**FUNCTIONAL DESIGN SPECIFICATION FOR KӦRBER**

**Revision History**

| **Rev. No.** | **Date** | **Prepared by** | **Remarks** | **Number of Pages** |
| --- | --- | --- | --- | --- |
| Rev 0 | 27.01.2024 | Mahesh N/ Raghunath K | Original Issue |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Tech Document Approvals**

| **Name of Authority** | **Designation** | **Date** |
| --- | --- | --- |
| **Raghunath K** | RTL Manager/Principle Engineer | 27-01-2024 |
| **Suchithra N P** | RTL Engineer |
| **Mahesh N** | RTL Engineer |

**Customer/Consultant Document Approvals**

|  |  |  |
| --- | --- | --- |
| **Name of Authority** | **Company Name** | **Date** |
| D.M. Sanford | KӦRBER |  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**Table of Contents**

[Abbreviations 4](#_Toc157786605)

[1. INTRODUCTION 6](#_Toc157786606)

[General 6](#_Toc157786607)

[Scope of Project: 6](#_Toc157786608)

[2. FPGA Board Details 8](#_Toc157786609)

[2.1 Cyclone V 5CGXC5 FPGA 8](#_Toc157786613)

[2.2 Specifications of Cyclone V 5CGXC5 FPGA: 9](#_Toc157786614)

[Adaptive Logic Modules (ALM) 9](#_Toc157786615)

[4.884 Mb 9](#_Toc157786616)

[Variable Precision 9](#_Toc157786617)

[Maximum User I/O Count 9](#_Toc157786618)

[Maximum Non-Return to Zero (NRZ) Transceivers 9](#_Toc157786619)

[Maximum Non-Return to Zero (NRZ) Data Rate 9](#_Toc157786620)

[Transceiver Protocol Hard IP 9](#_Toc157786621)

[Package Options 9](#_Toc157786622)

[M301, M383, U484, F484, F672 9](#_Toc157786623)

[2.3 A 9](#_Toc157786624)

[2.4 B 9](#_Toc157786625)

[3. Block Diagram 11](#_Toc157786626)

[3.1 System Architecture 11](#_Toc157786627)

[3.2 Intel Quartus Prime 17.1 Lite Edition 14](#_Toc157786628)

[3.3 Qsys 14](#_Toc157786629)

[3.4 A 14](#_Toc157786630)

[**3.5** B 14](#_Toc157786631)

[3.6 C 14](#_Toc157786632)

[3.7 D 14](#_Toc157786641)

[4. REGISTER TRANSFER LEVEL FUNCTIONALITY (RTL) 16](#_Toc157786644)

[4.1 General 16](#_Toc157786645)

[4.2 Design Considerations 16](#_Toc157786646)

[4.3 Flow of Information in RTL System 16](#_Toc157786647)

[4.4 OVERVIEW OF ADC Controller 16](#_Toc157786648)

[4.4.1 High Speed ADC data Capture 16](#_Toc157786654)

[4.4.2 S2P 17](#_Toc157786655)

[4.4.3 24-bit Shift Register 19](#_Toc157786656)

[4.6 SPI 19](#_Toc157786657)

# 

This page intentionally left blank.

# **Abbreviations**

GENERAL

* RTL : Register Transfer Level
* HDL : Hardware Description Language
* VHDL : VHSIC Hardware Description Language
* FPGA : Field-Programmable Gate Array
* ASIC : Application-Specific Integrated Circuit

FPGA RELATED

* Soc : System on Chip
* EDS : Embedded Design Suite
* AI : Analog Input
* AO : Analog Output
* DI : Digital Input
* DO : Digital Output

COMMUNICATION PROTOCOL

* SPI : Serial Peripheral Interface
* FIFO : First in, First Out

SECTION 1

SCOPE

# **INTRODUCTION**

## General

This document details the high-level design documentation for FPGA has been introduced.

It includes the FPGA RTL design development architecture.

FPGA RTL design development architecture requirement are defined in this document. In some cases, figures of external circuitry or subsystems may/will be illustrated to help clarify.

In addition to FPGA RTL design this specification may evolve as the development process intensifies.

