THE DINI GROUP

LOGIC Emulation Source

MainBus Specification

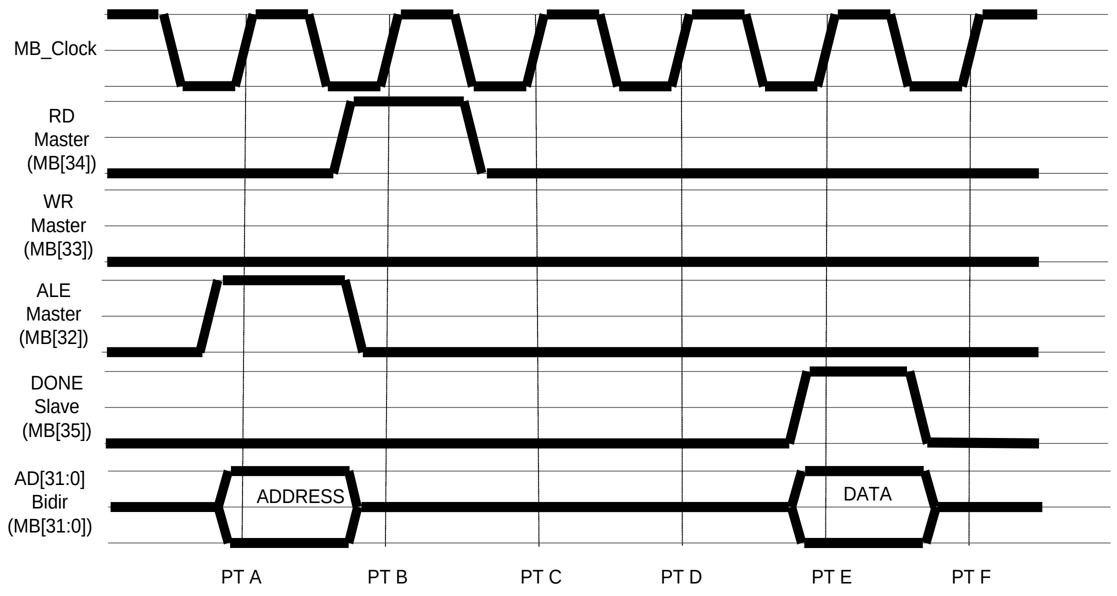
Table of Contents

1. Definition

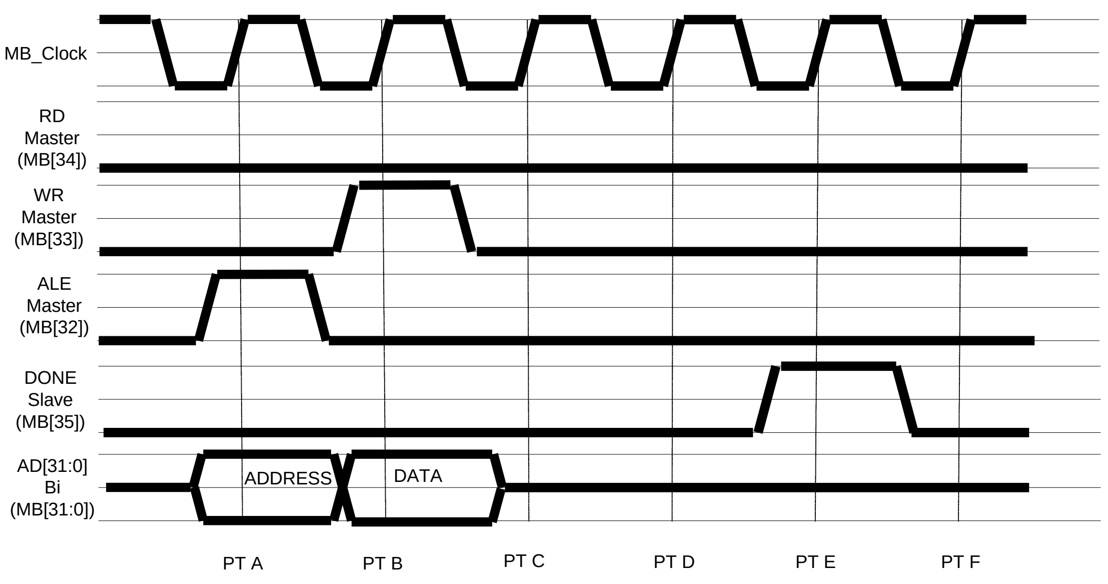
MainBus (MB) is a simple bus specification created by The Dini Group to facilitate transfer of data between FPGA's and host entities. Typically this is over a USB connection, but it may be over PCI, PCIe, or any other interface. In the MainBus specification there is one "master" and up to sixteen "slaves". On Dini Group products the ConfigFPGA acts as the "master" and the array FPGA's are all "slaves". MainBus is a 32-bit bus plus 4 control bits, for a total of 36 bits. It is synchronized to a clock signal and requires a reset signal (synchronous or asynchronous). The bus is shared for data and address, and is bi-directional for data. The four control signals are: ALE, RD, WR, and DONE.

