DESIGN OF DIRECT DIGITALLY SYNTHESIZED ARBITRARY WAVEFORM GENERATOR USING FIXED-POINT DIGITAL SIGNAL PROCESSOR

S. S. Prasad[1] and S. K. Sanyal[2]
[1]Department of Electronics and Communication Engineering
National Institute of Technology, Jamshedpur-831014, India
[2]Department of E.T.C.E, Jadavpur University, Kolkata-700 032, India
Email: jaiswal_shyam@hotmail.com, s_sanyal@ieee.org

Abstract: *This paper describes a prototype design to demonstrate the design of arbitrary waveform generator based on Direct Digital Synthesis (DDS) technique implemented on Digital Signal Processor. The key idea is the utilization of Look-up Table (LUT) supported with Code Composer Studio (CCS) platform. The arbitrary waveform is synthesized by storing samples in the memory of TMS320VC5416 Digital Signal Processor (DSP) in LUT form. The processor while executing the program recalls the stored samples sequentially and presents them to the Digital-to-Analog (D/A) Converter and the D/A converter generates the desired arbitrary waveform. The experimental results have been included.*

Keywords : *Direct Digital Synthesis, Arbitrary Waveform Generator, Look-Up Table, Code Composer Studio.*

1. Introduction

The phenomenal growth in the interest behind DDS has been largely driven by the corresponding impetus in digital and wireless communications during the last three decades. The synthesis of a sinusoid of a desired frequency is a fundamental requirement in communication systems. The recent shift to all-digital systems has also favored the growth in DDS research. The interest in DDS research is also driven by the advantage of precise and rapid control of the phase, frequency and amplitude of the digitally synthesized output [1]. The first paper on DDS using the most popular technique, the Sine Look-up Table method, was introduced by Tierney, Rader and Gold [2] in 1971. This method synthesizes a sine wave by successively scanning through a look-up table stored in Memory and then converting the recalled sine samples to an analog waveform through a Digital to Analog Converter (DAC).

LUT based DDS is one of the most popular techniques to synthesize AC signals for instrumentation, measurement and digital communication applications. Synthesized waveforms by DDS technique enjoy the benefits; like high frequency resolution, precise frequency control and low complexity [3-4]. DDS based designs are easy to implement on Digital Signal Processor. DSP has immense powers of computation and process control along with the power of re-programmability to reconfigure the changes in design without making a major redesign. The incorporation of Code Composer Studio (CCS) Integrated Development Environment (IDE) platform in this design technique enhances the development process without going through the rigorous knowledge of Assembly Level Instruction pertaining to the relevant processor. The knowledge of conventional C language is sufficient to carry out the design [14].

The DDS based sinusoid generation using TMS320VC5416 and implementing various modulations for digital communication has been well described in the prior work of Prasad and Sanyal [5-9]. In this paper an attempt has been made to utilize the LUT based DDS technique for arbitrary waveform generation.

Arbitrary Waveform Generators (AWG) allows the user to emulate virtually any system output [10]. The wave shape is created by a system based on the processor and memories, which provides the programmability, repeatability and reliability. The design of arbitrary waveform is Digital, and any wave shape can be obtained. Processor facilitates control of synthesis parameter by mere change in source code of program. In this paper the arbitrary waveform synthesis is demonstrated with an example and the result is included in the paper. The synthesis approach is generalized so that any mathematical function can be quickly implemented by DDS technique on the DSP as demonstrated in this experiment.

