rise to an m fold ambiguity in the phase estimates. A practical m-PSK modulation system adjusts for this by using differential data encoding, and performing the function of differential decoding at the receiver.

Let θ_n denote the estimator for the n-th symbol phase of a symmetric estimation interval ($-N \leq n \leq \leq N$). The variance of the estimator, var[θ_n], is derived in closed form in [10], where it is shown that it results independent of n. The analysis reported [10] has been carried out under the assumption of a time uncertainty a negligibly small fraction of the symbol time T. Thus a highly accurate and controlled timing must be envisaged.

Interesting feature of this carrier phase estimator are that preambles can be usually avoided and a short and defined acquisition time, equals N_E symbols, is required. Further this estimation method performs well with only two samples per symbol. Therefore a suitable sampling frequency rate variation from three samples per symbol to two samples per symbol must be implemented. The influence of a finite arithmetic implementation on the carrier estimated value can be derived only by simulations.

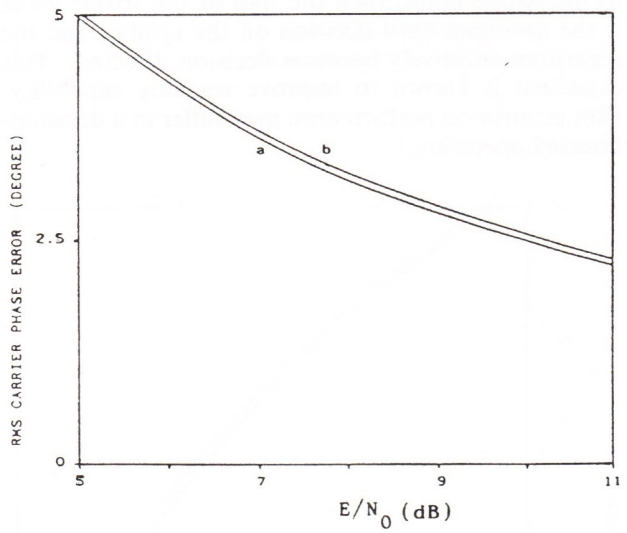

Fig. 8 - Carrier phase error as function of E/N_0.

a) floating point implementation;

b) finite precision implementation at 6 bits.

In Fig. 8 the mean square error on the carrier phase is shown as function of E/N_0, with E the energy per bit and N_0 the one-sided noise spectral density, by considering a floating point implementation (curve a) and a finite arithmetic implementation with $b_f = = 6$ bits (curve b). It can be noted from this figure that a carrier phase mean square error less than 5 degrees can be achieved for E/N_0 greater than 5 dB with $b_f = 6$ bits. This quantization loss can be significantly reduced by a more accurate quantization process. Naturally this has a direct impact on the size of the ROM. The implementation complexity of the proposed carrier phase estimator method can be derived as:

$$\begin{aligned} M_v &= 3R && \text{mults/s} \\ A_v &= 2(N+1)R && \text{adds/s} \end{aligned} \tag{17}$$

by considering two samples per symbol. The presence of a frequency uncertainty ϵ_f influences very much the performance of the considered carrier phase estimator. The variance of the estimate error can be explicitated as function of ϵ_f and N as:

$$\text{var}[\theta_n] = \frac{\Gamma(N, \epsilon_f)}{4(2N+1)\beta} \tag{18}$$

where $\beta = E/N_0$. The function $\Gamma(N, \epsilon_f)$ has been derived for a large class of nonlinearity $F(p_n)$ in reference [10]. Degradations introduced on the symbol error probability, naturally depends upon var[θ_n]. Hence for specified values of ϵ_f and β the degradations are influenced by the choice of N. In particular, with $\Gamma(N, \epsilon_f)$ as an explicit function of N, it can be shown that an optimum values of N exists [10]. This optimum value permits to achieve better carrier phase estimate accuracy. Fig. 9 shows the degradations introduced by a frequency uncertainty (normalized with respect to T) equal to 0.005 for $\beta = 10$ dB as function of N. In this case the best choice of N results 11 symbols. Therefore, to achieve better performance the optimum value of N must be used. In the rest of this paper we mainly consider the case $\epsilon_f = 0$. This condition can be guaranteed by providing good stable frequency source and by a reasonable accurate frequency estimation. In this case N must be chosen as a good tradeoff between short acquisition time and carrier phase estimate accuracy.

