Design and Evaluation of Chase Decoder Architecture for Medium Capacity TDMA Satellite Systems

Hélio César A.S. Salles and Walter da Cunha Borelli

This paper describes a digital machine architecture suitable for soft-decision decoding of binary linear block error-correcting codes, based on Chase's algorithm II. Specifically, it is presented the hardware implementation for the extended Golay (24, 12, 8) block code in a printed circuit board used in a satellite multi-frequency TDMA data transmission system, called SAMSAT, which was developed at CPqD-TELEBRÁS. This architecture may be implemented using off-the-shelf digital ICs and still reach up to 1 Mbit/s transmission rates, in a variety of digital satellite transmission applications. Cost and chip count of final circuitry are highly competitive within the class of performance to which this forward-error-correction (FEC) technique pertains.

1. Introduction

It is well known that, for any (n, k, d) binary linear block error-correcting code, conventional algebraic decoding techniques have INT [(d-1)/2] digits error correction capability, with n being the number of digits of the output block, k the number of bits of the input block or message, d the minimum Hamming distance of the code and INT [x] denoting the greatest integer number less than or equal to x.

In 1972, Chase [1] proposed a class of decoding algorithms using channel measurement information (soft-decision) in conjunction with any conventional algebraic (i.e., hard-decision) decoder, which can correct many patterns of up to (d-1) errors, i.e., practically a two-fold increase over the hard-decision decoders. Chase also showed that his algorithms are equivalent (or an approximation) to correlation block decoders, thus being maximum-likelihood and then optimum for binary linear block codes in an additive white Gaussian noise (AWGN) channel. He also simulated his algorithm II for the extended Golay code, denoted here as EGC (24, 12, 8), in an AWGN channel and found an asymptotic coding gain of 6dB over uncoded transmission and a 4dB improvement at an error probability $P_e=10^{-5}$. This is significant when compared to the same numbers for a binary hard-decision decoder (3 dB and 2

H.C.A.S. Salles is a Researcher at CPqD – TELEBRÁS, Caixa Postal 1579, 13100, Campinas. SP.
W.C. Borelli is a Professor at Faculdade de Engenharia Elétrica, UNICAMP, Caixa Postal 6101, 13081, Campinas, SP.

dB, respectively) and even when compared to typical Viterbi soft-decision decoders for short constraint length convolutional codes of the same rate (1/2) (e.g., Shenoy and Johnson [2]).

Clark and Cain [3] and Viterbi and Omura [4] obtained by simulation and calculation (respectively) that the degradation introduced by any quantization greater than or equal to 8 levels (3 bit), with respect to the non-quantized case, is less than or equal to 0.22 dB, regardless of the quantization scheme being used (uniform, optimum metric spacing, etc). Therefore, an 8 level quantization scheme can be used for soft-decision decoders with negligible degradation on the final performance. This 8 level quantization will be assumed for the following sections. Finally, Pessoa [5] simulated the sensitivity of Chase's algorithm II, associated to the EGC (24, 12, 8) and an 8 level uniform quantizer, with respect to automatic gain control effects. It has been found that, for ± 6 dB variations at the quantizer input, with respect to the optimum level, the corresponding degradation in the decoder performance is only ± 0.3 dB.

Other important aspects related to block decoders are: they do not have any source of degradation other than the levels of quantization, have no error propagation over subsequent blocks, have a fixed amount of delay from input to output and are very suitable for burst communications. On the contrary, soft-decision decoders for tree codes have degradations associated to the path memory length (both for Viterbi and sequential decoding), variable decoding delays (only for sequential decoders) and variable error propagation over the decoded bits.

Although well known for several years, system and circuit designers have given Chase decoders a secondary role in the error-control-coding field, driving all their efforts to Viterbi decoders. One exception to this trend can be found in Hackett [6], where a variation of Chase's algorithm, for low-speed applications, is implemented in software, using a general microprocessor.

The architecture proposed in the present paper may be implemented using standard digital ICs, available in the market (HCMOS series, TTL compatible; LS-TTL; S-TTL PROM's; CMOS EPROM's and NMOS RAM's). The resulting hardware still can reach transmission rates as high as 1 Mbit/s, with maximum operation speed being limited only by RAM's access time. Cost, complexity and chip count of final circuitry are highly competitive within the class of performance to which this FEC technique pertains. In order to achieve such results, we had to deal with some recent digital design techniques and concepts such as pipeline or parallel processing, multi-processing hardware synchronization, self-timed circuits, etc.

We start by presenting, in Section 2, the division of Chase's algorithm II in several parallel processes suitable for relatively simple and economical digital circuit synthesis. Then we thoroughly discuss the implementation of each section, detailing the necessary clocks, the synchronization problems and the interrelationship among the several processing units associated with those parallel processes.

In Section 3 we show, as the main application of the derived method and its associated hardware, the use of this architecture to decode the EGC (24, 12, 8). The resulting circuitry is part of the TDMA/QPSK modem at 1068 kbit/s trasmission rate for a satellite multi-frequency TDMA data transmission system, the SAMSAT, which will be briefly presented. Other possible satellite communications applications are also discussed.

Then we present the results obtained with the circuit that has been built on a double-face printed circuit board (PCB) with 124 standard ICs. We show how it complies with the system engineering specifications: for an input bit-error rate (BER) of 10^{-2}, output measured BER was 2.10^{-7} and with 4.10^{-3}, it was 1.4 10^{-9}, leading to a net gain of approximately 4.7 dB. Final goals of circuit design engineering are also achieved: a single double-face PCB with 115 square inches, powered by a single +5V supply, with a consumption of 1.3A and a decoding delay of 26 symbols, operating at rates of 1 Mbit/s could be implemented with low power needs, by using a relatively simple, modern and cost-effective architecture that still gives high performance at low SNR's (signal-to-noise ratios), i.e., low values of E_b/No (ratio between the energy per information bit and the one-sided noise power spectral donsity).

2. The Digital Architecture for Chase's Algorithm II Decoders

2.1 Chase's Algorithm II Review

Let y(t) be an antipodal binary signaling waveform received by a demodulator, corresponding to a noise corrupted waveform x(t) originally transmitted by a modulator in response to a codeword of n digits, denoted by $X = (x_1, x_2, \ldots, x_n)$, produced at the output of a binary linear block encoder whenever an information sequence of k bits, denoted by $A = (a_1, a_2, \ldots, a_k)$, is presented at its input. Let us assume that this bloc k encoder is based on a binary linear block code C(n, k, d). Then, if y(t) is quantized into Q levels prior to decoding, the Chase algorithm [1] for soft-decision decoding of binary linear block codes using channel measurement information can be described by a procedure consisting of the following steps:

(1) a quantized version of y(t) is available, sampled at the optimum sampling instants by a G-bit quantizer, such that, for each set of G bits associated with a single hard decision bit Y_i of the n-bit sequence $\mathbf{Y} = (y_1, y_2, \ldots, y_n)$:

a) there is 1 bit (the MSB or most significant bit) that represents the polarity (0 or 1) of the digit y_i;

b) there are (G-1) bits that represent a positive integer number in the range $[0, 2^{G-1}]$, carrying a reliability information of the received y_i and denoted by $\mathbf{r_i}$. Then there is a reliability vector $\mathbf{R} = (\mathbf{r_1}, \mathbf{r_2}, \ldots, \mathbf{r_n})$ associated with $\mathbf{Y} = (y_1, y_2, \ldots, y_n)$;

c) $Q = 2^G$ **(1)**

(2) make, on an a priori basis, polarity inversions on an arbitrary number F of digits in sequence **Y** (usually the least reliable ones), based on the generation of several test patterns T_j that are modulo-2 added to the received sequence Y, thus obtaining a new word $\mathbf{Y_j}$ as

$$Y_j = Y \oplus T_j \tag{2}$$

where $\oplus$ represents modulo-2 bit-by-bit addition;

(3) decode, with the aid of a conventional binary (hard-decision) decoder, the sequence Y_j – which is fed to this hard-decision decoder – and then find an associated error-pattern E_j;

(4) add E_j to T_j, obtaining Zt_j as a modulo-2 addition

$$Zt_j = E_j \oplus T_j \tag{3}$$

where Zt_j is an error pattern that carries both the binary decoded errors plus the possible a priori corrected errors via the polarity inversions;

(5) find the analog weight of Zt_j, Wa (Zt_j)

$$Wa(Zt_j) = \sum_{i=1}^{n} r_i Zt_{ji} \tag{4}$$

where r_i is the decimal representation of r_i (note that Wa $(\mathbf{Zt_j})$ is the sum of the reliabilities associated with the digits that are 1's in Zt_j);

(6) choose the pattern $\mathbf{Zt_j}$ with the minimum analog weight;

(7) if there are more test patterns, do steps 2 to 6 again; if there is none left, decode **C**, the codeword estimator, as

$$C = X_e = Y \oplus Zt_j \tag{5}$$

(8) transfer to the output the corresponding information bits associated to **C**.

Algorithms I, II, and III vary according to the test pattern set used to modify the received sequence. For Chase's algorithm II, the chosen test pattern set is the set of all test patterns that have any combination of 1's, which are located in the INT[d/2] positions of lowest confidence values r_i in the word **R**, including the all-zero pattern, that leads to the conventional hard-decision result.

Therefore, with Chase's algorithm II, we try to solve, via the test patterns, up to the F=INT[d/2] most probable errors, presumed to be the F least reliable digits of **Y**, letting the binary decoder solve up to t=INT[(d-1)/2] errors that are more difficult to detect by confidence values inspection. Operating in this way, the Chase's algorithm II decoder can handle up to

$$t^1 = t + F = (d-1) \tag{6}$$

error corrections, increasing considerably the error-correction capability of a code with minimum Hamming distance d. It should be observed that t^1 is not the error-correction capability of algorithm II because the decoder does not correct all the error patterns whose weight is less than or equal to (d-1). However, in a probabilistic way, it can correct the most likely error patterns whose weight is less than or equal do (d-1) and surely correct all error patterns whose weight is less than or equal to t, the error-correction capability of the inner hard-decision binary decoder.

Finally, the number N_e of elements T_j of the test pattern set is the number of possible subsequencies for the INT[d/2] least reliable positions in the sequence. Since the sequence is binary, then

$$N_e = 2^{INT[d/2]} \tag{7}$$

For the EGC (24, 12, 8), there are N_e = 16 test patterns.

2.2 A Novel Division of Chase's Algorithm II in Parallel Processes

The Chase algorithms, including algorithm II, have been proposed within a theoretical framework, as it can be inferred from the presented review, and they may be implemented by several different digital techniques. However, there are only few techniques that can be considered attractive from a practical point of view, for operation at relatively high data rates, with small power comsumption and reduced price.

To show this, we shall analyze some alternatives for decoding the EGC (24, 12, 8) at the desired rate (1068 kbit/s). The stationary decoding interval for a continuous transmission is one block and therefore the available processing time to make a decision over any test-pattern will be

$$T(\text{processing any } T_j) = (24/16).(1/1068.10^3) = 1.4\mu s \qquad \textbf{(8)}$$

This value is too low to accomplish all the necessary operations in order to find the analog weight of T_j with standard available microprocessors (even with an 80386 running at 16 MHz). Bit-slice processors and a RISC (Reduced Instruction Set Computers) architecture are discarded, mainly due to high power needs, EMC (electromagnetic compatibility) problems in a radio environment and economical reasons. On the same basis, a serial processing architecture using standard ICs must be discarded. A totally parallel architecture using standard ICs must be discarded too, due to its enormous chip count. Three alternatives remain to be analyzed. The first one is to use a DSP (Digital Signal Processor) based architecture. However, this solution must use several DSPs due to technical limitations of operating speed for DSPs currently available in the market. Then, this approach becomes expensive, although it seems to be a promising one in the near future, when DSPs are expected to speed up and to become cheaper than today. The second alternative is to make a custom IC design (ASICs), which is the way Viterbi decoders are currently implemented, but this is attractive only if a huge market is envisaged, for cost reduction due to large scale production, which is not our forecast for the potential Brazilian market.

Finally, the third alternative, that seems to be the most attractive one, is to find an architecture suitable for PCB design which is still applicable for ASIC design and which does not affect acceptable power and packing limitations imposed by the FEC function. This third alternative was the one chosen.

In order to explore such an alternative, the architecture presented in this work **transforms** Chase's algorithm II and treats it as a decoding macro-process which is divided in several distinct processes, each one suitable for hardware implementation. By careful selection, we can arrive at the following processes:

(1) store the received hard-decision sequence **Y** and the reliability vector **R**;

(2) find a "primary" syndrome **S=Y H'**, where **H'** is the transpose of the parity-check matrix **H**;

(3) find and store the addresses POS_i, $i \epsilon \{1, 2, \ldots, n\}$, of the INT[d/2] lowest confidence digits by inspection of the received r_i;

(4) generate the $2^{INT[d/2]}$ test patterns from the INT[d/2] POS_i;

(5) generate all associated "test-syndromes" ST_j using

$$ST_j = T_j \; H' \tag{9}$$

(6) add the "primary" syndrome to all "test-syndromes" to find the "modified syndromes"

$$\mathbf{SM_j} = S \oplus \mathbf{ST_j} \tag{10}$$

(7) find all the binary error patterns, represented here by E_j", with the aid of $\mathbf{SM_j}$ via a binary decoder, such as a look-up table, error trapping, etc, giving the addresses that are 1 in E_j";

(8) find the valid error-pattern E_j as

$$E_j = E_j \oplus T_j \tag{11}$$

by making the union of the sets of addresses, expurgating those appearing at both sets $\{\mathbf{E_j}"\}$ and $\{\mathbf{T_j}\}$;

(9) calculate all the $2^{INT[d/2]}$ analog wights $Wa(\mathbf{E_j})$, from the sum of all r_i addressed by those positions "i" that are 1 in $\mathbf{E_j}$;

(10) compare all $Wa\,(E_j)$ and select the E_j with the lowest analog weight;

(11) store both the min $[Wa(\mathbf{E_j})]$ and the correspondent $\mathbf{E_j}$;

(12) add this stored E to Y to obtain the codeword estimator **C**, correcting all the correctable errors in **Y**;

(13) send the information bits associated with Y; if the code is systematic, this is merely the transfer of the first k bits of **Y** to the output.

