MATHEMATICS OF COMPUTATION
Volume 70, Number 233, Pages 357–366
S 0025-5718(00)01196-0
Article electronically published on March 1, 2000

EXHAUSTIVE DETERMINATION OF (1023, 511, 255)-CYCLIC DIFFERENCE SETS

PETER GAAL AND SOLOMON W. GOLOMB

ABSTRACT. An exhaustive search for (1023, 511, 255)-cyclic difference sets has been conducted. A total of 10 non-equivalent (1023, 511, 255)-cyclic difference sets have been found, all of which are members of previously known or conjectured infinite families. A fast and effective autocorrelation test method was utilized that can also facilitate the testing of longer sequences.

1. INTRODUCTION

Cyclic Hadamard matrices, cyclic Hadamard difference sets, and the related two-level autocorrelation sequences—AC sequences for short—play a crucial role in both radar and communication systems. It is conjectured [1, pp. 91–92] that the length of each AC sequence falls into one of the following three categories:

- p, p prime, $p \equiv 3 (\mathrm{mod} 4)$
- pq, p, q prime, $q = p + 2$
- $2^n - 1$, n a positive integer.

(In a search involving lengths up to 9999 [2], no counterexamples to this conjecture were found.) Since out of these three categories, the third gives by far the most known or conjectured [3] difference sets of a given size, the exploration of these $(v, k, \lambda) = (2^n - 1, 2^{n-1} - 1, 2^{n-2} - 1)$-cyclic difference sets is well motivated. Complete searches in the following parameter set cases have been done already: (127, 63, 31) by Baumert and Fredericksen [4] in 1967; (255, 127, 63) by Cheng [5] in 1982; (511, 255, 127) by Dreier and Smith [6] in 1991 and later independently by Song [7] in 1997. The elapsed times between these successful attempts indicate the huge leaps in complexity between the consecutive cases. In fact, it was estimated [7] that even with applying all techniques used in previous searches, it would still take about 10^4 years of CPU time to do the complete search in the length 1023 case. In this article, we describe a new method that significantly reduces the computational time required to test a given sequence for the AC property, thus enabling us to undertake the exhaustive search for (1023, 511, 255)-cyclic difference sets.

A (v, k, λ)-cyclic difference set is defined in the following way: Let D be a set of k integers, distinct modulo v. D is a (v, k, λ)-cyclic difference set if and only if there are exactly λ ordered pairs (d_1, d_2), $d_1, d_2 \in D$, such that $d_1 - d_2 \equiv d \pmod{v}$ for all d, $0 < d < v$. We can use the following equivalent definition [8]:

Received by the editor April 21, 1998 and, in revised form, February 11, 1999.
2000 *Mathematics Subject Classification.* Primary 05B10, 94A55.
Key words and phrases. Cyclic Hadamard matrices, cyclic difference sets, ideal autocorrelation sequences.

Let

$$\theta(x) = \sum_{d_i \in D} x^{d_i}.$$

D is a cyclic difference set if and only if

$$\begin{aligned} \theta(x)\theta(x^{-1}) &\equiv k + \lambda \sum_{i=1}^{v-1} x^i \pmod{x^v - 1} \\ &= n + \lambda \sum_{i=0}^{v-1} x^i \pmod{x^v - 1}, \end{aligned} \tag{1}$$

where $n = k - \lambda$, and the coefficients are taken modulo v.

When v is not prime, each of its non-trivial divisors gives rise to a boundary condition that can be used in reducing the complexity of the search for all difference sets [4]. Let w be a proper divisor of v.

Then

$$\theta(x)\theta(x^{-1}) \equiv n + \frac{\lambda v}{w} \sum_{i=0}^{w-1} x^i \pmod{x^w - 1}. \tag{2}$$

A necessary, but of course not sufficient, condition for D to be a (v, k, λ) difference set [1, pp. 63–64] is the following.

