the consistency of the estimate when the noise has an unknown color. In Fig. 3, the observation lasts 600 symbol periods. The SNR varies from 0 to 18 dB.

The results thus obtained are satisfactory; however, the "best" choice of a sequence $f(n)$ is currently under investigation.

IV. Conclusion

When cyclostationarity is induced at the transmitter, we have shown that some second-order cyclo-spectra provide a way of identifying the unknown channel. In contrast to the conventional approches, the consistency of the proposed method is achieved when the observation is corrupted by interference, the second-order structures of which are unknown. We propose in Table IV a summary of the main points developed in the paper.

References

[1] Z. Ding, "Characteristics of band-limited channels unidentifiable from second-order statistics," *IEEE Signal Processing Lett.*, vol. 3, pp. 150–152, May 1996.

[2] A. J. van der Veen, "Resolution limits of blind multi-user channels identification schemes: The band-limited case," in *Proc. ICASSP*, 1996, pp. 2722–2725.

[3] Ph. Ciblat and Ph. Loubaton, "Égalisation aveugle au second ordre: le cas de signaux à bande limitée," *actes du GRETSI*, Grenoble, France, Sept. 1997.

[4] T. Kailath, *Linear Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1978.

[5] K. A. Meraim, P. Loubaton, and E. Moulines, "A subspace algorithm for certain blind identification problems," *IEEE Trans. Inform. Theory*, vol. 43, pp. 499–511, Mar. 1997.

[6] K. Abed-Meraim, J. F. Cardoso, A. Gorokhov, P. Loubaton, and E. Moulines, "On subspace methods for blind identification of single-input/multi-output FIR systems," *IEEE Trans. Signal Processing*, vol. 45, pp. 42–56, Jan. 1997.

[7] E. Moulines, P. Duhamel, J. F. Cardoso, and S. Mayrargue, "Subspace methods for the blind identification of multichannel FIR filters," *IEEE Trans. Signal Processing*, vol. 43, pp. 516–526, Feb. 1995.

[8] G. B. Giannakis, "A linear cyclic correlation approach for blind identification of FIR channels," in *Proc. Asilomar Conf.*, 1994, pp. 420–424.

[9] ____, "Filterbanks for blind channel identification and equalization," in *Proc. Wavelet, Subband, Block Transforms Commun. Symp.*, Hoboken, NJ, Mar. 1997.

[10] E. Serpedin and G. B. Giannakis, "Blind channel identification and equalization with modulation induced cyclostationarity," in *Proc. 31st Conf. Inform. Sci. Syst.*, Baltimore, MD, Mar. 1997.

[11] M. K. Tsatsanis and G. B. Giannakis, "Coding induced cyclostationarity for blind channel equalization," in *Proc. 29th Conf. Inform. Sci. Syst.*, Baltimore, MD, Mar. 22–24, 1995, pp. 685–690.

[12] G. B. Giannakis, "Filterbanks for blind channel identification and equalization," *IEEE Signal Processing Lett.*, vol. 4, pp. 184–187, June 1997.

[13] A. Chevreuil and P. Loubaton, "Modulation and second-order cyclostationarity: Structured subspace method and identifiability," *IEEE Signal Processing Lett.*, pp. 204–206, July 1997.

[14] D. Slock, "Blind fractionnally spaced equalizers, perfect reconstruction filter banks and multichannel linear prediction," in *Proc. ICASSP*, 1994, vol. 4, pp. 585–588.

[15] L. Tong, G. Xu, and T. Kailath, "A new approach to blind identification and equalization of multi-path channels," in *Proc. Asilomar Conf.*, 1991, pp. 856–860.

[16] M. J. Genossar, H. Lev-Ari, and T. Kailath, "Consistent estimation of the cyclic-autocorrelation," *IEEE Trans. Signal Processing*, vol. 42, pp. 595–603, Mar. 1994.

Memory Efficient Programmable Processor Chip for Inverse Haar Transform

G. A. Ruiz and J. A. Michell

***Abstract*— In this correspondence, a processor chip programmable between $N = 8$ and $N = 1024$ for the unidimensional inverse Haar transform (1-D-IFHT) is presented. The processor uses a low latency data-flow with an architecture that minimizes the internal memory and an adder/subtracter as the only computing element. The control logic has a single and modular structure and can be easily extended to longer transforms. A prototype of the 1-D-IFHT processor has been implemented using a standard-cell design methodology and a 1.0-μm CMOS process on a 11.7 mm^2 die. The maximum data rate is close to 60 MHz.**

***Index Terms*—Digital signal processors, Haar transforms, image processing, transform coding, very-large-scale integration.**

I. Introduction

Modern digital communications systems require efficient data coding in order to reduce transmission and/or storage costs [1], [2]. These coding systems use algorithms of one of the following types: waveform coders, transform coders, and model coders [2]. Although transform coding systems generally use the cosine transform [3], [4], the use of the Haar transform offers certain advantages over the former due to the wavelet characteristics of Haar functions [5].

