0.490 \text{ ns} = 0.888 \text{ ns}.$$

D. The area of the pipelined TOMA BK

This area can be evaluated as

$$A_{TOMA\_BK\_p} = A_{BK} + A_{BK\ m} + n_{BK} \cdot A_{FF} \quad (43)$$

where n_{BK} is the number of flip-flops in pipeline registers. For the structure from Fig. 7 with $n_{BK} = 51$ and $A_{FF} = A_{FD1Q} = 5.67$ GE, we get $A_{TOMA\_BK\_p} = 380.84$ GE.

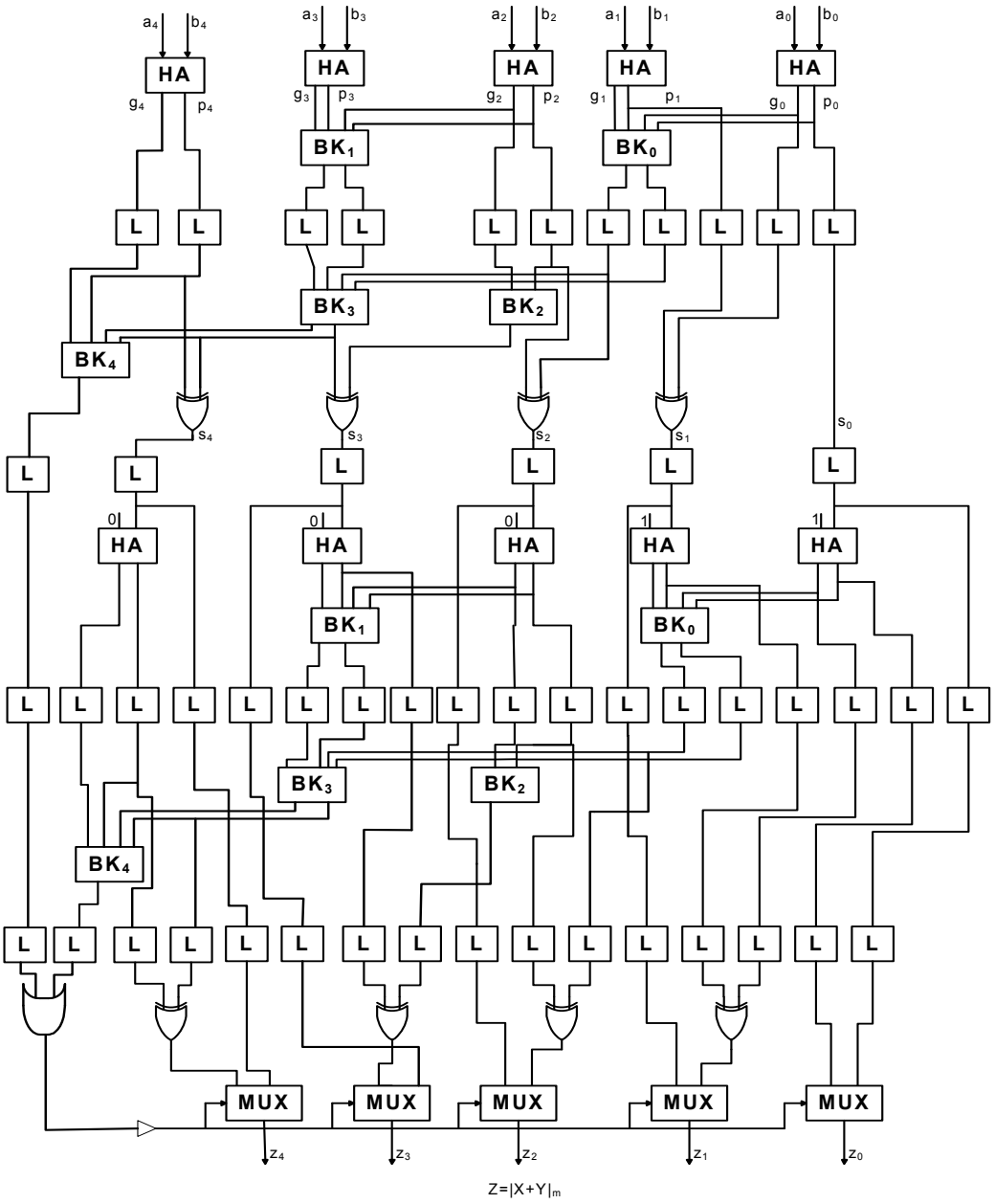

Fig. 7. Pipelined TOMA based on Brent-Kung adder.