As a conclusion, it should be noted that the Nonlinear Estimation method of a QPSK-modulated carrier phase achieves a good estimation accuracy, is less sensitive to a finite arithmetic implementation and requires a definite (and short) acquisition time equal to $N_E = (2N + 1)$ symbols.

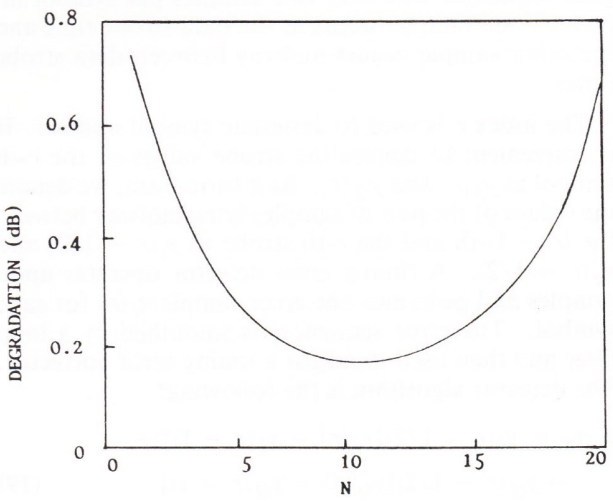

Fig. 9 - Degradation due to $\epsilon_f T$ equal to 0.005 as function of N for $\beta = 10$ dB.

b) *QPSK timing-error detector*

The timing error detector considered in this section can be directly related to that proposed in reference [11]. Its principle of operation will now be briefly described. For the considered approach only two samples per symbol are employed to achieve timing error detection with one sample coincident with the decision instant. These samples are derived by a suitable interpolation/decimation operation.

With reference to Fig. 10 for a block diagram of a typical I-Q receiving modem, the signal from DEMUX, already down converted to a low frequency range, is demodulated by a pair of quadrature-driven mixers. The phase of the local carrier must be adjusted to agree with that of the signal. The necessary carrier-recovery branch which is omitted from the diagram is irrelevant to the clock algorithm and discussion. Data filters are in general required; they perform receiver filtering to shape signal pulses, minimize noise, and suppress unwanted mixer products. It should be noted that in our case they are already implemented in the DEMUX.

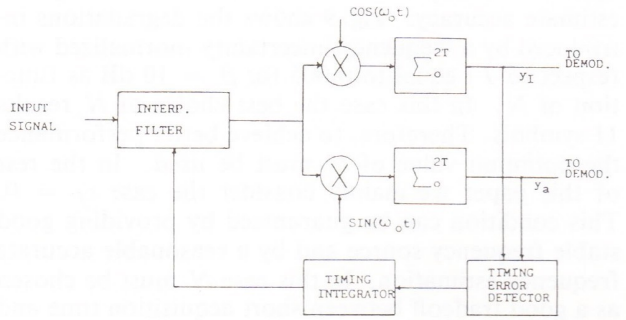

Fig. 10 - QPSK timing-error detector circuit.

Our interest is in sampled receivers. We do not specify the sampling point other that the filter outputs are available only in sampled form as the pair of real sequences $\{y_I(.)\}$ and $\{y_Q(.)\}$. Timing information must be retrieved from these sequences. As symbols are transmitted spaced by the time interval T each sequences will have two samples per symbol interval. One sample occurs at the data strobe time and the other sample occurs midway between data strobe times.

The index r is used to designate symbol number. It is convenient to denote the strobe values of the r-th symbol as $y_I(r)$ and $y_Q(r)$. As a formalism, we denote the values of the pair of samples lying midway between the $(r-1)$-th and the r-th strobe as $y_i(r-1/2)$ and $y_q(r-1/2)$. A timing error detector operates upon samples and generates one error sample $u_t(r)$ for each symbol. The error sequences is smoothed by a loop filter and then used to adjust a timing error corrector. The detector algorithm is the following:

$$u_t(r) = y_I(r-1/2)[y_I(r) - y_I(r-1)] + \\ + y_Q(r-1/2)[y_Q(r) - y_Q(r-1)] \qquad (19)$$

This algorithm is suitable for both tracking and acquisition modes of operation. It is proven in [11] that $u_t(r)$ is independent of carrier phase, so that timing lock can be achieved without depending upon prior phase acquisition.

