



# DESIGN AND ANALYSIS OF HYBRID WAVE PIPELINED PHASE ACCUMULATOR FOR DIRECT DIGITAL SYNTHESIZER

Menakadevi T.<sup>1</sup> and Madheswaran M.<sup>2</sup>

<sup>1</sup>Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamil Nadu, India

<sup>2</sup>Centre for Research and Development, Muthayammal Engineering College, Rasipuram, Namakkal, Tamil Nadu, India

E-Mail: [menaka\\_sar@rediffmail.com](mailto:menaka_sar@rediffmail.com)

## ABSTRACT

A Hybrid Wave Pipelining Phase Accumulator for Direct Digital Synthesizer is presented in this paper. Conventional Pipelining requires additional registers and it results in more area, power dissipation and clock routing complexity. Wave-pipelining does not have any of these disadvantages and it can be used to speed up the circuits without insertion of storage elements. The clock period in conventional pipeline scheme is proportional to the maximum delay while in hybrid wave-pipelining it is proportional to the maximum delay difference. To prove the better performance of the Hybrid wave pipelining Phase Accumulator DDS architecture compared it favorably with several existing DDS architecture. The focus of this paper is on design and analysis of hybrid wave pipelining Phase Accumulator DDS, using XilinxISE9.2i.

**Keywords:** hybrid wave pipeline, direct digital synthesis, FPGA, DDFS, numerical controlled oscillator, ASIC.

## 1. INTRODUCTION

Direct Digital Synthesis (DDS) is an electronic method for digitally creating arbitrary waveforms and frequencies from a single, fixed source frequency. With the development of VLSI technology and the requirement of modern communication systems, direct digital synthesizers have been widely used in Software Defined Radios and wireless transceivers. A DDS can achieve fast frequency switching in small frequency steps, over a wide band. In addition, it provides linear phase and frequency shifting with good spectral purity [1]. A DDS is used especially for a precise, high frequency and a phase tunable output. Traditional designs of high bandwidth frequency synthesizers employ the use of a phase locked-loop (PLL). DDS provides many significant advantages over the PLL approaches, such as fast settling time, sub-Hertz frequency resolution, continuous-phase switching response and low phase noise.

Several researchers have implemented DDS on FPGAs and ASICs [1, 2, 3] and [4]. However, these designs use more number of registers, multiplexers and their operating frequencies is up to 160 MHz and thus consume more power in FPGAs. In this paper, proposes a Hybrid wave pipelined Phase Accumulator along with ROMLUT for implementing DDS on FPGAs. Hybrid wave pipelined algorithm increases speed with minimum amount of register usage. The proposed Hybrid wave pipelined scheme modifies the wave pipeline scheme to achieve minimizes the area, improved power and performance gains [5, 6].

This hybrid wave pipeline architecture overcomes the low throughput problem in the previous implementations of the piecewise linear and angle rotation [7, 8, 9, 10]. First stage of DDS is Phase accumulator which is developed by hybrid wave pipelined scheme and this work is compared with various methods. This design greatly reduces the number of slices and increases the overall throughput.

The organization of this paper follows: In section 2, describes the overview of Direct Digital synthesizer. Section 3 gives the detail about Hybrid Wave Pipelining (HWP) technique and Section 4 is discussed about Implementation details and simulation results of Phase accumulator design using HWP. Conclusion is summarized in Section 5.

## 2. DDS OVERVIEW

The basic block diagram of a direct digital frequency synthesizer is shown in Figure-1 [11]. As shown in Figure-1, the main components of a DDS are a phase accumulator, phase-to-amplitude converter (a sine look-up table), a Digital-to-Analog Converter and filter. A DDS produces a sine wave at a given frequency. The frequency depends on three variables; the reference-clock frequency  $f_{clk}$  and the binary number programmed into the phase register (frequency control word,  $M$ ), length of  $n$ -bit accumulator. The binary number in the phase register provides the main input to the phase accumulator.

