A 256-Point Discrete Fourier Transform Processor Fabricated in a 2 μm NMOS Technology

JEF L. VAN MEERBERGEN AND FRANS J. VAN WYK

Abstract—**In this paper we describe a 2 μm NMOS single-chip processor, which computes a 256-point Fourier transform in 6.5 ms by means of the DFT algorithm. This chip is primarily intended for speech processing applications, such as voice recognizers, pitch extractors, and DFT vocoders. The DFT algorithm has been implemented with two parallel multipliers, an accumulator/shifter, and two ROM's with separate address computation units. The processor contains over 20 000 transistors and dissipates ~400 mW when operating at a clock frequency of 10 MHz. The die size is only 12.5 mm^2 as a result of the use of a full-custom design method and the incorporation of low-resistance implanted As$^+$ undercrossings within the logic circuitry.**

I. Introduction

THE Fourier transform is probably one of the most important algorithms in signal analysis. Several methods are known for computing this transform, each having their pros and cons.

This paper presents a single-chip processor, which computes a Fourier transform using the direct computation method [1]. In this paper we shall refer to this direct evaluation as the discrete Fourier transform (DFT).

In Section II, the DFT algorithm is explained. Then a comparison is made, in Section III, to the fast Fourier transform (FFT), with respect to a VLSI implementation. We explain why the DFT algorithm is preferred in a number of speech processing applications. Section IV gives a general overview of the DFT processor. The operation of the chip is also discussed here. Next, we give some performance figures and two application examples in Sections V and VI. This is followed by details of some of the key circuit blocks, such as the multiplier, the address ALU of the ROM, and the scan test hardware in Section VII. The technological features of the 2 μm NMOS process are given in Section VIII. We conclude with a discussion of the design method and CAD tools in Section IX, and a summary in Section X.

II. DFT Algorithm

The Fourier transform is implemented on the chip using the direct computation method, in this paper called the DFT algorithm. This algorithm is described as follows:

$$F_{\mathrm{Re}}(k) = \sum_{i=0}^{N-1} D(i)\, W(i) \cos(2\pi ik/N), \qquad k = 0, 1, \cdots, N-1 \tag{1}$$

$$F_{\mathrm{Im}}(k) = \sum_{i=0}^{N-1} D(i)\, W(i) \sin(2\pi ik/N), \qquad k = 0, 1, \cdots, N-1 \tag{2}$$

where $F_{\mathrm{Re}}(k)$ and $F_{\mathrm{Im}}(k)$ are the real and imaginary part of the frequency sample. The value $D(i)$ represents the data samples and N the number of input samples and thus the length of the transformation. $W(i)$ represents the values of a Hamming window, used to improve the spectral resolution. This function is given by

$$W(i) = 0.54 - 0.46 \cos(2\pi i/N). \tag{3}$$

III. DFT Versus FFT

The fast Fourier transform (FFT) is widely used to reduce the computation time of the spectral analysis [2]. In this section, we shall make a comparison between the FFT and DFT algorithms.

The DFT needs more basic computations than the FFT: roughly N^2 versus $N \log_2 N$. For example, when the transform length N equals 256, the use of the FFT algorithm reduces the number of computations by a factor of 30. However, real-time speech processing is the most important application at which we are aiming.

Chip count and system cost are our major criteria for a VLSI implementation [3], [4]. A 32-point FFT processor, which has been recently reported [5], requires too much external hardware for our area of application. When this off-chip, overhead is considered, then the DFT algorithm is more efficient.

Furthermore, the DFT needs only a small amount of on-chip control logic because the data flow is very regular. Compared to the FFT, there is no need for complex data handling, such as bit reversal and data shuffling. On-chip control overhead considerably influences the total chip area for any VLSI data processor. Therefore, the DFT algorithm is better suited to keep the chip cost at a minimum [6].

Also, the DFT is more accurate because intermediate results are stored in a double-length accumulator (Section IV). This is not the case for the FFT, which is an "in-place" algorithm. This means that intermediate results are truncated or rounded to a single-precision format each time they are temporarily written into on-chip or off-chip RAM memory. Furthermore, the FFT requires data scaling to prevent arithmetic overflow. The overflow problem can be easily solved for the DFT, by extending the accumulator at the MSB position with a few extra overflow bits.

Manuscript received April 25, 1983.

The authors are with the N.V. Philips Research Laboratories, P.O. Box 80.000, 5600 JA Eindhoven, The Netherlands.

