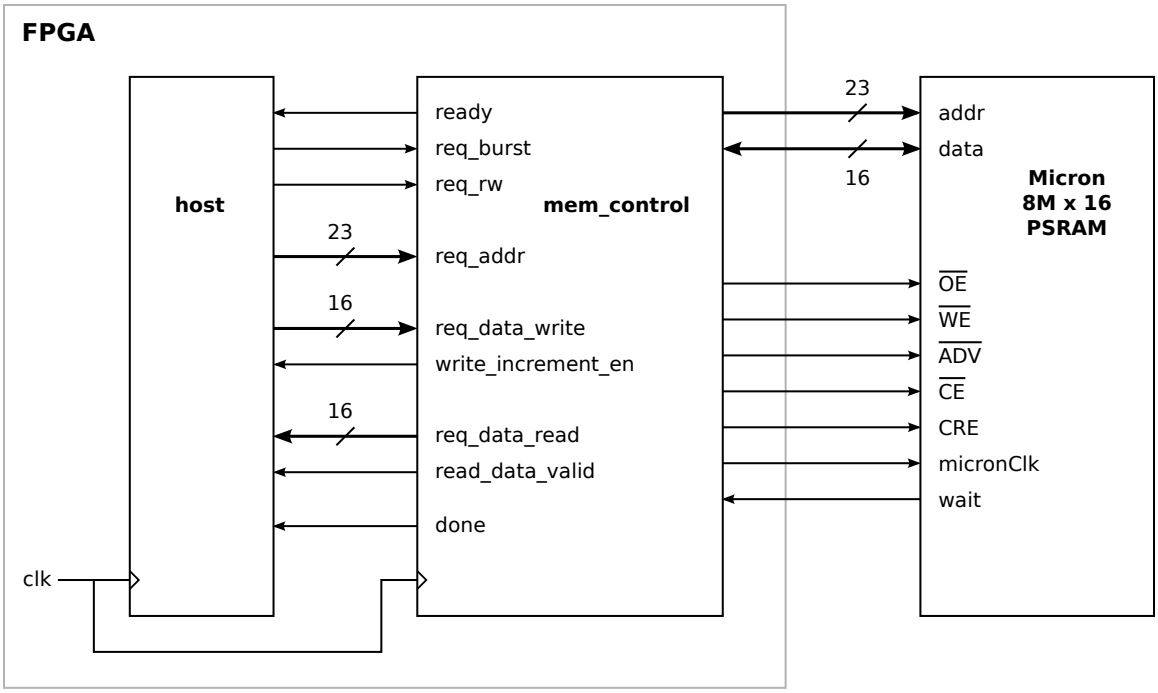
Micron CellularRAM Burst Memory Controller

1 Summary

This design provides a burst-mode synchronous interface to the Micron 16 MB (8M $\times$ 16) pseudo-SRAM chip commonly found on Digilent Nexys-2 and Nexys-3 FPGA development boards. Default settings are for 32-word burst length and 7-clock latency (to allow internal refresh cycles at the start of read or write bursts). At the maximum clock frequency of 80 MHz, the net transfer rate is approximately 120 MB/sec.



2 Interface

On FPGA startup the RAM is reconfigured to allow synchronous burst access; once this is completed, and after completion of each data transfer, `ready` is asserted high to indicate that a new transfer can be initiated. The `ready` signal must stay asserted for at least two 80 MHz clock cycles to ensure a refresh operation is triggered, so during the first clock cycle that `ready` is high the memory controller will not respond to read or write requests.

To request an operation the host drives the `req_burst` signal high; the direction of the transfer is determined by the state of `req_rw` (0 for write, 1 for read). The address is latched into the IO block on the first rising edge of `clk` while `req_burst` is high. The host must be prepared to handle the full burst length before initiating an operation.

The requested address must also be divisible by the burst length.

