Novel method to implement high frequency All Digital Phase-Locked Loop on FPGA

Dr Anu Gupta ,
Faculty, B.I.T.S Pilani,

Mohammad Waqar Ahamed,
Student B.I.T.S Pilani,

Abhishek Dhir ,
Student B.I.T.S Pilani,

Ravish Soni,
Student B.I.T.S Pilani,

Neeraj Kumar Sharma,
Student B.I.T.S Pilani

ABSTRACT

A programmable FPGA based implementation of a high frequency All Digital Phase Locked Loop (ADPLL) based clock generator is presented. The novelty of the design lies in its pipelined loop filter for improving the maximum tracked output frequency upto 70 MHz. The whole implementation of ADPLL consumes very low dynamic power of 32mW at highest frequency. The digital controlled oscillator (DCO) generates a clock signal with high frequency. The presented ADPLL has fast acquisition and large pull in range for output frequencies ranging from 10Mhz to 70 Mhz. Loop filter is designed to support high speed operation. The whole design including DCO has been done in synthesizable Verilog. It does not contain any library specific cells. The presented design has been implemented in a xc3s400a-4fg320 Xylinx Spartan FPGA. The maximum lock in time for the ADPLL is 39 clock cycles.

1. INTRODUCTION

The phase locked loop helps keep parts of our world orderly. A phase-locked loop (PLL) is a circuit synchronising the output signal of a controllable oscillator with a reference signal in frequency as well as in phase [1]. PLL can act as a frequency multiplier in a feedback loop as shown in figure 1.

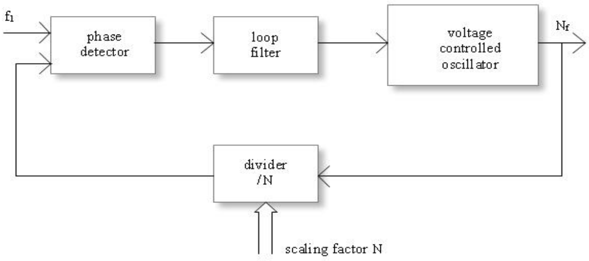

Fig.1. Frequency synthesizer based on PLL.

Designing in a Hardware Description Language (HDL) enables fast simulation and high level aspects can be easily tested and changed. Since no technology dependant constructs have been used in design, the presented design can be synthesized on a target ASIC with small changes in the code. The flexibility to generate multiple frequency clock signals from a reference clock can help save design effort and reduce design cost.

The architecture implemented in this paper is explained in section II.

2. IMPLEMENTED ARCHITECTURE OF ADPLL

a. *Phase Frequency Detector (PFD)*

Phase detector generates error signals depending upon the phase error in locked state and frequency error in unlocked state. We have used a synchronous version of PFD presented in [1]. This synchronous PFD can be easily implemented on an FPGA as all signals are synchronized to a clock.

b. *Loop Filter*

Loop filter takes error signals from PFD as input, these signals are processed to generate control signals for dco. Loop filter effectively performs the following calculations once on each cycle of dco clock period for duration of mismatch.

$$l = l_{prsv} \pm l_{prsv} \times \frac{R}{S} \qquad (1)$$

Decision about addition or subtraction is taken on the basis of error, i.e up or down. The implemented loopfilter is presented in figure 4.

This implementation is designed for low hardware requirements and high frequency of operation. Fixed point arithmetic is used to accommodate fractional values. The register error denoted in figure as 'Error_reg' is included to enhance maximum frequency of operation. Including this register requires change in generated control signals denoted in figure as 'clear' and 'update' so that equation 1 is calculated for equal number of clkdco cycles for which phase error exist. The required change is that update and clear signals must be delayed by one clkdco cycle which has been represented in figure 2 and figure 3 respectively.

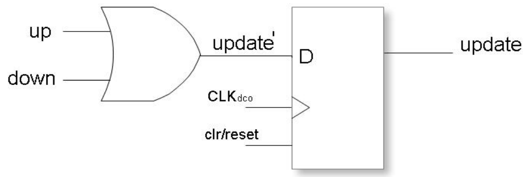

Fig.2. Generation of update signal

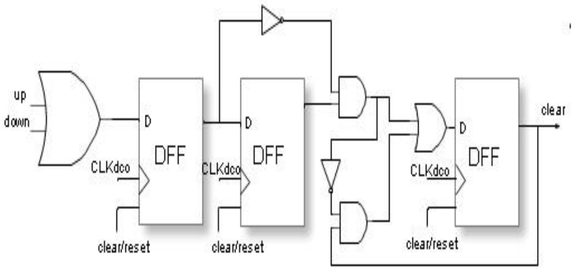

Fig.3. Generation of clear signal

To simplify hardware further we use the assumption that f_{clkdco}>>f_{ref} so delaying update and clear signals will not affect performance by large amount; we observed the same in our results.

