INTERFACING MIXED SIGNAL PERIPHERALS BY PROTOCOLS OF PACKET TYPE

Emil Gueorguiev Saramov, Angel Nikolaev Popov

Computer Systems Department, Technical University of Sofia, Kliment Ohriski blvd. No.8, 1797 Sofia, Bulgaria phone: +359 2 9653254, phone: +359 2 9652017, e-mail: egs@tu-sofia.bg

Keywords: mixed signal interface, FIFO, USB, endpoint

This paper presents several hardware-software approaches for optimization of mixed signal interface. The investigation is restricted to interfaces that use high speed Universal Serial Bus (USB 2.0). The paper is focused on the cases with data transfer direction from the peripheral device to the host computer. The data transfer rates of the proposed two-level FIFO structure for synchronous and asynchronous interfaces, for double, triple and quad buffering at the second level FIFO and for different host operating systems are compared. The data transfer path of the device side is configured by software, but the data path does not include software that transfers or processes data. The paper systematizes the transfer rates limitation factors for the examined cases.

1. INTRODUCTION

Mixed signal peripheral devices impose specific requirements on the interfacing to the host computer: data transfer rate, latencies, remote mixed signal units, hardware minimization and others.

The design of the peripheral interface is determined mainly by data transfer rate. It must handle a data flow with high average transfer rate. The high-speed differential lines dominate as a physical level of the interface [1].

The latencies introduced by the interface are relatively independent from data transfer rates. Latencies within the peripheral interface worsen the timings. They are critical in many situations and should be estimated and keep low.

Remote or at least outside PC mixed signal units is a key requirement when interfacing high-precision, high-resolution or high-speed analog modules.

Minimization of the hardware part of the interface is persistent in new hardware designs. The functions are transferred to the software, but data transfer paths should involve as little as possible software [5].

The requirements are significantly different for the application areas: test and measurement instruments [1], [2] (DSO, MSO, logic analyzers, protocol analyzers, data generators, AFG), digital video, industrial process control and monitoring.

The basic interface structure that is examined in the paper, targets test and measurement systems. High-speed USB is chosen as a low level interface. This determines the architecture of the system as single host - multiple devices and ensures low complexity/cost of the interface hardware. An important problem is to optimize the hardware and software in order to maximize the sustained real data rate. The transfer protocols are of packet type with hardware level of data integrity checks and retries.

2. RECONFIGURABLE MIXED SIGNAL TEST SYSTEM

2.1. Hardware

A reconfigurable mixed signal system is used for interface examination. The block diagram of the hardware is shown on Fig. 1. USB device controller is Cypress EZ-USB FX2. It provides innovative and flexible peripheral side interface, configurable by software as slave FIFO or General Programmable Interface (GPIF), [5]. It is essential that the CPU can exclude itself completely from the data path and can leave the hardware to handle the data flow.

An important part of the system is FPGA, [6], interconnected with ZBT/NOBL synchronous SRAM (Fig. 1). The master side of the peripheral interface is thoroughly reconfigurable [3] and controls the slave FIFO of the USB device controller. The master generates read and write command signals for the slave. The clock for the master is received either from the analog control unit or from USB device controller.

The mixed signal system involves two high-speed 8-bit analog input interfaces and a high-speed bi-directional digital interface. The data transfer rate for these interfaces is significantly higher than the slave FIFO can handle. One solution of this problem is to introduce another FIFO level, marked on the block diagram as level 1.

2.2. Software

The software for the tests involves 3 components:

- Firmware for FX2 CPU, downloaded either from PC during FX2 initialization, or from I^2C EEPROM. The firmware initializes USB and peripheral interfaces of FX2. CPU in all tests is in AutoIn modes and it is excluded from data path.
- PC drivers: USB drivers of the operating system (EHCI driver, usbd.sys); general purpose FX2 device driver. In the current tests general purpose driver downloads the firmware into FX2 RAM. USB Test and Measurement Class (USBTMC) is not used.
- Test program. It is a MFC based application that measures the transfer times then calculates the data rates and write them to a file.

