FPGA Based Intelligent Antenna Alignment System

Rabi Adhikary Sourav Nath Tapojyoti Bhattacharjee

Electronics and Communication Engineering Department.
Sikkim Manipal Institute of Technology, Sikkim Manipal University.
Majitar, Rangpo, East Sikkim 737136, India.

ABSTRACT
This paper describes how an antenna can be positioned according to the received signal strength. We used Field Programmable Gate Array (FPGA) based intelligent alignment system which aligns antenna automatically towards the maximum SNR. We have used a Yagi-Uda 7-element simple dipole antenna for transmission and folded dipole with reflector for reception, as a part of our demonstration. The individual signals that are obtained from the detector, after converted into their digital domain using Analog to Digital Converter (ADC), is fed into the FPGA. The signals are then compared and the point at which the maximum signal is noted. Finally it guides the antenna in the direction of the maximum signal received. The proposed system has been realized using Xilinx FPGA Spartan3E XC3S500E. The experimental results are found to be positive and effective.

Keywords
FPGA, SNR, ADC, P2P, LOS.

1. INTRODUCTION
In today's world, most of the communication system based on wireless where point to point (P2P) or line of sight communication (LOS) is required. In satellite communication, accurate alignment of the two communicating devices is the most crucial factor for establishing an effective and efficient communication. In our present work we attempt to implement an intelligent antenna system that can aligns itself to the direction of optimum SNR value of the received signal. The software platform used here VHDL language in Xilinx 9.2i. This software is ported in the Spartan3E starter kit [1]. We developed a prototype model which is completely operational at SMIT laboratory. Basically the parallelism strategy is explored here based on the Field Programmable Gate Array (FPGA). The FPGAs provide a flexible platform for fine-grained parallel computing based on reconfigurable hardware [2].

It is already mentioned above the antenna we used here Yagi-Uda 7–element which is a parasitic linear array of parallel dipoles and one of which is energized directly by a feed transmission line while the other act as parasitic radiator whose currents are induced by mutual coupling. The basic antenna is composed of one reflector (in the rear), one driven element, and one or more directors. The Yagi-Uda antenna has received exhaustive analytical and experimental investigations in the open literature and else where. The characteristics of a Yagi-Uda are affected by all of the geometric parameters of the array. Usually Yagi-Uda arrays have low input impedance and relatively narrow bandwidth. Improvements in both can achieved at the expense of others. Usually a compromise is made which depends on the particular design [3] and due to its directivity. This high directivity also requies a precise alignment with the satellite downlink beam [4].

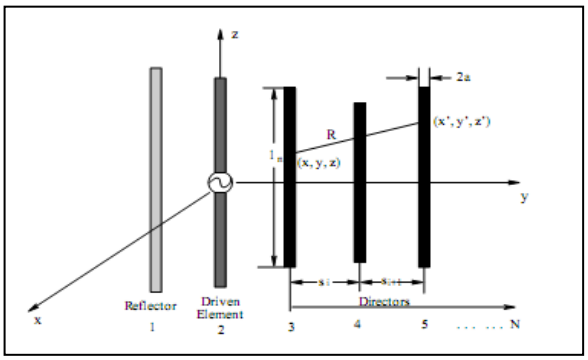

Fig.1 Geometrical structure of Yagi-Uda array [3]

The received power, Pr at the receiving antenna located at a distance, d from the transmitter which is given for free space propagation [6].

$$P_r = P_t \left(\frac{\lambda}{4\pi d}\right)^2 G_b G_m$$

If other losses (not related to propagation) are also present, we can rewrite the above equation as[6]

$$\frac{P_r}{P_t} = \left(\frac{\lambda}{4\pi d}\right)^2 \cdot \frac{G_b G_m}{L_0} = \frac{G_b G_m}{L_p L_0}$$

Where

Pr = Received Power
Pt = Transmitted Power
λ = Wavelength
G_b = gain of the transmitting antenna
G_m = gain of the receiving antenna
d = antenna separation distance between transmitter and receiver
L_o = other losses expressed as a relative attenuation factor

$L_p = \left[\frac{4\pi d}{\lambda}\right]^2$ = free space path loss, often expressed

as an attenuation in decibels(dB).

$$= 20\log\left(\frac{4\pi d}{\lambda}\right)(\text{dB})$$

In our experiment the L_o is not considered as there is minimum distance between the two antennas.

The polar plot of the Yagi Uda 7-element simple dipole antenna is shown in fig.2 which gives the justification of using it as our part of experiment.

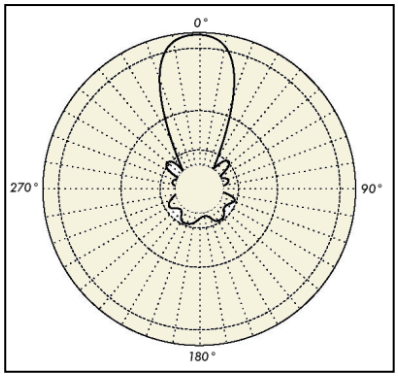

Fig.2 Plot of Yagi-Uda 7element dipole antenna [5]

2. ALIGNMENT SYSTEM

The system environment is the real-world environment with noise and unwanted multipath signals. These two factors are inevitable as the channel is wireless. For that we have used a additional capacitive filter at the input of the ADC. The total alignment system can be subcategorized into three different categories, i.e *Scanning*, *Data acquisition* and *Positioning* which are discussed under the following headings. The block diagram of the whole system is shown in the fig.3 which describes our complete work in the field of communication.

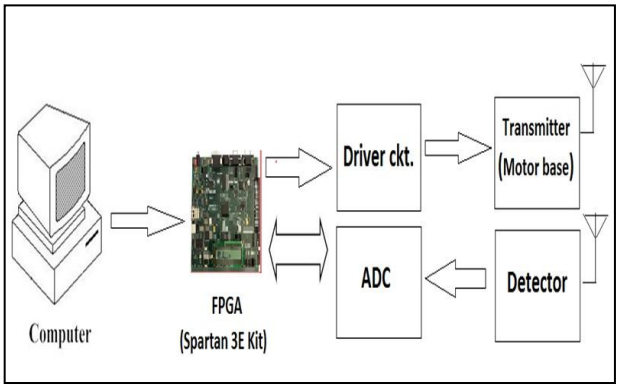

Fig.3 Block diagram of intelligent alignment system

2.1 Scanning

By scanning we are concentrating on the fact that both the systems are stationary with respect to their geographical aspect and both the systems must be able to either receive or transmit or reflect EM waves with considerable signal strength. As we are living in a 3D world so scanning should be done in three planes which are X, Y and Z planes. Normally Azimuthal Angle scan and Elevation Angle scan are formed by above three planes. The prototype we have designed only for X-Y plane scanning. For this purpose we have used a transmitting antenna which is radiating 750 MHz sine wave signal and mounted over a motorised tool. The motorised tool is controlled by FPGA which rotates the antenna through 360^0. The algorithm for scanning is shown in fig.4 which is the clockwise rotation of the complete 360^0 angle in X-Y plane.

```
begin
motor:="1100";
process(clk_for_motor)
begin
if clk_for_motor is enabled then;
motor:=shift the logic levels towards clockwise;
end if;
end process;
end;
```

