Custom Design of a VLSI PCM-FDM Transmultiplexer from System Specifications to Circuit Layout Using a Computer-Aided Design System

RAJEEV JAIN, MEMBER, IEEE, FRANCKY CATTHOOR, JAN VANHOOF,
BART J. S. DE LOORE, GERT GOOSSENS, NELSON F. GONCALVEZ, MEMBER, IEEE,
LUC J. M. CLAESEN, MEMBER, IEEE,
JOHAN K. J. VAN GINDERDEUREN, JOOS VANDEWALLE, SENIOR MEMBER, IEEE AND
HUGO J. DE MAN, SENIOR MEMBER, IEEE

Abstract—**The computer-aided design of a VLSI PCM-FDM transmultiplexer is presented. The entire design process from *system specifications to integrated circuit layout* is carried out with the aid of specialized computer programs for analysis, synthesis and optimization at each design level: *the filter network*, the *architecture* and the circuit *layout*. These CAD tools support a top-down custom design methodology based on bit-serial architectures and standard cells. A *customized architecture* is constructed, which is integrated using a *5-μm CMOS* cell library. The results are compared with a fully manual design and demonstrate the power of architecture based *computer-aided design methodologies for VLSI filtering*. By combining both *synthesis* as well as *optimization* aids at each design level, it is possible to achieve a high degree of automation while retaining an efficient use of silicon area, high throughput and moderate power consumption.**

I. Introduction

IN THIS PAPER we present a Computer-Aided Design system called CATHEDRAL [13], [19] for automating the design of custom integrated digital filters. As shown in Fig. 1 the CATHEDRAL design system consists of computer programs and a CMOS circuit cell library. An overview of the programs and the cell library is given in Section II and their application to the design of a VLSI PCM–FDM transmultiplexer [20] is illustrated in Section III.

A fundamental problem, in the custom design of a VLSI signal processing system, is to choose the combination of filter structure, signal processing architecture and integrated circuit technique, which leads to the most *economical* monolithic device, while satisfying all the system requirements and specifications. In the absence of a systematic design strategy to handle this complex problem, it becomes necessary to evaluate the *fabrication cost* of a selected combination of filter network, architecture and integrated circuit in terms of the chip area, power dissipation and throughput/unit area. If the cost is too high, iterative redesign and reevaluation become necessary, resulting in a long *design time*. Monolithic general purpose programmable devices known as *digital signal processors* can cut these costs and reduce the design time by relieving the designer from architecture and circuit design tasks. However, the fixed architecture of commercial general purpose signal processors is not always optimally suited for implementing custom VLSI systems. The FIRST [3] *silicon compiler* allows the design of a custom signal processing chips in a very short time, where the architecture can be customized to the special needs of the system. However, the designer has to manually perform the tedious job of translating the filter algorithm into a bit-serial architecture. Moreover, the fixed procedures in FIRST do not exploit the potential reduction in circuit complexity inherently possible in custom designs and hence result in substantially lower silicon efficiency compared to a fully manual custom design [4].

In order to achieve efficient design automation as well as cost-effective silicon utilization, De Man *et al.* [5] have proposed the concept of *computer-oriented architecture-based design methodologies* for custom integration of digital filters. The central ideas are the following:

1) The design time is cut by formal procedures (Fig. 1):

(a) for *synthesis* of filter algorithms, whose structure optimally matches the target architecture;

(b) for *customizing* or synthesizing the *architecture* to implement a given algorithm using parametrizable building blocks;

Manuscript received March 22, 1985; revised September 30, 1985. This work was done under the project ESPRIT 97 and supported by the European Communities, Siemens AG, West Germany, Philips, The Netherlands, Silvar-Lisco, Belgium, and Bell Telephone Manufacturing Company, Belgium.

R. Jain was with the ESAT Laboratory, Katholieke Universiteit Leuven, and IMEC, Belgium. He is now with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720.

F. Catthoor, J. Vanhoof, B. J. S. De Loore, G. Goossens, N. F. Goncalvez, L. Claesen, J. van Ginderdeuren, J. Vandewalle, and H. J. De Man are with the Inter-University Microelectronics Centre (IMEC), Kapeldreef 75, B-3030 Heverlee, Belgium, and the ESAT Laboratory, K. U. Leuven.

IEEE Log Number 8406488.

0018-9200/86/0200-0073$01.00

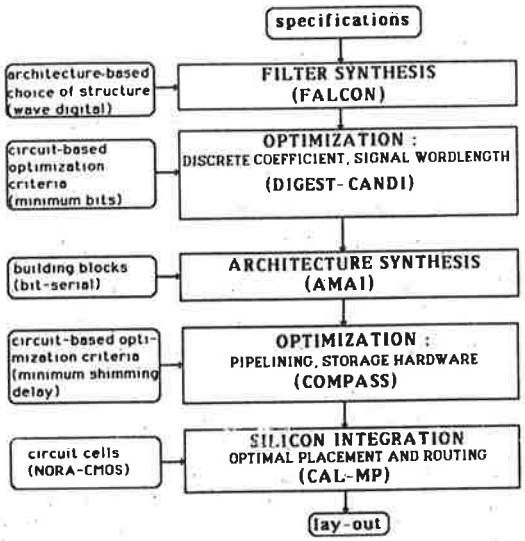


Fig. 1. Concept of the computer-aided design system CATHEDRAL for custom integration of digital filters using pipelined bit-serial architectures. The names in brackets correspond to the components of the CATHEDRAL design system which are described in Section II.

(c) for optimal *silicon implementation* of the resulting architectural configuration using pre-defined circuit cells for the building blocks.

2) The chip area, power and speed requirements are minimised based on *optimization criteria* at the circuit design level which are abstracted to the higher design levels, i.e., the algorithm and the architecture (Fig. 1).

An effective CAD system is created by combining efficient computer tools to automate the procedures 1a–1c, together with specialized computer-aided optimization programs to carry out the optimization tasks defined by 2). Such a CAD system (Fig. 1) leads to a simultaneous reduction of development costs as well as of fabrication costs.

In this paper, we demonstrate a CAD system called CATHEDRAL (Fig. 1) which is based on pipelined bit-serial architectures [4]–[7]. A brief overview of the design process with this architecture is given in *Section 2.1.* Extensive design experience [4], [12] shows that both wave digital lattice filter structures [14] as well as cascade-canonic second-order sections [15] are well suited for integration with pipelined bit-serial architectures. In this paper, we present the case-study of a PCM-FDM transmultiplexer designed with wave digital lattice filters [1]. For the synthesis (step 1a) of these filters the CATHEDRAL design system (Fig. 1) uses the FALCON program [14] described in *Section 2.2.* Optimization criteria at the filter algorithm level which minimize chip area and maximize throughput are discussed in Section 2.3. As shown in *Section 2.3,* to carry out this optimization CATHEDRAL (Fig. 1) employs the general purpose filter analysis program DIGEST [16] together with the coefficient optimization program CANDI [21]. For the implementation of the optimized filter algorithm in a bit-serial architecture (Step 1b), comprehensive procedures have been formulated in [4], [12]. In the CATHEDRAL system (Fig. 1) these procedures are automated by the knowledge-based architecture synthesis program AMAI which is described in *Section 2.4.* For optimizing the architecture with respect to circuit area and speed requirements the analytic tool COMPASS, which is presented in Section 2.4, has been integrated in the CATHEDRAL system (Fig. 1).

The silicon integration (step 1c) of the bit-serial architecture requires a library of integrated circuit cells which can implement the building blocks of the bit-serial architecture (Section 2.4). These cells can be designed for different technologies and with different circuit techniques. In *Section 2.5* we present a 5-μm single-metal CMOS cell library which is available in the CATHEDRAL design system. This cell library is designed with a special race-free dynamic CMOS circuit technique called NORA [17], [39]. The procedures for the computer-aided layout of the bit-serial architecture depend on the layout style of the cell library which is in turn influenced by the technology. For the single-metal CMOS cell library in CATHEDRAL the standard cell placement and routing program CAL-MP [18] is used (Fig. 1). The fundamental techniques behind most of the above mentioned CAD tools and cell library have been described individually in [14], [16]–[19], [21], [31], [29], [36], [37]. In Section II we illustrate how these have been integrated in the CATHEDRAL design system.

The CAD tools and cell library discussed in Sections 2.2–2.5 are applied in *Section III* to design a VLSI device for the transmultiplexer. In *Section 3.1.* the system level design of the transmultiplexer is described. It is shown how the knowledge of the target architecture can be exploited by the system designer to optimize the system configuration for minimum chip area. This optimization is performed manually. In *Section 3.2* the filter synthesis and analysis programs of Section 2.2 are applied. *Section 3.3* shows the optimization results with the computer-aids of Section 2.3. The synthesis and optimization of the bit-serial architecture for the transmultiplexer is demonstrated in *Section 3.4* using the programs described in Section 2.4. Finally in *Section 3.5* the complete layout of the transmultiplexer is generated with the cell library and layout techniques presented in Section 2.5.

II. Computer-Aided Design System: Cathedral

In this section we give an overview of the methodology used in the CATHEDRAL design system. The main design levels are illustrated in Section 2.1. The complete design process using the individual CAD tools and the cell library is explained in Sections 2.2–2.5.

