A Combinatorial Application of Necklaces: Modeling Individual Link Failures in Parallel Network-on-Chip Interconnect Links

Arseniy Vitkovskiy, Paul Christodoulides, Vassos Soteriou

Abstract— **The advent of the multicore era [1, 2] has made the execution of more complex software applications more efficient and faster. On-chip communication among the processing cores, in the form of packetized messages, is managed with the use of on-chip networks (NoCs) [3]. Routers handling on-chip communication are point-to-point topologically interconnected using parallel links laid onto the silicon surface comprising a number of individual parallel wires. With the underlying interconnect structure becoming denser, due to improvements in CMOS technology, parallel links become susceptible to wear-out [4], with permanent link failures inhibiting communication completely and indefinitely. It is hence critical to explore their failure patterns in the wires comprising these links and hence build mechanisms which can recover corrupted in-transit data [5, 6]; since no real data from chip manufacturers exist, the derivation of a mathematical model in aiding the understanding of the distribution of individual wire faults in parallel on-chip links becomes mandatory. This paper takes the first steps in such direction. First it is shown how the given problem reduces to an equivalent combinatorial problem through partitions and necklaces. Then a model that counts certain classes of necklaces is derived by making a separation between periodic and aperiodic cases. The model is tested against a brute-force algorithm to prove its exactness. Finally the obtained model is used in finding the probability distribution of the size of the fault segment of wires in a parallel NoC-based multicore chip.**

Index Terms—**Network-on-Chip, fault-tolerance, probability distribution, integer partition, necklace, combinatorics, modeling.**

I. INTRODUCTION

IN this article we derive and demonstrate the combinatorics-based models that can be used to calculate the spatial probability distribution of wire faults in a parallel Network-on-Chip (NoC) [3] interconnect link given its width (summation of the numbers of healthy and unhealthy[1] wires in this parallel link), and a given number of faulty wires that can appear in this link. In previous studies [5, 6] that target the recovery of corrupted packetized data being re-transmitted through Partially Faulty Links (PFLs) it has been shown that the

[1] The terms "unhealthy", "corrupted", and "faulty" are used interchangeably throughout this article.

consecutiveness[2] or "clustering" of these faulty wires directly determines the recovery latency required to restore received corrupted data among a pair of sender-receiver link-interconnected NoC routers. We are, therefore, particularly interested in the number of such consecutive faulty wires in a parallel NoC link as the "maximum wire fault clustering" directly determines the number of additional or overhead clock cycles that are required to re-transmit a flit over a faulty parallel link that are necessary for data reconstruction and recovery at the downstream (receiver) router.

To achieve this we derive a novel algorithm that can be used to determine the segmentation probability for an ordered collection of objects (i.e., parallel wires) of two distinct classes (faulty wires and healthy or "non-faulty" wires). The adjacent faulty wires form fault-segments (faulty wire clusters) separated by at least one healthy wire. As stated, the size of the largest segment determines the additional clock-cycle delay required by the Partially Faulty Link Recovery Mechanism (PFLRM) demonstrated in [5, 6] to recover corrupted flit[3] data at a receiver router. We pay particular attention to the wire fault segmentation since the PFLRM mechanism is based on a flit data recovery scheme that utilizes a 1-bit circular rotation (per cycle of flit recovery) of the received flit vectors at the receiver router to extract and combine healthy bit portions from these corrupted received flit copies (at each cycle) with the rotated intermediate flit data results from the previous clock cycle(s) to eventually re-generate the flit in its healthy form downstream (the reader is urged to refer to [5, 6] for details). Hence, to be able to construct our faulty wire spatial distribution probabilistic model, the link comprised of parallel wires can be abstractly represented as a ring of parallel wires (since the rotational nature of the PFLRM mechanism virtually connects together the two wires found at the two ends of the parallel wire; refer to Fig. 1 for demonstration). We assume a ***random spatial distribution*** of faulty wires in the parallel NoC link, and aim to determine the probability distribution of corrupted (and non-corrupted) flit data bits. A flit is assumed to consist of a number of binary bits, equal to the width of the parallel link, i.e. each wire in the parallel NoC link carries one bit of data.