Let

$$b_i^w = \#\{d \in D \mid d \equiv i \pmod{w}\}, \quad 0 \le i < w. \tag{3}$$

Then the b_i^w must satisfy all the following properties:

$$0 \le b_i^w \le v/w, \qquad \forall i, \quad 0 \le i < w, \tag{4}$$

$$\sum_{i=0}^{w-1} b_i^w = k, \tag{5}$$

$$\sum_{i=0}^{w-1} (b_i^w)^2 = n + \frac{\lambda v}{w}, \tag{6}$$

$$\sum_{i=0}^{w-1} b_i^w b_{i-j}^w = \frac{\lambda v}{w}, \qquad j \not\equiv 0 \pmod{w}. \tag{7}$$

To use this in a complete search, first we determine all possible $\{b_i^w\}$ sequences satisfying (4)–(7) for each $w|v$. The search can then be restricted to cases that by (3) correspond to one of the $\{b_i^w\}$ solutions for each $w|v$. Since $2^{10} - 1 = 1023$ is a product of three distinct primes, 3, 11 and 31, the most effective strategy is first to determine all $\{b_i^3\}$, $\{b_i^{11}\}$ and $\{b_i^{31}\}$ sequences; then using these, search for all $\{b_i^{33}\}$, $\{b_i^{93}\}$ and $\{b_i^{341}\}$ sequences; and then finally, using the latter three, determine all length 1023 AC sequences.

2. The autocorrelation test method

There are basically two ways by which we can reduce the computational complexity of the search. The first is to use necessary conditions, such as (5)–(7), to reduce the number of cases we actually have to test for the AC property. The second is to simplify the AC test itself. To calculate the complete autocorrelation function of a binary sequence of length L, we need approximately L^2 multiplications and

L^2 additions, $2L^2$ integer operations in all. The most obvious simplification can be achieved by stopping the calculation as soon as it becomes apparent that the sequence doesn't have the AC property. In all cases, we still need at least $L/2$ multiplications and $L/2$ additions, i.e. L integer operations before we can stop. The method proposed here will reduce the test to an average of five floating point—FP for short—operations per case by testing the Discrete Fourier Transform of the sequence instead of testing the sequence itself.

The DFT $\mathcal{F}\{a_i\}$ of a sequence $\{a_i\}$, $0 \leq i < L$, is defined as the vector $(A(0), \ldots, A(L-1))$, where

$$A(l) = \sum_{i=0}^{L-1} a_i e^{-\sqrt{-1} i l \frac{2\pi}{L}}.$$

Using well-known properties of the DFT, we can write

$$\mathcal{F}\left\{\sum_{i=0}^{L-1} a_i a_{i+\tau}\right\} = |A(l)|^2.$$

Since

$$\sum_{i=0}^{L-1} a_i a_{i+\tau} = \begin{cases} k, & \tau \equiv 0 \,(\mathrm{mod} v), \\ \lambda, & \text{otherwise}, \end{cases}$$

it follows that

$$|A(l)|^2 = \begin{cases} n + \lambda v, & l = 0, \\ n, & \text{otherwise} \end{cases}$$

Thus, a necessary condition for $\{a_i\}$ to be an AC sequence is that $|A(1)|^2 = n$ (note the "1", instead of "l", in the argument). Knowing $A(1)$, we only need two FP multiplications and one FP addition for this test. Intuitively, we can also expect this test to be efficient in the sense that non-AC sequences will fail it with high probability; thus, the further testing of passing sequences carries negligible cost. In other words, in almost all cases, it will be enough to test an arbitrary single DFT value—$A(1)$ in our case—to exclude non-AC candidate sequences. Of course, calculating $A(1)$ seemingly requires $2L$ additional FP operations, but due to the inherent redundancy of this calculation, we can reduce the resulting load significantly. To show how this can be done, consider a binary sequence $\{a_i\}$ of length $L = 2^n - 1$. By the multiplier theorem, we introduce no loss of generality if we assume that the sequence is constant on cyclotomic cosets, where by cyclotomic cosets we mean the equivalence classes under the equivalence relation $x \cong y$ iff $x \equiv 2^t y \pmod{v}$, for some $t \in \{0, 1, \cdots, n-1\}$. From now on, we will assume that sequence elements a_i and a_j are always equal if $i \equiv 2^t j \pmod{v}$, for some $t \in \{0, 1, \cdots, n-1\}$.