Now, in order to validate our idea, we shall prove that these processes together are equivalent to Chase's algorithm II and that they are very efficient for practical applications. This will be done by checking the following statements.

Statement 1

To work with "primary" and "test" syndrome addition is equivalent to work with received sequence and test pattern addition, and the former approach is preferable than the latter one.

Proof

We need to decode with binary decoding the sequence Y_j given in (2). All practical binary decoders need a syndrome to work with. Then we need to calculate

$$S_j' = Y_j H' = (Y \oplus T_j) H' = Y H' \oplus T_j \ H' = S \oplus ST_j = SM_j \qquad (12)$$

The syndrome addition needs only (n-k) adders while Y and T_j addition needs n adders.

Statement 2

To work with the addresses of the digits that are 1 in a binary vector with n digits is much more convenient than to work with the vector itself, provided n is big enough and the 1's are sparse.

Proof

(i) for binary decoding a binary linear block consider code C(n, k, d). The hard-decision decoder can correct up to t=INT[(d-1)/2] errors. The conventional decoder finds an error vector E" with n digits, then E" must be added to the received vector **Y** to produce the codeword estimator **C**. Then the number of necessary signal tracks L_1, which is also the number of circuit elements for each linear operation, is

$$L_1 = n \qquad (13)$$

However, if this decoder handles only the addresses of error-positions, since there are at most t correctable errors and since to address, in a binary form, a number ranging from 1 to n involves a number N_a which is

$$N_a = \begin{cases} g & \text{for } n = 2g \\ \text{INT}[(\log n)/(\log 2)] + 1 & \text{for any other } n \end{cases} \qquad (14)$$

there will be at most

$$L_2 = t\,N_a \qquad (15)$$

necessary signal tracks. Then, if d<<n, it is clearly wiser to work with addresses rather than the vector itself. For example, if t = 3, for any integer n greater than 9, we should prefer the addressed form of the vector.

(ii) analog weight calculation

Clearly there are at most (d-1) terms to add up when addresses are being used. On the contrary, if we use the error-vector itself, we must add all the products $r_i.E_{ji}$, since we do not know which digits will be 0's and which will be 1's.

(iii) valid error-pattern calculation

Note that n modulo-2 additions should be performed if a direct representation of vectors is used. If an addressed representation is available, it is merely a coincident address masking that is required.

Statement 3

The easiest generation of T_j is accomplished if addressing is used, because we can eliminate vector representation conversions. As far as algorithm II uses all possible subsequences for the least reliable positions in the sequence in order to generate the set $\{T_j\}$, we will always need to point out these positions by means of addressing. This is so even if we use a bit set-reset procedure to build a "true least reliable positions indicator vector", in case a direct representation form is being used.

Statement 4

To calculate the "test-syndrome" is simply to add the coresponding columns of the transpose of the parity-check matrix, addressed by the digits that are 1 in the test pattern T_j. Then, for algorithm II, it involves a maximum of INT[d/2] additions of INT[d/2] lines of the parity-check matrix **H**.

Based on this analysis, we can establish the decoding macro-process flowgraph shown in **Fig. 1**, where each process can be handled by a simple and economical hardware. It is thus defined a complete architecture, which may be used to decode, using Chase's algorithm II, any binary linear and preferably systematic code C(n,k,d).

If transmission is continuous, the decoder enabling signal ENDEC will always be 1 and if burst transmission is used, ENDEC will be pulsed and generated by the appropriate synchronization circuitry. If a simple coherent BPSK modem is used, the direct demodulated and quantized bit stream feeds the decoder. If coherent QPSK is used and a single encoder is provided, demodulated and quantized symbols must be recombined into a single bit quantized digit stream using the same commutation rule employed at the transmitter.

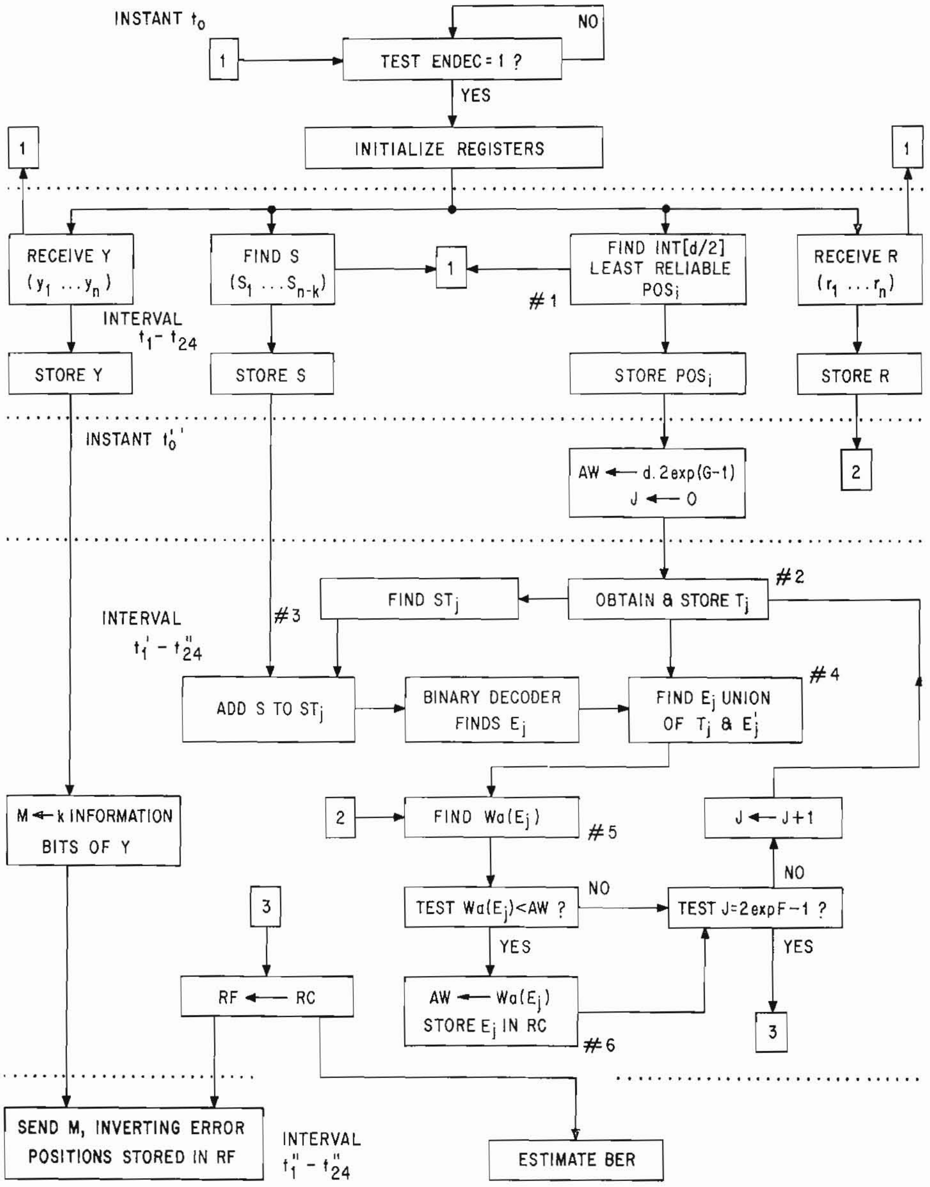

Figure 1. Decoding macro-process flowgraph.

