Programmable Single Board Spread Spectrum Digital Modem

E. Rossini [1], C. Campa [1], G. Chiassarini [1], G. D'orazio [1], D. Gianfelici [1], M. Marsella [1]

[1] Space Engineering SpA, via dei Berio 91 – I–00155 Roma – Italy – e-mail: rossini@space.it

ABSTRACT

Space Engineering SpA has recently developed a multi-processor, multi-FPGA (Field Programmable Gate Array), compact platform, including a multiple analog interface based on high-speed and high-accuracy ADC (Analog to Digital Converter) and DAC (Digital to Analog Converter).

The platform with related software is intended to be a potentially powerful development system for the implementation of spread spectrum digital modems for satellite and terrestrial communications. The platform however is compact and flexible in order to be adapted the real user needs in terms of performance and cost.

The platform is in fact a self-standing single board, housed in a small 19' drawer with 220 V power supply, local display and keyboard. It can be also remotely controlled by a personal computer via high-speed serial interfaces.

The platform has been used to implement modulators, demodulators and interference generators, for S-UMTS and T-UMTS (Satellite and Terrestrial Universal Mobile Telecommunication System) working in DS-CDMA (Direct Sequence Code Division Multiple Access) in the frame of several ESA (European Space Agency) and EC (European Community) programs.

Moreover complete terminals for voice/data transmission and wide-band interference generators have been implemented for wide-band FH-FDMA (Frequency Hopping Frequency Division Multiple Access).

The versatility of the board has allowed addressing the effort on the algorithm's design while maintaining the same hardware platform. A large variety of algorithms have been developed in the frame of digital modems mainly using the VHDL (Very high speed Hardware Description Language) language to configure the FPGA devices and the C language to program the DSP (Digital Signal Processor). This paper focuses the attention on the complexity analysis of the implemented algorithms and on their partitioning to achieve an optimum mapping on the available programmable devices.

HARDWARE DESCRIPTION

A block diagram of the designed hardware platform is given in Fig. 1. We can distinguish two SHARC microprocessors from Analog Devices working up to 40 MHz on the external busses and having an instruction cycle at frequency up to 100 MHz obtained exploiting the internal PLL. Also the peripherals section is visible. It consists of 6 FPGA (Field Programmable Gate Array) by Altera for an overall capacity of up to 6 Mgates. The FPGA are widely interconnected in order to implement all the foreseen algorithms. In particular some FPGA directly control some specific peripherals needed for particular functions. These peripherals include an external large fast memory bank, two Viterbi decoders, and bi-directional TTL (Transistor Transistor Logic) and LVDS (Low Voltage Differential Signalling) interface buffers. The FPGA control the board analog interface formed by three 12 bits ADC working at frequencies up to 105 MHz with 500 MHz of input bandwidth and a double 14 bits DAC converter working up to 135 MHz.

The DSP section includes 3 UART (Universal Asynchronous Receiver Transmitter) able to manage 3 independent serial links up to 921 Kbps and a bi-directional audio analog interface. The board is fully re-configurable in all its parts (DSP software and FPGA firmware) thanks to a large Flash memory bank that can be updated by an external computer via serial link.

The board has been designed in such a way only the required parts can be mounted to fulfil a specific function, in order to keep the overall board cost at the minimum. It is possible select three sizes of FPGA to be mounted: 400, 600 and 1000 Kgates. The Table 1 summarises the corresponding area characteristics of each of them.

Almost the totality of the algorithms has been implemented into the FPGA whilst the DSP has been used to interface the serial links with the FPGA and for controlling and monitoring purposes.

