Representing Kekule Cells in Stable Molecules

Aaron Germuth and Alex Aravind

Department of Computer Science
University of Northern British Columbia
E-mail: (germuth,csalex)@unbc.ca

July 19, 2013

Abstract

Silicon transistors may be slowly reaching their minimum size. Possible alternatives to allow the continual decrease of circuitry size is desired. One possibility is the use of π conjugated hydrocarbons as circuitry elements. Certain polycyclic polyunsaturated hydrocarbons have already shown promise in this area. Kekulé theory attempts to characterize the switching behavior of such hydrocarbons which allows them to act as circuitry elements. The Kekulé cell captures this switching behavior and all possible cells have been previously classified. We attempt to, given any cell, find a suitable molecule which has this cell, and therefore the required switching behavior. A genetic algorithm is used to evolve a population of graphs towards the required switching behavior. Most graphs obtained from the genetic algorithm resemble realistic graphs which in theory, could be used. The genetic algorithm is capable of quick producing results for all cells of rank 5 or lower, and most cells of rank 6.

1 Introduction

Moore's Law in 1965 [5] (not positive about cite here) predicts that, over the history of computing hardware, the number of transistors on a single circuit will double approximately every two years (see Figure 1). Although his prediction has proven to be accurate, there are signs which show this exponential progress is coming to an end [6]. Conventional silicon transistors may soon reach their minimum physical size. The amount of leakage current is approaching the order of magnitude of the actual current running through the transistor itself (cite v06[3]).

One alternative possibility to allow smaller circuitry is to use single molecules as computational elements. π conjugated organic molecules have been the main focus, and there have been solutions to make them perform as wires [9], diodes [10, 11], resistors [12, 13], transistors [14, 15, 16, 17], and switches [18, 19]. Following this it was shown a single molecule can represent an entire Boolean function or logic gate (AND[20] , OR[21], IF THEN (implication)[22], NAND[23], and

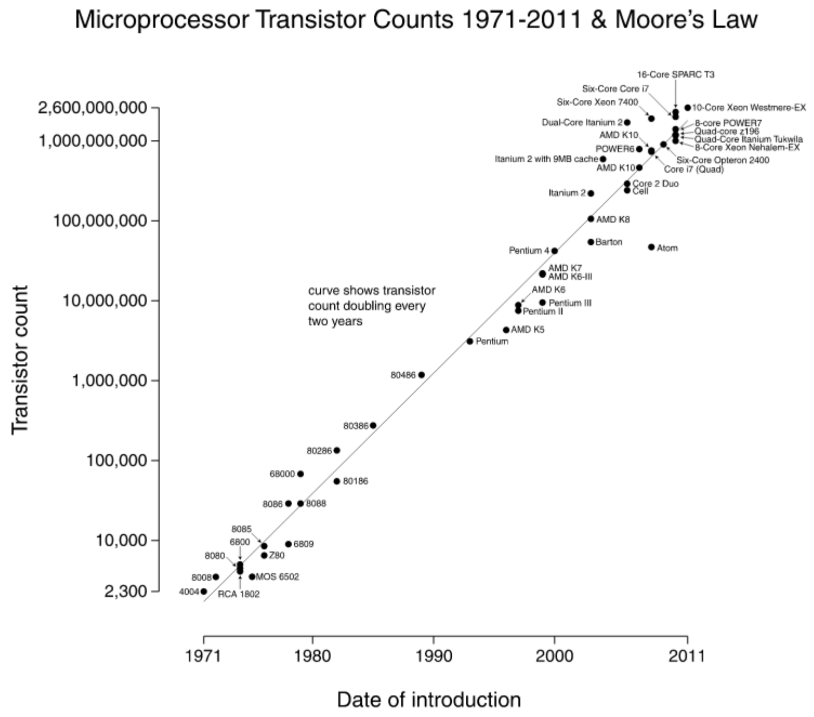


Figure 1: Plot of CPU transistor counts against dates of introduction, on a logarithmic vertical scale.

NOR[24]). Finally, M.H. van der Veen [3] has shown that a single molecule can facilitate all 16 fundamental logic operations. These molecules can even be combined to achieve bifunctional logic elements with increased complexity.

