A Low Power VLSI Design of an All Digital Phase Locked Loop

Nakkina Vydehi [1], A. S. Srinivasa Rao [2]

[1] *M. Tech, VLSI Design, Department of ECE,*
[2] *M.Tech, Ph.D, Professor, Department of ECE,*
[1,2] *Aditya Institute of Technology and Management, Tekkali, Srikakulam, INDIA*

***Abstract*—Phase locked loop is a familiar circuit for high frequency application and very short interlocking time. In this paper we have implemented and analysed All Digital Phase locked loop (ADPLL), as the present applications requires a low cost , low power and high speed Phase locked loops. The design is synthesized in Xilinx ISE software. This work Implements an ADPLL with Nyquist rate phase detector which is basically a digital multiplier, simulation results proves a very high speed of operation for low frequency ranges and resource utilization on FPGA proves the structure simpler.**

***Keywords*—** ADPLL, DCO, FPGA, Loop Filter, Phase Detector, PLL, wireless communications, Xilinx

I. INTRODUCTION

A Phase Locked Loop is a closed-loop control system that is used for the purpose of synchronization of the phase and frequency with that of an incoming signal. Analog PLLs are in wide use in Television, Radio, Pager, Telephony, Servo Motor control and several other areas. Advances in Telecommunication, Wireless & Wire line, and Intelligent Network concepts is posing greater demand towards design of PLLs. Faster and efficient operation of PLLs is very much desired. Implementation of a digital PLL on an FPGA helps to control the jitter involved in the operation of PLLs to a greater extent that is troubling the current communication industry [1]. This paper describes effective method for the modelling of an All Digital Phase Locked Loop and its verification using Xilinx ISE and Xilinx Spartan 3E FPGA kit.

II. COMPONENTS OF ADPLL

A Phase Locked Loop is used for the purpose of synchronization of the frequency and phase of a locally generated signal with that of an incoming signal. There are three components in a PLL. The Phase and Frequency detector (PFD), the loop filter and the Voltage Controlled Oscillator (VCO). The VCO is the heart of any PLL. The mechanism by which this VCO operates decides the type of the PLL circuit being used. There are basically four types of constructing PLLs:

1. Linear PLL(LPLL)
2. Digital PLL(DPLL)
3. All Digital PLL(ADPLL)
4. Software PLL(SPLL)

The analog PLL or the Linear PLL has been in use since a long time. It basically uses a multiplier circuit for serving the purpose of the PFD and a first order filter for the loop filter and a typical analog VCO. Though the name Digital is present in the DPLL, it's not exactly a complete Digital PLL. The All Digital PLL makes an attempt at digitizing all the three components required for the operation of a phase locked loop.

III. DESIGN OF ADPLL

The PD was the only component that was digitized long back. It's used in the Digital PLL. Similar idea can be extended to the ADPLL. The three common implementations of the digital PD are:

1. Exclusive-or (EXOR) Gate
2. Edge triggered JK Flip-Flop
3. Digital Phase-Frequency Detector

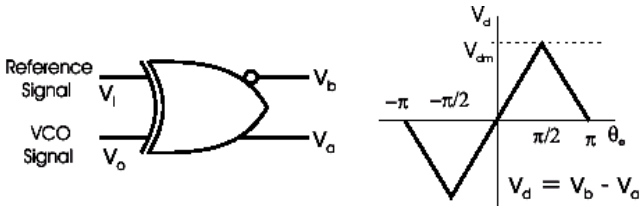


Fig 1: XOR Implementation of a Phase & Frequency Detector

The EXOR mechanism offers a simple yet reliable method of phase detection. One main drawback of this mechanism is its lack of sensitivity to edges. It's a flat triggered mechanism. To eliminate this drawback the edge triggered mechanism comes into picture. The edge triggered JK mechanism is the most popular and effective one. It is sensitive to the edges and hence instantaneous corrective action can be achieved. The incoming reference signal acts as one input and the output of the Digital Controlled Oscillator (feedback of the PLL) acts as the other input. This edge triggering mechanism has been used in the design of the current Jitter bounded ADPLL.