E. Pipelining frequency of pipelined TOMA BK:

layer 1

$t_D^{L1,BK}: t_{HAd2} + t_{BK1} = 0.092\ \text{ns} + 0.068\ \text{ns} = 0.160\ \text{ns}$,

layer 2

$t_D^{L2,BK}$:

$t_{BK3} + t_{BK4} + t_{XOR2d1} =$

$= 0.068\ \text{ns} + 0.074\ \text{ns} + 0.090\ \text{ns} = 0.232\ \text{ns}.$

layer 3

$t_D^{L3,BK}: t_{HAd1} + t_{BK1} = 0.088\ \text{ns} + 0.074\ \text{ns} = 0.162\ \text{ns}$,

layer 4

$t_D^{4,BK}: t_{BK3} + t_{BK4} = 0.152\ \text{ns}$,

layer5

$t_D^{L5,BK} = t_{XOR2d1} + t_{NID6} + t_{MX2d1} = 0.222\ \text{ns}.$

Using the $t_D^{L2,BK}$ as the maximum layer delay, we receive the maximum pipelining frequency

$$f_{PF\_\max}^{TOMA\_BK} = 1/(t_D^{L2,BK} + t_{FD1Q}) = 1/(0.232\ \text{ns} + 0.094\ \text{ns})$$
$$= 1/0.326\ \text{ns} = 3.06\ \text{GHz}.$$

6. New Five-bit TOMA

In this section we shall show a new TOMA structure and its pipelined form that requires smaller area than other TOMA structures. The TOMA is configured as a serial connection $X + Y$ adder and $X + Y - m$ adder that are designed in such a manner that leads to a substantial simplification and thus to a smaller delay or a smaller number of pipeline levels. Both adders are modifications of the standard CLA adder. In the first stage of the proposed structure the propagate's and generate's and transfer functions [30] $t_i = a_i + b_i$ are used. The first three carries c_1, c_2 and c_3 are computed simultaneously, and c_3 is used to generate c_4 and c_5.

Generally, the computation of the carry c_i can be expressed, assuming $c_0 = 0$, as

$$c_1 = g_0, \tag{44a}$$

$$c_2 = g_1 + c_1 \cdot t_1, \tag{44b}$$

$$c_3 = g_2 + c_2 \cdot t_2, \tag{44c}$$

$$c_4 = g_3 + c_3 \cdot t_3, \tag{44d}$$

$$c_5 = g_4 + c_4 \cdot t_4. \tag{44e}$$

In the above formulas instead of p_i, the transfer function $t_i = a_i + b_i$ is used, which is justified as follows

$$c_{i+1} = g_i + c_i \cdot p_i, \tag{45}$$

$$c_{i+1} = g_i + c_i \cdot p_i + c_i \cdot g_i = g_i + c_i \cdot (g_i + p_i) = g_i + c_i \cdot t_i, \tag{46}$$

with $t_i = a_i + b_i$, $g_i = a_i b_i$ and $p_i = a_i \oplus b_i$.

We may express c_2 and c_3 as the functions of g_i and t_i as

$$c_2 = g_1 + c_1 \cdot t_1, \tag{47}$$

$$c_3 = g_2 + g_1 \cdot t_2 + g_0 \cdot t_1 \cdot t_2. \tag{48}$$

Consequently, we receive

$$c_4 = g_1 + c_3 \cdot t_3, \tag{49}$$

and

$$c_5 = g_4 + g_3 \cdot t_4 + c_3 \cdot t_3 \cdot t_4. \tag{50}$$

In the adder realization the above equations are transformed to the NAND form. The sum bits are generated using $s_i = p_{i\text{-}1} \oplus c_{i\text{-}}$, 1, 2, 3, 4 with $s_0 = p_0$. The second stage of the TOMA implements the subtraction of $-m$ making use of the TCS representation of $-m$, $\widetilde{m} = (1, \widetilde{m}_4, \widetilde{m}_3, \widetilde{m}_2, \widetilde{m}_1, \widetilde{m}_0)$.

