PRL – TN – 2007 – 87

PRL Technical Note

on

# CHIRP TRANSFORM SPECTROMETER DESIGN
# FOR
# SUBMILLIMETER WAVE SPECTROSCOPY

**By**

**Jayesh P. Pabari**

**December, 2006**

**Physical Research Laboratory (PRL)**

**Ahmedabad**

# Acknowledgement

I am very much thankful to all the members of Submillimeter group of PRL and MEDF and MTLF groups of Space Application Centre who have helped directly or indirectly in carrying out this project on the Chirp Transform Spectrometer.

Jayesh P. Pabari
'Engineer – SC' (Electronics Engineer)
Submillimeter Wave Group
Physical Research Laboratory,
Navrangpura, Ahmedabad – 380 009
INDIA

Phone: +91-79-26314254/26314251
Fax: +91-79-26314771
Email: jayesh@prl.res.in

# Contents

# Abstract

The spectral information of molecular transitions of an astronomical source can be retrieved by processing the IF signal from a mixer device through a spectrometer. The purpose of writing this note is to show the system design of Chirp Transform Spectrometer (CTS), the choice of appropriate sub-systems to meet targeted frequency resolution of 40 kHz with 200 MHz system bandwidth, the design of Surface Acoustic Wave (SAW) Reflective Array Compressor (RAC) chirp filter at 450 MHz centre frequency with 200 MHz (~ 50 %) bandwidth, its fabrication and testing. The note suggests the instrument design consuming low power of few Watts only, essential for space payloads. The proposed SAW chirp filter is the key element in the design of the spectrometer capable of providing targeted 40 KHz frequency resolution and 200 MHz bandwidth in conjunction with expander. Some of the results are presented in the note. The proposed CTS can be used for any similar kind of applications having CW nature in the source.

# INTRODUCTION

The high-resolution spectroscopy is of great importance for studying the wind velocity, temperature, pressure, and chemical composition of the astronomical targets emitting signal in sub-millimeter (SMM) range of frequencies. The spectral information of the molecular transitions of an astronomical source can be retrieved by processing IF signal from an SMM mixer such as Schottky diode, Superconductor-Insulator-Superconductor (SIS) junction or Hot Electron Bolometer (HEB) through a spectrometer such as Acousto-Optic Spectrometer (AOS), Auto Correlator Spectrometer (ACS) and Chirp Transform Spectrometer (CTS) [1]. The requirement of bandwidth and frequency resolution for a particular source dictates the kind of spectrometer that can be used. The CTS is choice of astronomers when very high frequency resolution is sought for the study of objects like comets etc. Ideally any resolution is possible with this technique. Other advantages of CTS include compactness in structure, less power consumption and reliability.

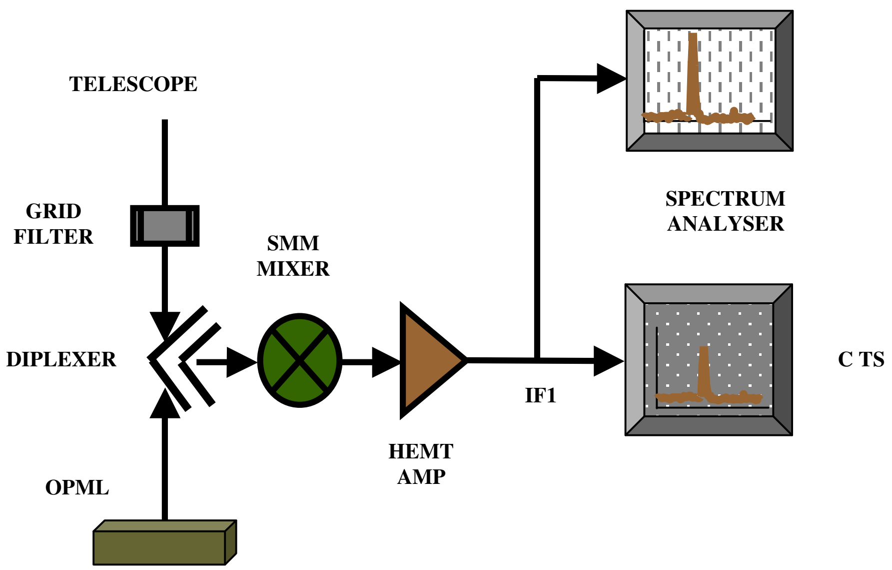



Figure 1: Submillimeter receiver block diagram

The SMM source signal and the local oscillator signal are combined using non-linearity of semiconductor device to give the Intermediate Frequency (IF) as a beat frequency of the two as shown in Figure 1. The local oscillator signal in the laboratory is derived from an Optically Pumped Molecular Laser (OPML) while the Schottky Barrier Diode is used as a mixer. The Schottky mixer gives the first IF in the range from 1 to 8 GHz, which contains the signal information. The output signal from the mixer is amplified by High Electron Mobility Transistor (HEMT) amplifier to get IF1 signal ready for further processing in the spectrometer. Design and development of a very high frequency resolution Chirp Transform Spectrometer (CTS) for Submillimeter (SMM) wave applications had been explored at PRL. The initial part gives the system design and digital expander for generating negative chirp signal while later part illustrates design of SAW RAC chirp filter carried out at PRL. It was being fabricated at Space Application Centre (SAC) as per design suggested.

