cadence

UART

Verification Plan

Version 1.0

Cadence Design Sytems, Inc., 555 River Oaks Parkway, San Jose, CA 95134, USA

Cadence Trademarks

Trademarks and service marks of Cadence Design Systems, Inc. (Cadence) contained in this document are attributed to Cadence with the appropriate symbol. For queries regarding Cadence's trademarks, contact the corporate legal department at the address above or call 800.862.4522.

1st Silicon Success®
Allegro®
Assura™
BuildGates®
Cadence® (brand and logo)
CeltIC™
ClockStorm®
CoBALT™
Conformal®
Connections®
Design Foundry®
Diva®
Dracula®
Encounter™
Fire & Ice®
First Encounter®
FormalCheck®
HDL-ICE®
Incisive™
IP Gallery™
Nano Encounter™
NanoRoute™
NC-Verilog®
OpenBook® online documentation library
Orcad®
Orcad Capture®
Orcad Layout®
PacifIC™
Palladium™
Pearl®
PowerSuite™
PSpice®
QPlace®
Quest®
SeismIC™
SignalStorm®
Silicon Design Chain™
Silicon Ensemble®
SoC Encounter™
SourceLink® online customer support
Specman
Spectre®
TtME®
UltraSim®
Verifault-XL®
Verilog®
Virtuoso®
vManager
VoltageStorm®

Other Trademarks

Open SystemC, Open SystemC Initiative, OSCI, SystemC, and SystemC Initiative are trademarks or registered trademarks of Open SystemC Initiative, Inc. in the United States and other countries and are used with permission.
All other trademarks are the property of their respective holders.

Confidentiality Notice

Introduction

This document defines the scope of the UART block level verification and specifies its solution. The verification problem is quantified in a set of coverage models derived from architectural and implementation features. The solution is specified in the functional specification of the simulation verification environment in section 4, "Verification Environment Design".

The UART is an Advanced Microcontroller Bus Architecture (AMBA) compliant System On Chip block peripheral. The key functions of this block are detailed below:

" Control and data access is provided by an Advanced Peripheral Bus (APB) bus interface

" APB access to Tx and Rx FIFO queues

" Interrupt Control

" Highly configurable Serial Interface

" Support for IrDA, Modem

Further technical information on the UART design may be found in the UART User Guide.

This document refers in several sections to serial data frames, for the context of this document a frame shall be considered as a single data word transmission (start bit, 6,7,8 bit data payload, optional parity bit and stop bit(s)).

The APB interface shall be tested in conjunction with an AHB/APB bridge, requiring therefore verification at an APB interface. The Design Under Verification (DUV) is depicted in Figure 1 - DUV Block Diagram.

The table below imports the APB Compliance vPlan..

Module Name	Verification Plan
SLAVE_APB	sv_cb_ex_lib/interface_uvc_lib/apb/vpm/APBvPlan.xml
SERIAL_UART	sv_cb_ex_lib/interface_uvc_lib/uart/vpm/UARTvPlan.xml

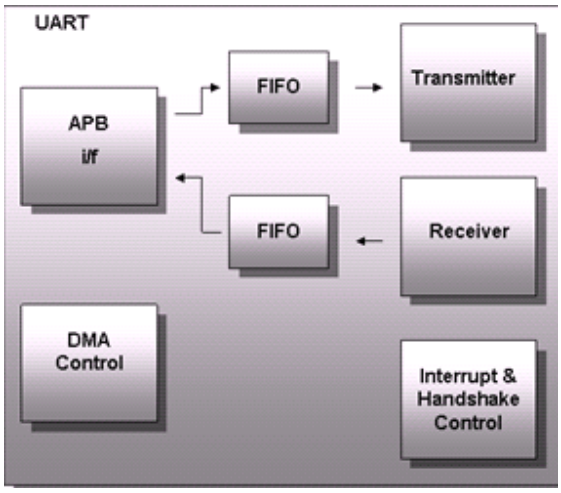

Figure 1 – DUV Block Diagram

IMPORTANT - The following features of the UART design shall not be tested:

- Fractional Baud Rate setting
- IrDA Serial Link Features
- Modem features
- Test Registers

Parameter Name	Parameter Values	Description
COV_CHK_MODELS	APB_IF_UVC, UART_IF_UVC, UART_ABV	The kind of metrics that is used in the project that instantiates this plan.