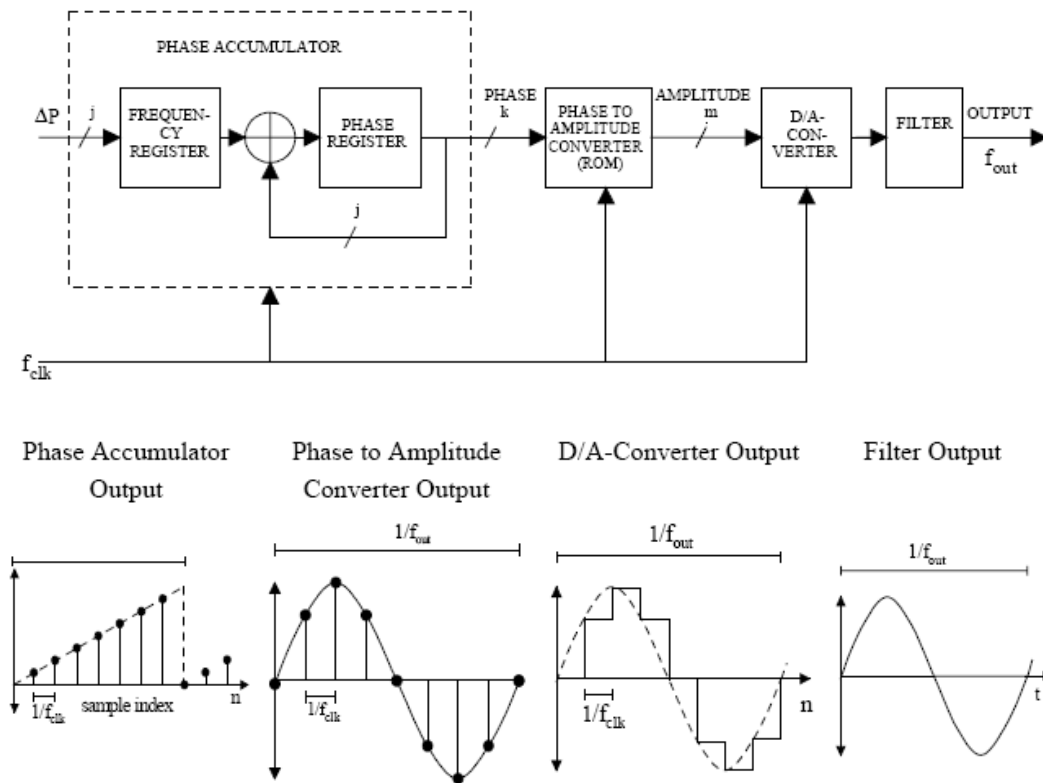


Figure-1. DDS function blocks and signal flow diagrams [11].

If a sine look-up table is used, the phase accumulator computes a phase (angle) address for the look-up table, which outputs the digital value of amplitude corresponding to the sine of that phase angle to the DAC. The DAC, in turn, converts that number to a corresponding value of analog voltage or current. To generate a fixed-frequency sine wave, a constant value (the phase increment—that is determined by the binary number  $M$ ) is added to the phase accumulator with each clock cycle. If the phase increment is large, the phase accumulator will step quickly through the sine look-up Table and thus generate a high frequency sine wave. If the phase increment is small, the phase accumulator will take many more steps, accordingly generating a slower waveform [2, 18]. The phase accumulator contents are updated once each clock cycle. Each time the PA is updated, the digital number or  $M$ , stored in the phase register is added to the number in the phase accumulator register. If the accumulator is 32-bits wide,  $2^n$  ( $n = 32$ ) clock cycles (over 4 billion) are required before the phase accumulator returns to 00...00, and the cycle repeats. The output of the phase accumulator serves as the address to a sine (or cosine) lookup table/ROM/phase-to-amplitude converter. Each address in the LUT corresponds to a phase point on the sine wave from  $0^\circ$  to  $360^\circ$ . The LUT contains the corresponding digital amplitude information for one complete cycle of a sine wave. The LUT, therefore, maps the phase information from the phase accumulator into a digital amplitude word, which in turn drives the DAC. For an  $n$ -bit phase accumulator ( $n$  generally ranges from 24 to

32 in most DDS systems), there are  $2^n$  possible phase points. The digital word in the phase register,  $M$  represents the amount the phase accumulator is incremented each clock cycle. If  $f_{clk}$  is the clock frequency, then the frequency of the output sine wave is equal to:

$$f_{out} = (M * f_{clk}) / 2^n \quad (1)$$

Above equation is known as the DDS "tuning equation." The frequency resolution of the system equals  $f_{clk} / 2^n$ . In a practical DDS system, all the bits out of the Phase Accumulator are not passed on to the LUT but are truncated, leaving only the first 13 to 15 MSBs. This reduces the size of the LUT and does not affect the frequency resolution. The phase truncation only adds a small but acceptable amount of phase noise to the final output [4].

## 2.1 Phase accumulator

The phase accumulator shown consists of a  $j$ -bit frequency register which stores a digital phase increment value  $\Delta p$  followed by a  $j$ -bit full adder and a phase register. The phase increment value is entered into the frequency register. The operation of the phase accumulator can be considered by looking at the phase advances around a circle as shown in Figure-2.

As the phase advances around the circle this corresponds to advances in the waveform, i.e., the greater the number corresponding to the phase, the greater the point is along the waveform [12]. By successively



advancing the number corresponding to the phase it is possible to move further along the waveform cycle.

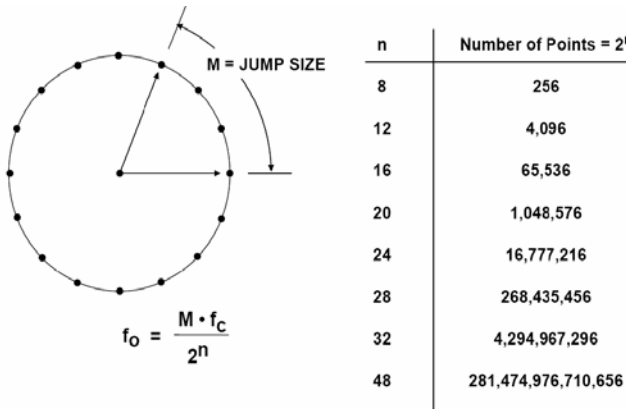


Figure-2. Digital phase wheel.

2.2. Phase to amplitude converter

Once the phase has been determined, it is necessary to convert this into a digital representation of the waveform. This is accomplished using a phase to waveform converter. This is a memory that stores a number corresponding to the voltage required for each value of phase on the waveform. The memory is either a read only memory (ROM) or programmable read only memory (PROM). This contains a vast number of points on the waveform. A very large number of points are required so that the phase accumulator can increment by a certain number of points to set the required frequency [4].

The output of the DDS is usually given to a DAC and then filtered to any unwanted signals. Phase quantization can be achieved in the DDS by truncating the phase information accumulator. Phase quantization is done to keep the memory requirements of the phase to waveform converter low. But it produces unwanted

spurious spectral components in the DDS output signals, know as spurs. Also sometimes images of the signals are generated on either side of the clock frequency and its multiples. These can be removed by low pass filtering [12].