2.1. Short Overview of the Bit-Serial Design Process

Based on the design experience gained with the manual designs of a large number of custom filter chips, a design methodology for bit-serial architectures has been developed. It is described in detail in [4], [12]. From this design methodology a set of formal rules has been derived which can be used by a designer to synthesize a bit-serial architecture, given the algorithmic description of the filter represented by a block diagram, as in Fig. 2(a). As a simple example of this methodology, Fig. 2 demonstrates the

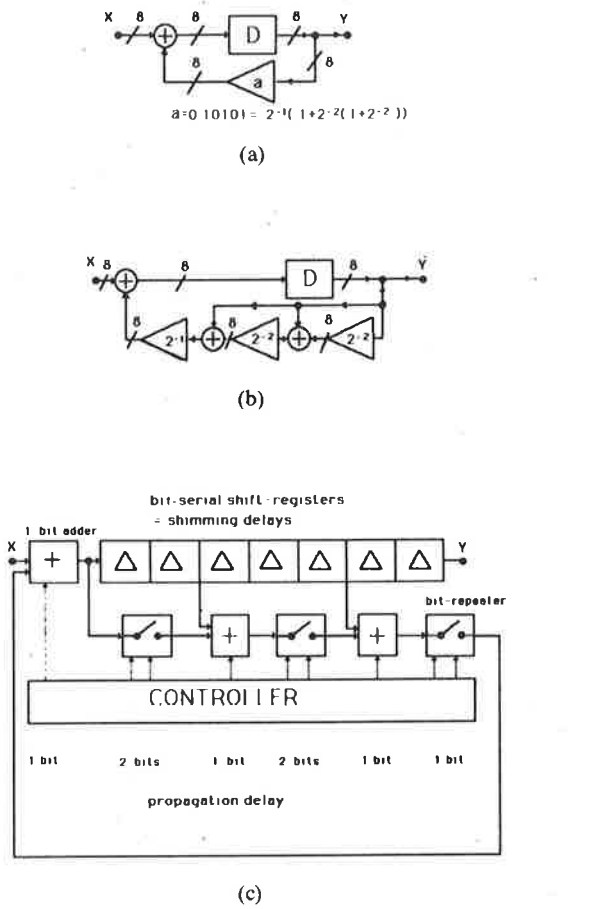


Fig. 2. Overview of design process for integrating a digital filter with a bit-serial architecture using CMOS standard cells: (a) "optimized" block diagram of a first-order filter with discrete coefficient $a = 0.10101$ and signal wordlength $W = 8$ bits; (b) Expansion of the discrete coefficient into power-of-two scalers and adders; (c) Bit-serial architecture for filter block diagram shown in (b); (d) Layout of architecture in (c) using the NORA CMOS cell library and CAL-MP; (e) Transistor schematic of a subtractor cell; (f) Layout of the subtractor cell in (e).

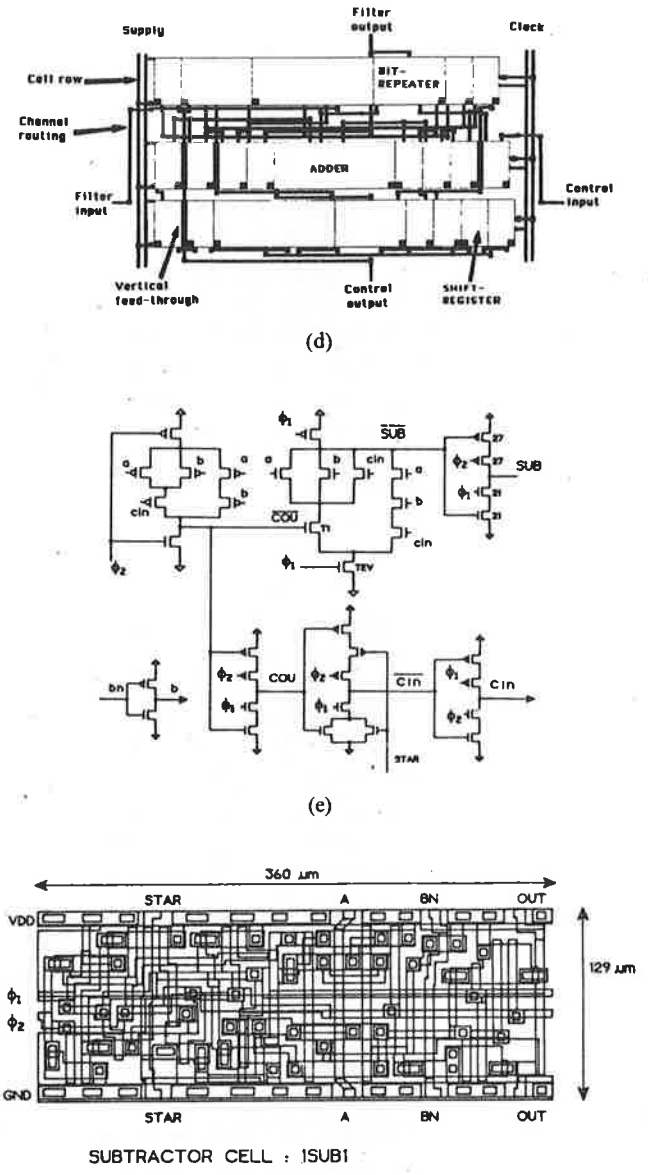


Fig. 2. (Continued).

integration of a first-order recursive filter. Fig. 2(a) shows the block diagram of a first-order filter with a binary coefficient $\alpha = 0.10101$. The signal wordlength is assumed to be-8 bits. The fixed nature of the coefficient is exploited by expanding the multiplication operation into power-of-two scaling (shift) and add operations as in Fig. 2(b). The synthesis of the expanded block diagram of Fig. 2(b) into a bit-serial architecture is given in Fig. 2(c). The architecture is constructed by interconnecting 1-bit pipelined building blocks, such as 1-bit adders, shift-registers, zero-injectors (used to perform a scaling with a positive power of two), and bit-repeaters (used to do a scaling with a negative power of two). A *controller* is added to the architecture to generate appropriate control signals to ensure correct synchronization of these bit-serial building blocks. The circuit and layout descriptions of cells which implement the building blocks of the bit-serial architecture are stored in a predefined cell library (Section 2.5). The layout technique used to integrate the architecture of Fig. 2(c) with the cell library depends on the technology of the cell library. For the CMOS single metal cell library presented in this paper a standard cell layout approach based on channel routing is adopted. For the architecture of Fig. 2(c) this results in the floorplan shown in Fig. 2(d).

2.2. *Filter Synthesis and Analysis: FALCON-DIGEST*

For a given specification on the frequency response of a digital filter (e.g., Fig. 7) different filter structures are available. The integration of 3 types of filter structures with bit-serial architectures has been investigated in [4], [12], namely wave digital [23], lossless discrete integrator (LDI) [24] and cascade canonic second-order sections [15].

These investigations show that the inherent concurrency in the wave digital *lattice* filter structure [14] (Fig. 6) leads to an efficient pipelining of the bit-serial architecture. Further, the low sensitivity, which is a common property of all wave digital structures, allows considerable reduction in the hardware for coefficient-signal multiplication (Section 2.3). In this paper we demonstrate the application of wave

digital lattice filters in a VLSI design (Section III). To design these filters the filter synthesis program FALCON [14] is used. This CAD tool can design wave digital lattice filters to satisfy arbitrary specifications in the frequency domain. The designer does not have to take care of the details of the synthesis method. With a minimal amount of interactive user-input, namely the filter type, cutoff frequency and stopband attenuation, FALCON automatically computes the required filter degree and the nominal "infinite precision" values for the coefficients of the cascaded "2-port adaptors" of the lattice structure (Fig. 6). The output of FALCON is a textual description of the filter block diagram (Fig. 6) consisting of the basic digital filter elements, i.e., adders, coefficient-signal multipliers and sample delays (D).

VLSI filtering systems such as the transmultiplexer (Section III) consist of several filter stages. The synthesis of each filter stage is usually done individually with a program like FALCON. Therefore, in the next design step the designer has to verify that the overall frequency specifications are satisfied when the filter stages are connected together to form the complete system. To do this the system transfer function is computed using the versatile filter analysis program DIGEST [16]. The network description of the filter block diagram obtained from the synthesis program FALCON is directly entered into DIGEST. The DIGEST program is also used to compute analysis information which is essential for optimization of the filter discrete coefficients and digital signal word-length as discussed in Section 2.3. DIGEST performs the network analysis directly in the z-domain [25] and exploits sparse matrix compaction techniques [26]. This leads to a high accuracy, numerical reliability and a high computation efficiency. To maintain a high efficiency for multirate VLSI filtering systems such as the transmultiplexer (Section III) special data structures are combined with efficient matrix search strategies as described in [27]. DIGEST also allows for time-domain simulation using a bit-true representation of signals, coefficients, and quantization characteristics of arithmetic operations.

2.3. Algorithm Optimization: DIGEST-CANDI

Having verified the correctness of the nominal filter design, the next step in the computer-aided design is to determine optimal values of the discrete coefficients and the signal wordlength (Fig. 2(a)), which minimize the silicon area required and maximize the throughput. For the bit-serial architecture this requires a *minimization of the nonzero bits* in the *discrete filter coefficients*, subject to the constraints imposed by the frequency specifications [28]. There are 3 nonzero bits in the discrete coefficient α in Fig. 2(a) and this directly contributes 2 adders and 3 bit-repeaters in the floorplan (Fig. 2(d)). Thus a *minimization of the nonzero bits* in the filter coefficients minimizes the chip area. The discrete coefficient values also determine the frequency response so that the minimization must be carried out under constraints imposed by the design specifications. In order to solve this discrete coefficient optimization problem the CAD tool CANDI [21] exploits an initial set of network analysis data from DIGEST (Section 2.2) to carry out large-change multiparameter sensitivity analysis [29] of the given filter. This capability is used to compute approximations to the *acceptable region* of coefficient values, which satisfy specifications on the filter response. Heuristic strategies [21] search for the discrete optimum within this region.