Table 1: Types of FPGA that can be mounted on the Hardware Platform

FPGA size (Kgates)	Logic Elements (LE)	RAM (Kbits)	RAM (Kbytes)
400	16 640	208	26
600	24 320	304	38
1000	38 400	320	40

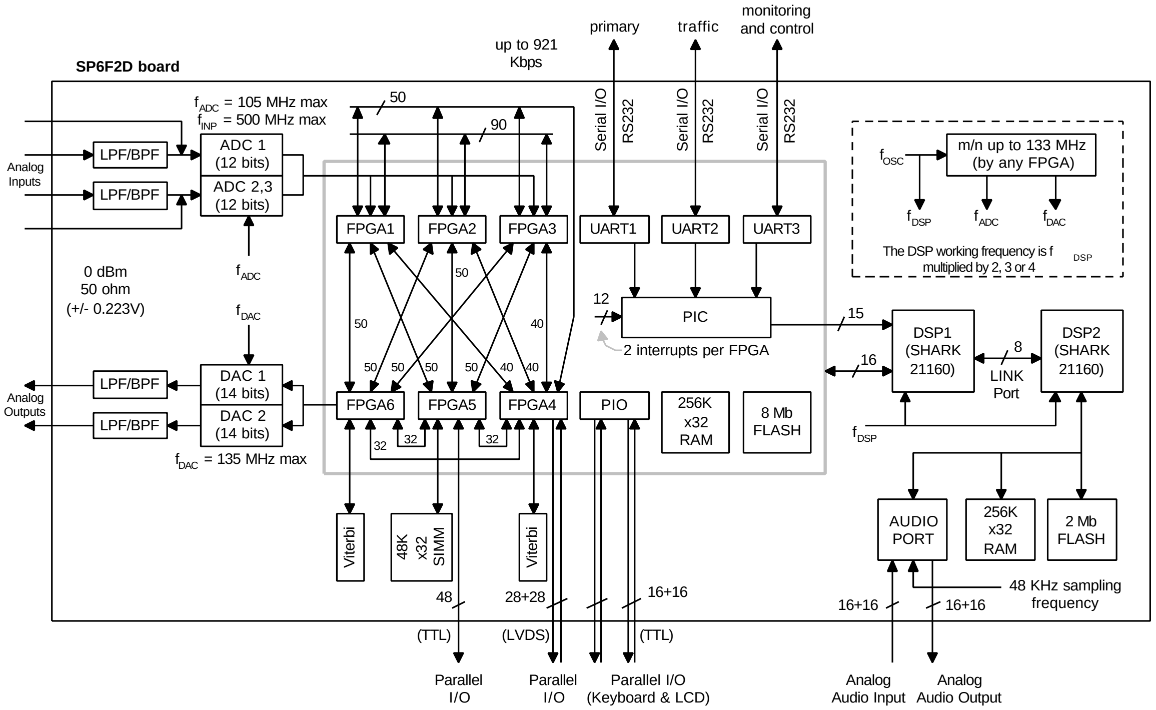

Fig. 1: Hardware Platform Block Diagram

The design flow used to translate the algorithms into electrical circuits to be hosted by the FPGA can be summarised in the following steps:

- VHDL (Very high speed Hardware Description Language) - RTL (Register Transfer Logic) DESIGN: VHDL language oriented entry [**1**] [**2**];
- FUNCTIONALVERIFICATION: simulation of the algorithm via MODELSIM VHDL simulator and analysis of its correctness through comparing with the results of a C++ bit true system simulation;
- CIRCUIT SYNTHESIS: translation of the algorithm in electronic circuit using LEONARDO SPECTRUM synthesiser;
- DEVICE FITTING: circuit fitting on the target FPGA device using the ALTERA QUARTUS II tool [**3**];
- VHDL-STRUCTURAL VERIFICATION: simulation through MODELSIM VHDL simulator and ALTERA libraries of the fitted circuit and test of its correctness through comparing with the results of the RTL simulation;
- ON BOARD TESTING: FPGA testing on the target hardware measuring signals through a Logic State Analyser;

The MODEM equipment based on this hardware platform appears as in the Fig. 2. The LCD (Liquid Crystal Display) display is visible in the centre of the frontal panel and the keyboard on its right. The system keyboard-display allowed setting and monitoring of the main modem parameters like the power of the transmitted signals, the status of the demodulation process and the performances in terms of BER (Bit Error Rate) and FER (Frame Error Rate). These features have made the use of the modem as stand alone equipment possible. A more sophisticated monitoring is also available using one of the serial links connected with an external PC. Data like the evolution of the loops and the scattering diagrams can be displayed using a GUI (Graphical User Interface) to have an immediate verification of the modem functionality.