3. HYBRID WAVE PIPELINE SCHEME

In this section, the proposed pipeline scheme called Hybrid Wave-Pipelining is introduced to compare with conventional pipeline system. Pipelining is a circuit design technique widely used in digital circuits in order to increase the clock rate. In the conventional pipelined circuit, a potential speed up of 'n' can be achieved when 'n' pipeline stages are used [13]. Each pipeline stage requires the insertion of register and global clock. The reciprocal of the maximum stage delay establishes the maximum pipeline clock rate.

Wave pipelining is a design technique for increasing the throughput of a digital system without introducing pipelining registers between adjacent combinational logic blocks. Rather, the signal propagation delay characteristics of the logic network are used at design time to manage the signal delays so as to ensure that operations do not interfere with their predecessor nor successor computations [14].

The concept of wave-pipelining has been described in a number of previous works [13, 15]. Figure-3 shows the simple pipelining scheme with combinational logic circuit surrounded by edge triggered input and output register. Figure-4 gives the associated timing diagram [13]. In Figure-4, the shaded regions bounded by the maximum and minimum delays through the logic (*D<sub>max</sub>* and *D<sub>min</sub>*) depict the flow of data through the combinational logic and the variations in the logic block with time.

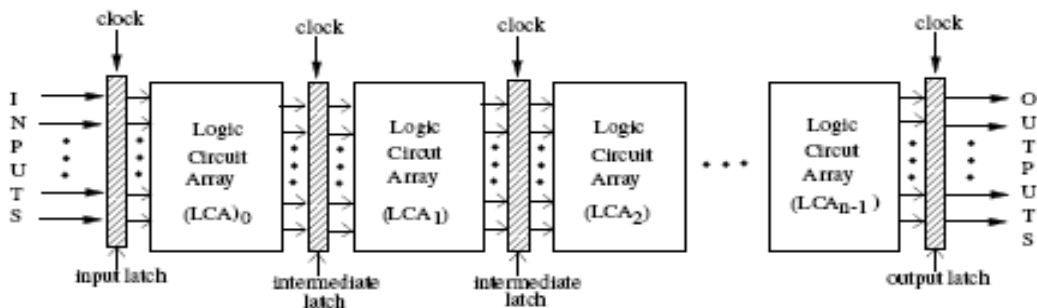


Figure-3. Simple pipelining scheme.

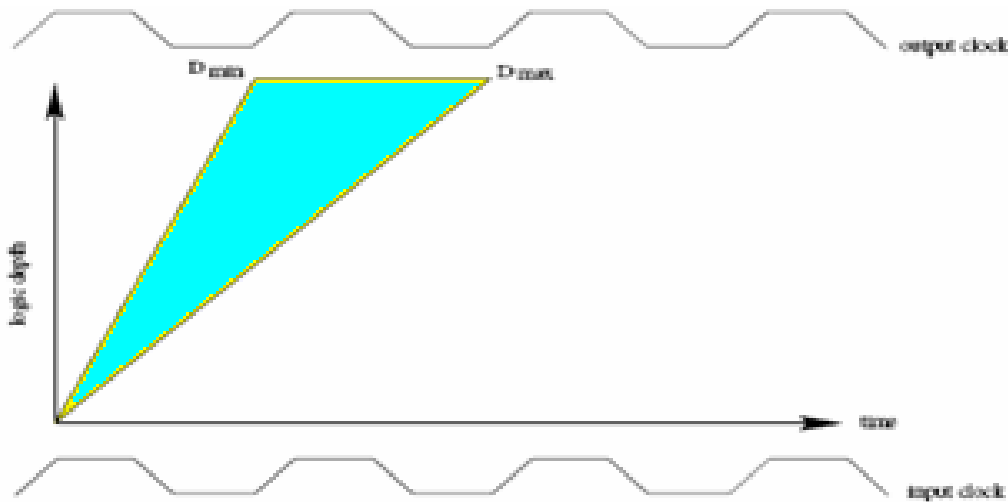


Figure-4. Delay difference to logic path.

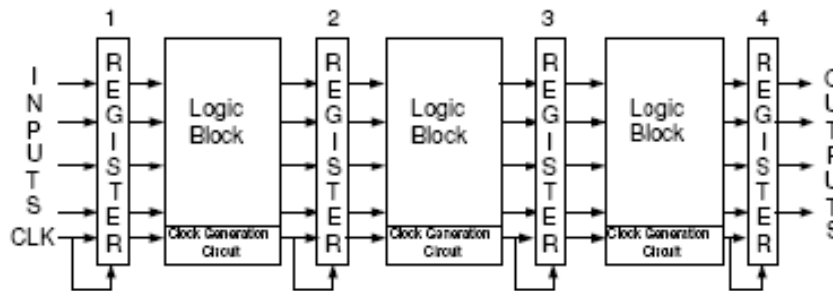


Figure-5. Hybrid wave pipelining scheme.

The proposed hybrid wave-pipeline scheme modifies the wave pipeline scheme [5, 6] to achieve improved power and performance gains. In this scheme the clock is also wave-pipelined as shown in Figure-5. The clock frequency is determined by the stage with the maximum delay difference. Contrary to these schemes similarities to regular pipeline scheme, it allows multiple data waves to exist in any stage similar to wave-pipelining. Higher clock frequencies are possible and

influence of clock uncertainties is mitigated. As can be seen, this scheme eliminates the need for complex clock distribution. Clock gating can be easily implemented to save power without affecting the pipelines performance. The temporal and spatial variation of the proposed hybrid wave-pipeline architecture is shown in Figure-6. The timing constraint analysis of this system can be done using Figure-6. In this scheme the clock is also wave pipelined and travels along with the data.