3. QUANTUM FIFO AND DOUBLE, TRIPLE AND QUAD BUFFERING

The dual port endpoint RAM is partitioned into blocks (256x16 in the cases of the tests). Each block is a FIFO, connected to the peripheral bus or to the SIE [5]. The connections change when one of the buffers is full and the other – empty (Fig. 2). The transition between the states does not insert latency. Double buffering is shown on Fig. 2. Triple and quad buffering use the same algorithm, but one/two FIFO blocks are added for triple/quad buffering. Drawback of the quantum FIFO is the overhead when sending data that does not fill entirely the FIFO – by activation of PKTEND input [5] or other methods. The FIFO level 1 can be of quantum type and double buffering, but the algorithms must be coordinated with data encoders (Fig. 1) that access FIFO level 1 more than once per word.

4. DATA TRANFER RATE MEASURING CONDITIONS

The data transfer rates are obtained by measuring the time, necessary to receive 10 MB for each point of the curves.

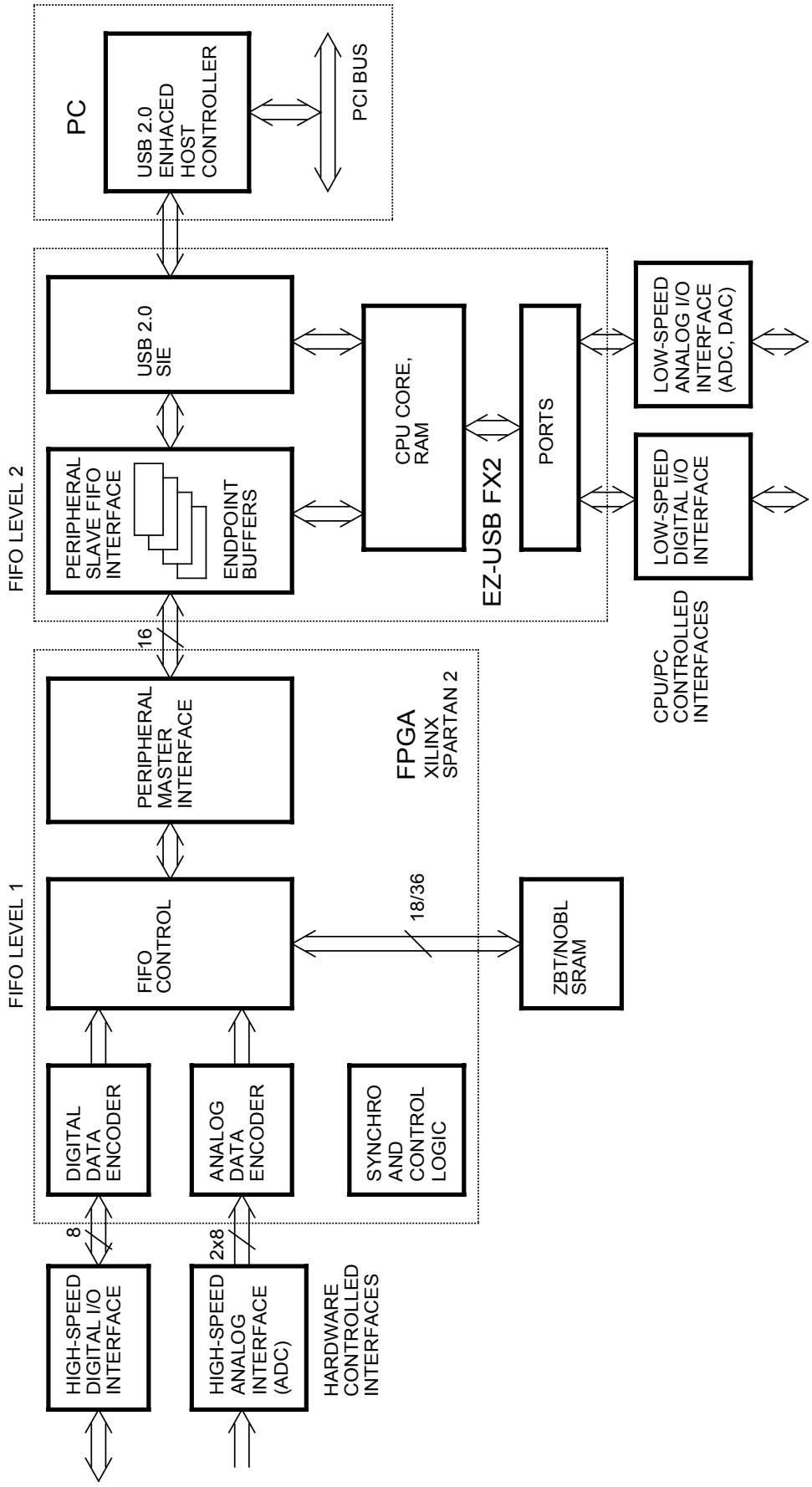
PC
USB 2.0 ENHACED HOST CONTROLLER
PCI BUS
USB 2.0 SIE
CPU CORE, RAM
PORTS
LOW-SPEED ANALOG I/O INTERFACE (ADC, DAC)
LOW-SPEED DIGITAL I/O INTERFACE
CPU/PC CONTROLLED INTERFACES
FIFO LEVEL 2
PERIPHERAL SLAVE FIFO INTERFACE
ENDPOINT BUFFERS
EZ-USB FX2
16
PERIPHERAL MASTER INTERFACE
FPGA XILINX SPARTAN 2
FIFO LEVEL 1
FIFO CONTROL
18/36
ZBT/NOBL SRAM
DIGITAL DATA ENCODER
ANALOG DATA ENCODER
SYNCHRO AND CONTROL LOGIC
8
2x8
HIGH-SPEED DIGITAL I/O INTERFACE
HIGH-SPEED ANALOG INTERFACE (ADC)
HARDWARE CONTROLLED INTERFACES