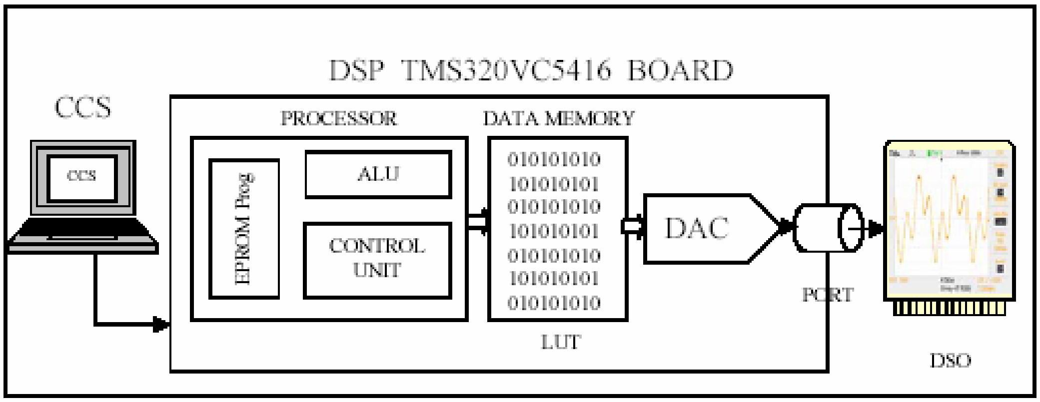

Fig. 1 Schematic diagram of DSP based AWG

2. Schematic of Arbitrary Waveform Generator

The Fig.1 shows a schematic of DSP based Arbitrary Waveform Generator. The CCS-IDE generates a TI-tagged Common Object Format File (COFF), which is downloaded to the EPROM programmer of the DSP board. The processor on execution of the program sequentially presents the samples of AWG-LUT to the DAC. The output of DAC is connected to Digital Storage Oscilloscope (DSO) which records the desired arbitrary waveform.

3. Look-up Table for Arbitrary Waveform Generator

The Look-up table in general is defined as an array of data values that can be quickly accessed to convert data from one form to another. For programmable processors, a look-up table is defined as a data structure. It is usually an array stored in memory, used to replace a runtime computation with a more simple lookup operation. The speed gain is significant, as retrieving a value from memory is faster than undergoing an expensive computation. Trigonometric tables are classic example of a lookup table. Calculating the sine of a value using a mathematical formula can be prohibitively slow in some applications. Most computers which only perform basic arithmetic operations, cannot directly calculate the sine of a given value. Instead, they use the CORDIC algorithm [11] or a complex formula like the Taylor series [12] to compute the value of sine to a high degree of precision:

$$\sin(x) \approx x - \frac{x^3}{6} + \frac{x^5}{120} - \frac{x^7}{5040} \qquad \textit{for x very close to } 0 \qquad \text{----------- (1)}$$

However, this approach can be expensive to compute, especially on slow processors.

There are many applications, particularly in signal processing, that need to compute many thousands of sine values per second. A common solution is to first precalculate the sine value for required phase and angle resolution and store it in the form of LUT. Later, when the program wants the sine of a value, it uses the LUT table to retrieve the corresponding data from the table instead of calculating it using a mathematical formula.

Any arbitrary waveform function can be synthesized using DDS based LUT method, on CCS platform. The arbitrary waveform synthesis is demonstrated with an example; let f(t) be a periodic arbitrary waveform function as described by (2). This is a combination of three cosine functions, two sine functions, and a dc term, which can be synthesized by DDS technique.

$$f(t) = 5 + 1.5 * \cos(2\pi t) + 0.5 * \cos(2 * 2\pi t) +$$

$$2*\sin(2\pi t)+2*\sin(3*2\pi t)+0.3*\cos(3*2\pi t)$$ ------------ (2)

In discrete form the equation can be modified as

$$f\left(\frac{k}{L}\right)=5+1.5*\cos\left(\frac{2\pi}{L}*k\right)+0.5*\cos\left(2*\frac{2\pi}{L}*k\right)+$$
$$2*\sin\left(\frac{2\pi}{L}*k\right)+2*\sin\left(3*\frac{2\pi}{L}*k\right)+0.3*\cos\left(3*\frac{2\pi}{L}*k\right)$$ -------------- (3)

where 'L' represents the total number of discrete samples and 'k' is an integer which can take values from 0 to L-1. Fig. 2 shows part of an actual plot for (3) and Fig.3 shows a part of the MATLAB simulated waveform for (3) when L=180 and 'k' is varied in a loop from 0 to 179 endlessly.