A phisical explanation can be found in (19). The detector samples the data stream midway between strobe locations in both the I and Q channels. If there is a transition between symbols, the average midway value should be zero, in the absence of timing error. A timing error gives a nonzero sample whose magnitude depends upon the amount of error, but the slope information necessary for the correction (lead or lag) is missing. To sort out the different possibilities, the algorithm examines the two strobe values on either side of the midway sample. If there is no transition, the strobe values are the same, their difference is zero, and so the midway sample is rejected. (No timing information is available in the absence of a transition.) If a transition is present, the strobe values will be different; the difference between them will provide slope information. The product of the slope information and the midway sample provides timing-error information. It may be worthwhile to use the signs of the strobe values instead of the actual values. This eliminates the effects of noise. If all data filtering has been performed prior to the strobe point, then the sign of the strobe value is the optimum hard decision on the symbol and the algorithm effectively becomes decision directed. This expedient is known to improve tracking capability. (But acquisition performance may suffer in a decision-directed operation.)

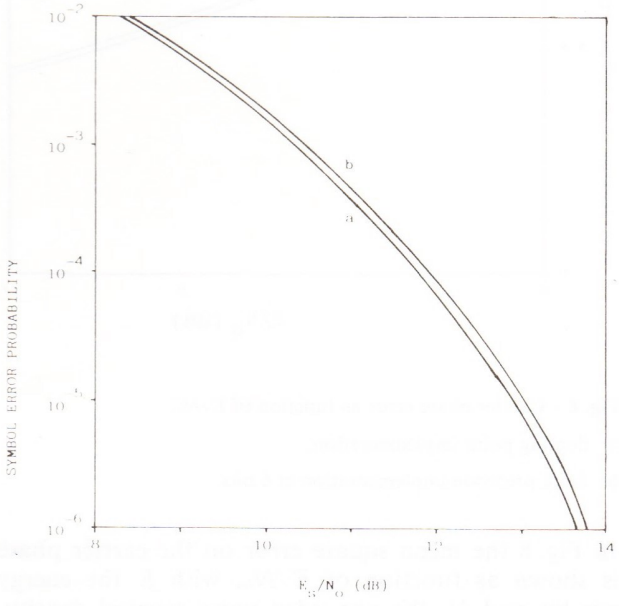

Fig. 11 - Symbol error probability versus E_s/N_0 achieved when the QPSK timing-error detector is used.

a) floating point implementation;

b) finite precision implementation at 8 bits.

Note that use of the strobe signs, instead of actual values, eliminates the need for actual multiplications in the algorithm: an attractive feature for digital processors. The decision-directed version of the algorithm will be recognized as very similar to the digital transition tracking loop of Lindsey and Simon [16]. The

implementation complexity for the timing error detector, including the interpolator filter used in [14], can be derived as:

$$M_c = 16R \quad \text{mults/s}$$
$$A_c = 16R \quad \text{adds/s} \tag{20}$$

where it has been assumed two samples per symbols.

The performance of the described clock recovery method, including the effects of a finite arithmetic implementation, has been derived through computer simulation.

In Fig. 11 the symbol error probability obtained by considering the timing offset error due to the clock recovery circuit and the channel noise are reported as function of E_s/N_0 with E_s the energy-per symbol. A finite arithmetic implementation of the clock recovery circuit at 8 bits has been considered. It can be noted from this figure that a degradation less than 0.1 dB is introduced in this case.

4. CLOCK SYNCHRONOUS SYSTEM

This section considers the benefits of the use of a network clock synchronization on the DEMUX and DEMOD design. Indeed, when all the carriers are clock synchronized at the satellite receiver, only one sample per symbol at the optimum decision time instant can be used at the DEMOD input.