Regarding that the second operand of the $X + Y - m$ adder is $\widetilde{m}$, we can write

$$c_1 = s_0 \cdot \widetilde{m}_0, \tag{51a}$$

$$c_2 = s_1 \cdot \widetilde{m}_1 + s_0 \cdot s_1 \cdot \widetilde{m}_0 + s_0 \cdot \widetilde{m}_0 \cdot \widetilde{m}_1, \tag{51b}$$

$$\begin{aligned} c_3 = {} & s_2 \cdot \widetilde{m}_2 + s_1 \cdot s_2 \cdot \widetilde{m}_1 + s_0 \cdot s_1 \cdot s_2 \cdot \widetilde{m}_0 + \\ & + s_0 \cdot s_2 \cdot \widetilde{m}_0 \cdot \widetilde{m}_1 + + s_1 \cdot s_2 \cdot \widetilde{m}_1 \cdot \widetilde{m}_2 \\ & + s_0 \cdot s_1 \cdot \widetilde{m}_0 \cdot \widetilde{m}_2 + s_0 \cdot \widetilde{m}_0 \cdot \widetilde{m}_1 \cdot \widetilde{m}_2 \end{aligned} \tag{51c}$$

$$c_4 = s_3 \cdot \widetilde{m}_3 + c_3 \cdot s_3 + c_3 \cdot \widetilde{m}_3, \tag{51d}$$

$$\begin{aligned} c_5 = {} & s_4 \cdot \widetilde{m}_4 + s_3 \cdot s_4 \cdot \widetilde{m}_3 + c_3 \cdot s_3 \cdot s_4 + c_3 \cdot s_4 \cdot \widetilde{m}_3 + \\ & + s_3 \cdot \widetilde{m}_3 \cdot \widetilde{m}_4 + c_3 \cdot s_3 \cdot \widetilde{m}_4 + c_3 \cdot \widetilde{m}_3 \cdot \widetilde{m}_4 \end{aligned} \tag{51e}$$

We may simplify the above equations by substituting $\widetilde{m}$ values of the individual five-bit moduli. The results of this simplification are given in Tab. 1.

m	c_1	c_2	c_5	c_4	c_5
17	s_0	$s_1 + s_0$	$s_2 + s_1 + s_0$	$s_3 + c_3$	$s_4 \cdot (s_3 + c_3)$
19	0	s_1	$s_2 + s_1$	$s_3 + c_3$	$s_4 \cdot (s_3 + c_3)$
21	s_0	$s_1 + s_0$	$s_2 \cdot (s_1 + s_0)$	$s_3 + c_3$	$s_4 \cdot (s_3 + c_3)$
23	s_0	$s_1 \cdot s_0$	$s_2 \cdot s_1 \cdot s_0$	$s_3 + c_3$	$s_4 \cdot (s_3 + c_3)$
25	s_0	$s_1 + s_0$	$s_2 + s_1 + s_0$	$s_3 \cdot c_3$	$c_3 \cdot s_4 \cdot s_3 \cdot$
27	s_0	$s_1 \cdot s_0$	$s_2 + s_1 s_0$	$s_3 \cdot c_3$	$c_3 \cdot s_4 \cdot s_3 \cdot$
29	s_0	$s_1 + s_0$	$s_2 \cdot (s_1 + s_0)$	$s_3 \cdot c_3$	$c_3 \cdot s_4 \cdot s_3 \cdot$
31	s_0	$s_1 \cdot s_0$	$s_2 \cdot s_1 \cdot s_0$	$s_3 \cdot c_3$	$c_3 \cdot s_4 \cdot s_3 \cdot$

Tab. 1. Logical functions for realizations of the carries of $X + Y - m$ adder.

In Fig. 8, the TOMA based on the new principle for $m = 29$ is depicted.

A. 5-bit new TOMA area

We shall analyze the area and delay of the new TOMA for $m = 29$.The area of the new TOMA can be computed as

$$A_{TOMA\_New} = A_{X+Y} + A_{X+Y-m}. \tag{52}$$

The hardware amount of the $X + Y$ adder can be expressed as

$$A_{X+Y} = A_{HA-stage} + A_{t_i} + A_{c_2} + A_{c_3} + A_{c_4} + A_{c_5} + A_{SU} \tag{53}$$

where $A_{HA-stage}$ is the area of the input summation stage (HAs and ORs), A_{c_i} are the areas of circuits generating the individual carries c_i.

