Simulation of Direct Digital Synthesizer with LabView

Elan Djaelani[1]

Abstract—***Direct Digital Synthesizer (DDS) is a frequency synthesizer to generate an arbitrary waveform. It is widely used for telecommunication applications such as RADAR and JAMMER. Nowadays, DDS is preferred than its traditional counterpart, the Phase-Locked Loop (PLL). However, the price of DDS is still relatively expensive. For this reason, it is important to develop a simulation for DDS as a cheaper alternative to learn about DDS, especially for students. National Instrument (NI) has developed a simulation for DDS. However, this simulation may not accessible for student either. In this research, a simulation for Direct Digital Synthesizer (DDS) is developed. The DDS simulation developed by the National Instrument is improved by adding new features. The added features are the storage system, an integrated DAC and the use of 20 bit data for lookup.***

Keywords—radio communication, direct digital synthesizer, simulation.

I. INTRODUCTION

Direct Digital Synthesizer (DDS) is widely used for telecommunication applications such as RADAR and JAMMER[1],[2]. Indonesian Institute of Sciences (IIOS) have developed a maritime radar and surveillance radar, which is used to measure distances and speed of ships and to guide them. This radar is a Frequency Modulation Continuous Wave (FMCW) radar. It is a new generation radar which continuously transmit a signal. This is different from the conventional pulse radar which transmit a pulse periodically. For FMCW radar, the pulse is replaced by a signal which is modulated with a signal, which is called a carrier, that has a period equals the period of the pulse. This carrier can be sinusoid, sawtooth, triangle, etc. In radar, DDS is used to generate a chirp signal. A chirp signal is a signal in which the frequency changes (increase or decrease) with time. This signal can be generated by modulating a signal with sawtooh signal. IIOS also have developed a jammer that operates in military frequency. Jammer is a radio transmitter which is used to interfere radio communications. This application also implement DDS.

DDS [3] is a technique to generate an arbitrary output signal from a single and fixed reference clock. DDS is commonly used for generating a signal, as oscillators, modulators, function generators, etc in telecommunication systems. In DDS, the reference clock is divided with a fixed scaled factors in a programmable binary tuning number. The tuning number is between 24 to 48 bits. For this reason, DDS can generate a high resolution of tuning frequency. DDS also has a high speed and high performance Digital to Analog (D/A) converter. The architecture of DDS can be seen in Fig. 1. These capabilities have made DDS is currently preferred over its analog counterpart, the Phase-Locked Loop (PPL). Besides DDS' higher resolution and higher speed than PPL, it is digitally controllable and it has very high hopping speed in the output frequency or phase.

Due its importance, the interest of learning DDS is increasing rapidly. However, the price of DDS is still relatively expensive, especially for students. For this reason, it is important to develop a simulation for DDS as a cheaper alternative to learn about DDS. A simulation of DDS has been developed by National Instrument (NI). While it provide a cheaper solution, this simulation may not accessible for student either. In this research, a simulation for Direct Digital Synthesizer (DDS) is developed. The DDS simulation developed by the National Instrument is improved by adding new features. The added features are the storage system, digitalization of the output signals and the use of 20 bit data for lookup memory. By doing so, we hope the simulation will be accessible for students to learn DDS.

II. METHOD

A. DDS simulation by National Instrument

In this section, the DDS simulation from NI is briefly explained [5]. Fig. 2 shows the block diagram of DDS simulation. The simulation from NI uses 8 bit data. A DDS basically consists of a lookup memory, an accumulator, a counter and a Digital to Analog Converter (DAC). The front panel of the simulation can be seen in Fig. 3. It consists of accumulator counter and look up memory address (in binary and decimal). From this figure we can see that the frequency can easily be set (in this case 7 Hz).

B. Proposed Design

The evaluation of 6-model rudders performance affect to the ship maneuvering performance was determined following [7]. In this computation, the 6-models of rudders was installed on the ship which have the main dimension of length between perpendicular (L) 99 m, width The block diagram of our DDS simulation can be seen in Figure 4. The block diagram consists of Lookup memory, Accumulator, and Measurement file. The design of look up memory can be seen in Fig. 5. The look up memory block has four components. They are divider, multiplier, degrees to radian converter, sine computer.

Look up memory is used to store the sine data. In this memory there are several samples. LM contains the information of one cycle of the desired waveform. For instance, we use 20 bits LM.