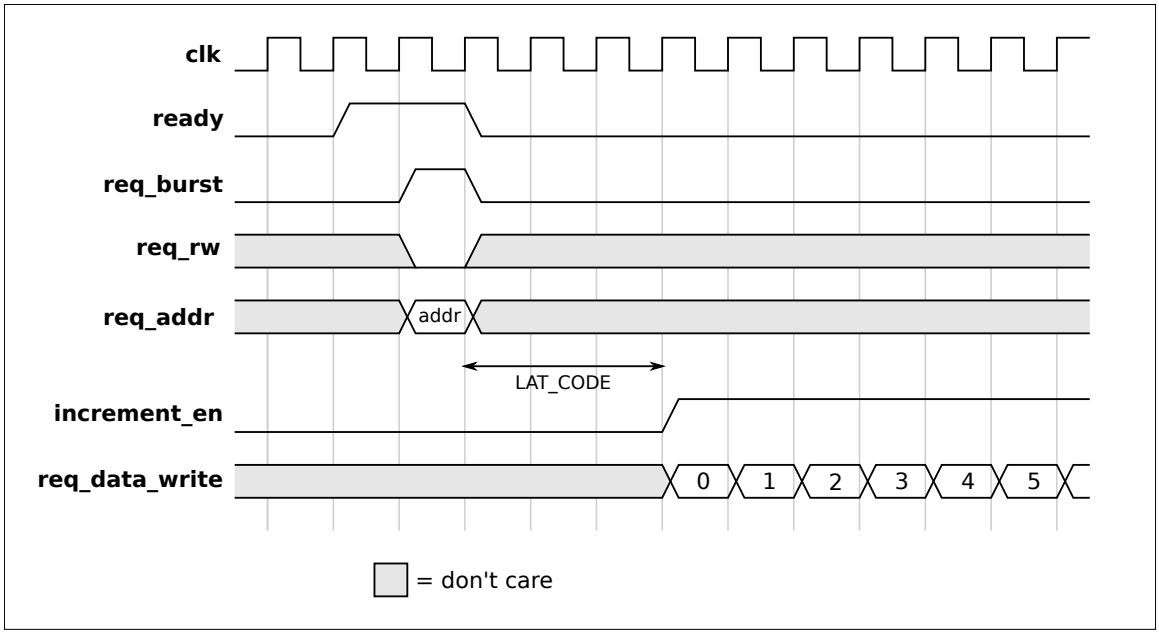


Figure 1: Write sequence timing diagram

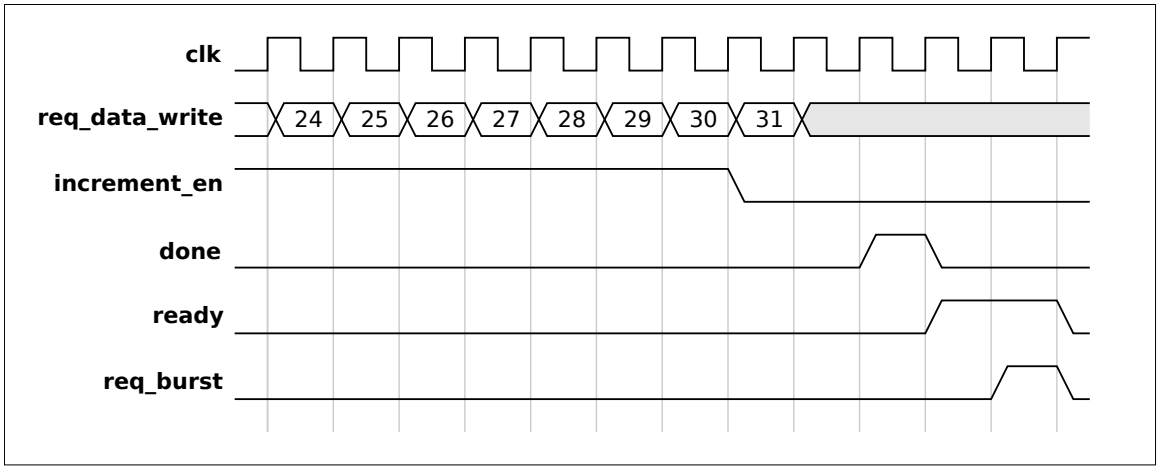


Figure 2: End of write sequence

During write operations, `write_increment_en` is asserted to indicate that data is being latched into the IO block; this signal is intended to control a FIFO or register file pointer, and eliminates the need for the host to keep track of burst lengths. After all data has been latched into the external RAM, `done` is asserted for exactly one clock cycle.