For bit-serial architectures, if F_c is the maximum clock rate at which the bits are propagated through the pipelined bit-serial building blocks (Fig. 2(c)) and W is the signal wordlength then the maximum possible throughput is Fc/W. So for a given clock rate the throughput is inversely proportional to the number of bits representing the internal signals. On the other hand, the total number of shift register blocks required in the bit-serial architecture (Fig. 2(c)) is directly proportional to the signal wordlength W. Hence a *minimization of signal wordlengths* maximizes the throughput/unit area. The signal wordlength also determines the signal/noise ratio, dynamic range and relative amplitude of limit-cycle oscillations. To minimize signal wordlengths within the specified constraints on these characteristics, the DIGEST program uses special techniques [16], [36], [37] to compute: (1) noise transfer functions, (2) L1-norm in the time-domain of the signals at various nodes in the system, (3) average signal energy (L2-norm) and (4) bounds on small scale limit cycles. These data are used to construct *templates* [16], which visualize the function of each bit in the digital signal at each node of the filter. The template of the signal at a node (Section 3.3) clearly indicates the maximum number of bits on the LSB side, as well as on the MSB side, which can be affected by each nonlinearity (e.g., product truncation, addition overflow) in executing the algorithmic operations. These templates are directly used to compute the minimum increase in signal wordlength necessary at each node to satisfy the specifications on the dynamic range, noise and parasitic oscillations.

The block diagram of the filter with optimised discrete coefficients and digital signal wordlength (Fig. 2(a)) is called the *optimized block diagram*.

2.4. Bit-Serial Architecture Synthesis and Optimization: AMAI-COMPASS

To integrate the optimized block diagram (Fig. 2(a)) of the filter, obtained with the CAD tools of Section 2.3, a bit-serial architecture (Fig. 2(c)) is constructed using a restricted set of building blocks, which can implement the operations of the optimized block diagram (Fig. 2(a)). This design task is called *architecture synthesis*. Rules for synthesizing bit-serial architectures have been formulated based on practical design experience. These rules are coded in a knowledge-based CAD program called AMAI, which automatically carries out the architecture synthesis. Once the architecture has been synthesized the optimization program COMPASS is used to minimize the total number of building blocks in order to reduce chip area. The design process with AMAI-COMPASS is described below.

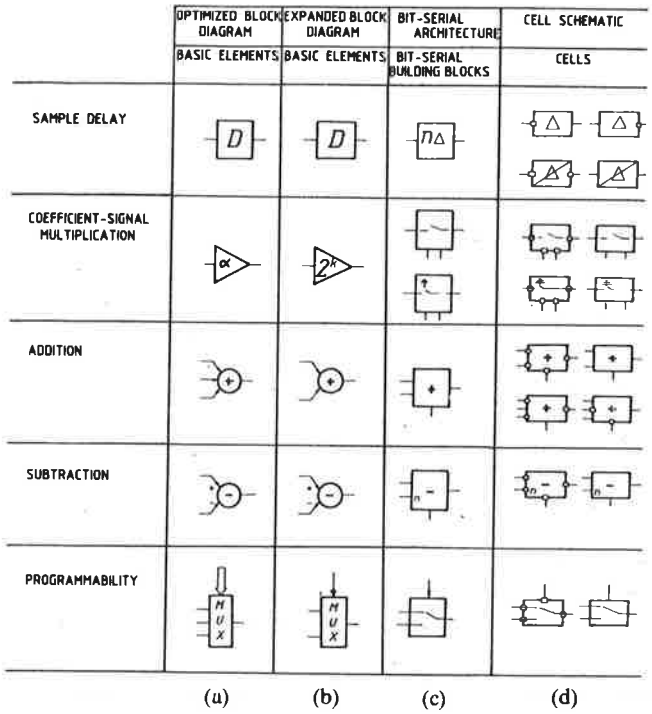


Fig. 3. The primitives used for implementing a digital filter (Section 2.4). (a) 5 basic elements of the optimized block diagram (Fig. 2(a)). (b) 5 elements of the expanded block diagram (Fig. 2(b)). (c) 6 bit-serial building blocks used to implement the 5 elements in (b) in a bit-serial architecture (Fig. 2(c)). (d) 16 NORA CMOS circuit cells to integrate the building blocks in (c).

The optimized block diagram of the digital filter (Fig. 2(a)) may in general contain 5 different basic elements, as shown in Fig. 3(a); these are (i) sample delays, (ii) coefficient-signal multipliers, (iii) adders, (iv) subtractors, and (v) multiplexers. The adders and multiplexers may have any number of inputs, and the coefficient-signal multiplier may have any binary value. To integrate the filter block diagram with a bit-serial architecture it should eventually be translated into an interconnection of integrated circuit cells which can implement the building blocks of the architecture. To perform this task AMAI and COMPASS carry out a step-wise refinement of the optimized block diagram (Fig. 2(a)) as follows:

Step 1: Expanded Block Diagram

In the first step the elements of the optimized block diagram (Fig. 2(a)) are expanded into simpler elements (Fig. 2(b)) according to the following rules which are programmed in AMAI:

Rule E1: A sample delay element of the optimized block diagram remains a sample delay element in the expanded block diagram.

Rule E2: A coefficient-signal multiplier of the optimized block diagram is expanded into a network of 2-input adders and power-of-two scalers in the expanded block diagram. Only the nonzero bits in the CSD code [32] of the coefficient are implemented.

Rule E3: An n-input adder (multiplexer) of the optimized block diagram is expanded into a tree structure of (n-1) 2-input adders (multiplexers) in the expanded block diagram.

Rule E4: A 2-input subtractor of the optimized block diagram remains a 2-input subtractor in the expanded block diagram.

Observe that the sample delay element D remains unchanged in the expansion process. The resulting expanded block diagram (Fig. 2(b)) may in general contain the basic elements drawn in Fig. 3(b).

Step 2: Bit-Serial Architecture Synthesis

In this step AMAI replaces each of the basic elements of the expanded block diagram (Fig. 3(b)) by a bit-serial architecture building block (Fig. 3(c)) as follows:

Rule B1: A sample delay in the expanded block diagram is replaced by a shift register block with a shift value n equal to the signal wordlength multiplied by the number of input channels over which the filter is multiplexed.

Rule B2: A scaler with a negative power-of-two is replaced by a bit-repeater.

Rule B3: A scaler with a positive power-of-two is replaced by a zero-injector.

Rule B4: A 2-input adder is replaced by a 2-input bit-serial adder building block.

Rule B5: A 2-input subtractor is replaced by a 2-input bit-serial subtractor.

Rule B6: A 2-input multiplexer becomes a 1-bit 2-input multiplexer.

Step 3: Architecture Optimization

The architecture resulting from Step 2 is pipelined and optimized for minimum chip area by the COMPASS program as described below. Observe that the only delays in the optimized block diagram (Fig. 2(a)) are the sample delay elements D, which are replaced by shift-register blocks ($n\Delta$) in the bit-serial architecture (Step 2, Rule B1). In order to pipeline the architecture the building blocks are clocked and this results in additional delays associated with the arithmetic building blocks (adders, subtractors, bit-repeaters, zero-injectors, multiplexers).

These delays are usually a fraction of the sample delay D. In order to keep the transfer function of the filter unaltered, these extra delays must be compensated for. To achieve this, COMPASS redistributes the sample delay ($n\Delta$) in feedback loops and assigns additional unit delays (Δ) in parallel signal paths with unequal pipeline delays (Fig. 9). These additional unit delays are called "shimming" delays [31] and require the addition of extra shift-register blocks (Fig. 2(c)) which eventually consume more chip area (Fig. 2(d)) and power. Hence COMPASS performs an *optimal assignment of "shimming" delays* which minimizes the total shift register count and hence the chip area and power consumption.

To find the global solution to this optimization problem the COMPASS program [31] exploits the technique of *retiming* [34] in order to reformulate the optimal delay management problem as a *linear programming* problem [35]. It is interesting to note that notwithstanding its in-

teger programming nature, the above problem can be solved with the simplex algorithm.

Step 4: Control Generation

When the architecture has been pipelined and optimized by COMPASS, AMAI constructs a customized control structure. This controller (Fig. 2(c)) is essentially a chain of shift-register blocks. It distributes appropriately delayed control pulses to the bit-serial building blocks (Fig. 2(c)) which need synchronization with the LSB of each successive sample. Such control pulses are used to initialize a building block for each sample, e.g., carry initialization of adders for LSB addition.

Step 5: Cell Schematic

In the final step AMAI translates the complete architecture (Fig. 2(c)) into a cell schematic, which is an interconnection of library cells. This cell schematic is generated using a cell library (Fig. 3(d)) as follows:

Rule C1: A shift-register block with a delay in $n\Delta$ is replaced by a cascade of shift-register cells.

Rule C2: A bit-repeater (zero-injector) is replaced by a bit-repeater cell (zero-injector cell).

Rule C3: A bit-serial adder block is replaced by an adder cell. If no carry input is required, an adder cell without carry input is selected.

Rule C4: A bit-serial subtractor block is replaced by a subtractor cell.

Rule C5: A multiplexer building block is replaced by a multiplexer cell.

In the NORA CMOS cell library used in this paper (Section 2.5) two different options exist for each cell as shown in Fig. 3(d). One option is clocked on phase ϕ_1 and the other option is clocked on phase ϕ_2. When applying the above rules (C1–C5) it is important to ensure that neighboring cells in the cell schematic are clocked on different phases. This guarantees a race-free operation [39] of the circuit. When executing rules C1–C5, the AMAI program automatically makes an appropriate choice of the cell option in order to satisfy the above condition.

The output of the AMAI program is the cell schematic in the form of a netlist of the library cells.

