ICSES 2012 – International Conference on Signals and Electronic Systems
WROCLAW, POLAND, September 18 – 21, 2012

A New Always Cancellation-free Approach to the Multilevel Symbolic Analysis for Very Large Electric Networks

Slawomir Lasota
Institute of Electronics, Silesian University of Technology
Akademicka 16, 44-100 Gliwice, Poland
e-mail: slawomir.lasota@polsl.pl

Abstract— **The paper presents an algorithm of the exact symbolic network function analysis that deals with circuits with any size. The only condition is to decompose the whole circuit into smaller sub-circuits. The decomposition is the multi-level hierarchical one. What is more, the calculation for each level can be done only once and the partial results can be reused any time. A higher level subcircuit does not need too much information about a lower one. The method can be easily implemented in multiprocessor or distributed systems. Although multilevel and compressed structure, symbolical value remains cancellation-free and any path from the root to the terminal vertex represent a single term. Thus, a large-scale and small-scale sensitivities calculation and elimination of less significant terms become simply and natural. To get the *s*-Expanded form, the fast algorithm based on sparse polynomial multiplication methods can be applied.**

I. Introduction

The symbolic analysis derives the analytical characterization of a circuit behavior in terms of circuit parameters. Symbolic simulation results are more instructive to designers than numerical ones. However, the number of symbolic terms in the circuit function increases in the exponential way with the number of nodes and branches in a circuit, [1]. It causes problems with computer resources and with the interpretation of huge formulae. That is why many traditional tools are suitable only for circuits of the moderate size, [2]. To deal with it, either *approximation* methods, [3], or hierarchical approach, [4], are used. Both methods have their own drawbacks. The *approximation* methods can cause some vital terms suppressing or some parasitic poles and zeroes appearance in the network formula, [5]. On the other hand, hierarchical methods (e.g. [4]) produce the exact results, but in a form of some *sequence of expressions* (SOE) that could be illegible for designers.

The Determinant Decision Diagrams (DDD), [6], and the attempt to apply them in the structural synthesis of analog circuits were the basis and the main inspiration of the presented works. These diagrams allow for good compression of the exact network functions, but values associated with vertices are entries of the Modified Nodal Admittance Matrix instead of straight electrical parameters. It leads to the cancellations and produce some pseudo-dependencies in functions. It also makes fast prediction of formulae changes after a new component addition quite difficult. Thus, DDDs are not suitable for the synthesis. A method without above drawbacks, called Parameter Decision Diagram (PDD), was presented in [7]. But to deal with small-signal models of semiconductor devices and large equivalent circuits the hierarchical approach should be applied, anyway.

II. Mathematical Background

A. Higher Order Summative Cofactors

In [8] the higher order summative cofactor (HOSC) of admittance matrix was introduced.

Definition 1: The higher order cofactor is a cofactor of a cofactor. The n-th order cofactor is represented by a symbol $\Delta_{r_1k_1,r_2k_2,\ldots r_nk_n}$, where $r_1, r_2, \ldots, r_n$ and $k_1, k_2, \ldots, k_n$ are the numbers of deleted rows and columns respectively (since then called *deletions* for simplicity). It is equal to the determinant of a matrix with rows $r_1, r_2, \ldots, r_n$ and columns $k_1, k_2, \ldots, k_n$ removed and multiplied by the sign $(-1)^{r_1+r_2+\ldots+r_n+row\_inv}_{k_1+k_2+\ldots k_n+col\_inv}$, where *row_inv* (*col_inv*) are the number of inversions that are necessary to reorder series $r_1, r_2, \ldots, r_n$ ($k_1, k_2, \ldots, k_n$) to get them in the increasing order. Counting the number of inversion is performed to avoid renumbering after application of each pair of deletions (r_ik_i).

The summative cofactors are cofactors with at least one deletion in a form $(a+d)$ or $(a-d)$. $(a+d)$ $[(a-d)]$ means: ***add [subtract] the row [column]*** a ***to [from] the row [column]*** d***, and then remove row [column]*** a. The reasonability of the presented notation comes from the following theorem:

Theorem 1: For cofactors of the same matrix:

$$\begin{aligned}\Delta_{(a+b)(c+d)} &= \Delta_{ac} - \Delta_{ad} - \Delta_{bc} + \Delta_{bd}\\ \Delta_{(a-b)(c-d)} &= \Delta_{ac} + \Delta_{ad} + \Delta_{bc} + \Delta_{bd}\end{aligned} \quad (1)$$

The following set of features of SHOC, which is necessary to manipulate on them, can be easily proved:

Theorem 2:

$$\varDelta_{\underline{a}b,\underline{c}d} = -\varDelta_{\underline{c}b,\underline{a}d} \tag{2a}$$

$$\varDelta_{(\underline{a}+\underline{b})(c+d)} = -\varDelta_{(\underline{b}+\underline{a})(c+d)} \tag{2b}$$

$$\varDelta_{(\underline{a}-\underline{b})(c+d)} = \varDelta_{(\underline{b}-\underline{a})(c+d)} \tag{2c}$$

$$\varDelta_{(a+b)(c+d),(e+\underline{a})(f+g)} = \varDelta_{(a+b)(c+d),(e+\underline{b})(f+g)} \tag{2d}$$

$$\varDelta_{(a+a)(c+d)} \equiv 0 \tag{2e}$$

$$\varDelta_{(a-a)(c+d)} = 2 \cdot \varDelta_{a(c+d)} \tag{2f}$$

$$\varDelta_{11,22,33,\ldots,nn} \equiv 1 \quad \text{even if a matrix is } \boldsymbol{0} \tag{2g}$$

B. Analysis of linear circuits

Let a circuit have 2 input signals, [7]: a voltage source (between nodes a_u and d_u) and a current source (between a_i and d_i), and 2 outputs: a voltage drop (between b_u and c_u) and a short-circuit current (between b_i and c_i). Generally, none of them need to be the reference one. The circuit is determined by the nodal admittance matrix (NAM) (not modified MNAM). The output values can be calculated with a set of 4 network functions:

$$U_O = U_i \cdot A_u + I_i \cdot M \tag{3a}$$

$$I_O = U_i \cdot N + I_i \cdot A_i \tag{3b}$$

$$A_u = \frac{\varDelta_{(a_u+d_u)(b_u+c_u),(b_i+c_i)(b_i+c_i)}}{\varDelta_{(a_u+d_u)(a_u+d_u),(b_i+c_i)(b_i+c_i)}} \tag{3c}$$

$$M = \frac{\varDelta_{(a_i+d_i)(b_u+c_u),(a_u+d_u)(a_u+d_u),(b_i+c_i)(b_i+c_i)}}{\varDelta_{(a_u+d_u)(a_u+d_u),(b_i+c_i)(b_i+c_i)}} \tag{3d}$$

$$N = \frac{\varDelta_{(a_u+d_u)(b_i+c_i)}}{\varDelta_{(a_u+d_u)(a_u+d_u),(b_i+c_i)(b_i+c_i)}} \tag{3e}$$

$$A_i = \frac{\varDelta_{(a_i+d_i)(b_i+c_i),(a_u+d_u)(a_u+d_u)}}{\varDelta_{(a_u+d_u)(a_u+d_u),(b_i+c_i)(b_i+c_i)}} \tag{3f}$$

More generally, the rule of thumb tells:

- In the common denominator a symmetrical pairs of deletions $(a+d)(a+d)$ that comes from each autonomic voltage source and each real short-circuit are added.
- In the numerator the first pair of deletions $(a+d)(c+b)$ always corresponds to any kind of the input signal between a and d and any kind of the output signal between c and b. Additionally, symmetrical deletions from any autonomous voltage sources and any short-circuits different than the input and output ones also should be added.

C. Parameter Extraction

If some network function has a form $T = \varDelta_N / \varDelta_D$ and T is a function of transconductance g_m, one could exclude g_m from an admittance matrix and determine, [7]:

$$T = \frac{g_m \cdot \varDelta_{N,(p+r)(k+l)} + \varDelta_N}{g_m \cdot \varDelta_{D,(p+r)(k+l)} + \varDelta_D} \tag{4}$$

Controlling voltage is placed between nodes k and l and the output current flows from node p and r. For any admittance, of course, $p = k = a$ and $r = l = b$.

For any impedance (4) can be transformed into a form:

$$T = \frac{Z \cdot \varDelta_N + \varDelta_{N,(a+b)(a+b)}}{Z \cdot \varDelta_D + \varDelta_{D,(a+b)(a+b)}} \tag{5}$$

Controlled sources different than VCCS cannot be included into the genuine NAM. This inconvenience can be easily circumvented by the extraction rules:

$$\begin{aligned}
T &= \frac{K \cdot \varDelta_{N,(p+r)(k+l)} + \varDelta_{N,(p+r)(p+r)}}{K \cdot \varDelta_{D,(p+r)(k+l)} + \varDelta_{D,(p+r)(p+r)}} \\
T &= \frac{\beta \cdot \varDelta_{N,(p+r)(k+l)} + \varDelta_{N,(k+l)(k+l)}}{\beta \cdot \varDelta_{D,(p+r)(k+l)} + \varDelta_{D,(k+l)(k+l)}} \\
T &= \frac{r_m \cdot \varDelta_{N,(p+r)(k+l)} + \varDelta_{N,(p+r)(p+r),(k+l)(k+l)}}{r_m \cdot \varDelta_{D,(p+r)(k+l)} + \varDelta_{D,(p+r)(p+r),(k+l)(k+l)}}
\end{aligned} \tag{6}$$

D. Pathological Components

In a few last years, the application of the pathological components to the symbolic analysis became popular, e.g. [9]. These components do not have any numerical values. They only modify the topology of a circuit, and decrease results dramatically. They are represented by the only single delation: from *norator* $(a_{nr} + b_{nr})$ for rows, from *current mirror* $(a_{cm} - b_{cm})$ for rows, for *nullator* $(a_{nl} + b_{nl})$ for columns, and for *voltage mirror* $(a_{vm} - b_{vm})$ for columns, [10].

