On Motifs and Functional Blocks in Complex Networks

Name of Author[1] and Name of Author[2]

[1] Name and Address of your Institute name@email.address
[2] Name and Address of your Institute name@email.address

This article addresses the presence and size of basic functional modules (motifs) in complex systems. The current view of motifs is that they are small modules of 3-4 “nodes”, where a node is interpreted as a primitive component, such as a primitive logic gate (AND, OR, NOT) or its correlative in other domains. For the domain of electrical circuits, we show that standard function blocks used to build real-world circuits are much larger than the proposed 3-4 node motifs, and in fact these function blocks themselves are typically comprised of a collections of several 3-4 node motifs. We argue that, for this domain, the statistical over-representation of 3-4 node motifs is due to function blocks consisting of motif-clusters, and not that motifs, in and of themselves, play any discernible functional role in a circuit.

1 Introduction

In recent years, many real-world complex networks including biological networks and engineering/technological networks have been characterized by global properties such as small world and long-tailed degree distribution [2, 4]. This has enabled the underlying topology of biological systems to readily be compared with engineering systems, which are traditionally described by networks such as flow charts and blueprints. Remarkably, when such a comparison is made, biological networks are seen to share structural principles with engineered networks [1].

One of the key thrusts of complex systems research has been the topological analysis of the network underlying a complex system. One key question is whether these structures are truly compositional, and, if so, what the underlying building blocks are. For example, biological networks have been shown to be decomposable into modular components that recur across and within given organisms: various researchers, e.g., (Lee et al. 2002; Shen-Orr et al. 2002; Zak et al. 2003), have argued that the underlying building blocks, or motifs, consist of interacting groups of between 2 and 4 genes, which control transcriptional regulation [Barkai and Leibler (2000)]. Given this evidence, (Shen-Orr et al. 2002) have proposed motifs as the basic building blocks in biological networks, and further argue that such motifs possess direct analogues in technological systems.

Studies of the most significant network motifs found in Escherichia coli (Shen-Orr et al. 2002) and in Saccharomyces cerevisiae (Lee et al. 2002) indicate that there exists a network structure governing biological regulation that consists of elementary modules or components, similar to those found in a digital circuit. It should be noted that these components work together in a sophisticated pattern of cascaded control loops, which are interconnected with other (transcriptional) control loops (doyle and stelling 2006). Hence, it is unclear whether the regulatory and transcriptional functionality is due to the motifs themselves, or to the complex interactions and structure of the motifs. In other words, it is not entirely clear what the basic structural building blocks are.

Modularity is a common property in engineering systems, such as libraries or cells used in electronic circuits [5] and subroutines in software [16]. It is also an oft-mentioned property of biological networks. For example, proteins are known to work in slightly overlapping, co-regulated groups such as pathways and complexes [1]. Alon [1] proposed a working definition of a module based on comparison with engineering: A module in a network is a set of nodes that have strong interactions and a common function. A module has defined input nodes and output nodes that control the interactions with the rest of the network. Modules in engineering, and presumably also in biology, have special features that make them easily embedded in almost any system [1].

The basic building block of the modular structure of biological and engineering networks, termed “motif”, has been defined as a subgraph that occurs significantly more frequently in real-world networks than expected by chance alone [15]. The observed over-representation of motifs has been interpreted as a manifestation of functional constraints and design principles that have shaped network architecture at the local level. Some researchers believe that motifs reflect the underlying processes that generated different networks and may have specific functions as elementary computational circuits in the networks [15]. Finding motifs in a new network may help explain what system-level function the network performs, and how it performs it.

At presently, it is not clear what determines the particular frequencies of all possible network motifs in a specific network [17]. In terms of the definitions the motifs and the functional modules are very close to each other. For the moment, we distinguish the two by emphasizing small size and recurrence for motifs, endowing modules with larger size, and perhaps a composition dominated by interconnected motifs. Wolf’s point of view on motifs and modules is that motifs - small, repeated, and conserved regulatory devices - are arranged by evolutionary or design processes into modules, which are larger, overlapping, and functionally significant subnetworks [19]. However the relation between motifs and functional blocks and their method of integration into full network has not been previously examined, so we need to investigate the real-world networks and verify the relation between motifs and functional modules or blocks. Once a dictionary of network motifs and corresponding functional modules is established, one could improve the understanding of the structure and underlying mechanism of complex networks.

In contrast with engineering networks, such as digital circuits, where design and control are explicitly engineered, our understanding of biological network design principles and of mechanisms that control the biological information is very poor. As the high-throughput assays are inherently noisy and biased in their nature, we can only get very incomplete data on biological networks. At present many of the connections, numbers and input functions in biological networks are not known, so the resulting network would still require careful interpretation to extract its underlying biological meaning. The recent research suggests that it is difficult to gain significant insights into biological function simply by considering the connection architecture of a gene network, or its decomposition into simple structural motifs [7].

Function modules can be defined for the main reaction pathways in a living cell, defined in terms of macromolecular machines with compact structure or ensembles that change their composition and/or organization during function [**?**]. These function modules can be distinguished using a number of properties, including topological distance, specificity and a characteristic time domain within which their function proceeds.

Biological systems are also characterized by their dynamical behaviour and the variability of the functional roles that the sub-structures can play. The dynamics of functional modules is critical, as all modules go through functional cycles, with phases of increasing and decreasing complexity of molecular interactions. Furthermore, each functional module can changes its function based on a number of intrinsic and extrinsic properties.

In contrast, electronic circuits are typical technology systems following well-defined engineering design principles. Though the structure of large scale circuits is also very complex, there is a great advantage of electronic circuits:

- Circuits have function blocks that have a stable function, in contrast to biological function blocks;
- Circuits have clearly-defined dynamical behaviour, and there are feedback-free circuits for which the dynamics is basically irrelevant;
- a lot of representative benchmark circuits such as ISCAS benchmark suites having complete netlists are available and convenient for our experiments and analysis [6]. Thus, electronic circuits will be a good point to begin our investigation. In particular, we focus on the class feedback-free circuits, combinational circuits.

