Design of IRNSS Tracking System using 1.5 bit ADPLL and Correlator

M. Udaya, D. Sony, D. Krishna Reddy

Abstract*: *IRNSS is an indigenous satellite navigation system consisting of 7 satellites that provide accurate positioning in the Indian sub-continent region. Each IRNSS satellite transmits a signal which contains information regarding satellite orbital and clock parameters (known as navigation message). The purpose of the receiver is to demodulate the satellite signal and extract navigation message, the receiver must know certain parameters of the signal like its doppler shift and code offset. However, in real-time, due to relative velocity of the satellite and ionospheric interference, these parameters vary with time. Therefore, the receiver must continuously perform the tracking operation to update the varying parameters. Existing tracking systems are based on SDR and SoC's, which require high-performance processors and iterative algorithms to perform both carrier and phase tracking. Though they are highly accurate, these designs are complex and expensive. In this paper, 1.5-bit ADPLL is used to track the carrier. This design does not require numerous computational loops to perform tracking of the carrier, thus reducing the complexity of the design. This work includes simulation results for 1.5-bit ADPLL. In this work, 2-bit, 1.5-bit, and modified 1.5-bit correlators are simulated and synthesized. It was found that modified 1.5-bit correlator architecture is less complex compared to 2-bit correlator and offers better SNR compared to 1.5-bit correlator. Therefore, modified 1.5-bit correlator is used for code tracking. The IRNSS signal tracking is performed in ModelSim. The system utilizes 77 standard LUTs and exhibit maximum settling time of 714μs and 31.28ms for carrier tracking and code tracking, respectively.

Keywords*: *Receiver, ADPLL, correlator, satellite navigation.

I. INTRODUCTION

The satellite navigation systems have become an essential infrastructure of major prominence in national security, information security, and economy. The research and development of satellite navigation receiver have gained much significance in the past decade. Indian Regional Navigation Satellite System (IRNSS) is an indigenous satellite navigation system developed and maintained by Indian space research organization (ISRO). This satellite system provides continuous, accurate, and reliable location, speed, and time information. The IRNSS offers two kinds of services Standard Positioning Service (SPS) and Restricted/Authorized Service (RS) at L5 band (1176.45MHz) and S-band (2472.5 MHz) frequencies, respectively. Each IRNSS satellite transmits a highly phase-coherent, frequency-stabilized continuous wave signal. It is encoded with binary data containing satellite orbital information and on-board clock parameters (known as ephemeris). The purpose of the receiver is to detect and demodulate the signal transmitted by IRNSS satellite. For the receiver to demodulate the navigation signal and extract navigation message, the receiver must know certain parameters of the signal. These parameters include the carrier frequency, code chip delay, and satellite visibility. These parameters are determined at the start-up of the receiver in a process called acquisition. However, in real-time, due to the relative velocity of satellite and ionospheric interference, these parameters vary with time. Therefore, the receiver continuously performs tracking operation to update the varying. The work by E. Schmidt presented in "Development of a Real-Time Software-Defined GPS Receiver in a LabVIEW-Based Instrumentation Environment" [5] describes GPS receivers based on software defined radios (SDR). Although SDR is most used method in navigation trackers due to their flexibility and short development time, based receivers usually use higher order bit quantization like 32 or 64 bits. As GPS algorithms must run in real time, faster processors are required. These incorporate highly parallelized architecture which makes the design power hungry, bulky, and expensive. The other implementations of GPS tracking are based on SoC platform. One such implementation is presented by E. Wang in "Implementation of an Embedded GPS Receiver Based on FPGA and MicroBlaze" [6]. This paper describes implementation of GPS tracker on a FPGA using Microblaze soft processor which uses 1167 LUTs. Although soft processor-based systems offer high flexibility at chip level as compared to SDR, the soft processors are very complex and expensive.

II. SIGNAL STRUCTURE

A. Transmitted Signal

Each satellite continuously transmits ephemeris (binary data containing its position and time). At the transmitter end, this data is multiplexed by CDMA scheme and then BPSK modulated onto L5 and S-band carrier signals [4].

Revised Manuscript Received on August 30, 2020.

* Correspondence Author

M. Udaya*, Department of Electronics and Communication, CBIT, Hyderabad, India. E-mail: jasperudaya@gmail.com

D. Sony, Department of Electronics and Communication, CBIT, Hyderabad, India. E-mail: sonyd_ece@cbit.ac.in

Dr. D. Krishna Reddy, Department of Electronics and Communication, CBIT, Hyderabad, India. E-mail: dkrishnareddy_ece@cbit.ac.in

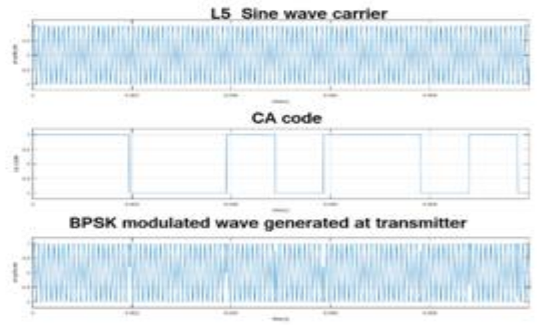


Fig. 1.Transmitted signal at IF = 10MHz

CDMA modulation is achieved by multiplying the ephemeris by the satellite's CA code. CA codes are a set of mutually orthogonal codes. The term orthogonal means that each CA code when correlated by another CA code or by a shifted version of itself, shifted by more than 1 bit will result in a negligible value, whereas CA code correlated with itself will result in a correlation peak. Since each satellite has a unique CA code, receiver can determine the identity of transmitting satellite simply by correlating received signal with all possible shifted versions of all the codes in the set of CA codes and by determining correlation peak. CDMA code thus obtained is used to modulate carrier wave through BPSK modulation scheme. In BPSK, each time the bit toggle occurs in modulating data, the phase of the sinusoidal carrier is shifted by 180 degrees. The ephemeris data, CA code data and L5 carrier have frequencies of 50 Hz, 1023 KHz, 1176.54 MHz, respectively. Fig 1 shows CDMA code modulating carrier. The signal transmitted by satellite can be mathematically represented as

$$s(t) = \sqrt{2P}\, C(t)D(t) \cos(\omega_e t) \quad \ldots (1)$$

where C(t) is CA code, D(t) is Ephemeris data, we are the L5 carrier frequency and P is the average power of the signal.

