| Owner | | Document | |
| --- | --- | --- | --- |
| Name |  | Project |  |
| E-mail |  | Filename |  |

# Abstract

*Write a short description of the project*

Table of Contents

[1 Introduction 3](#_Toc434002957)

[1.1 Background Reading 3](#_Toc434002958)

[1.2 Glossary 3](#_Toc434002959)

[1.3 Design Library 3](#_Toc434002960)

[1.4 People Involved 3](#_Toc434002961)

[2 General Description 4](#_Toc434002962)

[3 Pin List 5](#_Toc434002963)

[4 Functional Description 6](#_Toc434002964)

[4.1 Overview 6](#_Toc434002965)

[4.2 Procedural Diagram 6](#_Toc434002966)

[4.3 Detailed Description 6](#_Toc434002967)

[4.3.1 Subblock A 6](#_Toc434002968)

[4.3.2 Subblock B 6](#_Toc434002969)

[4.3.3 Subblock C 6](#_Toc434002970)

[5 Implementation 7](#_Toc434002971)

[5.1 <Subblock\_A> (<SUBA>) Subblock 7](#_Toc434002972)

[5.1.1 Pin List 7](#_Toc434002973)

[5.1.2 Interface Diagram 7](#_Toc434002974)

[5.1.3 Description 7](#_Toc434002975)

[5.1.4 <SUBA> Finite State Machine 8](#_Toc434002976)

[5.2 <Subblock\_B> (<SUBB>) Subblock 8](#_Toc434002977)

[5.3 <Subblock\_C> (<SUBC>) Subblock 8](#_Toc434002978)

[6 Verification 9](#_Toc434002979)

[6.1 List of Tests 9](#_Toc434002980)

[6.2 Detailed Test Description 9](#_Toc434002981)

[6.2.1 Test A 9](#_Toc434002982)

[6.2.2 Test B 9](#_Toc434002983)

[6.2.3 Test C 9](#_Toc434002984)

# Introduction

## Background Reading

[1]

[2]

## Glossary

i. e. UART [Universal Asynchronous Receiver Transmitter](https://en.wikipedia.org/wiki/Universal_asynchronous_receiver/transmitter)

## Design Library

Add design library, <top block>\_lib i.e. uart\_lib

## People Involved

| Name | E-mail address |
| --- | --- |
|  |  |
|  |  |
|  |  |

# General Description

Write a general description of the block. It should include:

* Some theory on where this block is based
* Key equations
* Simple description of the block partitioning

# Pin List

| Signal | | I/O | | Description |
| --- | --- | --- | --- | --- |
| Clock & Reset | | | | |
| clk | in | | System clock. | |
| rst\_n | in | | Global asynchronous active low reset. | |
| **Interface 0** | | | | |
| i\_signal0 | in | | Describe input signal. | |
| **Interface 1** | | | | |
| i\_signal1 | in | | Describe input signal. | |
| i\_signal2 | in | | Describe input signal. | |
| o\_signal0 | out | | Describe output signal. | |
| **Interface 2** | | | | |
| i\_signal3 | in | | Describe input signal. | |
| i\_signal4[7:0] | in | | Describe input vector. | |
| **Interface 3** | | | | |
| o\_signal1[31:0] | out | | Describe output vector. | |
| o\_signal2 | out | | Describe output signal. | |

*Table 3.1:*

# Functional Description

## Overview

The diagram of Figure 4.1 illustrates a single instantiation of <top block>. (This is the <top block> diagram where the Input/Output pins described in Table 3.1 are shown)

Figure 4.1:

## Procedural Diagram

Figure 4.2 depicts the internal subblocks of the <top block>. (those are the subblocks that the <top block> diagram is divided. i.e. <Subblock A>, <Subblock B>… You must show the Input/Output interface of the top block plus the internal interfaces of the subblocks )

Figure 4.2:

## Detailed Description

### Subblock A

Analyze the functions implemented in <Subblock A>.