From **Fig. 1**, it is evident that the stationary decoding delay is one block, and then we do not need duplication of decoders to handle any message time distribution. Also, a very handy function to system evaluation, namely the BER estimation, may be provided by the decoder.

At this point, instead of going further, detailing the generalized circuits to carry out each process, it will be easier to understand the principles of the architecture by showing the general block diagram of the hardware. We then analyze the block implementation for the specific application and finally generalize the requirements.

The general decoder block diagram is shown in **Fig. 2**. The input signals are power (VCC and GND), decoder enabling (ENDEC), clocks (which will be detailed in the next section), and input quantized (data and reliability) signals. Its output signals are the decoded data and BER alarm.

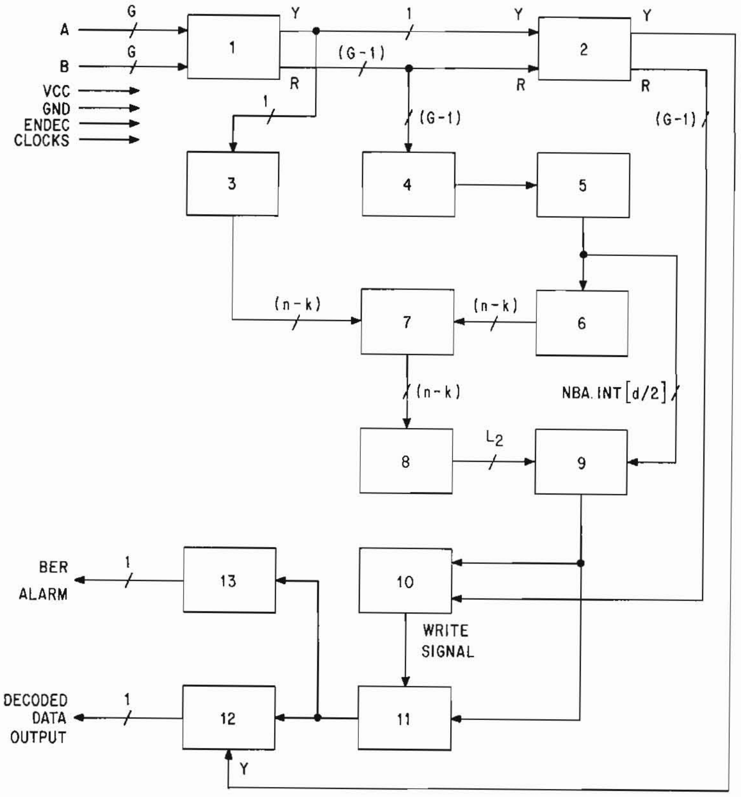

Figure 2. Decoder block diagram.

All presented findings allow us to draw a conclusion: combined pipeline and parallel processing, as used in the division of the macro-process shown in **Fig. 1**, can effectively provide an optimum architecture to perform Chase's algorithm II decoding.

3. Application to Digital Satellite Transmission Systems

3.1 Use of EGC (24, 12, 8) with Chase's Algorithm II in the SAMSAT System

As a concrete application example, we shall analyze the Chase decoder architecture composition in the SAMSAT environment. SAMSAT is the Brazilian narrow-band multi-frequency TDMA data transmission system by satellite for medium capacity domestic applications. Each SAMSAT system operating at its full capacity occupies one third of a 36 MHz transponder of the Brazilian domestic telecommunications satellite (BRASILSAT), and then there may be up to three SAMSATs per transponder. Each SAMSAT can handle up to 126 terminal stations (ETS) and has 2 reference stations: a master for primary reference (ER0) and a secondary reference (ER1). Both reference stations do not carry traffic. Total system capacity is 8.192 Mbit/s of information, allocated in 60 ms frames working at 512 kbit/s net information rate along 16 distinct frequencies.

Each ETS may support up to 16 ports (terrestrial interface modules) working at speeds ranging from 1.2 kbit/s up to 384 kbit/s in a variety of different interfaces such as CCITT V.24/V.28, V.36/V.11, G. 703 (both synchronous and plesiochronous). Non-switched voice service at 32 and 64 kbit/s are also supported. Burst assignment is pre-allocated (on a reservation basis) that may change once every 5 minutes, the minimum database change interval. The ETS multi-frequency TDMA/QPSK modem operates at a digit rate DR = 1068 kbit/s. The gross information rate (information plus overhead such as unique words, carrier recovery sequence, etc) is IR = 534 kbit/s, and an extended Golay code EGC (24, 12, 8) is applied to protect the already scrambled net information bits.

To convert the coded digit stream into two in-phase and quadrature streams to feed the modulator, decommutation is performed such that the odd positions of an encoded block are sent to the in-phase channel and the even ones to the quadrature channel. At the reception, coherent demodulation is used, with phase ambiguity removal and block synchronization achieved via the unique word detection, that also provides frame alignment and master station reference clock "windowing" for ETS clock recovery. The reference clock of any ETS is 16 times the digit rate, i.e., it is 17.088 MHz and then it is possible to generate any clock which is a rational multiple or submultiple of the digit rate (DR) up to 17.088 MHz.

SAMSAT system specifications stand for a BER $<10^{-7}$ for an operating E_b/N_o = 9.5 dB, including all allowable degradations, which led to an E_b/N_o = 8.5 dB at an IF loop of the modem with multi-frequency and burst operation. This requirement demands the use of FEC with soft-decision decoding. Adopted FEC scheme is the EGC(24, 12, 8) with soft-decision 3-bit quantized decoding using Chase's algorithm II mainly because block coding at rate 1/2 rate is easy to be done at the transmitter side, does not require any burst extension period as convolutional codes do, has simple clock generation and because it achieves the required performance, being as good as short-length rate 1/2 convolutional codes with Viterbi decoders.

We shall now detail how to design the blocks of the decoding architecture, showing their interrelationship. We begin by remembering that for 3-bit quantized EGC (24, 12, 8) sequences, using algorithm II, we shall have 4 least reliable positions per block, 16 possible test patterns, a maximum value of 3 for eack r_i. Therefore, an upper bound value for the analog weight is 3.8 = 24. Furthermore we assume a binary decoder that corrects all the error pattens up to 3 errors and some (1/6) of the 4-error patterns (for the EGC is a quasi-perfect code), with $2^{(n-k)}$=4096 correctable error patterns. Also, EGC may be characterized by a polynomial generator since it is derived from a cyclic code and we will assume that it is in its systematic format [3] and [4]. Also, block alignment and any rational multiple or submultiple of the basic clock rate are available. Then, if we state that any position within any block will be addressed by a number in the range 1 to 24, we may declare the value 0 as a dummy address, which will be very handy for hardware processing as we shall see soon. Finally, we assume the 3-bit quantizer output encoded as given in **Table 1**. With these input constraints in mind, we may now start the block implementation description. The block numbers referred to are always those shown in **Fig. 2**.