2.5. Silicon Integration: NORA-CALMP

The cell schematic of the optimized architecture, obtained from AMAI-COMPASS is entered in the database of the SL-2000 system [2] used for designing chips with standard cells. In the same system, the library of circuits and layout information for bit-serial cells is stored, which can be used to integrate the building blocks of the bit-serial architecture. For digital signal processing systems an important design characteristic is the fact that they share a common and restricted set of functional building blocks. In realizing several bit-serial prototype chips it has been our experience that with only a set of 6 building blocks (Fig. 3(c)) the basic operations of custom digital filters can be performed. In addition data-dependent decisions and programmable coefficients can be handled and testability can be provided. The establishment of a cell library to integrate these building blocks allows considerable reduction in the design time of a wide range of complex VLSI filtering devices. Also technology updating or adaptation to different silicon foundries is easy to achieve.

NORA [17] is the CMOS circuit technique employed to design the cell library [39]. In this technique, the combination of static, dynamic and C^2MOS circuits following a restricted set of composition rules, results in a racefree system, regardless of the amount of skew between the two synchronizing clock signals. For bit-serial applications another important characteristic of the NORA technique is the reduced silicon area required by the C^2MOS register. This allows a significant area saving compared with other techniques (e.g., DOMINO CMOS), because all bit-serial operators incorporate register elements to facilitate pipelining. As an example, Fig. 2(e) shows the NORA circuit of a bit-serial two's complement subtractor.

The cells have been designed for a 5-μm single-metal CMOS process. Since only a few reusable cells are required the layout of the cells has been done manually, making use of the Applicon interactive graphics system. The use of layout geometries at orientations that are integer multiples of 45 degrees made it possible to realize very dense layouts (average density = 745 tr/mm^2). Fig. 2(f) shows the layout of the subtractor cells of Fig. 2(e). Simulations indicate a maximum operating frequency of 10 MHz with a total capacitive load smaller than 1.5 pF. This has been verified by measurements on test chips [39].

The technology has a large impact on the layout strategy. For the single-metal CMOS technology it was found that, due to the lack of buried contacts, anti-latch-up design and high values of junction capacitance, routing outside cells is necessary. For this purpose the standard cell placement and routing program CAL-MP [18] is used. Clock, power and ground lines are interconnected by abutment. Interconnections between cells run horizontally in a dedicated routing channel parallel to the cells (Fig. 2(d)). Wherever necessary, "feeds" are inserted for the vertical connections (Fig. 2(d)). In each cell, contact terminals for datapath as well as control-supply connections are provided on the top as well as on the bottom side. The facilities for abutment between adjacent cells in the current library, allow the construction of serial register chains.

The NORA CMOS cells are stored in the SL-2000 database [2], which can be used to integrate the building blocks of the bit-serial architecture. The database contains both circuit and layout descriptions of each cell. Given the netlist of library cells obtained from the architecture synthesis program AMAI (Section 2.4), the SL-2000 database allows the designer to expand the design description down to two levels of abstraction. In order to carry out the cell placement and routing using CAL-MP [18], the netlist description is expanded down to the layout level by including geometrical information of the individual cell layout such as physical location of terminals. With this information CAL-MP does an automatic placement of the cells such that the channel widths (Fig. 2(d)) are minimized. This results in a floorplan such as shown in Fig. 2(d) for the filter of Fig. 2(a).

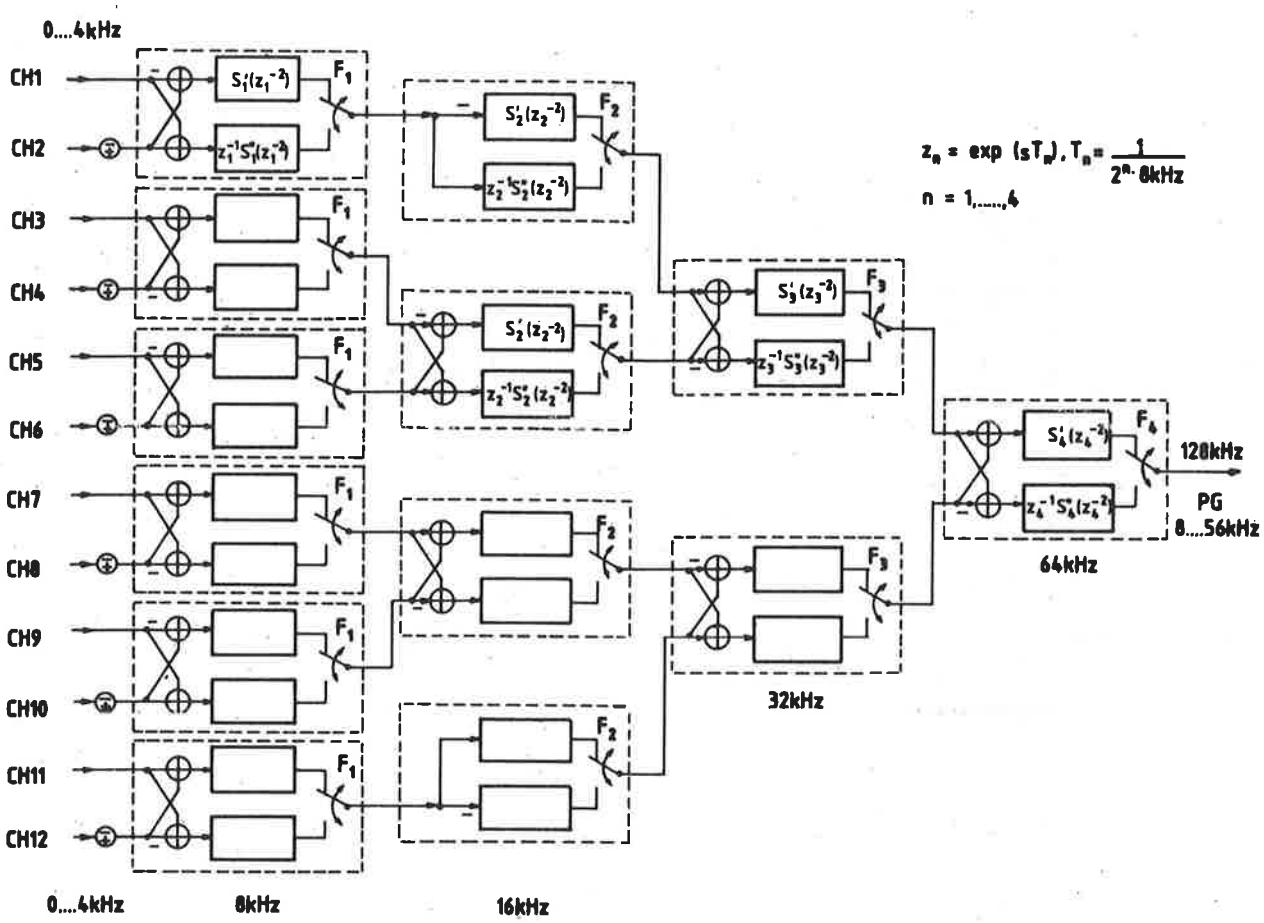


Fig. 4. System concept for PCM-FDM transmultiplexer using multirate branching filters.

Secondly, for the purpose of logic simulation using the BIMOS simulator [2], the SL-2000 database can expand the netlist description down to the transistor level by including information about the cell circuit (Fig. 2(e)). Since the input descriptions for both CAL-MP and BIMOS are generated by the same database, there exists a one-to-one correspondence between the simulation and the actual layout. The logic simulation with BIMOS is used to provide reference test patterns for later testing of the fabricated chip.

Finally, before a mask tape is created with the complete layout as generated by CAL-MP, the program LOADS [2] calculates the total interconnect capacitance in each net. This information can be used to verify whether the sampling rate requirements of the filter will be met. If this is not the case buffers have to be introduced in the bit-serial architecture (Section 2.4) to drive critical nets.

III. Transmultiplexer Design

In this section we apply the CATHEDRAL design system of Section II to design the layout of a VLSI transmultiplexer. Nossek *et al.* [1] have proposed a system design for a PCM-FDM transmultiplexer using multi-rate wave digital lattice filters. In *Section 3.1* the system level design of the transmultiplexer [11] is briefly described. Next, in *Section 3.2* and *Section 3.3* the filter synthesis and optimization is performed as described in Sections 2.2–2.3. For the integration, a customized bit-serial architecture is constructed in *Section 3.4* using the AMAI program (Section 2.4). In *Section 3.5* the layout is implemented with the library of 5-μm CMOS standard cells (Section 2.5). The results with the CAD system of Section II are discussed and compared with a fully manual design [11].