Let N_c be the number of cyclotomic cosets mod L. Define the DFT of a cyclotomic coset $\mathcal{C}_i$ by

$$C_i(l) = \sum_{j \in \mathcal{C}_i} e^{-\sqrt{-1} j l \frac{2\pi}{L}}.$$

Then the DFT of the sequence $\{a_i\}$ associated with a presumed difference set D is simply

$$A(l) = \sum_{i\,:\,\mathcal{C}_i \in D} C_i(l).$$

Since the values $C_i(1)$, $0 \leq i < N_c$, need to be calculated only once, getting the DFT value of $\{a_i\}$ now takes only $2N_c \approx 2L/n$ FP additions, where the factor 2 is a result of doing the computation in the complex field. To see how further reductions can be achieved, we have to consider the way a complete search is done. In the case of our previous example, we would use N_c nested loops, with each loop representing an assignment of 0 and then 1 to the coset $\mathcal{C}_i$ and invoking an inner loop assigning values to $\mathcal{C}_{i+1}$. We would test for the AC property in the innermost loop. Since we have N_c levels, the resulting search-tree will then consist of $2 \cdot (2^{N_c} - 1)$ branches and 2^{N_c} leaves. Because the DFT is a linear operator, we can build the Fourier transform gradually as we proceed along the branches of the tree. If a particular branch represents an assignment of 0 to a coset, then we do nothing. If it represents an assignment of 1, then we add the DFT value of that coset to the DFT value of the sequence. This takes two FP additions, one for the real and one for the imaginary part. Thus, on average, the cost of a branch is a single FP addition only. The cost of a leaf, as explained before, is two FP multiplications and one FP addition. Weighting these costs by the numbers of the branches and leaves, we can calculate that the total cost is approximately $3 \cdot 2^{N_c}$ additions and $2 \cdot 2^{N_c}$ multiplications, i.e. five FP operations per case.

The strength of this method is heightened by the fact that if we undertook searches for even longer sequences, the cost of testing for the AC property would not increase at all, as opposed to the linear growth associated with conventional autocorrelation test methods.

Further reductions are also possible. One method employed is the following. Let's precalculate the DFT of all combinations of cosets situated below a certain level in the tree, and select the combination with the maximal absolute DFT value for each possible Hamming weight value of the assignment. Then, upon entering this level during the search, the absolute value of the partial DFT sum can be tested against the preselected maximal absolute DFT value of the remaining cosets. Many subtrees can be excluded simply because the partial DFT sum is too big or too small to be brought to the desired $A(1)$ value even when adding or subtracting the biggest absolute DFT value of the remaining cosets.

As an example, assume that we chose the 6th level from the bottom in the tree to be the place where this test would be performed. Then we precalculate the DFT of the 6 cosets below with all $2^6 = 64$ possible coset assignments. We sort the coset assignments of these 6 cosets into 7 classes based on their Hamming weights. Thus, we will have 1 assignment with weight 0, 6 assignments with weight 10, 15 assignments with weight 20, etc. In each class, we determine which assignment has the maximal absolute value of the DFT. During the search, when we arrive at the 6th level, counted from the bottom, we already know the contribution of the $108 - 6 = 102$ higher cosets; call it $C_{0\ldots101}(1)$. We also know the required Hamming weight for the 6 lower level cosets, and from the precalculated list, we can determine the possible maximal absolute value of the DFT for these 6 cosets; call it $|C_{102\ldots107}(1)|_{max}$. Then if $|C_{0\ldots101}(1)| + |C_{102\ldots107}(1)|_{max}$ is less than the desired $|A(1)|$ or $|C_{0\ldots101}(1)| - |C_{102\ldots107}(1)|_{max}$ is more, then all possible completions of the partial coset assignment can be eliminated.

A corresponding condition could be set up based on the minimal absolute value of the DFT for the cosets at the bottom, but in practice this proved to be far less effective.

