Automatic Synthesis of Signal Processing Benchmark using the CATHEDRAL Silicon Compilers.

L.Claesen°, F.Catthoor°, D.Lanneer°, G.Goossens°, S.Note°, J.Van Meerbergen‡, H.De Man°*

°*Interuniversity Micro Electronics Center* IMEC, Kapeldreef 75, B-3030 Leuven, Belgium.
‡*Philips Res. Labs*, Prof. Holstlaan, P.O.Box 80000, 5600 JA Eindhoven, The Netherlands.

Abstract

For the joint ACM/SIGDA – IEEE/DATC workshop on High-Level Synthesis [1] a number of benchmarks have been proposed. The wave digital filter (WDF) benchmark has been used for silicon synthesis. The hardwired target architectures of CATHEDRAL-I and III have an application range dependent on the required throughput. The processor based target architecture of CATHEDRAL-II is mainly useful when versitile functionlity (e.g. decision-making) is required. All of the CATHEDRAL-tools start from high-level signal flow graph specifications and perform the synthesis down to layout. The conclusions are that specific target architectures and their compilers have their range of application. It is also concluded that digital filter realizations should be specified at an even higher behavioral level, i.e. frequency characterisitics and signal to noise ratios.

1 Introduction

At IMEC research is performed in the area of automatic synthesis methods for DSP applications. An overview of the synthesis approaches taken in the CATHEDRAL systems is given in [2]. First we will discuss the benchmark chosen. Section 3 illustates the importance of optimization considerations at the algorithmic level of the benchmark. The resulting signal flowgraph is synthesized in hardware according to three architectural implementation strategies: (1) CATHEDRAL-I [3] : bit-serial implementation. (2) CATHEDRAL-II [5] : customized micro-programmed multi-processor architecture (3) CATHEDRAL-III [6] : hardwired bit-sliced implementation. In sections 4, 5 and 6, each CATHEDRAL system is briefly discussed.

2 The benchmark chosen.

The wave digital filter benchmark (true ladder type [13]) [1,10] has been chosen for illustrating automatic synthesis with the set of CATHEDRAL silicon compilers. The true ladder filter topology is given in fig. 1.

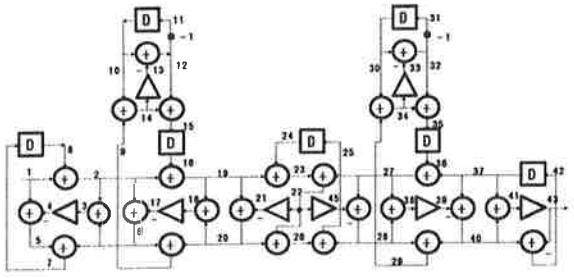

Figure 1: 5^{th} Order True Ladder Wave Digital Filter Benchmark.

*Professor at Kath. Univ. Leuven

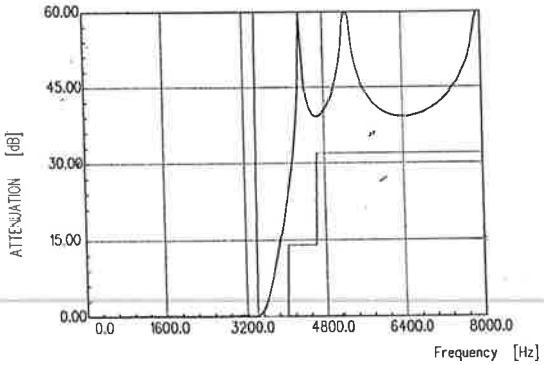

Figure 2: Specifications and a satisfying transfer function of a 5^{th} order PCM lowpass filter.

As the digital filter benchmark is not sufficiently specified, we propose to use the specifications of a specific fifth order filter e.g. the 5^{th} order PCM filter according to specification CCITT G712 PCM. This specification mainly consists of a passband ripple from 0Hz to 3kHz of +/- 0.125 dB and a stopband attenuation of -14 dB from 4.0kHz and -32 dB from 4.6kHz. Fig.2 illustrates the filter specifications and a satisfying transfer function. For the calculations we assumed a sample frequency of 16kHz. We set forward a data word length of 20 bits to be taken for the operations. In the next section the resulting signal to noise figures as calculated by DSP-DIGEST [2] are given. It would however be better that a specific signal to noise ratio was given and that as a consequence the word length (and the resulting hardware) can be optimized to the requirements.

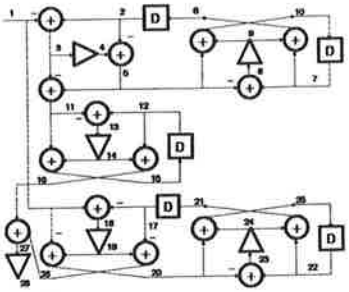

Figure 3: 5^{th} order WDF filter synthesized by FALCON

3 High-level Design with DSP-DIGEST.

3.1 Filter approximation and synthesis

The frequency response of the wave digital filter [10] proposed in [1] has been derived [13,11] from a 5^{th} order analog reference filter [12] and computed with the DIGEST analysis program [4,21] and fit into the

bounds. It should however be stressed that the implementation of fig.1 in a "true ladder" form *(referred to as LAD)* is very inefficient. Indeed, it is e.g. non-canonical as it contains 7 delays for a 5^{th} order response. In order to illustrate this, an alternative realisation has been designed with the WDF synthesis tool FALCON [14]. The result is a canonical 5^{th} order lattice filter with circulator sections (Fig.3) *(referred to as CIR)*. It has been designed with the same frequency specs as in fig. 2. This CIR structure exhibits many advantages: the arithmetic complexity is much smaller, the scaling properties are more favourable [13], the bit-serial delay management is simpler (see below) and all the other desirable properties of WDF's (such as reduced coefficient sensitivity and the absence of limit-cycles under certain conditions [13]) continue to hold.

3.2 Coefficient optimisation

The first step in the implementation requires the representation of the nominal real-valued coefficients with a limited number of bits. When a shift-add based realisation is adopted, the CSD-code [19] leads to a very compact result in terms of hardware. Area consuming multiplications are substituted by reduced shifts and add/substracts. As every non-zero bit eventually corresponds to some implementation cost, the total number of CSD-bits have to be minimized. The problem of finding the optimal coefficient set within the frequency constraints is very hard. This optimisation task has been solved with the simulated annealing based CAD-tool CAUCASIAN [20]. The results for the 2 filter structures is shown in Table 1.

