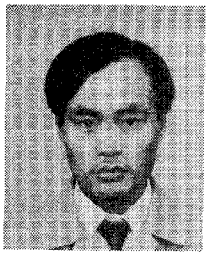
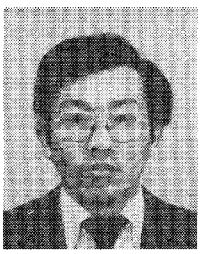


Katsuyuki Kaneko was born in Nagano, Japan, on July 21, 1955. He received the B.S. and M.S. degrees in communication engineering from Tohoku University, Sendai, Japan, in 1978 and 1980, respectively.

He joined the Matsushita Electric Industrial Co., Ltd., Osaka, in 1980. He has been working on the design of MOS LSI circuits.

Mr. Kenko is a member of the Institute of Electronics and Communication Engineers of Japan.



Juro Yasui received the B.S. and M.S. degrees in solid-state engineering from Osaka University, Osaka, Japan, in 1969 and 1971, respectively.

In 1974 he joined the Semiconductor Research Laboratory, Matsushita Electric Industrial Co., Ltd., Moriguchi, Osaka, Japan, where he has been working on the development of integrated circuit processing.

A Radix 4 Delay Commutator for Fast Fourier Transform Processor Implementation

EARL E. SWARTZLANDER, JR., SENIOR MEMBER, IEEE, WENDELL K. W. YOUNG, MEMBER, IEEE, AND SAUL J. JOSEPH, MEMBER, IEEE

Abstract — **This paper describes the development of a semicustom delay commutator circuit to support the implementation of high speed fast Fourier transform processors based on the McCellan and Purdy radix 4 pipeline FFT algorithm. The delay commutator is a 108000 transistor circuit comprising 12288 shift register stages and approximately 2000 gates of random logic realized with 2.5 micrometer design rule CMOS standard cell technology. It operates at a 10 MHz clock rate which processes data at a 40 MHz rate. The delay commutator is suitable for implementing processors that compute transforms of 16, 64, 256, 1024, and 4096 (complex) points. It is implemented as a 4 bit wide data slice to facilitate concatenation to accomodate common data word sizes and to use a standard 48 pin dual-in-line package.**

I. INTRODUCTION

ALTHOUGH the Cooley–Tukey FFT algorithm [1] developed nearly two decades ago has made it possible to apply digital signal analysis techniques to many applications, many others (e.g., radar and sonar beam forming, adaptive filtering, communications spectrum analysis, etc.) require both flexibility and speed that exceeds the present state of the art. Currently, there are three approaches for signal processing: software implemented on general purpose computers, software implemented on a general purpose computer augmented with a Programmable Signal Processor (PSP), and custom hardware development. Software only and Software-PSP implementations are adequate when the spectral bandwidth is under 10 MHz. Custom processors achieve analysis bandwidths of 10–50 MHz, but most are optimized for a specific application and would require extensive (and expensive) redesign to modify them to suit other applications. Thus general purpose computers with or without PSP augmentation are too slow while custom processors lack the required flexibility.

Current signal processing systems require many diverse functions: transform computation, time and frequency domain vector processing, and general purpose computing. We are developing a growing family of building block modules to facilitate the development and implementation of such systems on a semicustom basis. The result is the ability to quickly develop high performance signal processing systems for a wide variety of algorithms. The use of predesigned and precharacterized modules reduces cost, development time, and most importantly, risk. The initial set of modules was described in 1983 [2]. The modules defined include a data acquisition module, building block elements that are replicated to realize pipeline FFT and inverse FFT modules, a frequency domain filter module, a power spectral density computational module, and an output interface module.

The modules all have separate data and control interfaces. The separation of the data and control is analogous to the Harvard mainframe computer architecture which uses separate data and instruction memories to eliminate

Manuscript received April 11, 1984; revised June 14, 1984.