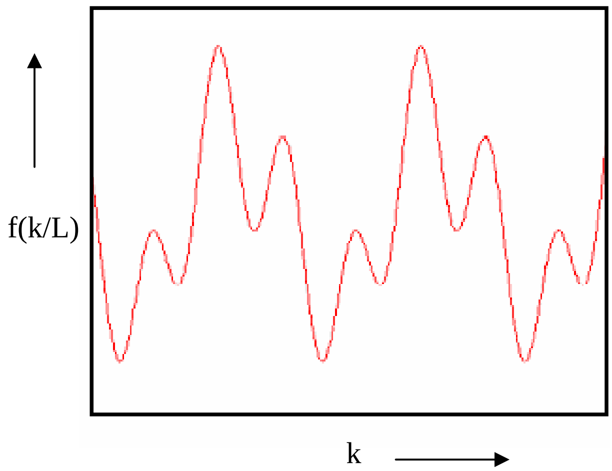

Fig. 2 Actual arbitrary waveform obtained from equation (3)

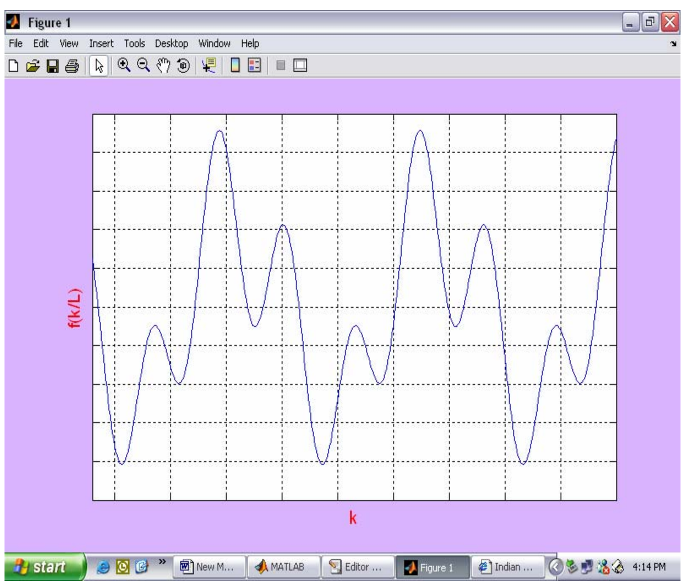

Fig. 3 MATLAB simulated arbitrary waveform for equation (3)

To create an LUT compatible for implementation on the TMS320VC5416 processor, the (3) is multiplied by a scalar factor 'A' to maximize the dynamic range of the output of arbitrary waveform synthesizer.

In order to keep all the values of LUT in positive region [13] the (3) is further modified as

$$LUT(n) = A * f(n) \quad ...(4)$$

where n=k/L.

This processor is also a fixed-point DSP [13] so the fraction resulting from (4) is to be truncated and the equivalent Hexadecimal values are to be tabulated in LUT, hence (4) can be modified as

$$LUT(n) = hex\langle floor(A * f(n))\rangle \quad . . . (5)$$

A program to calculate the discrete sample value of the final scaled function as described by (5) is developed in C- Language. This can accept any value of 'L' and create a LUT by changing 'k' from 0 to L-1 as shown in Program 1. For our experiment L=180 and A=256 is chosen. Table-I shows the rearranged tabulated values of LUT in the relevant Hexadecimal format, which can be referenced by the Processor for implementing DDS.

Program 1: C-Program for generation of LUT

```
#include<stdio.h>
#include<math.h>
#include<conio.h>
#define table(k) (int) floor(256*( 5 + ( (1.5*cos(2*3.14159*k/180)) +
    (0.5*cos(2*2*3.14159*k/180)) +(2*sin(2*3.14159*k/180))+
    (2*sin(3*2*3.14159*k/180))+ (0.3*cos(3*2*3.14159*k/180)))))

void main()
{
int i;
FILE  *ft=fopen("sample.txt","w");
for(i=0; i<180;i++)
{
fprintf(ft,"A[%d]=0x0%x;\n",i,table(i));
}
}
```

Table 1. LUT for synthesizing arbitrary wave described by equation (5)