## CONCEPT OF CHIRP TRANSFORMATION

The signal IF2 (obtained from IF1, as explained later) can be mathematically represented as f (t). The chirp transform of signal f (t) is derived from the Fourier transformation [2, 3]. For a given time dependent signal f (t), its Fourier transform is defined as

$$F(\omega) = \int_{-\infty}^{\infty} f(t)\, e^{-j\omega t}\, dt \quad \text{......................(1)}$$

Put $\omega = 2\pi\mu\tau$ in equation (1),

$$F(\omega) = \int_{-\infty}^{\infty} f(t)\, e^{-j(2\pi\mu\tau)t}\, dt$$

Substituting $2t\tau = t^2 + \tau^2 - (t-\tau)^2$, we get

$$F(2\pi\mu\tau) = e^{-j\pi\mu(\tau^2)} \int_{-\infty}^{\infty} [f(t)\, e^{-j\pi\mu(t^2)}]\, e^{j\pi\mu((\tau-t)^2)}\, dt \quad \text{.........................(2)}$$

Here, exponential signal in equation (2) above is known as chirp signal in RADAR terminology, $\mu$ is chirp slope (frequency by time), which transforms the frequency scale into time scale and $\tau$ is time delay. The three mathematical operations, i.e., multiplication, convolution and multiplication, in equation (2) are depicted in Figure 2 and such technique is used in RADAR compressive receivers. In first operation, the input signal f (t) is multiplied with negative chirp signal. The product of first operation is convolved with positive chirp signal. In the third multiplication operation, the result of second operation is multiplied with negative chirp signal to correct the phase. This Multiplication – Convolution – Multiplication (M – C – M) structure of Figure (2) is known as Chirp transform of the signal f (t).

Thus the Chirp transform maps the frequency axis in terms of time delay. Figure 3 shows how the horizontal input frequency time vectors are modified when they pass through chirp processor and become vertical finally to give pulses at different time delays. Thus the output spectrum from chirp processor would be pulses with different delay and amplitude according to frequency and amplitude of input spectral components respectively. The parameter $\mu$ is frequency separating parameter responsible for the speed of the sweep of a chirp signal while $\tau$ gives the proportional frequency position in the output spectrum.

In Figure 2 the $CH^-$ and $CH^+$ are negative and positive chirp signals involved in signal processing, M is the multiplier block and C indicates the convolution operation in the chirp filter. $CH^-$ block is known as an expander and $CH^+$ block is known as compressor.

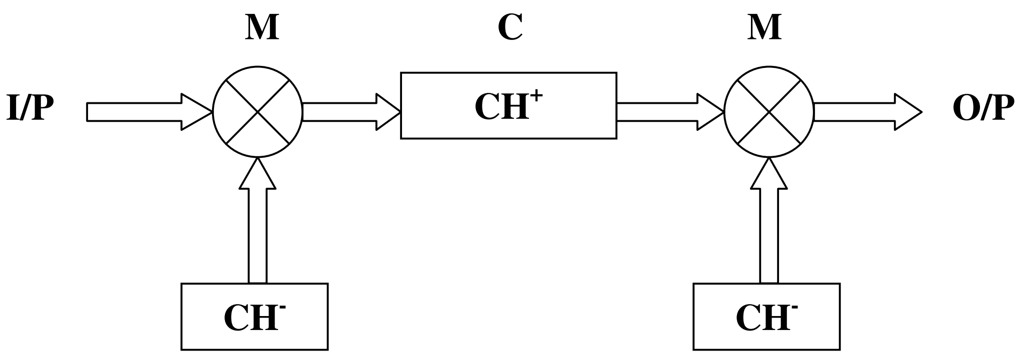



Figure 2: Chirp transform definition

## CTS DESIGN BLOCKS

The implementation of M-C-M structure can be done in two ways. (1) Firstly, in discrete time domain by digitising the signal known as Chirp Z Transform (CZT) spectrometer [4]. This method utilizes CZT software and it is useful for offline signal processing. (2) Secondly, in analog time domain by using expander and compressor in hardware form known as CTS.