Subsequently we have

$$A_{HA\_stage} = 2 \cdot A_{HAd1} + 3 \cdot A_{HAd2} + 4 A_{OR2d1} = 27.35 \text{ GE},$$

$$A_{c_2} = A_{IVd1} + 2 \cdot A_{NAND2d1} = 3 \text{ GE},$$

$$A_{c_3} = A_{IVd1} + 2 \cdot A_{NAND2d2} + A_{NAND3d1} + A_{NAND3d2} = 9.67 \text{ GE}$$

$$A_{c_4} = A_{IVd1} + 2 \cdot A_{NAND2d1} = 3 \text{ GE},$$

$$A_{c_5} = A_{IVd1} + A_{NAND2d1} + 2 A_{NAND3d1} = 4.67 \text{ GE},$$

$$A_{SU} = 5 \cdot A_{XOR2d1} = 15 \text{ GE}.$$

In effect, we receive

$$A_{X+Y} = 55.69 \text{ GE}.$$

For the $X + Y - m$ adder we have

$$A_{X+Y-m} = A_{c_2-m} + A_{c_3-m} + A_{c_4-m} + A_{c_5-m} + A_{SU-m},$$

where

$A_{c_1-m} = 0$ (direct connection),

$A_{c_2-m} = A_{OR2d2} = 1.67$ GE,

$A_{c_3-m} = A_{AND2d2} = 2$ GE,

$A_{c_4-m} = A_{AND2d1} = 1.67$GE,

$A_{c_5-m} = A_{AND3d1} = 2$ GE,

$A_{SU-m} = A_{OR2d1} + A_{NID6} + 5 \cdot A_{MX2d1} = 20.34$ GE.

We receive $A_{X+Y-m} = 27.68$ GE.

The total hardware amount is $A_{TOMA_New} = 110.06$GE.

B. 5-bit New TOMA delay

The delay of the new TOMA can be written as

$$t_{TOMA\_New} = t_{X+Y} + t_{X+Y-m}, \tag{54}$$

where

$t_{X+Y} = t_{HA\_Stage} + t_{c_1} + \max(t_{c_2}, t_{c_3}) + \max(t_{c_4}, t_{c_5}) + t_{SU}$ and t_{c_i}, $i = 1, 2,...,5$, denote the individual carry generator delays

$t_{c_1} = \max(t_{HAd2\_ACO}, t_{HAd2\_BCO})$,

$t_{c_2} = \max(t_{NAND2d1}, t_{IVd1}) + t_{NAND2d1}$,

$t_{c_3} = \max(t_{NAND2d1}, t_{NAND3d1}, t_{IVd1}) + t_{NAND3d1}$,

$t_{c_4} = \max(t_{NAND2d1}, t_{IVd1}) + t_{NAND2d1}$,

$t_{c_5} = \max(t_{NAND2d1}, t_{NAND3d1}, t_{IVd1}) + t_{NAND3d1}$,

$t_{X+Y} = t_{HAd2} + t_{NAND3d1} + t_{NAND3d2} + 2 \cdot t_{NAND2d1} + t_{XOR2d1}$,

$t_{X+Y} = 0.092\text{ns} + 0.052\text{ns} + 0.044\text{ns} + 2 \cdot 0.037\text{ns} + 0.09\text{ns}$
$= 0.352\text{ns}$

and for $X + Y - m$ adder we have

$t_{X+Y-m} = t_{c_5'} + \max(t_{OR2d1} + t_{NID6}, t_{XOR2d1})$, where

$t_{c_5'} = t_{OR2d1} + t_{AND2d1} + t_{AND3d1}$,

$t_{c_4\_c_5} = \max(t_{c_4}, t_{c_5})$,

$t_{X+Y-m} = t_{NAND2d1} + t_{NAND3d2} + t_{AND3d1} + t_{XOR2d1} + t_{MX2d1}$,

$t_{X+Y-m} = 0.037\text{ ns} + 0.044\text{ ns} + 0.066\text{ ns} +$
$+ 0.09\text{ ns} + 0.078\text{ ns} = 0.315\text{ ns},$

$t_{TOMA\_New} = 0.352\text{ ns} + 0.315\text{ ns} = 0.667\text{ ns}.$

D. The area of the pipelined new TOMA

This area is expressed as

$$A_{TOMA\_New\_p} = A_{X+Y} + A_{X+Y-m} + n_N \cdot A_{FF}$$

where n_N is the number of flip-flops in pipeline registers. For the structure from Fig. 8 with $n_N = 30$ and A_{FF}=A_{FD1Q}=5.67GE, we get . $A_{TOMA_New} = 280.82$GE