In this paper we firstly detect motifs and extract corresponding instances in benchmark circuits [18, 15]. We develop a library of circuit function blocks through a process of reverse-engineering, and identifying function blocks defined in IC databooks, cell libraries and in text books. We then search the common functional blocks used in benchmark circuits and discover their instances [5]. Furthermore we compare topologies and instances of motifs and functional blocks. This provides the first experimental demonstration on the relation between motifs and functional blocks. The experimental results show that all motifs are subgraphs of functional blocks, and give a clear explanation for observed over-representation of motifs. We also find that the common functional blocks discovered in benchmark circuits are motifs themselves and unlikely to present in the random graphs. This result will give us more clues on modeling real-world networks. At last we analyze the significance profile (SP) of benchmark circuits [14], and find that the circuits having similar functions are highly correlated in terms of the SP. It presents an effective approach based on the SP for comparing the local structure and function of benchmark circuits.

We organize the remainder of the article as follows. Section 2 discusses related work on motifs and corresponding functions. Section 3 elaborates our work on detecting motifs and functional blocks in benchmark circuits and analyzing the relation between motifs and functional blocks. Section 4 summarizes our contributions and research work in the future.

2 Notation

We assume that the underlying topology of a system is modeled using a graph $G(V, E)$, where V is the set of vertices and E the set of edges.[3] A subgraph $G' = (V', E')$ of G is a graph such that $V' \subseteq V$ and $E' contains all edges between$V'. A *size-k subgraph* is a subgraph that is induced by a vertex set of size k.

For a given integer k, the set of all size-k subgraphs in G can be extracted into sets $S_k^i(G)$, called subgraph classes, where two size-k subgraphs belong to the same subgraph class if and only if they are isomorphic (that is, if and only if they are topologically equivalent). Following [18], we define the *subgraph-frequency* as follows:

Definition 1 (Subgraph-frequency). *The* subgraph-frequency $C_k^i(G)$ *of a subgraph class* $S_k^i(G)$ *is defined as*

$$C_k^i(G) = |S_k^i(G)| \cdot \left(\sum_j |S_k^j(G)| \right)^{-1} .$$

We call $\mathcal{G}_R$ the class of generalized random graphs keeping the degree sequence of G [15, 8, 12, 9]. We call $C_k^i(\mathcal{G}_R)$ the subgraph-frequency of a subgraph class $S_k^i(G)$ within the class $\mathcal{G}_R$ of generalized random graphs.

Definition 2 (Motif). *A motif of a graph* G *is a subgraph class* $S_k^i(G)$ *such that* $C_k^i(G) \gg C_k^i(\mathcal{G}_R)$.

[3] We assume that all graphs we consider will be connected.

This definition of motif focuses on the rate of occurrence of the subgraph and is independent of the function of the subnetwork. In contrast, we can define a function block in terms of performing a key function within the overall system. To capture this notion, we introduce an abstract function $\phi(H, \mathcal{I})$ that a graph H will perform in a system given inputs $\mathcal{I}$. For example, in a circuit where H corresponds to a NOT gate, $\phi(H, \mathcal{I})$ just evaluates the boolean function $\neg\mathcal{I}$.

We introduce a first basic notion of function block.

Definition 3 (Function block). *A function block of a graph G is a subgraph class $S_k^i(G)$ such that function $\phi(S_k^i(G), \mathcal{I})$ is well-defined and all vertices $V \in S_k^i(G)$ participate in ϕ.*

3 Related Work

A number of publications have recently examined the occurrence and functions of the feed-forward loop motif in a variety of networks [7, 13]. They just focus on the dynamics of the motif alone and don't involve high-level functional blocks because our understanding of biological networks is not yet enough for the reverse-engineering. Additionally, it must be recalled that these motifs do not exist in isolation within the network, and their behavior will be heavily influenced by both global and local environment and the state of the network as a whole. These considerations alone may make attempts to draw positive conclusions about how a motif will behave overly optimistic Ingram looked in some detail at dynamics of the bi-fan motif and found that even with this relatively simple model, the bi-fan motif can exhibit a wide range of dynamical responses. This suggests that it is difficult to gain significant insights into biological function simply by considering the connection architecture of a biological network, or its decomposition into simple structural motifs [7].

It is of interest to understand how network motifs combine to form larger structures. Kashtan presents a systematic approach to define 'motif generalizations': families of motifs of different sizes that share a common architectural theme [11]. To define motif generalizations, he defines 'roles' in a subgraph according to structural equivalence. For example, the feed forward loop triad, a motif in transcription, neuronal and some electronic networks, has three roles, an input node, an output node and an internal node. The roles are used to define possible generalizations of the motif. The feed forward loop can have three simple generalizations, based on replicating each of the three roles and their connections. He presents algorithms for efficiently detecting motif generalizations and finds networks which share a common motif can have very different generalizations of that motif. Though this allows generalizing from small motifs to the larger complexes in which they appear, but a generalized motif is only a combination of one type of motifs and obviously the generalized motif is different from a functional block. And in experiments the author only analyzes generalizations of the feed-forward loop motif.