To reach the goal of a complete mathematical model describing the probability distribution of the length of a fault-segment for a given number of wires and faulty wires

[2] The terms "consecutive", "clustering", "adjacent", and "segment(s)" are used interchangeably throughout this article.
[3] A flit or "flow-control unit" is a logical segment of a packetized NoC message.

we make use of a series of combinatorial arguments that include partitions and necklaces. Necklaces, apart from their intrinsic usefulness in the field of Combinatorics, have proven to be a powerful tool in other areas of Mathematics and other sciences. Some customary notions and theories related to necklaces include the Lyndon word [7], the actual homonym *necklace problem* (see, for example [8]), the necklace spitting problem [9], and most notably a proof of Fermat's little theorem [10].

The rest of this article is organized as follows. In Section II the problem is formally defined. In Section III we explain the algorithm that leads to the determination of the probability distribution of the length of fault-segments for a given number of wires and faulty wires comprising a parallel NoC link, which is constructed through basic counting principles and probability rules, where appropriate, and through a derivation showing its correspondence to an equivalent necklace problem. In Section IV a series of results based on the derived model are presented, and then proven to coincide with the results of a *brute-force algorithm* implemented with the relevant Matlab software toolbox. In Section V an arithmetic example demonstrates the effectiveness of the obtained analytical model. Finally, Section VI concludes this article and sets future work directions with regard to the findings of this study.

II. PROBLEM DEFINITION

THERE is a [parallel] NoC link consisting of W wires that are placed in parallel, wrapped around a common axis forming a *ring* shape (refer to Fig. 1). Each of these wires may be either healthy or faulty, and cannot possess both of these states. The number F and *position* (placement) of faulty wires are both **random**, where $0 \le F \le W$.

Consecutively (adjacent) positioned faulty wires form a *fault segment*. Let S denote the *size* (or *length*) of the largest fault segment present in the link, where $0 \le S \le F$.

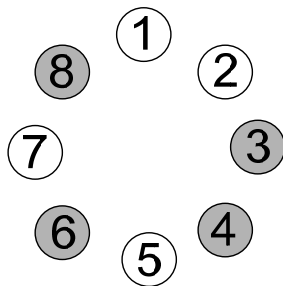For given values of W and F, we seek to find the probability distribution of S, $P_W(S \mid F)$.

Fig. 1. An example of a parallel link (cross section) consisting of $W = 8$ wires with $F = 4$ faulty wires (shaded), where the largest fault-segment size is $S = 2$ (formed by the clustering of faulty wires 3 and 4). In reality, wires 1 and 8 are physically located at the two opposite ends of the parallel link.

III. ALGORITHM DESCRIPTION

HEREAFTER we present an algorithm in order to find the probability $P_W(S \mid F)$ for each value of S for given values of W and F. We find it useful to demonstrate

the construction of the algorithm through arithmetic examples that clarify all notions involved.

A. Number of Possible Wire Arrangements

Let $A(W, F)$ denote the set of all possible wire arrangements, for given W and F values. The *cardinality* (i.e., number of elements) of set $A(W, F)$ is simply equal to the number of combinations in choosing F faulty wires out of W wires, given by

$$|A(W,F)| = \binom{W}{F} = \frac{W!}{F!(W-F)!}. \tag{1}$$

Similarly, let $A(W, F, S) \subset A(W, F)$ denote the set of all possible wire arrangements for given W, F and S values. Then, the problem reduces to finding $|A(W,F,S)|$ which when divided by $A(W, F)$ will yield exactly the required probability distribution $P_W(S \mid F)$.

B. Initial Values

As a first step, it is not difficult to see (using basic counting principles and probability rules) that for certain value choices of S and F, $P_W(S \mid F)$ can be obtained as follows

$$P_W(0 \mid 0) = 1, \quad P_W(1 \mid 1) = 1,$$
$$P_W(W \mid W) = 1, \quad P_W(W-1 \mid W-1) = 1,$$
$$P_W(S \mid F) = (S+1) \Big/ \binom{W}{F} \quad \text{(if } F = (W-F)S\text{)},$$
$$P_W(S \mid F) = W \binom{W-S-2}{F-S} \Big/ \binom{W}{F} \tag{2}$$
$$\text{(if } \lceil (F+1)/2 \rceil \le S \le F\text{)}.$$

Thus, the task is reduced to finding the probability distribution $P_W(S \mid F)$ for all $0 < S < \lceil (F+1)/2 \rceil$, for given values of W and F.