3.3 Word-length determination

The CAD tool DIGEST allows to compute "templates" [2] which collect most of the information which is needed to determine an appropriate, optimal word-length (WL) for the signal nodes. In Table 2, the template charts for the output nodes in the 2 filters are compared for a proposed

filter LAD				filter CIR			
mult. node	CSD value	mult node	CSD value	mult. node	CSD value	mult node	CSD value
4	0.1	45	0.1	4	0.010-1	24	0.01
13	0.1	33	1.00-1	9	0.001	28	0.1
17	0.01	39	0.10001	14	0.0100-1		
21	0.10-1	43	1.0-100-1	19	0.10-1		

Table 1: Multiplier coefficients in CSD notation for the two alternative filter implementations

word-length of 20 bits. Value truncation is used for quantization. At the LSB side, the statistical noise contributions (R) amount to less than 1 quantisation unit (q.u.). Hence only 1 bit is affected. Also the total, worst-case error bound (E) under arbitrary input excitation conditions has been computed with LIMCYC [4,21], which takes correlated effects into account. The results show that in the proposed LAD filter 2 more bits can be affected by worst-case errors than in CIR. At the MSB side, protected sign bits (S) are necessary to avoid overflow at the most critical internal nodes. For this purpose the L1-norm in the time-domain (L) is used. In this case, the CIR realisation requires only 3 bits, whereas the LAD necessitates 5 bits. When these 2 finite word-length effects are taken into account, the signal-to-noise (SNR) ratio amounts to 96 dB for CIR and only 84 dB for LAD. The CIR alternative is clearly more favorable in terms of required word lengths for equal SNR. Dependent on the actual SNR requirements, the WL figures have to be adapted.

LAD	SSSLLL##########EEER
CIR	SSLL###############ER

Table 2: Finite word length templates for two filter structures

Finally, there is also the possibility to investigate the actual worst-case quantisation effects due to correlated errors, usually referred to as limit-cycles. For this purpose, the CAD-tool SAILPLANE [4,20] has been applied. It executes a simulated-annealing based search for the worst-case finite word-length effects. When the default two's complement value truncation is assumed, worst-case zero-input limit-cycles of magnitude 1 and period 4 to 14 have been detected in LAD. For CIR only constant limit-cycles of amplitude 1 have been found. Hence, even if the requirements to eliminate these limit-cycles [13] are not incorporated, the quantisation effects tend to remain quite small, due to the favourable properties of the WDF structures.

In the next sections alternative architectural implementations are discussed. They all start from the optimized signal flowgraphs, CSD-coefficient representation, and a word-length of 20 bits. Table 3 gives an overview of the results of the synthesis in the different Cathedral approaches. The sample frequency is the maximum word sampling frequency that has been obtained for the adapted benchmark specifications (coefficients, word lengths). The chip area is given for a $3\mu m$ CMOS nwell technology and does not include bonding pads.

Whereas digital filters are normally intended to operate at a specific sampling frequency, we have indicated in table 3 the maximum sample frequency that can be obtained in the several implementation techniques. The frequency values in the specifications have to be scaled accordingly.

LAD ladder based filter benchmark.							
Tool-box	Word length	Sam. freq. kHz	Clock freq. MHz	Area (A) mm^2	Cyc-les	Proc. data path	A/F_s $\frac{mm^2}{MHz}$
C-I	38	500	20	3.2	38	bit-ser.	6.4
C-II	20	222	10	9.4	45	1 alu	42.3
C-II	20	400	10	14.3	25	2 alu's	35.8
C-II	20	500	10	18.4	20	3 alu's	36.8
C-III	20	5000	5	24.0	1	bit-par.	4.8
CIR circulator based filter benchmark.							
Tool-box	Word length	Sam. freq. kHz	Clock freq. MHz	Area (A) mm^2	Cyc-les	Proc.. data path	A/F_s $\frac{mm^2}{MHz}$
C-I	20	1000	20	2.3	20	bit-ser.	2.3
C-II	20	833	10	6.8	12	1 alu	8.2
C-II	20	1250	10	10.6	8	2 alu's	8.5
C-II	20	1111	10	22.8	9	1 alu, 1 mult	20.5
C-III	20	7500	7.5	14.8	1	bit-par.	2.0

Table 3: Results of the several synthesis methods.

4 CATHEDRAL-I: bit-serial digital filter implementation.

CATHEDRAL-I [3] is an automatic synthesis system targeted towards bit-serial digital filter implementations. Besides high-level synthesis and optimization tools, CATHEDRAL-I provides an automatic mapping into a bit-serial hardware implementation. The layout is generated using standard cell place and route techniques [22] with a library of only 20 cells.

Due to the long critical loop in the algorithm of the LAD benchmark that was proposed, delay management [3] of the bit-serial architecture required to use 38 bits instead of 20 bits. This results in a maximum sample frequency of 500kHz. Due to the fact that bit-serial architectures are hardwired and only 1 bit operations are implemented, the resulting chip area is the smallest of the three target architectures envisioned. The problem of the long loop of operations can be avoided if the more tractable, implementation CIR structure is used. In the CIR structure a sampling frequency of 1000kHz see tabel 3 can be obtained, satisfying to the same initial specifications of the filter without increasing the word length.

fig. 4 and 5 resp. show the layouts generated for the LAD and CIR implementations of the 5^{th} order PCM filter.

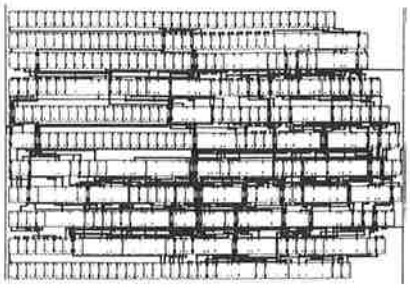

Figure 4: LAD benchmark generated by CATHEDRAL-I

5 CATHEDRAL-II: customized multi processor architecture synthesis.

The CATHEDRAL-II compiler and its methodology have been described in [5]. CATHEDRAL-II addresses highly complex DSP-applications with sample rates from a few kHz to 1 MHz. Algorithms in this range

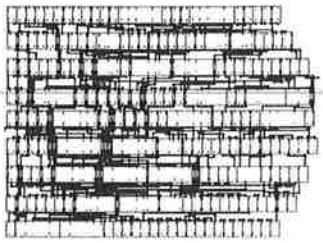

Figure 5: CIR benchmark generated by CATHEDRAL-I

are characterized by a high versatility in their functionality, combining arithmetic and decision making operations. For this reason, a flexible architectural style is required. The application range is much wider than digital filtering: current applications include echo-cancellation, pitch-extraction for speech, adaptive interpolation of audio signals, singular value decomposition, and control functions for Compact Disc.

