# Efficient Hardware Implementation for Maiorana-McFarland type Functions

Anupam Chattopadhyay[1], Subhamoy Maitra[2], Bimal Mandal[3], Manmatha Roy[2], Deng Tang[4]

[1] Nanyang Technological University, Singapore, anupam@ntu.edu.sg
[2] Indian Statistical Institute, Kolkata, subho@isical.ac.in, manmatha@isical.ac.in
[3] Indian Institute of Technology Jodhpur, India, bimalmandal@iitj.ac.in
[4] Shanghai Jiao Tong University, Shanghai, dtang@foxmail.com

**Abstract.** Maiorana–McFarland type constructions are basically concatenating the truth tables of linear functions on a smaller number of variables to obtain highly nonlinear ones on larger inputs. Such functions and their different variants have significant cryptology and coding theory applications. The straightforward hardware implementation of such functions using decoders (Khairallah et al., WAIFI 2018; Tang et al., SIAM Journal on Discrete Mathematics, 2019) requires exponential resources on the number of inputs. In this paper, we study such constructions in detail and provide implementation strategies for a selected subset of this class with polynomial many gates over the number of inputs. We demonstrate that such implementations cover the requirement of cryptographic primitives to a great extent. Several existing constructions are revisited in this direction, and exact implementations are provided with specific depth and gate counts for hardware implementation. Related combinatorial results of theoretical nature are also analyzed in this regard. Finally, we present a novel construction of a new class of balanced Boolean functions with very low absolute indicators and very high nonlinearity that can be implemented in polynomial-size circuits over the number of inputs. We underline that these constructions have immediate applications to resist the signature generation in Differential Fault Attack (DFA) and to implement functions on a large number of variables in designing ciphers for the paradigm of Fully Homomorphic Encryption (FHE).

**Keywords:** Balancedness, Bent Function, Boolean Function, Combinational Circuits, Hardware Implementation, Maiorana-McFarland Construction.

## 1 Introduction

Boolean functions (for details see [3, 4]) are an integral part of computing as well as communication science and technology. As a subclass, the bent functions [5, 6, 18] (detailed study is available in [15]) are one of the most interesting combinatorial structures that have applications to coding theory and cryptology. The

Maiorana-McFarland (we will call it MM from now on) construction is the simplest and one of the most fundamental techniques in this domain. It has been used in different kinds of construction that could achieve interesting properties relevant to cryptology, in particular for the design of symmetric ciphers. To advance the proceedings technically, let us immediately describe the MM bent functions on $n = 2k$ variables. Consider the input variables in two different sets as $\mathbf{x} = (x_1, x_2, \ldots, x_k)$ and $\mathbf{y} = (y_1, y_2, \ldots, y_k)$. The MM bent functions are of the form $\mathbf{x} \cdot \pi(\mathbf{y}) \oplus g(\mathbf{y})$, where $\pi$ is a permutation of $k$ bits to $k$ bits (a subclass of multiple output Boolean functions) and $g$ is a $k$-input 1-output Boolean function. How many such functions are there? One can see that there are $(2^k)!$ such permutations and $2^{2^k}$ possible ways to choose $g(\mathbf{y})$. Thus, the total number of such functions is $(2^{\frac{n}{2}})! \cdot 2^{2^{\frac{n}{2}}}$, for even $n = 2k$. It is easy to see that this is a huge class of functions. It is also needless to mention that implementation of $\pi$, $g$ may require an exponential number of gates, and the depth of the circuit can also be high. On the other hand, we require circuits that can be implemented with a significantly small number of gates as well as depth. For example, if we choose $\pi$ as the identity permutation and $g$ as identity 0, then the function can be implemented using a significantly small amount of circuit components. The function is then $x_1y_1 \oplus x_2y_2 \oplus \cdots \oplus x_ky_k$, which we refer henceforth as $\mathrm{MM}_0$ function. Indeed, It can be implemented with $k$ two-input AND gates placed in parallel and $k-1$ two-input XOR gates placed in the form of a complete binary tree with $\lceil \log_2 k \rceil$ depth.

How does such a function look like in truth table format? Suppose the left half of the input variables are $\mathbf{y}$ and the right half is $\mathbf{x}$. The linear functions $\bigoplus_{i=1}^{k} a_i x_i$, for all $2^k$ different options of $(a_1, a_2, \ldots, a_k)$. Each such linear function has a truth table of $2^k$ length, and concatenating different $2^k$ of them, one can get a truth table of length $2^{2k}$ for an $n = 2k$ variable function. In a generalized framework, the small truth tables of the linear functions can be permuted among each other, which is defined by $\pi$. Moreover, whether the linear function will be presented as it is or complemented will be decided by the function $g$. It can be proved that such functions are bent, i.e., they have nonlinearity $2^{n-1} - 2^{\frac{n}{2}-1}$. Now, it is well understood that, indeed, there exists a sub-class of permutation $\pi$ and a sub-class of any function $g$ that can be implemented efficiently (to be precise, with at most polynomial many components with respect to the input size). We explore such a class in Section 2.

The bent functions are generally not directly exploited as cryptologic primitives since they are not balanced. However, it should be noted that globally, these are the best possible Boolean functions available, where the Walsh–Hadamard (related to confusion and better properties to resist linear cryptanalysis) and autocorrelation (related to diffusion and better properties to resist differential cryptanalysis) spectra provide the provably optimized characteristics. The nonlinearity of a Boolean function $f$, denoted by $nl(f)$, is maximum when it is bent, and such functions exist only on an even number of variables. Modified versions of such bent functions are compromised a little in terms of nonlinearity and autocorrelation values, but certain other properties, such as balancedness

and resiliency, could be achieved. One well-known modification technique is to replace the truth table of the all zero linear function (i.e., $\bigoplus_{i=1}^{k} a_i x_i$, with all $a_i = 0$) by some nonlinear balanced functions on $k = \frac{n}{2}$ variables, as described in [7]. This technique provides balanced functions on (even) $n$ variables, with nonlinearity $2^{n-1} - 2^{\frac{n}{2}} + nlb(\frac{n}{2})$, where $nlb(t)$ is the maximum nonlinearity of a balanced Boolean functions in $t$ variables.

Due to different applications, efficient implementations of Boolean functions on a large number of inputs have been studied in the literature. In this regard, one may refer to [20, 21], where the circuits for cryptographically significant functions have been presented. The basic idea was to start with a Boolean function and then add a new variable in each step of the pipeline. Later in [10], efficient representation and software implementation of resilient Maiorana-McFarland S-boxes have been studied. However, the ideas did not refer to how one can implement functions on a large number of variables in hardware having very good nonlinearity as well as autocorrelation properties. Very recently, certain evolutionary algorithms have been exploited in [16] to construct functions with good parameters, and that improves certain results of [11]. However, such search methodologies can only work up to a certain number of variables, namely 26, as pointed out in [16]. Our motivation here is to provide generic constructions and implementations that are not limited to a certain range, and we will soon discuss the implementation of a 200-variable function. In fact, the question of such efficient implementation has been raised a few years back in [2], too. Presently, these questions are of renewed interest for the following reasons.

In recent years, symmetric-key cryptographic primitives, such as stream ciphers, block ciphers, hash functions, pseudo-random generators, pseudo-random functions, message authentication codes, and various related primitives serve as fundamental building blocks for many new applications of cryptography like secure Multi-Party Computation (MPC) [24], Zero-Knowledge proofs (ZK) [9], and Fully Homomorphic Encryption (FHE) [8, 17]. MPC allows different users who do not necessarily trust each other to evaluate a function on a shared secret without revealing it. FHE allows a user to operate on encrypted data without decrypting them. Finally, ZK is a technique that allows the authentication of secret information without disclosing it. In such scenarios, the inputs, outputs, and keys of symmetric-key cryptographic primitives are secretly shared or distributed between two or more parties and the cryptographic computations (expressed as an arithmetic circuit composed of gates which are multiplications and additions and connected by wires) are also processed in a distributed manner. In MPC/ZK/FHE, the bottleneck of cryptographic computations is the multiplicative complexity of the symmetric-key cryptographic primitives, and traditional standards like AES and SHA-3 are no longer efficient. Naturally, a line of research on MPC/ZK/FHE-friendly symmetric-key primitives has been developed. In this regard, if one looks into the design of the stream cipher family like FLIP [14], one can see that Boolean functions on a large number of variables are in use, but those do not have significantly good cryptologic parameters.

There is another important aspect from the fault attack point of view. One may note that most of the stream ciphers use only a few outputs from the Feedback Shift Registers (FSRs, Linear, and Nonlinear) and combine them to produce the key stream. However, that creates a situation where all the FSR points do not contribute to the output function. This is the reason, corresponding to a fault, signatures can be generated by comparing the stream without fault and the faulty key stream for several key-IV samples offline (see [19] and the references therein for more details). Consequently, these signatures can be used during the actual fault attack to identify the fault's location easily. If all the points of the FSRs are fed into a Boolean function, and the function has a good autocorrelation property, then such signatures cannot be generated efficiently. Since most modern lightweight hardware stream ciphers use around 200-length FSRs, we need to construct functions on those many variables. Using our method (see Remark 3 in Section 4.3), this is achievable as, for $n = 2k = 200$, $9.5k = 950$, and it is possible to implement these many logic gates in a lightweight circuit. A concrete stream cipher design in this direction could be a possible research direction.

### 1.1 Organization & Contribution

We first present our overall idea towards the efficient implementation of certain types of MM bent functions in section 2. The techniques we exploit are known in circuit design, but they are assembled from an engineering viewpoint to set up the constructions. We also present a preliminary understanding of circuit complexity results as passing remarks. The foundation of our construction is in the following section.

In section 3, we will identify how such modifications of bent functions can be accommodated with efficient circuit implementation strategies. Toward obtaining a very good autocorrelation spectrum and very good nonlinearity, some recent studies [11, 22, 23] considered certain modifications of MM type of constructions. Since such functions have immediate applications as cryptographic primitives, certain works ([12], [23, Section 4.3]) considered the implementation issues, and the gate count in those cases is exponential. This is because the circuit ideas are primarily based on decoders, and thus, $O(2^k)$ amount of logic gates are used. We investigate such constructions in detail and obtain efficient polynomial implementations in this regard. Such improvements are also presented in Section 3.

Finally, in section 4, we construct balanced Boolean functions having very good nonlinearity and very low absolute indicators. It should also be noted that the algebraic degree of the constructed functions is also high, as evident from Remark 2. This is a novel class and efficient implementation related to this complete framework are presented here. Section 5 concludes the paper. Before proceeding to the contributory sections, let us briefly outline a few definitions.

### 1.2 Preliminaries

First, we present some basic definitions and notations of Boolean functions. Let $\mathbb{F}_2$ be the prime field of characteristic 2 and $\mathbb{F}_{2^n}$ be an $n$th degree extension finite field over $\mathbb{F}_2$. Let $\mathbb{F}_2^n$ be an $n$-dimensional vector space over $\mathbb{F}_2$. An element of $\mathbb{F}_2^n$ is denoted by $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, where $x_i \in \mathbb{F}_2$, $1 \leq i \leq n$. Further, the set of nonzero elements of $\mathbb{F}_2^n$ is denoted by $\mathbb{F}_2^{n*}$. The weight of $\mathbf{x} \in \mathbb{F}_2^n$ is defined as $wt(\mathbf{x}) = \sum_{i=1}^{n} x_i$, where the sum is over integers. The cardinality of a set $S$ is denoted as $|S|$, the number of elements in $S$. Any function from the vector space $\mathbb{F}_2^n$ to $\mathbb{F}_2^m$ is called an $n$-variables Boolean function having $m$ outputs. Without mentioning specifically, we consider $m = 1$, and the set of all $n$-variable 1-output Boolean functions is denoted as $\mathcal{B}_n$. A Boolean function $f$ in $n$ variables can be written as a multi-variables polynomial of the form

$$f(\mathbf{x}) = \bigoplus_{\mathbf{a} \in \mathbb{F}_2^n} \mu_{\mathbf{a}} x_1^{a_1} x_2^{a_2} \cdots x_n^{a_n},$$

where $\mu_{\mathbf{a}} \in \mathbb{F}_2$ for all $\mathbf{a} \in \mathbb{F}_2^n$. This representation is called the Algebraic Normal Form (ANF) of $f$. The algebraic degree of $f$ is defined by $deg(f) = \max\{wt(\mathbf{a}) : \mu_{\mathbf{a}} \neq \mathbf{0}\}$. If $deg(f) \leq 1$, then $f$ is called an affine function. In particular, if the constant term of an affine function is zero, then it is called a linear function. The set $supp(f) = \{\mathbf{x} : f(\mathbf{x}) = 1\}$ is called the support of $f$ and its cardinality, i.e., $|supp(f)|$, is called weight of $f$. If $|supp(f)| = 2^{n-1}$ of an $n$-variable Boolean function $f$, then $f$ is said to be a balanced function. The Walsh–Hadamard transform of $f$ at $\mathbf{a} \in \mathbb{F}_2^n$ is defined as

$$W_f(\mathbf{a}) = \sum_{\mathbf{x} \in \mathbb{F}_2^n} (-1)^{f(\mathbf{x}) \oplus \mathbf{a} \cdot \mathbf{x}}.$$

The multiset $[W_f(\mathbf{a}) : \mathbf{a} \in \mathbb{F}_2^n]$ is called Walsh–Hadamard spectrum of $f$. By the well known Parseval's theorem, $\sum_{\mathbf{a} \in \mathbb{F}_2^n} W_f^2(\mathbf{a}) = 2^{2n}$ for all $f \in \mathcal{B}_n$. Thus, for any Boolean function $f$ in $n$ variables, we have $\max_{\mathbf{a} \in \mathbb{F}_2^n} |W_f(\mathbf{a})| \geq 2^{\frac{n}{2}}$. The nonlinearity of $f \in \mathcal{B}_n$ is the minimum Hamming distance from all affine functions in $n$ variables. According to the definition of Walsh–Hadamard transform, we have $nl(f) = 2^{n-1} - \frac{1}{2} \max_{\mathbf{a} \in \mathbb{F}_2^n} |W_f(\mathbf{a})|$. It is known that the nonlinearity of $f \in \mathcal{B}_n$ is upper-bounded by $2^{n-1} - 2^{\frac{n}{2}-1}$. A Boolean function that achieves the nonlinearity bound is called a bent function, i.e., for any bent function $f \in \mathcal{B}_n$, $nl(f) = 2^{n-1} - 2^{\frac{n}{2}-1}$. Thus, bent functions exist only for an even number of input variables and are not balanced since $|W_f(\mathbf{a})| = 2^{\frac{n}{2}}$, for all $\mathbf{a} \in \mathbb{F}_2^n$. Another important property, the autocorrelation of $f \in \mathcal{B}_n$ at point $\mathbf{a} \in \mathbb{F}_2^n$, $C_f(\mathbf{a})$, is defined as

$$C_f(\mathbf{a}) = \sum_{\mathbf{x} \in \mathbb{F}_2^n} (-1)^{f(\mathbf{x}) \oplus f(\mathbf{x} \oplus \mathbf{a})}.$$

The absolute indicator of $f \in \mathcal{B}_n$ is defined by $\Delta_f = \max_{\mathbf{a} \in \mathbb{F}_2^{n*}} |C_f(\mathbf{a})|$. For a bent function $f \in \mathcal{B}_n$, $C_f(\mathbf{a}) = 0$ for all nonzero $\mathbf{a} \in \mathbb{F}_2^n$ and $C_f(\mathbf{0}) = 2^n$.

From a cryptographic point of view, one needs to construct a balanced Boolean function having very high nonlinearity and very low absolute indicator to resist the linear and differential attacks on symmetric ciphers. In this direction, many such functions are constructed in [7, 11, 22, 23]. In [23], the number of logic gates required to implement some balanced functions is counted, and it was noted that an exponential order of components is required. In this paper, we discuss the hardware implementation of known balanced functions [7, 11, 22, 23] in more detail. Based on these, we show that it is possible to implement a subclass of such functions having very high nonlinearity and very low absolute indicator using a polynomial-size circuit.

## 2 Warm up: Identifying a Subclass of MM Bent Functions having Efficient Implementation

In this section, we study the circuit complexity of MM functions of the form $f(\mathbf{x}, \mathbf{y}) = \mathbf{x} \cdot \pi(\mathbf{y}) \oplus g(\mathbf{y})$, where $\pi$ is a permutation over $\mathbb{F}_2^k$, $g \in \mathcal{B}_k$ and $\mathbf{x}, \mathbf{y} \in \mathbb{F}_2^k$. It is convenient to study those constructions with the help of an implementation view using a decoder and without any decoder. Any MM function $f$ in $2k$ variables can be written as a concatenation of $2^k$ distinct linear functions in $k$ variables, called a linear block of length $2^k$. We present Fig. 1, which depicts a generic circuit implementation for the MM Bent function $f \in \mathcal{B}_{2k}$, directly following the definition. To select a linear function in $k$ variables $x_1, x_2, \ldots, x_k$,

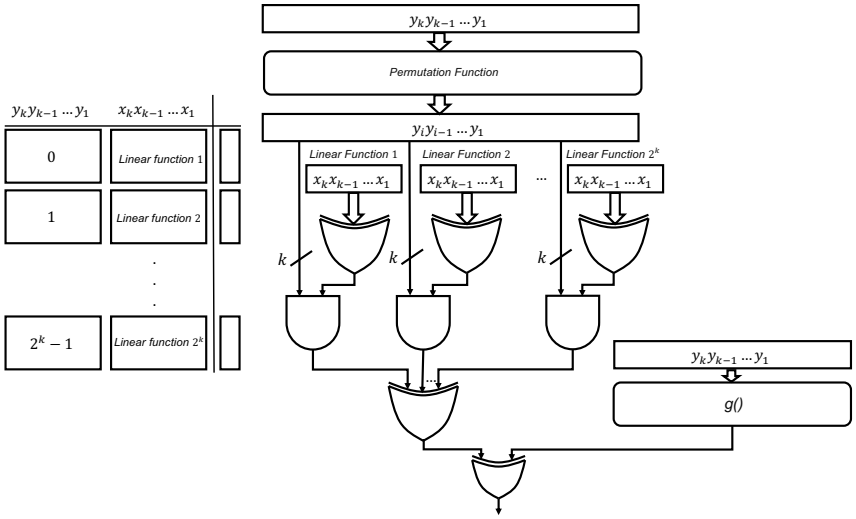


Fig. 1: Generic circuit implementation for MM bent function

a decoder of $k$ inputs and $2^k$ outputs are used, which require $2^k$ logic gates to implement. It can be easily observed that this naive construction leads to a circuit of exponential complexity, which we also write formally in the following.

**Lemma 1.** *Implementation of an $n$-input MM Bent function using AND, OR, XOR and NOT gates, where AND, OR, and XOR gates have a fan-in bound of 2 basis, following the circuit given in Fig. 1 in $\mathcal{O}(n2^{\frac{n}{2}})$-size.*

*Proof.* To establish the result, we first need to show a sub-circuit of $\mathcal{O}(2^{\frac{n}{2}})$ complexity. Considering the simplest case of the permutation function, $\pi$ being identity, the overall circuit size is dominated by the $k$-input $\oplus$ gates, each of which can be implemented with $\mathcal{O}(k)$-size circuit. Since we have $2^k$ linear functions to be implemented, the overall size is lower bounded by $\mathcal{O}(k2^k)$. Since $k$ is an additive sub-part of $n$, in particular $k = \frac{n}{2}$, in this case, we get the result. □

This is to explain that the straightforward implementations are not efficient, which is dependent on permutation $\pi$ over $\mathbb{F}_2^k$ and $g \in \mathcal{B}_k$. The kind of calculation we presented in Lemma 1 has been noted in [12], as well as in [23, Section 4.3]. However, this work explains that the circuit complexity can be lowered to polynomial order for certain subclasses without compromising any cryptographic properties. This is briefly touched upon earlier using the $\mathrm{MM}_0$ circuit. The construction can be generalized, leading to a large number of variants. We begin with a simple construction with $\pi$ as an identity permutation over $\mathbb{F}_2^k$ and $g(\mathbf{y}) = 0$, for all $\mathbf{y} \in \mathbb{F}_2^k$. The circuit implementation of $\mathrm{MM}_0$ functions is given in Fig. 2.