Three generic VLSI architectures have been proposed in [6] for the hardware implementation of the Haar transform. Each of these has different characteristics in terms of computation time, complexity, input/output format, and pipelinability. Some other important characteristics not considered in [6] are the size of the required memory, the input ordering, and the complexity of control. The selection of one architecture or another depends on the application and the design framework available so that in some cases, the resulting architecture has a structure that is a hybrid of other more general structures. This is the case of the direct Haar transform processor described in [7], whose architecture consists of three stages in pipeline, the last of these having sequential queue architecture.

The on-line computing of the bidimensional Haar transform can be carried out directly by implementing the two-dimensional (2-D) fast transform [8], [9] or indirectly by applying the property of separability [2] using the unidimensional (1-D) transform. The three processor parallel-pipeline architecture described in [8] has been designed for a raster ordering of input, minimizing the internal memory. The simplicity of the architecture proposed by Albanesi and Ferreti [9] is due to the fact that the input data ordering is not a raster ordering and that they use a less general bidimensional Haar-like transform.

This correspondence presents an inverse Haar transform processor (1-D-IFHT) programmable for transform lengths of between $N = 8$ and $N = 1024$. Its architecture, which is a variant of the sequential queue architecture proposed in [6], reduces the internal memory requirement from N to $\log_2 N$. Moreover, the control logic has a modular structure whose complexity increases linearly with $\log_2 N$.

Manuscript received July 25, 1996; revised August 4, 1997. This work was supported by the Spanish Government Research Commission (CICYT). The associate editor coordinating the review of this paper and approving it for publication was Dr. Konstantin Konstantides.

The authors are with Departamento de Electrónica y Computadores, Facultad de Ciencias, Santander, Spain (e-mail: grr@dyvci.unican.es; jmm@dyvci.unican.es).

Publisher Item Identifier S 1053-587X(98)00527-3.

1053–587X/98$10.00 © 1998 IEEE

A prototype chip of the 1-D-IFHT covering an area of 11.7 mm^2 has been implemented using standard cells and a 1-μm CMOS technology. Its maximum data rate is close to 60 MHz, which means that it can be applied in the computing of the 2-D-IFHT at video rates.

II. On-Line Computing of the 1-D-IFHT

Haar functions make up an ortonormal basis used in a wide variety of signal processing applications. The normalization of Haar functions [10] limiting their values to $\{0, \pm 1\}$ makes them easier to handle from a computing point of view. Thus, the normalized Haar functions of length N, power of 2, are defined as

$$\hat{H}_0(i) = 1$$

$$\hat{H}_k(i) = \begin{cases} +1, & \dfrac{N}{2^p} r \le i < \dfrac{N}{2^p}(r + 1/2) \\ -1, & \dfrac{N}{2^p}(r + 1/2) \le i < \dfrac{N}{2^p}(r + 1) \\ 0, & \text{elsewhere} \end{cases}$$

$$0 < k \le N - 1, \quad 0 \le i \le N - 1,$$

$$p = \text{INTEGER}[\log_2 k], \quad r = k - 2^p. \tag{1}$$

Using these functions, the direct and inverse Haar transforms of a sequence x_i of length N are defined as

$$X_k = \sum_{i=0}^{N-1} \hat{H}_k(i)\, x_i, \qquad 0 \le k \le N - 1 \tag{2}$$

$$x_i = \frac{1}{N} \sum_{k=0}^{N-1} A_k \hat{H}_k(i) X_k, \qquad 0 \le i \le N - 1$$

$$\text{being } A_0 = 1 \text{ and } A_k = 2^p \ \forall k > 0. \tag{3}$$

As with the other transforms, increased computing efficiency is achieved by means of fast algorithms. Fig. 1 shows the computing dataflow diagram corresponding to the algorithm of the inverse fast Haar transform for $N = 16$. As can be observed in this figure, the number of addition and subtraction operations required is 30. In general, the number of additions and subtractions needed in order to compute the 1-D-IFHT of length N is $2(N - 1)$ so that its on-line implementation can be carried out using a single adder/subtracter, as is the case of the direct transform [6], [7].