CATHEDRAL II synthesizes multi-processor architectures [7], starting from an applicative description. For every processor, a customized data path can be generated, composed of predefined parameterizable execution units (EXU's) or modules that are stored in a library. These EXU's communicate via dedicated bus interconnections. The layouts of the EXU instances are built by a module generator tool [9], combining symbolic layout of the leaf cells with procedural composition techniques. In the current definition of the target architecture, the following EXU's are included: a general ALU, a multiplier/accumulator, a comparator unit, an address ALU, a normalizer unit, RAM, ROM, and FIFO's. A prototype parameterizable module library has been designed in 3μ CMOS and is operating at a 10 MHz clock. The data paths in each processor are controlled by a powerful microcoded controller, allowing for fast decision making and multiple branching (e.g. loop constructs).

The CATHEDRAL II prototype is an interactive system, that allows to generate and compare several alternative designs in a short time [16], and finally produce layouts. The architecture synthesis system contains a rule-based mapping program [17] to transform the behavioral specification into a processor data path, and a number of optimization tools, e.g. a microcode compiler to schedule the microcode and perform hardware bindings [18].

In Table 3, the design figures for the LAD and CIR benchmark are shown. Three alternative processor implementations have been generated, by interactively allocating a different number of ALU's. Automatically generated data paths for LAD are shown in figures 6 and 7. For every alternative, an optimized binding of operations to ALU's is computed automatically. The trade-off between the allocated hardware and the obtained machine-cycle count is indicated. A fourth design

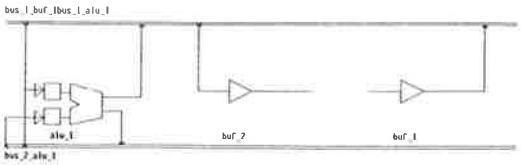

Figure 6: 1 ALU Datapath generated by CATHEDRAL-II.

containing a parallel multiplier requires a much larger area.

Two shortcomings in the current CATHEDRAL II prototype have resulted in slightly suboptimal designs. First of all, the current version doesn't exploit the CSD-coding of multiplication coefficients. Furthermore, at the moment the scheduler doesn't optimally exploit pipelines in the system. However, due to the long critical path in the block diagram of the LAD benchmark filter, pipelining possibilities are limited anyway. The scheduling for CIR results in much less processor cycles than for LAD.

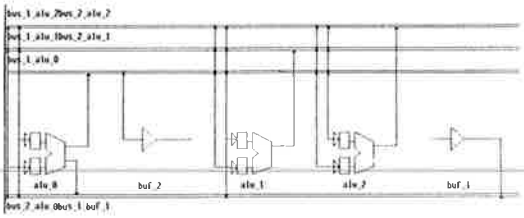

Figure 7: 3 ALU Datapath generated by CATHEDRAL-II.

6 CATHEDRAL-III Implementation

The CATHEDRAL-III system [6] is oriented towards *hardwired bit-sliced architectures* [23], which are intended for the implementation of algorithms in real-time video, image and communication domain, with sample rates ranging from 1 MHz to 40 MHz. The synthesis task for this kind of architecture can be divided into two subtasks, the *translation* and the *assembly* task.

During the translation step, a high-level description of the algorithm is translated to a Signal Flow Graph (SFG) suited for a hardwired bit-parallel architecture. For this architecture, in order to meet the high

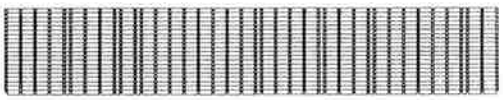

Figure 8: Floorplan layout representation of LAD benchmark synthesized by CATHEDRAL-III

speed requirements, typically each operation is mapped on a separate *functional building block* (FBB). The compiler maps arithmetic expressions of the algorithm into a structure of operations which can be evaluated with as much parallelism as is needed.

In the subsequent assembly task, which starts from a SFG description, a layout is generated taking both *performance, layout* and *area constraints* into account.

In the LAD benchmark, optimization at the register transfer level, which is based on retiming, did not improve anything. Neither did the optimization at the floorplanning level which tries to reorder the positions of the FBB's. Topological optimizations which attempt to transform the SFG in such a way that the performance increases, have not been applied. Only the optimization at the functional level has been successful: all simple adders have been replaced by faster carry bypass adders. Fig. 8 shows the floorplan layout of the data path.

As indicated in table 3, the CIR alternative filter is more efficient in terms of chip area and performance. The floorplan layout is given in fig. 9. The smaller critical loop in CIR is also favourable for the

Figure 9: Floorplan layout representation of CIR benchmark synthesized by CATHEDRAL-III

attainable sampling rate. The gain in sampling rate is not that better in CIR because a large part of the critical delay path still runs through the carry ripple of the 20 bits adders.

7 Conclusions.

In this paper we discussed the implementation of two WDF benchmarks that have been designed with three architecture specific silicon compilers.

The design time for high level synthesis and optimization is roughly one day. For each of the three synthesis systems, the elapsed design cycle starting from the specifications down to the optimized signal flow graph is another 1..2 days. For the architecture and layout generation (including the evaluation of the trade-offs), the design time ranges from a few hours to a day.

Specific *higher level* filter specifications have been used for the synthesis with three different implementation strategies in the CATHEDRAL silicon compilers. It has been shown that as an initial optimization step, it is important to initially choose a good algorithm. The CIR alternative is more favorable as compared to LAD in several aspects:

- less operations required.
- The structures in CIR are very modular, and therefore very well suited for modular hardware implementation.
- there is less sensitivity to limit cycles and better scaling properties which results in a smaller word length needed.
- CIR does not have such a long critical path of operations as LAD!
- WDF structures like CIR can be automatically synthesized from high level specifications using the FALCON program [14].

It should be noticed that digital filtering is only a special case of the more general application area of Digital Signal Processing. Whereas CATHEDRAL-I and III are mainly oriented towards linear signal processing, the CATHEDRAL-II architecture is targeted towards more general DSP applications. The results in table 3 indicate that the processor-based CATHEDRAL-II is only attractive when other, more complex operations (e.g. decision-making) have to be accommodated too. Moreover, the flexibility in terms of throughput range is higher because an area-time trade-off is possible by changing the number of EXU's. Hence the matching between the specification and the final result can be more optimal. For the hard-wired approaches, the bit-serial is clearly intended for a lower sample rate than the bit-sliced alternative. The area-time product (A/F_s) is however more favourable for the latter due to the less costly overhead in terms of register cells.

