The 1-to-7 Phase Decoder Circuit Design using VHDL

Dr. SERAFIM PORIAZIS
Phasetronic Laboratories
6 Depasta Str., 17122, N.Smirni, Athens
GREECE
http://www.phasetroniclab.com

Abstract: - The design of the 1-to-7 Phase Decoder (PD7) circuit is described by using a core written in synthesizable VHDL. The circuit decodes the input line that carries a sequential pattern of seven clock phases into seven output lines by streaming sets of forteen clock phases. A phase difference equal to the half period of a clock signal is used at the input line and this is maintained between the consecutive output line transitions. The core model of the PD7 is given in behavioral VHDL description and the simulation and synthesis results are generated for targeting an FPGA device. The inverse function of the PD7 is obtained through an EXOR7 gate in a serial configuration. An extension of the circuit behavior is verified by applying the phased outputs of the PD7 circuit into seven replicas of the core and the associated simulation and synthesis results are obtained by using a 7x7 phase aggregation configuration.

Key-Words: - Circuit design, Decoder, EXOR, FPGA, Phase, VHDL

1 Introduction

VHDL (VHSIC Hardware Description Language) is intended to describe digital electronics systems from the abstract to the concrete level. VHDL CAD tools allow a design written in VHDL to be synthesized and targeted to a suitable technology. An FPGA (Field Programmable Gate Array) can be the target technology of a specific decoder, such as the BCH (15, 7, 5) minimum weight decoder for double error as given by [1].

The VHDL description is run through a VHDL simulator which demonstrates the effectiveness of the model of a 3 to 8 line decoder as given by [2]. The detailed VLSI design of a turbo decoder based on Log-MAP decoding algorithm is given by [3] where the VLSI implementation is performed using VHDL as design and simulation entry. The logic design of the Serial Viterbi Decoder (SVD) is applied with top-down design methodology using VHDL, and simulated by VHDL logic simulator [4], while the logic timing is verified using output VHDL file by VHDL synthesis and FPGA place and routing. A Multi-Amplitude Continuous Phase Receiver that uses a new design of phase generating circuit has been designed by generating and synthesizing the VHDL code of the behavioral model [5].

In this paper the 1-to-7 Phase Decoder (PD7) circuit is designed by using VHDL and represents a reliable solution to the challenging problem of synchronizing the individual modules of a multiphase model [6], whose operation adopts a 7-phase timing pattern. The present work is based on the principles of operation of the Two-Phase Twisted Ring Counter (2P-TRC) circuit [7] and is targeting data streaming applications. The unit phase duration that is used throughout the text is equal to the half period of an input clock signal. The VHDL simulation and synthesis results are presented. The EXOR operator is associated to the aggregation of the phased outputs produced by the PD7. A transposition mechanism assists us to build a primitive PD7/EXOR7 and an expanded 7x7 aggregation configuration to verify that the phase relationships between the input, the decoded signals

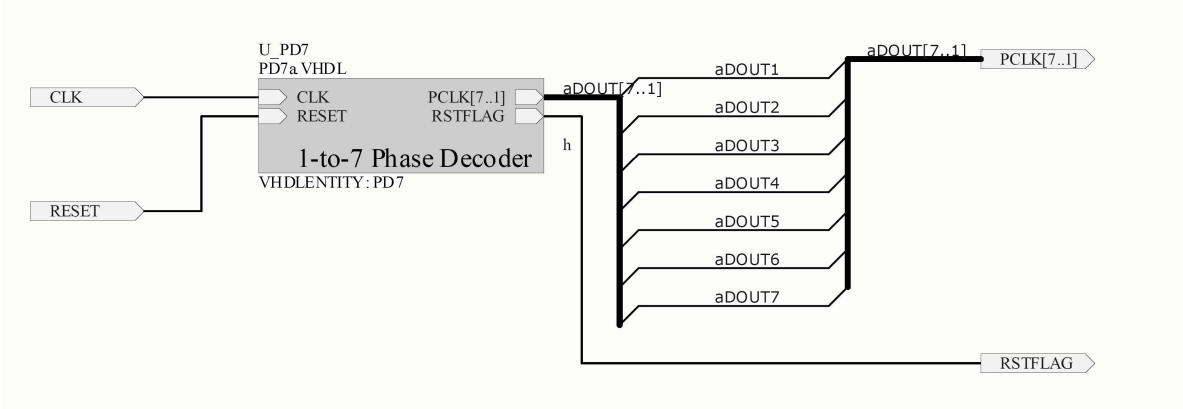

Fig. 1 The PD7 basic module block diagram

and the aggregated output are fully preserved.