E. E. Swartzlander, Jr. and W. K. W. Young are with TRW Defense Systems Group, Redondo Beach, CA 90278.

S. J. Joseph is with AT&T Bell Laboratories, Allentown, PA 18103.

0018-9200/84/1000-0702$01.00

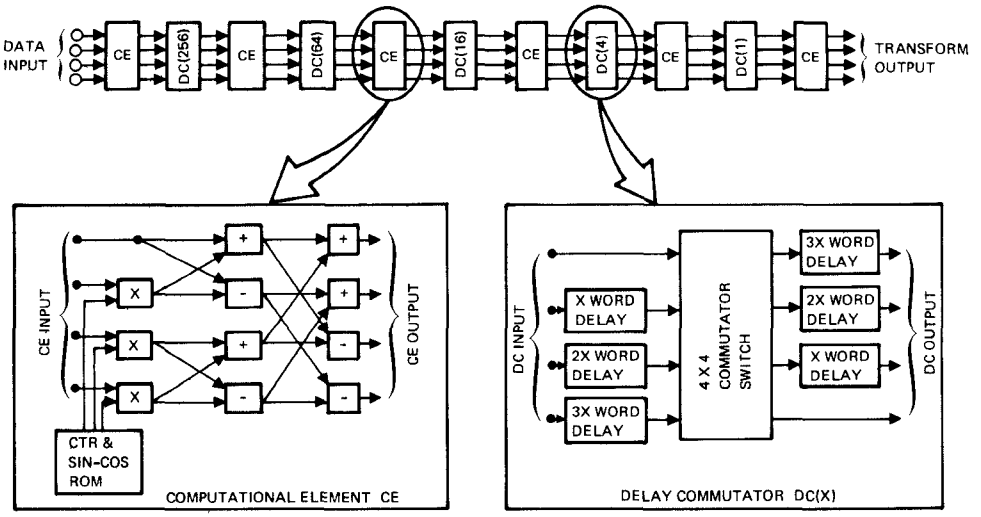


Fig. 1. Pipeline FFT architecture.

the "von Neumann bottleneck." In signal processing the separation of data and control allows the simple data interfaces to operate at high speed while the more flexible and complex control interfaces operate at a slower rate. All data interfaces satisfy a common interface protocol so that modules can be connected together to form architectures that match the data flow of each specific system.

II. Fourier Transform Algorithm Selection

Due to its importance in signal processing, the Fourier transform module was selected for initial development. The choice of algorithm for Fourier transform computation depends on many factors including the system requirements, component technology, and computational environment. There are three main classes of algorithms: the discrete Fourier transform (DFT), the fast Fourier transform (FFT) [1], and the Winograd Fourier transform (WFT) [3]. Each is optimum for certain situations. For example, if only a relatively small number of spectral components are required (as may occur in some speech and beam forming applications) then the DFT may be optimum because of its simple control and negligible memory demands, even though it may involve more arithmetic than the other approaches [4]. If the complete spectrum of long data sequences is required, either the FFT or WFT is generally best. The WFT minimizes multiplication operations but involves rather complex data manipulation and control sequencing, so it is generally used for minimum hardware mini or microcomputer based implementations. The FFT is attractive for VLSI implementation because of its modularity: processors can be realized based on repeated use of a single butterfly computation with simple control and data manipulation.

Recently descriptions of two "single chip" transform processors have been published [4], [5]. Both of these designs are tailored to perform transforms of specific lengths (i.e., 32 or 256 points). Concatenation of the short transforms computed by these implementations to compute the transform of long sequences (e.g., 1K, 4K, or 16K points) often requires complex logic, thereby mitigating the advantages of the VLSI realizations. They also lack the flexibility to efficiently transform sequences of varying lengths as may be required for many applications.

III. The Pipeline FFT Algorithm

Pipeline FFT algorithms are a small subset of the many Fourier transform algorithms that have been developed over the last two decades [6]. The pipeline algorithms [7], [8] are well suited for signal processing applications, where high data throughput is the dominant requirement. They are well suited to hardware implementation, due to their inherent modularity. A K^n length FFT is implemented by sequentially linking n modules where each module performs a radix K butterfly. Since K data paths are used, the pipeline processor achieves a data rate of K times the intermodule clock rate. The clock rate is independent of the transform length.

Our FFT processor uses the radix 4 pipeline algorithm developed a decade ago by McClellan and Purdy [7]. It represents an extension to radix 4 of the pipeline FFT concepts developed by Groginsky and Works [8]. With the radix 4 pipeline algorithm, data passes through a pipeline network comprised of computational elements and delay commutators as shown in Fig. 1. An important feature of this algorithm and architecture is that only two types of elements are used: computational elements and delay commutators. Only minor changes are required to implement forward and inverse transforms of lengths that are powers of 4 from 16 to 4096 points. The changes involve varying the number of stages connected in series, changing the counter step size on the computational elements, and changing the length of the delays on the delay commutator. The computational element performs a four point discrete Fourier transform in 22 bit floating point arithmetic with single chip adders, subtractors, and multipliers [9]. The