For the analytic signal approach the use of a network clock synchronization results in a reduction in the implementation complexity. Indeed in this case only one sample per symbol is required at the DEMUX output. Therefore, by maintain unchanged the signal bandwidth and the channel spacing, a frequency sampling reduction by a factor of three can be included in the low-rate stage of the DEMUX. The number of multiplications required per channel and per second is now given by:

$$M_{AS} = (L_G/3 + L_H/2)2W \tag{21}$$

From the previous equation it can be noted that a significant reduction for M_{AS} is achieved making use of the network clock synchronization. The saving in terms of multiplications required per channel and per second is enhanced by making use of the polyphase approach to implement the DEMUX. Indeed in this case a reduction by a factor of three of the output sampling frequency means that two out of every three samples of the FFT output are discarded. Hence, the FFT processor as well as the digital filters of the polyphase network can operate at the low sampling rate $2W/3$. Therefore, the total computation rate in the DEMUX system can be reduced by a factor three. Hence the number of multiplications required per channel and per second can be esplicitated as:

$$M_{\text{FFT}} = [L_f + 4\log_2 N_c]2W/3 \tag{22}$$

For the multistage approach a lower reduction is achieved. This is so because in this DEMUX approach a sampling frequency reduction by a factor three can be only performed in the last stage of the tree structure. Therefore the number of multiplications required per channel and per second is now defined as:

$$M_{MS} = \{[(N_F + 1)/2 + 1][\log_2 N_c - 1/2) + N_G/3\}2W \tag{23}$$

Finally, it can be pointed out that also the carrier recovery circuit can be simplified making use of a network clock synchronization. In fact, taking into account that the carrier phase estimation method presented in sect. 3.a) can operate with only one sample per symbol. Therefore, its implementation complexity results to be:

$$M_v = 2R$$
$$A_v = (2N + 1)R \tag{24}$$

For the clock recovery only one clock error estimator is required for all the N_c channels. It can estimate the clock error operating in time sharing on all the channels. Hence the contribution on the overall MCD implementation complexity due to the clock recovery circuit is greatly reduced.

5. SYSTEM DESIGN AND PERFORMANCE

The MCD design is discussed in this section. Different DEMUX and DEMOD implementation techniques are considered. To this regard an important consideration to be made is that the number of channels processed by the DEMUX (N_c) influences the input sampling frequency and consequently the processing rate and the complexity of the first stage of the MCD. In particular, a feasible constraint is to require that the input A/D converter sampling frequency (clock) should be close to its maximum possible value. Starting from these considerations, as the design goal, we have selected $N_c = 8$ at $R = 2048$ Kbit/s. The design of the digital multiplexer is first presented.

Table 1 - *Implementation complexity for the Analytic Signal method.*

N_c	R (Kbit/s)	L_H	L_G	$M_{AS} \times R$ (mults/ch/s)
8	2048	35	27	66.75
10	2048	44	27	73.5

R = Transmission rate;
L_H = High-rate low-pass filter coefficients number;
L_G = Low-rate low-pass filter coefficients number;
M_{AS} = Overall number of multiplications required per channel and per second.

For the Analytic Signal approach, the high-rate and low-rate lowpass prototypes have been designed as a

Table 2 - *Finite arithmetic DEMUX design* (Analytic Signal method).

N_c	R	Input Signal Quantization	Filters $H(fT_u)$			Filters $G(fT_d)$		
	(Kbit/s)	b_q	b_c	b_m	b_a	b_c	b_m	b_a
8	2048	8	12	11	8	11	11	8
10	2048	8	12	11	8	11	11	8

b_q = input signal word length;
b_c = filter coefficient word length;
b_m = filter arithmetic word length;
b_a = filter output word length;

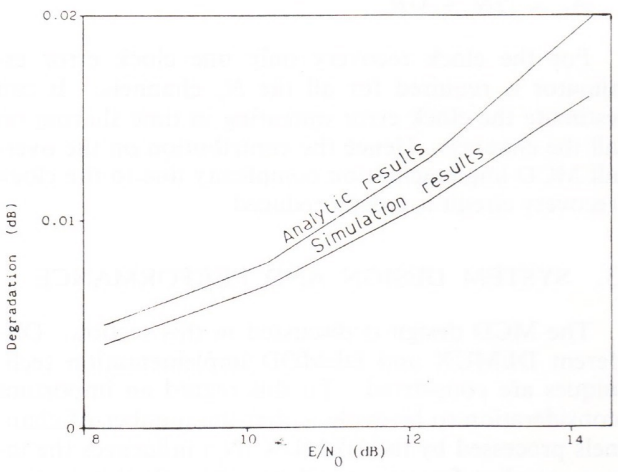

Fig. 12 - Performance degradation due to a finite precision implementation of the DEMUX ($N_c = 8$, $R = 2048$ Kbit/s).