A[0]=0X074C	A[30]=0X06EE	A[60]=0X0608	A[90]=0X03B3	A[120]=0X0291	A[150]=0X0377
A[1]=0X0793	A[31]=0X06AE	A[61]=0X0630	A[91]=0X036C	A[121]=0X02C1	A[151]=0X035F
A[2]=0X07D7	A[32]=0X0670	A[62]=0X0657	A[92]=0X0326	A[122]=0X02F2	A[152]=0X0348
A[3]=0X0817	A[33]=0X0633	A[63]=0X067C	A[93]=0X02E2	A[123]=0X0321	A[153]=0X0334
A[4]=0X0854	A[34]=0X05F8	A[64]=0X069E	A[94]=0X02A1	A[124]=0X034F	A[154]=0X0323
A[5]=0X088D	A[35]=0X05BF	A[65]=0X06BD	A[95]=0X0262	A[125]=0X037B	A[155]=0X0315
A[6]=0X08C2	A[36]=0X058A	A[66]=0X06D9	A[96]=0X0227	A[126]=0X03A5	A[156]=0X030B
A[7]=0X08F1	A[37]=0X0559	A[67]=0X06F0	A[97]=0X01F0	A[127]=0X03CD	A[157]=0X0306
A[8]=0X091A	A[38]=0X052C	A[68]=0X0703	A[98]=0X01BE	A[128]=0X03F1	A[158]=0X0305
A[9]=0X093E	A[39]=0X0504	A[69]=0X0711	A[99]=0X0190	A[129]=0X0411	A[159]=0X0308
A[10]=0X095B	A[40]=0X04E0	A[70]=0X071A	A[100]=0X0168	A[130]=0X042E	A[160]=0X0311
A[11]=0X0972	A[41]=0X04C2	A[71]=0X071E	A[101]=0X0145	A[131]=0X0447	A[161]=0X031F
A[12]=0X0983	A[42]=0X04A9	A[72]=0X071C	A[102]=0X0127	A[132]=0X045C	A[162]=0X0332
A[13]=0X098D	A[43]=0X0496	A[73]=0X0714	A[103]=0X0110	A[133]=0X046C	A[163]=0X034B
A[14]=0X0990	A[44]=0X0488	A[74]=0X0707	A[104]=0X00FE	A[134]=0X0478	A[164]=0X0369
A[15]=0X098C	A[45]=0X0480	A[75]=0X06F3	A[105]=0X00F3	A[135]=0X047F	A[165]=0X038C
A[16]=0X0982	A[46]=0X047D	A[76]=0X06DA	A[106]=0X00ED	A[136]=0X0483	A[166]=0X03B5
A[17]=0X0971	A[47]=0X0480	A[77]=0X06BA	A[107]=0X00EE	A[137]=0X0482	A[167]=0X03E2
A[18]=0X095A	A[48]=0X0488	A[78]=0X0696	A[108]=0X00F4	A[138]=0X047C	A[168]=0X0414
A[19]=0X093D	A[49]=0X0496	A[79]=0X066C	A[109]=0X0100	A[139]=0X0473	A[169]=0X044B

A[20]=0X091A	A[50]=0X04A8	A[80]=0X063D	A[110]=0X0111	A[140]=0X0467	A[170]=0X0486
A[21]=0X08F2	A[51]=0X04BE	A[81]=0X0609	A[111]=0X0128	A[141]=0X0457	A[171]=0X04C5
A[22]=0X08C5	A[52]=0X04D9	A[82]=0X05D1	A[112]=0X0143	A[142]=0X0444	A[172]=0X0507
A[23]=0X0894	A[53]=0X04F8	A[83]=0X0595	A[113]=0X0162	A[143]=0X042E	A[173]=0X054C
A[24]=0X085E	A[54]=0X0519	A[84]=0X0556	A[114]=0X0186	A[144]=0X0416	A[174]=0X0593
A[25]=0X0826	A[55]=0X053E	A[85]=0X0514	A[115]=0X01AD	A[145]=0X03FD	A[175]=0X05DC
A[26]=0X07EA	A[56]=0X0564	A[86]=0X04D0	A[116]=0X01D7	A[146]=0X03E3	A[176]=0X0625
A[27]=0X07AD	A[57]=0X058C	A[87]=0X048A	A[117]=0X0203	A[147]=0X03C7	A[177]=0X0670
A[28]=0X076E	A[58]=0X05B5	A[88]=0X0442	A[118]=0X0231	A[148]=0X03AC	A[178]=0X06BA
A[29]=0X072E	A[59]=0X05DF	A[89]=0X03FA	A[119]=0X0261	A[149]=0X0391	A[179]=0X0704