1 UART APB Verification Plan

This section identifies the features of the UART and describes the functional specification for each feature. Feature-related implementation details are also described when appropriate. Section weights, goals and required number of samples are included here or within a perspective.

1.1 Functional Interfaces

1.1.1 APB Interface

Parameters: COV_CHK_MODELS[*APB_IF_UVC*]

The UART is controlled from an APB interface via an AHB/APB bridge; this interface is compliant to the AMBA Specification (Rev2.0).

1.1.1.1 Legal Interface Behavior

The design implementation restricts accesses to be single beat with 32 bit data width. Split and Retry responses are not supported.

The table below details the coverage to be used from the APB Compliance vPlan.

Name	Path	Parameters	Instance Match	Actual Instances
APB_instance	SLAVE_APB::APB Compliance Plan			APB_BUS_MONITOR[APB_BUS_MONITOR]

1.1.2 UART Interface

Parameters: COV_CHK_MODELS[*UART_IF_UVC*]

The UART transmits and receives serial data. The data formats and speed are configurable to match those of an industry standard 16c550a UART.

1.1.2.1 Legal Interface Behavior

Name	Path	Parameters	Instance Match	Actual Instances
UART Interface	SERIAL_UART::UART Compliance Plan	Function_TYPE [ACTIVE]		UART_AGENT[UART_AGENT]

1.2 Black Box Behavior

Logical Instances: UART_APB_MONITOR

1.2.1 Serial Data FIFOs

The UART provides the option of the use of data FIFOs, with separate FIFOs for Rx and Tx data..

1.2.1.1 Rx FIFO

The Rx FIFO is a 12 bit wide data store with a depth of 16. The width is 12 bits because it contains 8 bits of data and 4 error/status bits.

Name	Item Pattern	Bucket Filter	Parameters
Rx FIFO level	UART_APB_MONITOR.monitor.uart_cover.dut_rx_fifo_cg.*		

1.2.1.2 Tx FIFO

The Tx FIFO is an 8 bit wide data store with a depth of 16.

Name	Item Pattern	Bucket Filter	Parameters
Tx FIFO level	UART_APB_MONITOR.monitor.uart_cover.dut_tx_fifo_cg.*		

1.3 White Box Behaviour

```
Logical Instances:      UART_DUT
```

There are no implementation specific features of the DUV.

1.3.1 Code Coverage

All module instances require code coverage

```
Cover group:      uart*(block)
Cover group:      uart*(expression)
```

1.3.2 UART Metrics

Assertion coverage collection from the UART DUT

Name	Item Pattern	Bucket Filter	Parameters
state is sr_idle and (srx_pad_i==1'b0 & ~break_error), next state will be "sr_rec_start"	UART_DUT.receiver.core_sr_idle_to_sr_rec_start		
current state is, sr_rec_bit and rcounter16_eq_0, next state is sr_end_bit	UART_DUT.receiver.core_sr_rec_bit_to_sr_end_bit		
current state is sr_end_bit and rbit_counter is zero, next state issr_rec_parity || sr_rec_stop	UART_DUT.receiver.core_sr_end_bit_to_next_state		
At reset output rf_count should be zero, this indicates rx fifo is empty	UART_DUT.receiver.output_rf_count_zero		

When RX fifo is reset count should be zero	UART_DUT.receiver.fifo_rx.output_count_zero		
Fifo overrun condition	UART_DUT.receiver.fifo_rx.output_overrun_high		
If TX is enabled, but tf_count is zero, state m/c stays in s_idle state	UART_DUT.transmitter.core_s_idle		
If current state is s_idle, stx_pad_o should be 1	UART_DUT.transmitter.core_stx_pad_o_high		
tf_pop is always of one clock cycle	UART_DUT.transmitter.core_tf_pop_pulse		
When TX fifo is reset count should be zero	UART_DUT.transmitter.fifo_tx.output_count_zero		

1.4 Topology

The topology of the verification environment is depicted below.

