OpenCores Wishbone Bus Architecture

Primary Documentation

Specification for the: WISHBONE System-on-Chip (SoC) Interconnection Architecture for Portable IP Cores, Revision: B.3, Released: September 7, 2002, OpenCores.org.
http://cdn.opencores.org/downloads/wbspec_b3.pdf

Wishbone B4: WISHBONE System-on-Chip (SoC)Interconnection Architecturefor Portable IP Cores, 2010, OpenCores.org.
http://cdn.opencores.org/downloads/wbspec_b4.pdf

Combining WISHBONE interface signals, Application note, Rev. 0.2, Richard Herveille, April 18, 2001, OpenCores.org.
http://cdn.opencores.org/downloads/appnote_01.pdf

OpenCores SoC Bus Review, Rev. 1.0, Rudolf Usselmann, January 9, 2001, OpenCores.org.
http://cdn.opencores.org/downloads/soc_bus_comparison.pdf

Wishbone (computer bus), From Wikipedia, the free encyclopedia
http://en.wikipedia.org/wiki/Wishbone_%28computer_bus%29

SoC Interconnection: Wishbone Web Site
http://opencores.org/opencores,wishbone

WISHBONE BUS ARCHITECTURE – A SURVEY AND COMPARISON, Mohandeep Sharma and Dilip Kumar, International Journal of VLSI design & Communication Systems (VLSICS) Vol.3, No.2, April 2012.
http://arxiv.org/ftp/arxiv/papers/1205/1205.1860.pdf

Introduction

System-on-Chip design and implementation requires specific architectural elements in order to integrate the various functions and components present in an integrated, embedded system. The most fundamental of these is the system interconnect.

The interconnection system must tie all the various subcomponents together in a manner that meets all communication and timing needs. It may consist of point-to-point connections, one-way or bi-directional capability, single or multiple master shared busses, or even a combination of these approaches.

As an embedded design, the implementation may be uniquely customized for every application or a common standard can be selected that allows for reuse, with any desired customization, across multiple designs and developments.

The WishBone bus is convenient candidate as,

> "The WISHBONE standard is not copyrighted, and is in the public domain. It may be freely copied and distributed by any means. Furthermore, it may be used for the design and production of integrated circuit components without royalties or other financial obligations."
> http://opencores.org/opencores,wishbone

In addition, there have been a few comparative reviews of available SOC buses. The following are two such opinions:

> "At the end I feel it would be a wise choice to adopt wishbone as a primary interface to our cores. It's signaling appears to be very intuitive and should be easily adopted to the other interfaces when needed. "
> OpenCores SoC Bus Review, Rev. 1.0, Rudolf Usselmann.

> "At the end, this paper endorses the view held by Rudolf Usselmann [33] that it would be a wise choice to adopt WISHBONE as a primary interface to our cores because its signaling appears to be very intuitive and should be easily adopted to the other interfaces when needed. "
> WISHBONE BUS ARCHITECTURE – A SURVEY AND COMPARISON, Mohandeep Sharma and Dilip Kumar

Wishbone B4: WISHBONE System-on-Chip (SoC)Interconnection Architecturefor Portable IP Cores, 2010, OpenCores.org.
http://cdn.opencores.org/downloads/wbspec_b4.pdf

Chapter 8.

Appendix A – WISHBONE Tutorial2

By: Wade D. Peterson, Silicore Corporation

Chapter 3. WISHBONE Classic Bus Cycles

3.1.3.1 Standard wishbone protocol

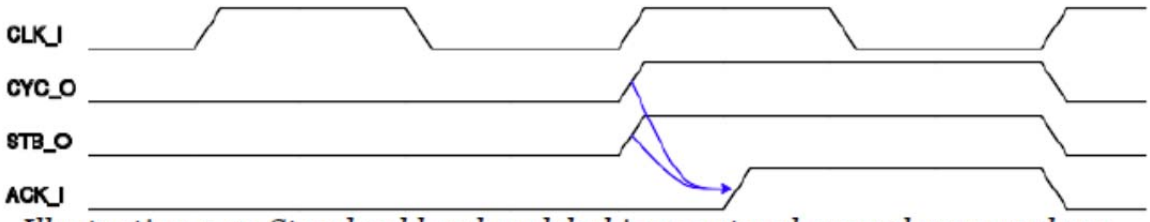

Illustration 3-2: Standard bus handshaking protocol, asynchronous slave.