2 The PD7 Core

2.1 The basic module operation

The fundamental module that decodes an input line signal *CLK* of frequency *f* into seven line phased signals of frequency *f*/7, is called 1-to-7 Phase Decoder (PD7) circuit and is shown in Figure 1. The seven overlapping output line signals CLK_1=aDOUT1, CLK_2=aDOUT2, CLK_3=aDOUT3, … , CLK_7=aDOUT7 have frequency equal to *f*/7 (period of each CLK_i, i=1,2,3,…,7 equals 7·T, where T the period of *CLK*) with signal CLK_1 leading CLK_2 , CLK_2 leading CLK_3 , … , CLK_6 leading CLK_7 by a T/2 phase difference. We note that a clock signal is used as the input line signal of the module throughout this work for implementing a pattern of a sequence of seven phases. The logic-'1' or logic-'0' pulse width of each of the above phased signals is equal to 7·T/2. Consecutive changes of logic value at each signal CLK_i, for i=1,2,3,…,7 occur alternatively at the rising and falling edges of *CLK* at a distance of 7·T/2.

2.2 The valid line codeword sequence

When the line signal *CLK* is applied to the circuit, the following cyclic sequence of line codewords is presented at the outputs: CLK_1,CLK_2,CLK_3, … , CLK_7= 0000000→ 1000000→1100000→1110000→1111000→1111100→111 1110→1111111→0111111→0011111→0001111→000011 1→0000011→0000001, which is considered as being the normal circuit operation. Each line codeword remains stable for the state time of the circuit, that is T/2, and the above sequence is repeated throughout the operation of the module. Thus the cycle time for the output pattern is defined by the forteen-tuple of codewords of length equal to 7·T. This duration forms the period of each phased output line signal.

The additional 114 line codewords out of the total 128 possible line codewords that are not included in the above cyclic sequence should be considered during the design of the circuit for achieving reliable operation of the PD7 core. If the circuit reaches any of these 114 invalid codewords, then an invalid codeword flag is set at the output of the module. This flag always forces the reset input ports of cascaded PD7 modules to operate in the reset state, which maintains the proper initializing behavior until a valid line codeword appears on the output port of the module.

2.3 The algorithm aspects of core operation

The PD7 core operation is implemented by the behavioral VHDL description shown in Figure 2. The VHDL entity section has an input port CLK on which the line signal of frequency *f* is applied and an input port RESET on which a reset flag is applied. The phased output line signals of frequency *f*/7 of the module are assigned to port PCLK[7..1] of width seven, that is PCLK1=aDOUT1, PCLK2=aDOUT2, … , PCLK7= aDOUT7. The output port RSTFLAG is signaling the invalid line codeword status of PCLK, or the core reset state, which suspends the streaming of phases from CLK toward the outputs of the module. The VHDL architecture section is of type "behavioral" and utilizes a state machine model, where two internal registers are being used, reg1 and reg2, one for the present state named "present_state1" clocked by the rising edge of the clock and the other for the present state named "present_state2" clocked by the falling edge of the clock, respectively. The next state logic block and the ouput logic

```
1   library IEEE;
2   use ieee.std_logic_1164.all;
3
4   entity PD7 is
5      port (CLK , RESET : in std_logic;
6           RSTFLAG : out std_logic;
7           PCLK : out std_logic_vector(7 downto 1));
8   end PD7;
9
10  architecture behavioral of PD7 is
11       type validcodewords is array(1 to 14) of std_logic_vector(7 downto 1);
12       constant phased_output : validcodewords :=
13       ( ( '0', '0', '0', '0' ,'0', '0', '0'), ( '0', '0', '0', '0', '0', '0', '1'), ( '0', '0', '0', '0', '0', '1', '1'),
14         ( '0', '0', '0', '0', '1', '1', '1'), ( '0', '0', '0', '1', '1', '1', '1'), ( '0', '0', '1', '1', '1', '1', '1'),
15         ( '0', '1', '1', '1', '1', '1', '1'), ( '1', '1', '1', '1', '1', '1', '1'), ( '1', '1', '1', '1', '1', '1', '0'),
16         ( '1', '1', '1', '1', '1', '0', '0'), ( '1', '1', '1', '1', '0', '0', '0'), ( '1', '1', '1', '0', '0', '0', '0'),
17         ( '1', '1', '0', '0', '0', '0', '0'), ( '1', '0', '0', '0', '0', '0', '0') );
18        signal index : integer := 0;
19        signal present_state1, present_state2, next_state: std_logic_vector(7 downto 1);
20        signal invalidcode_flag : std_logic := '0';
21       begin
22        reg1 : process (CLK, RESET)
23          begin
24            if RESET = '1' then present_state1 <= phased_output(1);
25            elsif (CLK='1' and CLK'event ) then present_state1 <= next_state ;
26            end if;
27        end process;
28        reg2 : process (CLK, RESET)
29          begin
30            if RESET = '1' then present_state2 <= phased_output(8);
31            elsif (CLK='0' and CLK'event ) then present_state2 <= next_state ;
32            end if;
33        end process;
34        next_state_logic : process (CLK, RESET)
35          begin
36            case CLK is
37             when '1' => if RESET = '1' then index <= 1; else index <= index + 1; end if;
38             when '0' => if RESET = '1' then index <= 8; else index <= index + 1; end if;
39             when others => null;
40            end case;
41            if index < 14 then next_state <= phased_output(index + 1);
42            else next_state <= phased_output(1); index <= 1; end if;
43            for i in 1 to 14 loop
44              if next_state = phased_output(i) then invalidcode_flag <= '0'; exit;
45              else invalidcode_flag <= '1'; end if;
46            end loop;
47        end process;
48        output_logic : process (index, present_state1, present_state2)
49          begin
50            case CLK is
51             when '1' => PCLK <= present_state1;
52                         if RESET = '1' then RSTFLAG <= '1'; else
53                         RSTFLAG <= invalidcode_flag; end if;
54             when '0' => PCLK <= present_state2;
55                         if RESET = '1' then RSTFLAG <= '1'; else
56                         RSTFLAG <= invalidcode_flag; end if;
57             when others => null;
58            end case;
59        end process;
60  end behavioral;
```