0018-9200/83/1000-0604$01.00 © 1983 IEEE

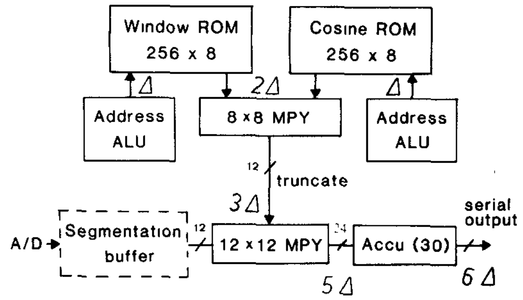


Fig. 1. Simplified block diagram of the DFT processor. Pipeline storage levels are located between each block. The various pipeline depths in multiples of the clock cycle time Δ are also shown.

Another advantage of the DFT algorithm is the fact that it is a serial algorithm, which means that frequency points are calculated one after another. Thus, it is possible to compute exactly those frequency points that are really needed. With the FFT, the entire spectrum is always calculated in parallel.

IV. General Overview of the DFT Processor

A block diagram of the DFT processor is shown in Fig. 1. The values of the Hamming window $W(i)$ and the sine/cosine coefficients are stored in two ROM's, each containing 256 words of 8 bits. Both ROM's have their own address computation unit.

For the computation of every frequency point, the contents of both ROM's are multiplied in an 8 X 8 bit parallel multiplier (MPY). At this point, the products are truncated to 12 bits and then multiplied by the data samples $D(i)$ in a 12 X 12-bit multiplier [7]. These 12-bit speech samples are externally supplied to the DFT chip. Next, the results are added in a 30-bit wide accumulator, of which 7 bits are overflow bits. Finally, 16 significant bits are selected and transferred externally in a sequential way.

The transform length N can be set to 32, 64, 128, or 256 points. Successive frequency points in the spectrum are stepped autonomously by the processor. In addition, an arbitrary frequency that is read in a serial way can be used as a starting point for the spectrum computation.

Whether the real part or the imaginary part of the spectrum is calculated, it is determined by a high level on one of two control pins (SIN, COS). When both pins are kept at a high level, the DFT processor calculates the real and imaginary values alternately: $F_{\mathrm{Re}}(1), F_{\mathrm{Im}}(1), F_{\mathrm{Re}}(2), F_{\mathrm{Im}}(2), F_{\mathrm{Re}}(3), \cdots$. The use of the window function $W(i)$ can also be suppressed by a high level on a third control pin.

The data output is controlled by three pins: ENABLE, NXBIT, and a tristate (data) OUTPUT. When both ENABLE and NXBIT are high, the next bit of a result is transferred externally. This way, the NXBIT pin is used as an external interrupt for the output bit stream. This makes it possible for the serial output transfer rate to be lower than the internal clock frequency.

V. Performance Figures

N						
256	6560	3280	1640	820	410	
128	1640	820	410	204	102	
64	410	204	102	51	25	
32	102	51	25	13	6.4	µsec.
	1	2	4	8	16	#

Fig. 2. Speed of the discrete Fourier transform as a function of chip count (#) and the length of the transform N.

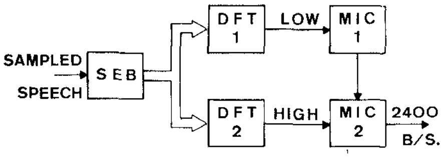


Fig. 3. Block diagram of the transmitter in a 2400-bit/s DFT vocoder.

Because of the regularity of the data flow in the DFT algorithm, pipeline techniques can be used to increase the data throughput on the chip. Pipeline registers are located between each block indicated in Fig. 1. The 12 X 12-bit array multiplier contains another pipeline storage level internally. This one is not shown in the block diagram. This leads to pipeline depths of Δ, 2Δ, 3Δ, 5Δ, and 6Δ where Δ equals the clock cycle time, as shown in Fig. 1. The DFT processor runs at a 10 MHz clock frequency by using these techniques. This enables a single chip to calculate a complete 256-point complex transform every 6.5 ms. When the transform length decreases to 32 points, we only need 102 µs.

Of course, several DFT processors can be used in parallel to reduce the computation time. This is illustrated in Fig. 2 where the computation time is shown as a function of the transform length N and the number of chips that are used in parallel. For example, 8 chips are required to compute a 64-point transform within 51 µs.

VI. Applications

In general, this DFT processor is applicable as a frequency analysis tool in real time speech processing such as DFT vocoders, pitch extractors, filter banks, word recognition systems, and touch-tone detection equipment. In this section, we discuss two application examples.