Therefore there are 1048576 samples which contain the information of a cycle of sine waveform. Every samples is a 16 bits integer value. A single cycle of a

[1]Elan Djaelani is with Puslit Informatika, Lembaga Ilmu Pengetahuan Indonesia, Indonesia. Email: elan@informatika.lipi.go.id.

sine waveform (360 degree) can be represented by 2^{20} points.

In this design, N= 2^{20}=1048576 samples. Therefore the angle for each sample is (360^0)/1048576 = 0,0003433 degree. Each point (in degrees) converts to radians (devided by 180 degree into the number multiplied by π to give the exact angle in radians). In radian this equals (0,0003433)/(180) x3,14=5.9x10^{-6} radian. Calculate the sine function for all angles in radians (2^{20} points).

Figure 6 shows the design of the accumulator. Accumulator consists of divider, multiplier, and a converter. The converter consists of numerical converter into 32 bits integer value, fixed or integer number converter into array, and Boolean array converter into 32 bits integer.

Accumulator works in a similar fashion as a counter. However, an accumulator has more advantages. One of them is the ability to count with almost all number with one. For this example, Accumulator is simulated with while loop and a shift register to do the counting by Frequency Control Word (FCW). The counting is represented by U32 to simulate a 32 bit counter. When the highest value achieved, a counter acts as an accumulator.

The value in accumulator is converted into a Boolean arrays to make it easy to take the highest bit, and then convert this value into an integer. The highest bit of the 20 bits of the accumulator is then sent into LM and the array index. The value of the index of the LM is an integer which represent a 16 bit samples of a sive waveform to be sent to DAC.

The accumulator works as follow. Let N be the number of counting to be perform and i is the index of iteration. The accumulator will run for N times. The angle obtained is in degrees, which is then converted into radian. The computation continues until we obtain N samples.

Figure 7. shows the design for measurement file. This block consists of :

1. Signals: the signals which value need to be stored should be connected to this block. If there are more than one signals then merge icon block should be used.
2. Enable: should be TRUE if the data wants to be stored.
3. File name: is the name file which also include the place where the data to be stored. For instanceD:\data.

Figure 8 is the front panel of the DDS simulation. There are two displays: Lookup Memory (a), Waveform of sample sent to DAC (b), and Waveform graph output DAC (c). In LM memory display, there is a cycle of sine. The vertical axis is the amplitude (-1 to 1) and the horizontal axis the sample number (0 to 1000000). In the display of the Waveform of sample sent to DAC, it contains a discrete signal. It consisted of discrete signal. The vertical axis is the amplitude (-1 to 1) and the horizontal axis is time (0 to 6 second). In the display of the waveform of output DAC, it contains an analog signals. The vertical axis is the amplitude (-1 to 1) and the horizontal axis is time (0 to 1250000). Waveform of output DAC show in Figure. 8 (c).

III. Result and Discussion

In our simulation, we add a data storage system. The format of this data storage is in text file. Therefore it can be opened using notepad or other similar software. By doing so, an offline analysis can be performed. The text format is used so the analysis can be done using other software such as excel and origin. The sample of the data stored can be seen in table 1. The data in the first column is the address memory and the second column is the sample value.

In addition to the storage systems, our simulation software has other advantages compared to the simulation software from DDS. The data from our simulation is already in digital form. This is because the data is already the output from DAC. Lastly, we implement 20 bits data for look up table. By doing so, the accumulator can take the address faster. The software also become simpler. For NI signal generators, a 48-bit phase register is used for maximum precision. Of these, 20 bits are used to choose a sample from the lookup table. So, not every address is used. This structures is simpler and allow faster information retrieval.

IV. Conclusion

We have developed a software simulation for DDS. In our simulation, we improved the DDS simulation from National Instruments by adding the following functions. First, a data storage system is added. Data is stored in an array while the program is running, the size of which depends on the numbers of variables being measured, the runtime of the experiment, and the sampling frequency. When the experiment ends the data is written to a tab separated value (lvm file) and saved on the hard disk of the PC. Data Storage is used to analyze data for meaningful information .Lvm files can be opened by Origin or Microsoft Excel for inspection and plotting the data. Second, the data is already in analog form. This is because the data is already the output from DAC. Third, 20 bits data is used for look up memory. By doing so, the software is simpler and faster. Our hope is by having this simulation software, it is easier and cheaper for students or the users to learn DDS.

Acknowledgement

The author would like to express thanks to LIPI and TNI AL for providing funds to support this research.