Fig. 1

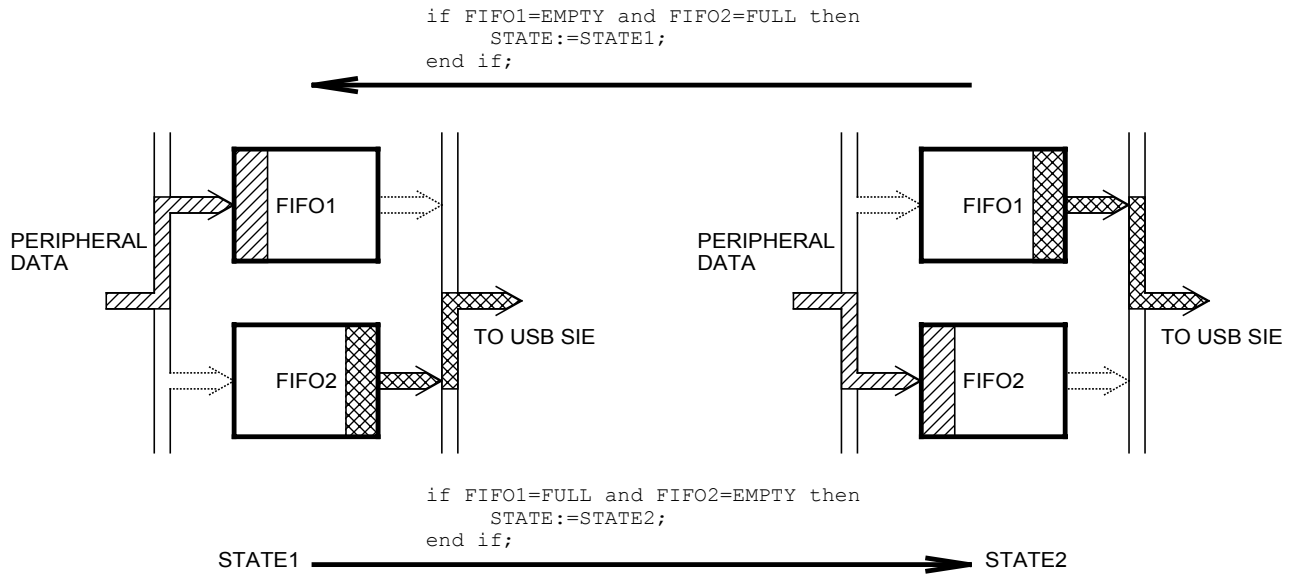

Fig. 2

Time measurement is implemented by the instruction that reads the internal time stamp counter of the Pentium processor – RDTSC. The received data is not processed. The data verification is implemented outside the measured time intervals.

No other USB devices except the universal mixed signal system are connected to the USB.

5. PERIPHERAL INTERFACE SYNCHRONIZATION

In a system that contains peripheral device and USB controller, usually the data/clock domains are different for the controller and the peripheral device. The hardware of the peripheral interface must provide data transition from the peripheral data/clock domain to the USB data/clock domain.