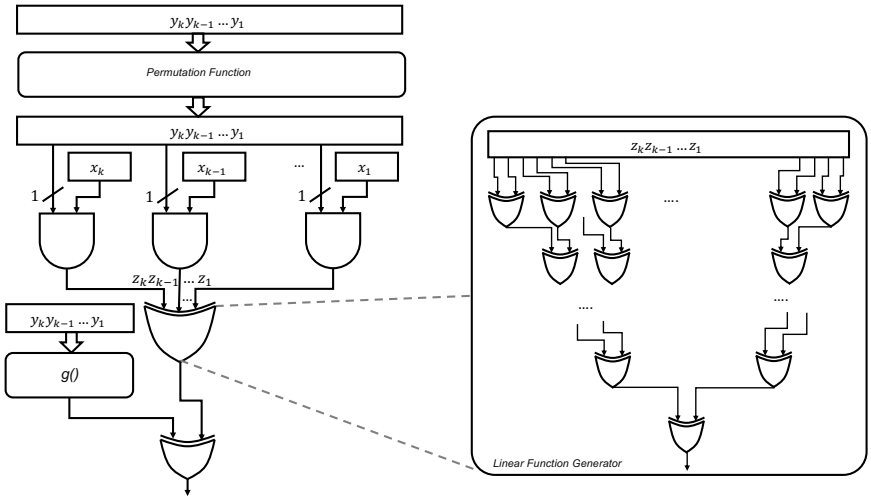


Fig. 2: Circuit implementation of $\mathrm{MM}_0$ functions

Let $\pi(y_k, y_{k-1}, \ldots, y_1) = (y_k, y_{k-1}, \ldots, y_1)$ for all $(y_k, y_{k-1}, \ldots, y_1) \in \mathbb{F}_2^k$. The output of the circuit given in Fig. 2 is $g(y_k, y_{k-1}, \ldots, y_1) \oplus (z_k \oplus z_{k-1} \oplus \cdots \oplus z_1) = y_k x_k \oplus y_{k-1} x_{k-1} \oplus \cdots \oplus y_1 x_1 = (x_k, x_{k-1}, \ldots, x_1) \cdot \pi(y_k, y_{k-1}, \ldots, y_1)$. The smallest case of such a circuit (i.e., $\mathrm{MM}_0$) is when $\pi$ and $g$ are taken as identity permutation and identity 0, respectively. For that, we need $k$ AND gates and $k-1$ XOR gates (or one XOR gate with a fan-in of $k$).

**Lemma 2.** *Implementation of a $2k$-input $\mathrm{MM}_0$ bent function using AND and XOR gates with fan-in of 2, given in Fig. 2, can be done in $\mathcal{O}(n)$-size circuit.*

Instead of identity permutation, we now discuss the cases considering other linear permutations and some nonlinear permutations that can be implemented in polynomial circuit size. Let us denote the set of all invertible and orthogonal binary matrices of order $k \times k$ by $GL(k, \mathbb{F}_2)$ and $SL(k, \mathbb{F}_2)$, respectively.
**Case (i):** Let the permutation be defined as $\pi(\mathbf{y}) = \mathbf{y}A$, for all $\mathbf{y} \in \mathbb{F}_2^k$, where $A \in GL(k, \mathbb{F}_2)$. If $\pi(\mathbf{y}) = (w_k, w_{k-1}, \ldots, w_1)$, then $w_i$, $1 \leq i \leq k$, can be written as a $\mathbb{F}_2$-linear of $\mathbf{y}$. To implement each coordinate of $\pi(\mathbf{y})$, we need $k-1$ XOR gates in the worst case. So, in the worst case, we need $k^2 - k$ XOR gates to implement the permutation $\pi(\mathbf{y}) = \mathbf{y}A$ (in particular, we need strictly $k^2 - k$ XOR gates). In this case, we can implement the function using $k^2 + k$ gates ($k$ AND and $k^2$ XOR gates), i.e., in $\mathcal{O}(k^2)$-size circuit.
**Case (ii):** Let $\pi(\mathbf{y}) = \mathbf{y}A$, for all $\mathbf{y} \in \mathbb{F}_2^k$, where $A \in SL(k, \mathbb{F}_2)$. Here, we need $k$ AND gates and $k-1$ XOR gates (fan-in of 2). So, the implementation of the corresponding $\mathrm{MM}_0$ function in $n = 2k$ variables is in $\mathcal{O}(n)$-size circuit.
**Case (iii):** Instead of linear permutation, we can consider certain kinds of nonlinear permutation, which can be realized in very low-depth and small-size circuits. For a suitable $f$, one can construct a subclass of permutation of the form $\pi(y_k, \ldots, y_2, y_1) = (y_k, \ldots, y_2, h(\mathbf{y}))$, where $h \in \mathcal{B}_k$ such that $deg(h) \geq 2$. The main advantage of such permutation is that one can control the algebraic degree of the permutation by picking suitable $h$, which in turn is reflected in the final function. For example, let $n = 4$ and $\pi(y_4, y_3, y_2, y_1) = (y_4, y_3, y_2, y_4 y_3 y_2 \oplus y_1)$. To implement $\pi$, it requires 2 AND and 1 XOR gates (with fan-in of 2), i.e., in polynomial size.

**Lemma 3.** *The circuit given in Fig. 2 can implement all* MM *bent functions realizable through the circuit given in Fig. 1.*

*Proof.* Since $\pi$ is a permutation over $\mathbb{F}_2^k$, for each $(y_k, y_{k-1}, \ldots, y_1)$, there exists unique $(w_k, w_{k-1}, \ldots, w_1) \in \mathbb{F}_2^k$ such that $(w_k, w_{k-1}, \ldots, w_1) = \pi(y_k, y_{k-1}, \ldots, y_1)$. We consider AND gate with fan-in of 2 for each pair of input $x_i$ and $w_i$, $1 \leq i \leq k$. So, we can implement any MM bent using this circuit. In particular, if $\pi$ is identity, $w_i = y_i$ for all $i = 1, 2, \ldots, k$. □

**Performance Trade-offs for Particular MM Function Construction:** One can get an interesting resource-sharing opportunity. The sub-circuit constructing the linear function can be easily extended by carrying certain input bits to the output. In this idea, both the linear function and the permutation function can be derived from the same circuit. We discuss such possibilities now.

**Construction 1** *The implementation scheme in Fig. 3 provides a* MM *bent function, which can share the same resources for the function g and the permutation π.*

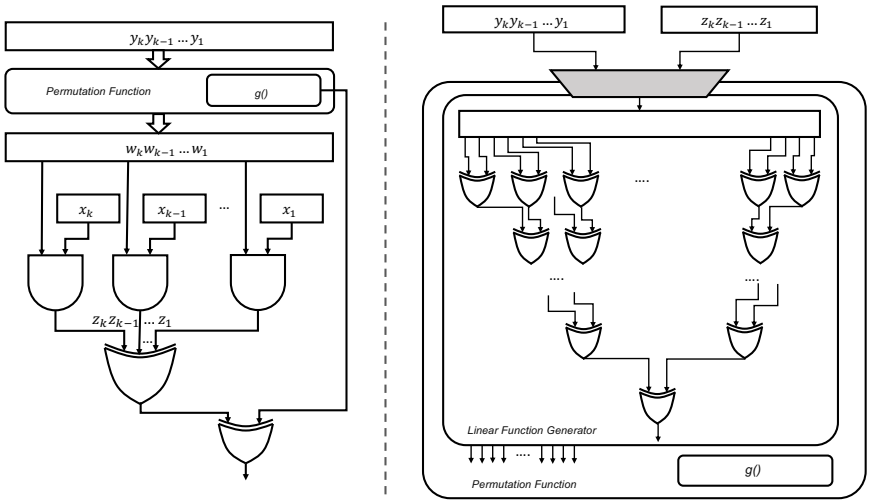


Fig. 3: Variants of MM bent function circuit with performance trade-off.

We schematically capture two variants of MM circuit implementations with resource sharing in Construction 1. Since $\pi$ is a $k$-input, $k$-output function and $g(.)$ is a $k$-input, 1-output function, there is always a choice to select $g(.)$ from the final outputs of $\pi$, or from a sub-circuit of $\pi$, allowing resource sharing. This is shown on the left portion of Fig. 3. Furthermore, as shown on the right part of Fig. 3, it is possible to share the circuits for linear function generator, permutation function $\pi$, and $g(.)$. A multiplexing logic is introduced for different inputs at different stages of computing.

By sharing the implementation of $g(.)$ with $\pi$, there is no performance overhead, though the possible choices of $g(.)$ get restricted. On the other hand, when the $k$-input XOR circuit is shared with the permutation function $\pi$, the circuit needs to process one set of inputs at a time. This is controlled by the multiplexer. Thus, the sharing of combinational logic leads to an increase in the runtime. Note that including multiplexing logic does not extend the complexity of the resulting circuit beyond polynomial size since the multiplexer is implemented using $\frac{n}{2}$ many $2 \times 1$ multiplexers.

## 3 Efficient Circuits for Modified MM Bent Functions

The Circuits given in Fig. 1, 2 and 3, while achieving excellent nonlinearity, do not achieve balancedness and resiliency. In 1994, Dobbertin [7] constructed balanced Boolean functions in even variables $2k$ by modifying the all-zero linear block with balanced functions in $k$ variables. That is, it is to replace the truth table of the all-zero linear function with another nonlinear balanced function on $k$ variables. We can implement this balanced function using a single $2 \times 1$ multiplexer. The balanced Boolean functions with very good autocorrelation spectrum and nonlinearity are recently constructed by modifying the MM functions in [11, 22, 23]. For that, Kavut et al. [11] and Tang et al. [22] modified two fixed positions in each linear block by two nonlinear balanced functions having certain properties. In [23], Tang et al. modified the first all-zero linear block by a nonlinear function in $k$ variables and the all-zero input of all other linear blocks by another nonlinear function in $k$ variables. These two small functions satisfy certain properties. We can implement these balanced functions using two $2 \times 1$ multiplexers. In [23], Tang et al. counted the number of logic gates required to hardware implement the balanced functions given in [22, 23], and it requires the exponential circuits. If we use the $2 \times 1$ multiplexers to modify some outputs of MM functions, then the hardware implementation cost of constructing such balanced functions depends on the hardware implementation cost of permutation and the small functions. Here, we observe that it is possible to construct balanced Boolean functions in polynomial size considering the suitable permutation and the small functions by using $2 \times 1$ multiplexers. Now, we present the hardware implementation of such functions.

### 3.1 Balanced functions given in [7]

Let $n = 2k$, $\pi$ be a permutation over $\mathbb{F}_2^k$ with $\pi(\mathbf{0}) = \mathbf{0}$, and $g \in \mathcal{B}_k$ with $g(\mathbf{y}) = 0$, for all $\mathbf{y} \in \mathbb{F}_2^k$. The new components added in Fig. 4 are a $k$-input XOR gate, a $2 \times 1$ multiplexer and a nonlinear balanced function, denoted as $f(x_k, x_{k-1}, \ldots, x_1)$. It is clear that the outputs of the circuit given in Fig. 4 are the outputs of a balanced Boolean function constructed by Dobbertin [7]. In order to still preserve the polynomial circuit size, the nonlinear balanced function and permutation need to be polynomial size. In that scenario, the proposed balanced function yields a circuit in polynomial size, shown in Fig. 4.

**Lemma 4.** *Let the permutation $\pi(\mathbf{y}) = \mathbf{y}A$, where $A \in GL(k, \mathbb{F}_2)$ and $f$ be in polynomial size. Then, the circuit given in Fig. 4 implements a balanced function with polynomial size.*

*Proof.* The implementation of permutation $\pi$ needs polynomial-size logic gates. To implement this function, we add $k$-input XOR gate, a $2 \times 1$ multiplexer, and a nonlinear function $f$ in $k$ variables in polynomial size. Thus, the constructed balanced Boolean function can be implemented in polynomial size. □

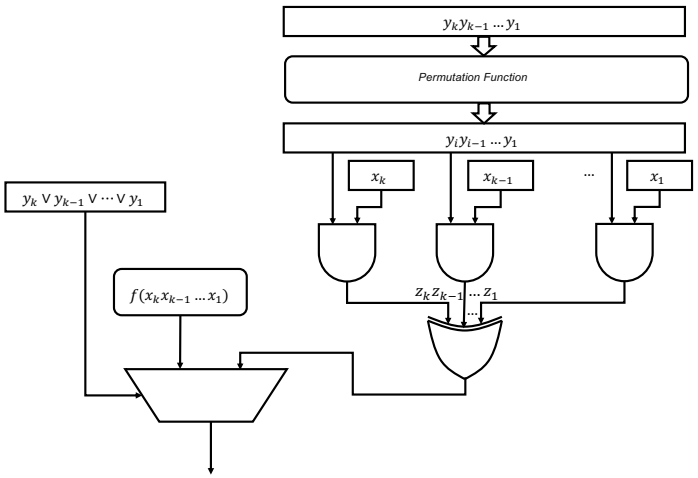


Fig. 4: Circuit implementation for modified MM bent function, Dobbertin construction

Instead of linear permutation, we can consider a nonlinear permutation, which can be realized in very low-depth and small-size circuits. For example we can consider the permutation $\pi(y_k, \dots, y_2, y_1) = (y_k, \dots, y_2, y_k y_{k-1} \cdots y_2 \oplus y_1)$ for $k \geq 3$. To implement $\pi$, we need $k-2$ AND gates and one XOR with a fan-in bound of 2, i.e., in polynomial size. Now, the question is to identify the nonlinear balanced function $f$ in $k$ variables in a polynomial-size circuit. Let us consider $k \geq 3$ and $f(x_1, x_2, \dots, x_k) = x_1 x_2 \oplus x_3 \oplus x_4 \oplus \cdots \oplus x_k$. Then $f$ is a balanced nonlinear function in polynomial size; in particular, we need one AND gate and $k-2$ XOR gates with fan-in bound of 2 for hardware implementation.

*Remark 1.* We like to point out here that the algebraic degree of these functions can be made substantially high so that they can be used as primitives in nonlinear combiner or filter generator model where LFSRs or NFSRs are used. It is possible to implement an MM function in polynomial circuit size as given in Fig. 2 with degree more than 3, i.e., $\deg(\pi) \geq 2$. One can construct a balanced function with high nonlinearity in polynomial size by choosing a suitable nonlinear balanced sub-function in $k$ variables with degree $k-1$ as in Fig. 4. In general, the algebraic degree of balanced Boolean functions constructed in [7, 11, 22, 23] is dependent on the degree of permutation and the sub-functions. Further, it is possible to choose the nonlinear sub-functions in polynomial size.

### 3.2 Balanced functions given in [11, 22, 23]

Kavut et al. [11] and Tang et al. [22, 23] presented constructions of balanced Boolean functions by modifying Maiorana-McFarland bent functions having good cryptographic properties in terms of their autocorrelation spectrum and nonlinearity. Instead of changing one linear block in the MM function, they modified MM functions by using two different nonlinear functions in small variables. The basic idea of this construction is to change the outputs of MM functions at two fixed positions of each linear block or the first all-zero input of each linear block and all all-zero blocks by two suitable small functions. Notably, the total number of gates required for hardware implementation of constructed balanced Boolean function is exponential in $k$, i.e., $\mathcal{O}(k2^k)$, using a decoder, which is exponential in $k$. The modification can be implemented by using two $2 \times 1$ multiplexers. For that, hardware implementation cost mainly depends on permutation $\pi$ and two small functions. We can consider the suitable permutations and small functions so that the hardware implementation of the balanced Boolean function given in [23] is polynomial over the input variables. Let $\pi$ be a permutation over $\mathbb{F}_2^k$ with $\pi(\mathbf{0}) = \mathbf{0}$, $g \in \mathcal{B}_k$ with $g(\mathbf{y}) = 0$, and $f_1, f_2 \in \mathcal{B}_k$ having certain properties. Suppose $h_1, h_2 \in \mathcal{B}_k$ are used to modify the outputs of the bent function at two fixed points. The circuit given in Fig. 5 generates balanced Boolean functions in $2k$ variables constructed in [11, 22, 23] for different $f_i$ and $h_i$, $i = 1, 2$, functions. The outputs of $2 \times 1$ multiplexers are decided by $h_1$ and $h_2$, which

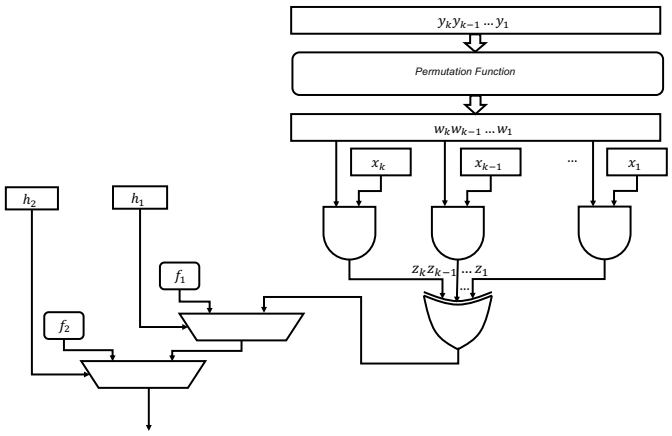


Fig. 5: Construction of balanced Boolean function proposed in [11, 22, 23]

are implemented in polynomial size logic gates. Thus, the total number of gates to hardware implement the circuit given in Fig. 5 depends on permutation $\pi$

and small functions $f_1$ and $f_2$. It is possible to identify a circuit of balanced functions implemented in polynomial size instead of exponential in $k$. Now, we discuss such cases one by one with examples.