Fig. 2 The VHDL description of the PD7 core

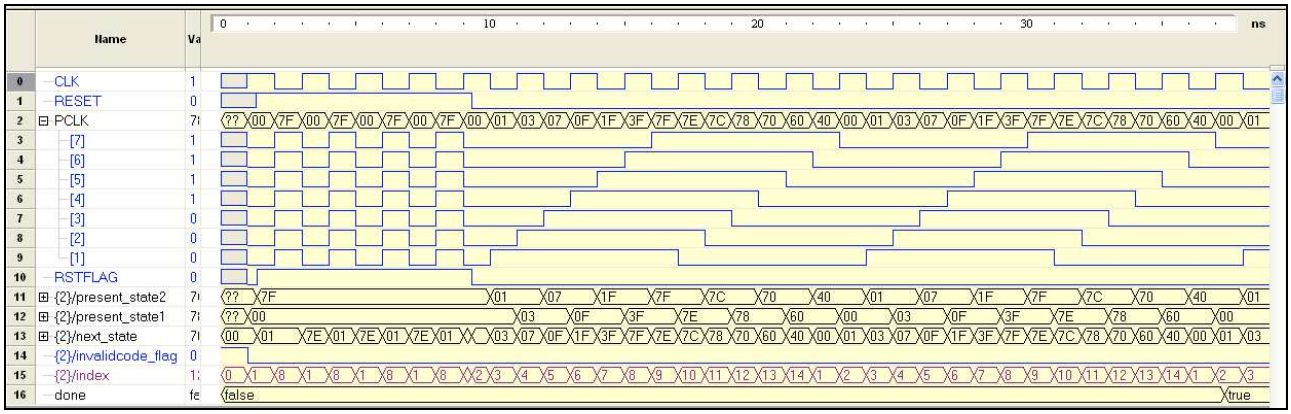

Fig. 3 The PD7 core operation (VHDL simulation results)

block of the model are specified by the corresponding processes "next_state_logic" and "output_logic". The set of forteen valid line codewords of the circuit are stored in an indexed array of size 14*7=98 bits, that is represented by the constant named "phased_output". The index of the above array cycles through the integer values 1 to 14 specifying the valid line codeword entry for the next state signal. Whenever the RESET input is set, that is RESET='1', each output signal from PCLK[7..1] can cycle only through the values "0000000" and "1111111" thus assuring proper initialization of the circuit at either the rising or the falling edge of CLK.

2.4 The VHDL simulation and synthesis

The VHDL testbench simulation results for the PD7 core are given in Figure 3. The duration of this simulation is defined by the value of the signal "done". Each of the signals "next_state", "present_state1", "present_state2" and PCLK have each a width of 7 bits. Each value shown on these signal waveforms is hexadecimal. The output port PCLK is analyzed into seven individual output line signals with waveforms that verify the correct operation of the circuit. The "index" signal has decimal values and defines the index value of the array of valid line codewords. The logic value changes of PCLK occur at each rising and at each falling edge of the input signal CLK.

