

cocotbext APB



March 31, 2025

Jay Convertino

# Contents

<b>1 Usage</b>	<b>2</b>
1.1 Introduction . . . . .	2
1.2 Dependencies . . . . .	2
1.3 In a Simulation . . . . .	2
<b>2 Architecture</b>	<b>2</b>
2.1 Directory Guide . . . . .	3
<b>3 Simulation</b>	<b>4</b>
3.1 cocotb . . . . .	4
<b>4 Code Documentation</b>	<b>5</b>
4.1 init . . . . .	6
4.2 monitor . . . . .	7
4.3 driver . . . . .	9
4.4 absbus . . . . .	12
4.5 busbase . . . . .	15
4.6 test extension python . . . . .	21
4.7 test extension verilog . . . . .	22

# 1 Usage

## 1.1 Introduction

Cocotb extension to test APB3 bus master, and slave devices.

## 1.2 Dependencies

The following are the dependencies of the cores.

- iverilog (simulation)
- cocotb (simulation)
- cocotb-bus (simulation)

## 1.3 In a Simulation

Below is a simple example for reading and writing data from register zero in the cocotb extension.

```
master = apb3Master(dut, "apb", dut.clk, dut.rstn)
slave = apb3EchoSlave(dut, "apb", dut.clk, dut.rstn)

await master.write(0, 0xAAAAAAAA)

rx_data = await master.read(0)

assert 0xAAAAAAAA == rx_data, "RECEIVED_DATA_DOES_NOT_
    ↳ MATCH"
```

# 2 Architecture

Please see 4 for more information.

apb3Master tests APB3 slave devices by executing read/write requests from the python test bench.

apb3EchoSlave provides a simple slave that will echo all register writes back over read when requested.

apb3Monitor tests to make sure signals are proper. Simple core at the moment, only checks for 0 at rest and if the penable is correct per pselect.

## 2.1 Directory Guide

Below highlights important folders from the root of the directory.

1. **docs** Contains all documentation related to this project.
  - **manual** Contains user manual and github page that are generated from the latex sources.
2. **cocotbext** Contains source files for the extension
  - **apb.three** Contains source files for the APB version three extension.
3. **tests** Contains test files for cocotb

## 3 Simulation

A simulation for testing the cores can be run to verify operation.

### 3.1 cocotb

To use the cocotb tests you must install the following python libraries.

```
$ pip install cocotb  
$ pip install -e .
```

Then you must enter the tests folder and enter the mil-std-1553 folder. From there you may execute the following command which will kick off the test.

```
$ make
```

## 4 Code Documentation

Natural docs is used to generate documentation for this project. The next lists the following sections.

- **init** Python init code.
- **monitor** Contains bus monitor code.
- **driver** Contains bus driver code.
- **absbus** Contains bus abstraction for monitor, and driver code.
- **busbase** Contains bus base for threads and read/write methods.
- **cocotb test** Python TestFactory code.
- **cocotb verilog test wrapper** Verilog wrapper module.

**\_\_init\_\_.py**

---

## **AUTHORS**

---

**JAY CONVERTINO**

---

## **DATES**

---

**2025/03/06**

---

## **INFORMATION**

---

### **Brief**

---

apb define for packages

### **License MIT**

---

Copyright 2025 Jay Convertino

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

\*\*\*\*\*

Copyright (c) 2020 Alex Forencich

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

\*\*\*\*\*

# monitor.py

---

## AUTHORS

---

JAY CONVERTINO

---

## DATES

---

2025/03/11

---

## INFORMATION

---

### Brief

---

Monitor for APB3

### License MIT

---

Copyright 2025 Jay Convertino

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## apb3Monitor

---

apb3Base

apb3Monitor

Check signals to make sure they are applied properly.

## FUNCTIONS

---

init

---



```
def __init__(
    self,
    entity,
    name,
    clock,
    resetn,

    args,
    kwargs
)
```

Setup defaults and call base class constructor.