III. Hierarchical Parameter Decision Diagram

A. 1-level Parameter Decision Diagram

Numerators and denominators in the formulae (4)÷(6) can be represented by a vertex in a decision diagram: $factor \cdot divisor + reminder$. The $factor$ is a parameter associated with a vertex, the $divisor$ is a 1-descendant and the $reminder$ is a 0-descendant.

Let a circuit is represented by a set of components. Each component is represented by a symbol and a set of circuit nodes to which the component is connected. Let us assume that components are arranged in the ordered list. Let us extract component by component according to the order in the list. Each parameter split each cofactor into 2 new ones. These new cofactors may differ from the parent one with some additional pairs of deletions that depend on the type of component (formulae (4)÷(6)). After extraction of the last component from the list, the set of cofactors of **0** matrix remains. A cofactor of **0** matrix differs from 0 only if the number of pairs of deletions is equal to the number of nodes, see (2g). This value can be equal to $\pm 2^p$, where p is the number of deletions in a form $(a-b)$, see (2f). The sign can be determined simultaneously with the validation of terms from rules (2). The algorithm is presented in [7].

Example 1: Let us determine the numerator and the denominator of the voltage gain for a circuit from Fig. 1. From (3c) $A_u = \varDelta_{13}/\varDelta_{11}$. Let the order of components be: A, $R1$, $C2$, $C1$, $R2$. The analysis looks as follows:

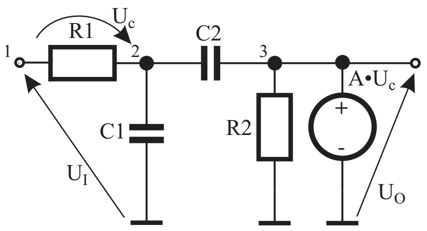

Fig. 1. The circuit to Example 1

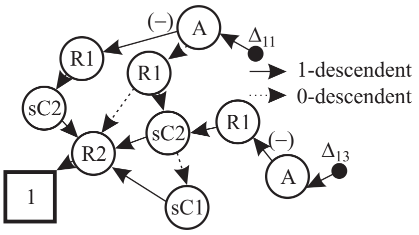

Fig. 2. PDD to the circuit from Fig. 1

1) The denominator root is represented by $\varDelta_{11}$ ([d1]); the numerator root – by $\varDelta_{13}$ ([d2])
2) VCVS A divides: [d1] into $\varDelta_{11,(0+3)(2+1)} = -\varDelta_{11,32}$ ([d3]) (and the sign "$-$" becomes the 1-descendant attribute) and $\varDelta_{11,(0+3)(0+3)} = \varDelta_{11,33}$ ([d4]); and [d2] into $\varDelta_{13,(0+3)(2+1)} = -\varDelta_{13,3(2+1)}$ ([d5]) and $\varDelta_{13,(0+3)(0+3)} \equiv 0$;
3) $R1$ divides: [d3] into $\varDelta_{11,32}$ ([d6]) and $\varDelta_{11,32,(2+1)(2+1)} \equiv 0$; [d4] into $\varDelta_{11,33}$ ([d7]) and $\varDelta_{11,33,(2+1)(2+1)} = \varDelta_{11,22,33}$ ([d8]); [d5] into $\varDelta_{13,3(2+1)}$ ([d9]) and $\varDelta_{13,3(2+1),(2+1)(2+1)} \equiv 0$;
4) $C2$ divides: [d6] into $\varDelta_{11,32,(3+2)(3+2)} = \varDelta_{11,22,33}$ ([d10]) and $\varDelta_{11,32} = 0^{(*)}$; [d7] into $\varDelta_{11,33,(3+2)(3+2)} = \varDelta_{11,22,33}$ ([d10]) and $\varDelta_{11,33}$ ([d11]); [d8] into $\varDelta_{11,22,33,(3+2)(3+2)} \equiv 0$ and $\varDelta_{11,22,33}$ ([d10]); [d9] into $\varDelta_{13,3(2+1),(3+2)(3+2)} = \varDelta_{11,22,33}$ ([d10]) and $\varDelta_{13,3(2+1)}$ ([d12]);
5) $C1$ divides: [d10] into $\varDelta_{11,22,33,22} \equiv 0$ and $\varDelta_{11,22,33}$ ([d13]); [d11] into $\varDelta_{11,22,33}$ ([d13]) and $\varDelta_{11,33} = 0^{(*)}$; [d12] into $\varDelta_{13,3(2+1),22} = \varDelta_{11,22,33}$ ([d13]) and $\varDelta_{13,3(2+1)} = 0^{(*)}$;
6) $R2$ divides: [d13] into $\varDelta_{11,22,33} = 1$ and $\varDelta_{11,22,33,33} \equiv 0$.

