basictrans

transaction

basictrans

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

basicMaster

busbase

basicMaster

basic bus master

VARIABLES

_signals

```
_signals
```

List of signals that are required

FUNCTIONS

__init__

```
def __init__(
self,
entity,
name,
clock,
reset,
                                                                     *
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.

basicEchoSlave

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

VARIABLES

_signals

```
_signals
```

List of signals that are required

FUNCTIONS

__init__

```
def __init__(
self,
 entity,
 name,
 clock,
 reset,
 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 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.

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.