XVII IMEKO World Congress
Metrology in the 3rd Millennium
June 22–27, 2003, Dubrovnik, Croatia

DSP-BASED DIGITAL SIGNAL GENERATOR WITH DYNAMIC OUTPUT UPDATING CAPABILITY

Leopoldo Angrisani(1), *Luigi Battaglia*(2)

(1) Dipartimento di Informatica e Sistemistica, Università di Napoli Federico II, Napoli.
(2) Laboratorio Nazionale di Comunicazioni Multimediali – CNIT, Napoli.

ITALY

Abstract − The main aim of the paper is to describe the most significant strategies, both hardware and software, adopted for the implementation of a new digital signal generator based on the TMS320C6701™ digital signal processor (DSP). The novelty of the generator essentially relies on its capability of dynamically updating the output signal, according to the selections the user can make through a suitable graphical interface, with no need of reloading the target application into the DSP. This feature along with the high portability, good flexibility, and, above all, low cost, make the proposed generator appropriate for educational purposes, mainly concerning measurements on telecommunication systems, at universities as well as research centres.

Keywords: *Digital signal generator*, *Digital signal processor*, *W-CDMA*, *Measurements on telecommunication systems*, *Education*.

1. INTRODUCTION

In the framework of the co-operation between the Department of Computer Science and Control System of the University of Naples Federico II and the National Multimedia Communications Laboratory of Naples– C.N.I.T. (Interuniversitary Consortium for the Telecommunications), a scaled version of the W-CDMA [1] down-link transmitter was already designed and implemented on the EVMTMS320C6701™ Evaluation Module (EVM) by Texas Instruments [2-5] according to the 3GPP (Third Generation Partnership Project) specifications [6-7].

In the paper the evolution of the aforementioned transmitter is illustrated. The software running on the DSP (target application) (Fig.1) is able to generate not only W-CDMA signals, but also square wave, amplitude modulation (AM), and quadrature phase shift keying (QPSK) signals. A user-friendly graphical interface, which has been developed in Visual C++ and runs on the PC hosting the EVM board (host application), allows the choice of the output signal.

With reference to the previous W-CDMA transmitter, the target application is considerably changed, and a complex software architecture is designed and implemented in order to allow the transmitter to dynamically update its output, depending on the choice from the host application. This is an important aspect of the research work. As soon as the user changes the selected signal, a command message is sent to the target application, which dynamically updates its output without stopping its execution; no reloading into DSP memory is thus needed. A shared memory allows both the host and target application to exchange data; data are sent and received through the PCI local bus. This way, the W-CDMA transmitter turns into an innovative digital signal generator, which could be very attractive for educational purposes, especially in the field of measurements on telecommunication systems.

In the following, the characteristics of the adopted hardware platform along with the software strategies peculiar to the target and host applications are described in detail. The results of a number of experiments highlighting the capability of the generator of dynamically changing its output according to user requests are also presented.

2. THE HARDWARE

The digital signal generator has been designed and implemented on the TMS320C6701 EValuation Module (EVM) by Texas Instruments [8-9]. It has been used as plug-in-card in a PCI expansion slot of an 800 MHz Pentium III with 265 MB RAM. The EVM is a relatively low-cost demonstration board allowing the assessment of the performance of the ‘C6701 floating point processors thanks to a number of programmable peripheral devices [10].

All analogue signals are synthesized through the 16-bit *stereo audio CODEC* CS4231A™ by Crystal, whose generation frequency can be selected in the range 5.5 – 48 kHz: so it is specifically designed for audio applications.

