WISHBONE
Interconnect Matrix
IP Core

Author: Rudolf Usselmann
rudi@asics.ws
www.asics.ws

Rev. 1.1
October 3, 2002

Revision History

Rev.	Date	Author	Description
1.0	10/19/01	Rudolf Usselmann	Internal release
1.1	10/3/2002	RU	Fixed a few minor typos.

1

Introduction

This core is a WISHBONE Interconnect Matrix. It can interconnect up to 8 Masters and 16 Slaves. It features a parameterized priory based arbiter in each Slave Interface, and allows for parallel communication between masters and slaves on different interfaces.

The WISBONE specification and additional information about WISHBONE SoC can be found at:
http://www.opencores.org/wishbone/

The Main features of the CONMAX are:

- Up to 8 Masters
- Up to 16 Slaves
- 1, 2 or 4 priority levels

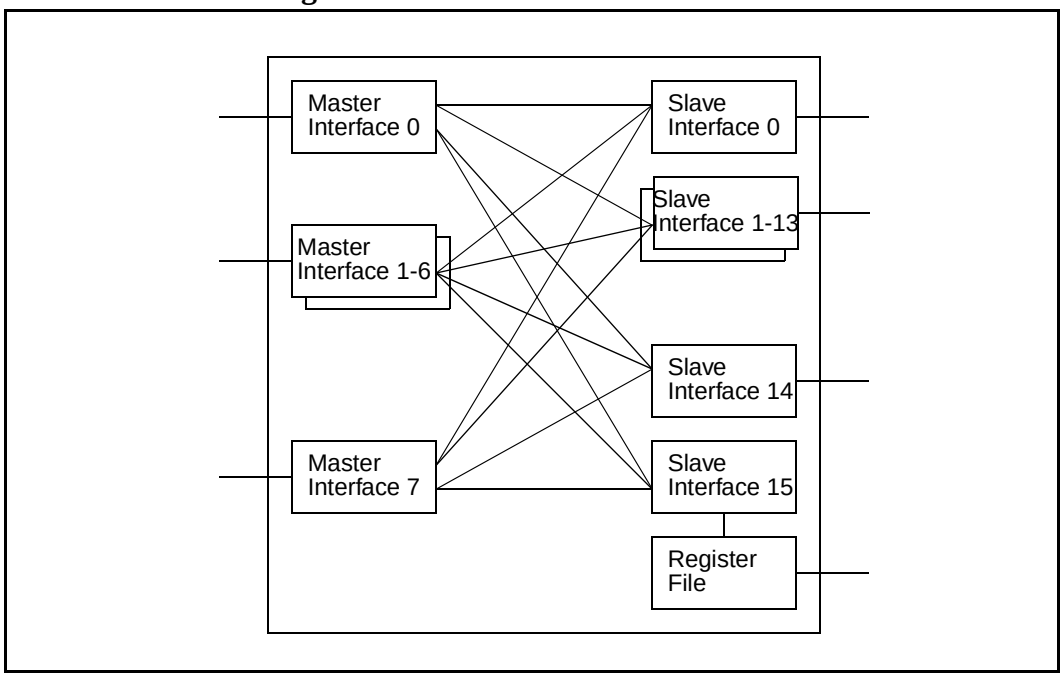
(This page intentionally left blank)

2

Architecture

Below figure illustrates the overall architecture of the core.

Figure 1: Core Architecture Overview



It consists of 3 main building blocks: a Master Interface, a Slave Interface and a Register File.

2.1. WISHBONE Master Interface

The Master Interface consists of a WISHBONE Slave that can connect to an external WISHBONE Master. When an external WISHBONE Master starts a WISHBONE cycle, the Master Interface will select the appropriate Slave Interface, based on WISHBONE Address bits [MSB:MSB-3].

2.2. WISHBONE Slave Interface

The Slave Interface consists of a WISHBONE Master that can connect to an external WISHBONE Slave. When an internal Master Interface starts a WISHBONE cycle, the Slave Interface will select the appropriate Master Interface, based on internal arbitration.

Both the WISBONE Master and WISHBONE Slave Interfaces are WISHBONE SoC Rev B2 compliant.

2.3. Register File

The Register File consists of 16 registers, each 16 bits wide. Each register is connected to one slave, and holds a 2 bit priority for each master.

