OpenRISC ASIC Requirement Specification

This document describes the functional requirement of the OpenRISC based ASIC

Michael Unneback
michael.unneback@orsoc.se
Table of Contents

1 Introduction .................................................................................. 1

2 Block Diagram .................................................................................. 2
  2.1 OpenRISC ........................................................................... 3
  2.2 Display controller .................................................................... 3
  2.3 Ethernet MAC ......................................................................... 4
  2.4 USB host/slave ......................................................................... 4
  2.5 SDHC controller ....................................................................... 4
  2.6 PCI .......................................................................................... 4
  2.7 Low Pin Count - LPC ................................................................. 5
  2.8 DDR2 SDRAM controller ......................................................... 5

3 Roadmap .......................................................................................... 6
1 Introduction

The design goal is to present a Linux enabled SoC design. The design will be based on open source IP available from OpenCores. Emphasis will be on performance and connectivity.

Design is centered around the popular OpenRISC, a 32 bit general purpose RISC processor. This processor comes with bleeding edge Linux and GCC support and does also incorporate a JTAG debug interface.

System bus is a Wishbone B3 compatible implementation supporting simultaneous open channels and multiple clock domains. This will enable high throughput for memory intensive functions such as display controller and Gigabit Ethernet.

External functions can easily be added over PCI or via a small pin count interface. In this way the design could be adopted to a large variety of uses.
2 Block Diagram

### Name Description

**OpenRISC**
32 bit RISC processor with instruction and data cache and MMU

**JTAG debug**
JTAG debug port

**GBE MAC**
Connects to an Gigabit Ethernet PHY over GMMI

**USB host**
Standard USB 1.1 host function

**SDHC**
SD FLASH card controller

**PCI**
PCI 32bit 33 MHz or 66 MHz

**DDR2 SDRAM**
DDR2 SDRAM controller with support for multiple clock domains with built in Wishbone arbiter

**BootROM**
Internal BootROM capable of loading software from an external SPI FLASH

Peripherals on IO Wishbone segment. This wishbone segment is its own clock domain, 33.333 MHz. All clock dividers are dependent on this clock and have no relation to the system wishbone clock.

### Name Description

**UART0, UART1**
16550 compatible standard UART

**I2C**
I2C host controller

**SPI**
SPI master controller

**AC’97**
AC’97 compatible audio controller

**LPC**
Intel defined low pin count interface
Also available is a LPC port using an external clock. This port can be used for high bandwidth external connections.

2.1 OpenRISC

The OR1200 is a 32-bit scalar RISC with Harvard microarchitecture, 5 stage integer pipeline, virtual memory support (MMU) and basic DSP capabilities.

![OpenRISC Block Diagram]

Default caches are 1-way direct-mapped 8KB data cache and 1-way direct-mapped 8KB instruction cache, each with 16-byte line size. Both caches are physically tagged.

By default MMUs are implemented and they are constructed of 64-entry hash based 1-way direct-mapped data TLB and 64-entry hash based 1-way direct-mapped instruction TLB.

Supplemental facilities include debug unit for real-time debugging, high resolution tick timer, programmable interrupt controller and power management support.

- Central CPU/DSP block
- IEEE 754 compliant single precision FPU
- Direct mapped data cache
- Direct mapped instruction cache
- Data MMU based on hash-based DTLB
- Instruction MMU based on hash-based ITLB
- Power management unit and power management interface
- Tick timer
- Debug unit and development interface
- Interrupt controller and interrupt interface
- Instruction and Data WISHBONE B3 compliant interfaces

IP available from OpenCores: http://opencores.org/openrisc,overview

2.2 Display controller

The OpenCores VGA/LCD Controller core is a WISHBONE revB.3 compliant embedded VGA core capable of driving CRT and LCD displays. It supports user programmable resolutions and video timings, which are limited only by the available WISHBONE bandwidth. Making it compatible with almost all available LCD and CRT displays.

The core supports a number of color modes, including 32bpp, 24bpp, 16bpp, 8bpp gray-scale, and 8bpp-pseudo color. The video memory is located outside the primary core, thus providing the most flexible memory solution. It can be located on-chip or off-chip, shared with the systems main memory (VGA on demand) or be dedicated to the VGA system. The color lookup table
is, as of core version 2.0, incorporated into the color-processor block.

Pixel data is fetched automatically via the Wishbone revB.3 Master interface, making this an ideal program-and-forget video solution. More demanding video applications like streaming video or video games can benefit from the video-bank-switching function, which reduces flicker and cluttered images by automatically switching between video-memory pages and/or color lookup tables on each vertical retrace.

The core can interrupt the host on each horizontal and/or vertical synchronization pulse. The horizontal, vertical and composite synchronization polarization levels, as well as the blanking polarization level are user programmable.

IP available from OpenCores: http://opencores.org/project,vga_lcd

2.3 Ethernet MAC
This design support up to two external standard Gigabit Ethernet PHY devices connected over GMMI interface. External interface can also be configured to support RMMI for 10/100 Mbps with reduced pin count. This interface can be used with, for instance, Micrel Ethernet PHY KSZ8021 in a 4x4 mm package to enable a small footprint solution.