$$\geq\ T_{c-q} + T_{combinational} + T_{setup} \qquad (2)$$

Since Tc,q and Tsetup are fixed so to decrease the required clock time period we need to have low Tcombinational to be able to operate our system at high frequency. For this reason we have implemented multiplier architecture with partial product generation with 'Radix-4 modified Booth recoding', partial product accumulation with 'Wallace tree' and Final stage addition with 'Brent-Kung adder'. Implemented adder-subtractor architecture is Kogge-Stone adder. We were able to achieve maximum operating frequency upto 71 MHz for our design.

c. DCO

We have implemented dco proposed in [2] which comprises of two parts. A ring oscillator with variable delay as shown in figure 5. The second part consists of control unit that constantly switches number of delay elements in the ring between two adjacent integral values to obtain fractional number of effective delay elements. This implementation has high frequency resolution. Switching between two adjacent values causes maximum peak to peak jitter of 2τde [2]. Complete architecture of DCO is shown in figure 6.

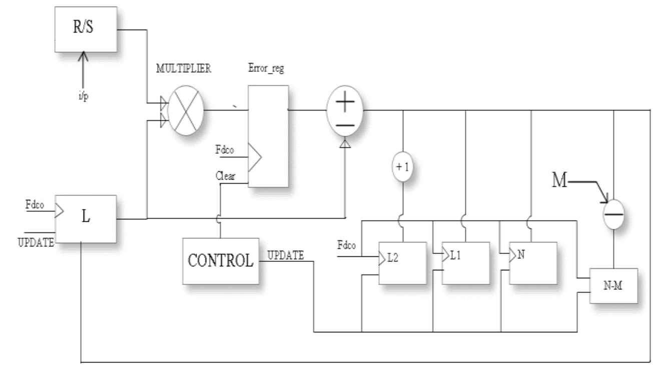

Fig.4. Loop Filter

d. Fractional Frequency Divider (FFD)

For high frequency resolution and flexibility the feedback of DCO has been implemented using a reprogrammable fractional frequency divider presented in [2]. Scaling factor of this accumulator switches periodically between two adjacent integers realizing a fractional scaling factor on average. Its implementation is shown in figure 6. The Most Significant Bit (MSB) of register shown in figure 6 is the output of FFD. The MSB also controls selection of N or N-M values. If MSB is zero, negative value N-M is selected. If MSB is one, the positive value N is selected. On average MSB bit is high for N out of M cycles of clkdco. The register N and N-M are made reprogrammable using a separate clock signal 'reprogram'.The values of these registers can be varied so as to attain different clkdco frequencies for different ref frequencies as per requirement.

The complete architecture of implemented ADPLL is shown in figure 8.