**Circuits of balanced Boolean functions given in [11, 22]:** In [11, 22], balanced Boolean functions in $2k$ variables (odd $k \geq 9$ in [22] and even $k \geq 10$ in [11]) with good cryptographic properties are constructed by modifying the simplest $\mathcal{PS}_{ap}$ bent function, a subclass of $\mathcal{PS}^-$, of the form $\mathrm{Tr}_1^k(\frac{\lambda x}{y})$, where $\mathrm{Tr}_1^k(x) = x \oplus x^2 \oplus \cdots \oplus x^{2^{k-1}}$, for all $x, y, \lambda \in \mathbb{F}_{2^k}$ with $\lambda \neq 0$. For that, they first constructed two Boolean function $f_1$ and $f_2$ in $k$ variables such that $wt(f_1) + wt(f_2) = 2^k$. The outputs of $\mathrm{Tr}_1^k(\frac{\lambda x}{y})$ is modified by $f_1(y)$ when $x = 0$, and by $f_2(y)$ when $x = \mu \in \mathbb{F}_{2^k} \setminus \{0\}$. In [23, Theorem 2], it is proved that the bent function $\mathrm{Tr}_1^k(\frac{\lambda x}{y})$ can be written as a concatenation of $2^k$ linear functions and changes are in two fixed points in each linear function. Here, the permutation $y \longmapsto y^{2^k-2}$ is nonlinear over a finite filed $\mathbb{F}_{2^k}$. The function corresponding to $\mathrm{Tr}_1^k(wx)$, where $w = y^{2^k-2}$ and $\lambda = 1$ over vector space $\mathbb{F}_2^k \times \mathbb{F}_2^k$ is fixed, is linear in $k$ variables. We first need to calculate the gate count of this permutation over $\mathbb{F}_2^k$. In 2012, Boyar et al. [1] proved that the hardware implementation of AES S-box requires 128 logic gates with depth 16. The AES S-box is a combination of the multiplicative inverse function over $\mathbb{F}_{2^8}$ and an affine transformation. Thus, the nonlinear permutation $y \longmapsto y^{2^k-2}$ for $k = 8$ might be implemented in polynomial size and depth. So, it might be possible to implement this permutation in polynomial size and depth for some $k \geq 10$. Using the circuit given in Fig. 5, we can generate the balanced function constructed in both [11] and [22]. Let us define

$$h_1(\mathbf{x}) = x_k \vee x_{k-1} \vee \ldots \vee x_1 \text{ and } h_2(\mathbf{x}) = (x_k \oplus \varepsilon_k) \vee (x_{k-1} \oplus \varepsilon_{k-1}) \vee \ldots \vee (x_1 \oplus \varepsilon_1),$$

where $(\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_1) \neq \mathbf{0}$. Thus, $h_1$ and $h_2$ can be implemented in polynomial circuit size. Suppose $\lambda = 1$. Then, the number of gates required for hardware implementation of balanced functions is dependent on the permutation $y \longmapsto y^{2^k-2}$ and small function $f_1$ and $f_2$. We identify the functions $f_1$ and $f_2$ as defined in [23, Definition 3] for small number of inputs $k = 9$ of the form

$$\begin{aligned} f_1(\mathbf{y}) &= y_2 \oplus y_1y_2 \oplus y_2y_3 \oplus y_2y_4 \oplus y_1y_2y_3 \oplus y_1y_2y_4 \oplus y_5y_6y_7 \oplus y_5y_8y_9 \oplus y_6y_8y_9, \\ f_2(\mathbf{y}) &= 1 \oplus y_1 \oplus y_2 \oplus y_1y_2 \oplus y_2y_3 \oplus y_1y_4 \oplus y_2y_4 \oplus y_1y_3y_3 \oplus y_2y_3y_4 \oplus y_5y_7y_9 \\ &\quad \oplus y_6y_7y_9 \oplus y_6y_8y_9. \end{aligned}$$

To hardware implement of $f_1$ (and $f_2$), it is required 13 (14, respectively) AND and 8 (11, respectively) XOR gates. That is, these functions are implemented in polynomial circuit size. We have also identified the functions $f_1$ and $f_2$ as defined in [11, Definition 1] for small number of inputs $k = 10$ of the form

$$\begin{aligned} f_1(\mathbf{y}) &= y_1 \oplus y_2 \oplus y_1y_3 \oplus y_1y_4 \oplus y_2y_4 \oplus y_3y_4 \oplus y_2y_5 \oplus y_4y_5 \oplus y_1y_2y_3 \oplus y_2y_3y_4 \\ &\quad \oplus y_1y_2y_5 \oplus y_1y_4y_5 \oplus y_6y_8y_{10} \oplus y_6y_9y_{10} \oplus y_7y_8y_{10}, \\ f_2(\mathbf{y}) &= 1 \oplus y_2y_3 \oplus y_1y_4 \oplus y_1y_2y_3 \oplus y_1y_2y_4 \oplus y_1y_4y_5 \oplus y_2y_4y_5 \oplus y_6y_8y_{10} \\ &\quad \oplus y_7y_8y_{10} \oplus y_7y_9y_{10}. \end{aligned}$$

To hardware implement of $f_1$ (and $f_2$), it is required 20 (16, respectively) AND and 14 (9, respectively) XOR gates. That is, polynomial-size gates are required to implement these functions. Here, the constructed balanced Boolean functions given in [11, 22] are highly nonlinear and can be implemented in polynomial circuit size.

**Polynomial size circuits of balanced Boolean functions given in [23]:** Tang et al. [23] constructed balanced Boolean functions in $2k$ variables having very high nonlinearity and very low absolute indicator (maximum absolute value of its autocorrelation spectrum) by modifying simple MM bent function of the form $\mathbf{x} \cdot \pi(\mathbf{y})$, where $\pi$ is permutation over $\mathbb{F}_2^k$ with $\pi(\mathbf{0}) = 0$. They used two Boolean function $f_1$ and $f_2$ in $k$ variables so that $f_1(\mathbf{0}) = f_2(\mathbf{0}) = 0$ and $wt(f_1) + wt(f_2) = 2^{k-1}$. The outputs of $\mathbf{x} \cdot \pi(\mathbf{y})$ is modified by $f_1(\mathbf{y})$ when $\mathbf{x} = \mathbf{0}$, and by $f_2(\mathbf{x})$ when $\mathbf{y} = \mathbf{0}$. It is noteworthy that the total number of gates required for hardware implementation of constructed balanced Boolean function is exponential in $k$, i.e., $\mathcal{O}(k2^k)$. Let us define

$$h_1(\mathbf{x}) = x_k \vee x_{k-1} \vee \ldots \vee x_1 \text{ and } h_2(\mathbf{y}) = y_k \vee y_{k-1} \vee \ldots \vee y_1.$$

The circuit given in figure 5 generates balanced Boolean functions in $2k$ variables constructed in [23] for $k \geq 10$. Now, we consider suitable permutations and small functions $f_1$ and $f_1$ to become polynomial-size circuits.

**Lemma 5.** *Let $\pi(\mathbf{y}) = \mathbf{y}A$, where $A \in GL(k, \mathbb{F}_2)$, $f_1$ and $f_2$ be two Boolean functions in $k$ variables in polynomial size such that $f_1(\mathbf{0}) = f_2(\mathbf{0}) = 0$ and $wt(f_1) + wt(f_2) = 2^{k-1}$. Circuits given in figure 5 implement a balanced Boolean function in $2k$ variables with polynomial size.*

*Proof.* Here, $\mathbf{x} \cdot \pi(\mathbf{y})$ can be implemented with $k$ many AND gates and $k^2 - k$ XOR gates (worst case) both fan-in of 2. Additionally, we need two $k$-input $\vee$ gates, two $2 \times 1$ multiplexers, and two nonlinear functions $f_1$ and $f_2$ in polynomial size. Thus, the combined circuit is implemented in polynomial size. □

Instead of linear permutation, we can consider a nonlinear permutation, which can be realized in very low-depth and small-size circuits, as discussed in section 3.1. We identify two functions $f_1$ and $f_2$ as constructed in [23, Theorem 9] and [23, Theorem 12], respectively, for small number of inputs $k = 6$ of the form

$$\begin{aligned} f_1(\mathbf{y}) &= y_1 \oplus y_2 \oplus y_3 \oplus y_4 \oplus y_5 \oplus y_6 \oplus y_1y_3 \oplus y_2y_4 \oplus y_1y_5 \oplus y_3y_5 \oplus y_2y_6 \\ &\quad \oplus y_4y_6 \oplus y_1y_3y_5 \oplus y_2y_4y_6 \\ f_2(\mathbf{x}) &= x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_5 \oplus x_6 \oplus x_1x_3 \oplus x_2x_4 \oplus x_1x_5 \oplus x_3x_5 \oplus x_2x_6 \\ &\quad \oplus x_4x_6 \oplus x_1x_3x_5 \oplus x_2x_4x_6 \oplus x_1x_2x_3x_4 \end{aligned}$$

For hardware implementation of $f_1$ (and $f_2$), we need 10 (13, respectively) AND and 13 (14, respectively) XOR gates. Here $f_1$ and $f_2$ are implemented in polynomial circuit size.

# 4 A class of ultra-lightweight Boolean functions with strong cryptographic properties

In the previous sections, we discussed the hardware implementation cost of some known balanced functions that are constructed by modifying MM bent functions. In particular, if we used multiplexers instead of a decoder, the number of gates required to implement such functions depends on sub-functions. In some known constructions, it is difficult to write the complete algebraic form of sub-functions for all input variables. In this section, we propose a new construction method for balanced Boolean functions by modifying a simple MM bent function. In fact, the constructed balanced functions are in polynomial circuit size. We first constructed two sub-functions with explicit algebraic forms. Let us recall the Maiorana-McFarland (MM) of bent functions, which is defined as $f(\mathbf{x}, \mathbf{y}) = \mathbf{x} \cdot \phi(\mathbf{y}) \oplus g(\mathbf{y})$, where $\mathbf{x}, \mathbf{y} \in \mathbb{F}_2^t$, $\phi$ is an arbitrary permutation on $\mathbb{F}_2^k$, and $g$ is an arbitrary Boolean function in $k$ variables. The MM bent functions were discovered independently by Maiorana and McFarland (see [5, 13]). In this section, we propose a class of ultra-lightweight balanced Boolean functions with good autocorrelation properties and almost optimal nonlinearity. Our construction is based on modifying the MM bent functions using two ultra-lightweight sub-functions.

## 4.1 Two ultra-lightweight sub-functions for main construction

We first define a balanced Boolean function in $t \geq 5$ variables, which is a concatenation of certain bent functions.

**Definition 1.** *Let $t$ be an integer such that $t \geq 5$. Let $g$ be a $t$-variable Boolean function defined as follows:*

$$g(\mathbf{r}) = \bigoplus_{i=1}^{\iota} r_i r_{i+\iota} \oplus \bigoplus_{i=2\iota+1}^{t} r_i,$$

*where $\mathbf{r} = (r_1, r_2, \ldots, r_t) \in \mathbb{F}_2^t$ and $\iota = \lfloor \frac{t-1}{2} \rfloor$.*

Since $\oplus_{i=1}^{\iota} r_i r_{i+\iota}$ is a bent function in $2\iota$ variables. If $t$ is odd, then $\iota = \frac{t-1}{2}$, so the function $g$ is a concatenation of two bent functions in $t-1$ variables, a bent and its complementary. If $t$ is even, then $\iota = \frac{t}{2} - 1$, and so, the function $g$ is a concatenation of four bent functions in $t-2$ variables of the form

$$\oplus_{i=1}^{\iota} r_i r_{i+\iota} || \oplus_{i=1}^{\iota} r_i r_{i+\iota} \oplus 1 || \oplus_{i=1}^{\iota} r_i r_{i+\iota} \oplus 1 || \oplus_{i=1}^{\iota} r_i r_{i+\iota}.$$

It is clear that $g$ is a balanced function in $t$ variables.

**Lemma 6.** *Let $t \geq 5$ be an integer and $g \in \mathcal{B}_t$ be the function defined in Definition 1. Then for any $\mathbf{d} = (d_1, d_2, \cdots, d_t) \in \mathbb{F}_2^t$, if $t$ is even, we have*

$$W_g(\mathbf{d}) = \begin{cases} 0, & \text{if } (d_t, d_{t-1}) \in \{(0,0), (0,1), (1,0)\} \\ (-1)^{\bigoplus_{i=1}^{\iota} d_i d_{i+\iota}} \cdot 2^{\frac{t}{2}+1}, & \text{if } (d_t, d_{t-1}) = (1,1) \end{cases},$$

*and if $t$ is odd, we have*

$$W_g(\mathbf{d}) = \begin{cases} 0, & \text{if } d_t = 0 \\ (-1)^{\bigoplus_{i=1}^{\iota} d_i d_{i+\iota}} \cdot 2^{\frac{t+1}{2}}, & \text{if } d_t = 1 \end{cases},$$

*where $\iota = \lfloor \frac{t-1}{2} \rfloor$.*

*Proof.* Let $g_1(\mathbf{x}) = \oplus_{i=1}^{\iota} x_i x_{i+\iota}$, where $\mathbf{x} \in \mathbb{F}_2^{2\iota}$, and $t$ be an even integer. Then $\iota = \frac{t}{2} - 1$ and $g(\mathbf{r}) = \oplus_{i=1}^{\iota} r_i r_{i+\iota} \oplus r_{t-1} \oplus r_t$. For any $\mathbf{d} = (\mathbf{d}', d_{t-1}, d_t) \in \mathbb{F}_2^{t-2} \times \mathbb{F}_2 \times \mathbb{F}_2$, we have

$$\begin{aligned} W_g(\mathbf{d}) &= \sum_{(\mathbf{r}', r_{t-1}, r_t) \in \mathbb{F}_2^{t-2} \times \mathbb{F}_2 \times \mathbb{F}_2} (-1)^{g(\mathbf{r}', r_{t-1}, r_t) \oplus \mathbf{d}' \cdot \mathbf{r}' \oplus d_{t-1} r_{t-1} \oplus d_t r_t} \\ &= \sum_{\mathbf{r}' \in \mathbb{F}_2^{t-2}} (-1)^{g_1(\mathbf{r}') \oplus \mathbf{d}' \cdot \mathbf{r}'} + \sum_{\mathbf{r}' \in \mathbb{F}_2^{t-2}} (-1)^{g_1(\mathbf{r}') \oplus 1 \oplus \mathbf{d}' \cdot \mathbf{r}' \oplus d_{t-1}} \\ &\quad + \sum_{\mathbf{r}' \in \mathbb{F}_2^{t-2}} (-1)^{g_1(\mathbf{r}') \oplus 1 \oplus \mathbf{d}' \cdot \mathbf{r}' \oplus d_t} + \sum_{\mathbf{r}' \in \mathbb{F}_2^{t-2}} (-1)^{g_1(\mathbf{r}') \oplus \mathbf{d}' \cdot \mathbf{r}' \oplus d_{t-1} \oplus d_t} \\ &= \big(1 + (-1)^{d_{t-1} \oplus 1} + (-1)^{d_t \oplus 1} + (-1)^{d_{t-1} \oplus d_t}\big) W_{g_1}(\mathbf{d}') \\ &= \begin{cases} 0, & \text{if } (d_t, d_{t-1}) \in \{(0,0), (0,1), (1,0)\} \\ 2^{\frac{t}{2}+1} (-1)^{\bigoplus_{i=1}^{\iota} d_i d_{i+\iota}}, & \text{if } (d_t, d_{t-1}) = (1,1) \end{cases}. \end{aligned}$$

Suppose $t$ is an odd integer. Then $\iota = \frac{t-1}{2}$ and $g(\mathbf{r}) = \oplus_{i=1}^{\iota} r_i r_{i+\iota} \oplus r_t$. For any $\mathbf{d} = (\mathbf{d}', d_t) \in \mathbb{F}_2^{t-1} \times \mathbb{F}_2$, we have

$$\begin{aligned} W_g(\mathbf{d}) &= \sum_{\mathbf{r} \in \mathbb{F}_2^t} (-1)^{g(\mathbf{r}) \oplus \mathbf{d} \cdot \mathbf{r}} = \sum_{(\mathbf{r}', r_t) \in \mathbb{F}_2^{t-1} \times \mathbb{F}_2} (-1)^{g(\mathbf{r}', r_t) \oplus \mathbf{d}' \cdot \mathbf{r}' \oplus d_t r_t} \\ &= \sum_{\mathbf{r}' \in \mathbb{F}_2^{t-1}} (-1)^{g_1(\mathbf{r}') \oplus \mathbf{d}' \cdot \mathbf{r}'} + \sum_{\mathbf{r}' \in \mathbb{F}_2^{t-1}} (-1)^{g_1(\mathbf{r}') \oplus 1 \oplus \mathbf{d}' \cdot \mathbf{r}' \oplus d_t} \\ &= \big(1 + (-1)^{d_t \oplus 1}\big) W_{g_1}(\mathbf{d}') \\ &= \begin{cases} 0, & \text{if } d_t = 0 \\ 2^{\frac{t+1}{2}} (-1)^{\bigoplus_{i=1}^{\iota} d_i d_{i+\iota}}, & \text{if } d_t = 1 \end{cases}. \end{aligned}$$

□

Thus, the maximum absolute Walsh–Hadamard spectrum value of $g \in \mathcal{B}_t$ is $2^{\frac{t}{2}+1}$ (and $2^{\frac{t+1}{2}}$) for $t$ is even (odd, respectively).

**Lemma 7.** *Let $t \geq 5$ be an integer and $g \in \mathcal{B}_t$ be the function defined in Definition 1. Then for any $\mathbf{d} = (d_1, d_2, \cdots, d_t) \in \mathbb{F}_2^t$, if $t$ is even, we have*

$$C_g(\mathbf{d}) = \begin{cases} 2^t (-1)^{d_{t-1} \oplus d_t}, & \text{if } (d_1, d_2 \ldots, d_{t-2}) = (0, 0, \ldots, 0) \\ 0, & \text{otherwise} \end{cases},$$

*and if $t$ is odd, we have*

$$C_g(\mathbf{d}) = \begin{cases} 2^t (-1)^{d_t}, & \text{if } (d_1, d_2 \ldots, d_{t-1}) = (0, 0, \ldots, 0) \\ 0, & \text{otherwise,} \end{cases}.$$

*Proof.* Let $\iota = \lfloor \frac{t-1}{2} \rfloor$ and $\mathbf{b} = (d_{\iota+1}, \ldots, d_{2\iota}, d_1, \ldots, d_\iota)$ for $\mathbf{d} \in \mathbb{F}_2^t$. Suppose $\mathbf{r}' = (r_1, r_2, \ldots, r_{2\iota})$ for any $\mathbf{r} \in \mathbb{F}_2^t$. Let $t \geq 5$ be an even integer. Then $2\iota = t-2$ and

$$\begin{aligned} Cg(\mathbf{d}) &= \sum_{\mathbf{r}\in\mathbb{F}_2^t} (-1)^{g(\mathbf{r}\oplus\mathbf{d})\oplus g(\mathbf{r})} = \sum_{\mathbf{r}\in\mathbb{F}_2^t} (-1)^{\bigoplus_{i=1}^{\iota} \left((r_i\oplus d_i)(r_{i+\iota}\oplus d_{i+\iota})\oplus r_i r_{i+\iota}\right)\oplus d_{t-1}\oplus d_t} \\ &= \sum_{\mathbf{r}\in\mathbb{F}_2^t} (-1)^{\oplus_{i=1}^{\iota}(r_i d_{i+\iota}\oplus r_{i+\iota} d_i)\oplus\oplus_{i=1}^{\iota} d_i d_{i+\iota}\oplus d_{t-1}\oplus d_t} \\ &= (-1)^{g(\mathbf{d})} \sum_{\mathbf{r}\in\mathbb{F}_2^t} (-1)^{\mathbf{r}'\cdot\mathbf{b}} \\ &= \begin{cases} 2^t(-1)^{d_{t-1}\oplus d_t}, & \text{if } (d_1, d_2 \ldots, d_{t-2}) = (0, 0, \ldots, 0) \\ 0, & \text{if otherwise} \end{cases}. \end{aligned}$$

Suppose that $t \geq 5$ is an odd integer. Then $2\iota = t-1$, and similarly, we get the results. □

Now, we define the two Boolean functions in $2t \geq 10$ variables, which are used to construct the sub-functions.