B. Received Signal

At the receiver end, a distorted version of the signal is received. The main factors causing distortion are ionospheric attenuation, Doppler shift in frequency and noise inherent in receiver equipment. The attenuation significantly reduces the signal strength to about 20 dB below noise floor. The doppler shift in the frequency occurs due to relative velocity between satellite and receiver. It is reflected as a shift in carrier frequency, which is proportional to relative velocity between satellite and receiver. Its typical range is ±10KHz. Equation (2) gives the mathematical model for signal at receiver end.

$$s^{'}(t) = \sqrt{2P^{'}}\, C(t - t_0)D(t - t_0)\cos[(\omega_e + \omega_d)\,t + \theta] + \phi(t) \quad \ldots\ldots (2)$$

where C(t) is CA code, D(t) is Ephemeris data, t_o is code offset, we is the L5 carrier frequency, ω_d is the frequency offset in carrier due to doppler effect, θ is phase offset in carrier, P' is average power of the received signal and ϕ(t) is Gaussian noise due to ionospheric interference.

C. Demodulation Scheme

The purpose of a receiver is to demodulate the received signal and retrieve D(t). In order to demodulate, s'(t) is firstly multiplied by with C(t-t_o). Since C(t) takes values of ±1, when C(t) is multiplied by exact replica of itself, two -1's multiply to produce +1 and two +1's multiply to give +1, thus C(t-t_o) component is removed from signal. In order to generate exact replica, receiver must determine time offset t_o. Since C(t) is 1024 bit long binary code that repeats itself, t_o is determined by multiplying s'(t) with each of 1023 possible shifts of C(t) and determining exact shift that gives the peak. This search is referred to as code search. Finally, s'(t) is multiplied by the exact replica of the carrier, cos[(w_e+w_d)t+θ], in order to remove carrier component.

$$\cos[(we + wd)t + \theta] * \cos[(we + wd)t + \theta'] = \cos[2(we + wd)t + (\theta + \theta')] + cos\,[(\theta - \theta')] \quad (3)$$

where θ' is the estimated phase offset. The high frequency component is removed by LPF filter and cos (θ-θ') appears as a DC component in final signal. However, receiver must know the frequency offset ω_d in order to remove carrier component. ω_d is determined by searching the possible frequency offset range of ±10KHz in steps of 50Hz bins till exact ω_d is determined. This process is called frequency search. Frequency search and code search together are referred to as "acquisition". The output of acquisition gives code(t_o), frequency(ω_d) and phase(θ) offset. However, these parameters change continuously due to motion of satellite, receiver and changes in ionospheric conditions. Therefore, parameters require to be updated continuously. A tracking system is essential to continuously track changes in code and frequency parameters and update them in demodulator.

III. TRACKING SYSTEM

The tracking system tracks and refines the rough estimates of code, frequency and phase parameters which are determined in the acquisition process. The basic block diagram of tracking system is shown in Fig 2 [8]. Initially, the incoming signal is multiplied by replica of carrier. This process will remove carrier wave from the signal. Finally, the signal is multiplied with locally generated CA code and ephemeris is retrieved. Therefore, the tracking system must generate replicas of both carrier and code [7].

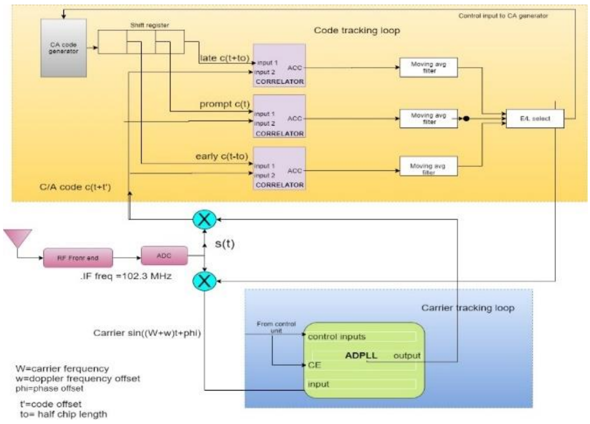


Fig. 2.Block diagram of tracking system

A. Code Tracking Loop

The purpose of the code loop is to determine whether the code offset of incoming signal is advanced or delayed with respect to locally generated code and adjust it accordingly. This is done by producing three replicas of locally generated code, early(e), late(l) and prompt(p) codes by using a shift register. The early(e), late(l) and prompt(p) codes are separated by ½ chip length. The incoming signal is multiplied with e, l, and p code in 3 different tracks and the product is accumulated. E/L select block compares the 3 accumulation (ACC) and determined whether code offset is advanced or delayed.

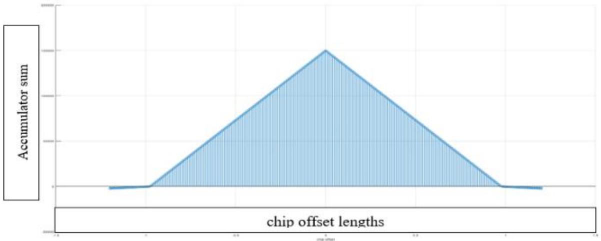


Fig. 3.Variation in accumulation peak with respect to chip offset lengths.

Code tracking loop shown in Fig 2 consists of the E/L select module, correlators and filters. The input to code tracking loop is free from carrier component as carrier tracking loop removes the carrier component. Within code carrier loop, the input wave is correlated by early, prompt, and late versions of CA. The correlator sum is directly proportional to the extent of alignment between the two inputs of correlator. The E/L module compares the outputs of all correlators and determines whether the local code is early or late compared to the incoming code. E/L select module has a decision block which generates an input to local CA generator to delay or advance its output. The number of signal samples that must be correlated in order to enable early or late select module to work properly depends on amount of SNR. If SNR is less, more samples must be correlated to make a decision. The major source of noise at the input of code tracking loop is the quantization error.