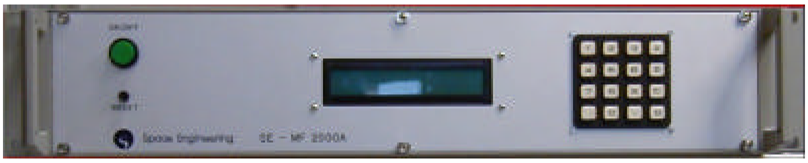

Fig. 2: MODEM Front Panel

DS-CDMA ALGORITHMS IN UMTS ENVIRONMENT

Since a MODEM includes both the modulator and the demodulator sections, many algorithms can be considered constituted by a transmitting part and by its dual part at the receiver. In particular the algorithms implementing the specifications of the standard UMTS (Universal Mobile Telecommunication System) physical layer [**4**] can be grouped into sets called 'processors' as indicated in the Table 2.

Table 2: Modulator Algorithms and Demodulator Dual Algorithms

Set Name	Modulator Algorithm		Demodulator Dual Algorithm	
Symbol Processor	1	CRC (Cyclic Redundancy Code) encoder	10	CRC checker
	2	FEC (Forward Error Correction) encoder	11	Viterbi Decoder
	3	Puncturing	12	De-Puncturing
	4	Interleaver	13	De-Interleaver
Chip Processor	5	Walsh-Hadamard Spreader	14	Walsh-Hadamard De-Spreader
	6	Gold Scrambler	15	Gold De-Scrambler
Sample Processor	7	SRRC (Square Root Rise Cos) TX Filter	16	SRRC RX Filter
Conversion Processor	8	Digital Up Converter	17	Digital Down Converter
	9	Interpolator Filter	18	Decimator Filter

A further class of algorithms is peculiar to the demodulator and can be classified according to the Table 3.

Table 3: Demodulator Peculiar Algorithms

Set Name	Demodulator Algorithm	
Demodulator Symbol Processor	19	AGC (Automatic Gain Control) Loop
	20	DLL (Delay Locked Loop)
	21	AFC (Automatic Frequency Control) Loop
	22	Channel Estimator and Phase Recovery
	23	Frame Sync Word Detector and Fine Frequency Acquisition
	24	Rake Receiver
Acquisition Processor	25	Chip and Coarse Frequency Acquisition

A brief description of each algorithm is given below:

CRC Encoded and CRC Checker: An 8 bits wide CRC field is added to each transmitted information frame. This field is filled with a value obtained from the frame itself after that it has been passed through a shifter based state machine (CRC Encoder). The received frame is then controlled carrying out the same computation and comparing the so obtained CRC value with the received one (CRC Checker).

FEC Encoded and Viterbi Decoder: A FEC Encoder with rate 1/3 is used to improve the BER. The decoding process is based on a soft-decision Viterbi algorithm [**5**] [**6**].

Puncturing and De-Puncturing: After the FEC redundancy, some bit is discarded following a given rule, with minimum losses in term of BER, in order to match the available channel bit rate with the information one. The received punctured frame is de-punctured inserting zeros in the same position where the bits have been discarded during the puncturing process.

Interleaver and De-Interleaver: This algorithm is based on exchanging the positions of the bits in order to avoid bursts of consecutive errors that are difficult to be recovered using the FEC/Viterbi approach.

Walsh-Hadamard Spreader and De-Spreader: This algorithm is responsible in transforming bits in chips. It characterises the CDMA (Code Division Multiple Access) modulation being each bit associated with N chips, where N is the so-called spreading factor. The N chips are chosen according to the Walsh-Hadamard matrix in order to achieve only orthogonal codes which don't cause interference in a synchronous code division access technique.