3.1. System Level Design

The original system design [1] is essentially a cascade of filter banks, operating at different sampling rates (Fig. 4). The range of sample rates varies between 8 kHz to 128 kHz, which is still lower than the maximum throughput capability of a hardwired bit-serial architecture, using the 5-μm CMOS cell library (Section 2.5). Hence one can optimize the use of silicon area, by *multiplexing hardware* among identical filters, which process different inputs. This, however, requires a completely modular filter configuration. To meet this condition a transformation [11] of the original system schematic [1] is carried out. The transformed system (Fig. 5) consists of 4 multiplexed wave-digital lattice filters (Fig. 6) operating at the same sample frequency. Compared to the original system (Fig. 4) this essentially involves a reorganization of the input channel sequences and the addition of "dummy" channels. Multiplexing the same hardware among N different input channels, only requires each sample delay to be repeated N times. Consequently, the total shift register count remains the same,

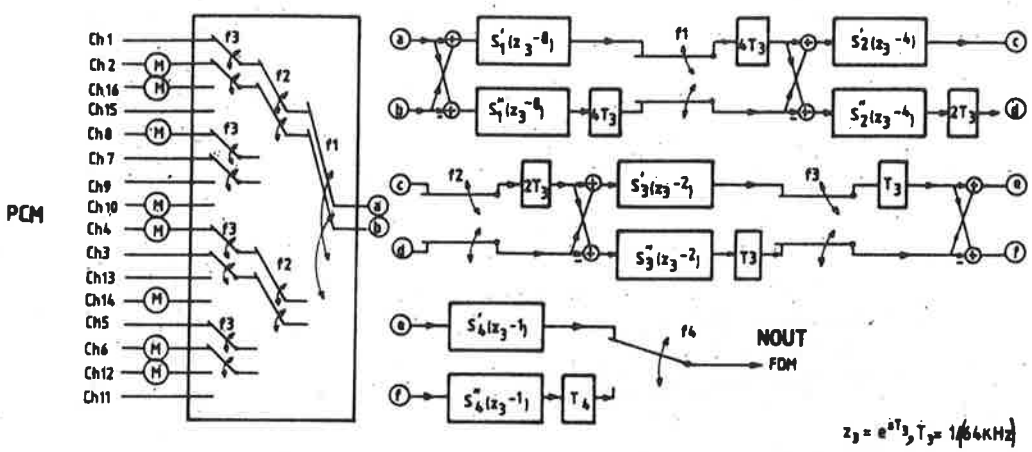


Fig. 5. Block diagram for tranformed transmultiplexer, obtained from Fig. 4 (after extension to 16 input channels) by sharing filter hardware among the channels.

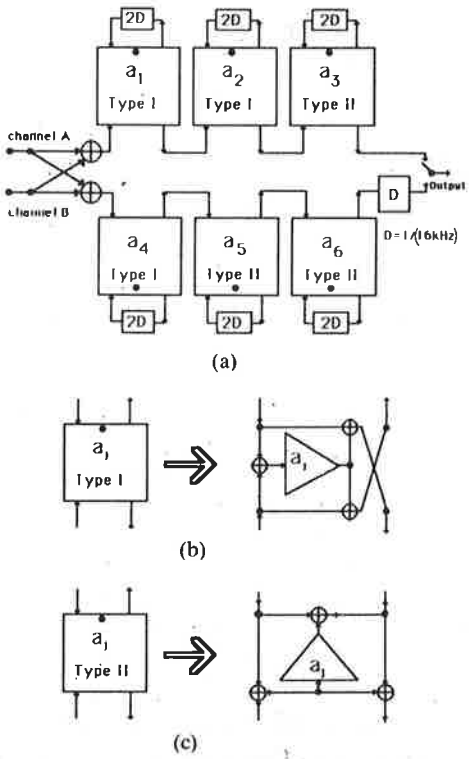


Fig. 6. Block diagram or network of 13th-order wave digital lattice filter used in the transmultiplexer design. The structure (a) is composed of 2-port adaptors as shown in (b) and (c). The FALCON program computes a nominal value for the coefficient a_j of the jth adaptor.

whereas the computational operators have to be implemented only once, resulting in *area savings up to 35 percent*. Additional circuitry has to be provided to satisfy the modified timing requirements. However, this overhead is negligible compared with the overall savings. The above transformation is an example of optimization at the system level, which directly reduces the chip area.

Various concepts have been described in the literature for realizing digital transmultiplexers [40], [41]. The system architecture used here [1] employs a single-path conversion scheme, which has an increased immunity to intelligble cross-talk. Further, the use of wave digital filters provides absolute stability under looped conditions [42].

3.2. Filter Synthesis and Analysis

The frequency-domain specifications for the wave-digital lattice filters of the respective filter stages (Fig. 5) are derived from the system specifications [1]. The synthesis program FALCON [14] computes the required degree and the nominal "infinite precision" values for the coefficients of the cascaded two-port adaptors of the lattice structure (Fig. 6). For the transmultiplexer, filters of 13th, 15th, 19th, and 21st degree are required to satisfy the specifications. The signal flow graph of the 13th-degree filter is shown in Fig. 6. FALCON generates a complete block diagram description of the filters. This task requires only 2 CPU-minutes (VAX780/VMS).

The next step is to analyze the response of the entire transmultiplexer (Fig. 5) constructed with the filters as designed above (Fig. 6) in order to verify that the overall system specifications are satisfied. The general digital filter evaluation program DIGEST [16] requires only 12 CPU minutes (VAX780/VMS) to compute the global system response (Fig. 7) over the full Nyquist range for 320 frequency points. The response shown in Fig. 7 corresponds to an excitation of the first channel (Fig. 5) which extends from 0 to 4 kHz in the FDM baseband. The aliasing effects in the other channels caused by this excitation represent the intelligible and unintelligible crosstalk levels. According to one of the CCITT-norms [10], these have to be attenuated by at least 60 dB, which is satisfied in Fig. 7.

3.3. Filter Optimization

Given the nominal design of Section 3.2, the optimization of discrete coefficients with CANDI (Section 2.3) reduces the total nonzero bits and hence the total shift-and-add operations by more than 30 percent at the cost of 1 CPU-hour (VAX780/VMS). The corresponding reduction in chip area is another 10 percent (Table I). The response with the discrete optimized coefficients computed by CANDI is shown in Fig. 8. One can observe that the tolerance limits have been exploited to reduce the nonzero bits. Further details of the optimization with CANDI can be found in [9].

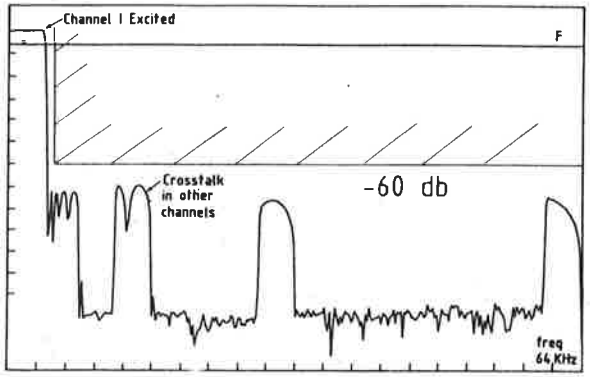


Fig. 7. Nominal response over 320 frequency points for transmultiplexer system of Fig. 5, analyzed by the DIGEST program using a nominal coefficient values as computed by FALCON.

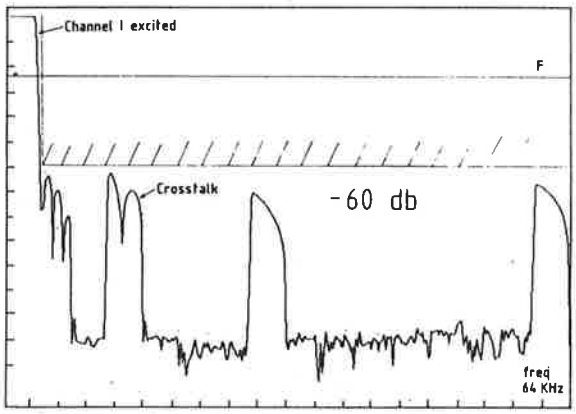


Fig. 8. Frequency domain response over 320 points for transmultiplexer system of Fig. 5, computed by DIGEST using optimized discrete coefficients obtained with CANDI.

The optimal signal wordlength W is determined with the aid of the templates generated by DIGEST (Section 2.3), as shown below for the FDM system output node Nout (Fig. 5) with $W=17$:

$$S\ S\ L\ \#\ \#\ \#\ \#\ \#\ \#\ \#\ \#\ \#\ \#\ \#\ R\ R\ R$$

(MSB) $\longrightarrow$ W bits $\longrightarrow$ (LSB)

For the statistical noise contribution at the node Nout (Fig. 5) due to the product truncation, an RMS-value of 4.9 quantization units is obtained. Hence, on the average less than 3 bits (letter R) are corrupted. For the overflow scaling analysis, the L1-norm in the time-domain (letter L) is computed to obtain a bound on the maximal amplitude gain factor in the presence of signal excitations whose amplitude is in the interval $[-1,1]$. In order to avoid overflow, the MSB (sign-bit; letter S) should not be affected. The worst-case value of L1 occurs for the output of the first section (Fig. 5), namely $L1=1.89$. Hence, to guarantee the absence of overflow conditions, the input signal level can be maintained up to the 15th-bit position (first value-bit). The computed L1 norm for Nout becames $L1=0.49$, and consequently only the bits up to the 14th position are contributing to the dynamic range at Nout, as represented in the figure above. Also the average energy can be computed (indicated with #). Combined with the noise-figure, the signal-to-noise ratio at Nout becomes 66 dB. When the special nature of the applicable CCITT-norm is taken into account (sophometric noise weighting and specified input signal levels), the specification is met for $W=17$. Each bit which can be dropped, corresponds to a reduction of the shift register lengths (Fig. 2(c)) and the area occupied by the register hardware decreases proportionally. In the original design [1] 20 bits were foreseen for W, so with the aid of the templates computed at a cost of 25 CPU-hours (VAX780/VMS), 3 bits can be saved. Consequently, the final active chip area is reduced by about 12 percent.

TABLE I
DISCRETE COEFFICIENT OPTIMIZATION OF TRANSMULTIPLEXER FILTERS WITH THE CAD PROGRAM CANDI.