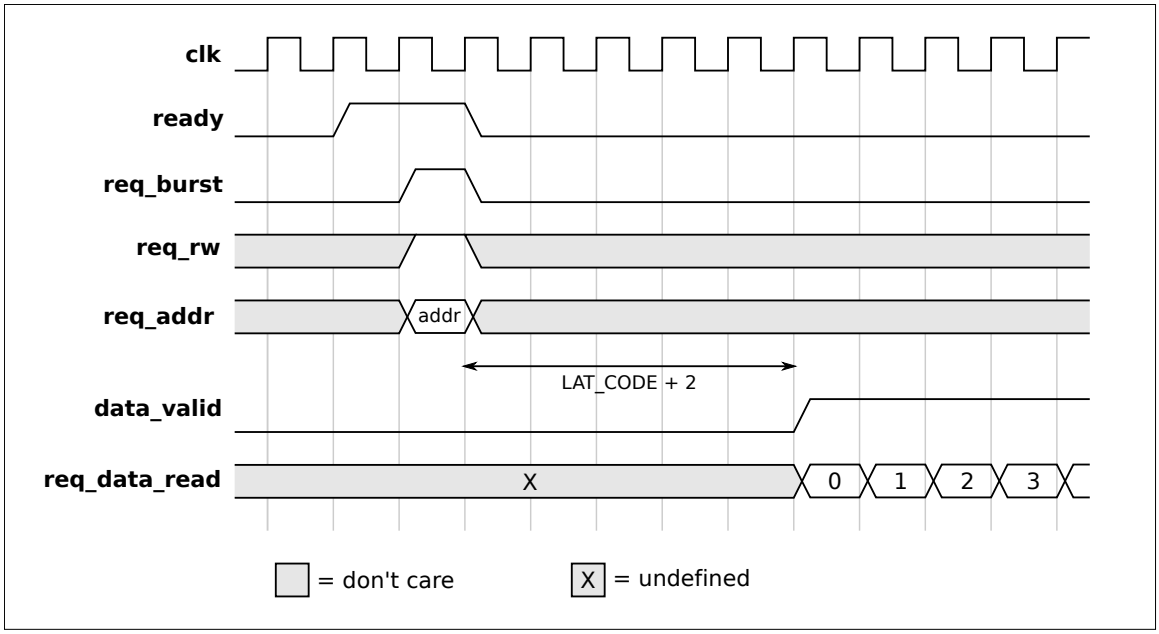


Figure 3: Read sequence timing diagram

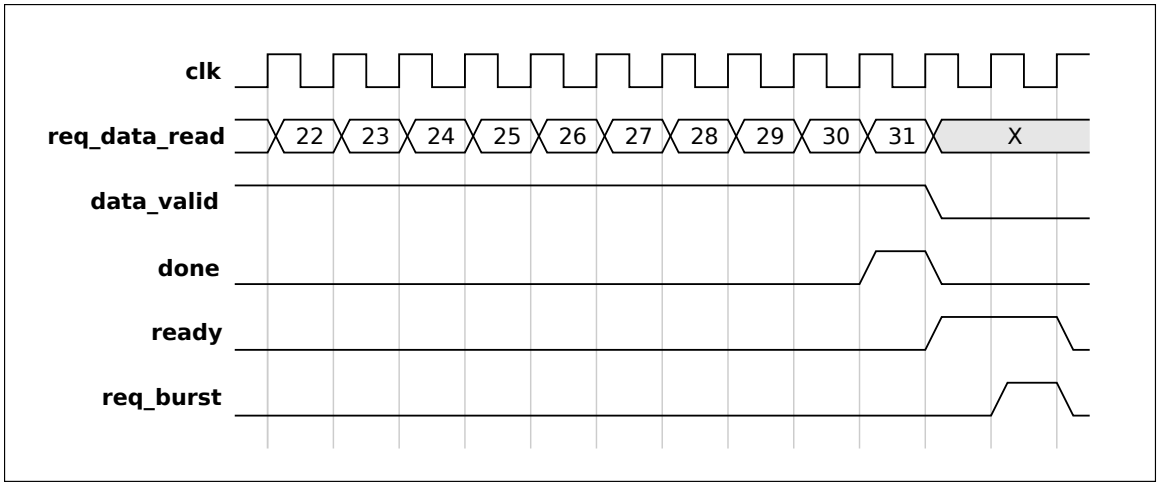


Figure 4: End of read sequence

During read operations, `read_data_valid` is asserted to indicate that valid data has been latched in the IO block and should be read by the host. Analogous to `increment_en`, `read_data_valid` is intended to be useful in driving data into a FIFO or register file. The `done` flag is asserted for one clock cycle at the end of a read operation.

3 Implementation

Internal FPGA signals are prone to glitches, so all outputs are registered in the IO block. The external RAM clock is generated by a double data rate register (DDR2 primitive), and is active only when a transfer is in progress. During read operations the external RAM clock is in-phase with the system clock and during write operations it is inverted, but all connections to the host circuit are synchronous with the system clock. The address valid signal (`micronADV`) is also generated by a DDR register, to enable it to bracket the clock in both read and write modes.