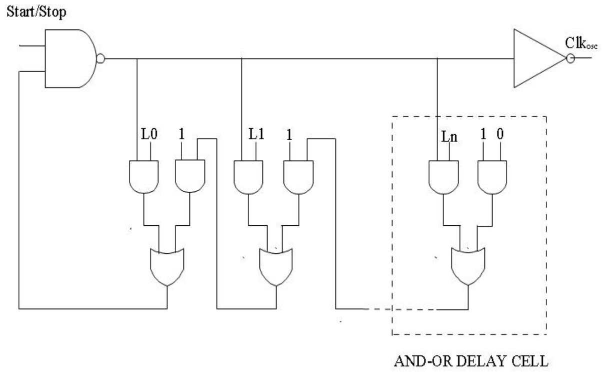

Fig.5. Ring oscillator with variable and-or delay matrix

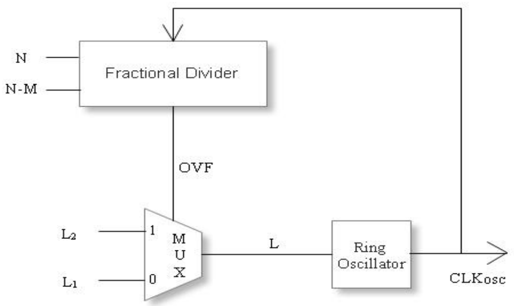

Fig.6. Block diagram of DCO

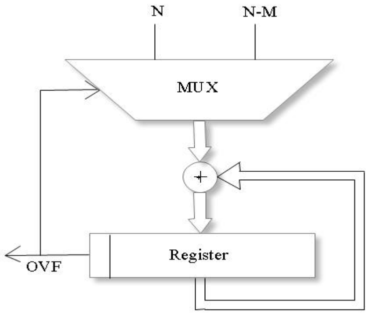

Fig.7. Fractional Frequency Divider

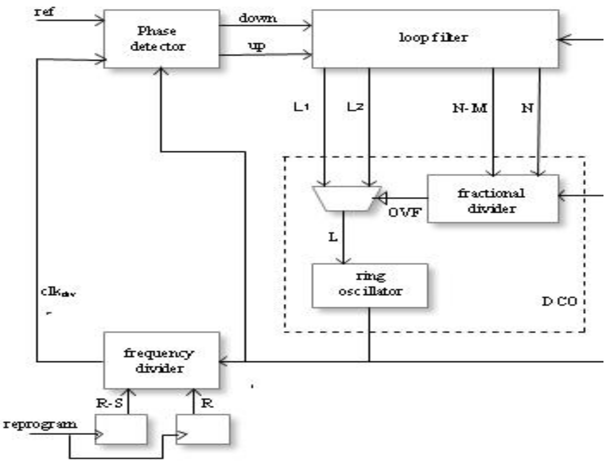

Fig.8. Complete Architecture of ADPLL

3.RESULTS

Table.1. Implementation results and results reported in [1]

	Technology	Slices	Maximum tracked output frequency	Maxi-mum Lock Time
Implemented Design	XC3S400A Xylinx FPGA	392	70 MHz	39 cycles
Design presented in [1]	V400BG432 Xylinx FPGA	672	40 MHz	30 cycles

Table 1 shows improvement in maximum tracked output frequency due to the changes made in loop filter. The maximum number of reference cycles to lock output signal is more than reported in [1]. We have obtained lock time of as low as 4 cycles for some reference frequencies, lock time results may improve by selecting a different free running frequency of DCO and division factor of frequency divider. In all calculations input and reference signals were considered lock if the phase error for 5 consequent cycles is less than π/4 and output follows reference thereafter.

Figure 9 indicates higher clock jitter at lower frequencies, with maximum jitter of 21% near 1100 KHz reference frequency when frequency division factor of feedback frequency divider is configured as 16. The percentage jitter flattens to 16% as we move to higher frequencies.

Figure 10 indicates that lock time (in microseconds) is maximum near 1000 KHz which is very near to frequency of maximum jitter.