2.4 USB host/slave
This design supports up to two external USB host or slave ports. Interface is compatible to USB 1.1.

IP available from OpenCores: http://opencores.org/project,usbhostslave

2.5 SDHC controller
The core is a combined SD/SDHC controller, for Secure Digital-card. Two designs is available, one full-feature core utilizing DMA and one smaller for PIO. The idea with the full-feature design is that it should provide as much performance as possible. Therefore it’s built to stall the CPU as little as possible and offload it some computations, this is achieved by:

- Error and flow control performed mainly in hardware i.e user specify when sending a command what kind of error check he wants to be performed. Result is then set in the response registers also can be set to generate interrupt.
- Command generation for writing/reading block of data is performed in hardware
- Buffer descriptors is used to to queue read/write data transmissions (Less delay between data transmissions)
- DMA for minimal CPU interruption as possible.

IP available from OpenCores: http://opencores.org/project,sdcard_mass_storage_controller

2.6 PCI
Peripheral Component Interconnect, PCI, is an industry standard peripheral bus. This implementation supports 33 or 66 MHz bus clock.

<table>
<thead>
<tr>
<th>Adr/Data bus size</th>
<th>Clock</th>
<th>Throughput</th>
</tr>
</thead>
<tbody>
<tr>
<td>32 bit</td>
<td>33 MHz</td>
<td>133 MB/s</td>
</tr>
<tr>
<td>32 bit</td>
<td>66 MHz</td>
<td>266 MB/s</td>
</tr>
</tbody>
</table>

IP available from OpenCores: http://opencores.org/project,pci
2.7 Low Pin Count - LPC

The Low Pin Count bus, or LPC bus, is used to connect moderate bandwidth devices to the ASIC. This Intel defined standard can be used for the following external resources:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>low speed serial IO</td>
<td>I2C, SPI, UART, CAN</td>
</tr>
<tr>
<td>audio</td>
<td>S/PDIF, AC’97, I2S</td>
</tr>
<tr>
<td>user i/f</td>
<td>keyboard, mouse, IR</td>
</tr>
<tr>
<td>analog</td>
<td>ADC, DAC</td>
</tr>
</tbody>
</table>

This module is driven by an external clock.

IP available from OpenCores: [http://opencores.org/project,wb_lpc](http://opencores.org/project,wb_lpc)

2.8 DDR2 SDRAM controller

The Versatile memory controller is a design supporting different type of external memory devices. This implementation uses DDR2. This designs support multiple clock domains. The controller and the memory itself is in one domain called sdram_clk, this is limited by the memory typically ~600 MHz and by the IO of the ASIC. The controller can have any number of groups of wishbone interfaces where each group is its own clock domain. In that way one group is the system wishbone clock, wb_clk, clock domain and others include ethernet MAC and display controller.

IP available from OpenCores: [http://opencores.org/project,versatile_mem_ctrl](http://opencores.org/project,versatile_mem_ctrl)
3 Roadmap

The goal of this project is to produce the following:

1. an affordable ASIC supporting all or a subset of the functions outlined in this document
2. a small formfactor module with the above ASIC and supported peripherals

As a prototype both for the ASIC and the small formfactor module an ALTEGRA Cyclone IV board is under development. Estimated size of that board is 70 x 36 mm. The board will include the following hardware functions:

<table>
<thead>
<tr>
<th>Function</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>EP4C22</td>
<td>FPGA with 22K LUT, n x 9kbit RAM</td>
</tr>
<tr>
<td>MT47H128M8CF</td>
<td>128 Mbyte DDR SDRAM</td>
</tr>
<tr>
<td>W25Q64BV</td>
<td>8 MByte SPI FLASH</td>
</tr>
<tr>
<td>KSZ9021</td>
<td>GigBit Ethernet PHY</td>
</tr>
<tr>
<td>USBT11</td>
<td>USB PHY</td>
</tr>
<tr>
<td>Combo</td>
<td>Combo connector with dual USB and RJ45 with built-in magnetics and LED</td>
</tr>
<tr>
<td>SD conn</td>
<td>SD connector for SDHC FLASH card</td>
</tr>
<tr>
<td>IO connector 1</td>
<td>SAMTEC 2x35 pin board to board connector, display controller</td>
</tr>
<tr>
<td>IO connector 2</td>
<td>SAMTEC 2x35 pin board to board connector, usage PIC, LPC, AC’97</td>
</tr>
<tr>
<td>FT4232H</td>
<td>Four channel USB serial, JTAG debug, SPI, UART, UART</td>
</tr>
<tr>
<td>USB conn</td>
<td>USB type B connector used with FT4234</td>
</tr>
<tr>
<td>EP5328</td>
<td>DC/DC 3.3V, 800mA, FPGA IO</td>
</tr>
<tr>
<td>EP5328</td>
<td>DC/DC 1.8V, 800mA, DDR2</td>
</tr>
<tr>
<td>EP5328</td>
<td>DC/DC 1.2V, 800mA, FPGA core</td>
</tr>
<tr>
<td>SP6201</td>
<td>linear DC/DC 2.5V, 100mA, FPGA PLL analog supply</td>
</tr>
</tbody>
</table>

The above prototype board can be used standalone or together with an expansion board. An expansion board with the following functions are planned:

<table>
<thead>
<tr>
<th>Function</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gigabit ETH</td>
<td>Realtek RTL8110S-32, PCI connected Gigabit Ethernet PHY/MAC with CPU offload</td>
</tr>
<tr>
<td>10/100 Fast Ethernet switch</td>
<td>Micrel KSZ8842-PMBL 2+1 port switch, PCI</td>
</tr>
<tr>
<td>USB2.0</td>
<td>5 port USB 2.0, PCI</td>
</tr>
<tr>
<td>FPGA</td>
<td>LPC IO expander, PCI arbiter</td>
</tr>
<tr>
<td>HDMI</td>
<td>Chrontel CH7301C HDMI driver</td>
</tr>
</tbody>
</table>