Acknowledgements

The work on CATHEDRAL-I, II and III is the result of the cooperative work of all members of the VSDM division within IMEC, its industrial partners and the Ruhr University Bochum. The work is sponsored by the EC under the ESPRIT-97 and ESPRIT-1058 projects and its associated industrial partners: Philips, Siemens, BTMC-Alcatel and Silvar-Lisco.

References

[1] –, "Workshop on High-Level Synthesis", *Rosario Resort, Orcas Island, Washington*, January 24-27, 1988, ACM/SIGDA and IEEE/DATC, Chaired by G.Borriello and E.Detjens.

[2] H.De Man e.a., "Synthesis of DSP Systems at Leuven", *IEEE Int. Conf. on Computer Design: VLSI in Computers & Processors*, Proc. ICCD'87, Rey Brook, NY, Oct.5-Oct.8, 1987, pp. 133-145.

[3] R.Jain, F.Catthoor, J.Vanhoof, B.De Loore, G.Goossens, N.Goncalves, L.Claesen, J.Van Ginderdeuren, J.Vandewalle, H.De Man, "Custom Design of a VLSI PCM-FDM Transmultiplexer from System Specifications to Circuit Layout Using a Computer-Aided Design System", *IEEE Journ. of Solid-State Circ.*, Volume SC-21, Number 1, February 1986, pp. 73-85.

[4] L.Claesen, F.Catthoor, H.De Man, J.Vandewalle, S.Note, K.Mertens, "A CAD environment for the thorough analysis, simulation and characterization of VLSI implementable DSP systems", *IEEE Int. Conf. on Computer Design: VLSI in Computers*, ICCD'86, Oct. 6-9, 1986, New York, pp. 72-75.

[5] H.De Man, J.Rabaey, P.Six, L.Claesen, "CATHEDRAL-II: A Silicon Compiler for Digital Signal Processing", *IEEE Design & Test of Computers*, December 1986, Vol.3, No.6, pp. 13-25.

[6] S.Note, J.Van Meerbergen, F.Catthoor, H.De Man, "Automatic Synthesis of a High Speed CORDIC Algorithm with the CATHEDRAL-III Compilation System", *IEEE Int. Symposium on Circ. and Syst.*, ISCAS-88, Helsinki University of Technology, Espoo, Finland, June 7-9, 1988.

[7] F.Catthoor, J.Rabaey, G.Goossens, J.Van Meerbergen, R.Jain, H.De Man, J.Vandewalle, "Architectural strategies for an application-specific synchronous multi-processor environment", *IEEE Trans. on Accoustics, Speech and Signal Processing*, February 1988.

[8] H.De Man, "Evolution of CAD tools towards third generation custom VLSI design", *Revue Phys. Appl. 22*, Vol.22, January 1987, pp. 31-45.

[9] P.Six, L.Claesen, J.Rabaey, H.De Man, "An Inteligent Module Gernator Environment", *Proc. of the 23rd Design Automation Conf.*, Las Vegas, June 29-July 2, 1986, pp. 730-735.

[10] P.Dewilde, E.Deprettere, R.Nouta, "Parallel and Pipelined VLSI Implementation of Signal Processing Algorithms", in "VLSI and Modern Signal Processing", edited by: S.Y.Kung, H.J.Whitehouse, T.Kailath, *Prentice-Hall Information and System Sciences Series*, 1985, pp.258-264.

[11] R.Nouta, "De vertaling van een 5^{de} orde Cauer-type laagdoorlaat filter in een golffilterstructuur", *private communication.*

[12] A.I. Zverev, "Handbook of Filter Synthesis", *Wiley*, New York, 1967.

[13] A.Fettweis, "Wave Digital Filters: Theory and Practice", *Proc. IEEE*, Vol.74, No.2, pp.270-327, February 1986.

[14] L.Gazsi, "Explicit formulae for lattice wave digital filter", *IEEE Trans. on Circ. and Syst.*, CAS Vol.32, January 1985.

[15] P.N.Hilfinger, "A High-Level Language and Silicon Compiler for Digital Signal Processing", *Proc. IEEE 1985 Custom Integrated Circ. Conf.*, CICC-85, pp. 213-216.

[16] J. Rabaey, J. Vanhoof, G. Goossens, F. Catthoor, H. De Man, "CATHEDRAL II : computer aided synthesis of digital signal processing systems", *Proc. IEEE-CICC'87*, 1987.

[17] J. Vanhoof, J. Rabaey, H. De Man, "A knowledge-based CAD system for synthesis of multi-processor digital signal processing chips", *Proc. IFIP WG 10.5 VLSI'87 Conf.*, Vancouver, August 1987.

[18] G. Goossens, J. Rabaey, J. Vandewalle, H. De Man, "An efficient microcode-compiler for custom DSP-processor", *Proc. IEEE-ICCAD'87*, pp. 24-27, Santa Clara, November 1987.

[19] S.W.Reitwiesner, "Binary Arithmetic", *in Advances in Computers, Vol.1, Academic Press*, pp.231-308, 1966.

[20] F.Catthoor, "Architectural design strategies for complex DSP-systems in an automated synthesis environment", *Doctoral dissertation, ESAT, K.U.Leuven*, Belgium, May 1987.

[21] L.Claesen, "Computer Aided Design of Integrated Systems for Digital and Analog Signal Processing", *Ph.D. Dissertation, Kath. Univ. Leuven*, Belgium, March 1984.

[22] H.Beke e.a., "CAL-MP: an advanced computer-aided layout program for MOS-LSI", in *Digest European Conf. on Solid-State Circ.*, pp.67-69, 1980.

[23] F.Catthoor, H.De Man, "Customised architectural methodologies for high-speed image and video processing", *Proc. ICASSP*, New York, April 1988.

Automatic Synthesis of Signal Processing Benchmark using the CATHEDRAL Silicon Compilers.

L.Claesen°, F.Catthoor°, D.Lanneer°, G.Goossens°, S.Note°, J.Van Meerbergen‡, H.De Man°*

°*Interuniversity Micro Electronics Center* IMEC, Kapeldreef 75, B-3030 Leuven, Belgium.
‡*Philips Res. Labs*, Prof. Holstlaan, P.O.Box 80000, 5600 JA Eindhoven, The Netherlands.