Filter	No. of coefficients	Straight-forward rounding (non-zero bits)	CANDI optimum (non-zero bits)	cpu minutes VAX 11/780
F1	6	19	14	3.3
F2	7	31	20	7.1
F3	9	38	25	12.0
F4	10	39	27	25.0

Total reduction in number of non-zero bits=30%

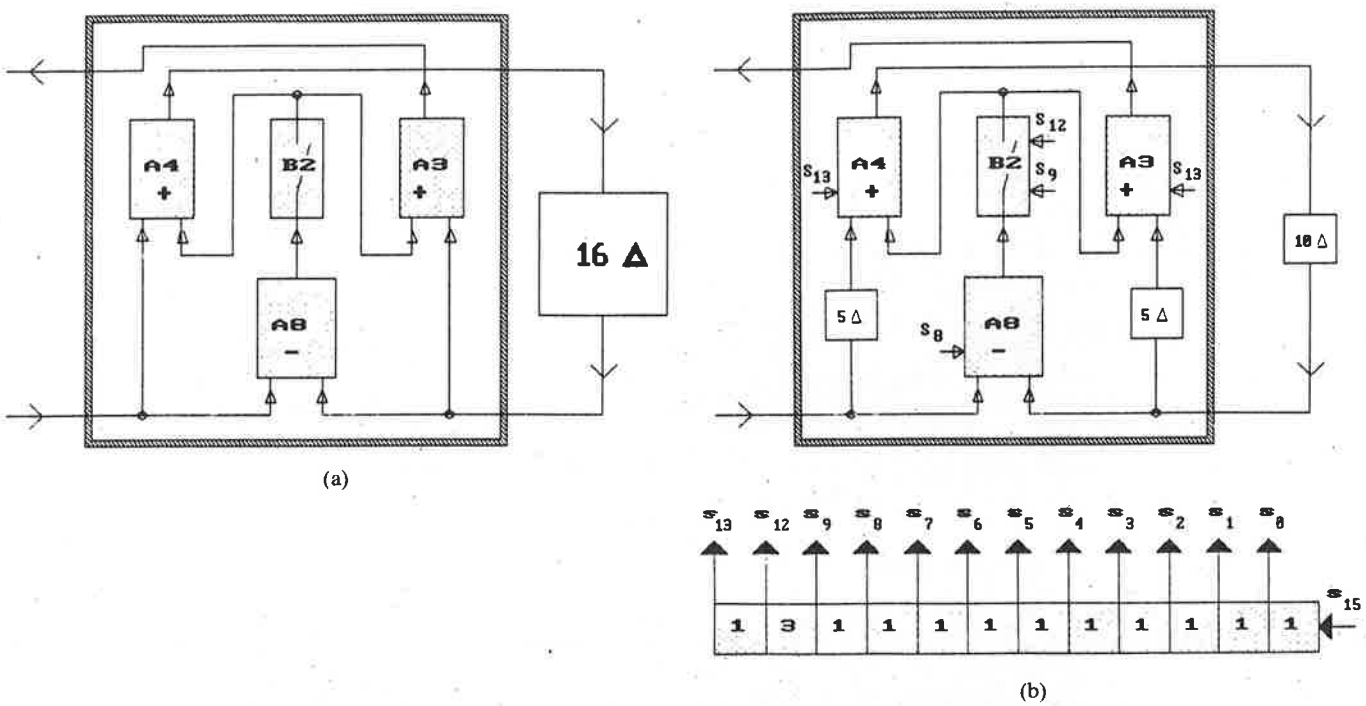
=> Total reduction in bit-serial chip area=10%

3.4. Bit-Serial Architecture Synthesis

The architecture synthesis program AMAI requires about 1 CPU-hour (VAX780/VMS) to "map" the optimized block diagram of the transmultiplexer filters (Section 3.3) into an interconnection of bit-serial architecture building blocks (Section 2.4). Current investigations are going on to automate the exploitation of the inherent system hierarchy (Fig. 5), as proposed in [11], where a small number of parametrizable building blocks, suffice for the description. This can reduce both the CPU-time and the dynamic memory requirements for executing AMAI considerably. In the next step, AMAI employs the COMPASS program to optimally assign shimming delays (Fig. 9) which are required to compensate for the propagation delay in the bit-serial operators (Section 2.4). This minimizes the total amount of additional shift registers, which in turn reduces the chip area further by 10 percent. The output of AMAI is a netlist description of the entire optimized architecture in terms of the CMOS library cells (Section 2.5). This forms the basis for the floorplan and layout design.

3.5. Integrated-Circuit Layout

The floorplanning task with the cell library of Section 2.5 has been carried out with the aid of the CAD tool CAL-MP [18], which directly accepts the netlist of stan-



(a)

(b)

Fig. 9. Bit-serial architecture of a 2-port adaptor (Fig. 6(b)) with discrete coefficient $a = 0.001$ and with one of its ports terminated in a delay element. (a) Synthesis of the architecture by AMAI using the building blocks in Fig. 3(c); (b) optimal assignment of shimming delays by COMPASS.

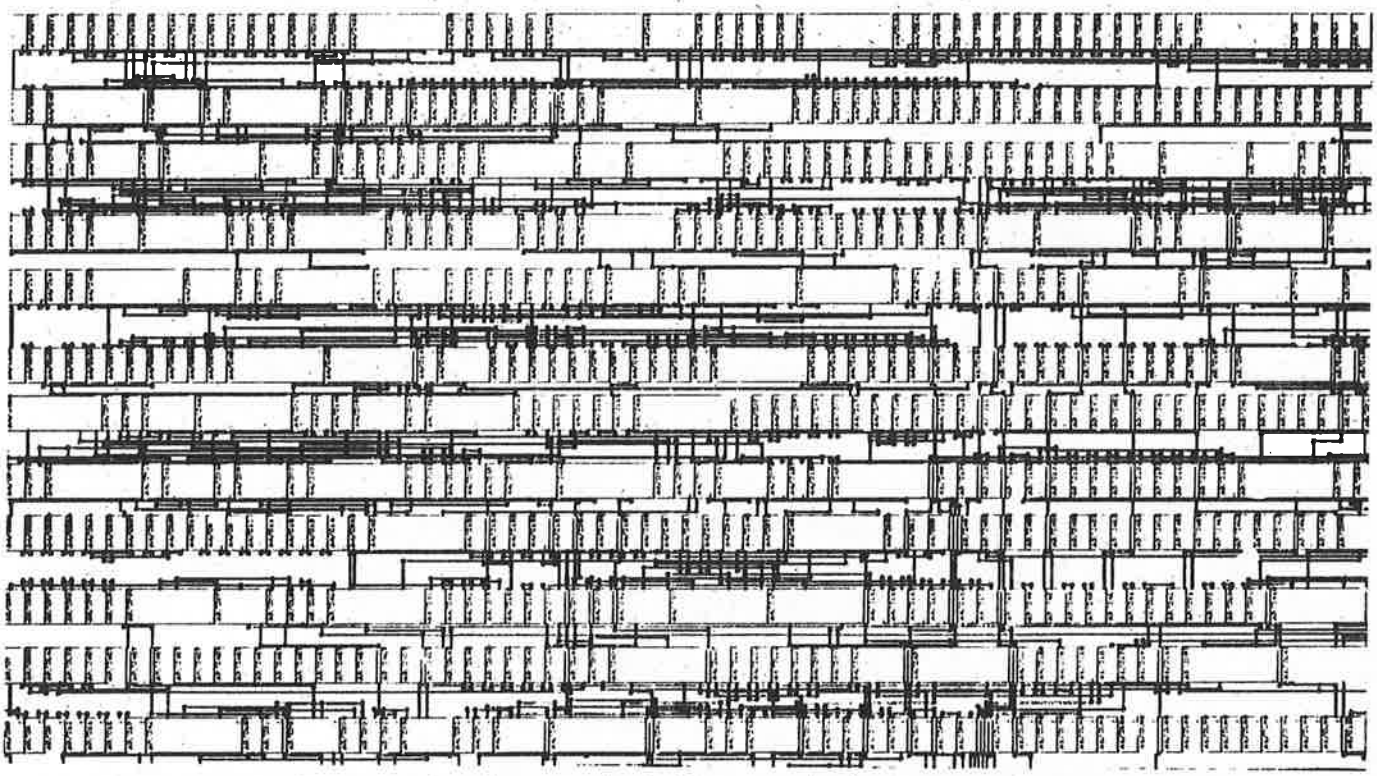


Fig. 10. Part of the transmultiplexer floor plan designed with CAL-MP showing 12 out of the 24 cell rows. The length of illustrated portion is about 40 percent of the total length and the illustrated width is about 50 percent of the total width. Bonding pads are not shown.

dard cells from AMAI (Section 3.4). An optimized placement combined with a 100 percent routing completion, has been achieved with a minimal amount of user intervention. In order to reduce the number of "independent" elements which have to be placed, delay clusters have been connected in a chain in advance, and defined as one entity, namely a "macro" in CAL-MP. Even with this simplification nearly 1600 components have to be placed. These include 1355 registers and register-chains, 158 adders and subtractors, 86 bit-repeaters and 7 multiplexers. About 6

iterations of placement, optimization and routing are required to arrive at the final floorplan of Fig. 10. For this task, CAL-MP consumes 2.5 CPU-hours in total on a VAX750/VMS. The floorplan (Fig. 10) contains 24 cell rows and 167 tracks covering 67 mm^2. The overall density is 435 transistors/mm^2. A comparable manual design would require only 58 mm^2 with a density of 500 transistors/mm^2 [11]. However, the time for manual placement and routing is estimated at 1 man-month.

Hence the automation with CAL-MP saves a substantial amount of design time and consumes only 10 percent more silicon area. The reasons for the area advantage of the manual design have been investigated in detail. Current research efforts are aimed at enhancing the CAD system so that the designer can interactively come arbitrarily close to the area efficiency of the fully manual custom design depending on the design time he is willing to invest.