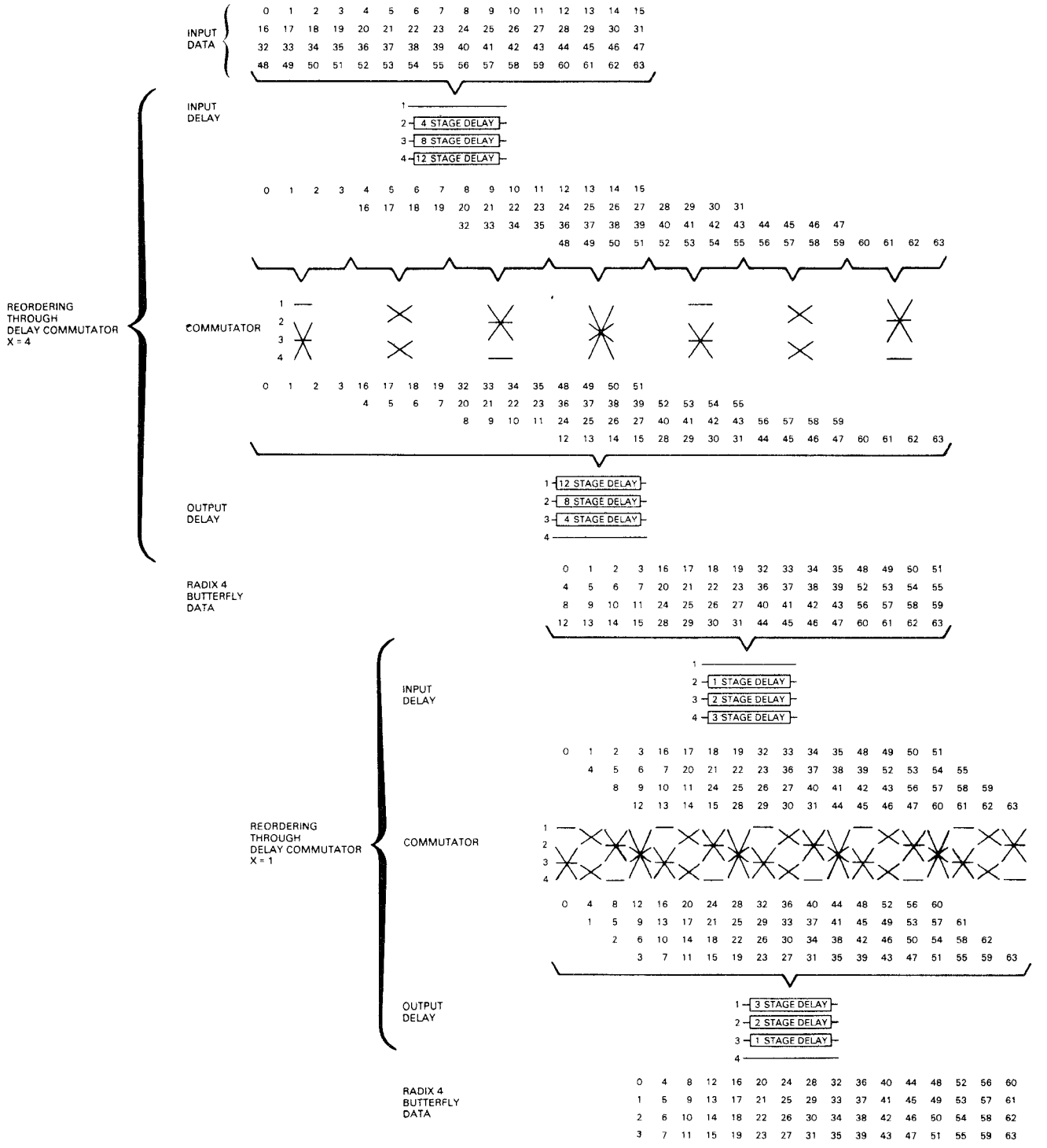


Fig. 2. Data patterns through two delay commutators for a 64 point FFT (after [10, p. 611]).

delay commutator reorders the data between computational stages as required for the FFT algorithm.

The interstage data reordering required at stage i in the implementation of a 4^n point transform is a base 4 digit reversal [10] of the data elements in a 4×4^n matrix. When data enter the delay commutator they pass through four parallel delay lines which skew the data. The first data path receives no delay, the second receives a delay of 4^{n-i-1}, the third receives a delay of $2 \times 4^{n-i-1}$, and the fourth receives a delay of $3 \times 4^{n-i-1}$. The data then pass through the commutator switch, where they are switched to selected data paths. Lastly, the data are deskewed through a second set of delay lines.

The routing of data that occurs in processing a 64 point transform with the radix 4 pipeline FFT algorithm is graphically charted in Fig. 2. Input data numbered 0–63 are shown on four parallel streams at the top of the figure. The action of the first delay commutator (set for $X = 4$) is shown. It transforms the four streams of data separated by 16 points into four streams where the data are separated by