---

## **`_check_type`**

```
def _check_type(
    self,
    trans
)
```

Check and make sure we are only sending apb3trans, this is only here to satisfy the need to have it.

---

## **`_run`**

```
async def _run(
    self
)
```

`_run` thread that deals with checking signals, simple check for now.

# driver.py

---

## AUTHORS

---

JAY CONVERTINO

---

## DATES

---

2025/03/11

---

## INFORMATION

---

### Brief

---

Bus Driver for APB3

### License MIT

---

Copyright 2025 Jay Convertino

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## apb3Master

---

apb3Base

apb3Master

Drive slave devices over the APB3 bus

## FUNCTIONS

---

init

---

```
def __init__(
    self,
    entity,
    name,
    clock,
    resetn,
    args,
    kwargs
)
```

Setup defaults and call base class constructor.

## read

---

```
async def read(
    self,
    address
)
```

Read from a address and return data

## write

---

```
async def write(
    self,
    address,
    data
)
```

Write to a address some data

## \_check\_type

---

```
def _check_type(
    self,
    trans
)
```

Check and make sure we are only sending 2 bytes at a time and that it is a bytes/bytearray

## \_run

---

```
async def _run(
    self
)
```

\_run thread that deals with read and write queues.

# apb3EchoSlave

---

apb3Base

apb3EchoSlave

Respond to master reads and write by returning data, simple echo core.

## FUNCTIONS

---

### \_\_init\_\_

```
def __init__(
    self,
    entity,
    name,
    clock,
    resetn,
    numreg
    =
    256,
    args,
    kwargs
)
```

Setup defaults and call base class constructor.

### \_check\_type

```
def _check_type(
    self,
    trans
)
```

Check and make sure we are only sending a type of apb3trans.

### \_run

```
async def _run(
    self
)
```

\_run thread that deals with read and write request over bus.

## absbus.py

---

### AUTHORS

---

JAY CONVERTINO

---

### DATES

---

2025/03/11

---

### INFORMATION

---

#### Brief

---

abstraction of the apb3 bus

#### License MIT

---

Copyright 2025 Jay Convertino

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## apb3trans

---

transaction

apb3trans

create an object that associates a data member and address for operation.

## apbState

---

enum.IntEnum

apbState

An enum class that provides the current state and will change states per spec.

## apb3Base

busbase

apb3Base

apb3EchoSlave

apb3Master

apb3Monitor

abstract base class that defines apb3 signals

## VARIABLES

### \_signals

\_signals

List of signals that are required

### \_optional\_signals

\_optional\_signals

List of optional signals, these will never be required but will be used if found.

## FUNCTIONS

### \_\_init\_\_

```
def __init__(
    self,
    entity,
    name,
    clock,
    resetn,

    args,
    kwargs
)
```

Setup defaults and call base class constructor.



## busbase.py

---

### AUTHORS

---

JAY CONVERTINO

---

### DATES

---

2025/03/11

---

### INFORMATION

---

#### Brief

---

classic bus define for packages

#### License MIT

---

Copyright 2025 Jay Convertino

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## transaction

---

ABC

transaction

ap b3trans

Abstract class for transaction types

## noSignal

---



noSignal

Class to use when a signal does not exist

## busbase

busbase

apb3Base

A busbase to transmit test routine.

## FUNCTIONS

### init

```
def __init__(
    self,
    entity
    :
    SimHandleBase,
    name
    :
    Optional[str],
    clock
    :
    SimHandleBase,
    args
    :
    Any,
    kwargs
    :
    Any
)
```

Initialize the object

## VARIABLES

### wqueue

self.wqueue

Queue to store write requests

### qqueue

self.qqueue

Queue to store read requests

## rqueue

---

```
self.rqueue
```

Queue to store result of read requests

## self.\_idle

---

```
self._idle
```

Event trigger for cocotb

## self.\_run\_cr

---

```
self._run_cr
```

Thread instance of \_run method

## FUNCTIONS

---

### \_restart

---

```
def _restart(  
    self  
)
```

kill and restart \_run thread.