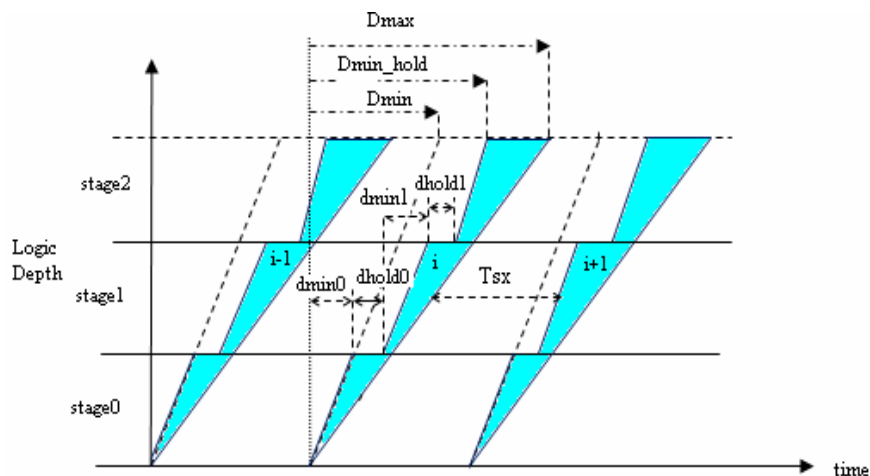


Figure-6. Temporal/spatial diagram of hybrid wave pipelining scheme [15].



The clock period  $T_{clk}$  of this system is determined by stage with the largest delay difference and safe time required before a new data vector/wave is admitted into this stage. The fundamental circuit limitations determine the safe time to separate any two adjacent data waves.

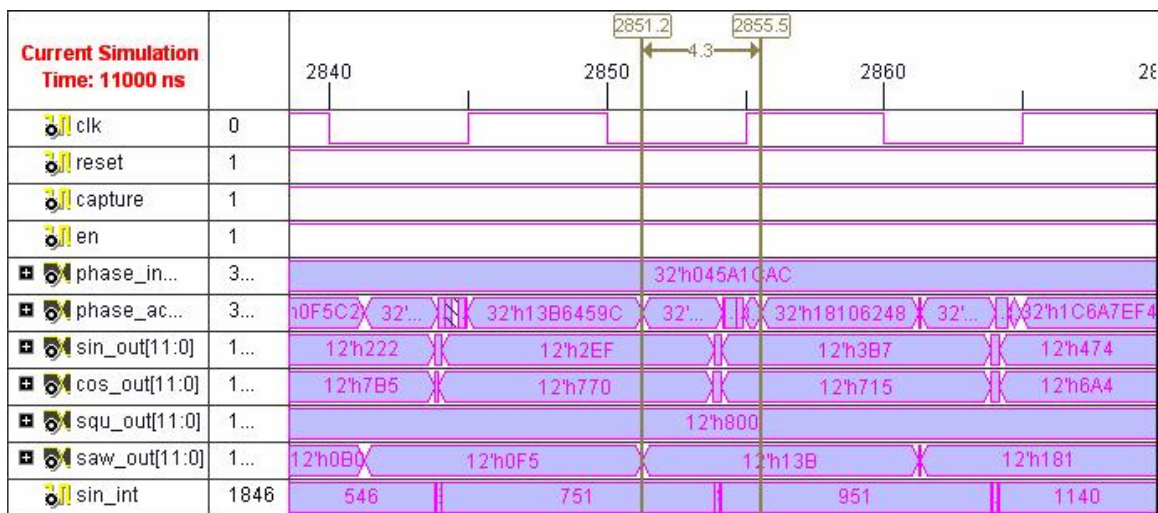
The smaller delay difference would result in a higher clock frequency. The delay difference can be minimized by balancing delay using buffers. Since each combinational logic stage in this scheme is wave-pipelined, the internal node constrains must also be considered so that any two adjacent data waves do not collide. This can be easily satisfied by designing the logic stages such that a stages delay difference is greater than the delay difference at any internal node in that stage or in other words the delay difference should monotonically increase from input to output of that stage [15].

#### 4. IMPLEMENTATION RESULTS AND DISCUSSIONS

Phase Accumulator is a heart of the DDS. The phase accumulator can be implemented in pipelined architecture, a pipelined version of accumulator can be implemented to achieve high throughput. A 32-bit phase accumulator has been chosen to be implemented in Hybrid wave-pipeline architecture as a proof of concept. The well-

known Pipelined Carry-look ahead Adder (CLA) technique has been used to implement the phase accumulator. The phase accumulator block is a 32 bit accumulator that is pipelined in 8 bit sections. VHDL code has been developed for 32 bit Pipelined CLA Phase accumulator and its functional and timing simulation results are verified by using Xilinx9.2i ISE Tool. For each stage maximum delay and minimum delay is identified by using timing simulation.

Hybrid wave-pipelining reduces the clock period by making the minimum delay ( $D_{min}$ ) at each stage of the system approach the maximum delay ( $D_{max}$ ). This in turn reduces the delay path difference and enables the reduction of  $T_{sx}$ , the separation between data waves at intermediate nodes. The results of these improvements have a bearing on the clock period. It can be made shorter and still enable data to propagate in its own wave. Hybrid Wave Pipelined concept is applied for Phase Accumulator. Using FPGA synthesis tool maximum and minimum path delays are identified. In this work  $D_{min}$  and  $D_{max}$  values are adjusted in the HWP PA architecture itself. Wherever the  $D_{max}$  and  $D_{min}$  difference is more it can be adjusted in two ways. (i) Delays are introduced to increase the  $D_{min}$  value. If  $D_{min}$  value is more the delay difference ( $D_{max}-D_{min}$ ) is less. (ii) Reducing the clock cycle period is another way to minimize the delay difference.