Itzkovitz proposes a repeated pattern discovery approach based on network motifs for simplifying networks by creating coarse-grained networks in which each node is a repeated pattern in the original network [10]. He defines coarse-graining units (CGU), which can be used as nodes in a coarse-grained version of the network. He demonstrates this approach by coarse-graining a small ISCAS89 benchmark electronic circuit which is given as a netlist of five gate types (AND, OR, NAND, NOR, NOT) and a D-flipflop (DFF). He first replaced every DFF occurrence with a standard implementation using four NAND gates and one NOT gate. All gates were then replaced with their standard transistor-transistor logic (TTL) implementation. The CGUs were detected in transistor network to recover the logic gates, and then applied coarse-graining process at the gate-level network to detect the DFF. At the last step he continued to detect higher-level CGUs at the gate-DFF-level and tried to discover the functional blocks by coarse-graining. But CGUs detected in the last step can only be called repeated patterns rather than real functional blocks. Since functional blocks should provide significant common functions easily embedded in almost any system and usually can be found in the standard libraries. And actually the detected CGUs are different from the functional blocks in the high-level analysis by electronic engineers. The corresponding repeated pattern discovery has rather high computing complexity and can be only applied on very small circuits. It is claimed that coarse-graining seeks a small dictionary of simple subgraph types in order to help understand the function of the network in terms of recurring independent building blocks without prior knowledge of library of functional blocks used in the design, but it is very hard to achieve in practical applications.

4 Analysis on motifs and functional blocks in digital circuits

Over the years, there have been many attempts to create and use neutral benchmark circuits for evaluation of different tools and algorithms. These benchmark sets are surrogate circuits chosen to represent the kinds of problems a tool will encounter in real use [6]. The widely accepted ISCAS-85 benchmark suite has been in use ever since being introduced in netlists of fundamental logic gates at the International Symposium of Circuits and Systems in 1985. The circuits are industrial designs whose functions and high-level designs have not been published, both for confidentiality reasons and to allow them to be viewed as random logic circuits with no significant high-level structure. But recently in researching test generation techniques, some researchers found that, in fact, the ISCAS-85 circuits have well-defined, high-level structures and functions based on common functional blocks such as multiplexers, adders, carry-look-aheads (CLA) and decoders. They reverse-engineered the benchmarks, starting with the original gate-level netlists and systematically recover the circuits' hidden functional and structural information [5]. The discovered high-level structures in ISCAS85 benchmark circuits provide an excellent opportunity to study the relation between motifs and functional modules.

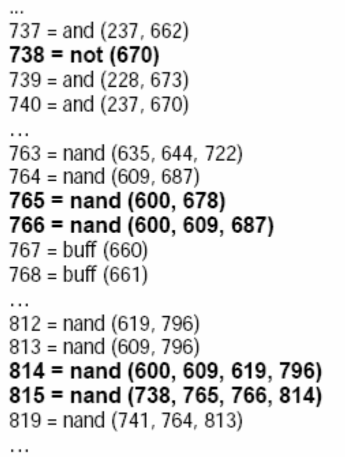
A network motif in the sense introduced by Milo is a pattern or small subgraph that occurs more often (at some statistically significant level) in the true network than in an ensemble of networks generated by randomly rewiring the edges in the true network, where the number of nodes and the degree of each node is kept fixed [15]. Of interest are the differences in the frequencies with which network motifs occur in real-world (biological as well as technological) networks. The recurrent presence of certain motifs has been linked to systematic differences in the functional properties required from networks, and it has been suggested that the motifs in networks reflect functional or computational units which combine to regulate network behavior as a whole. Electronic circuits can be viewed as networks in which nodes (or vertices) are electronic components (e.g. logic gates in digital circuits and resistors, capacitors, diodes and so on in analogic circuits) and edges (or connections) are wires in a broad sense [3]. Here, we address the question of whether a given network motif appears independently in the network or whether instances of the motif combine to form larger structures. If the latter occurs, what is the function of these larger structures? Do different networks that share a certain network motif also share the same structural combinations of that motif? These questions require analysis of large subgraphs, a computationally difficult problem.

Figure 1 shows part of the netlist defining one of ISCAS85 benchmark circuit - the C880[5]. Each line in the netlist describe a logic gate and its inputs, such as "765=nand(600,678)" show a nand gate which is assigned the ID 765 and has two inputs with the ID 600 and 678 respectively.

We treat the inputs in circuits as a special type of components, and convert the netlists of benchmarks into corresponding graphs. The netlist is presented in the logic gate level, so the number of nodes of the graph is the sum of the number of gates and the number of inputs. The table 1 shows properties of ISCAS85 circuits.

4.1 Detection of network motifs in electronic circuits

At first we detected 3-node and 4-node motifs in 10 circuits of ISCAS85 benchmark suite (Since the C17 has only 7 gates, we didn't run experiments on it). We applied a strict rule for filtering candidate subgraphs, which requires that frequencies of subgraphs are much higher than that in random graphs besides high Z-scores, and we think it can produce more reliable and convincing results. We found only one type of 3-node motif (Feed-forward loop) which appears in only 3 circuits, and two 4-node motifs (Bi-fan and Bi-parallel) are discovered in most circuits. The description of the motifs is listed in the table 2. The ID of a motif is a decimal number generated from its adjacent matrix which can be treated as a binary number. For example the adjacent matrix



```
...
737 = and (237, 662)
738 = not (670)
739 = and (228, 673)
740 = and (237, 670)
...
763 = nand (635, 644, 722)
764 = nand (609, 687)
765 = nand (600, 678)
766 = nand (600, 609, 687)
767 = buff (660)
768 = buff (661)
...
812 = nand (619, 796)
813 = nand (609, 796)
814 = nand (600, 609, 619, 796)
815 = nand (738, 765, 766, 814)
819 = nand (741, 764, 813)
...
```

Fig. 1. A portion of the ISCAS85 C880 benchmark netlist

Table 1. ISCAS85 benchmark circuits

circuit	input number	gate number	node number	edge number	clustering coefficient
C17	5	6	11	12	0
C432	36	160	196	336	0.003118
C499	41	202	243	408	0
C880	60	383	443	729	0
C1355	41	546	587	1064	0.217149
C1908	33	880	913	1497	0
C2670	233	1117	1350	2075	0.006926
C3540	50	1669	1719	2936	0.000892
C5315	178	2307	2485	4386	0.002067
C6288	32	2416	2448	4800	0.231808
C7552	207	3511	3718	6144	0.000241

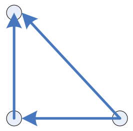
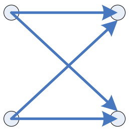
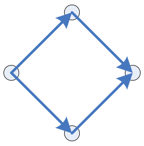
of the feed-forward loop can be converted into a binary number '000100110' which is equal to 38.