M. H. van der Veen [3] not only found a single molecule which has the above characteristics, but then went on to describe π logic, a way to determine which Boolean operations any π conjugated system can preform. All of this allows for, in theory the design of highly compact and complex logic circuits using π conjugated systems.

In π logic, molecules have terminals which consist of atoms where connections outside the molecule can be attached. π conjugated wires [9] or omniconjugated molecules [7] have been suggested to interconnect separate π conjugated systems. A pair of terminals is called a pathway. If there exists an alternating single-double bond path from terminal to terminal, the path is deemed 'open'. Open paths have been shown to conduct electricity to a much higher extent than closed paths [8]. A path can be alternated (all single bonds become double bonds and vice versa) by sending an electrical 'soliton' over the path [4]. A molecule can contain multiple intersecting paths, and in this way, when one path is toggled, another path may be opened or closed (see Figure 2).

To begin the analysis of a π conjugated system using π logic, all possible double bond patterns of the molecule are deduced. Two terminals are selected for an output, and whether the path between them is open or closed represents an output of '1' or '0' respectively. The other terminals are paired, and each pair represents an input of '1' or '0', again depending on the openness of the

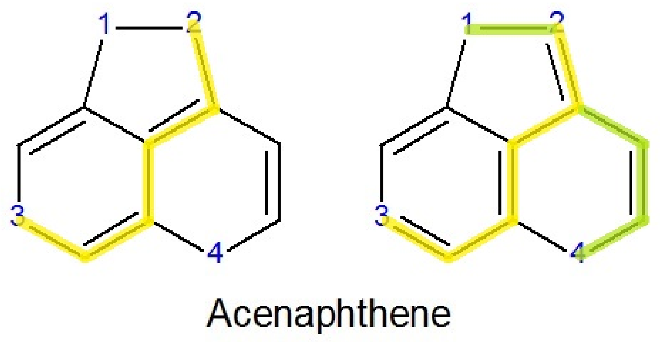


Figure 2: Acenaphthene displaying switching behavior. There exists an alternating path from terminal 2 to 3 (shown in yellow). When this path is toggled (right diagram), the path from terminal 1 to 4 (shown in green) is also opened.

path. From this you can observe all the different Kekulé structures of the molecule and map the inputs to the outputs, resulting with a Boolean function. If the terminals are paired in a different orientation, you may redo the process and have the same molecule represent another Boolean function.

Kekulé theory was introduced [2] to investigate and systematize this qualitative switching behavior. In Kekulé theory, terminals are called ports, and pathways are called channels. A Kekulé state is a configuration of bonds such that every node (other than the ports) has precisely one double bond. Ports may or may not have a double bond. Each Kekulé state (or resonance structure) of the molecule has a so called port assignment, the set of ports which contain a double bond. A Kekulé cell then consists of the set of port assignments for every resonance structure of that molecule (see Figure 3). The rank of a cell or molecule is defined as its number of ports.

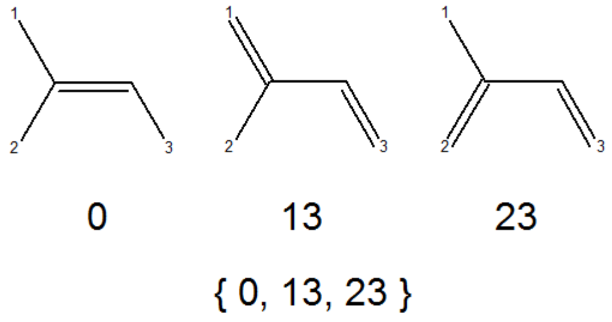


Figure 3: The Kekulé cell of a simple molecule. The 3 structures indicate the 3 possible arrangements of double bonds. The second row indicates the port assignment for each structure, and the third row is the Kekulé cell for that molecule. In Hesselink's notation, ports and labeled with letters rather than numbers, and the cell would be { 0, ac, bc }

.

The switching behavior of a molecule is completely determined by its Kekulé cell [2, 1]. This allows us to search in the opposite direction from before. Rather than searching for any switching behavior a given molecule may exhibit, we can take a required switching behavior needed in some circuitry element, and then search for a molecule who has this behavior. Hesselink [1] has developed a method to classify all possible Kekulé cells, and has given classifications for all cells with ports <7. Therefore, any switching behavior can be converted to a cell, and any cell can be converted to one of Hesselink's [1] classified cells.