Abstract

For the joint ACM/SIGDA – IEEE/DATC workshop on High-Level Synthesis [1] a number of benchmarks have been proposed. The wave digital filter (WDF) benchmark has been used for silicon synthesis. The hardwired target architectures of CATHEDRAL-I and III have an application range dependent on the required throughput. The processor based target architecture of CATHEDRAL-II is mainly useful when versitile functionlity (e.g. decision-making) is required. All of the CATHEDRAL-tools start from high-level signal flow graph specifications and perform the synthesis down to layout. The conclusions are that specific target architectures and their compilers have their range of application. It is also concluded that digital filter realizations should be specified at an even higher behavioral level, i.e. frequency characterisitics and signal to noise ratios.

1 Introduction

At IMEC research is performed in the area of automatic synthesis methods for DSP applications. An overview of the synthesis approaches taken in the CATHEDRAL systems is given in [2]. First we will discuss the benchmark chosen. Section 3 illustates the importance of optimization considerations at the algorithmic level of the benchmark. The resulting signal flowgraph is synthesized in hardware according to three architectural implementation strategies: (1) CATHEDRAL-I [3] : bit-serial implementation. (2) CATHEDRAL-II [5] : customized micro-programmed multi-processor architecture (3) CATHEDRAL-III [6] : hardwired bit-sliced implementation. In sections 4, 5 and 6, each CATHEDRAL system is briefly discussed.

2 The benchmark chosen.

The wave digital filter benchmark (true ladder type [13]) [1,10] has been chosen for illustrating automatic synthesis with the set of CATHEDRAL silicon compilers. The true ladder filter topology is given in fig. 1.

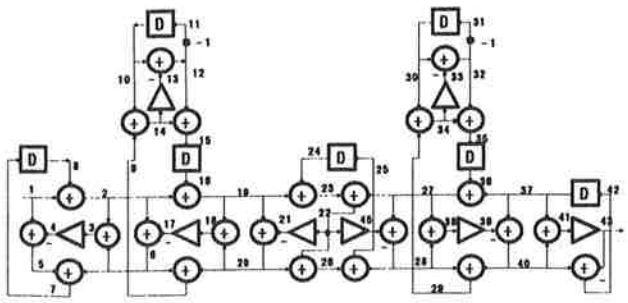

Figure 1: 5^{th} Order True Ladder Wave Digital Filter Benchmark.

*Professor at Kath. Univ. Leuven

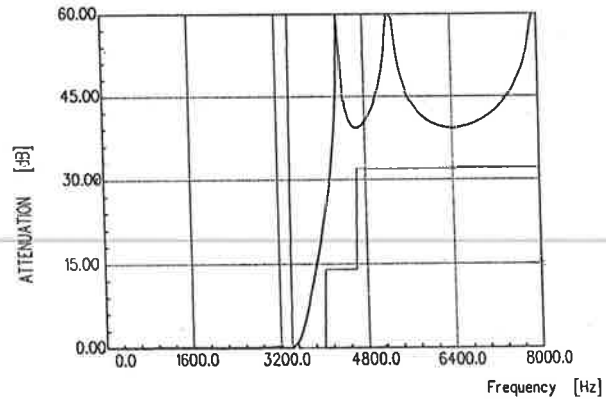

Figure 2: Specifications and a satisfying transfer function of a 5^{th} order PCM lowpass filter.

As the digital filter benchmark is not sufficiently specified, we propose to use the specifications of a specific fifth order filter c.q. the 5^{th} order PCM filter according to specification CCITT G712 PCM. This specification mainly consists of a passband ripple from 0Hz to 3kHz of +/- 0.125 dB and a stopband attenuation of -14 dB from 4.0kHz and -32 dB from 4.6kHz. Fig.2 illustrates the filter specifications and a satisfying transfer function. For the calculations we assumed a sample frequency of 16kHz. We set forward a data word length of 20 bits to be taken for the operations. In the next section the resulting signal to noise figures as calculated by DSP-DIGEST [2] are given. It would however be better that a specific signal to noise ratio was given and that as a consequence the word length (and the resulting hardware) can be optimized to the requirements.

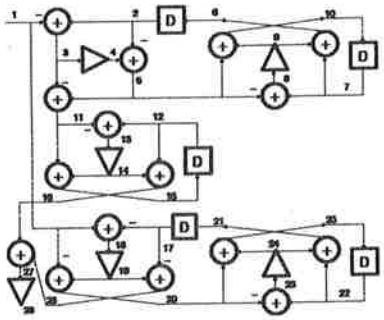

Figure 3: 5^{th} order WDF filter synthesized by FALCON

3 High-level Design with DSP-DIGEST.

3.1 Filter approximation and synthesis

The frequency response of the wave digital filter [10] proposed in [1] has been derived [13,11] from a 5^{th} order analog reference filter [12] and computed with the DIGEST analysis program [4,21] and fit into the

CH2584-1/88/0000-0087 $1.00 © 1988 IEEE

bounds. It should however be stressed that the implementation of fig.1 in a "true ladder" form *(referred to as LAD)* is very inefficient. Indeed, it is e.g. non-canonical as it contains 7 delays for a 5^{th} order response. In order to illustrate this, an alternative realisation has been designed with the WDF synthesis tool FALCON [14]. The result is a canonical 5^{th} order lattice filter with circulator sections (Fig.3) *(referred to as CIR)*. It has been designed with the same frequency specs as in fig. 2. This CIR structure exhibits many advantages: the arithmetic complexity is much smaller, the scaling properties are more favourable [13], the bit-serial delay management is simpler (see below) and all the other desirable properties of WDF's (such as reduced coefficient sensitivity and the absence of limit-cycles under certain conditions [13]) continue to hold.

3.2 Coefficient optimisation

The first step in the implementation requires the representation of the nominal real-valued coefficients with a limited number of bits. When a shift-add based realisation is adopted, the CSD-code [19] leads to a very compact result in terms of hardware. Area consuming multiplications are substituted by reduced shifts and add/substracts. As every non-zero bit eventually corresponds to some implementation cost, the total number of CSD-bits have to be minimized. The problem of finding the optimal coefficient set within the frequency constraints is very hard. This optimisation task has been solved with the simulated annealing based CAD-tool CAUCASIAN [20]. The results for the 2 filter structures is shown in Table 1.