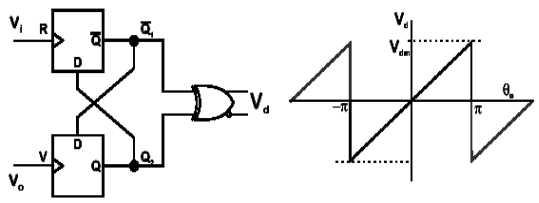


Fig 2: Edge Triggering Mechanism

Using the above shown edge triggering mechanism a logical extension can be done to the simple JK flip-flop such that it is sensitive to the edges and the clock can be eliminated. As shown in the Fig 3, the reference signal u1 and the output (or scaled-down output) signal u2 of the DCO are binary-valued signals. They are used to set or reset an edge triggered JK flip-flop. The time period in which the Q output of the flip-flop is a logic 1 is proportional to the phase error θe. The Q signal is used to gate the high-frequency clock signal into the counter. The counter is reset on every positive edge of the u1 signal. The content N of the counter is also proportional to the phase error θe, where N is the n-bit output of this type of phase detector. The frequency of the high-frequency clock is usually high frequency clock, where f0 is the frequency of the reference signal and M is a large positive integer.

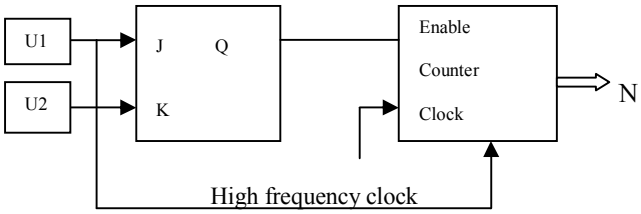


Fig 3: Edge Triggered JK Phase Detector

IV. ALL DIGITAL LOOP FILTER

There are two most important types of digital loop filters.

1. Up/Down counter loop filter
2. K Counter loop filter

The digital filter is not always present in phase locked loops. But in higher order loops where applications such as servo control, Telecommunications are involved the digital loop filter is necessary. Different Phase Detectors generate different types of signals. The simplest phase detector can be built from an Up/Down Counter.

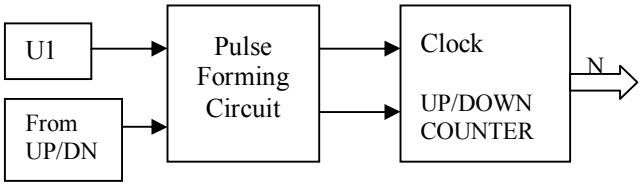
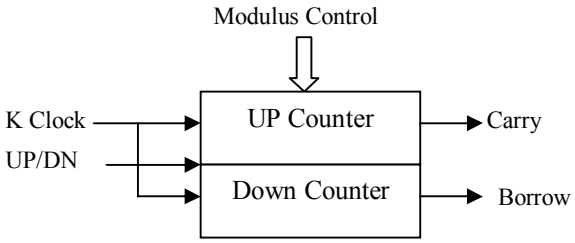


Fig 4: UP/Down Counter Loop Filter

It is easily adapted to operate in conjunction with an XOR or JK-flip-flop phase detectors and others. As shown in Fig 4, a pulse forming network is needed which converts the incoming UP and DN pulses into a counting clock and a direction (UP/DN) signal. On each UP pulse generated by the phase detector, the content N of the UP/DOWN counter is incremented by 1. A DOWN pulse will decrement N in the same manner. The content N is given by the n-bit parallel output signal UP of the loop filter.

One of the most important digital loop filters is the K counter shown in Fig 5. This loop filter always works together with the EXOR or the JK-flip flop phase detector.