**Definition 2.** *Let $k = 2t \geq 10$ be an even integer. We define two Boolean functions $p, q \in \mathcal{B}_k$ as follows:*

$$p(\mathbf{z}, \mathbf{r}) = \bigoplus_{i=1}^{t} z_i r_i,$$

*and*

$$\begin{aligned} q(\mathbf{z}, \mathbf{r}) &= \begin{cases} g(\mathbf{r}), & \text{if } \mathbf{z} = \mathbf{0} \\ \bigoplus_{i=2}^{t} z_i r_{i-1} \oplus z_1 r_t \oplus z_2 r_t, & \text{otherwise} \end{cases} \\ &= g(\mathbf{r}) \prod_{i=1}^{t} (z_i \oplus 1) \oplus \bigoplus_{i=2}^{t} z_i r_{i-1} \oplus z_1 r_t \oplus z_2 r_t, \end{aligned}$$

*where $\mathbf{z} = (z_1, z_2, \ldots, z_t), \mathbf{r} = (r_1, r_2, \ldots, r_t) \in \mathbb{F}_2^t$ and $g \in \mathcal{B}_t$ is given in Definition 1.*

Here, $p$ is a quadratic MM bent function in $2t$ variables, and the algebraic degree of $q \in \mathcal{B}_{2t}$ is $t+2$. If we consider a zero function for $\mathbf{z} = \mathbf{0}$ instead of $g$, then $q$ is also a quadratic bent function.

**Lemma 8.** *Let $t \geq 5$ be an arbitrary integer and $\phi(\mathbf{z}) = (z_2, z_3, \ldots, z_{t-1}, z_t, z_1 \oplus z_2)$, where $\mathbf{z} = (z_1, z_2, \ldots, z_t) \in \mathbb{F}_2^t$. Then both $\phi(\mathbf{z})$ and $\phi(\mathbf{z}) \oplus \mathbf{z}$ are permutations over $\mathbb{F}_2^t$. Moreover, the composition inverse of $\phi(\mathbf{z})$ is $\phi^{-1}(\mathbf{z}) = (z_1 \oplus z_t, z_1, z_2, \ldots, z_{t-1})$.*

*Proof.* It is clear that $\phi$ is a permutation over $\mathbb{F}_2^t$. Let $\mathbf{x}, \mathbf{y} \in \mathbb{F}_2^t$ such that $\phi(\mathbf{x}) \oplus \mathbf{x} = \phi(\mathbf{y}) \oplus \mathbf{y}$. Then

$$\begin{aligned}
&(x_1 \oplus x_2, \ldots, x_{t-1} \oplus x_t, x_1 \oplus x_2 \oplus x_t) = (y_1 \oplus y_2, \ldots, y_{t-1} \oplus y_t, y_1 \oplus y_2 \oplus y_t) \\
\Leftrightarrow\ & x_i \oplus x_{i+1} = y_i \oplus y_{i+1}, \text{ for all } 1 \le i \le t-1 \text{ and } x_1 \oplus x_2 \oplus x_t = y_1 \oplus y_2 \oplus y_t \\
\Leftrightarrow\ & x_i = y_i, \text{ for all } 1 \le i \le t.
\end{aligned}$$

Further, the composition inverse functions of $\phi$ is $\phi^{-1}(\mathbf{z})) = (z_1 \oplus z_t, z_1, \ldots, z_{t-1})$. □

**Lemma 9.** *Let $k = 2t \ge 10$ be an even integer and $p, q \in \mathcal{B}_k$ be the two functions defined in Definition 2. Then for any $\mathbf{c} = (c_1, c_2, \ldots, c_t), \mathbf{d} = (d_1, d_2, \ldots, d_t) \in \mathbb{F}_2^t$ we have*

$$W_p(\mathbf{c}, \mathbf{d}) = (-1)^{\mathbf{c}\cdot\mathbf{d}} \cdot 2^t,$$

$$W_q(\mathbf{c}, \mathbf{d}) = \begin{cases} 0, & \text{if } \mathbf{d} = \mathbf{0} \\ (-1)^{\mathbf{c}\cdot\mathbf{d}'} \cdot 2^t + W_g(\mathbf{d}), & \text{otherwise} \end{cases},$$

*and*

$$W_{p\oplus q}(\mathbf{c}, \mathbf{d}) = \begin{cases} 0, & \text{if } \mathbf{d} = \mathbf{0} \\ (-1)^{\mathbf{c}\cdot\mathbf{d}''} \cdot 2^t + W_g(\mathbf{d}), & \text{otherwise} \end{cases},$$

*$W_g(\mathbf{d})$ is given in Lemma 6, $\mathbf{d}'$ and $\mathbf{d}''$ are the composition inverse of $\phi(\mathbf{d})$ and $\phi(\mathbf{d}) \oplus \mathbf{d}$, respectively, where $\phi(\mathbf{d}) = (d_2, d_3, \ldots, d_{t-1}, d_t, d_1 \oplus d_2)$.*

*Proof.* Since $p$ is a bent function in $2t$ variables, for any $\mathbf{c}, \mathbf{d} \in \mathbb{F}_2^t$, we have $W_p(\mathbf{c}, \mathbf{d}) = 2^t(-1)^{\mathbf{c}\cdot\mathbf{d}}$. For any $\mathbf{c}, \mathbf{d} \in \mathbb{F}_2^t$,

$$\begin{aligned}
W_q(\mathbf{c}, \mathbf{d}) &= \sum_{\mathbf{z},\mathbf{r}\in\mathbb{F}_2^t} (-1)^{q(\mathbf{z},\mathbf{r})\oplus\mathbf{c}\cdot\mathbf{z}\oplus\mathbf{d}\cdot\mathbf{r}} \\
&= \sum_{\mathbf{r}\in\mathbb{F}_2^t} (-1)^{g(\mathbf{r})\oplus\mathbf{d}\cdot\mathbf{r}} + \sum_{\mathbf{z}\in\mathbb{F}_2^{t*}} \sum_{\mathbf{r}\in\mathbb{F}_2^t} (-1)^{\bigoplus_{i=2}^t z_i r_{i-1} \oplus z_1 r_t \oplus z_2 r_t \oplus \mathbf{c}\cdot\mathbf{z} \oplus \mathbf{d}\cdot\mathbf{r}} \\
&= W_g(\mathbf{d}) + \sum_{\mathbf{z}\in\mathbb{F}_2^{t*}} (-1)^{\mathbf{c}\cdot\mathbf{z}} \sum_{\mathbf{r}\in\mathbb{F}_2^t} (-1)^{\bigoplus_{i=2}^t z_i r_{i-1} \oplus z_1 r_t \oplus z_2 r_t \oplus \mathbf{d}\cdot\mathbf{r}} \\
&= W_g(\mathbf{d}) + \sum_{\mathbf{z}\in\mathbb{F}_2^{t*}} (-1)^{\mathbf{c}\cdot\mathbf{z}} \sum_{\mathbf{r}\in\mathbb{F}_2^t} (-1)^{(\mathbf{a}\oplus\mathbf{d})\cdot\mathbf{r}} \\
&= W_g(\mathbf{d}) + 2^t(-1)^{\mathbf{c}\cdot\mathbf{d}'},
\end{aligned}$$

where $\mathbf{a} = (z_2, z_3, \ldots, z_t, z_1 \oplus z_2)$ and $\mathbf{d}' = (d_1 \oplus d_t, d_1, \ldots, d_t)$. Since $W_g(\mathbf{0}) = 0$ and $\sum_{\mathbf{r}\in\mathbb{F}_2^t}(-1)^{\bigoplus_{i=2}^t z_i r_{i-1} \oplus z_1 r_t \oplus z_2 r_t} = 0$ for $\mathbf{z} \neq \mathbf{0}$, we have $W_q(\mathbf{c}, \mathbf{0}) = 0$. Also

$$\begin{aligned}
W_{p\oplus q}(\mathbf{c}, \mathbf{d}) &= \sum_{\mathbf{z},\mathbf{r}\in\mathbb{F}_2^t} (-1)^{p(\mathbf{z},\mathbf{r})\oplus q(\mathbf{z},\mathbf{r})\oplus \mathbf{c}\cdot\mathbf{z}\oplus\mathbf{d}\cdot\mathbf{r}} \\
&= \sum_{\mathbf{r}\in\mathbb{F}_2^t}(-1)^{g(\mathbf{r})\oplus\mathbf{d}\cdot\mathbf{r}} + \sum_{\mathbf{z}\in\mathbb{F}_2^{t*}}\sum_{\mathbf{r}\in\mathbb{F}_2^t}(-1)^{\mathbf{z}\cdot\mathbf{r}\oplus\bigoplus_{i=2}^t z_i r_{i-1}\oplus z_1 r_t\oplus z_2 r_t\oplus\mathbf{c}\cdot\mathbf{z}\oplus\mathbf{d}\cdot\mathbf{r}} \\
&= W_g(\mathbf{d}) + \sum_{\mathbf{z}\in\mathbb{F}_2^{t*}}(-1)^{\mathbf{c}\cdot\mathbf{z}}\sum_{\mathbf{r}\in\mathbb{F}_2^t}(-1)^{\mathbf{z}\cdot\mathbf{r}\oplus\bigoplus_{i=2}^t z_i r_{i-1}\oplus z_1 r_t\oplus z_2 r_t\oplus\mathbf{d}\cdot\mathbf{r}} \\
&= W_g(\mathbf{d}) + \sum_{\mathbf{z}\in\mathbb{F}_2^{t*}}(-1)^{\mathbf{c}\cdot\mathbf{z}}\sum_{\mathbf{r}\in\mathbb{F}_2^t}(-1)^{(\mathbf{b}\oplus\mathbf{d})\cdot\mathbf{r}} \\
&= W_g(\mathbf{d}) + 2^t(-1)^{\mathbf{c}\cdot\mathbf{d}''},
\end{aligned}$$

where $\mathbf{b} = (z_1 \oplus z_2, \ldots, z_{t-1} \oplus z_t, z_1 \oplus z_2 \oplus z_t)$ and $\mathbf{d}'' = (\bigoplus_{i=2}^t d_i, \bigoplus_{i=1}^t d_i, d_1 \oplus \bigoplus_{i=3}^t d_i, \ldots, d_1 \oplus d_t)$. Since $W_g(\mathbf{0}) = 0$ and $\sum_{\mathbf{r}\in\mathbb{F}_2^t}(-1)^{\mathbf{z}\cdot\mathbf{r}\oplus\bigoplus_{i=2}^t z_i r_{i-1}\oplus z_1 r_t\oplus z_2 r_t} = 0$ for $\mathbf{z} \neq \mathbf{0}$, we have $W_{p\oplus q}(\mathbf{c}, \mathbf{0}) = 0$. □

It is clear that $|W_p(\mathbf{c}, \mathbf{d})| = 2^t$ for all $\mathbf{c}, \mathbf{d} \in \mathbb{F}_2^t$, and

$$|W_q(\mathbf{c}, \mathbf{d})|, |W_{p\oplus q}(\mathbf{c}, \mathbf{d})| \leq \begin{cases} 2^t + 2^{\frac{t}{2}+1}, \text{ if } t \text{ is even} \\ 2^t + 2^{\frac{t+1}{2}}, \text{ if } t \text{ is odd} \end{cases}.$$

**Lemma 10.** *Let $k = 2t \geq 10$ be an even integer and $p, q \in \mathcal{B}_k$ be the two functions defined in Definition 2. Then for any $\mathbf{c} = (c_1, c_2, \ldots, c_t), \mathbf{d} = (d_1, d_2, \ldots, d_t) \in \mathbb{F}_2^t$ we have*

$$C_p(\mathbf{c}, \mathbf{d}) = \begin{cases} 2^k, \text{ if } \mathbf{c} = \mathbf{d} = \mathbf{0} \\ 0, \text{ otherwise} \end{cases},$$

$$C_q(\mathbf{c}, \mathbf{d}) = \begin{cases} 2^k, & \text{if } \mathbf{c} = \mathbf{d} = \mathbf{0} \\ C_g(\mathbf{d}) - 2^t, & \text{if } \mathbf{c} = \mathbf{0}, \mathbf{d} \neq \mathbf{0} \\ 2(-1)^{\mathbf{c}'\cdot\mathbf{d}} W_g(\mathbf{c}'), & \text{if } \mathbf{c} \neq \mathbf{0} \end{cases},$$

*where $\mathbf{c}' = (c_2, c_3, \ldots, c_t, c_1 \oplus c_2)$ for $\mathbf{c} \in \mathbb{F}_2^t$, and $W_g(\mathbf{c}')$ and $C_g(\mathbf{d})$ are given in Lemma 6 and Lemma 7, respectively.*

*Proof.* Since $p$ is a bent function in $2t$ variables, $C_p(\mathbf{0}, \mathbf{0}) = 2^{2t}$ and $C_p(\mathbf{c}, \mathbf{d}) = 0$ for all nonzero $\mathbf{c}, \mathbf{d} \in \mathbb{F}_2^t$. we have derived the autocorrelation values of $q$ in different cases.
Case (i): Let $\mathbf{c} = \mathbf{0}$. Then

$$q(\mathbf{z}, \mathbf{r}) \oplus q(\mathbf{z}, \mathbf{r} \oplus \mathbf{d}) = \begin{cases} g(\mathbf{r}) \oplus g(\mathbf{r} \oplus \mathbf{d}), & \text{if } \mathbf{z} = \mathbf{0} \\ \oplus_{i=2}^t z_i d_{i-1} \oplus z_1 d_t \oplus z_2 d_t, & \text{otherwise} \end{cases},$$

and so,

$$\begin{aligned}C_q(\mathbf{0},\mathbf{d}) &= \sum_{\mathbf{r}\in\mathbb{F}_2^t}(-1)^{g(\mathbf{r})\oplus g(\mathbf{r}\oplus\mathbf{d})} + \sum_{\mathbf{z}\in\mathbb{F}_2^{t*}}\sum_{\mathbf{r}\in\mathbb{F}_2^t}(-1)^{\oplus_{i=2}^t z_i d_{i-1}\oplus z_1 d_t\oplus z_2 d_t}\\ &= C_g(\mathbf{d}) + 2^t\Big(\sum_{\mathbf{z}\in\mathbb{F}_2^t}(-1)^{\mathbf{d}.\mathbf{z}'} - 1\Big),\ \text{where } \mathbf{z}' = (z_2, z_3, \ldots, z_t, z_1\oplus z_2)\\ &= C_g(\mathbf{d}) + 2^{2t}\delta_{\mathbf{0}}(\mathbf{d}) - 2^t\end{aligned}$$

Here $(z_1,\ldots,z_{t-1},z_t)\longmapsto(z_2,\ldots,z_t,z_1\oplus z_2)$ is a permutation over $\mathbb{F}_2^t$. If $\mathbf{d}=\mathbf{0}$, then $C_q(\mathbf{0},\mathbf{0}) = 2^k$, and if $\mathbf{d}\neq\mathbf{0}$, then $C_q(\mathbf{0},\mathbf{d}) = C_g(\mathbf{d}) - 2^t$.
Case (ii): Let $\mathbf{c}\neq\mathbf{0}$ and $\mathbf{d}=\mathbf{0}$. Suppose $\mathbf{c}' = (c_2, c_3, \ldots, c_t, c_1\oplus c_2)$ for any nonzero $\mathbf{c}\in\mathbb{F}_2^t$. Then

$$\begin{aligned}C_q(\mathbf{c},\mathbf{0}) &= \sum_{\mathbf{z},\mathbf{r}\in\mathbb{F}_2^t}(-1)^{q(\mathbf{z},\mathbf{r})\oplus q(\mathbf{z}\oplus\mathbf{c},\mathbf{r})}\\ &= 2\sum_{\mathbf{r}\in\mathbb{F}_2^t}(-1)^{g(\mathbf{r})\oplus(\oplus_{i=2}^t c_i r_{i-1})\oplus c_1 r_t\oplus c_2 r_t}\\ &\quad + \sum_{\mathbf{z}\in\mathbb{F}_2^t\setminus\{\mathbf{0},\mathbf{c}\}}\sum_{\mathbf{r}\in\mathbb{F}_2^t}(-1)^{\oplus_{i=2}^t c_i r_{i-1}\oplus c_1 r_t\oplus c_2 r_t}\\ &= 2\sum_{\mathbf{r}\in\mathbb{F}_2^t}(-1)^{g(\mathbf{r})\oplus\mathbf{c}'\cdot\mathbf{r}} + (2^t-2)\sum_{\mathbf{r}\in\mathbb{F}_2^t}(-1)^{\mathbf{c}'\cdot\mathbf{r}}\\ &= 2W_g(\mathbf{c}') + 2^t(2^t-2)\delta_{\mathbf{0}}(\mathbf{c}') = 2W_g(\mathbf{c}'),\ \text{as } \mathbf{c}\neq\mathbf{0}.\end{aligned}$$

Case (iii): Let $\mathbf{c}\neq\mathbf{0}$ and $\mathbf{d}\neq\mathbf{0}$. Then we have

$$\begin{aligned}q(\mathbf{z},\mathbf{r})\oplus q(\mathbf{z}\oplus\mathbf{c},\mathbf{r}\oplus\mathbf{d}) &= \begin{cases} g(\mathbf{r})\oplus\oplus_{i=2}^t c_i r_{i-1}\oplus c_1 r_t\oplus c_2 r_t\oplus & \\ \oplus_{i=2}^t c_i d_{i-1}\oplus c_1 d_t\oplus c_2 d_t, & \text{if } \mathbf{z}=\mathbf{0}\\ g(\mathbf{r}\oplus\mathbf{d})\oplus\oplus_{i=2}^t c_i r_{i-1}\oplus c_1 r_t\oplus c_2 r_t, & \text{if } \mathbf{z}=\mathbf{c}\\ \oplus_{i=2}^t z_i d_{i-1}\oplus z_1 d_t\oplus z_2 d_t\oplus & \\ \oplus_{i=2}^t c_i r_{i-1}\oplus c_1 r_t\oplus c_2 r_t\oplus & \\ \oplus_{i=2}^t c_i d_{i-1}\oplus c_1 d_t\oplus c_2 d_t, & \text{if } \mathbf{z}\neq\mathbf{0},\mathbf{c}\end{cases},\\ &= \begin{cases} g(\mathbf{r})\oplus\mathbf{c}'\cdot\mathbf{r}\oplus\mathbf{c}'\cdot\mathbf{d}, & \text{if } \mathbf{z}=\mathbf{0}\\ g(\mathbf{r}\oplus\mathbf{d})\oplus\mathbf{c}'\cdot\mathbf{r}, & \text{if } \mathbf{z}=\mathbf{c}\\ \mathbf{z}'\cdot\mathbf{d}\oplus\mathbf{c}'\cdot\mathbf{r}\oplus\mathbf{c}'\cdot\mathbf{d}, & \text{if } \mathbf{z}\neq\mathbf{0},\mathbf{c}\end{cases},\end{aligned}$$

where $\mathbf{z}' = (z_2, z_3, \ldots, z_t, z_1\oplus z_2)$ for $\mathbf{z}\in\mathbb{F}_2^n$, and so

$$\begin{aligned}C_q(\mathbf{c},\mathbf{d}) &= \sum_{\mathbf{z},\mathbf{r}\in\mathbb{F}_2^t}(-1)^{q(\mathbf{z},\mathbf{r})\oplus q(\mathbf{z}\oplus\mathbf{c},\mathbf{r}\oplus\mathbf{d})}\\ &= \sum_{\mathbf{r}\in\mathbb{F}_2^t}(-1)^{g(\mathbf{r})\oplus\mathbf{c}'\cdot\mathbf{r}\oplus\mathbf{c}'\cdot\mathbf{d}} + \sum_{\mathbf{r}\in\mathbb{F}_2^t}(-1)^{g(\mathbf{r}\oplus\mathbf{d})\oplus\mathbf{c}'\cdot\mathbf{r}}\\ &\quad + \sum_{\mathbf{z}\in\mathbb{F}_2^t\setminus\{\mathbf{0},\mathbf{c}\}}\sum_{\mathbf{r}\in\mathbb{F}_2^t}(-1)^{\mathbf{z}'\cdot\mathbf{d}\oplus\mathbf{c}'\cdot\mathbf{r}\oplus\mathbf{c}'\cdot\mathbf{d}}\\ &= 2(-1)^{\mathbf{c}'\cdot\mathbf{d}}W_g(\mathbf{c}'),\ \text{as } \mathbf{c}\neq\mathbf{0}.\end{aligned}$$

□

**Definition 3.** *Let $k = 2t \geq 10$ be an even integer. We define two Boolean functions $u, v \in \mathcal{B}_k$ as follows:*

$$u(\mathbf{z}, \mathbf{r}) = p(\mathbf{z}, \mathbf{r})q(\mathbf{z}, \mathbf{r}) = \bigoplus_{i=1}^{t} z_i r_i \bigoplus_{i=2}^{t} z_i r_{i-1} \oplus (z_1 r_t \oplus z_2 r_t) \bigoplus_{i=1}^{t} z_i r_i,$$

*and*

$$\begin{aligned} v(\mathbf{z}, \mathbf{r}) &= \big(p(\mathbf{z}, \mathbf{r}) \oplus 1\big)q(\mathbf{z}, \mathbf{r}) = u(\mathbf{z}, \mathbf{r}) \oplus q(\mathbf{z}, \mathbf{r}) \\ &= g(\mathbf{r}) \prod_{i=1}^{t} (z_i \oplus 1) \oplus \big(\bigoplus_{i=1}^{t} z_i r_i \oplus 1\big) \bigoplus_{i=2}^{t} z_i r_{i-1} \oplus (z_1 r_t \oplus z_2 r_t)\big(\bigoplus_{i=1}^{t} z_i r_i \oplus 1\big), \end{aligned}$$

*where $\mathbf{z} = (z_1, z_2, \ldots, z_t), \mathbf{r} = (r_1, r_2, \ldots, r_t) \in \mathbb{F}_2^t$ and $g \in \mathcal{B}_t$ is given in Definition 1.*

It is direct that the algebraic degree of the functions $u$ and $v$ in $2t \geq 10$ variables are 4 and $t + 2$, respectively. Now, we derive the Walsh spectrum values of the functions $u$ and $v$ given in Definition 3 using the following known results.

