Dominique Godard
Daniel Pilost

A 2400-Bit/s Microprocessor-Based Modem

This paper describes the main characteristics of a new microprocessor-implemented 2400-bit/s data modem, the IBM 3863. In addition to the execution of signal processing tasks, the microprocessor provides a variety of other significant functions such as diagnostics and aids in network problem determination. The lack of hardware multiplication capability imposes certain constraints in the design of the signal processing algorithms. The analytical approach and computational techniques, based on the processing of signals in polar coordinates, which are used to circumvent these constraints, are described. It is shown in particular that timing phase control, carrier recovery, and adaptive equalization can be achieved at the receiver by processing only the phase of the sampled signal. Additionally, experimental results are presented which demonstrate the superiority of this design over conventional coherent demodulators.

Introduction

Application of digital signal processing techniques in the design of data communication equipment has been intensively investigated in recent years, resulting in improved reliability and performance of communication systems. Most voiceband data modems are now microprocessor implemented, making it possible to utilize recent advances in signal theory which were not previously practical to implement.

Such microprocessor-based designs are of particular interest in high-speed modems (4800 bits/s and above) where severe channel distortions and impairments have to be compensated for, and in multipoint networks where fast start-up is of paramount importance and cannot be readily achieved with algorithms or circuits used for steady-state processing.

Another significant feature of microprocessor modems is the high flexibility possible through program control. This enables modems, through a set of microcoded routines, to contribute to communication network management by providing a variety of functions such as information on line quality, performance of self diagnostics, and the provision of automatic switching to and from full and fall-back speeds. These additional functions alone, which are implemented in the IBM 386X family of modems and are described in detail elsewhere [1], are sufficient to justify the use of a microprocessor in the IBM 3863 2400-bit/s data modem.

The general aspects of microprocessor modem implementations are well covered in the communications literature [2-4]. The algorithms and procedures that are reported commonly apply to data rates equal to or greater than 4800 bits/s, where adaptive equalization, precise timing, and carrier phase tracking are of crucial importance for good performance [5]. A microprocessor without multiplication capability is used in [3] to handle all the functions of a 4800-bit/s modem, and processing cycle reduction is obtained with the aid of decision-feedback equalization. In [2] and [4], LSI processors designed for performing specialized digital filtering operations are described.

At the relatively low data rate of 2400 bits/s, line distortions and impairments are usually not severe enough to require the implementation of sophisticated signal processing techniques, which explains why most 2400-bit/s modems are only equipped with *compromise*, possibly manually selectable, equalizers.

Copyright 1981 by International Business Machines Corporation. Copying is permitted without payment of royalty provided that (1) each reproduction is done without alteration and (2) the *Journal* reference and IBM copyright notice are included on the first page. The title and abstract may be used without further permission in computer-based and other information-service systems. Permission to *republish* other excerpts should be obtained from the Editor.

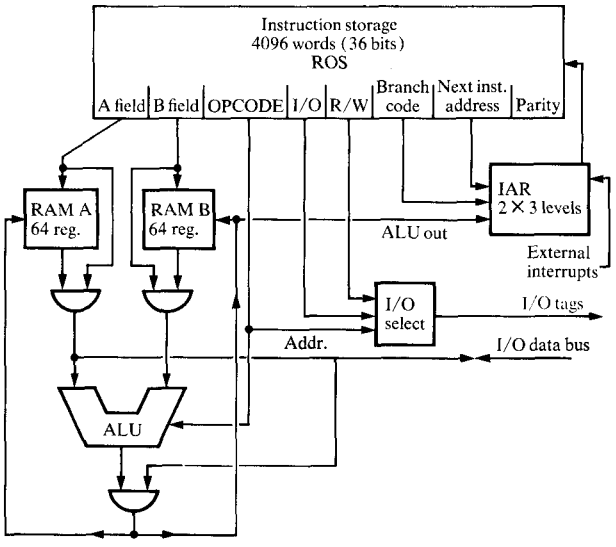

Figure 1 General microprocessor architecture.

In the design approach described in this paper we took the opportunity of having a microprocessor, although of limited capability, to enhance tolerance to distortions and to reach a level of performance superior to that of conventional coherent demodulators. This processor is not specialized to digital filtering applications, and no hardware multiplier is present, which imposes some constraints on algorithm design. However, it has a rather efficient instruction set and a special interrupt mechanism which allows the handling of independent and asynchronous tasks on its various interrupt levels.