## Scope of Project:

SECTION 2

FPGA Overview

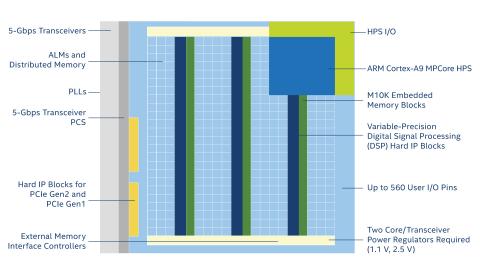
# **FPGA Board Details**



## Cyclone V 5CGXC5 FPGA

**Architecture**

* Cyclone® V FPGAs continue the Intel® Cyclone® device family tradition of an unprecedented combination of low power, high functionality, and low cost.
* The Cyclone® V FPGA now includes an optional integrated hard processor system (HPS) – consisting of processors, peripherals, and memory controller – with the FPGA fabric using a high-bandwidth interconnect backbone.
* The combination of the HPS with Intel's 28 nm low-power FPGA fabric provide the performance and ecosystem of an applications-class ARM\* processor with the flexibility, low cost, and low power consumption of the Cyclone® V FPGAs.



## Specifications of Cyclone V 5CGXC5 FPGA:

|  |  |
| --- | --- |
| **Resources** |  |
| Logic Elements (LE) | 77000 |
| Adaptive Logic Modules (ALM) | 116320 |
| Fabric and I/O Phase-Locked Loops (PLLs) | 6 |
| Maximum Embedded Memory | 4.884 Mb |
| Digital Signal Processing (DSP) Blocks | 150 |
| Digital Signal Processing (DSP) Format | Variable Precision |
| External Memory Interfaces (EMIF) | DDR2, DDR3, LPDDR2 |
| Maximum User I/O Count | 336 |
| Maximum Non-Return to Zero (NRZ) Transceivers | 6 |
| Maximum Non-Return to Zero (NRZ) Data Rate | 3.125 Gbps |
| Transceiver Protocol Hard IP | PCIe Gen1 |
| Package Options | M301, M383, U484, F484, F672 |