The CODEC and the DSP cannot exchange data directly. Data are processed by the DSP and then transferred to the serial interface *Multichannel Buffered Serial Port (McBSP)*. The EVM board has two serial modules, but only the McBSP0 can send/receive the data directly to/from the audio CODEC. Data are transferred to the serial interface by the *Direct Memory Access (DMA)*. Four independent channels and an Auxiliary DMA channel are able to transfer data independently from/to DSP memory space, which

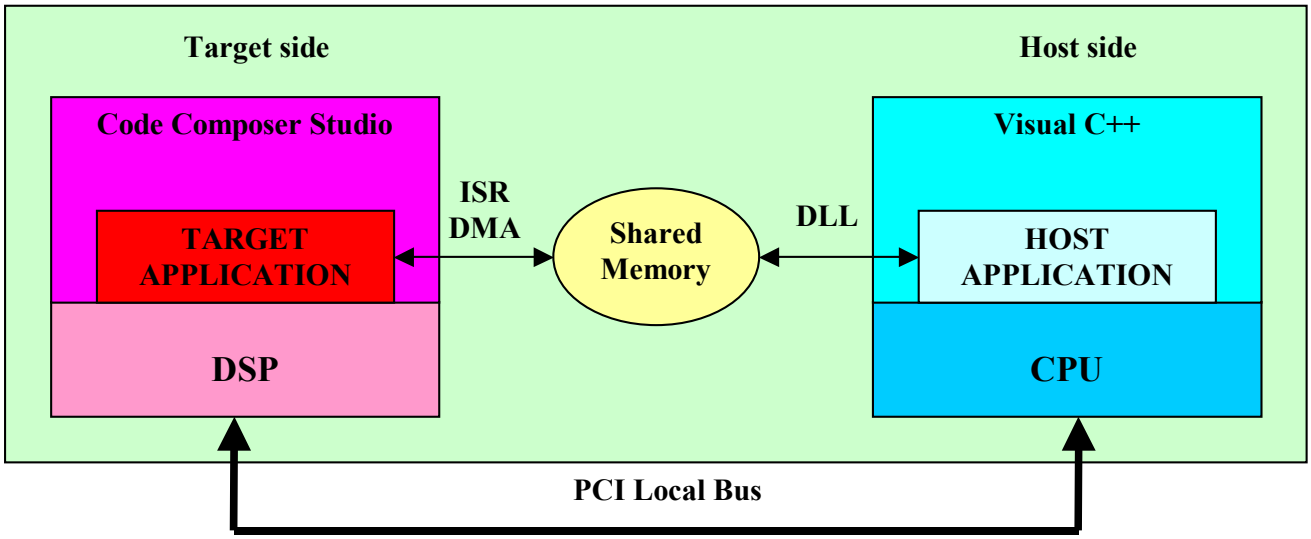

Fig.1. Software architecture of the proposed digital signal generator. The shared memory helps in creating a proper dynamic link between the target application and host application, which confers to the generator the capability of updating its output signal, according to user selections, without reloading the target application into the DSP.

includes external memory addresses and peripheral device registers. Accesses to external data occur through the *External Memory InterFace (EMIF)*. Two DMA channels have been used here in order to transfer data from a fixed memory area to the serial module McBSP0 and the PCI controller respectively [11-12].

The DSP is able to send or receive data to/from the Host application through the PCI local bus thanks to the *S5933 PCI controller* by AMCC [13-14]. On the one hand it interfaces directly to the 5-V, 32-bit PCI bus running up to 33 MHz (Fig.2), on the other it allows either the DSP and the host processor or other PCI devices to access its internal FIFO and mailbox registers by its general-purpose 16/32-bit add-on bus. The S5933 PCI controller have two sets of four 32-bit mailbox registers (*incoming mailbox* and *outgoing mailbox*) and two 32-bit wide and 8-word deep FIFOs (*FIFO read* and *FIFO write*) can be used to exchange data between PCI devices.