The K counter consists of two independent counters, which are usually referred to as "UP-counter" and "DOWN-counter". In reality, however, both counters are always counting upward. K is the modulus of both counters; that is, the contents of both counters are in a range from 0 . . . k-1. K can be controlled by the K modulus control input and is always an integer power of 2. The frequency of the clock signal (K clock) is by definition M times the center frequency f0 of the ADPLL, where M is typically 8, 16, 32 . . . The operation of the K counter is controlled by the DN/UP signal. If this signal is high, the "DN-counter" is active, while the contents of the UP-counter stay frozen. In the opposite case, the "UP-counter" counts up but the DN-counter stay frozen. Both counters recycle to 0 when the contents exceed K-1. The most significant bit of the "UP-counter" is used as a "carry" output, and the most significant bit of the "DN-counter" is used as a "borrow" output. Consequently, the carry is high when the content of the UP-counter is equal to or more than K/2. In analogy, the borrow output gets high when the content of the DN-counter is equal to or more than K/2. The positive-going edges of the carry and borrow signals are used to control the frequency of a digitally controlled oscillator as shown figure 7.[2][3][4]

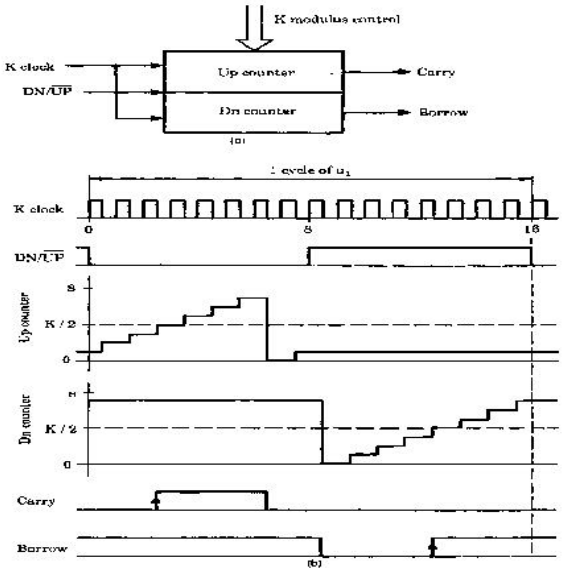


Fig 5: K Counter Loop Filter (a) Block Diagram (b) Corresponding waveforms.

A. *DIGITAL CONTROLLED OSCILLATORS*

A variety of DCOs can be designed. Depending upon output of the loop filter they change their frequency. Increment Decrement counter is used for our ADPLL design.

The simplest solution is the ÷N counter DCO. A÷N counter is used to scale down the signal generated by a high-frequency oscillator operating at a high frequency. The N-bit parallel output signal of a digital loop filter is used to control the scaling factor N of the ÷N counter.

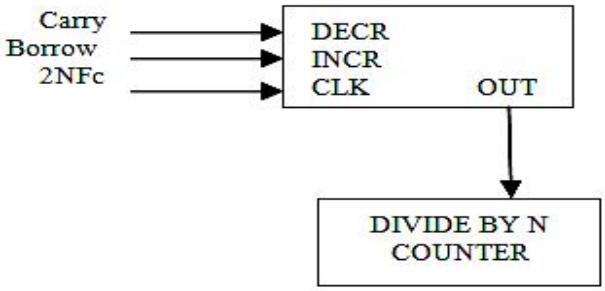


Figure 6: INCR and DECR counter

The main limitation of the ÷N Counter DCO is that it does not offer a jitter design criterion. That is, choice of a predefined jitter and tuning the ADPLL design based on desirable precision requirements is not possible in the case of a ÷N Counter DCO. Increment-Decrement Counter consists of two blocks. Carry is assigned to DECR input and Borrow is assigned to INCR input. ID counter with ÷ N counter for again dividing the OUT. Clock of increment-decrement counter is 2N times multiple of center frequency. Fig. 4 gives overall structure if no Carries and Borrows are present then ID counter divides OUT by 2 on the positive edges of ID clock. The logical function for ID counter is given by

ID out = (NOT (ID clock) AND (NOT (togg1e-FF))

If carry is present then half cycle is added and if borrow is present then half cycle is removed from OUT. Here out is output of increment-decrement counter. The adjusted waveform is shown below.

