Physical Realization and Synthesis of Walsh Pattern Generator for programmable devices

Mr. Shinde Navnath D\textsuperscript{1}, Mrs. Dhede Vaishali M.\textsuperscript{2}

\textsuperscript{1}ME-Signal Processing (E&T) Jaihind College of Engineering, Kuran, India nds.pune@gmail.com
\textsuperscript{2}Asst. Prof. & HOD, Department of E&T Jaihind College of Engineering, Kuran, India Vaishali_dhede@rediffmail.com

Received 2017-12-28; Accepted 2018-01-19
DOI: http://currentsciences.info/10.15520/ctst.v8i1.268.pdf

Abstract:
This paper presents design and physical realization and Synthesis of Walsh Pattern Generator on CPLD. The overall architecture for Walsh pattern generation is programmed in a Xilinx Design Tools 8.2i and to be synthesized for the programmable devices such as CPLD; thus to develop a SOC. A description of the hardware's structure and behavior is written in VHDL language and that source code is then compiled and downloaded prior to execution. Physical realization of the design was confirmed by obtaining RTL structure and Synthesis report. The simulation results obtained have been compared with the calculated results by hardware implementation.

Keyword – Hardware Implementation, RTL structure, Electronic Design Interchange Format, Walsh function.

I. Introduction:
Walsh functions and transforms are important analytical tools for signal processing and have wide applications in digital communication, digital image processing, statistical analysis, solving differential equations as well as in digital logic design. Walsh functions consist of trains of square wave pulses such that their transitions may only occur at fixed intervals of a unit time step and having the initial state is always +1. The function WAL\(i.k\) represents a waveform, known as Walsh Pattern, as a function of time k with i transitions over the period of the function defined as Sequency function. Thus, the parameter n can be interpreted as “one half the number of zero crossings per unit time. i is
known as the normalized sequency or sequency. For an index \( n \), \( i = \text{log}_2 n \). There are \( 2^n \) Walsh functions of length \( 2^n \). Furthermore, within the set of \( 2^n \) functions there is one function of zero sequency, one of (normalized) sequency \( 2^{n-1} \), and one pair (odd and even) of each (normalized) sequency from 1 to \( 2^{n-1} - 1 \).

II. Algorithm For Walsh Functions Generation:
In the algorithm for Walsh functions generation, the original \( 1 \)’s are kept, but \( -1 \)’s are replaced by 0, and all the Walsh functions are generated in the complete interval between 0 and 1 rather than between \(-\frac{1}{2}\) and \( \frac{1}{2} \). After the changing of original symbols the basic properties like a modulo-2 addition of two Walsh functions yields another Walsh function hold for all the Walsh functions. To generate, we use the properties described above.

**Step-I:** Let \( \vec{B} = (b_{n-1}, b_{n-2}, \ldots, b_j, \ldots, b_0) \) be a binary vector representing an index of the Walsh function (i.e. the sequency), where \( 0 \leq j \leq n-1 \).

**Step-II:** obtain the Gray code \( \vec{g} = (g_{n-1}, g_{n-2}, \ldots, g_j, \ldots, g_0) \) from the natural binary code \( \vec{B} = (b_{n-1}, b_{n-2}, \ldots, b_j, \ldots, b_0) \), where \( g_j = b_{j+1} \oplus b_j \) for \( 0 \leq j \leq n-2 \), and \( g_{n-1} = b_{n-1} \).

**Step-III:** Generate the Walsh functions in strict sequency ordering as follows (i is the index of walsh patterns):
- Some of the Walsh patterns (say WR patterns) can be obtained directly from the Rademacher functions as \( \text{WR}(2^{i+1} - 1, k) = R_i(k) \) (using property 2.1)
- \( W(0,k) = \text{WR}(0,k) = +1 \), for all \( k \)
- Remaining Walsh functions are obtained from these WR patterns as follows:
  \[ w(i) = \sum_{j=0}^{n-1} g_j \text{WR}(2^{i+1} - 1) = \sum_{j=0}^{n-1} g_j R_j, \] (using property 2.4)
- While obtaining the patterns using the above equation, reduce any combinations of 3 or more Walsh functions (wherever possible) to combinations of two Walsh functions which have already been obtained in the sequence of patterns. (Using property 2.2 and 2.3)

III. Structure Of Walsh Pattern Generator:

Fig. 1. Structure of Walsh Pattern Generation Circuitry

- 128 patterns can be generated of length 128 bits each for both the channels of the 30 antennae and these are selected by combination of antenna selection bits. Clock input is from the binary counter for Walsh generator. The control bits C5-C7 are used for pattern selection on each of the channels. Each pattern is a 128 bit pattern.

IV. Architecture For Clock Circuitry:

The clock circuitry consists of 3 binary counters as follows:
- One of the counters divides the incoming clock signal frequency from the clock oscillator by \( 2^{12} \) (i.e. 4096). The divided clock signal now becomes the driving clock signal for rest of the circuit operation. The remaining two counters are 3 stage counters (counts from 0 to 127) used for dividing the above clock signal further by \( 2^1 \) or \( 2^2 \) or \( 2^3 \) or none.
Two 4:1 multiplexers are to be used for selecting any one of the clock signals for Walsh generator and noise generator, respectively.

The select lines are the control bits C2-C3 for varying the clock frequency of the noise generator and C8-C9 are the select lines for varying the clock frequency of the Walsh and Sequency generator.

Fig. 2. Architectural view of Clock Circuitry.

V. RTL Structure:

Hardware compilation consists of two distinct steps. First step is called Synthesis which is an intermediate representation of the hardware design to be produced. This step is called synthesis and the result is a representation called a netlist. Next step is synthesis by which desired circuit behavior i.e. register transfer level (RTL) is transferred into a design implementation in terms of logic gates and flip flops. The netlist contents do not depend on the particulars of the FPGA or CPLD; it is usually stored in a standard format called the Electronic Design Interchange Format (EDIF).

Fig. 3. View of Synthesis report.

Fig. 4. RTL Structure of Walsh Pattern Generation Circuitry.
VI. Simulation Result:

The simulated results are as follows: Simulation results for Walsh generator shows the output 128 bit Walsh patterns on both the channels for different selections for antenna 1. End of each pattern is marked by a new 128 bit sequency pattern.

![Fig.5. Simulation Results for Walsh Generator](image)

![Fig.6. Measured results for 62 CAL Patterns](image)

VII. Conclusion:

Walsh pattern generator is designed and physically realized using VHDL source code successfully implementing Rademacher Functions. Synthesis report and measured results using Xilinx Software shows successful implementation of Walsh Generator.

References:


