New High-Speed and Low-Power radix-2r multiplication algorithms.

Abdelkrim K. Oudjida, Ahmed Liacha, Mohamed L. Berrandjia, Nicolas Chaillet

New High-Speed and Low-Power Radix-2^r Multiplication Algorithms

A.K. Oudjida, A. Liacha, M.L. Berrandjia
Microelectronics and Nanotechnology Division
Centre de Développement des Technologies Avancées
Algiers, Algeria
a_oudjida@cdta.dz

N. Chaillet
AS2M Department
FEMTO-ST Institute
Besançon, France
nicolas.chaillet@femto-st.fr

***Abstract*— In this paper, a new recursive multibit recoding multiplication algorithm is introduced. It provides a general space-time partitioning of the multiplication problem that not only enables a drastic reduction of the number of partial products (N/r), but also eliminates the need of pre-computing odd multiples of the multiplicand in higher radix (r≥3) multiplication. Based on a mathematical proof that any higher radix-2^r can be recursively derived from a combination of two or a number of lower radices, a series of generalized radix-2^r multipliers are generated by means of primary radices: 2^1 , 2^2, 2^5, and 2^8. A variety of higher-radix (2^3-2^{32}) two's complement 64x64 bit serial/parallel multipliers are implemented on Virtex-6 FPGA and characterized in terms of multiply-time, energy consumption per multiply-operation, and area occupation for r value varying from 2 to 64. Compared to a recent published algorithm, savings of 21%, 53%, 105% are respectively obtained in terms of speed, power, and area.**

Keywords—High-Radix Multiplication; Low-Power Multiplication; Multibit Recoding Multiplication; Partial Product Generator (PPG)

I. BACKGROUND AND MOTIVATION

The continuous refinement of the mostly-used design paradigm based on modified Booth algorithm [1] combined to a reduction tree (carry-save-adder array , Dadda,…) has reached saturation. In [2] only slight improvements are achieved. The proposal reduces the partial product number from $N/2+1$ to $N/2$ using different circuit optimization techniques of the critical path.

Theoretically, only the signed multibit recoding multiplication algorithm [3] is capable of a drastic reduction (N/r) of the partial product number, given that $r+1$ is the number of bits of the multiplier that are simultaneously treated ($1 \le r \le N$). Unfortunately, this algorithm requires the pre-computation of a number of odd multiples of the multiplicand (until $(2^{r-1}-1).X$) that scales linearly with r. The large number of odd multiples not only requires a considerable amount of multiplexers to perform the necessary complex recoding into PPG, but dramatically increases the routing density as well. Therefore, a reverse effect occurs that offsets speed and power benefits of the compression factor (N/r). This is the main reason why the multibit recoding algorithm was abandoned. In practice, designs do not exceed r=3 (radix-8).

This work is supported by "Centre de Développement des Technologies Avancées" (CDTA), Algiers, Algeria, in collaboration with FEMTO-ST Institute, Besançon, France.

The current trend [4][5] relies upon advanced arithmetic to determine minimal number bases that are representatives of the digits resulting from larger multibit recoding. The objective is to eliminate information redundancy inside $r+1$ bit-length slices for a more compact PPG. This is achievable as long as no or just very few odd multiples are required.

In [4], Seidel et al. have introduced a secondary recoding of digits issued from an initial multibit recoding for $5 \le r \le 16$. The recoding scheme is based on balanced complete residue system. Though it significantly reduces the number of partial products (N/r for $5 \le r \le 16$), it requires some odd multiples for $r \ge 8$. While in [5], Dimitrov et al. have proposed a new recoding scheme based on double base number system for $6 \le r \le 11$. The algorithm is limited to unsigned multiplication and requires a larger number of odd multiples.

Instead of looking for more effective number bases, which is a hard mathematical task, our approach consists in exploiting already existing odd-multiple *free* recoding algorithms (2^1, 2^2, 2^5, and 2^8) to recursively build up generalized odd-multiple *free* radix-2^r recoding schemes.

To achieve such a goal, the multibit recoding multiplication algorithm is revisited [3]. Its design space is extended by the introduction of a new recursive version that enables a hardware-friendly space-time partitioning of the multiplication problem. Depending on r value ranging from 2 to N, highly-scalable signed multipliers with various levels of parallelism and latencies can be systematically generated with insignificant control-complexity. The new algorithm has also the merit to recursively reduce the number of partial products (N/r) without any limit for the parameter r and any need for the odd multiples of the multiplicand. It also allows the combination of different recoding schemes proposed in the literature into the same architecture for better performances of the multiplier. Several higher radix (2^3-2^{32}) two's complement 64x64 bit serial/parallel multipliers based on combined recoding schemes are implemented on Virtex-6 FPGA and characterized in terms of speed, power, and area occupation for r value ranging from 2 to 64. Compared to a new signed version of Dimitrov et al. algorithm [5] and Seidel et al. algorithm [4], outstanding results are obtained with the new multibit recoding scheme for r=8 formed by the combination of Seidel algorithm (r=5), MacSorley algorithm (r=2) [1] and Booth algorithm (r=1) [6].

The respective savings are as follows: 21%, 53%, 105% and 8%, 52%, 63% are obtained in terms of multiply-time, energy consumption per multiply-operation, and total gate count, respectively.

The paper is organized as follows. Section I outlines the main requirement specifications for a generalized radix-2^r multiplication. Section II introduces the new recursive multibit recoding multiplication algorithm. A number of high-radix (2^3-2^{32}) variants of the new algorithm accompanied with their implementation results are presented in Section III.

II. THE NEW RECURSIVE MULTIBIT RECODING MULTIPLICATION ALGORITHM

The equation (2.1.2) of the original multibit recoding algorithm presented in [3] does not offer hardware visibility. Let us rewrite it in a simpler hardware-friendly form, as follows:

$$Y=\sum_{j=0}^{\frac{N}{r}-1}\left(y_{rj-1}+2^0\,y_{rj}+2^1\,y_{rj+1}+2^2\,y_{rj+2}+\cdots+2^{r-2}\,y_{rj+r-2}-2^{r-1}\,y_{rj+r-1}\right)2^{rj}=\sum_{j=0}^{\frac{N}{r}-1}Q_j\,2^{rj} \quad (1)$$

Where $y_{-1}=0$ and $r\in \mathrm{N}^*$. For simplicity purposes and without loss of generality, we assume that r is a divider of N.

In equation (1), the two's complement representation of the multiplier Y is split into N/r two's complement slices (Q_j), each of r+1 bit length. Each pair of two contiguous slices has one overlapping bit. In literature, equation (1) is referred to by radix-2^r equation, to which corresponds a digit set $D\left(2^r\right)$ such as $Q_j\in D\left(2^r\right)=\left\{-2^{r-1},\ldots,0,\ldots,2^{r-1}\right\}$. Thus, the multiplication between X and Y becomes: $X.Y=\sum_{j=0}^{\frac{N}{r}-1}X.Q_j.2^{rj}$ (2). Where each partial product can be expressed as follows: $X.Q_j.2^{rj}=(-1)^s.2^e.(m.X)$, with $m\in O_m\left(2^r\right)=\left\{1,3,\ldots,2^{r-1}-1\right\}$ such as $\left|O_m\left(2^r\right)\right|=2^{r-2}$. $O_m\left(2^r\right)$ represents the required set of odd-multiples of the multiplicand ($m.X$) for radix-2^r. Hence, the partial-product generation-process consists first in selecting one odd- multiple ($m.X$) among the whole set of pre-computed odd- multiples, which is then submitted to a hardwired shift of e positions, and finally conditionally complemented $(-1)^s$ depending on the bit sign s of Q_j term.. While lower $m.X$ can be obtained using just one addition ($3X$=$2X$+$1X$), the calculation of higher ones may require a number of computation steps ($11X$= $8X$+$2X$+$1X$).

To bypass the hard problem of odd-multiples, we exploit the fact that the two's complement multiplier Y on which equation (1) is applied, is composed of a series of two's complement digits (Q_j) on which equation (1) can be recursively applied again. Based on this observation, let us announce the two following theorems.

Theorem 1. *Any digit $Q_j\in D\left(2^r\right)$ can be represented in a combination of digits $P_i\in D\left(2^s\right)$, such as s is a divider of r.*

When theorem (1) is applied to equation (1), it gives:

$$Y=\sum_{j=0}^{\frac{N}{r}-1}\left[\sum_{i=0}^{\frac{r}{s}-1}P_{ji}\,2^{si}\right]2^{rj} \quad (3)\ ; \quad \text{where}$$

$P_{ji}\in D\left(2^s\right)=\left\{-2^{s-1},\ldots,0,\ldots,2^{s-1}\right\}$ with