A. DFT Vocoder

The first is the transmitter of a 2400-bit/s DFT vocoder [4], based on a new algorithm for pitch extraction [8]. In the design of real-time, digital vocoders LPC techniques play a predominating role. The major reason is that LPC algorithms can be efficiently implemented in commercial microcomputers and signal processors [9]. In comparison to an LPC-10 vocoder [4], a DFT vocoder performs better in a realistic environment because of the inherent robustness of this vocoder with respect to bandwidth limitations of the input speech and transmission errors.

Fig. 3 shows a block diagram of the transmitter of this DFT vocoder. This system also uses two other full-custom LSI circuits: a segmentation buffer (SEB) and a 16-bit microprocessor (MIC) [10]. Just for the pitch extraction, we need one DFT chip, one SEB buffer, and one MIC microprocessor. For a complete transmitter of a DFT vocoder, we must add a second DFT chip and another microprocessor.

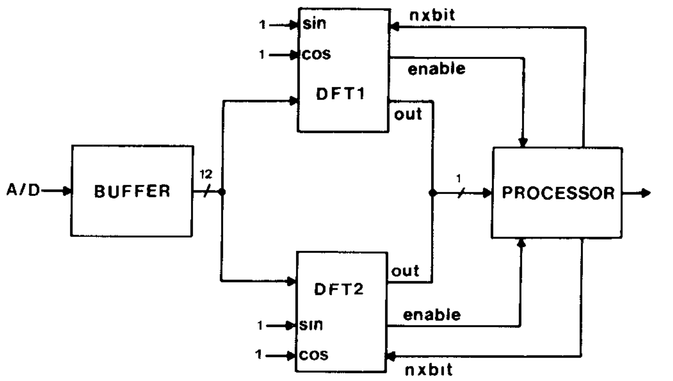


Fig. 4. Block diagram of the filter bank used in a voice recognition system.

In the transmitter input, speech is sampled at 8 kHz and after A/D conversion, the speech samples are buffered in the segmentation buffer chip (SEB). This SEB chip transfers the samples of the current segment of length 256 to both DFT circuits in a recirculating way, at the same time buffering new speech samples. The first DFT processor (DFT1) is programmed with the levels on the various control pins to compute the lower half of the spectrum, while the second one (DFT2) calculates the upper part. Both chips alternately compute the real and imaginary parts of successive frequency components.

These spectral values are transferred to the microprocessors (MIC), which then compute the pitch value and the channel information, and code these results in the right format, to be transmitted at a 2400-bit/s rate.

B. Filter Bank for Word Recognizer

The second application is a filter bank for a speaker-independent word recognition system for isolated words, shown in Fig. 4 [11]. This system is implemented with one segmentation buffer chip, two DFT processors, and one general-purpose signal processor. The functions of both DFT circuits are now slightly different from the vocoder application; the tristate data output pins of both DFT circuits are connected together with the serial input of the signal processor.

Both DFT circuits compute the various frequency points in parallel. Thus, when a summation according to (1) or (2) is finished, the signal processor first reads the output of DFT1. Since the serial input clock of this signal processor runs at a lower speed, the data transfer must be controlled by the appropriate use of the NXBIT pin. An internal counter disables DFT1 as soon as 16 bits have shifted externally. Next, the output of DFT2 is read in the same way. Finally, the signal processor computes the amplitude spectrum and performs the computation of the band filter spectra in the filter bank.

VII. Some Key Function Blocks

In this section, we discuss a few key circuits which have been used in the DFT processor, such as the address ALU's for the ROM's, the array multipliers, and the scan test hardware.

A. The Address Computation Units

A block diagram of the address ALU is shown in Fig. 5. The central part is an 8-bit accumulator, which can be reset to zero by a *clear* command. When this accumulator is enabled, it increments its output every clock cycle with the step Δ:

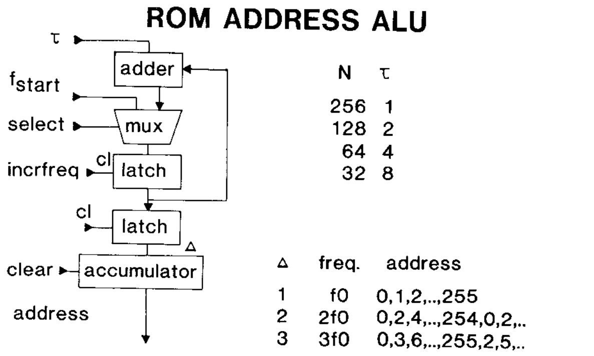


Fig. 5. Block diagram of the address computation unit.