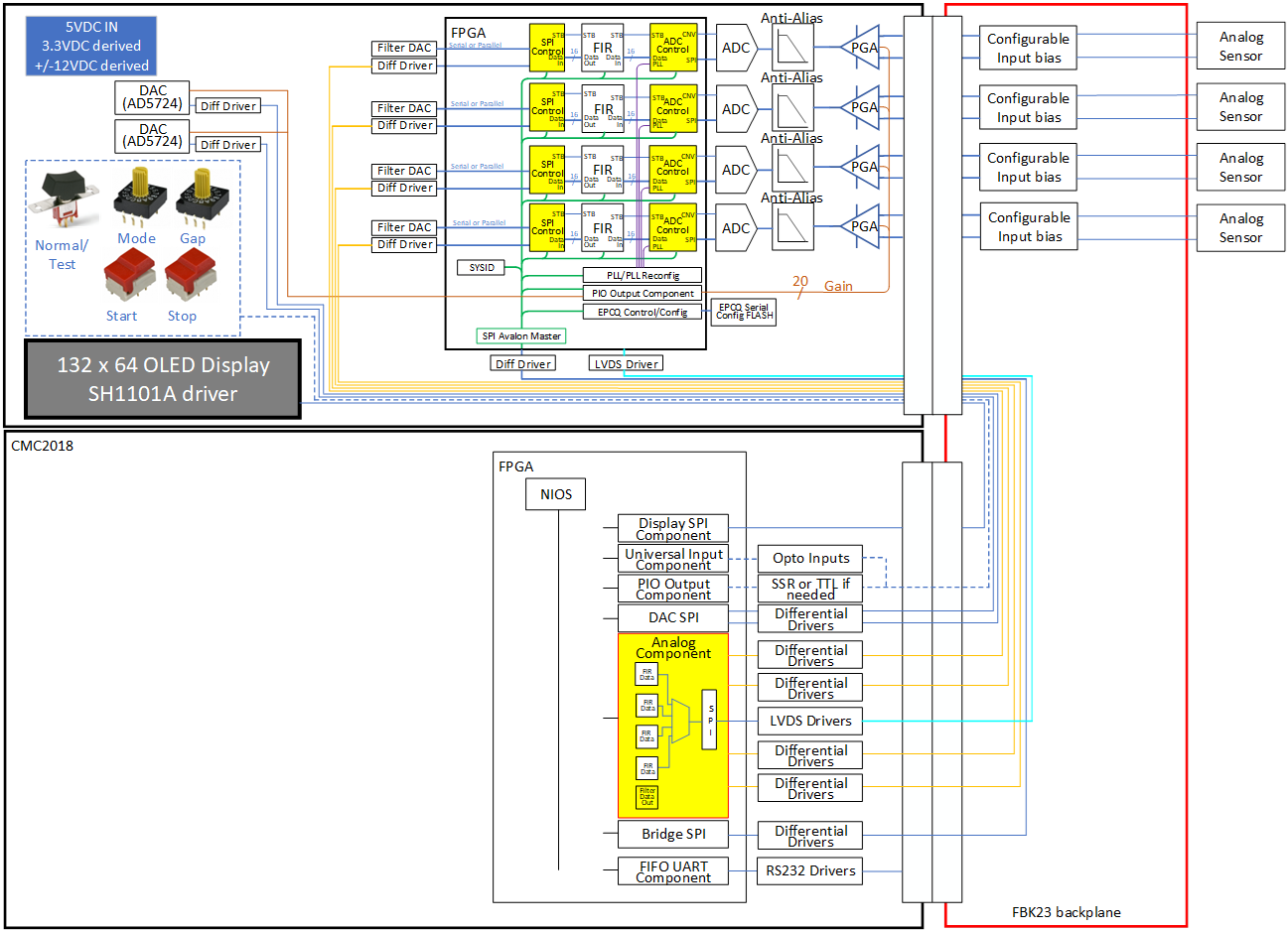
## A

## B

SECTION 3

Block Diagram

# **Block Diagram**



## System Architecture

AFE

FPGA

Clock Generator

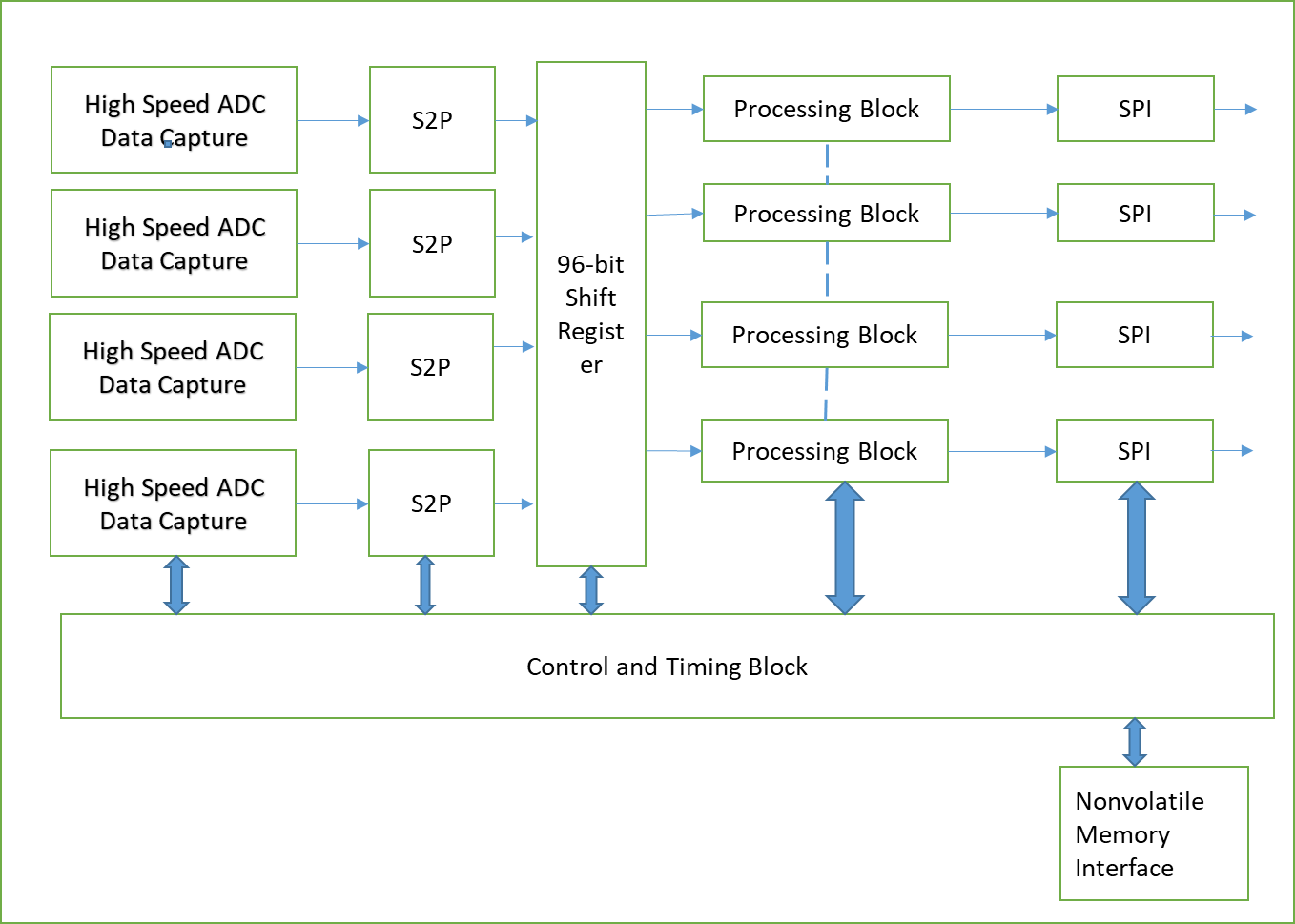
Memory

SPI Interface

Cyclone-v

Power Management

Serial Interface



High Speed ADC Data Capture

S2P

24-bit Shift Register

Processing Block

SPI

High Speed ADC Data Capture

S2P

High Speed ADC Data Capture

S2P

High Speed ADC Data Capture

S2P

Processing Block

SPI

Processing Block

SPI

Processing Block

SPI

Control and Timing Block

24-bit Shift Register

24-bit Shift Register

24-bit Shift Register

Nonvolatile Memory Interface

## Intel Quartus Prime 17.1 Lite Edition

## Qsys

## A

## B

## C



## D



SECTION 4

REGISTER TRANSFER LEVEL(RTL)

# **REGISTER TRANSFER LEVEL FUNCTIONALITY (RTL)**

# **General**

# **Design Considerations**

# **Flow of Information in RTL System**

# **OVERVIEW OF ADC Controller**



## High Speed ADC data Capture

Interfacing High speed ADC:

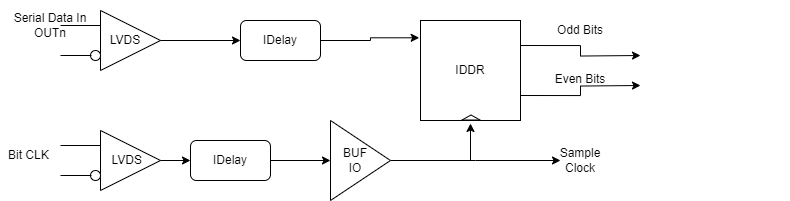
* There are various schemes of interfacing serialized low-voltage differential signaling (LVDS) data outputs from high-speed analog-to-digital converters (ADCs) to a field-programmable gate arrays(FPGAs) described below.
* The two key points required for reliable data capture
  + Bit clock edge selection
  + Frame alignment.
* Schemes for capturing data from multiple ADC devices
* One wire Interface:
  + ADC serial data rate of (Fs ×N) bits per second.