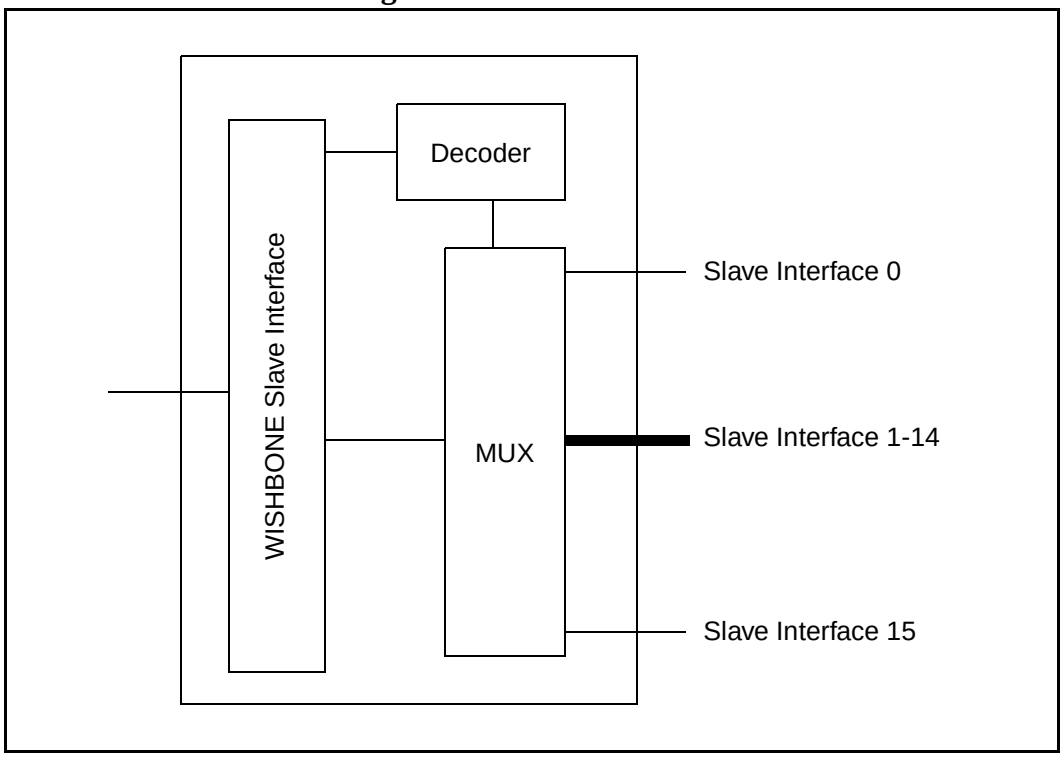
3

Operation

3.1. Master Interface

Each Master interface consists of a WISHBONE Slave interface and decoding logic.

Figure 2: Master interface



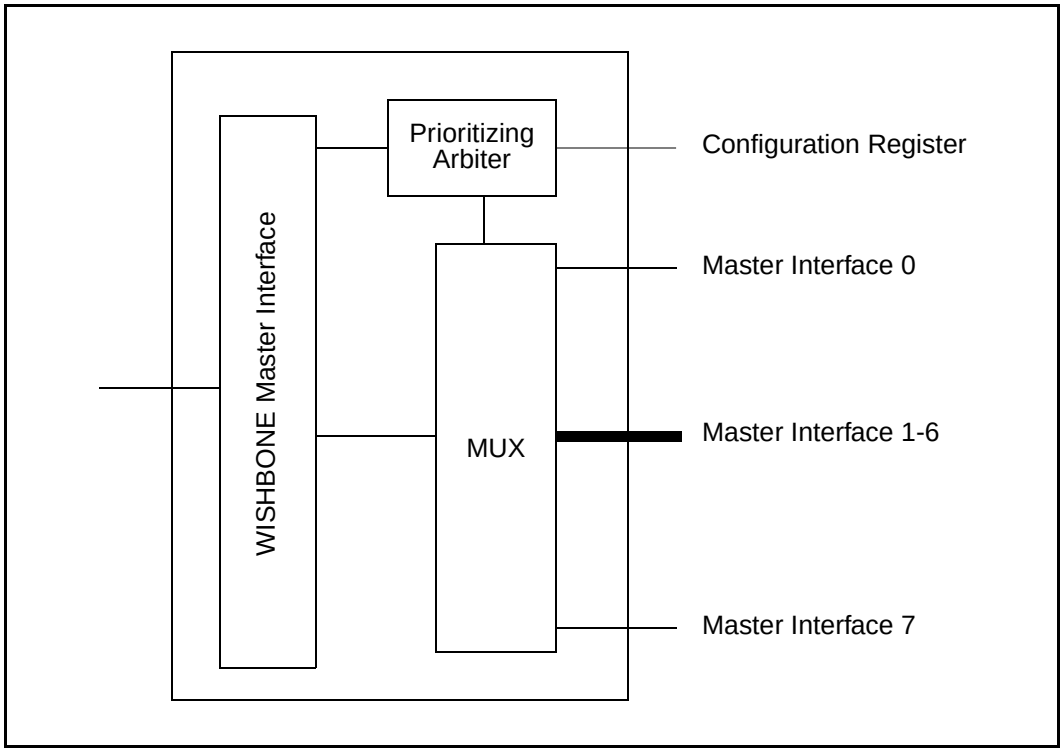
The Master Interface performs simple interface signal steering based on the upper four WISHBONE Address Bits. A 0 on the upper four WISHBONE address bits selects Slave 0, a 15 selects Slave 15.