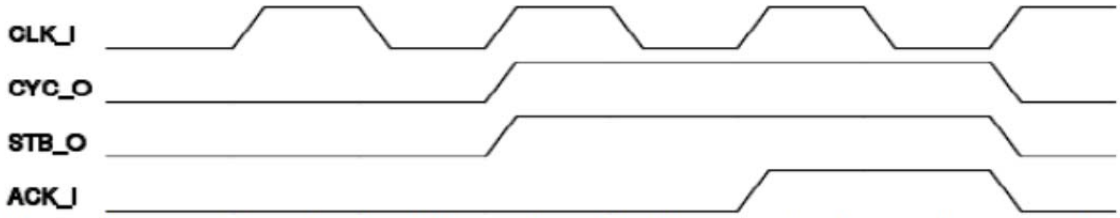

Illustration 3-3: Standard wishbone handshaking protocol, synchronous slave.

3.2.1 Classic standard SINGLE READ Cycle

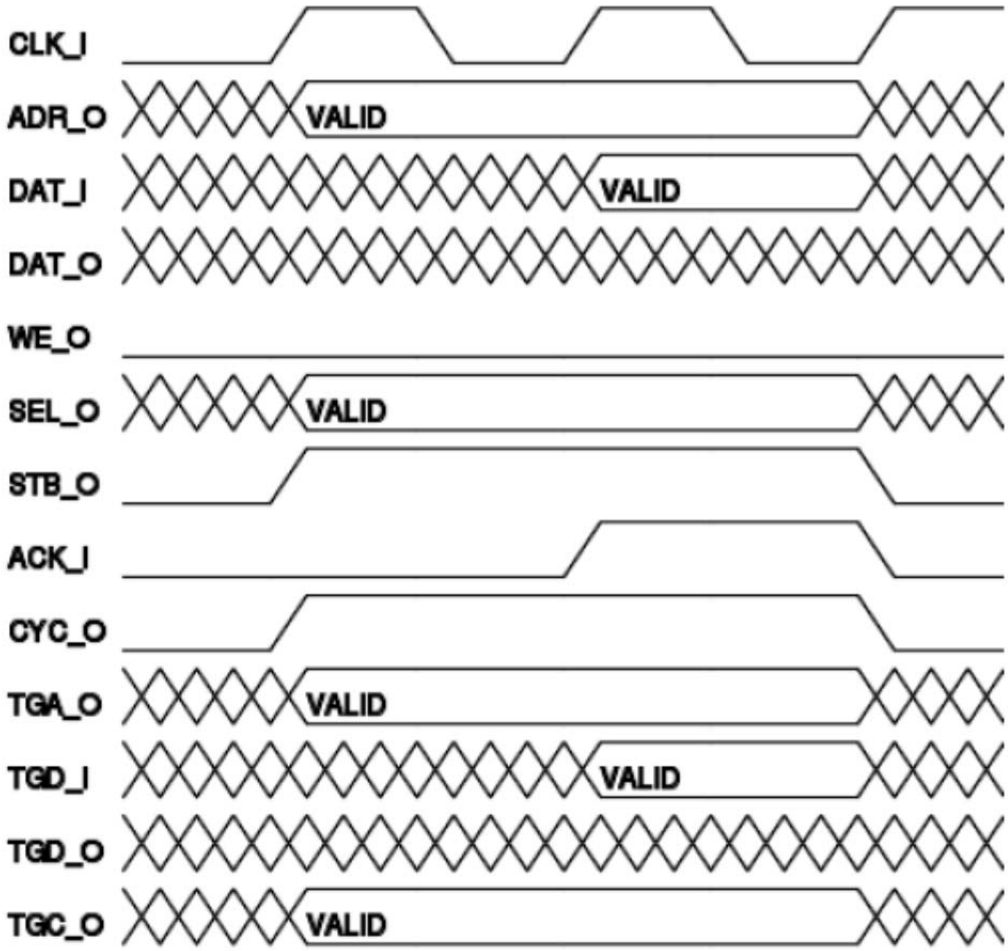

Illustration 3-5: Classic standard SINGLE READ cycle.

3.2.3 Classic standard SINGLE WRITE Cycle

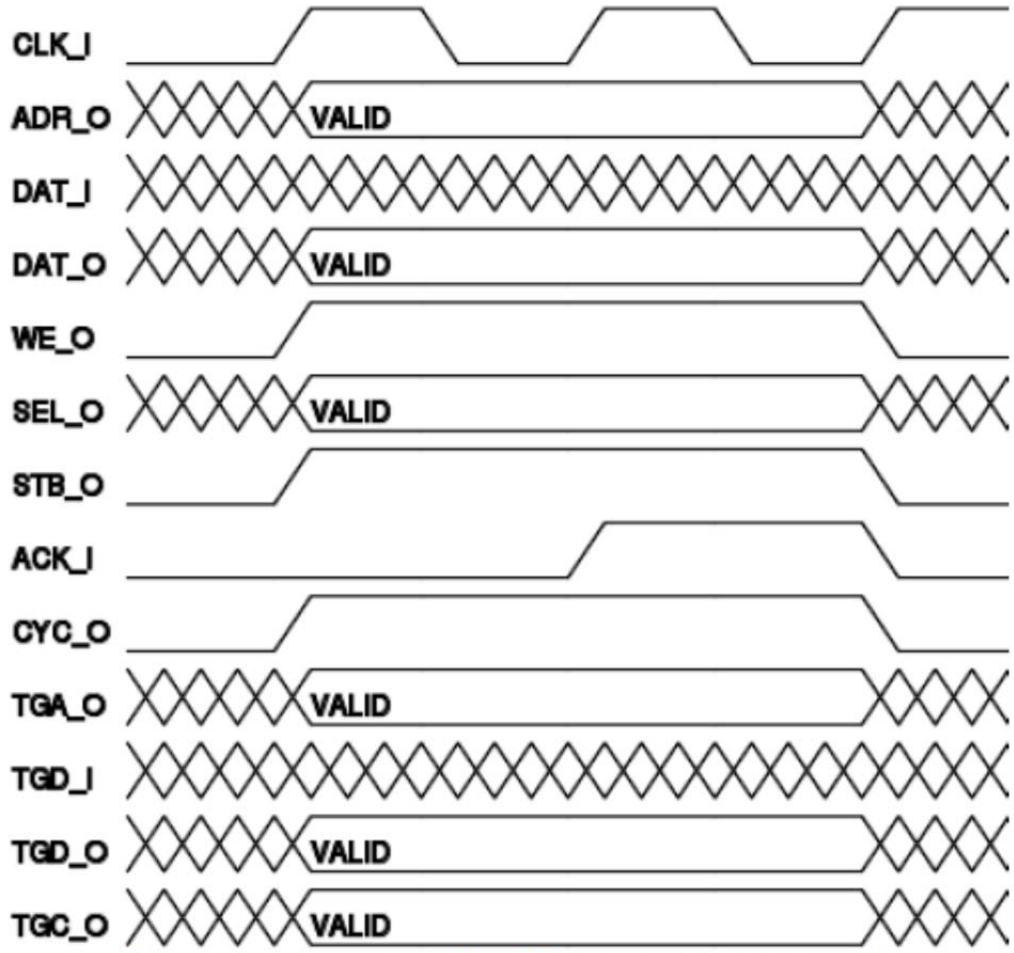

Illustration 3-7: Standard SINGLE WRITE cycle

3.3 BLOCK READ / WRITE Cycles

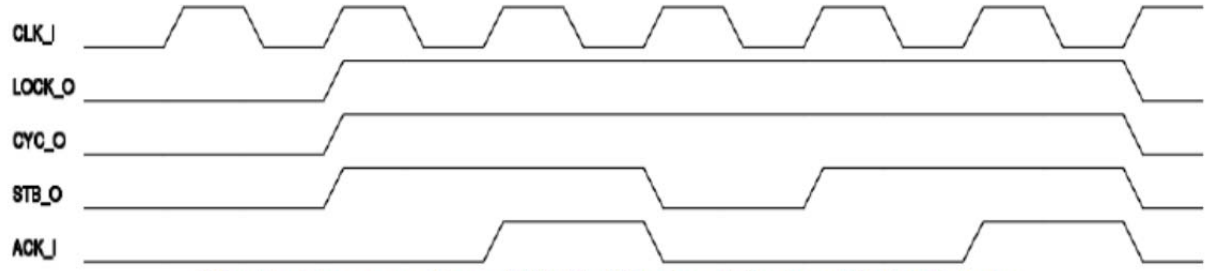

Illustration 3-9: Use of [CYC_O] signal during BLOCK cycles

3.3.1 BLOCK READ Cycle

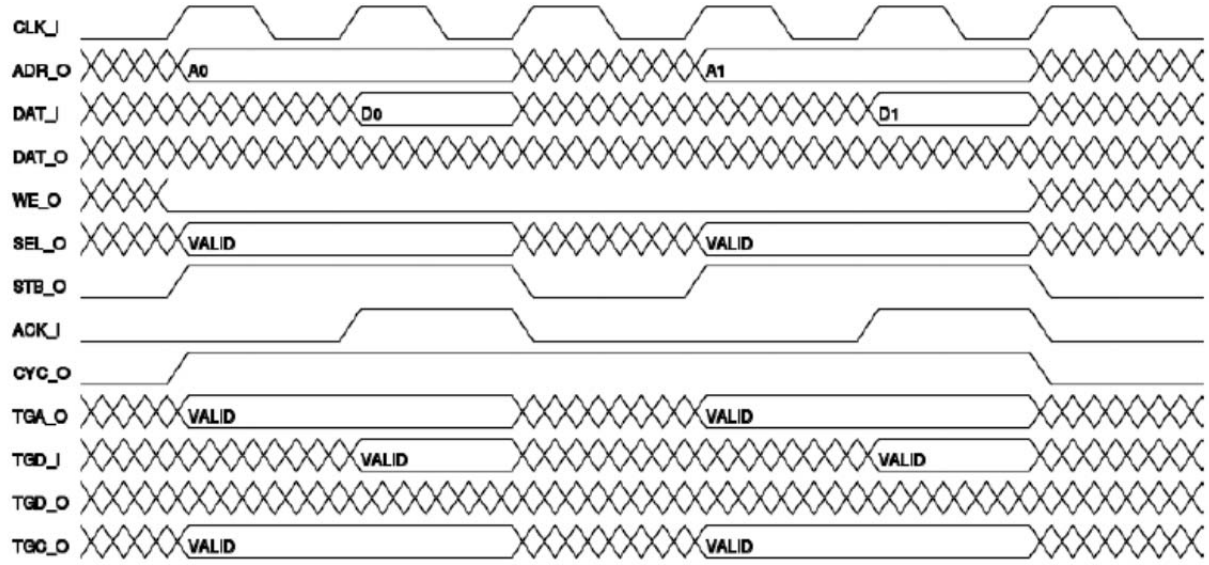

Illustration 3-10: Standard BLOCK READ cycle

3.3.2 BLOCK WRITE Cycle

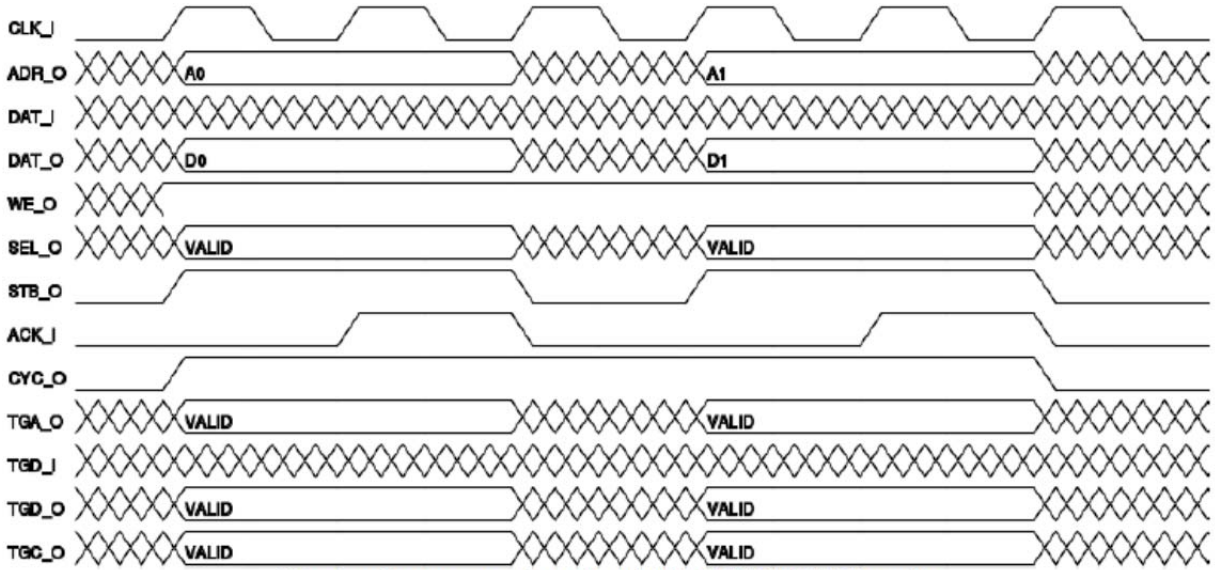

Illustration 3-12: Standard BLOCK WRITE cycle