The input sequence ordering conditions the size of internal memory required. The on-line implementation of the algorithm of Fig. 1 using normal input ordering requires the intermediate data to be stored in each computing level. The minimum of internal memory needed is limited by the number of intermediate data generated in the penultimate computing level, that is, $N/2$. This internal memory can be reduced by using the nonnormal input ordering produced by preorder listing [11] of the binary tree of X_i coefficients shown in Fig. 2(a). It can be seen from this tree that 1) any minimum path between the root node (X_0) and any terminal node ($X_{N/2}, \cdots, X_{N-1}$) includes the spectral coefficients needed to generate two points of the output sequence x_i, and 2) the number of nodes common to both of these paths coincides with the number of common intermediate results in the calculation of the respective output points. Taking $N = 16$ as an example, the path X_0–X_1–X_2–X_4–X_8 contains the coefficients needed to calculate x_0 and x_1, whereas the path X_0–X_1–X_2–X_4–X_9 contains the coefficients needed to calculate x_2 and x_3. X_0–X_1–X_2–X_4 are common coefficients so that all intermediate results obtained in the calculation of x_0 and x_1, except for the last one, are also used in the calculation of x_2 and x_3. By producing the preorder listing, the input sequence shown in Fig. 2(b) is obtained. Fig. 3 shows the 1-D-IFHT dataflow diagram resulting from this input ordering. Generally, the

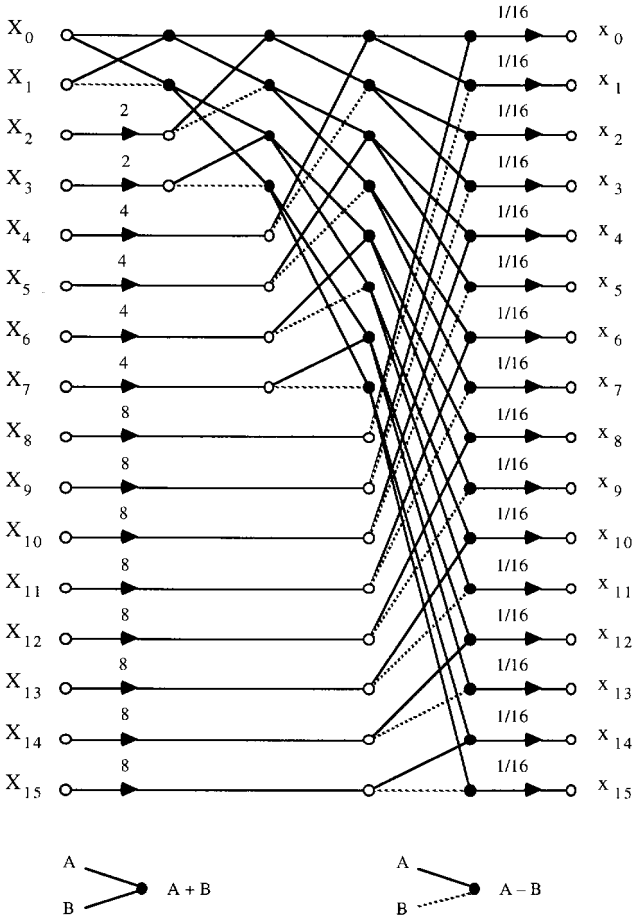


Fig. 1. Data-flow diagram of the 1-D-IFHT for $N = 16$.

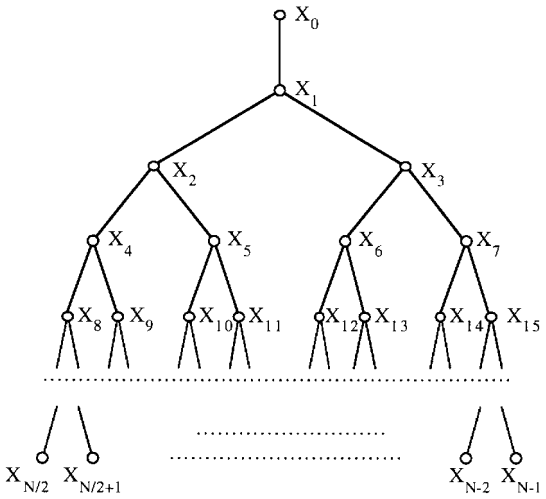
on-line implementation of the 1-D-IFHT with the proposed ordering requires $(\log_2 N))-1$ subtractions to be stored, that is, the number of computing levels minus one, and requires the last intermediate addition to be maintained as well. The internal memory size is thus reduced to $\log_2 N$. Moreover, the output sequence is generated with normal ordering, and its latency ($\log_2 N$) is minimal. Hence, we can say that the sequence in Fig. 2(b) has a minimum latency ordering.

III. Programable Length 1-D-IFHT Processor

The 1-D-IFHT processor has been designed for input sequences with minimum latency ordering, with a transform length programmable between $N = 8$ and $N = 1024$. The output sequence is generated with normal ordering. Its sequential queue architecture (Fig. 4) is made up of

- one adder/subtracter (A/S);
- one STACK memory of nine registers (R0 to R8) to store the subtraction output;
- one (RSUM) register to store the addition output;
- two multiplexers;
- two output registers (ROUT1 and ROUT2);
- three scaling circuits.

