|  |
| --- |
| Functional Specification Document  High Speed Phase Measurement |

**DOCUMENT VERSION 1**

**6/30/2020**

**AUTHORS**

|  |  |  |
| --- | --- | --- |
| **Name** | **CONTACT** | |
| Pratikto Sulthoni Hidayat (PSH) | mail | Pratikto.sulthoni.h@gmail.com |
| webex | https://ykgw.webex.com/meet/pr30043653 |
| phone | +62822 3644 8958 |

**CUSTOMER**

|  |  |  |
| --- | --- | --- |
| **Name** | **Contact** | |
| Wes Blankenship (WB) | mail | Wblankenship@symbrium.com |
| Radek Puldelko (RP) | mail | [rpudelko@symbrium](mailto:rpudelko@symbirum.com).com |
| Seth Thompson (ST) | mail | sthompson@symbrium.com |

**DOCUMENT HISTORY**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Document Revision Description** | **Document Author** |
| 6/22/2020 | 0 | First draft | PSH |
| 6/30/2020 | 1 | Add: project scope, hardware test design for customer, milestone and schedule | PSH |

**APPROVALS**

|  |  |  |  |
| --- | --- | --- | --- |
| **Approval Date** | **Approved Version** | **Approver Role** | **Approver** |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

[1 Introduction 1](#_Toc43852087)

[1.1 Purpose of the document 1](#_Toc43852088)

[1.2 Project Scope 1](#_Toc43852089)

[1.3 Related documents 1](#_Toc43852090)

[1.4 Terms/Acronyms and Definitions 2](#_Toc43852091)

[2 Solution Overview 2](#_Toc43852092)

[3 Functional Specifications 2](#_Toc43852093)

[3.1 Data Acquisition 3](#_Toc43852094)

[3.2 DSP 5](#_Toc43852095)

[4 Hardware Test 7](#_Toc43852096)

[5 Materials 8](#_Toc43852097)

[6 References **Error! Bookmark not defined.**](#_Toc43852098)

[6. Open Issues 9](#_Toc43852099)

[Appendix 9](#_Toc43852100)

# Introduction

The general purpose of this project is to enhance performance of existing high-speed measurement system. The existing system use NI 6602/6612 32-bit counters serve as Data Acquisition (DAQ) hardware to receive incoming signal from four rotary encoders. However, the maximum sampling frequency of this hardware is only limited to 80 MHz. Thus, it’s restricting the maximum rotational speed that can be measured.

This project will implement FPGA based System on Chip (SOC) solution to overcome this problem. This solution has several advantages over the existing system. This solution not only served as DAQ hardware but also doing post DSP action.

## Purpose of the document

The purpose of this document is to describe the functionality of high-speed measurement system that implemented using FPGA based System on Chip (SOC) solutions.

## Project Scope

The existing system consist of three parts which are:

* DAQ

4 Channel Phase Measurement Engine

* DSP

Post DSP Actions on the Phase Measurements algorithms

* Electronic Circuit

Encoder drivers and signal conditioning. see [1].

However, the scope of this project only covers

* DAQ part

Code and test result of Phase Measurement Engine

* DSP part

Code and test result of DSP

* Hardware prototype design

This design is used for conducting simultaneous testing by both of author and customer.

## Related documents

|  |  |  |
| --- | --- | --- |
| **title** | **Description** | **link** |
|  |  |  |

## Terms/Acronyms and Definitions

|  |  |
| --- | --- |
| **Term/Acronym** | **Definition** |
| DAQ | Data Acquisition |
| Dev Board | Development Board |
| DSP | Digital Signal Processing |
| FPGA | Field Programmable Gate Arrays |
| HLS | High Level Synthesis |
| PC | Personal Computer |
| SOC | System on Chip |

# Solution Overview

This project will implement FPGA based System on Chip (SOC) solution to overcome this problem. This solution has several advantages over the existing system. First, its offer higher clock frequency hence it can increase rotational speed that can be measured. Second, FPGA based SOC is flexible. FPGA can be configured to implement any arbitrary system, including embedded processors if needed. FPGAs can also be reconfigured as often as desired, thus offering a more fundamentally flexible platform. There is virtually no risk in deploying an FPGA in applications where system upgrades are required. Third, FPGA can perform parallel computation.