In the following sections we shall describe the major functional characteristics of, and the processing algorithms used in, the microprocessor-based IBM 3863 data modem. The main characteristics of the microprocessor used in implementing this modem are presented in the next section. This is followed by a brief presentation of the transmitter implementation (third section), and a description of the approach taken for processing signals at the receiver (fourth section). Although, in conventional approaches, quadrature-amplitude-modulated signals are processed in cartesian coordinates, the only information needed in this approach for realizing the functions of clock-phase control, adaptive equalization, and carrier recovery is the phases of signal samples. This eliminates the need for automatic gain control and allows all the receiver functions to be performed without any multiplication. These aspects are discussed in the fifth section. Experimental results and conclusions are then presented in the final section.

The general-purpose microprocessor

The IBM 386X family of products comprises three modems (IBM 3863, 3864, and 3865) operating at 2400, 4800, and 9600 bits/s, respectively. The designs of these modems are based on several common building blocks, one of them being a *general-purpose microprocessor* (GPM) implemented in MOSFET technology.

In the IBM 3864 modem, the GPM is supported by a *digital filtering microprocessor* (DFM) which is specialized in the computation of multiplications with accumulation of results, as required in digital filtering applications. The GPM is utilized for performing transmitter functions, modem monitoring, and certain receiver tasks which do not involve multiplications, such as decision taking, descrambling, differential decoding, and table look-up operations.

The IBM 3865 9600-bit/s modem contains two GPMs, one of which is reserved for transmitter tasks, as well as a more powerful version of the DFM.

Mainly for reasons of economy, the design objective for the IBM 3863 was to perform all required receiver and transmitter functions in one GPM, the general architecture of which is shown in Fig. 1.

The processor is provided with a relatively efficient instruction set and data flow. The operations of operand fetching, execution, result saving, and conditional-branch-on-result are completed within a single instruction, and the average instruction time is 1.2 μs. The program, stored in a 4K-word (16K-byte) *read-only storage* (ROS), is interrupt driven and operates on six priority levels with no latency time between interrupts.

Basically, the microprocessor may be viewed as consisting of two interleaved machines, one having even-numbered interrupts and the other odd, with both machines sharing the same *arithmetic/logic unit* (ALU) and *random access memroy* (RAM) registers (2 × 64 8-bit registers). For every alternate cycle, the processor executes an instruction from the highest priority level 0, 2, or 4; and for every interleaved cycle, it executes an instruction from the highest priority interrupt level 1, 3, or 5. Thus, interrupt level 0 for the "even" machine has the same priority as level 1 for the "odd" machine, and so on.

This interleaving arrangement gives the required flexibility of running, asynchronously on the same processor, transmitter and receiver functions executed on even and odd machines respectively. Independently from real-time signal processing, low-priority tasks such as monitoring and machine check are also run on the even side of the

microprocessor. The activation of an *instruction address register* (IAR) is caused either by external interrupt signals (D/A converter clock on level 0 and A/D converter on level 1) or by the program interrupt request register.

The ALU provides 8-bit addition and subtraction with a carry signal activated when a carry overflows from the most significant bit. The carry from a given operation can be included in a subsequent add or subtract operation, thus allowing for computation in double precision if required. Table look-up and logical operations such as AND, XOR, and shift left and right are also provided.

Transmitter implementation

As mentioned previously, the GPM is used in 386X modems for the computation of transmitted signal samples. Through efficient use of the ROS, this can be accomplished with no multiplication required by using the digital echo modulation technique [6, 7].

In the case of the IBM 3863 modem, the ratio between the carrier frequency and the symbol rate makes the transmitter implementation particularly simple. Conventionally, the modulation scheme employed is four-phase phase-shift keying (PSK) modulation. The signaling speed is 1200 baud with the carrier frequency located at 1800 Hz. Denoting the signal element by $h(t)$ and combining in-phase and quadrature components into complex-valued signals, the transmitted waveform $x(t)$ can be expressed as

$$x(t) = \text{Re} \sum_n \exp j\varphi_n h(t - nT) \exp j2\pi f_0 t, \quad (1)$$

where T denotes the symbol interval, φ_n are the transmitted phases taken from the four-phase constellation, and f_0 is the carrier frequency. It is easily verified that (1) may be put into the form

$$x(t) = \text{Re} \sum_n (-1)^n \exp j\varphi_n g(t - nT), \quad (2)$$

where $g(t)$ is a complex passband signal element defined by

$$g(t) = h(t) \exp j2\pi f_0 t. \quad (3)$$

The digital-to-analog conversion rate is common to the three 386X modems and is equal to 9600 Hz, while the time spread of $g(t)$ is chosen to be four symbol intervals. Line signal samples $x(kT + lT/8)$, $l = 0, \cdots, 7$, can be simply generated by having all possible products Re $\{\exp j\varphi \cdot g(mT + lT/8)\}$, $m = -2, \cdots, 1$, stored in a table, and by combining them to realize the sampled version of (2) by means of additions and subtractions. One hundred and twenty-eight ROS words are used for this purpose.