There is a trade-off in determining the optimal level of the tree at which the test ought to be performed. The higher the level, the more extensive the subtree that can be excluded when one of the conditions is met; however, the less likely this is to happen. Since the appropriate probabilities would be too difficult to calculate, the optimal level was found empirically.

It should be noted that every time DFT values were compared, windows for unavoidable calculation errors were left. In our programs, two double precision numbers were declared equal if the absolute value of the difference between them was less then 10^{-6}. The resulting "false alarms"—when two values were declared equal when indeed they were not—were easily filtered out by a conventional autocorrelation test performed in the integer domain. The rate of these events was sufficiently low so as not to create significant overhead.

3. The search

In order to organize the search efficiently for (1023, 511, 255)-cyclic difference sets, we have to consider how the cyclotomic cosets behave when we map the set of integers $\bmod v$ into the set of integers $\bmod w$, where $w|v$. First, to help us study the coset structure $\bmod v$, we introduce a simple classification of the cosets, explained as follows.

We will say that two integers x and y are equivalent $\bmod v$ when their greatest common divisors with v are the same, i.e.

$$x \cong y \text{ iff } (x, v) = (y, v).$$

Obviously, this is indeed an equivalence relation on the set of integers $\bmod v$; thus, it will provide us with equivalence classes tied to the individual divisors of v. It is also trivial that all the elements within any given coset are equivalent to each other; thus, the same equivalence relation can be applied to the cosets as well. This gives us a natural classification of the cyclotomic cosets. For convenience, we introduce the following notation. Let $\mathcal{C}_n^{(v)}$, $n|v$, mean the class of cosets satisfying

$$\mathcal{C}_i \in \mathcal{C}_n^{(v)} \text{ if and only if } (x, v) = n, \forall x \in \mathcal{C}_i.$$

Example 1. We take the case of $v = 33$ as an example. We have five cyclotomic cosets, listed as follows:

$$\begin{aligned}
\mathcal{C}_0 &= \{0\}, \\
\mathcal{C}_1 &= \{11, 22\}, \\
\mathcal{C}_2 &= \{3, 6, 12, 24, 15, 30, 27, 21, 9, 18\}, \\
\mathcal{C}_3 &= \{1, 2, 4, 8, 16, 32, 31, 29, 25, 17\}, \\
\mathcal{C}_4 &= \{5, 10, 20, 7, 14, 28, 23, 13, 26, 19\}.
\end{aligned}$$

Then, the classes are the following:

$$\begin{aligned}
\mathcal{C}_{33}^{(33)} &= \{\mathcal{C}_0\}, \\
\mathcal{C}_{11}^{(33)} &= \{\mathcal{C}_1\}, \\
\mathcal{C}_{3}^{(33)} &= \{\mathcal{C}_2\}, \\
\mathcal{C}_{1}^{(33)} &= \{\mathcal{C}_3, \mathcal{C}_4\}.
\end{aligned}$$

Thus, $\mathcal{C}_3$ and $\mathcal{C}_4$ are equivalent to each other, but the other cosets are equivalent only to themselves.

We now return to our original problem, namely determining the way the cosets are mapped when we map the set of integers mod v into the set of integers mod w, where $w|v$. It can be easily verified that cosets are mapped into cosets and equivalent cosets are mapped into equivalent cosets. Because of this, we could also say that coset equivalence classes are mapped into coset equivalence classes. This way, $\mathcal{C}_n^{(v)} \mapsto \mathcal{C}_m^{(w)}$, $w|v$, will mean that all cosets in equivalence class $\mathcal{C}_n^{(v)}$ are mapped into some cosets in equivalence class $\mathcal{C}_m^{(w)}$.

At this point, we introduce two characteristic values associated with this mapping. We say that class $\mathcal{C}_n^{(v)}$ has multiplicity $M \pmod w$ if there are exactly M cosets in $\mathcal{C}_n^{(v)}$ mapped into each of the cosets in $\mathcal{C}_m^{(w)}$. And we say that a class $\mathcal{C}_n^{(v)}$ has weight $W \pmod w$ if there are exactly W elements in any given coset in $\mathcal{C}_n^{(v)}$ mapped into each of the elements in the corresponding coset in $\mathcal{C}_m^{(w)}$. It can be easily seen that the multiplicity and the weight are properly determined by the following two identities:

$$M = \frac{\text{number of cosets in } \mathcal{C}_n^{(v)}}{\text{number of cosets in } \mathcal{C}_m^{(w)}},$$

$$W = \frac{\text{size of cosets in } \mathcal{C}_n^{(v)}}{\text{size of cosets in } \mathcal{C}_m^{(w)}}.$$

Using these notations, we have summarized all important information about the coset mappings in Table 1, at the end of the paper.