The A/S operates with the X_i input data and with the intermediate data stored in the STACK or in RSUM. The number of STACK registers used will vary between 2 (R0 and R1) in the case of $N = 8$ and 9 (R0 to R8) for $N = 1024$. The scaling of the input data by the factor 2^p is carried out with a programmable length shift to the left circuit (ASL[p]). The data output is twofold; one of them contains the even elements of the output sequence (x_{ei}) and the other one



(a)

$X_0 X_1 X_2 X_4 X_8 X_9 X_5 X_{10} X_{11} X_3 X_6 X_{12} X_{13} X_7 X_{14} X_{15}$

(b)

Fig. 2. (a) Binary tree of spectral coefficients. (b) Input sequence with minimum latency ordering for $N = 16$.

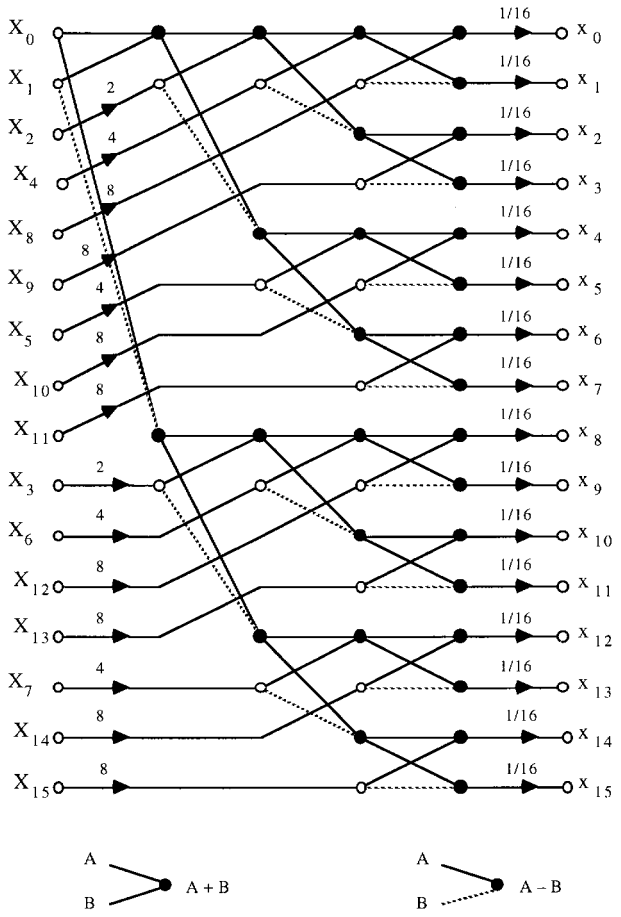


Fig. 3. Data-flow diagram of the 1-D-IFHT with minimum latency ordering for $N = 16$.

the odd elements (x_{oi}). The scaling of the output data by the factor $1/N$ is carried out by means of two fixed length shift to the right circuits (ASR[$\log_2 N$]).

TABLE I
NUMBER OF STANDARD CELLS OF THE CONTROL CIRCUIT AND THE STACK

N	Control Circuit			STACK
	data-path control	addressing generator	Total	
8	90	37	127	203
16	127	60	187	304
32	189	91	280	405
64	232	122	354	506
128	306	168	474	607
256	347	218	565	708
512	387	261	648	809
1024	431	312	743	910

The control circuit generates two functionally different types of signal: data path control and external addressing signals. The data path uses the control signals:

- WRITE (writing in R0);
- START (initialization of a new transform);
- DIV2 (selection of the data input to RSUM),
- SELREG (selection of the intermediate data),
- PUSH/PULL (control of the STACK).

The external addressing signals ($ADD_0, \cdots, ADD_9$) enable the input data with minimum latency ordering to be read from an external memory. The number of addressing lines that remain active is a function of the length of the transform; for example, in the case of $N = 8$, only the first three are used, the rest remaining inactive.

The control signals have the property of being generated recursively for any N from the outputs of a counter driven by the system clock. As an example, the generation of the WRITE signal is now presented. The writing signals WR_N are generated from the outputs of the counter a_r ($r = 0, 1, \cdots, 9$), using the circuit shown in Fig. 5(a). This circuit is made up of an initial stage [Fig. 5(b)] and seven elemental modules [Fig. 5(c)]. A multiplexer selects the WRITE signal corresponding to the programmed transform length. In general, the control signals corresponding to a transform length N can be obtained from those of length $N/2$ in a similar way to that described above. Because of this property, the control logic has a regular structure made up of elemental modules connected in cascade, which reduces its complexity and facilitates its extension to lengths greater than 1024, which is considered to be the maximum in this case. Table I lists the number of standard cells of the control circuit and the STACK as a function of the transform length. As can be observed, the complexity of both circuits, measured in number of cells, increases linearly with $\log_2 N$.