### write\_count

---

```
def write_count(  
    self  
)
```

How many items in the write queue

### read\_count

---

```
def read_count(  
    self  
)
```

How many items in the read queue

## write\_empty

---

```
def write_empty(  
    self  
)
```

Is the queue empty?

## read\_empty

---

```
def read_empty(  
    self  
)
```

Is the queue empty?self.bus.penable.value

## write\_clear

---

```
def write_clear(  
    self  
)
```

Remove all write items from queue

## read\_clear

---

```
def read_clear(  
    self  
)
```

Remove all read items from queue

## wait

---

```
async def wait(  
    self  
)
```

Wait for the run thread to become idle.

## idle

---

```
def idle(  
    self  
)
```

Are all the queues empty and the \_run is not active processing data.

## write\_trans

---

```
async def write_trans(  
    self,  
    trans  
    :  
    transaction  
)
```

Write transaction to send to write queue

---

## read\_trans

```
async def read_trans(  
    self,  
    trans  
    :  
    transaction  
)
```

Read bus and output and transaction.

---

## \_write

```
async def _write(  
    self,  
    trans  
    :  
    transaction  
)
```

Write data one element at a time

---

## \_queue\_read

```
async def _queue_read(  
    self,  
    trans  
    :  
    transaction  
)
```

Setup queue for read requests

---

## \_read

```
async def _read(  
    self,  
    trans  
    :  
    transaction  
)
```

Read data one element at a time

## **\_check\_type**

---

```
def _check_type(  
    self,  
    trans  
)
```

Check and make sure we are only sending the correct transaction type

## **\_run**

---

```
async def _run(  
    self  
)
```

Virtual method for \_run thread that deals with read and write queues.

## TB

---

TB

Create the device under test which is the master/slave.

## FUNCTIONS

---

### run\_test

---

```
async def run_test(  
    dut,  
    payload_data  
    =  
    None  
)
```

Tests the source/sink for valid transmission of data.

### incrementing\_payload

---

```
def incrementing_payload()
```

Generate a list of ints that increment from 0 to 2^8

### test

---

```
def test(  
    request  
)
```

Main cocotb function that specifies how to put the test together.

## test.v

---

## AUTHORS

---

JAY CONVERTINO

---

## DATES

---

2025/03/17

---

## INFORMATION

---

### Brief

---

Test bench for apb using cocotb

### License MIT

---

Copyright 2025 Jay Convertino

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## test

---

```
module test #(
  parameter
    ADDRESS_WIDTH
    =
    32,
  parameter
    BUS_WIDTH
    =
    4
) ( input clk, input rstn, inout [ADDRESS_WIDTH-1:0] apb_paddr, inout apb_ps
```

Test bench loop for apb

## Parameters

<b>ADDRESS_WIDTH</b> <small>parameter</small>	Width of the APB3 address port in bits.
<b>BUS_WIDTH</b> <small>parameter</small>	Width of the APB3 bus data port in bytes.

## Ports

<b>clk</b>	Clock
<b>rstn</b>	Negative reset
<b>apb_paddr</b>	APB3 address bus, up to 32 bits wide.
<b>apb_psel</b>	APB3 select per slave (1 for this core).
<b>apb_penable</b>	APB3 enable device for multiple transfers after first.
<b>apb_pready</b>	APB3 ready is a output from the slave to indicate its able to process the request.
<b>apb_pwrite</b>	APB3 Direction signal, active high is a write access. Active low is a read access.
<b>apb_pwdata</b>	APB3 write data port.
<b>apb_prdata</b>	APB3 read data port.
<b>apb_pslverror</b>	APB3 error indicates transfer failure, not implimented.