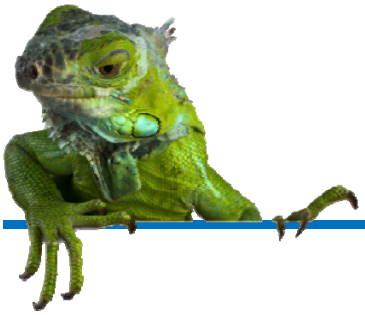
heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

REDS
Reconfigurable & embedded
Digital Systems



REPTAR

Reconfigurable **E**mbedded **P**latform for
Training **A**nd **R**esearch

User Guide

REDS Institute
July 2014
Ver 2.2

7.3.1.1 User interface timing

The *lba_ctrl* manage the data signals of the local bus interacting directly with the GPMC of the CPU and offers a simplified interface to the user, making easier the data transfers.

The User Interface block is shown in the figure 22, their input/outputs are described in the table 8.

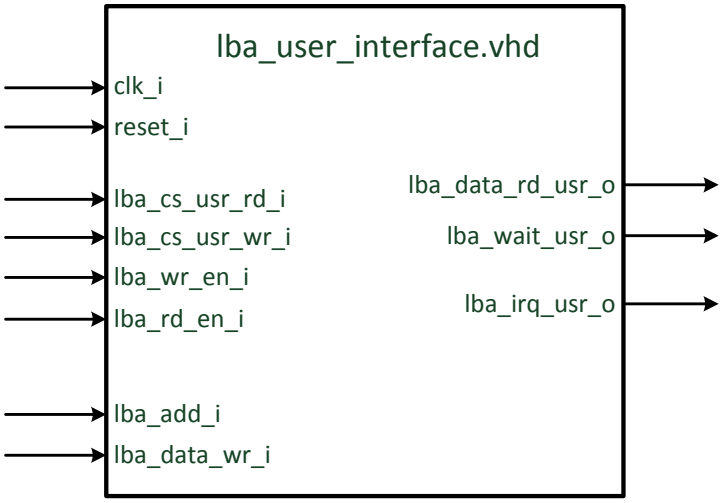


Figure 22 – Local Bus Asynchronous User interface

Signal	Direction	Width	Description
clk_i	input	1	System clock @ 200 MHz
reset_i	input	1	Asynchronous reset
lba_cs_usr_rd_i	input	1	Chip select for read access from the user interface
lba_cs_usr_wr_i	input	1	Chip select for write access to the user interface
lba_wr_en_i	input	1	Write enable
lba_rd_en_i	input	1	Read enable
lba_add_i	input	23	Address (16-bit-word, shift left 1 bit to get the byte address)
lba_data_wr_i	input	16	Data to be write in the user register addressed
lba_data_rd_o	output	16	Data read from the user register addressed
lba_wait_usr_o	output	1	Wait signal that indicate that the data in the bus is not yet valid
lba_irq_usr_o	output	1	Interrupt from the user interface to the CPU. See chapter [8]

Table 8 – Input/Outputs of the User Interface block

Read access

1. The User Interface block receives the address offset of the register to be read (clk 1)
2. The User Interface block receives the *lba_cs_usr_rd_i* signal to '1' (clk4)
3. The User Interface block must to put the data on the bus on the next clock, otherwise it can put the signal *lba_wait_usr_o* to '1' instead of the data to indicate that it's not ready. When the data is ready, the User Interface block puts the signal *lba_wait_usr_o* to '0' and the data on the bus at the same clock rising edge.

The data must remain on the bus until the CS signal (*lba_cs_usr_rd_i*) goes to '0'.

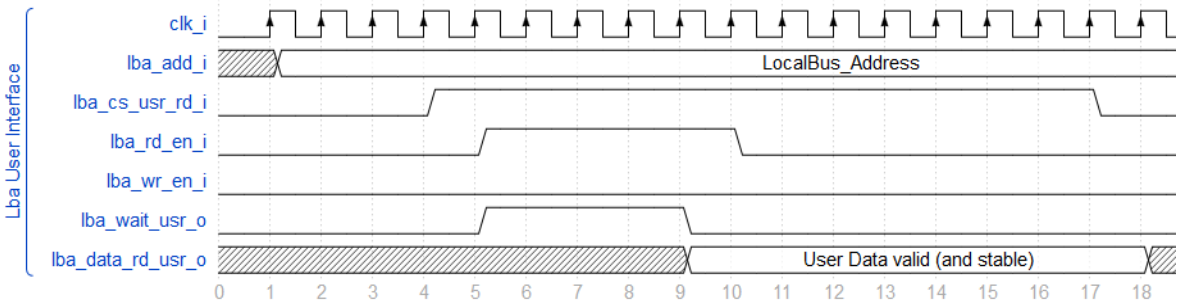


Figure 23 – Local Bus Asynchronous User Interface timing for read access with wait