References

[1] Elan Djaelani, "Perangkat Jamming Hasil Modifikasi Berbasis Direct Digital Synthesizser", Seminar Nasional Ilmu Pengetahuan Teknik Bandung, 28-29 November 2012.
[2] Purwoko Adhi, "Pembangkitan chrip untuk radar FMCW berbasis DDS", Jurnal Elektronika No.2, Vol.11, Juli Desember 2011.
[3] National Instruments, Understanding Direct Digital Synthesis (DDS)
[4] National Instrument, file:dds_32_14_bit_ex-1.vi, file dds 8bit.
[5] Elan Djaelani, "Simulasi Oscillator Blocking sebagai Sensor Level dengan Menggunakan LabView", Jurnal INKOM Vol 6, No 1, 2012, ISSN 1979-8059.

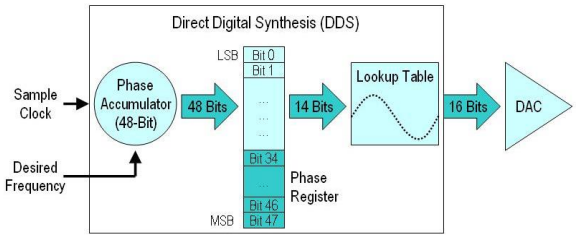


Figure 1. DDS architecture

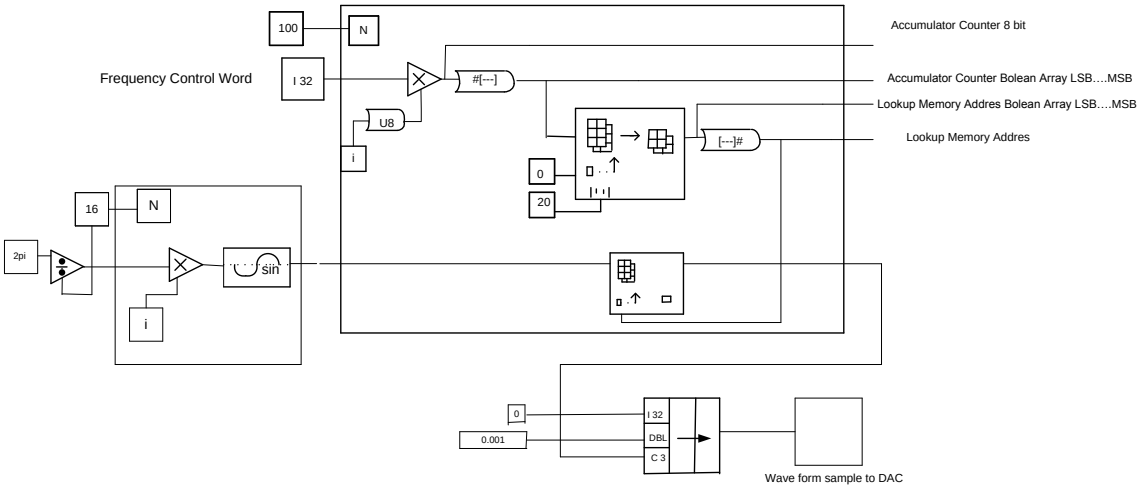


Figure 2. The block diagram of DDS simulation from National Instrument