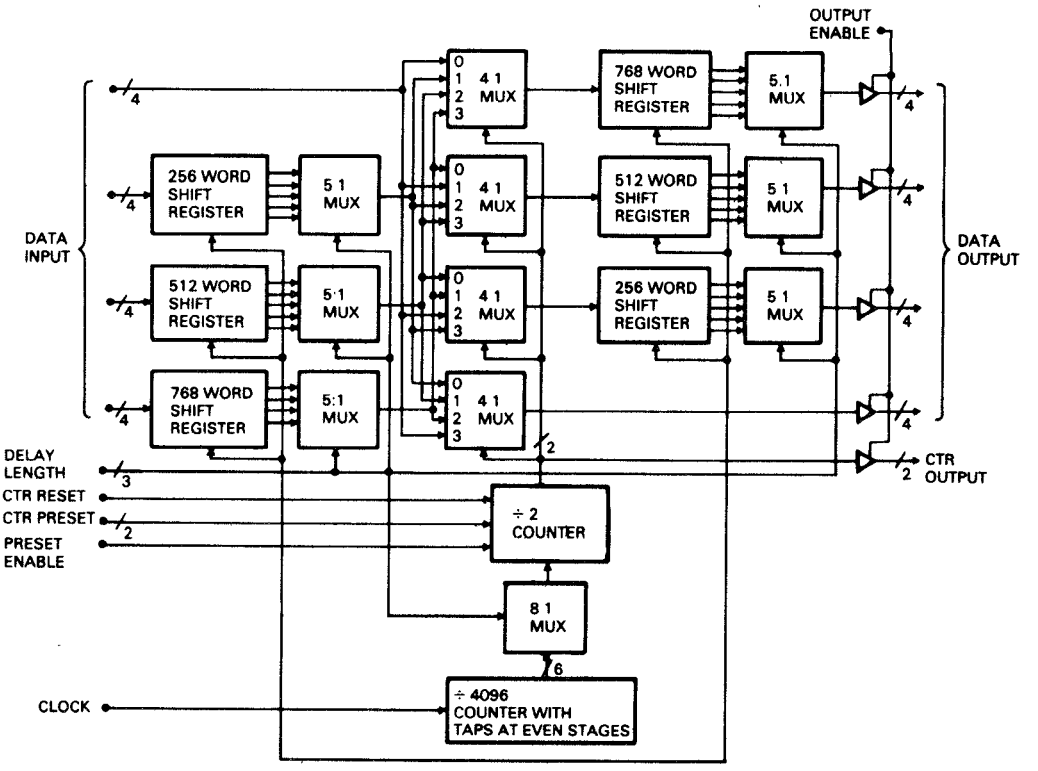


Fig. 3. Delay commutator block diagram.

four points. These data are operated upon by a radix 4 butterfly which does not change the data order. A second delay commutator (set for $X=1$) reorders the data to produce streams of adjacent data. This process is derived and explained in greater detail in [10].

IV. The Delay Commutator Circuit

Careful examination of our initial (off the shelf technology) FFT module design revealed that much of the complexity was due to the delay commutator element. Initial complexity estimates are 80 commercial integrated circuits for the computational element and 180 circuits for the delay commutator. The disparity in complexity arises because of the difficulty of realizing shift registers that can be set to a variety of lengths as required for the various delays. The most efficient approach involved simulating a delay line by using a RAM with write and read addresses displaced by a constant (i.e., the length of the simulated delay line). In view of the high complexity of the delay commutator, development of a semicustom implementation was undertaken. The resulting design of the delay commutator is a 4 bit wide slice that uses programmable length shift registers and a 4×4 switch as shown in Fig. 3. Data enter through shift registers with taps and multiplexers to set the delay at 1, 4, 16, 64, or 256 ($=X$) in the uppermost input register and multiplies of $2X$ and $3X$ in the middle and lower registers, respectively. Four 4:1 multiplexers implement the commutator function under the control of the programmable rate counter. The final 2 bit counter/decoder that controls the multiplexer settings can be reset and held in state 0 to disable the commutator switch function. In this mode the chip provides fixed length registers with delays of 256, $2X$ 768, and 1280 which are used to expand the delay commutator for 16384 point transforms. Data from the 4:1 multiplexers are output through programmable length shift registers that are similar to the input registers.

Gate array, standard cell, and custom technologies were considered for implementation of the delay commutator. An optimum balance between high circuit performance and low implementation cost was achieved using the AT&T Bell Laboratories' polycell (standard cell) CMOS technology. This technology was selected because it is well suited to the development of VLSI with high density shift registers and random logic. It is a twin tub 2.5 micrometer CMOS technology with chain-stops for device isolation and an epitaxial layer for latch-up protection [11].

The delay commutator circuit contains 12288 shift register stages and about 2000 gates of random logic, for a total complexity of 108000 transistors. At a clock rate of 10 MHz, the power dissipation is under 1/2 W. The chip size is 340×376 mil. The very high speed integrated circuit (VHSIC) program uses functional throughput rate (FTR) as a measure of circuit performance [12]. FTR is defined as the product of the number of gates times the chip clock rate divided by the area. By this measure (assuming a conversion rate of 3 transistors to 1 gate), the delay commutator FTR is 4.3×10^{11} gate·Hz/cm^2. Although not satisfying the important VHSIC environmental requirements, this is impressive performance for a semicustom circuit. The chip is shown on Fig. 4. Each of the four bit slices is constructed with input registers in a column,

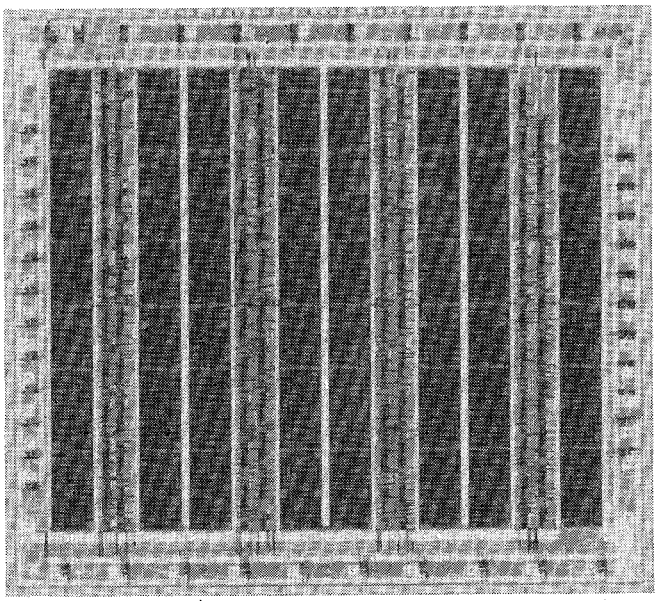


Fig. 4. Delay commutator photograph.