$$0, \Delta, 2\Delta, 3\Delta, \cdots, (N-1)\Delta.$$

The value of Δ is computed in the upper part of the address ALU, which contains an adder, two sample-and-hold registers, and a multiplexer; an increment τ is added here to the old value of Δ every time the calculation of a new frequency point is started. This increment τ depends on the length of the transform N: in the sine/cosine address ALU it can be set to 1, 2, 4, or 8 for $N = 256, 128, 64$, or 32 points, respectively. In the address ALU of the Hamming window ROM, τ is always set to 1.

Furthermore, an arbitrary starting frequency point f_{start} can be read in by the multiplexer MUX.

It can easily be shown that the value Δ defines the frequency of the sine/cosine function, which is read from the ROM table (f_0 is the base frequency)

Δ	freq.	address series
1	f_0	$0, 1, 2, \cdots, 127, 128, 129, \cdots, 254, 255$
2	$2f_0$	$0, 2, 4, \cdots, 254, 0, 2, \cdots, 252, 254$
3	$3f_0$	$0,3,6, \cdots, 255,2,5, \cdots .254,1,4, \cdots, 250,253.$

The Hamming window ROM contains 256 coefficients, which have been computed according to (3). The other ROM contains the sine function with period $1/f_0$. Because a cosine function can be obtained from a sine function by a phase shift of 90°, the cosine is extracted from this sine table by manipulating the two most significant address bits.

B. Array Multipliers

The multiplication of 2 two's complement numbers X (n bits) and Y (m bits) results in a $n + m - 1$ bit product P. The value P_v of this product equals

$$P_v = X_{n-1} Y_{m-1} 2^{m+n-2} + \sum_{i=0}^{n-2} \sum_{j=0}^{m-2} X_i Y_j 2^{i+j} - \sum_{j=0}^{m-2} X_{n-1} Y_j 2^{n-1+j} - \sum_{i=0}^{n-2} Y_{m-1} X_i 2^{m-1+i}. \tag{4}$$

In conventional fixed point notation, the most significant bits of the operands, X_{n-1} and Y_{m-1}, carry negative weight. As a

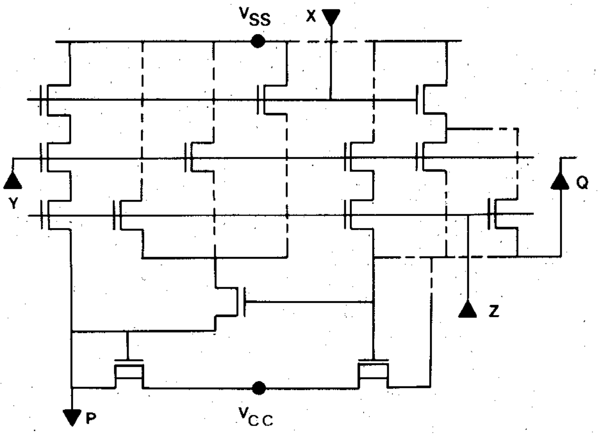


Fig. 6. Circuit diagram of the inverting full adder, used in the array multipliers. The dashed lines represent the undercrossings (see also Fig. 11).

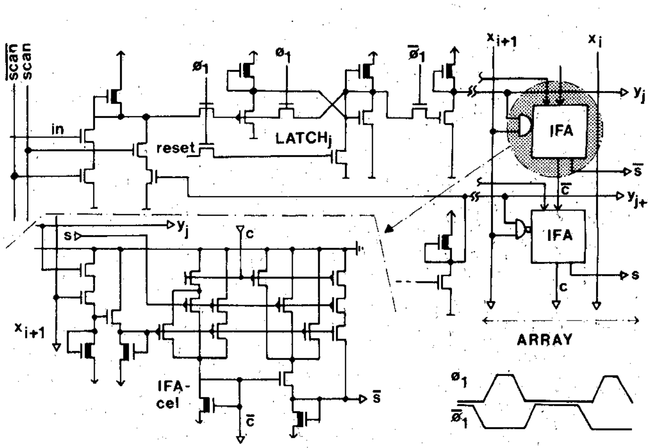


Fig. 7. Circuit diagram of the scan test logic at the multiplier input registers. Shown are two inverting full adders (IFA), one latch, and the AND-OR-INVERT gate as its input.

result, negative product bits arise, corresponding to the third and fourth term in (4). In array multipliers, all partial products $X_i\ Y_j$ are generated in parallel by $m \times n$ AND gates. Hence, the basic problem in the design of a multiplier is to reduce the time needed to add (or subtract) all partial products with equal weights 2^{i+j}.