The next step is to then, given a Kekulé cell, find a corresponding molecule which has that cell. In Kekulé theory, molecules are abstracted as undirected graphs, where nodes represent atoms, and edges their chemical covalent bonds. These graphs are hydrogen-suppressed as hydrogen plays no role in conjugation. Hesselink [1] has shown a recursive method which has found graphs for any cell with rank <6 and 210 out of 214 cells with rank 6.

However, graphs obtained by Hesselink do not represent realistic molecules. In regards to this, Hesselink mentions some methods which can alter a graph without changing it's cell [2, 3]. This method however, may be limited for graphs of larger ranks (justification?). We propose an alternative, possibly easier solution for obtaining more realistic graphs. We use a genetic algorithm to create a population of graphs, and evolve them towards their desired Kekulé cell.

1.1 Overview

Section 2 consists of the restrictions used to obtain more realistic graphs, and the reasoning behind them. Section 3 explains the genetic algorithm, and how mutation and crossover are applied on graphs. Section 4 presents the results of the genetic algorithms and assesses its validity. This includes how realistic graphs obtained are, and whether these structures can be easily synthesized chemically.

2 Restrictions

Carbon is tetravalent, meaning it has 4 valence electrons available to form covalent chemical bonds. This means in most cases, carbon is only stable when bonded with 4 bonding electrons to complete its octet. These 4 electrons may be provided from 4 distinct molecules, or there could be one molecule providing multiple electrons.

In π conjugated systems, every carbon atom contains precisely one double bond. Double bonds hold twice the amount of electrons and therefore each carbon can be at most connected to 3 distinct atoms. This is seen in all polycyclic polyunsaturated hydrocarbons. In this way, each node in our graphs has a maximum degree of 3. Ports however, must be able to interact with the outside world. If we assume this connection is through molecular wires, as in [9], or through an omniconjugated molecule [3], then each port must restrict one of it's valence electrons to be able to connect to atoms outside the molecule. For this reason, we restrict ports to have a maximum degree of 2.

Cycles must also be confined. Due to steric hinderance, cycles such as cyclopropane or cyclobutane are rarely stable in π conjugated systems. Huckel's rule states that only cycles with $4n + 2$ (2, 6, 10, etc.) delocalized electrons are considered aromatic and stable. However, Huckel's is mainly meant for monocyclic systems, meaning pentagons and heptagons can be stable in certain arrangements (see Azulene, Figure 4). For this reason, we currently only expand cycles of length

3 or 4.

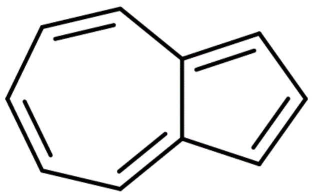


Figure 4: Azulene, a structure composed of an aromatic heptene and pentene ring attached together.

Cycle extension is done by inserting two additional carbon atoms into the cycle at the same position (see Figure 5). This has no effect on the Kekulé cell as the two atoms are conjugated with the rest of the ring and simply expand it. This procedure is nearly identical to operation v of [3]. Care must be taken to not select two nodes which also are a part of another cycle, as the resulting graph will contain unrealistic bonding distances and cycle connectivity. Currently cycles of length 8 and 9 are not reduced to their corresponding cycles of length 6 and 7.

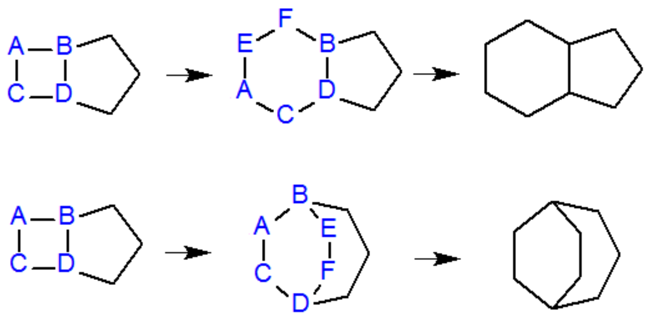


Figure 5: This shows the extension of the left 4 cycle into a proper 6 cycle. The top half shows a correct procedure when selecting to add two nodes in between A and B. The bottom half shows the incorrect procedure of adding two nodes in between B and D. Double bonds are excluded from this Figure for clarity.