$O_m\left(2^s\right)=\left\{1,3,\ldots,2^{s-1}-1\right\}$ such as $\dfrac{\left|O_m\left(2^r\right)\right|}{\left|O_m\left(2^s\right)\right|}=2^{ks}$ and

$$X.Y=\sum_{j=0}^{\frac{N}{r}-1}\left[\sum_{i=0}^{\frac{r}{s}-1}X.P_{ji}\,2^{si}\right]2^{rj} \quad (4)$$

Theorem 2. *Any digit $Q_j\in D\left(2^r\right)$ can be represented in a combination of digits P_i+T_i such as $P_i\in D\left(2^s\right)$ and $T_i\in D\left(2^t\right)$ with s+t a divider of r, and $t<s$.*

Likewise, when theorem (2) is applied to equation (1), we obtain: $Y=\sum_{j}^{\frac{N}{r}-1}\left[\sum_{i=0}^{\frac{r}{s+t}-1}\left[P_{ji}+T_{ji}\,2^s\right]2^{(s+t)i}\right]2^{rj}$ (5). Where

$P_{ji}\in D\left(2^s\right)=\left\{-2^{s-1},\ldots,0,\ldots,2^{s-1}\right\}$ with

$O_m\left(2^s\right)=\left\{1,3,\ldots,2^{s-1}-1\right\}$ and

$T_{ji}\in D\left(2^t\right)=\left\{-2^{t-1},\ldots,0,\ldots,2^{t-1}\right\}$ with

$O_m\left(2^t\right)=\left\{1,3,\ldots,2^{t-1}-1\right\}$ such as $\dfrac{\left|O_m\left(2^r\right)\right|}{\left|O_m\left(2^{s+t}\right)\right|}=2^{k(s+t)}$

and $X.Y=\sum_{j}^{\frac{n}{r}-1}\left[\sum_{i=0}^{\frac{r}{s+t}-1}\left[X.P_{ji}+X.T_{ji}\,2^s\right]2^{(s+t)i}\right]2^{rj}$ (6)

Theorem (1) and (2) allow an exponential reduction ($1/2^{ks}$ and $1/2^{k(s+t)}$, resp.) of the number of odd-multiples in equations (4) and (6) in comparison to equation (2), but at the expense of a linear augmentation (ks-1 and $k(s+t)$-1, resp.) in the number of additions. The advantage by far outweighs the cost, as practically shown in the next section.

The translation of equation (4) into architecture is depicted by Fig. 1, where each $\mathrm{PPG_j}$ (Q_j) is built up using identical $\mathrm{PPG_{ji}}$ (P_{ji}). This is not the case for equation (6) which requires two different $\mathrm{PPG_{ji}}$ (P_{ji} and T_{ji}). Theorem (1) and (2) can be merged together to produce $\mathrm{PPG_j}$ made of a number of different $\mathrm{PPG_{ji}}$ (P_{ji}, T_{ji}, U_{ji}, V_{ji},...). This is the general case that is thoroughly studied in the next section in order to determine the optimal multiplier.

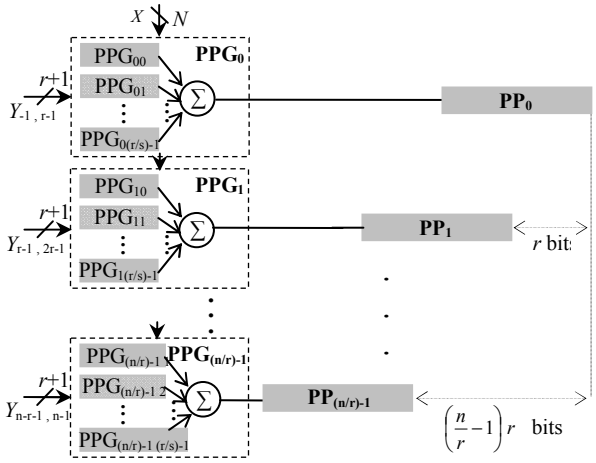

Figure. 1. Generalized $N\times N$ bit radix-2^r parallel multiplier based on sub-radix 2^s. Space partitioning according to r and s values.

III. SOME NEW HIGH RADIX (2^3-2^{32}) RECODING SCHEMES

Theorems (1) and (2) permit to build up any high radix-2^r multiplication algorithm based on lower sub-radices, employing much less odd-multiples. The objective is to generate high radix-2^r multiplication without odd-multiples for a maximum reduction of multiplexer complexity inside PPG_j. To achieve such a goal, a number of odd-multiple free low-radix algorithms are used, such as Booth algorithm [6] (radix-2^1), McSorley algorithm [1] (radix-2^2), Seidel et al. algorithms [4] (radix-2^5 and radix-2^8). The combination of these four algorithms enabled the generation of a series of higher radix recoding schemes (2^3-2^{32}) with minimum hardware resources (Table I). The generation process was manually guided by an heuristic (Table II) that evaluates the logic complexity (*Mux*) inside each PPG_j (Fig. 1).