C. Size of Fault-Segment

Let H denote the number of healthy wires in a parallel NoC link, i.e., $H = W - F$. From the problem definition, the size of the largest fault segment S has a lower bound which is equal to zero. However, it is possible to define the greatest lower bound of S more precisely (refer to Example 1 for demonstration) as

$$\lceil F/H \rceil \le S \le F. \tag{3}$$

Example 1: Let $W = 12$, $F = 7$. Then $H = W - F = 5$ and the greatest lower bound of S is $\lceil F/H \rceil = \lceil 7/5 \rceil = 2 \le S$. Consequently for this case S can never be equal to 0 or 1. An illustration of such a wire arrangement is

$$\times \times \circ \times \times \circ \times \circ \times \circ \times \circ,$$

where the link is shown as an "unwrapped" transverse section, with $\circ$ and $\times$ denoting a healthy and faulty wires,

respectively. The same link/wire representation is adopted throughput the remaining length of this article.

D. Number of Fault-Segments

Let σ denote the number of fault-segments in a parallel NoC link. It is not difficult to see (refer to the demonstration exhibited in Example 2) that

$$\lceil F / S \rceil \le \sigma \le \min(F - S + 1, H), \quad S > 0. \tag{4}$$

Example 2: Let $W = 12$, $F = 7$ and $S = 4$. Then, $\sigma_{min} = \lceil F / S \rceil = \lceil 7 / 4 \rceil = 2$. Moreover,

$$\sigma_{max} = \min(F - S + 1, H) = \min(4,5) = 4 (= F - S + 1).$$

Such wire arrangements for σ = 2, 3 and 4, are respectively, the following:

××××○××××○○○○, ××××○××○×○○○, ××××○×○×○×○○,

Now let $S = 2$ for the same W and F values. Then, $\sigma_{min} = \lceil F / S \rceil = \lceil 7 / 2 \rceil = 4$. Moreover,

$$\sigma_{max} = \min(F - S + 1, H) = \min(6,5) = 5 (= H).$$

Such wire arrangement for σ = 4 and 5, are respectively, the following:

××○××○××○×○○, ××○××○×○×○×○.

E. String Representation of Wire Arrangements

Let a wire arrangement $a \in A(W, F, S)$ be represented by the string $l = s_1 s_2 ... s_H$, where s_i is the size of the ith fault-segment followed by a single healthy wire. Making the convention that $s_1 = S$, we have:

$$\sum_{i=1}^{H} s_i = F \qquad (s_i \in Z_0^+ \le S, \quad s_1 = S). \tag{5}$$

Note that (5) allows $s_i = 0$, denoting an empty fault segment followed by a single healthy wire (refer to Example 3 below).

Example 3: Let $W = 12$, $F = 6$ and $S = 2$. Then $H = W - F = 6$. Clearly, one of the respective wire arrangements ××○○××○×○○×○ can be expressed by the string $l = s_1 s_2 s_3 s_4 s_5 s_6 = 202101$.

The string representation for a wire arrangement, as described above, will allow us to count the actual number of wire arrangements. We now introduce some terminology that will help us to reach this goal.

Definition 1: (a) The string $l = s_1 s_2 ... s_H$ is said to be *periodic iff* (i) H is a composite integer, and (ii) there exists $t \in Z^+$ such that $s_{i+t} = s_i, \forall i \in Z^+$. We call t the *period of string l*.

(b) If there is more than one t satisfying the condition (ii) above, then the string is said to have *multiple* periods that are all prime divisors of H, excluding 1 and H.

(c) If condition (ii) is not satisfied, although the string l is *non-periodic*, for the sake of simplicity, the period is considered to be $t = H$.

(d) The *period t_w of a wire arrangement $a \in A(W, F, S)$*

that is represented by the string l of period t, can be obtained as follows.

$$t_w = \sum_{i=1}^{t} (s_i + 1) = \sum_{i=1}^{t} s_i + t. \tag{6}$$

Example 4: As a demonstration of (6) consider the wire arrangement ××○×○××○×○ $\in A(10,6,2)$ which has a period of $t_w = 5$ (while the corresponding string $s_1 s_2 s_3 s_4 = 2121$ is of period $t = 2$).