In the DFT chip, the carry-save principle [7] is used, which requires $m(n - 1)$ full adders for an $m \times n$-bit array multiplication. The inverting full adder cell (IFA) that has been applied in the multiplier is shown in Fig. 6. It is basically an AND-OR-INVERT gate, in which the carry is used to form the sum function. This results in a compact cell. This cell has been used in both the 8 X 8-bit and the 12 X 12-bit array multipliers. Static enhancement-depletion logic synchronized by sample-and-hold latches has been used throughout the chip. These latches function as pipeline registers for improving the throughput in the data path, as has been discussed in Section V.

C. Built-In Test Hardware

Owing to increased circuit complexity, the pin-to-gate ratio in LSI and VLSI circuits is decreasing. Testability therefore becomes an important problem.

In the DFT processor, the scan test has been built in by extending the latches on the chip with a switch at their input. This switch is basically a 2-input AND-to-NOR gate, controlled by signals SCAN and $\overline{\text{SCAN}}$. Fig. 7 shows how this test hardware has been incorporated in the input registers of the array multipliers.

When SCAN = 0, they act as conventional latches. When SCAN = 1 the sample-and-hold latches are daisy-chained by connecting an output of one latch to the input of its nearest neighbor.

If the two ends of the shift register are connected to a pair of I/O pins, the states of all of the latches are easily accessible. In this SCAN mode, the circuit can be set to any desired state by supplying the test sequence to the input of the shift register chain of latches. The sequence of bits that appears on the output of this chain can be checked for whether the circuit is in a given state.

The incorporation of the scan test logic did not influence the total chip area very much, mainly due to the fact that we converted the scan path logic to a very compact AND-OR-INVERT gate.

Technology	2 µ ED - NMOS
Layout pitch	Al:5.5µ,Poly:4.5µ,Diff.:5µ
Threshold voltages	1.0V,0.15V,-1.9V,-1.1V
Effective channellength	1.6 µ
N^+-undercrossing	As,100 ohm/□
Die size	3.8 x 3.3 = 12.5mm^2
Number of transistors	20.000
Packing density	1600 transistors/mm^2
Power supply	5V
Clockfrequency	Fmax. = 10 MHz
Power dissipation	400 mW (f = 10 MHz)
TTL compatible I/O	
Package	28 pin DIP

Fig. 8. Main design features of the DFT circuit.

VIII. Technology

The main technological features of the DFT processor are given in Fig. 8. The chip has been manufactured in a 2 µm NMOS process with 500 Å gate oxide and LOCOS isolation. A photomicrograph of the DFT circuit is shown in Fig. 9.

The high packing density is the result of the regularity of the layout and the use of implanted n^+ crossings under the LOCOS field oxide (Fig. 10). These undercrossings lead to a reduced layout area of the logic circuits and also to lower wiring capacitances, thus improving the performance. The size of the basic cell in the array multipliers (full-adder + AND gate), for instance, is only 64 X 95 µm^2. A photomicrograph of this cell, in which undercrossings have extensively been used, is shown in Fig. 11. This layout corresponds to the circuit diagram given in Fig. 6.

Several threshold voltages have been applied in the DFT processor. Soft depletion transistors with $V_{TH} = -1.1$ V, for instance, are used in the internal drivers and TTL output buffers to reduce the active power dissipation. As a result, the total power consumption is only 400 mW when the DFT circuits operates at its maximum clock frequency of 10 MHz.