E. Pipelining frequency of the pipelined new TOMA

For the individual layers in the pipelined structure of the new TOMA, shown in Fig. 8, we have the following delays:

layer 1:

$t_D^{L1,N} = t_{HAd1} + 2 \cdot t_{NAND3d1} = 0.192$ ns,

layer 2:

$t_D^{L2,N} = 2 \cdot t_{NAND3d1} + t_{XOR2d1} = 0.194$ ns,

layer 3:

$t_D^{L3,N} = t_{OR2d1} + t_{AND2d1} + t_{AND3d1} = 0.185$ ns,

layer 4:

$t_D^{L4,N} = t_{XOR2d1} + t_{NID6} + t_{MX2d1} = 0.222$ ns.

The design of the pipelined structure aimed at the minimization of the number of pipeline stages while preserving possibly high pipelining frequency. The structure allows one to employ only three pipeline register stages with 30 flip-flops with the maximum pipelining frequency equal to

$$f_{PF\_\max}^{new\_TOMA} = 1/(0.222\text{ ns} + .094\text{ ns}) = 1/0.316\text{ ns} = 3.16\text{ GHz}.$$

In Tab. 2 the summary of the obtained TOMA parameters is given.

	TOMA-RCA	TOMA-BK	TOMA-Hiasat	New TOMA I
Area [GE] (nonpipelined)	81.68	99.01	127.03	110.72
Delay[ns]	0.747	0.888	0.886	0.667
Area x delay	61.01	87.64	112.55	73.41
Number of pipeline layers	6	4	5	3
Number of FFs	66	58	64	30
Area [GE] (pipelined)	472.90	380.84	489.91	280.82
Pipelining frequency max [GHz]	4.22	3.06	3.7	3.16

Tab. 2. TOMA parameters for $m = 29$.

It is seen that the area-delay product has the best values for the TOMA-RCA and the new TOMA, moreover the new TOMA requires the smallest area for the pipelined structure but at the cost of the reduced maximum pipelining frequency. In general the new pipelined TOMA calls for about 35% less area than the TOMA-BK, the best of three other considered structures.

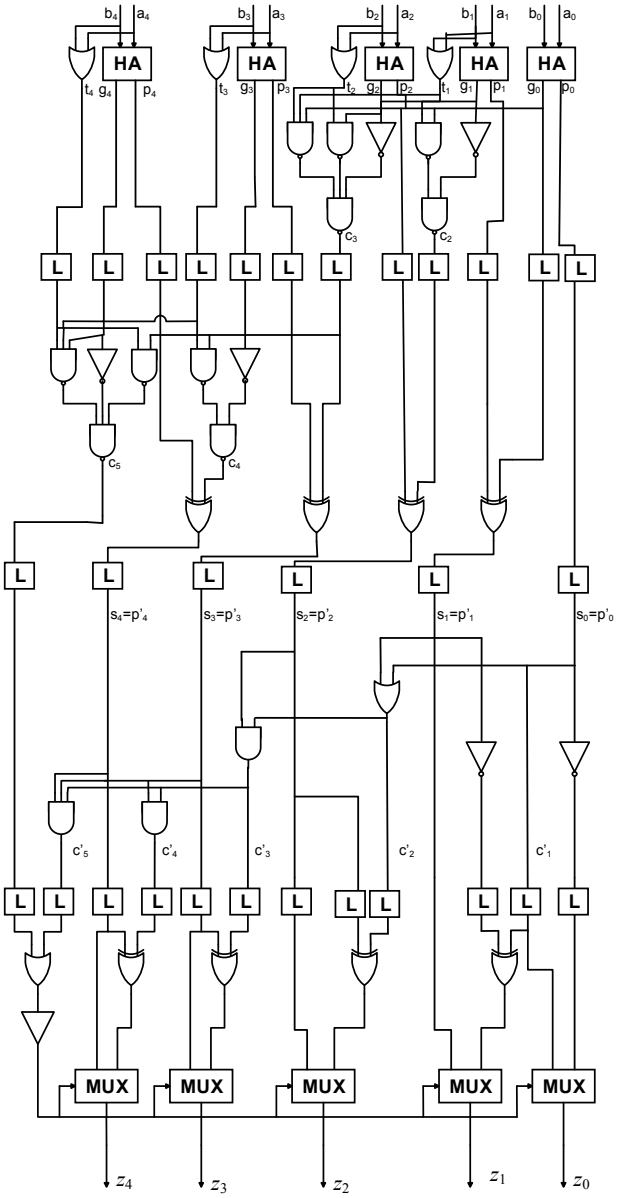

Fig. 8. New five-bit TOMA for $m = 29$.

