EXHIBIT G

US005848105A

United States Patent [19]

Gardner et al.

[11] **Patent Number: 5,848,105**

[45] **Date of Patent: Dec. 8, 1998**

[54] **GMSK SIGNAL PROCESSORS FOR IMPROVED COMMUNICATIONS CAPACITY AND QUALITY**

[76] Inventors: **William A. Gardner**, 6950 Yount St.; **Stephan V. Schell**, 1990 Yountville Cross Rd., both of Yountville, Calif. 94599

[21] Appl. No.: **729,625**

[22] Filed: **Oct. 10, 1996**

[51] **Int. Cl.**[6] **H03D 3/00**

[52] **U.S. Cl.** **375/336**; 375/350; 375/346; 364/572; 364/574; 370/478

[58] **Field of Search** 375/854, 274, 375/278, 334, 336, 346, 350; 364/572, 574, 581; 370/464, 465, 478

[56] **References Cited**

U.S. PATENT DOCUMENTS

5,255,210	10/1993	Gardner et al.	364/574
5,260,968	11/1993	Gardner et al.	375/200
5,299,148	3/1994	Gardner et al.	364/574
5,465,413	11/1995	Enge et al.	375/349
5,487,091	1/1996	Jasper et al.	375/347
5,515,378	5/1996	Roy, III et al.	370/334

OTHER PUBLICATIONS

Mendoza et al., "Interference Rejection Using the Time–Dependent Constant Modulus Algorithm (CMA) and the Hybrid CMA/Spectral Correlation Discriminator", IEEE Transactions on Signal Processing, vol. 39, No. 9, pp. 2108–2111, Sep. 1991.

Mendoza et al., "Interference Rejection Using Time–Dependent Constant Modulous Algorithms", Twenty–Third Asilomar Conference on Signals, Systems & Computers, pp. 273–278, Oct. 1989.

Mendoza et al., "Interference Rejection Using a Hybrid of a Constant Modulus Algorithm and the Special Correlation Discriminator", Proceedings of MILCOM 89, pp. III:491–497, Oct. 1989.

Greene et al., "An Optimal Receiver Using a Time–Dependent Adaptive Filter", Proceedings of MILCOM 89, pp. III:650–656, Oct. 1989.

Reed et al., "A Frequency Domain Time–Dependent Adaptive Filter for Interference Rejection", Proceedings of MILCOM 88, pp. II:391–397, Oct. 1988.

Widrow et al., "Adaptive Signal Processing", Prentice–Hall, pp. 15–22, 1985.

Treichler et al., "A New Approach to Multipath Correction of Constant Modulus Signals", IEEE Transaction on Acoustics, Speech, and Signal Processing, vol. ASSP–31, No. 2, pp. 459–472, Apr. 1983.

Treichler et al., "New Processing Techniques Based on the Constant Modulus Algorithm", IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. ASSP–33, No. 2, pp. 420–431, Apr. 1985.

Gooch et al., "The CM Array: An Adaptive Beamformer for Constant Modulus Signals", Proceedings of ICASSP 86, pp. 2423–2526, Apr. 1986.

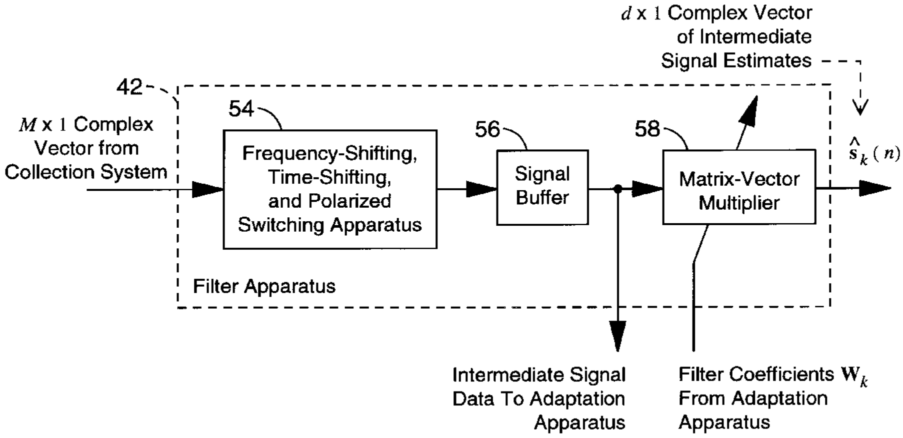
(List continued on next page.)

Primary Examiner—Jason Chan
Assistant Examiner—Congvan Tran
Attorney, Agent, or Firm—John P. O'Banion

[57] **ABSTRACT**

A method and apparatus for separating and removing distortion from interfering co-channel signals and suppressing adjacent-channel interfering signals of the Gaussian Minimum-Shift Keyed (GMSK) or other MSK type with filtering structures that exploit the cyclostationarity of the received GMSK or other MSK signals in order to accommodate a greater number (or the same number, but with greater quality) of transmitted signals received by one or more antennas than can be accommodated by existing filters. The parameters in these filtering structures are adapted by either of two adaptation apparatus that exploit both the known training sequence that is transmitted in most wireless communications systems, and the constant modulus property exhibited by each of the transmitted GMSK or other MSK signals.

12 Claims, 31 Drawing Sheets



OTHER PUBLICATIONS

Shynk et al., "Convergence Properties of the Multistage CMA Adaptive Beamformer", Proceedings of the Twenty–Seventh Asilomar Conference on Signals, Systems & Computers, pp. 622–626, Nov. 1993.

Schmidt, R., "Multiple Emitter Location and Signal Parameter Estimation", IEEE Transactions on Antennas Propagation, vol. AP–34, No. 3, pp. 276–280, Mar. 1986.

Viberg et al., "Detection and Estimation in Sensor Arrays Using Weighted Subspace Fitting", IEEE Signal Processing, vol. 39, No. 11, pp. 2436–2449, Nov. 1991.

Gardner, W.A., "Statistical Spectral Analysis: A Nonprobabilistic Theory", Prentice–Hall Information and System Sciences, pp. 481–488, 1987.

Brown, W.A., "On the Theory of Cyclostationary Signals", PhD thesis, University of California, Davis, pp. 70–102, Sep. 1987.

Reed, J.H., "Time–Dependent Adaptive Filters for Interference Rejection", PhD thesis, University of California, Davis, pp. 100–133, Dec. 1987.

Gardner et al., "Frequency–Shift Filtering Theory for Adaptive Co–Channel Interference Removal", Proceedings of the Twenty–Third Asilomar Conference on Signals, Systems & Computers, pp. 562–567, Nov. 1989.

Gardner et al., Performance of Optimum and Adaptive Frequency–Shift Filters for Cochannel Interference and Fading, Proceedings of the Twenty–Fourth Asilomar Conference on Signals, Systems & Computers, pp. 242–247, Nov. 1990.

Reed et al., "The Performance of Time–Dependent Adaptive Filters for Interference Rejection", IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 38, pp. 1373–1385, Aug. 1990.

Petrus et al., "Time Dependent Adaptive Arrays", IEEE Signal Processing Letters, vol. 2, No. 12, pp. 219–222, Dec. 1995.

He et al., "Spectral Correlation of AMPS Signals andits Application to Interference Rejection", Proceedings of MILCOM 1994, pp. III:1007–1011, Oct. 1994.

Reed et al., "An Optimal Receiver Using a Time–Dependent adaptive Filter", IEEE Transactions on Communications, vol. 42, No. 2/3/4, pp. 187–190, Feb.–Apr. 1995.

Aue et al., "An Interference Robust CDMA Demodulator That Uses Spectral Correlation Properties", Proceedings of IEEE 44th Vehicular Technology Conference, pp. 563–567, Jun. 1994.

Reed et al., "The Theoretical Performance of Optimal Time–Dependent Filters for Co–Channel Interference Rejection", Proceedings of MILCOM 90, pp. III:961–965, Sep. 1990.

Gardner, William A., "Cyclic Weiner Filtering: Theory and Method", IEEE Transactions on Communications, vol. 41. No. 1, Jan. 1993, pp. 151–163.

Hendessi F. et al., "The Structure and Performance of Fresh–Decision Feedback Equalizer in the Presence of Adjacent Channel Interference", Proceedings of the 1993 43rd IEEE Vehicular Technology Conference, May 18–20, 1993, pp. 641–644.

Webster, Mark, "Polyspectra–Based, Blind, MMSE, Fractionally–Spaced Equalization of a Cyclostationary Signal", Proceedings of the 1993 IEEE Statistical Signal and Array Processing Conference, Apr. 27–30, 1993, pp. IV–276 to IV–279.

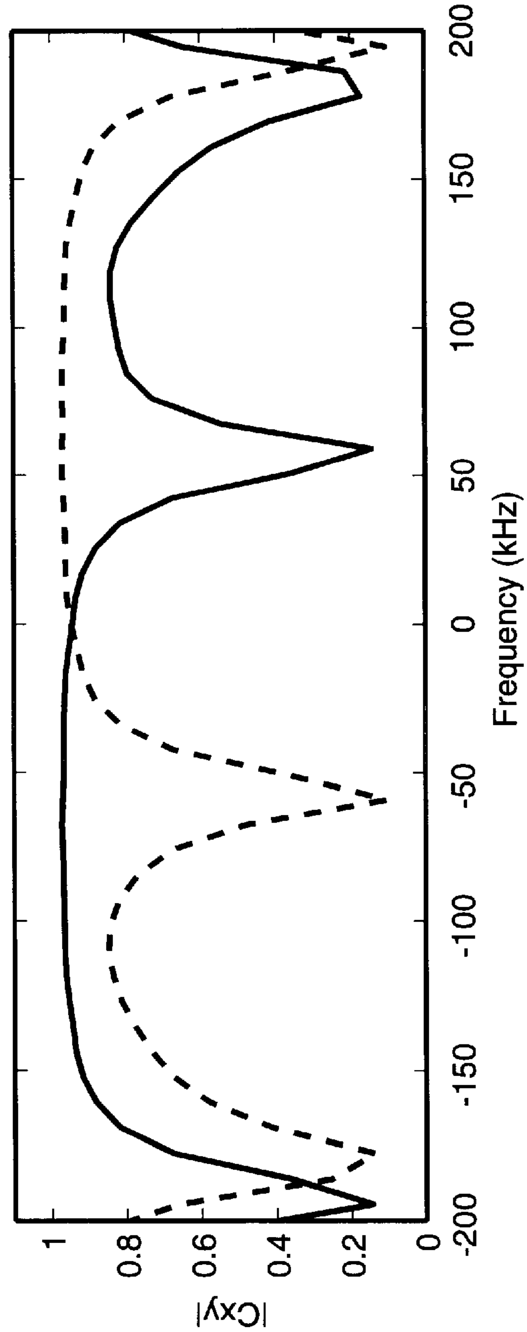

|Cxy|
0
0.2
0.4
0.6
0.8
1
-200
-150
-100
-50
0
50
100
150
200
Frequency (kHz)

FIG. – 1

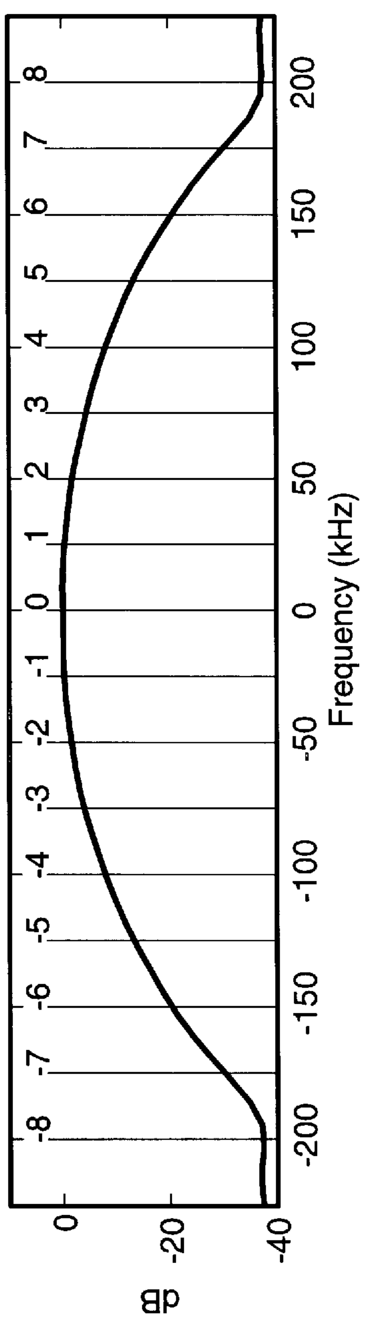

dB
0
-20
-40
-200
-150
-100
-50
0
50
100
150
200
Frequency (kHz)
-8
-7
-6
-5
-4
-3
-2
-1
0
1
2
3
4
5
6
7
8

FIG. – 2A

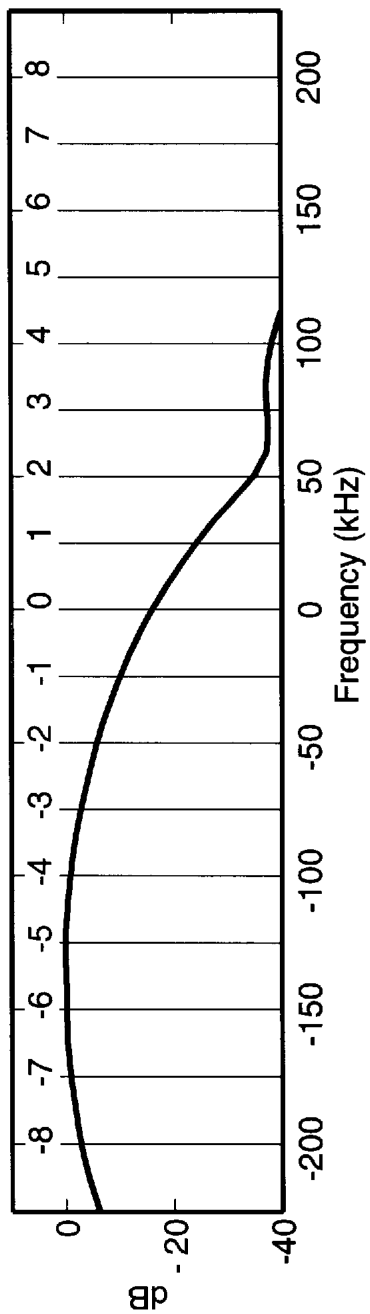

dB
0
-20
-40
-200
-150
-100
-50
0
50
100
150
200
Frequency (kHz)
-8
-7
-6
-5
-4
-3
-2
-1
0
1
2
3
4
5
6
7
8

FIG. – 2B

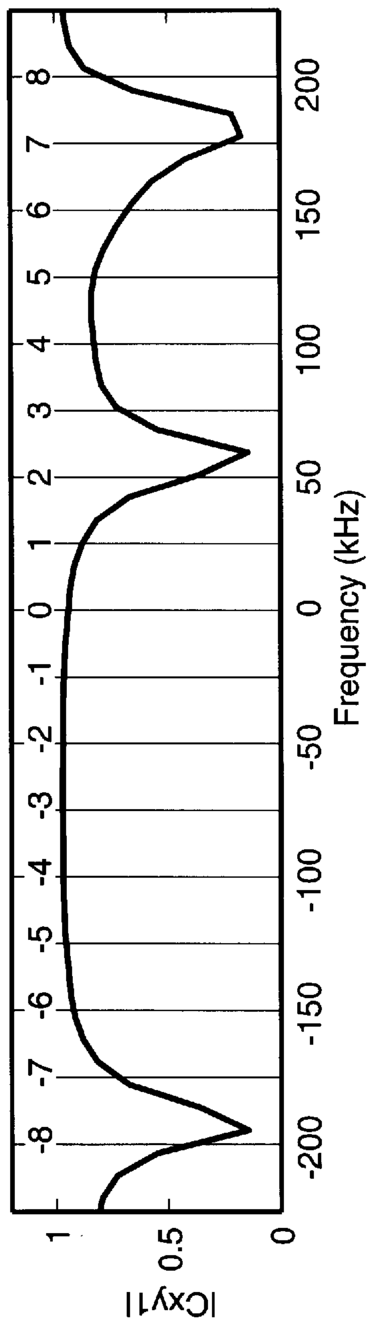
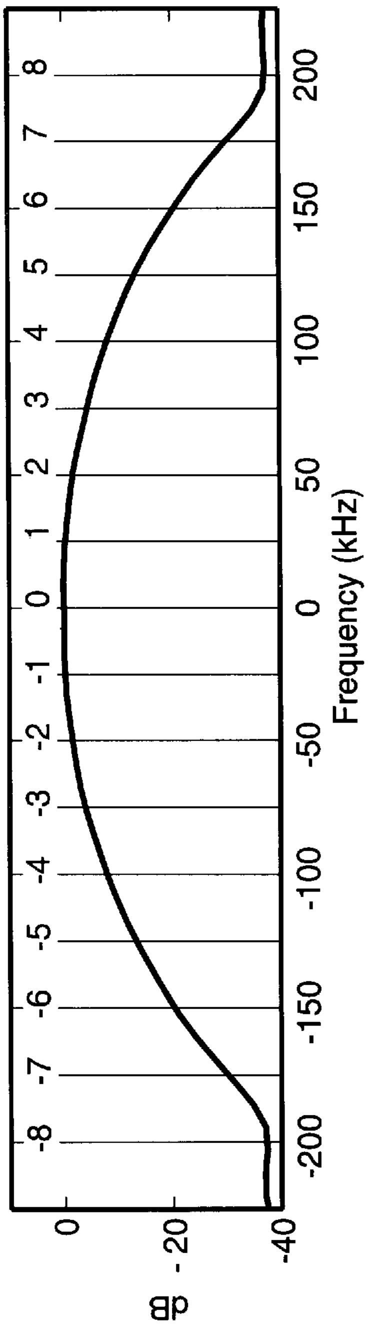
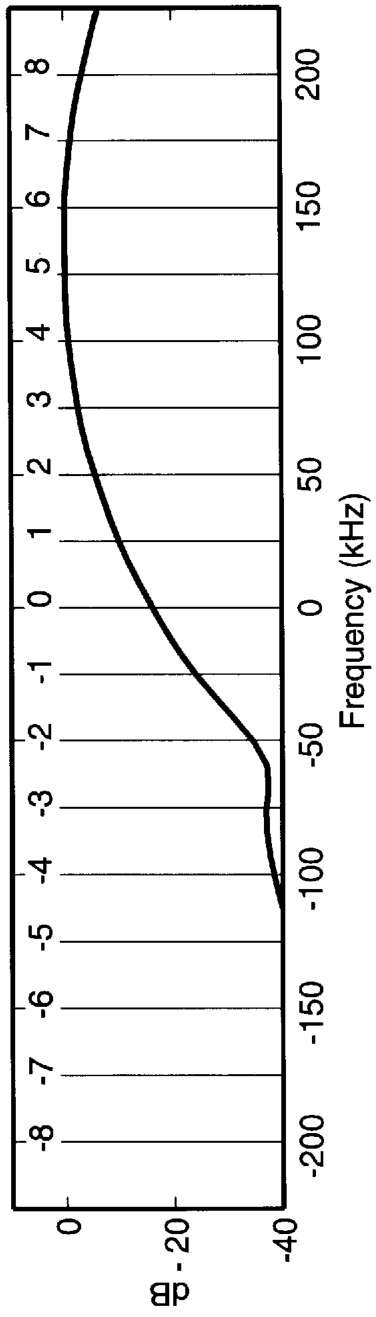

|Cxy1|
1
0.5
0
-200
-150
-100
-50
0
50
100
150
200
Frequency (kHz)
-8
-7
-6
-5
-4
-3
-2
-1
0
1
2
3
4
5
6
7
8

FIG. – 2C



dB
0
- 20
-40
-8
-7
-6
-5
-4
-3
-2
-1
0
1
2
3
4
5
6
7
8
-200
-150
-100
-50
0
50
100
150
200
Frequency (kHz)

FIG. – 3A



dB
0
- 20
-40
-8
-7
-6
-5
-4
-3
-2
-1
0
1
2
3
4
5
6
7
8
-200
-150
-100
-50
0
50
100
150
200
Frequency (kHz)

FIG. – 3B

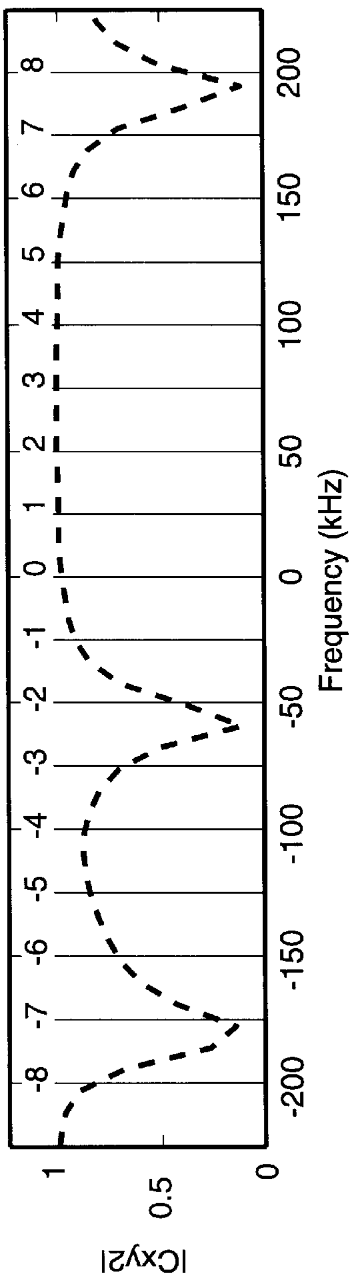

|Cxy2|
1
0.5
0
-8 -7 -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7 8
-200 -150 -100 -50 0 50 100 150 200
Frequency (kHz)

FIG. – 3C

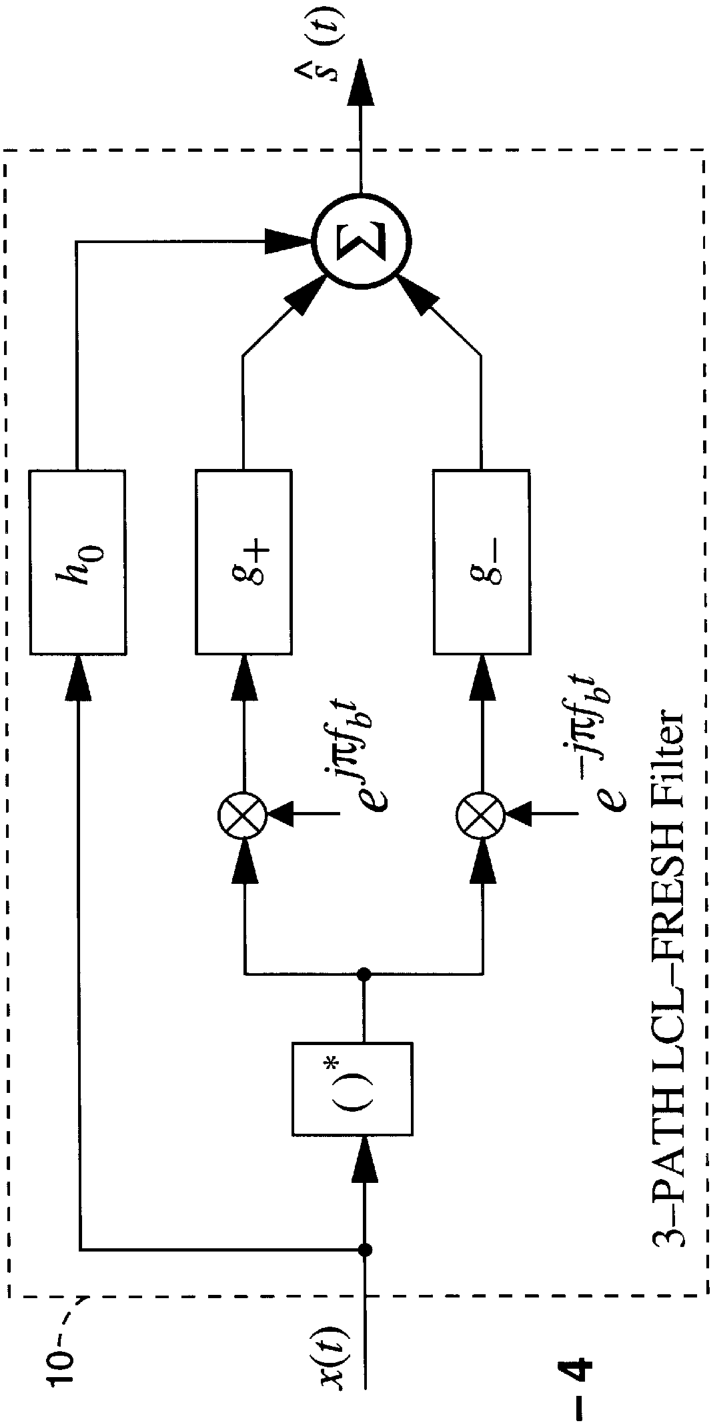

10
x(t)
( )*
h_0
g_+
g_-
$e^{j\pi f_b t}$
$e^{-j\pi f_b t}$
Σ
$\hat{s}(t)$
3–PATH LCL–FRESH Filter

FIG. – 4

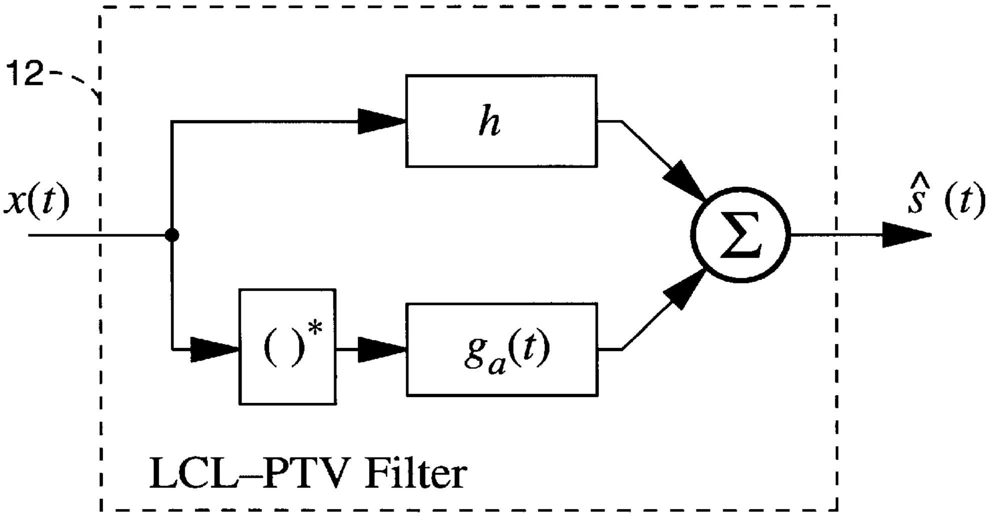

12
x(t)
h
( )*
$g_a(t)$
Σ
$\hat{s}(t)$
LCL–PTV Filter

FIG. – 5

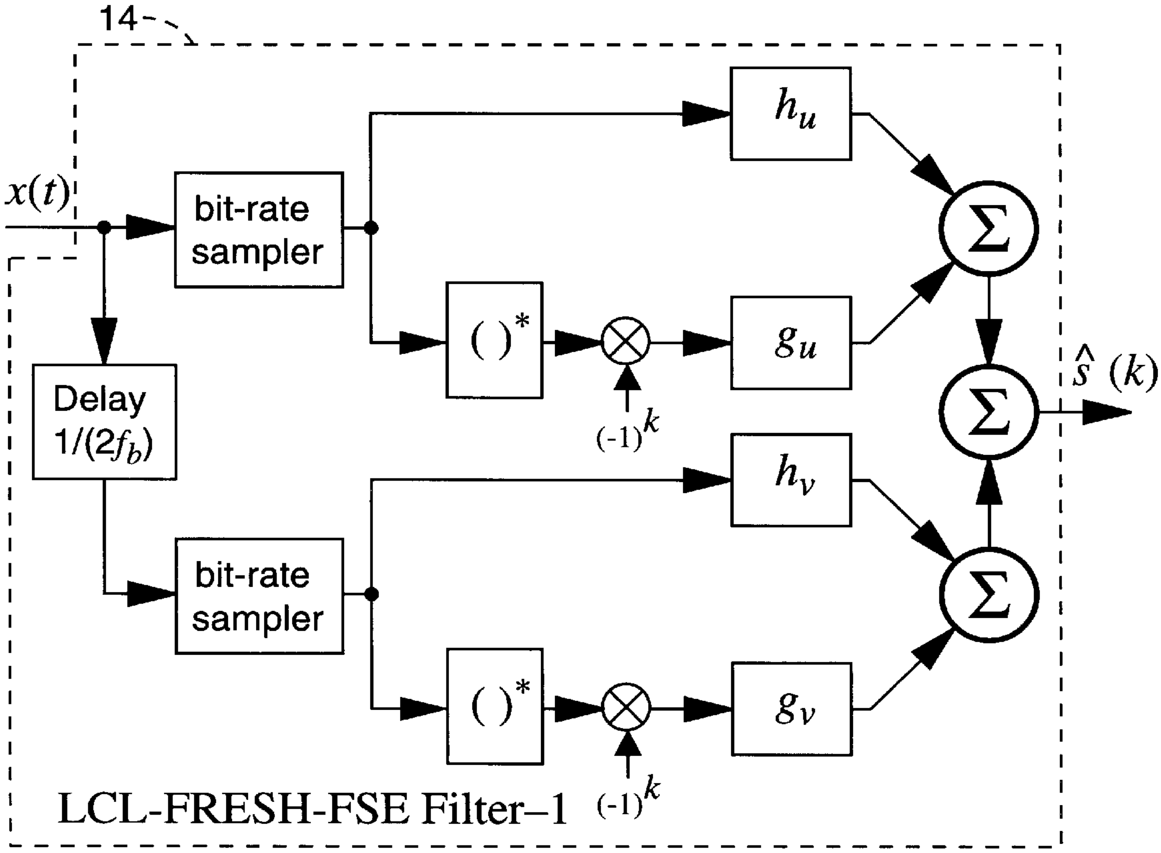

14
x(t)
bit-rate sampler
Delay $1/(2f_b)$
bit-rate sampler
h_u
( )*
$(-1)^k$
g_u
h_v
( )*
$(-1)^k$
g_v
Σ
Σ
Σ
$\hat{s}(k)$
LCL-FRESH-FSE Filter–1

FIG. – 6

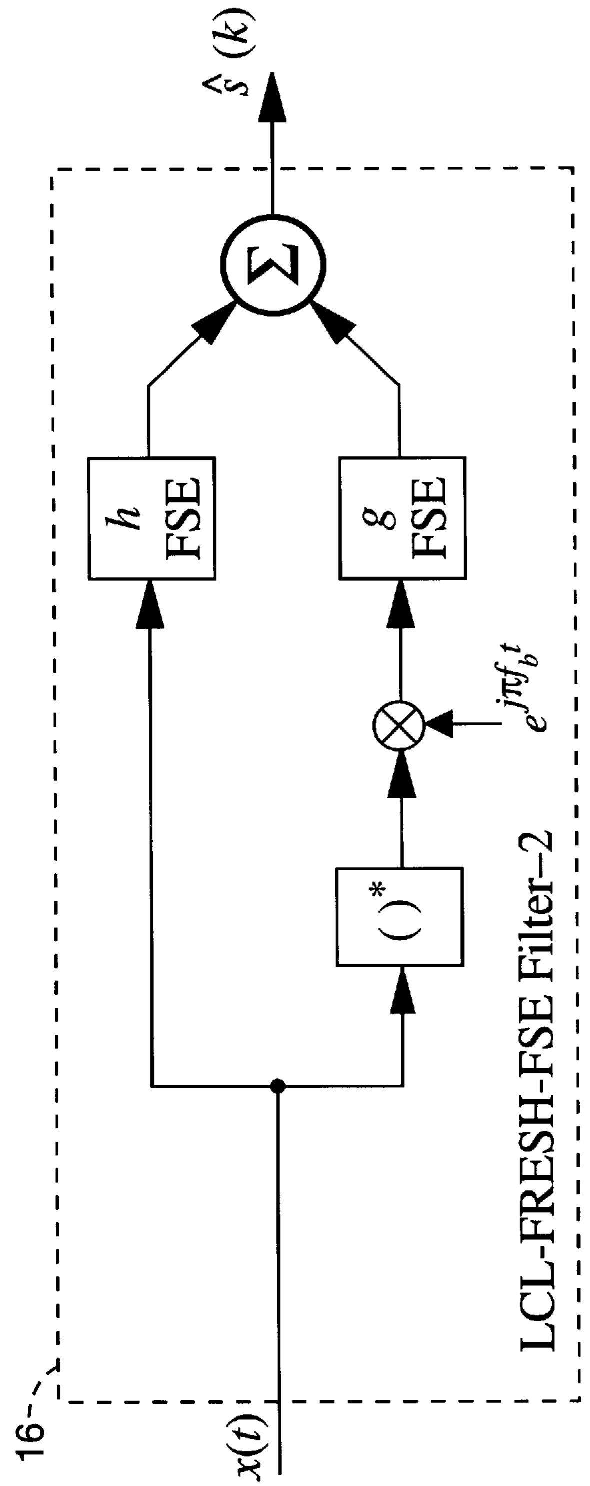
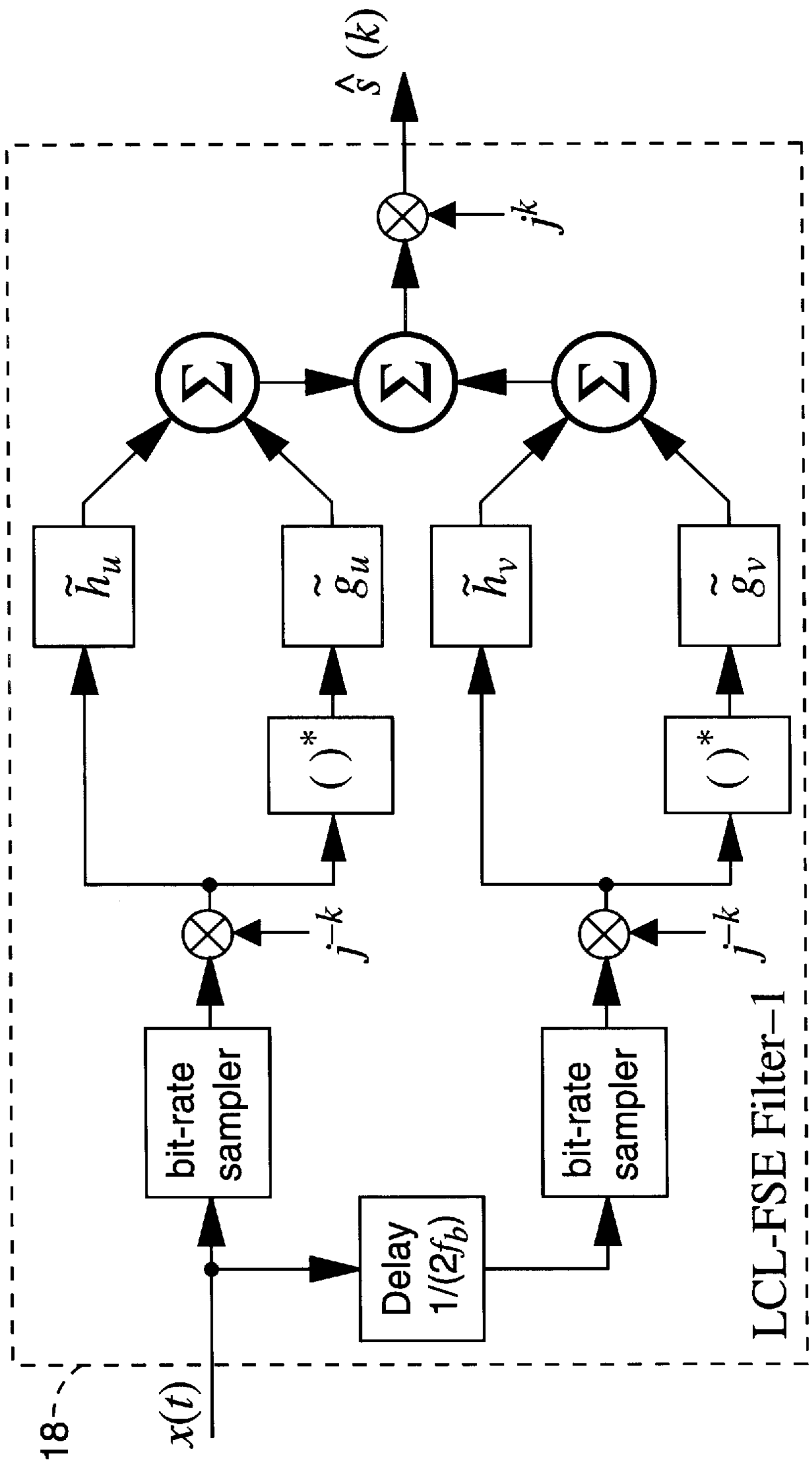


FIG. – 7



18
$x(t)$
Delay $1/(2f_b)$
bit-rate sampler
bit-rate sampler
j^{-k}
j^{-k}
$(\ )^*$
$(\ )^*$
$\tilde{h}_u$
$\tilde{g}_u$
$\tilde{h}_v$
$\tilde{g}_v$
Σ
Σ
Σ
j^k
$\hat{s}\,(k)$
LCL-FSE Filter-1

FIG. – 8

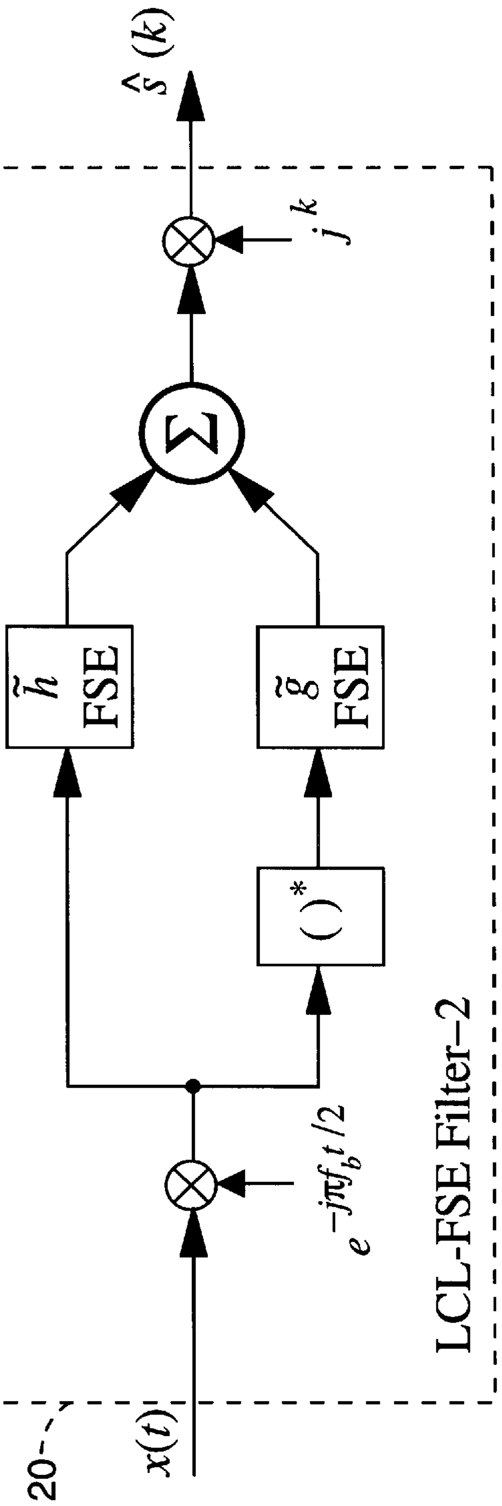

$\hat{s}(k)$
j^k
Σ
$\tilde{h}$ FSE
$\tilde{g}$ FSE
$(\ )^*$
$e^{-j\pi f_b t/2}$
LCL-FSE Filter–2
$x(t)$
20