Cycles are detected using an upper bounded implementation of the ’flooding’ algorithm (need cite here). In the flooding algorithm, each node sends out a packet to all if it’s neighbors. Any time a node receives a packet, it appends its name and sends the new packet to all of its neighbors. Packets are not sent back to the node who you just received it from. If at any time a packet has been appended by the same node twice, we have found a cycle. As soon as a packet finds a cycle, the packet is discontinued. This process continues to discover all cycles. Packets of length > 12 are also discontinued. This introduces the error that a graph could contain a cycle of length 13. The amount of computation time saved is usually worth it, considering how unlikely a cycle of length 13 or greater is.

Some optimizations can be made considering the subset of all possible graphs we are dealing with. Packets only begin on nodes with degree of 3, as this is guaranteed to find all cycles. Every point within a cycle has degree of 2, so if that cycle is connected to any other node, then at least one of that cycles nodes will have a degree of 3. If there are no nodes with degree of 3, the graph either contains no cycles, or the graph is one big cycle. Therefore, in this case, we only start a single packet at one random node with degree 2. Despite the optimization, this process is still the bottleneck of each iteration for the genetic algorithm.

Not only must cycle size be restricted, but cycle connectivity. In realistic polycyclic polyunsaturated systems, any two rings of the system share at most two nodes. This is seen in naphthalene (see Figure 6), both hexagons share two common atoms. Even in complex polycyclic molecules such as pyracylene (see Figure 6), each cycle shares at most two nodes. Two cycles cannot share a single node (spiro-compound) as the shared node would have a degree of 4. Notice in figure 5, the unrealistic compound has two cycles which share 4 nodes.

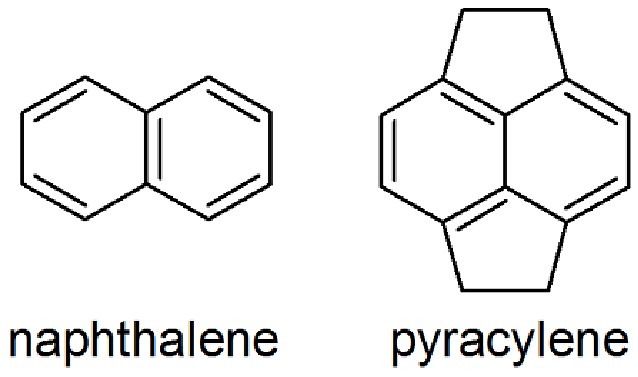


Figure 6: Napthalene shown two benzene rings with 2 carbon atoms is common. Even in complex polycylic hydrocarbons such as pyracylene, every ring shares only 2 carbon atoms.

Graphs must be planar. In order for a molecule to be considered aromatic, all of its contributing atoms must be coplanar. This is because aromatic systems are comprised of multiple conjugated bonds. Each π conjugated bond results from the overlap in the nodal planes of each bonding atom. This causes a cloud of electron probability above and below the covalent bond. Each cloud generates a small electromagnetic field, which works to keep the ring flat (need cite here). A consequence of planarity is that any graph of the molecule should be able to visualized such that no edges are intersecting. (currently we don't 100% detect whether graphs are planar or not)

Finally, graphs representing molecules must be connected. A disjoint graph represents two molecules rather than one. Two disjoint graphs can sometimes be connected using the following procedure. If we add internal vertices which must always conjugate within themselves, they will not interfere with the cell. For example, see Figure 7.

In the right-hand graph of Figure 7, F is not a port and so must always contain a double bond. The only other node which is available to form a double bond with F is E, so in every resonance structure of this molecule E and F will contain a double bond with each other (as shown in right diagram). Therefore d or b can never form a double bond with their new edge (that would result in two adjacent double bonds), and the Kekulé cell has not been changed. This now resembles a stable connected molecule. This procedure is similar to Operation iii and Operation vi of [3].