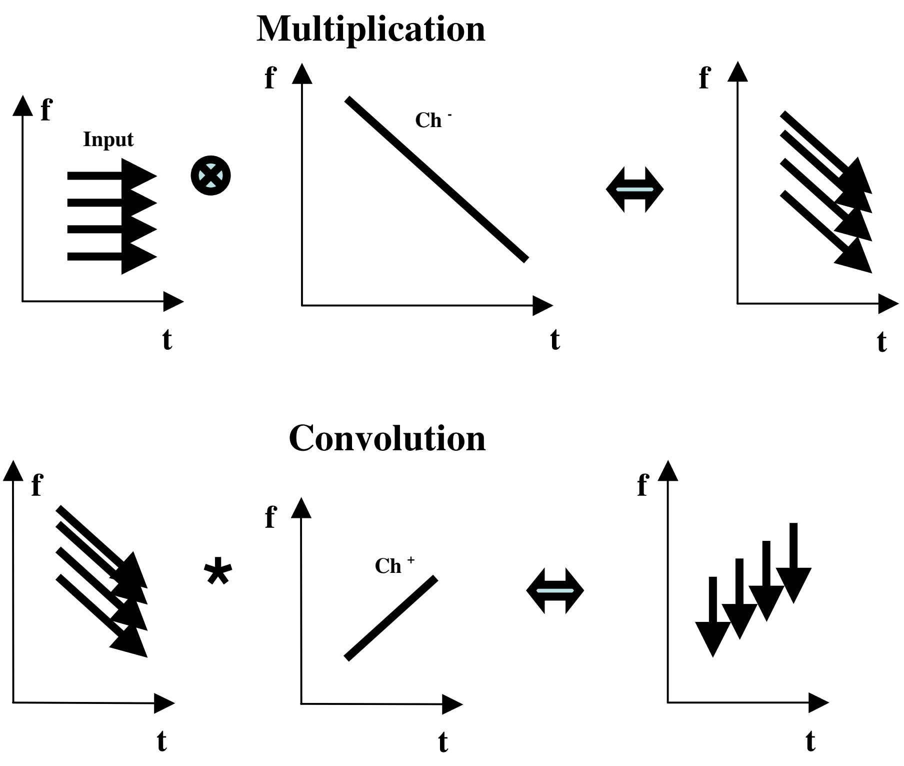



Figure 3: Vector modification during Chirp transformation

It is useful for real time operation of spectrometer. I have designed this real time CTS spectrometer using hardware approach, which is given here in main part of the note whose target specifications are given in Table 1.

| Specification | Value |
|---|---|
| Frequency Resolution | 40 kHz |
| Bandwidth | 200 MHz |
| Input Frequency | 1250 MHz |
| Dynamic Range | 40 dB |

Table 1: Target specifications of CTS

As shown in Figure 4, in the suggested design the whole CTS [5, 6] is divided into three stages, i.e., analog processor, chirp processor and digital processor out of which first two stages are specified. The analog processor is basically a second down conversion stage; the chirp processor is the heart of the instrument responsible for whole system's specifications. The third stage, i.e., digital processor is avoided here for the case of testing on ground and would be replaced by high-speed oscilloscopes.

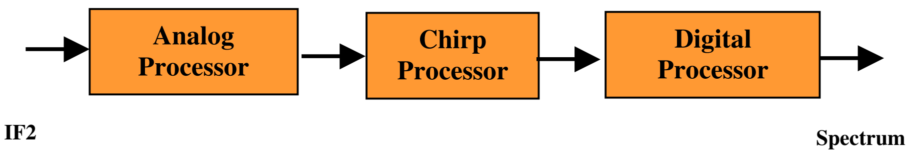



Figure 4: CTS design diagram

The IF1 signal, which is to be delivered in the range of 1 to 8 GHz, is further down-converted by using analog processor that includes the Low Noise Amplifier (LNA), microwave (MW) mixer, local oscillator and band pass filter. The output of analog processor is known as second IF (IF2) and it has centre frequency of 1250 MHz with and bandwidth of 200 MHz. This IF2 signal is amplified to achieve the signal level suitable for use in chirp processor.

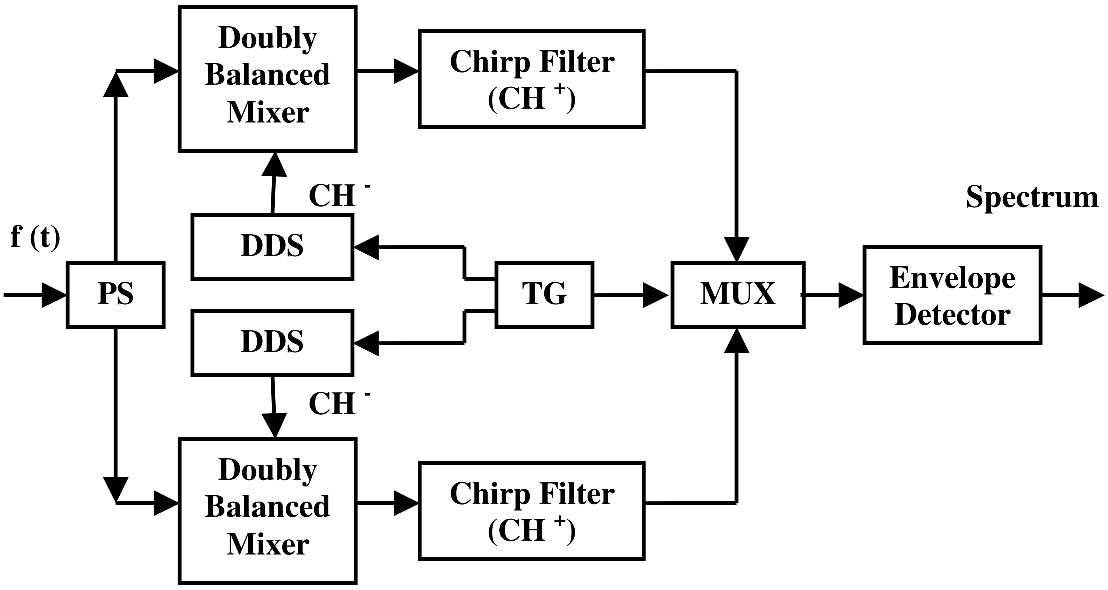
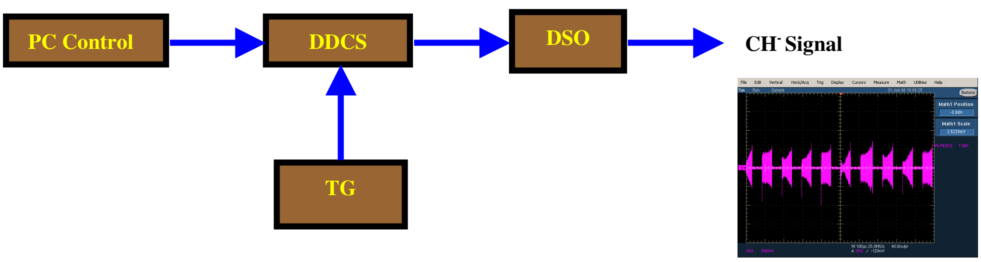



