OCP splitter

Functional specification

|  |  |
| --- | --- |
| **Ref** | OCP\_splitter\_func\_spec.doc |
| **Document version** | 0.1 |
| **Release date** | April 3th 2014 |
| **Department** | IP development |
| **Owner** | Ivan-Tolja Arsic |
| **Maturity level** | **Baseline (0.8)** |

**History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Notes** |
| 0.1 | April 03, 2014 | Arsic Ivan-Tolja | 1 |
|  |  |  |  |
|  |  |  |  |

NOTES:

|  |  |
| --- | --- |
| 1 | Initial version |
| 2 |  |
| 3 |  |

**Table of contents**

[1 Overview 7](#_Toc384306883)

[1.1 Introduction 7](#_Toc384306884)

[1.2 Module overview 7](#_Toc384306885)

[1.3 Feature list 7](#_Toc384306886)

[2 System interfaces 8](#_Toc384306887)

[2.1 Input OCP 32bits(output OCP 16bits) interface definition 8](#_Toc384306888)

[2.2 Interrupts 11](#_Toc384306889)

[2.3 PRCM interface and power management 11](#_Toc384306890)

[2.3.1 Reset 11](#_Toc384306891)

[2.3.2 Clock domains 11](#_Toc384306892)

[2.4 OCP input and OCP output port definition 12](#_Toc384306893)

**List of tables**

[Table 1 Reference documents 6](#_Toc384306865)

[Table 5 OCP 32 and OCP 16 ports 11](#_Toc384306866)

[Table 8 Input OCP 32bit port 12](#_Toc384306867)

[Table 9 Output OCP 16bit port 12](#_Toc384306868)

[Table 9 Clock reset interface 12](#_Toc384306869)

**List of figures**

[Figure 15 OCP write command timing 13](#_Toc384306894)

**Reference documents**

|  |  |  |  |
| --- | --- | --- | --- |
| **Ref** | **Title / source** | Owner | **Version** |
| [1] | OCP-IP 3.0 specification | Sonics | 1.0 |
| [2] |  |  |  |
| [3] |  |  |  |

Table Reference documents

# Overview

## Introduction

Section 13.1 of the IEEE standards style manual is adopted in this document.

* The word *shall* is used to indicate mandatory requirements strictly to be followed in order to conform to the standard and from which no deviation is permitted (shall equals *is required to*).
* The use of the word *must* is deprecated and shall not be used when stating mandatory requirements; must is used only to describe unavoidable situations.
* The use of the word *will* is deprecated and shall not be used when stating mandatory requirements; will is only used in statements of fact.
* The word *should* is used to indicate that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required; or that (in the negative form) a certain course of action is deprecated but not prohibited (should equals *is recommended that*).
* The word *may* is used to indicate a course of action permissible within the limits of the standard (may equals *is permitted*).
* The word *can* is used for statements of possibility and capability, whether material, physical, or causal (can equals *is able to*)

## Module overview

OCP splitter receives 32bit OCP (read/write) command and splits it into two 16bit OCP (read/write) commands.

Every OCP 32 write/read command on the input interface causes two OCP 16 write/read commands on the output interface.

This module supports just one transaction at the time – if one command is under process of splitting, the next one can’t be accepted on the input interface.

Next command on the input interface can be accepted even response phase for the current write transaction is not finished.

## Feature list

* + Target frequency: 304 MHz GS80 @ OPP100
  + Interfaces
    - OCP input interface:
      * OCP slave
      * 32-bit wide
      * OCP IP 3.0 compliant
    - OCP ouput interface
      * OCP master
      * 16-bits wide
      * OCP IP 3.0 compliant

