The Development of a Soft IP-core of a Multi-channel High Speed Serial Port with DMA Controller

Angelo Brito, Camila Nunes, Maria Silveira, Marcelo Albuquerque, Maryane Brasilino, Rafael Barbosa, Rafael Lima, Raphael Lima, Vanessa Carvalho and Edna Barros
Centro de Informática – CIn - UFPE
Av. Jornalista Anibal Fernandes, s/n - Cidade Universitária (Campus Recife)
50.740-560 - Recife – PE
(asb, can, mcrs, mra, mcbb, rcbil2, rdtl, rlna, vlac, ensb@cin.ufpe.br)

ABSTRACT
This paper presents the design of a McBSP IP-core with DMA capability based on a well-defined methodology that allows high-quality design of IP-cores. The Multi-channel Buffered Serial Port (McBSP) and DMA are very important devices for high speed signal processing. The IP-Cores have been specified in SystemVerilog and validated using a SystemVerilog based testbench. Both IP-cores have been prototyped in FPGA and integrated into an Altera® platform including a Nios II processor and AVALON bus.

General Terms
IP-core design

Keywords
Serial port, signal processing, DMA, IP-core design, SystemVerilog

1. INTRODUCTION
Due to the market’s pressure of consumer electronics, the industry has increased the use of IP-Cores in digital systems design supporting signal processing. On the other hand, the complexity such IP-Cores has been also increased significantly, because they are often needed to implement more functionalities and different operation modes. Due to this complexity, starting an IP-Core design from an RTL specification increases the codification and verification time and, consequently, increases the time to market. A solution to reduce the IP-Core development time is to construct the testbench concurrently with the RTL description.

This work shows the design of a high-speed serial port (McBSP)[1] and a direct memory access controller (DMA)[2], which have been prototyped in FPGA. High-speed serial ports with DMA have become popular due to their importance for signal processing, and they are present in the great majority of embedded systems.

This paper is organized as follows: Section 2 explains the McBSP and DMA IP-Core structure. Section 3 shows the IP-Core development; Section 4 shows how the FPGA prototyping of both IP-cores and some results. Section 6 concludes the paper.

2. THE MCBSP AND DMA OVERVIEW
The multichannel buffered serial port (McBSP) is an input / output device that allows buffered serial transmission and reception for applications requiring high-speed signal processing and transmission of audio and video. Among the McBSP features stand out serial transmission and reception with external devices. It also supports multi-channel transmission and multiple clock frequencies.

The processor communicates with the McBSP using 32-bit internal registers and control data, as shown in Figure 1.

![Figure 1 – McBSP Architecture](image)

The multichannel buffered serial port (McBSP) specification was based on the serial port of TMS320C6000 DSP family [1]. The McBSP provides the following functionalities: full-duplex communication, double-buffered data registers, which allow a continuous data stream. Additionally the McBSP supports independent framing and clocking for receive and transmit with an external shift clock or an internal, programmable frequency shift clock for data transfer. The McBSP has also capability of multichannel transmit and receive of up to 128 channels. It supports a wide selection of data sizes, including 8, 12, 16, 20, 24, and 32 bits, allowing 8-bit data transfers with the option of LSB or MSB first.
Both µ-Law and A-Law companding features are provided by the McBSP. The port also supports programmable polarity for both frame synchronization and data clocks.

The DMA controller transfers data between regions in the memory map without intervention by the CPU. The DMA controller allows movement of data to and from internal memory, internal peripherals, or external devices to occur in the background of CPU operation, allowing the CPU to remain active during data transfers. The DMA controller has a programmable numbers of elements per frame and of frames per block. In completing a frame transfer, the DMA controller moves all elements in a single frame and in completing a block transfer, the DMA controller moves all frames that it has been programmed to move.

The DMA controller has the following set of registers that must be configured prior to beginning a data transfer: Primary control register (PRICTL), used to configure the transfer; Secondary control register (SECCTL), used to enable interrupts to the CPU, and to monitor the channel activity; Transfer counter register (XFRCNT), used to keep track of the transferred elements; Source address register (SRC), which specifies memory location from which the element is transferred and Destination address register (DST), which specifies memory location to which the element is transferred.

As shown in figure 2, the DMA controller has two Avalon Bus interfaces, in which one is a Slave Interface and the other is a Master Interface. The first is to configure and observe the behavior of the DMA while the second serves to make the transfers between devices.

When the McBSP is operating with the DMA it takes advantage of its split-channel operation. Using this feature, a single DMA channel is used to perform both reads from, and writes to, the McBSP.

3. THE MCBSP-DMA DESIGN FLOW

The McBSP and DMA IP-Cores design have been done according to the ipPROCESS methodology, which has been already used in other IP-cores development [3], [4]. In short, the ipPROCESS methodology (or IPP methodology) is a rigorous and thorough engineering process that guides designers through the design process, so that they can acquire a clear and unique understanding of the IP-core functionality and behavior. It has been inspired on the combination of well-known software engineering methodologies, like RUP and XP, with IC design standards like VSIA and RMM. Figure 3 summarizes the design phases and activities, which are defined in the IP-process methodology.

In the McBSP and DMA designs, each IP-core has been designed as smaller modules, due to its complexity. The RTL description In SystemVerilog of each sub-module has been verified separately before being integrated to the top module. Most of the communication among the modules has been done at pin-accurate level, using FIFOs for the data channel implementation.

The McBSP and DMA design included four phases before the prototyping. These phases were: specification, testbench development, RTL specification, logical synthesis and FPGA prototyping.

The verification performed after each of these phases has covered the correctness of transmit and receive operations considering the various scenarios of a McBSP operating
scenarios, as for example multi-channel, with different clocks rates, in the general purpose mode and in the companding operating mode.

