Realizing All Index Generation Functions By the Row-Shift Method

Jon T. Butler
Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, CA 93943–5121 U.S.A.
Email: jon_butler@msn.com

Tsutomu Sasao
Department of Computer Science and Electronics
Meiji University
Kawasaki, Kanagawa 214-8571 JAPAN
Email: sasao@ieee.org

***Abstract*—We propose a method that allows the realization of all index generation functions using flexible decomposition charts. It is based on the first-fit decreasing heuristic used by Tarjan and Yao to store sparse matrices. We show that the first-fit-decreasing heuristic can yield nonminimal tables in the case of functions that do not satisfy the harmonic decay property. We show that an index generation function representation that just satisfies the harmonic decay property, called the perfect harmonic decay sequence, allows a simple matrix approach for calculating an error matrix, that describes the degree to which a given function representation departs from a perfect harmonic decay sequence. This gives insight into how function representations can be changed to realize the harmonic decay criteria. We also show the existence of sparse function representations for which no compression is possible. In such a case, we can still implement the corresponding index generation function, but it requires the largest resources possible.**

***Index Terms*—index generation functions, decomposition chart, row-shift decomposition, sparse table storage, harmonic decay, perfect harmonic decay sequence, first-fit-decreasing heuristic, nonmergable rows**

I. Introduction

Index generation functions [4] have application in the storage of sparse data. Applications include password files, memory-patch circuits, virus detection, and routing. Sasao [5] showed that the row-shift method in a decomposition chart representation of an index generation function often yields a significant reduction in complexity. To accomplish this, he developed a decomposition chart approach to accommodate index generation functions. The row-shift method is a way to adjust the position of indices in the decomposition chart so that it induces a smaller circuit [2], [5], [6].

In applying the row-shift method, the first-fit heuristic was used to find the amount of shift to apply. The row-shift method is often effective in reducing the complexity of the realization of index generation functions, especially when the function is sparse. Indeed, as shown in [2], the probability that the row-shift method can produce a realization depends on the degree of sparseness of the index values. A smaller number of index values corresponds to a larger probability of success. In this paper, we show a row-shift method can be applied to *all* index generation functions provided the decomposition chart has a flexible size.

II. Index Generation Functions

Definition 1. *An* **index generation function** f *is a multiple-valued function* $f : \{0,1\}^n = X \rightarrow \{0, 1, 2, \ldots, k\}$*, where there exists* $Y \subseteq X$*, such that an into and one-to-one mapping exists from* Y *to* $\{1, 2, \ldots, k\}$*, and where all other elements of* X *map to* 0*. Elements of* Y *are* **registered vectors***. The values of* $\{1, 2, \ldots, k\}$ *are* **indices***, while* 0 *is the* **ambient value***. Typically,* f *is* **sparse***, a term we use to (vaguely) describe the fact that there are many more ambient values than indices.*

Example 1. *Table I shows an index generation function with* $n = 4$ *variables,* x_1*,* x_2*,* x_3*, and* x_4 *and* $k = 7$ *indices,* 1*,* 2*,* 3*,* 4*,* 5*,* 6*, and* 7*. The table shows all assignments to the variables that map to an index. An assignment that does not map to an index is assumed to map to the ambient value* 0 *and is omitted. If this example represents a virus detection application, then a registered vector is a potential virus, and its index is an address where that virus is processed.* ■

TABLE I: Index Generation Function Example

x_1	x_2	x_3	x_4	f
0	0	0	0	**1**
0	0	1	1	**5**
0	1	0	0	**3**
0	1	1	0	**4**
1	0	0	0	**2**
1	1	1	0	**6**
1	1	1	1	**7**

III. Decomposition Charts

To modify a function so that it can be realized by physical circuits like LUTs in an FPGA, we seek to decompose the function into smaller functions. A useful tool is the **decomposition chart**. Table II shows a decomposition chart that realizes the function shown in Table I. Here, the four variables are divided into two parts. x_1x_2 label the rows, while x_3x_4 label the columns. By shifting the rows, we seek to redistribute the indices so that at most one index occurs in each column. Fig. 1a shows the circuit to accomplish this. Here,