Figure 5: CTS design diagram

To implement M – C – M structure of chirp processor, the second IF at 1250 MHz coming from analog processor is split into two channels as shown in Figure 5 [7] and each channel is operated at 50 % duty cycle. One of these split signal is to be multiplied with negative chirp signal of expander. The expander is implemented using digital technique, i.e., negative chirp signal is generated by using Direct Digital Synthesizer (DDS) operating at clock frequency of 1 GHz. The DDS has been interfaced with PC and programmed to generate burst of negative chirp signal. The different command words written for the synthesizer can generate different types of chirp signal using digital method. The chirp slope can be controlled using 32 bit 2's complement method and therefore sign of the slope can be selected. Figure 6a shows negative chirp signal obtained from DDS in burst form to be used for multiplication process and Figure 6b shows how the output of DDS varies with respect to frequency.




(a)

(b)

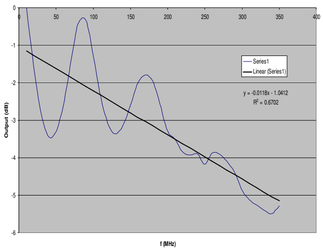



Figure 6: (a) Testing set up and output of synthesiser in burst form and (b) Output of synthesiser with respect to frequency

## DDS FIRMWARE

The firmware of DDS AD9858 for chirp generation designed by me is given in the following code. The programme sequence is the initialisation of AD9858 for chirp operation, loading of frequency change occurring in the process, its duration and the phase of interest. The output of AD9858 is obtained using the code written.

```
#include<iostream.h>
#include<dos.h>
#include<conio.h>
#define DATA 0X0378
#define CON 0x037A
void main()
{
clrscr();
int i,cfr;
char c,r;
cout<<"           PROGRAMME TO RUN AD9858 FOR CHIRP GENERATION\n";
cout<<"\nCheck power supply connections and switch on supply.";
cout<<"\nClock should be 2 GHz with -5 dBm level.";
cout<<"\nDo you want to reset ? If yes, press 'y'; if no, press 'n':
";
cin>>r;
switch(r){
case'n':{
      cout<<"\nYou have kept the synthesizer in previous condition.";
            break;
            }
case'y':{
      cout<<"\nYou have reset the synthesizer.\n";
      outportb(DATA,0x6D);
      outportb(CON,0x07);
      outportb(CON,0x0F);
      break;
      }
default:{;}
}
cout<<"\nDo you want to generate chirp ?";
cout<<"\nIf yes, press 'y' otherwise press 'n': ";
```

```
cin>>c;
switch(c){
case'n':{
      cout<<"\nYou are not interested!! We will meet later on.\n";
      break;
      }
case'y':{
      cout<<"\nNegative chirp is generated.";
      cout<<"\nStart frequency = 400 MHz";
      cout<<"Stop frequency = 0 Hz";
      cout<<"\nChirp rate = 500 Hz/microsecond";
      cout<<"     Dispersion time = 800 millisecond\n";
      outportb(DATA,0x00);
      outportb(CON,0x09);
      outportb(CON,0x0B);
      outportb(DATA,0x18);
      outportb(CON,0x0E);
      outportb(CON,0x0F);
      outportb(DATA,0x44);
      outportb(CON,0x07);
      outportb(CON,0x0F);
      outportb(DATA,0x01);
      outportb(CON,0x09);
      outportb(CON,0x0B);
      outportb(DATA,0x80);
      outportb(CON,0x0E);
      outportb(CON,0x0F);
      outportb(DATA,0x44);
      outportb(CON,0x07);
      outportb(CON,0x0F);
      outportb(DATA,0x02);
      outportb(CON,0x09);
      outportb(CON,0x0B);
      outportb(DATA,0x00);
      outportb(CON,0x0E);
      outportb(CON,0x0F);
      outportb(DATA,0x44);
      outportb(CON,0x07);
      outportb(CON,0x0F);
      outportb(DATA,0x03);
      outportb(CON,0x09);
      outportb(CON,0x0B);
      outportb(DATA,0x00);
      outportb(CON,0x0E);
      outportb(CON,0x0F);
      outportb(DATA,0x44);
      outportb(CON,0x07);
      outportb(CON,0x0F);
      outportb(DATA,0x04);
      outportb(CON,0x09);
