FPGA IMPLEMENTATION OF PHASE LOCKED LOOP (PLL) WITH SYNCHRONOUS RESET

[1]TANVIR ZAMAN KHAN, [2]PRAJOY PODDER, [3]MAMDUDUL HAQUE KHAN, [4]M. MUKTADIR RAHMAN

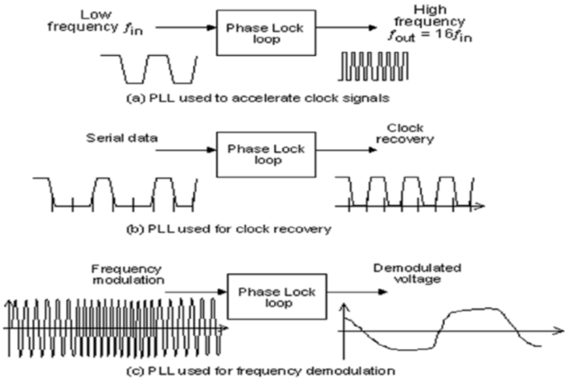
[1,2,3,4]Dept. of ECE, Khulna University of Engineering & Technology, Khulna-9203, Bangladesh
E-mail: [1]tzkhan19@gmail.com, [2]prajoypodder@gmail.com, [3]mehedi.35611@yahoo.com

Abstract- Modern high frequency, high performance system-on-chip design is heading to include more and more analog or mixed signal circuits as well as digital blocks. As the complexity of a system grows, it becomes more and more important to implement the system simulation and top-down design methodology as well. In this paper, we have designed a phase locked loop using Verilog and Xilinx .Considering the rapid growth in computer automation and computer networking sector, FPGA implementation technique of PLL has been adopted in this paper. Verilog is a hardware description language (HDL) used to model electronic systems. This PLL model basically used for synchronization of closed loop RF control signals. In ASIC system sometimes it is very important to synchronize the input signal with the output signal. Synchronous PLL can be used for this purpose. This paper has considered the minimum number of devices like Flip Flop, comparator, adder or subtractor due to enhance the efficiency of the system and the time delay of PLL is also considered to be much less.

Keywords- Voltage controlled oscillator, Phase detector, Synchronous reset and time delay.

I. INTRODUCTION

A phase-locked loop or phase lock loop (PLL) is a control system that generates an output signal whose phase is related to the phase of an input signal. It is widely used in modern wireless communication system owing to high frequency resolution and fast settling time. On the other hand, it also encounters some challenges. PLL frequency synthesizer is a complex mixed-signal system, and there is no stable periodic solution. Phase-locked loops are widely employed in radio, telecommunications, computers and other electronic applications.They can be used to demodulate a signal, recover a signal from a noisy communication channel & generate a stable frequency at multiples of an input frequency (frequency synthesis), or distribute precisely timed clock pulses in digital logic circuits such as microprocessors. Phase locked loop can be implemented in synchronous condition where after a definite time interval the PLL is locked.



(a) PLL used to accelerate clock signals

(b) PLL used for clock recovery

(c) PLL used for frequency demodulation

Fig.1 PLL IN SIMPLE DIGITAL SYSTEM

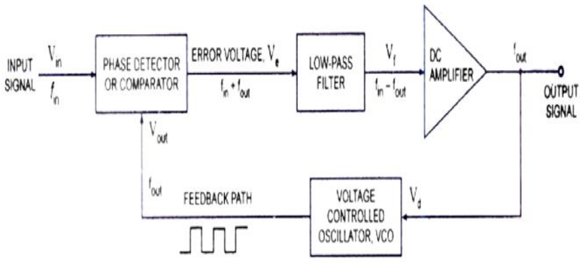


Fig.2 Block diagram of phase locked loop

Phase locked loop normally consists of voltage controlled oscillator, phase detector or comparator, low pass filter (fig.2). The oscillator generates a periodic signal. The phase detector compares the phase of that signal with the phase of the input periodic signal and adjusts the oscillator to keep the phases matched. Bringing the output signal back toward the input signal for comparison is called a feedback loop as the output is 'fed back' toward the input forming a loop.

Keeping the input and output phase in lock step also implies keeping the input and output frequencies the same. Consequently, in addition to synchronizing signals, a phase-locked loop can track an input frequency, or it can generate a frequency that is a multiple of the input frequency. These properties are used for computer clock synchronization, demodulation, and frequency synthesis, respectively.