TABLE II: Decomposition Chart Example

		0	0	1	1	x_3
x_1	x_2	0	1	0	1	x_4
0	0	**1**	0	0	**5**	
0	1	**3**	0	**4**	0	
1	0	**2**	0	0	0	
1	1	0	0	**6**	**7**	

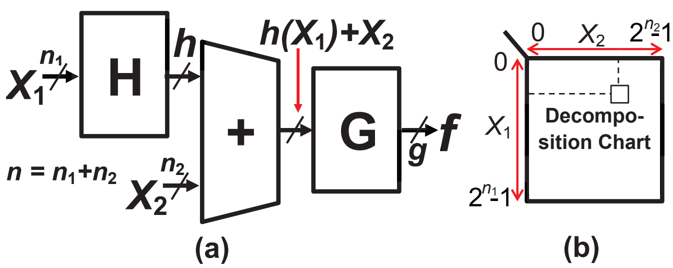

Fig. 1: Row-Shift Decomposition.

X_1 specifies the row number and h the amount of shift required for that row. The adder is a modulo adder, so that values shifted off the edge wrap around. Fig. 1b shows the corresponding decomposition chart. The process of designing the circuit shown in Fig. 1 is called the **row-shift method** [2], [5], [6].

From [2], we know that, for the example in Table II, no row-shift exists that results in at most one index per column. Specificially, there are seven indices and four columns, and, by the Pigeonhole Principle, at least one column has at least two indices. However, if we apply a different approach [8], we can use row-shifts to realize *any* decomposition chart. This is shown in Fig. 2 for the example shown in Table II.

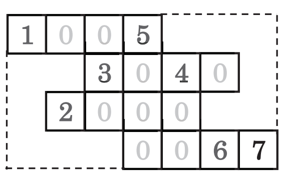

Fig. 2: Decomposition Chart Without a Size Limit.

Here, the row-shifts do not wrap around; they simply extend out, as needed. The row-shifts are chosen so that 1) no two indices occur in the same column and 2) as few columns as possible have only ambient values. In Fig. 2, the dotted lines define the extent of the decomposition chart needed for this row-shift. The circuit to realize this approach is shown in Fig. 1a where the adder is not a mod adder; rather, its output value accommodates the largest output value possible.

IV. Harmonic Decay

Tarjan and Yao [8] use the **first-fit** heuristic of Ziegler [9], in which $r(i)$ is computed one at a time, with $r(0)$ first, $r(1)$ second, etc., such that $r(i)$ is the smallest value such that no index of the i-th row falls into the same column as an index in a previously chosen row. This is improved by first arranging rows in decreasing order of the number of indices in each row, forming the **first-fit-decreasing** heuristic.

An important question is how to measure the extent to which an array is sparse. Tarjan and Yao [8] use the harmonic decay property.

Definition 2. *[8] Let $\eta(l)$, for $l \geq 0$, be the total number of indices in rows with more than l indices. Array A has the* **harmonic decay property***, if, for all l, $\frac{k}{l+1} \geq \eta(l)$. where k is the number of stored elements (indices).*

Example 2. *Consider an array in which the number of stored elements (indices) k is 144. Let there be 9 rows with 4 indices, 4 rows with 3 indices, 12 rows with 2 indices, and 72 rows with 1 index. This specification is shown in the first two columns of Table III. The third column shows $\eta(l)$, while the fourth (rightmost) column shows $\frac{k}{l+1}$. Comparing the third and fourth column shows that this array just satisfies the harmonic decay property, such that the condition $\frac{k}{l+1} \geq \eta(l)$ is always satisfied as $\frac{k}{l+1} = \eta(l)$.*

TABLE III: An array with the harmonic decay property

# of rows	l =# of indices	$\eta(l)$	$\frac{k}{l+1}$
9	4	0	
4	3	$9 \cdot 4 = 36$	$\frac{144}{3+1} = 36$
12	2	$36 + 12 = 48$	$\frac{144}{2+1} = 48$
72	1	$36 + 12 + 24 = 72$	$\frac{144}{1+1} = 72$
	0	$36 + 12 + 24 + 72 = 144$	$\frac{144}{0+1} = 144$

■

In an array with the harmonic decay property, the number of indices decreases relatively quickly as l increases, and is, therefore sparse. The following theorem by Tarjan and Yao [8] states that, if the harmonic decay property holds, then the first-fit-decreasing heuristic works well.