Table 3 - *Implementation complexity for the Fast Fourier Transform with polyphase network approach.*

N_c	R (Kbit/s)	L_{FT}	M_{FFT} XR (mults/ch/s)
8	2048	224	60

R = transmission rate;
L_{FT} = low/pass prototype coefficients including pulse shaping;
M_{FFT} = overall number of multiplications required per channel and per second.

FIR linear phase filter by using the equiripple method [19]. It can be noted that the low-rate lowpass prototype has been designed to include the required pulse-shaping function with a 40% roll-off factor equally shared between the transmitter and the receiver. The implementation complexity in terms of multiplications per second and per channel is reported in Table 1 for the different considered values of N_c.

Table 4 - *Finite arithmetic demultiplexer design. (FFT with polyphase network, $N_c = 8$, $R = 2048$ Kbit/s.)*

Input Signal Quantization	Polyphase Filters			FFT Processor		
b_q	b_c	b_m	b_a	b_{Fc}	b_{Fm}	b_{Fa}
8	12	13	10	11	13	8

b_q = input signal word length;
b_c = filter coefficient word length;
b_m = filter arithmetic word length;
b_a = filter output word length;
b_{Fc} = FFT coefficients word length;
b_{Fm} = FFT arithmetic word length;
b_{Fa} = FFT output word length.

Table 5 - *Implementation complexity for the Multistage approach.*

N_c	R (Kbit/s)	M_{MS} XR (mults/ch/s)
8	2048	74.25

A finite arithmetic implementation is necessarily required to implement any digital processing system. Al though it is possible to conceive and actually implement floating-point arithmetic for digital signal processing systems, it is deemed that the fixed-point arithmetic implementation will still represent the more convenient solution for the near term. To this end, the filtering specifications and the DEMUX finite precision design has been derived to introduce at each demultiplexer output a suitable degradation, with respect to the input signal-to-noise ratio SNR_i. The finite arithmetic wordlengths are reported in Table 2.

Table 6 - *Finite arithmetic demultiplexer design. (Multistage approach)*

N_c	R	Input Signal Quantization	Half-band Filters			Pulse Shaping Filters		
	(Kbit/s)	b_q	b_c	b_m	b_a	b_c	b_m	b_a
8	2048	8	12	14	8	12	14	8

b_q = input signal word length;
b_c = filter coefficient word length;
b_m = filter arithmetic word length;
b_a = filter output word length;

Table 7 - *DEMUX loss for $N_c = 8$ and $R = 2048$ Kbit/s. (Multistage approach)*

E/N_0 (dB)	DEMUX loss (dB)
8	0.05
10	0.09
12	0.16
14	0.22

In Fig. 12 the degradations in dB for the output signal-to-noise ratio introduced by the digital implementation of the DEMUX are reported as function of E/N_0. It can be noted that there is a good agreement between the results derived through the theoretical analysis and those obtained through computer simulations. It should be said that the theoretical analysis considers the worst case degradation.

For the polyphase approach as for the multistage method only N_c values that are an integer power of 2 must be taken into account. Therefore only the case $N_c = 8$ is considered. The low-pass digital filter prototype has been designed as a FIR linear phase filter. The required pulse shaping function has been included in this filter in order to reduce the overall implementation complexity. In Table 3 the overall DEMUX complexity in terms of multiplication per second and per channel (M_{FFT}) is reported for $N_c =$ $= 8$ and $R = 2048$ Kbit/s. The finite precision design has been carrier out following a suitable procedure described in details in [13]. It can be pointed out that in the complexity evaluation of this DEMUX approach the control circuitry for the FFT processor must also be taken into account. Based on in-house FFT implementations we can estimate the amount of FFT processor control circuitry of the order of 1.5 to 2 time the overall number of multiplications per second. This added control complexity has not been included in Table 3. In Table 4 the finite arithmetic wordlengths derived to introduce a degradation for the output signal-to-noise ratio less than 0.05 dB at $E/N_0 = 8$ dB are reported.