# Functional Specifications



Figure . System Architecture

Figure 3.1 Describes system architecture of this project. It consists of four encoders that connected to dev board through electronic circuit board. The electronic circuit board act as encoders driver and signal conditioner. Inside the dev board there is A SOC that processing incoming signal from encoders. The dev board also connected to a PC using UART communication protocol. It used for debugging purpose.

## Data Acquisition

### Description

DAQ part is used for counting incoming pulse signal from encoders. Incremental encoder has 6 outputs: A, Anot, B, Bnot, Z, and Znot. However, this part only using A and Z as input signal. Input A used for receiving pulse train whole Z input used for triggering the measurements. As depicted in Figure 3.1, there are 4 encoders connected to dev board, thus input A and Z contains 4-bit signals. Each bit corresponding with each encoder. Input signal A and Z will be passed on to the output A and Z for DSP calculation control. The other output is Diff\_nm. This output is result of subtraction between counter n and counter m.

### Functional Requirements

#### Clock Frequency

The clock frequency in this project is top priority. The higher clock frequency, the faster rotational speed that can be measured.

The existing system used NI 6602/6612 32-bit counters as DAQ hardware. However, that clock frequency can be achieved only with 2 channels. The same card only counts to when we are looking at 4 channels. Given from formula below, the maximum rotational speed that can be achieved is with encoder resolution and using 2 channels per card.

In other hand, the FPGA based SOC have internal clock speeds exceeding depend on phase-locked loop (PLL) and mixed-mode clock manager (MMCM) configuration. Thus, from the formula above the rotational speed that can be achieved with encoder resolution is at least .

#### Behavioral Specification



Figure . DAQ Architecture

Figure 3.2 describe the architecture of DAQ part. The input ports are A and Z. Since this port connected to 4 encoders, thus input A and Z contains 4-bit signals. Each bit corresponding with each encoder. Input A used for receiving pulse train whole Z input used for triggering the measurements. As depicted in Figure 3.1, there are Input signal A and Z will be passed on to the output A and Z for DSP calculation control. The other output is Diff\_nm. This output is result of subtraction between counter n and counter m.

>>> We need to have a good power supply and Schmidt trigger type cirucuit to covert 1 Vpp sine waves from the encoders to a suitable square wave for counting.

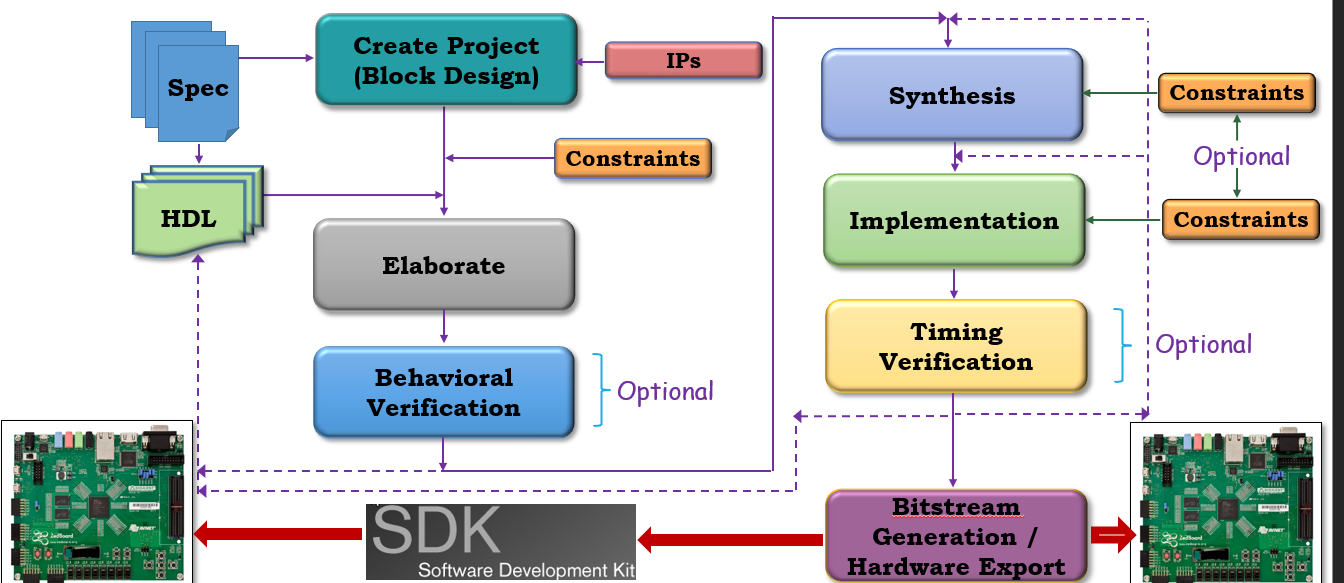
It is highly desirable to create a PCB circuit that has staqndar [1]d 1Vpp voltage inputs from a Heidenhain encoder (Something that replicates the function of an IBV 600 zero interpolation signal conditioning device:

<https://www.heidenhain.com/fileadmin/pdb/media/img/598011_en.pdf>

This circuitry is relatively straight forward and can be found in the literature. It consists of voltage supply, analog amplifier and high speed comparator with proper input impedance.

The counter board requires only A and Z as stated (Anot, B, Bnot and Znot are not used). However, it is always good to have a buffered output of the A and Z pulses available for other electronics that may want to use these same signals as a sampling clock and trigger, respectively.

### Design Procedure



### Behavioral Verification

Behavioral simulation employs a high level of abstraction to model the design. A behavioral design might, for example, contain high-level operations, such as a four-bit addition operator (this is not an adder, as in a structural design), without containing specifics on how the design will be implemented. Synthesis tools then take these behavioral designs and infer the actual gate structures and connections to be used, generating a netlist description.

Behavioral simulation is performed using a pre-synthesis Hardware Description Language (HDL) description of the design. Of the three simulation methods (behavioral, structural, and timing), behavioral simulation runs the fastest but provides the least design information.

Behavioral simulation allows you to verify syntax and functionality without timing information. During design development, most verification is accomplished through behavioral simulation. Errors identified early in the design cycle are inexpensive to fix compared to functional errors identified during silicon debug. After the required functionality is achieved, structural and timing simulation methods can be implemented to obtain more detailed verification data.

## DSP

### Purpose/ Description

DSP is used for calculating phase error algorithm.

### Functional Requirements

Functional requirement will be defined later since the algorithm is not available yet.

>> For Phase 0 – if we are able to stream the relative phase measurements to a file for post processing – this may be sufficient. Phase I would require on-board near real time processing.

### Design Procedure

DSP part will be design using High Level Synthesis (HLS) methodology. HLS is a technology that assists the transformation of a behavioral description of hardware into an RTL model. The input description is an untimed description of functionality written in C, C++ or [SystemC](https://semiengineering.com/knowledge_centers/languages/systemc/). HLS tools extract the available parallelism in the input description, schedule the operations, allocate the necessary resources and optimize the sharing of those resources to minimize the area while maintaining the necessary performance. Figure 3.3 show the design flow of HLS method.