Then analyze timing relationships (Wave diagram)

Figure 4.3: Timing relationships

### Subblock B

As above

Figure 4.4: Timing relationships

### Subblock C

As above

Figure 4.5: Timing Relationships

# Implementation

The following paragraphs describe the implementation of the <top block>. Figure 4.2 shows the interconnections of the subblocks included in the <top block>. **NB: The example analysed in this document does not correspond to a verified design; it is written in such way as to show how a designer could efficiently analyze a block.**

## <Subblock\_A> (<SUBA>) Subblock

### Pin List

| Signal | I/O | Description |
| --- | --- | --- |
| Clock & reset | | |
| clk | in | System clock. |
| rst\_n | in | Global asynchronous active low reset. |
| **Input interface** | | |
| i\_ip0 | in | Describe input signal. |
| i\_ip3 | in | Describe input signal. |
| i\_s1\_sig0 | in | Describe input internal signal. |
| **Output interface** | | |
| o\_s0\_sig0 | out | Describe output internal signal. |
| o\_s0\_sig1 | out | Describe output internal signal. |

*Table 5.1: <SUBA> Subblock Pin List*

### Interface Diagram

Draw the <SUBA> with its own Input/Output pins and the Subblocks that interact with.

Figure 5.2: <SUBA> Subblock Interface Diagram

### Description

Give a short description of what the subblock does.

Then start analysing EVERY BIT OF CODE you intend to write. Each signal generation or assignment MUST be explained and shown within a figure.

A first example can be the setting of an S-R register *cnt\_en*, which is set on assertion of input *i\_ip0* and reset when *i\_ip0* is deasserted and an internal counter *cnt\_q* becomes zero, as shown in Figure 4.3. Note that registers MUST ALWAYS be named and the resulting signals are named as *<register\_name>\_q*, where “\_q” denotes the output of this register. Also notice the white box, indicating how the register is reset. This should be included in all figures containing registers or synchronous logic.

Figure 5.3: Counter Enable Register

Since cnt\_q is used above, you need to describe this counter. Figure 4.4 is an example, showing a 4-bit counter. This counter is counting while cnt\_en\_q (above) is set and will count until it reaches the value 0xF. Then it is reset back to 0x0. Note also that when cnt\_q = 0xF, output o\_s0\_sig0 is set (driven by register s0\_sig1).

Figure 5.4: Counter Circuit Diagram

Finally, describe the generation of the remaining output *o\_sig0\_s1*. Figure 4.5 shows an example where register *s0\_sig1* (driving *s0\_sig1\_q* and hence *o\_s0\_sig1*) is set when input *i\_s1\_sig0* is high while *cnt\_en\_q* is also high.

Figure 5.5: Output o\_s0\_sig1 Generation

### <SUBA> Finite State Machine

This subblock could introduce a state machine, as depicted in Figure 5.6.

Figure 5.6: <SUBA> Subblock State Transition Diagram

State description:

* idle – the reset state of the FSM.
* state0 – give a brief description of state0.
* state1 – give a brief description of state1.
* state2 – give a brief description of state2.
* state3 – give a brief description of state3.

FSM operation:

Describe how this FSM works. Analyse all the transitions and state what happens in each case (e. g. what the next transition is on state1, depending *on i\_s0\_sig0*. Make sure that all the transitions are fully explained. Don't forget the white box in the diagram, explaining what happens on reset.

## <Subblock\_B> (<SUBB>) Subblock

As above

## <Subblock\_C> (<SUBC>) Subblock

As above

# Verification

## List of Tests

* Test A – Verification of <block name> function.
* Test B – Verification of <block name> function.
* Test C – Verification of <block name> function.

| Test | Description |
| --- | --- |
| Test A -<testname.v> | Short description |
| Test B-<testname.v> | Short description |
| Test C-<testame.v> | Short Description |

*Table 6.1: List of Tests*

## Detailed Test Description

### Test A

**The goal of this test is:** Describe the overall goal of this test

Unit under test: <module that you are testing>

input vectors (aka inputs to the unit under test)

output vectors(aka outputs from the unit under test)

Describe what the test does.

### Test B

As Above

### Test C

As Above