USB Dual-Mode Function IP Core Development

YAN Yong-ming, ZENG Yun
Department of Applied Physics, Hunan University Changsha 410007 China

Abstract This paper describes the specification and implement of a Universal Serial Bus (USB) dual-mode function IP core used for embedded system. Controlled by micro controller/CPU, the novel IP core can function as a USB host controller or USB peripheral controller. When configured as a USB host controller, it supports all USB 1.1 transaction types; supports automatic preamble insertion, and automatic SOF generation and transmission. Otherwise, when it is configured as a USB device by a microprocessor, it operates as a USB peripheral controller compliant with USB2.0 specification.

Key words universal serial bus; on-the-go; intellectual property

Announced on December 18th 2001 by the USB Implementers Forums, USB On-The-Go (OTG) is a new supplement to the USB 2.0 specification that augments the capability of existing mobile devices and USB peripherals by adding host functionality for connection to USB peripherals. Because OTG supplement gives portable devices a cost-effective means of conducting point-to-point communications using USB protocols, many USB OTG ASIC (Application-Specific Integrated Circuits) chips such as TD1120 (Trans Dimension, Inc.), ISPI761 (Philips, Inc.) and IP cores such as CI12323ug (ChipIdea, Inc.), MUSBFDRC (Metro Graphic, Inc.) are developing by famous silicon & IP vendors in recent years. In 2005, some OTG devices such as digital cameras, PDAs, and MP4 players (for example, Gmini-402, Archos, Inc. France) are available in worldwide market. Due to popularity of System-On-Chip (SOC), there are much more USB OTG IP cores announced than USB OTG single-chips. Typically, there are two kinds of USB OTG IP cores: UTMI interface IP cores such as DWC-usb2-hsotg-phy (announced by Synopsys, Inc.) which complete mixed-signal physical layer for USB 2.0 OTG applications; USB OTG controller IP cores such as CUSBOTG2 (by CAST, Inc.) which provide total performance USB OTG compliant functionality. Together with software support such as host controller driver, the cooperation of an UTMI interface IP core and an USB OTG controller IP core offer a complete hardware and software USB OTG solution.

This paper discusses a soft IP core called USB dual-mode function IP core described by verilog HDL (Hardware Description Language). This IP core is different from above two kinds of IP cores, which is especially targeted for utilization in embedded systems compliant with Wishbone architecture specification.

1 Dual-Mode Function IP Core Architecture

1.1 The Hardware Architecture

The hardware of USB dual-mode IP core consists of eight major functional blocks (see Fig.1). There are two scenarios to consider. First this USB IP core can be configured as a USB host, and secondly it can be configured as a USB peripheral. The article introduces the functionality of eight modules and describes how these blocks act in host/peripheral mode.

1.2 Two Common Blocks

1.2.1 USB OTG PHY (part1)

As is mentioned above, USB OTG PHY (part 1) and HostSlaveMux (Part 2) are common pipelines for host and peripheral. On the transmit path, USB OTG PHY implements sync insertion, CRC calculation and insertion, parallel to serial conversion, bit stuffing, and NRZI encoding. On the receive path, USB OTG PHY implements connection state detection, sync detection and stripping, clock recovery, NRZI decoding, bit de-stuffing, CRC calculation and checking, and serial to parallel conversion. The architecture of USB OTG PHY is shown in Fig.2.
1.2.2 HostSlaveMux (part 2) HostSlaveMux which includes system configuration and control registers allows host SIE and slave SIE to share access to the USB OTG PHY.

Wishbone interface

Part one

USB transceiver

Part two

Part three

Fig. 1 USB IP core architecture

Part four

Part five

Part six

Part seven

Wishbone architecture

USB1.1 Host SIE

USB2.0 Host controller

USB2.0 Peripheral SIE

Configuration controller

RAM

SIE

Peripheral

USB 2.0

USB 1.1

Input synchronizer

K/J/SEE change?

DPLL

Synch pattern finder

NRZI decoder

S-P converter

Output enable logic

NRZI decoder

EOP append logic

Synch pattern append

Bit stuffer

EP0

Configuration controller

Fig. 2 The architecture of USB OTG PHY(part 1)

1.3 Host Mode Blacks

There are four modules which supports the lowest level of the USB 1.1 host function in this USB IP core, which is also responsible for USB data I/O and control communications\[4\]. 1) USBHostSIE (part 3) supports the USB 1.1 host specific portion of the USB 1.1 protocol layer. 2) USBHostControl (part 4) includes USB host logic, USB host controller registers and memory blocks. These three parts of USBHostControl together with software support, function automatic preamble insertion, automatic SOF generation and transmission, and all USB 1.1 transaction types: bulk, setup, interrupt, and isochronous. 3) HostRx/HostTx (part 5) is two FIFOs which restore and exchange data with microprocessor. 4) wishbone Interface provides Wishbone compatible interface to host/slave controllers and the transmit/receive FIFOs.