In order to illustrate the operation of the processor, Fig. 6 shows the time diagram corresponding to the calculation of the 1-D-IFHT of the sequence $X_i = \{14, 20, -5, -15, 2, 5, 5, -16\}$, reordered in the form $\{14, 20, -5, 2, 5, -15, 5, -16\}$. This figure also illustrates the movement of data in the registers, as well as the operation of the arithmetic element. The transform is carried out in $N = 8$ clock cycles, and the output is generated with a latency of $\log_2 N - 3$ clock cycles. In the first cycle, the input data $X_0 = 14$ are stored in RSUM. In the second cycle, the processor operates with $\{X_0, X_1\}$, storing the addition (34) in RSUM and the subtraction (-6) in R0. In the third cycle, the contents of RSUM are processed with the input data, $X_2 = -5$, scaled by 2 (1 bit shifted to the left); the new additions (24) are stored in RSUM and the subtraction (44) in R0, shifting the previous content of R0 to R1. In the fourth cycle, the contents of RSUM are processed with the input data, $X_4 = 2$, scaled by 4 (2 bits shifted to the left); in this case, the results of the addition and subtraction, scaled by one eighth (3 bits shifted to the right), make

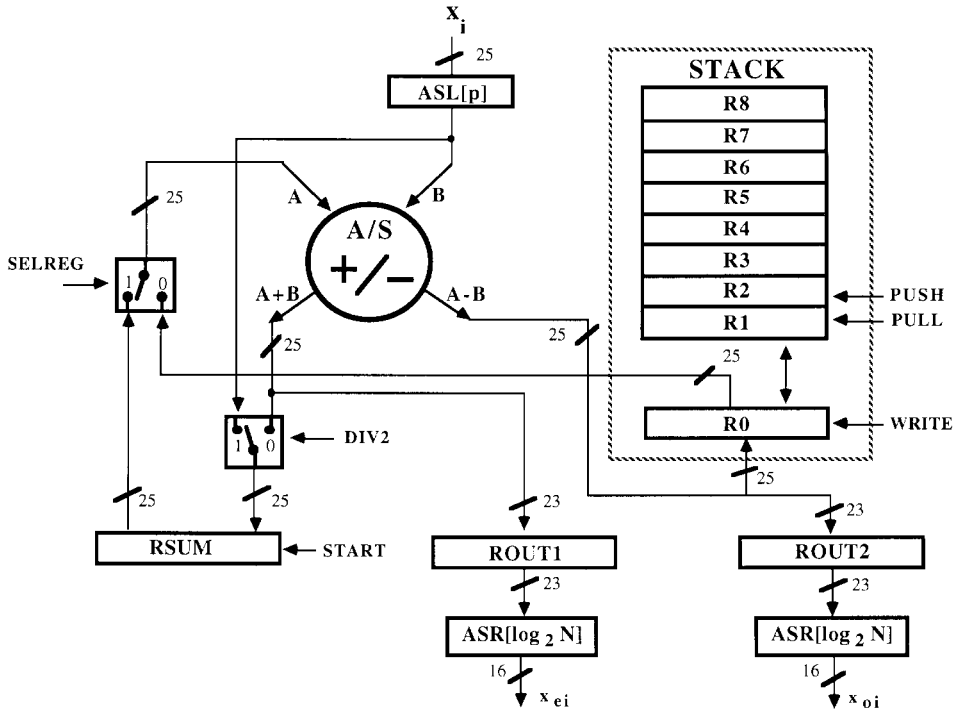


Fig. 4. 1-D-IFHT processor architecture.