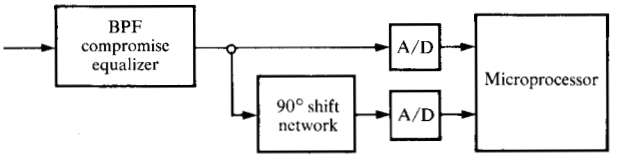

Figure 2 Analog signal processing.

Microprocessor flexibility also makes possible the implementation in the IBM 3863 of several options, such as compatibility with CCITT recommendation V26, the presence or absence of a scrambler, and the choice of particular start-up sequences for receiver training.

General receiver structure

In digitally implemented high-speed modems, the execution of receiver functions usually requires much more computing power than that of all the other tasks. In our case, however, due to the particular interleaved architecture of the microprocessor, only one-half of the computing power available is devoted to the receiver, which will now be described in more detail.

As previously mentioned, a major design objective was to enhance the tolerance of conventional coherent demodulators to channel distortions without resorting to algorithms requiring multiplication, the execution of which in our microprocessor would be extremely time-consuming. In particular, this precluded the use of an adaptive tapped delay line equalizer to cope with intersymbol interference, since in addition to requiring multiplications for computing the filter output signal and tap-gain increments, digital equalizers need automatic gain control (AGC) as well as complex demodulation due to operation in two-dimensional cartesian coordinates.

The three basic tasks to be continously carried out in a data receiver are timing phase control, carrier phase tracking, and equalization. Differential phase decoding and descrambling are straightforward procedures in digital modems and will not be addressed here. The start-up problems, although quite complex, have been solved; however, these will not be addressed in this paper because they are outside the scope of the main subject matter. Our approach was to achieve the three basic tasks mentioned by processing signals in polar instead of cartesian coordinates.

At the receiver, the analog data signal is first band-pass filtered (Fig. 2) for elimination of out-of-band noise. The band-pass filter (BPF) also achieves compromise equal-

ization. Because of the limited computing power available, the quadrature component of the received signal is not derived digitally but is generated by an analog 90-degree phase-shift network. Combined with the in-phase component, it forms the analytic signal [8]

$$y(t) = \sum_k s(t - kT) \exp j[2\pi(f_0 + \Delta f)t + \varphi_k] + w(t), \tag{4}$$

where $s(t)$ is the overall baseband equivalent impulse response, generally complex, $w(t)$ is complex filtered noise, and Δf represents frequency shift.

The real and imaginary parts of $y(t)$ are A/D-converted at the rate of 9600 samples per second. This rate is common to all IBM 386X modems and is convenient here for our timing phase control algorithm, which will be discussed later. The A/D converter produces 13-bit words in two's-complement form, denoted $y_1(mT/8)$ and $y_2(mT/8)$, providing enough precision to cover a 42-dB dynamic range. Included in the A/D conversion function is an automatic scaling which shifts y_1 and y_2 left until the two most significant bits of y_1 and/or y_2 are different. The number of shifts is averaged with different time constants, allowing the estimation of the signal energy and the detection of any sudden variations. The eight most significant bits of y_1 and y_2 are then transferred to the microprocessor which, through table look-up, determines the angles

$$\xi(mT/8) = \text{Arg}\,\{y_1(mT/8) + jy_2(mT/8)\}, \tag{5}$$

which comprise the only information processed for deriving timing phase control, carrier phase recovery, and equalization, as will be shown in the next section. Clearly, this algorithmic approach eliminates the need for automatic gain control.

Receiver processes

- *The equalization algorithm*

We first address the problem of compensating for channel linear distortions. From (4), the samples y_n at time instants $t = nT$, of $y(t)$, may be put into the form

$$y_n = \exp j[2\pi(f_0 + \Delta f)nT + \varphi_n] \times \left[s_0 + \sum_{p\neq 0} s_p \exp j(\varphi_{n-p} - \varphi_n)\right] + w_n, \tag{6}$$

where $\{s_p\}$ are samples of the complex pulse $s(t)$ at the rate $1/T$ Hz. The phase ξ_n observed at time $t = nT$ may be then written as

$$\xi_n = \varphi_n + n\pi + 2\pi\Delta f nT + \Delta\varphi_n + \beta_n, \tag{7}$$

where

$$\Delta\varphi_n = \text{Arg}\left\{s_0 + \sum_{p\neq 0} s_p \exp j(\varphi_{n-p} - \varphi_n)\right\} \tag{8}$$

is the phase error due to intersymbol interference, and β_n is a noise term. This phase error $\Delta\varphi_n$ is therefore a function of the phase-shift sequence $\{\varphi_{n-p} - \varphi_n\}$, $p = -\infty, \cdots, \infty$, which we denote

$$\Delta\varphi_n = \Phi(\varphi_\infty - \varphi_n, \cdots, \varphi_{n+1} - \varphi_n, \varphi_{n-1} - \varphi_n, \cdots, \varphi_{-\infty} - \varphi_n). \tag{9}$$