Definition 2: The *frequency f* of string $l = s_1 s_2 ... s_H$ is the number of occurrences of a repeating substring $l_t = s_1 s_2 ... s_t$ within l, where t is the period of l, and is given by

$$f = H / t. \tag{7}$$

Clearly, if string l has multiple periods, hence it has multiple frequencies as well.

Definition 3: The *frequency f_w* of the wire arrangement $a \in A(W, F, S)$ is the number of occurrences of a repeating sub-arrangement a_{t_w} within a, where t_w is the period of a, and it is given by

$$f_w = W / t_w. \tag{8}$$

Clearly, using (6), (7), and (8) and Definition 2, the respective frequencies f_w and f of the wire arrangement $a \in A(W, F, S)$ and the corresponding string l are equal.

Note here that, due to the convention in the definition of string l above (refer to (5)), one string l corresponds to t_w equivalent rotations of wire arrangements (refer to Example 5). If string l is non-periodic, i.e. $t = H$, then, by (5), the number of equivalent rotations of wire arrangements is

$$t_w = \sum_{i=1}^{t} s_i + t = F + H = W. \tag{9}$$

Example 5: Let $W = 8$, $F = 4$, and $S = 2$. Then $H = 4$. A non-periodic string $l = 2200$ corresponds to $t_w = W = 8$ equivalent rotations of wire arrangements, demonstrated as follows:

××○××○○○, ○××○××○○, ○○××○××○, ○○○××○××,
×○○○××○×, ××○○○××○, ○××○○○××, ×○××○○○×.

However the periodic string $l = 2020$ corresponds to only $t_w = s_1 + s_2 + t = 2 + 0 + 2 = 4$ equivalent rotations of wire arrangements:

××○○××○○, ○××○○××○, ○○××○○××, ×○○××○○×.

Clearly, with the introduction of the notion of the string as explained above has reduced the presented problem to finding all possible strings l with *non-intersecting* sets of equivalent rotations of wire arrangements.

F. Partitioning of the Number of Faulty Wires and Corresponding Necklaces

We use *integer partitions* in order to find all string representations of all wire arrangements.

Definition 4: A *k-partition* p of a positive integer n is a partition consisting of exactly k terms, adding zeros whenever necessary.

Returning to the presented problem for a parallel link arrangement of W wires, with F faulty wires and a largest fault segment S, an H-*partition* p of (integer) F consists of H (number of healthy wires) terms, with the largest term being equal to S (refer to Example 6).

Example 6: Let $W = 18$, $F = 12$ and $S = 3$. Then $H = 6$. The 6-partitions of $F = 12$, with the largest term being equal to $S = 3$, are given as follows:

$$3+3+3+3+0+0,\ 3+3+3+2+1+0,$$
$$3+3+3+1+1+1,\ 3+3+2+2+2+0,$$
$$3+3+2+2+1+1,\ 3+2+2+2+2+1.$$

Clearly there is no 1-1 correspondence between the 6-partitions above with the strings l of the wire arrangements. For instance $p = 3 + 3 + 3 + 3 + 0 + 0$ corresponds to a set of strings, namely 333300, 333030, 333003, and so forth. Hence, still, knowing the actual H-partitions corresponding to given W, F and S does not solve the problem, as the number of strings per partition must be found. This can be achieved by noting that the number of all possible strings l with non-intersecting sets of equivalent rotations of wire arrangements can be represented by the number of *necklaces* of each H-partition (refer to Example 7 for illustration). We recall the definition of a necklace as follows.

Definition 5: A *k-ary necklace* of length n is an equivalence class of n-character strings over an alphabet of size *k*, taking all rotations as equivalent [11].

Example 7: Let $W = 6$, $F = 3$ and $S = 2$. Then $H = 3$. It turns out that there is only one 3-partition of $F = 3$, with the largest term being equal to $S = 2$, namely

$$2 + 1 + 0.$$

All necklaces for the 3-partition above, with the corresponding equivalent rotations of wire arrangements, are

$$210: \quad \times\times\circ\times\circ\circ,\ \circ\times\times\circ\times\circ,\ \circ\circ\times\times\circ\times,$$
$$\times\circ\circ\times\times\circ,\ \circ\times\circ\circ\times\times,\ \times\circ\times\circ\circ\times.$$
$$201: \quad \times\times\circ\circ\times\circ,\ \circ\times\times\circ\circ\times,\ \times\circ\times\times\circ\circ,$$
$$\circ\times\circ\times\times\circ,\ \circ\circ\times\circ\times\times,\ \times\circ\circ\times\circ\times.$$

Note that there is a 1-1 correspondence between the necklaces above and (all possible, for this case) strings l, whose corresponding sets of equivalent rotations of wire arrangements do not intersect.