The above three motifs are also detected in several forward logic circuits in the ISCAS89 benchmark suite by Milo [15]. The detailed statistics of motifs in ISCAS85 are listed in Table 3, Table 4 and Table 5. For each motif or subgraph i, the statistical significance is described by the $Z - score$:

$$Z_i = (Nreal_i - < Nrand_i >)/std(Nrand_i)$$

Where $Nreal_i$ is the number of times the subgraph appears in the circuit, and $< Nrand_i >$ and $std(Nrand_i)$ are the mean and standard deviation of its appearance in the randomized network ensemble [15].

Table 2. Motifs detected in ISCAS85 benchmark circuits

ID	topology	Adjacent Matrix	Name	Circuit Number
38		000 100 110	feed-forward loop	3
204		0000 0000 1100 1100	bi-fan	8
2182		0000 1000 1000 0110	bi-fan	8

The $p - Value$ of a subgraph is the proportion of randomized graphs which have more instances of the subgraph than the real circuit. Obviously for a motif a less $p - Value$ is better.

Table 3. Statistics of feed-forward loop motifs in ISCAS85 benchmark circuits

Circuit	Frequency [real]	Frequency [random]	Standard-Dev [random]	Z-Score	p-Value
C1355	6.2575%	0.12839%	0.00059554	102.92	0
C2670	0.20253%	0.047177%	0.00027097	5.7333	0
C6288	6.1975%	0.027431%	0.00012762	483.46	0

Table 4. Statistics of bi-fan motifs in ISCAS85 benchmark circuits

Circuit	Frequency [real]	Frequency [random]	Standard-Dev [random]	Z-Score	p-Value
C432	0.28197%	0.047422%	0.00030755	7.6263	0
C880	1.2167%	0.015106%	0.0001347	89.208	0
C1355	0.05385%	0.010872%	7.8586e-005	5.4689	0
C1908	3.0596%	0.0078224%	5.9881e-005	509.64	0
C2670	0.87064%	0.0036917%	4.0357e-005	214.82	0
C3540	0.66965%	0.0051092%	3.2887e-005	202.07	0
C5315	1.2751%	0.0049267%	2.43e-005	522.68	0
C7552	1.4738%	0.0013707%	1.22e-005	1207	0

Table 5. Statistics of bi-fan motifs in ISCAS85 benchmark circuits

Circuit	Frequency [real]	Frequency [random]	Standard-Dev [random]	Z-Score	p-Value
C880	0.8709%	0.026338%	0.00017817	47.401	0
C1355	0.70005%	0.021553%	0.00011125	60.989	0
C1908	1.1727%	0.014336%	7.3778e-005	157.01	0
C2670	0.55909%	0.0083733%	5.8272e-005	94.507	0
C3540	0.23039%	0.0075035%	3.8007e-005	58.644	0
C5315	0.51197%	0.0057288%	2.7175e-005	186.29	0
C6288	0.68145%	0.0046884%	2.3736e-005	285.12	0
C7552	0.50259%	0.0033538%	1.9274e-005	259.01	0

Actually there are several other 4-node motifs with high Z-scores in C1355, C2670 and C6288. These motifs are listed in Figure 2. Obviously these 4-node motifs are directly derived from the 3-node motif feed-forward loop by adding an additional edge. We want to ensure that a high significance was not assigned to a pattern only because it has a highly significant subpattern [15], so these motifs are not listed in Table 3, Table 4 and Table 5.

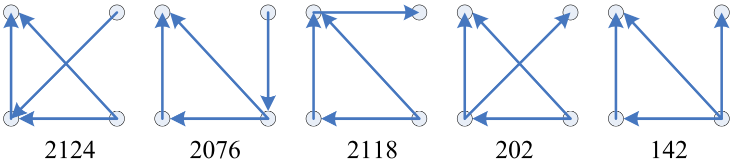


Fig. 2. Other 4-node motifs and corresponding IDs

4.2 Analysis of Functional Blocks in ISCAS85 circuits

We need to reverse-engineer netlists of the ISCAS85 circuits in order to discover what common functional blocks they may contain. Fortunately Hansen has done a good job on reverse-engineering and identifying high-level functional blocks in ISCAS85 [5]. His research showed that ISCAS85 circuits have well-defined structures and functions based on common small functional blocks such as multiplexers, adders, carry generators and decoders. If we find these common functional blocks and corresponding instances in ISCAS85 circuits, then we can compare them with detected motifs. Though Hansen's work provided valuable guidelines by summarizing high-level models of each circuit, not all implementations of common functional blocks are illustrated in details. So we have to search the functional blocks and verify their implementation. In general, circuit designers tend to follow published or textbook designs, and this is a boost for our work. We find the gate-level implementations or schematic diagrams of functional blocks in IC databooks or standard cell libraries and in textbooks, e.g.[**?**]. The functional blocks usually exist in variants due to differences in input size and gate types, so we developed a tool to automate small functional block recognition. This tool, which implements a simple subgraph-matching method, searches certain subcircuits corresponding to a given functional block. In following part we list the most widely-used small functional blocks in ISCAS85 circuits.