2. Waveforms

The following waveform shows a typical MainBus read transaction. The address is transferred at PT A with the assertion of ALE. A read request is then initiated at PT B with the assertion of RD. The MainBus Master then waits for the assertion of DONE, which occurs at PT E, which signals valid data on the bus and ends the transaction.



The next waveform shows a typical MainBus write transaction. Again, the address is transferred at PT A with ALE. The write is initiated with valid data at PT B, with the assertion of WR. The Master cannot start another transaction until the intended slave responds with DONE, shown at PT E. Note that DONE is active low on the DN7000K10PCI board, and active high on all other products.



3. Bus Ownership

The 32-bit address/data bus (AD) is defined to be "owned" by the MainBus master at all times except during a read operation. The MainBus master MUST tristate the AD bus the clock cycle following its assertion of the RD signal. It MUST continue to tristate the bus until the clock cycle after receiving a DONE pulse from a MainBus Slave, or until a MainBus timeout has occurred, as described in the next section. When ALE and WR are both inactive, the MainBus master may tristate the AD bus, but it is not required.

4. Timeouts

If a Slave is not present or fails to respond for some reason, the MainBus Master must eventually time-out and move on in order to prevent the system from locking up. The timeout for both reads and writes is defined to be 256 MainBus clock cycles. MainBus slaves MUST be able to respond in this time. On read transactions the MainBus Master is defined to return the value 0xDEADDEAD on a timeout condition. Other types of error conditions may return other error codes- for example, on Dini Group Products there is typically a register in the configFPGA to enable MainBus access, and a MainBus read attempted when MainBus is disabled will return 0x12345678. It is up to the user to determine that a timeout condition has occurred- there is no defined error handling protocol in the MainBus Specification. The MainBus master must continue operating normally after a timeout has occurred. Note that valid data may be confused with the timeout value by software, so care must be taken in implementing algorithms to detect MainBus timeouts.

5. FPGA Implementation

On Dini Group products the MainBus clock runs at 48Mhz and does not have a standardized name across products. It is often called "SysClk" or "USBClk". Check the reference design for your product to see which clock MainBus is synchronized to.

The FPGA's logic reset signal should be used to reset all MainBus logic to its initial state. The logic reset does not have a standardized name across all Dini Group products. It is often called "FPGA_GRSTN" or "RESET_FPGAS". See the reference design to determine appropriate reset usage.

The DONE signal is active high on all boards except the DN7000K10PCI. The configFPGA design on this board drives an active low DONE.

The address/data and control signals of MainBus are mapped to the MB[35:0] signals as follows:

Bus Line	MainBus Name	Usage	Driver
MB[31:0]	AD[31:0]	Shared Address/Data	Bidirectional
MB[32]	ALE	Address Strobe	Master (ConfigFPGA)
MB[33]	WR	Write Enable	Master (ConfigFPGA)
MB[34]	RD	Read Enable	Master (ConfigFPGA)
MB[35]	DONE	Done / Read Data Valid	Slave (Responding FPGA)

Input and Output flip-flops MUST be used on ALL MainBus signals. The IO flip-flops MUST be packed in IOB's in order to guarantee timing is met. The most common pitfall in implementing a MainBus target is to omit the IO flip-flops or allow them to be placed outside the IOB. To guarantee IO timing is met, constrain setup and clock-to-out times as well as frequency of the MainBus clock (48 Mhz).
MainBus clock frequency: 48 Mhz
Setup requirement at FPGA: 3 ns
Clock-to-Out requirement at FPGA: 5 ns

The driver characteristics of the FPGA can be important for MainBus to operate reliably. Set FPGA drivers as follows for maximum performance:
IO Standard: LVCMOS25[*]
Slew Rate: Fast
Drive Strength: 24mA

[*]A few Dini Group products implement MainBus at a signaling level other than 2.5V. Check the reference design for your product to verify the appropriate IO Standard, or check the schematic to verify the VCCIO for the FPGA bank on which MainBus is connected. For other voltages, use the appropriate LVCMOS IO Standard (ie LVCMOS33 or LVCMOS15), and use the same slew rate and drive strength settings specified above.