The results signed with $^{(*)}$ are not identically equal to 0. The only valid terminal result is any permutation of cofactor $\varDelta_{11,22,33}$, and for these signed cofactors the list of remaining components does not allow to obtain it. In practice, the incidence vectors allow for early detection of the *hopeless cases* and significantly reduce the analysis time.

The whole result is presented in Fig. 2. It can be seen in the figure that there is no 2 vertices that have the same associated symbols and the same descendants. This kind of *recycling* of the repeated parts of results is the source of the effectiveness of the decision diagram representation.

B. Pros and cons of using a hierarchical decomposition

A typical circuit is inherently hierarchical and consist of repeating building blocks, the building blocks consist of some semiconductor devices, and the semiconductor devices are represented by the same small-signal models. Thus, we should expect more advanced repetitions in results than in the *flat* analysis.

Unfortunately, the most of hierarchical methods leads to smaller but more complex results. The reason is that the subcircuit is usually represented by some *black box* described with the set of boundary parameters that are rational function. This inconvenience eliminates these methods from application in the structural synthesis. Furthermore, each hierarchical method which is known by the author generates cancellations, even if in the *flat* form is cancellation-free, e.g. [11]. Thus, there is a need to find another solution.

C. A new approach to the hierarchical decomposition – Let the topology decide

The main source of cancellations in the well known hierarchical methods comes from an operating on some universal and arbitrary defined set of boundary parameters and the arbitrary established local reference node that are insensitive to the real topology of the subcircuits. So let us assume there is no distinguished reference node, and let us allow *the subcircuite to define the boundary parameters*, [12].

Components can be taken in any order in a PDD creation, so we can start with the most basic structures like models of semiconductor devices, and then jump to another part of a circuit. If some part of a circuit is the same (e.g. the model of transistor) we can analyze it only once, and then clone many times. This subcircuit can be treated as a meta-component with a few boundary pins. The subcircuit is represented by a piece of PDD, and it can be treated as a meta-vertex with a few *descendants* that are disjoined. Each *descendant* is described with a set of deletions, similarly to 0- and 1-descendants for the basic vertex. Let us call such a structure a Hierarchical Parameter Decision Diagram (HPDD).

Example 2: Let an RC branch be our subcircuit , see 3a). Let resistor R be connected between nodes a and m and capacitor C – between m and b. Let us start the analysis with these components. We get a PDD from Fig. 3b). There are 4 descendants. The components R and C can be added separately to any leaf of the external circuit, and then we need to check 6 cofactors, or we can connect the whole structure at once, and than only 4 cofactor should be tested. Profit will be higher, if there are internal nodes that are not accessible within the external circuit. E.g. let us assume that m is the internal node. m is not accessible, thus it must be mandatory *linked* to some boundary node for the both rows and columns. That is why only leaves with m in deletions are valid. Thus, $1.\varDelta \rightarrow 0$, $0.\varDelta_{(b+m)(b+m)} \rightarrow \varDelta$, $3.\varDelta_{(a+m)(a+m)} \rightarrow \varDelta$, and $2.\varDelta_{(a+m)(a+m),(b+m)(b+m)} = \varDelta_{(a+b)(a+b),(b+m)(b+m)} \rightarrow \varDelta_{(a+b)(a+b)}$. The whole model can be included by analyzing only 2 cofactors, Fig. 3c).