Figure 3 displays a XOR functional block implemented by 4 NAND gates, which is frequently used in C1355. Its topological structure contains two 3-node feed-forward loops and one bi-parallel structure, and that's why the feed-forward loop and the bi-parallel are the two most significant motifs in C1355.

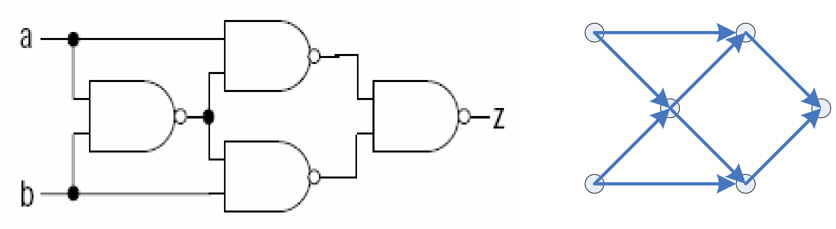


Fig. 3. A 2-input XOR functional block implemented by 4 NAND gates and the corresponding topology

The functional block in Figure 4 is a 2-to-1 multiplexer widely used in most of the ISCAS85 circuits.

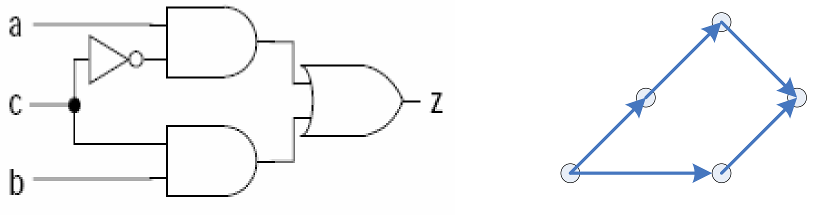


Fig. 4. A 2-to-1-multiplexer and the corresponding topology

C6288 consists of 240 full and half adder cells arranged in a 15x16 matrix. Figure 5 is a full adder cell functional block used in C6288. There are several types of half adders in C6288. The 15 top-row half adders lack the C_i input; each has two inverters at locations V. The single half adder in the bottom row lacks the B input, thereby acquiring two inverters at locations W. Obviously the topologies of both full adders and half adders contain abundant 3-node feed-forward loop and 4-node bi-parallel structures. Since the full and half adders are the only types of functional blocks in C6288, there is no bi-fan motif in C6288.

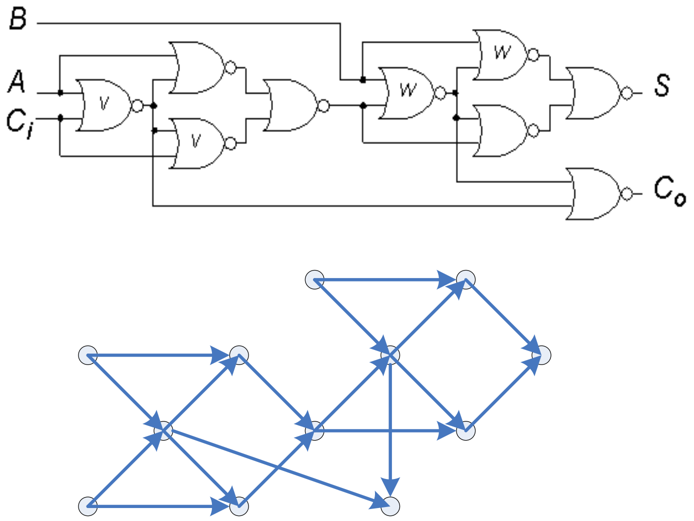


Fig. 5. A adder used in C6288 and the corresponding topology

It is interesting that we found another implementation of the half and full adders in Figure 6 from many text books [**?**]. The half adder itself appears as a bi-fan motif and the full adder contains two bi-fan motif structures. They are widely-used functional blocks in real circuits as well, although they don't appear in ISCAS85 circuits.

According to reverse-engineering analysis, many ISCAS85 circuits implement the function of the ALU or adder, so the carry look-ahead (CLA) generator is one of the most important common functional blocks in ISCAS85 circuits.

The highlighted block in Figure 7 shows a carry generator for the third bit of CLA in C880. Since the input of the NOT gate has no direct connection to other gates in the block, we can ignore the NOT gate and display the topology without it. There are four bi-parallel and one bi-fan structures in this block. The same CLA generator functional blocks also appear in C3540, and there is a different implementation of CLA generators in C7552, C5315 and C2670. Figure 8 shows a CLA generator block used in C2670, and the

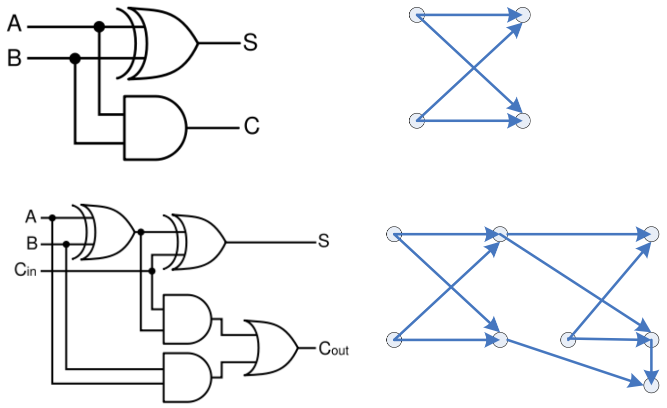


Fig. 6. The adders in textbooks and the corresponding topologies

highlight block is also a carry generator for the third bit. Though this carry generator uses gates with different types, its topology is the same as that of C880. We also can see that the block corresponding to the second bit in CLA present a bi-parallel structure.