      outportb(CON,0x0B);
      outportb(DATA,0xDC);
      outportb(CON,0x0E);
      outportb(CON,0x0F);
      outportb(DATA,0x44);
      outportb(CON,0x07);
      outportb(CON,0x0F);
      outportb(DATA,0x05);
      outportb(CON,0x09);
      outportb(CON,0x0B);
      outportb(DATA,0x46);
```

```
outportb(CON,0x0E);
outportb(CON,0x0F);
outportb(DATA,0x44);
outportb(CON,0x07);
outportb(CON,0x0F);
outportb(DATA,0x06);
outportb(CON,0x09);
outportb(CON,0x0B);
outportb(DATA,0x03);
outportb(CON,0x0E);
outportb(CON,0x0F);
outportb(DATA,0x44);
outportb(CON,0x07);
outportb(CON,0x0F);
outportb(DATA,0x07);
outportb(CON,0x09);
outportb(CON,0x0B);
outportb(DATA,0x00);
outportb(CON,0x0E);
outportb(CON,0x0F);
outportb(DATA,0x44);
outportb(CON,0x07);
outportb(CON,0x0F);
outportb(DATA,0x08);
outportb(CON,0x09);
outportb(CON,0x0B);
outportb(DATA,0xD4);
outportb(CON,0x0E);
outportb(CON,0x0F);
outportb(DATA,0x44);
outportb(CON,0x07);
outportb(CON,0x0F);
outportb(DATA,0x09);
outportb(CON,0x09);
outportb(CON,0x0B);
outportb(DATA,0x30);
outportb(CON,0x0E);
outportb(CON,0x0F);
outportb(DATA,0x44);
outportb(CON,0x07);
outportb(CON,0x0F);
outportb(DATA,0x0A);
outportb(CON,0x09);
outportb(CON,0x0B);
outportb(DATA,0x66);
outportb(CON,0x0E);
outportb(CON,0x0F);
outportb(DATA,0x44);
outportb(CON,0x07);
outportb(CON,0x0F);
outportb(DATA,0x0B);
outportb(CON,0x09);
outportb(CON,0x0B);
outportb(DATA,0x66);
outportb(CON,0x0E);
outportb(CON,0x0F);
outportb(DATA,0x44);
outportb(CON,0x07);
outportb(CON,0x0F);
outportb(DATA,0x0C);
outportb(CON,0x09);
```

```
	outportb(CON,0x0B);
	outportb(DATA,0x66);
	outportb(CON,0x0E);
	outportb(CON,0x0F);
	outportb(DATA,0x44);
	outportb(CON,0x07);
	outportb(CON,0x0F);
	outportb(DATA,0x0D);
	outportb(CON,0x09);
	outportb(CON,0x0B);
	outportb(DATA,0x66);
	outportb(CON,0x0E);
	outportb(CON,0x0F);
	outportb(DATA,0x44);
	outportb(CON,0x07);
	outportb(CON,0x0F);
	outportb(DATA,0x0E);
	outportb(CON,0x09);
	outportb(CON,0x0B);
	outportb(DATA,0x00);
	outportb(CON,0x0E);
	outportb(CON,0x0F);
	outportb(DATA,0x44);
	outportb(CON,0x07);
	outportb(CON,0x0F);
	outportb(DATA,0x0F);
	outportb(CON,0x09);
	outportb(CON,0x0B);
	outportb(DATA,0x00);
	outportb(CON,0x0E);
	outportb(CON,0x0F);
	outportb(DATA,0x44);
	outportb(CON,0x07);
	outportb(CON,0x0F);
	outportb(CON,0x09);
	outportb(DATA,0x74);
	outportb(CON,0x07);
	outportb(CON,0x0F);
	cout<<"\nDo you want to run synthesizer for CTS ?";
	cout<<"\nIf yes, press 'y'. If no, press 'n': ";
	cin>>r;
	switch(r){
	case'n':{
		cout<<"\nYou have kept the synthesizer running ";
		cout<<"for wide band chirping!\n";
		break;
		}
	case'y':{
	cout<<"\nPress any key if you want to stop synthesizer
generating ";
	cout<<"chirp for CTS.\n";
	while(!kbhit())
	{
	outportb(DATA,0x02);
	outportb(CON,0x09);
	outportb(CON,0x0B);
	outportb(DATA,0x18);
	outportb(CON,0x0E);
	outportb(CON,0x0F);
	outportb(DATA,0x44);
	outportb(CON,0x07);
```

```
        outportb(CON,0x0F);
        outportb(CON,0x09);
        outportb(DATA,0x74);
        outportb(CON,0x07);
        outportb(CON,0x0F);
        delay(10);
        outportb(DATA,0x02);
        outportb(CON,0x09);
        outportb(CON,0x0B);
        outportb(DATA,0x00);
        outportb(CON,0x0E);
        outportb(CON,0x0F);
        outportb(DATA,0x44);
        outportb(CON,0x07);
        outportb(CON,0x0F);
        outportb(CON,0x09);
        outportb(DATA,0x74);
        outportb(CON,0x07);
        outportb(CON,0x0F);
        delay(10);
        }
        getch();
        cout<<"\nSynthesizer will run for wide band chirp
application.\n";
        break;
        }
        default:{;}
        }
break;
}
default:{;}
}
cout<<"\nPress any key to go to programme.\n";
getch();
}
```