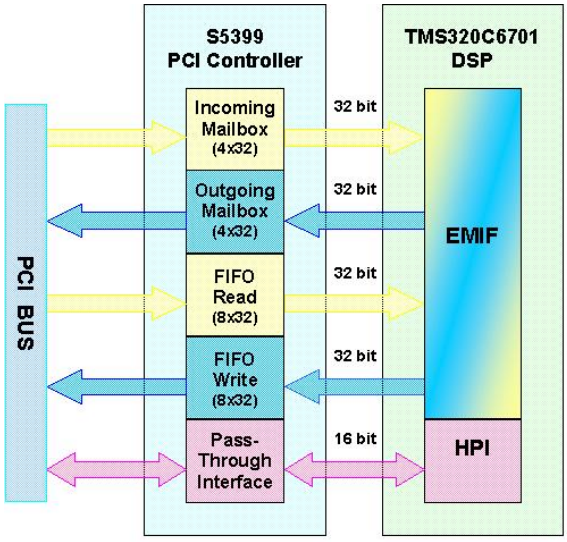

Fig.2. Data paths between the S5399 PCI controller and the TMS320C6701 DSP interface.

The DSP can send/receive data to/from the S5933 PCI controller trough both the EMIF and the *Host Port Interface (HPI)*. The data may be exchanged by two different mechanisms: the PCI master and the PCI slave data transfers. The EMIF allows to map the PCI controller registers into the DSP memory. So the DSP can access them and controls the data transferring. For this reason the EMIF is used to implement the PCI master transfers. Otherwise the HPI is used to implement the PCI slave data exchange. In fact, it allows a PCI external device to access directly the complete DSP memory space thanks to the dedicated Auxiliary DMA channel, but it cannot be controlled by the DSP. The PCI data exchange mechanism adopted here is the PCI master. It may allow better performance than the slave solution because the bus master data transfers through the EMIF are handled in hardware by means of a *Complex Programmable Logic Device (CPLD)*[14-15].

3. THE TARGET APPLICATION

The application running on the DSP is developed in Code Composer Studio 2.0™ Texas Instruments software development environment [9], [17-20]. It is able to output four modulated signals depending on the host choice: square wave, AM, QPSK and W-CDMA signals.

The developed software enlists two different phases (Fig.3). The former allows the proper configuration of the EVM board and all used peripheral devices, while the latter consists of an infinitive loop allowing the output of the selected signal. The main blocks shown in Fig.3 are described in the following subsections.

A. EVM Board Initialisation.

The board is reset and all peripheral device registers are initialised.