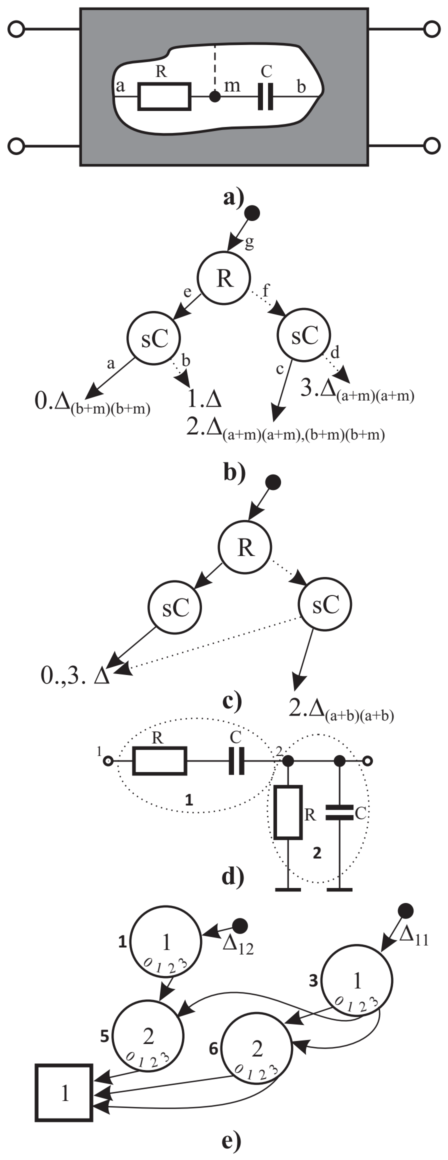

Fig. 3. An example of the hierarchical PDD

In Fig. 3d) there is a circuit that can be split into 2 RC subcircuits. The 1^{st} subcircuit has a node m inaccessible, and leaves are described as: 0 & 3 – $\mathit{\Delta}$, 1 – 0, and 2 – $\mathit{\Delta}_{(1+2)(1+2)}$. In the 2^{nd} subcircuit both nodes a and b are the reference one, and leaves should be described as: 0 & 3 – $\mathit{\Delta}_{22}$, 1 – $\mathit{\Delta}$, and 2 – 0. To determine the voltage transfer function cofactors $\mathit{\Delta}_{12}$ and $\mathit{\Delta}_{11}$ should be found. The procedure is similar to that for genuine vertices, see Example 2. The result can be seen in Fig. 3e). Descendants suppressed to 0 are left unconnected for the clarity of he diagram.

Let us summarize the features of HPDD:

1) The partitioning of a circuit into sub-circuits can be arbitrary, but it is recommended to do it according to the functionality in the circuit.
2) The largest number of internal nodes (inaccessible to the master circuit), the greater profit is.
3) The analysis is hierarchical, i.e. any sub-circuit can become a part of the larger sub-circuit.
4) Each once generated sub-graph for some sub-circuit can be stored in hard disk and can be reused any time it appears in any circuit.
5) The analysis consists of ”sticking together” of existing sub-graphs into the larger graph or sub-graph.
6) The sub-graph occurs many times in the master graph and is represented by the meta-vertices with disjoined descendants.
7) Contradictory to the well-known hierarchical methods, there is no problem to leave some boundary nodes unconnected.
8) When combining sub-graphs, the information about a terminal leaves is the only necessary one, without going into the internal structure of the circuit and a sub-graph as well. It gives the potential possibility of distributed network computing.

D. The meta-vertex development & interpretation

The presented form could be useless without possibility of interpretation and the numerical calculation of results. To take full advantage of the presented structure, the computation algorithm should determine results for sub-circuit only once, and the intermediate result should be reused as soon as the need arises.

Let us introduce the *intermediate result vector* (*irv*) for the sub-graph associated withe the meta-vertex $irv = \begin{bmatrix} d_0 & d_1 & \dots & d_{nd} \end{bmatrix}$, where nd is the number of descendants, and $d_1, d_2, \dots, d_{nd}$ are the current values. For *leaf* no. k $irv_k = \begin{bmatrix} 0 & 0 & \dots & 1 & \dots & 0 \end{bmatrix}$, where 1 lies at k-th position. For irv the standard addition of two vectors and multiplication by scalar are defined. For any vertex in a form $v = p \cdot d_1 + d_0$ $irv_v = p \cdot irv_{d_1} + irv_{d_0}$ should be determined. It can be generalized for met-vertex with nd descendants represented by *intermediate result vector* (*imv*) $inv_v = \sum_{i=0}^{nd} imv[i] \cdot d_i$.

Example 3: For the sub-graph from 3a) we have: $inv_a = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}$, $inv_b = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}$, $inv_c = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}$, $inv_d = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}$. $inv_e = sC \cdot inv_a + inv_b = \begin{bmatrix} sC & 1 & 0 & 0 \end{bmatrix}$ and $inv_f = sC \cdot inv_c + inv_d = \begin{bmatrix} 0 & 0 & sC & 1 \end{bmatrix}$. Finally $inv_f = R \cdot inv_e + inv_f = \begin{bmatrix} sC \cdot R & R & sC & 1 \end{bmatrix}$.