Theorem 1. *[8] Suppose that array A has the following harmonic decay property*

$$\frac{k}{l+1} \geq \eta(l),$$

where k is the total number of indices. Then, every row displacement $r(i)$ computed for A by the first-fit-decreasing heuristic satisfies $0 \leq r(i) \leq k$.

The specification $0 \leq r(i) \leq k$ guarantees that the linear array corresponding to A will not be 'too long'. However, in the next section, we show that the first-fit-decreasing heuristic may fail to find the absolute *minimum memory*.

V. The First-Fit-Decreasing Heuristic Does Not Always Minimize Memory

Example 3. *Table IV shows four rows of 0's and indices, with indices indicated by asterisks. On the left are the unshifted rows. Shown in the middle is the result of applying the first-fit-decreasing heuristic, and on the right is an optimum*

TABLE V: Beta Vectors for Perfect Harmonic Decay Sequences.

m	k	β_{10}	β_9	β_8	β_7	β_6	β_5	β_4	β_3	β_2	β_1	Density
2	2^2=4									1	2	0.66667
3	$2^2 3^2$=36								4	3	18	0.48000
4	$2^4 3^2$=144							9	4	12	72	0.37113
5	$2^4 3^2 5^2$=3,600						144	45	100	300	1,800	0.30138
6	$2^4 3^2 5^2$=3,600					100	24	45	100	300	1,800	0.25327
7	$2^4 3^2 5^2 7^2$=176,400				3,600	700	1,176	2,205	4,900	14,700	88,200	0.21822
8	$2^6 3^2 5^2 7^2$=705,600			11,025	1,800	2,800	4,704	8,820	19,600	58,800	352,800	0.19159
9	$2^6 3^4 5^2 7^2$=6,350,400		78,400	11,025	16,200	25,200	42,336	79,380	176,400	529,200	3,175,200	0.17071
10	$2^6 3^4 5^2 7^2$=6,350,400	63,504	7,840	11,025	16,200	25,200	42,336	79,380	176,400	529,200	3,175,200	0.15390

*arrangement. First-fit-decreasing results in 12 columns, while the optimum arrangement requires 11 columns. These memory counts include trailing 0's, which do not need to be stored. By not storing the trailing 0's, the optimum method is even better than the first-fit-decreasing method, requiring 9 columns versus 12 columns. This is shown in the last row of Table IV in the row labeled '*w/o Tr. 0s*'. We cannot do better than 9, since that is the number of stored indices.* ■

VI. Perfect Harmonic Decay Sequences

Definition 3. *The* **beta vector** $(\beta_m, \ldots, \beta_2, \beta_1)$ *of a harmonic decay sequence specifies how many rows contain a specified number of indices. Specifically, there are β_i rows with i indices.*

The total number of indices, k, in a sequence satisfies $k = \sum_{j=1}^{m} j\beta_j$. Further, $\eta(i) = \sum_{j=i+1}^{m} j\beta_j$ by definition.

Definition 4. *An array A has the* **perfect harmonic decay property** *if, for all i, $\frac{k}{i+1} - \eta(i) = \alpha_i = 0$, where $\eta(i)$ is the total number of indices in rows with more than i indices, for $i \geq 0$, and k is the total number of indices.*

Example 4. *Example 2 shows an array with the perfect harmonic decay property. Table V shows the beta vectors of arrays with the perfect harmonic decay property, where the vector width m ranges from 2 up through 10. Note that k takes on specific values depending on m, as shown in the second column in Table V.* ■

A. Calculation of β_i as a Function of k

Table V has some notable characteristics. For example, the values in the column labeled "β_1" are exactly 6 times the corresponding values in the column labeled "β_2", except for the first row. The values in the column labeled "β_2" are exactly 3 times the corresponding values in the column labeled "β_3", except for the first two rows, etc.. We can make the following general statement about these relationships.