7. Conclusions

The structures of pipelined two-operand modular adders for five-bit moduli based on ripple carry-adder, Brent-Kung adder and Hiasat adder have been presented and analyzed with respect to the area, number of layers and attainable pipelining frequency. Also a new structure of the two-operand modular adder based on the modified carry-look ahead adder has been proposed. It has been shown that the new pipelined adder has the smallest number of pipeline layers as well as the area smaller by about 35% than the best of other considered structures.

References

[1] SZABO, N. S., TANAKA, R. I. *Residue Arithmetic and its Applications to Computer Technology*. McGraw-Hill Inc., 1967.

[2] SODERSTRAND, M. A., JENKINS, M., JULLIEN, G. A., TAYLOR, F. J. *Residue Number System Arithmetic: Modern Applications in Digital Signal Processing*. IEEE Press, 1986.

[3] OMONDI, A., PREMKUMAR, B. *Residue Number Systems, Theory and Implementation.* Imperial College Press, 2007.

[4] ANANDA MOHAN, P.V. *Residue Number Systems, Algorithms and Architectures*. Kluwer Academic Publishers, 2002.

[5] JENKINS, W. K., KROGMEIER, J. V. The design of dual-mode complex signal processors based on quadratic modular number codes. *IEEE Transactions on Circuits and Systems*, 1987, vol. 34, no. 4, p. 354–364. DOI: 10.1109/TCS.1987.1086154

[6] KRISHAN, R., JULLIEN, G. A., MILLER, W. C. The modified quadratic residue number system (MQRNS) for complex high-speed signal processing. *IEEE Transactions on Circuits and Systems*, 1986, vol. 33, no. 3, p. 325–327. DOI: 10.1109/TCS.1986.1085897

[7] ULMAN, Z., CZYZAK, M. Highly parallel, fast scaling of numbers in nonredundant residue arithmetic. *IEEE Transactions on Signal Processing*, 1998, vol. 46, no. 2, p. 487–496. DOI: 10.1109/78.655432

[8] FRERKING, W. I., PARHI, K. K. Low-power FIR digital filters using residue arithmetic. In *Proceedings* of *the 31st Asilomar Conference on Signals, Systems and Computers ACSSC1997.* Pacific Grove (CA, USA), November 2 - 7, 1997, p. 739–743. DOI: 10.1109/ACSSC.1997.680542

[9] CARDARILLI, G. C., DEL RE, A., LOJACONO, R., NANNARELLI, A., RE, M. RNS implementation of high performance filter for satellite demultiplexing. In *Proceedings of the 2003 IEEE Aerospace Conference*. Big Sky (Montana, USA), March 8 - 15, 2003, vol. 3, p. 1365–1379. DOI: 10.1109/AERO.2003.1235253

[10] CARDARILLI, G. C., DEL RE, A. NANNARELLI, A., RE, M. Low-power implementation of polyphase filters in quadratic residue number system. In *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS 2004)*. Vancouver (Canada), 2004, p. 725–728. DOI: 10.1109/ISCAS.2004.1329374

[11] CZYZAK, M., SMYK, R. FPGA implementation of the two-stage high-speed FIR filter in residue arithmetic. *Elektronika*, 2011, no. 12, p. 90–92.

[12] CZYZAK, M., SMYK, R. Radix-4 DFT butterfly realization with the use of the modified quadratic residue number system. *Poznan University of Technology Academic Journals. Electrical Engineering*, 2010, no. 63, p. 39–51.

[13] GARCIA, A., MEYER-BAESE, U., TAYLOR, F. J. Pipelined Hogenauer CIC filters using field-programmable logic and residue number system. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing.* Seattle (Washington, USA), May 12 - 15, 1998, p. 3085–3088. DOI: 10.1109/ICASSP.1998.678178

[14] BARRACLOUGH, S. R., SOTHERAN, M., BURGIN, K., WISE, A. P., et al. The design and implementation of the IMS A110 image and signal processor. In *Proceedings of the 1989 IEEE Custom Integrated Circuits Conference* (*ICICC 1989*). San Diego (CA, USA), May 15 - 18, 1989, p. 24.5.1–24.5.4. DOI: 10.1109/CICC.1989.56826