IX. Design Method and CAD Tools

The most important problem in VLSI design is to deal with complexity. During the design of real-time signal processing

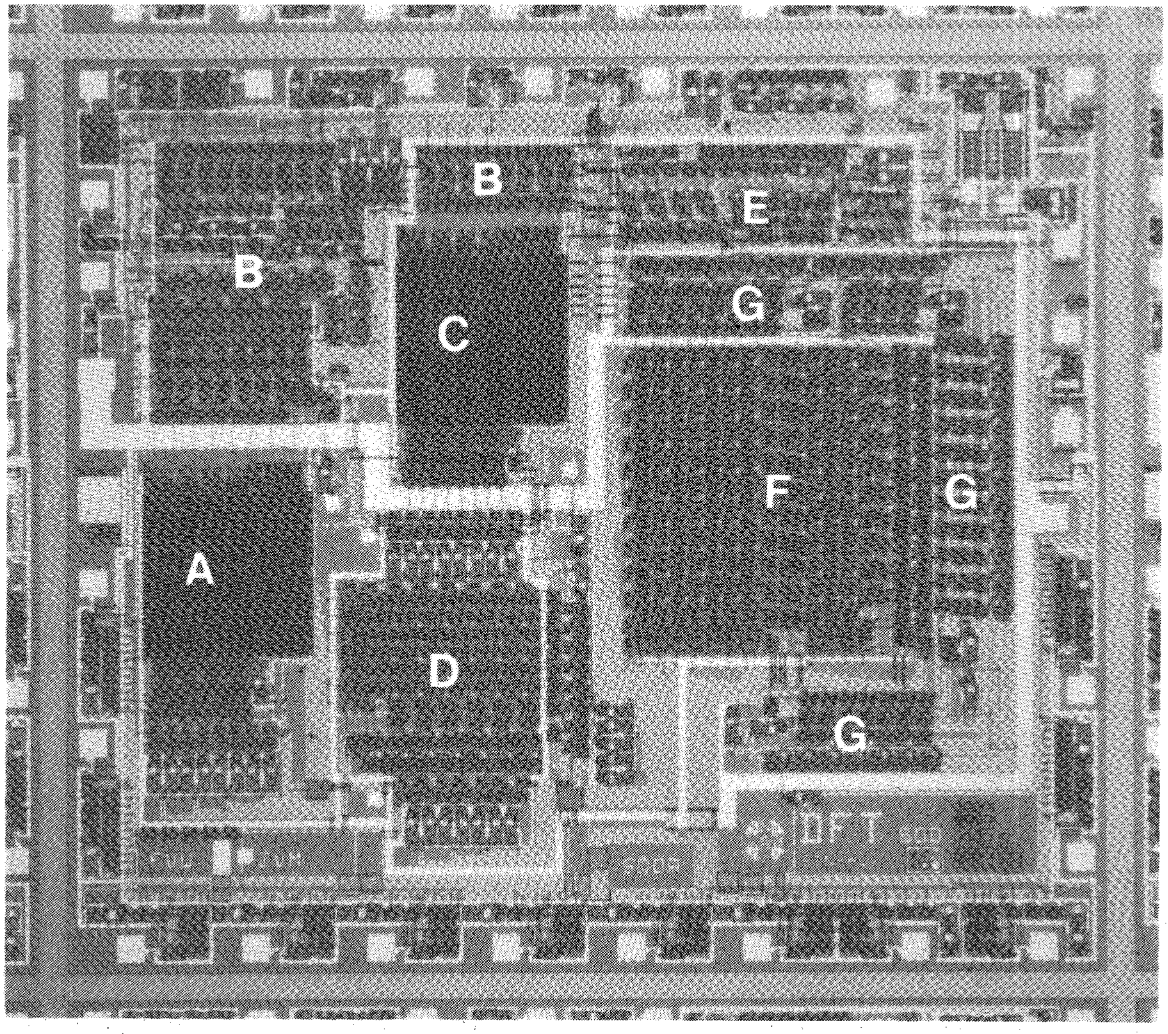


Fig. 9. Photomicrograph of the DFT process, showing (a) sine/cosine ROM, (b) address ALU's (c) Hamming-window ROM, (d) 8 × 8 MPY, (e) timing and control, (f) 12 × 12 MPY, and (g) accumulator/shifter.

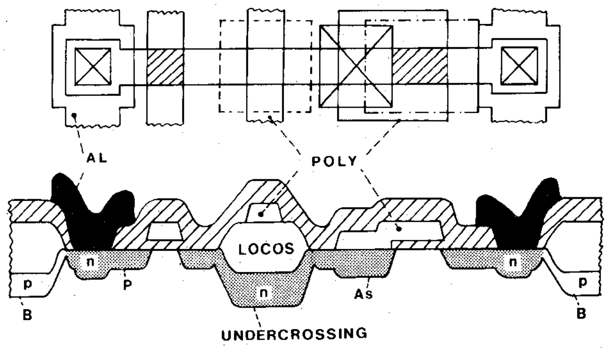


Fig. 10. Cross section of a typical inverter with implanted n^+ undercrossing.

circuits, we also must take care of the stringent performance requirements. The specific design method and CAD tools should therefore be chosen with care.