**Figure-7.** Timing simulation of conventional pipeline PA when  $f_{clk} = 10\text{GHz}$  and  $M = 045A1CAC$  and  $f_{out} = 1.7\text{GHz}$ .

In Conventional pipeline CLA, the propagation delay varies from 2851.2ns ( $D_{min}$ ) to 2855.5ns ( $D_{max}$ ) and the delay variation is 4.3ns as shown in Figure-7. Whereas in Hybrid Wave Pipelined Phase Accumulator, the propagation delay varied from 2843.2ns ( $D_{min}$ ) to 2845.8ns ( $D_{max}$ ) resulting in a delay variation of 2.6ns. The time period  $T_{clk}$  is mainly due to  $D_{max}$ . It is shown in Figure-8. It is clear that delay variation is minimized and  $T_{clk}$  depends on these delay difference. The functional

and timing simulation for HWP phase accumulator is carried out and delay values are adjusted. The work shows that the critical design issues when dealing with Hybrid Wave-Pipelining include; balancing delay paths, developing an appropriate clock delay scheme and reducing the overhead and latency associated with intermediate latches. The Hybrid Wave-Pipelined PA performs 60.4% better than the conventional pipelined PA.

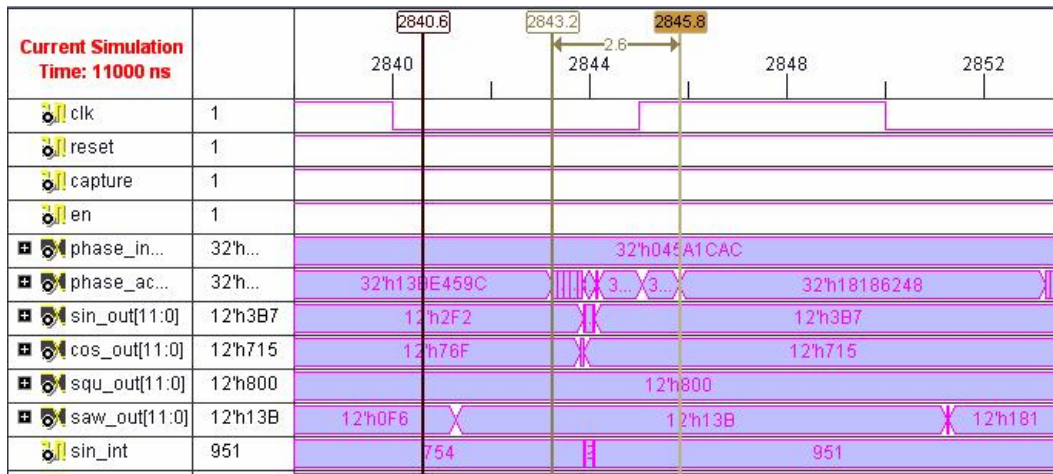


Figure-8. Timing simulation of HWP PA when fclk = 10GHz and M = 045A1CAC and fout = 1.7GHz.

With every clock pulse the contents of the frequency register (FCW) is added to that of PA. The PA generates the phase values of the output sine wave. The output of the PA serves as the address of the LUT. Each time the PA overflows, the LUT outputs sampled values of the sine wave. This output of the LUT represents one cycle of the sine waveform, since the LUT contains sampled

values of one cycle of the sine wave. The overflow rate of the PA depends on the bit-size of the PA (number of bits) and the frequency tuning word (FCW). Larger the size of the frequency tuning word faster the PA overflows. Simulations are done with input bits of 32 bits and its frequency resolution 0.0232Hz.

Device Utilization Summary				
Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Flip Flops	45	7,168	1%	
Number of 4 input LUTs	32	7,168	1%	
<b>Logic Distribution</b>				
Number of occupied Slices	23	3,584	1%	
Number of Slices containing only related logic	23	23	100%	
Number of Slices containing unrelated logic	0	23	0%	
<b>Total Number of 4 input LUTs</b>	<b>32</b>	<b>7,168</b>	<b>1%</b>	
Number of bonded IOBs	115	141	81%	
IOB Flip Flops	63			
Number of Block RAMs	6	16	37%	
Number of GCLKs	1	8	12%	
<b>Total equivalent gate count for design</b>	<b>394,461</b>			
Additional JTAG gate count for IOBs	5,520			

Figure-9. Design summary of conventional (Non-Pipelined PA) DDS.

The width of Sine ROMLUT is 12 bits and this block produces two outputs such as sine and cosine waveform. The number of slices, Look up Tables (LUTs) and registers are calculated for three designs: Conventional (Non-pipelined PA) DDS, Pipelined PA DDS and Hybrid Wave Pipelined (HWP) PA DDS. In fact, the proposed method uses 33 slice elements and 64 LUTs against 125 slices and 194LUTs for the pipelined method. Similarly Conventional DDS uses 45 slices and 32 LUTs. The slices and LUTs are minimum in the proposed method. Compare to other techniques the speed is also increased and its maximum frequency is 194.45MHz.

Synthesis results are summarized in Table-1. Compare to pipeline scheme the proposed scheme minimizes the slice count by the factor of 3.78-2.77 and reduces the four input LUTs by the factor of 6.06-3.03. The speed grade of proposed method is increased at 22% from pipeline scheme and 38% improvement with conventional method. The design summary of Conventional DDS, Pipelined PA DDS and Hybrid Wave Pipelined DDS using Spartan III FPGA is shown in Figures 9, 10 and 11. The frequency resolution is 0.023Hz and the SFDR is calculated to be 112dB.



www.arnpjournals.com

Device Utilization Summary				
Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Flip Flops	125	7,168	1%	
Number of 4 input LUTs	194	7,168	2%	
<b>Logic Distribution</b>				
Number of occupied Slices	135	3,584	3%	
Number of Slices containing only related logic	135	135	100%	
Number of Slices containing unrelated logic	0	135	0%	
<b>Total Number of 4 input LUTs</b>	<b>194</b>	<b>7,168</b>	<b>2%</b>	
Number of bonded IOBs	60	141	42%	
IOB Flip Flops	50			
Number of GCLKs	4	8	50%	
<b>Total equivalent gate count for design</b>	<b>2,654</b>			
Additional JTAG gate count for IOBs	2,880			

Figure-10. Design summary of pipelined PA DDS.