The input interface – block 1 – is simply composed of line receivers at all input signals and a multiplexer of the in-phase and quadrature demodulated signals, controlled by the information clock IR=534 kHz, to reconstruct the block sequence.

The primary syndrome generator – block 3 – may be implemented as a conventional syndrome register with (n-k)=12 flip-flops, as described in [3] and [7], and a (n-k) store register.

The least reliable digits selector – block 4 – performs process denoted as # 1 in **Fig. 1.** It must search for the addresses of the 4 least reliable positions and may be implemented as sketched in **Fig. 3.** The master counter plays a major role not only in this block but also in the whole architecture since it is responsible for all the address generation.

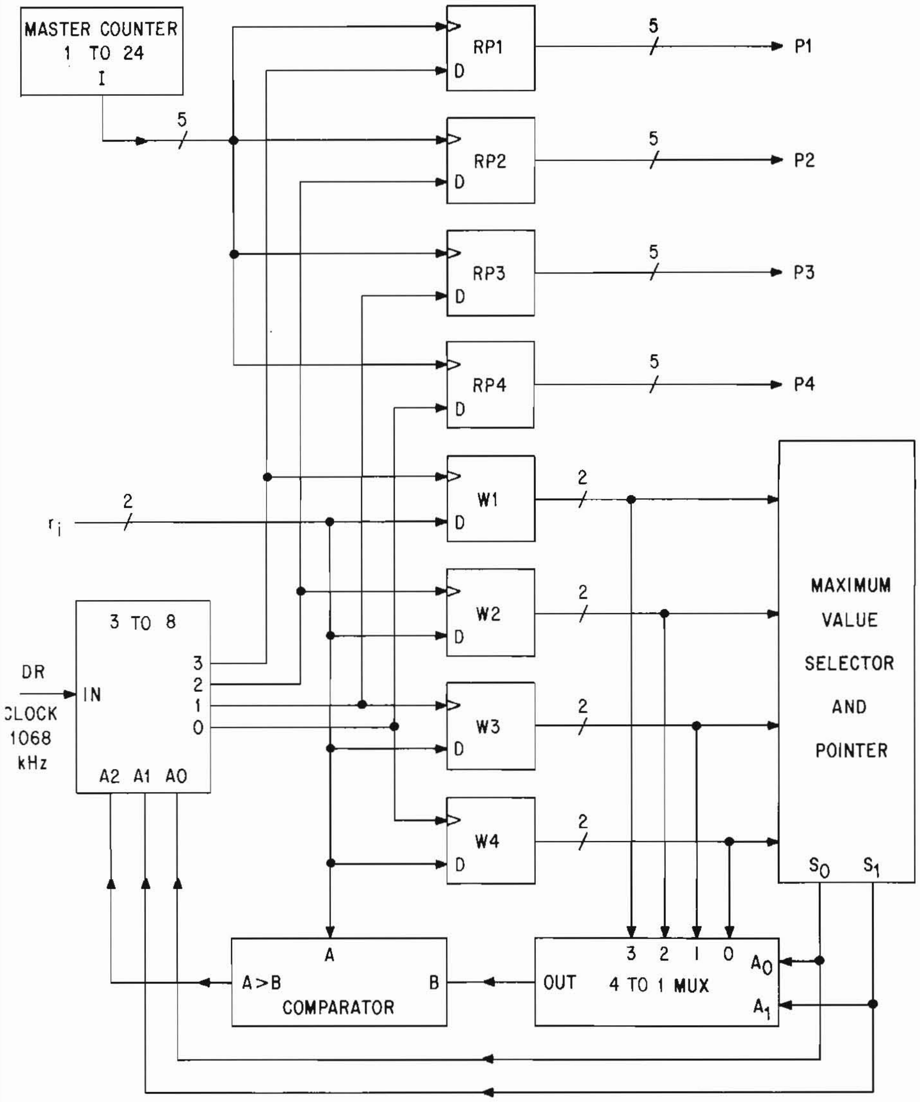

Figure 3. Least reliable digits selector.

Quantizer Outputs	03	02	01	Hypothetical Limiter Output	Confidence Levels	Waveform Status at the Sampling Instants
Significance	MSB		LSB			
Correspondence	y_i	r_{i2}	r_{i1}			
Threshold	1	1	1	+1	3	Strongest Positive
	1	1	0	+1	2	
	1	0	1	+1	1	
	1	0	0	+1	0	Weakest Positive
	0	0	0	−1	0	Weakest Negative
	0	0	1	−1	1	
	0	1	0	−1	2	
	0	1	1	−1	3	Strongest Negative

Table 1. Three bit quantization correspondence.

The test pattern generator – block 5 – responsible for process # 2 shown in **Fig. 1** is extremely simplified by using position addresses and the dummy address 0. **Fig. 4** shows the necessary hardware to generate the 16 vectors T_j represented by addresses DT_1 to DT_4. As only one $\mathbf{T_j}$ has 4 non-zero DT_j's, all the other will have at least one DT that is null, which means that this DT points to "nowhere" in the block. The process in block 5 can be seen as a position address masking function. The masking clock must guarantee the test pattern generation of all test patterns in one block period in order not to increase the decoding delay, nor to duplicate functions. Then it must have a frequency f_m which is equal to (16/24). DR, that is, 712 kHz. In general, we have

$$f_m = 2^{\text{INT}\,[d/2]} \cdot \frac{DR}{n} \qquad (16)$$

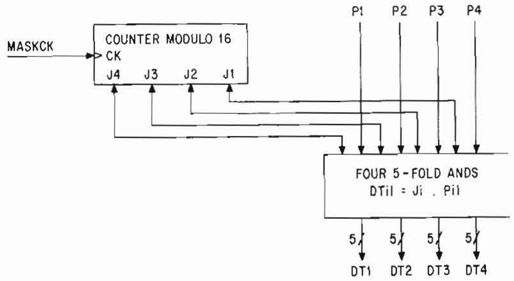

Figure 4. Test pattern generator.

The test syndrome generator – block 6 – responsible for process # 3 in Fig. 1 and the syndrome adder – block 7 – may be implemented together as shown in Fig. 5, because both blocks use the same number of adders, and we may eliminate a lot of circuitry by selecting three-state devices for hardware implementation, which give us a high-impedance state. The test pattern is multiplied by H' via 4 additions of lines stored in PROMs, addressed by the multiplexed DT_j's. If an extra addition with the primary syndrome is done, the modified syndrome $\mathbf{SM_j}$ that will be fed into the binary decoder results. The modulo-8 counter must have a clock frequency $f_{sum} = 2{\cdot}4{\cdot}f_m = (16/3).DR=5696$ kHz. In general, we have

$$f_{sum} = 2^{INT[d/2]} \quad f_m \tag{17}$$

It should be noticed that PROMs contents are not only the lines of the H matrix but also an all-zero line at address 0, in order not to change the modified-syndrome when any DT_j is 0. Also it is important to point out that output $\mathbf{SM_j}$ is ready only when VALID signal goes to 1. Finally, note that f_{sum} is the maximum necessary clock frequency to be supplied to the decoder's hardware.