Gold Scrambler and De-Scrambler: The Gold codes are used to distinguish channels using the same WH (Walsh Hadamard) code. This is necessary when more satellites transmit the same channel with the same WH code.

SRRC Filters: These FIR filters are used to shape the transmitted chips and as front end for the received base-band signal. This front end is also used as actuator of the timing recovering process being able to change its impulse response according to the DLL indication.

Digital Up/Down Converters and Interpolator/Decimator Filters: Exploiting the features of the hardware platform, the ADC and DAC conversions can be made directly at IF. This means that the signals are up and down converted (in transmission and reception respectively) to obtain the base band signals managed by the modulation and demodulation processes. The conversion processing includes also Interpolator and Decimator Filters (in transmission and reception respectively) to achieve the required sampling rate [**7**] [**8**].

AGC Loop: It is a digital gain control loop necessary to keep constant the signal power. The processing following this gain control requires that its input remain almost constant to assure good performances.

DLL: The DLL is the loop used to recover the timing of the incoming chips. It is based on an Early-Late error detector followed by a second order loop filter. The state variable of this filter is used to indicate the impulse response to be used to the SRRC receiving filter [**9**].

AFC Loop: It is the loop used to follow the carrier frequency during the tracking phase. Coarse and Fine frequency acquisitions are carried out by different algorithms included respectively in the Acquisition Processor and in the Frame Synch Word Detector [**9**].

Channel Estimator and Phase Recovery: Based on the Pilot Symbol of the P-CCPCH (Primary-Common Control Physical) whose scattering position is known a priori, this algorithm estimates both its amplitude and phase using these values to correct the amplitude and phase of all received symbols.

Frame Sync Word Detector and Fine Frequency Acquisition: It is the algorithm used to synchronise the frame. It is based on the cross correlation between the incoming FSW and the expected one and it is able also to extract from this correlation the phase error between two consecutive symbols in order to obtain a Fine estimation of the carrier frequency error.

Rake Receiver: This is the algorithm used to improve the performances of the satellite link exploiting the contemporary reception of three channels carrying the same information. After a realigning procedure of the three channels, which arrive in general with different delays, a suitable weighed sum is performed to collapse the three received frames into only one.

Chip and Coarse Frequency Acquisition: These are the algorithms performed by the Acquisition Processor to synchronise the demodulator to the incoming Gold and Walsh-Hadamard codes and to obtain a Coarse estimation of the carrier frequency error.

Each algorithm can be characterised in terms of ‘gates’ (area parameter) and ‘speed’ (time parameter) corresponding to the algorithm’s working frequency. Often, due to the internal architecture of the FPGA, the area parameter needs to be specified using different units like number of ‘logic elements’ or ‘memory locations’. We analysed the area complexity of each algorithm mainly implementing it in VHDL language and verifying the result of the synthesis tool. Specific design techniques have been used to minimise the area occupied by the gate consuming algorithms like filters. The working frequency of each algorithm depends on the activation rate of the corresponding processor which, in the UMTS framework, is summarised by the Table 4.

Table 4: Working Frequency of each Processor

Set Name	Working Frequency	Notes
Symbol Processors	Symbol Rate	Chip Rate / Spreading Factor
Chip Processors	Chip Rate = 3.84 MHz	
Sample Processors	Sample Rate = 15.36 MHz	Works at 4 samples per chip
Acquisition Processor	Chip Rate /2 = 1.92 MHz	
Conversion Processor	ADC/DAC frequency > 100 MHz	

DS-CDMA ALGORITHMS COMPLEXITY

In order to obtain an estimation of the algorithms complexity, we synthesised the algorithms under the following hypothesis:

- The modulation and demodulation include two UMTS-like channels: a Primary Control Channel and a Traffic Data Channel with bit rate of 4.6 Kbps and 192 Kbps respectively. Different bit rates don’t change the complexity of the algorithms but the used RAM. However the RAM necessary is always included in the RAM available on chip. The pair Primary plus Traffic is often referred as a Finger.
- The rake receiver algorithm is performed on three Traffic Data Channels and it doesn’t include the input FIFO, necessary to realign the three incoming frames, implemented as final stage of the algorithm 22 (Channel Estimator and Phase Recovery).
- The sample processors actually also include an up converter in the modulator section and a down converter in the demodulator section working at 3.84 MHz used to manage the carrier frequency error (e.g. through the AFC loop).

First of all we decided to carry out a rough partitioning aimed to minimise the number of signal exchanged between FPGA and grouping algorithms working at similar frequencies. Furthermore, we decided to implement the CRC

Encoded/Checker with a C program running on the DSP in order to provide/receive to/from the FPGA a complete information frame. Finally we synthesised the so obtained set of algorithms achieving the results reported in the Table 5 ((*) Implemented on the DSP).

Table 5: Algorithms Complexity

Algorithm Number	LE Used	RAM Used (Kbytes)	MOD/ DEM	Rate (MHz)	Algorithm
2 … 7	10 810	4.7	MOD	Sample (= 15.36)	MOD Symbol, Chip, Sample Proc.
8, 9	5 325	0	MOD	Conversion (> 100)	Digital Up Converter
1, 10	(*)	(*)	Both	Info Bit Rate	CRC Encoder/Checker
11	9 216	20	DEM	Symbol (= 3.84/SF)	Viterbi Decoder
12, 13	3 187	5.5	DEM	Symbol (= 3.84/SF)	DEM DeInterlever, DePuncturing
14 … 16 19 … 23	14 477	24.5	DEM	Chip (= 3.84) Symbol (= 3.84/SF)	DEM Symbol, Chip, Sample Proc. But DeInterleaver, DePunctuting
17, 18	6 656	0	DEM	Conversion (> 100)	Digital Down Converter
24	7 878	0	DEM	Symbol (= 3.84/SF)	Rake Receiver
25	14 592	1.6	DEM	Chip (= 3.84)	Acquisition Processor

DS-CDMA FPGA PARTITIONING

As far as the modulator is concerned, the most complex case includes three fingers and a digital up converter translating their sum to the required intermediate frequency. The hardware platform layout and the FPGA selection is shown for this case in the Table 6 (the FPGA number in the table corresponds to the FPGA number indicated in the block diagram of Fig. 1).

For all the positions the FPGA’s size of 400 Kgates is enough. The FPGA #4 and #5 are not mounted. The FPGA #6 has been left with a low density (occupation 32%: see Table 6) to reduce the working temperature (it works at frequencies >100 MHz). Sub-cases where only one or two fingers are required have been obtained simply leaving the FPGA #2 and/or the FPGA #3 not mounted.

Concerning the demodulator, many cases have been considered, depending on how many fingers must be received, whether the incoming signal is centred at 70 MHz or at 3.84 MHz, whether the on board commercial Viterbi is available or not, and so on. We give below three of these cases to show the flexibility of the hardware platform. If the received signal is centred at 3.84 MHz no digital down conversion from intermediate frequency is necessary but the digital down conversion driven by the AFC loop, already included in the sample processor. In this case three fingers can be demodulated and the configuration of the hardware platform is (Table 7).

Actually also the FPGA #6 is mounted only to interface one of the two commercial Viterbi available on the board. A second case we implemented has been obtained receiving only one finger at 70 MHz.