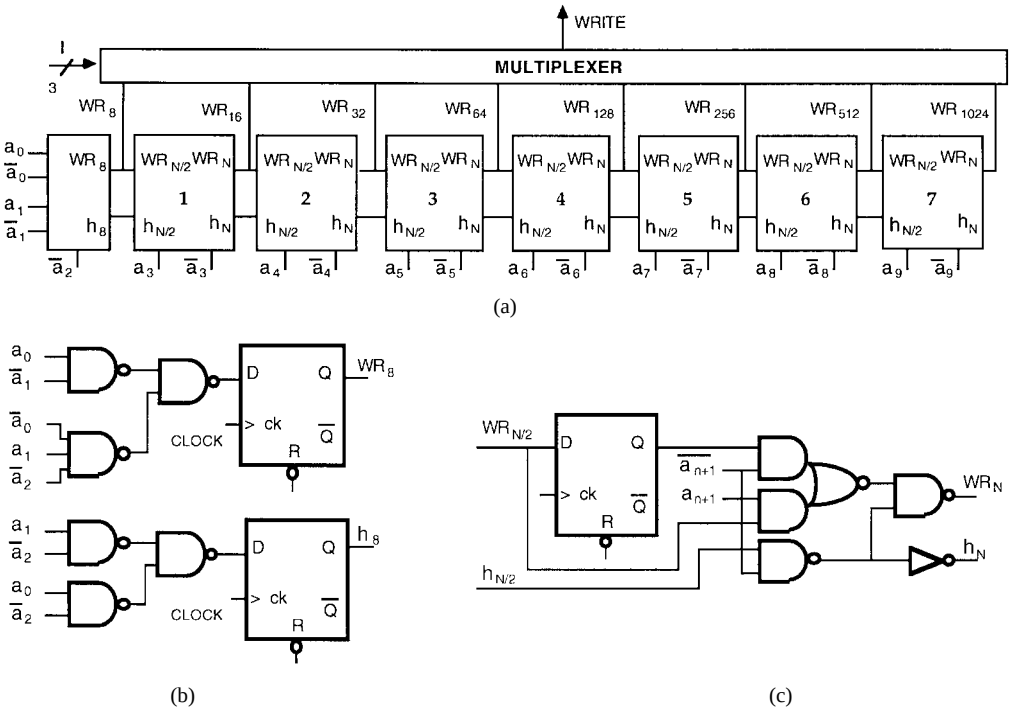


Fig. 5. WRITE signal generator. (a) Complete scheme. (b) Initial stage. (c) Elementary module.

up the first output data, $x_0 = 4$ and $x_1 = 2$ (x_{e0} and x_{o0}). In the fifth cycle, the data of R0 are processed with the input data, $X_5 = 5$, scaled by 4 (2 bits shifted to the left), and the contents of R1 shifts to R0; the results of the addition and subtraction, scaled by one eighth (3 bits shifted to the right), are now the output data $x_2 = 8$ and $x_3 = 3$ (x_{o1} and x_{e1}), respectively. In the sixth cycle, the data of R0 are processed with the input data $X_3 = -15$, scaled by 2, storing the result of the addition (-36) in RSUM and the subtraction (24) in R0. In the seventh cycle, the data of RSUM are processed with the input data $X_6 = 5$, scaled by 4; in this case, the results of the addition and subtraction, scaled by one eighth, are, respectively, the output data $x_4 = -2$ and $x_5 = -7$ (x_{o2} and x_{e2}). In the last cycle, the data of R0 are processed with the input data $X_7 = -16$, scaled by 4; the results of the addition and subtraction, scaled by one eighth, are, respectively, the last output data, $x_6 = -5$ and $x_7 = 11$ (x_{o3} and x_{e3}).

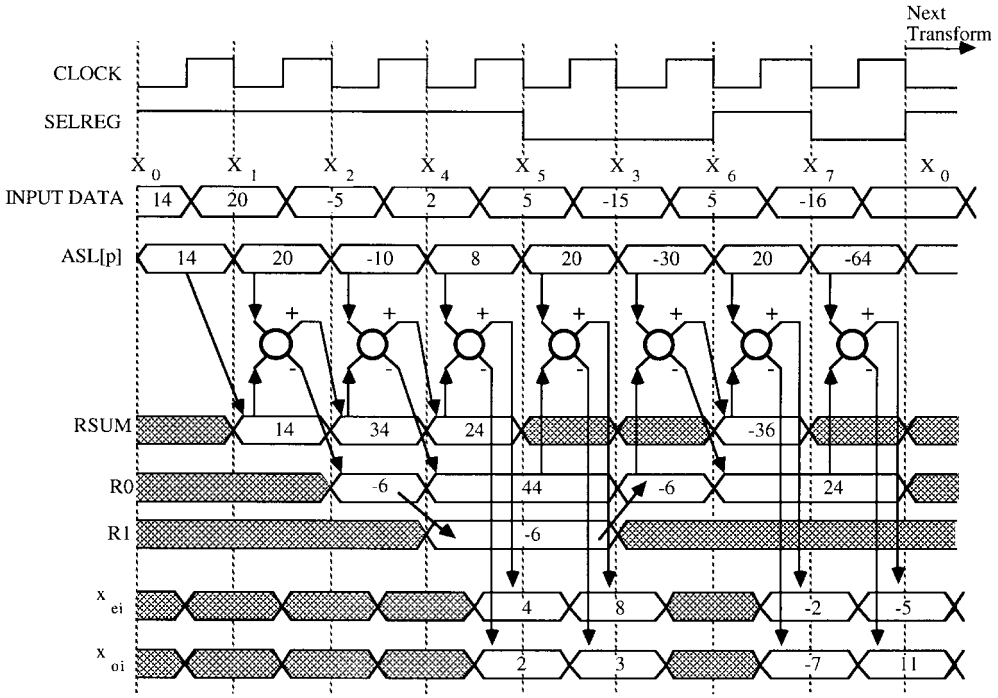


Fig. 6. Timing diagram of the 1-D-IFHT processor for ordered input sequence $\{14, 20, -5, 2, 5, -15, 5, -16\}$.