Next, we give two examples to show how the information contained in Table 1 can be utilized.

Example 2. We will determine the values assigned to the 1-, 2- and 5-element cosets mod 1023. First of all, we know that each coset has the value 0 or 1 assigned to it and that the sum of all elements is $511 \equiv 1 \pmod 5$. This forces the values assigned to the 1- and 2-element cosets to be 1 and 0, respectively. From Table 1, we get $\mathcal{C}_{1023}^{(1023)} \mapsto \mathcal{C}_{33}^{(33)}$ and $\mathcal{C}_{33}^{(1023)} \mapsto \mathcal{C}_{33}^{(33)}$, i.e. the 1- and 5-element cosets mod 1023 are mapped into the 1-element coset mod 33. Then, the value assigned to b_0^{33}, which is the same as the value assigned to the 1-element coset mod 33, must be congruent to 1 (mod 5). Using this, we can easily do a complete search for $\{b_i^{33}\}$ sequences that satisfy (4)–(7). The result of this search is that there is exactly one solution: $b_i^{33} = 31, 15, 15, \ldots, 15$. But this, in turn, means that 1 has to be assigned to every 5-element coset mod 1023, because otherwise the value assigned to the 1-element coset mod 33 would have to be less than 31.

Example 3. We will determine the $\{b_i^{31}\}$ sequences. Similarly to Example 2, using Table 1 and also the results of Example 2, we can determine that $b_0^{31} \equiv 1 (\text{mod } 10)$. Then, by (4), b_0^{31} =1, 11, 21 or 31 are the only possibilities. As for the other b_i^{31} values, we can claim that they must be odd because the contribution of the 5-element cosets mod 1023 is 1 and the contribution of all the other cosets is even since those cosets have weight 2 (mod 31). Using these necessary conditions, a simplified computer search sufficed to determine that there are exactly eight solutions mod 31.

As mentioned earlier, when w is a product of two or more primes, we can use the solutions for those prime moduli as boundary conditions. Consider $w = 341$, which has 11 and 31 as its prime factors, as an example. From Example 3, we

already know that there are eight solutions mod 31. It can also be seen that there is a unique $\{b_i^{11}\}$ solution: $b_i^{11} = 61, 45, 45, \ldots, 45$. This follows from the fact that the single $\{b_i^{33}\}$ solution found in Example 2 uniquely determines the $\{b_i^{11}\}$ solution through (3). Furthermore, we have from (4) that $0 \leq b_i^{341} \leq 3, \quad 0 \leq i < 341$. Then the problem of finding the $\{b_i^{341}\}$ sequences can be reformulated [6] as follows. We fill an 11×31 rectangle with numbers between 0 and 3 in such a way that the row sums form the unique $\{b_i^{11}\}$ sequence and the column sums form one of the eight possible $\{b_i^{31}\}$ sequences. Also, when we read out the numbers diagonally, extending the rectangle cyclically when necessary, we get a $\{b_i^{341}\}$ sequence that is constant on cosets and satisfies (5)–(7). The task of finding all solutions can then be done by an exhaustive computer search that takes all possible patterns giving permissible column and row sums and tests whether they satisfy (5)–(7) or not. Actually, using (5) is redundant here because the condition on either the row or the column sums already ensures that (5) will be satisfied. The compliance with (6) and (7) can be simultaneously checked using the AC test method described in Section 2.

To make the computation more efficient, we divided the search into eight distinct parts based on the eight $\{b_i^{31}\}$ solutions. This not only reduced the overhead by ensuring that we had unique boundary conditions in each case, but also supported a natural parallelization of the search process, which decreased the duration of the search significantly.