TABLE IV: Comparing the First-Fit-Decreasing Heuristic With An Optimum Algorithm

Original	F-F-Decr. (12)	Optimum (11)
0***00 **0000 *0000* 00**00	0***00 **0000 *0000* 00**00	*0000* **0000 00**00 0***00
# Columns	0********00*	********00
w/o Tr. 0's	**(12)**	**(9)**

Theorem 2. *Let β_i be the number of rows with i indices in a perfect harmonic decay sequence, where each row has m elements. Let k be the total number of indices across all rows; $k = \sum_{j=1}^{m} \beta_j$. Then,*

$$\beta_i = \frac{k}{m^2}, \text{ for } i = m \text{ and,} \tag{1}$$

$$\beta_i = \frac{k}{i^2(i+1)}, \text{ for } 1 \leq i \leq m-1. \tag{2}$$

Proof: Since this is a perfect harmonic decay sequence,

$$\frac{k}{1} - \eta(0) = \alpha_0 \geq 0,$$
$$\frac{k}{2} - \eta(1) = \alpha_1 \geq 0,$$
$$\frac{k}{3} - \eta(2) = \alpha_2 \geq 0,$$
$$\vdots$$
$$\frac{k}{\mu+1} - \eta(\mu) = \alpha_\mu \geq 0. \tag{3}$$

apply with the inequalities replaced by equalities. (1) follows directly from (3) by solving for β_m in $\eta(m-1) = m\beta_m = \frac{k}{m}$. Regarding (2), because this is a perfect harmonic decay sequence, we have, for any i, such that $1 \leq i \leq m-1$,

$$\frac{k}{i} = \eta(i-1) = i\beta_i + (i+1)\beta_{i+1} + (i+2)\beta_{i+2} + \ldots + m\beta_m. \tag{4}$$

But,

$$\frac{k}{i+1} = \eta(i) = (i+1)\beta_{i+1} + (i+2)\beta_{i+2} + \ldots + m\beta_m,$$

and we can write (4) as

$$\frac{k}{i} = i\beta_i + \frac{k}{i+1}.$$

Solving for β_i yields (2). □

TABLE VI: Relation Among Elements of Beta Vectors

β_i (2) $1 \le i \le m-1$				β_i (1) $i = m$
β_1	$= \frac{k}{1^2 2}$	$= \frac{k}{2}$	$= 6.00000 \times \beta_2$	$\frac{k}{1^2}$(=1)
β_2	$= \frac{k}{2^2 3}$	$= \frac{k}{12}$	$= 3.00000 \times \beta_3$	$\frac{k}{2^2}$(=4)
β_3	$= \frac{k}{3^2 4}$	$= \frac{k}{36}$	$= 2.22222 \times \beta_4$	$\frac{k}{3^2}$(=9)
β_4	$= \frac{k}{4^2 5}$	$= \frac{k}{80}$	$= 1.87500 \times \beta_5$	$\frac{k}{4^2}$(=144)
β_5	$= \frac{k}{5^2 6}$	$= \frac{k}{150}$	$= 1.68000 \times \beta_6$	$\frac{k}{5^2}$(=100)
β_6	$= \frac{k}{6^2 7}$	$= \frac{k}{252}$	$= 1.55556 \times \beta_7$	$\frac{k}{6^2}$(=3,600)
β_7	$= \frac{k}{7^2 8}$	$= \frac{k}{392}$	$= 1.46939 \times \beta_8$	$\frac{k}{7^2}$(=11,025)
β_8	$= \frac{k}{8^2 9}$	$= \frac{k}{576}$	$= 1.40625 \times \beta_9$	$\frac{k}{8^2}$(=78,400)
β_9	$= \frac{k}{9^2 10}$	$= \frac{k}{810}$	$= 1.35802 \times \beta_{10}$	$\frac{k}{9^2}$(=63,504)

The relations among various values of β_i noted at the beginning of this section can now be quantified. For example, we noted that the values in the column labeled "β_1" are exactly 6 times the corresponding values in the column labeled "β_2". Table VI shows the expressions for β_1 through β_9 for the β's shown in Table V, as specified by (1) and (2). For example, the observation that the values in the column labeled "β_1" are exactly 6 times the corresponding values in the column labeled "β_2" appears as the second line in Table VI.

B. Calculation of the Basic Perfect Harmonic Sequence

Up to this point, we have shown relations that must hold between various β_i values, as a function of k, the total number of indices. It remains to show how β_i depends on k.

Theorem 3. *Let k be the number of indices in a perfect harmonic decay sequence, where the row width is m. k is the least common multiple of* $\{1^2, 2^2, \ldots, m^2\}$.