B. Carrier Tracking Loop

The purpose of the carrier loop is to lock onto frequency and phase of the incoming signal and generate an exact replica locally. This is achieved by using a phase locked loop (PLL). PLL is a feedback system which generates an output signal whose phase is related to that of input signal.

When the incoming carrier and locally generated carrier are compared using phase detector of PLL, a phase error is generated and passed through low pass filter. The phase error is given to the input of the oscillator. When lock state is achieved, the frequency of the oscillator settles to that of the input signal. The output of oscillator is multiplied to input signal and passed through low pass filter to filter high frequency component, from (3). The carrier is thus removed from input signal.

IV. ALL DIGITAL PHASE LOCKED LOOP

A phase locked loop is a feedback control system whose purpose is to generate an output signal whose phase and frequency are equal to that of the input signal. Analog PLLs have limited lock range, less design flexibility and they cannot be implemented on FPGA. In this design, since the design is intended for digital implementation, we use an ADPLL. The input of ADPLL is sine wave. This is passed through a phase detector which compares the phase of input wave with that of reconstructed signal from NCO and generates error signal [9].

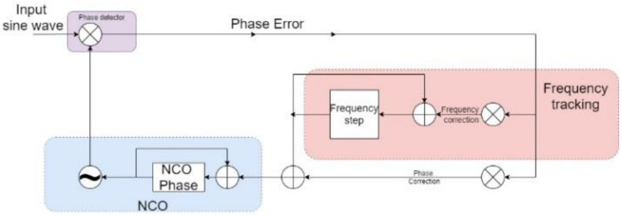


Fig. 4.Block diagram of ADPLL

The error signal acts as input to two other sub-modules: (1) frequency tracking (2) phase tracking. The two sub modules control NCO to generate reconstructed sine wave, which is the feedback signal into phase detector.

A. I/O Signals of ADPLL

The various IO signals required for an ADPLL are as following: The incoming signal is represented as sine wave. Frequency of NCO is set only when Load new frequency is set. Frequency control value gives the phase increment of NCO. Loop control bandwidth decides settling period of PLL. Since the PLL is synchronous, clock signal is used. CE is global clock enable. The fig 5 shows various IO signals of PLL.

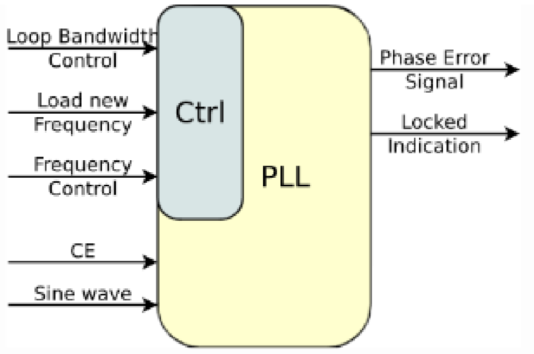


Fig. 5.IO signals of ADPLL

Error signal, Phase Error Signal is the output of phase detector. Locked Indication signal is high when lock state is achieved.

B. Design of Logical NCO

Numerically controlled oscillator is a digital oscillator that can generate an arbitrary frequency based on its phase step input. It basically operates through a look up table which stores one period of the required signal, such as sine or cos. The index to LUT is controlled by NCO accumulator. Accumulator is updated at each clock edge by a value specified by frequency control input. Higher the frequency control input, faster the index scales the LUT and higher the NCO frequency. The frequency of NCO is given by (4).

$$f = \frac{frequency\ control\ input}{2^{(PHASE\_STEP)}}\ f\_clk \qquad \text{…. (4)}$$

here PHASE_STEP is the NCO accumulator size.

C. Phase and Frequency Tracking Operation of PLL

While phase tracking, PLL accepts phase error and corrects phase accumulator. The phase error is multiplied by a constant specified as $2^{-\text{loop control bandwidth}}$.

If there is no phase error, NCO is continuously stepped by frequency control value. On contrary, if phase error does exist, along with frequency control value, a number corresponding to phase correction is added or subtracted from accumulator. The frequency of NCO is determined by input phase step, frequency control value. Therefore, the PLL tracks the phase error and adjusts frequency control value by phase error value scaled by frequency correction factor at each clock edge. However, the frequency correction factor must be chosen to assure stability and thus avoid oscillations in output.

The required lock range for PLL, for reliable operation is 9.995 MHz to 10.005 MHz and free running frequency must be 10MHz. Equation (5) shows relation between phase steps of NCO and NCO frequency

$$NCO\ phase\ step = \frac{2^N * f_o}{f_s} \qquad \text{………… (5)}$$

Where N is NCO accumulator size, f_o is output frequency and fs is sampling or clock frequency. Since, PLL is operates on 32-bit counter, 10.005MHz, 10MHz, 9.995MHz correspond to phase steps of 286474318.64, 286330153.066, 286187987.49, respectively.

Since ADPLL is designed to be critically damped, the initial oscillations in phase errors can be noticed in results before settling to final value.

