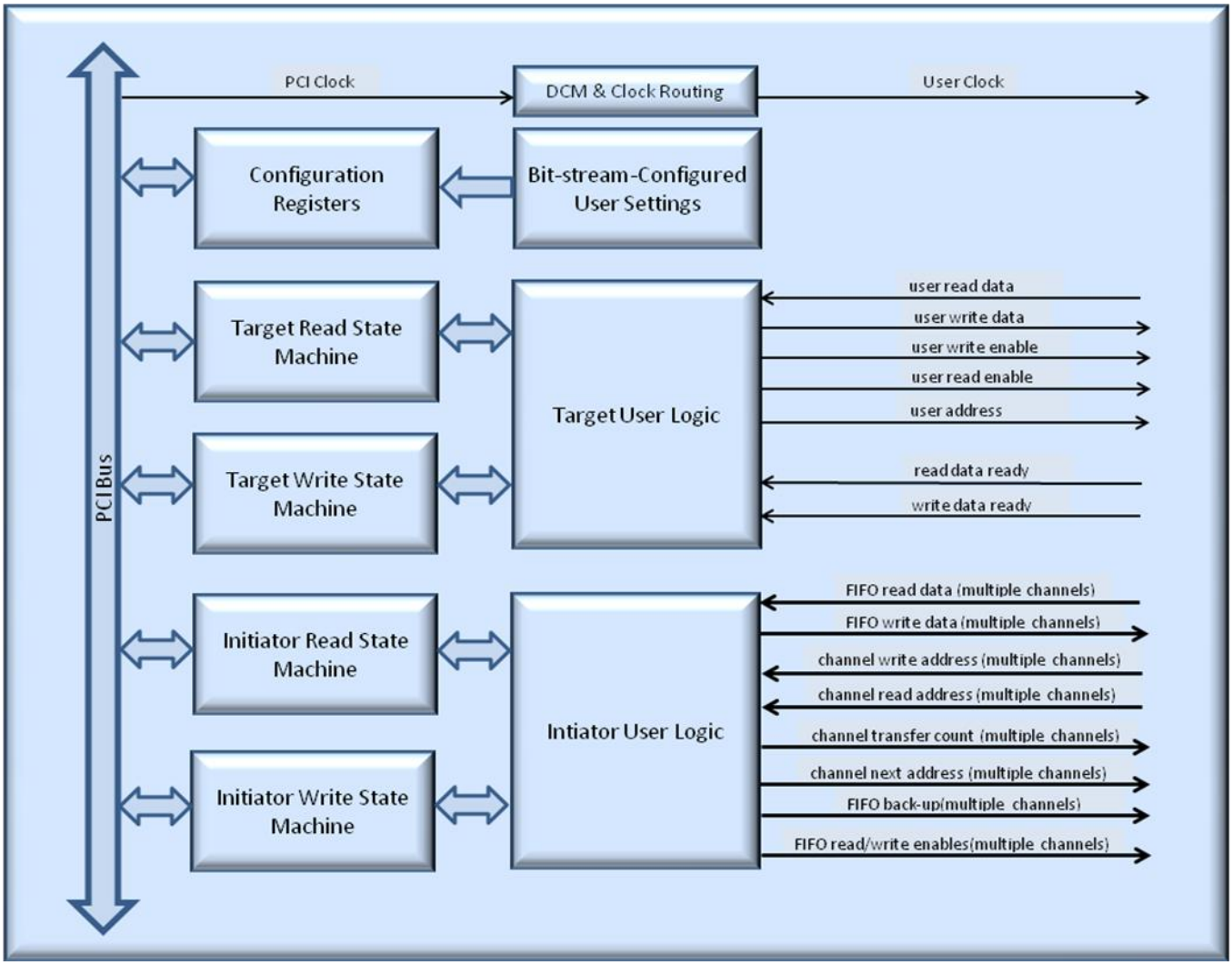

PCI Core Design

The project goal was to design a PCI core into a Xilinx Virtex 2 FPGA on circuit board that had already been placed and routed. The design was required to run at both 33 and 66 Mhz, and have full Initiator/Target capability. FPGA design, coding, simulation and implementation required eight fifty-plus-hour weeks, with another week for lab verification.

As can be seen from the diagram, functionality was split into "pure" PCI logic, on the left, and logic more geared to the specific user functions intended for the board. The PCI logic managed all the steps of PCI device configuration. It guaranteed compliance with initiator and target PCI protocol, including target responses to configuration and memory space accesses. It handled parity checking and generation, and the proper handling of data/address parity errors and addressing (Target Abort) errors. It also managed bus arbitration, and re-arbitration (in compliance with the Latency Timer protocol) for initiators,

The Target User Logic provided translation of target accesses to conventional address buses and mapped registers and memory arrays in the FPGA external to the core. Initially only single-data-phase reads and writes were supported. Burst attempts were responded to with "disconnect with data" terminations of the transaction. However, support for target burst accesses was added late in the development cycle.

The Initiator User Logic was, from the beginning, designed to support burst transactions. The user logic provided signals telling circuitry external to the core to provide, or accept, data (during writes or reads, respectively). It also counted data phases during bursts and, following early termination of a transaction, adjusted counters and issued FIFO "backup" commands. A later enhancement allowed multiple user read and write channels to access the PCI bus through the initiator state machines. This required a separate arbiter (distinct from PCI arbitration) within the Initiator User Logic.