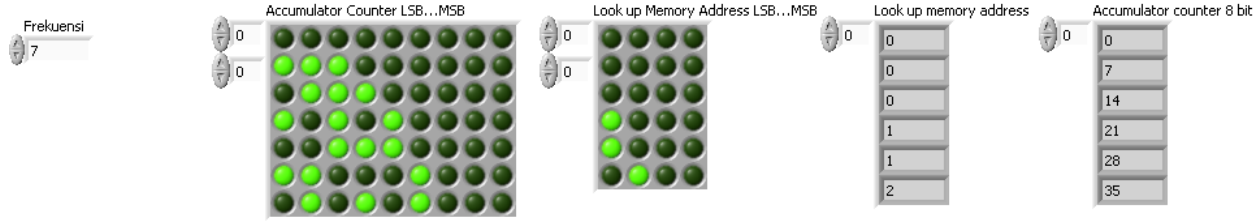


Figure 3. The front panel of DDS 8 bits

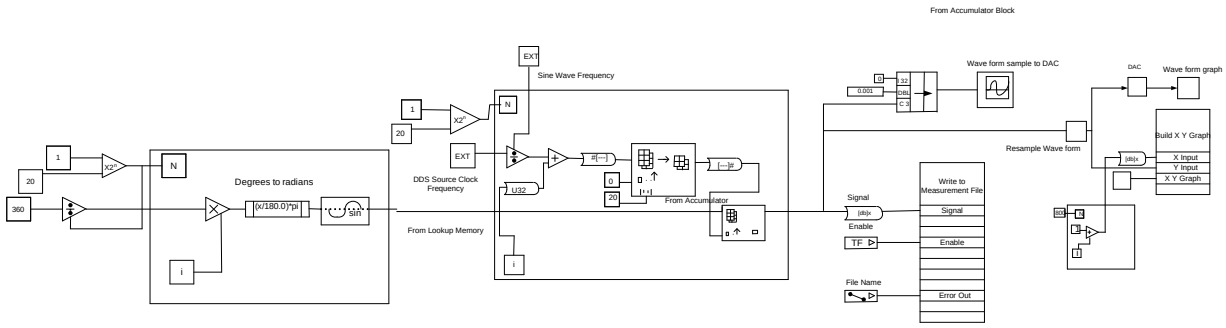


Figure 4. The block diagram of the proposed DDS simulation

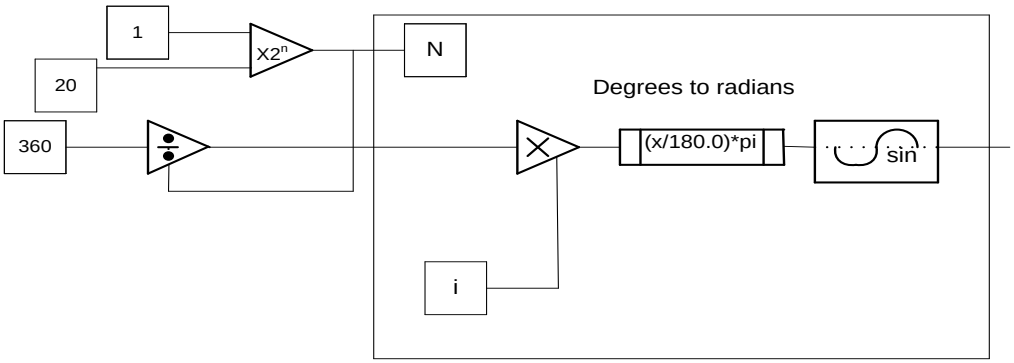


Figure 5 .Lookup Memory

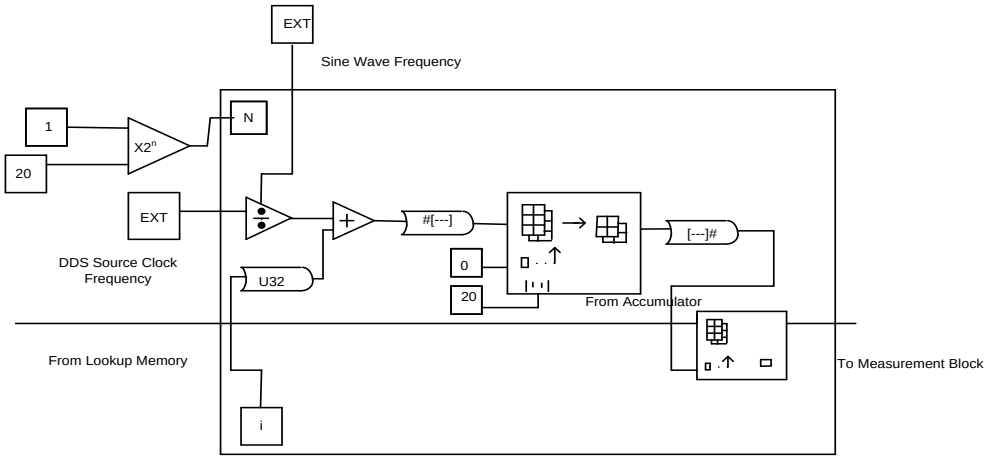


Figure 6. Accumulator

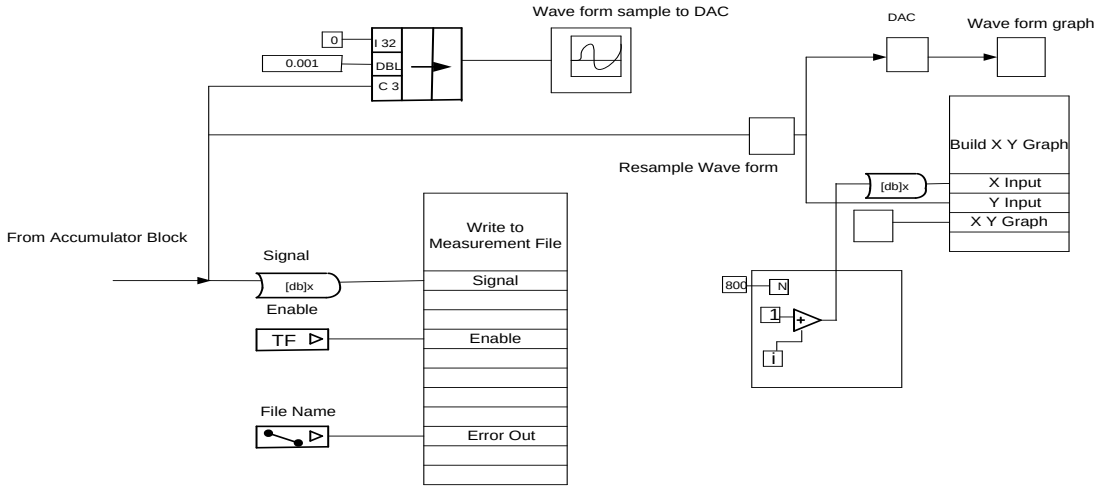
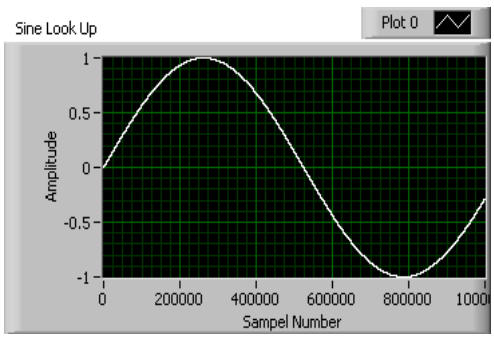
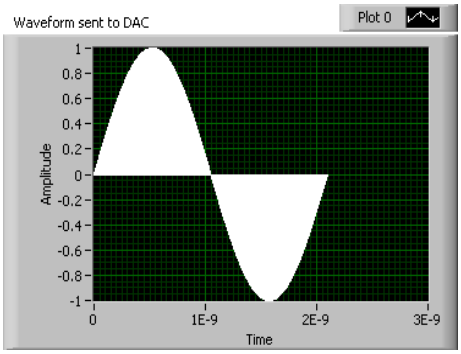


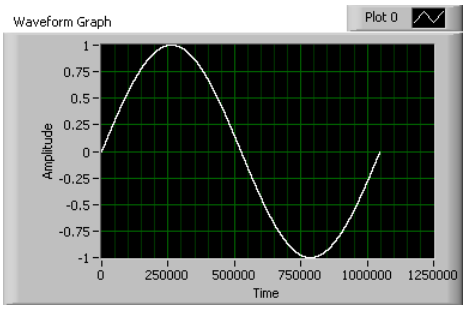
Figure 7. Measurement File



(a)



(b)



(c)

Figure 8. Front Panel (a) Lookup Memori, (b) Waveform of sample sent to DAC Waveform of sample sent to DAC, (c) Waveform of output DAC Waveform of output DAC

TABLE 1.
THE EXAMPLE OF THE RESULT

Sample	Data
0.000000	0.000000
1001.000000	0.190575
2001.000000	0.374164
3001.000000	0.544039
4001.000000	0.693971
5001.000000	0.818467
6001.000000	0.912962
7001.000000	0.973993
8001.000000	0.999322
9001.000000	0.988022
10001.000000	0.940506
11001.000000	0.858516
12001.000000	0.745058
13001.000000	0.604290
14001.000000	0.441371
15001.000000	0.262275
16001.000000	0.073565
17001.000000	-0.117842
18001.000000	-0.304929
19001.000000	-0.480839
20001.000000	-0.639124
21001.000000	-0.773983
22001.000000	-0.880471
23001.000000	-0.954686
24001.000000	-0.993907
25001.000000	-0.996697
26001.000000	-0.962953
27001.000000	-0.893913
28001.000000	-0.792107
29001.000000	-0.661266
30001.000000	-0.506187
31001.000000	-0.332553
31101.000000	-0.314410
32001.000000	-0.146730
32767.000000	0.000000