FIG. – 9

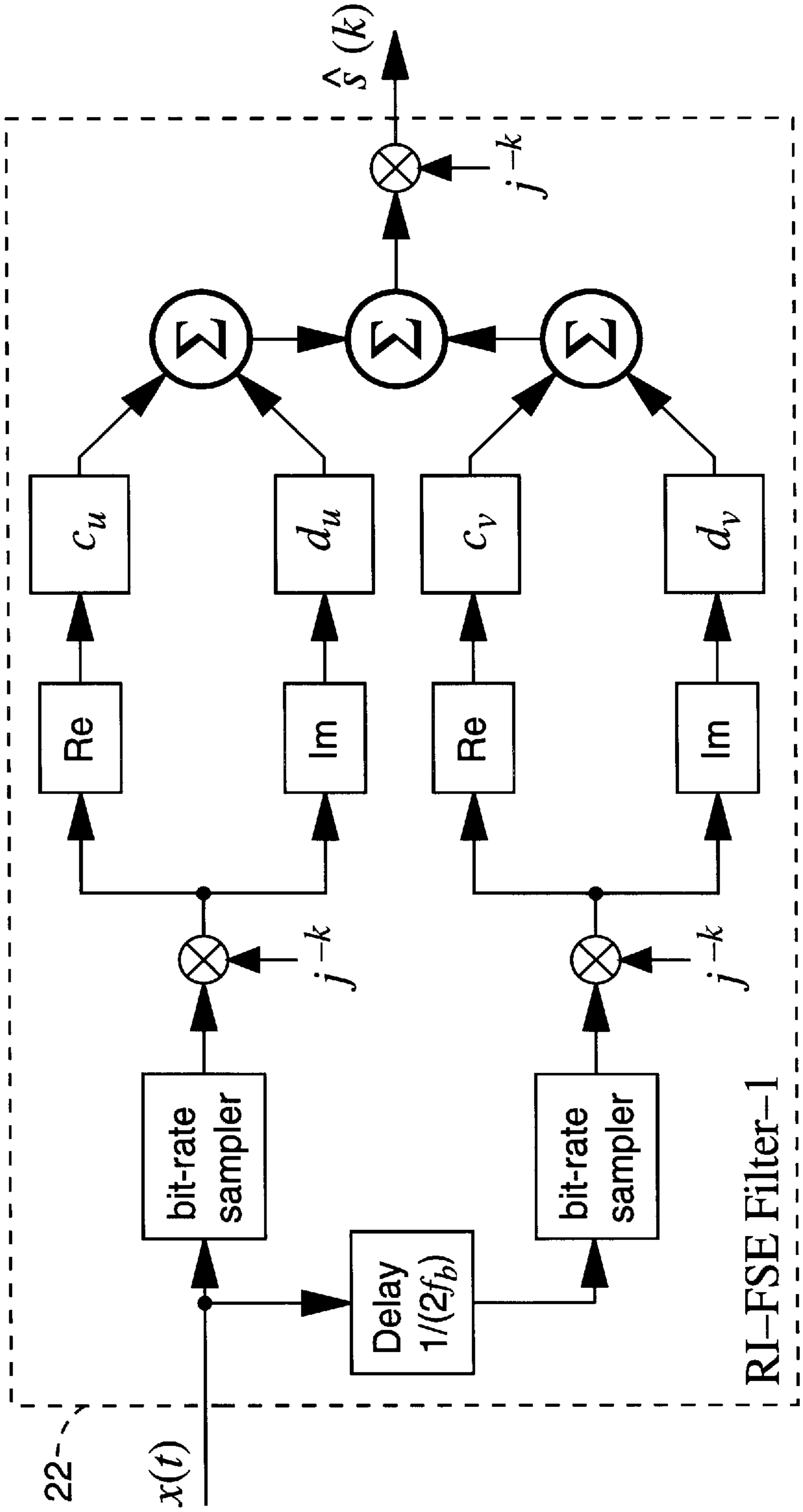

$\hat{s}(k)$
j^{-k}
Σ
c_u
d_u
c_v
d_v
Re
Im
Re
Im
j^{-k}
j^{-k}
bit-rate sampler
bit-rate sampler
Delay $1/(2f_b)$
RI–FSE Filter–1
22
$x(t)$

FIG. – 10

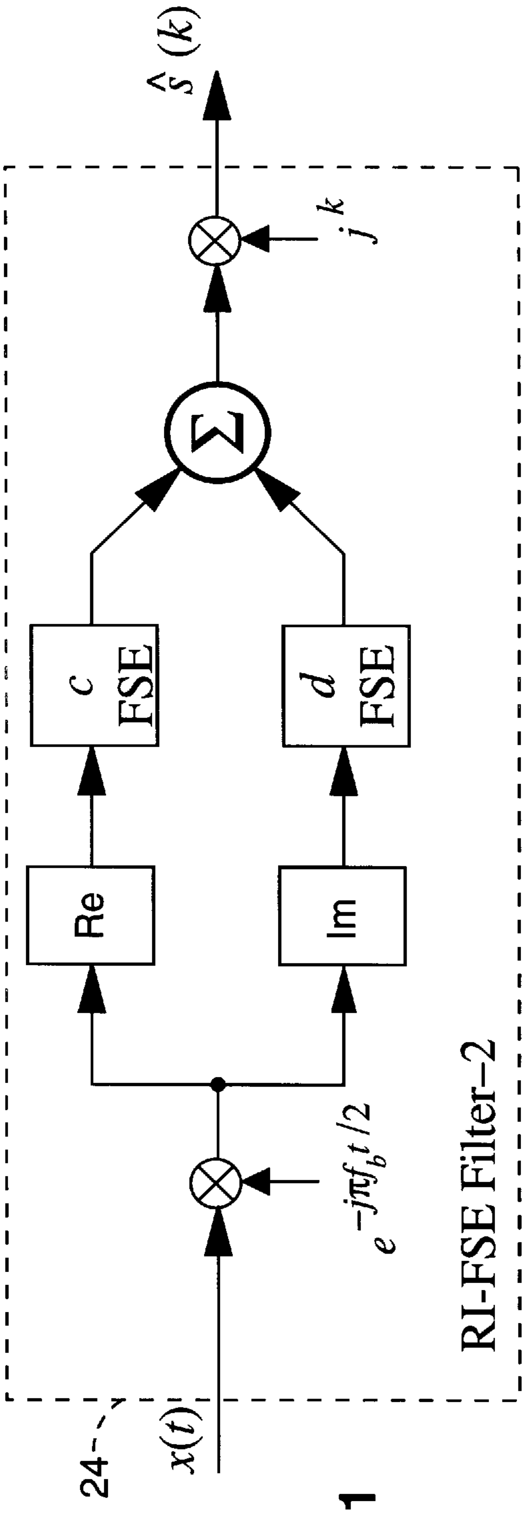

$\hat{s}(k)$
j^k
Σ
c FSE
d FSE
Re
Im
$e^{-j\pi f_b t/2}$
RI-FSE Filter–2
$x(t)$
24

FIG. – 11

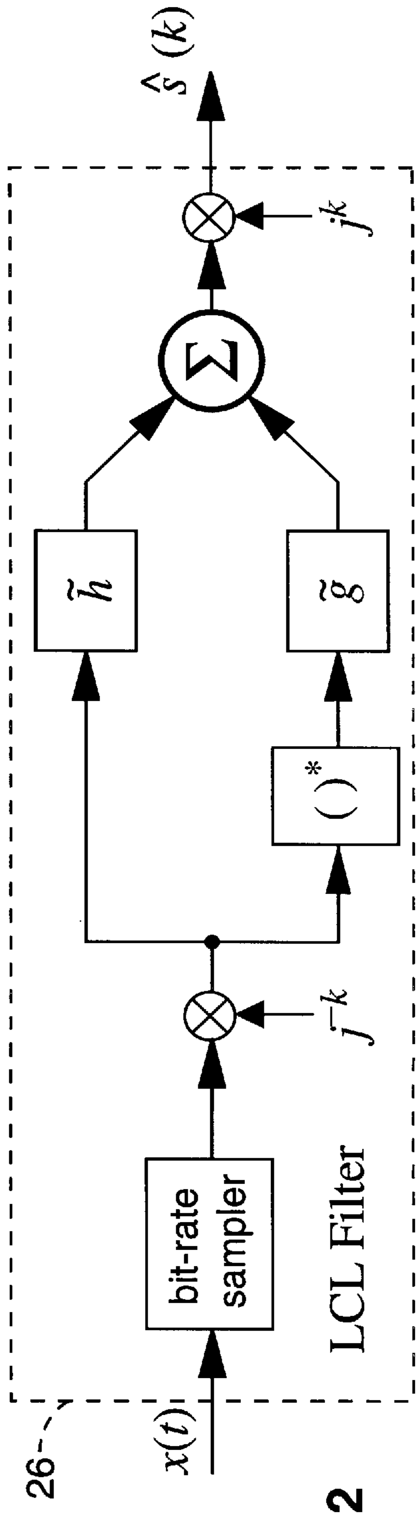

$\hat{s}(k)$
j^k
Σ
$\tilde{h}$
$\tilde{g}$
$(\ )^*$
j^{-k}
bit-rate sampler
LCL Filter
$x(t)$
26

FIG. – 12

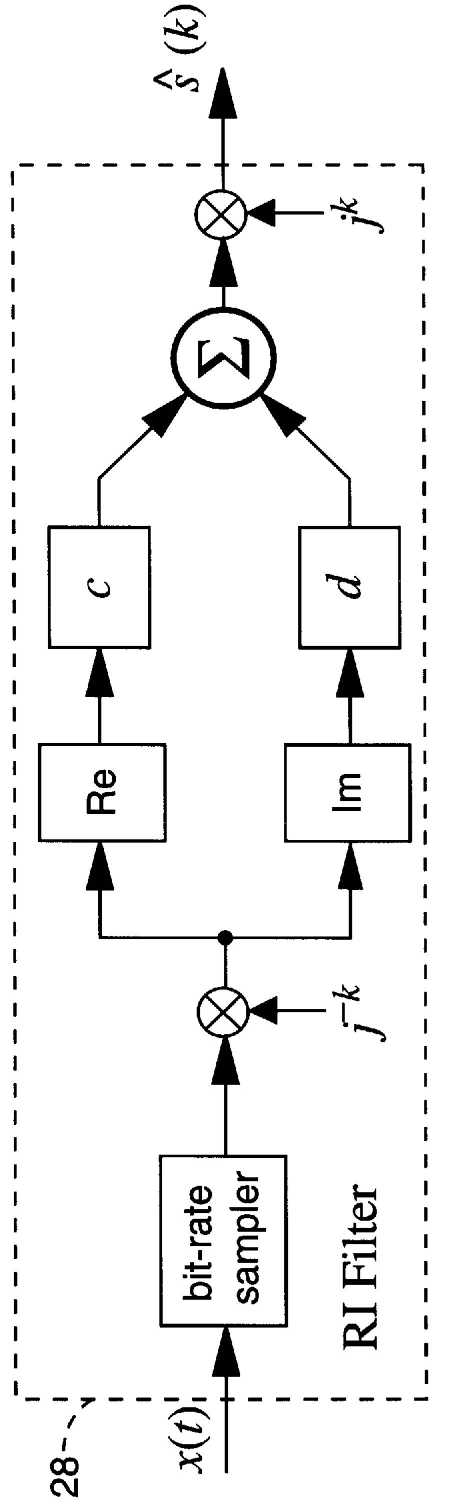
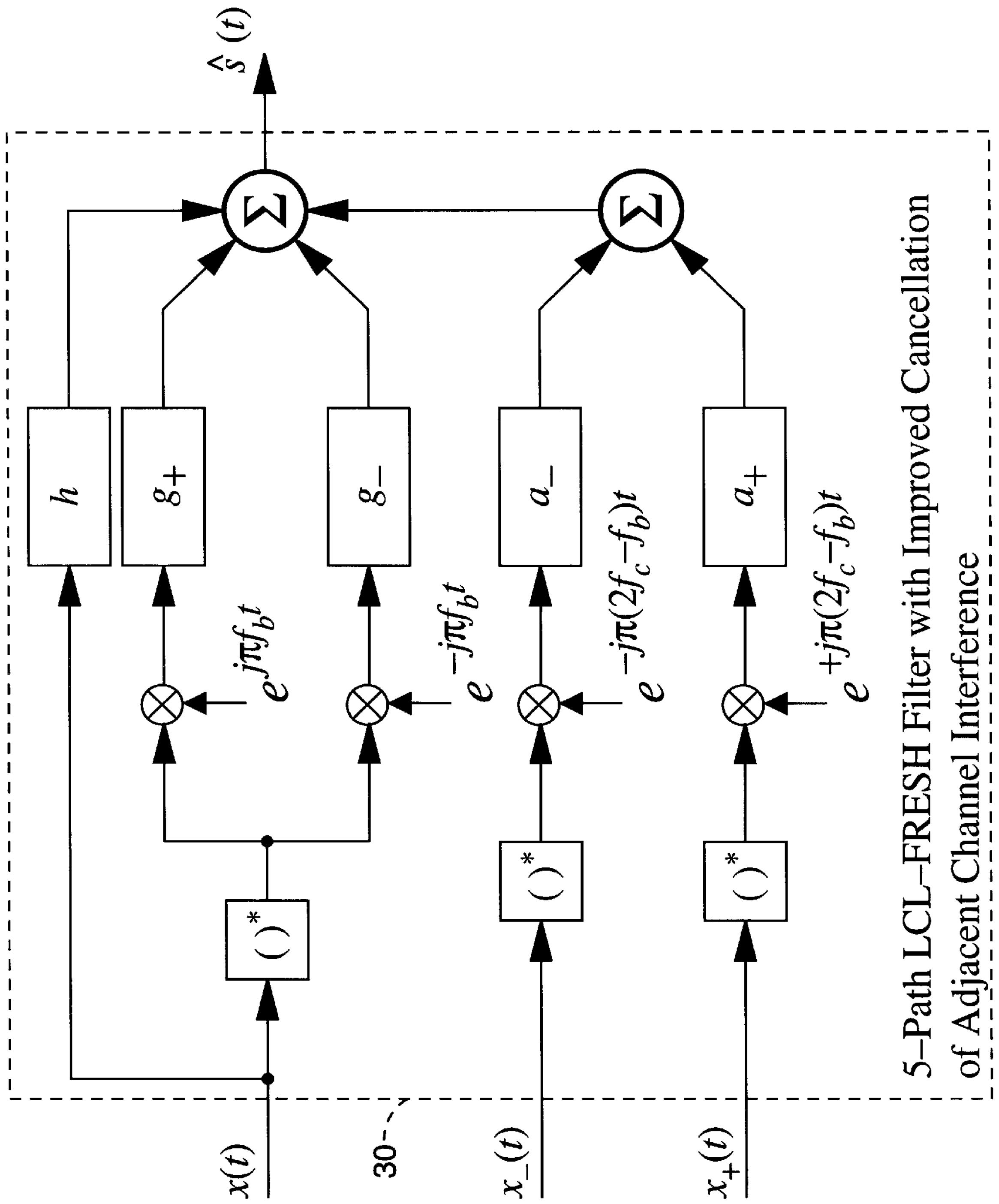
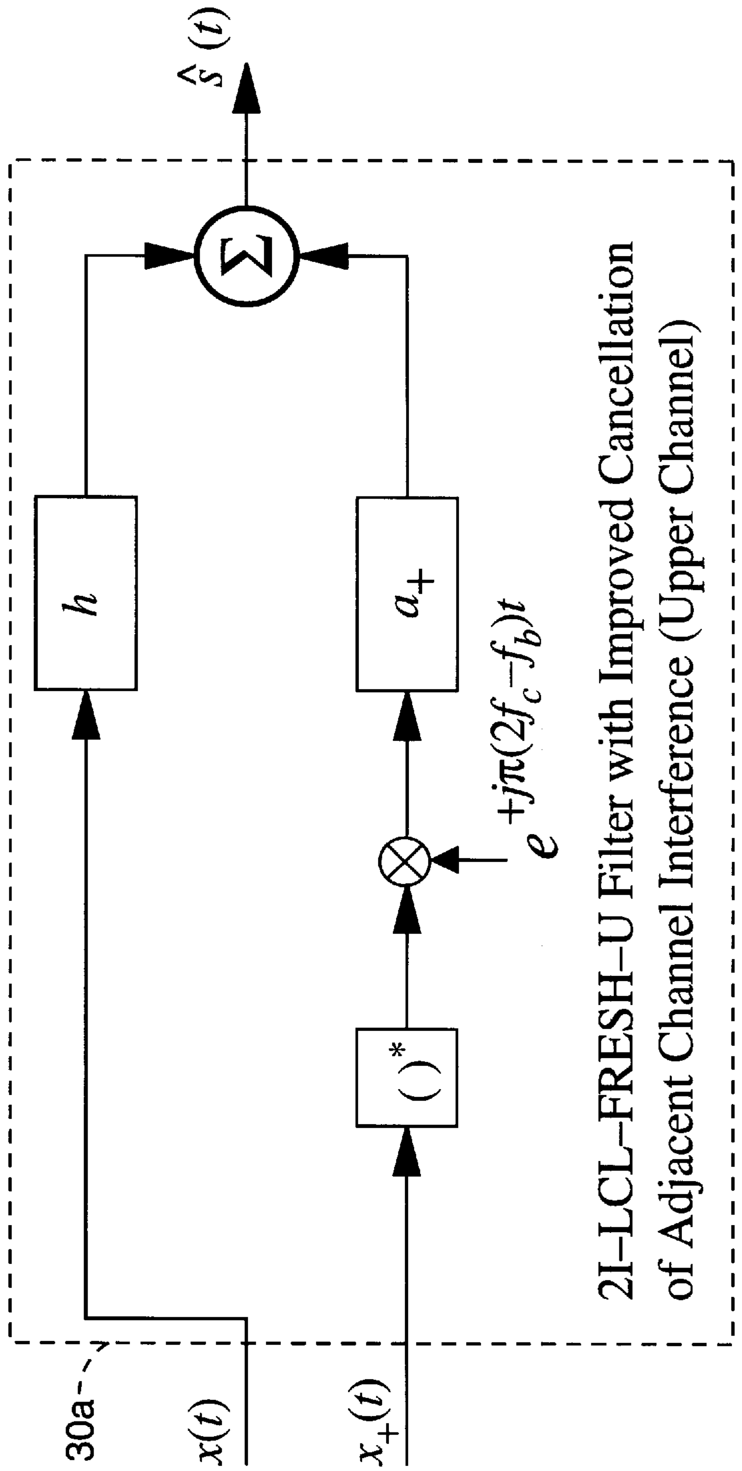


FIG. – 13



$\hat{s}(t)$
Σ
Σ
h
g_+
g_-
a_-
a_+
$e^{j\pi f_b t}$
$e^{-j\pi f_b t}$
$e^{-j\pi(2f_c - f_b)t}$
$e^{+j\pi(2f_c - f_b)t}$
$(\ )^*$
$(\ )^*$
$(\ )^*$
$x(t)$
30
$x_-(t)$
$x_+(t)$
5–Path LCL–FRESH Filter with Improved Cancellation of Adjacent Channel Interference

FIG. – 14



2I–LCL–FRESH–U Filter with Improved Cancellation of Adjacent Channel Interference (Upper Channel)

FIG. – 15

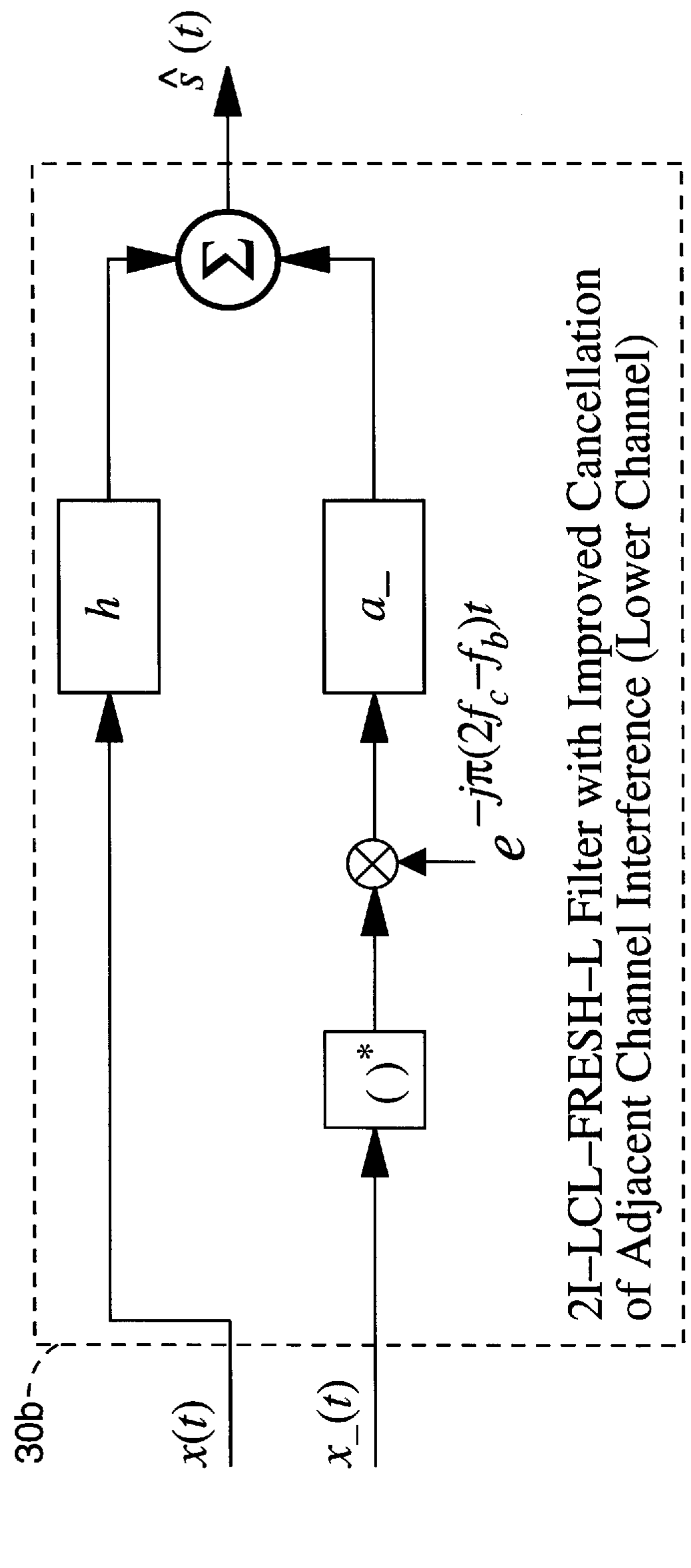
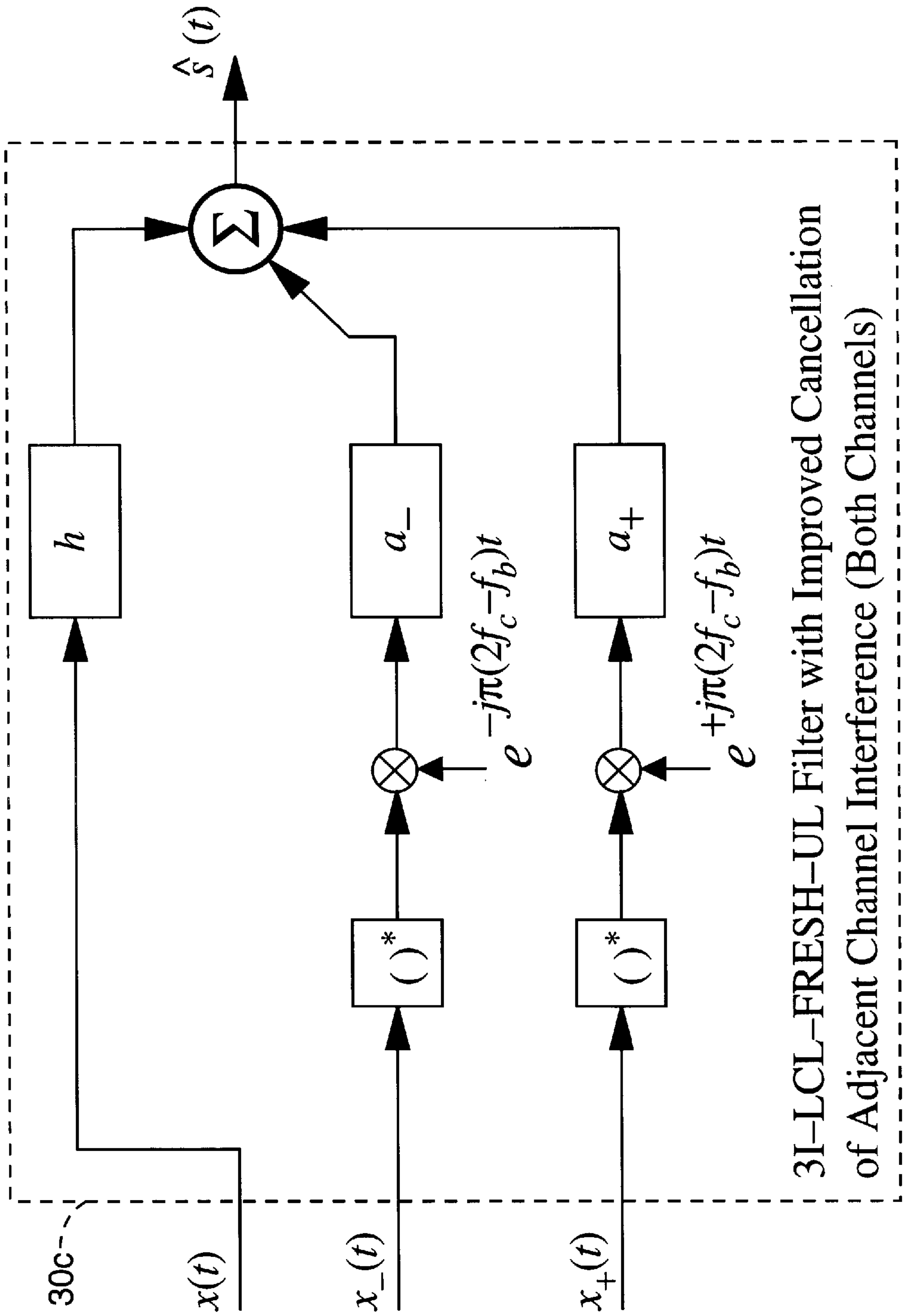

30b
x(t)
x_(t)
h
( )*
a_
$e^{-j\pi(2f_c-f_b)t}$
Σ
$\hat{s}(t)$
2I–LCL–FRESH–L Filter with Improved Cancellation
of Adjacent Channel Interference (Lower Channel)

FIG. – 16



30c
$x(t)$
$x_-(t)$
$x_+(t)$
$( )^*$
$( )^*$
$e^{-j\pi(2f_c-f_b)t}$
$e^{+j\pi(2f_c-f_b)t}$
h
a_-
a_+
Σ
$\hat{s}(t)$
3I–LCL–FRESH–UL Filter with Improved Cancellation of Adjacent Channel Interference (Both Channels)

FIG. – 17

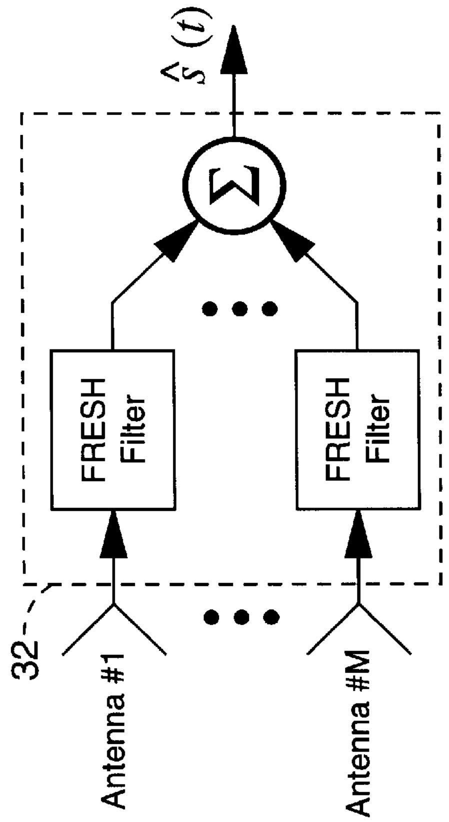


FIG. – 18

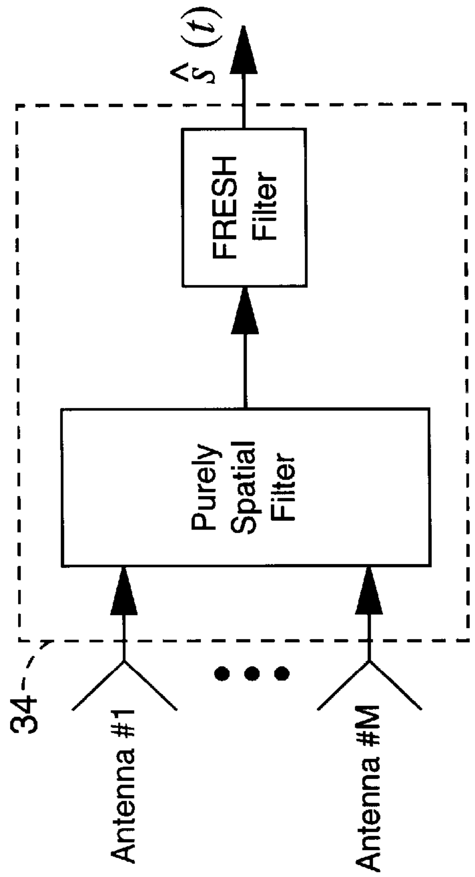


FIG. – 19

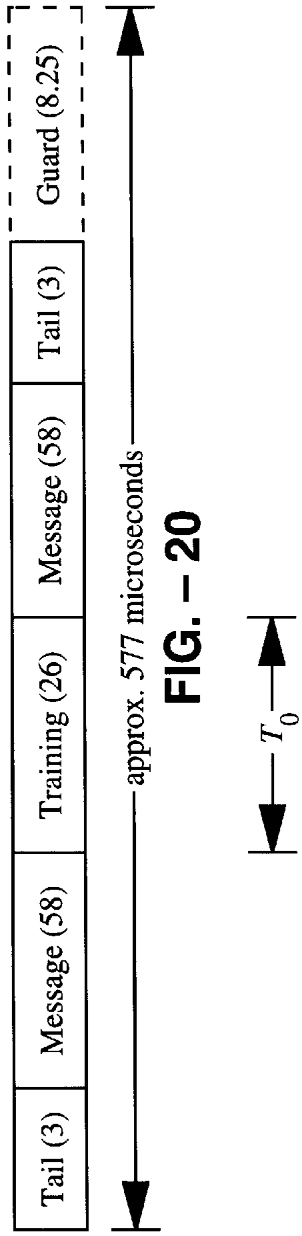


FIG. – 20

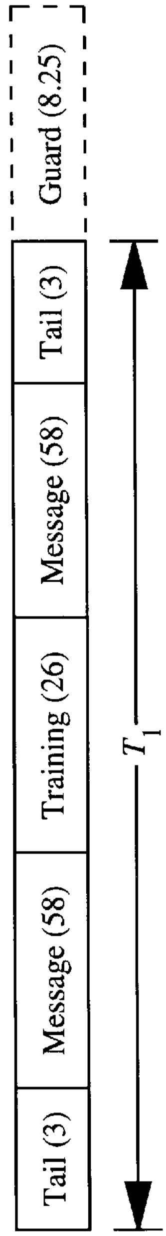
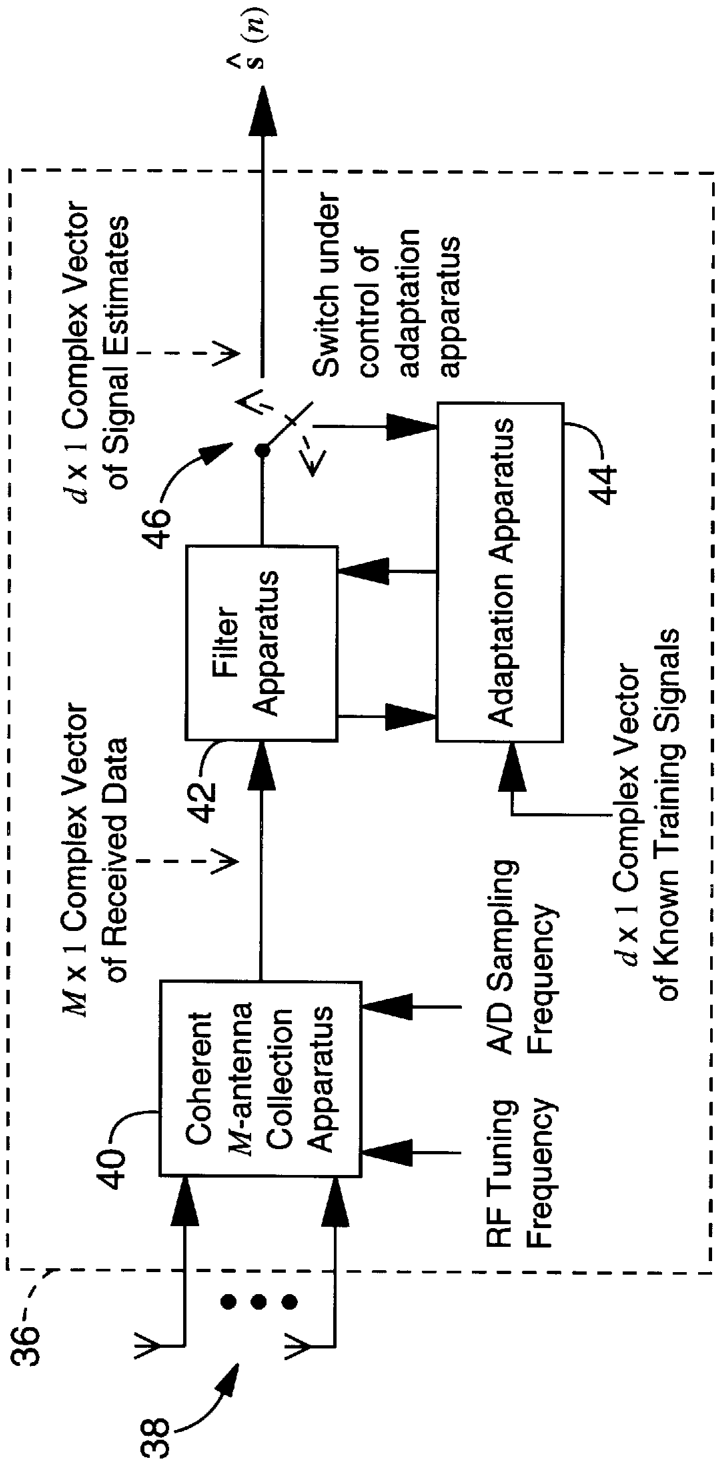


FIG. – 21



$\hat{s}(n)$
d x 1 Complex Vector of Signal Estimates
Switch under control of adaptation apparatus
46
Filter Apparatus
Adaptation Apparatus
44
M x 1 Complex Vector of Received Data
42
d x 1 Complex Vector of Known Training Signals
Coherent M-antenna Collection Apparatus
40
A/D Sampling Frequency
RF Tuning Frequency
36
38

FIG. – 22

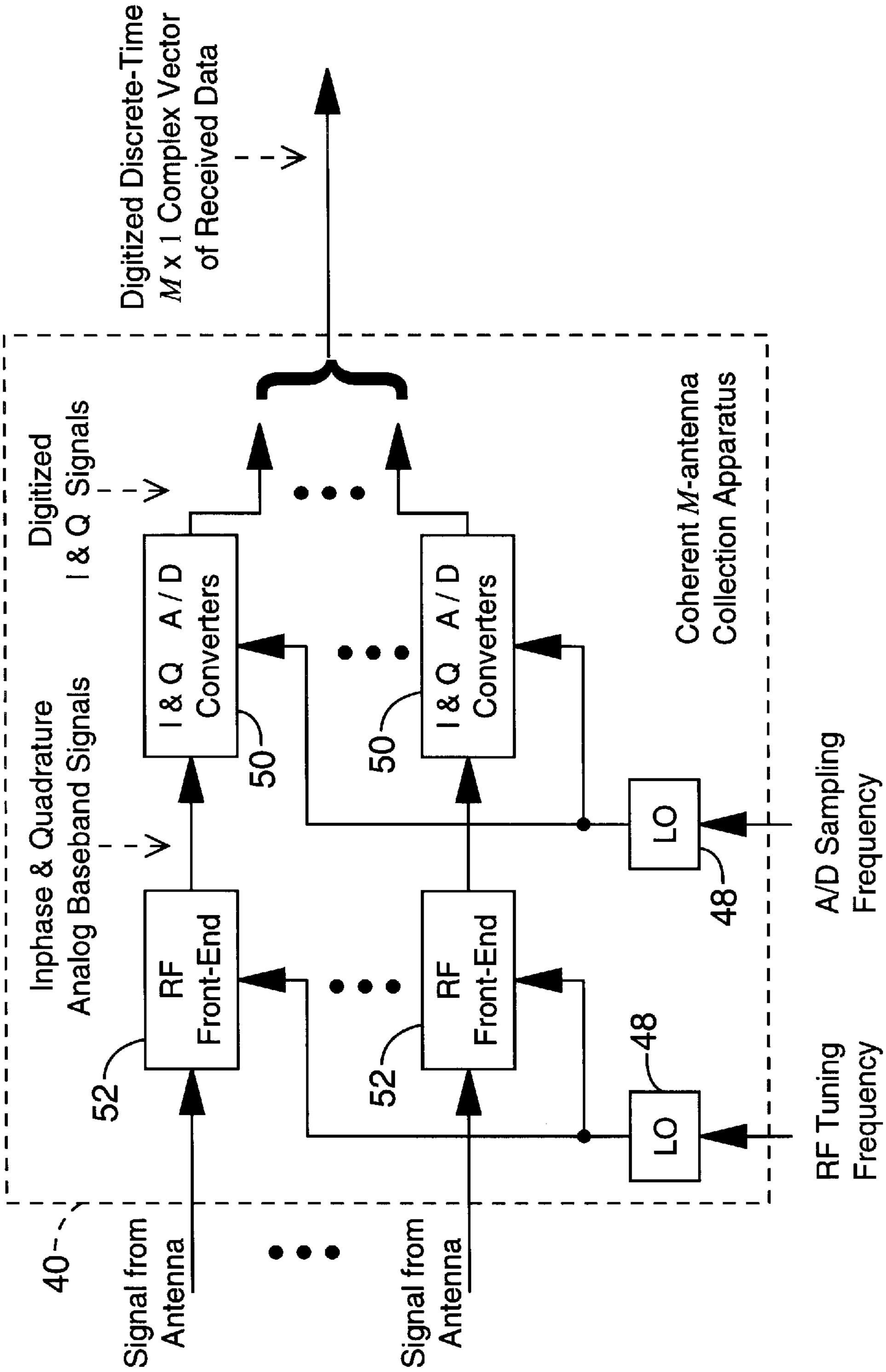

Digitized Discrete-Time
M x 1 Complex Vector
of Received Data
Digitized
I & Q Signals
Inphase & Quadrature
Analog Baseband Signals
Coherent M-antenna
Collection Apparatus
I & Q A / D
Converters
I & Q A / D
Converters
RF
Front-End
RF
Front-End
LO
LO
50
50
52
52
48
48
40
Signal from
Antenna
Signal from
Antenna
A/D Sampling
Frequency
RF Tuning
Frequency

FIG. – 23

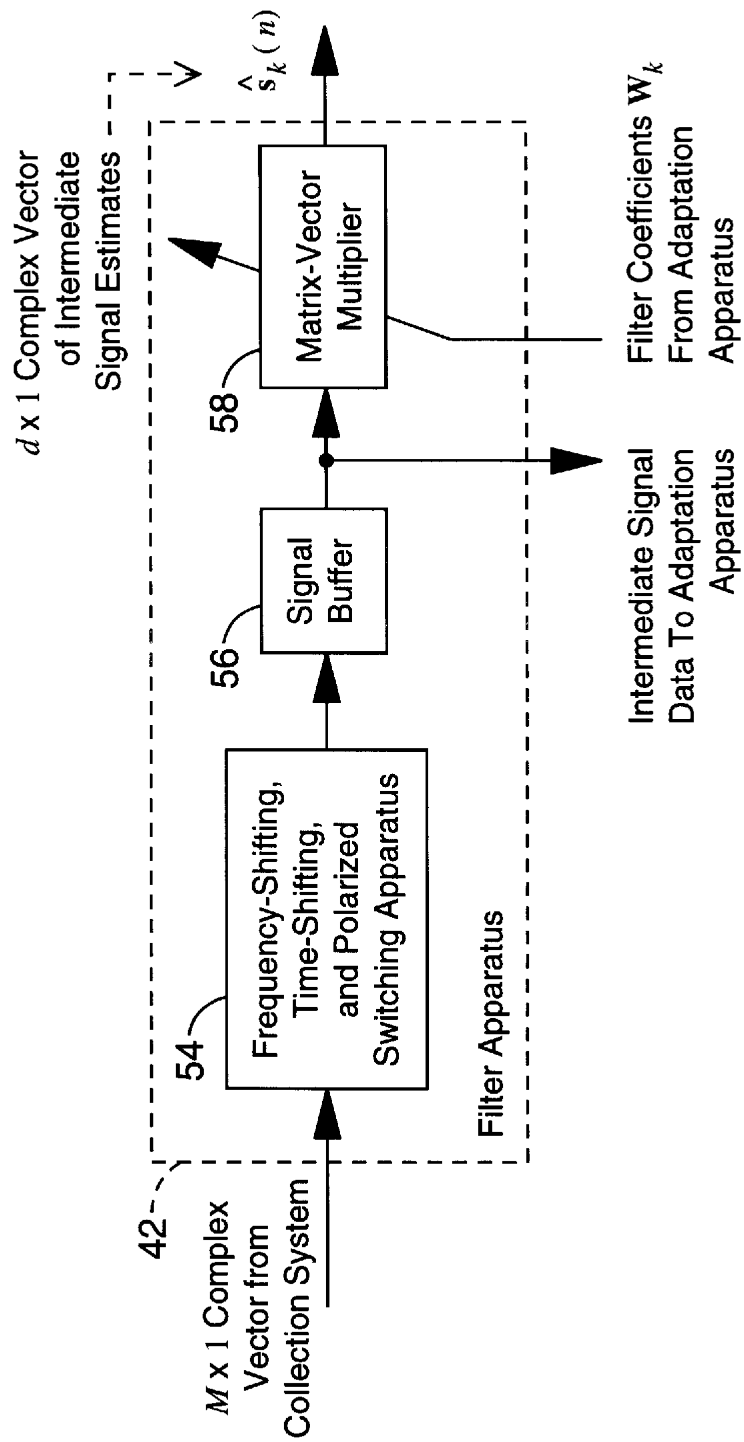

d x 1 Complex Vector
of Intermediate
Signal Estimates
$\hat{s}_k(n)$
58
Matrix-Vector
Multiplier
Filter Coefficients W_k
From Adaptation
Apparatus
56
Signal
Buffer
Intermediate Signal
Data To Adaptation
Apparatus
54
Frequency-Shifting,
Time-Shifting,
and Polarized
Switching Apparatus
Filter Apparatus
42
M x 1 Complex
Vector from
Collection System