II. VCO

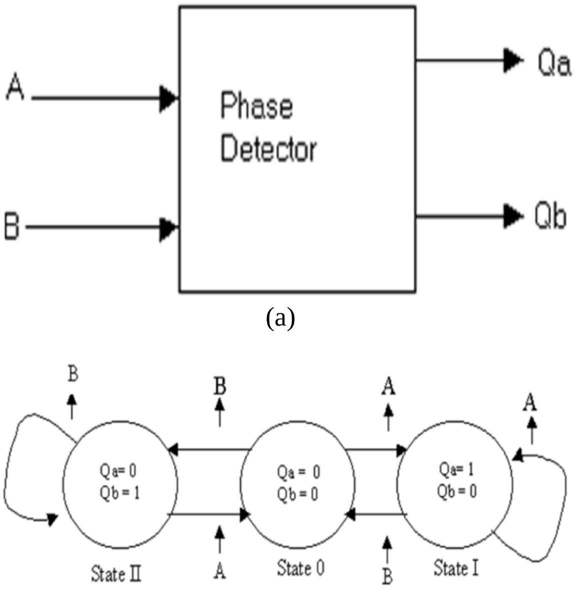
A voltage-controlled oscillator or VCO is an electronic oscillator. Its oscillation frequency is controlled by a voltage input. The applied input voltage states the instantaneous oscillation frequency. Subsequently, modulating signals applied to control input may cause frequency modulation (FM) or phase modulation (PM).VCOs may have sine and/or square wave outputs. Function generators are low-frequency oscillators which feature multiple waveforms, usually sine, square, and triangle waves. Analog phase-locked loops normally contain VCOs. High-frequency VCOs are applied in phase-locked loops for radio receivers. Low-frequency VCOs are used in analog music synthesizers. Therefore, sweep range, linearity, and distortion are often most important specs.

III. PHASE DETECTOR

A phase frequency detector (PFD), in electronics, is a device which compares the phase of two input signals. It has two inputs which correspond to two different input signals, usually one from a voltage-controlled oscillator (VCO) and another from some external source. It has two outputs which instruct subsequent circuitry on how to adjust to lock onto the phase. To form a Phase-locked loop (PLL) the PFD phase error output is fed to a loop filter which integrates the signal to smooth it. This smoothed signal is fed to a voltage-controlled oscillator which generates an output signal with a frequency that is proportional to the input voltage. The VCO output is also fed back to the PFD to form the PLL circuit.

The block diagram of a phase/frequency detector (PFD) is shown in Fig. 1. If the frequency of input A is less than that at input B, the PFD produces positive pulses at Qa, while Qb remains at zero. Conversely if the frequency at input B is higher than the frequency at input A, the PFD produces pulses at Qb and Qa remains at zero. If both frequencies are equal, then the circuit generates pulses at either Qa or Qb with a width equal to the phase difference between the two inputs.

The state diagram shown in Fig. 2 is used to implement the PFD. The circuit can change state only on a rising edge of A or B signals. If the PFD is in state 0, Qa= Qb = 0, then a transition on A takes it to state I, where Qa = 1, Qb = 0. The circuit remains in this state until a transition occurs on B, upon which the PFD returns to state 0.



(a)

(b)

Fig.3 (a) A simple phase detector (b) State diagram

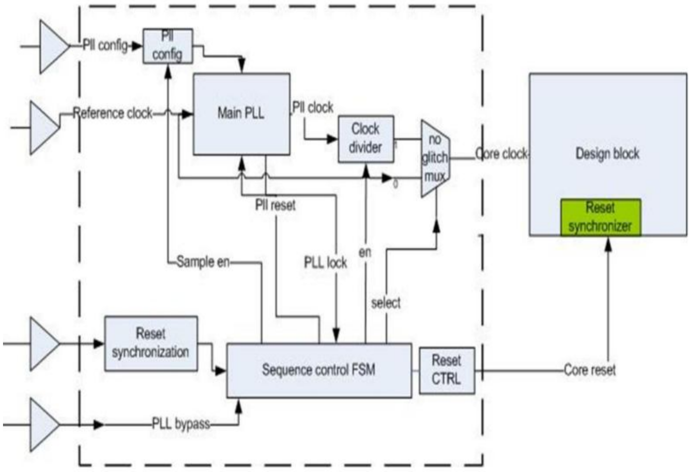


Fig.4 Block diagram of PLL synchronization with reset in FSM

IV. PLL WITH SYNCHRONOUS RESET

A power-up and initialization sequence, enabling clock to start and reset to be propagated through to all parts of the logic is required in every ASIC device. The clocking and reset initialization sequence is dissimilar from other purely logical parts such as it handles slow signals external to the device, an analog

PLL and depends on the sequencing of the device power rails, external clock generators and I/O pads. In many cases, testability and backup logic is also included, further complicating this delicate logic. This component is a basic example of a device clocking and reset sequence. It can be used as a reference and starting point for the implementation of this logic. This component contains RTL Verilog code for a reference design for a chip level clock and reset synchronization and initialization. The component contains the PLL startup logic, reset synchronization and propagation and handles clock selection and bypassing as well as reset distribution at the chip top level.