The synthesis of the PD7 core targeting an FPGA device was successfully performed giving us the following results:

- flip flops with asynchronous reset = 7
- flip flops with asynchronous preset = 7
- combinational feedback paths = 32
- combinational logic area estimate = 137 LUTs

3 The Phase Associations of the PD7 Line Signals

We examine the associations between the input line signal CLK and the phased output line signals PCLK1=CLK_1 , PCLK2= CLK_2 , … , PCLK7=CLK_7 of the PD7 core. It is evident that the EXOR function can be utilized to express the following relationships:

$$\begin{aligned} CLK &= CLK_1 \oplus CLK_2 \oplus CLK_3 \oplus ... \oplus CLK_7 \\ CLK_1 &= CLK \oplus CLK_2 \oplus CLK_3 \oplus ... \oplus CLK_7 \\ &... \\ CLK_7 &= CLK \oplus CLK_1 \oplus CLK_2 \oplus ... \oplus CLK_6 \end{aligned} \qquad (2)$$

The equation (2) satisfies an extended transposition mechanism for the EXOR operator that states the following: "if $f=g\oplus h$, then $g=f\oplus h$ and $h=g\oplus f$ " where f, g and h are binary variables. Thus an EXOR7 gate can be attached to the outputs of the PD7 module in order to aggregate the phased signals CLK_1 , CLK_2 , CLK_3 , … , CLK_7 and produce a replica of the input line signal, that is $CLK\_OUT= CLK$.

A primitive PD7/EXOR7 configuration is thus formed by the two modules, the PD7 module and the EXOR7 gate, which are being interconnected via the phased signals CLK_1 , CLK_2 , CLK_3 , … , CLK_7. All referenced signals of this configuration maintain the transposition mechanism stated above. We notice that the outputs CLK_1, CLK_2, CLK_3, … , CLK_7 of the PD7 are themselves periodic signals that can be used to drive in succession replicas of the module, thus forming a tree-like structure. A new phased line signal pattern of period $49{\cdot}T$ is thus produced with output line signals w_1, w_2, w_3 , …, w_{49} (b1DOUT1, b2DOUT1, b3DOUT1,…, b7DOUT7) with w_i leading w_{i+1} (i=1,2,3,…,49) having a phase difference equal to T/2. This structure (PD7*7) is composed of two levels of modules as shown in Figure 4. According to the frequency of the driving signals, the first level is driven by CLK of frequency f and the second level is driven by y_1=aDOUT1, y_2 =aDOUT2, … , y_7=aDOUT7 of frequency $f/7$. By using an inverse-tree-like structure of EXOR7 gates we can apply the EXOR function to the pattern of the phased line signals w_1, w_2, w_3 , … , w_{49} and form a 2-level expanded PD7/EXOR7 phase aggregation configuration. Its aggregated output signal CLK_OUT is a replica of the input line signal CLK , thus giving $w_1\oplus w_2\oplus w_3\oplus$ … $\oplus w_{49}= CLK\_OUT= CLK$ which have been verified by the associated simulation results.