IV. 1-D-IFHT Chip

A prototype of the 1-D-IFHT processor has been implemented in a chip, using a standard cell design methodology in a 1-μm two-level metal CMOS technology. In order to limit both the size of the chip and the number of I/O pads, a single data output bus has been used, multiplexing the outputs of the adder/subtracter at a frequency of double that of the processor clock, as shown in Fig. 7. Moreover, the processor has two additional outputs: MS, which indicates the completion of the computing of a transform; and EO, which indicates valid data output. The chip occupies an area of 11.7 mm^2 and has been encapsulated in a 64-pin package. Sixty-three pads and 2621 standard cells have been used in the design. The functionality of the prototype, whose microphotograph is shown in Fig. 8, has been fully tested up to frequencies of 25 MHz (the limit frequency of the test equipment). The circuit has also passed more simple tests that enable its maximum data rate to be placed close to 60 MHz. The adder/subtracter, whose structure is of the carry select type, has a propagation time of 13 ns, which limits the operating speed of the processor.

As an application of the 1-D-IFHT chip, the 2-D-IFHT processor shown in Fig. 9 has been designed and is capable of handling black and white images of 256×256 pixels with a coding of 8 bits. This processor is made up of two 1-D-IFHT chips, two RAM memories of $64k \times 16$, which are used to store the intermediate data, and an addressing circuit for these memories based on counters and multiplexers. The first 1-D-IFHT processes each of the 256 rows of the matrix of coefficients and writes the output data in one of the memories. Simultaneously, the second chip reads and processes the data stored in the other memory generated by the first processor, obtaining the output image with raster ordering and a latency of one image. The simulation results for 2-D-IFHT processor, using the standard cell 1.0-μm CMOS library, enable its maximum data rate to be estimated at around 50 MHz.

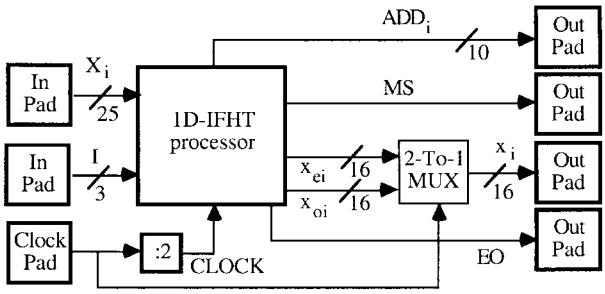


Fig. 7. 1-D-IFHT chip.

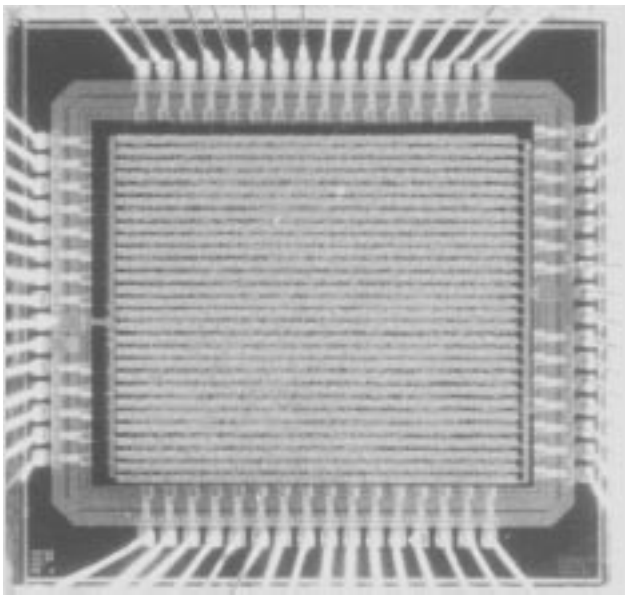


Fig. 8. Microphotograph of the chip.