FIG. – 24

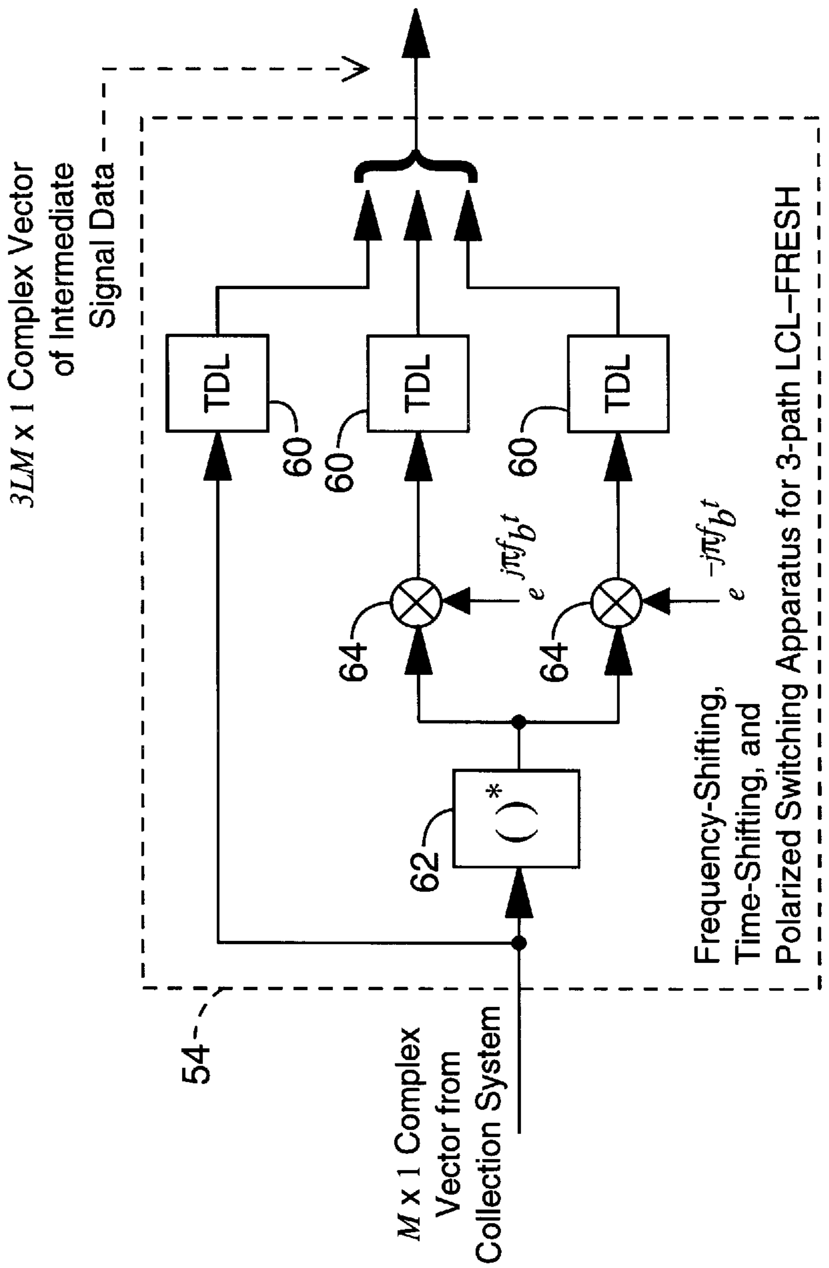

3LM x 1 Complex Vector
of Intermediate
Signal Data
M x 1 Complex
Vector from
Collection System
54
62
( )*
64
64
60
60
60
TDL
TDL
TDL
$e^{j\pi f_b t}$
$e^{-j\pi f_b t}$
Frequency-Shifting,
Time-Shifting, and
Polarized Switching Apparatus for 3-path LCL–FRESH

FIG. – 25

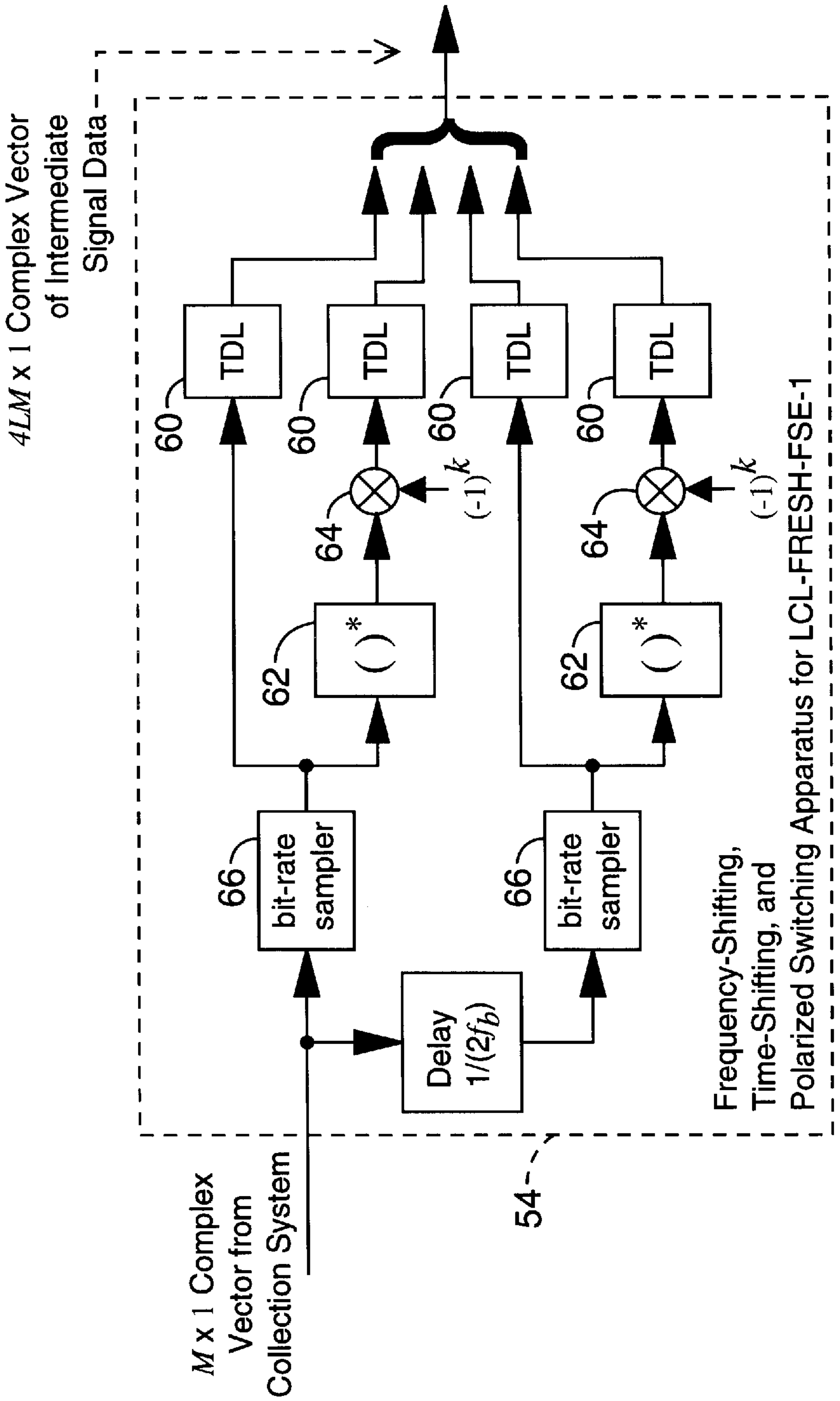
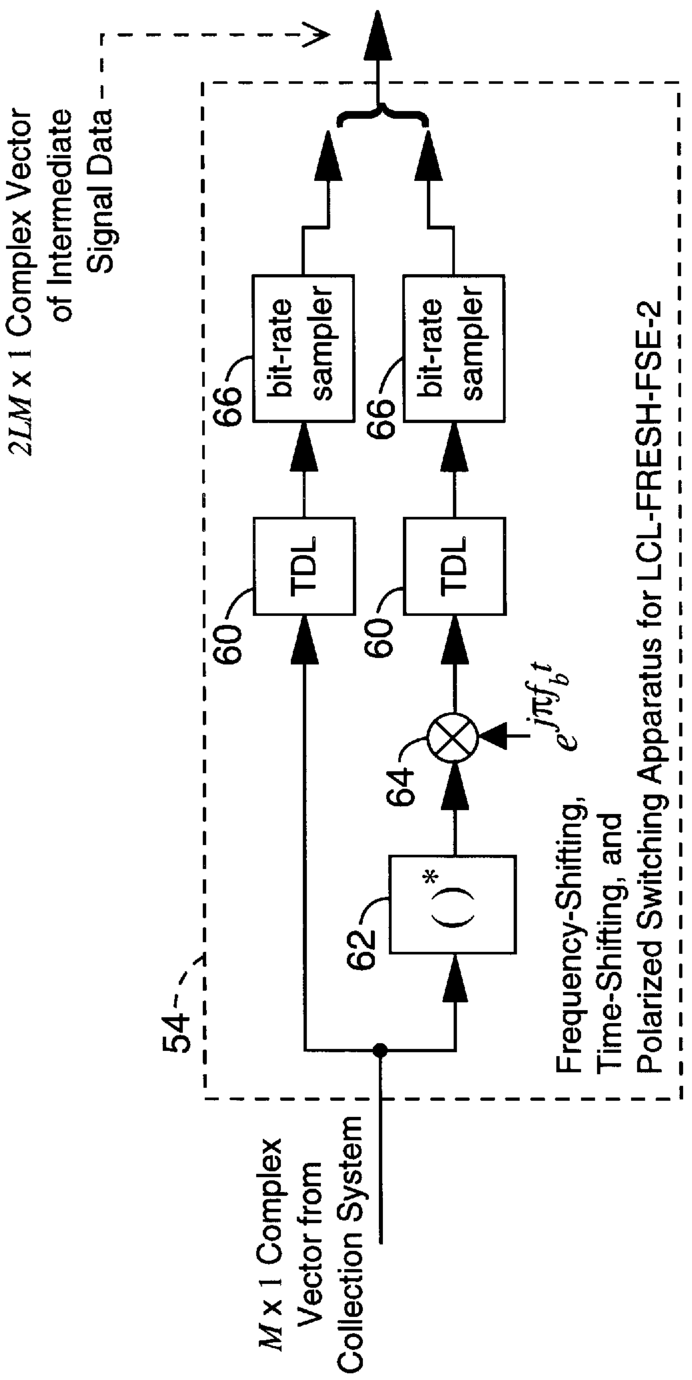
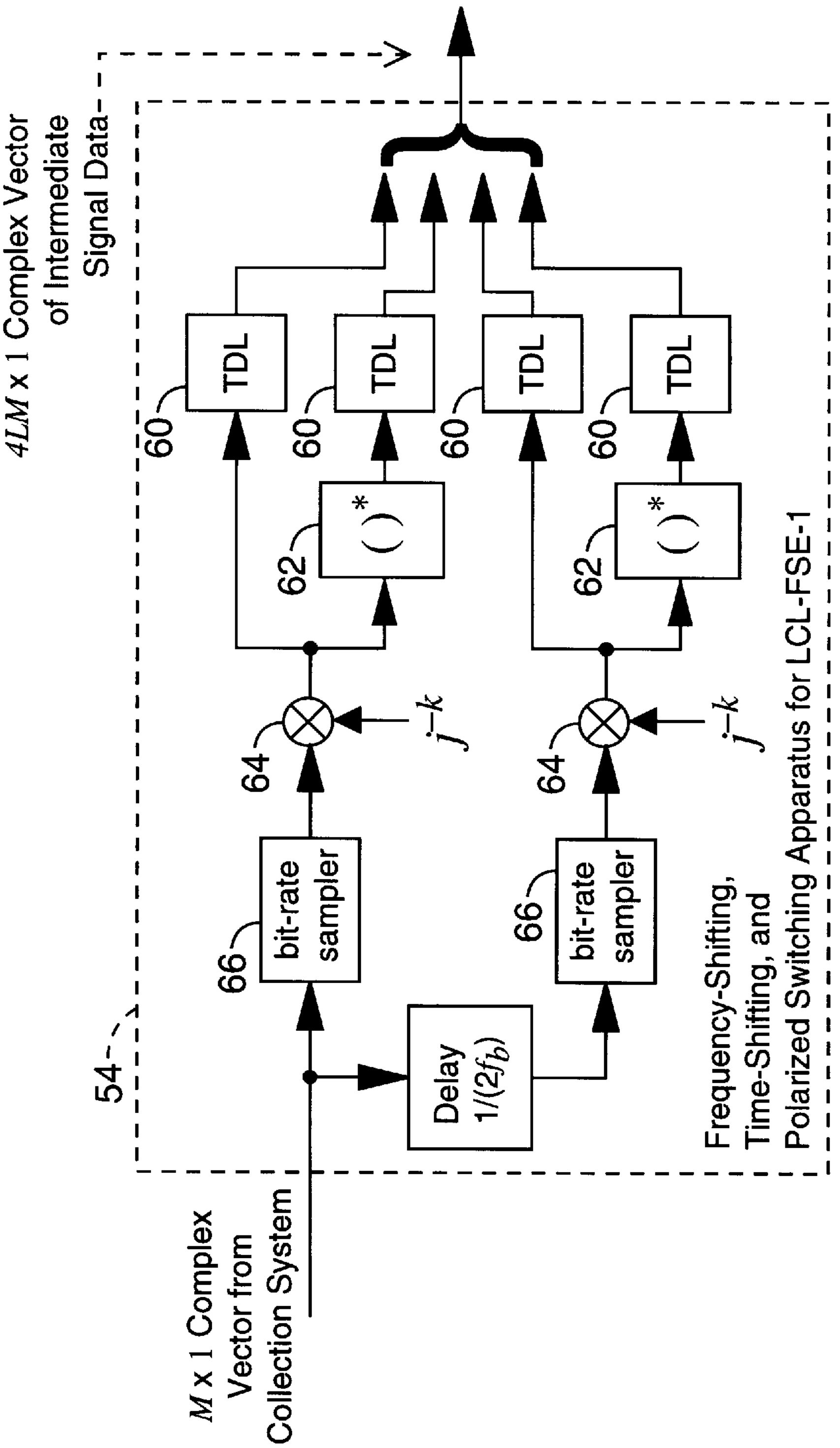

4LM x 1 Complex Vector
of Intermediate
Signal Data
60
TDL
60
TDL
60
TDL
60
TDL
64
(-1)^k
64
(-1)^k
62
( )*
62
( )*
66
bit-rate
sampler
66
bit-rate
sampler
Delay
1/(2f_b)
M x 1 Complex
Vector from
Collection System
54
Frequency-Shifting,
Time-Shifting, and
Polarized Switching Apparatus for LCL-FRESH-FSE-1

FIG. – 26



2LM x 1 Complex Vector of Intermediate Signal Data
54
60
TDL
66
bit-rate sampler
62
( )*
64
$e^{j\pi f_b t}$
60
TDL
66
bit-rate sampler
Frequency-Shifting, Time-Shifting, and Polarized Switching Apparatus for LCL-FRESH-FSE-2
M x 1 Complex Vector from Collection System

FIG. – 27



4LM x 1 Complex Vector of Intermediate Signal Data
60
TDL
60
TDL
60
TDL
60
TDL
62
( )*
62
( )*
64
j^{-k}
64
j^{-k}
66
bit-rate sampler
66
bit-rate sampler
Delay $1/(2f_b)$
54
Frequency-Shifting, Time-Shifting, and Polarized Switching Apparatus for LCL-FSE-1
M x 1 Complex Vector from Collection System

FIG. – 28

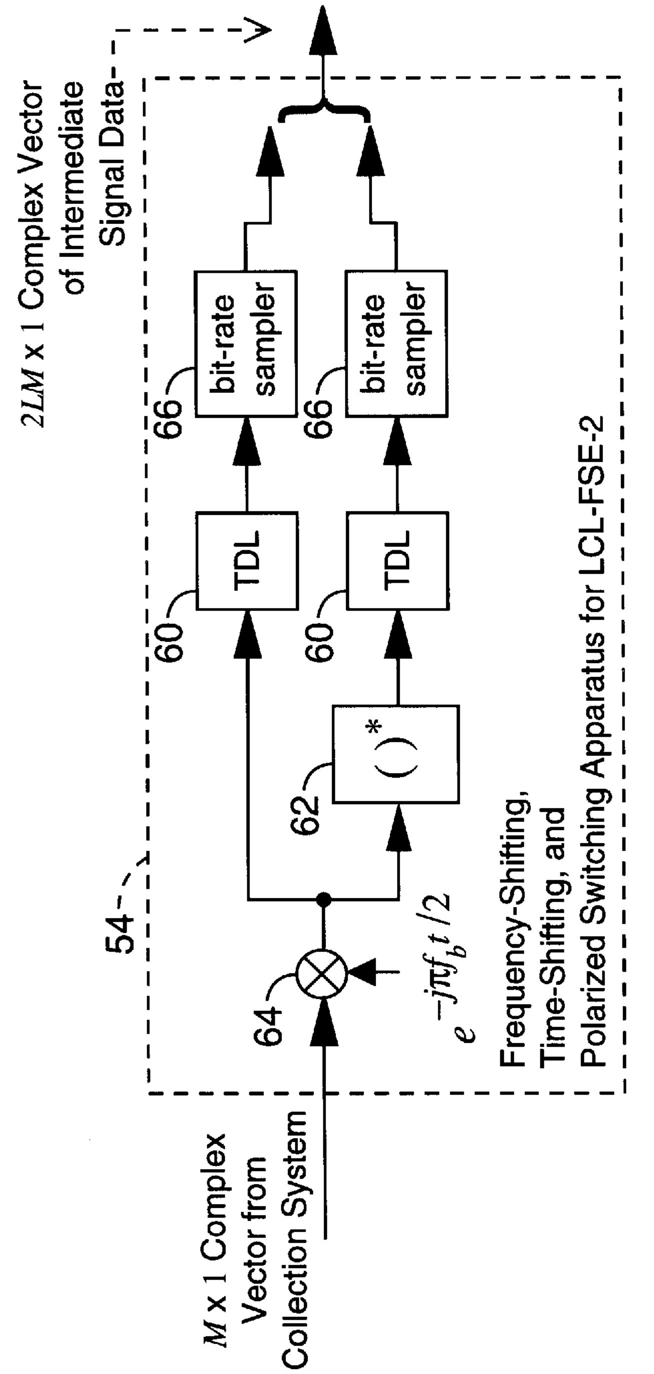

M x 1 Complex Vector from Collection System
$2LM$ x 1 Complex Vector of Intermediate Signal Data
54
64
$e^{-j\pi f_b t/2}$
62
( )*
60
TDL
60
TDL
66
bit-rate sampler
66
bit-rate sampler
Frequency-Shifting, Time-Shifting, and Polarized Switching Apparatus for LCL-FSE-2

FIG. – 29

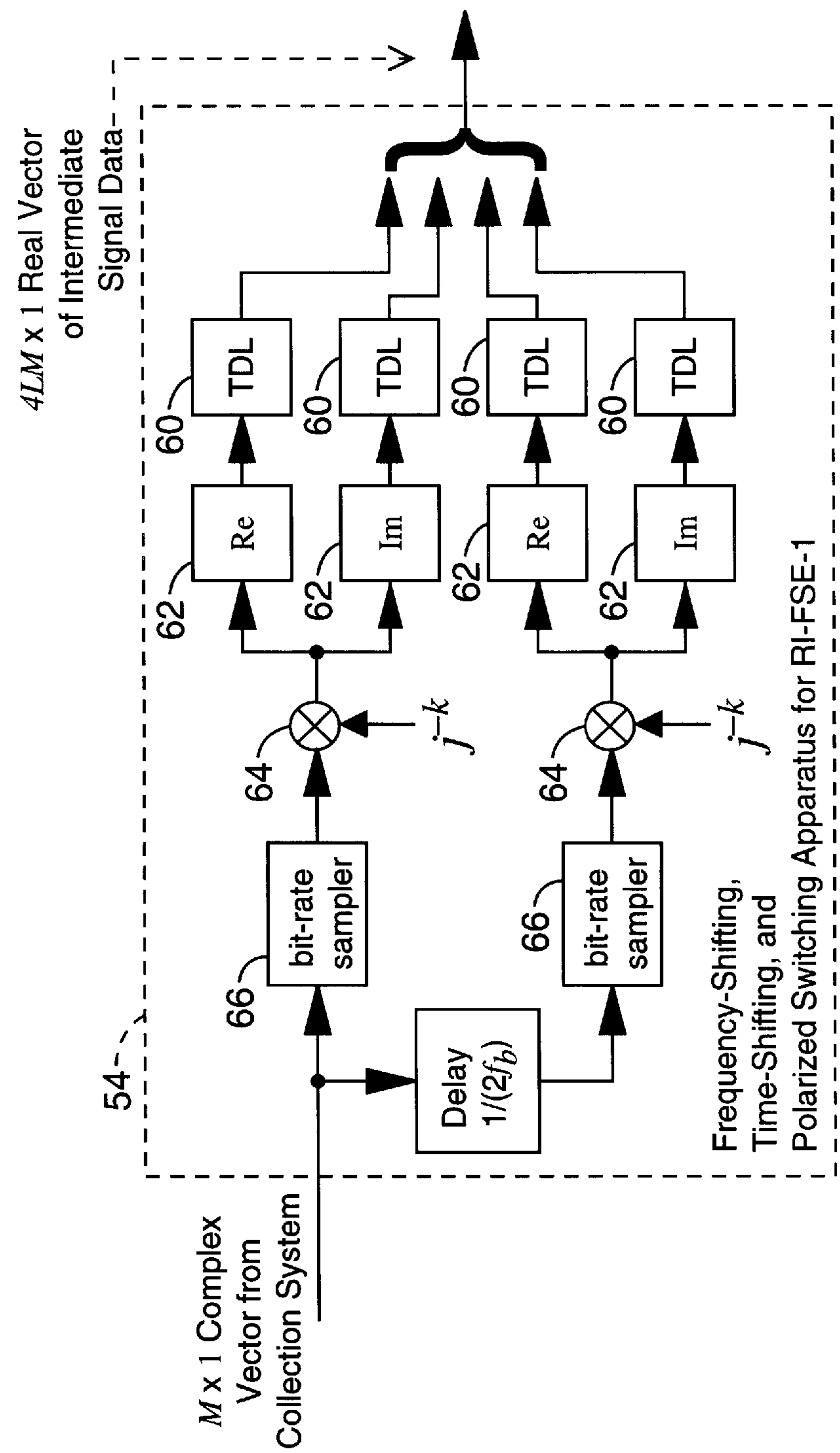

4LM x 1 Real Vector
of Intermediate
Signal Data
54
M x 1 Complex
Vector from
Collection System
Delay
1/(2f_b)
66
bit-rate
sampler
64
j^{-k}
62
Re
Im
60
TDL
Frequency-Shifting,
Time-Shifting, and
Polarized Switching Apparatus for RI-FSE-1

FIG. – 30

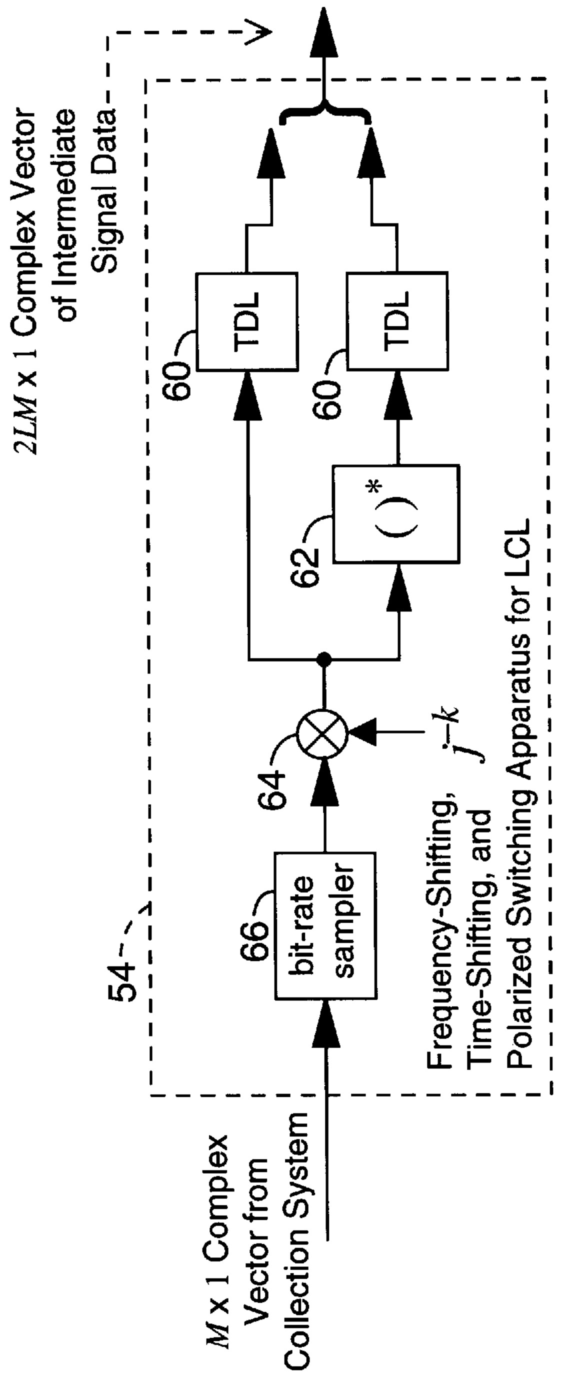

2LM x 1 Complex Vector of Intermediate Signal Data
54
bit-rate sampler
66
64
j^{-k}
62
( )*
60
TDL
60
TDL
M x 1 Complex Vector from Collection System
Frequency-Shifting, Time-Shifting, and Polarized Switching Apparatus for LCL

FIG. – 31

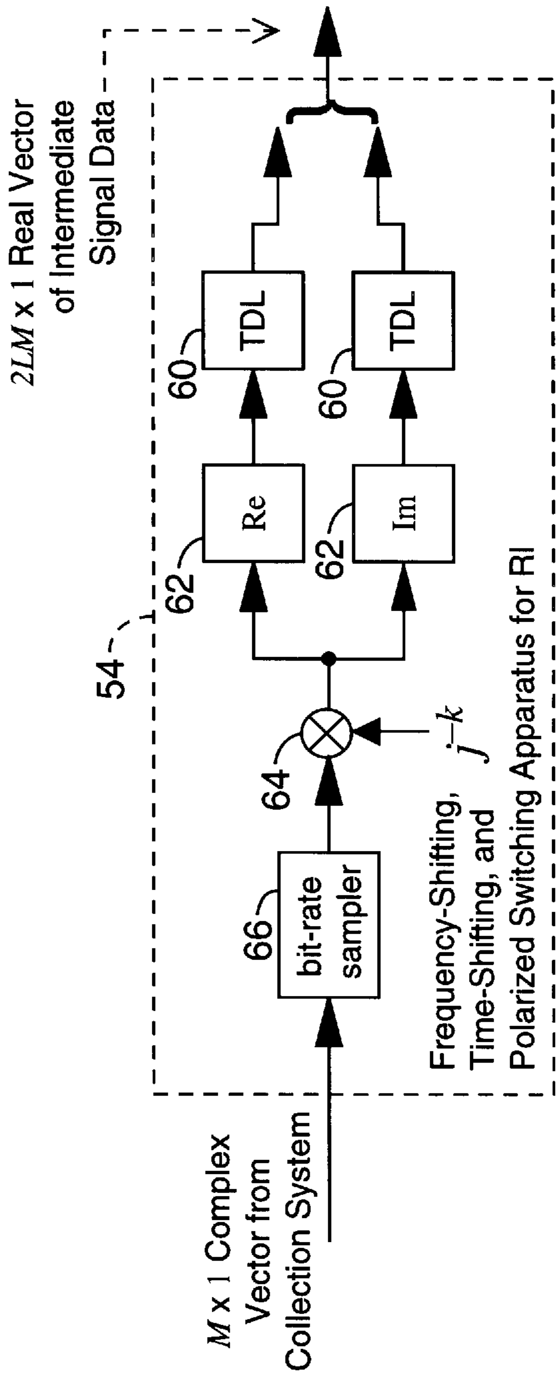

M x 1 Complex
Vector from
Collection System
bit-rate
sampler
66
64
j^{-k}
54
62
Re
62
Im
60
TDL
60
TDL
$2LM$ x 1 Real Vector
of Intermediate
Signal Data
Frequency-Shifting,
Time-Shifting, and
Polarized Switching Apparatus for RI

FIG. – 32

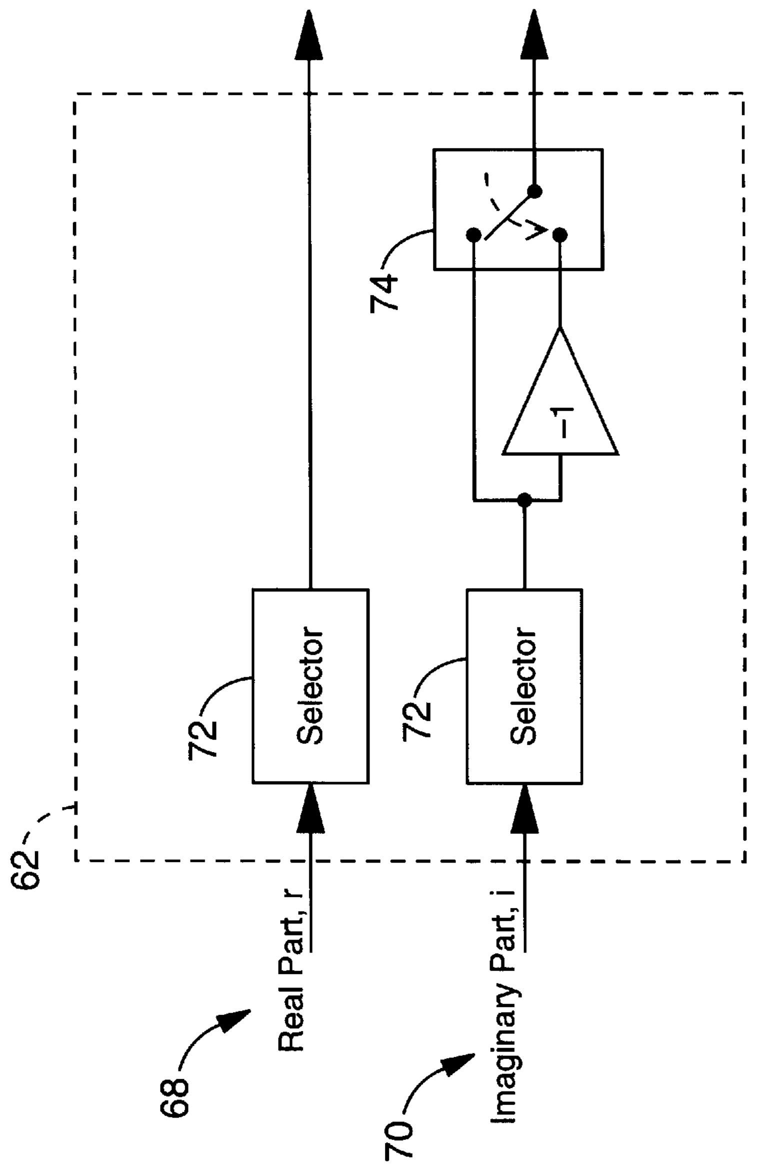

62
68
Real Part, r
70
Imaginary Part, i
72
Selector
72
Selector
−1
74

FIG. – 33

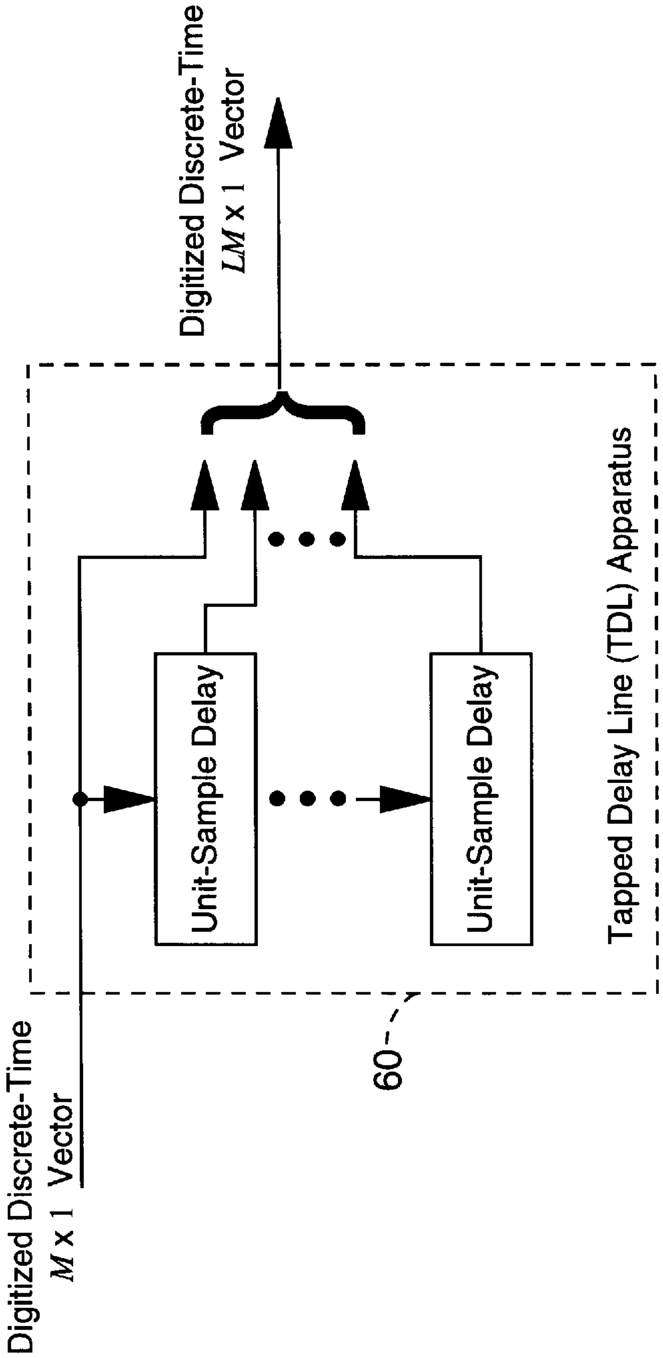

Digitized Discrete-Time
LM x 1 Vector
Digitized Discrete-Time
M x 1 Vector
Unit-Sample Delay
Unit-Sample Delay
Tapped Delay Line (TDL) Apparatus
60

FIG. – 34

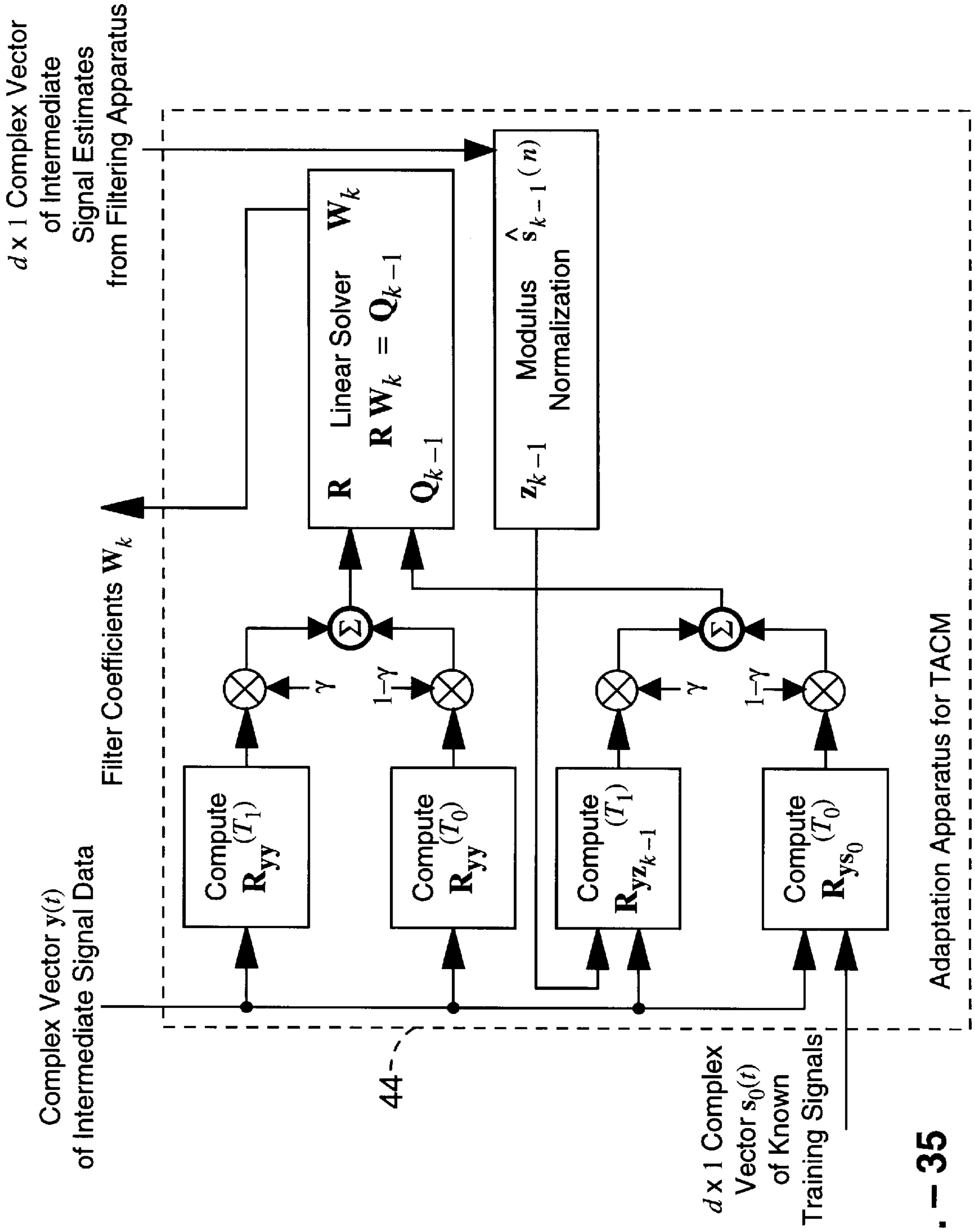

d x 1 Complex Vector of Intermediate Signal Estimates from Filtering Apparatus
Linear Solver
$\mathbf{R}$
$\mathbf{R}\,\mathbf{W}_k = \mathbf{Q}_{k-1}$
$\mathbf{Q}_{k-1}$
$\mathbf{W}_k$
Modulus Normalization
$\mathbf{z}_{k-1}$
$\hat{\mathbf{s}}_{k-1}(n)$
Filter Coefficients $\mathbf{W}_k$
Σ
γ
$1-\gamma$
Compute $\mathbf{R}_{\mathbf{yy}}^{(T_1)}$
Compute $\mathbf{R}_{\mathbf{yy}}^{(T_0)}$
Compute $\mathbf{R}_{\mathbf{yz}_{k-1}}^{(T_1)}$
Compute $\mathbf{R}_{\mathbf{ys}_0}^{(T_0)}$
Complex Vector $\mathbf{y}(t)$ of Intermediate Signal Data
Adaptation Apparatus for TACM
44
d x 1 Complex Vector $\mathbf{s}_0(t)$ of Known Training Signals

FIG. – 35

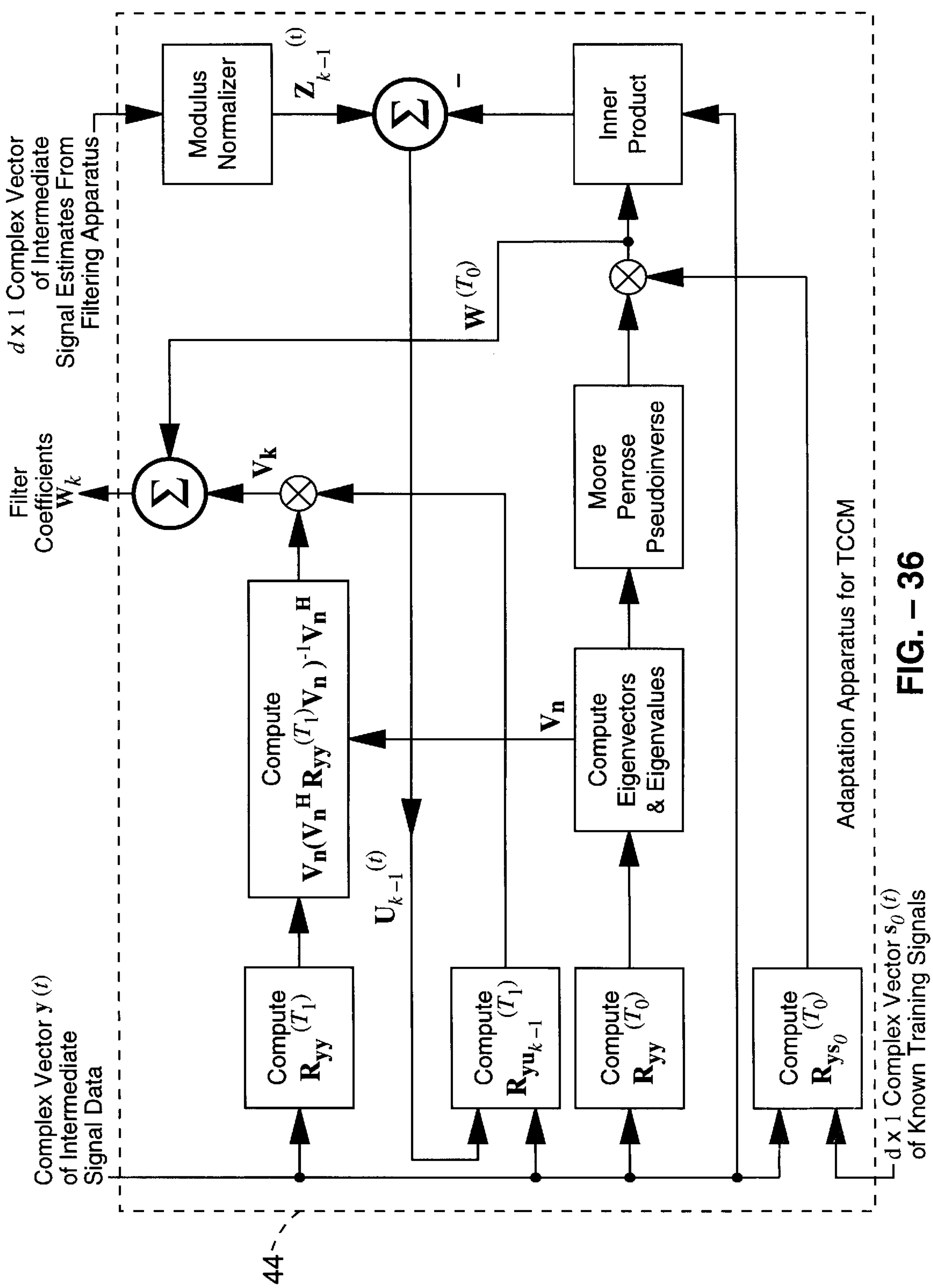

d x 1 Complex Vector of Intermediate Signal Estimates From Filtering Apparatus
Modulus Normalizer
$Z_{k-1}(t)$
Inner Product
$W(T_0)$
Moore Penrose Pseudoinverse
Filter Coefficients W_k
V_k
Compute $V_n(V_n^H R_{yy}^{(T_1)} V_n)^{-1} V_n^H$
V_n
Compute Eigenvectors & Eigenvalues
Adaptation Apparatus for TCCM
$U_{k-1}(t)$
Complex Vector y (t) of Intermediate Signal Data
Compute $R_{yy}^{(T_1)}$
Compute $R_{yu_{k-1}}^{(T_1)}$
Compute $R_{yy}^{(T_0)}$
Compute $R_{ys_0}^{(T_0)}$
d x 1 Complex Vector $s_0(t)$ of Known Training Signals
44

FIG. – 36

GMSK SIGNAL PROCESSORS FOR IMPROVED COMMUNICATIONS CAPACITY AND QUALITY

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention pertains generally to reducing interference in wireless communications systems using Gaussian Minimum Shift Keyed (GMSK) or other Minimum Shift Keyed (MSK) signals, and more specifically to a method and apparatus for (i) rejecting interfering signals received at a communications receiver and occupying the same or adjacent spectral band as one or more desired signals, (ii) separating, if needed, multiple desired communication signals received at a communications receiver, and (iii) correcting distortion that affects one or more desired communication signals received at a communications receiver such as that caused by a multipath radio-frequency (RF) propagation channel.

