Internal dynamic partial reconfiguration for real time signal processing on FPGA

Sheetal U. Bhandari¹, Shaila Subbaraman², Shashank Pujari³ and Rashmi Mahajan⁴

¹, ³, ⁴Dept. of Microelectronics and VLSI Design, International Institute of Information Technology, Pune-411057 India
²Department of Electronics Engineering, Walchand College of Engg., Sangli 416415 India
¹sheetalubhandari@gmail.com, ²shailasubbaraman@yahoo.co.in, ³pujarishashank@gmail.com, ⁴mahajanrashmi@gmail.com

Abstract

Few FPGAs support creation of partially reconfigurable systems when compared to traditional systems based on total reconfiguration. This allows dynamic change of the functionalities hosted on the device when needed and while the rest of the system continues its working. Runtime partial reconfiguration of FPGA is an attractive feature which offers countless benefits across multiple industries. Xilinx has supported partial reconfiguration for many generations of devices. This can be taken advantage of substituting inactive parts of hardware systems and to adapt the complete chip a different requirement of an application. This paper describes an innovative implementation for real time audio and video processing using run time internal partial reconfiguration. System is implemented on Virtex-4 FPGA. Internal reconfiguration is handled using internal configuration access port (ICAP) driven by soft processor core. The considerable savings in device resources, bit stream size and configuration time is observed and tabulated in this paper.

Keywords: Reconfigurable computing; partial reconfiguration; run time reconfiguration; internal reconfiguration.

Introduction

Nowadays, pervasive systems take more and more space in our life as an embedded systems which automatically adapts to change in their environment and act on the base of user needs. Security aspects, seamless communication and self configuration are the key challenges raised by pervasive systems. Flexibility in hardware platform is important to achieve high performance. Run time partial reconfiguration (PR) is an exact candidate for this. Run time PR is a recent method in reconfigurable computing to update selectively the circuitry of a programmable board, while still being active (Fig.1). This allows changing a group of logic very quickly when the application needs it. As it is not possible to plan at design time the evolution of the functions used by pervasive systems (for e.g cryptography, signal processing & communication protocols), the system must be able to update itself in order to adapt to its environment (Lagger, 2006).

The pervasive systems in consumer electronic domain provide day to day larger amount of functionalities like video-audio processing, communications, entertainment etc. At the same time these functionalities demand more complex support: operating systems, secured communication etc. Guaranteeing high performance is not possible when the processing is fully performed by software. A common approach to improve performance is to include specialize hardwired coprocessor. However, given their static architecture, these systems lack flexibility and having specialized coprocessor for each task is not feasible, due to the amount of logic required and the possible incompatibility of upgraded version of the algorithms. The system must be able to update its own hardware platform in order to adapt to an environment in autonomous way. Dynamic reconfiguration can deal with this problem by modifying their function without altering the rest of the system with minimum resources.

In this paper we present innovative real time audio, video signal processing using Internal PR feature of FPGA. This approach shows considerable savings in area and reconfiguration time too. In the following section, the concept of internal run time partial reconfiguration is discussed; followed by briefing of processing real time audio and video signal. Hardware platform, processor and operating system are also described before presenting results and discussions.

Run time internal partial reconfiguration

With rising gate densities and increased power of FPGA, co-existence of processor and digital logic components is possible on single device. This provides flexibility of combining software and hardware based control in one chip. One more important feature is being added to FPGA to further enhance the performance with minimum resources is partial reconfiguration. This feature is provided in few FPGAs from Xilinx like Virtex II, Virtex IV etc (Koo, 2005). On the fly PR provides a way to modify the implemented logic in FPGA when the device is on. More clearly PR allows reconfiguring selected areas of a FPGA when other part of FPGA is still
Real time signal processing

Sheetal et al.

Fig. 2. Setup

Fig. 3. System block

For video processing spatial filtering techniques are used for noise reduction by mean and median filters. The used architecture is based on an impulse response array of 5 x 5 mask. The idea of mean filtering is simply to replace each pixel value in an image with the mean value of its neighbors, including itself. This has the effect of eliminating pixel values, which are unrepresentative of their surroundings. Image averaging is a digital image processing technique (Oppenheim & Schafer, 2000; Chanda et al., 2003) that is often employed to enhance video images that have been corrupted by random noise. The algorithm operates by computing an average or arithmetic mean of the intensity values for each pixel position in a set of captured images from the same scene or view field. The methodology of diffusing the intensity is based on following four modules. (1) Real time calculation of Intensity of each pixel. (2) Storage of 5 lines of RGB pixels. (3) Calculation of average intensity of center pixel w.r.t. surrounding 5 x 5 pixels’ intensity. (4) Regeneration of R, G, B of center pixel as function of this new average intensity. In median filtering the input pixel is replaced by the median of the pixels contained in the neighborhood.