**Lemma 11 ([23]).** *Let $u_1, u_2$ be two $k$-variable Boolean functions and define $u(\mathbf{x}) = u_1(\mathbf{x})u_2(\mathbf{x})$, where $\mathbf{x} = (x_1, x_2, \ldots, x_k) \in \mathbb{F}_2^k$. Then for any $\mathbf{b} \in \mathbb{F}_2^k$ we have*

$$W_u(\mathbf{b}) = 2^{k-1}\delta_{\mathbf{0}}(\mathbf{b}) + \frac{1}{2}\Big(W_{u_1}(\mathbf{b}) + W_{u_2}(\mathbf{b}) - W_{u_1 \oplus u_2}(\mathbf{b})\Big),$$

*where $\delta_{\mathbf{0}}(\cdot)$ is the Dirac (or Kronecker) symbol, which is defined by $\delta_{\mathbf{0}}(\mathbf{b}) = 1$ if $\mathbf{b} = \mathbf{0}$ and $\delta_{\mathbf{0}}(\mathbf{b}) = 0$ otherwise. For any $\mathbf{d} \in \mathbb{F}_2^{k*}$, we have*

$$\begin{aligned} C_u(\mathbf{d}) = 2^{k-2} + \frac{1}{4}\Big(C_{u_1}(\mathbf{d}) + C_{u_2}(\mathbf{d}) + C_{u_1 \oplus u_2}(\mathbf{d})\Big) + \frac{1}{2}\Big(W_{u_1}(\mathbf{0}) + W_{u_2}(\mathbf{0}) \\ - W_{u_1 \oplus u_2}(\mathbf{0})\Big) + \frac{1}{2}\Big(C_{u_1, u_2}(\mathbf{d}) - C_{u_1, u_1 \oplus u_2}(\mathbf{d}) - C_{u_2, u_1 \oplus u_2}(\mathbf{d})\Big). \end{aligned}$$

**Lemma 12.** *Let $k = 2t \geq 10$ and $u, v \in \mathcal{B}_k$ be the functions defined by Definition 3. For any $\mathbf{c}, \mathbf{d} \in \mathbb{F}_2^t$, we have*

$$W_u(\mathbf{c}, \mathbf{d}) = \begin{cases} 2^{k-1} + 2^{t-1}, & \text{if } \mathbf{c} = \mathbf{d} = \mathbf{0} \\ 2^{t-1}, & \text{if } \mathbf{c} \neq \mathbf{0}, \mathbf{d} = \mathbf{0} \\ 2^{t-1}((-1)^{\mathbf{c}\cdot\mathbf{d}} + (-1)^{\mathbf{c}\cdot\mathbf{d}'} - (-1)^{\mathbf{c}\cdot\mathbf{d}''}), & \text{if } \textit{otherwise} \end{cases},$$

*and*

$$W_v(\mathbf{c}, \mathbf{d}) = \begin{cases} 2^{k-1} - 2^{t-1}, & \text{if } \mathbf{c} = \mathbf{d} = \mathbf{0} \\ -2^{t-1}, & \text{if } \mathbf{c} \neq \mathbf{0}, \mathbf{d} = \mathbf{0} \\ W_g(\mathbf{d}) - 2^{t-1}((-1)^{\mathbf{c}\cdot\mathbf{d}} - (-1)^{\mathbf{c}\cdot\mathbf{d}'} - (-1)^{\mathbf{c}\cdot\mathbf{d}''}), & \text{if } \textit{otherwise} \end{cases},$$

*where $\mathbf{d}' = (d_1 \oplus d_t, d_1, \ldots, d_{t-1})$ and $\mathbf{d}'' = (\bigoplus_{i=2}^{t} d_i, \bigoplus_{i=1}^{t} d_i, d_1 \oplus \bigoplus_{i=3}^{t} d_i, \ldots, d_1 \oplus d_t)$.*

*Proof.* From [23, Theorem 20] and Lemma 9, we have for any $\mathbf{c}, \mathbf{d} \in \mathbb{F}_2^t$

$$\begin{aligned}
W_u(\mathbf{c},\mathbf{d}) &= 2^{k-1}\delta_{\mathbf{0}}(\mathbf{c},\mathbf{d}) + \frac{1}{2}\big(W_p(\mathbf{c},\mathbf{d}) + W_q(\mathbf{c},\mathbf{d}) - W_{p\oplus q}(\mathbf{c},\mathbf{d})\big) \\
&= 2^{k-1}\delta_{\mathbf{0}}(\mathbf{c},\mathbf{d}) + 2^{t-1}(-1)^{\mathbf{c}\cdot\mathbf{d}} + \begin{cases} 0, & \text{if } \mathbf{d}=\mathbf{0} \\ 2^{t-1}(-1)^{\mathbf{c}\cdot\mathbf{d}'} + \frac{W_g(\mathbf{d})}{2}, & \text{otherwise} \end{cases} \\
&\qquad - \begin{cases} 0, & \text{if } \mathbf{d}=\mathbf{0} \\ 2^{t-1}(-1)^{\mathbf{c}\cdot\mathbf{d}''} + \frac{W_g(\mathbf{d})}{2}, & \text{otherwise} \end{cases} \\
&= \begin{cases} 2^{k-1}+2^{t-1}, & \text{if } \mathbf{c}=\mathbf{d}=\mathbf{0} \\ 2^{t-1}, & \text{if } \mathbf{c}\neq\mathbf{0}, \mathbf{d}=\mathbf{0} \\ 2^{t-1}((-1)^{\mathbf{c}\cdot\mathbf{d}} + (-1)^{\mathbf{c}\cdot\mathbf{d}'} - (-1)^{\mathbf{c}\cdot\mathbf{d}''}), & \text{if otherwise} \end{cases}.
\end{aligned}$$

where $\mathbf{d}' = (d_1 \oplus d_t, d_1, \ldots, d_t)$ and $\mathbf{d}'' = (\bigoplus_{i=2}^t d_i, \bigoplus_{i=1}^t d_i, d_1 \oplus \bigoplus_{i=3}^t d_i, \ldots, d_1 \oplus d_t)$. Since $W_{p\oplus 1}(\mathbf{c},\mathbf{d}) = -W_p(\mathbf{c},\mathbf{d})$ for all $\mathbf{c},\mathbf{d} \in \mathbb{F}_2^t$, we have

$$\begin{aligned}
W_u(\mathbf{c},\mathbf{d}) &= 2^{k-1}\delta_{\mathbf{0}}(\mathbf{c},\mathbf{d}) + \frac{1}{2}\big(W_{p\oplus 1}(\mathbf{c},\mathbf{d}) + W_q(\mathbf{c},\mathbf{d}) - W_{p\oplus q\oplus 1}(\mathbf{c},\mathbf{d})\big) \\
&= 2^{k-1}\delta_{\mathbf{0}}(\mathbf{c},\mathbf{d}) + \frac{1}{2}\big(-W_p(\mathbf{c},\mathbf{d}) + W_q(\mathbf{c},\mathbf{d}) + W_{p\oplus q}(\mathbf{c},\mathbf{d})\big) \\
&= \begin{cases} 2^{k-1}-2^{t-1}, & \text{if } \mathbf{c}=\mathbf{d}=\mathbf{0} \\ -2^{t-1}, & \text{if } \mathbf{c}\neq\mathbf{0}, \mathbf{d}=\mathbf{0} \\ W_g(\mathbf{d}) - 2^{t-1}((-1)^{\mathbf{c}\cdot\mathbf{d}} - (-1)^{\mathbf{c}\cdot\mathbf{d}'} - (-1)^{\mathbf{c}\cdot\mathbf{d}''}), & \text{if otherwise} \end{cases}.
\end{aligned}$$

□

From the above results, it is clear that the maximum absolute Walsh spectrum values of the functions $u$ and $u$ in $k = 2t \geq 10$ variables are $2^{k-1} + 2^{t-1}$ and $2^{k-1} - 2^{t-1}$, respectively.

**Lemma 13.** *Let $k = 2t \geq 10$ and $u, v \in \mathcal{B}_k$ be the functions defined by Definition 3. For any $\mathbf{c}, \mathbf{d} \in \mathbb{F}_2^t$, we have*

$$\begin{aligned}
2^{k-2} - 2^t \leq & C_u(\mathbf{c},\mathbf{d}) \leq 2^{k-2} + 2^{t+2}, \text{ and} \\
2^{k-2} - 3\cdot 2^{t+1} \leq & C_v(\mathbf{c},\mathbf{d}) \leq 2^{k-2} + 3\cdot 2^{t+1}.
\end{aligned}$$

*Proof.* We first consider the autocorrelation spectrum of the Boolean function $u$. Let us define $q'(\mathbf{z},\mathbf{r}) = \bigoplus_{i=2}^{t} z_i r_{i-1} \oplus z_1 r_t \oplus z_2 r_t = \phi(\mathbf{z})\cdot\mathbf{r}$ which is a quadratic bent function, where $\phi(\mathbf{z}) = (z_2, z_3, \ldots, z_{t-1}, z_t, z_1 \oplus r_2)$ is a permutation over $\mathbb{F}_2^t$. Then we have $u(\mathbf{z},\mathbf{r}) = p(\mathbf{z},\mathbf{r})q(\mathbf{z},\mathbf{r}) = p(\mathbf{z},\mathbf{r})q'(\mathbf{z},\mathbf{r})$ since $p(\mathbf{z},\mathbf{r}) = 0$ with $\mathbf{z} = \mathbf{0}$. Obviously, we have $C_u(\mathbf{c},\mathbf{d}) = 2^k$ if $(\mathbf{c},\mathbf{d}) = (\mathbf{0},\mathbf{0})$. In what follows, we consider the values of $C_u(\mathbf{c},\mathbf{d})$ with $(\mathbf{c},\mathbf{d}) \in \mathbb{F}_2^k \setminus \{\mathbf{0},\mathbf{0}\}$ and our strategy is based on Lemma 4.6. First, we can easily see that $W_p(\mathbf{0},\mathbf{0}) = W_{q'}(\mathbf{0},\mathbf{0}) = W_{p\oplus q'}(\mathbf{0},\mathbf{0}) = 2^t$. In addition, for any $(\mathbf{c},\mathbf{d}) \in \mathbb{F}_2^k \setminus \{\mathbf{0},\mathbf{0}\}$, we have $C_p(\mathbf{c},\mathbf{d}) = C_{q'}(\mathbf{c},\mathbf{d}) = 0$ and $C_{p\oplus q'}(\mathbf{c},\mathbf{d}) = 0$ since $(p \oplus q')(\mathbf{z},\mathbf{r})$ is also a bent function due

to $\mathbf{z} \oplus \phi(\mathbf{z})$ is a permutation on $\mathbb{F}_2^t$ by Lemma 8. Furthermore, we have

$$\begin{aligned} C_{p,q'}(\mathbf{c},\mathbf{d}) &= \sum_{\mathbf{z}\in\mathbb{F}_2^t}\sum_{\mathbf{r}\in\mathbb{F}_2^t}(-1)^{\mathbf{z}\cdot\mathbf{r}\oplus(\phi(\mathbf{z}\oplus\mathbf{c})\cdot(\mathbf{r}\oplus\mathbf{d}))} = \sum_{\mathbf{z}\in\mathbb{F}_2^t}\sum_{\mathbf{r}\in\mathbb{F}_2^t}(-1)^{\mathbf{z}\cdot\mathbf{r}\oplus(\phi(\mathbf{z})\oplus\phi(\mathbf{c}))\cdot(\mathbf{r}\oplus\mathbf{d}))} \\ &= \sum_{\mathbf{z}\in\mathbb{F}_2^t}\sum_{\mathbf{r}\in\mathbb{F}_2^t}(-1)^{(\mathbf{z}\oplus\phi(\mathbf{z}))\cdot\mathbf{r}\oplus\phi(\mathbf{z})\cdot\mathbf{d}\oplus\phi(\mathbf{c})\cdot\mathbf{r}\oplus\phi(\mathbf{c})\cdot\mathbf{d}} \\ &= \sum_{\mathbf{z}\in\mathbb{F}_2^t}\sum_{\mathbf{r}\in\mathbb{F}_2^t}(-1)^{(\phi^{-1}(\mathbf{z})\oplus\mathbf{z})\cdot\mathbf{r}\oplus\mathbf{z}\cdot\mathbf{d}\oplus\phi(\mathbf{c})\cdot\mathbf{r}\oplus\phi(\mathbf{c})\cdot\mathbf{d}} \\ &= (-1)^{\phi(\mathbf{c})\cdot\mathbf{d}}W_{\phi'(\mathbf{z})\cdot\mathbf{r}}(\mathbf{d},\phi(\mathbf{c})) \in \{2^t,-2^t\}, \end{aligned}$$

where $\phi'(\mathbf{z}) = \phi^{-1}(\mathbf{z})\oplus\mathbf{z}$ is a permutation over $\mathbb{F}_2^t$. Similarly, we have $C_{p,p\oplus q'}(\mathbf{c},\mathbf{d})$ $\in \{2^t,-2^t\}$ and $C_{q',p\oplus q'}(\mathbf{c},\mathbf{d}) \in \{2^t,-2^t\}$. Therefore, for any $(\mathbf{c},\mathbf{d}) \in \mathbb{F}_2^k \setminus \{\mathbf{0},\mathbf{0}\}$ we have

$$2^{k-2} - 2^t \le C_{pq}(\mathbf{c},\mathbf{d}) = C_{pq'}(\mathbf{c},\mathbf{d}) \le 2^{k-2} + 2^{t+2}.$$

We now consider the autocorrelation spectrum of Boolean function $v$. Similar to the case of $u$. We can get that

$$2^{k-2} - 2^{t+1} \le C_{(p\oplus 1)q'}(\mathbf{c},\mathbf{d}) \le 2^{k-2} + 2^{t+1}.$$

Note that the support of $v = (p \oplus 1)q$ equals $supp((p \oplus 1)q') \cup supp(g)$ and $|supp(g)| = 2^{t-1}$. Then we have

$$2^{k-2} - 3\cdot 2^{t+1} \le C_{(p\oplus 1)q}(\mathbf{c},\mathbf{d}) \le 2^{k-2} + 3\cdot 2^{t+1}.$$

This completes the proof. □

### 4.2 Main construction of ultra-lightweight Boolean functions

In this section, we construct a class of balanced Boolean functions by modifying a simple MM bent function using two sub-functions defined in Definition 3.

**Construction 2** *Let $n = 2k = 4t$ be an even integer not less than 20. We construct an $n$-variable Boolean function $f$ over $\mathbb{F}_2^k \times \mathbb{F}_2^k$ as follows*

$$f(\mathbf{x},\mathbf{y}) = \begin{cases} u(\mathbf{y}), & \text{if } \mathbf{x} = \mathbf{1}, \mathbf{y} \in \mathbb{F}_2^k \setminus \{\mathbf{1}\} \\ v(\mathbf{x}), & \text{if } \mathbf{y} = \mathbf{x} \in \mathbb{F}_2^k \\ (\mathbf{x}\oplus\mathbf{y})\cdot\mathbf{y}, & \text{otherwise} \end{cases},$$

*where $u$ and $v$ are the two Boolean functions over $\mathbb{F}_2^k$ defined in Definition 3.*

Since the outputs of MM bent function $(\mathbf{x}\oplus\mathbf{y})\cdot\mathbf{y} = \mathbf{x}\cdot\mathbf{y}\oplus\bigoplus_{i=1}^k y_i$ is modified by a sub-function $u$ when $\mathbf{x} = \mathbf{1}$ and $\mathbf{y} \neq \mathbf{0}$, and a sub-function $v$ when $\mathbf{x} = \mathbf{y}$, where $\mathbf{x},\mathbf{y} \in \mathbb{F}_2^k$. The function $f$ can be written as

$$\begin{aligned} f(\mathbf{x},\mathbf{y}) = (x_1x_2\cdots x_k(y_1y_2\cdots y_k \oplus 1)\oplus 1)(\mathbf{x}\oplus\mathbf{y})\cdot\mathbf{y} \oplus x_1x_2\cdots x_k(y_1y_2\cdots y_k \\ \oplus 1)u(\mathbf{y}) \oplus (x_1\oplus y_1\oplus 1)(x_2\oplus y_2\oplus 1)\cdots(x_k\oplus y_k\oplus 1)v(\mathbf{x}) \end{aligned}$$

*Remark 2.* Here, the algebraic degree of balanced Boolean functions defined in Construction 2 depends on the sub-functions $u$ and $v$. Both the sub-functions are nonlinear, in particular, $\deg(u) = 4$ and $\deg(v) = t+2$. Thus, $f$ is a nonlinear function where the degree can be made substantially high. Further, in subsection 4.3, it is observed that $f$ can be implemented in polynomial circuit size.

**Balancedness, autocorrelation properties, and nonlinearity** We now derive the Walsh spectrum and autocorrelation values of balanced Boolean function $f$ in $2k$ variables defined in Construction 2. Then, we compute the nonlinearity and absolute indicator of $f$.