Here the output of DDS, i.e., negative chirp signal has 400 MHz bandwidth and 50 microsecond duration and it does not remain constant with respect to frequency due to limitation of the device. The negative chirp signal from DDS and second (split) IF signal are applied to balanced mixer acting here as a multiplier block proving first operation of multiplication necessary for Chirp transformation. One of the solutions to the limitation of the device is to use mixer acting as a multiplier having wider range of LO power to tolerate such variations in amplitude.

## TIMING GENERATOR

The timing generator (TG) block of Figure 5 and 6 (a) is selected as 1 GHz clock from RFXO family of clocks to drive various circuits including DDS as shown in Figure 7. Its PCB has been made and it has been attached with DDS to minimize the space.

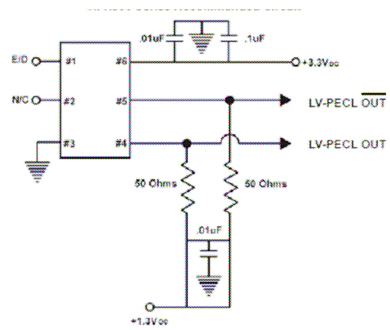



Figure 7: Implemented circuit of 1 GHz clock on PCB

## SAW RAC DESIGN

The product after first multiplication is passed through SAW RAC [8] chirp filter. This chirp filter decides the system bandwidth and frequency resolution. The impulse response of SAW-RAC is positive chirp signal as shown in Figure 8. Its slope is exactly opposite to that of negative chirp signal from DDS. The duration of this positive chirp signal is 25 microsecond and its bandwidth is 200 MHz. The second operation of convolution is carried out in the chirp filter. The output of chirp filter would be pulses with different delay and amplitude according to input spectral components. The outputs of the two chirp filters placed in two channels are combined into multiplexer to obtain 100 % duty cycle of the spectrometer.

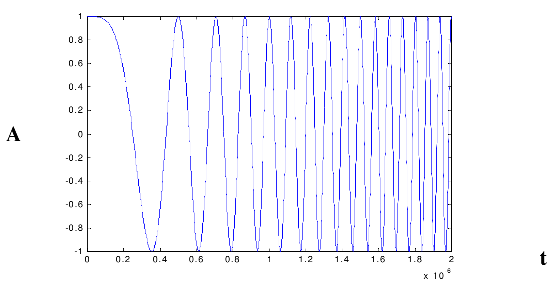



Figure 8: Simulated impulse response of designed SAW RAC chirp filter

Due to availability of RAC type chirp filter, very high time-bandwidth product of 5000 can be realized. I have designed this SAW RAC chirp filter in which the design centre frequency is 450 MHz, target bandwidth is 200 MHz (actually taken as

224 MHz to avoid side effects in chirp filter) and target duration of chirp signal is 25 microsecond (actually taken as 28 microsecond to avoid side effects in chirp filter). The designed SAW RAC chirp filter is shown in Figure 9. The insertion loss is expected to be about 35 dB. The design involves two arrays of 5000 grooves along with other items and the total computation involves complexity of roughly square of the number of grooves, which has been carried out using my own design code written in C language. The software gives final response curve of chirp filter for a given set of input design parameters like depth of grooves in arrays, width of grooves, SAW launching aperture etc. I have also written separate design code to generate lithography data to be fed directly to lithography machine at SAC. The normal way of etching the grooves is making the length of all grooves the same, which gives the frequency response of the filter as inclined (higher value) towards higher frequency side as shown in Figure 10a. In order to achieve the appropriate frequency response, I have suggested tapered length of grooves as shown in Figure 9 [9]. This helps in avoiding the higher magnitudes of filter response at higher frequencies and further calibration needed due to filter response. Similar to the groove length variation, it is also possible to vary the depth and width of the grooves to favourably change the response. This has been verified using design software.