With these and similar methods, we were able to find all solutions of (4)–(7) for all possible moduli. Here, we only list the number of inequivalent solutions found in each case:

(mod 3)	:	1 solution
(mod 11)	:	1 solution
(mod 31)	:	8 solutions
(mod 33)	:	1 solution
(mod 93)	:	17 solutions
(mod 341)	:	14 solutions

When we do the final search for mod 1023 sequences, we follow a very similar path. The only difference is that here we have a 3-D object to work with. Namely, we have a $3 \times 11 \times 31$ rectangular solid to be filled with zeros and ones in such a way that the projection sums taken over the three faces form permissible $\{b_i^{33}\}$, $\{b_i^{93}\}$ and $\{b_i^{341}\}$ sequences. Note that using the $\{b_i^{3}\}$, $\{b_i^{11}\}$ or $\{b_i^{31}\}$ sequences would provide no further information, because the correct projection sums over the faces already ensure the correct sums along the edges. This implies, of course, that the combination of a certain $\{b_i^{93}\}$ and $\{b_i^{341}\}$ solution can only occur if they both reduce to the same $\{b_i^{31}\}$ sequence mod 31. The unique $\{b_i^{33}\}$ sequence can occur with any $\{b_i^{93}\}$ or $\{b_i^{341}\}$ solutions, because the $\{b_i^{3}\}$ and $\{b_i^{11}\}$ solutions are both unique. When we divide the search into parts based on these permissible combinations, we have to remember that, in general, the decimations of the modular solutions must be treated as inequivalent cases. We are still allowed, however, to fix the decimation of one of the modular sequences, and we can also use the fact that the unique $\{b_i^{33}\}$ solution is invariant under decimations. This led to a total of 43 permissible combinations.

Interestingly, it turned out that the computationally most intensive part wasn't the search for the length 1023 AC sequences but rather the intermediate search for the $\{b_i^{341}\}$ sequences, which involved the testing of approximately $5 \cdot 10^{12}$ putative

solutions. Utilizing the test method described in the previous section, however, we were able to accomplish this task in less than two days, with the bulk of the computation done on USC's Convex Exemplar computer. The other moduli took significantly less time to check completely.

4. Results

We give one representative for each of the ten inequivalent (1023, 511, 255)-cyclic difference set classes found. Every difference set can be obtained by a decimation and a cyclic shift based on one of these examples. We used the trace representation of the associated two-level autocorrelation sequence—where α is an arbitrary primitive element of GF(1024)—to obtain a short listing.

Using the same methods as described in this paper, the search was later repeated by Song [7] with identical results.

- m-sequence
  - $\mathrm{Tr}(\alpha^i)$
- GMW sequences
  - $\mathrm{Tr}(\alpha^i + \alpha^{219i})$
  - $\mathrm{Tr}(\alpha^i + \alpha^{63i})$
  - $\mathrm{Tr}(\alpha^i + \alpha^{39i} + \alpha^{157i} + \alpha^{221i})$
  - $\mathrm{Tr}(\alpha^i + \alpha^{101i} + \alpha^{159i} + \alpha^{187i})$
  - $\mathrm{Tr}(\alpha^i + \alpha^{39i} + \alpha^{47i} + \alpha^{95i} + \alpha^{101i} + \alpha^{159i} + \alpha^{171i} + \alpha^{187i})$
- Extended GMW construction based on length 31 quadratic residue sequences
  - $\mathrm{Tr}(\alpha^i + \alpha^{5i} + \alpha^{7i} + \alpha^{9i} + \alpha^{19i} + \alpha^{25i} + \alpha^{69i})$
  - $\mathrm{Tr}(\alpha^i + \alpha^{5i} + \alpha^{25i} + \alpha^{45i} + \alpha^{69i} + \alpha^{87i} + \alpha^{101i} + \alpha^{107i} + \alpha^{121i} + \alpha^{159i} + \alpha^{187i} + \alpha^{237i} + \alpha^{245i} + \alpha^{479i})$
- 5-term construction [3]
  - $\mathrm{Tr}(\alpha^i + \alpha^{9i} + \alpha^{57i} + \alpha^{73i} + \alpha^{121i})$
- Construction using the Welch-Gong transformation [3] on the previous example
  - $\mathrm{Tr}(\alpha^i + \alpha^{10i} + \alpha^{11i} + \alpha^{12i} + \alpha^{13i} + \alpha^{14i} + \alpha^{15i} + \alpha^{138i} + \alpha^{139i} + \alpha^{140i} + \alpha^{141i} + \alpha^{142i} + \alpha^{143i})$