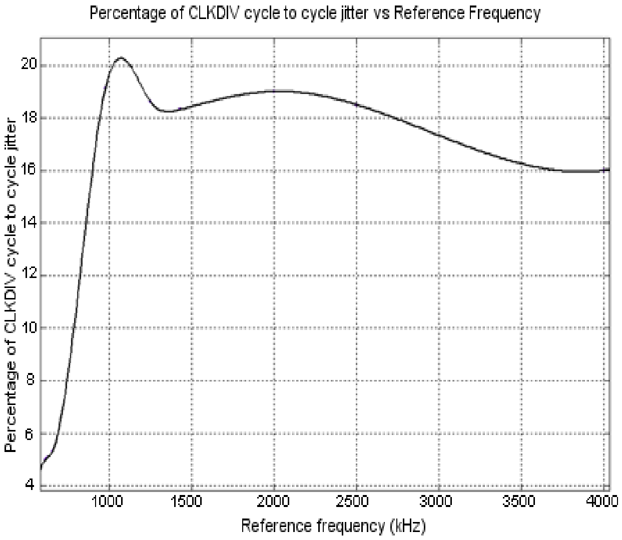

Fig.9. CLKDIV cycle to cycle jitter vs Reference Frequency

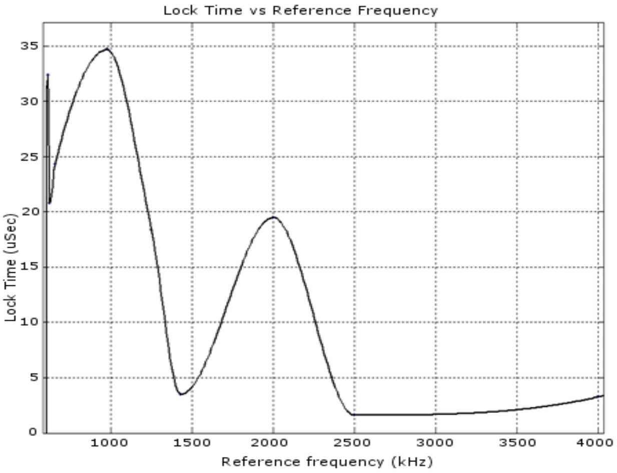

Fig.10. Lock Time vs Reference Frequency

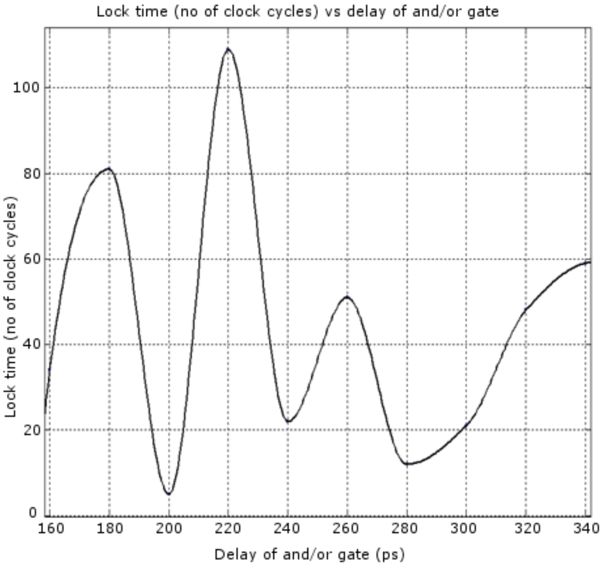

Fig.11. Lock Time in Cycles vs and/or gate delay of delay matrix for clkdco= 16.384 MHz

Figure 11 was plotted for a fixed reference frequency of 1024 KHz and changing the delays of and/or gate. The results indicate that locking time can change widely with temperature as and or gate delays change. Using a coarse lock circuit may help to improve performance under variation of and/or gate delay.

For figure 9-11 the values were measured at random intervals and plots were interpolated.

Table.2. Quiescent and Dynamic power of circuit at different frequencies

DCO Frequency	50 MHz	70 MHz
Total Quiescent Power	0.050 W	
Total Dynamic Power	0.023 W	0.032 W

Figure 12 shows CLKDIV signal being tracked to Reference signals. The DCO frequency is 16 times CLKDIV frequency.

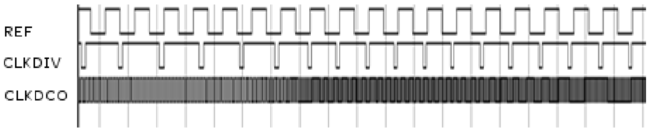

Fig.12. Waveform of CLKDIV signal being tracked to Reference signal

4. CONCLUSION

A programmable high frequency ADPLL based clock synthesizer was successfully synthesized on FPGA. Very few papers are available on FPGA based implementation of ADPLL. The implemented design supports 70 MHz tracked output frequency (clkdco) on FPGA which is more than all papers known to author. The novelty of implemented design is the pipelined loop filter. The proposed design can easily be adapted for different processes without the need to redesign any of its components since it does not use any technology specific construct. Dynamic power dissipated by the circuit is 32 mW at 70 MHz. It has fast acquisition and large pull in range which can be varied by changing the programmable feedback division factor of our design. The whole design including dco has been implemented in synthesizable verilog. It has been successfully implemented in xc3s400a-4fg320 Xylinx Spartan FPGA. Maximum lock in time is 39 cycles which may be improved further. The locking time is very sensitive to variations in temperature. The maximum tracked output frequency is 70 MHz.

5. REFERENCES

[1] Ronald Best, Phase-Locked Loops, McGraw-Hill,USA, 6th Edition.

[2] Stefo, R.; Schreiter, J.; Schlussler, J.-U.; Schuffny, R.; , "High resolution ADPLL frequency synthesizer for FPGA- and ASIC-based applications," *Field-Programmable Technology (FPT), 2003. Proceedings. 2003 IEEE International Conference on* , vol., no., pp. 28- 34, 15-17 Dec. 2003

[3] Walters, S.M.; Troudet, T.; , "Digital phase-locked loop with jitter bounded," *Circuits and Systems, IEEE Transactions on* , vol.36, no.7, pp.980-987, Jul 1989

[4] Tarar, M.A.; Sun, J.; Sampson, A.; Wilcox, R.; Zhizhang Chen; , "The Implementation of a New All-Digital Phase-Locked Loop on an FPGA and Its Testing in a Complete Wireless Transceiver Architecture,"*Communication Networks and Services Research Conference, 2009. CNSR '09. Seventh Annual* , vol., no., pp.238-244, 11-13 May 2009

[5] Kumm, M.; Klingbeil, H.; Zipf, P.; , "An FPGA-Based Linear All-Digital Phase-Locked Loop," *Circuits and Systems I: Regular Papers, IEEE Transactions on* , vol.57, no.9, pp.2487-2497, Sept. 2010

[6] Qiang Zhang; Kai Huang; Zuojun Liu; Zhigang Li; , "Research and Application of All Digital Phase-Locked Loop," *Intelligent Networks and Intelligent Systems, 2009. ICINIS '09. Second International Conference on* , vol., no., pp.122-125, 1-3 Nov. 2009