The binary decoder – block 8 - is simply an EPROM array that performs a table look-up operation. Given an address which is the 12-bit $\mathbf{SM_j}$, it gives as output

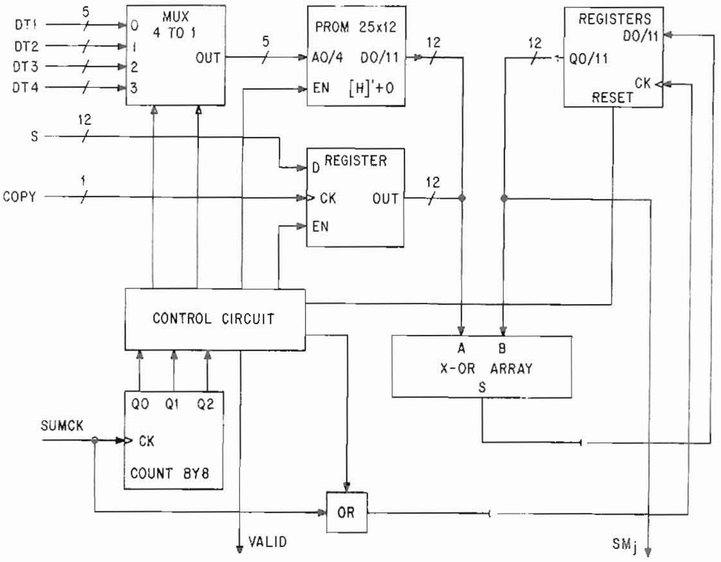

Figure 5. Test syndrome generation and syndrome adder.

the addresses of digits that are 1 in the decoded error-pattern. This is a set of four 5-bit addresses, namely PE_1, PE_2, PE_3 and PE_4. Generally, look-up table decoding needs an (n-k)-bit syndrome input to respond with a set of t addresses of valued 1 digit positions in the error-pattern. There shall be an enabling signal, such as VALID is, to indicate a valid syndrome. Other hard-decision decoding might be used, but this may not be practical, for today's EPROMs are quite cheap.

The coincident address analyzer – block 9 – carries out process # 4 in **Fig. 1** and its structure is depicted in **Fig. 6.** Again a bus strucuture is used to multiplex (do the union of) the signals PE_i and DT_i by enabling 3-state devices at appropriate instants. However, to get the proper valid error pattern E_Δ, we need to erase the coincident positions between sets $\{PE_i\}$ and $\{DT_i\}$. The multiplexed positions of valued 1 digits of $\mathbf{E_i}$ will be used in the next blocks, the metric analyzer and the error-position memory. Since there are 4 PE_i's and 4 DT_i's, the frequency f_{aw} of the AWCK ("Analog Weight Clock") must be $8.f_m$, that is, 5696 kHz. In general, we must have

$$f_{aw} = \{ \text{INT}[\, d/2 \,] + t" \} \,.\, f_m \tag{18}$$

where

$$t" = \begin{cases} \text{INT}[\,(d-1)/2\,] & \text{for perfect codes} \\ 1 + \text{INT}[\,(d-1)/2\,] & \text{for quasi-perfect codes} \end{cases} \tag{19}$$

thus $f_{aw} = f_{sum} = d.f_m$ whenever the code being used is quasi-perfect. Also, although Chase decoders correct up to (d-1) errors, the hardware will always need to work with d addresses due to the random nature of the ordering of error positions in the sets $\{PE_i\}$ and $\{DT_i\}$.

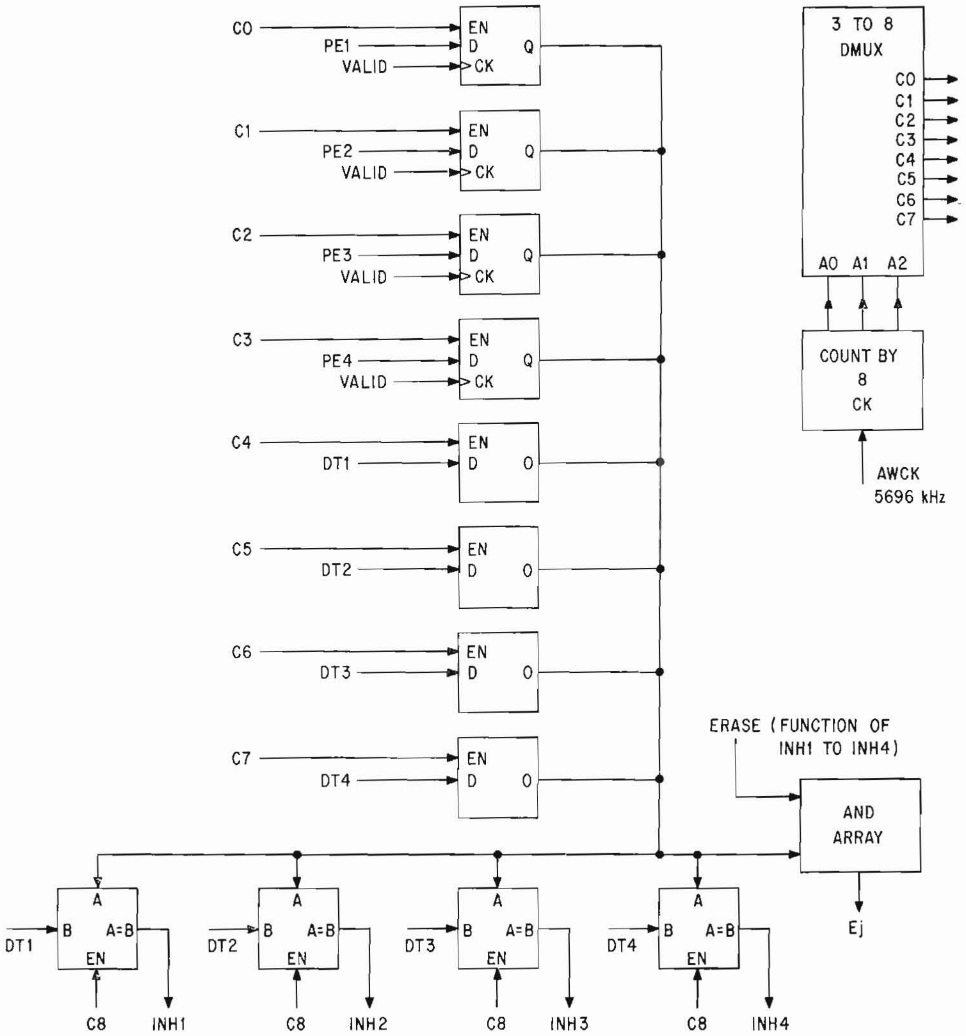

Figure 6. Coincident address analyzer.

The metric analyzer shown in **Fig.** 7 – block 10 – performs process # 5 and part of process # 6 shown in **Fig.** 1 and always works together with the reliability and data memory. Its output carries a WRITE information to the error-position memory. The accumulator clock must be the same AWCK used by block 9. The metric register initial value IAW must always satisfy

$$IAW > (d-1)(2^{G-1} - 1) \qquad (20)$$

In our case, we need IAW > 7·3 and we selected IAW = 8·3 = 24.