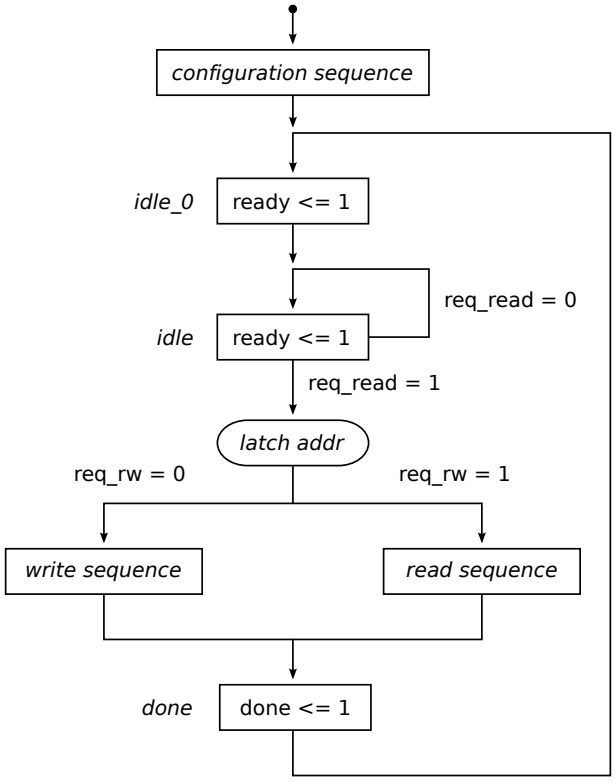


Figure 5: Simplified state diagram

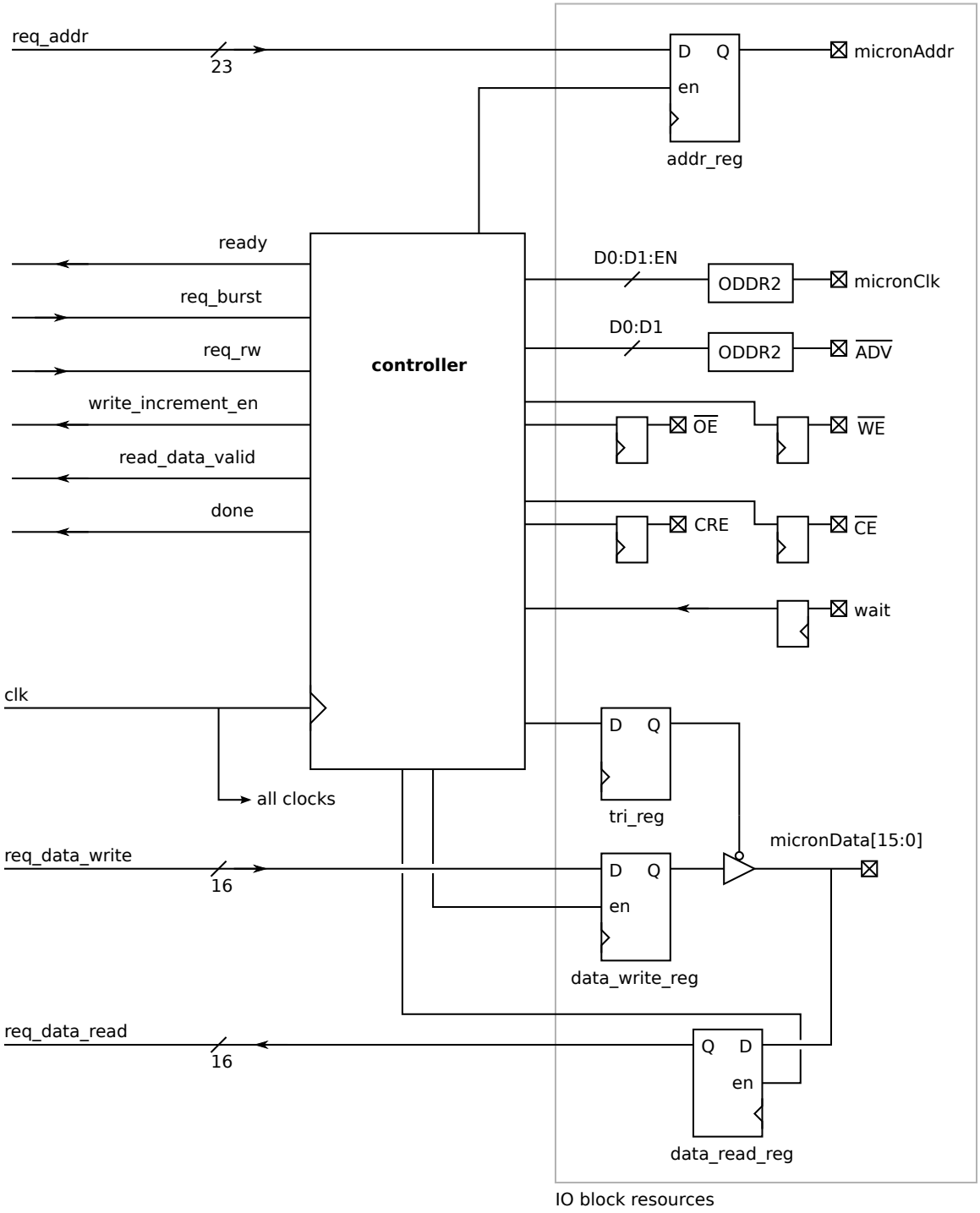


Figure 6: Detailed block diagram