5.1 Peripheral interface clock source

- USB controller is the source of the interface clock

The peripheral interface is synchronized with USB related clocks. If the whole peripheral device is synchronized with the interface clock, only one clock/data domain exists. This situation is difficult to implement without loss of optimality.

The tests are run with interface clock from the USB controller, but the analog interface has a different clock. The timing requirements of the peripheral interface are related to the USB controller clock, therefore this option provides higher data transfer rates.

- The peripheral device is the source of the interface clock

In this case the data transition from peripheral device clock domain to USB controller clock domain is implemented by level 2 FIFO.

5.2 Asynchronous peripheral interface

In asynchronous mode the interface clock is not used externally, but the data is synchronized in slave FIFO by USB related clock (pseudo asynchronous mode).

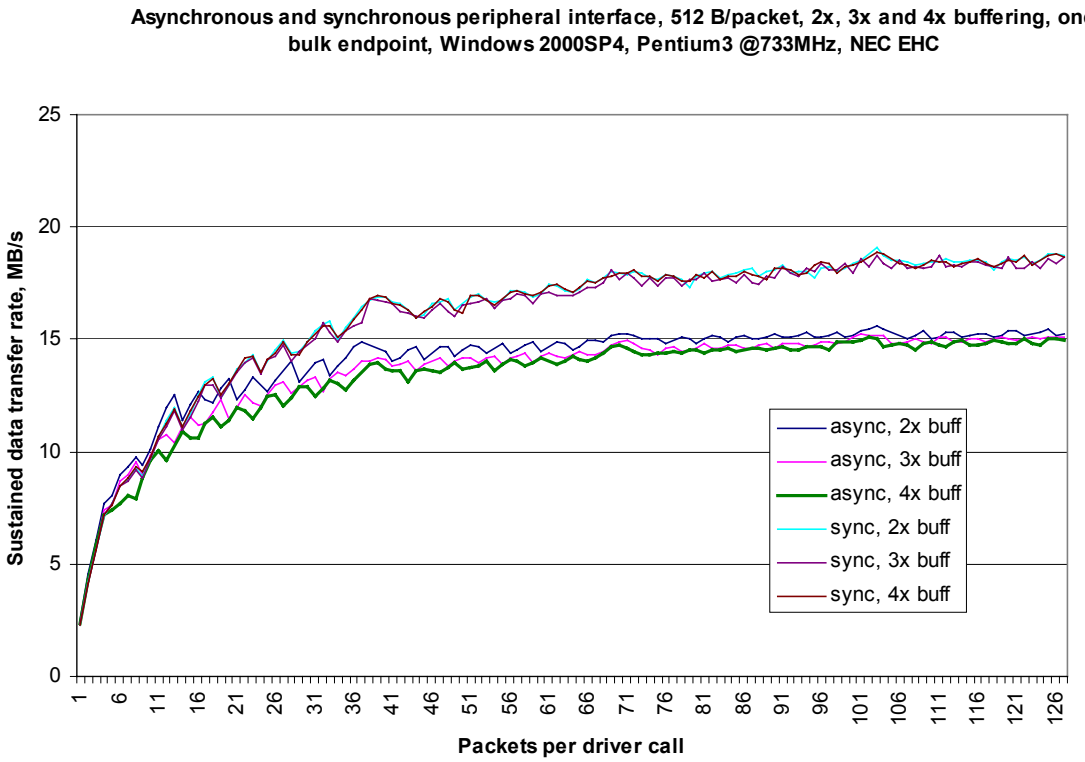
Asynchronous and synchronous peripheral interface, 512 B/packet, 2x, 3x and 4x buffering, one bulk endpoint, Windows 2000SP4, Pentium3 @733MHz, NEC EHC
Sustained data transfer rate, MB/s
Packets per driver call
async, 2x buff
async, 3x buff
async, 4x buff
sync, 2x buff
sync, 3x buff
sync, 4x buff