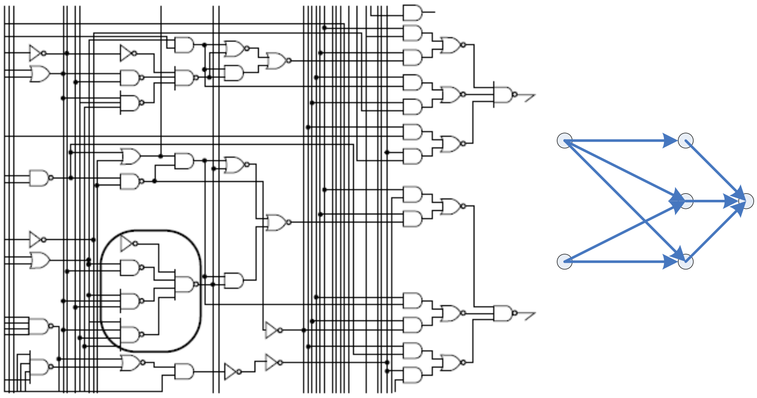


Fig. 7. A carry generator for the third bit of a CLA in C880 and the corresponding topology

There are also some other common functional blocks such as the 4-to-1 multiplexer and decoder in ISCAS85. Figure 9(a) shows a 4-to-1-multiplexer used in C2670 which also contains abundant bi-fan and bi-parallel structures.

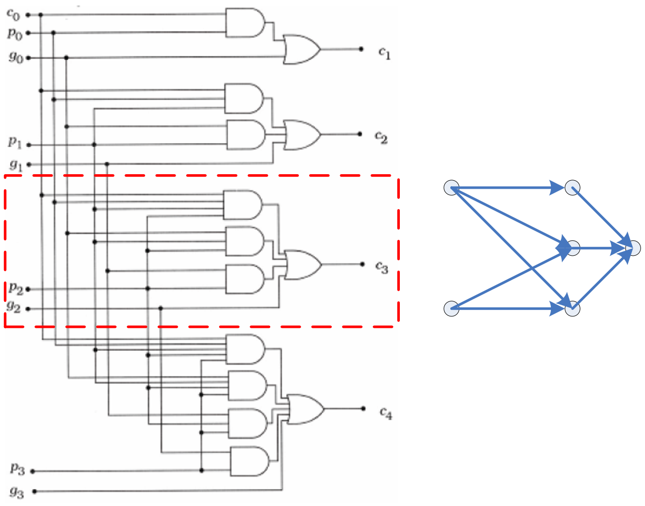


Fig. 8. A carry generator for the third bit of a CLA in C2670 and the corresponding topology

In Figure 9(b) there is an example of the 2-to-4 decoder. We can find a lot of bi-fan and bi-parallel structures in them as well. The topological structure of the 2-to-1-multiplexer can also be found in the 4-to-1-multiplexer, since the 4-to-1-multiplexer is extended from the 2-to-1-multiplexer.

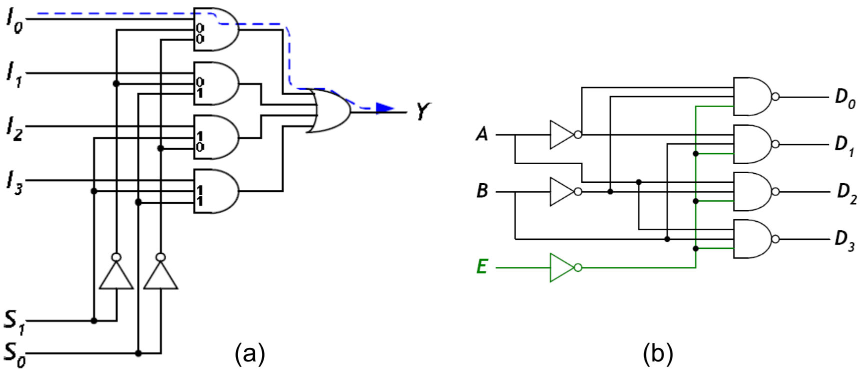


Fig. 9. A 4-to-1 multiplexer (a) and a 2-to-4-decoder (b)

We discover the feed-forward loop, bi-fan and bi-parallel motifs in the ISCAS85 circuits because they are contained in common functional blocks and combined to form the main structure of these functional blocks.

The 2-to-1 multiplexer is a 5-node common functional block used in almost every circuit. We only didn't find any 2-to-1 multiplexer in C432 and C6288, and in some circuits this functional block has a couple of variant implementations, such as replacing a 2-input OR by a 2-input NOR and replacing a 2-input AND by a 2-input NAND. Actually we found the structure of 2-to-1 multiplexer itself present as a motif. In Table 6 we compared its occurrence in ISCAS85 with that in corresponding random graphs.

Table 6. Statistics of 2-to-1 multiplexers in ISCAS85 benchmark circuits

Circuit	Frequency [real]	Frequency [random]	Standard-Dev [random]	Z-Score	p-Value
C499	0.034862%	0.014656%	3.9158e-005	5.1602	0
C880	0.037032%	0.015961%	6.6586e-005	3.1646	0.003
C1355	0.027103%	0.012835%	3.6355e-005	3.9247	0.001
C1908	0.087428%	0.0070623%	2.0987e-005	38.292	0
C2670	0.35125%	0.0051655%	2.175e-005	159.12	0
C3540	0.10784%	0.0037717%	1.1234e-005	92.639	0
C5315	0.1908%	0.0027487%	7.4177e-006	253.52	0
C7552	0.18692%	0.0020339%	6.5669e-006	281.54	0

Furthermore we investigated the carry generator for the third bit of a CLA, and Table 7 shows the results.