**Theorem 1.** *Let $n = 2k \geq 20$ and $f \in \mathcal{B}_n$ be a Boolean function generated by Construction 2. Then for any $(\mathbf{a}, \mathbf{b}) \in \mathbb{F}_2^k \times \mathbb{F}_2^k$ we have*

$$W_f(\mathbf{a},\mathbf{b}) = \begin{cases} W_u(\mathbf{b}) + W_v(\mathbf{b}) + 2^k, & \text{if } (\mathbf{a},\mathbf{b}) \in \{\mathbf{0}\} \times \mathbb{F}_2^{k*} \\ W_u(\mathbf{0})(-1)^{\mathbf{1}\cdot\mathbf{a}} + W_v(\mathbf{a}), & \text{if } (\mathbf{a},\mathbf{b}) \in \mathbb{F}_2^{k*} \times \{\mathbf{0}\} \\ W_u(\mathbf{b})(-1)^{\mathbf{1}\cdot\mathbf{a}} + W_v(\mathbf{a}\oplus\mathbf{b}) & \\ +2^k(-1)^{\mathbf{a}\cdot(\mathbf{1}\oplus\mathbf{b})}, & \text{if } (\mathbf{a},\mathbf{b}) \in \mathbb{F}_2^{k*} \times \mathbb{F}_2^{k*} \setminus U \\ W_u(\mathbf{b})(-1)^{\mathbf{1}\cdot\mathbf{b}} + W_v(\mathbf{0}) & \\ -2^k\delta_{\mathbf{0}}(\mathbf{b})(-1)^{\mathbf{1}\cdot\mathbf{b}}, & \text{if } (\mathbf{a},\mathbf{b}) \in U \end{cases},$$

*where $U = \{(\mathbf{c},\mathbf{c}) : \mathbf{c} \in \mathbb{F}_2^k\} \subseteq \mathbb{F}_2^k \times \mathbb{F}_2^k$ and $\delta_{\mathbf{0}}(\cdot)$ is the Dirac (or Kronecker) symbol which is defined by $\delta_{\mathbf{0}}(\mathbf{b}) = 1$ if $\mathbf{b} = \mathbf{0}$ and $\delta_{\mathbf{0}}(\mathbf{b}) = 0$ otherwise.*

*Proof.* We can easily get that $\sum_{\mathbf{x}\in\mathbb{F}_2^k}(-1)^{\mathbf{c}\cdot\mathbf{x}\oplus\mathbf{d}\cdot\mathbf{y}}$ equals 0 if $\mathbf{c} \in \mathbb{F}_2^{k*}$ and equals $2^k$ otherwise, where $\mathbf{d}$ and $\mathbf{y}$ are arbitrary vectors in $\mathbb{F}_2^k$. Then for any $(\mathbf{a},\mathbf{b}) \in \mathbb{F}_2^k \times \mathbb{F}_2^k$, according to the definition of the Walsh transform we have

$$\begin{aligned} W_f(\mathbf{a},\mathbf{b}) &= \sum_{(\mathbf{x},\mathbf{y})\in\mathbb{F}_2^k\times\mathbb{F}_2^k} (-1)^{f(\mathbf{x},\mathbf{y})\oplus\mathbf{a}\cdot\mathbf{x}\oplus\mathbf{b}\cdot\mathbf{y}} \\ &= \sum_{(\mathbf{x},\mathbf{y})\in\{\mathbf{1}\}\times\mathbb{F}_2^k\setminus\{\mathbf{1}\}} (-1)^{f(\mathbf{x},\mathbf{y})\oplus\mathbf{a}\cdot\mathbf{x}\oplus\mathbf{b}\cdot\mathbf{y}} + \sum_{(\mathbf{x},\mathbf{y})\in U} (-1)^{f(\mathbf{x},\mathbf{y})\oplus\mathbf{a}\cdot\mathbf{x}\oplus\mathbf{b}\cdot\mathbf{y}} \\ &\quad + \sum_{(\mathbf{x},\mathbf{y})\in\{(\mathbf{x},\mathbf{y}):\mathbf{x}\in\mathbb{F}_2^k\setminus\{\mathbf{1}\},\mathbf{y}\in\mathbb{F}_2^k,\mathbf{y}\neq\mathbf{x}\}} (-1)^{f(\mathbf{x},\mathbf{y})\oplus\mathbf{a}\cdot\mathbf{x}\oplus\mathbf{b}\cdot\mathbf{y}} \\ &= \sum_{\mathbf{y}\in\mathbb{F}_2^k\setminus\{\mathbf{1}\}} (-1)^{u(\mathbf{y})\oplus\mathbf{1}\cdot\mathbf{a}\oplus\mathbf{b}\cdot\mathbf{y}} + \sum_{\mathbf{x}\in\mathbb{F}_2^k} (-1)^{v(\mathbf{x})\oplus\mathbf{a}\cdot\mathbf{x}\oplus\mathbf{b}\cdot\mathbf{x}} \\ &\quad + \sum_{(\mathbf{x},\mathbf{y})\in\{(\mathbf{x},\mathbf{y}):\mathbf{x}\in\mathbb{F}_2^k\setminus\{\mathbf{1}\},\mathbf{y}\in\mathbb{F}_2^k,\mathbf{y}\neq\mathbf{x}\}} (-1)^{\mathbf{x}\cdot\mathbf{y}\oplus\mathbf{1}\cdot\mathbf{y}\oplus\mathbf{a}\cdot\mathbf{x}\oplus\mathbf{b}\cdot\mathbf{y}}. \end{aligned}$$

Note that

$$\begin{aligned} \sum_{\mathbf{y}\in\mathbb{F}_2^k\setminus\{\mathbf{1}\}} (-1)^{u(\mathbf{y})\oplus\mathbf{1}\cdot\mathbf{a}\oplus\mathbf{b}\cdot\mathbf{y}} &= W_u(\mathbf{b})(-1)^{\mathbf{1}\cdot\mathbf{a}} - (-1)^{u(\mathbf{1})\oplus\mathbf{1}\cdot(\mathbf{a}\oplus\mathbf{b})} \\ &= W_u(\mathbf{b})(-1)^{\mathbf{1}\cdot\mathbf{a}} - (-1)^{\mathbf{1}\cdot(\mathbf{a}\oplus\mathbf{b})} \text{ (since } u(\mathbf{1}) = 0). \end{aligned}$$

Since $\sum_{\mathbf{x}\in\mathbb{F}_2^k}(-1)^{v(\mathbf{x})\oplus\mathbf{a}\cdot\mathbf{x}\oplus\mathbf{b}\cdot\mathbf{x}} = W_v(\mathbf{a}\oplus\mathbf{b})$, and

$$\begin{aligned}
&\sum_{(\mathbf{x},\mathbf{y})\in\{(\mathbf{x},\mathbf{y}):\mathbf{x}\in\mathbb{F}_2^k\setminus\{\mathbf{1}\},\mathbf{y}\in\mathbb{F}_2^k,\mathbf{y}\neq\mathbf{x}\}} (-1)^{\mathbf{x}\cdot\mathbf{y}\oplus\mathbf{1}\cdot\mathbf{y}\oplus\mathbf{a}\cdot\mathbf{x}\oplus\mathbf{b}\cdot\mathbf{y}} \\
&= \sum_{(\mathbf{x},\mathbf{y})\in\mathbb{F}_2^k\times\mathbb{F}_2^k\setminus U} (-1)^{\mathbf{x}\cdot\mathbf{y}\oplus\mathbf{1}\cdot\mathbf{y}\oplus\mathbf{a}\cdot\mathbf{x}\oplus\mathbf{b}\cdot\mathbf{y}} - \sum_{(x,y)\in\{\mathbf{1}\}\times\mathbb{F}_2^k\setminus\{\mathbf{1}\}} (-1)^{\mathbf{x}\cdot\mathbf{y}\oplus\mathbf{1}\cdot\mathbf{y}\oplus\mathbf{a}\cdot\mathbf{x}\oplus\mathbf{b}\cdot\mathbf{y}} \\
&= \left[\sum_{(\mathbf{x},\mathbf{y})\in\mathbb{F}_2^k\times\mathbb{F}_2^k} (-1)^{\mathbf{x}\cdot\mathbf{y}\oplus\mathbf{1}\cdot\mathbf{y}\oplus\mathbf{a}\cdot\mathbf{x}\oplus\mathbf{b}\cdot\mathbf{y}} - \sum_{x\in\mathbb{F}_2^k}(-1)^{\mathbf{x}\cdot\mathbf{x}\oplus\mathbf{1}\cdot\mathbf{x}\oplus\mathbf{a}\cdot\mathbf{x}\oplus\mathbf{b}\cdot\mathbf{x}}\right] \\
&\quad - \left[\sum_{y\in\mathbb{F}_2^k}(-1)^{\mathbf{1}_k\cdot\mathbf{y}\oplus\mathbf{1}\cdot\mathbf{y}\oplus\mathbf{a}\cdot\mathbf{1}\oplus\mathbf{b}\cdot\mathbf{y}} - (-1)^{\mathbf{1}\cdot(\mathbf{a}\oplus\mathbf{b})}\right] \\
&= \left[\sum_{x\in\mathbb{F}_2^k}(-1)^{\mathbf{a}\cdot\mathbf{x}}\sum_{y\in\mathbb{F}_2^k}(-1)^{(\mathbf{x}\oplus\mathbf{1}\oplus\mathbf{b})\cdot\mathbf{y}} - \sum_{x\in\mathbb{F}_2^k}(-1)^{(\mathbf{a}\oplus\mathbf{b})\cdot\mathbf{x}}\right] \\
&\quad - \left[(-1)^{\mathbf{1}\cdot\mathbf{a}}\sum_{y\in\mathbb{F}_2^k}(-1)^{\mathbf{b}\cdot\mathbf{y}} - (-1)^{\mathbf{1}\cdot(\mathbf{a}\oplus\mathbf{b})}\right] \\
&= 2^k(-1)^{\mathbf{a}\cdot(\mathbf{1}\oplus\mathbf{b})} - 2^k\delta_{\mathbf{0}}(\mathbf{a}\oplus\mathbf{b}) - 2^k\delta_{\mathbf{0}}(\mathbf{b})(-1)^{\mathbf{1}\cdot\mathbf{a}} + (-1)^{\mathbf{1}\cdot(\mathbf{a}\oplus\mathbf{b})}.
\end{aligned}$$

Then we have

$$\begin{aligned}
W_f(\mathbf{a},\mathbf{b}) &= W_u(\mathbf{b})(-1)^{\mathbf{1}\cdot\mathbf{a}} + W_v(\mathbf{a}\oplus\mathbf{b}) + 2^k(-1)^{\mathbf{a}\cdot(\mathbf{1}\oplus\mathbf{b})} - 2^k\delta_{\mathbf{0}}(\mathbf{a}\oplus\mathbf{b}) \\
&\quad -2^k\delta_{\mathbf{0}}(\mathbf{b})(-1)^{\mathbf{1}\cdot\mathbf{a}} \\
&= \begin{cases}
W_u(\mathbf{b}) + W_v(\mathbf{b}) + 2^k, & \text{if } (\mathbf{a},\mathbf{b}) \in \{\mathbf{0}\}\times\mathbb{F}_2^{k*} \\
W_u(\mathbf{0})(-1)^{\mathbf{1}\cdot\mathbf{a}} + W_v(\mathbf{a}), & \text{if } (\mathbf{a},\mathbf{b}) \in \mathbb{F}_2^{k*}\times\{\mathbf{0}\} \\
W_u(\mathbf{b})(-1)^{\mathbf{1}\cdot\mathbf{a}} + W_v(\mathbf{a}\oplus\mathbf{b}) & \\
+2^k(-1)^{\mathbf{a}\cdot(\mathbf{1}\oplus\mathbf{b})}, & \text{if } (\mathbf{a},\mathbf{b}) \in \mathbb{F}_2^{k*}\times\mathbb{F}_2^{k*}\setminus U \\
W_u(\mathbf{b})(-1)^{\mathbf{1}\cdot\mathbf{b}} + W_v(\mathbf{0}) & \\
-2^k\delta_{\mathbf{0}}(\mathbf{b})(-1)^{\mathbf{1}\cdot\mathbf{b}}, & \text{if } (\mathbf{a},\mathbf{b}) \in U
\end{cases}.
\end{aligned}$$

It follows from Theorem 1 that $W_f(\mathbf{0}) = W_u(\mathbf{0}) + W_v(\mathbf{0}) - 2^k = 0$, so we have the following results. We also get the nonlinearity of the balanced function $f$.

**Corollary 1.** *Let $n = 2k = 4t$ be an even integer not less than 20. The $n$-variable Boolean function $f \in \mathcal{B}_n$ generated by Construction 2 is balanced.*

**Corollary 2.** *Let $n = 2k = 4t$ be an even integer not less than 20 and $t$ be an even integer. The nonlinearity of $n$-variable Boolean function $f \in \mathcal{B}_n$ generated by Construction 2 is*

$$nl(f) = 2^{2k-1} - 2^{k-1} - 3\cdot 2^{t-1} - 2^{\frac{t}{2}}.$$

*Further, if $t$ is odd, the nonlinearity of $f$ is $2^{2k-1} - 2^{k-1} - 3\cdot 2^{t-1} - 2^{\frac{t-1}{2}} \leq nl(f) \leq 2^{2k-1} - 2^{k-1} - 3\cdot 2^{t-1}$.*

*Proof.* Let $\mathbf{a} = \mathbf{0}$ and $\mathbf{b} \neq \mathbf{0} \in \mathbb{F}_2^k$. Suppose $\mathbf{b} = (\mathbf{c}, \mathbf{d}) \neq (\mathbf{0}, \mathbf{0}) \in \mathbb{F}_2^t \times \mathbb{F}_2^t$. If $\mathbf{d} = \mathbf{0}$, then $\mathbf{c} \neq \mathbf{0}$, and $W_u(\mathbf{c}, \mathbf{0}) + W_v(\mathbf{c}, \mathbf{0}) + 2^k = 2^k$. If $\mathbf{d} \neq \mathbf{0}$, then $W_u(\mathbf{c}, \mathbf{d}) + W_v(\mathbf{c}, \mathbf{d}) + 2^k = 2^k + W_g(\mathbf{d}) + 2^t(-1)^{\mathbf{c}\cdot\mathbf{d}'}$, where the value of $W_g(\mathbf{d})$ is given in Lemma 6. Thus,

$$\max_{\mathbf{c},\mathbf{d}\in\mathbb{F}_2^t} |W_f(\mathbf{0}, \mathbf{b})| = \begin{cases} 2^k + 2^t + 2^{\frac{t+1}{2}}, & \text{if } t \text{ is odd} \\ 2^k + 2^t + 2^{\frac{t}{2}+1}, & \text{if } t \text{ is even} \end{cases}.$$

Let $\mathbf{a} \neq \mathbf{0} \in \mathbb{F}_2^k$ and $\mathbf{b} = \mathbf{0}$. Suppose $\mathbf{a} = (\mathbf{c}, \mathbf{d}) \neq (\mathbf{0}, \mathbf{0}) \in \mathbb{F}_2^t \times \mathbb{F}_2^t$. Then $W_f(\mathbf{a}, \mathbf{0}) = W_u(\mathbf{0}, \mathbf{0})(-1)^{wt(\mathbf{c},\mathbf{d})} + W_v(\mathbf{c}, \mathbf{d})$, and so

$$\max_{\mathbf{c},\mathbf{d}\in\mathbb{F}_2^t} |W_f(\mathbf{a}, \mathbf{0})| = \begin{cases} 2^{k-1} + 2^{t+1} + 2^{\frac{t+1}{2}}, & \text{if } t \text{ is odd} \\ 2^{k-1} + 2^{t+1} + 2^{\frac{t}{2}+1}, & \text{if } t \text{ is even} \end{cases}.$$

Let $\mathbf{a} = \mathbf{b} = (\mathbf{c}, \mathbf{d}) \in \mathbb{F}_2^k$, where $\mathbf{c}, \mathbf{d} \in \mathbb{F}_2^t$. Then $W_f(\mathbf{a}, \mathbf{a}) = W_u(\mathbf{c}, \mathbf{d})(-1)^{wt(\mathbf{c},\mathbf{d})}$ $+W_v(\mathbf{0}, \mathbf{0}) - 2^k(-1)^{wt(\mathbf{c},\mathbf{d})}\delta_{(\mathbf{0},\mathbf{0})}(\mathbf{c}, \mathbf{d})$, and so $\max_{\mathbf{a}\in\mathbb{F}_2^k} |W_f(\mathbf{a}, \mathbf{a})| = 2^{k-1} + 2^t$. Let $\mathbf{a} \neq \mathbf{0}, \mathbf{b} \neq \mathbf{0} \in \mathbb{F}_2^k$ and $\mathbf{a} = \mathbf{b}$. Suppose $\mathbf{a} = (\mathbf{c}, \mathbf{d})$ and $\mathbf{b} = (\mathbf{e}, \mathbf{h}) \in \mathbb{F}_2^t \times \mathbb{F}_2^t$. Since $(\mathbf{d} \oplus \mathbf{h})' = \mathbf{d}' \oplus \mathbf{h}'$ and $(\mathbf{d} \oplus \mathbf{h})'' = \mathbf{d}'' \oplus \mathbf{h}''$. If $t$ be an even integer, then there exist values of $\mathbf{a}, \mathbf{b}$ such that $wt(\mathbf{a})$ is even, $\mathbf{a} \cdot \mathbf{b} = 0$, $\mathbf{c} \cdot \mathbf{d} = 0$, $\mathbf{c} \cdot \mathbf{d}' = 0$, $\mathbf{c} \cdot \mathbf{d}'' = 1$, $(\mathbf{c} \oplus \mathbf{e}) \cdot (\mathbf{d} \oplus \mathbf{h}) = 1$, $(\mathbf{c} \oplus \mathbf{e}) \cdot (\mathbf{d}' \oplus \mathbf{h}') = 0$, $(\mathbf{c} \oplus \mathbf{e}) \cdot (\mathbf{d}'' \oplus \mathbf{h}'') = 0$ and $(d_{t-1} \oplus h_{t-1}, d_t \oplus h_t) = (1, 1)$. Then $\max_{\mathbf{a},\mathbf{b}\in\mathbb{F}_2^k} |W_f(\mathbf{a}, \mathbf{b})| = 2^k + 3 \cdot 2^t + 2^{\frac{t}{2}+1}$. Thus, the nonlinearity of $f$ is $nl(f) = 2^{2k-1} - 2^{k-1} - 3 \cdot 2^{t-1} - 2^{\frac{t}{2}}$.