Table 6: Modulator FPGA Partitioning

FPGA	Algorithm Number	LE Used	FPGA Size (Kgates)	LE Used (%)	Algorithm
#1	2 … 7	10 810	400	65	MOD Symbol, Chip, Sample Proc.
#2	2 … 7	10 810	400	65	MOD Symbol, Chip, Sample Proc.
#3	2 … 7	10 810	400	65	MOD Symbol, Chip, Sample Proc.
#6	8, 9	5 325	400	32	Digital Up Converter

Table 7: Demodulator FPGA Partitioning: Case 1

IF @ 3.84 MHz – COMMERCIAL VITERBI – THREE FINGERS					
FPGA	Algorithm Number	LE Used	FPGA Size (Kgates)	LE Used (%)	Algorithm
#1	14 … 16 19 … 23	14 477	600	60	DEM Symbol, Chip, Sample Proc. But DeInterleaver, DePunctuting
#2	14 … 16 19 … 23	14 477	600	60	DEM Symbol, Chip, Sample Proc. But DeInterleaver, DePunctuting
#3	14 … 16 19 … 23	14 477	600	60	DEM Symbol, Chip, Sample Proc. But DeInterleaver, DePunctuting
#4	4*(12+13) 24	20 626	600	84	DEM DeInterlever, DePuncturing & Rake Receiver
#5	25	14 595	600	60	Acquisition Processor

Table 8: Demodulator FPGA Partitioning: Case 2

IF @ 70 MHz – COMMERCIAL VITERBI – ONE FINGER					
FPGA	Algorithm Number	LE Used	FPGA Size (Kgates)	LE Used (%)	Algorithm
#1	12 … 16 19 … 23	14 477	600	60	DEM Symbol, Chip, Sample Proc. But DeInterleaver, DePunctuting
#2	17, 18	6 656	600	28	Digital Down Converter
#4	2*(12+13)	6 374	600	26	DEM DeInterlever, DePuncturing
#5	25	14 595	600	60	Acquisition Processor

Table 9: Demodulator FPGA Partitioning: Case 3

IF @ 70 MHz –VITERBI ON FPGA – TWO RX FINGER plus ONE TX FINGER					
FPGA	Algorithm Number	LE Used	FPGA Size (Kgates)	LE Used (%)	Algorithm
#1	14 … 16 19 … 23	14 477	600	60	DEM Symbol, Chip, Sample Proc. But DeInterleaver, DePunctuting
#2	17, 18	6 656	600	28	Digital Down Converter
#3	14 … 16 19 … 23	14 477	600	60	DEM Symbol, Chip, Sample Proc. But DeInterleaver, DePunctuting
#4	2*(12+13) 2 … 7 11	26 400	1000	69	DEM DeInterlever, DePuncturing & MOD Symbol, Chip, Sample Proc. & Viterbi Decoder
#5	25	14 595	600	60	Acquisition Processor
#6	8, 9	5 325	400	32	Digital Up Converter

No rake receiver is necessary and the hardware platform is configured as reported in the Table 8. The FPGA #3 and #6 are not mounted reducing the cost of the equipment. Finally also a case of a complete modem has been taken into account. Its FPGA partitioning is shown in the Table 9. Here the capabilities of the board are fully exploited introducing also a 1000 Kgates device. This kind of modem can be used as slave modulator where the transmitting finger can be locked, through suitable timing and frequency error detector to one of the two received fingers considered as master. The other received finger is the finger transmitted by the slave modulator itself came back after the satellite path and used to extract the frequency and timing differences through comparing with the master one.

WIDE BAND FH-FDMA ALGORITHMS

The platform is used to build the terminals of a wide band FH-FDMA communication system. These terminals are of several kinds, both dedicated to specific links (say payload telecontrol/telemetry link and pilot link for network signalling broadcasting) and for user traffic managing.