When operating at 1200 baud, intersymbol interference spreads only over a few symbol intervals. By analyzing a recent survey of voiceband channels [9], it was found that a good compromise between performance and complexity was to consider that the first leading and trailing echoes s_{-1} and s_1 are dominant, and also that the second trailing echo s_2 is small enough to allow approximating (9) by

$$\Delta\varphi_n \approx \varepsilon(\varphi_{n+1} - \varphi_n, \varphi_{n-1} - \varphi_n) + \theta(\varphi_{n-2} - \varphi_n) + \beta'_n, \tag{10}$$

where β'_n is considered as additional noise. In (10), ε and θ, which will be termed "phase echoes," can take on 16 and 4 possible values, respectively, depending on the data sequence transmitted.

Note that demodulation in this case simply consists of subtracting from ξ_n an estimate $\varphi_c(n)$ of the carrier phase (the adjustment of which will be described later), yielding

$$\psi_n = \xi_n - \varphi_c(n) \approx \varphi_n + \varepsilon(\varphi_{n+1} - \varphi_n, \varphi_{n-1} - \varphi_n) + \theta(\varphi_{n-2} - \varphi_n) + \gamma_n, \tag{11}$$

where γ_n is random noise.

Now, let us first assume that all possible values of phase echoes ε and θ are known. Under the approximation that γ_n is white and gaussian, the problem of determining the most probable data sequence, given the received phases $\{\psi_n\}$, is that of finding the sequence $\{\hat{\varphi}_n\}$, which minimizes the squared error:

$$D^2 = \sum_n [\psi_n - \hat{\varphi}_n - \varepsilon(\hat{\varphi}_{n+1} - \hat{\varphi}_n, \hat{\varphi}_{n-1} - \hat{\varphi}_n) - \theta(\hat{\varphi}_{n-2} - \hat{\varphi}_n)]^2. \tag{12}$$

Theoretically, this maximum likelihood sequence estimation problem could be solved using the Viterbi algorithm [10, 11] but in our environment that would require too much computing power. Instead, the problem may be considerably simplified under the following assumptions:

- At time $t = nT$, past decisions $\hat{\varphi}_{n-1}$ and $\hat{\varphi}_{n-2}$ are supposed correct, as is done in decision-feedback equalizers [12].
- The binary eye is open, which means that phase errors $\Delta\varphi_n$ due to intersymbol interference have a magnitude smaller than $\pi/2$.

Then, denoting by φ^i and φ^j the two closest neighbors of ψ_n in the four-phase constellation, the two possible decisions are $\hat{\varphi}_n = \varphi^i$ and $\hat{\varphi}_n = \varphi^j$. Similarly, we denote by φ^k and φ^l the two closest neighbors of ψ_{n+1}. Now, only four sequences are likely to be transmitted, and for each of these sequences an error signal can be computed. The decision $\hat{\varphi}_n$ is then taken as follows:

- First compute the four error signals:

$$
\begin{aligned}
e_n^{ki} &= \psi_n - \varphi^i - \varepsilon(\varphi^k - \varphi^i, \hat{\varphi}_{n-1} - \varphi^i) - \theta(\hat{\varphi}_{n-2} - \varphi^i),\\
e_n^{kj} &= \psi_n - \varphi^j - \varepsilon(\varphi^k - \varphi^j, \hat{\varphi}_{n-1} - \varphi^j) - \theta(\hat{\varphi}_{n-2} - \varphi^j),\\
e_n^{li} &= \psi_n - \varphi^i - \varepsilon(\varphi^l - \varphi^i, \hat{\varphi}_{n-1} - \varphi^i) - \theta(\hat{\varphi}_{n-2} - \varphi^i),\\
e_n^{lj} &= \psi_n - \varphi^j - \varepsilon(\varphi^l - \varphi^j, \hat{\varphi}_{n-1} - \varphi^j) - \theta(\hat{\varphi}_{n-2} - \varphi^j);
\end{aligned}
\tag{13}
$$

- then determine the error $e_{\min}$ having the smallest magnitude;
- and, finally, decide $\hat{\varphi}_n = \varphi^i$ if $e_{\min} = e_n^{ki}$ or e_n^{li}, and $\hat{\varphi}_n = \varphi^j$ otherwise.

Obviously, phase echoes ε and θ are initially unknown at the receiver and must be determined adaptively. The criterion for their adjustment is the minimization of the mean-squared phase error.