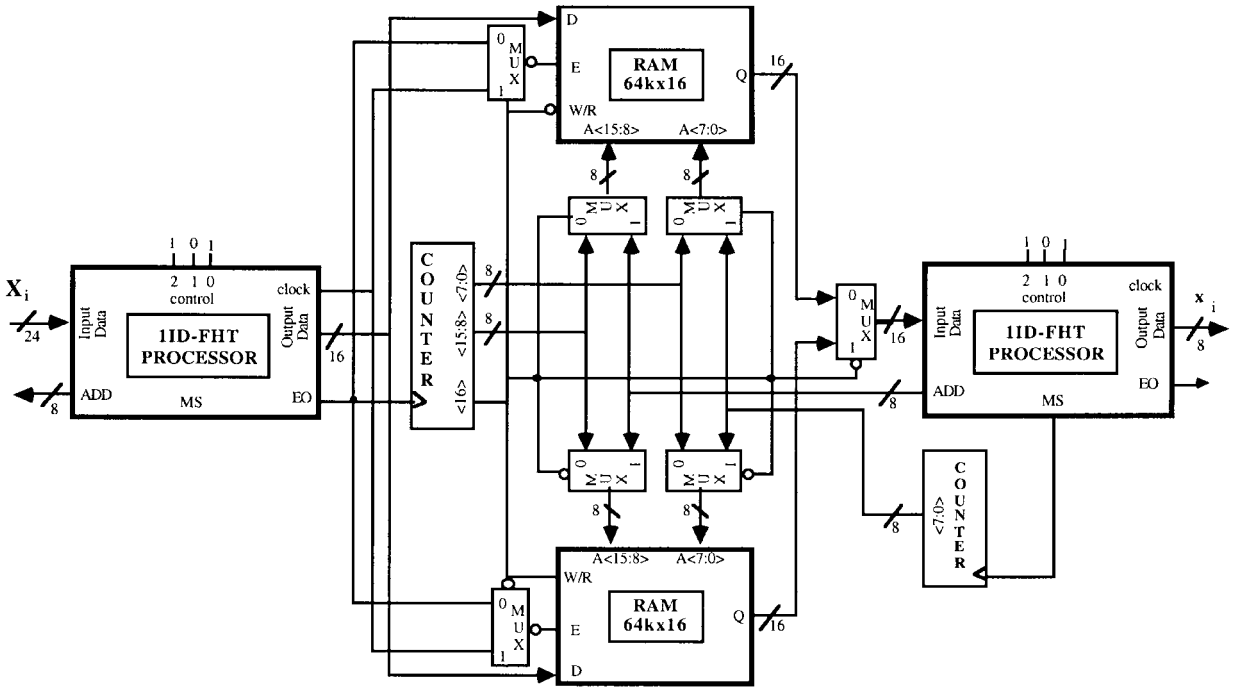


Fig. 9. 256×256 2-D-IFHT processor.

V. Conclusion

This correspondence describes an inverse fast Haar transform processor (1-D-IFHT) of programmable length between 8 and 1024 points. The only computing element of this processor is an adder/subtracter and its data flow has been conceived with the purpose of minimizing the internal memory. One important characteristic is that the control logic and the internal memory have a modular structure that enables it to be enlarged to greater lengths with a linear growth in the hardware of $\log_2 N$. A prototype of the processor has been implemented in a 1-μm CMOS process using a standard cell design methodology. The maximum data rate is close to 60 MHz.

Acknowledgment

The authors would like to thank the independent reviewers of this paper for their constructive comments.

References

[1] R. B. Yates, N. A. Thacker, S. J. Evans, S. N. Walker, and P. A. Ivey, "An array processor for general purpose digital image compression," *IEEE J. Solid-State Circuits*, vol. 30, pp. 244–249, Mar. 1995.

[2] J. S. Lim, *Two-Dimensional Signal and Image Processing.* Englewood Cliffs, NJ: Prentice-Hall, 1990.

[3] P. Lee and F. Huang, "An efficient prime-factor algorithm for the discrete cosine transform and its hardware implementations," *IEEE Trans. Signal Processing*, vol. 42, pp. 1996–2005, Aug. 1994.

[4] N. R. Murthy and M. N. S. Swamy, "On the real-time computation of DFT and DCT through systolic architectures," *IEEE Trans. Signal Processing*, vol. 42, pp. 988–991, Aug. 1994.

[5] C. K. Chui, *An Introduction to Wavelets.* New York: Academic, 1992.

[6] K. J. R. Liu, "VLSI computing architectures for Haar transform," *Electron. Lett.*, vol. 26, no. 23, pp. 1962–1963, Nov. 1990.

[7] A. M. Burón, J. A. Michell, and J. M. Solana, "Single-chip fast Haar transform at MHz rates," in *Proc. 3rd. Int. Workshop Spectral Technol.*, Dortmund, Germany, Oct. 1988, pp. 8–17.

[8] J. A. Michell, A. M. Burón, J. M. Solana, and G. Ruiz, "A three processor parallel-pipelined architecture of the 2D-FHT at video rates," in *Proc. Fourth ISMM/IASTED Int. Conf. Parallel Distributed Comput. Syst.*, Washington, DC, Oct. 1991, pp. 170–174.

[9] M. G. Albanesi and M. Ferreti, "A high speed Haar transform implementation," *J. Circuits, Syst. Comput*, vol. 2, no. 3, pp. 207–226, 1992.

[10] S. L. Hurst, D. M. Miller, and J. C. Muzio, *Spectral Techniques in Digital Logic.* London, U.K.: Academic, 1985.

[11] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *Data Structures and Algorithms.* Reading, MA: Addison-Wesley, 1983.