  + An associated 50% duty cycle bit clock is output with a frequency of (Fs ×N/2). The bit clock is typically center-aligned and both clock edge scan be used to latch serial ADC data. Therefore, the bit clock is referred to as a double data rate(DDR)bit clock.
  + An associated 50% duty cycle frame clock is output with a frequency of Fs. As the name suggests, the frame clock rising edge transitions are aligned with the framing ADC data bits(D0andDN-1). This alignment helps the receiver to correctly load parallel data after de-serialization. Note that the frame clock rising and falling edges are aligned with the transitions of data.
* Two-wire interface
* MSB-first
* All the above interface will have one-bit clock.
* Capturing data from a serialized ADC interface should implement two process namely
  + Latching the serialized data into S2P shift register using the ADC bit clock (serialized clock domain).
  + Aligning the parallel output data from the S2P registers correctly (parallel clock domain).
  + ***Important Note: Most FPGA shave a DDR flip-flop and register as part of the logic library. The DDR IO element accepts a single clock and registers data at the input data pin at both clock edges. The DDR IO element is also physically located close to the FPGA receiver pins, thus helping minimize any delays caused by routing in the FPGA. Therefore, TI recommends using the DDR IO element to interface to the ADC serial LVDS interface. Ideally, there is no additional skew between the serial LVDS data, LVDS bit clock, and LVDS frame clock signals caused by delays in the PCB and FPGA internal routing. In a real situation, the effects of trace delays in the PCB and FPGA routing cannot be ignored.***