To obtain values for the circuit from Fig. 3d) and graph 3e) we have $inv_{inp_5} = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}$, so $inv_5 = \begin{bmatrix} R2 \end{bmatrix}$. $inv_{inp_1} = \begin{bmatrix} 0 & 0 & R2 & 0 \end{bmatrix}$, so $\mathit{\Delta}_{12} = inv_1 = \begin{bmatrix} sC1 \cdot R2 \end{bmatrix}$. Similarly, $inv_{inp_6} = \begin{bmatrix} 1 & 0 & 0 & 1 \end{bmatrix}$, so $inv_6 = \begin{bmatrix} sC2 \cdot R2 + 1 \end{bmatrix}$. $inv_{inp_3} = \begin{bmatrix} sC2 \cdot R2 + 1 & 0 & R2 & sC2 \cdot R2 + 1 \end{bmatrix}$, so finally $\mathit{\Delta}_{11} = inv_3 = \left[\left(sC2 \cdot R2 + 1 \right) \cdot \left(sC1 \cdot R1 + 1 \right) + sC1 \cdot R2 \right]$.

E. s*-Expanded result development*

The most instructive to designers form is so called *s*-Expanded one – the numerator and the denominator of transfer functions are represented by a polynomial of *s* with the symbolical coefficients. Unfortunately this form is usually much less *compressed* than genuine one. One of the reason

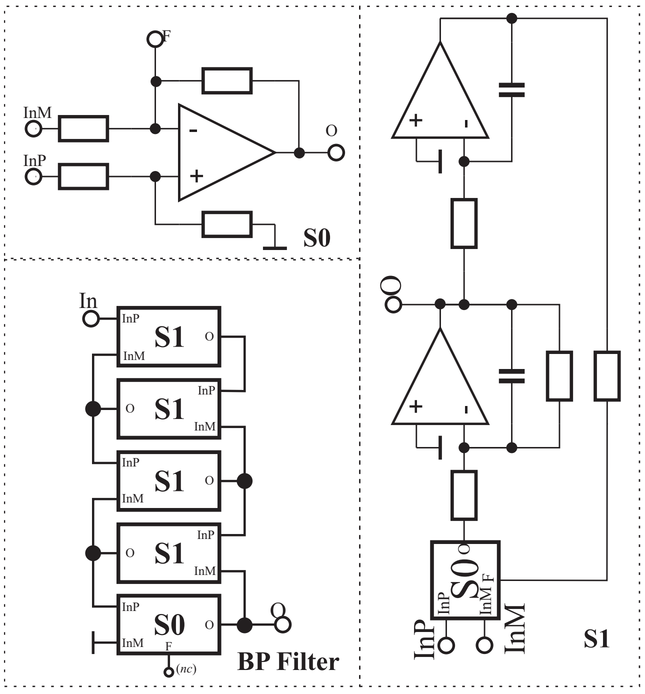

Fig. 4. Multilevel 6th order band-pass filter.

is that each capacitor and inductor increase the order of polynomials and it follows naturally to multi-root decision diagram.

If each entry in *intermediate result vector* (see III-D) is represented by a polynomial instead of symbol or number the s-Expansion becomes natural and it reduces to the multiplication of polynomials. On the base of simulations it can be concluded that the algorithm runs faster and consumes less resources if one uses special algorithms for rare polynomials, e.g. [13]. In the future, the comparison with the classical Karatsuba, Schönhage and Strassens, and based on FFT algorithms should be tested.

IV. Analysis of some real circuits

Of course using of the presented method for circuit from Example 2 is the triumph of form over content. However, the limited space of the paper does not allow to present the analysis of any real electronic circuits in details. For the practical circuit analysis a program *PDDEngine* in C++ was created and tasted on a PC computer with the processor Intel Core Duo 3.0GHz, 3 GB memory, and the operation system Windows XP.

*1) **B-P FILTER**:* First, the benchmarks from the web page: `http://rodanski.net/ben/work/symbolic/index.htm` were tested, see Fig. 4. The op-amp macromodel by Vlach, [14], has 5 nodes (4 boundary, including local reference one), 2 VCCS, 5 resistors, 1 capacitor. The circuit was cut into subcircuits: S0 – 6 nodes (5 boundary, including local reference one), 1 op-amp, 4 resistors; S1 – 9 nodes (4 boundary, including local reference one), 1 subcircuit S0, 2 op-amps, 3 resistors and 2 capacitors. The whole circuit has 7 nodes and consists of 4 subcircuits S1 and 1 S0. The op-amp meta-component was created in 1.8 ms, S0 – in 2.9 ms, S1 – in 47,2 ms, and the whole voltage transmittance – in 134 ms. Op-amp is modeled as 25 vertices and 13 leaves, S0 – as 31 meta-vertices and 38 leaves, S1 – as 99 meta-vertices and 41 leaves. To represents both the numerator and the denominator 91 meta-vertices are needed. The program allows to get the *flat* PDD representation from the hierarchical one. The whole PDD consists of 24,964 vertices, but thanks to *recycling* of repeating subcircuits only 1,487 are enough to represent them in the computer memory. In s-Expanded form the voltage transmittance is a rational function of 2 polynomials of s with the highest power 21. The s-Expanded 1-level PDD consists of 252,639 vertices but it can be represented by 4,832 meta-vertices. The time of 1-level PDD creation was about 7.75 s, and the s-Expanded numerical calculations took 9.22 s. While the same calculation for HPDD took only 31.95 ms. This example was to weak for the intensive tests of the algorithm.