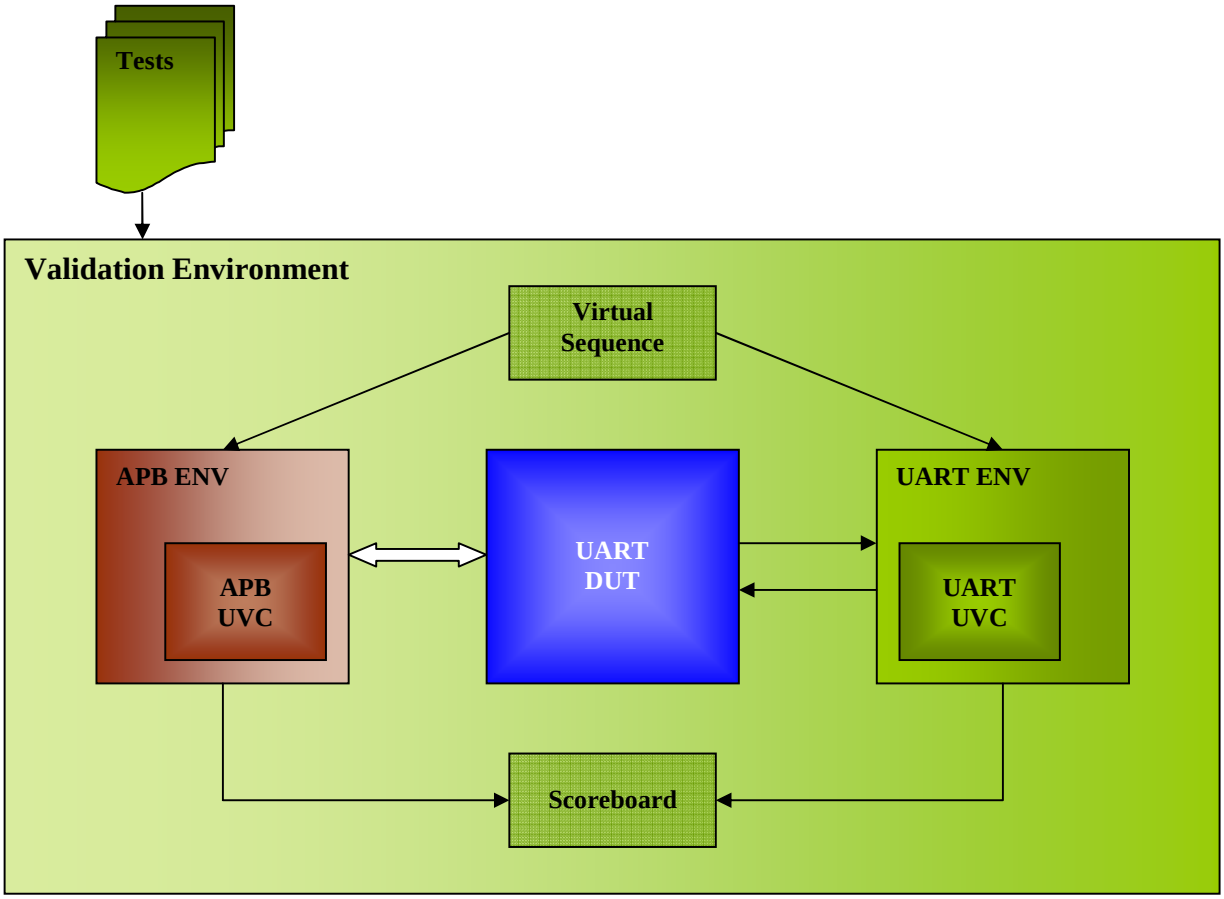

Figure 2 - Verification Topology