Let $\hat{\varepsilon}_n$ and $\hat{\theta}_n$ estimates of phase echoes at time $t = nT$. These estimates are used in (13), instead of actual values, for computing the four possible error signals. Since the decision on φ_{n+1} has not yet been taken, updating takes place with one baud interval delay. Errors determined at the previous sampling instant have been stored in memory, and phase echoes corresponding to the detected phase shifts, *i.e.*, $\hat{\varepsilon}_n(\hat{\varphi}_n - \hat{\varphi}_{n-1}, \hat{\varphi}_{n-2} - \hat{\varphi}_{n-1})$ and $\hat{\theta}_n(\hat{\varphi}_{n-3} - \hat{\varphi}_{n-1})$ are updated using the appropriate error. Let, for instance, $\hat{\varphi}_n = \varphi^i$, $\hat{\varphi}_{n-1} = \varphi^p$, $\hat{\varphi}_{n-2} = \varphi^r$ and $\hat{\varphi}_{n-3} = \varphi^s$. Then, phase-echo updating is defined by the stochastic approximation algorithms

$$
\begin{aligned}
\hat{\varepsilon}_{n+1}(\varphi^i - \varphi^p, \varphi^r - \varphi^p) &= \hat{\varepsilon}_n(\varphi^i - \varphi^p, \varphi^r - \varphi^p) + \lambda e_{n-1}^{ip},\\
\hat{\theta}_{n+1}(\varphi^s - \varphi^p) &= \hat{\theta}_n(\varphi^s - \varphi^p) + \lambda e_{n-1}^{ip},
\end{aligned}
\tag{14}
$$

where λ is a small positive step-size parameter. At each symbol interval, one particular ε out of sixteen and one θ out of four are updated.

Weighting errors by λ in (14) can be realized by shift-right operations, so that the whole equalization and decision processes involve only additions, subtractions, and simple logical operations.

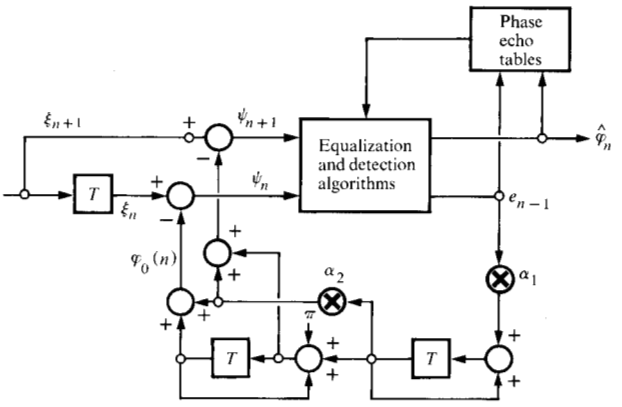

Figure 3 Carrier tracking loop.

Computer simulations have shown that, with the receiver operating in decision-directed mode, phase echoes converge to their steady-state values within about 600 symbol intervals, even for initial symbol error probabilities as high as 10^{-1}.

- *Carrier recovery*

As mentioned previously, demodulation consists simply of subtracting from the received phase ξ_n an estimate $\varphi_c(n)$ of the carrier phase. In order to lock on the zero-mean steady-state phase error in the presence of frequency offset, this estimate is provided by a second-order phase-locked loop driven by phase errors determined in the detection process discussed previously. This means that at time $t = nT$, as for phase echo updating, only the phase error e_{n-1} at the previous sampling instant is available. As indicated in Fig. 3, the reference carrier phase $\varphi_c(n)$ evolves according to

$$
\varphi_c(n+1) = \varphi_c(n) + \pi + \alpha_1\alpha_2 e_{n-1} + \alpha_1 \sum_{k=2}^{\infty} e_{n-k}. \tag{15}
$$

The closed-loop transfer function of the "phase filter" of Fig. 3 is easily shown to be given by

$$
G(z) = \frac{(1 - z^{-1})^2}{1 - 2z^{-1} + (1 + \alpha_1\alpha_2)z^{-2} + \alpha_1(1 - \alpha_2)z^{-3}}, \tag{16}
$$

and to be stable for loop gains α_1 and α_2 equal to 1/64 and 16, respectively (powers of two are selected to avoid multiplications).