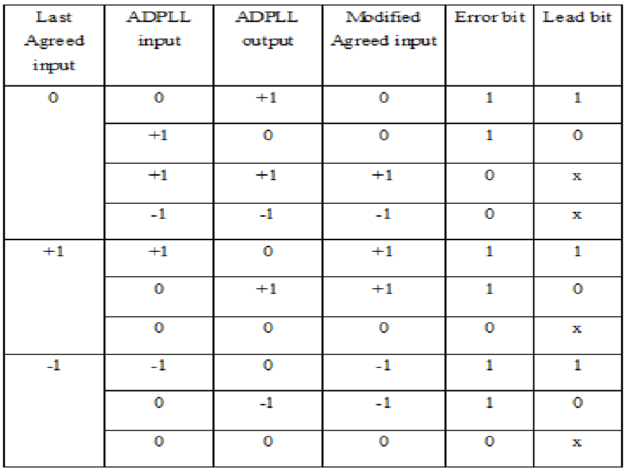
D. Design of Phase Detector

The traditional ADPLL operates on logic input, i.e. input is either 0 or 1. However, to track IRNSS signal, ADPLL must track a sinusoidal carrier which is quantized to 1.5-bit by ADC in front end circuit. Therefore, the phase detector of pll must be modified to determine phase difference between two sine waves, each represented in 1.5-bit format.

The phase detector produces a signal that is proportional to deviation in phase between incoming signal and NCO signal. If NCO signal changes prior to input then NCO is said to be leading, and a negative phase error is generated to slow down NCO. Similarly, if NCO signal changes after input signal, NCO is lagging, and positive error is generated. A phase error flag indicates any disagreement between input and NCO signal, while lead flag indicates lead or lag. In order to determine whether NCO is leading or lagging, we keep track of last agreed output. A lead flag is set if the last agreed output is high and NCO goes low before input. Because BPSK modulation is done on carrier, whenever the navigation message bit toggles, 180^0 phase shift occurs. Therefore, to account for this, the ADPLL must be designed such that it is insensitive to 180 degrees phase shift in carrier. This is done by modifying the phase detector to generate lead and lag signals based on 1.5-bit inputs rather than 1 bit input.

Table I depicts the working of 1.5-bit phase detector. The transition of outputs is clear from the fact that sine wave is represented in three values, 1, 0 and -1.

Table I - Logic Implementation Of 1.5-Bit Phase Detector.



Last Agreed input	ADPLL input	ADPLL output	Modified Agreed input	Error bit	Lead bit
0	0	+1	0	1	1
	+1	0	0	1	0
	+1	+1	+1	0	x
	-1	-1	-1	0	x
+1	+1	0	+1	1	1
	0	+1	+1	1	0
	0	0	0	0	x
-1	-1	0	-1	1	1
	0	-1	-1	1	0
	0	0	0	0	x

V. ARCHITECTURES OF CORRELATOR

Correlation is a statistical technique that shows how strongly pairs of variables are related. In satellite receivers, in order to lock onto a particular satellite signal, we must determine its CA code, carrier frequency and phase. This is accomplished by correlating the incoming IF signal with locally generated carrier and CA replica signals. 1 bit and 2-bit correlators are generally used in satellite communications.

A. 1-Bit Correlator and 2 Bit Correlator

1 bit correlator is the simplest correlator architecture. In this architecture both IF and carrier are represented by 1-bit. It is a single 3-input XOR gate followed by 1 bit accumulator. When the inputs are equal, XOR gate produces high output and when inputs are complementary, low output is produced and added to accumulator sum. In traditional 2-bit correlator IF samples and carrier are quantified in 2 bit representation, the sign bit is determined by XOR operation of sign bits of IF samples, carrier and CA code. The magnitude bits are determined by XOR for magnitude bits of IF samples and carrier. The value of base band accumulation can be obtained by the multiplication of IF samples, carrier and CA code. During accumulation, the number of four quantization values will be counted first, and multiply and add operations will be performed to get the final results [10].

$$R_{SGN} = +1 \quad S_{+1} = X_{SGN}\text{^}C_{SGN}\text{^}L_{SGN}$$

$$R_{SGN} = -1 \quad S_{-1} = \overline{X_{SGN}\text{^}C_{SGN}\text{^}L_{SGN}}$$

$$R_{SGN} = 1 \quad M_1 = \overline{X_{MAG}}\&\overline{L_{MAG}}$$

$$R_{SGN} = 2 \quad M_2 = \overline{X_{MAG}}\&L_{MAG}$$

$$R_{SGN} = 3 \quad M_3 = \overline{L_{MAG}}\&X_{MAG}$$

$$R_{SGN} = 6 \quad M_6 = X_{MAG}\&L_{MAG} \qquad \text{……… (6)}$$

The accumulation is computed as follows:

$$SUM= \sum [Bitsum(M_1S_{+1})+ 2\ Bitsum(M_2S_{+1})+ 3Bitsum(M_3S_{+1})+ 6\ Bitsum(M_6S_{+1}) - Bitsum(M_1S_{-1}) - 2Bitsum(M_2S_{-1}) - 3\ Bitsum(M_3S_{-1}) - 6\ Bitsum(M_6S_{-1})] \quad \ldots\ldots\ldots (7)$$

B. 1.5-Bit Correlator

1.5 bit quantization uses three values (−1, 0 and 1). It uses 1.5 bit quantization for both of its inputs [11]. The relation between inputs and outputs of 1.5 bit correlator are described in table III. It has reduced complexity and reduced SNR.

C. Modified 1.5-Bit Correlator

Unlike 1.5 bit quantization methods, modified 1.5 bit quantization uses 2 bit quantization for one input and 1.5 bit quantization for its second input. Because zero value exists in 1.5-bit quantization, the complexity is similar with 1.5 bit quantization, while the quantization noise introduced is similar with 2 bit quantization [11]. The table II shows the output R(t) for correlator under 2 bits IF samples and 1.5 bit carrier for various inputs. The table III describes the values of RMAG, M0, M1, M3 based on inputs IF and carrier when they are quantized in 1.5 or 2 bits. These are given as inputs to the accumulation module of the correlator.