The algorithm for median filtering requires sorting the pixel gray values in the neighborhood in increasing or decreasing order and picking up the median of the array (Oppenheim & Schafer, 2000; Chanda et al., 2003). The methodology of median filtering is based on following four modules. (1) Real time calculation of Intensity of each pixel. (2) Storage of 5 lines of RGB pixels. (3) Finding out the median intensity of center pixel w.r.t. Surrounding 5 x 5 pixels’ intensity. (4) Regeneration of R, G, B of center pixel as function of this new average intensity. As any one filter would require at any time, instead of implementing two filters, third step of above is implemented as dynamic module of PR design while hardware realization of step 1, 2 and 4 makes up static implementation. For stereo audio processing, 4 filters per channel are designed for different cut off frequencies to provide user with selective band hearing facility. Instead of having all the filters present at a time on device one filter per channel is loaded using PR and can be changed by user on the fly. Selection of filters was kept simple as the focus of work was to prove the usability of PR concept. Implementation of advanced filters using PR are expected to show same benefits as reconfigurable module is treated as black box as far as PR implementation is concerned.

System description

The implementation is done on ML-402 board consisting of Virtex-4 with support from VDEC1 card from Digilent. Video source is from NTSC/PAL compatible camera and the output display is 640 X 480 resolution VGA PC monitor. Audio source is from MP3 player and hardware.
the output device is headphone or speaker. The laboratory set up used for experimentation is shown in Fig. 2. Microblaze system is designed using Xilinx platform studio hardware implementation of audio, video is carried out in Xilinx ISE web pack and partial reconfiguration flow is carried out using early access plan-ahead tool (Jackson, 2007). Fig. 3 shows the block diagram of complete system. It consists of microblaze processor, hardware implementation of audio, video signals processing with partially reconfigurable filters.

There are three reconfigurable modules, two audio filters for two audio channels and one video filter. The processor can reconfigure the audio and video filters as per user’s choice by loading respective partial processor can reconfigure the audio and video filters as for two audio channels and one video filter. The processor can reconfigure the audio and video filters as per user’s choice by loading respective partial configuration bitstream through the ICAP port. The standalone operating system is running on microblaze allows managing the system, peripherals and PR.

<table>
<thead>
<tr>
<th>Implementation</th>
<th>Type of bit file</th>
<th>Size of bit file</th>
<th>Reconfig. time (Cable freq: 6MHz) using JTAG</th>
<th>Reconfig. time using ICAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Without PR_ ALL Filters</td>
<td>Static bit file</td>
<td>1673 KB</td>
<td>6 sec</td>
<td>-</td>
</tr>
<tr>
<td>Partial mean filter</td>
<td>190KB</td>
<td>1 sec</td>
<td>360mSec</td>
<td></td>
</tr>
<tr>
<td>Partial median filter</td>
<td>244KB</td>
<td>1sec</td>
<td>460mSec</td>
<td></td>
</tr>
<tr>
<td>Video blank</td>
<td>146KB</td>
<td>1sec</td>
<td>280 mSec</td>
<td></td>
</tr>
<tr>
<td>Partial bit file left_LPF</td>
<td>146 KB</td>
<td>1 sec</td>
<td>280 mSec</td>
<td></td>
</tr>
<tr>
<td>Partial bit file left_BPF</td>
<td>135 KB</td>
<td>1 sec</td>
<td>250 mSec</td>
<td></td>
</tr>
<tr>
<td>Partial bit file left_BSF</td>
<td>130 KB</td>
<td>1 sec</td>
<td>240 mSec</td>
<td></td>
</tr>
<tr>
<td>Partial bit file right_LPF</td>
<td>135 KB</td>
<td>1 sec</td>
<td>250 mSec</td>
<td></td>
</tr>
<tr>
<td>Partial bit file right_HPF</td>
<td>152 KB</td>
<td>1 sec</td>
<td>290 mSec</td>
<td></td>
</tr>
<tr>
<td>Partial bit file right_BPF</td>
<td>138 KB</td>
<td>1 sec</td>
<td>260 mSec</td>
<td></td>
</tr>
<tr>
<td>Partial bit file right_BSF</td>
<td>150 KB</td>
<td>1 sec</td>
<td>285 mSec</td>
<td></td>
</tr>
<tr>
<td>Left blank</td>
<td>58 KB</td>
<td>1 sec</td>
<td>108 mSec</td>
<td></td>
</tr>
<tr>
<td>Right_blank</td>
<td>75 KB</td>
<td>1 sec</td>
<td>140 mSec</td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Resource utilization for implementations with and without PR & percentage saving in PR

<table>
<thead>
<tr>
<th>Logic utilization</th>
<th>Used without PR</th>
<th>Used With PR</th>
<th>% saving</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of slice flip flops</td>
<td>6086</td>
<td>3120</td>
<td>48.73</td>
</tr>
<tr>
<td>No. of 4 input LUTs</td>
<td>9091</td>
<td>4166</td>
<td>54.17</td>
</tr>
</tbody>
</table>

Table 2. Bit file sizes & downloading time for implementations with & without PR

Table 3. Power dissipation for implementations with and without PR and percentage saving in PR

Hardware board
ML 402 (UG083, 2006) board has Virtex-4sx35ff668 (UG070, 2005) device which supports partial reconfiguration and has two ICAP cores on it. It has compact flash which is being used to store system ace and partial bit files. The UART port is used to provide user interface to select appropriate filter by user.