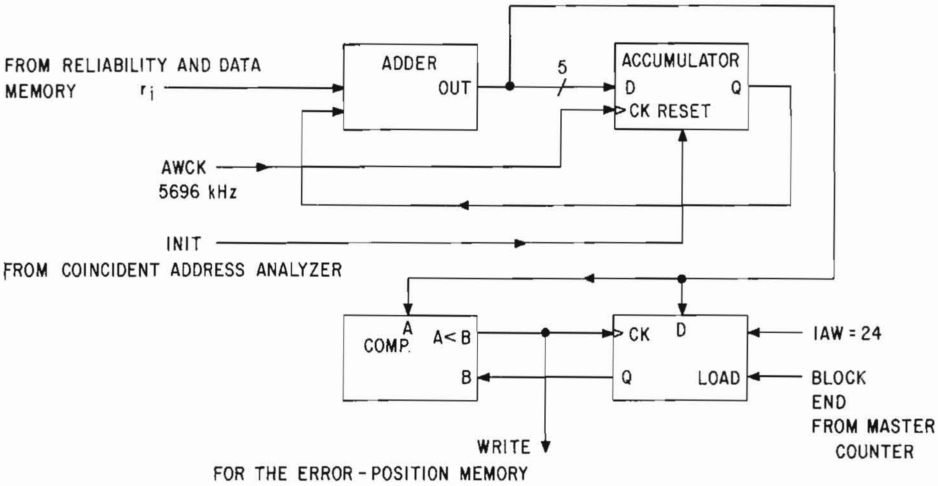

Figure 7. Metric analyzer.

The error correction and output interface unit – block 12 – receives the error positions, stored in the error-position memory, via a read sweeping operation over this memory during an information bit period. If the information bit address agrees with any stored error position, an address comparator pulses and this pulse is fed into an exclusive-or gate to invert the current information bit, providing the correction. This bit stream is regenerated and fed into a line driver which is the output source. The information bits come from a rate converter (1068 kbit/s to 534 kbit/s) that reads them from the reliability and data memory.

BER meter – block 13 – just counts the number of correction pulses during a certain number of bits. If the counter of correction pulses overflows, an alarm of excessive input (prior to correction) BER is activated. BER must be measured within a given tolerance and with a high confidence level [**8**].

The last two blocks – the memories – should be carefully analyzed due to its pagination schemes. The reliability and data memory – block 2 – will be always a 2.(n+1) long and G-bit wide RAM, that is a 50x3 RAM in our example. This memory performs the following functions:

(i) receive and store the reliability and data sequences, with 24 write cycles per codeword, at 1068 kbit/s;

(ii) store 0 at relative address 0 of current selected memory page, with 24 write cycles per codeword, at 1068 kbit/s;

(iii) read the information bits to send to output, with 12 read cycles at 1068 kbit/s per codeword;

(iv) read the reliability values to feed the metric analyzer, with 16·8=128 read cycles per codeword at 5696 kHz.

This memory has two pages that avoid duplication of other decoding functions because while a block read from one page is being decoded, the subsequent block may be being stored at the other page.

The error-position memory – block 11 – also has two pages, each one divided in two segments. In general, it will be always a 4.{INT[d/2]+t"} long and N_a-bit wide RAM, with N_a as given by (**14**). In our case, it is a 32x5 RAM. The two pages are provided for mutiplexed write cycles for the current processing codeword and for read cycles for the current output codeword (that is being corrected). Segmentation is needed due to the fact that a decision on any $Wa(E_j)$ is possible only at the end of the current E_j time interval. Therefore, we first need to write E_j in a segment of the current page and then, based on the metric comparator output, we can either declare this segment good and change of segment, or declare this segment as scratch memory and continue to use it. At the end of decoding, the minimum analog weight error pattern will be stored at the segment which is not being pointed as "able to write" by the write address buffer.

At last, we shall derive the required RAM access times. Since the maximum write/read frequency is 5696 kHz which is (16/3).DR and since a divide-by-3 counter has an output duty cycle of 1/3, it follows that we need an access time T_a such that

$$T_a < 58.5 \text{ ns} \qquad \textbf{(21)}$$

Due to propagation delays, parasitic capacitances, jitter and skew, we have chosen $T_a = 45$ ns.

To evaluate the maximum operating speed of this architecture, we note that the fastest NMOS RAM available today has $T_a = 25$ ns. Then the maximum theoretical speed would be

$$DR_{max} = 2.5 \text{ Mbit/s} \qquad (22)$$

or

$$IR_{max} = 1.25 \text{ Mbit/s} \qquad (23)$$

3.2 Other Application Examples

There are several possible applications of this architecture in digital satellite communications systems. Since all blocks are completely defined by the code parameters (n, k, d), the digit rate DR and the number G of quantization bits, it is very easy to make an assessment of viability or fitness for any Chase algorithm II decoder using this architecture. TDMA systems or any "RF framed" system may be directly implemented. Application to continuous "unframed" transmission systems would need some extensions, in order to design the codeword alignment auxiliary circuits.

This work shows clearly that this architecture is very suitable to implement a standard PCB – using standard ICs – or even an ASIC to decode binary linear and preferably systematic and cyclic block codes, that have moderately large minimum distance (d<12), operating at medium data rates.

An extension of the specific EGC(24, 12, 8) decoder considered here to continuous "unframed" transmission applications is not complicated since EGC(24, 12, 8) is transparent to phase ambiguities that lead to word inversion. This is so because the all-ones word is a codeword, i.e., it has symmetrical weight distribution. DBPSK (Differential BPSK) applications do not need any additional circuitry. DQPSK applications must have a block alignment circuit that may use the BER alarms as thresholds for changing a channel commutator and inverter. A recursive polynomial division of Y by the code generator polynomial might be done by additional hardware, in order to ensure a small alignment search time. Coherent demodulation for "unframed" transmission is impossible in this case due to the specific code transparency to phase rotation.

4. Measurement Results

The hardware presented for the EGC (24, 12, 8) used in SAMSAT has been designed using 124 standard ICs on a 13.05 x 8.87 square inch double-sided PCB whose density is 0.63 square inches/equivalent 14-pin IC. A single +5V power supply is required and in the worst case the measured operating current was 1300 mA at the nominal digit rate of 1068 kbit/s. Ten PCBs have been built to check repeatability and to provide enough hardware for SAMSAT field trial.