*2) **CT filter**:* The benchmark was crated from benchmarks for analog and mixed signal testing from `http://www.cs.wright.edu/~emmert/analog_bm_ckts/main_bm.htm`. The main circuit is the universal analog filter (UAF) with one input and 4 outputs (low-pass, band-pass and high-pass), see Fig. 5a). Each op-amp is represented by 9 MOS transistor circuit from Fig. 5b). Each transistor is represented by sub-micron Level 3 SPICE model from Fig. 5c). The PDD for MOS transistor was created in 4.3 ms, the model of op-amp in 246,8 ms and the whole result (4 cofactors) in 1.6416 s. The transistor is represent by 69 vertices and 24 leaves, op-amp by 403 meta-vertices and 97 leaves and 3 numerators and the common denominator are represented by 896 meta-vertices. It gives 3,548 meta-vertices. The 1-level PDD consists of 901,854 vertices. Unfortunately, s-Expanded form is huge. The limit of the available memory for the single process in Windows XP forced calculations only for the numerator for high-pass output. 1-level PDD for that numerator consist of 191 223, while in HPDD only 149. In s-Expanded form we has obtained polynomial with highest power 68 and we needed 6,868,137 vertices. The creations of 1-level PDD took 17.25 s, and the numerical calculation 74.77 s. The same calculations for HPDD needed only 333 ms. The calculation of the whole result (3 numerators and 1 common denominator) took only 847 ms.

In the next step an attempt to calculate the higher order low-pass filter containing 2 UAFs were made. The creation of PDD sub-graph for UAF took 2.22 s and the transfer function determination 1 ms. The s-Expanded numerical values calculation needed 7.07 s and we have obtained numerator polynomials of power 136 and denominator polynomial of power 140. Of course this highest power coefficients are much less significant than the lowest one.

V. Conclusion

The obtained results are promising. The attempts to calculate transfer function of the circuit that contains 48 MOS transistors represented by the 5-node Level-3 small signal