[15] WEI, W., SWAMY, M.N.S., AHMAD, M.O. RNS application for digital image processing. In *Proceedings of the 4th IEEE International Workshop System-on-Chip for Real-Time Applications*. Banff (Alberta, Canada), July 19 - 21, 2004, p. 77 to 80. DOI: 10.1109/IWSOC.2004.1319854

[16] BANERJI, D. K. A novel implementation method for addition and subtraction in residue number systems. *IEEE Transactions on Computers*, 1974, vol. 23, no. 1, p. 106–109. DOI: 10.1109/T-C.1974.223790

[17] AGRAWAL, D. P., RAO, T. R. N. Modulo 2^n +1 arithmetic logic. *IEEE Journal on Electronic Circuits and Systems*, 1978, vol. 2, no. 6, p. 186–188. DOI: 10.1049/ij-ecs.1978.0037

[18] SODERSTRAND, M. A. A new hardware implementation of modulo adders for residue number systems. In *Proceedings of the 26th IEEE Midwest Symposium on Circuits and Systems*. Puebla (Mexico), August 15 - 16, 1983, p. 412–415.

[19] BAYOUMI, M., JULLIEN. G. A. VLSI implementation of residue adders. *IEEE Transactions Circuits and Systems*, 1987, vol. 34, no. 3, p. 284–288. DOI: 10.1109/TCS.1987.1086130

[20] DUGDALE, M. VLSI implementation of residue adders based on binary adders. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 1992, vol. 39, no. 5, p. 325–329. DOI: 10.1109/82.142036

[21] PIESTRAK, S. J. Design of high-speed residue-to-binary number system converter based on the chinese remainder theorem. In *Proceedings* of *the International Conference on Computer Design ICCD'94, VLSI in Computers and Processors*. Cambridge (MA, USA), 1994, p. 508–511. DOI: 10.1109/ICCD.1994.331962

[22] ZIMMERMANN, R. Efficient VLSI implementation of modulo ($2^n \pm 1$) addition and multiplication. In *Proceedings of the 14th IEEE Symposium on Computer Arithmetic.* Adelaide (Australia), April, 1999, p. 158–167. DOI: 10.1109/ARITH.1999.762841

[23] HIASAT, A. A. High-speed and reduced-area modular adder structures for RNS. *IEEE Transactions on Computers*, 2002. vol. 51, no. 1, p. 84–89. DOI: DOI: 10.1109/12.980018

[24] PATEL, R. A., BENAISSA, M., POWELL, N., BOUSSAKTA, S. Novel power-delay-area-efficient approach to generic modular addition. *IEEE Transactions on Circuits and Systems–I Regular Papers*, June 2007, vol. 54, no. 6, p. 1279–1292. DOI: 10.1109/TCSI.2007.895369

[25] KELLIHER, T. P., OWENS, R.M., IRWIN, M. J., HWANG, T.-T. ELM - a fast addition algorithm discovered by a program. *IEEE Transactions on Computers*, 1992, vol. 41, no. 9, p. 1181–1184. DOI: 10.1109/12.165399

[26] KOGGE, P. M., STONE, H. S. A parallel algorithm for the efficient solution of a general class of recurrence equations. *IEEE Transactions on Compute*rs, 1973, vol. 22, no. 8, p. 786–793. DOI: 10.1109/TC.1973.5009159

[27] LADNER, R. E., FISCHER, M. J. Parallel-prefix computation. *Journal of Association of Computing Machinery*, 1980, vol. 27, no. 4, p. 831–838. DOI: 10.1145/322217.322232

[28] BRENT, R. P., KUNG, H. T. A regular layout for parallel adders. *IEEE Transactions on Computers*, 1982, vol. 31, no. 3, p. 260 to 264.

[29] *0.13 micron Standard Cell Logic Library STDH 150*. Samsung Inc., 2004.

[30] PARHAMI, B. *Computer Arithmetic: Algorithms and Hardware Designs.* New York: Oxford University Press, 2000.

About the Authors …

Maciej CZYŻAK was born in 1950, Chorzow, Poland. He received his M. Sc. in Automatic Control and Computer Engineering from the Faculty of Electronics, Gdansk University of Technology (GUT) in 1973, and his Ph.D. in Automatic Control from the Faculty of Electrical Engineering, GUT in 1985. He authored and co-authored over 60 technical papers. His interests encompass digital signal processing with the use of residue arithmetic, VLSI and FPGA design.