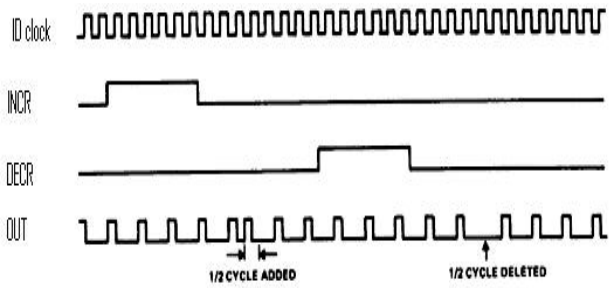


Fig 7: Waveforms of Increment-decrement counter

B. *ADPLL DESIGN USING VERILOG*

ADPLL components are Phase Detectors (PDs), Loop Filters (LFs) and Digital Controlled Oscillators (DCOs) [10]. Based on all these components we can design various types of ADPLLs. For designing ADPLL, Verilog hardware description language is used[9]. It is simulated in ISE Xilinx 12.1 and then this circuit is implemented into FPGA Spartan 3E kit.

C. *DESIGN PROCESS*

All basic building blocks of ADPLL[9] i.e. Phase Detector (PD), Loop Filter (LF) and Digital Controlled Oscillator (DCO) are designed using Verilog (HDL). EXOR gate and JK Flip flop are used as PD. Output of EXOR or JK Flip-flop PD is fed into K counter. This signal is represented by Dn/up. K counter clock is Kclk signal. Kclk is M time multiple of center frequency. Up counter and down counter are two independent counters of K counter. The Dn/up signal controls the K counter. If this signal is low then up counter is active and down counter becomes inactive. In other case when this signal is high then down counter is active and up counter becomes inactive. Outputs of K counter are carry and borrow pulses. Carry and borrow are MSB of the up and down counter respectively. Carry pulse is fed into INC input of ID counter .Whereas borrow pulse is fed into DEC input of ID counter .Output of ID counter is IDout .In general one carry pulse adds half cycle to IDout and one borrow pulse delete half cycle to IDout signal.ID clock is 2N time multiples of center frequency. Output of ID counter (IDout) is fed into divide by N counter, which is the last stage of DCO.ID out is used as clock pulses for divide by N counter. Design is verified and simulated in Xilinx ISE 12.1 tool. A diagram of designed and implemented ADPLL [8] is shown in the Fig.7. Also M=2N, so both clock frequency are taken from the same source. The important parameters used for designing ADPLL

are listed in the TABLE I. Table II gives synthesis details of the designed ADPLL.