2. Description of the Background Art

The background of the present invention falls into two general categories: (i) work pertaining to the filtering structures that perform interference rejection, signal separation, and distortion correction; and (ii) work pertaining to the adaptive algorithms that adjust the coefficients in these filtering structures to allow them to operate in signal environments that are endemic to mobile, portable, and personal wireless communication systems. This work, however, differs substantially from the present invention.

A. Filtering Structures

With the exception of the single-sensor FRESH filter and the Joint Maximum Likelihood Sequence Estimator, all prior work on interference rejection and signal separation that could be applied to a system that receives multiple GMSK or MSK signals in the same spectral band and time interval requires that multiple sensors (antennas) be used. Another requirement of this body of work is that the spatial characteristics of any two signals received by the multiple sensors must be sufficiently different; if this requirement is not met (or is poorly met) then the filtering structure will not be able to reject the interference and separate the desired signals (or will do so with poor quality). This body of work includes all filtering structures that belong to the class of linear time-invariant (LTI) spatio-temporal filters; this class of filters is defined by the fact that their outputs can be modeled mathematically as sums of amplitude-scaled, phase-shifted, and time-shifted versions of the output signals of the multiple sensors. Notably, this class of filters characterizes virtually all filtering structures that have been previously developed to address the problem (and problems similar to it) solved by the present invention.

With regard to the single-sensor FRESH filter, none of the prior works (i) discuss or describe application of FRESH filtering to GMSK or other MSK signal types, (ii) discuss or describe application to GSM or DECT standards, (iii) suggest using frequency shifts equal to plus or minus one-half the bit rate of a signal or equal to two times the carrier frequency plus or minus one-half the bit rate, (iv) show a 3-path LCL-FRESH structure relevant to multiple GMSK signals with equal carrier frequencies and bit rates which (for complex envelope with zero carrier offset) contains no unconjugated frequency-shifted paths, (v) show a 2-path, 2-input LCL-FRESH structure relevant to adjacent-channel GMSK interference, (vi) show a 3-path, 3-input LCL-FRESH structure relevant to adjacent-channel GMSK interference, (vii) show a multi-sensor 3-path LCL-FRESH

structure comprising one each of identical versions of a 2-path or 3-path LCL-FRESH structure at the output of each sensor with all 2-path LCL-FRESH filter outputs summed, (viii) describe any partially blind methods for adaptation of any filter to separate GMSK or MSK signals, (ix) describe any training-aided constant modulus algorithm (CMA) methods for adaptation for any signal type, (x) demonstrate separability of any GMSK or MSK signals using any FRESH filtering method, (xi) describe the theoretical spectral redundancy properties of GMSK that provide for signal separability using FRESH filtering, or (xii) show the measured spectral redundancy properties of GMSK signals. Furthermore, the single-sensor FRESH filter has a severe limitation that is transcended by the present invention: it cannot accommodate multiple sensors in any way that allows it to reject and/or separate more than 2 co-channel and 2 adjacent-channel GMSK or MSK signals, in contrast to the present invention.

Although a Joint Maximum Likelihood Sequence Estimator (JMLSE) operating with a single sensor may theoretically have the ability to reject and/or separate more than 2 GMSK or MSK signals, the quality of the separated signals must degrade substantially when the number of these exceeds two. Furthermore, the computational complexity of the JMLSE is extremely high and is not practical to implement in real-time without expensive hardware that draws copious amounts of electrical energy. These drawbacks become more severe (at an exponential rate) as the number of sensors to be accommodated increases beyond one and the number of signals to be jointly separated increases.

The filtering structures of the present invention do not suffer from the severe implementational drawbacks of the JMLSE; they accommodate single or multiple sensors to great advantage; and they can reject and/or separate multiple signals even when one or more pairs of these signals cannot be rejected and/or separated by the LTI spatio-temporal filters. More specifically, the filtering structures of the present invention provide capabilities, for rejecting interfering signals and separating desired signals and correcting their distortion, that are not provided by any prior filtering structures.

B. Adaptation Methods

The filtering structures used in this invention, and in other means and apparatus for interference rejection, signal separation, and correction of distortion are controlled through the selection of values of filter parameters or coefficients. In some embodiments of these filtering structures, these coefficients can include the gains and phases used to weight various signals prior to summing. In typical wireless communication systems, the values of these coefficients must vary properly because the spatio-temporal characteristics of the received signals vary in time. That is, the values of the coefficients of these filtering structures must be chosen adaptively.

With the exception of the present invention, there are two classes of methods for adaptively choosing these coefficient values: (i) conventional methods that rely exclusively on the knowledge of training sequences embedded in the desired signals, this knowledge being used to choose an initial setting for the coefficient values, typically according to a least-squared error criterion; and (ii) blind methods that do not use embedded training sequences but instead exploit one or more properties of the interference or of the desired signals.

Conventional methods that use a known training signal are well-known to those skilled in the art of signal processing, and are described in the open literature pertaining to adaptive signal processing and adaptive filtering. Conventional methods are limited in their applicability, however, because the number of coefficient values that they can adapt successfully must be less (much less for good performance) than the number of known samples within the embedded training sequence. This number of known samples is fixed by the wireless communication system (e.g., for all signals in GSM that convey voice traffic, this number is the product of 26 and the number of effectively independent samples per bit (which is typically 2) used in the digitization of the continuous-time received signals, assuming a discrete-time implementation). With regard to the problem addressed by the present invention, in which the filtering structures can have, potentially, a large number of adjustable coefficients, the number of known samples within the embedded training sequence can be insufficient to provide a useful initial setting. A filtering structure initialized in this way would not provide sufficiently high quality estimates of the desired signals to be useful in the communications system.

In contrast, blind methods can make use of the entire record of the received signals over which the spatio-temporal characteristics (i.e., angles of arrival, relative delays, and relative phases of the multiple propagation paths of the multiple signals, and carrier and bit phases of the transmitted signals) are approximately constant. This circumvents the limit imposed by conventional methods on the number of filter coefficients that can be adapted. However, the various blind methods have other severe drawbacks that prevent them from operating successfully in many situations, including the one addressed by the present invention. Specifically, the class of Constant Modulus Algorithms (CMA) and related methods can require much longer data records, to converge to a high-quality solution, than are available. Furthermore, their convergence to a solution is not guaranteed, leading to a confusion between interfering signals (which should be rejected) and desired signals (which should be separated from each other and from the interference). Another class of blind methods relies on estimating the directions of arrival of the received signals; these estimates are then used in a table lookup (through the array manifold, which is a table of calibration data for the multiple-sensor receiver) to find the coefficients of an LTI spatio-temporal filter. However, these so-called direction-finding methods tend to perform poorly in the presence of severe multipath, unknown number of signals, and unknown interference and noise fields, these three impairments being endemic to the wireless communication systems of interest in the present invention.

Furthermore, the direction-finding methods do not provide the types of parameter estimates that are needed to adapt the filtering structures of the present invention.

The adaptation methods of the present invention exploit in a novel manner both the embedded training sequences and the constant modulus property of the desired signals, and in doing so circumvent the undesirable aspects of the conventional methods and of the blind methods. That is, the adaptation methods of the present invention, which choose the values of coefficients in filtering structures, such as those of the present invention, provide capabilities that are not provided by prior adaptation methods.

SUMMARY OF THE INVENTION

The present invention provides a method and apparatus for rejecting undesired interference and separating and correcting the distortion of desired interfering signals of the Gaussian Minimum Shift Keyed (GMSK) or other Minimum Shift Keyed (MSK) type, which arise for example in wireless communication systems based on either the Digital European Cordless Telecommunications (DECT) standard or the Global System for Mobile Communications (GSM) standard and, as a result, improving the quality and quantity of service offered by a wireless communications system that uses GMSK or other MSK signals.

The utility of the present invention may be appreciated, for example but not exclusively, by manufacturers of wireless communications equipment and providers of wireless communications services who wish to improve the quality of communications offered to users of their systems, and/or who wish to increase the number of users of their systems (e.g., to increase revenue in a commercially operated system).

Quantity of service improves when a base station is able to support a greater number of users for a fixed frequency allocation by, for example, being able to reduce frequency reuse distance, extend range, possibly reuse frequencies within a cell, and more frequently assign users to adjacent channels. Quality of service improves, for example, when requests made to a central coordinator (hereafter referred to as the base station) by users to initiate calls are rejected less frequently (i.e., blocking probability is reduced), when calls in progress by mobile users are not abnormally terminated (i.e., probability of dropped calls is reduced), and when the intelligibility of the voice messages in the call is better (i.e., the parties to the call hear less noise, interference, abnormal silences, and the like).

The present invention addresses these general problems by means of a method and apparatus that perform the following functions: (1) rejection of interfering signals received at a communications receiver and occupying the same (or adjacent) spectral band as one or more desired signals, (2) separation (if needed) of multiple desired communication signals received at a communications receiver, and (3) correction of distortion that affects one or more desired communication signals received at a communications receiver (e.g., distortion caused by a multipath radio-frequency (RF) propagation channel).

Accordingly, an object of the invention is to provide an array of one or more antennas whose outputs are processed by conventional RF front-ends and are then processed by a filtering structure that rejects interference, separates one or more desired signals, and corrects distortion to obtain high-quality estimates of desired signals.

Another object of the invention is to provide a class of filtering structures that can reject interfering signals, separate one or more desired signals, and correct the distortion in these desired signals.

Another object of the invention is to provide a class of filtering structures that can operate on the outputs of any number of antennas after these outputs are downconverted by conventional RF front-ends, including the special case when only one antenna is used.

Another object of the invention is to provide a linear-conjugate-linear (LCL) frequency-shift (FRESH) filter that uses the particular pattern of frequency-shifts and conjugations to exploit the cyclostationarity properties that are characteristic of GMSK and other MSK signals, thereby providing the desired interference-rejection, signal-separation, and distortion-correction capabilities.

Another aspect of the invention is to provide a linear-conjugate-linear polyperiodically time-variant (LCL-PTV) filter that uses the particular pattern of conjugations and time-variant filter coefficients to exploit the cyclostationarity properties that are characteristic of GMSK and other MSK signals, thereby providing the desired interference-rejection, signal-separation, and distortion-correction capabilities.

Another object of the invention is to provide an LCL-FRESH fractionally-spaced equalizer (LCL-FRESH-FSE) that operates on higher-than-bit-rate-sampled data to provide bit-rate-sampled output and uses the particular pattern of frequency-shifts and conjugations to exploit the cyclostationarity properties that are characteristic of GMSK and other MSK signals, thereby providing the desired interference-rejection, signal-separation, and distortion-correction capabilities.

Another object of the invention is to provide an LCL-FSE filter that operates on higher-than-bit-rate-sampled and frequency-shifted (by one-fourth of the bit rate) data and uses the sum of FSE processed data and conjugated data to exploit the cyclostationarity properties that are characteristic of GMSK and other MSK signals, thereby providing the desired interference-rejection, signal-separation, and distortion-correction capabilities.

Another object of the invention is to provide a real-imaginary (RI-FSE) filter that provides a computationally efficient alternate embodiment of the LCL-FSE filter.

Another object of the invention is to provide an LCL filter that operates on bit-rate-sampled and frequency-shifted (by one-fourth of the bit rate) data and uses the sum of linearly filtered data and conjugated data to exploit cyclostationarity properties that are characteristic of GMSK and other MSK signals, thereby providing the desired interference-rejection, signal-separation, and distortion-correction capabilities.

Another object of the invention is to provide a real-imaginary (RI) filter that provides a computationally efficient alternate embodiment of the LCL filter.

Another object of the invention is to provide a class of methods for adapting the adjustable coefficients in any of the filtering structures of this invention so that the filter can properly perform the functions of rejecting interference, separating one or more desired signals, and correcting distortion.

Another aspect of the invention is to provide a means of adapting the adjustable coefficients, by using both the known training sequence(s) embedded in the desired signal(s) and the knowledge that the desired signal(s) have constant modulus, thus providing training-augmented constant-modulus (TACM) adaptation.

Another object of the invention is to provide a means of adapting the adjustable coefficients, by using the known training sequence(s) embedded in the desired signal(s) to constrain adaptation that uses the knowledge that the desired signal(s) have constant modulus, thus providing training-constrained constant-modulus (TCCM) adaptation.

These general aspects and their details give this invention significant capabilities of great utility in wireless communication systems. The capability provided by this invention can be used to improve the quality and quantity of communication services provided by such systems. The quality of service can be improved by using the invention to suppress interference in a particular frequency band and time slot jointly referred to as a channel) and spatial cell caused by other users in the same or adjacent channels and in the same or adjacent spatial cells and to correct signal distortion; and the quantity of service can be increased by reallocating channels in a manner that intentionally results in more interfering, but separable by this invention, signals per channel and cell, in order to accommodate more users.

These improvements are made possible by this invention because this invention, using only one sensor, can separate multiple spectrally overlapping signals of certain types and can correct even severe frequency-selective fading and other distortion and, because the number of such signals that can be separated and corrected by this invention, using multiple sensors increases proportionately with the number of sensors. In cellular communication systems that use base stations, the allocation of channels can (but need not) be made so that the introduced interference occurs only in up-link channels (from mobile units to base stations) so that the invention need only be implemented in the base station receivers and not in the mobile receivers. The invention can also be used at base stations (or at mobile units, or both) using only a single receiving antenna or multiple antennas.

With reference to the specific example of GMSK and other MSK signals, between 2 and 4 GMSK and other MSK signals in a single channel can be accommodated with one antenna, depending on how many signals arise from users in the same channel and how many arise from adjacent channels. More generally, with a number M of antennas, between 2M and 4M signals can be accommodated.

Further objects and advantages of the invention will be brought out in the following portions of the specification, wherein the detailed description is for the purpose of fully disclosing preferred embodiments of the invention without placing limitations thereon.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention will be more fully understood by reference to the following drawings which are for illustrative purposes only:

FIG. **1** is a graph showing the cross-coherence functions of a sample GMSK signal and its conjugated and frequency-shifted versions.

FIG. **2**A is a graph showing the spectral content of a GMSK (0.3) signal, x(t).

FIG. **2**B is a graph showing the spectral content of a conjugated and frequency shifted version of the signal shown in FIG. **2**A, $y_1(t)=x^*(t)e^{-j\pi f_b t}$.

FIG. **2**C is a graph showing the spectral density of the degree to which x shown in FIG. **2**A and y, shown in FIG. **2**B are correlated.

FIG. **3**A is a graph showing the spectral content of a GMSK (0.3) signal, x(t).

FIG. **3**B is a graph showing the spectral content of a conjugated and frequency shifted version of the signal shown in FIG. **3**A, $Y_2(t)=x^*(t)e^{j\pi f_b t}$.

FIG. **3**C is a graph showing the spectral density of the degree to which x shown in FIG. **3**A and y_2 shown in FIG. **3**B are correlated.

FIG. **4** is a functional block diagram of a 3-path LCL-FRESH filter in accordance with the present invention for performing interference rejection and/or co-channel separation and/or distortion correction for GMSK or other MSK signals.

FIG. **5** is a functional block diagram of an LCL-PTV filter in accordance with the present invention for performing interference rejection and/or co-channel separation and/or distortion correction for GMSK or other MSK signals.

FIG. **6** is a functional block diagram of a "type-1" LCL-FRESH-FSE filter in accordance with the present invention for performing interference rejection and/or co-channel separation and/or distortion correction for GMSK or other MSK signals.

FIG. **7** is a functional block diagram of a "type-2" LCL-FRESH-FSE filter in accordance with the present invention for performing interference rejection and/or co-channel separation and/or distortion correction for GMSK or other MSK signals.

FIG. **8** is a functional block diagram of a "type-1" LCL-FSE filter in accordance with the present invention for performing interference rejection and/or co-channel separation and/or distortion correction for GMSK or other MSK signals.

FIG. **9** is a functional block diagram of a "type-2" LCL-FSE filter in accordance with the present invention for performing interference rejection and/or co-channel separation and/or distortion correction for GMSK or other MSK signals.

FIG. **10** is a functional block diagram of a "type-1" RI-FSE filter in accordance with the present invention for performing interference rejection and/or co-channel separation and/or distortion correction for GMSK or other MSK signals.

FIG. **11** is a functional block diagram of a "type2" RI-FSE filter in accordance with the present invention for performing interference rejection and/or co-channel separation and/or distortion correction for GMSK or other MSK signals.

FIG. **12** is a functional block diagram of an LCL filter in accordance with the present invention for performing interference rejection and/or co-channel separation and/or distortion for GMSK or other MSK signals.

FIG. **13** is a functional block diagram of an RI filter in accordance with the present invention for performing interference rejection and/or co-channel separation and/or distortion correction for GMSK or other MSK signals.

FIG. **14** is a functional block diagram of a 5-path LCL-FRESH filter in accordance with the present invention for performing interference rejection and/or co-channel signal separation and/or distortion correction, with improved cancellation of adjacent channel interference, for GMSK or other MSK signals.

FIG. **15** is a functional block diagram of a 2-input, 2-path LCL-FRESH filter in accordance with the present invention for canceling adjacent upper channel interference.

FIG. **16** is a functional block diagram of a 2-input, 2-path LCL-FRESH filter in accordance with the present invention for canceling adjacent lower channel interference.

FIG. **17** is a functional block diagram of a 3-input, 3-path LCL-FRESH filter in accordance with the present invention for canceling both adjacent upper and adjacent lower channel interference.

FIG. **18** is a functional block diagram of a joint space-time multi-sensor FRESH filter in accordance with the present invention.

FIG. **19** is a functional block diagram of space-time factorized processor in accordance with the present invention.

FIG. **20** is a data structure timing diagram for a time slot in a GSM system, showing the number and location of each type of bit.

FIG. **21** is a data structure timing diagram showing the time periods T_0 and T_1 for the time slot shown in FIG. **20**.

FIG. **22** is a functional block diagram of a receiver apparatus in accordance with the present invention.

FIG. **23** is a functional block diagram of an embodiment of the coherent M-antenna collection apparatus block shown in FIG. **22**.

FIG. **24** is a functional block diagram of an embodiment of the filter apparatus block shown in FIG. **22**.

FIG. **25** is a functional block diagram of an embodiment of the frequency-shifting, time-shifting, and polarized switching apparatus block of the filter shown in FIG. **24** where the filter is configured as a 3-path LCL-FRESH filter.

FIG. **26** is a functional block diagram of an embodiment of the frequency-shifting, time-shifting, and polarized switching apparatus block of the filter shown in FIG. **24** where the filter is configured as a type-1 LCL-FRESH-FSE filter.

FIG. **27** is an alternative embodiment of the frequency-shifting, time-shifting, and polarized switching apparatus shown in FIG. **26** where the filter is configured as a type2 LCL-FRESH-FSE filter.

FIG. **28** is a functional block diagram of an embodiment of the frequency-shifting, time-shifting, and polarized switching apparatus block of the filter shown in FIG. **24** where the filter is configured as a type1 LCL-FSE filter.

FIG. **29** is an alternative embodiment of the frequency-shifting, time-shifting, and polarized switching apparatus shown in FIG. **28** where the filter is configured as a type2 LCL-FSE filter.

FIG. **30** is a functional block diagram of an embodiment of the frequency-shifting, time-shifting, and polarized switching apparatus block of the filter shown in FIG. **24** where the filter is configured as a type1 RI-FSE filter.

FIG. **31** is a functional block diagram of an embodiment of the frequency-shifting, time-shifting, and polarized switching apparatus block of the filter shown in FIG. **24** where the filter is configured as an LCL filter.

FIG. **32** is a functional block diagram of an embodiment of the frequency-shifting, time-shifting, and polarized switching apparatus block of the filter shown in FIG. **24** where the filter is configured as an RI filter.

FIG. **33** is a functional block diagram depicting the polarized switch portion of the frequency-shifting, time-shifting, and polarized switching apparatus shown in FIG. **24** through FIG. **32**.

FIG. **34** is a functional block diagram of an embodiment of the tapped time delay (TDL) apparatus block of the frequency-shifting, time-shifting, and polarized switching apparatus shown in FIG. **25** through FIG. **32**, where the input and output vectors can be real or complex.

FIG. **35** is a functional block diagram of a TACM adaptation apparatus in accordance with the present invention.

FIG. **36** is a functional block diagram of a TCCM adaptation apparatus in accordance with the present invention.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

Referring more specifically to the drawings, for illustrative purposes the present invention is generally shown with reference to FIG. **1** through FIG. **36** where like reference numerals denote like parts. It will be appreciated that the apparatus may vary as to configuration and as to details of the parts and that the method may vary as to the steps and their sequence without departing from the basic concepts as disclosed herein.

A. Filtering Structures and Their Theoretical Foundations

The filtering structures of the present invention exploit the 100% spectral redundancy exhibited by GMSK and other MSK signals. That is, these filtering structures exploit the fact that information conveyed in one subband of the GMSK or other MSK signal is also conveyed in another subband. In fact, for any GMSK or other MSK signal, the waveforms in any two such related subbands are nearly perfectly correlated. Thus, this spectral redundancy augments the more widely known and exploited spatial redundancy in which the waveforms due to a given source and received by two different antennas in an array are perfectly correlated.

Just as a receiver without multiple antennas and the spatial filtering structure needed to properly combine their outputs loses out on the well-known performance gains due to the exploitation of spatial redundancy, a receiver without the filter structures of the present invention loses out on the significant performance gains due to the exploitation of spectral redundancy.

By exploiting both spatial redundancy and spectral redundancy (also referred to as spectral correlation), the filtering structures of the present invention provide at least twice the level of processing gain of an LTI spatio-temporal filtering structure. For example, an M-antenna array can separate up to 2M to 4M GMSK or other MSK signals using these filtering structures, or can separate up to the more conventionally attainable number M of GMSK or other MSK signals but can do so with significantly higher reliability and quality than a conventional spatial filter. These capabilities can be explained by reference to the spectral correlation of the GMSK or other MSK signals, and by reference to the relationship between correlation (regardless of its origin or type) and linear filtering.

B. Relationship Between Correlation and Linear Filtering

Although the relationship between correlation and linear filtering is fundamental, it is not widely understood even by those skilled in the art of signal processing. Therefore, for purposes of describing the present invention we will explain this relationship by a simple, although abstract, analogy: if x(t)=a(t)+b(t), and y(t)=c(t)+d(t), where a(t) and c(t) are highly correlated, and where b(t) and d(t) are highly correlated, then an appropriate linear combination of x(t) and y(t) (i.e., the output of an appropriately chosen linear filtering structure applied jointly to x(t) and y(t)) can provide high-quality estimates of one or more of the signals a(t), b(t), c(t), and d(t).

C. Spectral Correlation of GMSK and Other MSK Signals

To understand how such performance can be possible, and to understand what filtering structures are applicable to GMSK and other MSK signals, it is necessary to understand the spectral correlation properties of GMSK and other MSK signals. For this purpose, these properties are best described by the cross-coherence function, which is a frequency-domain cross-correlation coefficient. Specifically, let x(t) denote the complex envelope of a GMSK signal having zero carrier-frequency offset and bit rate f_b, and let y(t) be a conjugated and frequency-shifted version of x(t), where the frequency-shift is either $f_b/2$ or $-f_b/2$:

$$y_1(t)=x^*(t)e^{-j2\pi(f_b/2)t} \quad (1)$$

$$y_2(t)=x^*(t)e^{+j2\pi(f_b/2)t} \quad (2)$$

with $y(t)=y_1(t)$ or $y(t)=y_2(t)$. We consider the cross-coherence function $C_{xy}(f)$ defined by

$$C_{xy}(f)=\frac{S_{xy}(f)}{\sqrt{S_{xx}(f)S_{xy}(f)}} \quad (3)$$

where $S_{xy}(f)$ denotes the cross spectral density between x(t) and y(t), and $S_{xx}(f)$ and $S_{yy}(f)$ denote the power spectral densities of x(t) and y(t), respectively. The magnitude $|C_{xy}(f)|$ versus frequency f for a GMSK signal is shown in FIG. **1**, where the solid line represents $|C_{xy_1}(f)|$ and the dashed line represents $|C_{xy_2}(f)|$, and is clearly very nearly equal to one over the entire frequency channel (−100 kHz to +100 kHz) allocated to the signal.

For any f such that $|C_{xy}(f)|$ is nearly equal to one, the narrowband components of x(t) and y(t) at frequency f fare nearly perfectly correlated. For our definitions of x(t) and y(t), this is equivalent to stating that the narrowband component of x(t) at f and the narrowband component of x*(t) at either $f+f_b/2$ or $f-f_b2$ are nearly perfectly correlated. Referring to FIG. **2**A through FIG. **2**C, the fact that $|C_{xy_1}(f)|$ in FIG. **2**C is nearly equal to unity for narrowband components numbered −6, −5, . . . ,−1, 0 implies that each pair of like-numbered narrowband components taken from x(t) in FIG. **2**A and $y_1(t)=x^*(t)e^{-j\pi f_b t}$in FIG. **2**B are nearly perfectly correlated. The same is true for the continuum of pairs of narrowband components in x(t) and y,(t) at identical values off Accordingly, an appropriate linear combination of one of these components from x(t) and the corresponding one from y,(t), can provide the same level of processing gain against other signals, interference, and noise as an appropriate linear combination of the signals received by at least twice as many antennas as used to receive x(t). Similar statements apply to linear combinations involving x(t) and $Y_2(t)=x^*(t)e^{j\pi f_b t}$. Thus, a filtering structure that uses linear combinations of x(t),y_1(t), and y_2(t) can provide the same level of processing gain against other signals, interference, and noise as an appropriate linear combination of the signals received by twice as many antennas as used to receive x(t).

Based on these facts about the spectral correlation properties of GMSK and other MSK signals, two specific interference scenarios are cited here as examples to illustrate, but not to limit, the applicability of the filtering structures of this invention to practical problems in wireless communication systems. For simplicity, only a single antenna is used in both scenarios; the capability of the proposed filtering structures increases proportionately with the number of antennas used.

Co-Channel Interference Rejection

A single antenna receives a desired GMSK signal and a co-channel GMSK signal having the same carrier frequency and comparable signal power. Neither signal can be recovered by a linear time-invariant filtering structure. However, with reference to FIG. **2**A through FIG. **2**C, each of the components numbered −6 to 0 (and the components in the continuum from −150 kHz to 0 kHz) in x(t) can be linearly combined with its corresponding component in y_1(t) to yield a clean estimate of the desired signal while rejecting the co-channel interference. Similarly, with reference to FIG. **3**A through FIG. **3**C, each of the components numbered 0 to 6 (and the components in the continuum from 0 to 150 kHz) in x(t) can be linearly combined with its corresponding component in y_2(t) to yield a clean estimate of the desired signal while rejecting the co-channel interference. Thus, by using both y_1(t) and Y_2(t), the desired signal can be recovered over the entire band from −150 kHz to +150 kHz, or over just the allocated GSM frequency channel from −100 kHz to +100 kHz if desired. In this way, the filtering structures in this invention can reject a GMSK co-channel interfering signal even when only one antenna is used.

Adjacent Channel Interference Rejection

A single antenna receives a desired GMSK signal and an adjacent channel GMSK signal possibly having much greater signal power. Specifically, let the adjacent channel signal have a carrier frequency that is 200 kHz (per GSM carrier allocation protocols) less than that of the desired signal. Due to spectral leakage, the adjacent channel interferer degrades the desired signal, which cannot be recovered by a linear time-invariant filtering structure. However, with reference to FIG. **3**A through FIG. **3**C, each of the components numbered **4** to **8** (and the components in the continuum from 100 kHz to 200 kHz) in the adjacent channel interference (now downconverted to zero carrier offset), which comprise the spectral leakage into the desired signal, can be linearly combined with its corresponding component in $y_1(t)$ to cancel out the effects of the spectral leakage. Similarly, an adjacent channel interferer whose carrier is 200 kHz above that of the desired signal can be canceled out. In this way, the filtering structures in this invention can reject two GMSK, adjacent channel interfering signals even when only one antenna is used.

Signal Distortion Correction

When a transmitted signal undergoes propagation through a non-ideal channel, such as a multipath propagation channel, the received signal can be a severely distorted version of the transmitted signal, including for example frequency-selective fading (i.e., deep spectral nulls). However, with reference to FIG. **2**A through FIG. **2**C, any of the narrowband components numbered −6, −5, . . . −1, 0 (and the components in the continuum from −150 kHz to 0 kHz) in x(t) that have been severely attenuated by frequency-selective fading (to such an extent that they cannot simply be amplified without strongly increasing the noise level), can be replaced with corresponding properly phase-adjusted and amplitude-scaled components in $y_1(t)$, thereby removing the spectral nulls. Similarly, with reference to FIG. **3**A through FIG. **3**C, spectral nulls in the band from 0 kHz to +150 kHz can be removed.

D. Specific Filtering Structures

Various filtering structures in accordance with the present invention will now be described with reference to FIG. **4** through FIG. **17**. In the filtering structures depicted therein, all blocks with only a letter inside denote standard linear time-invariant filters (ignoring time variation due only to adaptation), those blocks with a letter followed by (t) denote linear periodically time-variant filters, and those blocks with a letter and FSE denote linear time-invariant fractionally spaced equalizers (filters), all of which are known in the signal processing art. Furthermore, blocks with the symbol ( )* inside denote conjugators, blocks with "Re" inside denote real-part selectors (extractors), and blocks with "Im" inside denote imaginary-part selectors (extractors), each of which operates on a complex signal and each of which is also referred to herein as a polarized switcher. These elements are known to those having ordinary skill in the signal processing art or are as otherwise described herein.

LCL-FRESH Filter

The preceding observations about $|C_{xy}(f)|$ and the role of high or nearly perfect correlation in motivating linear filtering structures are taken into account in the 3-path time-domain linear-conjugate-linear (LCL) frequency-shift filter (FRESH filter) **10** in accordance with the present invention whose structure for GMSK and other MSK signals is depicted in block-diagram form in FIG. **4**. When applied to the signal x(t) received at a single antenna, or the other filter structure or the other filter structures described below can separate and/or correct the distortion of two co-channel GMSK or other MSK signals. The 3-path LCL-FRESH filter is appropriate for data that is sampled at a rate higher than the bit rate. Accordingly, the output of the 3-path LCL-FRESH filter must be sampled at the bit rate prior to being processed, for example, by a GMSK demodulator that operates on bit-rate sampled data to recover the bit stream.

LCL-PTV Filter

As an alternative to the 3-path LCL-FRESH filter, the effect of the two complementary frequency shifts $e^{j\pi f_b t}$and $e^{-j\pi f_b t}$and the associated filters g_{13} and g_+can be achieved also with a single time-variant filter g(t). The resulting structure is referred to herein as a linear-conjugate-linear periodically time-variant (LCL-PTV) filter **12** shown in block-diagram form in FIG. **5**. The filter g(t) has a time-variant impulse response according to the possibly periodic time-variations in the possibly cyclostationary statistics of the received signals. In some applications, it is appropriate for the coefficients of time-variant filter g(t) to vary with frequency f_b, in which case there exists a one-to-one mapping between the LCL-PTV filter and the 3-path LCL-FRESH filter. In other applications, such as those in which the time-slot boundaries of the signals do not line up sufficiently well and the length of a time-slot is a non-integer number of bit periods (e.g., as in GSM), the coefficients of filter g(t) must change every time one or more of the received signals encounters a time-slot boundary. The LCL-PTV filter is appropriate for data that is sampled at a rate higher than the bit rate. Accordingly, the output of the LCL-PTV filter must be sampled at the bit rate prior to being processed, for example, by a GMSK demodulator that operates on bit-rate sampled data to recover the bit stream.

LCL-FRESH-FSE Filter

In both the 3-path LCL-FRESH filter and the LCL-PTV filter, the sampling rate of the filter output exceeds the bit-rate (typically for a linear filter processing GMSK, it is an integer multiple of the bit-rate f_b, such as $2f_b$), in which case the filter output is typically sampled at the bit-rate prior to being processed by the demodulator (which provides an estimate of the bit stream conveyed in the desired signal). Thus, an alternative to the 3-path LCL FRESH filter is to move this bit-rate sampling operation back through the filtering structure to the extent possible without changing the capability of the filter and to eliminate redundant operations. Alternative embodiments of the resulting filter structure **14** (type-1) and **16** (type2), which are referred to herein as LCL-FRESH fractionally-spaced equalizers (LCL-FRESH-FSE), are depicted in block-diagram form in FIG. **6** and FIG. **7**, respectively. This filter differs significantly from the LCL time-invariant filter by its particular use of conjugated signal paths and by its special use of frequency shifts to properly exploit the cyclostationarity properties of the GMSK or other MSK signals. The bit-rate-sampled signal estimate $\hat{s}(k)$ must be processed by a demodulator to recover the bit stream.

LCL-FSE and RI-FSE Filters

Straightforward manipulations can be performed on the LCL-FRESH-FSE filter to obtain a filter structure in which the output of each bit-rate-sampler is frequency-shifted by j^{-k}. This shifted signal is then split into two paths: the upper path is processed by an LTI filter, and the lower path is conjugated and processed by another LTI filter. That is, the lower path following each bit-rate-sampler no longer contains a frequency shift of $(-1)^k$. However, the output final summer must be frequency-shifted by j^k. Alternative embodiments of the resulting filter structure **18** (type-1) and

20 (type2) are shown in FIG. **8** and FIG. **9**, respectively. This filter structure will be referred to herein as an LCL-FSE filter rather than an LCL-FRESH-FSE because no path is frequency-shifted any differently than any other path.

The LCL-FSE filter can be simplified without any loss of performance by observing that the two paths at the output of each bit-rate sampler and j^{-k} frequency-shifter form an LCL filter, which can be equivalently realized by modifying the upper path to operate on the real part (instead of operating on the entire complex signal) and by modifying the lower path to operate on the imaginary part (instead of operating on the conjugated complex signal). Since this structure replaces the two LCL filters by two Real-Imaginary (RI) filter pairs, the resulting structure is referred to as the RI-FSE filter and can be interpreted to be an alternate but mathematically equivalent embodiment of the LCL-FSE filter.

Alternative embodiments of this filter structure, **22** (type-1) and **24** (type-2), are shown in FIG. **10** and FIG. **11**, respectively.

LCL and RI Filters

The LCL and RI filters are related to the LCL-FSE and RI-FSE filters, but operate on bit-rate-sampled data. This is possible for GMSK, and for some other MSK signals, because the bit-rate is larger than the RF bandwidth occupied by the signal, and so the bit-rate satisfies Nyquist's sampling criterion for this signal.

The result of simplifying the LCL-FSE filter as a result of the input being sampled at the bit-rate is the LCL filter **26**, shown in FIG. **12**.

The LCL filter can be simplified without any loss of performance by observing that it can be equivalently realized by modifying the upper path to operate on the real part (instead of operating on the entire complex signal) and by modifying the lower path to operate on the imaginary part (instead of operating on the conjugated complex signal). Since this structure replaces the LCL filter by a Real-Imaginary (RI) filter pair, the resulting structure **28**, shown in FIG. **13**, is referred to as an RI filter and can be interpreted to be an alternate but mathematically equivalent embodiment of the LCL filter.

The RI filter offers the same implementation advantage over the LCL filter as the RI-FSE offers over the LCL-FSE: since the data paths being processed by various LTI or FSE filters are real, both the application of the filters to the data and the adaptation of the filters require less arithmetic than if the data paths were complex. Significantly, since the RI filter is mathematically equivalent to the LCL filter, there is no loss of quality in the estimated signal at the output.

Time-Variant Versions of LCL-FRESH, LCL-FRESH-FSE, LCL-FSE, RI-FSE, LCL, and RI

Although the 3-path LCL-FRESH, LCL-FRESH-FSE, LCL-FSE, RI-FSE, LCL, and RI filters are described herein as if their filter coefficients are fixed over one time-slot, they can in fact be made time-variant (as with the LCL-PTV filter) so as to accommodate the changes in statistical properties of the received signals when, for example, one or more of the signals encounters a time-slot boundary. However, in systems in which there is a high degree of control over the synchronization of the time-slot structures of the multiple co-channel signals, it is advantageous to force the time-slot boundaries of the multiple signals to be as close together as possible. Such desired synchronization could certainly arise if the multiple signals originate from mobile units operating in the same cell in a wireless cellular communication system; similarly, if the multiple signals originate from mobile units operating in different cells, then the desired synchronization could be made to arise through cooperation of the base stations in these different cells.

Variations for Improved Rejection of Adjacent Channel Interference

The aforementioned filtering structures are ideally suited for the rejection of co-channel interference and separation of co-channel desired signals and removal of signal distortion. However, variants of these filtering structures can provide rejection of adjacent channel interference by exploiting the spectral correlation properties of the adjacent channel interference in addition to those of the co-channel signals. Capability to reject adjacent channel interference can be especially important when the adjacent channel signals are much stronger than the desired channel signals (e.g., due to the near-far problem). The aforementioned filtering structures can be generalized (or modified) to exploit the cyclostationarity properties of the adjacent channel interference in addition to those of the desired channel signals (or, if desired, only those of the adjacent channel interference). For example, the 3-path LCL-FRESH filtering structure can be generalized to include two additional paths, one for each of the upper and lower adjacent channels that can be present. The signals present in these two additional paths are highly correlated with the interference, from the corresponding adjacent channels, that contaminates the desired signals; thus, correctly linearly combining these additional paths with the other three paths can significantly reduce the adjacent channel interference. The resulting 5-path LCL-FRESH filtering structure **30** is depicted in FIG. **14**, wherein x_{13} (t) and $x_+(t)$ are the complex envelope signals from the lower and upper adjacent channels (i.e., downconverted to complex base-band from their respective RF carriers), respectively.

Or, if only adjacent channel interference rejection is desired, such that functionality is limited to the cancellation of adjacent channel interference, instead of joint cancellation of adjacent- and co-channel interference, signal separation, and distortion correction, then the two-input two-path LCL-FRESH filtering structures **30***a*, **30***b*, which are shown in FIG. **15** (2 I-LCL-FRESH-U) and FIG. **16** (2 I-LCL,-FRESH-L), respectively, can be used to remove interference from either the upper adjacent channel by using input $x_+(t)$ and frequency shift $f_c-f_b/2$, or the lower adjacent channel by using input $x_-(t)$ and frequency shift $-f_c+f_b/2$, respectively, where f_c is the separation between adjacent carriers. As an alternative, interference from both upper and lower adjacent channels can be simultaneously removed using the 3-input 3-path LCL-FRESH filter **30***c* shown in FIG. **17** (3 I-LCL-FRESH-UL).

Based on the aforementioned descriptions of the manner in which the LCL-PTV, LCL-FRESH-FSE, LCL-FSE, RI-FSE, LCL, and RI filtering structures can be obtained from the 3-path FRESH filter, those skilled in the art of signal processing will appreciate that these filtering structures can also be generalized to improve their ability to reject adjacent channel interference. Such generalization may be accomplished by altering these filtering structures to exploit the cyclostationarity properties of the adjacent channel interference.

Finally, it is noted that combinations of the outputs of various filtering structures may be used advantageously. For example, each of two copies of the 3-path LCL-FRESH filter can be applied to each of the lower and upper adjacent channels to provide improved estimates of the adjacent channel signals; these estimates can then be appropriately frequency-shifted, adaptively filtered, and then subtracted from the received data in the desired frequency channel, thus reducing the adjacent channel interference. The signal resulting thereby can then be processed by the 3-path LCL-FRESH filter to separate co-channel signals and correct for distortion. The adjacent channel interference canceling filters can either be adapted jointly with the co-channel separation and distortion correction filters or be adapted separately.