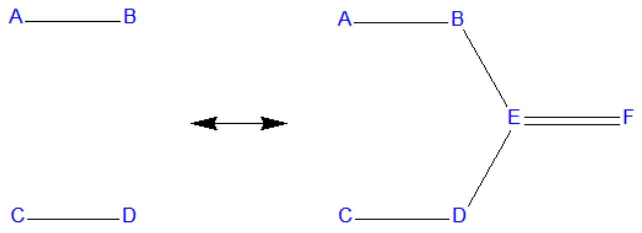


Figure 7: Process used to turn a disjoint graph to a connected graph with the same cell. Two new nodes E and F can be added which always conjugate to each other.

Although these restrictions do a good job of trimming out almost all unrealistic graphs, they are not absolute. Cycles of length 5 and 7 are seen in naturally occurring hydrocarbons, but it is difficult to determine what arrangement makes than stable (need more research here). Additionally, our procedure makes no attempt to ensure all bonding distances are realistic and the molecules displayed can be synthesized (more on this in Section 4).

3 Genetic Algorithm

A genetic algorithm is used to search for graphs which apply to a given cell. The algorithm begins by randomly generating a population of graphs. Each starting graph has anywhere from rank to 31 nodes, and rank - 1 to 20 edges. There must be at least rank - 1 edges in order to connect the the minimum amount of nodes together. Graphs with fitness 0 or lower (normally arises from disconnected ports) are deleted and a new graph will take its place. This makes up the first generation.

In each iteration, the survivors of last generation's population must be chosen. This is done by selecting the best preforming graphs (fitness-wise), along with a smaller amount of random graphs. Random graphs are added to ensure genetic diversity in the population. This sub-population of now undergoes genetic operations such as mutation and crossover.

The genetic algorithm has two terminating conditions. Either the maximum number of iterations has been reached, or some threshold of graphs which satisfy the cell has been reached. In some smaller cases, the genetic algorithm terminates immediately after generating the initial population, as it contained at least x graphs which satisfy the cell (where x is the threshold of graphs desired).

If no graphs are found which satisfy the cell and the restrictions of Section 2, it is possible the genetic algorithm will display graphs who satisfy the cell, but do not satisfy all restrictions. (Not sure if we want this. If we do, need to explain why fitness decremented instead of removed for bad graphs.)

Resulting graphs are converted to SMILES (small molecular input line entry system) in order to be visualized. Conversion first involves finding a spanning tree of the graph. All edges in the original graph which are not in the spanning tree must be edges which complete a cycle. Therefore each node on either side of the edge is labeled as a cycle point. We then preform depth first search over the spanning tree, and append each nodes label as we reach it.

The ports are labeled by atoms with a P (normally reserved for phosphorous, but we can ignore this here). We need not label each port individually since Kekulé cells are invariant across port translation [1]. The open source Chemistry Development Kit (citation) is used to parse the SMILES and generate a two-dimensional structure. Other than the ports, the structure resembles the hydrogen-suppressed carbon skeleton of the molecule.

3.1 Mutation

Mutation involves randomly perturbing graphs to obtain new ones. We normally generate enough mutants so that it is likely each graph will be mutated more than once. Mutations include:

1. 20 % chance to add a new node
2. 20 % chance to remove a random node, and all edges to it
3. 40 % chance to add a random edge [1]
4. 40 % chance to remove a random edge
5. 5 % chance to extends the ports

Port extension involves adding a new node in the position of every port, and then adding the port connected to that node. This will always be viable since any port can have at most degree 2, and after port extension, the new nodes will have at most degree 3. This is often useful for spreading the ports out and allowing new edges without breaking our degree restrictions. Since each port will always have a degree of 1 after extension, each port can facilitate a new bond. See Figure 8.

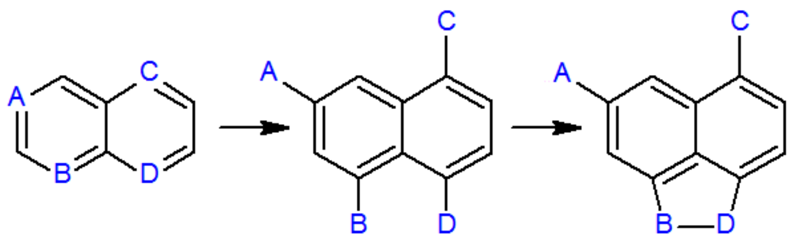


Figure 8: Napthalene (left) undergoing a port extension (center). Notice that after the port extension, it is now feasible to add an edge between B and D, which can then create Acenaphthene (right).