4. Code Composer Studio

Code Composer Studio (CCS) software is an Integrated Development Environment (IDE) supporting Texas Instruments TMS320VC5416 based DSP kit. CCS slashes the development and integration time for the DSP software. CCS integrates all host and target tools in a unified environment to simplify DSP system configuration and application design [14]. The environment integrates traditional tools for editing, building, debugging, code profiling and project management. Code Composer Studio IDE user interface such as signal probing, multi-processor support, data, system visualization, flexible C-based scripting language for automated testing and customization are very helpful. To synthesize arbitrary waveform a project named “awg.pjt” is created in CCS. The C- program to implement the DDS based logic is written on a C-source file of CCS and saved as “awg.c”. An equivalent assembly file is generated and saved as “awg.asm” as shown in Program-2. Then the assembler translates the assembly file into Common Object Format File (COFF); the linker then combines the COFF and library modules into a single executable COFF. A hex conversion utility is used to convert a COFF object into TI-tagged object formats. The converted file “awg.out” is ready to be downloaded to the EPROM programmer of the DSP kit.

Program 2: Assembly program for arbitrary wave synthesis

```
;*************************************
;* TMS320C54x ANSI C Codegen Version 3.70
;* Date/Time created: Mon May 28
18:40:13 2007            *
;*************************************
            .mmregs
    FP      .set      AR7
            .c_mode
            .file     "awg_jour.c"
    ;       c:\new
    ti\c5400\cgtools\bin\acp500.exe
     -
    @C:\DOCUME~1\ADMINI~1\LOCAL
    S~1\Temp\TI3176_4

            .sect     ".text"
            .global   _main
            .sym      _main,_main, 36, 2, 0
            .func     3

;*************************************
    ;*    FUNCTION    DEF:    _main
    *
;*************************************

    _main:
            .line     2
            .sym      _i,0, 4, 1, 16
            .sym      _A,1,  52,  1,  2880,,
    180
            .sym      _x,181, 4, 1, 16
        PSHM     AR1
        PSHM     FP
        ADDM     #-182,*(SP)
        NOP
        NOP
        MVMM     SP,FP
        ST       #1868,*SP(1)        ; |7|
        ST       #1939,*SP(2)        ; |8|
            :
            :
            :
            :
```

```
        ; Similarly all 180 values of
LUT are stored in SP and FP, not shown
here  just to save space
        :
        :
        :
        :
        :
    ST      #1573,*FP(177)      ; |183|
    ST      #1648,*FP(178)      ; |184|
    ST      #1722,*FP(179)      ; |185|
    ST      #1796,*FP(180)      ; |186|

L1:
    SSBX    SXM
    LD      #180,A
    ST      #1,*SP(0)           ; |194|
    SUB     *SP(0),A            ; |194|
    BC      L1,ALEQ             ; |194|
    ; branch occurs ; |194|

L2:
    LDM     SP,A
    ADD     #1,A
    ADD     *SP(0),A            ; |196|
    STLM    A,AR1
    NOP
    NOP
    LD      *AR1,A
    STL     A,*FP(181)
    PORTW   *FP(181),04H        ; |197|
    LD      #180,A
    ADDM    #1,*SP(0)           ; |198|
    SUB     *SP(0),A            ; |198|
    BC      L2,AGT              ; |198|
    ; branch occurs ; |198|
    B       L1                  ; |199|

branch occurs                   ; |199|

.endfunc 202,000040400h,184
;*************************************
;*TYPE  INFORMATION
;**************************************
```