3.3 Word-length determination

The CAD tool DIGEST allows to compute "templates" [2] which collect most of the information which is needed to determine an appropriate, optimal word-length (WL) for the signal nodes. In Table 2, the template charts for the output nodes in the 2 filters are compared for a proposed

filter LAD				filter CIR			
mult. node	CSD value	mult node	CSD value	mult. node	CSD value	mult node	CSD value
4	0.1	45	0.1	4	0.010-1	24	0.01
13	0.1	33	1.00-1	9	0.001	28	0.1
17	0.01	39	0.10001	14	0.0100-1		
21	0.10-1	43	1.0-100-1	19	0.10-1		

Table 1: Multiplier coefficients in CSD notation for the two alternative filter implementations

word-length of 20 bits. Value truncation is used for quantization. At the LSB side, the statistical noise contributions (R) amount to less than 1 quantisation unit (q.u.). Hence only 1 bit is affected. Also the total, worst-case error bound (E) under arbitrary input excitation conditions has been computed with LIMCYC [4,21], which takes correlated effects into account. The results show that in the proposed LAD filter 2 more bits can be affected by worst-case errors than in CIR. At the MSB side, protected sign bits (S) are necessary to avoid overflow at the most critical internal nodes. For this purpose the L1-norm in the time-domain (L) is used. In this case, the CIR realisation requires only 3 bits, whereas the LAD necessitates 5 bits. When these 2 finite word-length effects are taken into account, the signal-to-noise (SNR) ratio amounts to 96 dB for CIR and only 84 dB for LAD. The CIR alternative is clearly more favorable in terms of required word lengths for equal SNR. Dependent on the actual SNR requirements, the WL figures have to be adapted.

LAD	SSSLLL##########EEER
CIR	SSLL################ER

Table 2: Finite word length templates for two filter structures

Finally, there is also the possibility to investigate the actual worst-case quantisation effects due to correlated errors, usually referred to as limit-cycles. For this purpose, the CAD-tool SAILPLANE [4,20] has been applied. It executes a simulated-annealing based search for the worst-case finite word-length effects. When the default two's complement value truncation is assumed, worst-case zero-input limit-cycles of magnitude 1 and period 4 to 14 have been detected in LAD. For CIR only constant limit-cycles of amplitude 1 have been found. Hence, even if the requirements to eliminate these limit-cycles [13] are not incorporated, the quantisation effects tend to remain quite small, due to the favourable properties of the WDF structures.

In the next sections alternative architectural implementations are discussed. They all start from the optimized signal flowgraphs, CSD-coefficient representation, and a word-length of 20 bits. Table 3 gives an overview of the results of the synthesis in the different Cathedral approaches. The sample frequency is the maximum word sampling frequency that has been obtained for the adapted benchmark specifications (coefficients, word lengths). The chip area is given for a $3\mu m$ CMOS nwell technology and does not include bonding pads.

Whereas digital filters are normally intended to operate at a specific sampling frequency, we have indicated in table 3 the maximum sample frequency that can be obtained in the several implementation techniques. The frequency values in the specifications have to be scaled accordingly.

LAD ladder based filter benchmark.							
Tool-box	Word length	Sam. freq. kHz	Clock freq. MHz	Area (A) mm^2	Cyc-les	Proc. data path	A/F_s $\frac{mm^2}{MHz}$
C-I	38	500	20	3.2	38	bit-ser.	6.4
C-II	20	222	10	9.4	45	1 alu	42.3
C-II	20	400	10	14.3	25	2 alu's	35.8
C-II	20	500	10	18.4	20	3 alu's	36.8
C-III	20	5000	5	24.0	1	bit-par.	4.8
CIR circulator based filter benchmark.							
Tool-box	Word length	Sam. freq. kHz	Clock freq. MHz	Area (A) mm^2	Cyc-les	Proc.. data path	A/F_s $\frac{mm^2}{MHz}$
C-I	20	1000	20	2.3	20	bit-ser.	2.3
C-II	20	833	10	6.8	12	1 alu	8.2
C-II	20	1250	10	10.6	8	2 alu's	8.5
C-II	20	1111	10	22.8	9	1 alu, 1 mult	20.5
C-III	20	7500	7.5	14.8	1	bit-par.	2.0

Table 3: Results of the several synthesis methods.

4 CATHEDRAL-I: bit-serial digital filter implementation.

CATHEDRAL-I [3] is an automatic synthesis system targeted towards bit-serial digital filter implementations. Besides high-level synthesis and optimisation tools, CATHEDRAL-I provides an automatic mapping into a bit-serial hardware implementation. The layout is generated using standard cell place and route techniques [22] with a library of only 20 cells.

Due to the long critical loop in the algorithm of the LAD benchmark that was proposed, delay management [3] of the bit-serial architecture required to use 38 bits instead of 20 bits. This results in a maximum sample frequency of 500kHz. Due to the fact that bit-serial architectures are hardwired and only 1 bit operations are implemented, the resulting chip area is the smallest of the three target architectures envisioned. The problem of the long loop of operations can be avoided if the more tractable, implementation CIR structure is used. In the CIR structure a sampling frequency of 1000kHz see tabel 3 can be obtained, satisfying to the same initial specifications of the filter without increasing the word length.

fig. 4 and 5 resp. show the layouts generated for the LAD and CIR implementations of the 5^{th} order PCM filter.

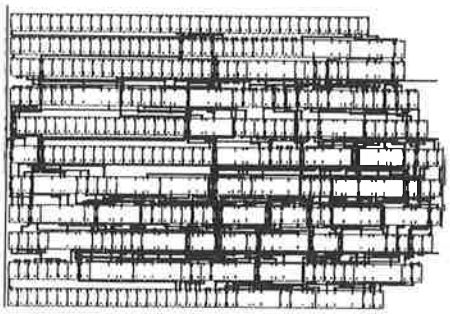

Figure 4: LAD benchmark generated by CATHEDRAL-I

5 CATHEDRAL-II: customized multi processor architecture synthesis.

The CATHEDRAL-II compiler and its methodology have been described in [5]. CATHEDRAL-II addresses highly complex DSP-applications with sample rates from a few kHz to 1 MHz. Algorithms in this range

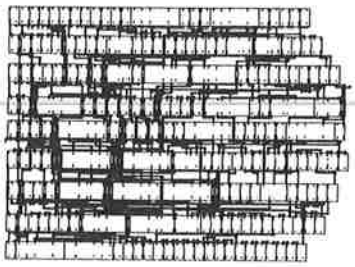

Figure 5: CIR benchmark generated by CATHEDRAL-I

are characterized by a high versatility in their functionality, combining arithmetic and decision making operations. For this reason, a flexible architectural style is required. The application range is much wider than digital filtering: current applications include echo-cancellation, pitch-extraction for speech, adaptive interpolation of audio signals, singular value decomposition, and control functions for Compact Disc.