Device Utilization Summary				
Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Flip Flops	33	7,168	1%	
Number of 4 input LUTs	64	7,168	1%	
<b>Logic Distribution</b>				
Number of occupied Slices	33	3,584	1%	
Number of Slices containing only related logic	33	33	100%	
Number of Slices containing unrelated logic	0	33	0%	
<b>Total Number of 4 input LUTs</b>	<b>64</b>	<b>7,168</b>	<b>1%</b>	
Number of bonded IOBs	115	141	81%	
IOB Flip Flops	43			
Number of Block RAMs	6	16	37%	
Number of GCLKs	1	8	12%	
<b>Total equivalent gate count for design</b>	<b>394,397</b>			
Additional JTAG gate count for IOBs	5,520			

Figure-11. Design summary of HWP PA DDS.

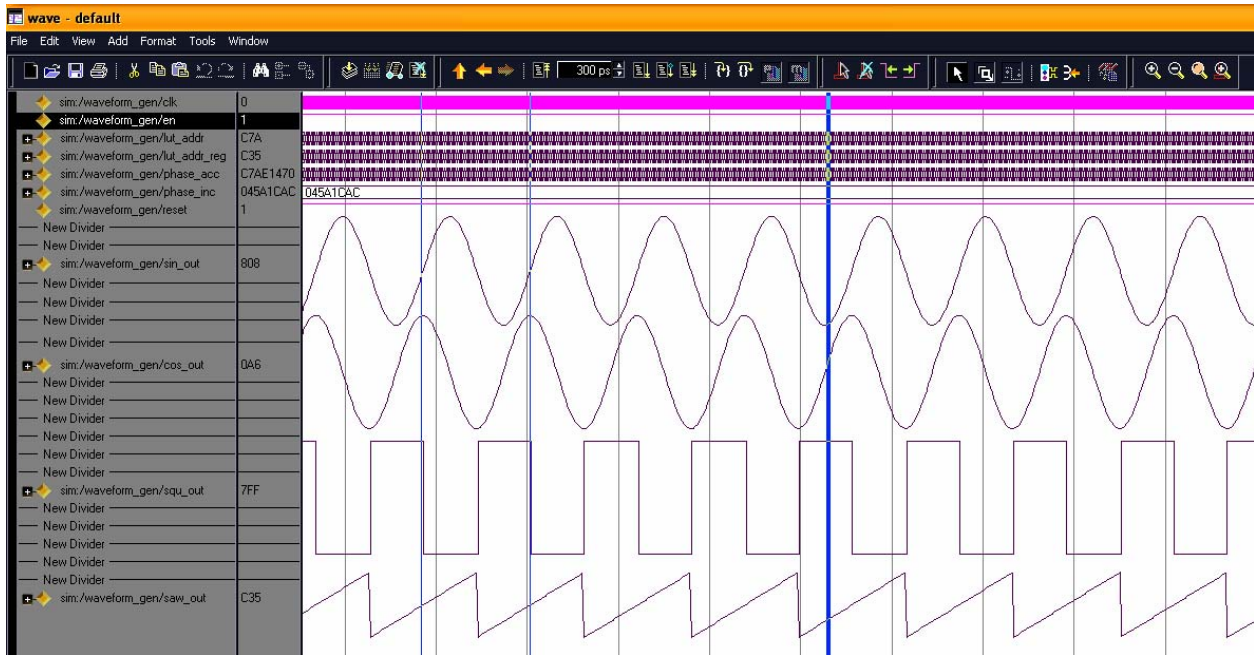
Table-1. Speed and area performance.

Technique	No. of slices used	No. of 4 input LUTs	No. of IOB registers	Maximum frequency (MHz)
Conventional DDS	45	32	63	122.63 MHz
Pipelined PA DDS	125	194	50	152.374MHz
Hybrid wave pipelined PA DDS	33	64	43	194.445MHz

The proposed DDS architecture is worked with some MHz reference frequency to little GHz reference frequency. For that two reference frequencies taken in this work for simulation. In the first case to generate an output frequency of 1.7MHz with a reference clock frequency of 100MHz, a frequency tuning word ( $M$ ) of 045A1CACH is stored in the frequency Register. The value of the

frequency tuning word ( $M$ ) is calculated using the frequency tuning equation in Equation 2. The Phase Accumulator is 32-bits wide. This control word  $M$  is added to the previous value of PA with each clock pulse.