For each H-partition one can compute the corresponding number of necklaces (refer to (11)), which in turn can be used to compute the number of wire arrangements. Hence, the problem reduces to finding: (a) all such H-partitions, as described above, and subsequently, (b) their corresponding number of necklaces.

1) H-partitions

There are a number of known algorithms that can actually generate such a list of H-partitions in a *constant amortized time* [12]. Note here that a partition can be extended to an H-partition by simply adding the necessary number of zeros.

However, we use an alternative and simpler approach. We define a string $r = n_0 n_1 n_2 ... n_S$, where n_j is equal to the number of occurrences of integer j in a string l. Then, from the way the strings l and r are constructed and from (4) and (5), we set the following system of equations. This system has a limited number of solutions for n_j, with each solution giving a distinct string r.

$$\begin{cases} \sum_{j=1}^{S}(S+1-j)\cdot n_j = F, \\ \sum_{j=1}^{S} n_j \in [\lceil F/S \rceil, \min(F-S+1,H)], \\ n_0 + \sum_{j=1}^{S} n_j = H, \\ n_j \in Z_0^+, \quad n_1 \neq 0. \end{cases} \quad (10)$$

2) Necklaces

Using the substring $n_0 n_1 n_2 ... n_{k-1} \subset r$, where $n_j \neq 0$ and k is the number of non-zero terms in string r, we can find the number of necklaces of an initial string l as follows.

$$N(n_0, n_1, ..., n_{k-1}) = \quad (11)$$
$$\frac{1}{n} \sum_{j|\gcd(n_0,n_1,...,n_{k-1})} \varphi(j)\left(\frac{(n/j)!}{(n_0/j)!(n_1/j)!...(n_{k-1}/j)!}\right),$$

where $N(n_0, n_1, ..., n_{k-1})$ denotes the number of necklaces composed of n_j occurrences of $j = 0,1,...,k-1$, $n = \sum_{j=0}^{k-1} n_j = H$ and $\varphi(j) = j\prod_{q|j}\left(1-\frac{1}{q}\right)$ is the *Euler's Totient Function*, defined as a number of positive integers less than or equal to j, that are coprime to j [13, 14].

For computational purposes (refer to Example 5), it is necessary to separate *periodic* and *aperiodic* necklaces. To calculate the periodic cases one needs to find the prime common factors of all elements in a string r, which are no more than the frequencies f of string l (refer to Definition 2). Once we know the frequencies of each periodic necklace we divide each of the elements in the corresponding string r by frequency f, with the result corresponding to the repeating substring l_t (refer to Definition 2). Consequently, the number of periodic necklaces is given by

$$K(n_0, n_1, ..., n_{k-1}) = \sum_j N_j\left(\frac{n_0}{f_j}, \frac{n_1}{f_j}, ..., \frac{n_{k-1}}{f_j}\right), \quad (12)$$

where j runs through all frequencies f of string l.

G. Full Model for the Probability Distribution

To complete the construction of the model describing the probability distribution $P_W(S \mid F)$ we first establish a conjecture based on empirical data.

Conjecture: The number of periodic wire arrangements is given by $\left| \bigcup_m A(\frac{W}{f_m}, \frac{F}{f_m}, S) \right|$, where f_m are the prime common factors of W and F. This can be illustrated by Example 8.

Example 8: Let $W = 12$, $F = 6$ and $S = 2$. Then $H = 6$. The prime common factors of W and F are $f_1 = 2$ and $f_2 = 3$ and the periodic wire arrangements for $A(12,6,2)$ are the following:

$t_{w1} = 6$, $f_1 = 2$:   ××o×oo××o×oo, ××oo×o××oo×o,
o××o×oo××o×o, o××oo×o××oo×,
oo××o×oo××o×, ×o××oo×o××oo,
×oo××o×oo××o, o×o××oo×o××o,
o×oo××o×oo××, oo×o××oo×o××,
×o×oo××o×oo×, ×oo×o××oo×o×,

$t_{w2} = 4$, $f_2 = 3$:   ××oo××oo××oo, o××oo××oo××o,
oo××oo××oo××, ×oo××oo××oo×.