The management of the hopping bandwidth grounds on an analog expander, at TX side, and compressor, at RX side, based on frequency multipliers. The signal processing between the source voice/data and the IF unit is enclosed into the BB (Base Band) unit, hosted into the platform. The TX section of the BB unit delivers to the IF unit the hopped signal to be transmitted. This was the very critical point, for TX section, in which the signal processing at sampling rate foresees agile tone synthesising and sync shaping pre-compensation. The RX section of the BB unit is fed with a de-hopped signal coming from the IF unit. The agile tone to do de-hopping comes from the RX agile synthesiser, the same as for TX section. In the following Table 10 the algorithms are listed:

Table 10: Modulator/Demodulator Algorithms

Set Name	Modulator Algorithm		Demodulator Dual Algorithm	
Symbol Processor	1	Voice encoder	9	Voice decoder
	2	Frame formatter	10	Frame de-formatter
	3	FEC encoder	11	Viterbi Decoder
	4	8-FSK Modulator	12	FSK de-Modulator
Hop Processor	5	TranSec	13	TranSec
Sample Processor	6	Agile synthesiser	14	Agile synthesiser
	7	sync pre-compensation	15	sync pre-compensation
Synchronisation	8	Timer	16	Timer

A further class of algorithms is peculiar to the demodulator and can be classified according to the Table 11.

Table 11: Demodulator Peculiar Algorithms

Set Name	Demodulator Algorithm	
Demodulator Sample Processor	17	Digital Down converter
	18	Decimator filter
Demodulator Hop Processor	19	Fading emulator
Demodulator Symbol Processor	20	AGC Loop
	21	DLL (Timing Recovery Loop)
	22	AFC Loop (Frequency Recovery Loop)
	23	Decision logic
	24	Frame Sync Word Detector
Acquisition Processor	25	Hop synchronisation

A brief description of most critical algorithms is given below:

Voice Encoder/Decoder: a standard floating point speech vocoder is hosted into the DSP #2.

FEC Encoder and Viterbi Decoder: the selected encoding algorithm is a *non-binary dual-k, constraint length two convolutional* encoder with code rate ½ [**10**]. Because of the low rate of the encoded symbol stream the Encoder is implemented into the DSP #1. The decoding process is based on a soft-decision Viterbi algorithm. Also in this case the the Decoder is implemented via FW into the DSP #1 because of the low rate of the decoded symbol stream.

FSK Modulator/de-Modulator: the input to the Modulator is the encoded symbol stream coming from the encoder. The output of the Modulator is a stream of digital samples making the single FSK channel, properly allocated into the group bandwidth. The FSK de-Modulator is in charge of detecting the wanted FSK channel performing complex correlations between the sample stream incoming from the RX decimator LP filter and complex tones locally generated. The de-Modulator delivers output values at hopping rate. These values produce the statistics to feed the decision logic. The FSK de-Modulator is the most cumbersome block of the entire signal processing chain. Indeed further these complex correlations, it is in charge of both the computing of early/late correlations, in order to feed the DLL loop, and the computing of further correlations, to feed the AFC loop [**10**]. The high number of complex correlations to be computed requires a smart reuse of a sub-set of correlators.

TranSec: this function drives the agile synthesiser giving the hopping law. It foresees the capability to perform parametric normalisation to adjust the number of hopping tones inside the hopping bandwidth. Furthermore, the capability to skip forbidden bandwidth inside the hopping bandwidth was foreseen, too. The hopping law to be generated can be updated at run time.

Agile synthesiser: the agile synthesiser is made of a standard phase accumulator and a LUT tabulating just a quarter of the round angle. The agile synthesiser development was a good challenge in that it has to work at high clock rate.

Sync pre-compensation: the sync pre-compensation of agile tone magnitude is required to do analog de-hopping into the IF unit in order to cope the sync shaping of output spectrum produced by the hold of DAC over the clock period.

Timer: the timers are split into several sections. The quicker section is driven by the DLL loop. It split the hop time into several time steps in order to allow very fine time corrections.

Fading emulator: in order to take into account the impact of frequency selective fading a proper emulator has been implemented. It grounds on a bank of complex, Gaussian, noise generators to produce the random amplitude modulation, Rician distributed, into a set of several bands. The sample of the signal to be de-hopped is multiplied by the proper random amplitude factor, selected by the frequency value of the tone used to do de-hopping.