Table 7. Statistics of the carry generators for the third bit of CLAs in ISCAS85 benchmark circuits

Circuit	Frequency [real]	Frequency [random]	Standard-Dev [random]	Z-Score	p-Value
C880	0.0013813%	0%	0	undefined	0
C2670	0.0024862%	0%	0	undefined	0
C3540	0.00021153%	0%	0	undefined	0
C5315	0.0021738%	0%	0	undefined	0
C7552	0.0063876%	0%	0	undefined	0

Similarly we investigate the 6-node structure corresponding to the implementation of the XOR functional block in Figure 3. This structure is also presented in the full and half adders in C6288 with the different function.

Experimental results show that 3-node and 4-node motifs are subgraphs in common functional blocks used in ISCAS85 circuits, and these functional blocks themselves are also motifs. The random graph generators keeping the same degree distribution are unlikely to even produce one instance of small common functional blocks having only 6 nodes. An engineering system with limited number of the common functional blocks can is easy to be maintained.

Table 8. Statistics of the 6-node structure in Figure 3 in ISCAS85 benchmark circuits

Circuit	Frequency [real]	Frequency [random]	Standard-Dev [random]	Z-Score	p-Value
C1355	0.016731%	0%	0	undefined	0
C6288	0.013197%	0%	0	undefined	0

Similarly a gene circuit must be robust to such perturbations imposes severe constraints on its design: only a small percentage of the possible circuits that perform a given function can perform it robustly [1]. Both engineering systems and biological systems share the similar principles.

Li also demonstrated that in the context of such complex and highly engineered systems as the Internet, it is largely impossible to understand any nontrivial network properties while ignoring all domain-specific details such as technological constraints and function demand, as is typical in methods inspired by statistical mechanics [?]. Her experimental results show that the networks with high performance in real-world Internet are highly unlikely to generate by probabilistic approaches like random rewiring or preferential attachment and provide macroscopic evidence to display that the networks resulting from a careful design process are vanishingly rare from a conventional probabilistic graph point of view. Our microscopic analysis is consistent with her macroscopic analysis.

4.3 Analysis of the Significance Profile in ISCAS85 circuits

Milo presented an approach for comparing network local structure based on the so called significance profile (SP). The SP is the vector of Z-scores normalized to length 1:

$$SP_i = Z_i/(\sum Z_i^2)^{1/2}$$

The normalization emphasizes the relative significance of subgraphs, rather than the absolute significance. This is important for comparison of networks of different sizes, because motifs in large networks tend to display higher Z-scores than motifs in small networks [14].

Based on the SP approach, we can compare structures of the ISCAS85 circuits more fully. Firstly we extract all 17 subgraphs appearing in ISCAS85 circuits and assign each of them a number from 1 to 17. The following Table 9 show these 17 subgraphs with their motif IDs. The No.5 and No. 10 are the bi-fan motif and the bi-parallel motif respectively.

Table 9. The list of detected 4-node motifs in ISCAS85 circuits

NO.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Motif ID	14	140	142	202	204	2076	2116	2118	2124	2182	2184	2188	28	392	536	652	74

We present in Figure 10 the SP of 17 detected 4-node motifs for ISCAS85 circuits.

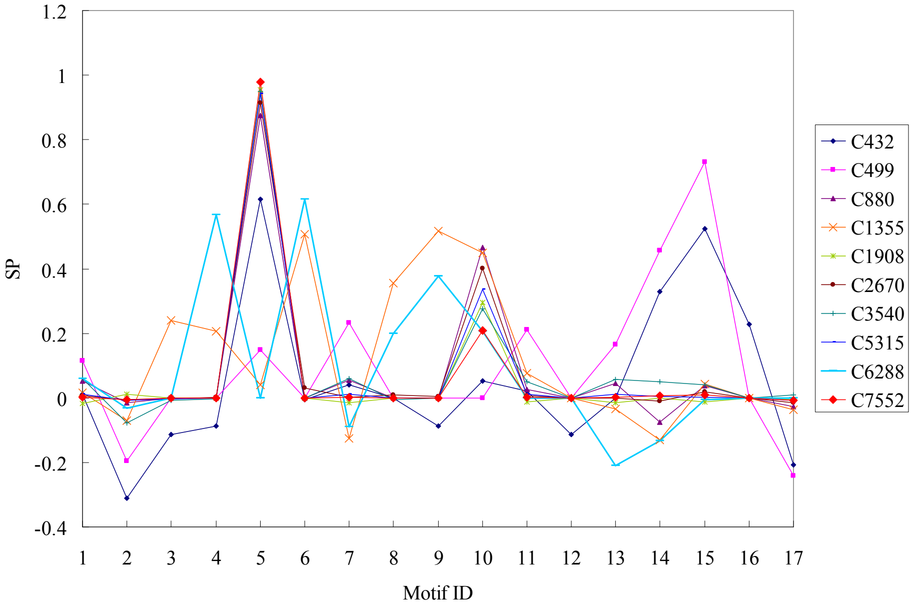


Fig. 10. The 4-node SP of ISCAS85 circuits

It is very interesting that the SP approach can not only display local structures of circuits intuitively but also distinguish functions of circuits. For example, the C880, C2670, C3540, and C5315 are ALUs and C7552 is Adder/Comparator. The main functions of these 5 circuits are similar and related to the fundamental addition operation. The SPs of circuits implementing the similar functions are highly correlated each other (correlation coefficient $c > 0.95$). But C6288 is a multiplier and obviously its SP is quite different from the 5 circuits related to the addition operation (correlation coefficient $c < 0$).