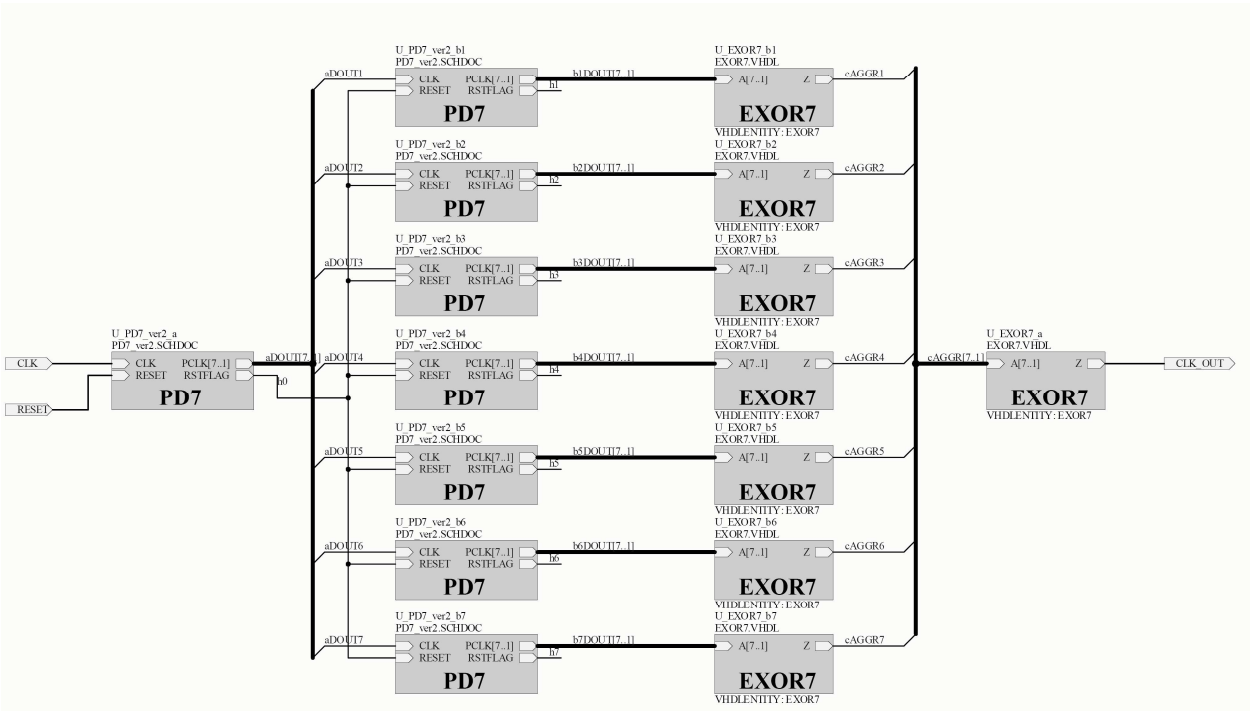

Fig. 4 The 2-level expanded PD7/EXOR7 circuit 7x7_phase_aggregation configuration (block diagram)

The synthesis of the 2-level expanded PD7/EXOR7 circuit 7x7 phase aggregation configuration targeting an FPGA device was successfully performed giving us the following results:

- flip flops with asynchronous reset = 56
- flip flops with asynchronous preset = 56
- combinational feedback paths = 256
- combinational logic area estimate = 1136 LUTs

4 Conclusion

The operation and the design concepts of the 1-to-7 Phase Decoder (PD7) circuit are considered in this paper. The behavioral VHDL description of the PD7 core is given. The corresponding circuit simulation and synthesis results targeting an FPGA device successfully verify the proper core behavior.

A 7x7 phase aggregation configuration is formed by tree-like structures of the basic core and the EXOR gate. The input line is decoded into 49 phased line signals, by a two-level expansion of the core, which are then aggregated into an output line by preserving all the phase associations of the above signals. The configuration has been simulated and synthesized successfully, thus verifying the extension capabilities of the given core.

References:

[1] El-Medany, et al. "*VHDL implementation of a BCH minimum weight decoder for double error*," in Proceedings of the Eighteenth National Radio Science Conference, NRSC 2001, Vol.2, pp.361-368, 2001.

[2] M.B.I. Reaz, S.Z. Islam, "*VHDL modeling of built-in-self-test (BIST) for system on chip (SOC) design*," in Proceedings of the IEEE International Conference on Semiconductor Electronics, ICSE 2002, pp.222-225, 19-21 Dec. 2002.

[3] Tong Yanhui, T.H.Yeap, J.Y.Chouinard, "*VHDL implementation of a turbo decoder with log-MAP-based iterative decoding*," in IEEE Transactions on Instrumentation and Measurement, Vol.53, Issue 4, pp.1268-1278, Aug. 2004.

[4] Park Jang-Hyun, Rho Yea-Chul, "*Performance test of Viterbi decoder for wideband CDMA system*," in Proceedings of the ASP-DAC 1997, Asia and South Pacific Design Automation Conference, pp.19-23, 28-31 Jan. 1997.

[5] H.H. Ali, "*VLSI implementation of a multi-amplitude continuous phase receiver*," in Proceedings of the 44th IEEE 2001 Midwest Symposium on Circuits and Systems, MWSCAS 2001, Vol.1, pp.393-396, 2001.

[6] S. Poriazis, *Logic Design of Multiphase Finite State Machines*, Monograph, Phasetronic Laboratories, http://www.phasetroniclab.com Athens, Greece, 2001.

[7] S. Poriazis, "*The Two-Phase Twisted-Ring Counter Circuit,*" in Proceedings of the 2002 IEEE International Symposium on Circuits and Systems (ISCAS'02), Phoenix, Arizona, USA, vol. IV, pp. 858-861, 26-29 May, 2002.