The lower chip area of the manual design of the transmultiplexer [11] is due to two reasons. Firstly, the manual approach makes use of dense shift register matrices, which are important since nearly 80 percent of the active chip-area is comprised of shift register cells. In the automated strategy with CAL-MP, these register matrices are not allowed yet, as they do not fit into the adopted standard cell approach of single cell-rows connected with tracks running in dedicated channels. The main difference between the automated and manual designs, however, results from an efficient strategy for exploiting the system hierarchy in the manual approach [11]. This can be mimicked in CAL-MP with a limited amount of overhead in design time. In CAL-MP a feature is included to define clusters or "classes" which are initially placed together in a restricted range. In fact in the mapping tool AMAI, a possibility exists to generate classes by specifying the nodes located on the border of the cluster (cut set). However, this approach does not yet allow the full flexibility present in the manual placement. Therefore, current investigations are going on with the transmultiplexer test-vehicle, to evaluate the CAD tool GRIP [2], which allows preplacement of the components in little groups of arbitrary size. The necessary routing is then generated by the program. With this facility the designer can speed up the design time while interactively approaching the result of a full custom design.

IV. Conclusion

The top-down realization of VLSI filters from system specification to silicon integration with a computer-aided design system is demonstrated by the design of a monolithic PCM-FDM transmultiplexer device. On this device 184 orders of filtering with 92 coefficients are integrated. The design system called CATHEDRAL consists of a comprehensive set of CAD tools and a 5-μm CMOS cell library (Fig. 1), which support a bit-serial architecture methodology. These result in a fast turnaround time as well as in a highly efficient use of the silicon area. The design time with CATHEDRAL amounts to about 1 week as opposed to 2–3 man-years for a fully manual design. At the same time the chip areas of the computer-aided and manual designs are comparable. The CATHEDRAL design system results in a chip area of 67 mm^2 with a complexity of nearly 30 000 transistors. When scaled down to a 2.5 micron technology less than 25 mm^2 can be attained. This implies less than 0.15 mm^2 per order of filtering which, due to the high degree of multiplexing, is more area efficient than a switched-capacitor design. The CAD system leads to a total reduction of 67 percent in the chip area. This capability of CATHEDRAL is a result of combining a systematic design methodology with fundamental techniques to provide efficient computer-aids at every design level.

The CATHEDRAL design system presented in this paper can be used to design VLSI filtering systems with sample rates up to 500 kHz. Current research is aimed at extending the approach in this paper to develop CAD systems for bit-parallel architectures with a higher throughput and broader functional capability.

Acknowledgment

The authors are grateful to Dr. H. J. Pfleiderer for stimulating the interaction with Siemens AG which led to this work. The authors would also like to thank Dr. J. Nossek (Siemens AG) for the transmultiplexer system design, E. Van Geemert (Silvar-Lisco nv) for his invaluable help with the automated layout investigation and Dr. L. Gaszi (Ruhr Univ. Bochum) for the filter designs with FALCON.

References

[1] J. A. Nossek *et al.*, "Wave digital lattice filters with applications in communication systems," *Proc. IEEE Int. Symp. Circuits and Systems*, Newport Beach, CA, pp. 845–848, May 1983.

[2] SL-2000 overview: An integrated system for custom IC design, Doc. No. M-044-1, Silvar-Lisco, Abdijstraat 34, B-3030 Heverlee, Belgium.

[3] P. B. Denyer *et al.*, "A silicon compiler for VLSI signal processors,"in *Dig. Tech. papers, European Conf. on Solid-State Circuits*, Brussels, pp. 215–218, Sept. 1982.

[4] J. K. J. Van Ginderdeuren, H. De Man, N. Goncalves, and W. Van Noije, "Compact NMOS building blocks and a methodology for dedicated digital filter applications," *IEEE J. Solid-State Circuits*, vol. SC-18, pp. 306–316, June 1983.

[5] H. De Man *et al.*, "Custom design of hardware digital filters on IC's," in *Proc. Custom Integrated Circuits Conf.*, Rochester, NY, pp. 173–177, June 1982.

[6] S. L. Freeny, "Special purpose hardware for digital filtering," *Proc. IEEE*, vol. 63, pp. 633–648, Apr. 1975.

[7] R. F. Lyon, "A bit-serial VLSI architectural methodology for signal processing," in *Proc. VLSI Conf.*, Edinburgh, Academic, Aug. 1981.

[8] W. Ulbrich *et al.*, "MOS-VLSI pipelined digital filters for video applications," *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, San Diego, CA, pp. 44.7.1–44.7.4, Mar. 1984.

[9] R. Jain, "Computer-aided discrete coefficient optimisation for custom integrated digital filters," Ph.D. dissertation, ESAT Lab., Katholieke Universiteit Leuven, May 1985.

[10] C. F. Kurth *et al.*, "A per channel, memory oriented transmultiplexer with logarithmic processing," *IEEE Trans. Commun.*, vol. 30, 1982.

[11] F. Catthoor, "Bit-serial transmultiplexer design," ESPRIT 97 pilot phase final report, July 1985, IMEC, Kapeldreef 75, B-3030 Leuven, Belgium.

[12] J. K. J. Van Ginderdeuren, H. De Man, F. Catthoor, S. Beckers, "A design methodology for compact integration of wave digital filters," *Digest Europ. Conf. on Solid-State Circuits*, Edinburgh, pp. 210–213, Sept. 1984.

[13] H. De Man *et al.*, "A unified box of CAD tools for the design of dedicated signal processing chips," in *Proc. IEEE Int. Conf. on Computer Design*, Port Chester, NY, Oct. 84, pp. 838–844, Oct. 1984.

[14] L. Gazsi, "Explicit formulae for lattice wave digital filters," *IEEE Trans. Circ. Syst.*, vol. CAS-32, pp. 68–88, Jan. 1985.
[15] A. V. Oppenheim *et al.*, *Digital Signal Processing*, Englewood Cliffs, NJ: Prentice Hall, 1975.
[16] L. Claesen *et al.*, "DIGEST a digital filter evaluation and simulation tool for MOSVLSI filter implementations filters," *IEEE J. Solid-State Circuits*, vol. SC-19, pp. 414–424, June 1984.
[17] N. F. Goncalves *et al.*, "NORA: A racefree dynamic CMOS technique for pipelined logic structures," *IEEE J. Solid-State Circuits*, vol. SC-18, pp. 261–267, June 1983.
[18] H. Beke *et al.*, "CAL-MP: an advanced computer-aided layout program for MOS-LSI," in *Dig. Europ. Conf. on Solid-State Circuits*, pp. 67–69, 1980.
[19] R. Jain *et al.*, "CAD tools for the optimised design of custom VLSI wave digital filters," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Tampa, FL, pp. 1465–1468, Mar. 1985.
[20] F. Catthoor, R. Jain, H. De Man, and J. Vandewalle, "Custom integration of a VLSI transmultiplexer using a computer-aided design methodology based on bit-serial architectures," in *Proc. IEEE Int. Symp. Circuits Syst.*, Kyoto, pp. 251–254, June 1985.
[21] R. Jain, J. Vandewalle, and H. De Man, "Efficient CAD tools for coefficient optimisation of arbitrary integrated digital filters," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, San Diego, CA, pp. 30.11.1-4, Mar. 1984.
[22] S. Pope *et al.*, "Macrocell design for concurrent signal processing," *3rd Caltech. Conf. on VLSI*, Ed. by R. Bryant, Springer, 1983.
[23] A. Fettweis, "Digital filter structures related to classical filter networks," *Arch. Elek. Uebertragung*, vol. 25, pp. 79–89, Feb., 1971.
[24] L. T. Bruton, "Low-sensitivity digital ladder filters," *IEEE Trans. Circuits Syst.*, vol. CAS-22, pp. 168–176, Mar. 1978.
[25] J. Vandewalle, H. De Man, and J. Rabaey, "Time, frequency and z-domain modified nodal analysis of switched-capacitor networks," *IEEE Trans. Circuits Syst.*, vol. CAS-28, pp. 186–195, Mar. 1981.
[26] L. Claesen, J. Vandewalle, and H. De Man, "Efficient computer analysis of ideal switched-capacitor circuits using matrix compaction techniques," *Int. J. Circuit Theory Appl.*, vol. 11, pp. 241–264, 1983.
[27] F. Catthoor *et al.*, "SMAC: Efficient Sparse matrix compaction techniques in DIGEST," ESPRIT pilot phase final report, IMEC, Leuven, Belgium, 1984.
[28] R. Jain, J. Vandewalle, and H. De Man, "Discrete coefficient optimisation for the CAD of arbitrary integrated digital filters," in *Proc. European Conf. on Circuit Theory and Design*, 1983, Stuttgart, FRG.
[29] R. Jain, J. Vandewalle, and H. De Man, "Efficient and accurate multiparameter analysis of digital filters using a multivariable feedback representation," *IEEE Trans. Circuits and Syst.*, vol. CAS-32, pp. 225–235, Mar. 1985.
[30] W. F. Clocksin, and C. S. Mellish, *Programming in PROLOG*,. Berlin, Germany: Springer-Verlag.
[31] G. Goossens *et al.*, "An optimal and flexible delay management techique for VLSI," in *Proc. 7th Int. Symp. on the Mathematical Theory of Networks and Systems (MTNS)*, Stockholm, Sweden, June 1985.
[32] R. W. Reitwiesner, "Binary arithmetic" in *Advances in Computers*, vol. 1, Academic, 1966, pp. 231–308.
[33] M. Renfors, and Y. Neuvo, "The maximum sampling rate of digital filters under hardware speed constraints," *IEEE Trans. Circuits Syst.*, vol. CAS-28, pp. 196–202, Mar. 1981.
[34] C. E. Leiserson, and J. B. Saxe, "Optimizing synchronous systems," in *Proc. IEEE Symp. Foundations of Computer Science*, Nashville, TN, Oct. 1981.
[35] C. E. Leiserson, F. M. Rose, and J. B. Saxe, "Digital circuit optimization," MIT private communication, Apr. 1982.
[36] F. Catthoor, H. De Man, and J. Vandewalle, "SAILPLANE: A CAD tool for the Analysis of Limit-cycle Behaviour with the aid of Simulated Annealing," in *Proc. IEEE Int. Conf. on Computer Design*, Port Chester, NY, pp. 244–247, Oct. 1985.
[37] F. Catthoor, J. Vandewalle, and H. De Man, "The computation and use of Templates in multi-rate filter design," ESPRIT Interim Rep., IMEC, Leuven, Belgium, Mar. 1985.
[38] J. Pandel and W. Ulbrich, "Implementation of a single-path multiplier-free transmultiplexer scheme," in *Proc. IEEE Int. Symp. on Circuits and Systems*, Kyoto, Japan, pp. 759–762, June 1985.
[39] N. Goncalvez, NORA: A racefree CMOS technique for register transfer systems," Ph.D. dissertation, ESAT Lab., Katholieke Universiteit Leuven, Nov. 1984.
[40] H. Scheuermann and H. Goeckler, "A comprehensive survey of digital transmultiplexer methods," *Proc. IEEE*, vol. 69, pp. 1419–1450, Nov. 1981.
[41] A. Fettweis, "Transmultiplexers with either analog conversion circuits, wave digital filters or SC filters—A review," *IEEE Trans. Commun.*, vol. COM-30, pp. 1575–1586, July 1982.
[42] A. Fettweis and K. Meerkoetter, "On parasitic oscillations in digital filters under looped conditions," *IEEE Trans. Circuits Syst.*, vol. CAS-24, pp. 475–481, Sept. 1977.

Rajeev Jain received the B.Tech. degree in electrical engineering from the Indian Institute of Technology, Delhi, in 1978, and the Ph.D. degree in applied sciences from the Katholieke Universiteit Leuven, Belgium, in 1985.

From 1978 to 1980 he was a graduate engineer with Associated Instruments Manufacturers (India) Ltd. From 1980 to 1982 he was a design engineer with the Telecom Application Laboratories, Siemens AG, Munich, Germany, where he worked on the development of digital signal processing techniques and architectures for VLSI telecom devices. In 1982 he moved to the ESAT laboratory of the Katholieke Universiteit Leuven, Belgium where he completed his doctoral research. From 1982 to 1983 he had a Fellowship from the Research Foundation (Onderzoeksfonds) of the K.U.L. In Jan. 1984 he was appointed Research Associate under the ESPRIT 97 project and subsequently Group Leader for research on Architectural and Algorithmic Strategies at the Inter-University Microelectronics Centre (IMEC) Leuven. In August 1985, he moved to the Electronic Research Laboratory, University of California, Berkeley, where he is currently working as a Visiting Scholar. His current research interests are in the computer-aided design of integrated circuits for speech and image processing.

✠

Francky Catthoor was born in Temse, Belgium, on October 12, 1959. He received the Electrical Engineering Degree combined with a Master's degree from the Katholieke Universiteit Leuven (KUL), Belgium in 1982. From September 1983 he has been at KUL working towards the Ph.D. degree in electrical engineering, as a member of the research group for Digital Signal Processing of Prof. H. De Man and Prof. J. Vandewalle, first, until December 1984, at ESAT, KUL, and then, from January 1985 at the Inter-University Micro-Electronics Center (IMEC), Heverlee, Belgium.

His research activities partly belong to the domain of digital filter analysis. But he is also involved in the definition of architectural strategies for the VLSI realization of DSP-algorithms, combined with the design of test-vehicles. These topics are part of the joint ESPRIT 97 project, sponsored by the EC, in cooperation with industrial partners.

✠

Jan Vanhoof was born in Balen, Belgium, on June 19, 1961. He received the degree in electrical engineering from the Katholieke Universiteit Leuven, Belgium, in 1984.

Currently he is a Research Assistant at the Inter-University Microelectronics Centre (IMEC), Leuven, Belgium, working on silicon compilation, more specifically on the synthesis of multiprocessor architectures for digital signal processing and on the mapping of signal processing algorithms onto these architectures. His research interests are in the use of artificial intelligence techniques in VLSI design.

Bart J. S. De Loore was born in Gent, Belgium, on November 5, 1960. He received the electrical engineering degree from the Katholieke Universiteit Leuven, Belgium, in 1983. From January 1985 he has been working for IMEC, Belgium, towards the Ph.D. degree in the field of CMOS circuit design.

In October 1983 he joined ESAT, KU-Leuven, Belgium, where he has participated in an ESPRIT microelectronics project.

✠

Gert Goossens was born in Genk, Belgium, on October 10, 1960. He received the degree in electrical engineering from the Katholieke Universiteit Leuven, Belgium in 1984.

He joined the Inter-University Microelectrons Centre (IMEC), Leuven, Belgium, as a research assistant. He is currently working on the development of CAD-methodologies for the synthesis of multiprocessor architectures for signal processing, and algorithm mapping. His research interests are mainly in the application of analytical optimization techniques in VLSI-design.

✠

Nelson F. Goncalvez was born in Brazil on September 26, 1952. He received the B.S. and M.S. degrees from the University of Sao Paulo, Sao Paulo, Brazil, in 1975 and 1978, respectively. In November 1984, he received the Ph.D. degree from the Katholieke Universiteit Leuven, Belgium.

From 1976 to 1978, he worked on the characterization of the interface SI/SIO. In September 1980, he joined ESAT, Katholieke Universiteit Leuven, where he had a Fellowship from FAPESP, Brazil. He has been working on the design of dynamic CMOS circuits.

✠

Luc J. M. Claesen (S'77–M'85) was born in Hasselt, Belgium, on May 23, 1956. He received the electrical engineering degree and Ph.D. degree in applied science from the Katholieke Universiteit Leuven, Heverlee, Belgium in 1979 and 1984, respectively.

After graduation in 1979 he joined the ESAT-laboratory, Katholieke Universiteit Leuven, as a Research Assistant. There he has been working in the field of computer-aided design of integrated systems for digital and analog signal processing. Since October 1983 he has been working on a database system for VLSI–CAD applications. Since July 1984 he has been at the IMEC laboratory in Heverlee (Belgium), where he is currently heading research in the Design Management and Verification group within the Design Methodologies for VLSI division. His research interests are in computer-aided design and integrated digital signal processing circuits.

Dr. Claesen is a member of the Koninklijke Vlaamse Ingenieurs Vereniging (KVIV).

✠

Johan K. J. Van Gindereuren was born in Genk, Belgium, on June 23, 1957. He received the electrical engineering degree from the Katholieke Universiteit Leuven, Heverlee, Belgium, in 1980. He is presently working towards the Ph.D. Degree on custom IC implementation of digital filters.

From 1980 to 1983, he obtained an I.W.O.N.L. Fellowship which allowed him to work as a Research Assistant at the ESAT Laboratory, Katholieke Universiteit Leuven. From 1983 he was appointed as Research Assistant under an ESPRIT project sponsored by Siemens, Philips, BTMC (ITT), Silvar-Lisco and the EEC, at the Katholieke Universiteit Leuven and IMEC, respectively. His current research interests are in computer-aided design and integration of digital signal processing algorithms and architectures.

✠

Joos Vandewalle (S'71 – M'79 – SM'82) was born in Kortrijk, Belgium, on August 31, 1948. He received the engineering degree and a doctorate degree in applied sciences, both from the Katholieke Universiteit Leuven, Belgium, in 1971 and 1976, respectively.

From 1972 to 1976 he was Assistant at the E.S.A.T. Laboratory, Laboratory of the Katholieke Universiteit Leuven, Belgium. From 1976 to 1978 he was Research Associate and from July 1978 to July 1979, he was Visiting Assistant Professor, both at the University of California, Berkeley. Since July 1979 he has been back at the E.S.A.T. Laboratory of the Katholieke Universiteit Leuven, Belgium, where he is currently Assistant Professor and Research Fellow. His research interests are mainly in mathematical system theory and its applications in circuit theory, control, signal processing and cryptography. He has authored or coauthored about 70 papers in these areas.

✠

Hugo J. De Man was born in Boom, Belgium on September 19, 1940. He received the electrical engineering degree and the Ph.D. degree in applied sciences from the Katholieke Universiteit Leuven, Haverlee, Belgium, in 1964 and 1968, respectively.

In 1968 he became a member of the staff of the Laboratory for Physics and Electronics of Semiconductors at the University of Leuven working on integrated circuit technology. From 1969 to 1971 he was at the Electronic Research Laboratory, University of California, Berkeley, as an ESRO-NASA Postdoctoral Research Fellow, working on Computer-Aided Device and Circuit Design. In 1971 he returned to the University of Leuven, as a Research Associate of the NFWO (Belgian National Science Foundation). In 1974 he became a Professor at the University of Leuven. During the winter quarter of 1974–1975 he was a Visiting Associate Professor at the University of California, Berkeley. He was an Associate Editor for the IEEE JOURNAL OF SOLID-STATE CIRCUITS from 1975–1980 and is presently Associate Editor for the IEEE Transactions on CAD. He received a Best Paper Award at the ISSCC of 1973 on Bipolar Device Simulation and at the 1981 ESSCIRC conference for work on an integrated CAD system. His actual field of research is the design of Integrated circuits and Computer-Aided Design. Since 1984 he is Vice-President of the VLSI systems design group of IMEC (Leuven, Belgium)