5 Further Discussion

Dobrin found that individual motifs aggregate into homologous motif clusters and a supercluster forming the backbone of the E. coli transcriptional regulatory network and play a central role in defining its global topological organization [**?**]. The supercluster coalesced by motif clusters establishes distinct topological hierarchies that show global statistical properties similar to the whole network. We also found that motifs detected in ISCAS85 circuits are usually overlapped and clustered in functional blocks. It will be interesting to make similar investigation on ISCAS85 circuits, and can present more global information on motifs and functional blocks. The problems of detecting features in a given network and of generating synthetic but realistic networks have received considerable attention recently. For instance, a huge amount of benchmark circuits with suitable characteristic parameters is required for the development and evaluation of tools and algorithms of digital circuits [?]. Appropriate models are stringently required to match and generate networks with real-world properties. We applied a machine learning method based on counting subgraphs to evaluate several most widely used models in biological and engineered networks, which has been successfully used to inferring the best mechanism underlying a protein interaction network, and inferred the model most accurately capturing real circuit structure [?]. Our experimental results show that the generalized random graph model keeping the same degree distribution outperforms all other models including the preferential model, the preferential model with spatial constraints, the small-world graph model and the optimization model under multi-objective constraints [2, 4, ?, ?, ?]. It is not really surprising that the generalized random graph model match real circuits better than other models. The input of generalized random model is the degree sequence of the original real circuit, but in general other models only get the number of nodes and the number edges as the input, so basically there are more constraints on the generalized random graph model. We can not simply say that the generalized random graph is a model better than other models, since they have no different intentions and application cases. Essentially the generalized random graph model tries to capture degree distribution of real network directly and is different from our models such as the preferential attachment model and the optimization model, which try to give a general underlying mechanism shaping the network. As the null model in motif detection the generalized random model has been proved to fail to produce functional blocks of real circuits well in our experiments. But compared with other models, the generalized model is still best. This situation presents us a big challenge on generating synthetic but realistic networks, and we still have a lot of work to do on this topic.

Recently Mahadeven proposed a new, systematic approach for analyzing network topologies, which introduces the dK-series of probability distributions specifying all degree correlations within d-sized subgraphs of a given graph G [?]. Increasing values of d capture progressively more properties of G at the cost of more complex representation of the probability distribution. When d = 0, the approach is just the generalized random graph model. She found that the d = 2 case is sufficient for most practical purposes, while d = 3 essentially reconstructs the Internet AS- and router-level topologies exactly. Though dK-series is an approach describing the networks instead of explaining networks, but at least it provide us a systematic approach to quantitatively measure the distance between two graphs and construct random graphs that accurately reproduce virtually all metrics proposed in the literature. So it maybe provides us more insight on the real-world networks such as electronic circuits and offer useful guidelines for generating realistic networks. We can investigate if common small function blocks will emerge in the synthetic networks at an appropriate value of d.

References

1. U. Alon. Biological networks: the tinkerer as an engineer. *Science*, 301(5641):1866–1867, September 2003.
2. Stefano Boccaletti, Vito Latora, Yamir Moreno, Mario Chavez, and D-U. Hwang. Complex networks : Structure and dynamics. *Physics Reports*, 424(4-5):175–308, 2006.
3. Ramon Ferrer i Cancho, Christiaan Janssen, and Ricard V. Solé. Topology of technology graphs: Small world patterns in electronic circuits. *Phys. Rev. E*, 64(4):046119, Sep 2001.
4. Deepayan Chakrabarti and Christos Faloutsos. Graph mining: Laws, generators, and algorithms. *ACM Comput. Surv.*, 38(1), 2006.
5. Mark C. Hansen, Hakan Yalcin, and John P. Hayes. Unveiling the iscas-85 benchmarks: A case study in reverse engineering. *IEEE Des. Test*, 16(3):72–80, 1999.
6. Justin E. Harlow. Overview of popular benchmark sets. *IEEE Design and Test of Computers*, 17(3):15–17, 2000.
7. Piers J. Ingram, Michael PH Stumpf, and Jaroslav Stark. Network motifs: structure does not determine function. *BMC Genomics*, 7:108, 2006.
8. S. Itzkovitz, R. Milo, N. Kashtan, G. Ziv, and U. Alon. Subgraphs in random networks. *Physical Review E*, 68:026127, 2003.
9. S. Itzkovitz, 1 N. Kashtan R. Milo, M. E. J. Newman, and U. Alon. Reply to "comment on 'subgraphs in random networks' ". *Physical Review E*, 70:058102, 2004.
10. Shalev Itzkovitz, Reuven Levitt, Nadav Kashtan, Ron Milo, Michael Itzkovitz, and Uri Alon. Coarse-graining and self-dissimilarity of complex networks. *Physical Review E*, 71:016127, 2005.
11. N. Kashtan, S. Itzkovitz, R. Milo, and U. Alon. Topological generalizations of network motifs. *Physical Review E*, 70:031909, 2004.
12. Oliver D. King. Comment on "subgraphs in random networks". *Physical Review E*, 70:058101, 2004.
13. S. Mangan and U. Alon. Structure and function of the feed-forward loop network motif. *Proc Natl Acad Sci U S A*, 100(21):11980–11985, October 2003.
14. R. Milo, S. Itzkovitz, N. Kashtan, R. Levitt, S. Shen-Orr, I. Ayzenshtat, M. Sheffer, and U. Alon. Superfamilies of evolved and designed networks. *Science*, 303(5663):1538–1542, March 2004.
15. R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network motifs: simple building blocks of complex networks. *Science*, 298(5594):824–827, October 2002.
16. C. R. Myers. Software systems as complex networks: structure, function, and evolvability of software collaboration graphs. *Physical Review E*, 68:046116, 2003.

17. Robert J J. Prill, Pablo A A. Iglesias, and Andre Levchenko. Dynamic properties of network motifs contribute to biological network organization. *PLoS Biol*, 3(11), October 2005.
18. Sebastian Wernicke. Efficient detection of network motifs. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 3(4):347–359, 2006.
19. D. M. Wolf and A. P. Arkin. Motifs, modules and games in bacteria. *Curr Opin Microbiol*, 6(2):125–134, April 2003.