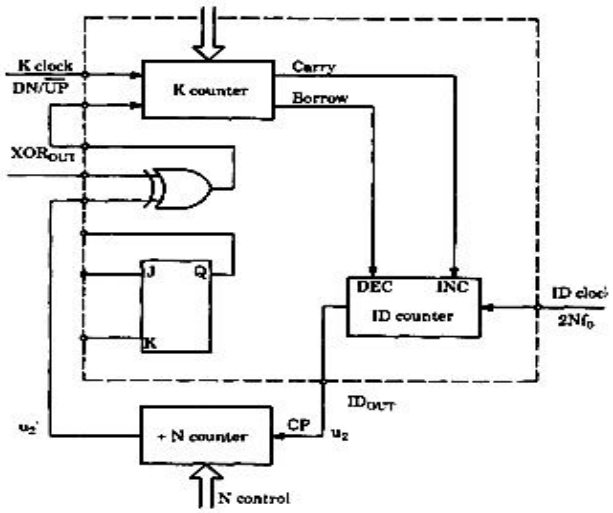


Fig 8: ADPLL circuit diagram

TABLE I
DESIGN PARAMETERS OF ADPLL

PARAMETERS	ADPLL DESIGN
K	8
M	16
N	8
Center frequency(fo)	300 khz

TABLE II
SYNTHESIS DETAILS OF ADPLL

Logic Utilization	Used	Available	Utilization
Number of Slice Flip Flops	45	9,312	1%
Number of 4 input LUTs	104	9,312	1%
Number of occupied Slices	71	4,656	1%
Number of Slices containing only related logic	71	71	100%
Number of Slices containing unrelated logic	0	71	0%
Total Number of 4 input LUTs	130	9,312	1%

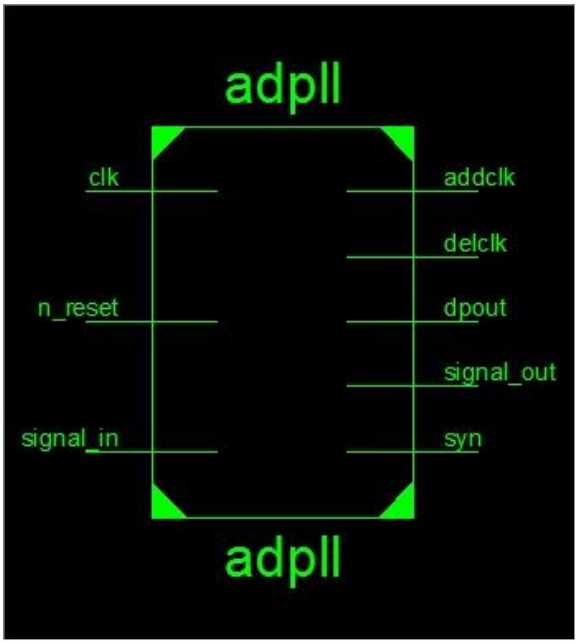


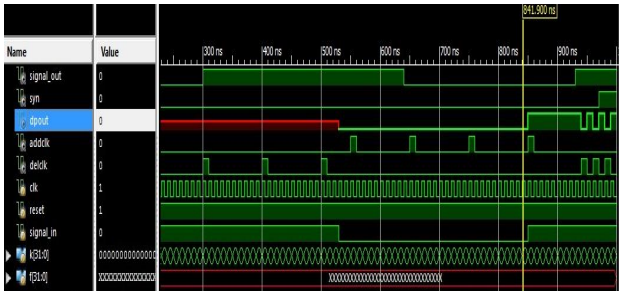
Fig 9: RTL module of designed ADPLL

V. RESULTS

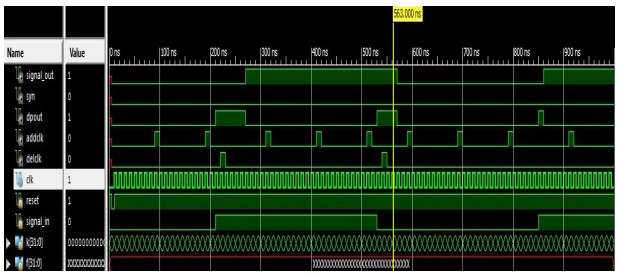
The ADPLL design has been implemented onto a Xilinx Spartan XC3S500E FT256 FPGA chip[8]. The chip is having 9312 LUT's and a maximum of 129 are used. A maximum of 8 are used out of 190 I/O's. Spartan XC3S500E FT256 is one of the primitive chips and after it several chips of greater densities have been released. The current design of ADPLL occupies around a maximum of 1.3% of the total chip space there by leaving a lot of space for fitting the other components present in a bigger and complete SOC IC.