An implementation that requires less than the provided 8 Master Interfaces should leave the outputs of the unused interfaces unconnected, and connect the inputs of the unused interfaces to ground. The unused interfaces will then be automatically removed during the synthesis process.

3.2. Slave Interface

Each Slave interface consists of a WISHBONE Master interface and a prioritizing Arbiter.

Figure 3: Slave Interface



The Slave Interface performs interface signal steering based on the output of the prioritizing arbiter.

An implementation that required less than the provided 16 Slave interfaces should leave the outputs of the unused interfaces unconnected, and connect the inputs of the unused interfaces to ground. The unused interfaces will then be automatically removed during the synthesis process. Note that Slave Interface 15 is also used for accessing the register file and therefore can not be completely removed.

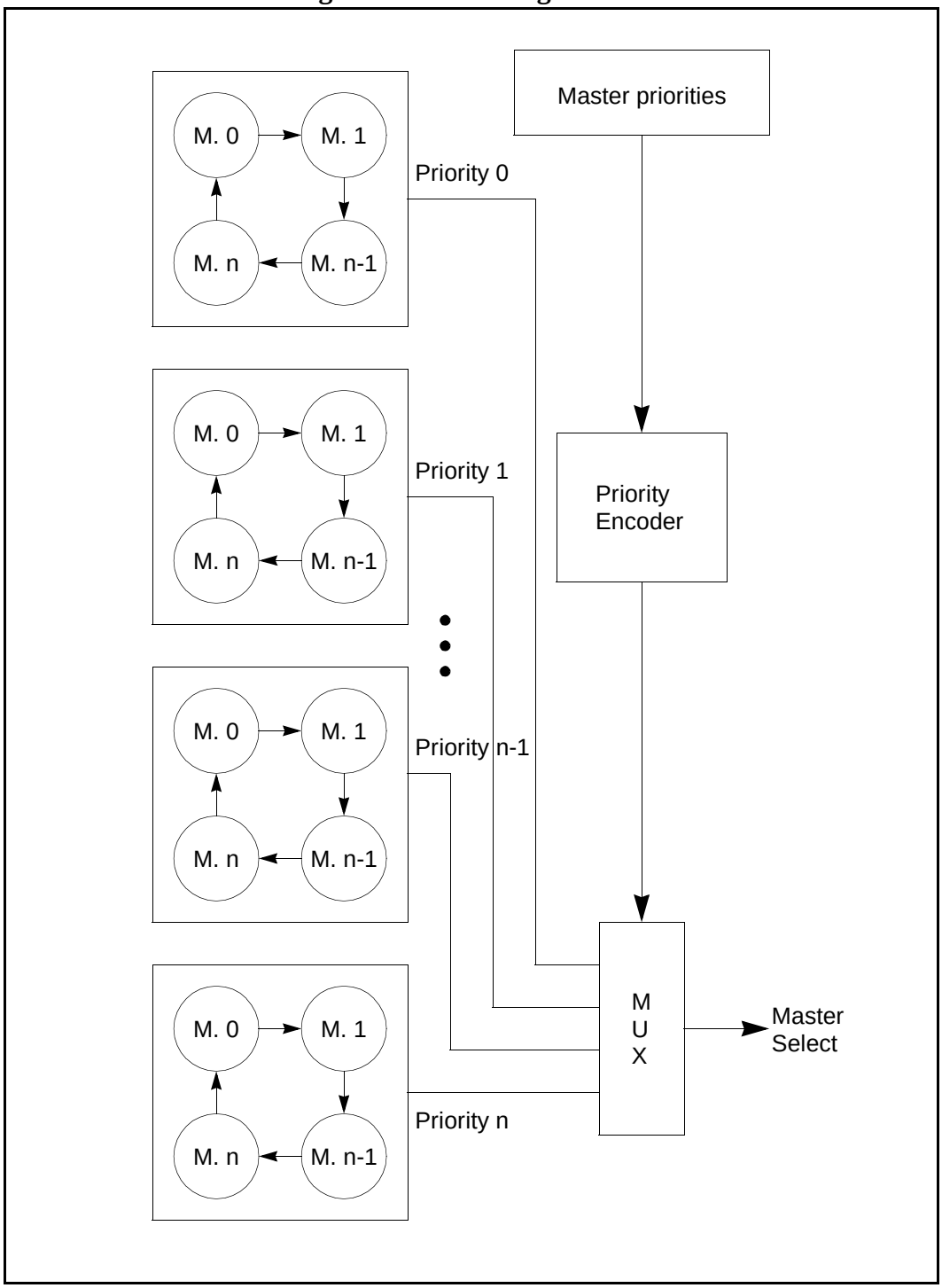
3.3. Prioritizing Arbiter

The prioritizing arbiter selects the Master that will own the Slave, based first on priority, and secondarily, if all priorities are equal, in a round robin way. Each master interface has a 2^1 bit priority value associated with it. A value of 0 identifies a master with very low priority, a value of 3 identifies a master with very high priority. Masters with the same priority are processed in a round robin way, as long as there are no Masters with a higher priority.