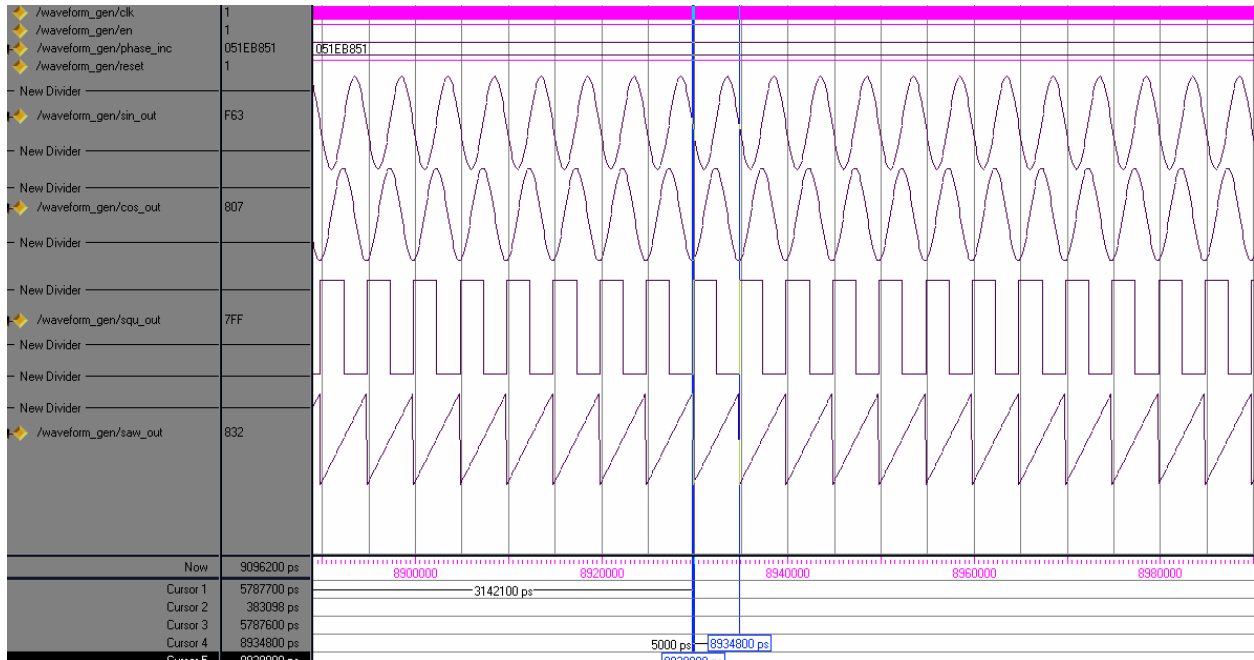
$$\text{Frequency Control word } (M) = (2^n * f_{out}) / f_{clk} \quad (2)$$



**Figure-12.** DDS output fclk = 100MHz, FCW = 045A1CAC, fout = 1.7MHz using modelsim6.2g.

From the second case input reference clock frequency is 10GHz. In order to produce 200MHz output frequency the frequency tuning word of 051EB851H is

applied in to frequency register. Figure-13 shows the ModelSim6.2g simulation result of 200MHz output frequency with 10GHz reference clock.



**Figure-13.** DDS output, fclk = 10GHz, FCW = 051EB851H, fout = 200MHz.



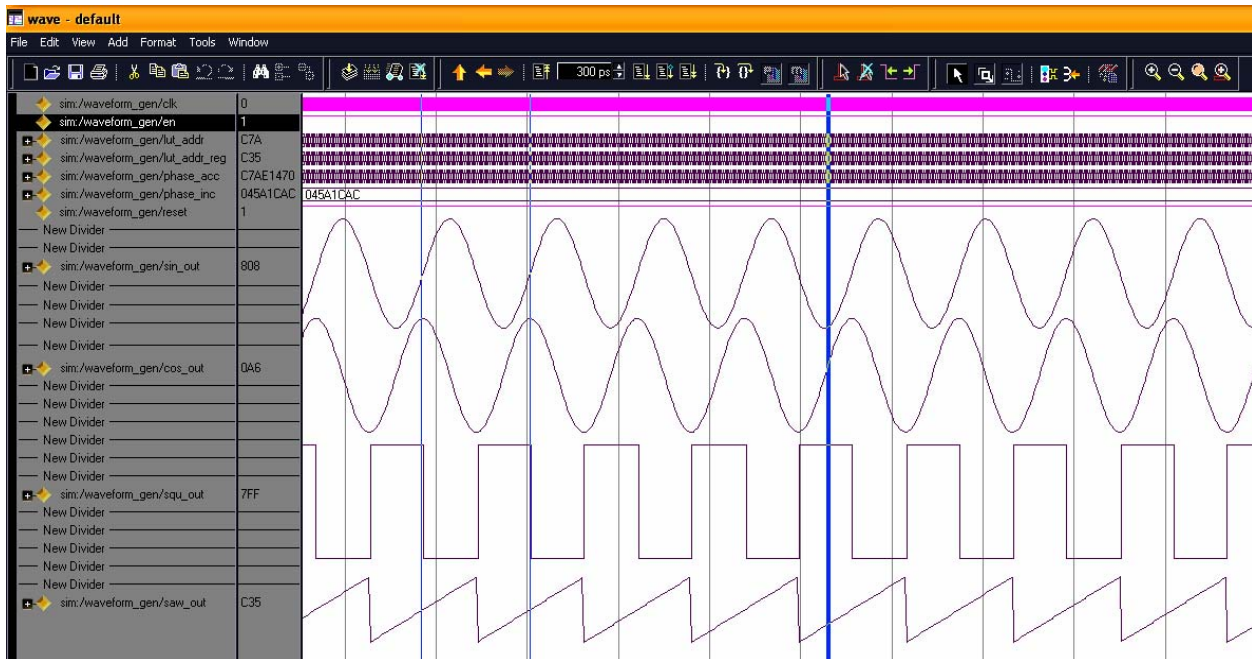


Figure-14. DDS output fclk = 100MHz, FCW = 045A1CAC<sub>h</sub>, fout = 1.7MHz using ModelSim6.2g.