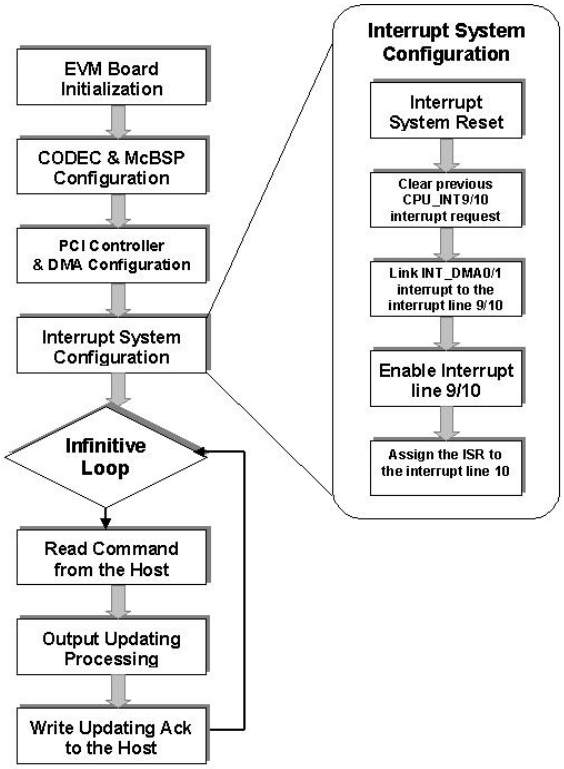

Fig.3. Flow Diagram of the target application

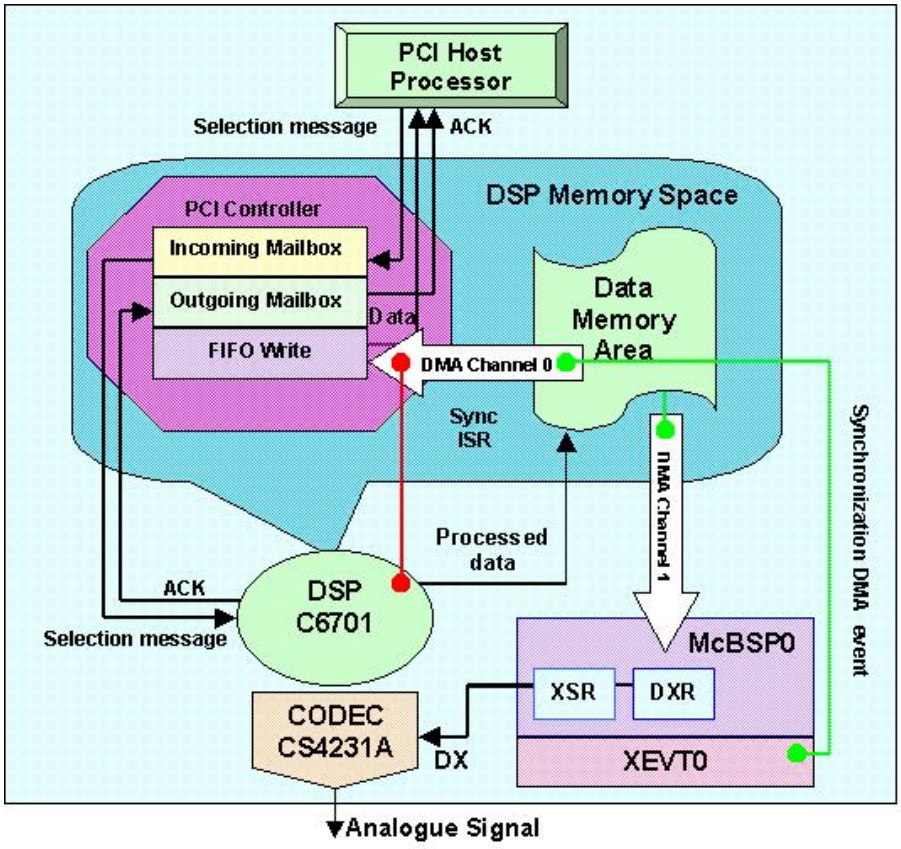

Fig.4. The data paths.

B. CODEC and McBSP configuration.

The serial module McBSP0 is configured in order to send/receive 32-bit data to/from the 16-bit stereo audio CODEC (Fig.4). Data are stored into the Data Transmit Register (DXR) and, then, shifted into the Transmit Shift Register (XSR). At last, they are transferred serially to the CODEC by the Data Transmit (DX) terminal line. The McBSP0 data transfers are synchronised through a signal generated by the CODEC itself. As soon as the DXR is ready to process new data the McBSP0 transmission interrupt flag XEVT0 is enabled in order to synchronise the DMA data transfers.

C. PCI controller and DMA configuration.

To allow the host application to send selection messages and receive acknowledgement (ACK) messages to/from the target application, an incoming and an outgoing mailbox are allocated into the DSP memory space by calculating their starting addresses. Then, a FIFO write is set-up, thus allowing the application running on the DSP to send processed data to the host application. Data transfers are performed by two DMA channels: *DMA channel 0* and *DMA channel 1*[11]. They are configured in such a way as to transfer blocks of 256 elements of 32 bits from the DSP memory area (Fig.4) to the destination addresses: PCI controller's FIFO write and McBSP0 DXR register respectively. Data transfers are synchronised through the McBSP0 event XEVT0, which occurs every 5.3 ms. As soon as the DMA data transfers are performed, the DMA interrupt flags DMA_INT0 and DMA_INT1 is enabled in order to synchronise the ACK messaging.