In Table 5 the overall implementation complexity achieved when the Multistages Demultiplexer approach is used, is reported for $N_c = 8$ and $R = 2048$ Kbit/s. It can be noted that also in this case the required pulse shaping function is performed in the DEMUX. The finite precision design has been carried out in order to introduce acceptable loss. The obtained finite precision wordlengths are reported in Table 6. In Table 7 the corresponding actual degradation are reported for different values of E/N_0.

The overall DEMOD implementation complexity results equal to $19R$ (mults/ch/s) with $N_E = 33$. The degradation of the demodulator (demod loss) are due to the phase jitter introduced by the carrier phase estimate and to a symbol timing offset introduced by the symbol timing estimate.

In order to evaluate the loss due to the phase jitter and symbol timing offset, we have assumed them as two independent noise contributions. The loss due to a phase jitter can be derived through the following equation [20]:

$$\text{loss(dB)} = 4.34\,(1 + 2\Omega)[1 + (1 + 2\Omega)/2\alpha]/\alpha \qquad (25)$$

where Ω is equal to E/N_0 and α for moderate to high signal-to-noise ratios, can be assumed equal to $1/\sigma_\theta^2$ with σ_θ the root-mean-square (r.m.s) value of the phase error. In particular, for the considered non-linear estimate method, in the steady state condition, through simulations it results equal to 5 degrees. The degradation is reported in Fig. 13 as function of E/N_0 with a finite precision implementation at 6 bits.

It can be noted that an acceptable loss (less than 0.25 dB) can be achieved. The degradation due to the symbol timing offset has been derived by computer simulations. The overall DEMOD loss, also including the effects of a finite precision implementation for the two different approaches results equal to 0.33 dB at $E/N_0 = 8$ dB.

The benefits of the use of a network clock synchronization on the MCD design are now illustrated. The resulting overall implementation complexity (not including the common clock recovery circuit) in terms of number of multiplications per channel and per second is reported in Table 8a) for $N_c = 8$ and $R = 2048$ Kbit/s. In the same Table the achieved DEMOD

complexity and the parameters M_{AS}, M_{FFT} and M_{MS} are also reported. In Table 8b) the same parameters are reported for comparison when a clock network synchronization is not used.

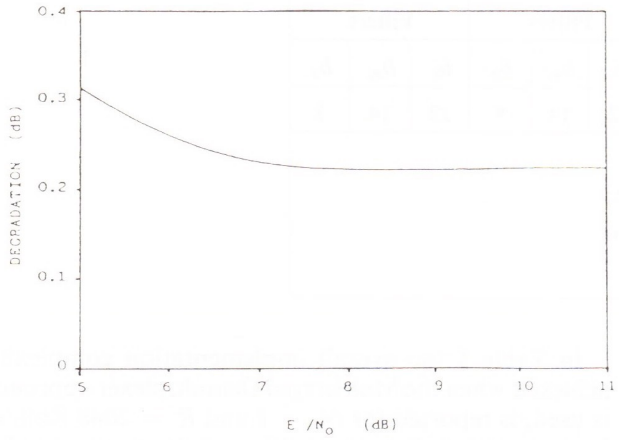

Fig. 13 - Degradation due to the carrier recovery circuit (finite precision implementation at 6 bits).

From the previous results it can be pointed out that, when a network clock synchronization is used, the Polyphase approach represents a suitable solution for a digital DEMUX implementation. However, if a network clock synchronization is not possible solution, the Analytic Signal approach seems to be a convenient choice to implement a digital DEMUX on the basis of the following main evaluation criteria:

a) implementation complexity;
b) possible integration of same demultiplexer and demodulator functions;
c) low sensitivity to a finite precision implementation;
d) low implementation degradations.

In addition the Analytic Signal method has the further following advantage with respect to the other approaches:

e) suitability for flexible modes of operation, i.e. variable bit rate or number of channels processed by a single MCD.