The 32 bit adder output is truncated into 12 bits [16]; only higher order bits are taken. The phase value will not be affected if the truncation is carried out because the phase information is only available in higher order bits. This module produces phase register output which is 12 bit wide along with sine and cosine output. These outputs are input of the Sine ROMLUT module.

This simulation, performed with a reference clock of 100MHz. Finally, to enhance the DDS survey,

comparisons with some previous works are presented in Table-2. In this table, it is shown that the types of targets are used for implementation and techniques used. For FPGA implementation, the proposed architecture gives high spectral performance (SFDR = 112 dBc) and high operating frequency at 194.45MHz.

Table-2. Comparison of DDS Performance.

References	Target	Technique	Input bits	Maximum frequency (MHz)	SFDR (dBc)
JYOTHI [8]	FPGA	Slope piecewise	32	25	89
WANG [17]	FPGA	Tailor series corrected	32	61.44	105
MORAN [9]	FPGA	Modulus Change	28	160	110
Proposed work	FPGA	Hybrid Wave pipelined with ROM LUT	32	194.45	112

### 5. CONCLUSIONS

This proposed method takes advantage of Hybrid Wave Pipelined scheme which uses a minimum number of registers with high throughput and decreases the whole system latency. This method uses a large phase accumulator to perform a high frequency resolution. The architecture has been compared with the prior art architectures in simulations and measurements. The simulated and measured results demonstrated that the maximum frequency is 194.45MHz, frequency resolution of 0.0232Hz, Phase resolution 0.088° and spurious

performance is improved at 112dBc. This proposed design provides better SFDR, high maximum frequency and minimum resource utilization compare to existing DDS architectures.

### REFERENCES

[1] J. Tierney, C. M. Radre and B. Gold. 1971. A Digital Frequency Synthesizer. IEEE Transactions on Audio and Electroacoustics. March.



- [2] 2008. A Technical Tutorial on Digital Signal Synthesis, Online Available WWW: [http://www.analog.com/UploadedFiles/Tutorials/450968421DDS\\_Tutorial\\_rev12-2-99](http://www.analog.com/UploadedFiles/Tutorials/450968421DDS_Tutorial_rev12-2-99)
- [3] M. I. Bellaouar, A., O'brecht M. S., Fahim A. M., and Elmasry. 2000. Low-power direct digital frequency synthesis for wireless communications. *IEEE Journal of Solid-State Circuits*. 35(3): 385-390.
- [4] Nicholas H. T. and Samueli H. An analysis of output spectrum of direct digital frequency synthesizer in the presence of phase accumulator truncation. *Proceedings of the 41<sup>st</sup> IEEE Annual Frequency Control Symposium*. pp. 495-502.
- [5] C. T. Gray, W. Liu and R. K. Cavin 1994. Timing Constraints for Wave-pipelined Systems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. 13(8): 987-1004.
- [6] W. P. Burleson, M. Ciesielski, F. Klass and W. Liu. 1998. Wave-Pipelining: A Tutorial and Research Survey. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*. 6(3): 464-474.
- [7] A. M. Sodagar and G. R. Lahiji. 2001. A pipelined ROM-less architecture for sine-output direct digital frequency synthesizers using the second-order parabolic approximation. *IEEE Transaction on Circuits and Systems II*. 48(9): 850-857.
- [8] L. S. Jyothi, M. Ghosh, F. F. Dai and R. C. Jaeger. 2006. A novel DDS using nonlinear ROM addressing with improved compression ratio and quantization noise. *IEEE Transactions on Ultrasonic Ferroelectrics and Frequency Control*. 53(2): 274-283.
- [9] D. R. Moran, J. G. Menoyo and J. L. Martin. 2006. Digital Frequency Synthesizer Based on Two Coprime Moduli DD. *IEEE Transaction on Circuit and Systems II*. 53(12): 1388-1392.
- [10] D. De Caro and A. G. M. Strollo. 2005. High-Performance Direct Digital Frequency Synthesizers Using Piecewise-Polynomial Approximation. *IEEE Trans. on Circuit and Syst. Part- I*. 52: 324-336.
- [11] 2001. *Direct Digital Synthesizers: Theory, Design and Applications*. Jouko Vankka Boston; London. Kluwer Academic Publishers.
- [12] Sreerama Reddy G. M. and P. Chandrashekara Reddy. 2009. Design and FPGA Implementation of High Speed, Low Power Digital up Converter for Power Line Communication Systems. *European Journal of Scientific Research*. 25(2): 234-249.
- [13] K. K. Parhi. 1999. *VLSI Signal Processing Systems*. John Wiley and Sons, New York, NY, USA.
- [14] G. Seetharaman, B. Venkataramani and G. Lakshminarayanan. 2008. VLSI Implementation of Hybrid Wave-Pipelined 2D DWT Using Lifting Scheme. *Journal of VLSI Design*. Hindawi Publishing Corporation. 2008 (Article ID 512746): 8.
- [15] W. P. Burleson, M. Ciesielski, F. Klass and W. Liu. 1998. Wave-pipelining: a tutorial and research survey. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*. 6(3): 464-474.
- [16] Maher Jridi and Ayman Alfalou. 2009. Direct Digital Frequency Synthesizer with CORDIC Algorithm and Taylor Series Approximation for Digital Receivers. *European Journal of Scientific Research*. 30(4): 542-553.
- [17] W. Wang, Z. Yifang and Y. Yang. 2006. Efficient wireless Digital up Converters design using system generator. *International conference on Signal Processing, ICSP*. pp. 443-446.
- [18] A. Ashrafi and R. Adhami. 2005. A Direct Digital Frequency Synthesizer Utilizing Quasi-Linear Interpolation Method. *Proc. of IEEE 37<sup>th</sup> Southeastern Symposium on System Theory*. March. pp. 114-118.