Fig.4 Scanning algorithm

2.2 Data Acquisition

The purpose of data acquisition is to measure an electrical or physical phenomenon such as voltage, current, temperature, pressure and sound. PC-based data acquisition uses a combination of modular hardware, application software and a computer to take measurements. While each data acquisition system is defined by its application requirements then every system shares a common goal of acquiring, analyzing and presenting information. Data acquisition systems incorporate signals, sensors, actuators, signal conditioning, data acquisition device and application software. Here the data acquisition is done for comparing the incoming data with the initial data for finding the maximum SNR that are received from the input port of the Spartan3E which is stored in a temporary variable. After that comparison is done with the initial data, if the incoming data is greater than the stored initial data then initial data is updated with the recent data and process will continue till 360^0 rotation of the antenna. The algorithm of data comparing and saving the maximum signal value is shown in fig 5.

```
begin
process(data_clock)
If data_clock is enabled then
clock_count=0;
maximum_snr <= initial data;
data_var <= adc_data;
maximum_snr <= data_var
clock_count++;
else
clock_count <= clock_count;
end if;
end if;
end process;
end;
```

Fig.5 Data comparison algorithm

2.3 Positioning

The Positioning of antenna means, after getting the maximum SNR by Scanning the maximum signal position is noted by counting the number of clock during the scanning. With the number of clock obtained, the antenna is again rotated in anticlockwise direction up to the number of clock as such the antenna is positioned in the direction of maximum SNR. The algorithm for Positioning of the antenna is shown in fig.6.

```
begin
motor:="1100";
process(clk_for_motor)
begin
if clk_for_motor is enabled then
motor:= shift the logic levels towards anticlockwise;
end if;
end process;
end;
```

Fig.6 Positioning algorithm

3. MOTOR CONTROL

A stepper motor is a good choice whenever controlled movement is required and data is totally measured by a measurement unit. That can be used to advantage in applications where we need to control rotation angle, speed, position and synchronism. Thus, selection of stepper motor in our work completely fits the profile. The Transmitting antenna is mounted by an antenna mast over a belt driven platform to support heavier load. The FPGA is programmed to rotate the stepper motor through an angle of 360^0 with a step size of 1.8^0 (approx.) which in turn rotates the belt driven larger diameter platform with a step angle of 0.54^0 by switching the motor coils at 20 Hz frequency. The 50 MHz internal clock of FPGA is divided to generate the 20 Hz clock cycle for motor rotation thus exempting usage of any external crystal. The algorithm for the clock division is shown in fig.7.

```
clk_in:= onboard clock(50 MHz);
begin
process(clk_in)
begin
clk_in_count:=initial_value;
clk_generated:=logic_low;
if clk_in_count=present_count_value then
clk_generated:= logic_high;
end if;
end process;
end;
```

Fig.7 Clock division

4. ADC INTERFACING

The interfacing of the A-to-D converter is configured around ADC 0808 IC with the supporting control signals from FPGA. The ADC 0808 IC is an 8-bit A-to-D converter, having data lines of D0-D7. It works on the principle of successive approximation and there are some control signals which drives the ADC to give digital output to Spartan3E input port. The control signal is generated by the FPGA and it is fed to the ADC through the Spartan3E output port.

```
begin
process(clk_adc)
begin
If clk_adc is enabled then
SOC, EOC, OE is initialized to their respective states;
SOC:='0';
SOC:='1';
  If EOC is low then
  OE:= high;
  data_var <= ADC_data;
  end if;
end if;
end process;
end;
```

Fig.8 ADC conversion

The main control signals are Start of Conversion (SOC), End of Conversion (EOC) and Output Enable (OE). Start of Conversion initiates the ADC to start the conversion and End of Conversion is the output signal of the ADC which should be checked by the FPGA, if the EOC is low then it has to generate a control signal which is Output Enable. Output Enable is a signal which is the identification of collecting the digital data from data bus of the ADC. The ADC is made operational by generating afore said control signal through the code. The algorithm of the ADC interfacing is shown in fig.8.

5. RESULTS

Internal clock of the FPGA worked in high frequency, but normally low frequency clock controlled the stepper motor rotation. So we required to divide the internal clock into very low clock cycle. The simulation results of clock division is shown in fig.9.