Because data throughput is one of the most important requirements in digital signal processing, along with a small silicon area and low power consumption, we have chosen a top-down specification and a bottom-up implementation and verification approach. Fig. 12 shows how the various CAD tools which have been used during the design, interact. Five different programs have been used: a logical simulator (PHILSIM), a circuit simulator (PHILPAC), a layout description language (CIRCUITMASK), a circuit extractor (LOCAL), and a logic gate extractor (FUN 2).

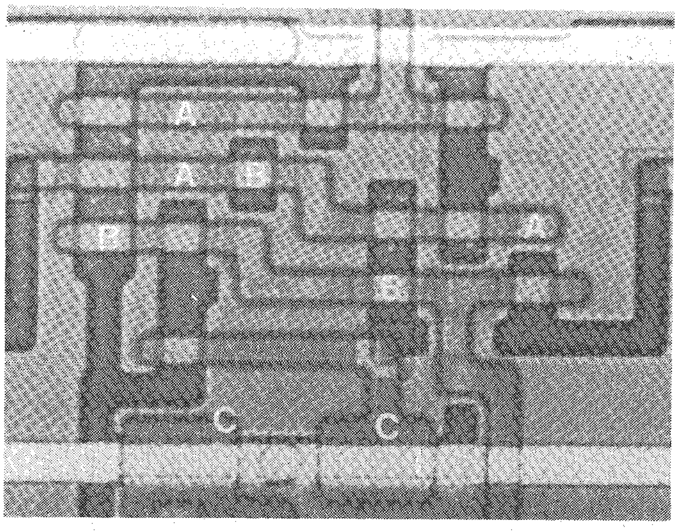


Fig. 11. Photomicrograph of the inverting full adder, showing (a) n^+ undercrossings, (b) enhancement transistors, and (c) depletion loads.

Simulation at the register transfer level was skipped because the DFT algorithm is simple and partitioning in functional blocks is obvious and straightforward. During the top-down

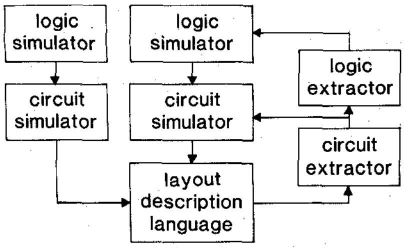


Fig. 12. Design method and CAD tools that have been used during the design of the DFT circuit.

specification (Fig. 12), a logic simulation was performed for the complete chip. The width of the data path, however, was reduced from 12 to 4 bits, resulting in acceptable run times.

The major functional blocks, such as address ALU's, accumulator, and multipliers, have been implemented during the layout design as "bit-slices." Since this chip layout design was more or less "full-custom," there was a need to verify the artwork bottom-up. A circuit extractor (LOCAL) was used which regenerates the complete circuit schematically, including W-over-L ratios and all parasitic capacitances. Circuit simulations then give results which are much closer to reality. Nevertheless, a full circuit analysis of more than 300 transistors is about the upper limit for this circuit simulator.

For complex data processors such as the DFT chip, there was a need to verify larger parts of the circuit, even at the expense of lower accuracy. To solve this problem, a program (FUN2) was written which extracts logic gates from the transistor circuit diagram generated by LOCAL. This program, which is primarily intended for ED-NMOS logic, first finds all "pull-up" nodes in the network. Next, it detects all pass transistors. Finally, starting from the "pull-up" nodes, it searches paths to ground and generates a network of AND-OR-INVERT logic gates. This network can be inputted to the PHILSIM logic simulator for logic verification. In the logic network extractor, primitive delay modeling has been built in since all W-over-L ratios and load capacitances C_l are known. These features make a complete bottom-up verification of the DFT chip ("from bonding pad to bonding pad") possible.

X. Summary

In this paper, we have described a 2 μm NMOS circuit which computes a 256-point complex Fourier transform in 6.5 ms. Computing this transform by way of a direct evaluation instead of the fast Fourier transform has a number of advantages. In speech processing, we do not need the speed of the FFT algorithm. The direct evaluation (DFT) has a regular data flow, does not require off-chip overhead, and has almost no on-chip control logic. Most of these advantages make it better suited for a VLSI implementation with low chip-count and system cost as the major goals.

The use of state-of-the-art technology with n$^+$ undercrossings, together with a full-custom design approach using bit slices, has resulted in a very small silicon area (12.5 mm^2).