The multipliers were mapped to Virtex-6 FPGA and characterized in terms of multiply-time, energy consumption per multiply-operation, and area occupation for r value varying from 2 to 64. The obtained results (Fig. 2, 3, and 4) showed an outstanding superiority of our algorithms over their recent counterparts [4][5]. When comparing our algorithms to each other, ß2^2 algorithm is the most area and energy efficient algorithm for any value of r (Table II). For r ranging from 8 to 64, ß"2^8 is the fastest algorithm, but it is outperformed by ß2^{32} for r values greater than 64. ß2^2 algorithm served to design a 16-bit set-point PID. The implementation results outperformed the published ones at all levels [7].

TABLE II
THEORETICAL ESTIMATION OF AREA OCCUPATION AND DELAY

Recoding Algorithm	Area Occupation		Delay (levels)		
	Mux	*Add*	*Mux Delay*	*PPG Adders*	*Linear Reduction Tree*
ß2^2	5r	r/2	d_2	0	r/2
ß2^3	5r	(2/3)r	d_2	1	r/3
ß2^5	27r	(3/5)r	d_5	2	r/5
ß2^8	194r	(7/8)r	d_8	6	r/8
ß'2^8	520r	r/4	d'_8	1	r/8
ß"2^8	19r	(5/8)r	d_5	4	r/8
ß2^{13}	130r	(10/13)r	d_8	9	r/13
ß2^{16}	100r	(11/16)r	d_8	10	r/16
ß2^{24}	74r	(16/24)r	d_8	15	r/24
ß2^{32}	60r	(21/32)r	d_8	20	r/32

Mux is an heuristic measure of the multiplexer logic inside PPG_i. *Add is* the exact umber of adders. d_i is the delay due to *Mux* logic ($d_2 < d_5 < d_8 < d'_8$)

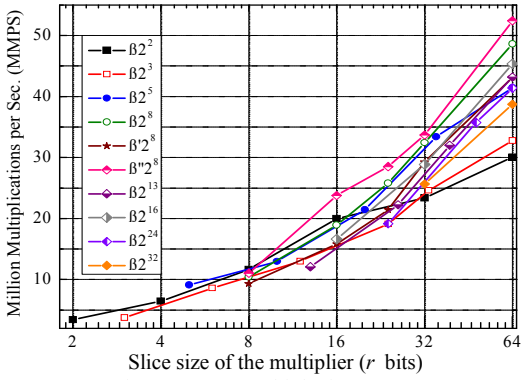

Figure 2. Max. multiply time versus r.

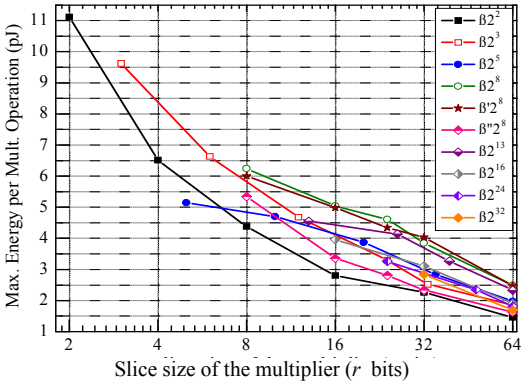

Figure 3. Max. energy consumption per mult. operation versus r.

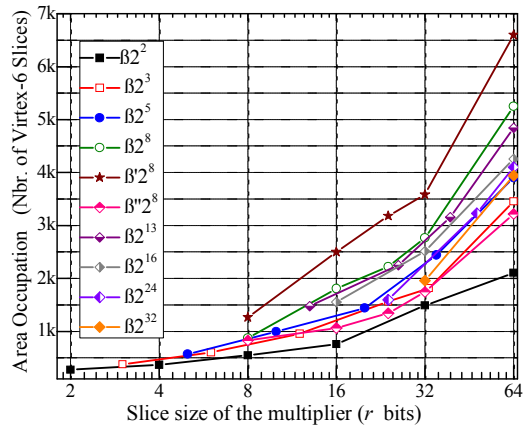

Figure. 4. Area occupation versus r

REFERENCES

[1] O.L. MacSorley, "High-Speed Arithmetic in Binary Computers," Proceedings of the IRE, Vol. 49(1), pp. 67-91, January 1961.

[2] F. Lamberti, "Reducing the Computation Time in (Short Bit-Width) Two's Complement Multiplier," IEEE Trans. on Computers, vol. 60, N° 2, pp. 148-156, February 2011.

[3] H. Sam, and A. Gupta, "A Generalized Multibit Recoding of Two's Complement Binary Numbers and its Proof with Application in Multiplier Implementation," IEEE Trans. on Computers, vol. 39, N° 8, August 1990.

[4] P.M. Seidel et al., "Secondary Radix Recodings for Higher Radix Multipliers," IEEE Trans. on Computers, vol. 54, N°2, February 2005.

[5] V.S. Dimitrov et al., "Area Efficient Multipliers Based on Multiple-Radix Representations," IEEE Trans. on Computers, vol. 60, N° 2, pp 189-201, February 2011.

[6] A. D. Booth, "A Signed Binary Multiplication Te:chnique," Quarterly J. Mech. Appl. Math., Vol. 4, part 2, pp. 236-240,1951.

[7] A.K. Oudjida et al., "High-Speed and Low-Power PID Structures for Embedded Applications" Proceedings of the 21th edition of the International Workshop on Power and Timing Modeling, Optimization and Simulation PATMOS, LNCS 6951, pp. 257-266, Springer-Verlag Editor. Madrid, Spain, Sep. 26-29 2011.