The total number of wire arrangements is

$$\left| \bigcup_m A(\frac{W}{f_m}, \frac{F}{f_m}, S) \right| = \left| A(\frac{W}{f_1}, \frac{F}{f_1}, S) \cup A(\frac{W}{f_2}, \frac{F}{f_2}, S) \right| = 16,$$

as $A(W/f_1, F/f_1, S) = A(6,3,2)$ and $A(W/f_1, F/f_1, S) = A(4,2,2)$ are disjoint with cardinalities 12 and 4 respectively (these values were pre-calculated specifically for this example).

Now it is not difficult to derive the following recursive equation that yields the desired number of wire arrangements (once the number of H-partitions; strings, i.e. necklaces, and frequencies are known):

$$|A(W,F,S)| = W(\sum_i N_i(n_0, n_1, ..., n_{k-1}) \tag{13}$$

$$- \sum_j N_j(\frac{n_0}{f_j}, \frac{n_1}{f_j}, ..., \frac{n_{k-1}}{f_j})) + \left| \bigcup_m A(\frac{W}{f_m}, \frac{F}{f_m}, S) \right|,$$

where i denotes the index for each of the necklaces corresponding to the H-partitions of F, j the index of the prime common factors of all elements in string r (corresponding to a periodic necklace), and m the index of the prime common factors of W and F.

Finally, combining (1), (2), and (13), one can obtain the following formula for the probability distribution $P_W(S \mid F)$.

$$P_W(S \mid F) = W \binom{W-S-2}{F-S} \Big/ \binom{W}{F} \quad (\text{if } \lceil \frac{F+1}{2} \rceil \leq S \leq F),$$

$$\tag{14}$$

$$P_W(S \mid F) = |A(W,F,S)| \Big/ \binom{W}{F} \quad (\text{if } 0 < S < \lceil \frac{F+1}{2} \rceil)$$

The formula above has been numerically tested and is verified by the results obtained from a *brute-force algorithm* implemented in Matlab. The brute-force algorithm calculates wire combinations in a straight-forward manner and spends considerable time (in terms of days) to count combinations for $W = 40$ and $0 \leq F \leq 23$, running on a workstation consisting of an Intel Xeon X5560 Quad Core CPU at 2.80 GHz and 12 GB of DDR3 random access memory.

TABLE I.
RESULTS OBTAINED USING THE SET OF EQUATIONS IN (14) FOR $W = 16$ SHOWING THE NUMBER OF ALL POSSIBLE FAULTY WIRE ARRANGEMENTS.

	S=1	S=2	S=3	S=4	S=5	S=6	S=7	S=8	S=9	S=10	S=11	S=12	S=13	S=14	S=15	S=16
F=1	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F=2	104	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F=3	352	192	16	0	0	0	0	0	0	0	0	0	0	0	0	0
F=4	660	968	176	16	0	0	0	0	0	0	0	0	0	0	0	0
F=5	672	2640	880	160	16	0	0	0	0	0	0	0	0	0	0	0
F=6	336	4224	2568	720	144	16	0	0	0	0	0	0	0	0	0	0
F=7	64	4032	4704	1920	576	128	16	0	0	0	0	0	0	0	0	0
F=8	2	2212	5432	3304	1344	448	112	16	0	0	0	0	0	0	0	0
F=9	0	608	3776	3696	2016	896	336	96	16	0	0	0	0	0	0	0
F=10	0	56	1400	2560	1976	1120	560	240	80	16	0	0	0	0	0	0
F=11	0	0	208	1184	896	560	320	160	64	16	16	0	0	0	0	0
F=12	0	0	4	136	360	424	336	240	160	96	48	16	0	0	0	0
F=13	0	0	0	0	32	80	112	96	80	64	48	32	16	0	0	0
F=14	0	0	0	0	0	0	8	16	16	16	16	16	16	16	0	0
F=15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	0
F=16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