This DFT circuit is a powerful frequency analysis tool for small low cost speech processing systems.

Acknowledgment

The authors wish to acknowledge the contributions made by J. Lagerberg, G. Pasman, L. Pfennings, the members of the maskershop under the direction of F. Stoots, and F. Smolders and his colleagues in the processing department.

Special thanks are also due to R. Sluyter, G. Bosscha, H. Kotmans, P. Zuidweg, and M. van der Meulen for the many stimulating discussions.

References

[1] J. L. van Meerbergen and F. J. van Wyk, " A 2 μm NMOS 256-point discrete Fourier transform processor," *Dig. Tech. Papers, Int. Solid State Circuits Conf.*, Feb. 1983, pp. 124–125.

[2] L. R. Rabiner and B. Gold, *Theory and Application of Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1975, pp. 356–371.

[3] R. J. Sluyter, H. J. Kotmans, and A. van Leeuwaarden, "A novel method for pitch extraction from speech and a hardware model applicable to vocoder systems," in *Proc. Int. Conf. Acoust., Speech, Signal Processing*, Apr. 1980, pp. 45–48.

[4] G. J. Bosscha and R. J. Sluyter, "DFT vocoder using harmonic sieve pitch extraction," in *Proc. Int. Conf. Acoust., Speech, Signal Processing*, May 1982, pp. 1952–1955.

[5] B. L. Troutman, G. W. McIver, W. E. Kingsley, and B. H. Whalen, "A 2μ CMOS/LSI 32-point FFT processor," *Dig. Tech. Papers, Int. Solid State Circuits Conf.*, Feb 1982, pp. 26–27.

[6] R. W. Broderson, "Signal processing using MOS-VLSI technology," in *VLSI Electronics: Microstructure Science*. N. G. Einspruch, Ed., New York: Academic, 1981, vol. 2, pp. 189–226.

[7] N. F. Benschop and L. C. M. Pfennings, "Compact NMOS array multipliers with inverting full adders," *Philips J. Res.*, vol. 36, pp. 173–194, 1981.

[8] R. J. Sluyter, H. J. Kotmans, and T. A. C. M. Claasen, "Improvements of the harmonic sieve pitch extraction scheme and an appropriate method for voiced-unvoiced detection," in *Proc. Int. Conf. Acoust., Speech, Signal Processing*, May 1982, pp. 188–191.

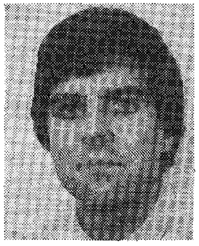
[9] J. A. Feldman, E. M. Hofstetter, and M. L. Malpass, "A compact, flexible LPC vocoder based on a commerical signal processing microcomputer," *IEEE J. Solid-State Circuits*, vol. SC-18, pp. 4–9, Feb. 1983.

[10] P. Zuidweg, J. van Meerbergen, and M. van der Meulen, "Custom LSI chip set for speech analysis," in *Proc. Int. Conf. Acoust., Speech, Signal Processing*, May 1982, pp. 521–524.

[11] M. Kuhn, R. Geppert, and H. Tomaschewski, "Isolated word recognition with a real time DFT circuit," in *Proc. FASE/DAGA*, Goettingen, West Germany, Sept. 1982, pp. 963–966.

Jef L. van Meerbergen was born in Aarschot, Belgium, in 1951. He received the engineering and Ph.D. degrees from the University of Louvain in 1975 and 1980, respectively. His dissertation was on the device physics of silicon solar cells.

In 1979 he joined the Philips Research Laboratories, Eindhoven, The Netherlands, where he is now engaged in the design of MOS digital circuits and signal processors.



Frans J. van Wyk was born in The Hague, The Netherlands, in 1957. He received the degree of Ingenieur in electrical engineering from the Delft University of Technology, Delft, The Netherlands, in 1982. The design of an NMOS LSI circuit for real time signal processing was the subject of his thesis research. At the university he was also employed as a technologist in the Integrated Circuits Laboratory from 1979 to 1981, working on bipolar processes and devices.

In 1980 he was employed as a summer student at the N.V. Philips' Research Laboratories, Eindhoven, The Netherlands where he worked on NMOS integrated circuit design. Upon receiving his degree, he again joined the Research Laboratories of N.V. Philips in Eindhoven, where he is presently engaged in the definition and design of NMOS and CMOS VLSI circuits for professional and consumer applications.