Filter Generalization

It will be appreciated that each of the foregoing filter structures include filters (comprising time-shifters and linear combiners), frequency shifters, polarized switchers (conjugators or real- or imaginary-part selectors), and summers (linear combiners). Thus, common core elements of all of these filtering structures are time-shifters, frequency shifters, polarized switchers, and linear combiners. Table 1 below maps conjugate cycle frequencies and input and output sampling rates into frequency-shift values and polarized switch functions (e.g., conjugation or real- or imaginary-part extraction) defining the specific filter structures described herein.

than the latter. Alternatively, the 3-path LCL-FRESH filter can be replaced by either the LCL-PTV filter or the 5-path LCL-FRESH filter in either FIG. **18** or FIG. **19**, or by any of the LCL-FRESH-FSE, LCL-FSE, RI-FSE, LCL, and RI filters filter in either FIG. **18** or FIG. **19** to yield a bit-rate-sampled estimate $\hat{s}(k)$ of the desired signal waveform.

Variations: Single vs. Multiple Sensors, & Single vs. Multiple Desired Signals

By using the spectral redundancy theory of cyclostationary signals, we have proven mathematically that the best performing method for combining FRESH-filtering capability and spatial-filtering (multi-sensor linear combining) capability is to replicate the best single-sensor FRESH filter by passing the data from each sensor through its own FRESH-filter prior to linearly combining the multiple data sets. This results in a special form of periodically time-variant linear-conjugate spatio-temporal filter. For example, using the single-sensor 3-path LCL-FRESH filter with M sensors yields the 3-path M-sensor LCL-FRESH (3-path MS-LCL-FRESH) spatio-temporal filter. Therefore, it follows that the LCL-FRESH, LCL-PTV, LCL-FRESH-FSE, LCL-FSE, RI-FSE, LCL, and RI filtering structures can apply not only to single-sensor receivers (in which case x(t) is a scalar signal and the LTI or FSE filters in each signal path are single-input filters) but also to multiple-sensor receivers (in which case x(t) is a vector signal and the LTI

TABLE 1

Conjugate Cycle Frequencies, Input/Output Sampling Rates†	Filtering Structures*
$\pm f_b/2$, O/O	FIG. 4, LCL-FRESH (FIG. 5, LCL-PTV)
$\pm f_b/2$, O/B	FIG. 7, LCL-FRESH-FSE-2 (FIG. 6, 8, 9, 10, 11)
$\pm f_b/2$, B/B	FIG. 12, LCL (FIG. 13)
$2f_c - f_b/2$, O/O	FIG. 15, 2I-LCL-FRESH-U
$-2f_c + f_b/2$, O/O	FIG. 16, 2I-LCL-FRESH-L
$2f_c - f_b/2$, $-2f_c + f_b/2$, O/O	FIG. 17, 3I-LCL-FRESH-UL
$\pm f_b/2$, $-2f_c + f_b/2$, $2f_c - f_b/2$, O/O	FIG. 14, 5-path LCL-FRESH

†f_c = separation between adjacent carriers, O = oversampled, and B = bit rate sampled.
*The filter structures listed in parenthesis are mathematically equivalent to the basic structure not in parenthesis, and the former can be derived from the latter by standard block diagram manipulations.

It will further be appreciated that each of the foregoing filtering structures are baseband structures designed for complex baseband signals corresponding to a desired down-converted channel centered at zero frequency, and if the desired channel is centered at a non-zero frequency, then standard baseband-to-passband transformations can be applied to these baseband filtering structures to obtain their passband counterparts for processing passband signals.

Joint Space-Time Filtering and Factorized Filtering

As depicted in FIG. **18** and FIG. **19**, any of the aforementioned filtering structures can be used in either of two ways: (1) in a joint space-time filtering structure **32**, in which each antenna output is processed by one of these filters, and then the filtered outputs are summed, or (2) in a space-time factorized filtering structure **34**, in which a purely spatial filter is applied to the received array data, and then the spatially filtered data is processed by one of these filters. The former structure has the capability to attain higher performance at the expense of greater complexity

or FSE filters in each signal path are multiple-input filters). Similarly, the LCL-FRESH, LCL-PTV, LCL-FRESH-FSE, LCL-FSE, RI-FSE, LCL, and RI LCL filtering structures can be applied to estimate a single desired signal (in which case the LTI or FSE filters in each signal path are single-output filters; in this case, if multiple signals are desired to be estimated, then multiple copies of the filtering structure can be implemented in parallel) as well as to estimate multiple desired signals (in which case the LTI or FSE filters in each signal path are multiple-output filters). It will also be appreciated that the frequency-shifters and conjugators employed in the filters, as well as the real and imaginary part extractors, become multiple-input devices. These observations emphasize the fact that the novelty and utility of the filtering structures reside largely not in the dimensions of the signals in any particular signal path within the structure but rather within the patterns of conjugations, real-part and imaginary-part selections, frequency shifts, and sampling operations and the ways in which linear combinations of signals subjected to these transformations are formed.

Implementation of LTI and FSE Filters

It will be appreciated by those skilled in the art of signal processing that the linear-time-invariant (LTI) filters and FSE filters describing the various filtering structures of the present invention can be implemented in numerous ways. In one such implementation, an LTI filter is implemented as an FIR filter and is applied to its input by using the well-known overlap-and-add or overlap-and-save algorithms that utilize the fast Fourier transform (FFT). In another such implementation, an LTI filter is implemented using a tapped delay line, multipliers attached to each tap, and a summer. In yet other implementations, the LTI filter might be implemented using an infinite-impulse response (IIR) filter instead of an FIR filter; however, it is noted that adapting such filters can be computationally expensive and/or unreliable.

Demodulation of Recovered Signals

Finally, the interaction between the filtering structures (used for interference rejection and/or co-channel signal separation) and the demodulator (used for recovering the bit stream from the estimate of the desired signal) must be considered. Several demodulators are currently in use for GMSK signals: the maximum likelihood sequence estimator (MLSE) based on the Viterbi algorithm, the multiple-symbol differential detector, and the decision feedback equalizer (DFE). Any one of these demodulators can be used following the filtering structures to demodulate the estimated signal of interest. Alternatively, since the DFE is itself a filtering structure, it is reasonable to merge the demodulation function implemented by the conventional DFE and the interference-rejection and/or signal-separation functions implemented by the space-time filtering structure; the resulting filtering structure would have the desirable property of very low complexity relative to the receivers based on MLSE or multiple-symbol differential detection, and may be capable of attaining a higher-quality recovered bit stream due to its smaller number of adjustable coefficients for the same memory length.

Because all of the alternative filtering apparatus described herein include the capability of the feedforward part of the DFE, the result of merging the filtering apparatus with the DFE is that the filtering apparatus, be it single- or multiple-sensor, is simply followed by the feedback part of the DFE. Thus, the feedback part alone functions as the demodulator. Algorithms for adaptively adjusting this demodulator are not within the scope of this invention.

Similarly, the complexity of the MLSE demodulator can be reduced and its performance can possibly be increased by the effects of the distortion-removal capability of the FRESH filtering structures. For example, the requirements of the channel estimator that is an integral part of the MLSE demodulator can be eased by virtue of the distortion removal or even just distortion reduction.

D. Adaptation Methods

The following discussion of adaptation methods is made in the context of the GSM system, which is one type of wireless communication system to which the present invention applies. Those skilled in the art of communication systems and signal processing will appreciate that these adaptation methods can apply also to any other communication system that uses signals with embedded training sequences and constant modulus.

In GSM, each time slot lasts 577 μs (156.25 bit periods at 270.833 kbits/sec) and the bits therein are structured as shown in FIG. **20**. The three tail bits on either side of the main burst are all zeros, and the guard time lasting 8.25 bit periods prevents consecutive time slots from colliding due to different bulk propagation delays. The 116-bit message in each slot can be either digitized speech (after vocoding, convolutional coding, and interleaving) or control information. The 26-bit training sequence embedded in the middle of the slot is used in existing conventional GSM receivers, for example, to perform channel estimation prior to MLSE demodulation.

In the present invention, not only is this convenient training sequence exploited but also the very useful constant-envelope (or constant-modulus) property of GMSK and other MSK signals is exploited to adapt the filter coefficients. Four adaptation algorithms are explained, the first two of which are conventional and the second two of which are novel and unique to the present invention:

1. conventionally trained adaptive equalization;
2. constant-modulus (CM) blind adaptive equalization;
3. training-augmented constant-modulus (TACM) partially blind adaptive equalization; and
4. training-constrained constant-modulus (TCCM) partially blind adaptive equalization.

The first two adaptation algorithms provide the building blocks for the second two. All four of these algorithms are directed to finding the values of the filter coefficients in a matrix W that controls a linear combiner (the number of columns in this matrix is set to be equal to the number of desired signals to be estimated and the number of rows is set to be equal to the number of weights in the linear combiner to be adaptively adjusted). The linear combiner applies these complex weights to the elements of a vector y(t) containing various differently delayed, frequency-shifted, and possibly conjugated versions of the received data from the M-sensors (denoted by the vector x(t)) to produce a vector of signal estimates $\hat{s}(t)=W^H y(t)$, where superscript H denotes conjugate transposition. For example, in the 3-path LCL-FRESH filter, let each of the three filters be implemented by a tapped delay line (TDL) containing L taps, followed by a linear combiner. Then the signal estimate $\hat{s}(t)$ can be expressed as

$$\hat{s}(t) = W^H \begin{bmatrix} x(t) \\ x(t-1) \\ \cdot \\ \cdot \\ \cdot \\ x(t-L+1) \\ x^*(t)e^{j\pi f_b t} \\ x^*(t-1)e^{j\pi f_b t} \\ \cdot \\ \cdot \\ \cdot \\ x^*(t-L+1)e^{j\pi f_b t} \\ x^*(t)e^{-j\pi f_b t} \\ x^*(t-1)e^{-j\pi f_b t} \\ \cdot \\ \cdot \\ \cdot \\ x^*(t-L+1)e^{-j\pi f_b t} \end{bmatrix} = W^H y(t), \tag{4}$$

where y(t) is defined in the obvious way. Similar definitions of y(t) can be constructed for the LCL-FRESH-FSE, LCL-FSE, RI-FSE, LCL, and RI filtering structures, as well as for the 5-path LCL-FRESH filter and any extensions of these in which the number of paths, patterns of conjugations or real-part and imaginary-part selectors, choices of frequency shifts, and origins of input signals for the various paths are altered. Thus, in the following descriptions, no further reference to the specific dependencies of y(t) on x(t) is made, since all of the filtering structures of the present invention can be represented using this common framework. Furthermore, to simplify the descriptions only the case of W having a single column is considered; to emphasize that the matrix degenerates to a vector in this case, the set of filter coefficients is denoted by w.

Conventionally Trained Adaptive Equalization

In conventionally trained adaptive equalization, the receiver knows that a training signal $s_0(t)$ is transmitted by the desired user starting at time t_{01} and ending at time t_{02}. This knowledge and the received data are used to find w. Specifically, w is chosen so as to minimize the time-averaged squared error between the training signal $s_0(t)$ and the estimate $\hat{s}(t)$ over the training interval (i.e., for all $t\epsilon T_0$, where $T_0=[t_{01}, t_{02}]$):

$$\min_{w}\left\langle \left|\hat{s}(t)-s_0(t)\right|^2\right\rangle_{t\epsilon T_0}, \leftrightarrow w=(R_{yy}^{(T_0)})^{-1}R_{ys_0}^{(T_0)}, \tag{5}$$

where $<\bullet>_{t\epsilon T_0}$ denotes time-averaging over the period $t_{01}\leqq t\leqq t_{02}$, and the correlation matrix between any two vectors (or scalars) a(t) and b(t) is defined by $R_{ab}^{(T_0)}=<a(t)b^H(t)>_{t\epsilon T_0}$. With reference to GSM systems and FIG. **21**, T_0 is the time interval containing the embedded training signal. This method is simple to implement, with low computational complexity when implemented using the recursive least squares (RLS) algorithm or any of the various fast algorithms (e.g., the modular multichannel method or the fast affine projection algorithm), and it converges reliably and quickly when the length of the training sequence exceeds the number of filter coefficients (i.e., the length of w). However, in the presence of strong co-channel interference and/or when the length of w is comparable to or greater than the length of the training signal, this conventionally trained adaptive equalization algorithm may not provide sufficiently reliable or high quality signal estimates.

Constant Modulus (CM) Blind Adaptive Equalization

In total contrast to the conventionally trained adaptive equalizer, the constant-modulus (CM) blind adaptive equalizer ignores completely any training signal information. Instead, in this application it exploits the fact that GMSK and other MSK signals have constant modulus (constant envelope), a property which is degraded or destroyed by the presence of co-channel interference and channel distortion. The CM algorithm attempts to solve the following minimization problem:

$$\min_{w}\left\langle \left|\hat{s}(t)-\frac{\hat{s}(t)}{|\hat{s}(t)|}\right|^2\right\rangle_{t\epsilon T_1}, \tag{6}$$

where T_1 our case is the set of all time instants within a particular time slot. Since this is a nonlinear minimization problem with no closed-form solution, and a least-mean-squares (LMS) stochastic gradient descent algorithm would converge much too slowly, an iterative block approach (the so-called least-squares CM algorithm) is used. At the k^{th} iteration, the weight vector W_k is found by solving

$$\min_{w_k}\left\langle \left|w_k^H y(t)-z_{k-1}(t)\right|^2\right\rangle_{t\epsilon T_1} \tag{7}$$

where

$$z_{k-1}(t)=\frac{w_{k-1}^H y(t)}{|w_{k-1}^H y(t)|}.$$

That is, the modulus normalized term in the squared error is obtained from the filter output during the previous iteration. This allows a simple linear solution to be found:

$$W_k=(R_{yy}^{(T_1)})^{-1}R_{yz_{k-1}}^{(T_1)}. \tag{8}$$

These two steps, finding the modulus-normalized filter output and solving this linear equation, are iterated until convergence. The CM algorithm allows the equalizer to be adapted successfully in the absence of a training signal, although it is known to exhibit unreliable convergence in some situations and its reliability may be lower than desired when the observation interval is short (as it is for GSM time slots). Another complication with the CM algorithm arises when it is applied to environments containing multiple co-channel signals. Multi-target CM algorithms have been developed to allow multiple sets of filter weights to be adapted, each to extract a different signal. A subsequent signal sorting operation must then be performed to identify the correspondence between extracted signals and users; this correspondence can be achieved easily if the users are assigned different training sequences. Notice though that this training signal information might be more profitably used for adaptation of the filter weights themselves rather than simply to sort the output of the blindly adapted filters. This observation is pursued through two other adaptive algorithms, discussed below.

Training-Augmented CM (TACM) Partially Blind Adaptive Equalization

Unlike the conventionally trained adaptive equalizer and the CM blind adaptive equalizer, each of which ignores significantly useful information that is exploited by the other, the TACM partially blind adaptive equalizer exploits both is the training signal available in GSM and the constant modulus property. The motivation for these approaches is the belief that minimization of a convex linear combination of the two cost functions in Equation (5) and Equation (7) will yield better performance than minimization of either cost function alone, since the two cost functions cannot be simultaneously minimized in general. The resulting cost function is given by

$$\gamma<|w_k^H y(t)-z_{k-1}(t)|^2>_{t\epsilon T_1}+(1-\gamma)<|w_k^H y(t)-s_0(t)|^2>_{t\epsilon T_0}. \tag{9}$$

As in the conventionally training adaptive equalizer for use in GSM systems, T_0 is the time interval occupied by the embedded training signal (FIG. **20**). The time period T_1 over which the CM algorithm is applied can be either the entire time slot, or the time slot excluding the training signal. Either way, the closed-form solution is given by

$$w_k=[\gamma R_{yy}^{(T_1)}+(1-\gamma)R_{yy}^{(T_0)}]^{-1}[\gamma R_{yz_{k-1}}^{(T_1)}+(1-\gamma)R_{ys_0}^{(T_0)}]. \tag{10}$$

The convex linear combiner weight γ controls the influence of the two cost functions and thus also controls the emphasis given to the two distinct types of knowledge used in them.

Values of γ closer to unity give more influence to the exploitation of the CM property, whereas values closer to zero give more influence to the exploitation of the known training signal.

It is noted that a special case of the TACM arises when the initial condition is zero (i.e., w=0) and $\gamma=1$. In this case, the TACM yields an algorithm in which the known training signal is used to initialize the conventional CM algorithm at iteration k=1 with $w_1=[R_{yy}^{(T_0)}]^{-1}R_{ys_0}^{(T_0)}$.

In the event that $s_0(t)$ is a constant-modulus signal (e.g., as it is for GMSK, CPFSK, and other MSK signals), then $s_0(t)$ should be pre-normalized to have unity modulus prior to its use in the TACM algorithm. In the event that $s_0(t)$ is not a constant-modulus signal, then an adaptive normalization constant must be inserted into the TACM cost function above to yield the adaptively normalized TACM (AN-TACM) algorithm:

$$\gamma\langle|w_k^H y(t)-z_{k-1}(t)|^2\rangle_{t\in T_1}+(1-\gamma)\langle|w_k^H y(t)-c^* s_0(t)|^2\rangle_{t\in T_0}. \tag{11}$$

where c is a complex scalar. The solution to this problem is obtained by equating to zero the complex gradients with respect to w^*_k and c^* to obtain

$$[\gamma R_{yy}^{(T_1)}+(1-\gamma)R_{yy}^{(T_0)}]w_k-(1-\gamma)\,R_{ys_0}^{(T_0)}c=\gamma R_{yz_{k-1}}^{(T_1)} \tag{12}$$

$$R_{s_0y}^{(T_0)}w_k-R_{s_0s_0}^{(T_0)}c=0. \tag{13}$$

This system of equations is solved by

$$\begin{bmatrix} w_k \\ c \end{bmatrix}=\begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{bmatrix}^{-1}\begin{bmatrix} \gamma R_{yz_{k-1}}^{(T_1)} \\ 0 \end{bmatrix} \tag{14}$$

where $R_{11}=\gamma R_{yy}^{(T_1)}+(1-\gamma)R_{yy}^{(T_0)}$, $R_{12}=-(1-\gamma)R_{ys_0}^{(T_0)}$, $R_{21}=R_{s_0y}^{(T_0)}$, and $R_{22}=-R_{s_0s_0}^{(T_0)}$.

Training-Constrained CM (TCCM) Partially Blind Adaptive Equalization

The TCCM differs from the TACM in the way that it attempts to exploit both types of knowledge. In particular, when the number of filter coefficients is greater than or equal to the number of samples of the training signal, the linear system used by the conventionally trained adaptive equalizer (implied in Equation (5)) is under-determined. That is, the training signal is sufficient only to restrict W_k to lie in some subspace but not to define it completely. Specifically, in this case all that can be said so far is that w_k is given by

$$w_k=(R_{yy}^{(T_0)})^+R_{ys_0}^{(T_0)}+v_k=w^{(T_0)}+v_k, \tag{15}$$

where superscript $(\ )^+$ denotes the Moore-Penrose pseudoinverse, v_k is any vector in the null space of $R_{yy}^{(T_0)}$, and $w^{(T_0)}$ is defined in the obvious way. The constraint on v_k ensures that $W_k^H y(t)$ exactly reproduces the training interval signal over time interval T_0. The CM property is then used to select v_k, where k denotes as before the iteration in the block least-squares implementation of the CM algorithm. The resulting cost function is obtained simply by substituting Equation (15) into Equation (7) and minimizing with respect to v_k to obtain the solution

$$v_k=V_n(V_n^H R_{yy}^{(T_1)}V_n)^{-1}V_n^H R_{yu_{k-1}}^{(T_1)}, \tag{16}$$

where

$$u_{k-1}(t)=z_{k-1}(t)-(w^{(T_0)})^H y(t) \tag{17}$$

and the columns of V_n are the eigenvectors associated with the negligible eigenvalues of $R_{yy}^{(T_0)}$.

An alternative algorithm can be obtained by first solving for the CM weight vector according to Equation (8) and then projecting this vector onto the affine space $\{w_k=w^{(T_0)}+V_n c_k$: for all $c_k\}$. The resulting solution is

$$w_k=w^{(T_0)}+V_nV_n^H(R_{yy}^{(T_1)})^{-1}R_{yz_{k-1}}^{(T_1)} \tag{18}$$

and thus $w_k^H y(t)$ exactly reproduces the training signal over the time interval T_0. As with the TACM, if the desired signal $s_0(t)$ has constant modulus, as is the case with GMSK, CPFSK, and other MSK signals, then $s_0(t)$ should be normalized to have unity modulus. Finally, each of the two versions of the TCCM algorithm may be extended, through the adaptive normalization technique used in deriving the AN-TACM algorithm, to accommodate desired signals having non-constant modulus. These extensions are straightforward to obtain and so are not elaborated upon here.

Additional Adaptation Methods

In addition to the aforementioned TACM and TCCM methods and their various extensions and special cases, other adaptation methods can also be used to select the values of the coefficients in the filter. For example, the so-called decision-direction method of adaptation, in which a first phase of adaptation proceeds under control of some method (e.g., any of the aforementioned methods) to produce a medium-to-high quality signal estimate which can then be demodulated, and the resulting detected bits are used to form a known training signal which can be used together with the original perfect training data in a second phase of adaptation (e.g., through the conventional least-squares method).

E. Preferred Embodiment of an Overall Receiver

FIG. **22** depicts a preferred embodiment of an overall receiver apparatus **36** in accordance with the present invention. RF signals at the outputs of the M antennas **38** are coherently downconverted to complex baseband and digitized by a Coherent M-Antenna Collection Apparatus **40**, the output of which is a digitized discrete-time sequence of M×1 complex vectors, denoted elsewhere herein as x(t). This vector of received data is then provided as input to a Filter Apparatus **42**, which takes as its other input the filter coefficients (W) provided by an Adaptation Apparatus **44**, to produce a digitized discrete-time sequence of d×1 complex vector signal estimates $\hat{s}_k(n)$. During the adaptation process, a switch **46** at the output of the Filter Apparatus **42** directs the output of the Filter Apparatus **42** into the Adaptation Apparatus **44**; upon the detection of convergence by the Adaptation Apparatus **44**, such as when a prescribed number of iterations are completed, the switch **46** is moved to direct the output of the Filter Apparatus **42**, the final signal estimate $\hat{s}(n)$, to subsequent processors such as demodulators that will render bit decisions to recover the binary data streams conveyed by the desired signals. An intermediate output of the Filter Apparatus **42** (denoted by y(t) herein) is also provided as input to the Adaptation Apparatus **44**.

A preferred embodiment of the Coherent M-Antenna Collection Apparatus **40** is shown in FIG. **23**. This apparatus is controlled by user-supplied values for the RF tuning frequency and the A/D sampling clock frequency to control the local oscillators **48** used for the downconversion from real RF to complex baseband (also known as in-phase and quadrature-phase signals) and for the A/D converters **50**, respectively. The RF front-ends **52** that perform this downconversion can be implemented in a variety of ways known well to those skilled in the art of RF design. Preferably the sampling clock frequency of the A/D converters is equal to an integer multiple of the bit rate f_b. With the exception of the LCL and RI filters for which a sampling rate off f_b is needed, $2f_b$ represents a good compromise between the temporal resolution provided by oversampling and the excess of data and associated computational load induced by oversampling. The digitized discrete-time inphase and quadrature signals at the output of each pair of A/D converters are combined to form a representation of a single complex-valued digitized discrete-time signal. The M signals so formed, one per chain of hardware associated with a single antenna, are grouped into a digitized discrete-time M×1 complex vector of received data.

A preferred embodiment of the Filtering Apparatus **42** is shown in FIG. **24**. The Filtering Apparatus **42** accepts as input the M×1 complex vector of received data and the filter coefficients output from the Adaptation Apparatus **44**. The received data is first processed by a Frequency-Shifting, Time-Shifting, and Polarized Switching (i.e., conjugation, real-part selection or imaginary-part selection) Apparatus **54** whose purpose is to form the vector y(t). The record of such transformed data y(t) is stored in a Signal Buffer **56**, which can be implemented using dynamic RAM chips or other high-speed storage media, to allow y(t) to be made available to the Adaptation Apparatus **44** over multiple iterations of the adaptation algorithm implemented therein. A Matrix-Vector Multiplier **58** applies the filter coefficients to the received data, producing output $\hat{s}_k(n)$ for use in the subsequent iteration (number k+1) performed by the Adaptation Apparatus **44**.

A preferred embodiment of the Frequency-Shifting, Time-Shifting, and Polarized Switching Apparatus **54** is shown in FIG. **25** for the particular filtering structure corresponding to the 3-path LCL-FRESH filter described previously herein. It will be appreciated that the other filters previously described herein could be alternatively used. The apparatus accepts as an input the M×1 vector x(t) of received data and produces as an output the 3LM×1 vector y(t) according to Equation (4). The Tapped Delay Lines (TDL) **60** produce sets of L time-shifted versions of their respective inputs. The Polarized Switcher **62** which, in this embodiment is a conjugator, and the conjugator **62** and Frequency-Shifters **64** are depicted using standard block symbols from signal processing and communications. Similarly, the preferred embodiments of the Frequency-Shifting, Time-Shifting, and Polarized Switching Apparatus **54** for the particular filtering structures corresponding to the LCL-FRESH-FSE (2 alternate embodiments), the LCL-FSE (2 alternate embodiments), the RI-FSE (a second alternate embodiment can be obtained for the RI-FSE, similar to the LCL-FRESH-FSE and LCL-FSE), the LCL, and the RI filters are shown in FIG. **26** through FIG. **32**, respectively, in which the intermediate signal vector at the output is 4LM×1 or 2LM×1. Note that the embedded training sequences that appear in the desired signal(s) must also be bit-rate sampled by a Bit-Rate Sampler **66** and frequency-shifted by j^{-k} prior to using them in adaptation methods applied to the LCL-FSE, RI-FSE, LCL, and RI filters; it is similarly noted that this frequency shift j^{-k} must be compensated for at the filter output by a conjugate frequency shift j^{-k} if it is desired to obtain an unshifted estimate of the desired signal. However, many GMSK demodulators require their inputs to be shifted by j^{-k} anyway, and so the conjugate frequency shift j^{k} and the subsequent frequency shift j^{-k} cancel each other. It is also noted that the RI-FSE and RI filters can be more efficiently implemented due to their use of real-valued intermediate signal vectors (and consequent arithmetic simplification of filter application and adaptation), instead of the complex-valued intermediate signal vectors that are used in the LCL-FSE and LCL filters.

Referring to FIG. **33**, Polarized Switcher **62** is depicted in a functional block diagram form. It can be seen at this point that Polarized Switcher **62** represents a family of filter design parameters wherein the real **68** and imaginary **70** parts of the complex input signal are optionally selected or deselected by selectors **72** according to the conjugate cycle frequencies and input/output sampling rates shown in Table 1. Note also that the sign of the imaginary part of the input signal, if selected, is either unaffected or flipped by switch **74**. In other words, the complex input signal is either conjugated, its real part is selected, or its imaginary part is selected. It will be appreciated, however, that Polarized Switcher **62** is preferably implemented as a single function conjugator or real-part selector or imaginary-part selector rather than an apparatus that provides for the alternative selection of each of those functions.

Referring to FIG. **34**, a preferred embodiment of the Tapped Delay Line (TDL) **60** is shown. This is a standard implementation of what is known in the art. The LM×1 output vector b(t) produced thereby is defined simply for any M×1 input vector a(t) to be

$$b(t) = \begin{bmatrix} a(t) \\ a(t-1) \\ \cdot \\ \cdot \\ \cdot \\ a(t-L+1) \end{bmatrix}. \tag{19}$$

Thus, TDL **60** can accommodate either a real input or a complex input, with appropriate modification to the underlying unit-sample delay devices and signal paths.

Referring now to FIG. **35** and FIG. **36**, alternative embodiments of Adaptation Apparatus **44** are shown where TACM and TCCM are the chosen adaptation algorithms, respectively, and all of the d desired signals in the d×1 vector $s_0(t)$ have constant unit modulus. Adaptation Apparatus **44** takes as inputs the intermediate signal data y(t), provided by the signal buffer **56** in the Filtering Apparatus **42**, and the output signal $\hat{s}_k(t)$ computed by the Matrix-Vector Multiplier **58** in the Filtering Apparatus **42**. It provides as output the matrix W_k of filter coefficients, and it also controls the output switch **46** shown in FIG. **22**. Adaptation Apparatus **44** further moves the switch **46** from its lower position, in which Filtering Apparatus **42** output is directed to the Adaptation Apparatus **44**, to its upper position, in which the Filtering Apparatus **42** output is the d×1 vector $\hat{s}(t)$ of desired signal estimates, suitable for subsequent use by an appropriate demodulator. It should be understood by those skilled in the art of implementing (e.g., on programmable digital signal processing chips) that alternate embodiments of the Adaptation Apparatus **44** follow directly from the descriptions of the various versions of TACM and TCCM in this document.

As can be seen, the present invention can be used to extract a signal of interest from a plurality of spectrally overlapping communications signals, separate and remove distortion from interfering co-channel signals and suppress adjacent-channel interfering signals of the GMSK or MSK type. These signals characteristically have real and imaginary components, exhibit spectral and temporal overlap, exhibit temporal redundancy, have conjugate cycle frequencies equal to twice their carrier frequencies plus and minus one-half of their data bit rate, and exhibit conjugate spectral redundancy for spectral components having frequencies separated by their conjugate cycle frequencies. The present invention provides for time-shifting, frequency-shifting, and conjugation or extraction of real or imaginary parts of complex signals, to produce an estimate of the signal of interest. While the description above contains many specificities, these should not be construed as limiting the scope of the invention but as merely providing illustrations of some of the presently preferred embodiments of this invention. Thus the scope of this invention should be determined by the appended claims and their legal equivalents.

We claim:

1. An apparatus for extracting a signal of interest from a plurality of spectrally and temporally overlapping input signals containing digital data having a bit rate, said input signals having carrier frequencies, said input signals having conjugate cycle frequencies equal to twice their carrier frequencies plus and minus one-half of their data bit rate, said input signals exhibiting conjugate spectral redundancy for spectral components having frequencies separated by said conjugate cycle frequencies, said input signals exhibiting temporal redundancy, said apparatus comprising:

(a) time-shifting means for producing a time-shifted output signal wherein said signal of interest is time-shifted;

(b) frequency-shifting means for producing a frequency-shifted output signal wherein said signal of interest is frequency-shifted by an amount determined by its cycle frequencies; and

(c) linear combining means for weighting and summing said output signals to produce an estimate of said signal of interest.

2. An apparatus as recited in claim **1**, wherein said input signals having real and imaginary components, and further comprising polarized switching means for producing a polarized output signal wherein said real and/or imaginary components of said signal of interest are selected or deselected and wherein said real and imaginary components have signs which are changed or unchanged by said polarized switching means, wherein said polarized output signal is time-shifted, and wherein said polarized output signal is weighted and summed by said linear combining means.

3. An apparatus as recited in claim **1**, further comprising receiving means for receiving said plurality of spectrally and temporally overlapping input signals.

4. An apparatus as recited in claim **1**, further comprising demodulator means for extracting data from said estimate of said signal of interest.

5. An apparatus for extracting a signal of interest from a plurality of spectrally and temporally overlapping communications signals, said communications signals having real and imaginary components, said communications signals having carrier frequencies, said communications signals containing digital data having a bit rate, said communications signals exhibiting temporal redundancy, said communications signals having conjugate cycle frequencies equal to twice their carrier frequencies plus and minus one-half of their data bit rate, said communications signals exhibiting conjugate spectral redundancy for spectral components having frequencies separated by said conjugate cycle frequencies, said apparatus comprising:

(a) time-shifting means for producing a time-shifted output signal wherein said signal of interest is time-shifted;

(b) polarized switching means for producing a polarized output signal wherein said real and/or imaginary components of said signal of interest are selected or deselected and wherein said real and imaginary components have signs which are changed or unchanged by said polarized switching means;

(c) frequency-shifting means for producing a frequency-shifted output signal wherein said signal of interest is frequency-shifted by an amount determined by its cycle frequencies; and

(d) linear combining means for weighting and summing said output signals to produce an estimate of said signal of interest.

6. An apparatus as recited in claim **5**, further comprising receiving means for receiving said plurality of spectrally and temporally overlapping communications signals.

7. An apparatus as recited in claim **6**, further comprising demodulator means for extracting data from said estimate of said signal of interest.

8. An apparatus for extracting a signal of interest from a plurality of spectrally and temporally overlapping communications signals containing digital data having a bit rate, said communications signals having carrier frequencies, said communications signals having conjugate cycle frequencies equal to twice their carrier frequencies plus and minus one-half of their data bit rate, said communications signals exhibiting conjugate spectral redundancy for spectral components having frequencies separated by said conjugate cycle frequencies, said communications signals exhibiting temporal redundancy, said apparatus comprising:

(a) sensor means for receiving said communications signals;

(b) filter means for frequency-shifting, time-shifting and polarized switching of said signal of interest contained in said communications signals, wherein said signal of interest is frequency-shifted by an amount determined by its cycle frequencies;

(c) means for adapting said filter means; and

(d) means for producing an estimate of said signal of interest.

9. An apparatus as recited in claim **8**, wherein said sensor means comprises an antenna and a radio frequency receiver.

10. An apparatus as recited in claim **9**, further comprising signal buffering means for buffering an output signal from said filter means and producing and intermediate signal as an input to said adapter means.

11. An apparatus as recited in claim **10**, further comprising matrix-vector multiplier means, coupled to said signal buffer means and said adapter means, for weighting and linearly combining signals from said filter means.

12. An apparatus as recited in claim **11**, further comprising demodulator means for extracting digital data contained in said estimate of said signal of interest.

* * * * *

EXHIBIT H

US007383453B2

(12) United States Patent
Youngs

(10) **Patent No.: US 7,383,453 B2**

(45) **Date of Patent: *Jun. 3, 2008**

(54) **CONSERVING POWER BY REDUCING VOLTAGE SUPPLIED TO AN INSTRUCTION-PROCESSING PORTION OF A PROCESSOR**

(75) Inventor: **Lynn R. Youngs**, Cupertino, CA (US)

(73) Assignee: **Apple, Inc**, Cupertino, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 55 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **11/213,215**

(22) Filed: **Aug. 25, 2005**

(65) **Prior Publication Data**

US 2005/0283628 A1 Dec. 22, 2005

Related U.S. Application Data

(63) Continuation of application No. 11/103,911, filed on Apr. 11, 2005, now Pat. No. 6,973,585, which is a continuation of application No. 10/135,116, filed on Apr. 29, 2002, now Pat. No. 6,920,574.

(51) **Int. Cl.**
G06F 1/00 (2006.01)
G06F 1/26 (2006.01)

(52) **U.S. Cl.** **713/300**; 713/320; 713/324

(58) **Field of Classification Search** 713/300
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,666,537	A *	9/1997	Debnath et al.	713/322
6,347,379	B1 *	2/2002	Dai et al.	713/320
6,721,892	B1 *	4/2004	Osborn et al.	713/300
6,792,551	B2 *	9/2004	Dai	713/320
6,795,896	B1 *	9/2004	Hart et al.	711/118
2003/0056127	A1 *	3/2003	Vaglica	713/300
2003/0120959	A1	6/2003	Bohrer et al.	713/320
2003/0120962	A1	6/2003	Dai et al.	713/320

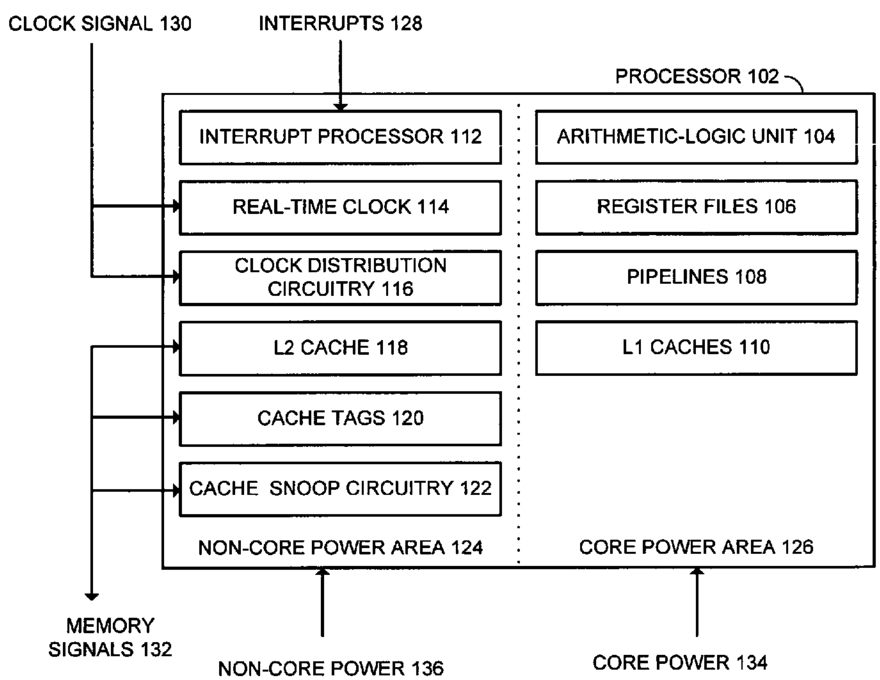
* cited by examiner

Primary Examiner—Rehana Perveen
Assistant Examiner—Stefan Stoynov
(74) *Attorney, Agent, or Firm*—Park, Vaughan & Fleming, LLP; Edward J. Grundler

(57) **ABSTRACT**

One embodiment of the present invention provides a system that facilitates reducing static power consumption of a processor. During operation, the system receives a signal indicating that instruction execution within the processor is to be temporarily halted. In response to this signal, the system halts an instruction-processing portion of the processor, and reduces the voltage supplied to the instruction-processing portion of the processor. Full voltage is maintained to a remaining portion of the processor, so that the remaining portion of the processor can continue to operate while the instruction-processing portion of the processor is in reduced power mode.