6. Provided HDL

The Dini Group provides HDL (verilog and VHDL) to implement a MainBus Slave in an FPGA. The source code can be found on the Reference CD in the Reference Design area in the folder "common/MainBus". The module "MB_target" implements a MainBus Slave with a parameterized 4-bit address. On the back-end, this module provides 4 chip selects, address, and read/write enables. The module "registers" gives an example of implementing a simple register bank that can be directly connected to "MB_target". The module "mainbus_tf" provides a test fixture for testing MainBus Slave designs. The description in this file shows how to call the test fixture functions to generate MainBus reads and MainBus writes in your test bench.

7. Address Map

By convention, the upper 4 bits of address are used to decode which device on a bus is acting as the Slave (for subsequent WR and RD commands). This convention limits the number of slaves on the bus to 16.

By convention, FPGA A (or F0), is selected by sending an ALE command with 0x0 in the upper 4 bits of address. FPGA B (or F1) is selected by sending an ALE command with 0x1 in the upper 4 bits, and so on. The remaining 28 bits of address are left for decoding within each Slave. In Dini Group Product Reference Designs, the next two bits typically decode between internal registers, external memory (ie DDR2 DRAM), and FPGA Interconnect Test Control.

This convention leaves 28 bits of addressing for each device on the bus. The customer does not need to follow this convention, but the Dini Group recommends following the convention if it meets the customer’s needs.

When an ALE command causes a new device to be selected as Slave, control of the DONE signal needs to be given to the new slave device. The de-selected device must tri-state this signal.

8. Host Interfaces

On Dini Group Products MainBus can be used to transfer data from a Host PC to FPGA designs on the board. It is quick and simple to implement, and examples are provided in the Dini Group Reference Designs. Both the USBController software (USB) and the AETest software (PCI) provide ready-to-use solutions for moving data between a Host PC and your FPGA designs.

For PCI hosted boards, the following steps may be used to access MainBus. See the AETest Source Code for implementation examples in the mb_read() and mb_write() functions.

1. Write the 32-bit (4 byte) “MainBus” address you wish to access to BAR 0, offset 0x240. This sets the ConfigFPGA MainBus address register. This also causes an “ALE” pulse to occur over MainBus, setting the current address.
2. To read from MainBus at the current address read 4 bytes from BAR 0, offset 0x250. This will cause the ConfigFPGA to read from MainBus and return the 32-bit result.
3. To write to an address on MainBus, write 4 bytes to BAR 0, offset 0x248, LSB first. The four bytes of data will be sent as a WR command over main bus to the current address.
4. The ConfigFPGA is capable of auto-incrementing the MainBus address after read and write operations, which gives an enormous performance increase when reading or writing large chunks of consecutive addresses. By default this feature is turned OFF for MainBus accesses over PCI. (For accesses over USB this feature is turned ON by default). To enable the address auto-increment feature for MainBus over PCI write "0x1" to BAR 0, offset 0x270.

For USB hosted boards, vendor requests are used to transfer data to and from MainBus. See the functions mb_write() and mb_read() in the USBController source code for implementation examples.

9. Legacy Issues

Older products may use a fifth control signal, "rd_data_valid". This was placed on MB[35], and DONE was placed on MB[36]. As it became clear that "rd_data_valid" was not necessary, it was dropped and DONE was moved down to MB[35]. If you are implementing a MainBus Slave and are experiencing trouble that may be related to this issue it is strongly suggested that you contact support@dinigroup.com and inquire about a firmware upgrade for your product, which will update the configuration section to be compatible with the MainBus Specification described in this document.
The DONE signal was changed to active low on the DN7000K10PCI. This move caused a lot of confusion, and in the end it was decided to keep DONE as an active high signal on all future products.