The design has been done keeping in mind the portability, flexibility and optimality criterion. It can be used in any design suiting the given frequency specifications. A system clock of 5 MHz is used. The design is implemented for a center frequency of 300 kHz. It's mainly meant for low frequency applications. The current design offers a operating frequency range of 290 kHz to 320 kHz approximately. The top module of the designed ADPLL has been presented in Fig 8. The design can be extended beyond that also. But the logic criterion needs to be changed to overcome the propagation delay that is introduced due to greater number of bits involved in the computation.

The XPower utility lets to analyze the total power consumption. By setting the appropriate routing path and utilizing these tools the design can be further optimized. The change of platform also decides the power consumption. The latest Spartan 6 FPGAs consume only 1.5 V power.



(a)



(b)

Fig 10: ADPLL output (a)with JK Flip-Flop and (b) Without JK Flip-Flop.

VI. CONCLUSION

Design of an ADPLL for low frequency range has been performed, in view its applications in various fields like wireless communication, biomedical etc, which require a low power, high speed and small devices. The designed can be extended for accumulator based DCO which improves accuracy. Area optimization obtained from the synthesis report shows only 1.3% of device utilization Thus leaving behind a lot of space for other fittings in an SOC IC. This architecture avoids the use of analog vco, provides fine frequency steps, DDs allows exhibits much faster channel switching, Also as a further direction different types of ADPLL architectures can be designed and implemented for various frequencies, depending on its applications. The design and architecture can be modified to increase the range of frequencies and performance of ADPLLS. The primary issue here is the speed from Nyquist sampling theorem, the clock frequency must be at least twice the desired output frequency 1.8GHz in a 900 MHz system, in fact to enhance the DCO requirements Clock should be typically about three to four times the output. In present VLSI technologies it is difficult to perform such operations at such speeds especially if power dissipation is critical.

ACKNOWLEDGMENT

Our sincere thanks to Prof. V. V. Nageswara Rao, Director, Dr. K. B. Madhu Sahu, Principal and Dr. G. Sateesh Kumar, Head of the ECE Department of Aditya institute of technology and management for their encouragement and support in bringing out this paper. I am grateful to all Teaching members of the department for their valuable suggestions and encouragement. The authors sincerely acknowledge the contributions made by other team members.

REFERENCES

[1] Phase-Locked Loops: Design, Simulation & Applications By Roland E. Best, 4th Edition, McGraw-Hill Professional Engineering.

[2] Martin Kumm, HaraldKlingbeil, Peter Zipf, "An FPGA Based Linear All-Digital Phase-Locked Loop", IEEE Trans. on Circuits and Systems 57-I(9): 2487-2497 (2010).

[3] Kwang-Jin Lee, Hyo-Chang Kim, Uk-Rae Cho, Hyun- GeunByun, Suki Kim, "A Low Jitter ADPLL for Mobile Applications", IEICE Transactions 88-C(6), 1241-1247 (2005)

[4] Nursani Rahmatullah "Design of All Digital FM ReceiverCircuit", Institute Technology Bandung, Indonesia, March

[5] VHDL: Programming By Example by Douglas L. Perry - McGraw-Hill Professional, 2002.

[6] Xilinx Spartan 3 FPGA Complete Data Sheet. http://direct.xilinx.com/ bvdocs/publications/ds001.pdf.

[7] Xilinx Software Manuals and Help for Xilinx ISE CAD Tools http://www.xilinx.com/support/sw_ manuals/xilinx5/index.htm.

[8] Digital Signal Processing with Field Programmable Gate Arrays by Uwe Meyer-Baese, Springer-Verlag publishers

[9] Using FPGAs for DPLL Applications, Actel Programmable Logic Solutions http://www.actel.com /documents/s04_18.pdf.

[10] S. Walters and T. Troudet, "Digital Phase-Locked Loop with Jitter Bounded", IEEE Transactions on Circuits and Systems, Vol. 36, No. 7, July 1989.[7]C. Rowen, "Reducing SoC Simulation and Development Time," IEEE Computer, vol.35, no.12, December 2002, pp.29-35.