Jacek HORISZNY was born in 1962, Gdansk Poland. He received his M. Sc. in Electrical Engineering from the Faculty of Electrical Engineering, Gdansk University of Technology (GUT) in 1986, and his Ph.D. in Electrical Engineering from the Faculty of Electrical and Control Engineering, GUT in 1996. He is the author and co-author of over 40 technical papers. His current interests include the transformer design and applications of digital signal processing for transformer protection.

Appendix A

Area [GE]	Delay [ns]
A_{AND2d1} = 1.67	t_{AND2d1}=0.054
A_{AND2d2} = 2.00	t_{AND2d2}=0.055
A_{AND3d1} =2.00	t_{AND3d1} = 0.066
A_{AND3d2} =2.67	t_{AND3d2} = 0.068
A_{AND4d1} =2.33	t_{AND4d1}=0.082
A_{AND4d2} =2.67	t_{AND4d2}=0.085
A_{FAd1}=8.00	t_{FAd1}=0.143
A_{FAd2}=9.00	t_{FAd2}=0.150
A_{HAd1}=4.67	t_{HAd1}=0.088
A_{HAd2}=5.67	t_{HAd2}=0.092
$A_{NAND2d1}$=1.00	$t_{NAND2d1}$=0.037
$A_{NAND2d2}$=2.00	$t_{NAND2d2}$=0.031
$A_{NAND3d1}$=1.67	$t_{NAND3d1}$=0.052
$A_{NAND3d2}$=3.00	$t_{NAND3d2}$=0.044
$A_{NAND4d1}$=2.00	$t_{NAND4d1}$=0.067
$A_{NAND4d2}$=3.67	$t_{NAND4d2}$=0.059
A_{NOR2d1} =1.33	t_{NOR2d1}= 0.050
A_{NOR2d2} =2.00	t_{NOR2d2}= 0.040
A_{OR2d1} =1.67	t_{OR2d1}=0.065
A_{OR2d2} = 2.00	t_{OR2d2}=0.069
A_{OR3d1} =2.00	t_{OR3d1}=0.090
A_{OR3d2} =2.67	t_{OR3d2}=0.090
A_{OR4d1} =3.00	t_{OR4d1}= 0.076
A_{OR4d2} =3.33	t_{OR4d2}= 0.082
A_{OR5d1} =3.33	t_{OR5d1}=0.094
A_{OR5d2} =3.67	t_{OR5d2}= 0.105
A_{XOR2d1}=3.00	t_{XOR2d1}=0.090
A_{IVd1} =1	t_{IVd1}=0.029
A_{NID6} =3.67	t_{NID6} =0.054
A_{MX2d1} =3.00	t_{MX2d1}=0.078
A_{MX2d2} =3.33	t_{MX2d2}=0.076
A_{MX2d4} = 4.33	t_{MX2d4} = 0.092
A_{MX4d1} =6.33	t_{MX4d1}=0.105
A_{FD1Q} = 5.67	t_{FD1Q_SU} = 0.094

Tab. 3. Hardware amount and time delays for STDH150 basic elements.

Half-adder (HA) delays [ns]	Full-adder (FA) delays [ns]
t_{HAd1_ACO}=0.054	t_{FAd1_CICO}=0.083
t_{HAd1_BCO}=0.055	t_{FAd1_ACO}=0.122
t_{HAd1_AS}= 0.088	t_{FAd1_BCO}=0.143
t_{HAd1_BS}= 0.073	t_{FAd1_AS}=0.121
t_{HAd2_ACO}=0.057	t_{FAd1_BS}=0.139
t_{HAd2_BCO}=0.058	t_{FAd2_CICO}=0.089
t_{HAd2_AS}= 0.092	t_{FAd2_ACO}=0.130
t_{HAd2_BS}= 0.074	t_{FAd2_BCO}=0.150
	t_{FAd2_AS}=0.129
	t_{FAd2_BS}=0.150

Tab. 4. Individual delays between input and output nodes for FAs and HAs (STDH150).

Robert SMYK was born in 1978, Malbork, Poland. He received his M. Sc. in Automatic Control from the Faculty of Electrical and Control Engineering, Gdansk University of Technology (GUT) in 2008, and his Ph.D. in Automatic Control from the Faculty of Electrical and Control Engineering, GUT in 2008. He co-authored and authored over 20 technical papers. His interests encompass digital signal processing, residue arithmetic, and FPGA design.