At this point, several remarks should be made. First, the phase echoes and the carrier tracking loop are both able to compensate for a constant phase offset. Trunca-

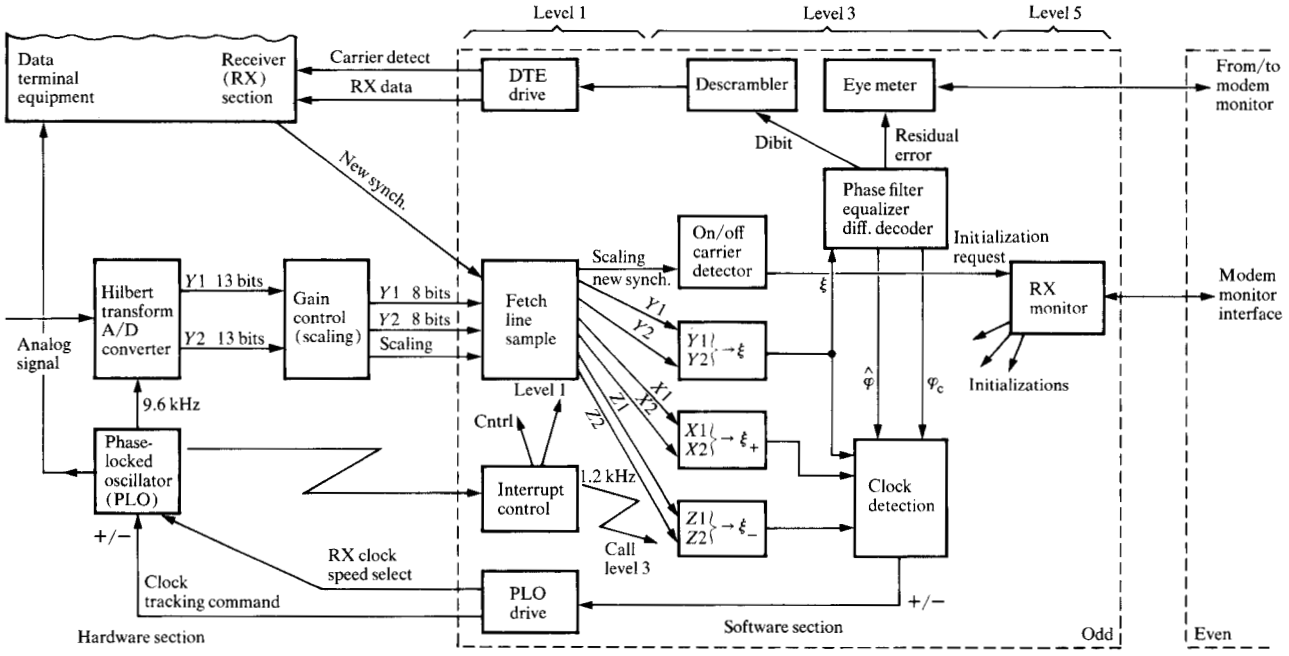

Figure 4 Receiver block diagram.

tion errors due to finite-length arithmetic may then lead to some kind of instability. This problem is avoided by keeping fixed to zero the phase echoes corresponding to the all-zero phase shift sequence.

It should also be noted that, in steady state, the content of the first integrator approximates $2\pi\Delta fT$, a value which in most cases is small. This requires that the carrier tracking function be executed in double precision.

Finally, one observes from (15) that the phase filter is capable of tracking phase jitter if the product $\alpha_1\alpha_2$ is sufficiently high [13]. However, this makes the carrier recovery loop more sensitive to noise and residual intersymbol interference. In addition to stability considerations, the gains α_1 and α_2 were also chosen to offer a good compromise between phase jitter tracking capability and tolerance to noise.

• *Timing phase control*

The criterion for controlling the receiver clock phase is that of maximum eye opening; *i.e.*, it is desired to sample the received signal at time instances when the phase error $\Delta\varphi_n$ due to intersymbol interference has minimum variance.

Let τ be the symbol clock phase; *i.e.*, signals are sampled at receiver input at instances $\tau + kT/8$. As pointed out by Kobayashi [14], τ should evolve according to the gradient algorithm

$$\tau(m+1) = \tau(m) - \mu \frac{\partial}{\partial \tau}[E(\Delta\varphi_n)^2]_{\tau=\tau(m)}, \tag{17}$$

where E indicates expectation and μ is a positive step-size parameter. As usually done in equalizer adjustment algorithms for minimizing mean square error [15], one can drop the expectation term and transform (17) into the stochastic approximation algorithm

$$\tau(n+1) = \tau(n) - \mu'\Delta\varphi_n\left[\frac{\partial}{\partial\tau}\Delta\varphi_n\right]_{\tau=\tau(n)}, \qquad \mu' > 0, \tag{18}$$

where $\Delta\varphi_n$ is the phase error at instant $nT + \tau(n)$.