To accomplish this verification, distinct testbenches have been developed and integrated. Figure 2 shows the testbench structure for the McBSP verification and how the testbenches have been integrated.

The validation of McBSP and DMA modules was done following a functional verification approach, which is based on the VERISC verification methodology [5]. This methodology includes the testbench architecture and an strategy for developing testbenches, which allow to starts the testbench development in an early design phase and to validate the complete testbench before using it for DUV functional verification.

The functional validation basically consists of sending stimulus from “Source” to the reference model and the RTL model. The component called "Checker" compares the results obtained from the reference model and RTL model to check the equivalence between models.

For each module was developed a Design Under Verification (DUV), a Reference Model which implements the DUV functionalities in an ideal way and the testbench for the functional verification. The testbench is responsible for generating stimulus to the DUV and compare the responses with the responses of the Reference Model. The Reference Model, testbench and the DUV have been specified using the SystemVerilog hardware description language.

The Reference Model models the functionality, but not the interface. Thus the testbench also includes drivers and monitors modules. The driver receives incoming transactions from source and converts them into signals of the DUV input interface and the Monitor observes signals from the DUV output interface and generates output transactions and passes them to the checker. These modules can be seen in Figure 4.

The main aim of the tests case set was to validate both transmitting and receiving operations, using multi-channel configuration and different clock rates.

The functional verification has been done using VCS verification library. The logical synthesis has been done using the Synplify tool. The Quartus tool, from Altera®, has been used for FPGA prototyping.

4. FPGA PROTOTYPING AND RESULTS

To validate the functionalities of the proposed McBSP and DMA IP-cores, a FPGA implementation of both IPs has been done. For this purpose a platform including a memory, a Nios II processor and an Avalon bus, has been developed and both the McBSP and the DMA have been integrated into the Platform. This platform has been prototyped in a Cyclone II EP2C35F672C6 FPGA.

Table 1 shows the main results of McBSP and DMA designs with prototyping in FPGA. Both IP-Cores working at a clock rate of 50 MHz and are able to read a wave file of 30.6 MB in 6 minutes. Figures 5 and 6 show the screenshots with captured waveforms during transmitting and receiving of a wave file, respectively.

<table>
<thead>
<tr>
<th>Results</th>
<th>McBSP</th>
<th>DMA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architecture</td>
<td>12 modules</td>
<td>10 modules</td>
</tr>
<tr>
<td>RTL Code</td>
<td>9277 code lines</td>
<td>12377 code lines</td>
</tr>
<tr>
<td>Testbench</td>
<td>73181 code lines</td>
<td>46049 code lines</td>
</tr>
<tr>
<td>CLB’s</td>
<td>4.630</td>
<td>3.865</td>
</tr>
<tr>
<td>Gate count</td>
<td>158,902</td>
<td>132,647</td>
</tr>
<tr>
<td>FPGA area</td>
<td>14%</td>
<td>12%</td>
</tr>
</tbody>
</table>

Figures 5 and 6 show the behavior of the McBSP CLKR, FSR e DR signals when operating transmit and receive. Once transmit frame synchronization occurs, the value in the transmit shift register (XSR) is shifted out and driven on the DX pin after the appropriate data delay as set by XDATDLY. XRDY is activated after every DXR-to-XSR copy on the following falling edge of CLKK, indicating that the data transmit register (DXR) can be written with the next data to be transmitted. XRDY is deactivated when the DXR is written by the CPU or the DMA controller.

Figure 5 illustrates serial transmission. Once the receive frame synchronization signal (FSR) transitions to its active state, it is detected on the first falling edge of the receiver’s CLKR.
The contents of RSR register is copied to RBR at the end of every element on the rising edge of the clock, provided RBR is not full with the previous data. Then, an RBR-to-DRR copy activates the RRDY status bit to 1 on the following falling edge of CLKR. This indicates that the receive data register (DRR) is ready with the data to be read by the CPU or the DMA controller.

To validate the McBSP and DMA functionalities in FPGA and emphasize the DMA importance on chip as a whole, the McBSP was integrated into a FPGA considering two scenarios: one with DMA and another one without the DMA module. The experiments have taken into account the system efficiency and the CPU use (CPU interruptions mainly). The simulation showed that the number of CPU interrupts was, approximately, 164 interrupts per second, making it very difficult to run the transmission in parallel with some tasks.

A classical implementation of the recursive calculation of the factorial of a number could not be executed in parallel with the transmission of a wave file. Even simple tasks like writing on the seven-segment display ran extremely inefficient due to the high number of interruptions.

**5. CONCLUSIONS**

This paper presented the implementation of an IP-core of a high speed multi-channel serial port with DMA capabilities. This IP-core is very important for signal processing digital applications since it allows the transmission and receiving of multi-signals using buffers. Additionally the IP-core allows encoding the data to be transmitted for increasing the bandwidth. The availability of an IP-core for serial transmission with DMA capabilities is extremely important for the implementation of systems in the area of signal processing as it allows transmission and reception of signals is made without the participation of the processor making the system more efficient.

The proposed design of the McBSP serial port with DMA has been validate through exhaustive simulation. For this purpose a complete testbench has been developed, which has been used to validate all sub modules of the McBSP serial port and DMA separately and also the top modules. The IP-cores have also been prototyped in FPGA and a platform based on the NIOS processor has been developed for validating both the McBSP and DMA IP-cores.

The designed IP-cores can be integrated into any platform. The current version has capability for interfacing with AVALON bus. The connection with others bus protocols can easily do by design of wrappers.

Integrating the proposed IP-cores into a processor based platform enhances the processor capability of signal processing.

Future works include the validation of both IP-cores as ASIC. The logic synthesis, test design and layout synthesis are under development.

**5. REFERENCES**


