BPC-NDI Auxiliary Arithmetic Unit

Requirement Specification

SIGNATURES AND APPROVAL

	Name	Signature	Date
Prepared by:	Vojtěch Dvořák		1.11.2023
Checked by:			
Approved by:			

RESTRICTIONS ON USE AND DISTRIBUTION

This document contains Company Proprietary information intended only for persons with a need-to-know. Distribution, dissemination or disclosure of this information to third parties requires written authorization and shall only be used for the purpose specified in this document.

1. DOCUMENT CHANGE LOG

Issue	Date	Author	Pages	Description
1.0	11.10.2021	V. Dvorak		Initial issue
2.0	1.10.2022	V. Dvorak		Minor updates
2.1	1.11.2023	V. Dvorak		Updated requirement REQ_ AAU_I_023

2. TABLE OF CONTENT

3. TABLE OF FIGURES

4. TABLE OF TABLES

1 INTRODUCTION

1.1 Purpose and Scope of the Document

This document defines requirements for Auxiliary Arithmetic Unit as a topic for BPC-NDI semestral project.

1.2 Background of the Work

Student shall be familiar with basics of digital circuits and VHDL.

1.3 Order of Precedence

Only in specified cases.

2 APPLICABLE AND REFERENCE DOCUMENTS

The applicable and reference documents used during document preparation are listed in tables bellow. If exact date of issue is unknown the 1st day of the month is used. If either month is unknown the 1st of January is used. If revision of the document is unknown number 0 is used.

2.1 Applicable Documents

The following documents of the exact issue and/or revision shown form a part of this document to the extend specification herein. Where no issue is shown the latest issue is applicable.

Ref	Description	Doc. Number	Date	Issue
AD01	-	-		-
AD02	-	-		-

Table 2-1 Applicable Documents

2.2 Reference Documents

The following documents are for reference and/or guideline only.

Ref	Description	Doc. Number	Date	Issue
RD01	-	-		-
RD02	-	-		-

Table 2-2 Reference Documents

3 DEFINITIONS, ABBREVIATIONS AND ACRONYMS

3.1 Definitions

For the objective-requirement specification the following terms are used:

- The word “shall” is used to indicate a mandatory requirement
- The word “should” and “may” express non-mandatory provisions.

Each requirement is composed of:

- Unique Identifier of the structure “REQ-AAU-*x*-*yyy*”, where
  x corresponds to group of requirements (G – General requirements, F – functional requirements, I – interface requirements)
  yyy is a unique 3-digit number
- Unique Name
- Parrent Req. - identifies the parent requirement or the applicable documents from which the objective requirement comes
- Verification Method – determines one of the methods to verify the objective requirement
  - Review of Design [R]
  - Analysis [A]
  - Simulation on RTL level [S]
- Description of the object requirement

3.2 Signal active value

If not specified otherwise, all one-bit signals (e.g. enable signals) are active HIGH (log 1).

3.3 Number writing

Numbers in the document are written in these number systems:

- Decimal (integer) – decimal number, i.e. 123
- Hexadecimal – number starts with 0x
- Binary – number is in a quotation marks
  - single bit ' '
  - vector of bits “ ”

3.4 Finite State Machine Diagrams

In this document, the finite state machines with the conditional transitions between states and the outputs are all visually described by state diagrams. To improve readability of the state diagrams, if conditional transitions from a state A to other states are depicted in a state diagram, the state machine remains in that state A until one of the conditions for the transitions to the other states is true.

3.5 Units

Only SI units are used.

3.6 Abbreviations

AAU	Auxiliary Arithmetic Unit
BUT	Brno University of Technology
FPGA	Field Programmable Gate Array
I/F	Interface
LSb	Least Significant Bit
LSB	Least Significant Byte
MCU	Microcontroller Unit
MTBF	Mean Time Between Failures
SPI	Serial Peripheral Interface
TBC	To Be Confirmed
TBD	To Be Defined
-	-

4 AUXILIARY ARITHMETIC UNIT OVERVIEW

The Auxiliary Arithmetic Unit (AAU) is intended to be used as an extension unit for simple microcontroller (MCU) to perform arithmetic operations. The AAU is connected to the MCU via Serial Peripheral Interface (SPI), where MCU is master and AAU is slave in link schema. Data are transmitted through the link from master to slave and the AAU autonomously return results of arithmetic operations in the next packet. Two arithmetic operations are performed by AAU, addition and multiplication while numbers are represented in fixed-point.

