WISHBONE

Revision History

Rev.	Date	Author	Description
0.1	23/1/01	Rudolf Usselmann	First Draft Internal release
0.2	28/1/01	RU	First public release
0.3	13/3/01	RU	- Removed buffers and all references to buffered transfers - Updated WISHBONE interface e i341nated WI.trfatedass-Through mod30(ecscri

2

Architecture

Below figure illustrates the overall architecture of the core.

This implementation implements a 32 bit bus width and does not support other bus widths.

2.2. DMA Engine

The DMA engine is a up to 31 channel DMA engine that supports transfers

3

Operation

The WISHBONE DMA/Bridge consists of up to 31 DMA channels, the actual DMA engine, and a channel prioritizing arbiter (see “Figure 2: DMA Engine”).

Channels with the same priority are processed in a round robin way, as long as there are no channels with a higher priority.

"Figure 3: Channel Arbiter" on page 6 illustrates the internal operation of the

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

3.2. DMA Engine

The DMA engine can be programmed to perform various transfer operations. This section will illustrate several transfer options and their operation.

3.2.1. Normal (Software) DMA Operation

CSR register. In this case the DMA channel will immediately stop and indicate an error condition by setting the ERR bit in the channel CSR register and asserting an error interrupt (if enabled).

If CHK_SZ is not zero, the channel has to re-arbitrate for the interfaces after each CHK_SZ of words has been transferred. This is particularly useful when setting up all channels with the same priority and requiring “fair” bus usage distribution and low latency.

3.2.2. HW Handshake Mode

Below fi

```
CH0_BW = CH0_CHK_SZ/TOT_BW*100 // Channel 0 bandwidth (percent)
CH1_BW = CH1_CHK_SZ/TOT_BW*100 // Channel 1 bandwidth (percent)
CH2_BW = CH2_CHK_SZ/TOT_BW*100 // Channel 2 bandwidth (percent)
CH3_BW = CH3_CHK_SZ/TOT_BW*100 // Channel 3 bandwidth (percent)

Example:
CH0_CHK_SZ = 8
CH1_CHK_SZ = 4
CH2_CHK_SZ = 4
CH3_CHK_SZ = 1
TOT_BW = 8+4+4+1 = 17 (100%)
CH0_BW = 8/17*100 = 47%
CH1_BW = 4/17*100 = 23.5%
CH2_BW = 4/17*100 = 23.5%
CH3_BW = 1/17*100 = 5.8%
```

Figure 10: Back to Back DMA Transfers

3.9. Forcing Next Descriptor

The DMA core provides a special feature that allows a device to force the

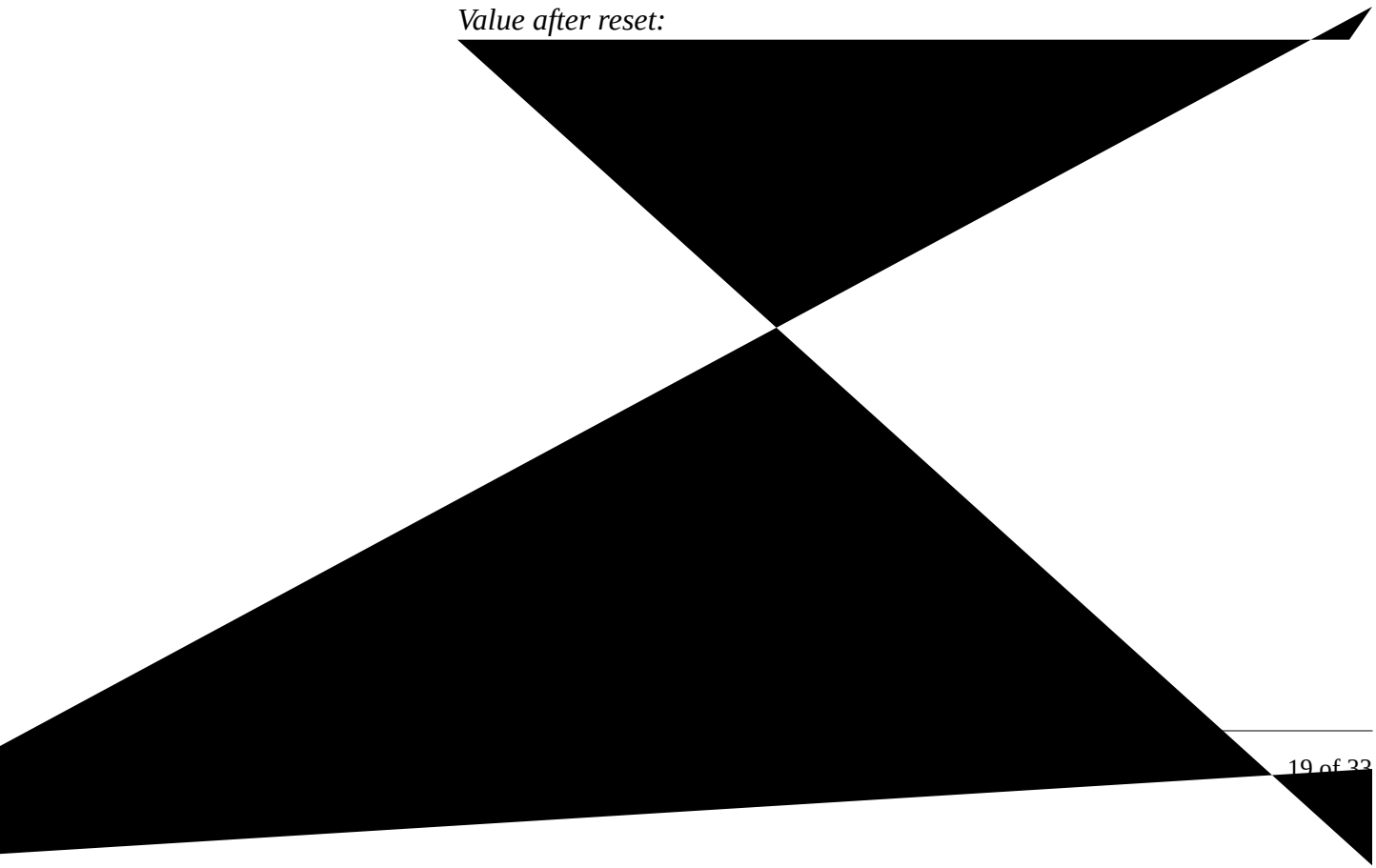
Starting Addr. 120

4.1. Main Configuration Status Register (CSR)

This is the main configuration register of the DMA/Bridge core.

Value after reset:

Value after reset:



```
`define     HAVE_CBUF9  1
`define     HAVE_CBUF10 1
`define     HAVE_CBUF11 1
`define     HAVE_CBUF12 1
`define     HAVE_CBUF13 1
`define     HAVE_CBUF14 1
`define     HAVE_CBUF15 1
`define     HAVE_CBUF16 1
`define     HAVE_CBUF17 1
`define     HAVE_CBUF18 1
`define     HAVE_CBUF19 1
`define     HAVE_CBUF20 1
`define     HAVE_CBUF21 1
`define     HAVE_CBUF22 1
`define     HAVE_CBUF23 1
`define     HAVE_CBUF24 1
`define     HAVE_CBUF25 1
`define     HAVE_CBUF26 1
`define     HAVE_CBUF27 1
`define     HAVE_CBUF28 1
`define     HAVE_CBUF29 1
`define     HAVE_CBUF30 1
```