Proof: Because the sequence is a perfect harmonic sequence, (1) and (2) hold. Since β_i is an integer, k must be a multiple of i^2, *for all* i, such that $1 \le i \le m$. Because k is a multiple of i^2, for $1 \le i \le m$, k is a multiple of $i+1$ for $1 \le i \le m-1$. Since the sequence is a perfect harmonic sequence, k is the *least* common multiple of $\{1^2, 2^2, \ldots, m^2\}$. □

The second column in Table V shows the calculation of k for various m. This sequence, (4, 36, 144, 3,600, 3,600, 176,400, 705,600, 6,350,400, 6,350,400), is Sloane's integer sequence A051418 [7].

There are as many conditions as there are divisors of m. Computationally, it is easier to check only the prime divisors of k. If p^m divides k, so does $p^{m'}$, where $m > m'$. This suggests that we need only check the largest multiple of prime divisors. Therefore, we can restate Theorem 3 as follows.

Corollary 1. *Let k be the total number of indices in a perfect harmonic decay sequence, and let m be the maximum number of indices in any row. Let 2, 3, …, and z be prime divisors of k. Then,*

$$n = [2^{\xi_2} 3^{\xi_3} \ldots z^{\xi_z}]^2, \tag{5}$$

where ξ_i is the largest k such that $i^k \le m$.

Theorem 2 and either Theorem 3 or Corollary 1 completely characterize the sequences with the perfect harmonic decay property.

C. Density of Indices in Sequences That Have the Perfect Harmonic Decay Property

From the rightmost column of Table V, it can be seen that the density of indices decreases as m increases. These values were computed in a straightforward computation, in which we take the size of the array to be $(\beta_1 + \beta_2 + \ldots + \beta_m)m$, since $\beta_1 + \beta_2 + \ldots + \beta_m$ is the total number of rows with indices, and where m is the array width just wide enough to contain the longest column of all indices, a column of width m.

Theorem 4. *The density δ of indices in a perfect harmonic array approaches 0 as $m \to \infty$.*

Proof: From the discussion above, the density of indices in a perfect harmonic array is

$$\delta = \frac{(\beta_1 1 + \beta_2 2 + \beta_3 3 + \ldots + \beta_m m)}{(\beta_1 + \beta_2 + \beta_3 + \ldots + \beta_m)m} = \frac{N}{D}. \tag{6}$$

The numerator N is k, the total number of indices. From (1) and (2), we can write the denominator as

$$D = (\beta_1 + \beta_2 + \beta_3 + \ldots + \beta_m)m =$$

$$mk\left(\frac{1}{1^2 2} + \frac{1}{2^2 3} + \frac{1}{3^2 4} + \ldots + \frac{1}{(m-1)^2 m} + \frac{1}{m^2}\right).$$

From $\frac{1}{i(i+1)} = \frac{1}{i} - \frac{1}{i+1}$, we have

$$D = mk\left(\frac{1}{1^2} + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \ldots + \frac{1}{(m-1)^2} + \frac{1}{m^2}\right) -$$

$$mk\left(\frac{1}{1\cdot 2} + \frac{1}{2\cdot 3} + \frac{1}{3\cdot 4} + \frac{1}{4\cdot 5} + \frac{1}{5\cdot 6} + \ldots + \frac{1}{(m-1)m}\right).$$

After some algebra and repeating $\frac{1}{i(i+1)} = \frac{1}{i} - \frac{1}{i+1}$, we have

$$D = mk\left(1 + \frac{1}{2^2} + \frac{1}{3^2} + \ldots + \frac{1}{(m-1)^2} + \frac{1}{m^2}\right) - mk.$$

The series is the Basel series, which was solved by Euler in 1734[1]: specifically, $\sum_{i=1}^{m} \frac{1}{i^2} \to \frac{\pi^2}{6}$ as $m \to \infty$. After cancellations and substitutions, we have $\delta = N/D \to 0$, as $m \to \infty$. □

D. Fraction of Square Arrays That Have the Harmonic Decay Property

Table VII shows the fraction of square arrays that have the harmonic decay property. The first three rows of Table VII were calculated by exhaustive enumeration. Specifically, *all* square arrays were enumerated and tested. An array that has the harmonic decay property contributes 1 to the count. In the case of 2×2 ($m = 2$) arrays, there are a total of 16 arrays, and nine or 0.5625 satisfied the criteria. In the case of $m = 3$ and 4, the fraction of arrays that have the harmonic decay property is 0.2832 and 0.0506, respectively.