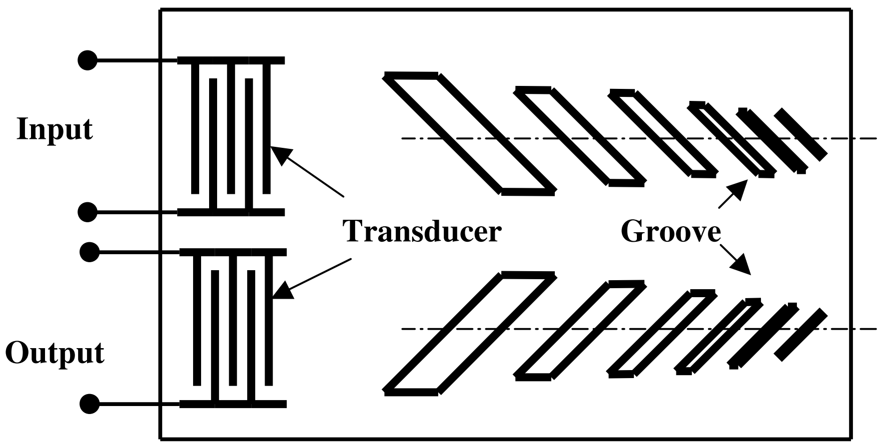



Figure 9: Designed SAW RAC chirp filter

Weighting in magnitude response can be obtained by depth variation; however there need not be any change in its phase response. Also, there is a very minor difference in magnitude responses for the two cases of groove width variation, (i) first when the groove width is kept constant corresponding to centre wavelength of the chirp filter as $\lambda_o/2$ and (ii) second is when the width of the grooves is varied as per running wavelength $\lambda/2$ throughout the length of the array.

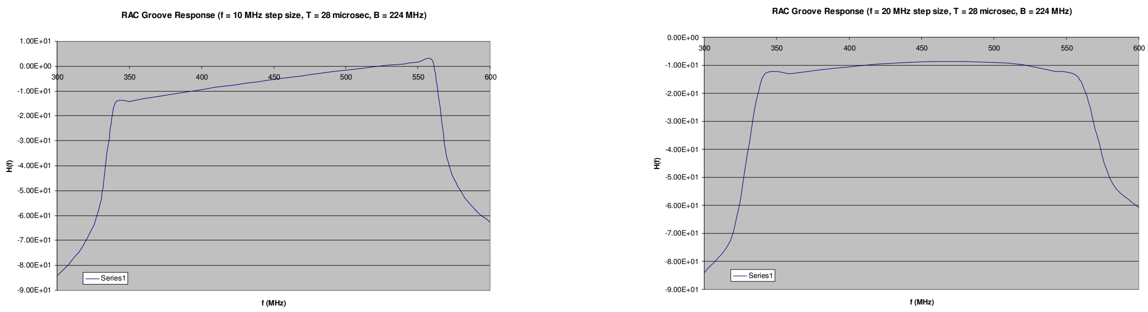



Figure 10: (a) Simulated frequency response of SAW RAC with same length of grooves and (b) Simulated frequency response of SAW RAC with tapered length of grooves

Figure 10b shows this improvement in frequency response because of tapering as compared to that shown in Figure 10a. It is possible to generate any type of response by this technique.

## FABRICATION OF SAW RAC CHIRP FILTER

Fabrication of this chirp filter could be done using lithography on appropriate substrate at Space Application Centre (SAC). Lithium Niobate ($LiNbO_3$) substrate was selected for achieving 200 MHz bandwidth. In the design of RAC chirp filter, there are two etched arrays of inclined grooves and two Inter Digital Transducers (IDT) of metal. The mask design and resist image of grooves are shown in Figure 11.

In M – C – M structure of chirp processor, I omit the last multiplication stage since only the amplitude spectrum is of concern and the phase information is not required for the astronomical objects. The amplitude of pulses coming out from multiplexer is detected using envelope detector.

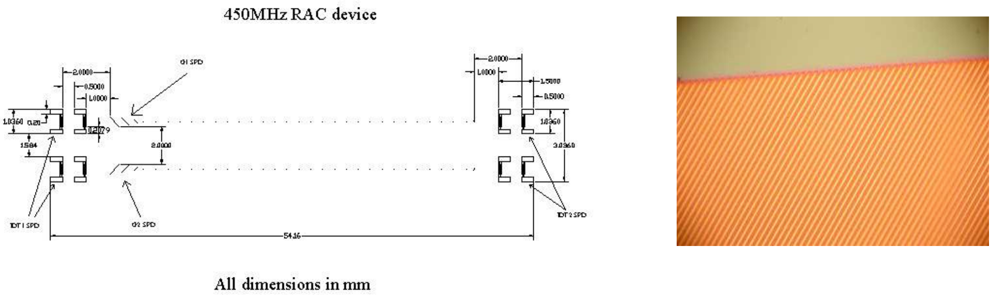



Figure 11: (a) Mask design of SAW RAC chirp filter (b) Resist image of grooves etched on SAW RAC chirp filter fabricated as SAC with step size of 10 nm

In our design the duration of expander chirp (50 microsecond) is larger than that of compressor chirp (25 microsecond) which is known as M (l) – C (s). It gives the 'Sliding Fourier Transform' of the given input signal. If the signal comprises a finite sum of CW waveforms, the magnitude of the output given by the sliding transform is the same as that given by the conventional transform; the distinction is then of no significance, provided the spectral phase is not required.