As mentioned before, the ten cases are inequivalent in the sense that none of them can be transformed into another by cyclic shifts and/or decimations. A separate question is whether any pair of the ten underlying designs is isomorphic, i.e. whether their incidence matrices can be transformed into each other by permuting rows and columns. This is obviously impossible when the incidence matrices have different ranks. The rank of a cyclic incidence matrix is the same as the linear complexity of the corresponding sequence, which in our case equals ten times the number of terms in the trace description above, for each sequence. The only pairs where isomorphism would be possible are the first and second pairs of GMW sequences. To try to settle these cases, first we examined the triple intersection numbers. Interestingly, it turned out that these numbers were identical for all four sequences. Then, we examined the quadruple intersection numbers. This test was quite lengthy, as it required the execution of $\sim 1.5 \cdot 10^{12}$ integer operations per case; but in the end, we were able to determine that no two designs were isomorphic.

TABLE 1. Coset class mappings

mod1023 N, S	mod341 N, S M, W	mod93 N, S M, W	mod33 N, S M, W	mod31 N, S M, W	mod11 N, S M, W	mod3 N, S M, W
$\mathcal{C}_{1023}^{(1023)}$	$\mathcal{C}_{341}^{(341)}$	$\mathcal{C}_{93}^{(93)}$	$\mathcal{C}_{33}^{(33)}$	$\mathcal{C}_{31}^{(31)}$	$\mathcal{C}_{11}^{(11)}$	$\mathcal{C}_{3}^{(3)}$
1, 1	1, 1	1, 1	1, 1	1, 1	1, 1	1, 1
	1, 1	1, 1	1, 1	1, 1	1, 1	1, 1
$\mathcal{C}_{341}^{(1023)}$	$\mathcal{C}_{341}^{(341)}$	$\mathcal{C}_{31}^{(93)}$	$\mathcal{C}_{11}^{(33)}$	$\mathcal{C}_{31}^{(31)}$	$\mathcal{C}_{11}^{(11)}$	$\mathcal{C}_{1}^{(3)}$
1, 2	1, 1	1, 2	1, 2	1, 1	1, 1	1, 2
	1, 2	1, 1	1, 1	1, 2	1, 2	1, 1
$\mathcal{C}_{33}^{(1023)}$	$\mathcal{C}_{11}^{(341)}$	$\mathcal{C}_{3}^{(93)}$	$\mathcal{C}_{33}^{(33)}$	$\mathcal{C}_{1}^{(31)}$	$\mathcal{C}_{11}^{(11)}$	$\mathcal{C}_{3}^{(3)}$
6, 5	6, 5	6, 5	1, 1	6, 5	1, 1	1, 1
	1, 1	1, 1	6, 5	1, 1	6, 5	6, 5
$\mathcal{C}_{93}^{(1023)}$	$\mathcal{C}_{31}^{(341)}$	$\mathcal{C}_{93}^{(93)}$	$\mathcal{C}_{3}^{(33)}$	$\mathcal{C}_{31}^{(31)}$	$\mathcal{C}_{1}^{(11)}$	$\mathcal{C}_{3}^{(3)}$
1, 10	1, 10	1, 1	1, 10	1, 1	1, 10	1, 1
	1, 1	1, 10	1, 1	1, 10	1, 1	1, 10
$\mathcal{C}_{31}^{(1023)}$	$\mathcal{C}_{31}^{(341)}$	$\mathcal{C}_{31}^{(93)}$	$\mathcal{C}_{1}^{(33)}$	$\mathcal{C}_{31}^{(31)}$	$\mathcal{C}_{1}^{(11)}$	$\mathcal{C}_{1}^{(3)}$
2, 10	1, 10	1, 2	2, 10	1, 1	1, 10	1, 2
	2, 1	2, 5	1, 1	2, 10	2, 1	2, 5
$\mathcal{C}_{11}^{(1023)}$	$\mathcal{C}_{11}^{(341)}$	$\mathcal{C}_{1}^{(93)}$	$\mathcal{C}_{11}^{(33)}$	$\mathcal{C}_{1}^{(31)}$	$\mathcal{C}_{11}^{(11)}$	$\mathcal{C}_{1}^{(3)}$
6, 10	6, 5	6, 10	1, 2	6, 5	1, 1	1, 2
	1, 2	1, 1	6, 5	1, 2	6, 10	6, 5
$\mathcal{C}_{3}^{(1023)}$	$\mathcal{C}_{1}^{(341)}$	$\mathcal{C}_{3}^{(93)}$	$\mathcal{C}_{3}^{(33)}$	$\mathcal{C}_{1}^{(31)}$	$\mathcal{C}_{1}^{(11)}$	$\mathcal{C}_{3}^{(3)}$
30, 10	30, 10	6, 5	1, 10	6, 5	1, 10	1, 1
	1, 1	5, 2	30, 1	5, 2	30, 1	30, 10
$\mathcal{C}_{1}^{(1023)}$	$\mathcal{C}_{1}^{(341)}$	$\mathcal{C}_{1}^{(93)}$	$\mathcal{C}_{1}^{(33)}$	$\mathcal{C}_{1}^{(31)}$	$\mathcal{C}_{1}^{(11)}$	$\mathcal{C}_{1}^{(3)}$
60, 10	30, 10	6, 10	2, 10	6, 5	1, 10	1, 2
	2, 1	10, 1	30, 1	10, 2	60, 1	60, 5