Fig. 3

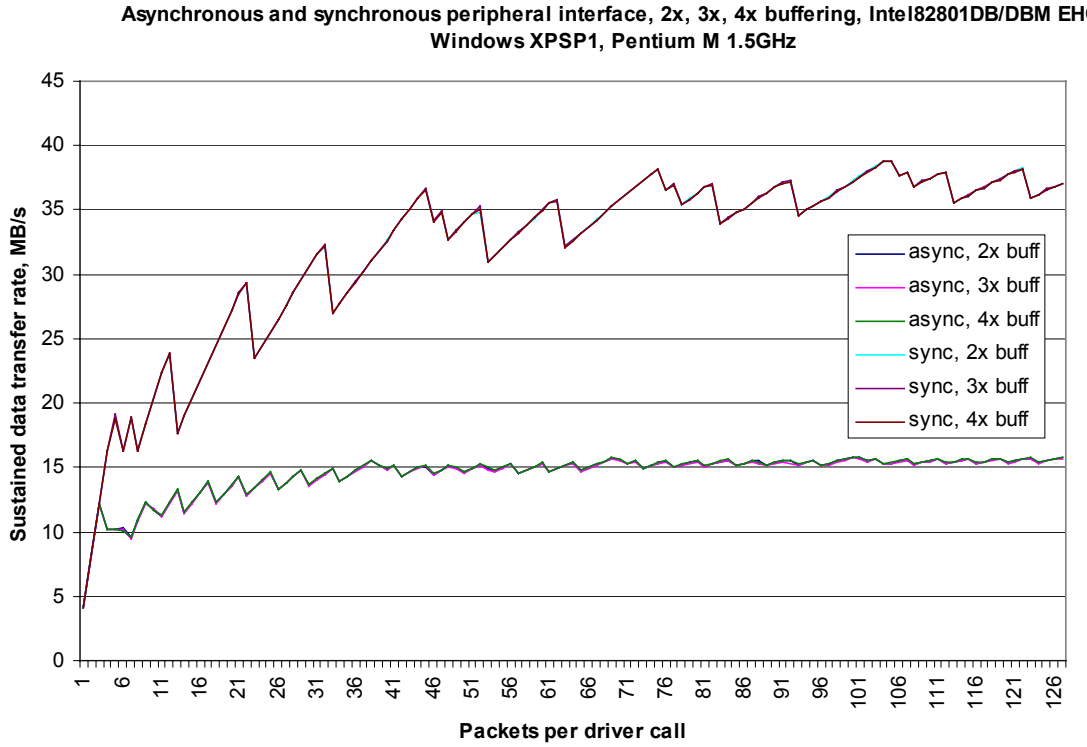
Asynchronous and synchronous peripheral interface, 2x, 3x, 4x buffering, Intel82801DB/DBM EHC, Windows XPSP1, Pentium M 1.5GHz
Sustained data transfer rate, MB/s
Packets per driver call
async, 2x buff
async, 3x buff
async, 4x buff
sync, 2x buff
sync, 3x buff
sync, 4x buff

Fig. 4

The maximum value of data rates from the tests is near the theoretical maximum – 16.6 MB/s (Fig. 3). The write strobe SLWR_N, generated for the tests from FPGA exactly meets the minimum requirements in [4].

5.3 Synchronous peripheral interface

In this case the read/write strobes are clock enables for FX2 FIFO.

Theoretical maximal value of data rate of the peripheral interface is 96MB/s at 48MHz internal clock, therefore the peripheral interface is not a bottleneck (as asynchronous interface).

The test results are 35% and 74% of the USB 2.0 data transfer rate maximum 53 MB/s (Fig. 4).

The bottleneck is in the PC, it limits at 19MB/s (Windows 2000/PentiumIII-733MHz, NEC EHCI) and at 39MB/s (Windows XP SP1/Pentium4-1.5GHz, Intel EHCI).

Using isochronous transfers produces similar results, but streaming is not suitable for transfer of output data of compression methods without error correction.