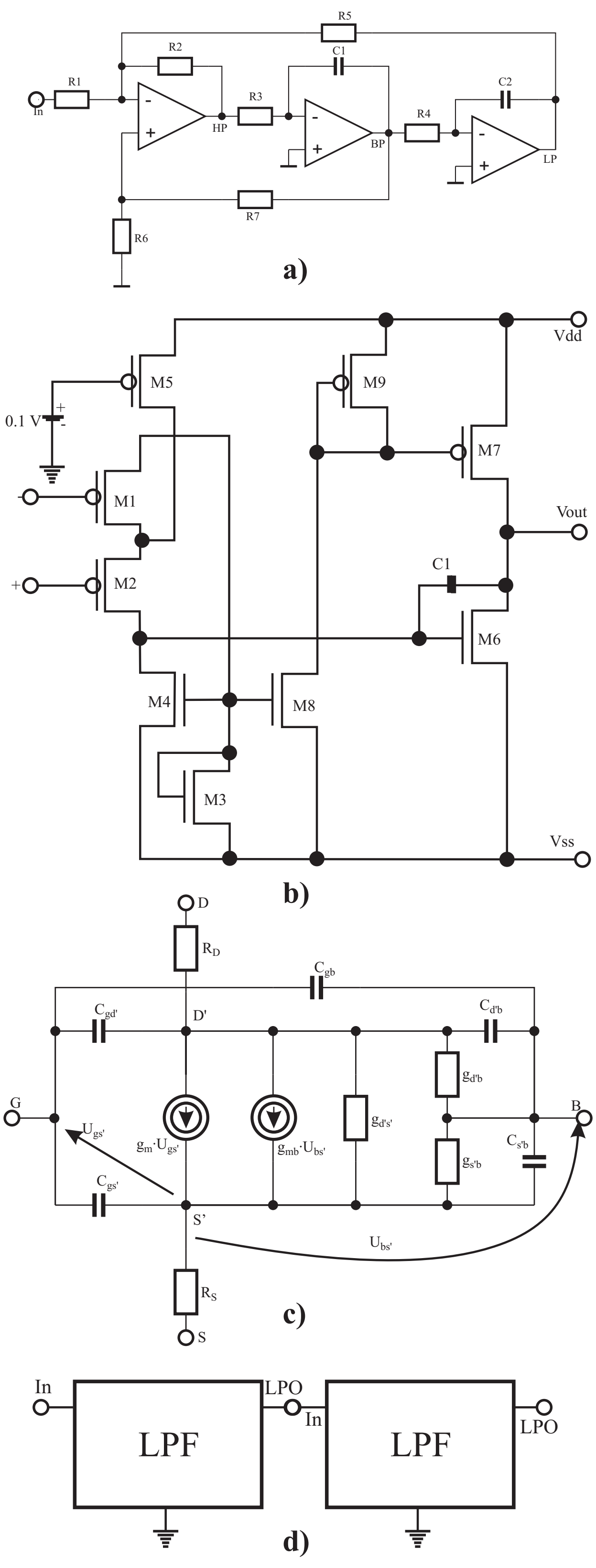

Fig. 5. Large multilevel circuit - the universal active filter.

model was easy and quite fast. But the work on the HPDDs is still in progress. PDDs allow also for the implementation of an effective algorithm for determining the divisor and reminder of selected parameters. These operations are the basic ones for the small- and large-scale sensitivities calculation. The algorithm is quite effective, because the operation can be made locally inside a meta-vertex. Now, the works on the direct generation of simplified flat PDD (which should contain only the most significant values) from HPDD are made. The key to the simplification is based on the form of any vertex: $factor \cdot divisor + reminder$. It is not too difficult to decide which descendent is much lower then the other one and suppress the whole less significant sub-trees.

ACKNOWLEDGMENT

The author would like to appreciate prof. Ben Rodanski for the benchmarks delivery and supporting in the researches.

REFERENCES

[1] P. Lin, *Symbolic Network Analysiss*. Amserdam, the Netherlands: Elsevier Science, 1994.

[2] G. Gielen, P. Wambacq, and W. Sansen, "Symbolic analysis methods and applications for analog circuits: a tutorial overview," *Proceedings of the IEEE*, vol. 82, no. 2, pp. 287 – 304, feb 1994.

[3] Q. Yu and C. Sechen, "Approximate symbolic analysis of large analog integrated circuits," in *Computer-Aided Design, 1994., IEEE/ACM International Conference on*, nov 1994, pp. 664 – 671.

[4] M. Hassoun and P.-M. Lin, "A hierarchical network approach to symbolic analysis of large-scale networks," *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, vol. 42, no. 4, pp. 201 – 211, apr 1995.

[5] P. Wambacq, P. Dobrovolny, G. Gielen, and W. Sansen, "Symbolic analysis of large analog circuits using a sensitivity-driven enumeration of common spanning trees," *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, vol. 45, no. 10, pp. 1342 – 1350, oct 1998.

[6] C.-J. Shi and X. Tan, "Canonical symbolic analysis of large analog circuit with determinant decision diagram," *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, vol. 19, no. 1, pp. 1–18, 2000.

[7] S. Lasota, "Parameter decision diagrams in the symbolic analysis and the structural synthesis. part i, ii, iii," in X^{th} *Int. Workshop on Symbolic and Numerical Methods, Modeling and Application to Circuit Design* SM^2ACD *'08*, Erfurt, Germany, oct. 2008, pp. 149–157,172–179,180–187.

[8] V. Sigorskij and A. Petrenko, *Algorithms of the Analysis of Electronic Circuits*. Kiev: Tehnika, 1971, in Russian.

[9] C. Sanchez-Lopez, F. Fernandez, E. Tlelo-Cuautle, and S. Tan, "Pathological element-based active device models and their application to symbolic analysis," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 58, no. 6, pp. 1382 –1395, june 2011.

[10] S. Lasota and A. Malcher, "A cancellation-free algorithm for the symbolic analysis of circuits containing current conveyors," in *International Conference on Signals and Electronic Systems, ICSES '04.*, Poznań, Poland, Sept. 2004, pp. 171 – 174.

[11] V. Filaretov, "A topological analysis of electronic circuits by a parameter extraction method," *Electrical technology Russia*, vol. 2, 1998.

[12] S. Lasota, "Fast multilevel & always cancelation-free method for large electric networks exact symbolic analysis," in *Programmable Devices and Embedded Systems PDeS 2012, 11th IFAC/IEEE International Conference on*, Brno, Czech Republic, may 2012, pp. 188 – 193.

[13] M. Monagan and R. Pearce, "Sparse polynomial division using a heap," *Submitted September 2008 to the JSC special issue on Milestones in Computer Algebra*. [Online]. Available: http://www.cecm.sfu.ca/CAG/papers/MonHeapsRevised.pdf

[14] K. Singhal and J. Vlach, *Computer Methods for Circuit Analysis and Design*. Springer, 2010.