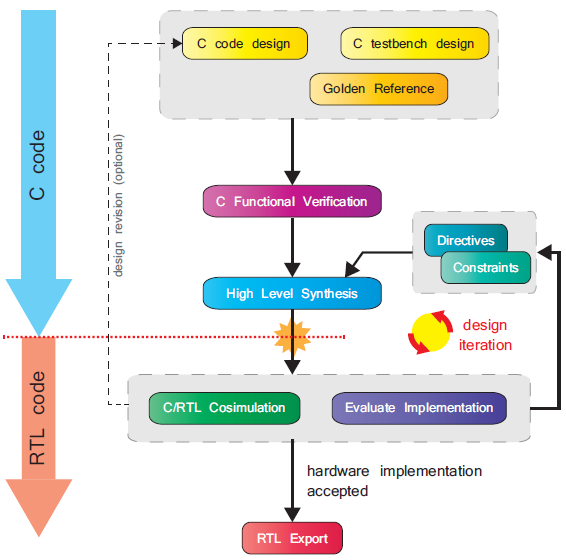


Figure . High Level Synthesis (HLS) Design Flow

### Verification

Given the importance of verification, it is useful to further detail the *C functional verifi­cation* and *C/RTL cosimulation* processes. These are described graphically in Figure 3.4.

Depicted on the left-hand side, a C-based testbench has been designed to create and supply input test vectors to the functional C module. The same test vectors are passed through a ‘known good’ golden reference design, or alternatively read from a prepared file, to give golden reference output test vectors. These are compared with the outputs from the C module, and the testbench reports a pass if the two set of results matches, or failure otherwise. The testbench may also be designed to report the total number of errors, or to provide other automated feedback on the results.

As part of the Vivado HLS C/RTL cosimulation process, an equivalent testbench config­uration is automatically created by Vivado HLS (shown on the right of Figure 14.6). The testbench verifies the RTL version of the original C module, i.e. the primary output of HLS, against the golden reference, and reports success or failure as before.

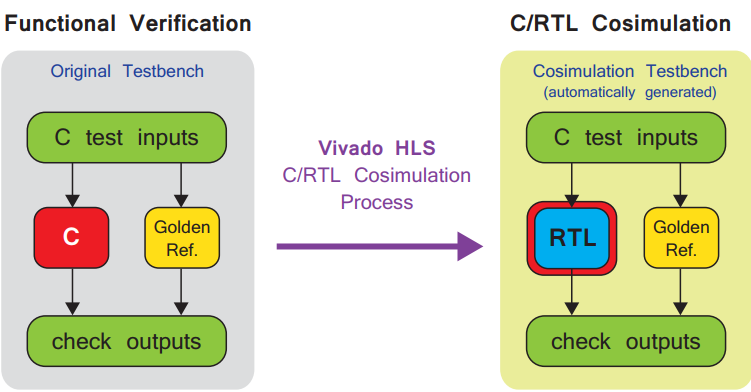


Figure . C functional verification and C/RTL cosimulation in Vivado HLS

# Hardware Test



Figure . Proposed Hardware Test Configuration for Author

Figure 4.1 show the configuration of Hardware test for author. The system will be tested again using signal coming from signal generator to verify its result. The signal generator will be replicate encoders output signal.



Figure . Proposed Hardware Test Configuration for Customer

Figure 4.2 describe the Proposed Hardware Test Configuration for customer. The system will be tested again using signal coming from customer testing machine. Level converter circuit is used for interfacing test machine output to Dev Board input since there is possibility that this device has different operating voltage.

# Materials

|  |  |
| --- | --- |
| **Hardware** | **Usage** |
| Zybo Z7: Zynq-7000 ARM/FPGA SoC Development Board | Dev board to implement DAQ and DSP part |
| Analog Discovery 2 | Signal generator to hardware testing |

|  |  |
| --- | --- |
| **Software** | **Usage** |
| Vivado 2019.2 | Synthesize and analyze HDL design |
| Vivado HLS | Synthesis rtl using HLS method |
| Xilinx SDK | Programm Zynq processor |

Note: This material is not included in bid/budget

# Project Schedule and Milestones

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Milestones | July | | August | | | | September | | | | October | | | |
| 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| DAQ FPGA design |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| DSP FPGA Design |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Hardware Test |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

# References

|  |  |
| --- | --- |
| [1] | HEIDENHEIN, Product Information IBV 600 Series Interpolation and Digitizing Electronics, 2010. |

# Open Issues

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Issue ID** | **Issue** | **Raised By** | **Raised On** | **Solution/ Decision** | **Resolved By** | **Resolved On** | **Status** |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

# Appendix