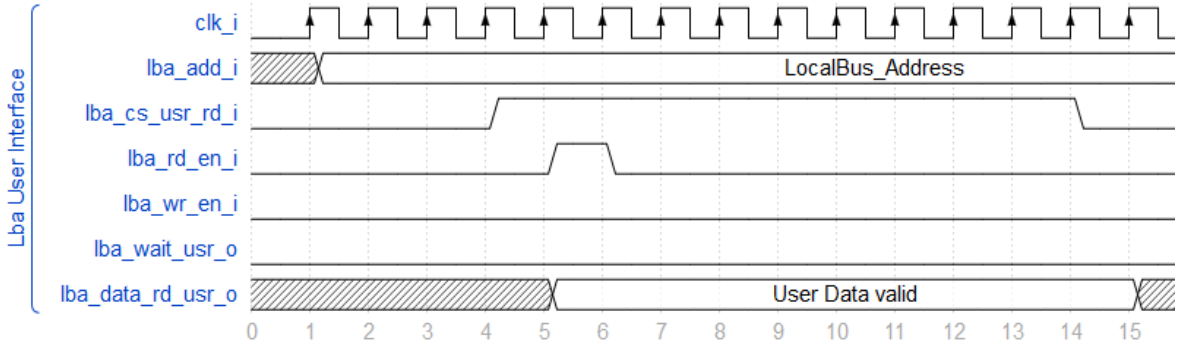


Figure 24 – Local Bus Asynchronous User Interface timing for read access without wait

Write access

The User Interface block receives the address offset, the data and the chip-select and write-enable signals at the same time. The control signals *lba_cs_usr_wr_i* and

lba_wr_en_i stay active during one clock cycle and can be directly used as the strobe of the register to catch the data.

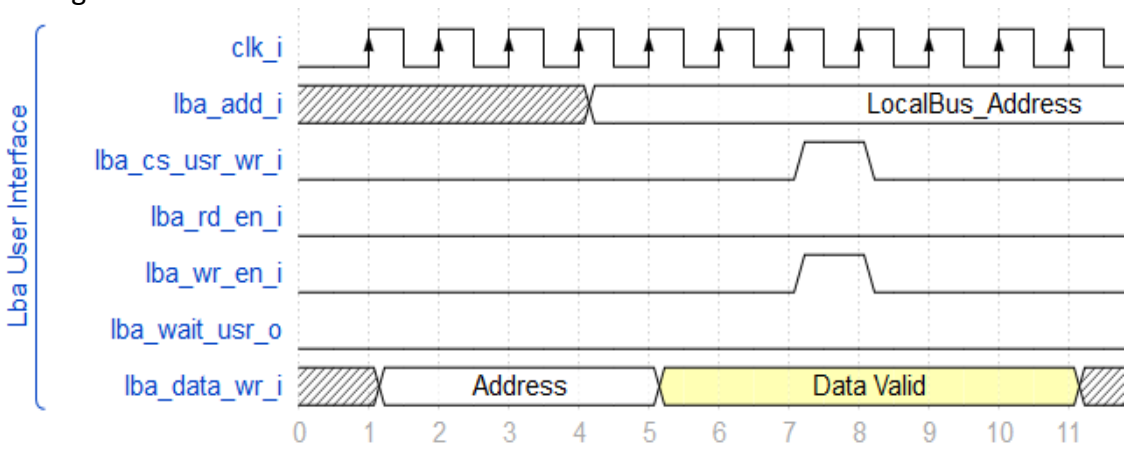


Figure 25 – Local Bus Asynchronous User Interface timing for write access

7.3.2 Synchronous access (CS4)

There is another chip select going from CPU to FPGA: the chip select CS4 of the GPMC used to allow the CPU access to the DDR connected to the Spartan6. In this case, the Local Bus is used in synchronous mode, that is, the GPMC clock is used to run the state machine of the Local Bus Controller of the FPGA.

In synchronous mode, we use a frequency of the GPMC clock of 50MHz (FCLK/4). Currently, transfers use bursts of 4 x 16 bits (equivalent to a *long long unsigned int* in C programming), the maximum allowed by the GPMC being 16 x 16 bits. In order to implement the maximum burst, DMA transfers can be used. This functionality is not yet included on the REPTAR BSP.

The synchronous access is explained in the section 10.1.5.10 of the DM3730 TRM. The Figures 22/23 show a synchronous multiple read/write operation on an address/data-multiplexed device with a GPMC_CLK = GPMC_FCLK/2.

The Table 7 gives the parameter values for the REPTAR release2014_v1, with i=4 (CS4).

The Figure 25 shows the states diagram of the Finite State Machine implemented on the FPGA to manage the Local Bus synchronous accesses (source file reptar_hard\reptar_series_1\cpld_fpga\fpga\src\ lbs_ctrl.vhd). This FSM gets the data from the Local Bus and then transfer it to the FIFOs of the Xilinx DDR Controller IP, called MCB Memory Controller Block, and vice-versa.