IV. NUMERICALLY VERIFIED ANALYTICAL RESULTS

THE following results for a parallel NoC link with $W = 16$ wires were obtained with the help of Matlab using our spatial faulty wire probability distribution model from (13) and (14). Again, the results were verified to be correct when compared against the results of extensive brute-force simulations in Matlab. Table 1 shows the number of wire arrangements for all possible combinations of F and S, while Fig. 2 shows the distribution of the corresponding probabilities $P_W(S \mid F)$.
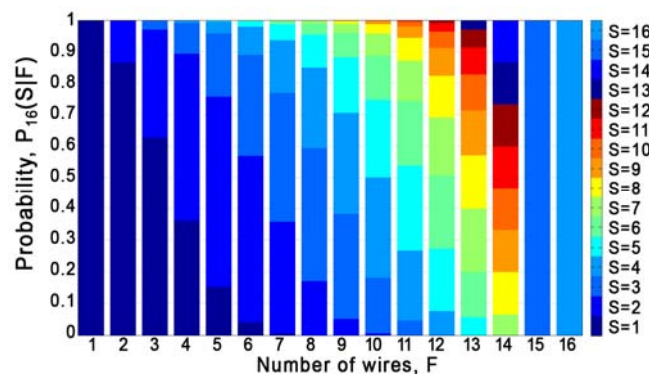

Fig. 2. Probability distribution $P_W(S/F)$ for a parallel NoC link with a total of $W = 16$ parallel wires including F faulty wires, where S is the size of the largest fault-segment.

V. AN ARITHMETIC EXAMPLE AND DEMONSTRATION OF THE EFFECTIVENESS OF THE DERIVED MODEL

WE demonstrate the applicability and, consequently, the effectiveness of the derived model using the following parameters that were chosen at random (relatively large numbers have been picked to show both the efficiency and accuracy of the derived model).

Let $W = 18$, $F = 12$ and $S = 3$. Hence $H = 6$. The 6-partitions of $F = 12$, with the largest term being equal to

$S = 3$ are given as follows with the corresponding string r (refer to Section III.*F.1*):

$$p = 3 + 3 + 3 + 3 + 0 + 0 \sim (4, 0, 0, 2) = r,$$
$$p = 3 + 3 + 3 + 2 + 1 + 0 \sim (3, 1, 1, 1) = r,$$
$$p = 3 + 3 + 3 + 1 + 1 + 1 \sim (3, 0, 3, 0) = r,$$
$$p = 3 + 3 + 2 + 2 + 2 + 0 \sim (2, 3, 0, 1) = r,$$
$$p = 3 + 3 + 2 + 2 + 1 + 1 \sim (2, 2, 2, 0) = r,$$
$$p = 3 + 2 + 2 + 2 + 2 + 1 \sim (1, 4, 1, 0) = r.$$

Then, the number of necklaces for each of the partitions above is (refer to (11)):

$$N(4,2) = 3, \quad N(3,1,1,1) = 20, \ N(3,3) = 4,$$
$$N(2,3,1) = 10, \quad N(2,2,2) = 16, \ N(1,4,1) = 5,$$

giving a total of 58 necklaces. Moreover, the number of periodic necklaces is (refer to (12))

$$K(4,2) = N(2,1) = 1, \quad K(3,3) = N(1,1) = 1,$$
$$K(2,2,2) = N(1,1,1) = 2,$$

giving a total of 4 periodic necklaces.

Using the conjecture established in Section III.*G*, we find, recursively, the number of periodic wire arrangements

$$\left| \bigcup_m A\left(\frac{W}{f_m}, \frac{F}{f_m}, S\right) \right| = \left| A(18/2, 12/2, 3) \cup A(18/3, 12/3, 3) \right| = 33,$$

as $A(9,6,3)$ and $A(6,4,2)$ are disjoint with cardinalities 27 and 6, respectively.

The next step is to proceed with the calculation of the total number of wire arrangements using (13) as follows.

$$\left| A(18,12,3) \right| = W\left(\sum_i N_i - \sum_j K_j\right) + \left| \bigcup_m A\left(\frac{18}{f_m}, \frac{12}{f_m}, 3\right) \right|$$
$$= 18(58 \ 4) + 33 = 1005,$$

which is in complete agreement with the brute-force computation.