TABLE VII: Fraction of Square Arrays That Have the Harmonic Decay Property

m	# Inst.	Max./Sample Sz.	Fraction	Avg Dens δ
Exhaustive Search				
2	9	16	.562500	0.33333
3	145	512	.283203	0.35862
4	3313	65536	.050552	0.26954
Monte Carlo Simulation				
2	56404	100000	.5640400	0.33297
3	28336	100000	.2833600	0.35895
4	5005	100000	.0500500	0.26897
5	2178	1000000	.0021780	0.20825
6	820	10000000	.0000820	0.20986
7	153	100000000	.0000015	0.19394
8	4	1000000000	.0000000	0.15625
9	0	1000000000	.0000000	-

Excessive computation time prevented the computation of all 5×5 or larger arrays. The next eight entries were computed in a Monte Carlo simulation, where $m \times m$ matrices are randomly generated and tested to determine if they have the harmonic decay property. The first three of these entries correspond to arrays for which we can do an exhaustive enumeration. This serves as a comparison between the Monte Carlo simulation and exhaustive enumeration.

In Table V, the rightmost column shows the exact density of indices for each perfect harmonic decay sequence. In the case of Table V, the array size is $(\beta_1+\beta_2+\ldots+\beta_m)m$, where m is the largest i such that β_i is nonzero. This is the smallest array that will fit around the specified perfect harmonic decay sequence. The density is then computed as $(\beta_1 \cdot 1 + \beta_2 \cdot 2 + \ldots + \beta_i \cdot i + \ldots + \beta_m m)/((\beta_1 + \beta_2 + \ldots + \beta_m)m)$. Table V shows that the densities associated with the perfect harmonic decay sequences are similar to those shown in Table VII. It should be noted that each row in Table V represents exactly *one* array, albeit a special one, namely the perfect harmonic decay sequence that just satisfies the harmonic decay sequence inequalities.

Tarjan and Yao [8] state "It is useful to reflect a bit on the meaning of harmonic decay. If A has the harmonic decay property, at least half the nonzeros in A must be in rows with only a single index." Note, from Table V, that, in the case of perfect harmonic sequences, *exactly* half of the indices are in rows with only a single index.

E. Matrix Representation of the Alpha Vector Values

The perfect harmonic sequence can be used as a method for the direct calculation of the alpha vectors from the beta vectors. This can be seen in Tables VIII and IX. Consider $m = 4$, whose perfect harmonic beta vector is $(\beta_4, \beta_3, \beta_2, \beta_1) = (9, 4, 12, 72)$. Table VIII shows the alpha vector calculation to the left and the calculation of $\eta(i)$ to the right. What this shows is how a change in the beta vector affects the alpha vector. For example, in Table VIII if $\beta_1 = 72$ is increased by 1, then all of the elements of the alpha vector are increased, thus 'improving' the harmonic decay property.

TABLE VIII: Calculation of the Alpha Vector and $\eta(i)$

α_i	$\eta(i)$
$\frac{k}{1} - \eta(0) = \alpha_0$	$72 \cdot 1 + 12 \cdot 2 + 4 \cdot 3 + 9 \cdot 4 = \eta(0) = 144$
$\frac{k}{2} - \eta(1) = \alpha_1$	$12 \cdot 2 + 4 \cdot 3 + 9 \cdot 4 = \eta(1) = 72$
$\frac{k}{3} - \eta(2) = \alpha_2$	$4 \cdot 3 + 9 \cdot 4 = \eta(2) = 48$
$\frac{k}{4} - \eta(3) = \alpha_3$	$9 \cdot 4 = \eta(3) = 36$