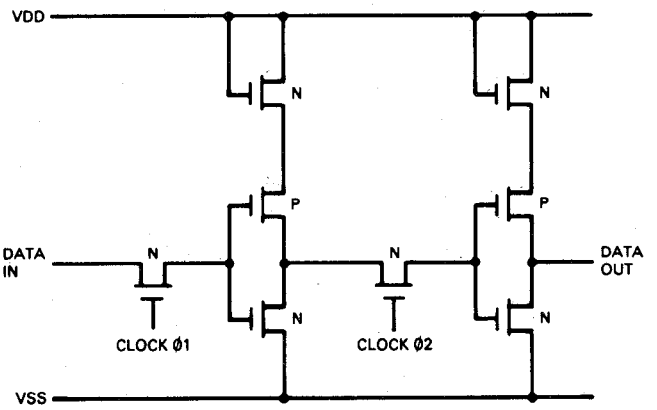


Fig. 5. Shift register cell.

switching logic in two standard cell "random logic" columns, and output registers in a column. The four nearly identical slices are about four times as tall as they are wide producing a roughly square chip when they are properly stacked. There is a minor variation in the random logic of each bit slice to account for sharing of the counters, decoders, click drivers, etc. The circuit is packaged in a 48 pin ceramic dual-in-line package to accomodate the 32 data, 11 control, and 5 power and ground connections.

The shift register cells consist of eight transistors each. Each cell measures 3.8 square mil and uses the same two-phase nonoverlapping master-slave clocks as the random logic. The circuit diagram of a single shift register cell is shown in Fig. 5. The upper transistor is connected in a diode configuration to reduce the effective "1" level within the cell. This eliminates the threshold voltage drop across the clocked transmission gate since the clock maintains a full swing from V_{SS} to V_{DD}. This approach eliminates static dc power consumption while using a simple two-phase nonoverlapping clock since there is no p-type transmission gate in parallel with the n-type transmission gate. A photomicrograph of the cell is shown in Fig. 6.

The 2000 gates of random logic were implemented using a library of standard cells. The geometrical descriptions of each cell serves as a common data base. Descriptions of each cell are automatically generated for use in the automated layout and simulation programs for fault, logic, and timing analysis.

The complete design cycle comprised two phases. In the initial phase the functional block diagram was translated into a logic design where the logic elements are drawn from the polycell library. In the second phase the logic design is translated into a silicon layout. While translating the functional design into the logic design, the designer uses a schematic capture routine which permits automatic polycell implementation of high level functions and automatically generates a machine readable circuit description which is used to detect design flaws and give a measure of the testability of the logic. While translating the logic design into the silicon layout, a layout routine is used to automatically produce an optimum layout with respect to size and parasitics. The parasitics are used during timing simulation to verify the performance of the circuit. Masks are then generated and wafers are processed, tested, and packaged.

The output waveform for one of the channels of the delay commutator is shown in Fig. 7. The 10 MHz clock waveform is shown on the upper channel. A 5 MHz output data waveform is shown on the lower channel. Note that the rise and fall times are in the 10–15 ns range.

A logic analyzer test pattern for one bit of the 4 bit delay commutator slice is shown in Fig. 8. Here input channels 1, 2, 3, and 4 are commutated to output channels 4, 3, 2, and 1, respectively. The varying delays between the input channels and the outputs are due to the shift registers (set here for $X=1$). Tracing through the delay commutator circuit in Fig. 1 indicates that the delays should be 6, 4, 2, and 0 for output channels 1, 2, 3, and 4, respectively.

The point of semicustom chip development is to reduce system complexity. This circuit succeeds admirably as shown in Table I. Development of the delay commutator chip reduces the complexity of a 40 MHz 4096 point FFT from 1375 commercial integrated circuits to 546 circuits (of which 66 are delay commutator chips) [13]. This is a 60 percent complexity reduction achieved through use of a single semicustom chip. For larger transforms, the complexity of a 16384 point FFT processor is reduced from 1634 integrated circuits to 670 circuits with the VLSI delay commutator circuit. Such a reduction greatly improves system reliability since connections between circuits represent the dominant failure mechanism in modern systems [14]. With 60 percent fewer circuits (and a corresponding reduction in the number of interconnections) the reliability is greatly improved.