D. Interrupt System Configuration.

The 'C6701 DSP family has 16 interrupt sources, but only 12 of them can be set by the users. So they have been mapped with the CPU interrupts INT4 - INT15 by using two Interrupt Multiplexed Registers. The CPU interrupts 9 and 10 are here enabled and coupled to the DMA_INT0 and the DMA_INT1 interrupt events. Then, a specific Interrupt Service Routine (ISR) is assigned to the INT9 by refreshing the Interrupt Service

Table (IST). At the end the interrupt system of the DSP is enabled.

E. Read Command from the Host

The target application reads asynchronously the information related to the selected signal from the *incoming mailbox,* thanks to the CPLD which detects when the mailboxes are full or empty and communicates it to the application running on the DSP.

F. Output Updating Processing.

Signal samples are processed by the DSP according to the current selection message and stored into the Data Memory Area. Then they are transferred to the McBSP0 and the PCI FIFO Write by the DMA channels. The data transfers occur continuously in background without overloading the DSP.

G. Write Updating Ack message to the Host.

As soon as the data processing is performed, an ACK message is sent to the outgoing mailbox. This allows the host application to detect possible errors.

H. The Interrupt Service Routine.

The CPU and the DMA channels are able to access the same memory locations. In order to avoid it, the ping-pong technique [11] is adopted here. As soon as the DMA channel 1 has performed its data transfers, an interrupt request is sent to the CPU and the associated Interrupt Service Routine (ISR) switches a pointer to the used memory buffers. So the DSP stores processed data in a buffer different from that is currently being transferred by the DMA channels.

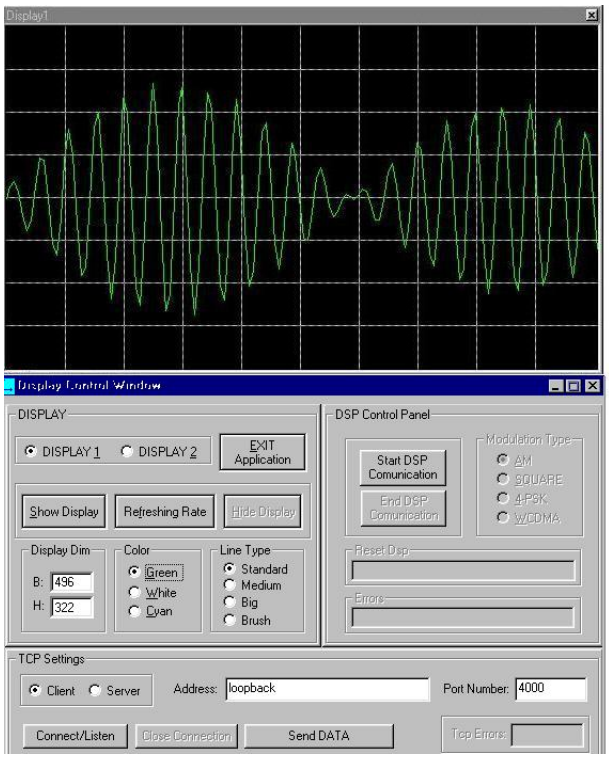

Fig.5. Graphical User Interface.

4. THE HOST APPLICATION

The host application consists of a user friendly graphical interface (Fig.5), which has been implemented through the Visual C++ by the Microsoft Visual Studio 6.0™. It provides a number of radio-buttons allowing to select the signal that the target application has to synthesise, and a screen similar to a digital oscilloscope allowing to plot the data received from the target application. When the START DSP Communication button is pushed, the host application resets the EVM board, loads the target application into the DSP internal program memory and sets up a driver connection to it. At first, the target application synthesises a 480 Hz square signal representing the default output. As soon as the user changes the selected waveform by clicking on the proper radio-button a little binary information is sent to the *incoming mailbox*. Then the host application waits for an acknowledgement message from the *outgoing mailbox*: in this way possible errors during the communication processing can be detected. Information about possible errors and the selected signal are sent asynchronously, whereas processed data representing the target application's output are read continuously from the *FIFO write* and plotted on the screen with a refreshing rate fixed to 5.3 ms. The implemented graphical interface also provides a set of plotting facilities.