[1]Edges are randomly chosen until an edge is found that can be added without breaking our restrictions on the degree of vertices.

3.2 Crossover

Crossover begins by randomly selecting 2 graphs from the sub-population. These graphs are combined to create a new child graph. On average, each graph produces two children. Children receive all edges with both parents share, and have a 50% chance to get any other edge found in one of the parents. This creates children which are a hybrid of both their parents.

3.3 Fitness Function

Fitness values are integers given to a graph. Any graphs cell can be calculated using Hesselink's procedure here: (explain a lot). The fitness is calculated by iterating over the graphs cell compared to the cell we are evolving towards. For every port assignment they share, the fitness is incremented by one. For every port assignment that is missing, or is extra, fitness is decremented one point. If a cell is empty, with no port assignments [2], it is given a fitness value of -10, in attempt to discourage that graph from evolving onward.

3.4 Additional Note

It should be noted that rather than using the genetic algorithm to approximate a solution (as seen in most applications), an answer generated by it is only useful if it found an actual solution. However, for all cells of rank <6 and most cells of rank 6, the genetic algorithm can converge to multiple solutions for a given cell.

4 Results

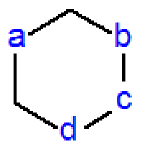
In most small cases (ports <6), the genetic algorithm is able to obtain 10 different answers immediately. For some of the later classifications of rank 5, and cells of rank 6 or higher, it can take longer (up to 5 minutes). Some cells of rank 6 it is unable to find a graph for (more testing needed here). As it is possible that no such realistic graph exists for some cells, it is difficult to determine a solution. The number of nodes is also currently maxed at 31 nodes, as the nodes are stored in a bit array, which would overflow the integer at 32. Removing this may yield more graphs.

Below is graphs for all cells of rank 4 and 5. As stated previously, the genetic algorithm is capable of coming to multiple solutions for each cell, and the ones shown below is simply one selected by us.

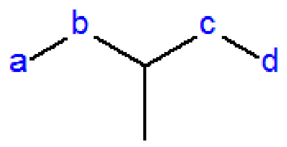
Rank 5 picture here (much larger, 24 graphs to display).

Paragraph about Rank 6, and perhaps a website which shows graphs for as many cells as possible. Then we can give a quote on how many graphs we found.

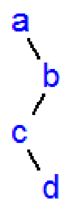
[2]This can happen when the genetic algorithm creates a graph which has a (non-port) vertex with no edges to it (degree 0). There is no way for this graph to have a Kekulé structure since the lone vertex can never have a double bond.



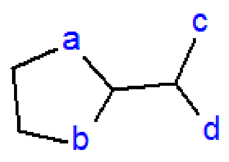
$K_1 = \{\emptyset, ab, ac, ad\}$



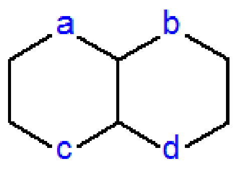
$K_2 = \{\emptyset, bc, ad, abcd\}$



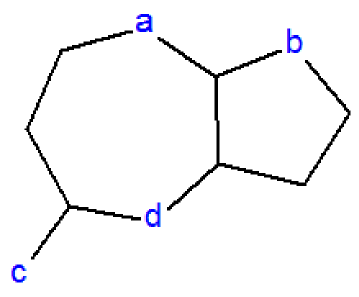
$K_3 = \{\emptyset, ac, bc, ad, bd\}$



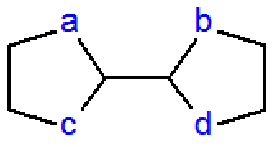
$K_4 = \{\emptyset, ab, ac, bc, ad, bd\}$



$K_5 = \{\emptyset, ac, bc, ad, bd, abcd\}$



$K_6 = \{\emptyset, ab, ac, bc, ad, bd, cd\}$



$K_7 = \{\emptyset, ba, ca, cb, da, db, dc, dcba\}$

Figure 9: Realistic Graphs for each cell in Hesselink's classification of all cells with rank 4. K_1 resembles benzene, K_5 napthalene, and K_6 azulene.