V. Conclusions

A high performance semicustom circuit has been developed to implement the interstage reordering required for radix 4 pipeline FFT computation. The chip is realized

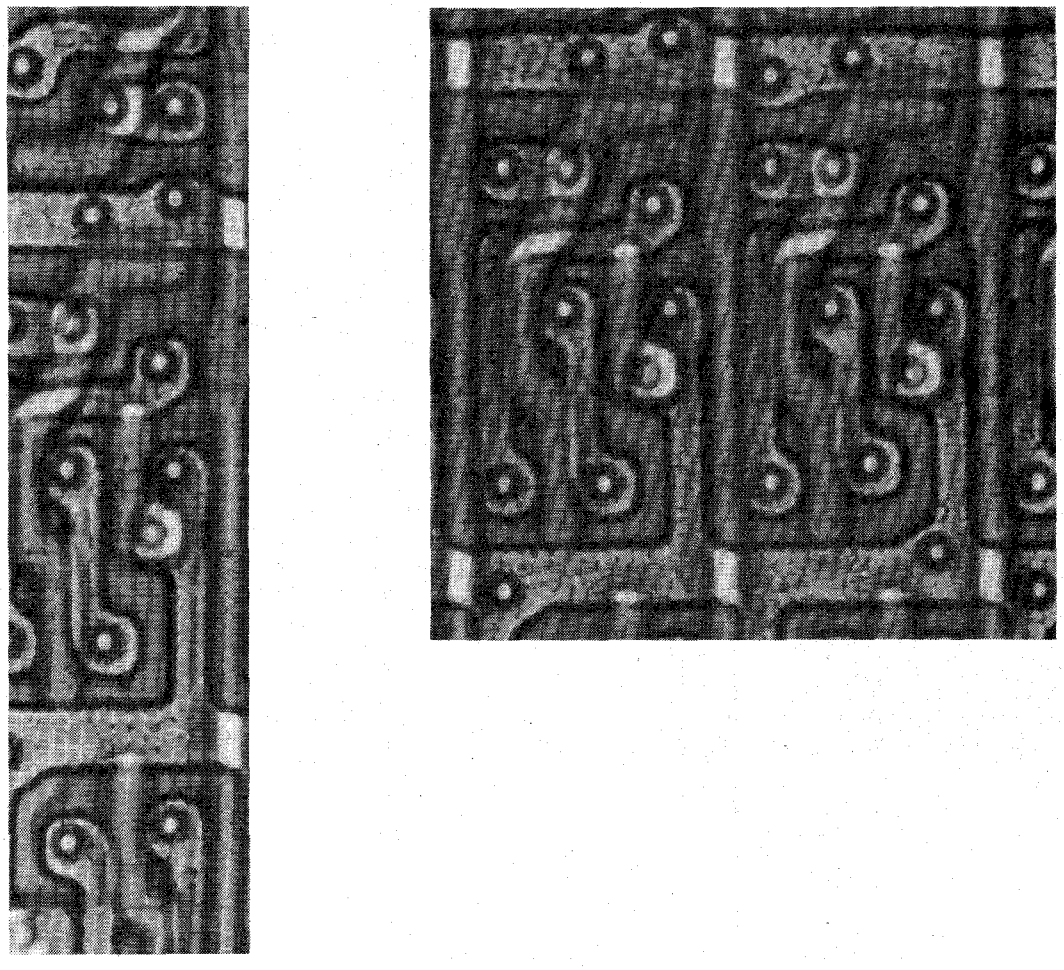


Fig. 6. Photograph of shift register cell.

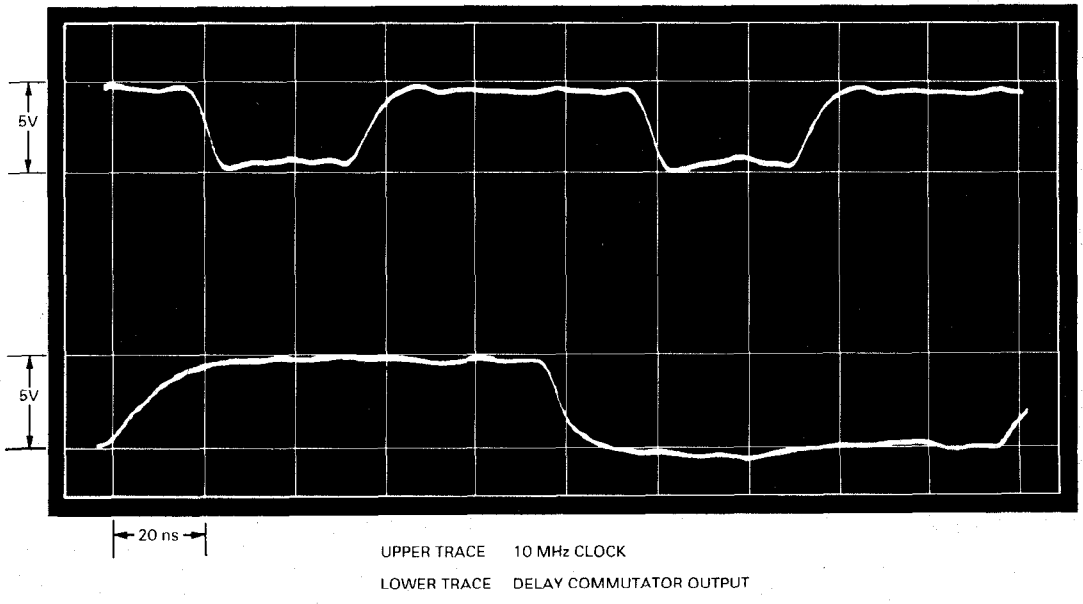


Fig. 7. Clock and data output waveforms (10 MHz clock rate).