Let $t$ be an odd integer. Then the maximum absolute value of $W_f(\mathbf{a}, \mathbf{b})$ satisfies the inequality

$$2^k + 3 \cdot 2^t \leq \max_{\mathbf{a}\neq\mathbf{b}\in\mathbb{F}_2^k\setminus\{\mathbf{0}\}} |W_f(\mathbf{a}, \mathbf{b})| \leq 2^k + 3 \cdot 2^t + 2^{\frac{t+1}{2}}.$$

Thus, the nonlinearity of $f$ is $2^{2k-1} - 2^{k-1} - 3 \cdot 2^{t-1} - 2^{\frac{t-1}{2}} \leq nl(f) \leq 2^{2k-1} - 2^{k-1} - 3 \cdot 2^{t-1}$. □

**Theorem 2.** *Let $n = 2k = 4t \geq 8$ and $f \in \mathcal{B}_n$ be a Boolean function generated by Construction 2. Then for any $(\mathbf{a}, \mathbf{b}) \in \mathbb{F}_2^k \times \mathbb{F}_2^k$, we have*

$$C_f(\mathbf{a}, \mathbf{b}) = \begin{cases} 2^n, & \text{if } (\mathbf{a}, \mathbf{b}) = (\mathbf{0}, \mathbf{0}) \\ C_u(\mathbf{b}) + 2W_v(\mathbf{b})(-1)^{\mathbf{1}\cdot\mathbf{b}} & \\ -2(-1)^{u(\mathbf{1})\oplus u(\mathbf{1}\oplus\mathbf{b})} - 2(-1)^{v(\mathbf{1})} & \\ +2(-1)^{v(\mathbf{1})\oplus u(\mathbf{1}\oplus b)} - 2^k + 2, & \text{if } (\mathbf{a}, \mathbf{b}) \in \{\mathbf{0}\} \times \mathbb{F}_2^{k*} \\ C_v(\mathbf{a}) + 2W_u(\mathbf{a})(-1)^{\mathbf{1}\cdot\mathbf{a}} - 2^k, & \text{if } (\mathbf{a}, \mathbf{b}) \in U' \\ 2W_u(\mathbf{a})(-1)^{\mathbf{a}\cdot\mathbf{b}} + 2W_v(\mathbf{a} \oplus \mathbf{b})(-1)^{(\mathbf{a}\oplus\mathbf{1})\cdot\mathbf{b}} & \\ -2(-1)^{u(\mathbf{1}\oplus\mathbf{a}\oplus\mathbf{b})} + 2(-1)^{v(\mathbf{1}\oplus\mathbf{a})\oplus u(\mathbf{1}\oplus\mathbf{a}\oplus\mathbf{b})} & \\ -2(-1)^{v(\mathbf{1}\oplus\mathbf{a})} + 2, & \text{if } (\mathbf{a}, \mathbf{b}) \in \mathbb{F}_2^{k*} \times \mathbb{F}_2^k \setminus U' \end{cases},$$

*where $U' = \{(\mathbf{c}, \mathbf{c}) : \mathbf{c} \in \mathbb{F}_2^{k*}\} \subset \mathbb{F}_2^k \times \mathbb{F}_2^k$.*

*Proof.* By the definition of autocorrelation function, we immediately get that $C_f(\mathbf{0}, \mathbf{0}) = 2^n$. We now consider the values of $C_f(\mathbf{a}, \mathbf{b})$ for all $(\mathbf{a}, \mathbf{b}) \in \mathbb{F}_2^k \times \mathbb{F}_2^k \setminus$

$\{(\mathbf{0},\mathbf{0})\}$. Our discuss is mainly based on the fact that $\sum_{\mathbf{x}\in\mathbb{F}_2^k}(-1)^{\mathbf{c}\cdot\mathbf{x}\oplus\mathbf{d}\cdot\mathbf{y}}$ equals 0 if $\mathbf{c}\in\mathbb{F}_2^{k*}$ and equals $2^k$ otherwise, where $\mathbf{d}$ and $\mathbf{y}$ are arbitrary vectors in $\mathbb{F}_2^k$. We consider the values of $C_f(\mathbf{a},\mathbf{b})$ for all $(\mathbf{a},\mathbf{b})\in\mathbb{F}_2^k\times\mathbb{F}_2^k\setminus\{(\mathbf{0},\mathbf{0})\}$ from the following three cases:

Case (i): Let $(a,b)\in\{\mathbf{0}\}\times\mathbb{F}_2^{k*}$. In this case, we have

$$\begin{aligned}
C_f(a,b) &= \sum_{(\mathbf{x},\mathbf{y})\in\{\mathbf{1}\}\times\mathbb{F}_2^k}(-1)^{f(\mathbf{1},\mathbf{y})\oplus f(\mathbf{1},\mathbf{y}\oplus\mathbf{b})}+\sum_{(\mathbf{x},\mathbf{y})\in T\times\mathbb{F}_2^k}(-1)^{f(\mathbf{x},\mathbf{y})\oplus f(\mathbf{x},\mathbf{y}\oplus\mathbf{b})}\\
&= \sum_{\mathbf{y}\in\mathbb{F}_2^k\setminus\{\mathbf{1},\mathbf{1}\oplus\mathbf{b}\}}(-1)^{u(\mathbf{y})\oplus u(\mathbf{y}\oplus b)}+(-1)^{v(\mathbf{1})\oplus u(\mathbf{1}\oplus b)}+(-1)^{u(\mathbf{1}\oplus\mathbf{b})\oplus v(\mathbf{1})}\\
&\quad+\sum_{\mathbf{x}\in T}\Bigg(\sum_{\mathbf{y}\in\mathbb{F}_2^k\setminus\{\mathbf{x},\mathbf{x}\oplus\mathbf{b}\}}(-1)^{(\mathbf{x}\oplus\mathbf{y})\cdot\mathbf{y}\oplus(\mathbf{x}\oplus\mathbf{y}\oplus\mathbf{b})\cdot(\mathbf{y}\oplus\mathbf{b})}+(-1)^{v(\mathbf{x})\oplus\mathbf{b}\cdot(\mathbf{x}\oplus\mathbf{b})}\\
&\qquad+(-1)^{\mathbf{b}\cdot(\mathbf{x}\oplus\mathbf{b})\oplus v(\mathbf{x})}\Bigg)\\
&= \sum_{\mathbf{y}\in\mathbb{F}_2^k\setminus\{\mathbf{1},\mathbf{1}\oplus\mathbf{b}\}}(-1)^{u(\mathbf{y})\oplus u(\mathbf{y}\oplus b)}+2(-1)^{v(\mathbf{1})\oplus u(\mathbf{1}\oplus b)}\\
&\quad+\sum_{\mathbf{x}\in T}\Big((2^k-2)(-1)^{\mathbf{b}\cdot\mathbf{x}\oplus\mathbf{1}\cdot\mathbf{b}}+2(-1)^{v(\mathbf{x})\oplus\mathbf{b}\cdot\mathbf{x}\oplus\mathbf{1}\cdot\mathbf{b}}\Big)\\
&= C_u(\mathbf{b})-2(-1)^{u(\mathbf{1})\oplus u(\mathbf{1}\oplus\mathbf{b})}+2(-1)^{v(\mathbf{1})\oplus u(\mathbf{1}\oplus b)}\\
&\quad+(2^k-2)\sum_{\mathbf{x}\in\mathbb{F}_2^k}(-1)^{\mathbf{b}\cdot\mathbf{x}\oplus\mathbf{1}\cdot\mathbf{b}}-(2^k-2)+2W_v(\mathbf{b})(-1)^{\mathbf{1}_k\cdot\mathbf{b}}-2(-1)^{v(\mathbf{1})}\\
&= C_u(\mathbf{b})+2W_v(\mathbf{b})(-1)^{\mathbf{1}_k\cdot\mathbf{b}}-2(-1)^{u(\mathbf{1}_k)\oplus u(\mathbf{1}_k\oplus\mathbf{b})}+2(-1)^{v(\mathbf{1})\oplus u(\mathbf{1}\oplus b)}\\
&\quad-2(-1)^{v(\mathbf{1})}-2^k+2,
\end{aligned}$$

where $T=\mathbb{F}_2^k\setminus\{\mathbf{1}\}$. It can be easily verified that $v(\mathbf{1})=0$ if $t$ is odd and $v(\mathbf{1})=1$ if $t$ is even. In addition, we can check that $u(\mathbf{1})=0$. Thus, in this case, we have

$$C_f(a,b)=C_u(\mathbf{b})+2W_v(\mathbf{b})(-1)^{\mathbf{1}\cdot\mathbf{b}}-2^k+2\big(1-(-1)^{t+1}\big)\big(1-(-1)^{u(\mathbf{1}\oplus\mathbf{b})}\big).$$

Case (ii): Let $(\mathbf{a},\mathbf{b})\in U'$. We can get that

$$\begin{aligned}
C_f(a,b) &= \sum_{\mathbf{x}=\mathbf{1},\mathbf{y}\in\mathbb{F}_2^k}(-1)^{f(\mathbf{1},\mathbf{y})\oplus f(\mathbf{1}\oplus\mathbf{a},\mathbf{y}\oplus\mathbf{a})}+\sum_{\mathbf{x}=\mathbf{1}\oplus\mathbf{a},\mathbf{y}\in\mathbb{F}_2^k}(-1)^{f(\mathbf{1}\oplus\mathbf{a},\mathbf{y})\oplus f(\mathbf{1},\mathbf{y}\oplus\mathbf{a})}\\
&\quad+\sum_{(\mathbf{x},\mathbf{y})\in E_{\mathbf{a}}\times\mathbb{F}_2^k}(-1)^{f(\mathbf{x},\mathbf{y})\oplus f(\mathbf{x}\oplus\mathbf{1},\mathbf{y}\oplus\mathbf{a})}\\
&= \sum_{\mathbf{y}\in\mathbb{F}_2^k\setminus\{\mathbf{1},\mathbf{1}\oplus\mathbf{a}\}}(-1)^{u(\mathbf{y})\oplus(\mathbf{1}\oplus\mathbf{a}\oplus\mathbf{y}\oplus\mathbf{a})\cdot(\mathbf{y}\oplus\mathbf{a})}+(-1)^{v(\mathbf{1})\oplus v(\mathbf{1}\oplus\mathbf{a})}+(-1)^{u(\mathbf{1}\oplus\mathbf{a})\oplus\mathbf{1}\cdot\mathbf{a}}\\
&\quad+\sum_{\mathbf{y}\in\mathbb{F}_2^k\setminus\{\mathbf{1},\mathbf{1}\oplus\mathbf{a}\}}(-1)^{(\mathbf{1}\oplus\mathbf{a}\oplus\mathbf{y})\cdot\mathbf{y}\oplus u(\mathbf{y}\oplus\mathbf{a})}+(-1)^{v(\mathbf{1}\oplus\mathbf{a})\oplus v(\mathbf{1})}+(-1)^{\mathbf{1}\cdot\mathbf{a}\oplus u(\mathbf{1}\oplus\mathbf{a})}\\
&\quad+\sum_{\mathbf{x}\in E_a}\Bigg(\sum_{\mathbf{y}\in\mathbb{F}_2^k\setminus\{\mathbf{x}\}}(-1)^{(\mathbf{x}\oplus\mathbf{y})\cdot\mathbf{y}\oplus(\mathbf{x}\oplus\mathbf{a}\oplus\mathbf{y}\oplus\mathbf{a})\cdot(\mathbf{y}\oplus\mathbf{a})}+(-1)^{v(\mathbf{x})\oplus v(\mathbf{x}\oplus\mathbf{a})}\Bigg)
\end{aligned}$$

$$
\begin{aligned}
&= 2 \sum_{\mathbf{y}\in\mathbb{F}_2^k\setminus\{\mathbf{1},\mathbf{1}\oplus\mathbf{a}\}} (-1)^{u(\mathbf{y})\oplus\mathbf{a}\cdot\mathbf{y}\oplus\mathbf{1}\cdot\mathbf{a}} + (-1)^{v(\mathbf{1})\oplus v(\mathbf{1}\oplus\mathbf{a})} + (-1)^{u(\mathbf{1}\oplus\mathbf{a})\oplus\mathbf{1}\cdot\mathbf{a}} \\
&\quad + \sum_{\mathbf{x}\in E_a} \left( \sum_{\mathbf{y}\in\mathbb{F}_2^k} (-1)^{\mathbf{a}\cdot\mathbf{y}\oplus\mathbf{a}\cdot\mathbf{x}} - (-1)^{\mathbf{a}\cdot\mathbf{x}\oplus\mathbf{a}\cdot\mathbf{x}} + (-1)^{v(\mathbf{x})\oplus v(\mathbf{x}\oplus\mathbf{a})} \right) \\
&= \left[ 2W_u(\mathbf{a})(-1)^{\mathbf{1}\cdot\mathbf{a}} - 2(-1)^{u(\mathbf{1})} + 2(-1)^{v(\mathbf{1})\oplus v(\mathbf{1}\oplus\mathbf{a})} \right] \\
&\quad + \left[ C_v(\mathbf{a}) - 2(-1)^{v(\mathbf{1})\oplus v(\mathbf{1}\oplus\mathbf{a})} - (2^k - 2) \right] \\
&= C_v(\mathbf{a}) + 2W_u(\mathbf{a})(-1)^{\mathbf{1}\cdot\mathbf{a}} - 2^k,
\end{aligned}
$$

where $E_{\mathbf{a}} = \mathbb{F}_2^k \setminus \{\mathbf{1}, \mathbf{1}\oplus\mathbf{a}\}$.
Case (iii): Let $(\mathbf{a},\mathbf{b}) \in \mathbb{F}_2^{k*} \times \mathbb{F}_2^k \setminus U'$. In this case, we can obtain that

$$
\begin{aligned}
C_f(\mathbf{a},\mathbf{b}) &= \sum_{\mathbf{x}=\mathbf{1},\mathbf{y}\mathbb{F}_2^k} (-1)^{f(\mathbf{1},\mathbf{y})\oplus f(\mathbf{1}\oplus\mathbf{a},\mathbf{y}\oplus\mathbf{b})} + \sum_{\mathbf{x}=\mathbf{1}\oplus\mathbf{a},\mathbf{y}\in\mathbb{F}_2^k} (-1)^{f(\mathbf{1}\oplus\mathbf{a},\mathbf{y})\oplus f(\mathbf{1},\mathbf{y}\oplus\mathbf{b})} \\
&\quad + \sum_{(\mathbf{x},\mathbf{y})\in E_{\mathbf{a}}\times\mathbb{F}_2^k} (-1)^{f(\mathbf{x},\mathbf{y})\oplus f(\mathbf{x}\oplus\mathbf{1},\mathbf{y}\oplus\mathbf{b})} \\
&= \sum_{\mathbf{y}\in\mathbb{F}_2^k\setminus\{\mathbf{1},\mathbf{1}\oplus\mathbf{a}\oplus\mathbf{b}\}} \Big( (-1)^{u(\mathbf{y})\oplus(\mathbf{1}\oplus\mathbf{a}\oplus\mathbf{y}\oplus\mathbf{b})\cdot(\mathbf{y}\oplus\mathbf{b})} + (-1)^{v(\mathbf{1})\oplus(\mathbf{a}\oplus\mathbf{b})\cdot(\mathbf{1}\oplus\mathbf{b})} \\
&\quad + (-1)^{u(\mathbf{1}\oplus\mathbf{a}\oplus\mathbf{b})\oplus v(\mathbf{1}\oplus\mathbf{a})} \Big) + \sum_{\mathbf{y}\in\mathbb{F}_2^k\setminus\{\mathbf{1}\oplus\mathbf{a},\mathbf{1}\oplus\mathbf{b}\}} \Big( (-1)^{(\mathbf{1}\oplus\mathbf{a}\oplus\mathbf{y})\cdot\mathbf{y}\oplus u(\mathbf{y}\oplus\mathbf{b})} \\
&\quad + (-1)^{v(\mathbf{1}\oplus\mathbf{a})\oplus u(\mathbf{1}\oplus\mathbf{a}\oplus\mathbf{b})} + (-1)^{(\mathbf{a}\oplus\mathbf{b})\cdot(\mathbf{1}\oplus\mathbf{b})\oplus v(\mathbf{1})} \Big) \\
&\quad + \sum_{\mathbf{x}\in E_{\mathbf{a}}} \Big( \sum_{\mathbf{y}\in\mathbb{F}_2^k\setminus\{\mathbf{x},\mathbf{x}\oplus\mathbf{a}\oplus\mathbf{b}\}} (-1)^{(\mathbf{x}\oplus\mathbf{y})\cdot\mathbf{y}\oplus(\mathbf{x}\oplus\mathbf{a}\oplus\mathbf{y}\oplus\mathbf{b})\cdot(\mathbf{y}\oplus\mathbf{b})} \\
&\quad + (-1)^{v(\mathbf{x})\oplus(\mathbf{a}\oplus\mathbf{b})\cdot(\mathbf{x}\oplus\mathbf{b})} + (-1)^{(\mathbf{a}\oplus\mathbf{b})\cdot(\mathbf{x}\oplus\mathbf{a}\oplus\mathbf{b})\oplus v(\mathbf{x}\oplus\mathbf{a})} \Big) \\
&= 2 \sum_{\mathbf{y}\in\mathbb{F}_2^k\setminus\{\mathbf{1},\mathbf{1}\oplus\mathbf{a}\oplus\mathbf{b}\}} \Big( (-1)^{u(\mathbf{y})\oplus\mathbf{a}\cdot\mathbf{y}\oplus\mathbf{a}\cdot\mathbf{b}} + (-1)^{v(\mathbf{1})\oplus\mathbf{a}\cdot(\mathbf{1}\oplus\mathbf{b})} \\
&\quad + (-1)^{v(\mathbf{1}\oplus\mathbf{a})\oplus u(\mathbf{1}\oplus\mathbf{a}\oplus\mathbf{b})} \Big) + \sum_{\mathbf{x}\in E_{\mathbf{a}}} \Big( \sum_{\mathbf{y}\in\mathbb{F}_2^k\setminus\{\mathbf{x},\mathbf{x}\oplus\mathbf{a}\oplus\mathbf{b}\}} (-1)^{\mathbf{a}\cdot\mathbf{y}\oplus(\mathbf{a}\oplus\mathbf{1}\oplus\mathbf{x})\cdot\mathbf{b}} \\
&\quad + (-1)^{v(\mathbf{x})\oplus(\mathbf{a}\oplus\mathbf{b})\cdot\mathbf{x}\oplus\mathbf{a}\cdot\mathbf{b}\oplus\mathbf{1}\cdot\mathbf{b}} + (-1)^{v(\mathbf{x}\oplus\mathbf{a})\oplus(\mathbf{a}\oplus\mathbf{b})\cdot\mathbf{x}\oplus\mathbf{1}\cdot(\mathbf{a}\oplus\mathbf{b})} \Big) \\
&= 2 \sum_{\mathbf{y}\in\mathbb{F}_2^k\setminus\{\mathbf{1},\mathbf{1}\oplus\mathbf{a}\oplus\mathbf{b}\}} \Big( (-1)^{u(\mathbf{y})\oplus\mathbf{a}\cdot\mathbf{y}\oplus\mathbf{a}\cdot\mathbf{b}} + (-1)^{v(\mathbf{1})\oplus\mathbf{a}\cdot(\mathbf{1}\oplus\mathbf{b})} \\
&\quad + (-1)^{v(\mathbf{1}\oplus\mathbf{a})\oplus u(\mathbf{1}\oplus\mathbf{a}\oplus\mathbf{b})} \Big) + \sum_{\mathbf{x}\in E_{\mathbf{a}}} \Big( - (-1)^{(\mathbf{a}\oplus\mathbf{b})\cdot\mathbf{x}\oplus\mathbf{b}\cdot(\mathbf{1}\oplus\mathbf{a})}
\end{aligned}
$$

$$
\begin{aligned}
&-(-1)^{(\mathbf{a}\oplus\mathbf{b})\cdot\mathbf{x}\oplus\mathbf{1}\cdot(\mathbf{a}\oplus\mathbf{b})}+(-1)^{v(\mathbf{x})\oplus(\mathbf{a}\oplus\mathbf{b})\cdot\mathbf{x}\oplus\mathbf{a}\cdot\mathbf{b}\oplus\mathbf{1}\cdot\mathbf{b}}\\
&+(-1)^{v(\mathbf{x}\oplus\mathbf{a})\oplus(\mathbf{a}\oplus\mathbf{b})\cdot\mathbf{x}\oplus\mathbf{1}\cdot(\mathbf{a}\oplus\mathbf{b})}\Big)\\
=\ &\big(2W_u(\mathbf{a})(-1)^{\mathbf{a}\cdot\mathbf{b}}-2(-1)^{u(\mathbf{1})\oplus\mathbf{a}\cdot(\mathbf{1}\oplus\mathbf{b})}-2(-1)^{u(\mathbf{1}\oplus\mathbf{a}\oplus\mathbf{b})}\\
&+2(-1)^{v(\mathbf{1})\oplus\mathbf{a}\cdot(\mathbf{1}\oplus\mathbf{b})}+2(-1)^{v(\mathbf{1}\oplus\mathbf{a})\oplus u(\mathbf{1}\oplus\mathbf{a}\oplus\mathbf{b})}\big)\\
&+\big(2W_v(\mathbf{a}\oplus\mathbf{b})(-1)^{(\mathbf{a}\oplus\mathbf{1})\cdot\mathbf{b}}-2(-1)^{v(\mathbf{1})\oplus\mathbf{a}\cdot(\mathbf{b}\oplus\mathbf{1}_k)}\\
&-2(-1)^{v(\mathbf{1}\oplus\mathbf{a})}+2(-1)^{\mathbf{a}\cdot(\mathbf{1}\oplus\mathbf{b})}+2\big)\\
=\ &2W_u(\mathbf{a})(-1)^{\mathbf{a}\cdot\mathbf{b}}+2W_v(\mathbf{a}\oplus\mathbf{b})(-1)^{(\mathbf{a}\oplus\mathbf{1})\cdot\mathbf{b}}-2(-1)^{u(\mathbf{1}\oplus\mathbf{a}\oplus\mathbf{b})}\\
&+2(-1)^{v(\mathbf{1}\oplus\mathbf{a})\oplus u(\mathbf{1}\oplus\mathbf{a}\oplus\mathbf{b})}-2(-1)^{v(\mathbf{1}\oplus\mathbf{a})}+2,
\end{aligned}
$$

where $E_{\mathbf{a}}=\mathbb{F}_2^k\setminus\{\mathbf{1},\mathbf{1}\oplus\mathbf{a}\}$. □

From the above results, we can derive the maximum absolute autocorrelation value of the constructed balanced function in $2k$ variables, which is strictly less than $2^k$.

