Implementation of the INOUT Block in Buses

The INOUT block can be used in serial communication protocols like SPI and I2C, which require bidirectional signals.

1. Implementation on the SPI Bus

The SPI protocol (Serial Peripheral Interface) typically uses four signals:

- MOSI (Master Out Slave In): Master output, slave input.
- MISO (Master In Slave Out): Master input, slave output.
- SCK (Clock): Managed by the master.
- SS (Slave Select): Activates a slave device.

The MISO signal is a great example of a bidirectional pin because the master configures it as an input, while the slave configures it as an output. The INOUT block can be implemented in this case as follows:

```
assign miso = (slave_enable) ? data_out : 1'bZ;
```

Here, the slave writes to MISO only when it is enabled (slave_enable = 1). Otherwise, the pin remains in high impedance (Z), allowing another slave to control the line.

2. Implementation on the I2C Bus

The I2C bus (Inter-Integrated Circuit) uses two main signals:

- SDA (Serial Data): Bidirectional data line.
- SCL (Serial Clock): Clock line.

SDA must be an open-drain pin, meaning devices can only pull it to logical zero (0) or leave it in high impedance (Z). This is implemented with the INOUT block as follows:

```
assign sda = (write_enable) ? data_out : 1'bZ;
```

When **write_enable = 1**, the pin takes the value of data_out (it can be 0 or 1).
When **write_enable = 0**, the pin remains in high impedance (Z), allowing another device to control the line.

Revision #3
Created 6 March 2025 16:24:34 by Caroline
Updated 14 March 2025 11:56:10 by Caroline