CATHEDRAL II synthesizes multi-processor architectures [7], starting from an applicative description. For every processor, a customized data path can be generated, composed of predefined parameterizable execution units (EXU's) or modules that are stored in a library. These EXU's communicate via dedicated bus interconnections. The layouts of the EXU instances are built by a module generator tool [9], combining symbolic layout of the leaf cells with procedural composition techniques. In the current definition of the target architecture, the following EXU's are included: a general ALU, a multiplier/accumulator, a comparator unit, an address ALU, a normalizer unit, RAM, ROM, and FIFO's. A prototype parameterizable module library has been designed in 3μ CMOS and is operating at a 10 MHz clock. The data paths in each processor are controlled by a powerful microcoded controller, allowing for fast decision making and multiple branching (e.g. loop constructs).

The CATHEDRAL II prototype is an interactive system, that allows to generate and compare several alternative designs in a short time [16], and finally produce layouts. The architecture synthesis system contains a rule-based mapping program [17] to transform the behavioral specification into a processor data path, and a number of optimization tools, e.g. a microcode compiler to schedule the microcode and perform hardware bindings [18].

In Table 3, the design figures for the LAD and CIR benchmark are shown. Three alternative processor implementations have been generated, by interactively allocating a different number of ALU's. Automatically generated data paths for LAD are shown in figures 6 and 7. For every alternative, an optimized binding of operations to ALU's is computed automatically. The trade-off between the allocated hardware and the obtained machine-cycle count is indicated. A fourth design

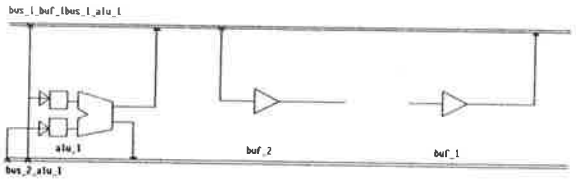

Figure 6: 1 ALU Datapath generated by CATHEDRAL-II.

containing a parallel multiplier requires a much larger area.

Two shortcomings in the current CATHEDRAL II prototype have resulted in slightly suboptimal designs. First of all, the current version doesn't exploit the CSD-coding of multiplication coefficients. Furthermore, at the moment the scheduler doesn't optimally exploit pipelines in the system. However, due to the long critical path in the block diagram of the LAD benchmark filter, pipelining possibilities are limited anyway. The scheduling for CIR results in much less processor cycles than for LAD.

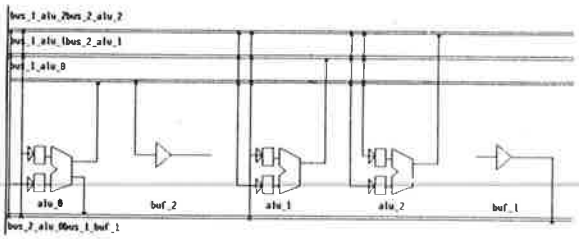

Figure 7: 3 ALU Datapath generated by CATHEDRAL-II.

6 CATHEDRAL-III Implementation

The CATHEDRAL-III system [6] is oriented towards *hardwired bit-sliced architectures* [23], which are intended for the implementation of algorithms in real-time video, image and communication domain, with sample rates ranging from 1 MHz to 40 MHz. The synthesis task for this kind of architecture can be divided into two subtasks, the *translation* and the *assembly* task.

During the translation step, a high-level description of the algorithm is translated to a Signal Flow Graph (SFG) suited for a hardwired bit-parallel architecture. For this architecture, in order to meet the high

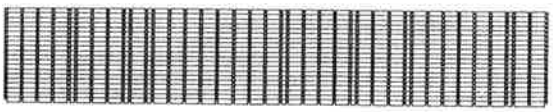

Figure 8: Floorplan layout representation of LAD benchmark synthesized by CATHEDRAL-III

speed requirements, typically each operation is mapped on a separate *functional building block* (FBB). The compiler maps arithmetic expressions of the algorithm into a structure of operations which can be evaluated with as much parallelism as is needed.

In the subsequent assembly task, which starts from a SFG description, a layout is generated taking both *performance, layout* and *area constraints* into account.

In the LAD benchmark, optimization at the register transfer level, which is based on retiming, did not improve anything. Neither did the optimization at the floorplanning level which tries to reorder the positions of the FBB's. Topological optimizations which attempt to transform the SFG in such a way that the performance increases, have not been applied. Only the optimization at the functional level has been successful: all simple adders have been replaced by faster carry bypass adders. Fig. 8 shows the floorplan layout of the data path.

As indicated in table 3, the CIR alternative filter is more efficient in terms of chip area and performance. The floorplan layout is given in fig. 9. The smaller critical loop in CIR is also favourable for the

Figure 9: Floorplan layout representation of CIR benchmark synthesized by CATHEDRAL-III

attainable sampling rate. The gain in sampling rate is not that better in CIR because a large part of the critical delay path still runs through the carry ripple of the 20 bits adders.

7 Conclusions.

In this paper we discussed the implementation of two WDF benchmarks that have been designed with three architecture specific silicon compilers.

The design time for high level synthesis and optimization is roughly one day. For each of the three synthesis systems, the elapsed design cycle starting from the specifications down to the optimized signal flow graph is another 1..2 days. For the architecture and layout generation (including the evaluation of the trade-offs), the design time ranges from a few hours to a day.

Specific *higher level* filter specifications have been used for the synthesis with three different implementation strategies in the CATHEDRAL silicon compilers. It has been shown that as an initial optimization step, it is important to initially choose a good algorithm. The CIR alternative is more favorable as compared to LAD in several aspects:

- less operations required.
- The structures in CIR are very modular, and therefore very well suited for modular hardware implementation.
- there is less sensitivity to limit cycles and better scaling properties which results in a smaller word length needed.
- CIR does not have such a long critical path of operations as LAD!
- WDF structures like CIR can be automatically synthesized from high level specifications using the FALCON program [14].

It should be noticed that digital filtering is only a special case of the more general application area of Digital Signal Processing. Whereas CATHEDRAL-I and III are mainly oriented towards linear signal processing, the CATHEDRAL-II architecture is targeted towards more general DSP applications. The results in table 3 indicate that the processor-based CATHEDRAL-II is only attractive when other, more complex operations (e.g. decision-making) have to be accommodated too. Moreover, the flexibility in terms of throughput range is higher because an area-time trade-off is possible by changing the number of EXU's. Hence the matching between the specification and the final result can be more optimal. For the hard-wired approaches, the bit-serial is clearly intended for a lower sample rate than the bit-sliced alternative. The area-time product (A/F_s) is however more favourable for the latter due to the less costly overhead in terms of register cells.

