

# driver.py

---

## AUTHORS

---

JAY CONVERTINO

---

## DATES

---

2025/03/27

---

## INFORMATION

---

### Brief

---

Bus Driver for Xilinx FIFO

### 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.

## xilinxFIFOsource

---

xilinxFIFObase

xilinxFIFOsource

Drive xilinx FIFO write interfaces

## 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,
    fwft
    =
    False,
    ack
    =
    False,
    args,
    kwargs
)
```

\*

\*\*

Setup defaults and call base class constructor.

---

### **write**

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

Write to a address some data

---

### **\_check\_type**

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

Check and make sure we are only sending xilinxFIFOtrans

## \_run

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

\_run thread that deals with read and write queues.

## xilinxFIFOsink

xilinxFIFObase

xilinxFIFOsink

Drive xilinx FIFO read interfaces

## 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,  
    fwft  
    =  
    False,  
    args,  
    kwargs  
)
```

Setup defaults and call base class constructor.

## write

---

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

Write to a address some data

## read

---

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

Read from a address and return data

## check\_type

---

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

Check and make sure we are only sending xilinxFIFOtrans

## \_run

---

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

\_run thread that deals with read and write queues.