Table - II Values of R(t) under 2 bits IF samples and 1.5 bit carrier

R(t)	R_{SGN}	$X_{SGN}*C_{SGN}*L_{SGN}$	R_{MAG}	X_{MAG}	L_{MAG}
+1	+1	+1	1	1	1
+3			3	3	1
-1	-1	-1	1	1	1
-3			3	3	1
0	-	-	-	-	0

Table - III Magnitude judgment under different 1.5-bit quantization methods

Architecture	IF	Carrier	Judgment
Modified 1.5 bit correlator	2	1.5	$R_{MAG}=0,\quad M0=\overline{L_{MAG}}$ $R_{MAG}=1,\quad M1=\overline{X_{MAG}}\&\ L_{MAG}$ $R_{MAG}=3,\quad M3=X_{MAG}\&\ L_{MAG}$
	1.5	2	$R_{MAG}=0,\quad M0=\overline{X_{MAG}}$ $R_{MAG}=1,\quad M1=X_{MAG}\&\ \overline{L_{MAG}}$ $R_{MAG}=2,\quad M3=X_{MAG}\&\ L_{MAG}$
1.5 bit correlator	1.5	1.5	$R_{MAG}=0,\quad M0=\overline{X_{MAG}}\mid\overline{L_{MAG}}$ $R_{MAG}=1,\quad M1=X_{MAG}\&\ L_{MAG}$

VI. RESULTS AND DISCUSSION

The verification and simulation results of designed ADPLL for various cases and simulation and synthesis results for various correlator is performed in ModelSim. The tracking system design is verified step by step to clearly state the coherence in demodulation process.

A. Simulation and Synthesis of ADPLL

The ADPLL designed in section IV was simulated and synthesized in ModelSim. The fig 6 shows ADPLL settling to final value. The variable NCO_out and o_output shows the incoming carrier and output of ADPLL, respectively. Variable (o_freq) represents the output frequency of ADPLL. Initially o_freq is 10MHz i.e free running frequency of ADPLL. Fig 6 shows the output frequency of ADPLL settling to its final value, 10.005MHz .

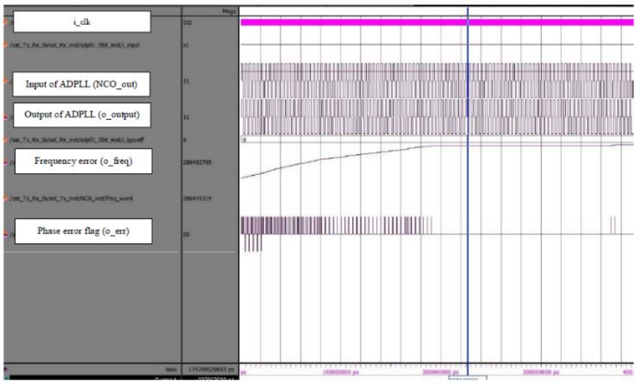


Fig. 6.Output of ADPLL showing the settling from 10MHz to final value of 10.005MHz

It can also be observed that frequency of occurrence of output error flag (o_err), reduces to zero implying that there is no phase error between the output and the input of ADPLL. It can be observed from the simulation results of the designed 1.5bit ADPLL that the 90% settling time is 714µs for the frequency resolution of 1.16Hz

The fig 7 shows the output of ADPLL indicating its insensitivity upon inversion of input. The input and output of ADPLL are i_input and NCO_out respectively. The signal *inv* shows the input being inverted after the lock state is achieved. The variable actualPhaseError in the simulation output shows phase error between input and the output. It can be observed that there is no change in actualPhaseError variable when input is inverted.

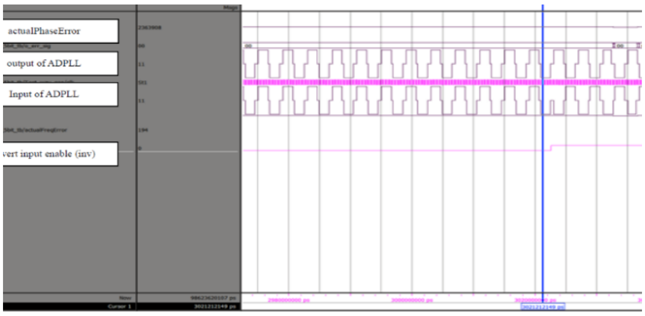


Fig. 7.Output of ADPLL demonstrating insensitivity to 180° inversion of carrier

Fig 8 shows ADPLL settling characteristics to 1MHz step. Although, sudden frequency steps do not occur practically, step change are used to show the stability of the system. It can be observed from fig 8 that the system remains stable for 1MHz step change in input signal.

Retrieval Number: J75820891020/2020©BEIESP
DOI: 10.35940/ijitee.J7582.0891020
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication

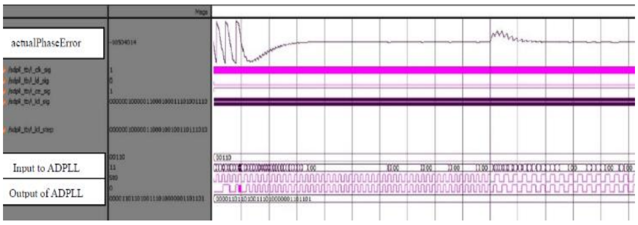


Fig. 8.ADPLL output depicting settling from 5MHz to 6MHz

B. Simulation and Synthesis of Correlator Architectures

Three correlator architectures are studied, simulated and synthesized in order to determine the architecture that offers a reasonable tradeoff between complexity and SNR, to be used in tracking system.

1) 2-Bit Correlator

In 2-bit correlator IF, carrier and CA code are quantized as 2 bits. Fig 9 shows the output of 2 bit correlator for correlated inputs and uncorrelated inputs. It can be observed that correlation, is very large(4454) in case of correlated inputs compared to that of uncorrelated inputs(814).