Decision logic: The decision logic integrates the correlations delivered from the de-Modulator over the hops of one symbol, so as to produce the metrics to be passed to the decoder and to be detected to locally close the gain, timing and frequency loops of RX. Besides it is in charge of acting a countermeasure against the CW jammer [**10**].

WIDE BAND FH-FDMA FPGA PARTITIONING

The modem board final configuration foresees three FPGA. They are: the FPGA#6, connected to the two DAC, the FPGA#1, connected to the ADC, and the FPGA#5, connected to the fast digital interface. The hardware platform layout and the FPGA selection are shown for this case in Table 12.

FPGA #2, #3 and #4 are not mounted. The FPGA #6 has been left with a low density (occupation 60%: see Table 12) to reduce the working temperature (because of work frequencies >100 MHz).

The Modem Control section is made of both fast, floating point DSP. The master DSP acts as symbol processor, it is in charge of booting all the FPGA, too. A slave DSP is connected with a dedicated fast link to the master DSP. It can be used as generic coprocessor and in particular as vocoder, being connected to an audio interface. Into the following Table 13 the mapping onto the two DSP of the functions demanded to the control section is shown:

Table 12: Modem functions partitioning on FPGA

FPGA	Algorithm Number	LE Used	FPGA Size (Kgates)	LE Used (%)	Algorithm
#1	12, 17, 18	13462	400	80	Hop, Sample Processors
#5	19	14070	400	84	Hop, Sample Processors
#6	4-8, 13-16	10107	400	60	Hop, Sample Processors

Table 13: Modem functions partitioning on DSP

DSP	Algorithm Number	Device	Algorithm
#1	2, 3, 10, 11, 20-25	SHARC 21160	Symbol Processors
#2	1, 9	SHARC 21160	Symbol Processors

ABOUT THE AUTHORS

All the authors belong to the DTA (Digital Technology Area) department of the Space Engineering SpA. C.Campa designed the hardware platform described in the paper. E.Rossini defined the system and carried out the bit true simulations concerning the DS-CDMA algorithms. G.Chiassarini and D.Gianfelici defined the FH-FDMA system and its algorithms. Furthermore G.Chiassarini is the person in charge of the DTA dept. G.D'Orazio implemented the DSP C firmware and M.Marsella implemented the FPGA VHDL firmware. A special acknowledgement goes to B.Tacca, G.D'Angelo and G.Bogo also belonging to the DTA dept. for their contributions to the projects using the material described in this paper.

REFERENCES

[1] D.L.Perry, "*VHDL*", McGraw Hill, Third Edition 1998.
[2] L. Wanhammar, "*DSP Integrated Circuits*" Academic Press, 1999.
[3] "*APEX 20K Programmable Logic Data Sheet*" ALTERA, 2000.
[4] ETSI TS 101 851 –3, "*Satellite Component of UMTS/IMT 2000, Part 3: Spreading and modulation*", V.1.1.1.
[5] A.J.Viterbi, "*Convolutional Codes and Their Performance in Communication Systems*" IEEE Transactions On Communications Technology, VOL. COM-19, NO.5, October 1971.
[6] "*Forward Error Correction Data Book*", Qualcomm CDMA Technologies, 2000.
[7] N.J.Fliege, "*Multirate Digital Signal Processing*" Wiley, 1994.
[8] R.E.Crochiere, L.R.Rabiner, "*Multirate Digital Signal Processing*" Prentice-Hall, 1983.
[9] H.Meyr and G.Ascheid, "*Synchronization in Digital Communications*" Wiley Interscience, 1990.
[10] M.K.Simon, J.K.Omura, R.A.Scholtz, B.K.Levitt, "*Spread Spectrum Communications*" VOLL. I-II-III, Computer Science Press, 1985.
[11] R.R.Ziemer, R.L.Peterson, "*Digital Communications and Spread Spectrum Systems*" MACMILLAN, 1985.
[12] M.Bellanger, "*Digital Processing of Signals*" Wiley, 1989.