The driver connection to the EVM board is closed as soon as the END DSP communication button is pushed. The host application performs all previous operations by means of the functions implemented into the specific Dynamic Link Library – DLL *evm6x.dll* [8], [16].

The host application is also able to manage data flow through the Internet. In fact the Client/Server radio-buttons allow to open and close a remote TCP connection in client and server mode respectively. The remote client application can receive the data plotted locally on the graphical interface and eventually set the waveform type. The server application manages the communication with the target application.

5. EXPERIMENTS

Figs.6,7,8,9 show the results related to different experiments. Different transitions from a particular output signal to another one have been tested in order to assess the reliability and efficacy of the proposed digital signal generator. As soon as the user selects a new output signal, the generator dynamically updates its output. As said before, updating does not need resetting the target application running on the DSP and, consequently, reloading a new version into the DSP.

The possible output signals have the following characteristics:

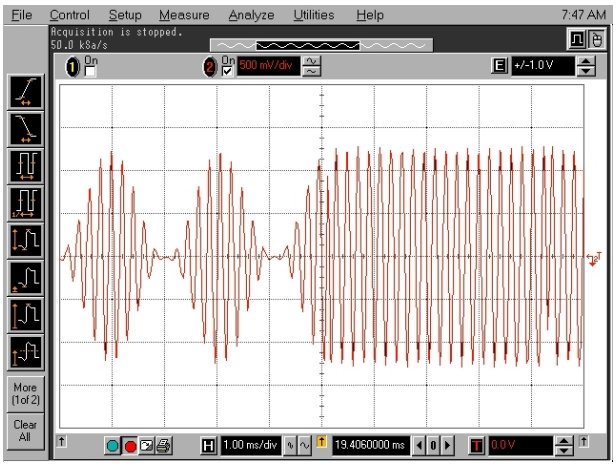

Fig.6. Dynamic updating of the output from the AM to the QPSK signal

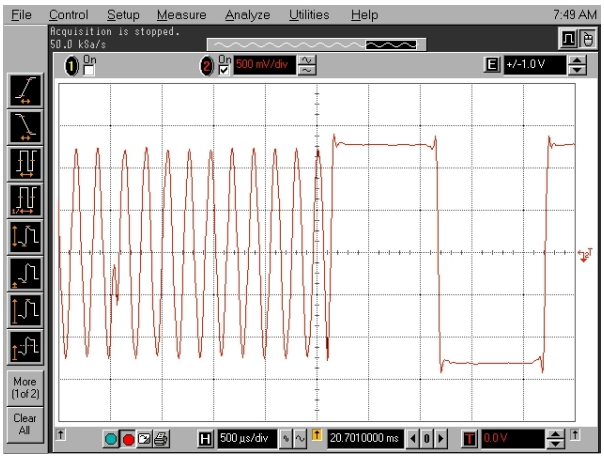

Fig.7. Dynamic updating of the output from the QPSK to the square signal.

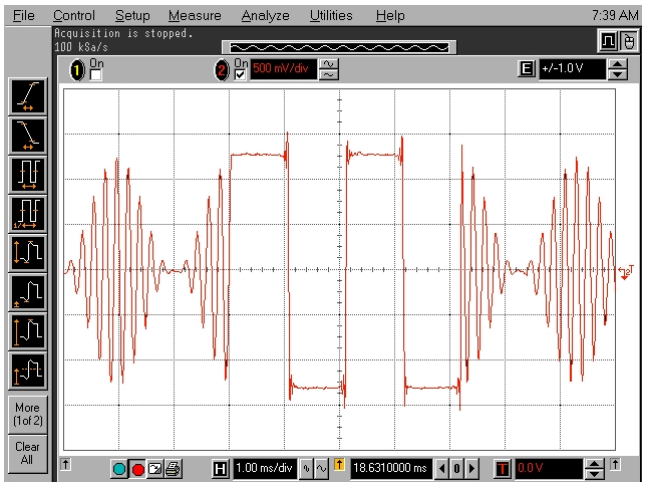

Fig.8. Dynamic updating of the output from the AM to the square signal.