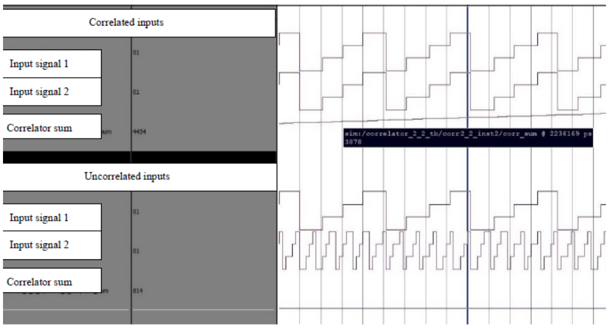


Fig. 9.Output of 2-bit correlator

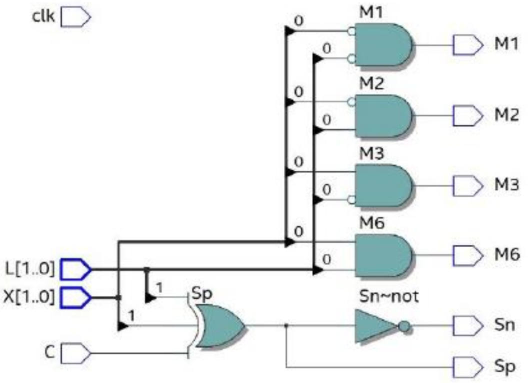


Fig. 10. RTL schematic for 2 bit correlator

The synthesis of 2bit correlator requires 3 standard LUTs. Fig 10 shows the RTL schematic for 2-bit correlator. It requires 3-bit accumulator which utilizes 17 LUTs. Since 2 bit quantization is used for both IF and local carrier, SNR=27.48 dB.

2) 1.5-Bit Correlator

In 1.5-bit correlator IF, carrier and CA code are quantized as 1.5 bits. Fig 11 shows the output of 1.5 bit correlator for correlated inputs and uncorrelated inputs. The correlation sum is found to be very large (489) in case of correlated inputs compared to that of uncorrelated inputs (258). The synthesis of 1.5-bit correlator requires 2 standard LUTs. Fig 12 shows the RTL schematic for 1.5-bit correlator. It requires 2-bit accumulator which utilizes 13 LUTs. Since 1.5 bit quantization is used for both IF and local carrier SNR=21.46 dB.

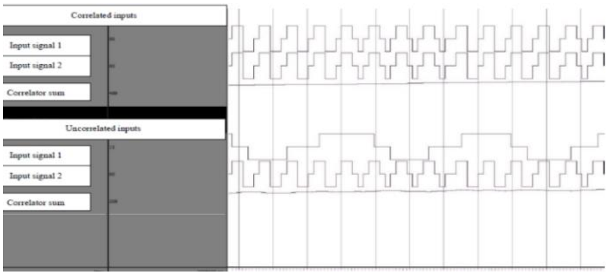


Fig. 11. Output of 1.5-bit correlator

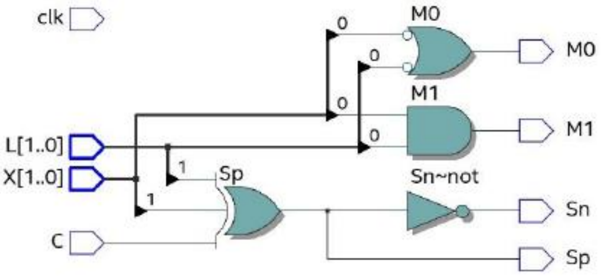


Fig. 12. RTL schematic for 1.5 bit correlator

3) Modified 1.5-Bit Correlator.

In modified 1.5-bit correlator IF is quantized as 1.5 bits and carriers quantized as 2 bits. Fig 13 shows the output of 1.5-bit correlator. Output is similar to 2-bit correlator in terms of SNR and close to 1.5 correlator in terms of complexity.

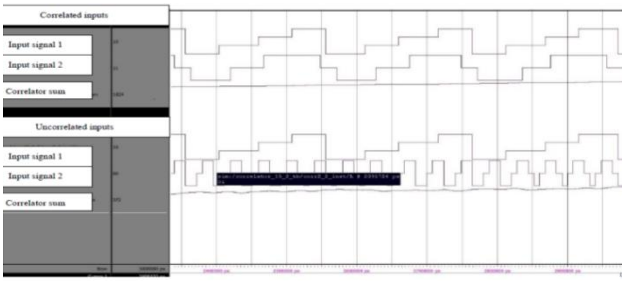


Fig. 13. Output of modified 1.5-bit correlator under 2-bit IF sample and 1.5-bit carrier quantization

The synthesis of 1.5-bit correlator requires 2 standard LUTs. Fig 14 shows the RTL schematic for modified 1.5-bit correlator. It requires 2-bit adder accumulator which utilizes 13 LUTs.

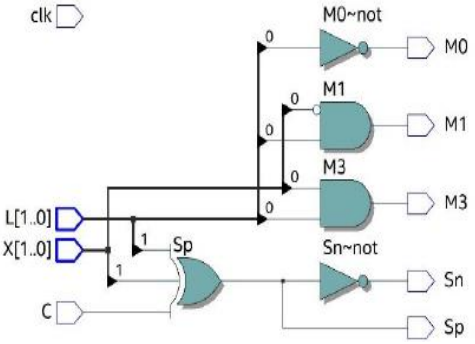


Fig. 14. RTL schematic for modified 1.5 bit correlator

Since 2-bit IF samples and 1.5-bit carrier samples are used, SNR is given by root mean square, SNR=25.48 dB.

The table IV indicates the complexities and the SNR values (in dB) for various architectures of correlators simulated.

Table - IV Comparison of complexities and SNR of 2-bit, 1.5 bit and modified 1.5 bit correlators

Correlator architecture	Complexity							SNR (dB)
	From architecture (No. of gates)					From synthesis (No. of LUT's)		
	XOR	AND	OR	NOT	Adder size	Correlator	Adder	
1 bit	1	0	0	0	1 bit	1	10	15.56
2 bit	1	4	0	5	4 bit	3	17	27.48
1.5 bit	1	1	1	3	2 bit	2	12	21.46
Modified 1.5 bit	1	2	0	3	2 bit	2	14	25.48

C. Simulation and Synthesis of Tracking System

The designed tracking system was simulated and synthesized using ModelSim and Quartus II software. The results show that design reliably recovers navigation message. To verify the functionality of the designed tracking system, the standard IRNSS satellite signal transmitter is simulated in Verilog and used to generate input to designed tracking system.