Processor
The choice of implementation for processor on FPGA is hard core or soft core processor. Soft-core processor provides benefits of easy implementation and upgradeability while hard core processors are best optimized hence provides higher Performance. As no hardcore processor is available on the device Soft-core processor Microblaze is used from several available choices of processor cores like OpenRISC, LEON etc.

Audio video processing
The filters for audio and video signals are implemented on hardware. Video decoder ADC ADV7183B is available on FPGA and filtering is done as discussed in section III of this paper. Need of mean and median filter is mutually exclusive. Hence, PR is used to change averaging block by median calculating block to change the type of filter. Processed RGB signal is fed to Video DAC ADV7123 available on board; which converts it to VGA compatible signal. For audio signal LM 4550 audio codec is available on board. To control and configure it AC 97 core is generated and implemented in FPGA. Audio filters are also implemented using PR so as to have one filter per channel available on FPGA.

Self reconfiguration & user Interface
Self reconfiguration refers to handling on the fly PR by FPGA itself. Internal configuration access port (ICAP) is used for this (DS280, 2006). Partial bit files are stored in compact flash available on the board along with system ACE file. Microblaze drives the ICAP for changing the filters on the fly (Blodget, 2003). Decision of changing the filter can be taken by processor based on software running on it. For demonstration purpose we use UART communication to show the menu and allow user to enter the choice of filter for audio and video signal processing. Though ICAP is used for reconfiguration FPGA can be partially or fully reconfigured using appropriate bit file using JTAG mode. This allows user to update hardware internally as well as externally if required (Xapp138, 2000).

Operating system
Selection of operating system is critical in any embedded system design. We have used standalone operating system provided by Xilinx as it is easy to use and good amount of documentation is available.

Experimental set up and result
- To validate and demonstrate
the benefits of partial reconfiguration for early adaptation and resource utilization for computation intensive real time application in pervasive audio video filtering application is chosen as a bench mark circuit.

- To find out the benefit in terms of resource utilization; PR implementation, where only one filter for processing respective signals at a time is available in a system is compared with a system in which all filters are available all the time on the FPGA.

- To validate the early adaptation we have measured the reconfiguration time: time to partially reconfigure the FPGA through the ICAP. This time is compared with the time of total reconfiguration to adapt new filter. To observe the benefits of internal PR, reconfiguration time is also compared with our earlier attempt of external PR; in which bit files were loaded with JTAG using Impact tool by Xilinx (Bhandari et al., 2009).

We also measured the third important parameter of any design i.e. power. Since the board hosting FPGA does not measure power consumption, the power dissipation of the entire board is measured using the method of Paulsson et al. (2008). The clock frequency of ML 402 board is 100Mhz. Cable frequency for JTAG downloading is 6 MHz.

### Resource utilization

Microblaze and other blocks of static module have a maximum area of 1587 slices. This leaves large space for hardware accelerator. The slices required for video filters are 206 and 668 for mean and median filters respectively. For audio filters required slices are 334 per filter per channel. The resource utilization for with and without PR implementation is presented in Table 1.

### Reconfiguration time

In Table 2, sizes and downloading time of all static and partial bit files are loaded. After running through the partial reconfiguration, flow and assembling the PR project static and partial bit files get generated. As only module needs to alter compared to the entire FPGA, the sizes of these partial bit files are smaller than static bit files. To download the bitstream we have used JTAG and ICAP ports. In the implementation where PR feature is not used we need to reconfigure entire FPGA which is done through JTAG. In the implementation where PR is used but not the ICAP bitstream downloading is done through ICAP. In the implementation where PR is done internally, ICAP is used. As in Table 2, the speed ups in adaptability are promising with the use of partial reconfiguration and ICAP port to handle it internally. This data allow determining that use of ICAP is justified. In Spartan 3 device, ICAP is not available. Recently, special core for such devices have been developed (Bayar & Yurdakul, 2008).

Power dissipation and % saving is shown in Table 3.

### Conclusion

We have developed on the fly internal partial reconfiguration system for real time audio and video signal processing. The system consists of 3 independent reconfigurable modules with 2 and 4 choices for video and both audio filters. To handle reconfiguration internally, ICAP is used which is driven by software running on microblaze processor. Though audio and video filters are taken as case study, the observed benefits are independent of the application. Hence, the feature permits a virtually infinite number of coprocessor’s configuration allowing wide performance for a wide amount of applications. Good amount saving in resource utilization is observed by using PR feature. For the applications where resources are not enough on the device can be fit by using PR. Use of PR feature also makes system easy and quick to adapt to the change in environment or as per need. Hence it is promising solution to the challenge raised by pervasive systems about easy configurability. Along with performance and utilization this feature proves itself beneficial for third critical dimension of design; i.e. power saving in PR system is also attractive benefit. Use of ICAP makes the system more autonomous in terms of changing the logic on the fly. It also reduces the reconfiguration time.

### References