Finally, we compute the desired probability (from (14)) to be

$$P_{18}(3 \mid 12) = \left| A(18,12,3) \right| \bigg/ \binom{18}{12} = \frac{1005}{18564}.$$

VI. CONCLUSION

THIS article derives and demonstrates the combinatorics-based model which can be used to calculate the spatial probability distribution of wire faults in a parallel NoC interconnect link given its width and a given number of faulty wires which can appear in this link. Particular emphasis was paid upon the adjacency of the faulty wires that form *fault-segments* separated by at least one healthy wire, as the size of the largest segment determines the additional delay required by data recovery mechanisms, such as the Partially Faulty Link Recovery Mechanism (PFLRM) [9, 10], to recover corrupted flit data at the receiver router. The developed model constitutes a combinatorial application of partitions and necklaces through a systematic approach that derives the correspondence between the presented problem and necklaces, where periodicity plays a crucial role. The model proves to be highly accurate when compared against extensive brute-force numerical simulations and associated results. Next, future, steps, include the rigorous proof of the recursive conjecture established in the presented algorithm, while the derivation of a fully non-recursive algorithm persists as a challenge.

REFERENCES

[1] S.R. Vangal, J. Howard, G. Ruhl, S. Dighe, H. Wilson, J. Tschanz, D. Finan, A. Singh, T. Jacob, S. Jain, V. Erraguntla, C. Roberts, Y. Hoskote, N. Borkar, S. Borkar, "An 80-Tile Sub-100-W TeraFLOPS Processor in 65-nm CMOS," IEEE Journal of Solid-State Circuits, Vol. 43, No. 1, pp. 29–41, Jan. 2008.
[2] S. Bell, B. Edwards, J. Amann, R. Conlin, K. Joyce, V. Leung, J. MacKay, M. Reif, Liewei Bao, J. Brown, M. Mattina, Chyi-Chang Miao, C. Ramey, D. Wentzlaff, W. Anderson, E. Berger, N. Fairbanks, D. Khan, F. Montenegro, J. Stickney, J. Zook, "TILE64 Processor: A 64-Core SoC with Mesh Interconnect," Proc. of the IEEE Int'l Solid-State Circuits Conference, pp. 588–598, Feb. 2008.
[3] W.J. Dally and B. Towles, "Route Packets not Wires: On-Chip Interconnection Networks," Proc. of the IEEE Design Automation Conference, pp. 684–689, May 2001.
[4] J.R. Black, "Electromigration Failure Modes in Aluminum Metallization for Semiconductor Devices," Proc. of the IEEE, Vol. 57, No. 8, pp. 1587–1594, Sept. 1969.
[5] A. Vitkovskiy, V. Soteriou and C. Nicopoulos, "A Fine-Grained Link-Level Fault-Tolerant Mechanism for Networks-on-Chip," Proc. IEEE International Conference on Computer Design, pp. 447–454, Oct. 2010.
[6] A. Vitkovskiy, V. Soteriou and C. Nicopoulos, "A Dynamically Adjusting Gracefully Degrading Link-Level Fault-Tolerant Mechanism for NoCs," To appear in the IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2012.
[7] R.C. Lyndon, "On Burnside's Problem," Transactions of the American Mathematical Society, Vol. 77, pp. 202–215, 1954.
[8] L. Pebody, "Reconstructing Odd Necklaces," in Combinatorics, Probability and Computing, Vol. 16, pp. 503–514, Feb. 2007.
[9] N. Alon, "Splitting Necklaces," Advances in Mathematics, Vol. 63, No. 3, pp. 247–253, March 1987.
[10] S.W. Golomb, "Combinatorial Proof of Fermat's 'Little Theorem," in The American Mathematical Monthly, Vol. 63, No 10, pp. 718, 1956
[11] E.W. Weisstein, "Necklace," MathWorld--A Wolfram Web Resource, available [online]: http://mathworld.wolfram.com/Necklace.html
[12] D.E. Knuth, "The Art of Computer Programming: Vol. 4A Combinatorial Algorithms, Part 1," Upper Saddle River, New Jersey: Addison-Wesley, 2011, pp xvi+883.
[13] The Object Server, "Necklaces, Unlabelled Necklaces, Lyndon Words, De Bruijn Sequences," Available [online]: http://www.theory.cs.uvic.ca/~cos/inf/neck/NecklaceInfo.html
[14] E.W. Weisstein, "Totient Function," MathWorld--A Wolfram Web Resource, available [online]: http://mathworld.wolfram.com/TotientFunction.html.