# System interfaces

## Input OCP 32bits(output OCP 16bits) interface definition

The configuration for OCP 32bit (OCP 16bit) port is provided in the table below.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| OCPIP revision | 2.2 |  | | | |
| Parameter | Default | Type | Value | Tied Off? | Notes |
| Clock Control |  |  |  |  |  |
| enableclk | 0 | boolean | 1 |  |  |
| Data flow |  |  |  |  |  |
| Request |  |  |  |  |  |
| Address |  |  |  |  |  |
| addr | 1 | boolean | 1 |  |  |
| addr\_wdth |  | width | 32 |  |  |
| addrspace | 0 | boolean | 0 |  | 1 address space only |
| addrspace\_wdth |  | width |  |  |  |
| burstlength | 0 | boolean | 0 |  | No burst support. However, interface has default command accept and can handle one access every two functional clock cycles |
| burstlength\_wdth |  | width |  |  |  |
| burstseq | 0 | boolean |  |  |  |
| burstseq\_blck\_enable | 0 | boolean |  |  |  |
| burstseq\_dflt1\_enable | 0 | boolean |  |  |  |
| burstseq\_dflt2\_enable | 0 | boolean |  |  |  |
| burstseq\_incr\_enable | 1 | boolean |  |  |  |
| burstseq\_strm\_enable | 0 | boolean |  |  |  |
| burstseq\_unkn\_enable | 0 | boolean |  |  |  |
| burstseq\_wrap\_enable | 0 | boolean |  |  |  |
| burstseq\_xor\_enable | 0 | boolean |  |  |  |
| burstprecise | 0 | boolean |  |  |  |
| burstsinglereq | 1 | boolean |  |  |  |
| atomiclength | 0 | boolean |  |  |  |
| atomiclength\_wdth |  | width |  |  |  |
| blockheight | 0 | boolean |  |  |  |
| blockheight\_wdth |  | width |  |  |  |
| blockstride | 0 | boolean |  |  |  |
| blockstride\_wdth |  | width |  |  |  |
| burst\_aligned | 0 | boolean |  |  |  |
| reqlast | 0 | boolean |  |  |  |
| reqrowlast | 0 | boolean |  |  |  |
| Command |  |  |  |  |  |
| broadcast\_enable | 0 | boolean | 0 |  |  |
| rdlwrc\_enable | 0 | boolean | 0 |  |  |
| read\_enable | 1 | boolean | 1 |  | Reads supported |
| readex\_enable | 0 | boolean | 0 |  |  |
| write\_enable | 1 | boolean | 1 |  | Posted writes supported |
| writenonpost\_enable | 0 | boolean | 1 |  | Non poested writes supported |
| In-band request extensions |  |  |  |  |  |
| reqinfo | 0 | boolean | 0 |  |  |
| reqinfo\_wdth |  | width |  |  |  |
| connid | 0 | boolean | 0 |  |  |
| connid\_wdth |  | width |  |  |  |
| Request flow control |  |  |  |  |  |
| cmdaccept | 1 | boolean | 1 |  |  |
| sthreadbusy | 0 | boolean | 0 |  |  |
| Data |  |  |  |  |  |
| data\_wdth |  | width | 32(16) |  |  |
| byteen | 0 | boolean | 1 |  | Support unaligned BYTEEN for legacy reasons |
| force\_aligned | 0 | boolean | 0 |  |
| endian | little | endian | little |  |  |
| Write data |  |  |  |  |  |
| mdata | 1 | boolean | 1 |  | Needed to support writes |
| In-band write data extensions |  |  |  |  |  |
| mdatainfo | 0 | boolean | 0 |  |  |
| mdatainfo\_wdth |  | width |  |  |  |
| mdatainfobyte\_wdth |  | width |  |  |  |
| Write data phasing/flow control |  |  |  |  |  |
| datahandshake | 0 | boolean | 0 |  |  |
| mdatabyteen | 0 | boolean |  |  |  |
| reqdata\_together | 0 | boolean |  |  |  |
| datalast | 0 | boolean |  |  |  |
| datarowlast | 0 | boolean |  |  |  |
| dataaccept | 0 | boolean |  |  |  |
| sdatathreadbusy | 0 | boolean |  |  |  |
| sdatathreadbusy\_exact | 0 | boolean |  |  |  |
| sdatathreadbusy\_pipelined | 0 | boolean |  |  |  |
| Read data |  |  |  |  |  |
| sdata | 1 | boolean | 1 |  | Needed to support reads |
| In-band read data extensions |  |  |  |  |  |
| sdatainfo | 0 | boolean | 0 |  |  |
| sdatainfo\_wdth |  | width |  |  |  |
| sdatainfobyte\_wdth |  | width |  |  |  |
| Response |  |  |  |  |  |
| resp | 1 | boolean | 1 |  | Response group required |
| writeresp\_enable | 0 | boolean | 1 |  | Needed for error reporting and non posted writes |
| resplast | 0 | boolean |  |  |  |
| resprowlast | 0 | boolean |  |  |  |
| In-band response extensions |  |  |  |  |  |
| respinfo | 0 | boolean | 0 |  |  |
| respinfo\_wdth |  | width |  |  |  |
| Response flow control |  |  |  |  |  |
| respaccept | 0 | boolean | 0 |  |  |
| mthreadbusy | 0 | boolean | 0 |  |  |
| mthreadbusy\_exact | 0 | boolean |  |  |  |
| mthreadbusy\_pipelined | 0 | boolean |  |  |  |
| Tagging |  |  |  |  |  |
| tags | 1 | positive | 1 |  | No tag support |
| taginorder | 0 | boolean |  |  |  |
| tag\_interleave\_size | 1 | positive |  |  |  |
| Threading |  |  |  |  |  |
| threads | 1 | positive | 1 |  | Single threaded interface |
| Sideband |  |  |  |  |  |
| Master sideband |  |  |  |  |  |
| merror | 0 | boolean | 0 |  |  |
| mflag | 0 | boolean | 0 |  |  |
| mflag\_wdth |  | width |  |  |  |
| mreset |  | boolean | 0 |  |  |
| Slave sideband |  |  |  |  |  |
| interrupt | 0 | boolean | 0 |  | Don't use OCP-interrupts. |
| serror | 0 | boolean | 0 |  | In band error reporting used instead |
| sflag | 0 | boolean | 0 |  |  |
| sflag\_wdth |  | width |  |  |  |
| sreset |  | boolean | 0 |  | No functional need |
| Control sideband |  |  |  |  |  |
| control | 0 | boolean | 0 |  |  |
| control\_wdth |  | width |  |  |  |
| controlwr | 0 | boolean |  |  |  |
| controlbusy | 0 | boolean |  |  |  |
| Status sideband |  |  |  |  |  |
| status | 0 | boolean | 0 |  |  |
| status\_wdth |  | width |  |  |  |
| statusrd | 0 | boolean |  |  |  |
| statusbusy | 0 | boolean |  |  |  |
| Test |  |  |  |  |  |
| scanport | 0 | boolean | 0 |  |  |
| scanport\_wdth |  | width |  |  |  |
| scanctrl\_wdth | 0 | width |  |  |  |
| clkctrl\_enable | 0 | boolean | 0 |  |  |
| jtag\_enable | 0 | boolean | 0 |  |  |
| jtagtrst\_enable | 0 | boolean |  |  |  |