5. Experimental Setup with TMS320VC5416 based DSP Kit

In this experiment, a Texas Instruments 16 bit fixed-point Digital Signal Processor (DSP); TMS320VC5416 is used to implement Direct Digital Synthesis based on the Look up table method for generating arbitrary waveform. The C5416 DSP has a high degree of operational flexibility and speed. It combines an advanced modified Harvard architecture (with one program memory bus, three data memory buses; four address buses), a CPU with application-specific hardware logic, on-chip memory, on-chip peripherals and a highly specialized instruction set [15]. A separate program and data spaces allow simultaneous access to program instructions and data, providing a high degree of parallelism. Instructions with parallel store and application-specific instructions fully utilize this architecture. In addition, data can be transferred between data and program spaces. Such parallelism supports a powerful set of arithmetic, logic, and bit-manipulation operations that can all be performed in a single machine cycle. Also, the C5416 DSP includes the control mechanisms to manage interrupts, repeated operations and function calls.

For this experimental setup the DSP kit is connected to the serial port of the Personal Computer (PC) in order to download the TI tagged format program to run on the processor. The CCS debugging tool is used to debug the program by using a controlled execution and the monitoring support is provided by the debugging environment. The TMS320VC5416 based DSP hardware kit has a dual port DAC. The output port address of DAC is 04H and 05H; for this experiment Port 04H is used. The file “awg.out” is downloaded to the EPROM programmer of the DSP kit and executed.

6. Experimental Results

On execution of the program, the processor scans through the LUT and presents the hexadecimal sample to DAC of the DSP kit. The arbitrary waveform is obtained at the DAC output. The analog output providing the arbitrary waveform at Port 4 is recorded on Tektronix TPS 2014 digital storage oscilloscope. Fig.4 shows an experimentally obtained arbitrary waveform for (5) which is modified version of (3) for processor implementation. The actual waveform obtained from (3) is already shown in Fig. 2. As a comparison between the two it is found that both the waveforms are almost similar. For a known function like (3) the MATLAB simulated waveform is already shown in Fig.3. The simulated waveform is also very close to the experimental one.

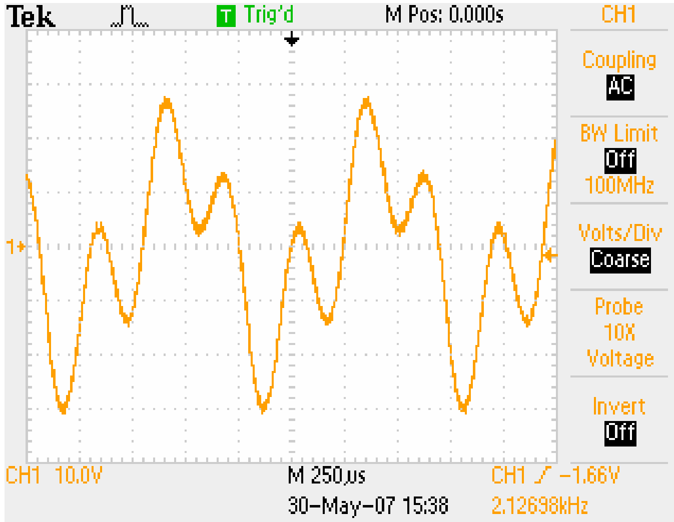

Fig. 4 Experimentally obtained arbitrary waveform for equation (5)

7. Conclusion

A Direct Digital Synthesis technique based arbitrary waveform generation has been described and implemented, using Texas Instruments TMS320VC5416 digital signal processor. The change in design parameter can be incorporated by mere change in Program, so hardware modification is not required, thus it's cost effective. The commercially available AWG is customized only for pre-defined arbitrary waveform generation. However, our approach has the advantage of synthesising any type of arbitrary waveform being expressed by mathematical function, on the DSP using DDS technique supported with CCS-IDE. For a known function, the actual, simulated and experimentally generated waveforms have been obtained and all are found to be almost similar.

8. References

[1] L.Cordesses, "Direct digital synthesis: A tool for periodic wave generation (part-1)", *IEEE Signal Processing Magazine*, Vol. 21, Issue 4, July 2004, pp. 50 – 54.

[2] J.Tierney, C.M.Rader, and B.Gold, "A digital frequency synthesizer", *IEEE Trans. on Audio Electroacoustics*, Vol. AU-19, 1971 pp. 48-57.