21 Claims, 3 Drawing Sheets



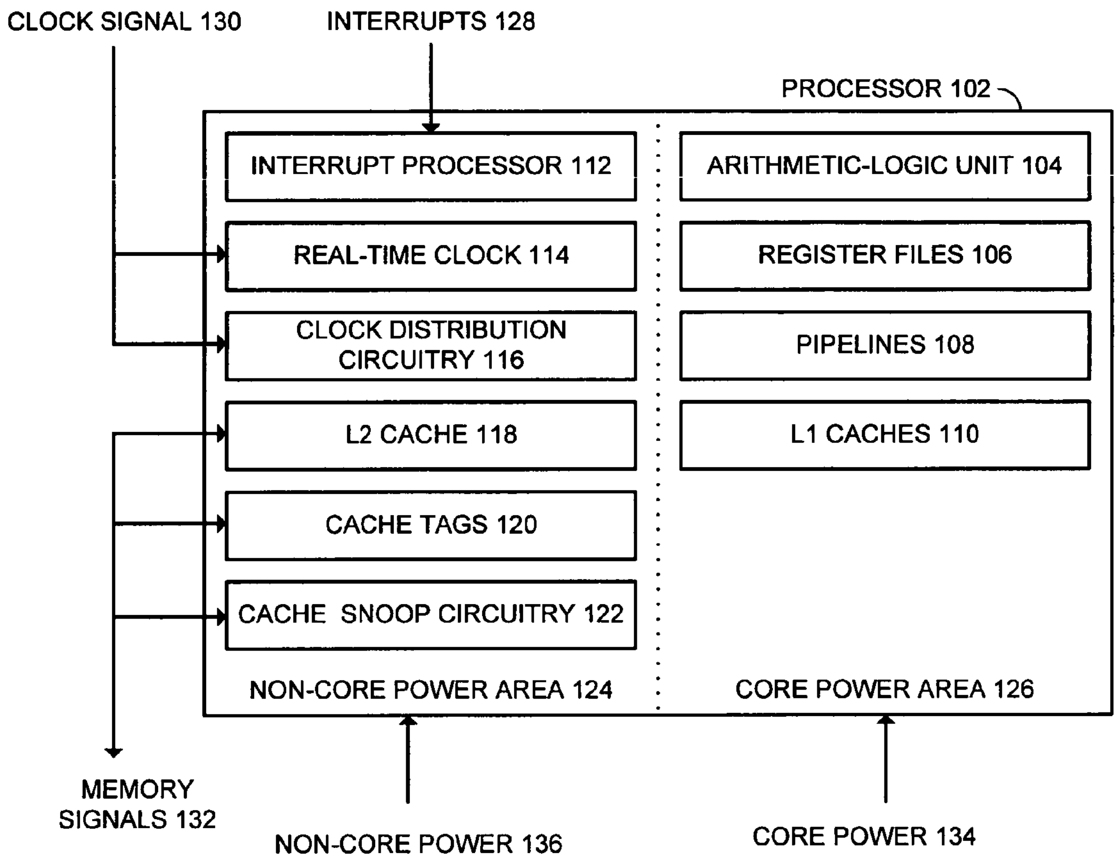

CLOCK SIGNAL 130
INTERRUPTS 128
PROCESSOR 102
INTERRUPT PROCESSOR 112
ARITHMETIC-LOGIC UNIT 104
REAL-TIME CLOCK 114
REGISTER FILES 106
CLOCK DISTRIBUTION CIRCUITRY 116
PIPELINES 108
L2 CACHE 118
L1 CACHES 110
CACHE TAGS 120
CACHE SNOOP CIRCUITRY 122
NON-CORE POWER AREA 124
CORE POWER AREA 126
MEMORY SIGNALS 132
NON-CORE POWER 136
CORE POWER 134

FIG. 1A

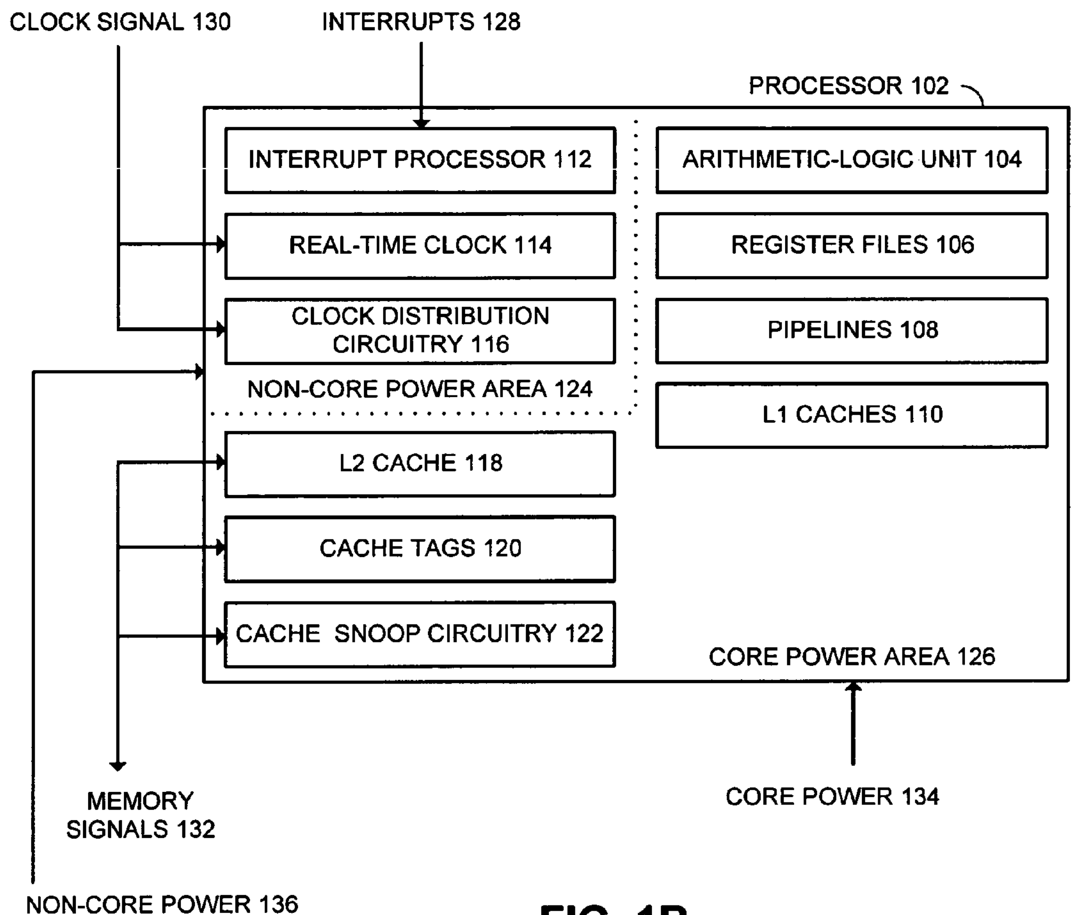

CLOCK SIGNAL 130
INTERRUPTS 128
PROCESSOR 102
INTERRUPT PROCESSOR 112
ARITHMETIC-LOGIC UNIT 104
REAL-TIME CLOCK 114
REGISTER FILES 106
CLOCK DISTRIBUTION CIRCUITRY 116
PIPELINES 108
NON-CORE POWER AREA 124
L1 CACHES 110
L2 CACHE 118
CACHE TAGS 120
CACHE SNOOP CIRCUITRY 122
CORE POWER AREA 126
MEMORY SIGNALS 132
CORE POWER 134
NON-CORE POWER 136

FIG. 1B

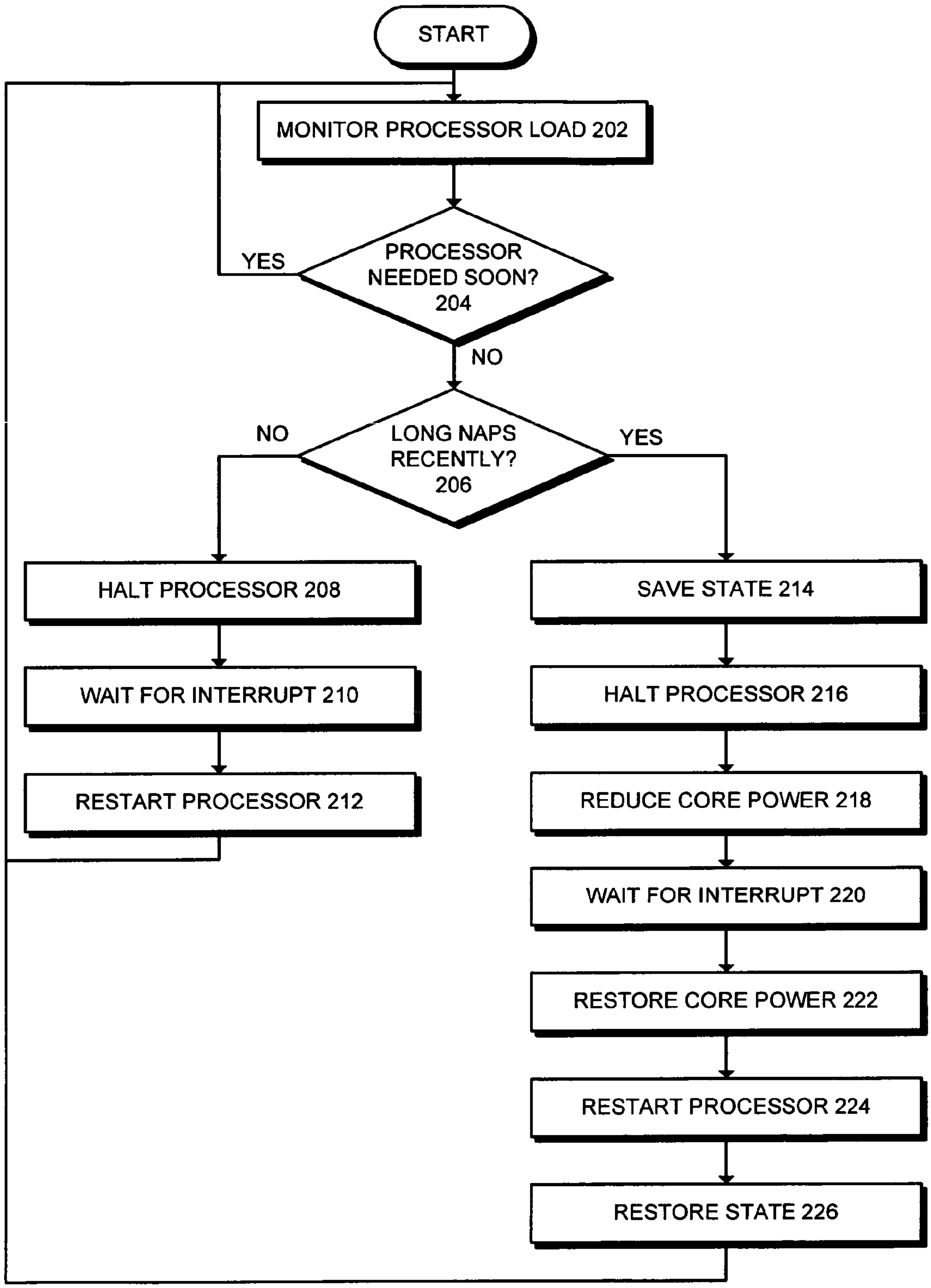

START
MONITOR PROCESSOR LOAD 202
PROCESSOR NEEDED SOON? 204
YES
NO
LONG NAPS RECENTLY? 206
NO
YES
HALT PROCESSOR 208
WAIT FOR INTERRUPT 210
RESTART PROCESSOR 212
SAVE STATE 214
HALT PROCESSOR 216
REDUCE CORE POWER 218
WAIT FOR INTERRUPT 220
RESTORE CORE POWER 222
RESTART PROCESSOR 224
RESTORE STATE 226

FIG. 2

CONSERVING POWER BY REDUCING VOLTAGE SUPPLIED TO AN INSTRUCTION-PROCESSING PORTION OF A PROCESSOR

RELATED APPLICATION

This application is a continuation of U.S. patent application Ser. No. 11/103,911, filed 11 Apr. 2005 now U.S. Pat No. 6,973,585. This application hereby claims priority under 35 U.S.C. §120 to the above-listed application. Note that pending U.S. patent application Ser. No. 11/103,911 is itself a continuation of U.S. patent application Ser. No. 10/135,116, filed 29 Apr. 2002 now U.S. Pat. No. 6,920,574.

BACKGROUND

1. Field of the Invention

The present invention relates to techniques for conserving power usage in computer systems. More specifically, the present invention relates to a method and an apparatus for reducing power consumption in a processor by reducing voltage supplied to an instruction-processing portion of the processor, while maintaining voltage to other portions of the processor.

2. Related Art

Dramatic advances in integrated circuit technology have led to corresponding increases in processor clock speeds. Unfortunately, these increases in processor clock speeds have been accompanied by increased power consumption. Increased power consumption is undesirable, particularly in battery-operated devices such as laptop computers, for which there exists a limited supply of power. Any increase in power consumption decreases the battery life of the computing device.

Modern processors are typically fabricated using Complementary Metal Oxide Semiconductor (CMOS) circuits. CMOS circuits typically consume more power while the circuits are switching, and less power while the circuits are idle. Designers have taken advantage of this fact by reducing the frequency of (or halting) clock signals to certain portions of a processor when the processor is idle. Note that some portions of the processor must remain active, however. For example, a cache memory with its associated snoop circuitry will typically remain active as well as interrupt circuitry and real-time clock circuitry.

Although reducing the frequency of (or halting) a system clock signal can reduce the dynamic power consumption of a processor, static power consumption is not significantly affected. This static power consumption is primarily caused by leakage currents through the CMOS devices. As integration densities of integrated circuits continue to increase, circuit devices are becoming progressively smaller. This tends to increase leakage currents, and thereby increases static power consumption. This increased static power consumption results in reduced battery life, and increases cooling system requirements for battery operated computing devices.

What is needed is a method and an apparatus that reduces static power consumption for a processor in a battery operated computing device.

SUMMARY

One embodiment of the present invention provides a system that facilitates reducing static power consumption of a processor. During operation, the system receives a signal indicating that instruction execution within the processor is to be temporarily halted. In response to this signal, the system halts an instruction-processing portion of the processor, and reduces the voltage supplied to the instruction-processing portion of the processor. Full voltage is maintained to a remaining portion of the processor, so that the remaining portion of the processor can continue to operate while the instruction-processing portion of the processor is in reduced power mode.

In one embodiment of the present invention, reducing the voltage supplied to the instruction-processing portion of the processor involves reducing the voltage to a minimum value that maintains state information within the instruction-processing portion of the processor.

In one embodiment of the present invention, reducing the voltage supplied to the instruction-processing portion of the processor involves reducing the voltage to zero.

In one embodiment of the present invention, the system saves state information from the instruction-processing portion of the processor prior to reducing the voltage supplied to the instruction-processing portion of the processor. This state information can either be saved in the remaining portion of the processor or to the main memory of the computer system.

In one embodiment of the present invention, upon receiving a wakeup signal, the system: restores full voltage to the instruction-processing portion of the processor; restores state information to the instruction-processing portion of the processor; and resumes processing of computer instructions.

In one embodiment of the present invention, maintaining full voltage to the remaining portion of the processor involves maintaining full voltage to a snoop-logic portion of the processor, so that the processor can continue to perform cache snooping operations while the instruction-processing portion of the processor is in the reduced power mode.

In one embodiment of the present invention, the system also reduces the voltage to a cache memory portion of the processor. In this embodiment, the system writes cache memory data to main memory prior to reducing the voltage.

In one embodiment of the present invention, the remaining portion of the processor includes a control portion of the processor containing interrupt circuitry and clock circuitry.

In one embodiment of the present invention, the remaining portion of the processor includes a cache memory portion of the processor.

BRIEF DESCRIPTION OF THE FIGURES

FIG. **1**A illustrates different power areas within processor **102** in accordance with an embodiment of the present invention.

FIG. **1**B illustrates alternate power areas within processor **102** in accordance with an embodiment of the present invention.

FIG. **2** is a flowchart illustrating the process of monitoring processor load and switching to power saving modes in accordance with an embodiment of the present invention.

DETAILED DESCRIPTION

The following description is presented to enable any person skilled in the art to make and use the invention, and is provided in the context of a particular application and its requirements. Various modifications to the disclosed embodiments will be readily apparent to those skilled in the art, and the general principles defined herein may be applied to other embodiments and applications without departing from the spirit and scope of the present invention. Thus, the present invention is not intended to be limited to the embodiments shown, but is to be accorded the widest scope consistent with the principles and features disclosed herein.

Processor **102**

FIG. **1**A illustrates different power areas within processor **102** in accordance with an embodiment of the present invention. Processor **102** is divided into a core power area **126**, and a non-core power area **124**. Core power area **126** includes the instruction-processing portion of processor **102**. Specifically, core power area **126** includes arithmetic-logic unit **104**, register files **106**, pipelines **108**, and possibly level one (L**1**) caches **110**. Note that L**1** caches **110** can alternatively be located in non-core power area **124**.

Arithmetic-logic unit **104** provides computational and logical operations for processor **102**. Register files **106** provide source operands, intermediate storage, and destination locations for instructions being executed by arithmetic-logic unit **104**. Pipelines **108** provides a steady stream of instructions to arithmetic-logic unit **104**. Instructions in pipelines **108** are decoded in transit. Therefore, pipelines **108** may contain instructions in various stages of decoding and execution. L**1** caches **110** include data caches and instruction caches for arithmetic-logic unit **104**. L**1** caches **110** are comprised of very high-speed memory to provide fast access for instructions and data. In one embodiment of the present invention, L**1** caches **110** includes a write-through data cache.

Non-core power area **124** comprises the remaining portion of processor **102** and includes interrupt processor **112**, real-time clock **114**, clock distribution circuitry **116**, level two (L**2**) caches **118**, cache tags **120**, and cache snoop circuitry **122**. In general, non-core power area **124** includes portions of processor **102** that are not directly involved in processing instructions, and that need to operate while instruction processing is halted.

Interrupt processor **112** monitors interrupts **128** and periodically interrupts the execution of applications to provide services to external devices requiring immediate attention. Interrupt processor **112** can also provide a wake-up signal to core power area **126** as described below. Real-time clock **114** provides time-of-day services to processor **102**. Typically, real-time clock **114** is set upon startup from a battery operated real-time clock in the computer and thereafter provides time to the system. Clock distribution circuitry **116** provides clock signals for processor **102**. Distribution of these clock signals can be switched off or reduced for various parts of processor **102**. For example, clock distribution to core power area **126** can be stopped while the clock signals to non-core power area **124** continue. The acts of starting and stopping of these clock signals are known in the art and will not be described further. Real-time clock **114** and clock distribution circuitry **116** receive clock signal **130** from the computer system. Clock signal **130** is the master clock signal for the system.

L**2** cache **118** provides a second level cache for processor **102**. Typically, an L**2** cache is larger and slower that an L**1** cache, but still provides faster access to instructions and data than can be provided by main memory. Cache tags **120** provide an index into data stored in L**2** cache **118**. Cache snoop circuitry **122** invalidates cache lines base primarily on other processors accessing their own cache lines, or I/O devices doing memory transfers, even when instruction processing has been halted. L**2** cache **118**, cache tags **120**, and cache snoop circuitry **122** communicate with the computer system through memory signals **132**.

Non-core power area **124** receives non-core power **136** and core power area **126** receives core power **134**. The voltage applied for non-core power **136** remains at a voltage that allows circuitry within non-core power area **124** to remain fully active at all times. In contrast, non-core power **136** may provide different voltages to non-core power area **124** based upon the operating mode of processor **102**. For example, if processor **102** is a laptop attached to external electrical power, the voltage provided to non-core power **136** (and to core power **134** during instruction processing) may be higher than the minimum voltage, thus providing faster execution of programs.

The voltage applied to core power **134** remains sufficiently high during instruction processing so that core power area **126** remains fully active. However, when processor **102** receives a signal that processing can be suspended, the voltage supplied by core power **134** can be reduced.

In one embodiment of the present invention, the voltage in core power **134** is reduced to the minimum value that will maintain state information within core power area **126**, but this voltage is not sufficient to allow processing to continue. In another embodiment of the present invention, the voltage at core power **134** is reduced to zero. In this embodiment, the state of core power area **126** is first saved before the voltage is reduced to zero. This state can be saved in a dedicated portion of L**2** cache **118**, in main memory, or in another dedicated storage area. Upon receiving an interrupt or other signal indicating that processing is to resume, the voltage in core power **134** is restored to a normal level, saved state is restored, and processing is restarted.

FIG. **1**B illustrates an alternative partitioning of power areas within processor **102** in accordance with an embodiment of the present invention. As shown in FIG. **1**B, L**2** cache **118**, cache tags **120**, and cache snoop circuitry **122** are included in core power area **126** rather than in non-core power area **124**. In this embodiment, the voltage supplied as core power **134** is reduced or set to zero as described above, however, the cache circuitry within processor **102** is also put into the reduced power mode. Prior to reducing the voltage supplied to core power area **126**, data stored in L**2** cache **118** is flushed to main memory. Additionally, if the voltage at core power **134** is reduced to zero, the state of processor **102** is first saved in main memory.

Monitoring and Switching

FIG. **2** is a flowchart illustrating the process of monitoring processor load and switching to power saving modes in accordance with an embodiment of the present invention. The system starts by monitoring the processor load (step **202**). Next, the system determines if the processor will be needed soon (step **204**). This determination is made based on the current execution pattern and the cost of entering and recovering from nap mode. This cost, calculated in power usage, must be less than the power wasted by not going into nap mode. If the processor will be needed soon at step **204**, the process returns to step **202** to continue monitoring the processor load.

If the processor will not be needed soon at step **204**, the system determines if the processor has been taking long naps recently (step **206**). If not, the system enters a normal nap mode, which involves halting the processor without reducing any voltages (step **208**). Typically, halting the processor involves removing the clock signals to the core power area of the processor. After halting the processor, the system waits for an interrupt (step **210**). Upon receiving an interrupt or other signal requiring a restart, the system restarts instruction processing (step **212**). After restarting instruction processing, the process returns to step **202** to continue monitoring the processor load.

If the processor has recently been taking long naps at step **206**, the system enters a deep nap mode, which involves saving the state information from the core power area (step **214**), halting the processor (step **216**), and then reducing the voltage supplied to the core power area (step **218**). After reducing the voltage, the system waits for an interrupt (step **220**).

Upon receiving the interrupt or other signal requiring a restart, the system restores the voltage to the core power area (step **222**). Next, the modules within the core power area are restarted (step **224**). The system then restores the state information that was saved at step **214** (step **226**). After the processor has been restarted, the process returns to step **202** to continue monitoring the processor load. Note that the above description applies when the processor is used to save and restore the state information. In cases where dedicated hardware saves and restores the state information, steps **214** and **216**, and steps **224** and **226** can be reversed. Note also that if the voltage supplied to the core power area **126** is reduced but maintained at a level where modules in the core power do not lose state information, steps **216** and **224** are not required.

The foregoing descriptions of embodiments of the present invention have been presented for purposes of illustration and description only. They are not intended to be exhaustive or to limit the present invention to the forms disclosed. Accordingly, many modifications and variations will be apparent to practitioners skilled in the art. Additionally, the above disclosure is not intended to limit the present invention. The scope of the present invention is defined by the appended claims.

What is claimed is:

1. An instruction-processing system with minimal static power leakage, the instruction-processing system comprising:
- a core with instruction-processing circuitry;
- an area coupled to the core;
- a core voltage provided to the core; and
- an area voltage provided to the area;
- wherein in a normal operation mode:
  - a clock signal to the core is active;
  - the core voltage is a first value;
  - the core is active;
  - the area voltage is a second value; and
  - the area is active;
- wherein in a first power-saving mode that is exited upon receipt of an interrupt signal:
  - the clock signal to the core is inactive;
  - the core voltage is equal to or greater than the first value; and
  - the area voltage is equal to or greater than the second value;
- wherein in a second power-saving mode that can be exited upon receipt of a signal that is not an interrupt signal:
  - the clock signal to the core is inactive;
  - the core voltage is less than the first value; and
  - the area voltage is equal to or greater than the second value.

2. The instruction-processing system of claim **1**, wherein the first power-saving mode can be exited upon receipt of a signal that is not an interrupt signal.

3. The instruction-processing system of claim **1**, wherein the area comprises a cache.

4. The instruction-processing system of claim **3**, wherein the area further comprises cache tags.

5. The instruction-processing system of claim **1**, wherein prior to entering the second power-saving mode, the state of the core is saved to a memory.

6. The instruction-processing system of claim **1**, wherein upon exiting the second power-saving mode, the state of the core is restored.

7. The instruction-processing system of claim **1**, wherein in the second power-saving mode, the core voltage is at zero.

8. A method for minimizing static power leakage in an instruction-processing system, wherein the instruction-processing system comprises a core with instruction-processing circuitry, an area coupled to the core, a core voltage provided to the core, and an area voltage provided to the area, the method comprising:
- entering a normal operation mode by:
  - providing a clock signal to the core;
  - providing the core with a core voltage that is equal to a first value;
  - providing the area with an area voltage that is equal to a second value;
- entering a first power-saving mode by:
  - disabling the clock signal to the core;
  - providing the core with a core voltage that is equal to or greater than the first value; and
  - providing the area with an area voltage that is equal to or greater than the second value;
- exiting the first power-saving mode upon receipt of an interrupt signal;
- entering a second power-saving mode by:
  - disabling the clock signal to the core;
  - setting the core voltage to a value less than the first value; and
  - providing the area with an area voltage that is equal to or greater than the second value; and
- exiting the second power-saving mode upon receipt of a signal that is not an interrupt signal.

9. The method of claim **8**, further comprising exiting the first power-saving mode upon receipt of a signal that is not an interrupt signal.

10. The instruction-processing system of claim **8**, wherein the area comprises a cache.

11. The method of claim **10**, wherein the area further comprises cache tags.

12. The method of claim **8**, further comprising saving the state of the core to a memory prior to entering the second power-saving mode.

13. The method of claim **8**, further comprising restoring the state of the core upon exiting the second power-saving mode.

14. The method of claim **8**, wherein in the second power-saving mode, setting the core voltage to the value less than the first value comprises setting the core voltage to zero.

15. A computer-readable medium containing data representing an instruction-processing system with minimal static power leakage, the instruction- processing system comprising:
- a core with instruction-processing circuitry;
- an area coupled to the core;
- a core voltage provided to the core; and
- an area voltage provided to the area;
- wherein in a normal operation mode:
  - a clock signal to the core is active;
  - the core voltage is a first value;
  - the core is active;

the area voltage is a second value; and

the area is active;

wherein in a first power-saving mode that is exited upon receipt of an interrupt signal:

the clock signal to the core is inactive;

the core voltage is equal to or greater than the first value; and

the area voltage is equal to or greater than the second value;

wherein in a second power-saving mode that can be exited upon receipt of a signal that is not an interrupt signal:

the clock signal to the core is inactive;

the core voltage is less than the first value; and

the area voltage is equal to or greater than the second value.

16. The computer-readable medium of claim **15**, wherein the first power-saving mode can be exited upon receipt of a signal that is not an interrupt signal.

17. The computer-readable medium of claim **15**, wherein the area comprises a cache.

18. The computer-readable medium of claim **17**, wherein the area further comprises cache tags.

19. The computer-readable medium of claim **15**, wherein prior to entering the second power-saving mode, the state of the core is saved to a memory.

20. The computer-readable medium of claim **15**, wherein upon exiting the second power-saving mode, the state of the core is restored.

21. The computer-readable medium stem of claim **15**, wherein in the second power-saving mode, the core voltage is at zero.

* * * * *

EXHIBIT I

US005455599A

United States Patent [19]

Cabral et al.

[11] **Patent Number: 5,455,599**

[45] **Date of Patent: Oct. 3, 1995**

[54] **OBJECT-ORIENTED GRAPHIC SYSTEM**

[75] Inventors: **Arthur W. Cabral**; **Rajiv Jain**, both of Sunnyvale; **Maire L. Howard**, San Jose; **John Peterson**, Menlo Park; **Richard D. Webb**, Sunnyvale; **Robert Seidl**, Palo Alto, all of Calif.

[73] Assignee: **Taligent Inc.**, Cupertino, Calif.

[21] Appl. No.: **416,949**

[22] Filed: **Apr. 4, 1995**

Related U.S. Application Data

[63] Continuation of Ser. No. 145,840, Nov. 2, 1993, abandoned.

[51] **Int. Cl.**[6] **G09G 5/00**

[52] **U.S. Cl.** **345/133**; 395/118

[58] **Field of Search** 345/112, 132, 345/133, 153, 154, 155; 395/118, 275

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,821,220	4/1989	Duisberg	364/578
4,885,717	12/1989	Beck et al.	364/900
4,891,630	1/1990	Friedman et al.	340/706
4,953,080	8/1990	Dysart et al.	364/200
5,041,992	8/1991	Cunningham et al.	364/518
5,050,090	9/1991	Golub et al.	364/478
5,060,276	10/1991	Morris et al.	382/8
5,075,848	12/1992	Lai et al.	395/425
5,093,914	3/1992	Coplien et al.	395/700
5,119,475	6/1992	Smith et al.	395/156
5,125,091	6/1992	Staas, Jr. et al.	395/650
5,133,075	7/1992	Risch	395/800
5,136,705	8/1992	Stubbs et al.	395/575
5,151,987	9/1992	Abraham et al.	395/575
5,181,162	1/1993	Smith et al.	364/419
5,241,625	8/1993	Epard et al.	395/163
5,265,206	11/1993	Shackelford et al.	395/200
5,297,279	3/1994	Bannon et al.	395/600

FOREIGN PATENT DOCUMENTS

0459683	12/1991	European Pat. Off. .
0603095	6/1994	European Pat. Off. .
91/20032	12/1991	WIPO .

OTHER PUBLICATIONS

"Object Oriented Approach to Design of Interactive Intelligent Instrumentation User Interface", Nikola Bogunovic, Automatika vol. 34, No. 3–4, May–Dec. 1993, pp. 143–146.

"Object–oriented versus bit–mapped graphics interfaces: performance and preference differences for typical applications", Michael Mohageg, Behaviour & Inforamtion Technology, vol. 10, No. 2, Mar.–Apr. 1991 pp. 121–147.

"Porting Apple© Macintosh© Applications to the Microsoft© Windows Environment", Schulman et al., Microsoft System Journal, vol. 4, No. 1, Jan. 1989, pp. 11–40.

Computer, vol. 22(10), Dec. 1989, Long Beach, US, pp. 43–54, Goodman "Knowledge–Based Computer Vision".

Software–Practice and Experience, vol. 19(10), Oct. 1989, Chicester UK, pp. 979–1013, Dietrich, "TGMS: An Object–Oriented System for Programming Geometry".

Proceedings of the SPIE, vol. 1659, Feb. 12, 1992, US, pp. 159–167, Haralick et al. "The Image Understanding Environment".

Intelligent CAD Oct. 6, 1987, NL, pp. 159–168, Woodbury et al., "An Approach to Geometric Reasoning".

Computer, vol. 22(10), Dec. 1989.

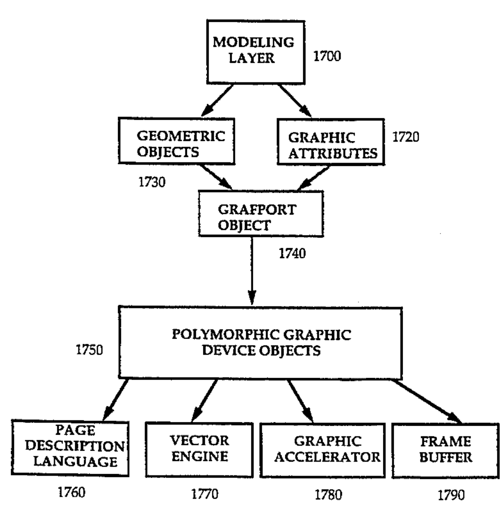
Primary Examiner—Jeffery Brier

Attorney, Agent, or Firm—Keith Stephens

[57] **ABSTRACT**

An object-oriented graphic system is disclosed including a processor with an attached display, storage and object-oriented operating system. The graphic system builds a component object in the storage of the processor for managing graphic processing. The processor includes an object for connecting one or more graphic devices to various objects responsible for tasks such as graphic accelerators, frame buffers, page description languages and vector engines. The system is fully extensible and includes polymorphic processing built into each of the support objects.

26 Claims, 16 Drawing Sheets



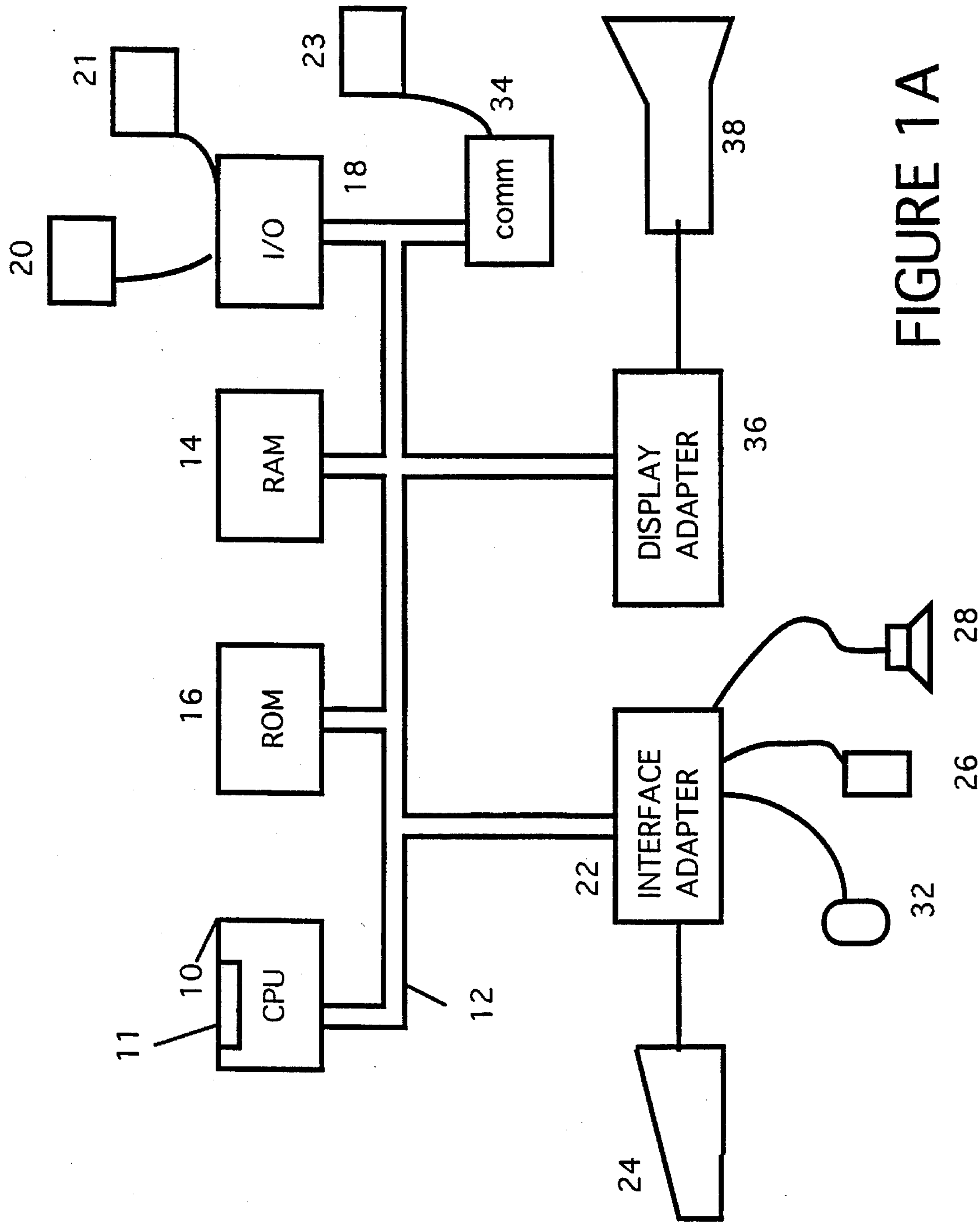

CPU
ROM
RAM
I/O
comm
DISPLAY ADAPTER
INTERFACE ADAPTER
10
11
12
14
16
18
20
21
22
23
24
26
28
32
34
36
38

FIGURE 1A

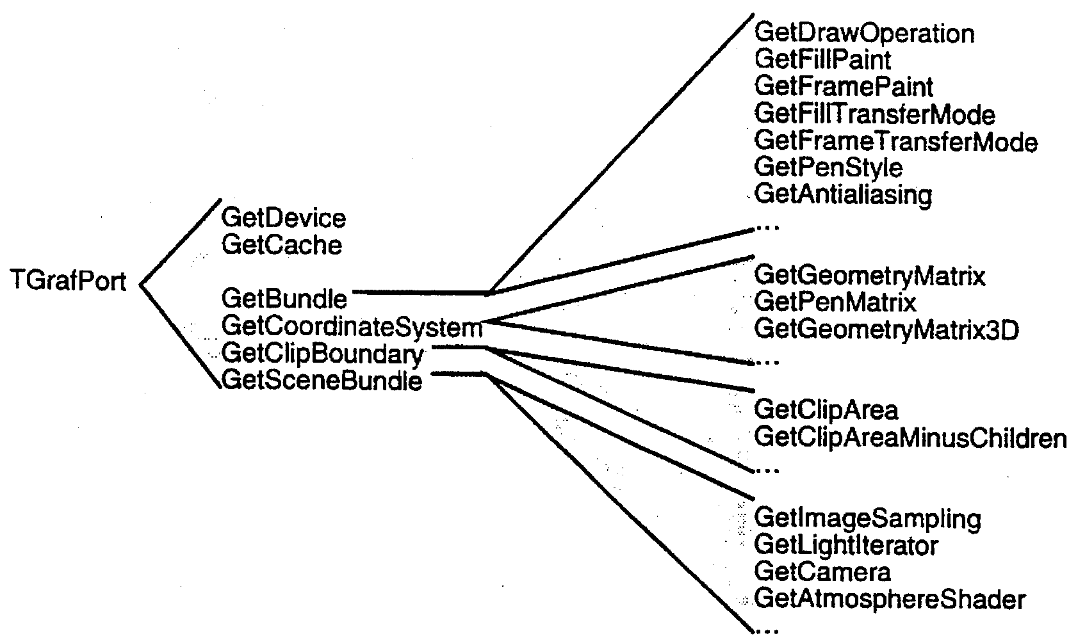

TGrafPort
GetDevice
GetCache
GetBundle
GetCoordinateSystem
GetClipBoundary
GetSceneBundle
GetDrawOperation
GetFillPaint
GetFramePaint
GetFillTransferMode
GetFrameTransferMode
GetPenStyle
GetAntialiasing
...
GetGeometryMatrix
GetPenMatrix
GetGeometryMatrix3D
...
GetClipArea
GetClipAreaMinusChildren
...
GetImageSampling
GetLightIterator
GetCamera
GetAtmosphereShader
...

FIGURE 1B

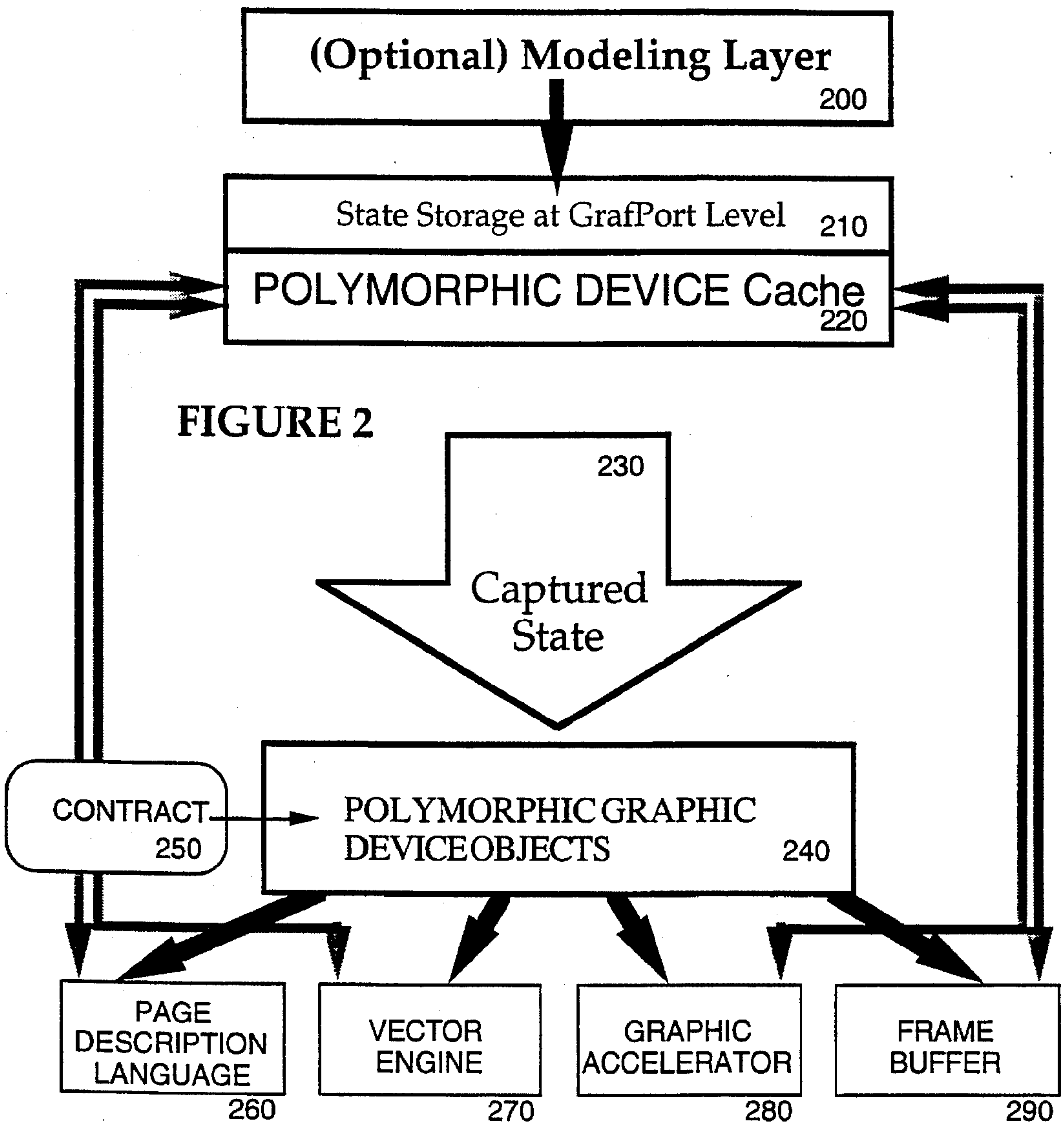

(Optional) Modeling Layer
200
State Storage at GrafPort Level
210
POLYMORPHIC DEVICE Cache
220
230
Captured State
CONTRACT
250
POLYMORPHIC GRAPHIC DEVICE OBJECTS
240
PAGE DESCRIPTION LANGUAGE
260
VECTOR ENGINE
270
GRAPHIC ACCELERATOR
280
FRAME BUFFER
290