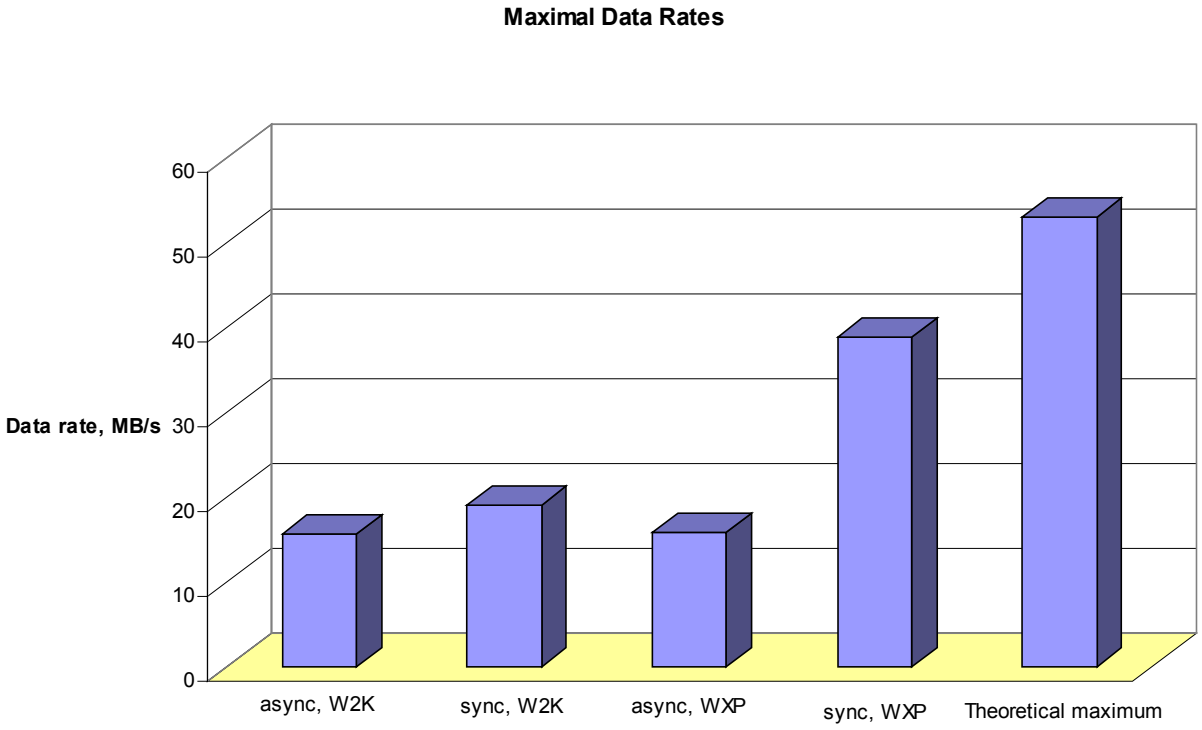

Fig. 5

Low data transfer rates when a small number of packets (1 - 5) in one DeviceIOControl indicate that calls from User Mode to Kernel Mode slow down the communication.

The rate deviation is significant for high data transfer rates (Fig. 4). Small differences between data transfer rates for double, triple and quad buffering show that double buffering does not lead to NAK of the IN tokens. Fig. 5 compares the maximal transfer rates of synchronous/asynchronous interfaces on the two OS.

6. CONCLUTION

The results for sustained data transfer rates about 75% of the maximum for bulk transfers on USB 2.0 shows that it can be used in most of the medium to high speed mixed signal systems.

Appropriate methods for data buffering and synchronization should be used to remove data path bottlenecks. Data transfer/processing by software in high transfer rate systems have to be limited or avoided. Driver call number must be as low as possible.

The significant differences between data transfer rates of analog/digital interfaces and USB makes important the use of high speed data compression methods.

REFERENCES

[1] Williams S. R., *Test and measurement equipment adopts computer industry standards*, Autotestcon Proceedings 2002 IEEE, ISSN 1080-7725, October 2002, pp 367-378.

[2] Mueller J. E., *Achieving Instrument Interchangeability with IVI Instrument Drivers*, Autotestcon 2003 IEEE System Readiness Technology Conference. Proceedings , ISSN 1080-7725, September 2003.

[3] Compton K., S. Hauck, *Reconfigurable Computing: A Survey of Systems and Software*, ACM Computing Surveys, Vol. 34, No. 2, June 2002, pp. 171–210.

[4] Cypress Semiconductor Corp., *CY7C68013A/CY7C68015A, EZ-USB FX2LP™ USB Microcontroller, High-Speed USB Peripheral Controller*, Cypress Semiconductor Corp., Doc. #: 38-08032 Rev. E, June 8, 2004.

[5] Cypress Semiconductor Corp., *EZ-USB FX2 Technical Reference Manual v2.2*, Cypress Semiconductor Corp., 2003.

[6] Xilinx Inc., *Spartan-II 2.5V FPGA Family: Complete Data Sheet Product Specification*, Xilinx Inc., DS001 September 3, 2003.