Table 8.a) - *Overall MCD implementation complexity for N_c = 8 and R = 2048 Kbit/s when a network clock synchronization is used*

	Analytic Signal Approach	Polyphase Approach	Multistage Approach
DEMUX (mults/ch/s)	39.75	20	47.25
DEMOD (mults/ch/s)	2	2	2
Overall MCD (mults/ch/s)	41.75	22	49.75

Table 8.b) - *Overall MCD implementation complexity for N_c = 8 and R = 2048 Kbit/s when a network clock synchronization is not used*

	Analytic Signal Approach	Polyphase Approach	Multistage Approach
DEMUX (mults/ch/s)	66.75	60	74.25
DEMOD (mults/ch/s)	19	19	19
Overall MCD (mults/ch/s)	85.75	79	93.25

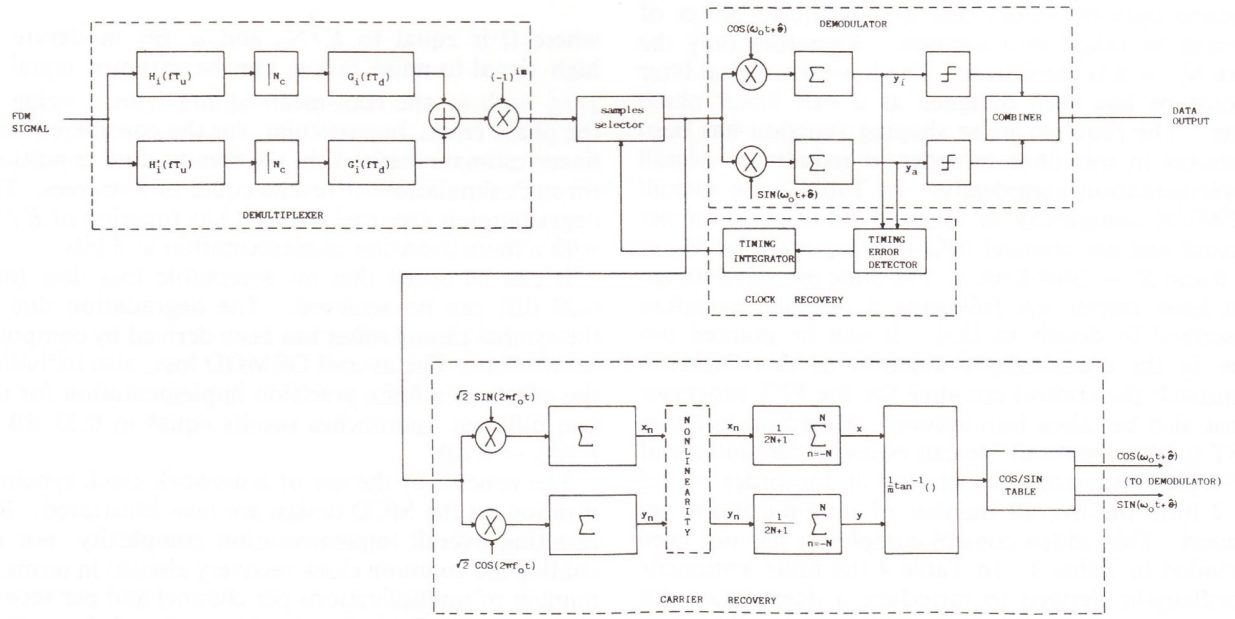

Fig. 14 - Implementation structure of the considered MCD.

The block diagram of a possible implementation of the resulting MCD system is shown in Fig. 14. The achieved overall MCD loss are reported in Table 9 for different values of E/N_0 by considering the case $N_c = 8$ and $R = 2048$ Kbit/s.

Table 9 - *Overall MCD loss for $N_c = 8$ and $R = 2048$ Kbit/s when a network clock synchronization is not used.*

E/N_0 (dB)	DEMUX loss (dB)	DEMOD loss (dB)	Overall loss (dB)
8	0.004	0.330	0.334
10	0.007	0.305	0.312
12	0.010	0.280	0.290
14	0.016	0.230	0.246

In conclusion the alternatives for a digital MCD system described herein represent an appropriate solution for digital communication systems directly interfacing FDMA and TDM links on-board the satellite. In particular, it is suitable for implementation by custom or semicustom VLSI digital circuits.

Acknowledgment

The topics of this work have been thoroughly discussed with many people from Telespazio, Roma (Italy). In particular, the authors wish to thank E. Saggese, G. Chiassarini and G. Gallinaro for their helpful cooperation.

Manuscript received on July 29, 1988.

REFERENCES

[1] G. Benelli, E. Del Re, R. Fantacci, F. Mandelli: *Performance of uplink random-access and downlink TDMA techniques for packet satellite networks.* "IEEE Proceedings", vol. 72, n. 11, Nov. 1984, p. 1583-1593.