1) Results for Carrier Tracking Loop:

The input to ADPLL is a BPSK modulated carrier wave which is received from satellite through the antenna, passed through noise filter and quantized to 1.5 bits (in this paper, for this purpose the similar signal is generated). The signal can assume values -1, 0 and 1. In fig 15, input signal of carrier tracking loop, i_input is of frequency 10.005 MHz (which is the maximum possible positive deviation for IRNSS system). The ADPLL is designed to have a free running frequency of 10MHz.

From the simulation results, it can be observed that the output of ADPLL, NCO_out has perfectly locked onto incoming wave, both in terms of frequency and phase. For the loop coefficient of value 10, the settling time required to attain 90% of final value is 714µs.

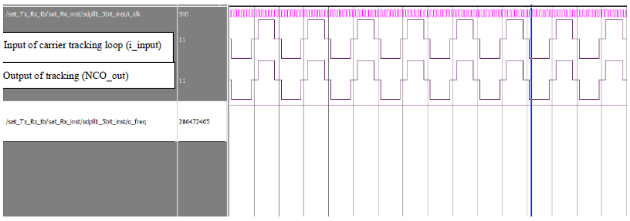

Input of carrier tracking loop (i_input)
Output of tracking (NCO_out)

Fig. 15. Output for carrier tracking using 1.5-bit ADPLL after lock state is achieved.

2) Results for Code Extraction from Tracked Carrier Signal:

To extract the CA code, we multiply the incoming signal(in) with the ADPLL output (pll_out). If both are in phase with each other, the associated code is +1, if both are out of phase, the associated code is -1. However, due to jitter, the output of multiplication operation (multiplier) is passed through LPF. Fig 16 shows the output for code extraction.

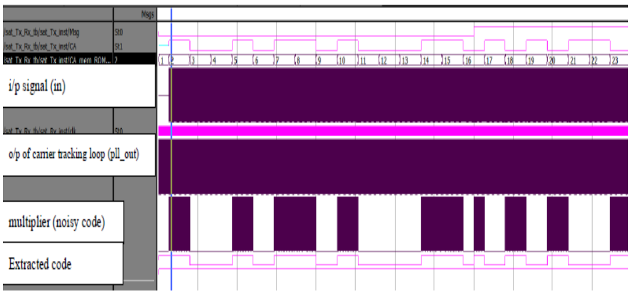


Fig. 16. Output for code extraction

3) Results for Code Tracking Loop Through Correlation:

The code thus extracted is tracked to recognize if it is leading or lagging with respect to locally generated code. This is performed through process of correlation. Fig 17 shows the results of code tracking loop. The extracted code (in) is correlated with early, prompt and late version of locally generated CA code. The early/late select module (E/L select module) is designed to estimate the moving average of 32 signal samples. Since delay of E/L select module is negligible, that the maximum time required for settlement of code tracking loop is calculated to be 31.28ms.

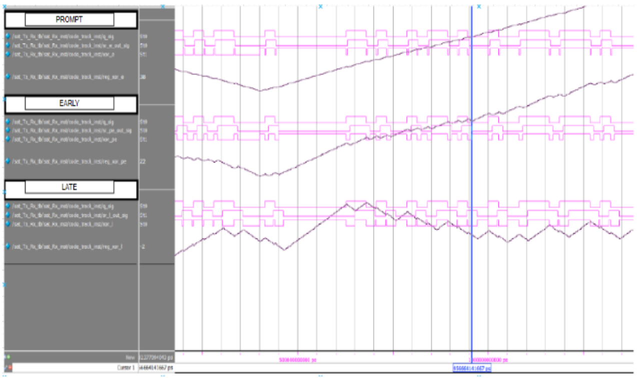


Fig. 17. Results of code tracking loop

In fig 17, the inputs and outputs of early, prompt, and late correlators are grouped separately. Among the three, the correlator whose accumulator holds the largest sum indicates the most closely aligned version of the code. If any code offset is observed, E/L module generates a control input to compensate for the offset and to obtain perfect alignment in prompt correlator. So prompt correlator gives the largest accumulation sum.

4) Results for Navigation Message Extraction:

The carrier and code tracking loop demodulate the incoming signal and navigation message is obtained as binary data. From correlator output, the change in slope of accumulator indicates the toggling of navigation message bit. Fig 18 shows that by performing xor operation between extracted CA code and perfectly aligned version of CA code obtained from the prompt correlator of the code tracking loop, each bit of navigation message is obtained (Msg) and is fed into a shift register to fully reconstruct the navigation message.

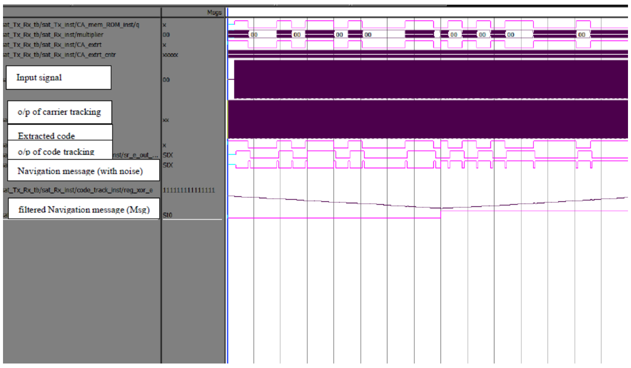


Fig. 18. Results for extraction of navigation message.