FIGURE 2

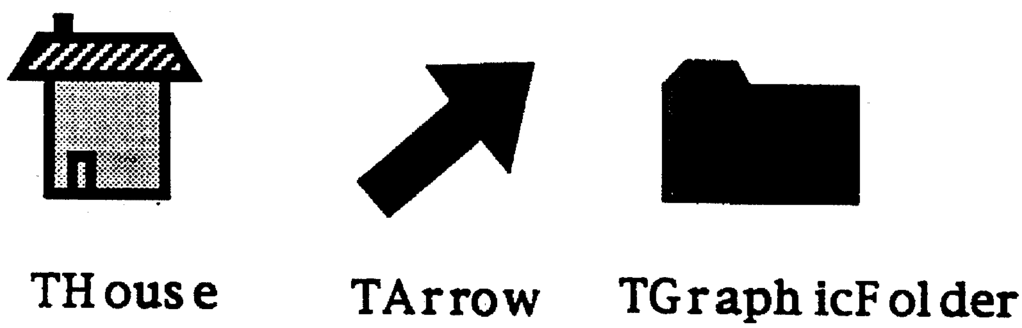


FIGURE 3

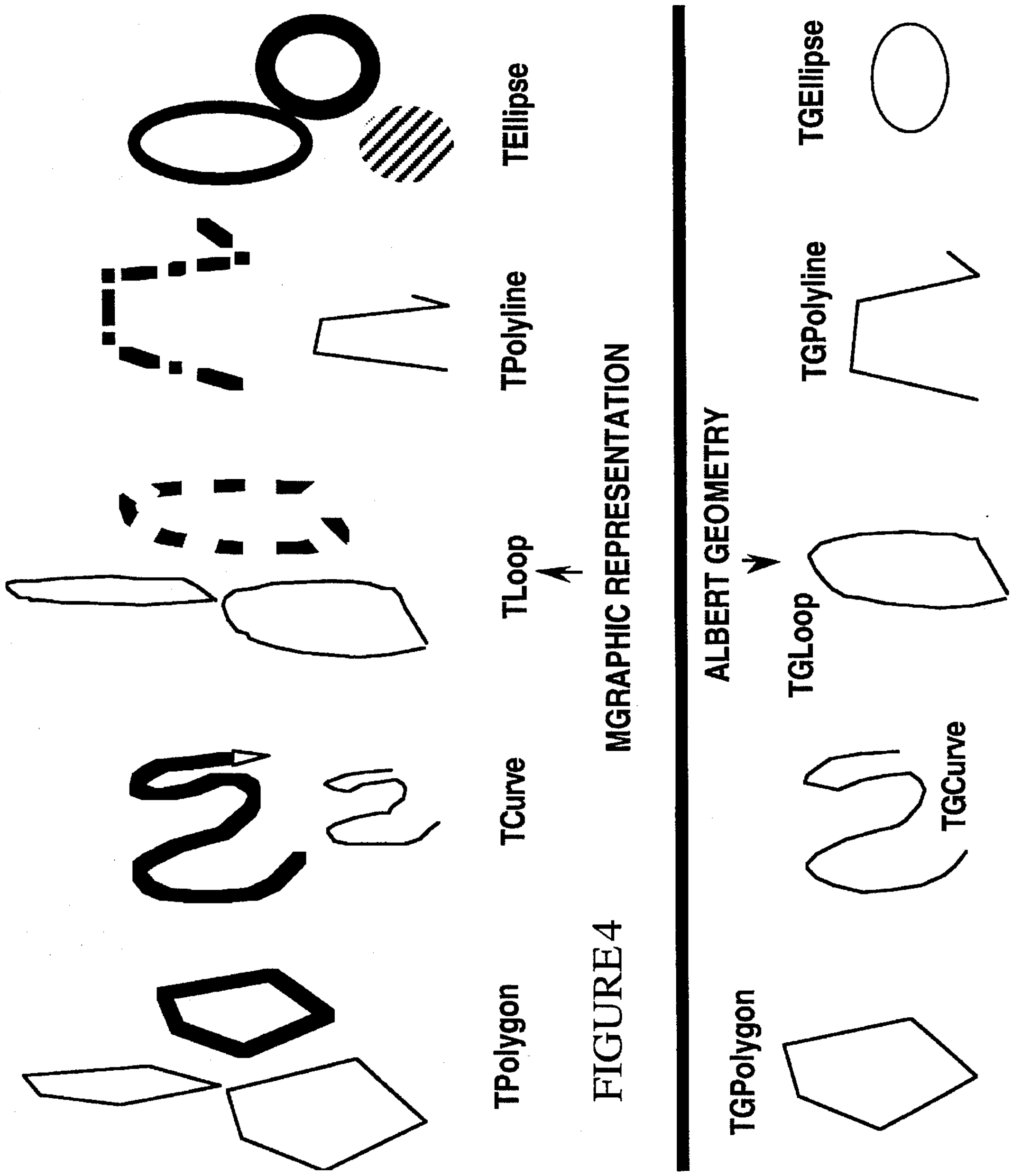

TPolygon
TCurve
TLoop
TPolyline
TEllipse
MGRAPHIC REPRESENTATION
ALBERT GEOMETRY
TGPolygon
TGCurve
TGLoop
TGPolyline
TGEllipse

FIGURE 4

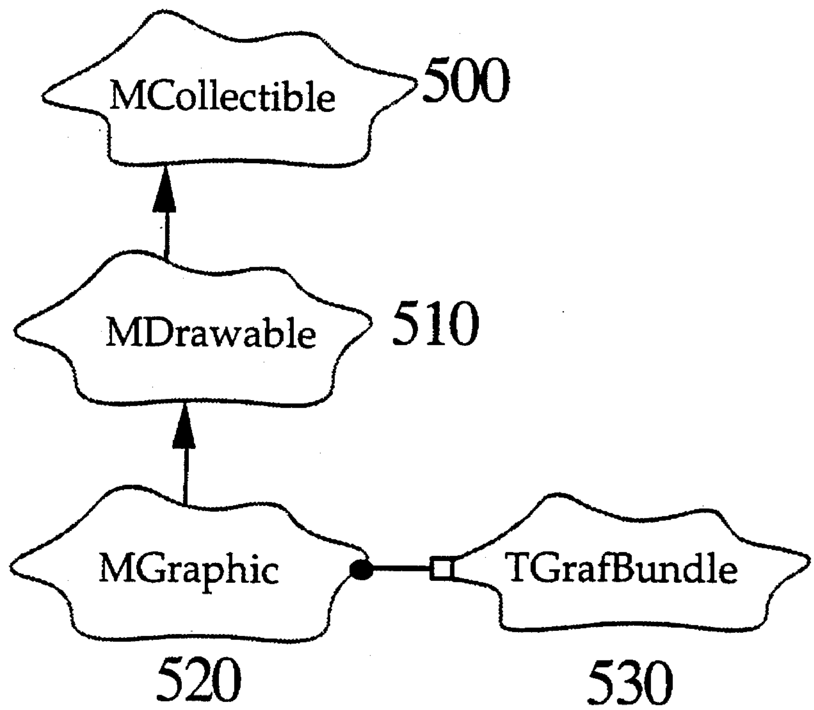


FIGURE 5

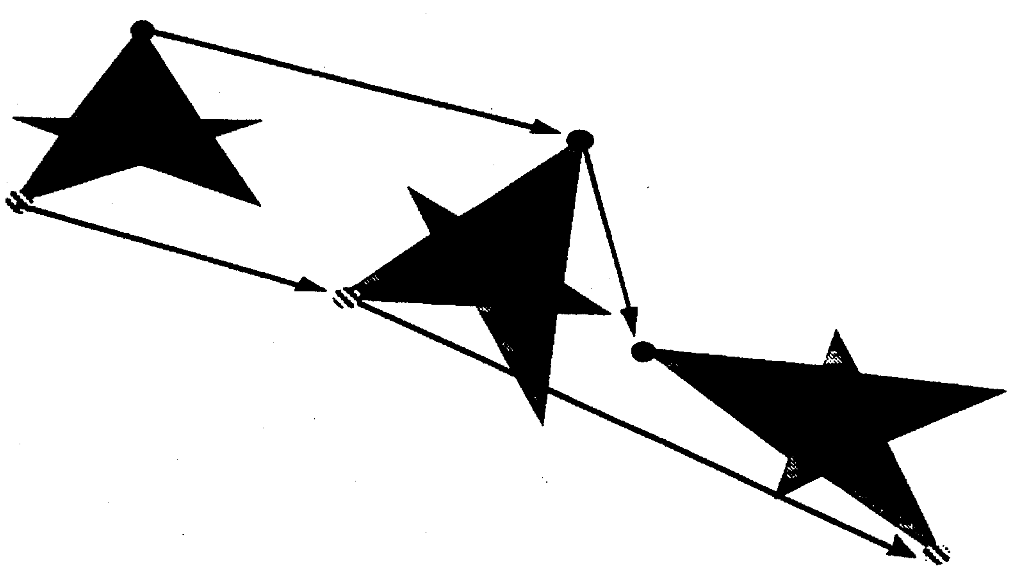


FIG 6

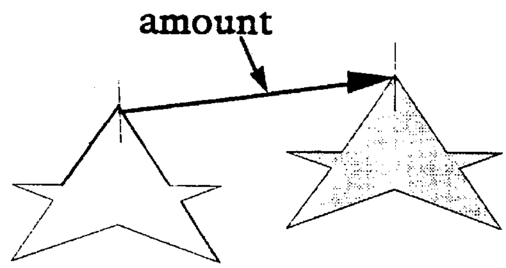


FIGURE 7

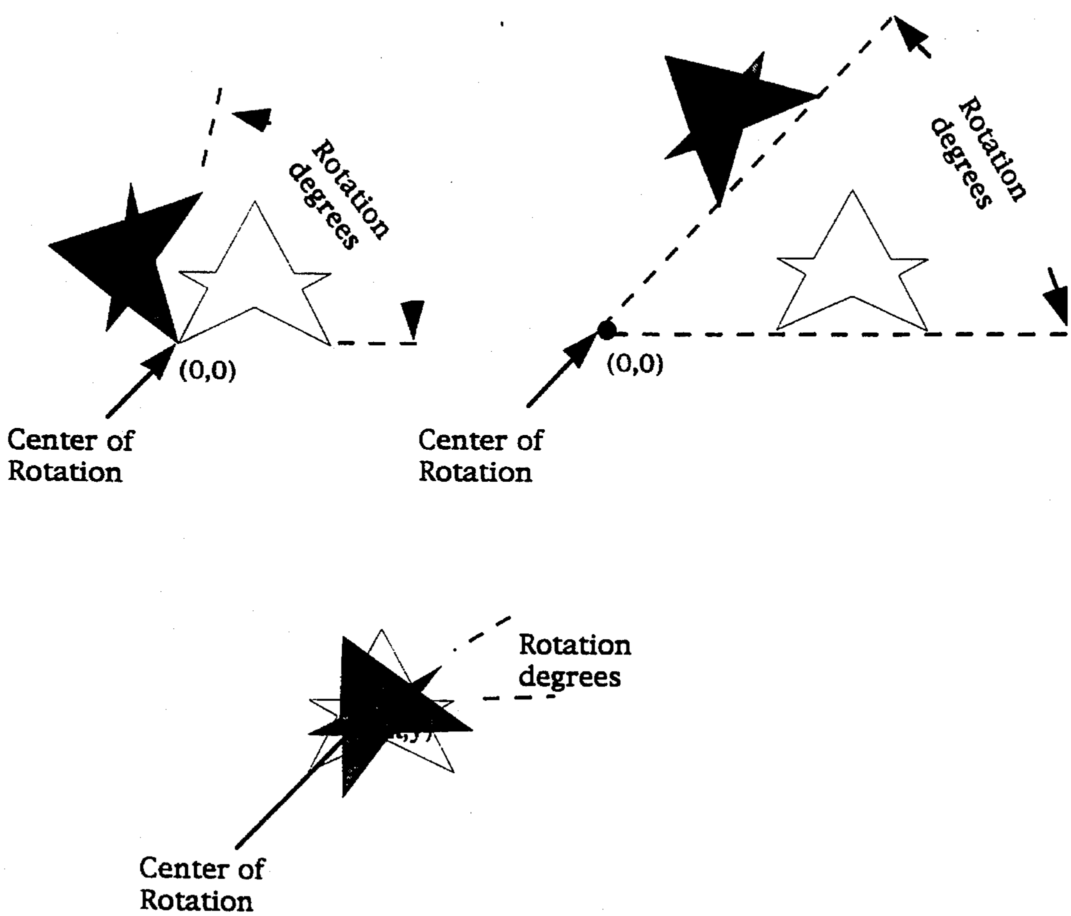
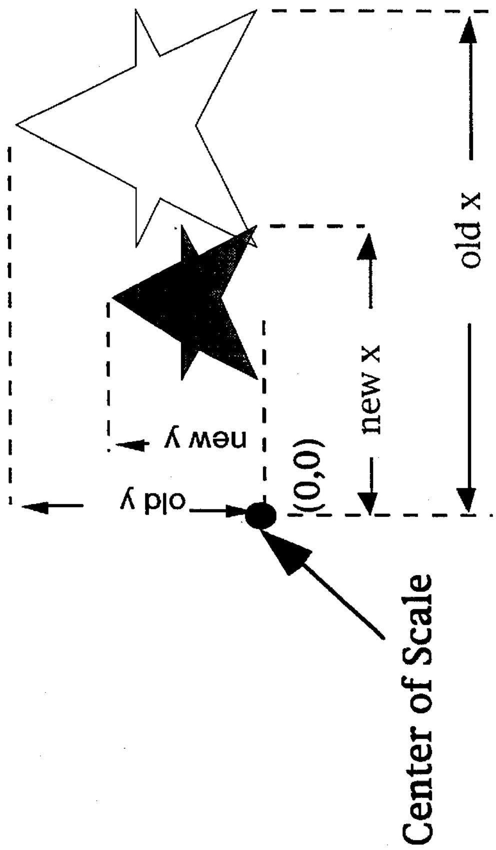


FIGURE 8



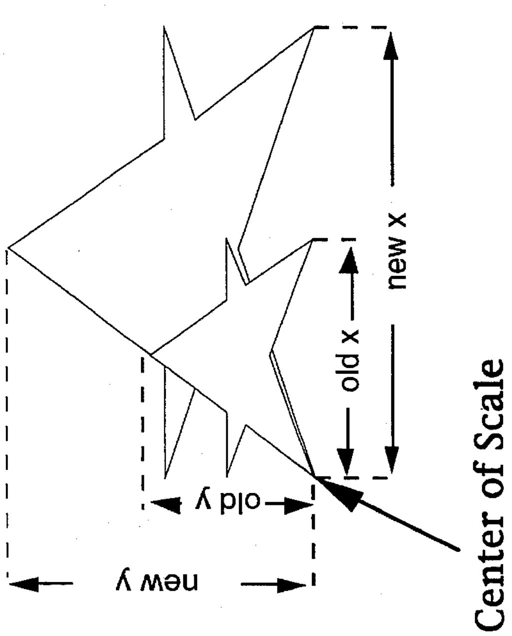


FIGURE 9A

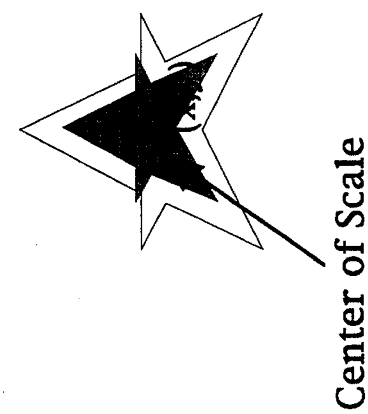


FIGURE 9B

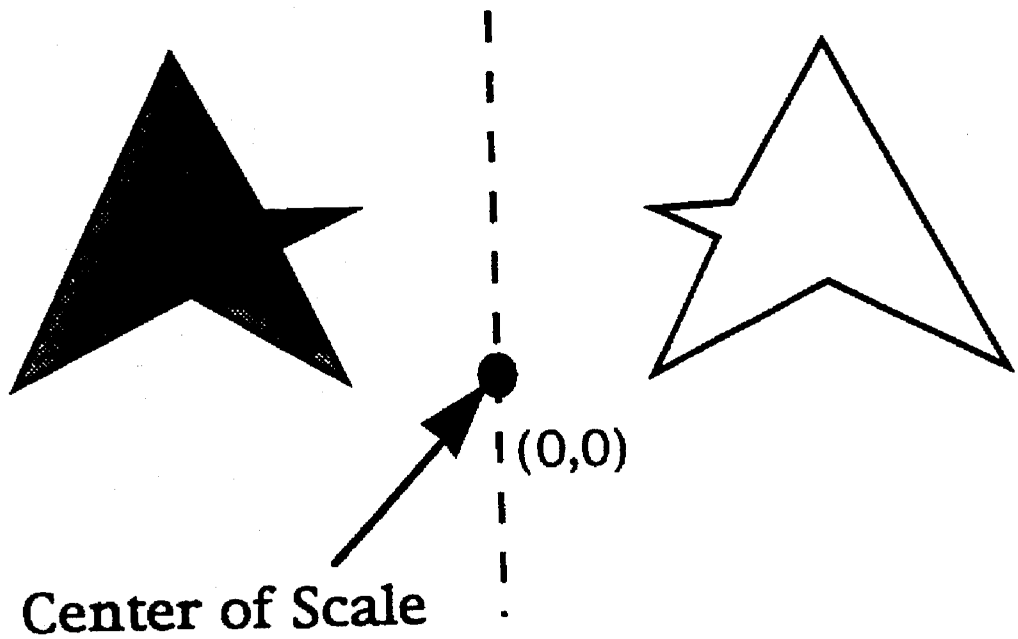


FIGURE 10

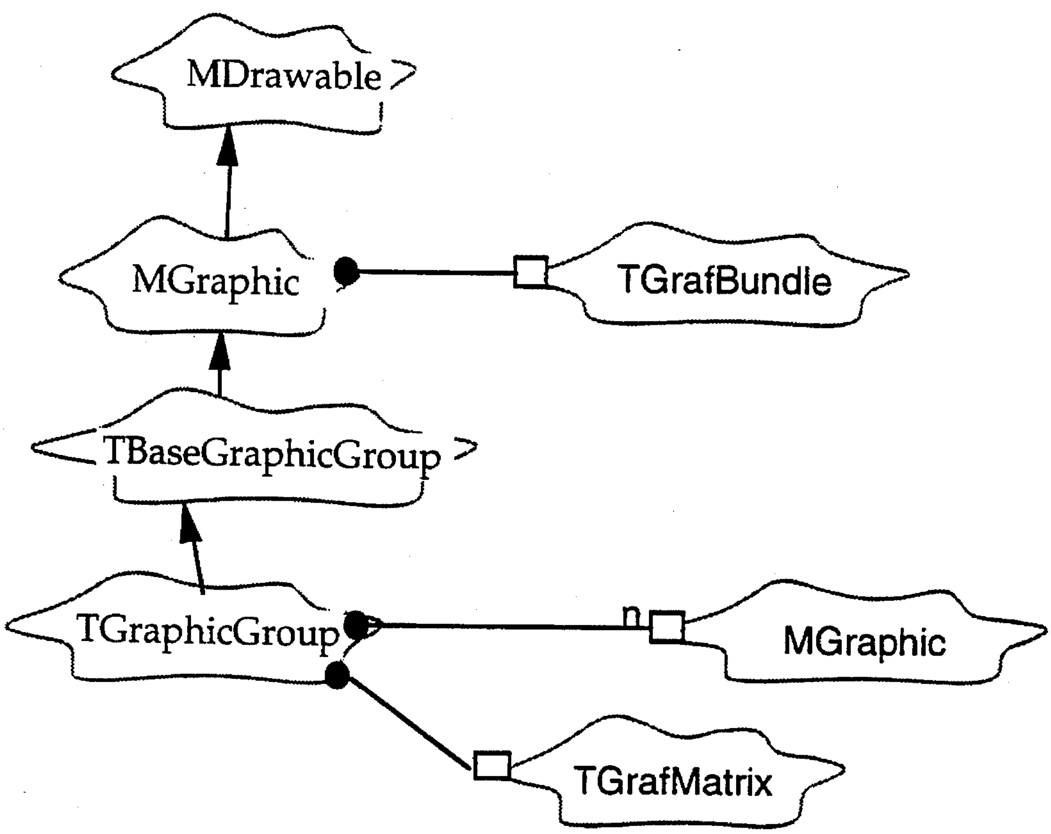

MDrawable
MGraphic
TGrafBundle
TBaseGraphicGroup
TGraphicGroup
n
MGraphic
TGrafMatrix

FIGURE 11

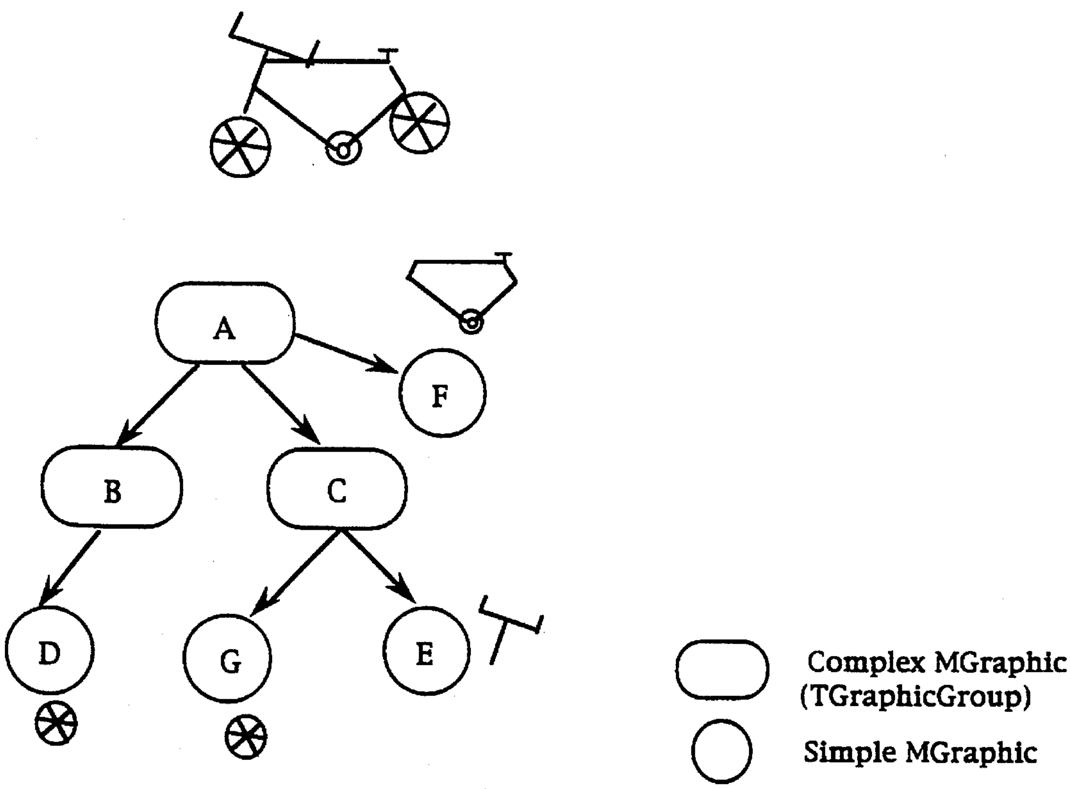


FIGURE 12

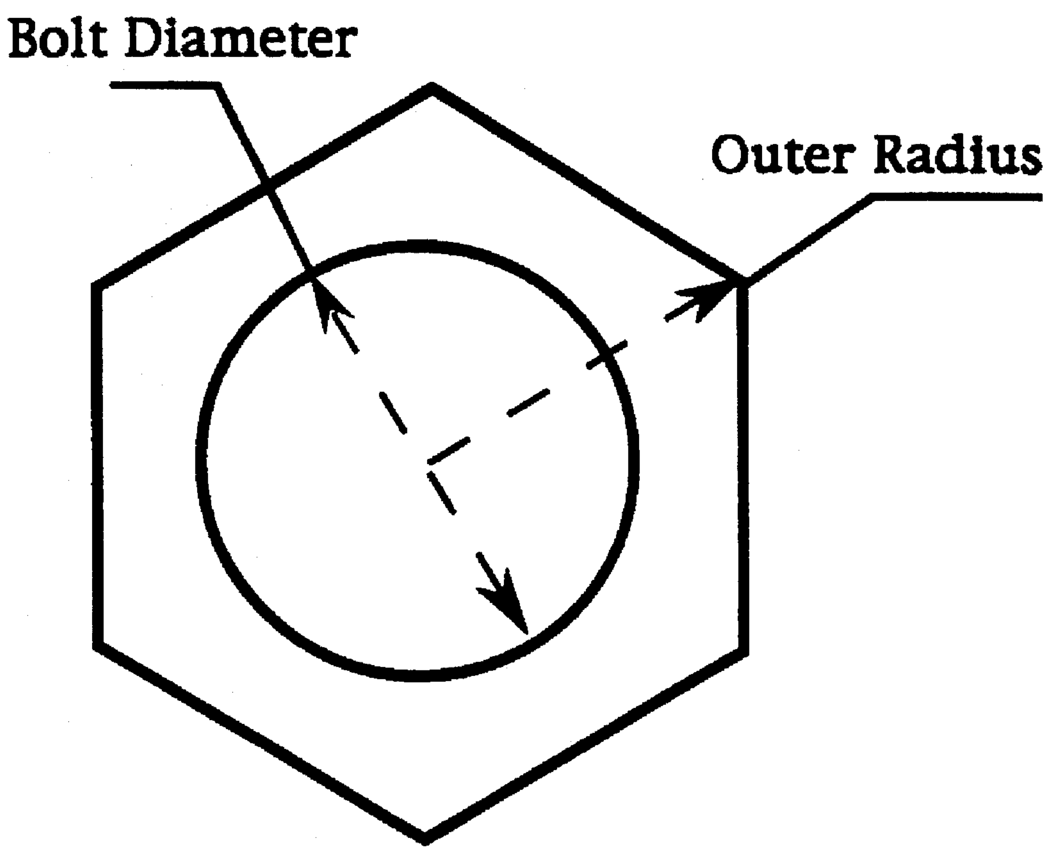


FIGURE 13

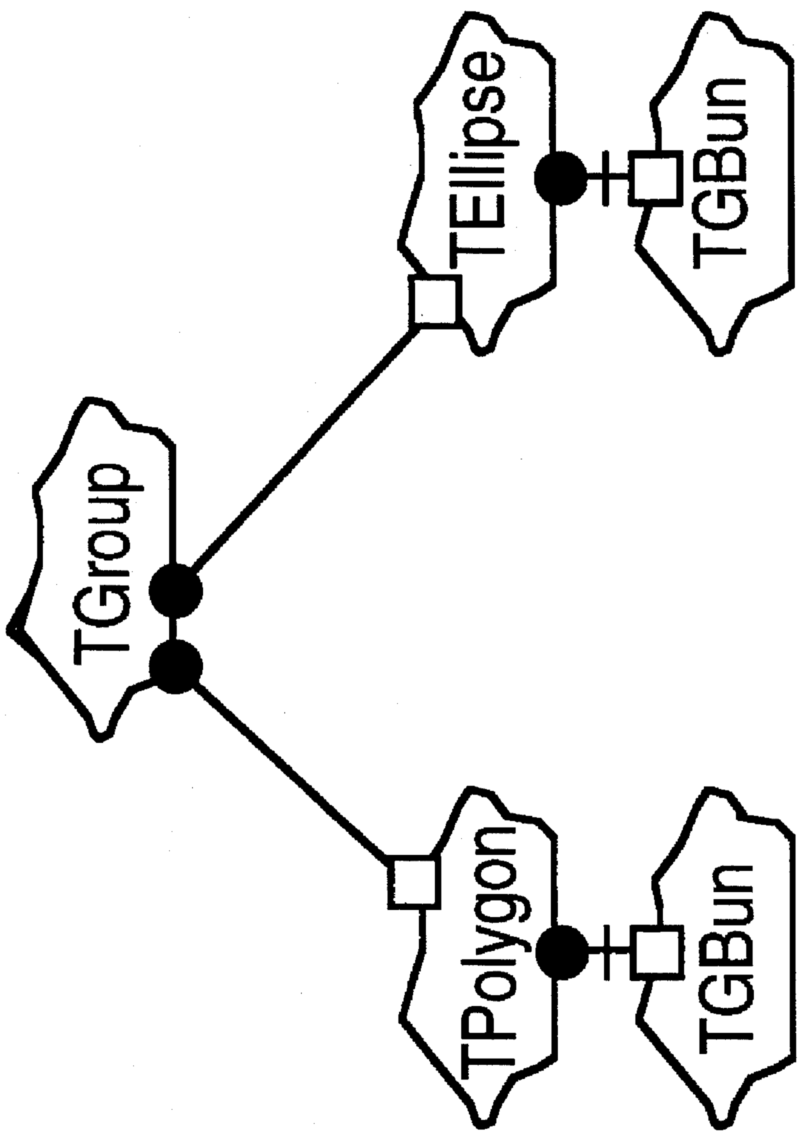


FIGURE 14

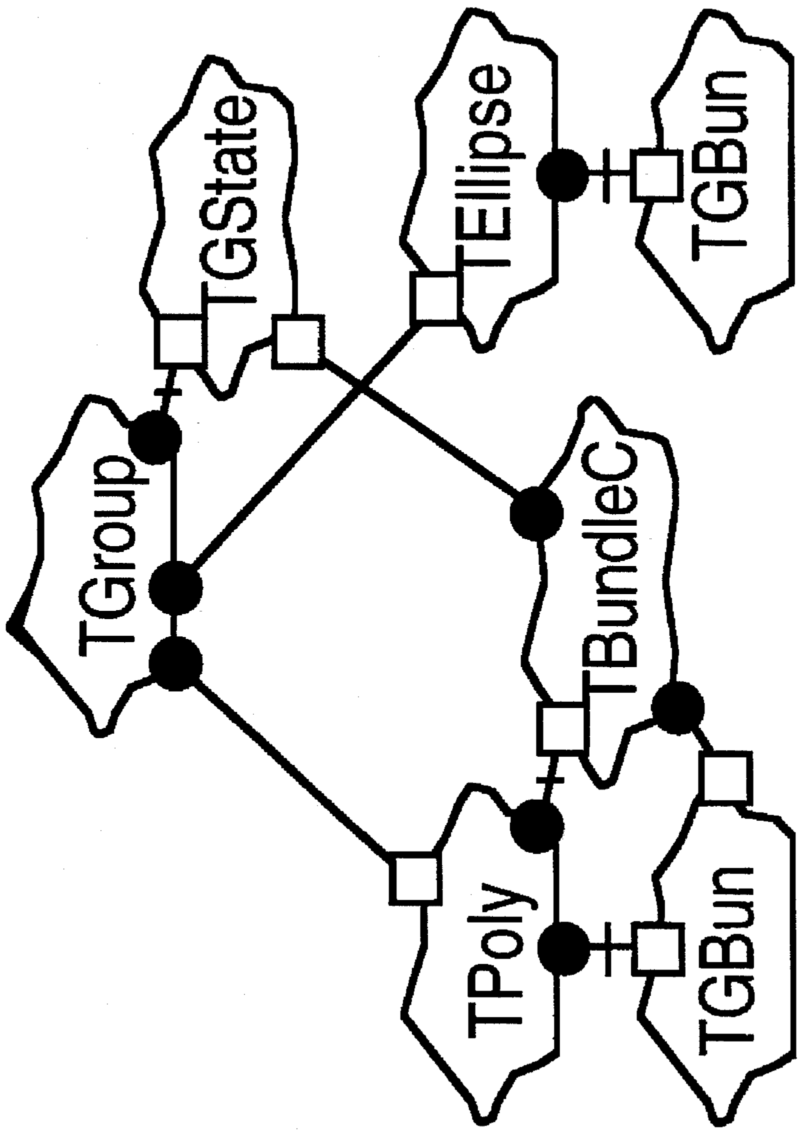


FIGURE 15

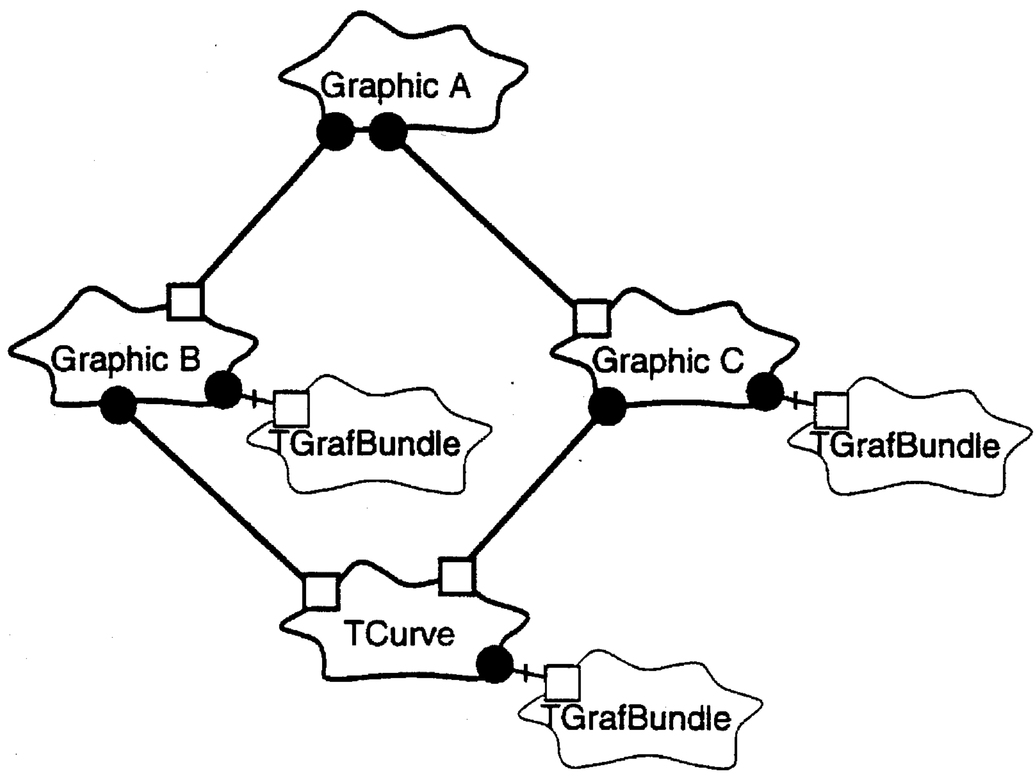

Graphic A
Graphic B
Graphic C
TGrafBundle
TGrafBundle
TCurve
TGrafBundle

FIGURE 16

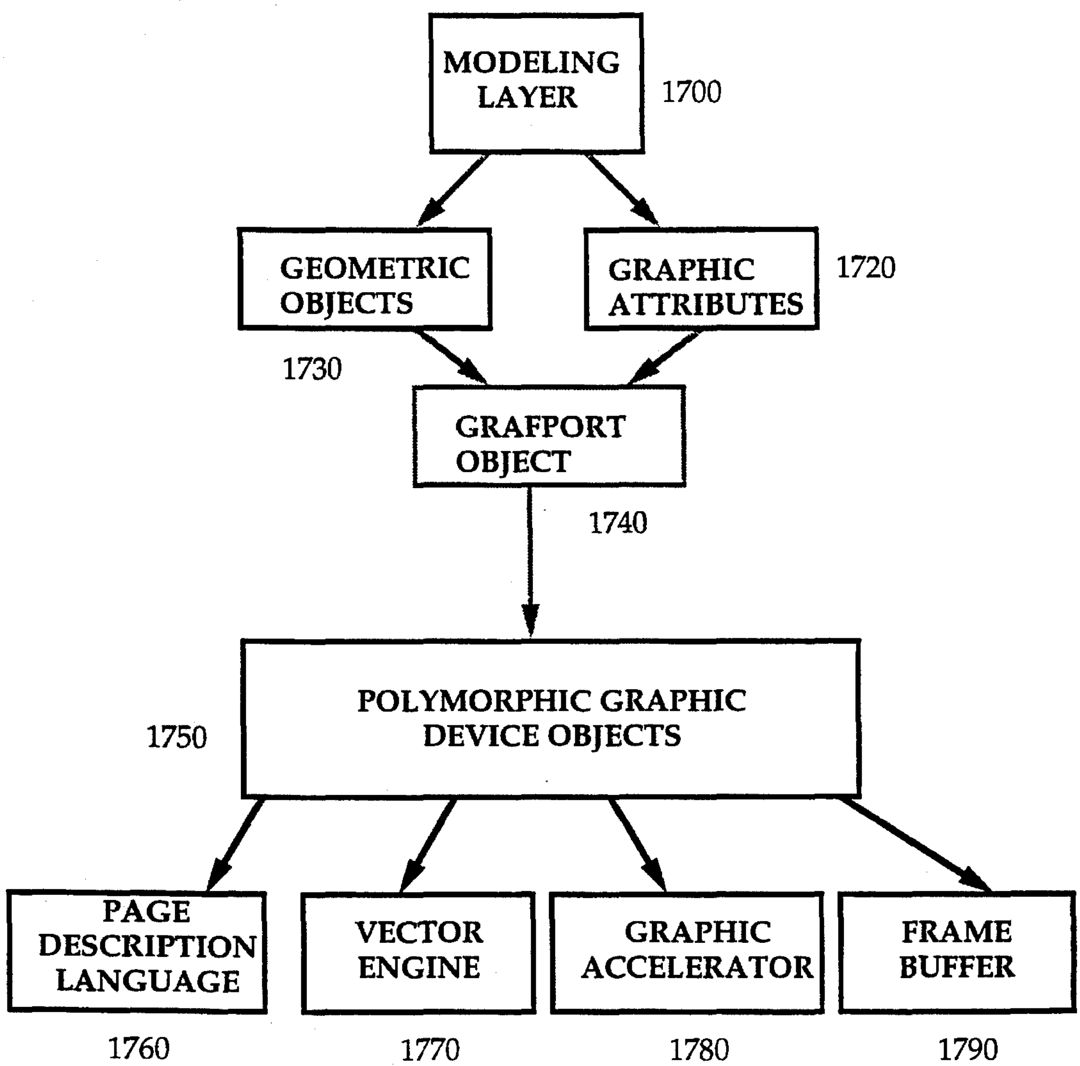

MODELING LAYER
1700
GEOMETRIC OBJECTS
GRAPHIC ATTRIBUTES
1720
1730
GRAFPORT OBJECT
1740
POLYMORPHIC GRAPHIC DEVICE OBJECTS
1750
PAGE DESCRIPTION LANGUAGE
VECTOR ENGINE
GRAPHIC ACCELERATOR
FRAME BUFFER
1760
1770
1780
1790

FIGURE 17

OBJECT-ORIENTED GRAPHIC SYSTEM

This is a continuation, of application Ser. No. 08/145,840, filed Nov. 2, 1993, abandoned.

COPYRIGHT NOTIFICATION

1. Field of the Invention

This invention generally relates to improvements in computer systems and more particularly to a system for enabling graphic applications using an object-oriented operating system.

2. Background of the Invention

Computer pictures or images drawn on a computer screen are called computer graphics. Computer graphic systems store graphics internally in digital form. The picture is broken up into tiny picture elements or pixels. Thus, a computer picture or graphic is actually an aggregation of individual picture elements or pixels. Internally, in the digital world of the computer, each pixel is assigned a set of digital values which represent the pixel's attributes. A pixel's attributes may describe its color, intensity and location, for example. Thus to change the color, intensity or location of a pixel, one simply changes the digital value for that particular attribute.

Conventional computer graphic systems utilize primitives known as images, bitmaps or pixel maps to represent computer imagery as an aggregation of pixels. These primitives represent a Two Dimensional (2D) array of pixel attributes and their respective digital values. Typically, such a primitive is expressed as a "struct" (data structure) that contains a pointer to pixel data, a pixel size, scanline size, bounds, and possibly a reference to a color table. Quite often, the pixels are assumed to represent Red, Green, and Blue (RGB) color, luminance, or indices into a color table. Thus, the primitive serves double duty as a framebuffer and as a frame storage specification.

The burgeoning computer graphics industry has settled on a defacto standard for pixel representation. All forms of images that do not fit into this standard are forced into second class citizenship. Conventional graphics systems, however, are nonextendable. They are usually dedicated to a particular application operating on a specific class of images. This is unacceptable in today's rapidly changing environment of digital technology. Every day a new application, and with it the need to process and manipulate new image types in new ways. Thus, the use of a graphics system with a nonextensible graphic specification is not only short sighted, it is in a word, obsolete. Graphical applications, attributes, and organizational requirements for computer output media are diverse and expanding. Thus, dedicated, single-purpose graphic systems fail to meet current application requirements. There is a need for a robust, graphic system that provides a dynamic environment and an extensible graphic specification that can expand to include new applications, new image types and provide for new pixel manipulations.

For example, two applications rarely require the same set of pixel attributes. Three Dimensional (3D) applications store z values (depth ordering), while animation and paint systems store alpha values. Interactive material editors and 3D paint programs store 3D shading information, while video production systems may require YUV 4:2:2 pixel arrays. Hardware clippers store layer tags, and sophisticated systems may store object IDs for hit detection. Moreover, graphical attributes such as color spaces are amassing constant additions, such as PhotoYCC™. Color matching technology is still evolving and it is yet unclear which quantized color space is best for recording the visible spectrum as pixels. Thus, there are a variety of data types in the graphics world. There are also a variety of storage organization techniques. To make matters even worse, it seems that every new application requires a different organization for the pixel memory. For example, Component Interleaved or "Chunky" scanline orientations are the prevailing organization in Macintosh ® video cards, but Component Interleaved banked switched memory is the trend in video cards targeted for hosts with small address spaces. Component planar tiles and component interleaved tiles are the trend in prepress and electronic paint applications, but output and input devices which print or scan in multiple passes prefer a component planar format. Multiresolution or pyramid formats are common for static images that require real-time resampling. Moreover, images that consume large amounts of memory may be represented as compressed pixel data which can be encoded in a multitude of ways.

The variety and growth of graphic applications, data types and pixel memory manipulations is very large. There is a requirement for a multipurpose system that can handle all the known applications and expand to handle those applications that are yet unknown. A single solution is impractical. Although it may handle every known requirement, it would be huge and unwieldy. However, if such an application is downsized, it can no longer handle every application. Thus, there is a need for a general graphic framework that suits the needs of many users, but allows the individual user to customize the general purpose graphic framework.

SUMMARY OF THE INVENTION

An object-oriented system is well suited to address the shortcomings of traditional graphic applications. Object-oriented designs can provide a general purpose framework that suits the needs of many users, but allows the individual user to customize and add to the general purpose framework to address a particular set of requirements. In general, an object may be characterized by a number of operations and a state which remembers the effect of these operations.

Thus it is a goal of the present invention to provide a method and apparatus which facilitates an object-oriented graphic system. A processor with an attached display, storage and object-oriented operating system builds a component object in the storage of the processor for managing graphic processing. The processor includes an object for connecting one or more graphic devices to various objects responsible for tasks such as graphic accelerators, frame buffers, page description languages and vector engines. The system is fully extensible and includes polymorphic processing built into each of the support objects.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1A** is a block diagram of a personal computer system in accordance with a preferred embodiment;

FIG. **1B** is a hierarchical layout of a graphic port in accordance with a preferred embodiment;

FIG. **2** is a block diagram of the architecture in accordance with a preferred embodiment;

FIG. 3 illustrates examples of graphic extensions of MGraphic in accordance with a preferred embodiment;

FIG. 4 illustrates MGraphics and their corresponding geometries in accordance with a preferred embodiment;

FIG. 5 is a booch diagram setting forth the flow of control of the graphic system in accordance with a preferred embodiment;

FIG. 6 illustrates a star graphic object undergoing various transformations in accordance with a preferred embodiment;

FIG. 7 depicts a star moved by an amount in accordance with a preferred embodiment;

FIG. 8 illustrates rotating-the star about various centers of rotation in accordance with a preferred embodiment;

FIG. 9 illustrates scaling a star about different centers of scale in accordance with a preferred embodiment;

FIG. 10 shows the effects of scaling an asymmetric star by (-1.0, 1.0) in accordance with a preferred embodiment;

FIG. 11 illustrates a hierarchical graphic in accordance with a preferred embodiment;

FIG. 12 illustrates a bike graphic in accordance with a preferred embodiment;

FIG. 13 illustrates a bolt object in accordance with a preferred embodiment;

FIG. 14 illustrates a hierarchical graphic in accordance with a preferred embodiment;

FIG. 15 illustrates an object that exists inside the TPolygon's Draw call in accordance with a preferred embodiment;

FIG. 16 illustrates a graphic hierarchy that supports sharing of two or more graphics in accordance with a preferred embodiment; and

FIG. 17 is a flowchart setting forth the detailed logic in accordance with a preferred embodiment.