1. Implementation Dependent. This core supports 1, 2 and 4 priority levels. Please see Appendix A “Core HW Configuration” on page 13 for more information.

"Figure 4: Prioritizing Arbiter" on page 7 illustrates the internal operation of the prioritizing arbiter.

Figure 4: Prioritizing Arbiter



Care should be taken when using priorities, as Masters with lower priorities may be locked out and never serviced, if masters with higher priority are being continuously serviced.

3.4. Register File

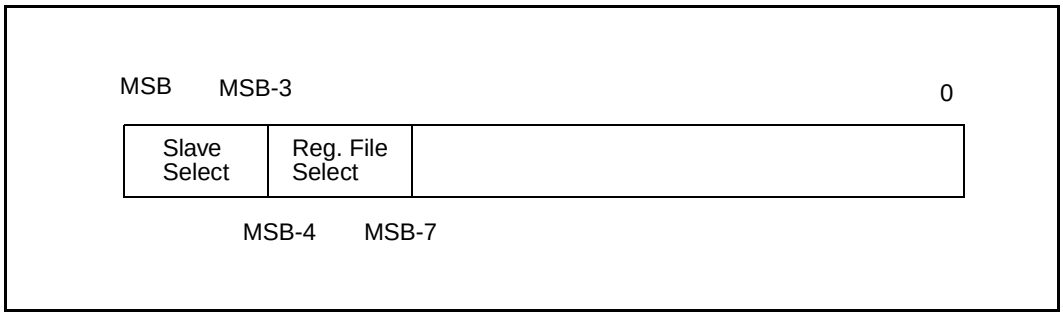
The Register File consists of 16 registers of 16 bits each. The register file sits behind slave interface 15.

The Register file is selected when a Master selects Slave Interface 15 and the second nibble from the top address bits (MSB-4 through MSB-7) are equal to the parameter “rf_addr”.

3.5. Addressing

The Address Space for each Master is divided as follows:

- Address bits [MSB:MSB-3] select the Slave Interface
- Address bits [MSB-4:MSB-7] == rf_addr select the Register File (for Slave 15 ONLY)



4

Core Registers

This section describes all control and status registers inside the WISHBONE CONMAX core. The *Address* field indicates a relative address in hexadecimal. *Width* specifies the number of bits in the register, and *Access* specifies the valid access types to that register. RW stands for read and write access, RO for read only access. A 'C' appended to RW or RO indicates that some or all of the bits are cleared after a read.

All RESERVED bits should always be written with zero. Reading RESERVED bits will return undefined values. Software should follow this model to be compatible to future releases of this core.

Table 1: Control/Status Registers

Name	Addr.	Width	Access	Description
CFG0	0	32	RW	Configuration Register for Slave 0
CFG1	4	32	RW	Configuration Register for Slave 1
CFG2	8	32	RW	Configuration Register for Slave 2
CFG3	c	32	RW	Configuration Register for Slave 3
CFG4	10	32	RW	Configuration Register for Slave 4
CFG5	14	32	RW	Configuration Register for Slave 5
CFG6	18	32	RW	Configuration Register for Slave 6
CFG7	1c	32	RW	Configuration Register for Slave 7
CFG8	20	32	RW	Configuration Register for Slave 8
CFG9	24	32	RW	Configuration Register for Slave 9
CFG10	28	32	RW	Configuration Register for Slave 10
CFG11	2c	32	RW	Configuration Register for Slave 11
CFG12	20	32	RW	Configuration Register for Slave 12
CFG13	34	32	RW	Configuration Register for Slave 13
CFG14	38	32	RW	Configuration Register for Slave 14

Table 1: Control/Status Registers

Name	Addr.	Width	Access	Description
CFG15	3c	32	RW	Configuration Register for Slave 15

4.1. Configuration Register (CFGn)

This is the configuration for master priorities. Each Slave has it's own priority register. A priority value[1] of 0 represents a master with lowest priority, a value of 3 a master with highest high priority.

Table 2: CFGn Register

Bit #	Access	Description
31:16	RO	RESERVED
15:14	RW	Master 7 Priority
13:12	RW	Master 6 Priority
11:10	RW	Master 5 Priority
9:8	RW	Master 4 Priority
7:6	RW	Master 3 Priority
5:4	RW	Master 2 Priority
3:2	RW	Master 1 Priority
1:0	RW	Master 0 Priority

Value after reset:

CFGn: 00 h

1. Implementation Dependent. This core supports 1, 2 and 4 priority levels. Please see Appendix A "Core HW Configuration" on page 13 for more information.

5

Core IOs

5.1. Interface IOs

All interfaces are WISHBONE Rev. B2 compliant. Actual interface signals are prefixed with "*m0_*" through "m7_" for Master Interfaces, and "s0_" through "s15_" for slave interfaces.

Table 3: CONMAX Interface

Name	Width	Direction	Description
clk_i	1	I	Clock Input
rst_i	1	I	Reset Input (asynchronous, active high)
Master Interface			
addr_i	32	I	Address Bus (Parametarizable)
data_i	32	I	Data Input (Parametarizable)
data_o	32	O	Data Output (Parametarizable)
sel_i	4	I	Byte Select Signals. (Parametarizable)
we_i	1	I	Write Enable
cyc_i	1	I	Cycle Signal
stb_i	1	I	Strobe Signal
ack_o	1	O	Acknowledge Signal
err_o	1	O	Error Signal
rty_o	1	O	Retry Signal
Slave Interface			
addr_o	32	O	Address Bus
data_i	32	I	Data Input
data_o	32	O	Data Output
sel_o	4	O	Byte Select Signals.

Table 3: CONMAX Interface

Name	Width	Direction	Description
we_o	1	O	Write Enable
cyc_o	1	O	Cycle Signal
stb_o	1	O	Strobe Signal
ack_i	1	I	Acknowledge Signal
err_i	1	I	Error Signal
rty_i	1	I	Retry Signal

Appendix A

Core HW Configuration

All configurable items are passed to the core as parameters. This chapter describes all parameters and other user adjustable defines.

A.1. Core Parameters

When instantiating the core, the user must pass various parameters to the core:

```
wb_conmax_top#(dw, aw, rf_addr,
                    pri_sel0 ... pri_sel15 u0(<IO Ports ...>);
```

A.1.1. dw and aw

This two parameters specify the Data Bus and Address Bus width respectively. The MSB of the address bus is aw-1, the MSB for the data bus is dw-1.

A.1.2. rf_addr

This four bit vector specifies the second from the top nibble of the register file address.

A.1.3. pri_selN

This two bit vector specifies the number of priorities slave N supports.

0 indicates 1 priority level
1 indicates 2 priority levels
2 indicates 4 priority levels

A.2. Example

```
wb_conmax_top
#(      32,             // Data Bus width
        32,             // Address Bus width
        4'hf,           // Register File Address
        2'h1,           // Number of priorities for Slave 0
        2'h1            // Number of priorities for Slave 1
        // Priorities for Slave 2 through 15 will default to 2'h2
        )
        u0(<IO Ports ...>);
```

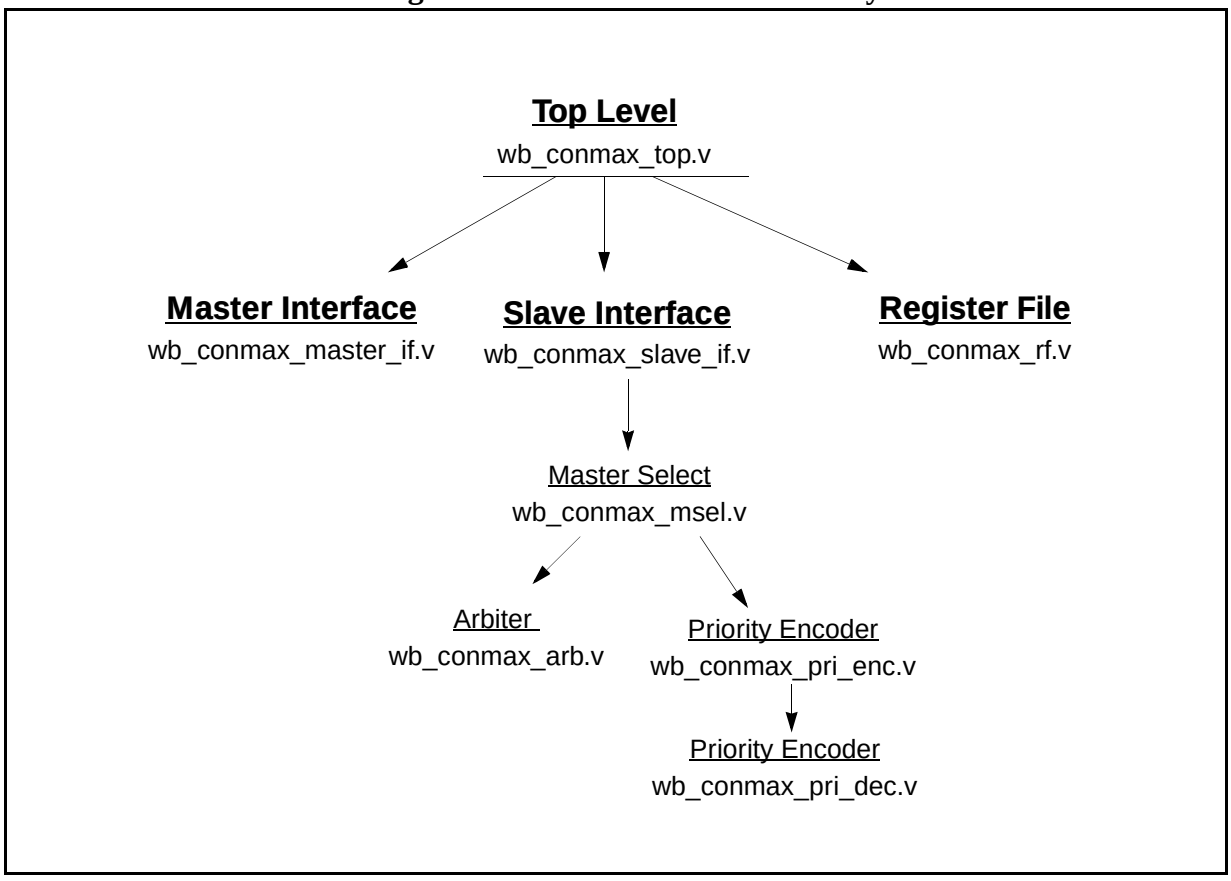
(This page intentionally left blank)

Appendix B

File Structure

This section outlines the hierarchy structure of the WISHBONE CONMAX IP Core Verilog Source files.

Figure 5: CONMAX Core Hierarchy Structure



(This page intentionally left blank)

Appendix C

Application Example

This section demonstrates how the WISHBONE CONMAX IP core can be used.