## S2P

Latching Serialized ADC Data Bits into the S2P Shift Register

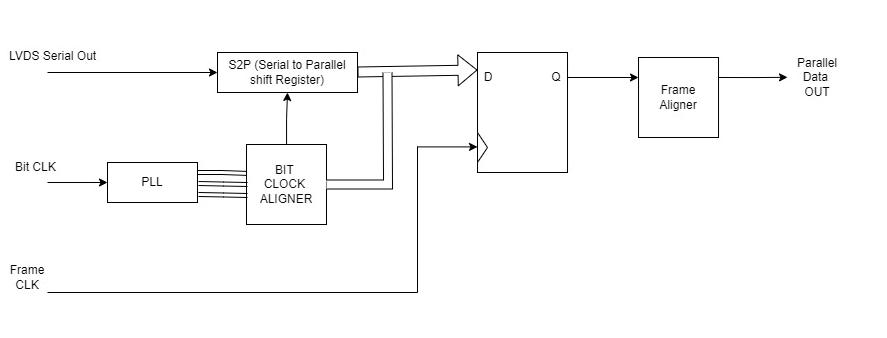
* Two methods: -

Method1: Delay elements

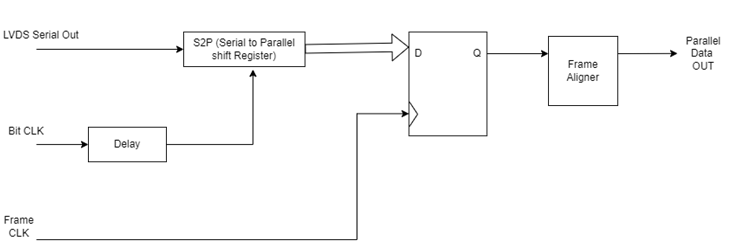


Method2: PLL Replacing PLL with various delays for controlling data

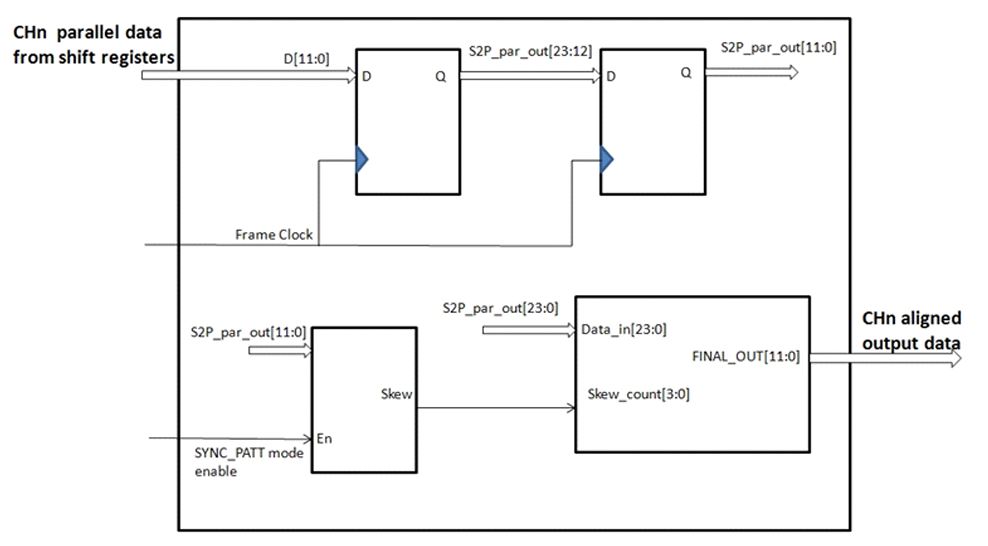
Capture Scheme Using PLL: One-Wire Interface



Capture Scheme Using Delays: One- Wire Interface



**Frame Aligner**



## 24-bit Shift Register

* 1. **Processing Block[FIR]**

# **SPI**