In a digital processor, the time derivative

$$\frac{\partial}{\partial\tau}\Delta\varphi_n$$

is not readily available, but can be approximated by

$$\left[\frac{\partial}{\partial\tau}\Delta\varphi_n\right]_{\tau=\tau(n)} \approx \frac{4}{T}[\Delta\varphi(nT + \tau(n) + T/8) - \Delta\varphi(nT + \tau(n) - T/8)]. \tag{19}$$

From (6) and (7), one obtains

$$\left[\frac{\partial}{\partial\tau}\Delta\varphi_n\right]_{\tau=\tau(n)} \approx \frac{4}{T}\left[\xi_+ - \xi_- + \frac{3\pi}{4}\right], \tag{20}$$

where $\xi_+ = \xi[nT + \tau(n) + T/8]$ and $\xi_- = \xi[nT + \tau(n) - T/8]$.

On the other hand, one has

$$\Delta\varphi_n = \xi[nT + \tau(n)] - \varphi_c(n) - \hat{\varphi}_n. \quad (21)$$

Finally, in order to avoid multiplications, only the product of the signs of (20) and (21) is used to control the phase of a phase-locked oscillator (PLO). This leads to a binary clock control scheme of the form

$$\tau(n + 1) = \tau(n) \pm \Delta\tau,$$

where $\Delta\tau$ is chosen in the order of $10^{-4}T$, which is enough to compensate for drift between transmitter and receiver clocks.

The receiver implementation architecture is summarized in the block diagram of Fig. 4.

Experimental results and conclusions

The effectiveness of the signal processing algorithms described in this paper, particularly that of the equalization and detection technique, was tested experimentally. Bit error rates obtained when transmitting over channels defined by the group delay distortions given in Fig. 5 were measured with two different versions of the modem (with amplitude distortion produced by the compromise equalizer). In the first version, phase echoes ε and θ were kept fixed at zero, and therefore no equalization was achieved, so that the receiver behaved as a conventional coherent demodulator. In the second version, the equalization and detection processes discussed previously (in the section on "Receiver Processes") were in effect. The results of these tests are given in Fig. 6, where the signal-to-noise ratio (SNR) is measured through a *C-message* filter [16]. The results illustrated clearly indicate the improvements obtained with the phase-echo approach. For channel 1, the coherent demodulator requires an SNR greater than 32 dB for achieving 10^{-5} bit error rate, and it does not operate at all on channel 2. With the IBM 3863 modem, 10^{-5} bit error rates are obtained at SNRs equal to 10.7 dB for channel 1 and 13.5 dB for channel 2. Theoretically, assuming perfect filtering of out-of-band noise and ideal channel characteristics, 10^{-5} bit error rate should be obtained for an SNR (still measured through a C-message filter) equal to 8.8 dB. The performance of the coherent demodulator for channel 3 may be considered as satisfactory. The gain in SNR obtained with the phase echo approach is again very significant.

This paper has presented the main architectural characteristics of the IBM 3863 2400-bit/s data modem, the design of which was based on a general-purpose micro-

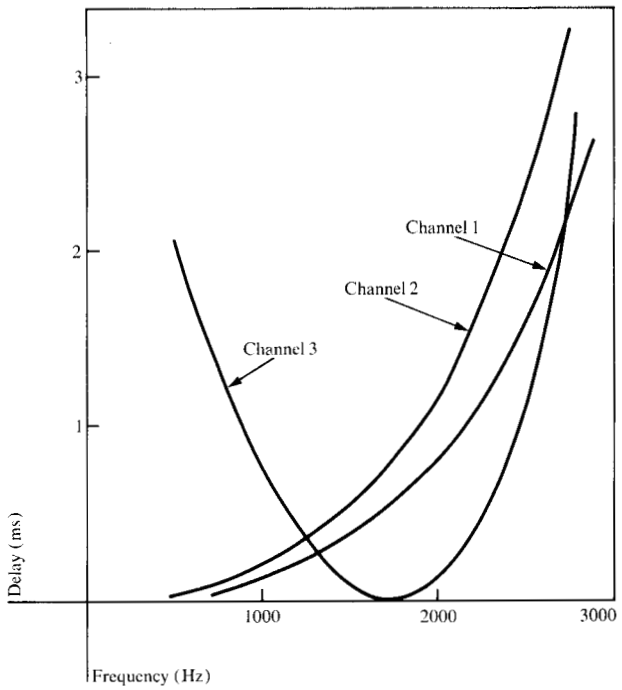

Figure 5 Group delay distortions.

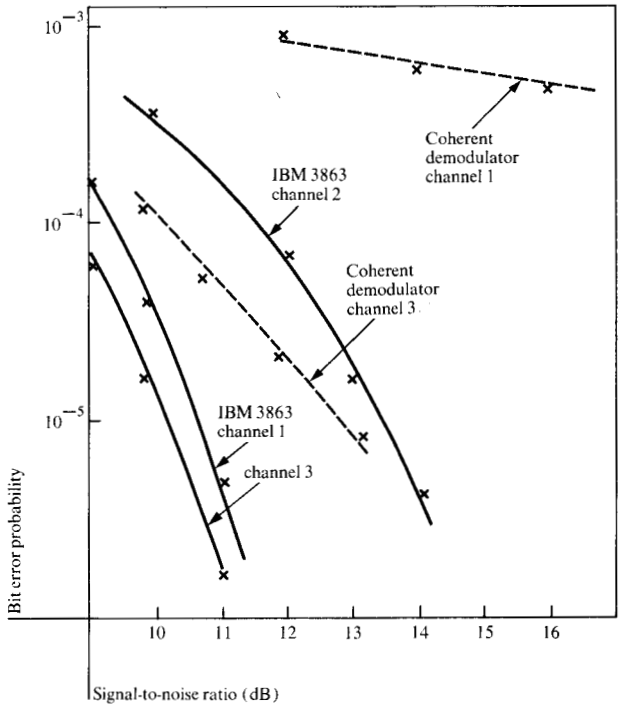

Figure 6 Experimental error rates.

processor. New algorithms for adaptive equalization, carrier phase tracking, and clock phase control, tailored for ease of implementation in such a processor, have been described. These algorithms process only signal phases for minimizing the mean-squared phase error, and they can be implemented without any multiplication. Experimental results show that the particular signal processing approach described in this paper allows for reliable data transmission at 2400 bits/s over channels which could not be realized by a coherent demodulator without the use of manually selectable equalizers.

References

1. Simon Huon and Robert Smith, "Network Problem-Determination Aids in Microprocessor-Based Modems," *IBM J. Res. Develop.* **25,** 3-16 (1981, this issue).
2. H. L. Logan and G. D. Forney, "A MOS/LSI Multiple-Configuration 9600-Bps Data Modem," *Proceedings of the International Conference on Communications*, Philadelphia, 1976, pp. 48.7-48.12.
3. K. Watanabe, K. Inone, and Y. Sato, "A 4800 bit/s Microprocessor Data Modem," *IEEE Trans. Commun.* **COM-26,** 493-498 (1978).
4. K. Murano, S. Unagami, and T. Tsuda, "LSI Processor for Digital Signal Processing and its Application to 4800 bit/s Modem," *IEEE Trans. Commun.* **COM-26,** 499-506 (1978).
5. D. N. Godard, "Passband Timing Recovery in an All-Digital Modem Receiver," *IEEE Trans. Commun.* **COM-26,** 367-376 (1970).
6. A. Croisier and J. M. Pierret, "The Digital Echo Modulation," *IEEE Trans. Commun. Technol.* **COM-18,** 367-376 (1970).
7. P. J. Van Gerwen, N. A. M. Verhoeckx, H. A. Van Essen, and F. A. M. Snijders, "Microprocessor Implementation of High-Speed Data Modems," *IEEE Trans. Commun.* **COM-25,** 238-250 (1977).
8. L. E. Franks, *Signal Theory*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1969, pp. 79-82.
9. F. P. Duffy and T. W. Thatcher, "Analog Transmission Performance on the Switched Telecommunications Network," *Bell Syst. Tech. J.* **50,** 1311-1347 (1971).
10. G. D. Forney, "Maximum Likelihood Sequence Estimation of Digital Sequences in the Presence of Intersymbol Interference," *IEEE Trans. Info. Theory* **IT-18,** 363-378 (1972).
11. G. Ungerboeck, "Adaptive Maximum Likelihood Receiver for Carrier-Modulated Data Transmission Systems," *IEEE Trans. Commun.* **COM-22,** 624-636 (1974).
12. M. E. Austin, "Decision-Feedback Equalization for Digital Communication over Dispersive Channels," *Technical Report 461*, Massachusetts Institute of Technology, Cambridge, MA, 1967.
13. D. D. Falconer, "Jointly Adaptive Equalization and Carrier Recovery in Two-Dimensional Digital Communication Systems," *Bell Syst. Tech. J.* **55,** 317-334 (1976).
14. H. Kobayashi, "Simultaneous Adaptive Estimation and Decision Algorithms for Carrier Modulated Data Transmission Systems," *IEEE Trans. Commun. Technol.* **COM-19,** 268 (1971).
15. G. Ungerboeck, "Theory on the Speed of Convergence in Adaptive Equalizers for Digital Communication," *IBM J. Res. Develop.* **16,** 546 (1972).
16. "Data Communications Using Voice-Band Private Line Channels," *Bell Syst. Tech. Ref.*, October 1973, pp. 23-24.

Received November 27, 1979; revised April 29, 1980

The authors are located at the IBM Laboratory (Centre d'Etudes et Recherches) at La Gaude, France 06610.