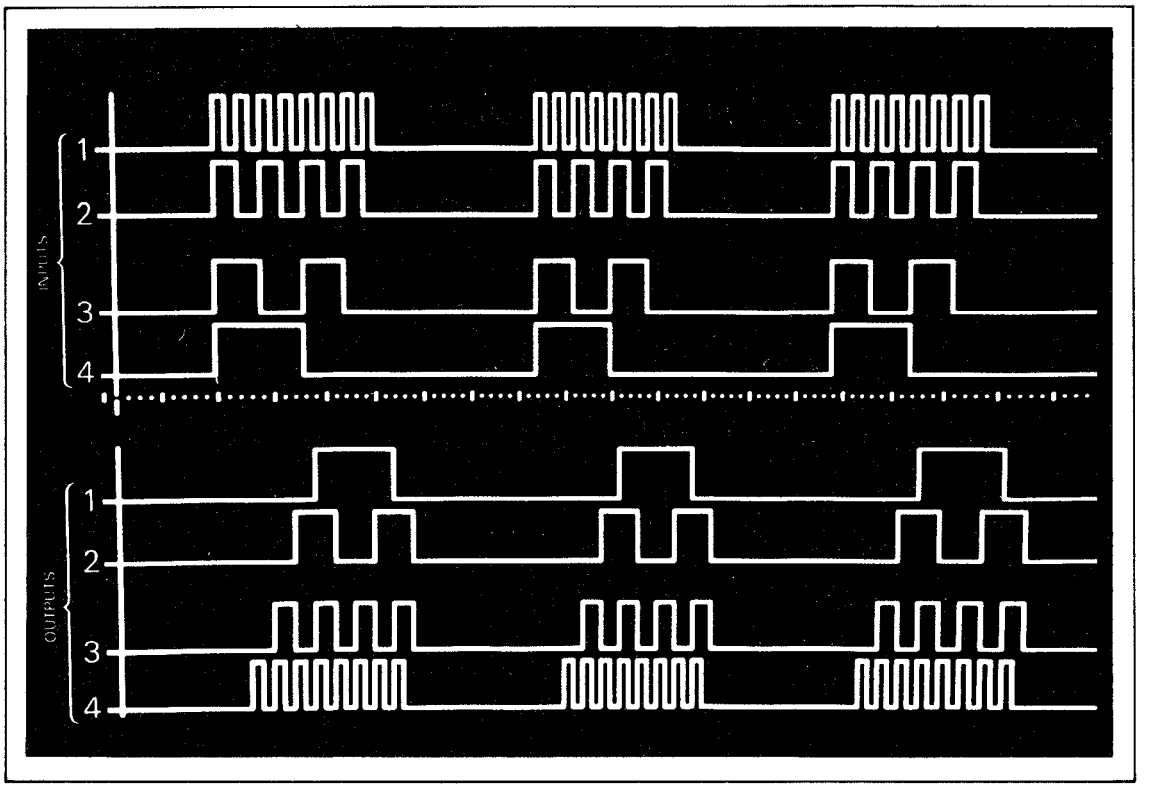


Fig. 8 Logic analyzer display of delay commutator operation showing mapping of input channels 1, 2, 3, and 4 onto output channels 4, 3, 2, and 1, respectively (with delay $X=1$).

TABLE I
COMPLEXITY REDUCTION ACHIEVED WITH THE DELAY COMMUTATOR CIRCUIT

	SYSTEM COMPLEXITY (INTEGRATED CIRCUIT COUNT)			
	WITHOUT DELAY COMMUTATOR CURCUIT		WITH DELAY COMMUTATOR CIRCUIT	
4096 POINT FFT				
COMPUTATIONAL ELEMENT	6 CARDS AT 80 CKTS/CARD	480 CKTS	6 CARDS AT 91 CKTS CARD	546 CKTS
DELAY COMMUTATOR	5 CARDS AT 179 CKTS CARD	895 CKTS	(INCLUDED ON CE)	
TOTAL	11 CARDS	1375 CKTS	6 CARDS	546 CKTS
16384 POINT FFT				
COMPUTATIONAL ELEMENT	7 CARDS AT 80 CKTS CARDS	560 CKTS	7 CARDS AT 91 CKTS CARD	637 CKTS
DELAY COMMUTATOR	6 CARDS AT 179 CKTS CARDS	1074 CKTS	1 EXTENDED DC (1024)	33 CKTS
TOTAL	13 CARDS	1634 CKTS	8 CARDS	670 CKTS

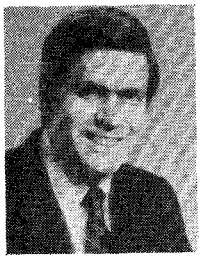
with twin tub 2.5 μm CMOS technology. It achieves an FTR of approximately 5×10^{11} gate·Hz/cm^2.