While the data transfer is driven by the master unit, it is still possible some errors might appear on the link and completeness of incoming data must be checked prior to perform arithmetic operations and sending out the results.

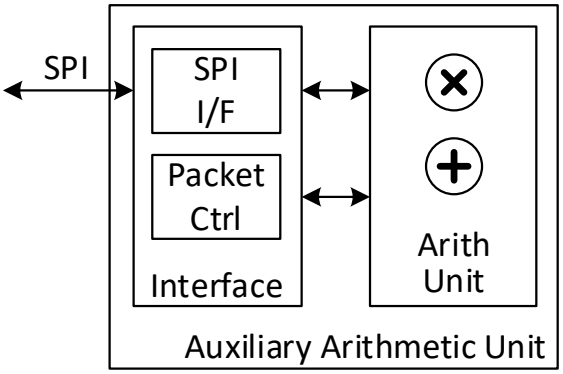

Figure 4-1 Top Level overview of Auxiliary Arithmetic Unit module

5 GENERAL REQUIREMENTS

General requirements put constraints on implementation of the other requirements. Most of these requirements are verified only by review and compliance of the design to them must be clearly shown in the documentation.

REQ_AAU_G_001	**Target technology**
Parent Req: none	Verification: R
The Auxiliary Arithmetic Unit shall be designed for FPGA Xilinx Spartan 3.	

REQ_AAU_G_002	**Synchronous design**
Parent Req: none	Verification: R
Design shall be fully synchronous, only one clock in the whole design is allowed. *Comment: Only statement "if rising_edge(clk) then" can be in the code!*	

REQ_AAU_G_003	**Output signals**
Parent Req: none	Verification: R,(S)
All output signals shall be glitch-free.	

REQ_AAU_G_004	**Input signals**
Parent Req: none	Verification: R,(S)
All input signals are asynchronous in nature and proper synchronization technique shall be implemented to ensure no metastability issues can appear.	

REQ_AAU_G_005	**FSM Safe Implementation**
Parent Req: none	Verification: R,(S)
All FSMs should be protected against environmental effects such as SEE to ensure no FSM can stuck in non-operational state.	

REQ_AAU_G_006	**Documentation**
Parent Req: none	Verification: R

As a minimum, following points shall be documented:

- Development flow with flowchart
- Description of design with signal flow diagram
- Verification plan with verification matrix
- Verification description
- Verification report
- Implementation report such as used resources, maximal operational frequency and MTBF

6 FUNCTIONAL REQUIREMENTS

Functionality of the unit is defined here. Since the AAU has only one function, which is to perform arithmetic operations, functional requirements are mainly related to arithmetic operations.

REQ_AAU_F_010	**Auxiliary Arithmetic Unit**
Parent Req: none	Verification: R
The Arithmetic Unit shall perform two operations with signed numbers, addition and multiplication.	

REQ_AAU_F_011	**Format of numbers**
Parent Req: none	Verification: S
The AAU shall perform operations in fixed-point arithmetic with 16 bit numbers, where whole part is 8 bits and fractional part is 8 bits. This is applicable on both input and output numbers of AAU. *Note: Negative numbers are implemented in 2nd complements.*	

REQ_AAU_F_012	**Number rounding**
Parent Req: none	Verification: S
Where rounding of results of arithmetic operations is necessary, *floor* method shall be used. *Note: Floor method is also called rounding to minus infinity.*	

REQ_AAU_F_013	**Overflow of arithmetic operations**
Parent Req: none	Verification: S
When result of arithmetic operation overflows (i.e. result is outside of range of number representation), the result shall be saturated to maximum positive or negative value (0x7FFF, 0x8000).	

7 INTERFACE REQUIREMENTS

Interface to transmit data to and from the AAU is Serial Peripheral Interface (SPI). This link two has control signals (CS_b, SCLK) and two data signals (MOSI, MISO). Typically, link has two participants, *master* and *slave*. Master has control over the link and defines when communication is performed via pair of control signals. In this project, the AAU serves as a slave and accepts control signals and data sent by master (Master-Out-Slave-In, MOSI). When transaction on link is started, slave transmits its data via dedicated data signal (Master-In-Slave-Out, MISO).
Signal sequence of SPI interface is shown on figure 7-1. Master starts data transfer by forcing signal CS_b to '0' followed by falling edge of SCLK. Data from master to slave (MOSI) is set on falling edge of SCLK. Data from slave (MISO) is sampled at rising edge of SCLK.