TABLE IX: Delta Calculation of Alpha Vectors

Elements of Beta Vector	Change in Alpha Vector
$72 + \Delta_1$	$\alpha_0 \to \alpha_0$
	$\alpha_1 + \frac{1}{2}\Delta_1 \to \alpha_1$
	$\alpha_2 + \frac{1}{3}\Delta_1 \to \alpha_2$
	$\alpha_3 + \frac{1}{4}\Delta_1 \to \alpha_3$
$12 + \Delta_2$	$\alpha_0 \to \alpha_0$
	$\alpha_1 - \Delta_2 \to \alpha_1$
	$\alpha_2 + \frac{2}{3}\Delta_2 \to \alpha_2$
	$\alpha_3 + \frac{1}{2}\Delta_2 \to \alpha_3$
$4 + \Delta_3$	$\alpha_0 \to \alpha_0$
	$\alpha_1 - \frac{3}{2}\Delta_3 \to \alpha_1$
	$\alpha_2 - 2\Delta_3 \to \alpha_2$
	$\alpha_3 + \frac{3}{4}\Delta_3 \to \alpha_3$
$9 + \Delta_4$	$\alpha_0 \to \alpha_0$
	$\alpha_1 - \Delta_4 \to \alpha_1$
	$\alpha_2 - 2\frac{2}{3}\Delta_4 \to \alpha_2$
	$\alpha_3 - 3\Delta_4 \to \alpha_3$

The data in Table IX can be represented in matrix form, $\overrightarrow{\alpha} = \overrightarrow{A}\,\overrightarrow{\Delta}$, where

$$\overrightarrow{\alpha} = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix}, \overrightarrow{A} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{1}{2} & -1 & -\frac{3}{2} & -1 \\ \frac{1}{3} & \frac{2}{3} & -2 & -2\frac{2}{3} \\ \frac{1}{4} & \frac{1}{2} & \frac{3}{4} & -3 \end{bmatrix}, \overrightarrow{\Delta} = \begin{bmatrix} \Delta_1 \\ \Delta_2 \\ \Delta_3 \\ \Delta_4 \end{bmatrix}. \quad (7)$$

From the data, it can be seen that, when Δ_1 is increased, then *all* elements of the α_i vector are increased. This is due to the fact that a positive Δ_1 value corresponds to an increase in β_1, the number of indices due to rows that have exactly one index. This increases η_1 and leaves all other η_i values unchanged. Since it increases k, the total number of indices, it increases *all* α_i values, except α_0. Thus, an increase in β_1 increases the margin by which a binary sequence has the harmonic decay property.

The matrix representation shows quantitatively what is intuitive. That is, when there are more rows with many indices, then it is less likely that the two-dimensional array has the harmonic decay property. Quantitatively, the larger β values cause *decreases* in the α values, which decrease the α values, making it less likely that the α values will be positive.

VII. Worst Case Arrangements

In this section, we consider worst case arrangements of indices. These are arrangements in which it is impossible to merge any two rows because, regardless of the relative shift of each row to each other, there will always be at least one column with two indices. Tarjan and Yao [8] show that, if the indices are distributed according to the harmonic decay property, the number of memory locations will be close to k, the number of indices.

Fig. 3 shows an example 2×57 array. Here, indices are shown as large dots, while ambient values are shown as small dots. The two rows cannot be merged together. That is, when any algorithm is applied to these two rows to form a shortest sequence of memory locations to store the indices, the rows must be contiguous; i.e. the maximum of $2 \cdot 57 = 114$ locations is needed. There are $8 + 16 = 24$ indices, for a density of 0.21. We now use another measure for sparseness besides the harmonic decay property. More specifically,

Definition 5. *The* **density** *of an $m \times p$ array is the total number of indices divided by the total number of array elements, $\frac{k}{m \cdot p}$.*

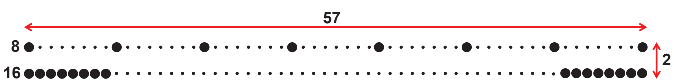

Fig. 3: Array With Two Nonmergable Rows

Example 5. *In the array shown in Fig. 3, the density is $\frac{24}{114} \simeq 0.21$. However, a "stretched" version of Fig. 3 has a lower density. In Fig. 3, the density of the first row is $\frac{8}{57}$, which is approximately $\frac{1}{8}$. Indeed, if this row is extended in a natural way, with an index occurring every eighth position, then the density approaches $\frac{1}{8}$ as row length increases. For the second row, the density, in the limit, approaches 0. Thus, the density of this $2 \times p$ array approaches $\frac{1}{2} \cdot \frac{1}{8} = \frac{1}{16} \simeq 0.06$, as p increases.* ■