Let's consider the host configuration first. On power up, when the dual-mode function IP core is connected to a USB device, it will report the connection event and the connection speed to the microprocessor through Wishbone architecture—a kind of SOC on-chip architecture. Now the microprocessor configures the system configuration and control registers with the following dual-mode function IP core parameters: USB speed, USB polarity, the USB address of the attached device (address zero is the default device address), the USB endpoint (endpoint zero is used for set up) within the attached device. Now that the host has been configured, the host can attempt to complete a transfer with the attached USB device.
1.4 Peripheral Mode Blacks

Now let’s take a look at the case where USB dual-mode function IP core is configured as a USB device. On power up, the microprocessor configures this IP core as a USB device[5].

Fig.3 Architecture of USB peripheral SIE

1.4.1 USB2.0 Peripheral SIE (part 6)

USB2.0 Peripheral SIE supports the lowest level of the USB 2.0 protocol layer. This protocol layer is responsible for all USB data IO and control communication which mainly includes five blocks. 1) DMA and memory interface provides random memory access and also DMA block transfers. This block also performs pre-fetching when byte misaligned data must be written (RMW cycles). 2) protocol engine handles all the standard USB protocol handshakes, such as SOF tokens and acknowledgment of data transfers (ACK, NACK, NYET), replying to PING tokens. 3) packet assembly assembles packets and places them in to the output FIFO. It first assembles the header, inserting a proper PID and check sums, then adds a data field if requested. 4) packet disassembly decodes all incoming packets and forwards the decoded data to the appropriate blocks. The decoding includes extracting of PID and sequence numbers, as well as header check sum checking. 5) UTMI I/F is the interface block to the UTMI compliant PHY (transceiver). The architecture of USB peripheral SIE is shown Fig.3.

1.4.2 DoubleBuffer and DMA Controller (part 7)

DoubleBuffer and DMA Controller connect with Wishbone architecture and SSRAM. This USB core supports a double buffering feature which reduces the latency requirements on the functions micro controller and driver software. Double buffering is enabled when all buffer pointers have been set. Data is being retrieved/filled from/to the buffers in a round robin fashion. When data is sent to/from an endpoint, first buffer 0 is used. When the first buffer is empty/full, the function controller may be notified via an interrupt. The function controller can refill/empty buffer0 now. The USB core will now use buffer 1 for the next operation. When the second buffer is full/empty, the function controller is interrupted, and the USB core will use buffer 0 again, and so on. Any buffer that is not allocated will be skipped. A buffer that has the used bit set will cause the core to stall, replying with NAK/NYET acknowledgments to the host. This procession is shown in Fig.4.

Fig.4 Buffering procession

2 Test of USB Dual-Mode Function IP Core

Using the methods and design flow for system-on-programmable-chip (SOPC[6]), we design a special MCU for test of USB dual-mode function IP core. This MCU mainly includes three blocks: a dual-mode function IP core, a MC8051[7] IP core and an UART IP core. The MC8051 IP core is a
microprocessor which controls, configures and exchanges data with the dual-mode function IP core, and the UART IP core is a hardware debugger and a software debugger by transmitting the verification data to a personal computer. This MCU is integrated and realized on a FPGA chip (EP1C12Q240C8), and the hardware architecture of test board for dual-mode function IP core is shown in Fig.5.

![Fig.5 Hardware architecture of test board for dual-mode USB IP core](image)

### Tab.1 Difference between this IP Core and a popular USB OTG Controller

<table>
<thead>
<tr>
<th>IP Core</th>
<th>Hardware characters</th>
<th>Software support</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>A popular USB OTG controller</td>
<td>Complex hardware architecture includes root Hub, Vbus controller and HNP/SRP logic inside. Gate count about 40K gates. UTMI+PHY IP core needed.</td>
<td>Running under most real time operating systems such as palm OS, WinCE, Linux needing memory &gt;500 K.</td>
<td>Multi-ports, true dual-role capability, LS-FS host capability, LS-FS-HS peripheral capability.</td>
</tr>
</tbody>
</table>

The test of host mode of the dual-mode function IP: by using the test board, we have tested USB host controls mode operation, automatic preamble insertion, automatic SOF generation and transmission, operating in conjunction with Host Controller Driver\(^8\) (HCD) software and translating data with another USB device in all USB 1.1 transaction types with low-speed (LS) and Full-Speed (FS) mode.

Peripheral mode of USB dual-mode function IP core has 7 endpoints: EP0-control (in and out), EP1-bulk in, EP2-bulk out, EP3-interrupt in, EP4-interrupt out, EP5-ioschronous in, and EP6-ioschronous out. By using this test board, a PC, a USB WDM device driver developed by using software package DDK (Driver Development Kits)\(^9\) and Windriver (provided by Jungo Ltd.), we have tested all four transaction types: bulk, control, interrupt, and ioschronous in FS and LS mode of USB peripheral.

The main test data we have received and conclusion resulting from comparing this IP core and a popular USB OTG controller IP core are shown in Tab.1.

As shown above, the USB dual-mode function IP core has simpler hardware architecture, smaller software programs, fewer gate counts—about 40% of an ordinal USB OTG controller IP core plus an UTMI PHY IP core, and is easier for utilization in small embedded systems, especially for SOC.

### References


(Continued on page 76)