**Corollary 3.** *Let $n=2k=4t\geq 20$ and $f\in\mathcal{B}_n$ be a Boolean function generated by Construction 2. Then the absolute indicator of $f\in\mathcal{B}_n$ is $\Delta_f\leq 2^k-2^{k-2}+9\cdot 2^t$.*

*Proof.* If $\mathbf{a}=\mathbf{b}=\mathbf{0}\in\mathbb{F}_2^k$, then $C_f(\mathbf{a},\mathbf{b})=2^n$. We derive the maximum absolute autocorrelation value of $f$ on different cases.

Case(i): Let us consider $\mathbf{a}=\mathbf{b}\neq\mathbf{0}\in\mathbb{F}_2^k$. Suppose $\mathbf{a}=(\mathbf{c},\mathbf{d})\in\mathbb{F}_2^t\times\mathbb{F}_2^t\setminus\{(\mathbf{0},\mathbf{0})\}$. Then

$$
\begin{aligned}
C_f(\mathbf{a},\mathbf{a})&=C_v(\mathbf{c},\mathbf{d})+2W_u(\mathbf{c},\mathbf{d})(-1)^{wt(\mathbf{c},\mathbf{d})}-2^k\\
&=\begin{cases}C_v(\mathbf{c},\mathbf{0})+2W_u(\mathbf{c},\mathbf{0})(-1)^{wt(\mathbf{c})}-2^k, & \text{if } \mathbf{c}\neq\mathbf{0},\mathbf{d}=\mathbf{0}\\ C_v(\mathbf{c},\mathbf{d})+2W_u(\mathbf{c},\mathbf{d})(-1)^{wt(\mathbf{c},\mathbf{d})}-2^k, & \text{if } \mathbf{d}\neq\mathbf{0}\end{cases}.
\end{aligned}
$$

Since $-2^t\leq 2W_u(\mathbf{c},\mathbf{0})(-1)^{wt(\mathbf{c})}\leq 2^t$ for $\mathbf{c}\neq\mathbf{0}$ and $-3\cdot 2^t\leq 2W_u(\mathbf{c},\mathbf{d})(-1)^{wt(\mathbf{c},\mathbf{d})}\leq 3\cdot 2^t$ for $\mathbf{d}\neq\mathbf{0}$. From Lemma 13, we have $\max_{\mathbf{a}\in\mathbb{F}_2^k\setminus\{\mathbf{0}\}}|C_f(\mathbf{a},\mathbf{a})|\leq 2^k-2^{k-2}+9\cdot 2^t$.

Case(ii): Let $\mathbf{a}=\mathbf{0}$ and $\mathbf{b}\neq\mathbf{0}\in\mathbb{F}_2^k$. Suppose $\mathbf{b}=(\mathbf{e},\mathbf{h})\in\mathbb{F}_2^t\times\mathbb{F}_2^t\setminus\{(\mathbf{0},\mathbf{0})\}$. Then

$$
\begin{aligned}
C_f(\mathbf{0},\mathbf{b})&=C_u(\mathbf{b})+2W_v(\mathbf{b})(-1)^{wt(\mathbf{b})}-2^k+2\\
&\quad-2(-1)^{u(\mathbf{1})\oplus u(\mathbf{1}\oplus\mathbf{b})}+2(-1)^{v(\mathbf{1})\oplus u(\mathbf{1}\oplus\mathbf{b})}-2(-1)^{v(\mathbf{1})}\\
&=A(\mathbf{b})+B(\mathbf{b}),
\end{aligned}
$$

where $A(\mathbf{b}) = C_u(\mathbf{b}) + 2W_v(\mathbf{b})(-1)^{wt(\mathbf{b})} - 2^k$ and $B(\mathbf{b}) = 2 - 2(-1)^{u(\mathbf{1})\oplus u(\mathbf{1}\oplus\mathbf{b})}$ $+2(-1)^{v(\mathbf{1})\oplus u(\mathbf{1}\oplus\mathbf{b})} - 2(-1)^{v(\mathbf{1})}$ for all $\mathbf{b} \in \mathbb{F}_2^k$. Since $\max_{\mathbf{b}\in\mathbb{F}_2^{k*}} |B(\mathbf{b})| \leq 8$ and

$$\begin{aligned} A(\mathbf{b}) &= C_u(\mathbf{e},\mathbf{h}) + 2W_v(\mathbf{e},\mathbf{h})(-1)^{wt(\mathbf{e},\mathbf{h})} - 2^k \\ &= \begin{cases} C_u(\mathbf{e},\mathbf{0}) + 2W_v(\mathbf{e},\mathbf{0})(-1)^{wt(\mathbf{e})} - 2^k, & \text{if } \mathbf{e} \neq \mathbf{0}, \mathbf{h} = \mathbf{0} \\ C_u(\mathbf{e},\mathbf{h}) + 2W_v(\mathbf{e},\mathbf{h})(-1)^{wt(\mathbf{e},\mathbf{h})} - 2^k, & \text{if } \mathbf{h} \neq \mathbf{0} \end{cases} \\ &= \begin{cases} C_u(\mathbf{e},\mathbf{0}) - 2^t - 2^k, & \text{if } \mathbf{e} \neq \mathbf{0}, wt(\mathbf{e}) \text{ is even}, \mathbf{h} = \mathbf{0} \\ C_u(\mathbf{e},\mathbf{0}) + 2^t + 2^k, & \text{if } \mathbf{e} \neq \mathbf{0}, wt(\mathbf{e}) \text{ is odd}, \mathbf{h} = \mathbf{0} \\ C_u(\mathbf{e},\mathbf{h}) + 2W_v(\mathbf{e},\mathbf{h})(-1)^{wt(\mathbf{e},\mathbf{h})} - 2^k, & \text{if } \mathbf{h} \neq \mathbf{0} \end{cases}. \end{aligned}$$

Thus from Lemma 13, we have $\max|A(\mathbf{b})| \leq 2^k - 2^{k-2} + 2^{t+1}$ for $\mathbf{b} = (\mathbf{e},\mathbf{0})$ with $\mathbf{e} \neq \mathbf{0}$. If $\mathbf{h} \neq \mathbf{0}$, then we have

$$\max|A(\mathbf{b})| \leq \begin{cases} 2^k - 2^{k-2} + 4\cdot 2^t + 2^{\frac{t+3}{2}}, & \text{if } t \text{ is odd} \\ 2^k - 2^{k-2} + 4\cdot 2^t + 2^{\frac{t}{2}+2}, & \text{if } t \text{ is even} \end{cases},$$

and so, $\max_{\mathbf{b}\in\mathbb{F}_2^{k*}} |C_f(\mathbf{0},\mathbf{b})| \leq \begin{cases} 2^k - 2^{k-2} + 2^{t+2} + 2^{\frac{t+3}{2}} + 8, & \text{if } t \text{ is odd} \\ 2^k - 2^{k-2} + 2^{t+2} + 2^{\frac{t}{2}+2} + 8, & \text{if } t \text{ is even} \end{cases}$.

Case(iii): Let $\mathbf{a} \neq \mathbf{0}$, $\mathbf{b} \neq \mathbf{0}$ with $\mathbf{a} \neq \mathbf{b}$. Then we have

$$\max_{\mathbf{a},\mathbf{b}\in\mathbb{F}_2^{k*},\mathbf{a}\neq\mathbf{b}} |C_f(\mathbf{a},\mathbf{b})| \leq \begin{cases} 3\cdot 2^{t+1} + 2^{k-2} + 2^{\frac{t+3}{2}} + 8, & \text{if } t \text{ is odd} \\ 3\cdot 2^{t+1} + 2^{k-2} + 2^{\frac{t}{2}+2} + 8, & \text{if } t \text{ is even} \end{cases}.$$

Combining all three cases, we get the result. □

Recently, the balanced Boolean functions with good cryptographic properties are constructed in [11, 22, 23] by modifying MM bent functions. Here, we add one more class of balanced Boolean functions in a polynomial-size circuit having good cryptographic properties.

### 4.3 Efficient implementation of Boolean functions defined as in Construction 2

It is known that a $2k$-variable MM bent function can be written as a concatenation of $2^k$ affine functions in $k$ variables. Using a decoder, we can implement the balanced Boolean function $f$ in $2k = 4t \geq 20$ variables generated by Construction 2. However, in that case, the naive implementation will require $O(2^k)$ gates. Instead of using a decoder, one can use the circuit given in Fig. 5 considering an identity permutation over $\mathbb{F}_2^k$ and a nonzero $g$ function in $k$ variables. The number of logic gates that are required to implement a balanced function in $2k$ variables generated in Construction 2 is dependent on the $k$-variable Boolean functions $g$, $u$,$v$, $h_1$ and $h_2$. Here we consider two $2 \times 1$ multiplexers. Let us define $f_1(\mathbf{y}) = u(\mathbf{y})$ and $f_2(\mathbf{x}) = v(\mathbf{x})$, for all $\mathbf{x}, \mathbf{y} \in \mathbb{F}_2^k$, and

$h_1(\mathbf{x}) = x_k x_{k-1} \cdots x_1 \oplus 1$ and $h_2(\mathbf{x}) = (x_k \oplus y_k) \vee (x_{k-1} \oplus y_{k-1}) \vee \cdots \vee (x_1 \oplus y_1)$.

Let the output of the original bent function be $\varepsilon \in \mathbb{F}_2$. The output after applying the first multiplexer is $\varepsilon_1 = (1 \oplus h_1(\mathbf{x}))f_1(\mathbf{y}) \oplus h_1(\mathbf{x})\varepsilon \in \mathbb{F}_2$. Similarly, the output of the circuit, i.e., after applying the second multiplexer, is $(1 \oplus h_2(\mathbf{x}))f_2(\mathbf{x}) \oplus h_2(\mathbf{x})\varepsilon_1$.

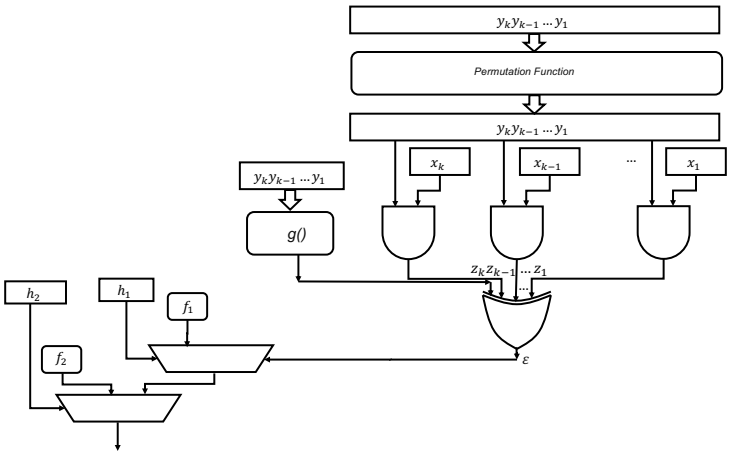


Fig. 6: Construction of balanced Boolean function proposed in [11, 22, 23]

*Remark 3.* The circuit's implementation of the balanced Boolean function generated from Construction 2 is given in Fig. 6. To implement two multiplexers, we need 8 logic gates (4 AND and 4 XOR gates). The required logic gates (fan-in of 2) to implement the functions $g$, $u$,$v$, $h_1$ and $h_2$ in $k = 2t \geq 10$ variables are as follows:

- $g$ requires $k - 1$ logic gates,
- $h_1$ requires $k$ logic gates,
- $h_2$ requires $2k$ logic gates,
- $f_1$ requires $2k$ logic gates,
- $f_2$ requires $3k + \frac{k}{2} + 1$ logic gates,

i.e., a total of $9.5k$.

Here, all these small functions can be implemented in polynomial circuit size over the input variables. Thus, the balanced Boolean function generated from Construction 2 can be implemented in polynomial circuit size.

## 5 Conclusion

In this work, we have first studied multiple polynomial-size constructions for MM bent functions. These constructions present various performance and design

space choices, thus significantly augmenting the arsenal of designers. Further, we consider the construction of cryptographically significant balanced functions with polynomial-size circuit size by modifying the MM bent functions. It is proved that the constructed function has very good nonlinearity and a very low absolute indicator value. In this regard, we would like to underline that the functions constructed by our method have low multiplicative complexity as evident from Construction 2 (Figure 6). Here we need AND gates to implement the functions $\mathbf{x} \cdot \mathbf{y}$, $h_1$, $f_1$, $f_2$ and multiplexers, where $\mathbf{x}, \mathbf{y} \in \mathbb{F}_2^k$. It is clear that $k+4$ AND gates are required to implement multiplexers and $\mathbf{x} \cdot \mathbf{y}$. As discussed above, we need $k-1$ AND gates for $h_1$, $k+1$ AND gates for $f_1$, and $3t+\iota+2$ AND gates for $f_2$, where $k = 2t$ and $\iota = \lfloor \frac{t-1}{2} \rfloor$. Thus, we need $3k+t+\iota+6$ AND gates to implement the function $f \in \mathcal{B}_{2k}$ given in Figure 6, where $k = 2t$ and $\iota = \lfloor \frac{t-1}{2} \rfloor$. This way, we can implement the function $f$ given in Construction 2 using polynomial size AND gates. Therefore, such functions can be used in the design of MPC/ZK/FHE-friendly symmetric-key primitives due to their low multiplicity complexity and additionally very good autocorrelation properties, very high nonlinearity, and good algebraic degree. Such functions will also have immediate applications in resisting Differential Fault Attacks in stream ciphers based on Feedback Shift Registers.

## References


1. J. Boyar and R. Peralta, *A small depth-16 circuit for the AES S-box*, In: D. Gritzalis, S. Furnell, M. Theoharidou (eds) Information Security and Privacy Research, SEC 2012, IFIPAICT 376 287–298 Springer.
2. A. Canteaut, S. Maitra, H. Yoshida, L. Perrin, S. Jha, R. Rohit A. Baksi, *Design of filtering functions for lightweight stream ciphers*. Presentation in ASK 2018, Kolkata, India.
3. C. Carlet, *Boolean functions for cryptography and error correcting codes*, In: Y. Crama, P. Hammer (eds.), Boolean Methods and Models, Cambridge Univ. Press, Cambridge 257–397 (2010).
4. T. W. Cusick and P. Stănică, *Cryptographic Boolean functions and applications*, Elsevier–Academic Press, (2009).
5. J. F. Dillon, *Elementary Hadamard Difference Sets*, PhD Thesis, University of Maryland (1974).
6. J. F. Dillon, *Elementary Hadamard Difference Sets*, In: proceedings of 6th S. E. Conference of Combinatorics, Graph Theory, and Computing, Utility Mathematics, Winnipeg, 237–249 (1975).
7. H. Dobbertin, *Construction of bent functions and balanced Boolean functions with high nonlinearity*, Fast Software Encryption 1994 LNCS 1008 61–74 (1994).
8. C. Gentry, *Fully homomorphic encryption using ideal lattices*, in *Proceedings of the forty-first annual ACM symposium on Theory of computing*, 2009, pp. 169–178.
9. S. Goldwasser, S. Micali, and C. Rackoff, *The knowledge complexity of interactive proof-systems (extended abstract)*, in *Proceedings of the 17th Annual ACM Symposium on Theory of Computing, May 6-8, 1985, Providence, Rhode Island, USA*, R. Sedgewick, Ed. ACM, 1985, pp. 291–304. `https://doi.org/10.1145/22145.22178`

10. K. C. Gupta and Palash Sarkar, *Efficient Representation and Software Implementation of Resilient Maiorana-McFarland S-boxes*, WISA 2004: 317-331, LNCS 3325.
11. S. Kavut, S. Maitra and D. Tang, *Construction and search of balanced Boolean functions on even number of variables towards excellent autocorrelation profile*, Designs, Codes and Cryptography 87(2-3) 261–276 (2019).
12. M. Khairallah, A. Chattopadhyay, B. Mandal and S. Maitra, *On Hardware Implementation of Tang-Maitra Boolean Functions*, Arithmetic of Finite Fields - 7th International Workshop, WAIFI 2018, LNCS 11321 111–127 (2018).
13. R. L. McFarland, *A family of difference sets in non-cyclic groups*, Journal of Combinatorial Theory, Series A 15(1) 1–10 (1973).
14. P. Méaux, A. Journault, F.-X. Standaert, and C. Carlet, *Towards stream ciphers for efficient fhe with low-noise ciphertexts*, in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2016, pp. 311–343.
15. S. Mesnager, *Bent Functions – Fundamentals and Results*, Springer, Switzerland ISBN 978-3-319-32593-4 1–544 (2016).
16. E. Ozcekic, S. Kavut and H. Kutucu, *Genetic Approach to Improve Cryptographic Properties of Balanced Boolean Functions Using Bent Functions*, Computers, 12(8), 159, 14 pages, 2023.
17. R. L. Rivest, L. Adleman and M. L. Dertouzos, *On data banks and privacy homomorphisms*, *Foundations of secure computation*, vol. 4, no. 11, pp. 169–180, 1978.
18. O. S. Rothaus, *On bent functions*, Journal of Combinatorial Theory, Series A 20 300–305 (1976).
19. D. Roy, B. N. Bathe and S. Maitra, *Differential Fault Attack on Kreyvium & FLIP*, IEEE Trans. Computers 70(12): 2161-2167 (2021).
20. P. Sarkar and S. Maitra, *Efficient Implementation of "Large" Stream Cipher Systems*, CHES 2001: 319-332, LNCS 2162.
21. P. Sarkar and S. Maitra, *Efficient Implementation of Cryptographically Useful 'Large' Boolean Functions*, IEEE Trans. Computers 52(4): 410-417 (2003).
22. D. Tang and S. Maitra, *Constructions of n-variable ($n \equiv 2 \bmod 4$) balanced Boolean functions with maximum absolute value in autocorrelation spectra $< 2^{\frac{n}{2}}$*, IEEE Transactions on Information Theory 64(1) 393–402 (2018).
23. D. Tang, S. Kavut, B. Mandal and S. Maitra, *Modifying Maiorana–McFarland type bent functions for good cryptographic properties and efficient implementation*, SIAM Journal on Discrete Mathematics 33(1) 238–256 (2019).
24. A. C. Yao, *Protocols for secure computations*, in *23rd annual symposium on foundations of computer science (FOCS 1982)*, IEEE, 1982, pp. 160–164.