## RESULTS

Figure 12 shows simulation of Multiplication and Convolution operations and indicates how the input spectral components can be separated using Chirp transform for a given input signal having three frequencies. Figure 13 shows a typical output of chirp processor for a given input with single (low) frequency. The operation of chirp processor matches in both cases.

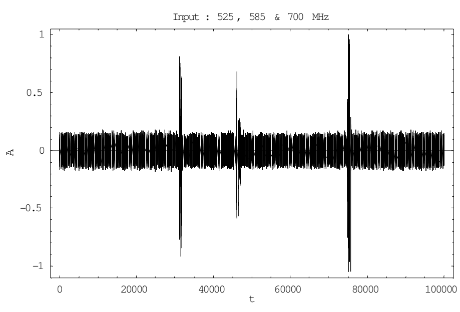



Figure 12: Simulation of M-C structure

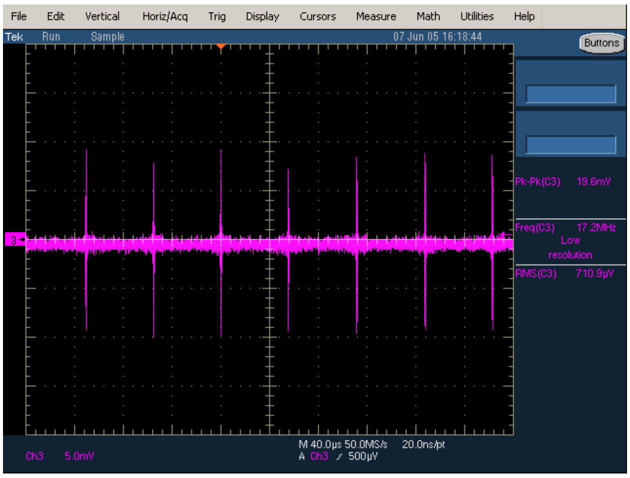



Figure 13: Typical output of CTS

## SUMMARY AND FUTURE SUGGESTIONS

The design of CTS had been considered for a very high frequency resolution. The firmware and design software were developed for controlling DDS using external PC and for the design of SAW RAC filter respectively. Tapering of grooves in the design of SAW RAC chirp filter has been suggested to achieve required flat frequency response. Moreover, lithography data generating software was also made for fabrication of the SAW RAC chirp filter.

The convolution operation with different signs can be tested for optimum performance. The SAW RAC chirp filter can replace digital expander to reduce power consumption for future space and other applications where power supply is limited. The CTS is best suited for ground-based observatory, airborne observatory, balloon payload, and space payloads for obtaining very high frequency resolution. For space applications, a digital processor with high-speed analog to digital converter is needed to send the data through telemetry to ground.

# REFERENCES


[1] GRATE – German Receiver for Astronomy at Terahertz Frequencies, website "http://www.ph1.uni-koeln.de/workgroups/astro_instrumentation/receiver/great/great.html"

[2] D. P. Morgan, "Chirp filters and their applications" in "Surface-wave devices for signal processing", Elsevier Publication, 1991, pp. 268-275

[3] M. A. Jack, P. M. Grant and J. H. Collins, "The theory, design and applications of surface acoustic wave Fourier transform processors" Proceedings of the IEEE, Vol. 68, no. 4, April 1980, pp. 450-468.

[4] Jayesh Pabari and Hemant Dave, "Chirp transform spectrometer for heterodyne receiver system", Proceedings on Sensor Technology, Sept. 2002, pp. 543-546

[5] X. Zheng, Y. Zhou, Z. Xu, J. Gong, S. Li and J. E. Ken, "A new radio spectral receiving system at l=13 mm in Urumqi astronomical station", Astronomy and Astrophysics Suppl. Scr. 137, 1999, pp. 537-541

[6] P. Hartogh and G. K. Hartmann, "A high-resolution chirp transformed spectrometer for microwave measurements", Meas. Sci. Technology 1, 1990, pp. 592-595

[7] Jayesh Pabari, Hemant Dave, R. P. Singh, Ashish Dubey, Satheesh Thampi, Vinay Kumar, N. M. Vadher, and A. B. Shah, "High resolution chirp transform spectrometer for terahertz spectroscopy", Proceedings of 1st International Conference on Microwaves, Antenna, Propagation and Remote Sensing, Jodhpur, December, 2003, pp. 3.14-3.17

[8] Richard C. Williamson and Henry I. Smith, "The use of surface-elastic wave reflection gratings in large time-bandwidth pulse-compression filter", IEEE Transactions on MTT, Vol 1, No. 4, April, 1973, pp. 195-205

[9] Jayesh Pabari et. al., "Chirp transform spectrometer development for submillimeter wave spectroscopy", Proceedings of International Conference on Submillimeter Science and Technology, Ahmedabad, October 2004, pp. 82-85