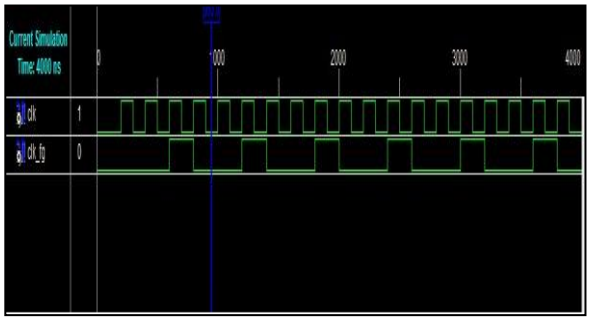

Fig.9 Clock division

As we know the stepper motor we have used is four coil stepper motor whose specification is stated above, that can be rotated by energising the coil in different ways. We have energised two coils of the motor at the same time so that maximum torque can be produced. Through our code we are giving the pulse to the coil. The result of the code for controlling motor is shown in fig.10.

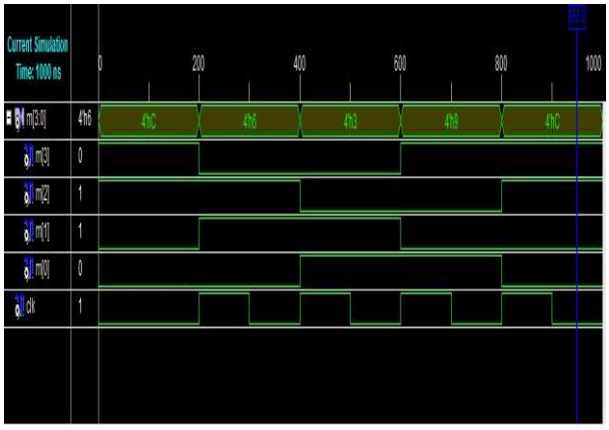

Fig.10 Motor control

The combined result of our system is shown in fig.11. It has some undefined signal shown in the figure due the non real time simulation, as some signals are not coming from input port at the time of simulation.

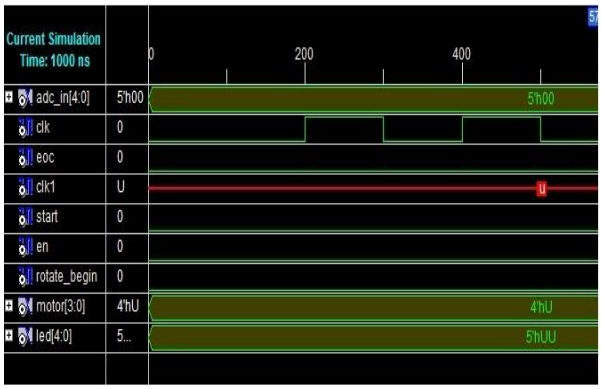

Fig.11 Complete simulation of our system

As we have done our work in different sections and combined the individual section and corresponding results are shown, to form the complete system. The whole works can be summarized under the following flow chart which is given in fig12.