VII. CONCLUSION

In this paper, tracking system for IRNSS satellite navigation has been designed using 1.5 bit All digital PLL and correlator. The receiver accurately detects and demodulates the satellite signal to extract binary navigation message. The design used 1.5-bit ADPLL for carrier tracking and 1.5-bit correlator is used for code tracking. Unlike existing architecture for satellite tracking. The tracking operation presented in this paper does not relay on SDR's and tediously iterative algorithms that require to perform several loop operations but rather uses a PLL which is a feedback control system to lock onto carrier frequency, thus the complexity is reduced. The designed ADPLL exhibits settling time of 714µs and offers a frequency resolution and free running frequency of 1.16Hz and 10 MHz, respectively, which is suitable for high frequency communication applications. For code tracking, modified 1.5-bit correlator is used. The correlator utilizes 16 LUTs and gives 25.48 dB SNR which is better than 1.5-bit correlator. The design is simulated and synthesizes in ModelSim. It uses utilizes 77 standard LUT which is less compared to logic utilization of general-purpose processor, MicroBlaze which uses 1169 LUTs [13].

REFERENCES

1. M. Braasch and A. Dempster, "Tutorial: GPS receiver architectures, front-end and baseband signal processing," in IEEE Aerospace and Electronic Systems Magazine, vol. 34, no. 2, pp. 20-37, Feb. 2019.W.-K. Chen, *Linear Networks and Systems* (Book style). Belmont, CA: Wadsworth, 1993, pp. 123–135.
2. Dr. S. Muthukumar, M. K. Subramaniya Raman, D Jagadeeshwaran, V Saveetha, 2020, Signal Acquisition of IRNSS Signals, International Journal Of Engineering Research & Technology (IJERT) Volume 09, Issue 06 (June 2020).
3. D.J.R.Van Nee and A.J.R.M. Coenen, "New Fast GPS code-acquisition technique using FFT," in Electronics Letters, vol. 27, no. 2, pp. 158-160, 17 Jan. 1991.
4. A.R.Yashaswinia P.Siva Nagendra Reddy G.N.Kodanda Ramaiahb "Generation and implementation of IRNSS Standard Positioning Signal" Engineering Science and Technology, an International Journal Volume 19, Issue 3, September 2016, Pages 1381-1389.
5. E. Schmidt, D. Akopian and D. J. Pack, "Development of a Real-Time Software-Defined GPS Receiver in a LabVIEW-Based Instrumentation Environment," in IEEE Transactions on Instrumentation and Measurement, vol. 67, no. 9, pp. 2082-2096, Sept. 2018, doi: 10.1109/TIM.2018.2811446.
6. E. Wang, S. Zhang, Q. Hu, J. Yi and X. Sun, "Implementation of an Embedded GPS Receiver Based on FPGA and MicroBlaze," 2008 4th International Conference on Wireless Communications, Networking and Mobile Computing, Dalian, 2008, pp. 1-4, doi: 10.1109/WiCom.2008.335.
7. S. B. Totad, K. S. Rakesh, N. N. Nagendra and M. Sowmya, "Design and simulation of tracking loops for IRNSS receiver in SIMULINK®," 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), Chennai, 2017, pp. 1548-1553, doi: 10.1109/ICECDS.2017.8389705
8. Borre, K., Akos, D.M., Bertelsen, N., Rinder, P., Jensen, S.H. "A Software-Defined GPS and Galileo Receiver- A Single-Frequency Approach" 2007, 978-0-8176-4540-3.
9. A. K. Chaudhary and M. Kumar, "Design and implementation of ADPLL for Digital communication applications," 2017 2nd International Conference for Convergence in Technology (I2CT), Mumbai, 2017, pp. 397-401, doi: 10.1109/I2CT.2017.8226159.
10. Hongwei Zhou1, Tian Jin2, and Fangyao Lü2 &ldquo "Design of 1.5 bit quantization correlator in satellite navigation software receiver" rdquo; Journal of Systems Engineering and Electronics Vol. 27, No. 2, April 2016, pp. 449 − 456.
11. J. Tao, W. Yu. "A Real-time GPS software receiver correlator design for embedded platform". Proc. of Institute of Navigation Global Navigation Satellite Systems Conference, 2011: 808−812.
12. Y. Yang, X. Ba and J. Chen, "A Novel VLSI Architecture for Multi-Constellation and Multi-Frequency GNSS Acquisition Engine," in IEEE Access, vol. 7, pp. 655-665, 2019.
13. Xilinx "MicroBlaze Processor Reference Guide" UG984 (v2017.3) October 4, 2017.

AUTHORS PROFILE

M. Udaya received her B.Tech degree in ECE from Keshav Memorial Institute of Technology, Hyderabad. She is pursuing M.E in Embedded Systems and VLSI Design from Chaitanya Bharathi Institute of Technology. Her areas of interest include VLSI design for digital signal processing.

D. Sony is presently working as Assistant professor in ECE Dept, Chaitanya Bharathi Institute of Technology, Hyderabad. Having 11 years of experience in teaching, her areas of interest include Embedded System Design, Digital system design and GNSS/NavIC applications.

Dr. D. Krishna Reddy has obtained B.E. degree in ECE from Andhra University, Waltair with distinction. Obtained his M.E. and Doctoral degrees from Osmania University. He is Recipient of National Merit Loan Scholarship from Govt. of India. Worked as Graduate Engineer with Captive Power Plant of ICCL, Bhuvaneswar for one year. About 100 publications to his credit, co-authored 13 technical reports and 03 Book/Chapters. Successfully completed one R&D project Sponsored by SERB, DST and one consultancy project under CARS scheme sponsored by RCI, DRDO. Member of Critical Design Review committee of project entitled "SHAKTI" of program SAMUDRIKA being executed at DLRL, DRDO Hyderabad. Three Research scholars got awarded Ph.D degree and 14 research scholars are pursuing under his guidance. Received Best Teacher award from the Institute during 2018 Shruthi celebrations. Visited Georgia Institute of Technology Atlanta, University of Illinois Chicago and Illinois Institute of Technology Chicago. Successfully organized AICTE sponsored Seminar along with two intensive courses "Electronic Intelligent Systems" and "Communication Intelligence Systems" With Grant-in-Aid from DRDO, New Delhi jointly with DLRL, DRDO, Hyderabad. His areas of research interest include Navigation, 5G and Signal Processing. He is a fellow of IETE, member of IEEE, life member of ISTE and life member of SOEMCEI.