Use of this circuit reduces the complexity of high speed FFT processors by 60 percent relative to a commercial circuit implementation. Since it is organized as a 4 bit wide data slice, it is applicable to the realization of FFT processors for common data word sizes. It is used directly to implement processors for transform lengths of 16, 64, 256, 1024, and 4096 (complex) points, and can be expanded to accommodate 16384 point transforms.

REFERENCES

[1] J. W Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Math. Comput.*, vol. 19, pp. 297–301, 1965.
[2] E. E. Swartzlander, Jr., L. S. Lome, and G. Hallnor, "Digital signal processing with VLSI technology," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, 1983, pp. 951–954.
[3] S Winograd, "On computing the discrete Fourier transform," *Math. Comput.*, vol. 32, pp. 175–199, 1978.
[4] J. L. van Meerbergen and F J. van Wyk, "A 256-point discrete Fourier transform processor fabricated in a 2 μm NMOS technology," *IEEE J. Solid-State Circuits*, vol SC-18, pp. 604–609, 1983.
[5] G. D. Covert, "A 32 point monolithic FFT processor chip," in

Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing, 1982, pp. 1081–1083.
[6] H. J. Nussbaumer, *Fast Fourier Transform and Convolution Algorithms*. New York: Springer-Verlag, 1982.
[7] J. H. McClellan and R. J. Purdy, "Applications of digital signal processing to radar, " in *Applications of Digital Signal Processing*, A. V. Oppenheim, Ed. Englewood Cliffs, NJ: Prentice-Hall, 1978, Ch. 5.
[8] H. L. Groginsky and G. A. Works, "A pipeline fast Fourier transform," *IEEE Trans. Comput.*, vol. C-19, pp. 1015–1019, 1970.
[9] J. A. Eldon and C. Robertson, "A floating point format for signal processing," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, 1982, pp. 717–720.
[10] L. R. Rabiner and B. Gold, *Theory and Applications of Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1975, ch. 10.
[11] L. C. Parrillo *et al.*, "Twin-tub CMOS II—An advanced VLSI technology," in *Proc. Int. Electron Devices Meeting*, 1982, pp. 706–709.
[12] L. W. Sumney, "VHSIC: A status report," *IEEE Spectrum*, vol. 19, pp. 34–39, Dec. 1982.
[13] E. E. Swartzlander, Jr., and G. Hallnor, "Fast transform processor implementation," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, 1984, pp. 25A.5.1–25A.5.4.
[14] G. W. Preston, "The very large scale integrated circuit," *Amer. Scientist*, vol. 71, pp. 466–472, 1983.



Earl E. Swartzlander, Jr. (S'64–M'72–SM'79) received the B.S.E.E. degree from Purdue University, Lafayette, IN, in 1967, the M.S.E.E. degree from the University of Colorado, Boulder, in 1969, and the Ph.D. degree from the University of Southern California, Los Angeles, in 1972. He obtained his doctorate in computer design with the support of a Howard Hughes Doctoral Fellowship.

He is currently the Manager of the Advanced Development Office in the Systems Engineering Operations of the TRW Defense Systems Group. This involves the conceptual definition and development of advanced signal processing systems. His current activity focuses on the related issues of algorithms, architecture, and implementation with off-the-shelf as well as custom VLSI. He has directed the development of a variety of advanced systems, and has developed the architectural and functional design of VLSI components which are in varying stages of design, development, and production. He has published over 50 papers in the fields of computer architecture, VLSI implementation, and computer arithmetic, and is the Editor of the books *Computer Design Development* (Rochelle Park, NJ: Hayden Book Co., 1976) and *Computer Arithmetic* (Stroudsburg, PA: Dowden, Hutchinson & Ross, 1980).

Dr. Swartzlander is an Editor of the IEEE TRANSACTIONS ON COMPUTERS. He is a member of the Association for Computing Machinery and belongs to Eta Kappa Nu, Sigma Tau, and Omicron Delta Kappa honorary fraternities, and is a registered Professional Engineer in Alabama, California, and Colorado.

Wendell K. W. Young (M'83) received the B. S. degree in information computer sciences from the University of Hawaii, Honolulu, in 1978.

In 1979 he joined TRW, Redondo Beach, CA, as a Member of the Technical Staff conducting analysis work on the post-flight navigation data system for the Space Shuttle program. In 1980, he joined Magnavox Government Systems where he was engaged in the development of integrated navigation systems. He is currently a Systems Engineer in the TRW Defense Systems Group involved in the development of advanced signal processing circuits and systems.

Saul J. Joseph (S'77–M'80) received the B.S.E.E. degree from Rutgers University, New Brunswick, NJ, in 1978 and the M.S.E.E. degree from Lehigh University, Bethlehem, PA, in 1979.

In 1978 he joined AT&T Bell Laboratories. He is currently a Member of the Technical Staff in the VLSI Design Laboratory in Allentown, PA. He is responsible for the design of semicustom CMOS circuits using the Polycell family of standard cells.

Mr. Joseph is a member of Tau Beta Pi, Eta Kappa Nu, and the National Society of Professional Engineers.