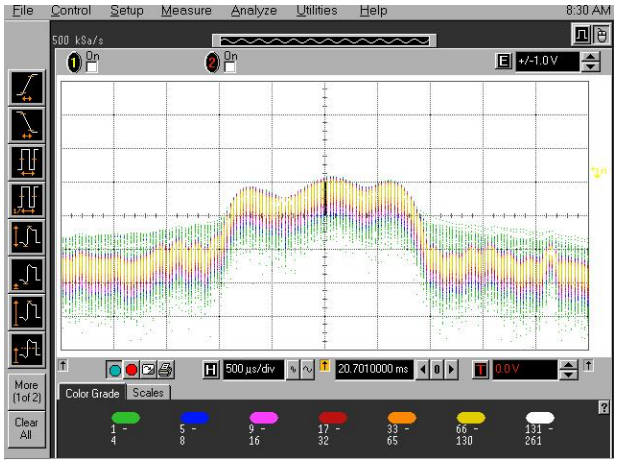

Fig.9. Spectrum of the W-CDMA signal.

- square wave signal at a frequency rate of 480 Hz;
- AM signal with a carrier frequency of 4800 Hz and a 480 Hz sine modulating signal;
- QPSK signal with a carrier frequency and a bit-rate of, respectively, 4800 Hz and 960 bit/s. The information source is an NRZ signal;
- W-CDMA signal chip-rate equal to 6 kchip/s with a bit rate fixed to 1.5 kbit/s(the spreading factor has been fixed to 8). The measured bandwidth is equal to 6.4 kHz. As the generation rate is fixed to 48 kHz, the centre frequency has been fixed to 12 kHz.

Signals are shown in the time domain by a digitising oscilloscope, namely Agilent Technologies 54820A™ (2 GS/s maximum sample rate, 500 MHz bandwidth). The sampling rate has been chosen equal to 100 kS/s in the experiments providing for a transition of the output signal from AM to square wave, and to 50 kS/s for the other experiments. The vertical scale has been fixed to 500 mV/div. The horizontal scale has been chosen equal to 500 µs/div for the output signals turning from QPSK to square wave, and to 1 ms/div for the others. At last, the spectrum of the W-CDMA signal is shown in Fig.9; it has been obtained from the same oscilloscope, used as an FFT analyser and in colour grade mode. The sampling rate has been chosen equal to 500 kS/s. The vertical scale has been fixed to 20.0 dBm/div. The horizontal scale has been set to 2.0 kHz/div, respectively. The reference level has been chosen equal to -28.0 dBm.

6. CONCLUSIONS

The paper has presented the design and implementation of a DSP-based digital signal generator for didactic applications. It allows the user to change the desired output signal without stopping the application running on the DSP; a dynamic output updating is evidenced. The software

strategies allowing the real-time data exchange between target and host applications have been discussed in detail.

Future research activities will be oriented to:

- stress the software architecture in order to exhaustively assess its performance and reliability;
- enhance the interaction between a remote host application and the developed local host application;
- use the proposed platform to implement an automated distributed measurement system, where processing is distributed on a cluster of DSP and signals are acquired by means of an array of sensors which can be located anywhere.

Acknowledgements

The authors wish to thank Prof. O.Andrisano, Prof. F.Davoli, Prof. L.Paura, Prof. S.Zappatore, and the whole CNIT staff for their valuable support, and Dr. Alessandro Cerulo for its significant contribution in developing the graphical user interface.