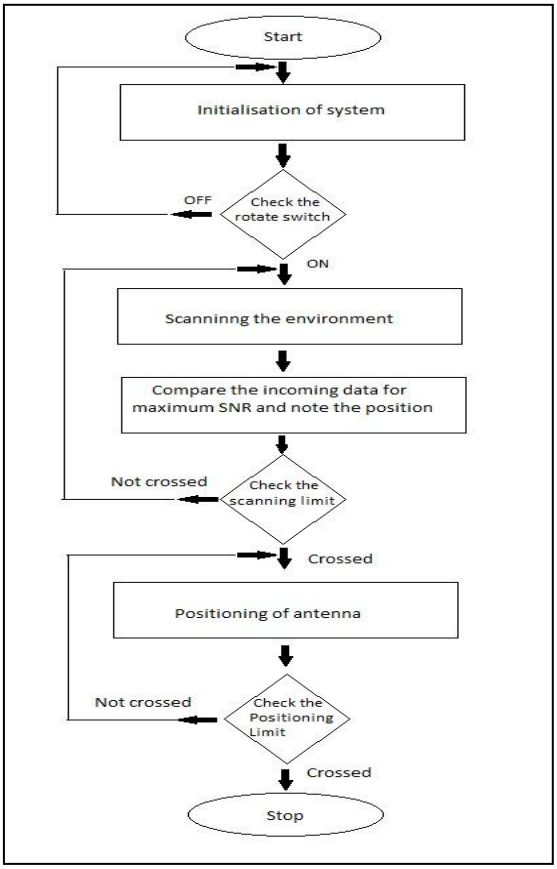

Fig.12 Flow chart for intelligent alignment system

The complete experimental setup shown in fig.13. It comprises with transmitting and receiving antenna system, spartan kit, driver ckt, ADC and PC.

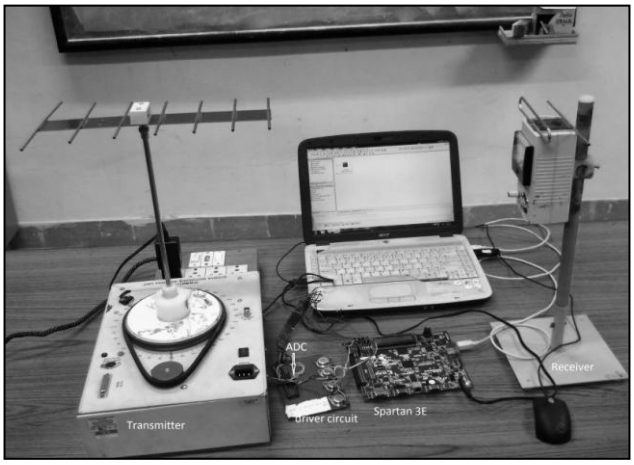

Fig. 13 Experimental setup

6. CONCLUSION

The system works well in the SMIT laboratory and it will reduce the human efforts of aligning different type of antenna. Our work can be directly implemented in DTH antenna alignment. The driver circuit for driving the motor coils can be realized through IC based current buffers but we have implemented using power transistors (2N3055) for the sake of simplicity. The circuit arrangement can also be modified by using onboard ADC in Spartan3E board. The VHDL code is simulated in Xilinx 9.2i and downloaded in Spartan3E xc3s500e FPGA board. We can improve our processing speed using the Vertex IV supported Software Define Radio (SDR) which is itself a real time development platform.

7. REFRENCE

[1] Xilinx Tutorial Documentation, "*ISE 9.1i Quick Start Tutorial*.", Copyright © Xilinx, Inc. All right reserve, 1995-2007.

[2] Tim Oliver, Bertil Schmidt, Darran Nathan, Ralf Cldemens and Douglas Maskell "*Using Reconfigurable hardware to accelerated multiple sequence alignment with clustalW*" Vol.21 no.16 2005,pages 3431-3432

[3] Dong Xue, Department of Engineering Mechanics " *Yagi-Uda Antenna*"

[4] Sourav Dhar, R. Bera, D. Ghosh, Biswadeep Dasgupta and Bandaru Viswavikyath "*Automatic* Antenna Assembly for DVB Access on the Move" International Journal of Networks and Application. Volume 1, Number 1 (2010), pp. 19—28

[5] Wireless Networking, available at www.vias.org/wirelessnetw/wndw_06_05_05.html

[6] Vijay K. Garg, " Wirless Communication and Networking" 1ed., Elsvier 2007

[7] Kariyappa B. S., Hariprasad S. A. and R. Nagaraj "*Position Control of an AC Servo Motor Using VHDL & FPGA*" World Academy of Science, Engineering and Technology 49 2009

[8] Douglas L. Perry "*VHDL Programming By Example*", 4 ed., Mc Graw-Hill, USA, 2002.

[9] XILINX, available at www.xilinx.com/support/documentation/boards_and../ug230