[3] C.L.Turner, "Recursive discrete-time sinusoidal oscillators", *IEEE Signal Processing Mag.*, Vol. 20, No. 3, May 2003, pp. 103–111.

[4] J. Vankka, "Methods of Mapping from Phase to Sine Amplitude in Direct Digital Synthesis", *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, Vol. 44, March 1997, pp. 526-534.

[5] S.S.Prasad and S.K.Sanyal, "A Novel Technique for the Generation of Good Quality Tunable Sinusoid based on DDS using Digital Signal Processor", *Proceedings of 2nd IEEE International Conference on Information & Communication Technologies: From Theory to Applications, ICTTA'06*, Damascus, Syria, April 2006, Vol. 1, pp. 717-718.

[6] S.S.Prasad and S.K.Sanyal, "Direct Digital Synthesis of Carrier and ASK Modulation using TMS320VC5416 Digital Signal Processor", *Proceedings of IEEE sponsored International Conference on Computer & Communication Engineering, ICCCE 2006*, Kuala Lumpur, Malaysia, , May 2006, Vol. 2, pp.1030-1032.

[7] S.S.Prasad and S.K.Sanyal, "A Novel Direct Digital Synthesized FSK Modulator using Code Composer Studio for applications in Digital Communication: Extension to 4-FSK Modulation", *Proceedings of IEEE sponsored, 3rd International Conference on Computers and Devices for Communication (CODEC)*, Kolkata, India, Dec. 18-20. 2006, pp.581-584.

[8] S.S.Prasad and S.K.Sanyal, "Development of a prototype direct digital synthesized QAM modulator using code composer studio for application in mobile communication systems", *Proceedings of IEEE 4th International Conference on Electrical & Computer Engineering (ICECE-2006)*, Dhaka, Bangladesh, Dec. 19-21, 2006, pp. 72-75.

[9] S.S.Prasad and S.K.Sanyal, "A Novel application of TMS320VC5416 Digital Signal Processor for Direct Digital Frequency Synthesis", *Proceedings of IEEE sponsored International Conference on Computer & Communication Engineering, ICCCE'06*, Kuala Lumpur, Malaysia, May 9-11, 2006, Vol.2, pp. 962-965.

[10] A.D.V.N. Kularatna and N.Kulratna, *Digital and Analogue Instrumentation: testing and measurement*, The Institution of Engineering and Technology, London, 2003.

[11] N. Takagi,, T. Asada and S. Yajima,, "Redundant CORDIC Methods with a Constant Scale Factor for Sine and Cosine Computation", *IEEE Tranactions. Computers*, Vol. 40, No. 9, Sept. 1991, pp. 989-995.

[12] R.Tehrani, L.C.Ludeman, "Use of generalized Taylor series expansion", *Proceedings of IEEE International Symposium on Circuits and Systems, ISCAS '92.* 3-6, May 1992, Vol. 2, pp. 979 – 982.

[13] *TMS320VC5416 Fixed-Point Digital Signal Processor*, Data Manual, Literature Number: SPRS095J March 1999 – Revised April 2003.

[14] *Code Composer Studio Getting Started Guide*, Literature Number: SPRU509C, (November 2001).

[15] *TMS320C54x DSP Reference Set Volume 1: CPU and Peripherals*, Literature Number: SPRU131G, March 2001.

S. S. Prasad received the B.E., M.Sc. (Engg.) degrees in 1996 and 2001. He completed his Ph.D research work at Jadavpur University, Kolkata, India in June 2007. He is Lecturer in the Department of Electronics and Communication Engineering at National Institute of Technology, Jamshedpur, India since 1997. His research interests include direct digital synthesis, arbitrary waveform generation, multi-user detection etc. for digital communication.

S. K. Sanyal received the B.E.Te1.E., M.E.Te1.E. and Ph.D. degrees from Jadavpur University, India, in 1977, 1979 and 1990 respectively. He is currently a Professor in the Department of E.T.C.E, Jadavpur University, Kolkata-700 032, India. His research interests include tunable phased arrays, Analog/ Digital/ Genomic/ Radar Signal Processing, Digital Communication, VLSI circuit design. Mr. Sanyal is also the Chairman of the Calcutta Section of IEEE.