REFERENCES

[1] L.Angrisani, L.Battaglia, A.Iudici, "Design, Implementation and Metrological Characterization of a Cost-Effective DSP-based W-CDMA Transmitter ", *12th IMEKO TC4 International Symposium*, Zagreb, Croatia, September 25–27, 2002.

[2] L.Angrisani, M.D'Apuzzo, M.D'Arco, "Advanced digital processing solutions for performance testing and troubleshooting of CDMA Transmitters", in Proc. of *IEEE IMTC/2001*, Budapest, Hungary, May 21-23, 2001, vol.I, pp.534-539.

[3] G.Betta, C.Liguori, A.Pietrosanti , "A Multi-Application FFT Analyzer Based on a DSP Architecture", IEEE Transactions on Instrumentation and Measurement, Vol.50, No.3, June 2001.

[4] D.Dardari, A.Conti, "Experimental DSP-based CDMA Modem for Ka-band Satellite Systems: Low Complexity Code Acquisition and Tracking Scheme", Personal, Indoor and Mobile Radio Communications, 2001 12th IEEE International Symposium Vol. 2, Sep/Oct 2001, Page(s): 100–104, vol.2.

[5] Designing and testing 3GPP W-CDMA Base Stations – Application Note 1335 – Agilent Tecnologies.

[6] 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; UTRAN Overall Description (Release 1999).

[7] 3G TS 25.213 V3.4.0 (2000-12) Technical Specification 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Spreading and modulation (FDD) (Release 1999).

[8] TMS3206201/6701 Evaluation module technical reference - SPRU305 – December 1998.

[9] Code Composer Studio User's Guide – SPRU328B – February 2000.

[10] TMS320C6000 Peripherals Reference Guide - SPRU190D – February 2001.

[11] David Bell, "TMS320C6000 DMA Example Applications", Application Report SPRA529A, April 2002, Texas Instruments.

[12] D.Bell, E.Biscondi, "TMS320C620x/TMS320C6701 DMA and CPU: Data Access Performance" Application Report SPRA614A , August 2000, Texas Instruments.

[13] S5933 PCI Controller Data Book, Applied Micro Circuits Corporation, 1997.

[14] Brian G.Carlson, "Interfacing the TMSC6000 EMIF to a PCI Bus Using the AMCC S5933 PCI Controller", Application Report SPRA479A, September 2001, Texas Instruments.

[15] Z.Nikolic, "TMS320C6201/6701 DSP Host Port Interface (HPI) Performance", Application Report SPRA449, April 1998, Texas Instruments.

[16] Deborah Keil, "Real-Time Data Exchange", White Paper SPRY012, February 1998, Texas Instruments.

[17] Naim Dahnoun: Digital Signal Processing Implementation – Prentice Hall

[18] W.H.Press, W.T.Vetterling, S.A.Teukolsky, B.P.Flannery, Numerical Recipes in C(II edition), Cambridge University Press, 1999.

[19] W.H.Press, W.T.Vetterling, S.A.Teukolsky, B.P.Flannery, Numerical Recipes – Example Book(C), Cambridge University Press, 1999.

[20] J.G.Proakis, D.G.Monolakis, Digital Signal Processing – Principles, Algorithms, and Applications, Prentice Hall, 1996.

Authors: Prof. **Leopoldo Angrisani**, Department of Computer Science and Control System of the University of Naples Federico II, Via Claudio 21, 80125 Naples, Italy. Phone number: +390817683170, Fax number: +390812396897, E-mail address: angrisan@unina.it;

Researcher Junior **Luigi Battaglia**, National Multimedia Communications Laboratory – C.N.I.T. (Interuniversitary Consortium for the Telecommunications), Via Diocleziano 328, Naples, Italy. Phone number: +390812303311; Fax number: +390816171255; E-mail address: luigi.battaglia@cnit.it.