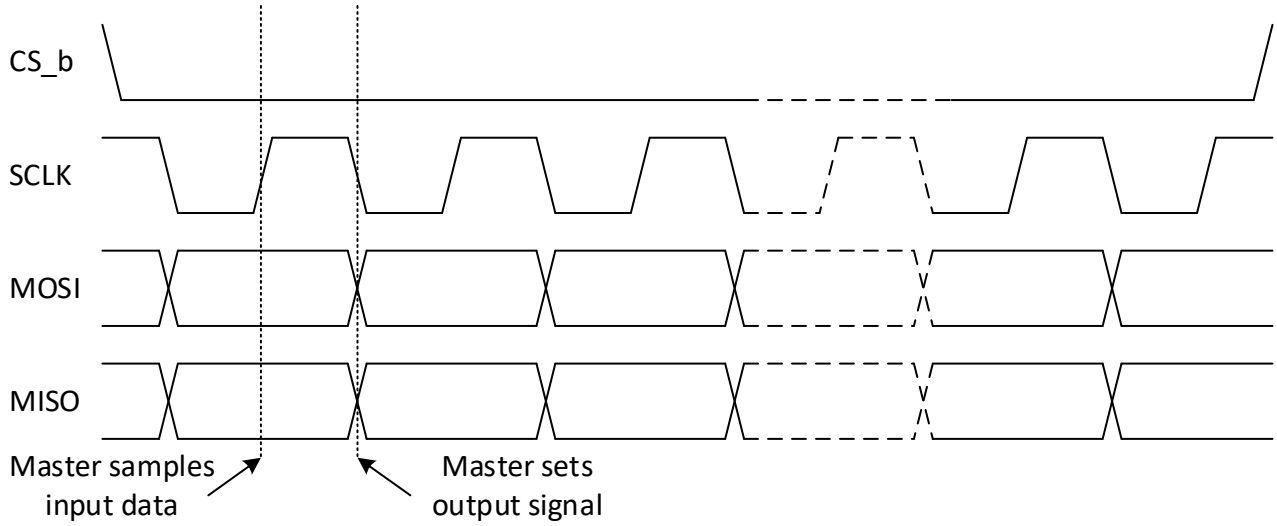

Figure 7-1 SPI link

For purpose of this project, elementary data chunk is called *frame*. One frame is one number and it is defined by events on CS_b signal, where falling edge marks start of the frame and rising edge of CS_b identifies end of the frame. Full transaction is performed with *packets*, where one packet is composed of two frames. In data transfer, master sends data to slave in one packet and results are expected in the next packet.

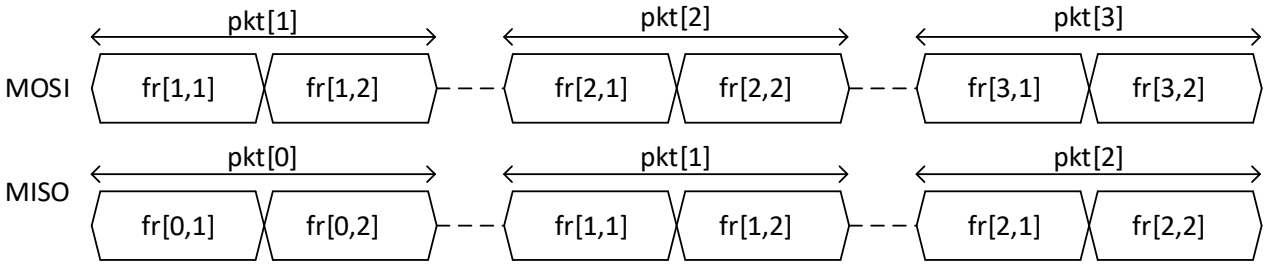

Figure 7-2 Packet transfer

REQ_ AAU_I_020	**SPI clock frequency**
Parent Req:	Verification: S
The AAU interface shall be compatible with SPI clock (SCLK) 10 kHz, 100 kHz and 1 MHz.	

REQ_ AAU_I_021	**Bit ordering**
Parent Req:	Verification: S
In data transfer, LSb shall be sent first.	

REQ_ AAU_I_022	**Incomplete frame**
Parent Req:	Verification: S
Frame with wrong number of bits shall be ignored.	

REQ_ AAU_I_023	**Link reset**
Parent Req:	Verification: S
When reception of the second frame of a packet is not started in 1ms after the first frame was received, such packet shall be considered invalid.	

REQ_ AAU_I_024	**Packet format**
Parent Req:	Verification: S
Arithmetic results shall be sent out to master in the following order: 1. Sum 2. Product	