Baseband continuous non-scrambled transmission tests in an AWGN (Additive White Gaussian Noise) channel have given the typical results listed in **Table 2** (by-pass correction and hard/soft-decision selection jumpers are available at PCB). Numbers in **Table 2** are ensemble average figures, where a minimum of 5 consecutive measurements were made with 100 accumulated bit errors per measurement. Log-linear regression was made to find a function relating the output BER (BER_{out}) and the input BER (BER_{in}). We have obtained for soft and hard-decision, respectively

$$\log BER_{out} = 5.05 \log BER_{in} + 3.19 \qquad \text{(soft-decision)} \tag{24}$$

$$\log BER_{out} = 4.03 \log BER_{in} + 3.13 \qquad \text{(hard-decision)} \tag{25}$$

BER_{in}	BER_{out} (hard)	BER_{out} (soft)
$5.90.10^{-2}$	$1.18.10^{-2}$	$7.50.10^{-4}$
$5.02.10^{-2}$	$6.70.10^{-3}$	$3.16.10^{-4}$
$3.93.10^{-2}$	$3.15.10^{-3}$	$9.48.10^{-5}$
$2.87.10^{-2}$	$9.46.10^{-4}$	$2.53.10^{-5}$
$1.99.10^{-2}$	$1.84.10^{-4}$	$3.63.10^{-6}$
$1.64.10^{-2}$	$1.12.10^{-4}$	$2.27.10^{-6}$
$1.53.10^{-2}$	$7.00.10^{-5}$	$1.32.10^{-6}$
$1.18.10^{-2}$	$2.52.10^{-5}$	$3.65.10^{-7}$
$1.02.10^{-2}$	$1.48.10^{-5}$	$1.73.10^{-7}$
$7.61.10^{-3}$	$3.66.10^{-6}$	$3.78.10^{-8}$
$6.40.10^{-3}$	$1.97.10^{-6}$	$1.83.10^{-8}$
$4.01.10^{-3}$	$2.53.10^{-7}$	$1.39.10^{-9}$ (1)

Table 2. Measured average performance of EGC (24, 12, 8) Chase decoder,

1) less than 100 accumulated bit errors per measured point, 12 for this specific point.

Finally, measured results for IF loop with TDMA multi-frequency operation of the modem have shown that at $E_b/N_0 = 8.5$ dB the BER is better than 10^{-7}, i.e., complies with the specification. SAMSAT system tests with 3 ETSs and the reference stations have demonstrated that the whole system works very well through the BRASILSAT and no BER associated problems have been detected so far at the user's interfaces.

5. Conclusions

A novel approach to Chase's Algorithm II Decoding Method for binary linear block codes was derived. This approach was based on the division of the several steps of the algorithm into pipelined parallel processes suitable for hardware implementation. It was shown that this hardware is optimum for logical circuitry minimization. Then the implementation for the specific case of an Extended Golay (24, 12, 8) code applied to a narrow-band multi-frequency TDMA data transmisson system by satellite developed at CPqD – the SAMSAT – was presented and discussed. Possible extensions of the derived circuits for the general case of a binary linear block code (n, k, d) have also been considered. It was also shown that the resulting hardware is very suitable to decode binary linear and preferably systematic and cyclic block codes with moderately large minimum distance ($d<12$) operating at medium data rates.

Finally, practical measured results for the above mentioned case were presented, showing the BER performance at several system operating points. From these results, the improvement of the coding system on the overall system performance can be obtained.

The presented architecture is being submitted for patent registration.

Acknowledgements

The authors wish to thank D. S. Arantes and A. C. F. Pessoa for their early work on Chase's algorithm II simulation, A. C. Marchiori and R. Cardoso for device progamming and debug efforts, F. B. Brandão for his patient laboratory help, J. M. Pitsch for the fruitful discussions and guidance and CPqD-Telebrás for the sl ipoort and funding of this research.

References

[1] D. Chase, "A Class of Algorithms for Decoding Block Codes with Channel Measurement Information", IEEE Transactions on Information Theory, vol. IT-18, no. 1, January 1972, pp. 170-182.

[2] A. Shenoy and P. Johnson, "Serial Implementation of Viterbi Decoders", COMSAT Technical Review, vol. 13, no. 2, Fall 1983, pp. 315-329.

[3] G.C. Clark and J. B. Cain, "Error-Correction Coding for Digital Communications", Plenum Press, 1982.

[4] A.J. Viterbi and J. K. Omura, "Principles of Digital Communication and Coding", McGraw Hill Kogakusha, Ltd., 1979.

[5] A.C.F. Pessoa, "On the Sensitivity of Chase Decoders to Variations of AGC Output in the Reception", Internal Report, TELEBRÁS-UNICAMP, August 1986 (in portuguese).

[6] C. M. Hackett, "An Efficient Algorithm for Soft-Decision Decoding of the (24, 12) Extended Golay Code", IEEE Transactions on Communications, vol. COM-29, no. 6, June 1981, pp.909-911.

[7] W.W. Peterson and E.J. Weldon, Jr., "Error-Correcting Codes", The MIT Press, 2nd edition, 1972.

[8] R.J. Kerczewski, E.S. Daugherty and I. Kramarchuk, "Gauge Bit-Error Rate as a Function of Signal-to-Noise Ratio", Microwaves & RF, vol. 27, no. 2, February 1988, pp. 107-113.

HÉLIO CÉSAR ALVES SEABRA SALLES naceu em 20 de março de 1961 no Rio de Janeiro. Formou-se em Engenharia Elétrica – Automação e Controle – na UNICAMP, em 1982, e obteve o grau de Mestre em Ciências em Engenharia Elétrica – Eletrônica e Comunicações – na UNICAMP, em 1990. Durante a graduação foi bolsista de iniciação científica da FAPESP, tendo atuado em controle de máquinas c.c. por modulação em largura de pulso. Ingressou como engenheiro no CPqD da TELEBRÁS em 1983 onde atuou no desenvolvimento de uma Unidade de Canal de Dados para Comunicações via Satélite. Em 1985 ingressou na IBM atuando na área de desenvolvimento de periféricos de comunicações para "mainframes". Retornou ao CPqD-TELEBRÁS em 1986 onde atuou no projeto de um sistema AMDT para Comunicações via Satélite (SAMSAT), tendo projetado a unidade decodificadora de correção de erros e exercido a função de responsável pelas unidades de banda básica do modem do sistema até 1989. Atualmente é responsável pelo grupo de engenharia de sistemas rádio da Coordenação de Áreas de Sistemas Rádio do Departamento de Transmissão e Acesso. Suas principais áreas de interesse são códigos corretores de erro, modelamento de canais de comunicação via rádio/satélite e aplicações de rádio/satélite em redes de telecomunicações.

WALTER DA CUNHA BORELLI graduou-se no ano de 1972 em Engenharia Elétrica pela Escola de Engenharia Elétrica de São Carlos da Universidade de São Paulo. Obteve o grau de Mestre no ano de 1975 em Engenharia Elétrica, na área de Telecomunicações, junto ao então Departamento de Engenharia Elétrica da Faculdade de Engenharia de Campinas da UNICAMP. Defendeu seu doutorado no ano de 1983 na área de Teoria de Informação e Codificação junto aos "Electronic Labs." da Universidade de Kent em Canterbury na Inglaterra. Trabalha na UNICAMP como professor desde 1973 sendo atualmente Professor em nível MS-4 do Departamento de Telemática da Faculdade de Engenharia Elétrica. Atualmente desenvolve pesquisa na área de Códigos Corretores de Erros, Modulação Codificada,Comunicação Multi-Usuários e na área de Procolos de Comunicação para Redes Digitais de Serviços Integrados e Redes de Computadores.