[2] R.E. Crochiere, L.R. Rabiner: *Multirate digital signal Processing.* Prentice Hall, Englewood Cliffs, New Jersey, 1983.

[3] M. Bellanger: *Digital processing of signals: theory and practice.* John Wiley, London, 1984.

[4] E. Del Re, F. Ronconi, P. Salvi, P. Semenzato: *Comparison of Non-FFT methods of TDM-FDM transmultiplexing.* "Alta Frequenza", vol. LI, n. 1, Jan-Feb. 1982, p. 9-21.

[5] E. Del Re, P.L. Emiliani: *An analytic signal approach for transmultiplexers: theory and design.* "IEEE Trans. on Comm.", vol. COM-30, n. 7, July 1982, p. 1623-1628.

[6] M.G. Bellanger, J.L. Dauguet: *TDM-FDM transmultiplexer: digital polyphase and FFT.* "IEEE Trans. on Comm.", vol. COM-22, n. 9, Sept. 1974, p. 1199-1205.

[7] E.S. Yam, M.D. Redman: *Development of a 60 channel FDM-TDM Transmultiplexer.* "Comsat Technical Review", vol. 13, n. 1, 1983, p. 1-55.

[8] H. Soheruermann, H. Gockler: *A comprehensive survey of digital transmultiplexeing methods.* "IEEE Proc.", vol. 69, n. 11, Nov. 1981.

[9] E. Del Re: *A property of digital quadrature filters.* "IEEE Proc.", vol. 69, Dec. 1981, p. 1577-1578.

[10] A.J. Viterbi, A.M. Viterbi: *Non linear estimator of PSK modulated carrier phase with application to burst digital transmission.* "IEEE Trans. on Inf. Theory", vol. II-29, n. 4, July 1983, p. 543-551.

[11] F.M. Gardner: *A BPSK/QPSK timing-error detector for sampled receivers.* "IEEE Trans. on Comm.", vol. COM-34, n. 5, May 1986, p. 423-429.

[12] E. Del Re, R. Fantacci: *A digital communication receiver with integrated MAP synchronization and ML demodulation.* "Electronics Letters", vol. 21, Sept. 1985, p. 862-864.

[13] E. Del Re, R. Fantacci, et al.: *Multicarrier demodulator design.* Estec Contract 6096/84/NL/GM(SC), Final Report, 1986.

[14] F.M. Gardner: *On board processing for mobile-satellite communications.* Estec Contract 5889/84/NL/GM, Final Report, 1985.

[15] O. Agazzi, C.P.J. Tzeng, D.G. Messerschmitt, D.A. Hodges: *Timing recovery in digital subscriber loops.* "IEEE Trans. on Comm.", vol. COM-33, June 1985, p. 558-569.

[16] W.C. Lindsey, M.K. Simon: *Telecommunication systems engineering.* Englewood Cliffs, Prentice-Hall, New York 1973.

[17] H.L. Van Trees: *Detection estimation and modulation theory.* New York J. Willey, 1968.

[18] R.W. Booth: *An illustration of the MAP estimation method for deriving closed loop phase tracking topologies: the MSK signal structure.* "IEEE Trans. on Comm.", vol. COM-28, Aug. 1980, p. 1137-1142.

[19] *Programms for Digital Signal Processing.* ASSP DIGITAL Signal Processing Committee (Ed.). IEEE Press, N.Y., 1979.

[20] R. Matyas: *Effect of noisy phase references on coherent detection of FFSK signals.* "IEEE Trans. on Comm.", vol. COM-26, June 1978, p. 807-815.

[21] V.K. Bhargava, D. Haccaun, R. Matjas, P.P. Nuspl: *Digital communication by satellite.* John Wiley, New York, 1984.

[22] E. Del Re, R. Fantacci: *Alternatives for on-board digital multicarrier demodulator.* "Int. Journal of satellite communications", vol. 6, July-Sept. 1988, p. 267-281.

E. Del Re, R. Fantacci: **Digital Multicarrier Demodulator for Regenerative Communication Satellites**

ALTA FREQUENZA, VOL. LVII - N. 10 DECEMBER 1988, p. 545-559