TABLE I

SUMMARY OF OUR NEW RADIX-2^r MULTIBIT RECODING ALGORITHMS

Recoding Algorithm	Recoding Equation and Main Features
$\beta 2^r$	$Y=\sum_{j=0}^{\frac{N}{r}-1} Q_j.2^{rj}$; BR: 2^r ; OM: $\{1,3,...,2^{r-1}-1\}$; DV: $Q_j \in \{-2^{r-1},...,0,...,2^{r-1}\}$;
$\beta 2^2$	$Y=\sum_{j=0}^{\frac{N}{r}-1}\left[\sum_{i=0}^{\frac{r}{2}-1} Q_{ji}\,2^{2i}\right].2^{rj}$; BR: 2^2 ; OM: $\{1\}$; DV: $Q_{ji} \in \{-2,-1,0,1,2\}$;
$\beta 2^3$	$Y=\sum_{j}^{\frac{N}{r}-1}\left[\sum_{i=0}^{\frac{r}{3}-1}\left(Q_{ji}+P_{ji}\,2^2\right).2^{3i}\right].2^{rj}$; BR: $2^1, 2^2$; OM: $\{1\}$; DV: $Q_{ji} \in \{-2,-1,0,1,2\}$, $P_{ji} \in \{-1,0,1\}$
$\beta 2^5$	$Y=\sum_{j=0}^{\frac{N}{r}-1}\left[\sum_{i=0}^{\frac{r}{5}-1}\left(7.Q_{ji}+P_{ji}\right)2^{5i}\right].2^{rj}$; BR: 2^5 ; OM: $\{1\}$; DV: $Q_{ji} \in \{-2,-1,0,1,2\}$, $P_{ji} \in \{-4,-2,-1,0,1,2,4\}$
$\beta 2^8$	$Y=\sum_{j=0}^{\frac{N}{r}-1}\left[\sum_{i=0}^{\frac{r}{8}-1}\left(11^2.Q_{ji}+11.P_{ji}+T_{ji}\right)2^{8i}\right].2^{rj}$; BR: 2^8 ; OM: $\{1\}$; DV: $Q_{ji} \in \{-2,-1,0,1,2\}$, $P_{ji}, T_{ji} \in \{-16,-8,-4,-2,-1,0,1,2,4,8,16\}$;
$\beta' 2^8$	$Y=\sum_{j=0}^{\frac{N}{r}-1}\left[\sum_{i=0}^{\frac{r}{8}-1}\left(2^k.Q_{ji}+(-1)^e\,2^h.P_{ji}\right)2^{8i}\right].2^{rj}$ BR: 2^8 ; OM: $\{1,3,5,7\}$; DV: $Q_{ji}, P_{ji} \in \{1,3,5,7\}$ $k, h \in \{0,1,2,3,4,5,6,7\}$; $e \in \{0,1\}$
$\beta'' 2^8$	$Y=\sum_{j=0}^{\frac{N}{r}-1}\left[\sum_{i=0}^{\frac{r}{8}-1}\left[\left(7.Q_{ji}+P_{ji}\right)+\left(R_{ji}+S_{ji}\,2^2\right).2^5\right].2^{8i}\right].2^{rj}$ BR: $2^1, 2^2, 2^5$; OM: $\{1\}$; DV: $Q_{ji} \in \{-2,-1,0,1,2\}$, $P_{ji} \in \{-4,-2,-1,0,1,2,4\}$, $R_{ji} \in \{-2,-1,0,1,2\}$, $S_{ji} \in \{-1,0,1\}$
$\beta 2^{13}$	$Y=\sum_{j=0}^{\frac{N}{r}-1}\left[\sum_{i=0}^{\frac{r}{13}-1}\left[\left(11^2.Q_{ji}+11.P_{ji}+T_{ji}\right)+\left(7.R_{ji}+S_{ji}\right).2^8\right].2^{13i}\right].2^{rj}$ BR: $2^5, 2^8$; OM: $\{1\}$; DV: $Q_{ji} \in \{-2,-1,0,1,2\}$, $P_{ji}, T_{ji} \in \{-16,-8,-4,-2,-1,0,1,2,4,8,16\}$, $R_{ji} \in \{-2,-1,0,1,2\}$, $S_{ji} \in \{-4,-2,-1,0,1,2,4\}$
$\beta 2^{16}$	$Y=\sum_{j=0}^{\frac{N}{r}-1}\left[\sum_{i=0}^{\frac{r}{16}-1}\left[\left(11^2.Q_{ji}+11.P_{ji}+T_{ji}\right)+\left(\sum_{k=0}^{3} M_{kji}.2^{2k}\right).2^8\right].2^{16i}\right].2^{rj}$ BR: $2^2, 2^8$; OM: $\{1\}$; DV: $Q_{ji} \in \{-2,-1,0,1,2\}$, $P_{ji}, T_{ji} \in \{-16,-8,-4,-2,-1,0,1,2,4,8,16\}$, $M_{kji} \in \{-2,-1,0,1,2\}$
$\beta 2^{24}$	$Y=\sum_{j=0}^{\frac{N}{r}-1}\left[\sum_{i=0}^{\frac{r}{24}-1}\left[\left(11^2.Q_{ji}+11.P_{ji}+T_{ji}\right)+\left(\sum_{k=0}^{3} M_{kji}.2^{2k}\right).2^8+\left(7.R_{ji}+S_{ji}\right)2^{16}+\left(U_{ji}+V_{ji}\,2^2\right).2^{21}\right].2^{24i}\right].2^{rj}$ BR: $2^1, 2^2, 2^5, 2^8$; OM: $\{1\}$; DV: $Q_{ji} \in \{-2,-1,0,1,2\}$, $P_{ji}, T_{ji} \in \{-16,-8,-4,-2,-1,0,1,2,4,8,16\}$, $M_{kji} \in \{-2,-1,0,1,2\}$, $R_{ji} \in \{-2,-1,0,1,2\}$, $S_{ji} \in \{-4,-2,-1,0,1,2,4\}$, $U_{ji} \in \{-2,-1,0,1,2\}$, $V_{ji} \in \{-1,0,1\}$
$\beta 2^{32}$	$Y=\sum_{j=0}^{\frac{N}{r}-1}\left[\sum_{i=0}^{\frac{r}{32}-1}\left[\left(11^2.Q_{ji}+11.P_{ji}+T_{ji}\right)+\left(\sum_{k=0}^{3} M_{kji}.2^{2k}\right).2^8+\sum_{k=0}^{1}\left(\left(7.R_{kji}+S_{kji}\right)2^{16+8k}+\left(U_{kji}+V_{kji}\,2^2\right).2^{21+8k}\right)\right].2^{32i}\right].2^{rj}$ BR: $2^1, 2^2, 2^5, 2^8$; OM: $\{1\}$; DV: $Q_{ji} \in \{-2,-1,0,1,2\}$, $P_{ji}, T_{ji} \in \{-16,-8,-4,-2,-1,0,1,2,4,8,16\}$, $M_{kji} \in \{-2,-1,0,1,2\}$, $R_{kji} \in \{-2,-1,0,1,2\}$, $S_{kji} \in \{-4,-2,-1,0,1,2,4\}$, $U_{kji} \in \{-2,-1,0,1,2\}$, $V_{kji} \in \{-1,0,1\}$

BR: Based on Radix ; DV: Digit Variations ; OM: Odd-Multiples