DETAILED DESCRIPTION OF THE INVENTION

The invention is preferably practiced in the context of an operating system resident on a personal computer such as the IBM® PS/2® or Apple® Macintosh® computer. A representative hardware environment is depicted in FIG. 1, which illustrates a typical hardware configuration of a workstation in accordance with the subject invention having a central processing unit **10**, such as a conventional microprocessor, and a number of other units interconnected via a system bus **12**. The workstation shown in FIG. 1 includes a Random Access Memory (RAM) **14**, Read Only Memory (ROM) **16**, an I/O adapter **18** for connecting peripheral devices such as disk units **20** to the bus, a user interface adapter **22** for connecting a keyboard **24**, a mouse **26**, a speaker **28**, a microphone **32**, and/or other user interface devices such as a touch screen device (not shown) to the bus, a communication adapter **34** for connecting the workstation to a data processing network and a display adapter **36** for connecting the bus to a display device **38**. The workstation has resident thereon an operating system such as the Apple System/7® operating system.

In a preferred embodiment, the invention is implemented in the C++ programming language using object oriented programming techniques. As will be understood by those skilled in the art, Object-Oriented Programming (OOP) objects are software entities comprising data structures and operations on the data. Together, these elements enable objects to model virtually any real-world entity in terms of its characteristics, represented by its data elements, and its

behavior, represented by its data manipulation functions. In this way, objects can model concrete things like people and computers, and they can model abstract concepts like numbers or geometrical concepts. The benefits of object technology arise out of three basic principles: encapsulation, polymorphism and inheritance.

Objects hide, or encapsulate, the internal structure of their data and the algorithms by which their functions work. Instead of exposing these implementation details, objects present interfaces that represent their abstractions cleanly with no extraneous information. Polymorphism takes encapsulation a step further. The idea is many shapes, one interface. A software component can make a request of another component without knowing exactly what that component is. The component that receives the request interprets it and determines, according to its variables and data, how to execute the request. The third principle is inheritance, which allows developers to reuse pre-existing design and code. This capability allows developers to avoid creating software from scratch. Rather, through inheritance, developers derive subclasses that inherit behaviors, which the developer then customizes to meet their particular needs.

A prior art approach is to layer objects and class libraries in a procedural environment. Many application frameworks on the market take this design approach. In this design, there are one or more object layers on top of a monolithic operating system. While this approach utilizes all the principles of encapsulation, polymorphism, and inheritance in the object layer, and is a substantial improvement over procedural programming techniques, there are limitations to this approach. These difficulties arise from the fact that while it is easy for a developer to reuse their own objects, it is difficult to use objects from other systems and the developer still needs to reach into the lower, non-object layers with procedural Operating System (OS) calls.

Another aspect of object oriented programming is a framework approach to application development. One of the most rational definitions of frameworks came from Ralph E. Johnson of the University of Illinois and Vincent F. Russo of Purdue. In their 1991 paper, Reusing Object-Oriented Designs, University of Illinois tech report UIUCDCS91-1696 they offer the following definition: "An abstract class is a design of a set of objects that collaborate to carry out a set of responsibilities. Thus, a framework is a set of object classes that collaborate to execute defined sets of computing responsibilities." From a programming standpoint, frameworks are essentially groups of interconnected object classes that provide a pre-fabricated structure of a working application. For example, a user interface framework might provide the support and "default" behavior of drawing windows, scrollbars, menus, etc. Since frameworks are based on object technology, this behavior can be inherited and overridden to allow developers to extend the framework and create customized solutions in a particular area of expertise. This is a major advantage over traditional programming since the programmer is not changing the original code, but rather extending the software. In addition, developers are not blindly working through layers of code because the framework provides architectural guidance and modeling but at the same time frees them to then supply the specific actions unique to the problem domain.

From a business perspective, frameworks can be viewed as a way to encapsulate or embody expertise in a particular knowledge area. Corporate development organizations, Independent Software Vendors (ISV)s and systems integrators have acquired expertise in particular areas, such as manufacturing, accounting, or currency transactions. This expertise is embodied in their code. Frameworks allow organizations to capture and package the common characteristics of that expertise by embodying it in the organization's code. First, this allows developers to create or extend an application that utilizes the expertise, thus the problem gets solved once and the business rules and design are enforced and used consistently. Also, frameworks and the embodied expertise behind the frameworks, have a strategic asset implication for those organizations who have acquired expertise in vertical markets such as manufacturing, accounting, or bio-technology, and provide a distribution mechanism for packaging, reselling, and deploying their expertise, and furthering the progress and dissemination of technology.

Historically, frameworks have only recently emerged as a mainstream concept on personal computing platforms. This migration has been assisted by the availability of object-oriented languages, such as C++. Traditionally, C++ was found mostly on UNIX systems and researcher's workstations, rather than on computers in commercial settings. It is languages such as C++ and other object-oriented languages, such as Smalltalk and others, that enabled a number of university and research projects to produce the precursors to today's commercial frameworks and class libraries. Some examples of these are InterViews from Stanford University, the Andrew toolkit from Carnegie-Mellon University and University of Zurich's ET++ framework. Types of frameworks range from application frameworks that assist in developing the user interface, to lower level frameworks that provide basic system software services such as communications, printing, file systems support, graphics, etc. Commercial examples of application frameworks are MacApp (Apple), Bedrock (Symantec), OWL (Borland), NeXTStep App Kit (NeXT), and Smalltalk-80 MVC (ParcPlace).

Programming with frameworks requires a new way of thinking for developers accustomed to other kinds of systems. In fact, it is not like "programming" at all in the traditional sense. In old-style operating systems such as DOS or UNIX, the developer's own program provides all of the structure. The operating system provides services through system calls-the developer's program makes the calls when it needs the service and control returns when the service has been provided. The program structure is based on the flow-of-control, which is embodied in the code the developer writes. When frameworks are used, this is reversed. The developer is no longer responsible for the flow-of-control. The developer must forego the tendency to understand programming tasks in term of flow of execution. Rather, the thinking must be in terms of the responsibilities of the objects, which must rely on the framework to determine when the tasks should execute. Routines written by the developer are activated by code the developer did not write and that the developer never even sees. This flip-flop in control flow can be a significant psychological barrier for developers experienced only in procedural programming. Once this is understood, however, framework programming requires much less work than other types of programming.

In the same way that an application framework provides the developer with prefab functionality, system frameworks, such as those included in a preferred embodiment, leverage the same concept by providing system level services, which developers, such as system programmers, use to subclass/override to create customized solutions. For example, consider a multimedia framework which could provide the foundation for supporting new and diverse devices such as audio, video, MIDI, animation, etc. The developer that needed to support a new kind of device would have to write a device driver. To do this with a framework, the developer only needs to supply the characteristics and behaviors that are specific to that new device.

The developer in this case supplies an implementation for certain member functions that will be called by the multimedia framework. An immediate benefit to the developer is that the generic code needed for each category of device is already provided by the multimedia framework. This means less code for the device driver developer to write, test, and debug. Another example of using system frameworks would be to have separate I/O frameworks for SCSI devices, NuBus cards, and graphics devices. Because there is inherited functionality, each framework provides support for common functionality found in its device category. Other developers could then depend on these consistent interfaces for implementing other kinds of devices.

A preferred embodiment takes the concept of frameworks and applies it throughout the entire system. For the commercial or corporate developer, systems integrator, or OEM, this means all the advantages that have been illustrated for a framework such as MacApp can be leveraged not only at the application level for such things as text and user interfaces, but also at the system level, for services such as graphics, multimedia, file systems, I/O, testing, etc. Application creation in the architecture of a preferred embodiment will essentially be like writing domain-specific pieces that adhere to the framework protocol. In this manner, the whole concept of programming changes. Instead of writing line after line of code that calls multiple API hierarchies, software will be developed by deriving classes from the preexisting frameworks within this environment, and then adding new behavior and/or overriding inherited behavior as desired. Thus, the developer's application becomes the collection of code that is written and shared with all the other framework applications. This is a powerful concept because developers will be able to build on each other's work. This also provides the developer the flexibility to customize as much or as little as needed. Some frameworks will be used just as they are. In some cases, the amount of customization will be minimal, so the piece the developer plugs in will be small. In other cases, the developer may make very extensive modifications and create something completely new.

In a preferred embodiment, as shown in FIG. **1**, a multimedia data routing system manages the movement of multimedia information through the computer system, while multiple media components resident in the RAM **14**, and under the control of the CPU **10**, or externally attached via the bus **12** or communication adapter **34**, are responsible for presenting multimedia information. No central player is necessary to coordinate or manage the overall processing of the system. This architecture provides flexibility and provides for increased extensibility as new media types are added. A preferred embodiment provides an object-oriented graphic system. The object-oriented operating system comprises a number of objects that are clearly delimited parts or functions of the system. Each object contains information about itself and a set of operations that it can perform on its information or information passed to it. For example, an object could be named WOMAN. The information contained in the object WOMAN, or its attributes, might be age, address, and occupation. These attributes describe the object WOMAN. The object also contains a set of operations that it can perform on the information it contains. Thus, WOMAN might be able to perform an operation to change occupations from a doctor to a lawyer.

Objects interact by sending messages to each other. These messages stimulate the receiving object to take some action, that is, perform one or more operations. In the present invention there are many communicating objects. Some of the objects have common characteristics and are grouped together into a class. A class is a template that enables the creation of new objects that contain the same information and operations as other members of the same class. An object created from a certain class is called an instance of that class. The class defines the operations and information initially contained in an instance, while the current state of the instance is defined by the operations performed on the instance. Thus, while all instances of a given class are created equal, subsequent operations can make each instance a unique object.

Polymorphism refers to object-oriented processing in which a sender of a stimulus or message is not required to know the receiving instance's class. The sender need only know that the receiver can perform a certain operation, without regard to which object performs the operation or what class to which it belongs. Instances inherit the attributes of their class. Thus, by modifying the attribute of a parent class, the attributes of the various instances are modified as well, and the changes are inherited by the subclasses. New classes can be created by describing modifications to existing classes. The new class inherits the attributes of its class and the user can add anything which is unique to the new class. Thus, one can define a class by simply stating how the new class or object differs from its parent class or object. Classes that fall below another class in the inheritance hierarchy are called descendants or children of the parent class from which they descend and inherit. In this polymorphic environment, the receiving object is responsible for determining which operation to perform upon receiving a stimulus message. An operation is a function or transformation that may be applied to or by objects in a class. The stimulating object needs to know very little about the receiving object which simplifies execution of operations. Each object need only know how to perform its own operations, and the appropriate call for performing those operations a particular object cannot perform.

When the same operation may apply to many different classes, it is a polymorphic operation. The same operation takes on a different form in a variety of different classes. A method is the implementation of a particular operation for a given class. For example, the class Document may contain an operation called Read. Depending on the data type of the document, for example, ASCII versus BINARY, a different method might be used to perform the Read operation. Thus while both methods logically perform the same task, Read, and are thus called by the same name, Read, they may in fact be different methods implemented by a different piece of executable code. While the operation Read may have methods in several classes, it maintains the same number and types of arguments, that is, its signature remains the same. Subclasses allow a user to tailor the general purpose framework. It allows for different quantization tradeoffs, sets of pixel attributes, and different pixel memory organizations. Each subclass can encapsulate the knowledge of how to allocate, manage, stream, translate, and modify its own class of pixel data. All subsystems of a preferred embodiment use polymorphic access mechanisms, which enable a user to extend buffer types that can be rendered to or copied.

Fortunately, some commonalty exists among the various types of buffers. As it turns out, there are eight basic functions or categories that are necessary to satisfy the majority of client needs. Most clients want polymorphic management and the ability to specify the relationship between discrete and continuous space. Clients want to characterize color capabilities for use in accurate color reproduction. Clients want mechanisms for pixel memory alteration in the form of Get and SetPixel, specialized "blit loops" tailored for scan converting clients, BitBit, and CopyImage. Clients want mechanisms to supply clients with variants which match a key formed from the combination of client supplied attributes. Clients desire the ability to perform polymorphic queries regarding traits or stored attributes. Clients require mechanisms allowing clients to polymorphically create, maintain, and query buffer caches. And finally, clients require mechanisms which allow them to polymorphically create, and maintain correlated back-buffers.

Graphic Application Programming Interface (API)

The basic components of a graphic system include a fixed set of Geometric Primitives: Point, Rectangle, Line, Curve, Polygon, Polyline, Area in 2D, Line, Polyline, Curve and Surface in 3D. This set of geometry is not intend to be user extensible. This limits the complexity of the lower level graphic devices, and provides a "contract" between the user-level API and the low level device for consistent data. Discretized data sets: which include 2D raster images with a number of possible components and triangulated 3D datasets. High level modeling tools: that can express hierarchical groups of graphic objects. Transforms: these objects represent the operations available with a traditional 3×3 (in 2D) or 4×4 (in 3D) matrices to rotate, scale, translate, etc. objects. Bundles: these objects encapsulate the appearance of the geometry. Standard attributes include (2D & 3D) frame and/or fill color, pen thickness, dash patterns, etc. In 3D, bundles also define shading attributes. Custom attributes may be specified via a keyword/value pair. All numeric values are expressed in IEEE standard double precision floating point in the graphic system. Graphic Ports: a graphic port is an application-level view that encapsulates the state of the application. The graphic port re-routes any draw calls to an appropriate one of a number of possible devices (monitors, off screen frame buffers, PostScriptPrinter on a network, a window, etc.). Graphical "state" (current transform, bundle, clipping region, etc) is managed at the port level. However, at the device level the system is "stateless". In other words, the complete state for a particular rendering operation is presented to the device when that rendering occurs. Note that a device may turn around and invoke other devices. For example, a device for the entire desktop may first decide which screen the geometry falls on, and then invoke the render call for that particular screen.

Architectural Introduction

In past graphics architectures, a graphic typically stores its state (such as color, transfer mode, clip area, etc.) privately. When asked to draw, the graphic procedurally copies these state variables into a graphic port, where they are accessed by the rendering code. Thus, the graphic's state is available only during this explicit drawing operation. This is not object-oriented, and is a restriction a modern graphic system cannot afford to make. A preferred embodiment provides a framework for a graphic to store its state. The framework supports a "don't call us, we'll call you" architecture in which clients can get access to the graphic state outside the context of any particular function. This is the purpose of the graphic port class. It is an abstract class that defines the interface for accessing the state variables. Concrete subclasses define the actual storage and concatenation behavior of the state variables.

Graphic Port Class

A design employing graphic port classes groups the graphic states into four different groups, which then are grouped into a single class called graphic port. The four "sub-states" are TGrafBundle, TCoordinateSystem, TClipBoundary, and TSceneBundle. A graphic port object can be referenced by other classes that need access to the full graphic state. Additionally, a child's graphic state can be concatenated to its parent's graphic port object, producing a new graphic port object. FIG. 1B is a hierarchical layout of a graphic port in accordance with a preferred embodiment. A graphic port class also contains methods to access a device and a device cache. GetDevice returns a pointer to the device to which rendering is done. Typically, this device is inherited from the parent graphic port. GetCache returns a pointer to the cache used by the device to cache devicedependent objects. This cache must have been created by the device at an earlier time. The main purpose for subclassing graphic port and the four sub-states is to define how storage and concatenation of the graphic state, device, and device cache is done. A simpler, flat group of state variables would not be flexible enough to support customization of state concatenation for a subset of the state variables. Also, the sub-states assist in splitting the state variables into commonly used groups. For instance, a simple graphic typically needs only a TGrafBundle; more complex graphic objects may need a matrix and possibly a clip area.

A graphic class, such as MGraphic, must describe itself to a TGrafPortDevice in terms of the basic set of geometries, and each geometry must have a graphic port object associated with the geometry. The graphic port allows a graphic object to conveniently "dump" its contents into a TGrafDevice object. This is accomplished by supplying a set of draw functions in the graphic port class that mirrors a set of render functions in the TGrafDevice class. Each draw function takes a geometry and passes the geometry and the contained graphic state to the appropriate render call in the device. For convenience, an overriding bundle and model matrix are also passed.

FIG. **2** is a block diagram of the architecture in accordance with a preferred embodiment. In the preferred embodiment, a modeling layer **200** generates calls to a Graphic port **210** using the API **210** described above. This GraphPort interface accepts only a specific, fixed set of primitives forming a "contract" **250** between the user level API and the device level API **240**. The graphic port captures state information including transform, appearance ("bundle"), and clipping into a polymorphic cache **220** that is used across multiple types of devices. For each render call, the geometry and all relevant accumulated state information **230** is presented to the device via a polymorphic graphic device object **240**. A device managed by the graphic device object **240** may take the form of a page description language **260** (such as postscript), a vector plotting device **270**, a device with custom electronic hardware for rendering geometric primitives **280**, a traditional framebuffer **290**, or any other graphic device such as a display, printer or plotter.

Modeling Layer

Above the graphic port and geometry layers there is an optional modeling layer. A preferred embodiment provides a modeling layer, but an application can override the default. The default modeling layer is called a "MGraphic" layer. An MGraphic object encapsulates both geometry and appearance (a bundle). To render an MGraphic, a draw method is used. This method takes the graphic port the MGraphic is drawn into as an argument. The MGraphic draw method turns this information into a graphic port call. The goal behind separating the MGraphic layer from-the graphic port / geometry layer is to avoid a rigid structure suited to only one type of database. If the structure provided by the MGraphic objects does not satisfy the client's requirements, the architecture still permits a different data structure to be used, as long as it can be expressed in terms of primitive geometries, bundles, and transforms.

MGRAPHIC LAYER

The graphic system provides two distinct ways of rendering geometries on a device. An application can draw the geometry directly to the device. The class graphic port supports a well defined, but fixed set of 2D geometries. It supports these by a set of overloaded draw methods. When using this approach, attributes and transformation matrices are not associated with geometry, making it suitable for immediate mode rendering only. The following pseudo code is an example of how an application may use this approach to create a red line.

```
{
    create a  displayPort an instance of TGrafPort
    TGLine line( TGPoint( 0.0, 0.0 , TGPoint( 1.0, 1.0 ));          //Creates a line
    TGrafBundle redColor( TRGBColor( 1.0, 0.0, 0.0 ));              //Creates a red color bundle
    displayPort->Draw(line, redColor);                  //Render the line on to the GrafPort
}
```

Alternatively, an application can draw the geometry via a higher level abstraction called MGraphic. This is a retained mode approach to rendering of graphical primitives. MGraphic is an abstract base class for representing the 2D primitives of the graphic system. It is a higher level manifestation of graphical objects which can be held in a collection, be transformed and rendered to a graphic device (TGrafDevice). Each MGraphic object holds a set of its own attributes and provides streaming capability (with some restrictions on some of its subclasses). Hit testing methods provide a mechanism for direct manipulation of MGraphic objects such as picking. MGraphic provides extensibility through subclassing that is one of the key features of MGraphics. A particular subclass of MGraphic also creates hierarchies of MGraphic objects and provides the capability to extend the graphic system. FIG. **3** illustrates some examples of graphic extensions of MGraphic in accordance with a preferred embodiment.

MGraphic is a utility class for applications to hold geometry related data that includes geometry definition, grafbundle (set of graphical attributes defining the representation of the geometry) and a set of transformation methods. MGraphic objects also hold any other information required by a user and will copy and stream this user specific data to an application. This class may not be needed for applications interested in pure immediate mode rendering. For immediate mode rendering of the primitives the applications render geometry by passing an appropriate geometry object, a grafbundle and a transformation matrix to the graphic port. FIG. **4** illustrates MGraphics and their corresponding geometries in accordance with a preferred embodiment. FIG. **5** is a Booch diagram setting forth the flow of control of the graphic system in accordance with a preferred embodiment. In the Booch diagram of FIG. **5**, "clouds" depicted with dashed lines indicate classes or aggregations of classes (e.g. application **500**). Arrows connecting classes are directed from subclass to superclass and indicate a hierarchy including the properties of encapsulation, inheritance and polymorphism as is well understood in object technology and graphic notations accepted in the art which are illustrative thereof. Double lines indicate use of the class in the implementation or interface. A circle at one end of a line segment indicates containment or use in the class with the circle on the end of the line segment. For a more complete description of this notation, reference can be made to "Object Oriented Design" by Grady Booch, published by the Benjamin/Cummings Publishing Co., Copyright 1991. The current MGraphic **520** inherits from MDrawable **510** which inherits from MCollectible **500** to inherit the streaming, versioning and other behaviors of MCollectible **500**. Each MGraphic **520** also has a bundle, TGrafBundle **530**, which holds a set of attributes. These attributes are used by the MGraphic at rendering time.

The MGraphic abstract base class represents only 2D graphical primitives. In general it has been observed that 2D and 3D primitives do not belong to a common set unless users clear the 3D plane on which 2D primitives lie. 2D and 3D primitives have different coordinate systems and mixing them would confuse users. Clients can mix the two sets based upon their specific application requirements. The class MDrawable **510** is the abstract base class common to both MGraphic **520** and MGraphic3D abstracting the common drawing behavior of the two classes. This class is useful for clients interested only in the draw method and do not require overloaded functionality for both 2D and 3D.

MDrawable Drawing Protocol

All MGraphics (2D and 3D) draw onto the graphic port which is passed to the MGraphic as a parameter. Besides the state information, which is encapsulated by the GrafPort, all other information is contained in the MGraphic object. This information includes the geometry, attribute bundle and any transformation information. All MGraphics draw synchronously and do not handle updating or animating requirements. It is up to the client to create subclasses. When drawing 2D and 3D primitives as a collection, such as in a list of MDrawable objects, the drawing sequence is the same as it would be when 2D and 3D draw calls are made on the graphic port. Thus, drawing a 2D polygon, a 3D box and a 2D ellipse will render differently depending upon the order in which they are rendered. The graphic port passed to this method is a passive iterator which is acted upon by the MGraphic to which it is passed.

MGraphic Transformations

FIG. **6** illustrates a star undergoing various transformations in accordance with a preferred embodiment. Transformations can alter an MGraphic's shape, by scaling or perspective transformation, and position, by rotating and moving. The transformation methods allow applications to change an existing MGraphic's shape and location without having to recreate the MGraphic. All transformation methods apply only relative transformation to the MGraphic. Methods ScaleBy, MoveBy and RotateBy are special cases of the more general method TransformBy. Subclasses apply the transform directly to the geometry they own to directly change the geometry.

All MGraphic subclasses are closed to arbitrary transformations i.e. a TGPolygon when transformed by an arbitrary transformation will still be a TGPolygon. However, certain geometries do not possess this closure property. For example, a rectangle, when transformed by a perspective matrix, is no longer a rectangle and has no definition for either width or height. The original specification of the rectangle is insufficient to describe the transformed version of the rectangle. All MGraphic subclasses must be closed to arbitrary transformations. Since all transformations are relative, a transformed MGraphic cannot be "untransformed" by passing an identity matrix to the MGraphic method TransformBy().

FIG. **7** depicts a star moved by an amount in accordance with a preferred embodiment. This method moves the MGraphic by an amount relative to its current position. FIG. **8** illustrates rotating the star about various centers of rotation in accordance with a preferred embodiment. The amount of rotation is specified in degrees and is always clockwise. However, subclasses can override the default and optimize for a specific geometry and usage. FIG. **9** illustrates scaling a star about different centers of scale in accordance with a preferred embodiment. The factor is a vector which allows non-uniform scaling namely in X and Y. In FIG. **9** the X coordinate of the parameter amount will be (new x/old x) and the Y coordinate will be (new y/old y). In case of uniform scaling both the X and the Y coordinate will be the same. FIG. **9** also shows scaling about different centers of scale.

Negative scale factors are allowed, and the effects of negative scale factors is the same as mirroring. Scaling by −1.0 in the X direction is the same as mirroring about the Y axis while a negative scale factor in the Y direction is the same as mirroring about the X axis. FIG. **10** shows the effects of scaling an asymmetric star by (−1.0, 1.0) in accordance with a preferred embodiment. Like RotateBy() and TranslateBy(), the effect of this transform is the same as creating a scaling matrix and passing it to TransformBy() and this is the default implementation. Subclasses can override this default implementation and optimize for a specific geometry and usage. TransformBy is a pure virtual member function that transforms the MGraphic by matrix. All concrete subclasses of MGraphics must define this member function. Subclasses that own a TGrafMatrix for manipulation must post multiply the parameter matrix with the local matrix for proper effect.

MGraphic Attribute Bundles

As seen in FIG. **5**, all MGraphic objects have an associated attribute bundle, TGrafBundle. This bundle holds the attribute information for the graphic object such as its color, pens, filled or framed. When an MGraphic is created, by default, the GrafBundle object is set to NIL. If GrafBundle is equal to a NIL, then the geometry is rendered by a default mechanism. When used in a hierarchy, the parent bundle must be concatenated with the child's bundle before rendering the child. If a child's bundle is NIL, then the child uses the parent's bundle for rendering. For example, in the hierarchy in FIG. **12**, object E will inherit the attributes of both A, C and E before it is rendered, and a change of attribute in A will trickle down to all its children namely B, C D, E, G and D.

It is important to note that a bundle has a significant amount of information associated with it. Thus, copying of the bundle is generally avoided. Once the bundle is adopted, MGraphic object will take full responsibility to properly destroy the bundle when the MGraphic object is destroyed. When a client wishes to modify an attribute of an MGraphic object, they do so by orphaning the bundle, changing the attribute, and then having the MGraphic adopt the bundle. Also, all caches that depend upon bundles must be invalidated when the bundle is adopted or orphaned. When an object orphans data, it returns a pointer to the data and takes no further data management responsibility for the data. When an object adopts data, it takes in the pointer to the storage and assumes full responsibility for the storage. Default implementations of all bundle related member functions has been provided in the base MGraphic class and subclasses need not override this functionality, unless the subclasses have an attribute based cache which needs to be invalidated or updated whenever the bundle is adopted and orphaned. For example, the loose fit bounds, when cached, need to be invalidated (or reevaluated) when the attributes change.

C++ Application Program Interfaces (API) for Bundle Management

virtual void AdoptBundle(TGrafBundle *bundle)

MGraphic adopts the bundle.

If an MGraphic object already holds a bundle, it is deleted, and the new bundle is attached. As pointers are passed, it is important for the clients not to keep references to the bundle passed as the parameter. The MGraphic object will delete the bundle when it gets destroyed.

virtual const TGrafBundle* GetBundle() const

This method allows users to inquire a bundle and then subsequently inquire its attributes by iterating through them. This method provides an alias to the bundle stored in the MGraphic object.

virtual TGrafBundle* OrphanBundle()

This method returns a bundle to a calling application for its use. Once this method is called, it is the calling application's responsibility to delete the bundle unless it is adopted again by an MGraphic object. When orphaned, the MGraphic bundle is set to NIL, and when the graphic is subsequently drawn, the MGraphic uses the default mechanism of attributes/bundles for its parent's bundle. This kind of MGraphic subclass references other MGraphic objects. Although all manipulative behavior of complex MGraphic objects is similar to a MGraphic object, these objects do not completely encapsulate MGraphic objects they refer to. Of the subclasses supported by a preferred embodiment, the one that falls in this category is TGraphicGroup. TGraphicGroup descends from the abstract base class TBaseGraphicGroup which makes available polymorphically the methods to create iterators for traversing groups. It is important for clients creating groups or hierarchies to descend from the base class TBaseGraphicGroup for making available the iterator polymorphically. FIG. 11 illustrates the class hierarchy in accordance with a preferred embodiment.

TBaseGraphicGroup Iterator Support

Since GraphicGroup facilitates creation of hierarchies, support for iterating the hierarchy is built into this base class and is available polymorphically. This method is virtual in the abstract base class TBaseGraphicGroup and all subclasses provide an implementation. Subclasses which desire a shield for their children may return an empty iterator when this member function is invoked.

Protocol: TGraphicIterator* CreateGraphicIterator() const=0

This method creates a Graphic iterator which iterates through the first level of a hierarchy. For example in FIG. 12, the graphic iterator created a concrete subclass to iterate over B, C and F. To iterate further, iterators must be created for both B and C as these are TBaseGraphicGroups. All subclasses creating hierarchies must provide a concrete implementation.

TGraphicIterator is an active iterator that facilitates the iteration over the children of a TBaseGraphicGroup.

TGraphicIterator methods include:

const MGraphic *TGraphicIterator::First()

const MGraphic *TGraphicIterator::Next()

const MGraphic *TGraphicIterator::Last()

TGraphicGroup

The graphic system provides a concrete subclass of TBaseGraphicGroup, namely TGraphicGroup, which supports creation of trees. TGraphicGroup creates a collection of MGraphic objects forming a group. As each of the MGraphic objects can be a TGraphicGroup, clients can create a hierarchy of objects. FIG. 12 is an example of a hierarchy created by TGraphicGroup. FIG. 12 contains TGraphicGroups A, B and C. D, E, F and G are different simple MGraphics encapsulating more than one geometry. A has references to B, C and F. B refers to D while C refers to G. Group C also refers to the MGraphic E. FIG. 12 can be considered as an over simplified bike, where A refers to MGraphic F-the body of bike, and groups B and C which refer to the transformations associated with the rear and the front wheel respectively. The two wheels are represented by the primitive geometries D and G. E represents the handle-bar of the bike. Moving node C will move both the front wheel and the handle-bar, and moving node A will move the entire bike.

While applying a transformation matrix to the children at the time of rendering, the group creates a temporary GrafPort object and concatenates its matrix with that stored in the GrafPort. This new GrafPort is used to render its children and is destroyed once the child is completely rendered. The GrafPort objects are created on the stack. TGraphicGroup does not allow its children to have more than one parent in a team. TGraphicGroup inherits directly from MGraphic and thus each of the nodes own its own grafbundle and can affect its own side of the hierarchy. The destructor of TGraphicGroup destroys itself and does not destroy its children. It is up to an application to keep track of references and destroy MGraphic objects when they are not referenced.

GraphicGroup Iterator

Graphic Group provides a concrete implementation for iterating its children. The Graphic Iterator created iterates only one level. Clients interested in iterating more than one level deep can do so by creating iterators on subsequent TGraphicGroups.

Attribute and Transformation Hierarchy

Each TGraphicGroup, if it so chooses, defines its own attributes and transformation. By default, an attribute bundle is NIL and the transformation matrix is set to the identity matrix. As TGraphicGroup is a complex MGraphic, it has references to other MGraphics, and its children. By definition, each of the children must inherit the attribute traits and transformations of its parent. However, since each child can contain multiple references, it inherits these attributes by concatenating the parents information, without modifying its own, at the time of rendering. The concatenation of these attributes is achieved at the time of the Draw call. Both the attribute and the matrix are concatenated with the TGrafPort object which is passed as a parameter to the Draw call. In FIG. **12**, attributes and transformations of object A (body of bike) are concatenated with the GrafPort object passed to A (as parameter to member function Draw) and a new GrafPort object, APortObject, is created on the stack. APortObject is passed to object C which concatenates its state and creates a new port object, CPortObject. The new CPortObject is passed to object E to be rendered. Object E concatenates its state with CPortObject and renders itself using the new state.

MGRAPHIC EXAMPLE

As an example, a graphic is subclassed from MGraphic to create a special 2D primitive which corresponds to a top view of a bolt. This class stores a transformation matrix for a local coordinate system, and is a very simple example without taking into account performance and efficiency. FIG. **13** illustrates a bolt object in accordance with a preferred embodiment. The code below is a C++ source listing that completely defines the bolt object in accordance with a preferred embodiment.

```
class TBoltTop : public MGraphic {
public:
        TBoltTop(GCoord BoltDiameter, GCoord outerradius, TGPoint center);
        TBoltTop(const TBoltTop&);
        TBoltTop& operator= (const TBoltTop&);
        virtual void Draw(TGrafPort&) const;
        virtual TGPoint GetAlignmentBasePoint() const;
        virtual TGRect GetLooseFitBounds() const;
        virtual TGRect GetGeometricBounds() const;
        virtual void TransformBy(const TGrafmatrix& matrix);
        virtual Boolean Find(TGrafSearcher& searcher) const;
private:
        TBoltTop();              / /For streaming purposes only.
        TGrafMatrix fMatrix;
        TGPolygon fPolygon;      / / This is the outer polygon
        TGEllipse fCircle;       / / This is the inner circle
        void ComputePolygon(GCoord outerRad, int numOfSides);
};
TBoltTop::TBoltTop()
{
}
TBoltTop::TBoltTop(GCoord boltDia, GCoord outerDia, TGPoint center)
                                                : fCircle(boltDia, center)
{
        calculate the hexagon polygon from these paramters
        The side of the polygon = outerDiameter / 2.0
        TGPointArray polygonPoints(6);
        TGPoint tmpPoint;
        for (unsigned long i = 0, theta = 0.0;i < 6;i ++,
                                                theta += kPi/6) {
                tmpPoint.fX = center.fX + outerDia * sin(theta);
                tmpPoint.fY = center.fY + outerDia * cos(theta);
                polygonPoints.SetPoint(i, tmpPoint);
        }
}
void TBoltTop::Draw(TGrafPort &port) const
{
        /*
        * draw the geometry with the Grafllundle and the matris
        * associated with this primitive
        * /
        port.Draw(fPolygon, fGrafBundle, fMatrix);
        port.Draw(fCircle, fGrafBundle, fMatrix);
        /*
        * If there are a large number of primitives with same attributes
        * it is efficient to construct a local port and then render
        * geometries into this local port.
        * The semantics will be as:
        *
        * TConcatenatedGrafPort newPort(port, fGrafBundle, fMatrix);
        * TConcatenatedGrafPort is a port that concatenates bundle and
        * matrix with the state information of the old port.
        *
        * newPort.Draw(fPolygon);
        * newPort.Draw(fCircle);
        */
}
TGPoint TBoltTop::GetAlignmentBasePoint() const
{
        / / The alignment point is the center of the circle.
```

-continued

```
        TGPoint point;
        point.x = fCircle.GetCenterX();
        point.y = fCircle.GetCenterY();
        return point;
}
TGRect TBoltTop::GetLooseFitBounds() const
{
        TGRect bounds;
        / / Get bounds of the polygon
        / / pass the bounds to the bundle for altering.
        GetGeometricBounds(bounds);
        fGrafBundle->AlterBounds(bounds);
        return bounds;
}
TGRect TBoltTop::GetGeometricBounds() const
{
        / / Get bounds of the polygon
        / / pass the bounds to the bundle for altering.
        bounds = fPolygon.GetBounds();
}
void TBoltTop::TransformBy(const TGrafMatrix& matrix)
{
        fMatrix.ConcatWith(matrix);
}
void TGrafSearch::EFindResult TBoltTop::Find(TGrafSearch& search) const
{
        if (!search.find(fPolygon, fgrafBundle, fMatrix)) {
                return search.find(fCircle, fGrafBundle, fMatrix);
        }
        return TGrafSearch::kDoneSearching;
}
```

The Device Cache

The device cache can potentially be a large object, so care must be taken to ensure that device caches do not proliferate throughout the system unexpectedly. If the same base, GrafPort, is utilized for a number of hierarchies, the hierarchies would automatically share the cache in the base GrafPort.

Graphic State Concatenation

FIG. 14 illustrates a hierarchical graphic in accordance with a preferred embodiment. The graphic consists of a polygon and an ellipse in a group. Each graphic in the hierarchy can store a graphic state. For instance, the polygon and the ellipse each have a TGrafBundle, while the TGroup stores no graphic state. This architecture is easily understood until hierarchical states for matrices are considered. To produce the correct geometry matrix, a graphic's local view matrix must be concatenated with the view matrix of its parent. This concatenated matrix may then be cached by the graphic that provided it. A graphic's state must be "concatenated" to that of its parent graphic, producing a new, full set of states that applies to the graphic. When TGroup::Draw is called, its parent's graphic port object is passed in. Since the TGroup has no state of its own, it doesn't perform any concatenation. It simply passes its parent's graphic port object to the polygon's Draw call and then to the ellipse's Draw call.

The polygon has a TGrafBundle object that must be concatenated to its parent's graphic port object. This is facilitated by creating a local graphic port subclass that can perform this concatenation. It then makes a call to TBundleConcatenator::Draw. FIG. 15 illustrates an object that exists inside the TPolygon's Draw call in accordance with a preferred embodiment. Because the TBundleConcatenator object is created locally to a TPolygon's Draw call, this type of concatenation is transient in nature. This processing is required for particular types of graphic hierarchies. For instance, a graphic hierarchy that allows a particular graphic to be shared by two or more other graphics must implement transient concatenation because the shared graphic has multiple parents. FIG. 16 illustrates a graphic hierarchy that supports sharing of two or more graphics in accordance with a preferred embodiment. The curve object in this example is shared by graphics B and C. Thus, the concatenation must be transient because the results of the concatenation will be different depending on the branch taken (B or C).

Graphic objects in a persistent hierarchy require knowledge of parental information, allowing a graphic to be drawn using its parent's state without drawing its parent. A graphic in the hierarchy cannot be shared by multiple parents. Extra semantics, such as a ConcatenateWithParent call and a Draw call with no parameters, must be added to the graphic classes used in the hierarchy. A graphic may use a graphic port subclass that stores more state, such as a coordinate system and clip boundary. Thus, each graphic may also want to keep its own private device cache.

FIG. 17 is a flowchart of the detailed logic in accordance with a preferred embodiment. Processing commences at function block **1700** where a modeling layer object communicates with the grafport object **1740** with a fixed set of geometric objects **1730** and an extensible set of graphic attribute objects **1720**. The grafport object **1740** passes the geometric object **1730** and graphic attributes **1720** to a polymorphic graphic device object **1750** which manages devices (hardware and software) such as a page description language object **1760**, a vector engine object **1770**, a graphic accelerator object **1780**, a frame buffer object **1790**; or more traditional graphic devices such as displays, printers or plotters as depicted in FIG. **1**.

While the invention has been described in terms of a single preferred embodiment, those skilled in the art will recognize that the invention can be practiced with modification within the spirit and scope of the appended claims.

Having thus described our invention, what we claim as new, and desire to secure by Letters Patent is:

1. An object-oriented graphic system, comprising:

(a) a processor;

(b) a storage under the control of and attached to the processor;

(c) one or more graphic devices under the control of and attached to the processor;

(d) a grafport object in the storage of the processor;

(e) a graphic device object in the storage of the processor for managing one of the one or more graphic devices;

(f) a graphic object in the storage of the processor for managing graphic processing; and

(g) means for connecting the graphic device object to the grafport object to output graphic information on the one of the one or more graphic devices under the control of the graphic object.

2. A system as recited in claim 1, including a graphic accelerator graphic device object.

3. A system as recited in claim 1, including a frame buffer graphic device object.

4. A system as recited in claim 1, including a page description language graphic device object.

5. A system as recited in claim 1, including a vector engine graphic device object.

6. A system as recited in claim 1, wherein the grafport object, the graphic device object and the graphic object are polymorphic.

7. A system as recited in claim 1, wherein the grafport object, the graphic device object and the graphic object are fully extensible.

8. A system as recited in claim 1, including a modeling layer in the graphic object.

9. A system as recited in claim 8, including a geometric object and a graphic attribute object in the modeling layer.

10. A system as recited in claim 1, wherein the geometric object includes geometry for the graphic information.

11. A system as recited in claim 1, wherein the graphic device objects include displays, printers and plotters.

12. A method for graphic processing in an object-oriented operating system resident on a computer with a processor, a storage attached to and under the control of the processor and a graphic device attached to and under the control of the processor, comprising the steps of:

(a) building a modeling layer object in the storage;

(b) generating calls from the modeling layer object to grafport object using a predefined set of graphic primitives;

(c) capturing state information and rendering information at the grafport object; and

(d) passing the state information and the rendering information to a graphic device object for output on the graphic device.

13. The method as recited in claim 12, including state information with transform, appearance and clipping information.

14. The method as recited in claim 12, wherein the graphic device is a software or a hardware graphic processor.

15. An apparatus for graphic processing, comprising:

(a) a processor,

(b) a storage attached to and under the control of the processor;

(c) a graphic device attached to and under the control of the processor;

(d) a modeling layer object in the storage;

(e) a grafport object in the storage;

(f) means for generating calls from the modeling layer object to the grafport object using a predefined set of graphic primitives;

(g) means for capturing state information and rendering information at the grafport object; and

(h) means for passing the state information and the rendering information to a graphic device object for output on the graphic device.

16. The apparatus as recited in claim 15, wherein the state information includes transform, appearance and clipping information.

17. The apparatus as recited in claim 15, wherein the graphic device is a vector engine.

18. The apparatus as recited in claim 15, wherein the graphic device is a graphic accelerator.

19. The apparatus as recited in claim 15, wherein the graphic device is a frame buffer.

20. The apparatus as recited in claim 15, wherein the graphic device is a plotter.

21. The apparatus as recited in claim 15, wherein the graphic device is a printer.

22. The apparatus as recited in claim 15, wherein the graphic device is a display.

23. The apparatus as recited in claim 15, wherein the graphic device is a postscript processor.

24. The apparatus as recited in claim 15, wherein the modeling layer object includes at least one geometric object and at least one graphic attribute object.

25. The apparatus as recited in claim 15, wherein an object includes a method and data.

26. The apparatus as recited in claim 25, wherein the object is polymorphic and extensible.

* * * * *