Acknowledgements

The work on CATHEDRAL-I, II and III is the result of the cooperative work of all members of the VSDM division within IMEC, its industrial partners and the Ruhr University Bochum. The work is sponsored by the EC under the ESPRIT-97 and ESPRIT-1058 projects and its associated industrial partners: Philips, Siemens, BTMC-Alcatel and Silvar-Lisco.

References

[1] –, "Workshop on High-Level Synthesis", *Rosario Resort, Orcas Island, Washington*, January 24-27, 1988, ACM/SIGDA and IEEE/DATC, Chaired by G.Borriello and E.Detjens.

[2] H.De Man e.a., "Synthesis of DSP Systems at Leuven", *IEEE Int. Conf. on Computer Design: VLSI in Computers & Processors*, Proc. ICCD'87, Rey Brook, NY, Oct.5-Oct.8, 1987, pp. 133-145.

[3] R.Jain, F.Catthoor, J.Vanhoof, B.De Loore, G.Goossens, N.Goncalves, L.Claesen, J.Van Ginderdeuren, J.Vandewalle, H.De Man, "Custom Design of a VLSI PCM-FDM Transmultiplexer from System Specifications to Circuit Layout Using a Computer-Aided Design System", *IEEE Journ. of Solid-State Circ.*, Volume SC-21, Number 1, February 1986, pp. 73-85.

[4] L.Claesen, F.Catthoor, H.De Man, J.Vandewalle, S.Note, K.Mertens, "A CAD environment for the thorough analysis, simulation and characterisation of VLSI implementable DSP systems", *IEEE Int. Conf. on Computer Design: VLSI in Computers*, ICCD'86, Oct. 6-9, 1986, New York, pp. 72-75.

[5] H.De Man, J.Rabaey, P.Six, L.Claesen, "CATHEDRAL-II: A Silicon Compiler for Digital Signal Processing", *IEEE Design & Test of Computers*, December 1986, Vol.3, No.6, pp. 13-25.

[6] S.Note, J.Van Meerbergen, F.Catthoor, H.De Man, "Automatic Synthesis of a High Speed CORDIC Algorithm with the CATHEDRAL-III Compilation System", *IEEE Int. Symposium on Circ. and Syst.*, ISCAS-88, Helsinki University of Technology, Espoo, Finland, June 7-9, 1988.

[7] F.Catthoor, J.Rabaey, G.Goossens, J.Van Meerbergen, R.Jain, H.De Man, J.Vandewalle, "Architectural strategies for an application-specific synchronous multi-processor environment", *IEEE Trans. on Accoustics, Speech and Signal Processing*, February 1988.

[8] H.De Man, "Evolution of CAD tools towards third generation custom VLSI design", *Revue Phys. Appl. 22*, Vol.22, January 1987, pp. 31-45.

[9] P.Six, L.Claesen, J.Rabaey, H.De Man, "An Inteligent Module Gernator Environment", *Proc. of the 23rd Design Automation Conf.*, Las Vegas, June 29-July 2, 1986, pp. 730-735.

[10] P.Dewilde, E.Deprettere, R.Nouta, "Parallel and Pipelined VLSI Implementation of Signal Processing Algorithms", in "VLSI and Modern Signal Processing", edited by: S.Y.Kung, H.J.Whitehouse, T.Kailath, *Prentice-Hall Information and System Sciences Series*, 1985, pp.258-264.

[11] R.Nouta, "De vertaling van een 5^{de} orde Cauer-type laagdoorlaat filter in een golffilterstructuur", *private communication.*

[12] A.I. Zverev, "Handbook of Filter Synthesis", *Wiley*, New York, 1967.

[13] A.Fettweis, "Wave Digital Filters: Theory and Practice", *Proc. IEEE*, Vol.74, No.2, pp.270-327, February 1986.

[14] L.Gassi, "Explicit formulae for lattice wave digital filter", *IEEE Trans. on Circ. and Syst.*, CAS Vol.32, January 1985.

[15] P.N.Hilfinger, "A High-Level Language and Silicon Compiler for Digital Signal Processing", *Proc. IEEE 1985 Custom Integrated Circ. Conf.*, CICC-85, pp. 213-216.

[16] J. Rabaey, J. Vanhoof, G. Goossens, F. Catthoor, H. De Man, "CATHEDRAL II : computer aided synthesis of digital signal processing systems", *Proc. IEEE-CICC'87*, 1987.

[17] J. Vanhoof, J. Rabaey, H. De Man, "A knowledge-based CAD system for synthesis of multi-processor digital signal processing chips", *Proc. IFIP WG 10.5 VLSI'87 Conf.*, Vancouver, August 1987.

[18] G. Goossens, J. Rabaey, J. Vandewalle, H. De Man, "An efficient microcode-compiler for custom DSP-processor", *Proc. IEEE-ICCAD'87*, pp. 24-27, Santa Clara, November 1987.

[19] S.W.Reitwiesner, "Binary Arithmetic", *in Advances in Computers, Vol.1, Academic Press*, pp.231-308, 1966.

[20] F.Catthoor, "Architectural design strategies for complex DSP-systems in an automated synthesis environment", *Doctoral dissertation, ESAT, K.U.Leuven*, Belgium, May 1987.

[21] L.Claesen, "Computer Aided Design of Integrated Systems for Digital and Analog Signal Processing", *Ph.D. Dissertation, Kath. Univ. Leuven*, Belgium, March 1984.

[22] H.Beke e.a., "CAL-MP: an advanced computer-aided layout program for MOS-LSI", in *Digest European Conf. on Solid-State Circ.*, pp.67-69, 1980.

[23] F.Catthoor, H.De Man, "Customised architectural methodologies for high-speed image and video processing", *Proc. ICASSP*, New York, April 1988.

Proceedings of the IEEE 1988 CUSTOM INTEGRATED CIRCUITS CONFERENCE

Rochester Plaza **May 16-19, 1988**
Holiday Inn-Genesee Plaza
Rochester Riverside Convention Center
Rochester, New York

The CICC '88 is sponsored by the IEEE Electron Devices Society, the IEEE Solid-State Circuits Council, and co-sponsored by the IEEE Rochester Section. Its aim is to bring together designers, producers and users of custom ICs to discuss recent developments and future directions in custom integrated circuits.

88CH2584-1