Table OCP 32 and OCP 16 ports

## Interrupts

Interrupt definition – TBD.

## PRCM interface and power management

### Reset

An active low asynchronous hardware reset is provided by the PRCM. **HW resets are executed unconditionally and may therefore violate the OCP protocol.**

### Clock domains

OCP splitter receives one functional clock from PRCM that’s also used for both OCP ports. The maximum frequency is 304 MHz.

## OCP input and output port definition

The following tables summarize the OCP interfaces signals.

|  |  |  |
| --- | --- | --- |
| **Signal name** | **IO** | **Description** |
| m\_cmd\_in\_32 | I[2:0] | OCP command on the input interface |
| m\_addr\_in\_32 | I[31:0] | OCP 32 bits address on the input port |
| m\_data\_in\_32 | I[31:0] | OCP 32 input data |
| s\_cmd\_accept\_out\_32 | O[31:0] | command accept active when both transactions are accepted on the 16bit OCP output interface |
| s\_data\_out\_32 | O | 32 bit output data - used for read command |

Table Input OCP 32bit port

|  |  |  |
| --- | --- | --- |
| **Signal name** | **IO** | **Description** |
| m\_cmd\_out\_16 | O[2:0] | OCP command on the input interface |
| m\_addr\_out\_16 | O[31:0] | OCP 16 bits address on the output port |
| m\_data\_out\_16 | O[16:0] | OCP data out bus, used for write command |
| s\_cmd\_accept\_in\_16 | I | OCP command accept on the output interface |
| s\_data\_in\_16 | I[16:0] | OCP data out bus, used for read command |

Table Output OCP 16bit port

|  |  |  |
| --- | --- | --- |
| **Signal name** | **IO** | **Description** |
| pi\_ocp\_rstn | I | Asynchronous hardwar reset active low |
| pi\_ocp\_clk. | I | Ocp clock input pin(304 MHz) |

Table 9 Clock reset interface

The figures below show timing diagrams for WRITE and READ transaction on both OCP ports.

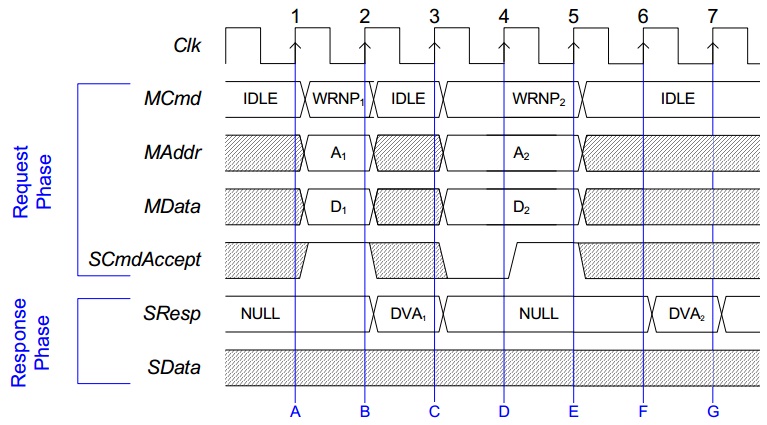


Figure OCP write command timing

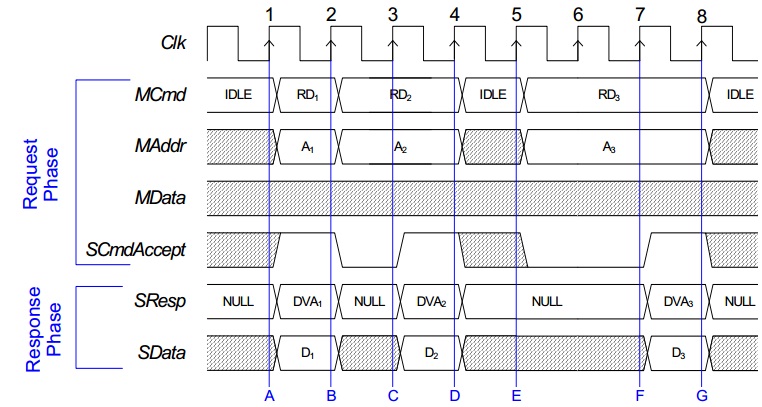


Figure 16 OCP read command timing