Example 6. *The arrays shown in Fig. 3 can only be chosen once. That is, if there is more than one copy of the first row or more than one copy of the second row, the copies can be merged with themselves. Fig. 4 shows a row that cannot be merged with itself. Note that there can be more than one copy of this row if the indices are all distinct.* ■

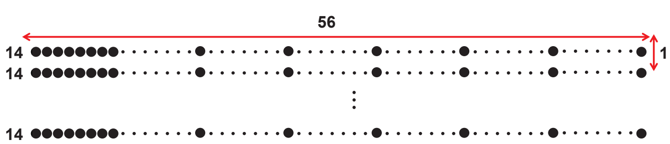

Fig. 4: A Row That Is Nonmergable With Itself

In general, we can say

Theorem 5. *There exists an $m \times p$ array consisting of m nonmergable sequences whose density approaches $\frac{1}{\eta}$, as p increases, where $\eta > 2$ is some fixed (possibly large) integer.*

Theorem 5 says that, for some (possibly very small) density, we can find an $m \times p$ array such that all m rows are nonmergable for any m. This is counter to the intuition that low-density arrays are likely to have mergeable rows. Indeed, this is counterintuitive to the result of Tarjan and Yao [8], which says that, if the number of indices in rows of an $m \times p$ array has the harmonic decay property, then the number of memory locations is approximately the same as the total number of indices.

VIII. Concluding Remarks

Index generation functions are multiple-valued functions [4] in which the input variables are binary valued, and the output variables are multiple-valued. While the row-shift method is effective in the design of index generation functions when the function is sparse, it fails in the case of functions that are less sparse. [2] addresses the issue of when the row-shift method is effective. It leaves open the question of whether some variant of the row-shift method applies to all index generation functions. In this paper, we show a row-shift method that can be applied to all index generation functions regardless of the degree of sparseness. It is shown in [8] that the first-fit-decreasing heuristic produces 'reasonably' good compression when the array has the harmonic decay property.

However, we show that the first-fit-decreasing heuristic produces nonminimal arrays. We analyze the perfect harmonic decay property, which is an extreme of the harmonic decay property. We show that a vector $\overrightarrow{\Delta}$ of delta values can be used to specify a distance measure for sparseness, and gives insight as to how arrays can be modified to have the harmonic decay property. We show the existence of sparse arrays for which absolutely no compression is possible by a row-shift method.

Acknowledgment

This research is partly supported by a Japan Society for the Promotion of Science (JSPS) Grant-in-Aid for Scientific Research. Three anonymous reviewers provided comments to improve the paper.

References

[1] R. Ayoub, "Euler and the zeta function," *Amer. Math. Monthly* Vol. 81, pp. 106786, 1974.

[2] J. T. Butler and T. Sasao, "Analysis of cyclic row-shift decompositions for index generation functions," *The Workshop on Synthesis and Sys. Integration of Mixed Info. Tech. (SASIMI-2018)*, March 26–27, 2018.

[3] S. Even, D. I. Lichtenstein, and Y. Shiloach, Remarks on Ziegler's method for matrix compression, *Unpublished manuscript*, 1977.

[4] T. Sasao, *Memory-Based Logic Synthesis*, Springer, New York, Dordrecht, Heidelberg, London. 2011.

[5] T. Sasao, "Row-shift decompositions for index generation functions," *Design, Automation and Test in Europe, (DATE-2012)*, March 12-16, 2012, Dresden, Germany, pp. 1585–1590.

[6] T. Sasao, "Cyclic row-shift decompositions for incompletely specified index generation functions," *The 22nd International Workshop on Logic Synthesis, (IWLS)*, June 7–8, 2013, Austin, TX.

[7] N. J. A. Sloane, https://oeis.org/A051418/internal.

[8] R. E. Tarjan and A. C.-C. Yao, "Storing a sparse table," *Comm. of the ACM*, Vol. 22, No. 11, November 1979, pp. 606–611.

[9] S. F. Ziegler, "Smaller faster table driven parser," *Unpublished manuscript*, Madison Academic Computing Center, University of Wisconsin, Madison, Wisconsin, 1977.