Paragraph talking about whether these molecules can be synthesized.

Paragraph about downside of genetic algorithm, perhaps a picture of the rare molecules which aren't very realistic.

Paragraph about possible improvements. Increase number of nodes. More efficient cycle detection algorithm? Calculate fitness every 2nd iteration?

References

[1] W. H. Hesselink, Graph Theory for alternating hydrocarbons with attached ports. Indagationes Mathematicae, Elsevier, 24:115141, 2013.

[2] W.H. Hesselink, J.C. Hummelen, H.T. Jonkman, H.G. Reker, G.R. Renardel de Lavalette, M.H. van der Veen, Kekule Cells for Molecular Computation. Cornell University Library Online, 2013.

[3] M.H. van der Veen. π-Logic. PhD thesis, University of Groningen, May 2006.

[4] A.J. Heeger, S. Kivelson, J.R. Schrieffer, and W.-P Su. Solitons in conducting polymers, Rev. Mod. Phys.,60:781, 1988.

[5] G. E. Moore, Cramming more components onto integrated circuits. Electronics Magazine. p. 4. Retrieved 2006-11-11.

[6] C. A. Mack, Fifty Years of Moore's Law. IEEE Transactions on semiconductor manufacturing, 24:2 2011.

[7] M. H. van der Veen, M. T. Rispens, H. T. Jonkman, and J. C. Hummelen, Molecules with Linear π-Conjugated Pathways between all Substituents: Omniconjugation. Adv. Function Mater, 14:3, 2004.

[8] S.N. Yalirahi, M.A. Ratner, Interplay of topology and chemical stability on the electronic transport of molecular junctions, Ann. New York Acad. Sci. 960 (2002) 153.

[9] J. Reichert, R. Ochs, D. Beckmann, H. B. Weber, M. Mayor, H. von Lhneysen,Phys. Rev. Lett. 2002, 88, 176804.

[10] A. Aviram, M. A. Ratner, Chem. Phys. Lett. 1974, 29, 277.

[11] D. B. Strukov, K. K. Likharev, Nanotechnology 2005, 16, 137.

[12] Y. Karzazi, J. Cornil, J. L. Brdas, J. Am. Chem. Soc. 2001, 123, 10076.

[13] J. M. Tour, M. Kozaki, J. M. Seminario, J. Am. Chem. Soc. 1998, 120, 8486.

[14] A. Aviram, J. Am. Chem. Soc. 1988, 110, 5687.

[15] H. W. Ch. Postma, T. Teepen, Z. Yao, M. Grifoni, C. Dekker, Science 2001, 293, 76.

[16] C. Joachim, J. K. Gimzewski, Chem. Phys. Lett. 1997, 265, 353.

[17] T. D. Anthopoulos, C. Tanase, S. Setayesh, E. J. Meijer, J. C. Hummelen, P. W. M. Blom, D. M. de Leeuw, Adv. Funct. Mater. 2004, 16, 2174.

[18] G. M. Tsivgoulis, J.-M. Lehn, Chem. Eur. J. 1996, 2, 1399.

[19] J. J. D. de Jong, L. N. Lucas, R. M. Kellogg, J. H. van Esch, B. L. Feringa, Science 2004, 304, 278.

[20] A. P. de Silva, H. Q. N. Gunaratne, C. P. McCoy, Nature 1993, 364, 42.

[21] F. M. Raymo, S. Giordani, J. Org. Chem. 2003, 68, 4158.

[22] K. Rurack, A. Koval'chuck, J. L. Bricks, J. L. Slominskii, J. Am. Chem. Soc. 2001, 123, 6205.

[23] D. Parker, J. A. G. Williams, Chem. Commun. 1998, 245.

[24] A. P. de Silva, I. M. Dixon, H. Q. N. Gunaratne, T. Gunnlaugsson, P. R. S. Maxwell, T. E. Rice, J. Am. Chem. Soc. 1999, 121, 1393.