N : Number of cosets in the coset class,
S : Size of the cosets in the coset class,
M : Multiplicity of the mod 1023 cosets when mapped to other moduli,
W : Weight of the mod 1023 cosets when mapped to other moduli.

ACKNOWLEDGMENTS

The authors would like to acknowledge useful discussions with Guang Gong, Lloyd Welch, P. Vijay Kumar, Hong-Yeop Song and Roland Dreier.

REFERENCES

[1] L. D. Baumert, *Cyclic Difference Sets*, Lecture Notes in Math., vol. 182, Springer-Verlag, New York, 1971. MR **44:97**

[2] S. W. Golomb and H.-Y. Song, *On the Existence of Cyclic Hadamard Difference Sets*, IEEE Trans. Info. Theory, vol. 40, no. 4, July 1994, pp. 1266-1268.

[3] J.-S. No et. al., *Binary Pseudorandom Sequences of Period* $2^n - 1$ *with Ideal Autocorrelation*, IEEE Trans. Info. Theory, vol. 44, no. 2, March 1998, pp. 814-817. CMP 98:08

[4] L. D. Baumert and H. Fredricksen, *The Cyclotomic Numbers of Order Eighteen with Applications to Difference Sets*, Math. Comp. 21, 1967, pp. 204-219. MR **36:**6370
[5] U. Cheng, *Exhaustive Construction of (255, 127, 63)-Cyclic Difference Sets*, J. Combin. Theory, Ser. A 35, no. 2, 1983, pp. 115-125. MR **85b:**05038
[6] R. B. Dreier and K. W. Smith, *Exhaustive Determination of (511, 255, 127)-Cyclic Difference Sets*, Unpublished, 1991.
[7] H.-Y. Song, *Personal Communication*, April 1997 and August 1998.
[8] M. Hall, Jr., *A Survey of Difference Sets*, Proc. Am. Math. Soc. **7** (1956), 975-986. MR **18:**560h

Communication Sciences Institute, University of Southern California, Los Angeles, CA 90089-2565
E-mail address: pgaal@qualcomm.com; milly@mizar.usc.edu

Communication Sciences Institute, University of Southern California, Los Angeles, CA 90089-2565