V. DESIGN STEPS

Some important design steps must be followed to accomplish desired operation.

TABLE I: Input and Output Declaration

Input	Output
reset	Counter used to lock onto the clock
Limit(The upper limit for the counter)	Clk_out(Clock output)
Clk(Fast system clock)	
Clk_in(Clock input)	

i. Define module and parameters like clock to output delay, maximum frequency of the PLL and the duty cycle of the output clock
ii. Input and output declarations
iii. Signal declarations in register and wire
iv. Look at the rising edge of system clock
v. Store the clock input for finding rising edges
vi. Look for the rising edge of the input clock which ideally happens at the same time as counter =limit
vii. Add or remove counter cycles to synchronize its phase with that of the input clock.
viii. If (counter $\neq$ limit) and (counter < (limit >> 1)).Then the output clock edge came too late, so subtract two from the counter instead of the usual one.
ix. Otherwise add an extra cycle by not decrementing the counter this cycle.

VI. SIMULATION RESULT

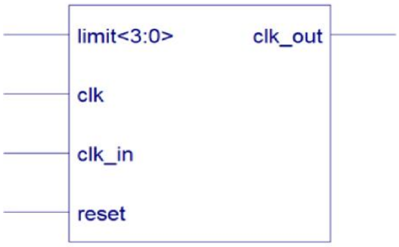


Fig.5: Pin Diagram of Phase Locked Loop (PLL)

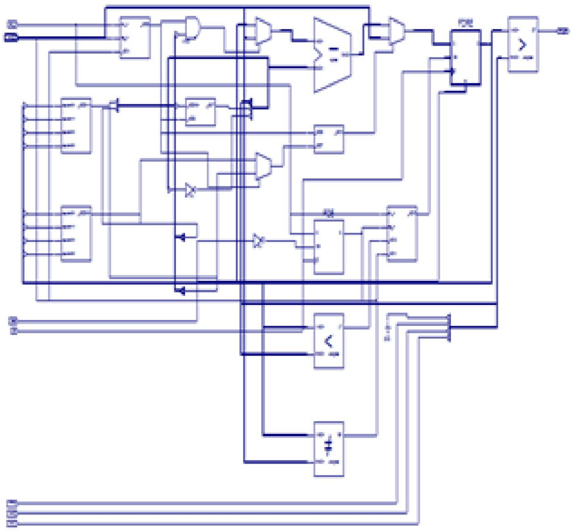


Fig.6: RTL schematic

FPGAs can be used to implement any logical function and cost-effective. They have lots of advantages over microcontrollers, such as greater speed, number of I/O ports and performance. The proposed design is modeled in HDL, using Xilinx 6.3i with SPARTAN 3 family (XC3S1500), FG320 package logically verified and then synthesized in XST synthesis tool. The timing summary and Device utilization summary and are given in Table II and Table III respectively. In the timing summary, it is seen that the maximum delay is 10.367ns. From the table it is seen that only 3flip flop is used among 384. In table III it is seen delay with respect to device connected to Phase locked loop (PLL).

VII. TIMING DIAGRAM

A counter is used in the simulation code to generate a clock input to the PLL. The counter is a down counter. The output clock frequency is caused to change by changing the upper limit of the counter. By observing for a low value of the output clock on the previous cycle and a high value on the current cycle, a rising edge on the output clock can be determined. When this rising edge occurs during the same cycle as a rising edge on the input clock, three times in a row, then the PLL is considered to have locked to the incoming clock. A rising edge on the input clock happens when the counter reaches zero.

This paper attempts to get the PLL to lock onto several different frequencies by changing the counter limit. The simulation ends successfully if all of the frequencies can be locked ON by the PLL. The simulation stops and reports an error If the PLL does not lock within a certain large number of cycles. It should be noted that the input clock is purposely a clock with a lopsided duty cycle. The simulation waveforms can be examined to realize that the output clock is approximately a square wave and does not depend on the duty cycle of the input clock.

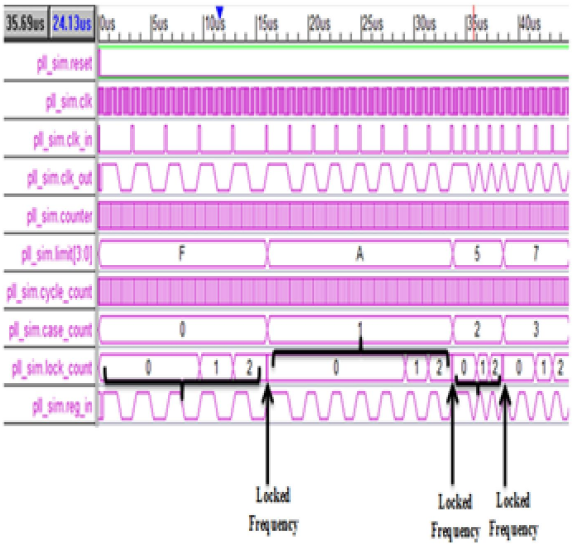


Fig.7: Timing diagram of PLL with synchronous reset

Table II: TIMING SUMMARY

Parameter	Seconds
Minimum period:	12.770ns
Minimum input arrival time before	12.056ns
Maximum output required time after	11.081ns
Maximum delay:	10.367ns

Table III: DEVICE UTILIZATION SUMMARY

Name	Used Blocks	Percentages (%)
Number of Slices	18 out of 192	9
Number of Slice Flip Flops	5 out of 384 (FDE:1 FDRR:4	1
Number of 4 input LUTs	32 out of 384	8
Number of bonded IOBs	7 out of 90 IBUF:6 OBUF:1	7
Number of GCLKs	1 out of 4	25

Table IV: HDL Synthesis Report

Name	Sub- Name	quantity
# Adders/Subtractors	4-bit subtractor	1
# Registers	1-bit register	1
	4-bit register	1
# Comparators	4-bit comparator not equal	1
	4-bit comparator less	1
	4-bit comparator greater	1
# Multiplexers	4-bit 2-to-1 multiplexer	2
	1-bit 2-to-1 multiplexer	1

CONCLUSION

PLL is widely used in wireless communication systems as well as telecommunication system. FPGA implementation ensures easy and computer control over the system. This paper has discussed the PLL basic structure and designed a PLL that can be used efficiently for device synchronization purpose. Xilinx is used for designing the schematic diagram using the RTL logic. Verilogger pro 6.5 is used for observing the timing diagram of the designated PLL with synchronous reset. From the synthesis report, it has been also observed that how many subtractors, flip flops, registers, comparators and multipliers are needed to design. In future our aim is to design a PLL for frequency jittering purposes which will have a built in self-test capability.

REFERENCES

[1] G. Eynard; N.Lewis; D.Dallet; B.Le Gal, "Efficient phase noise modeling of a PLL-based frequency synthesizer" 4th International Symposium on Communications, Control and Signal Processing (ISCCSP), 2010.

[2] Oura, T. ; Shizuoka Univ., Hamamatsu, Japan ; Hiraku, Y. Suzuk, T. ; Asai, H.," Modeling and simulation of phase-locked loop with verilog," Circuits and Systems, 2004. Proceedings. The 2004 IEEE Asia-Pacific Conference on (Volume:1)

[3] en.wikipedia.org/wiki/Phase-locked_loop

[4] http://www.sentex.ca/~mec1995/gadgets/pll/pll.html

[5] Martin Kumm; Harald Klingbeil; Peter Zip,'' An FPGA-Based Linear All-Digital Phase-Locked Loop", IEEE transactions on CIRCUITS AND SYSTEMS—I: REGULAR PAPERS, VOL. XX, NO. Y, SEPTEMBER 2009

[6] B.Alecsa; Gh. Asachi;Iasi ;'' An FPGA Implementation of an All Digital Phase Locked loops", A.Intelligent Computer Communication and Processing, 2009. ICCP 2009,IEEE 5th International Conference on 2009

[7] Dr Anu Gupta; Mohammad Waqar Ahamed; Abhishek Dhir;''Novel method to implement high frequency All Digital Phase-Locked Loop on FPGA", International Conference on VLSI Communication & Instrumentation (ICVCI) 2011 Proceedings published by International Journal of Computer Applications(IJCA).

[8] H. Klingbeil ;"A fast DSP-based phase-detector for closed-loop RF control in synchrotrons", IEEE Trans. Instrum. Meas., vol. 54, no. 3, pp.1209 -1213 2005

[9] W. Namgoong ;"Observer-controller based digital PLL", IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 57, no. 3, pp.631 -641 2010

[10] en.wikipedia.org/wiki/Phase_frequency_detector

[11] VLSI projects/FREQUENCY DIVIDER USING PLL(vhdl) _ VLSI For You.htm

[12] DIGITAL VLSI DESIGN WITH VERILOG: A TEXTBOOK FROM SILICON VALLEY TECHNICAL Institute-John Williams

[13] Verilog Designers library-Bob Zeidman

★★★