

LSCIC Pre coder for Image and Video Compression

¹Muhammad Kamran, ²Shi Feng and ²Wang YiZhuo

¹*Department of Electrical Engineering, University of Engineering and Technology,
Lahore-54890, Pakistan*

²*Department of Computer Science and Engineering, Beijing Institute of Technology,
Beijing-100081, China*

1. Introduction

Image and video compression schemes are implemented for the optimum reconstruction of image with respect to speed and quality. LSCIC (Layered Scalable Concurrent Image Compression) pre coder is introduced here to utilize best available resources to obtain reasonable good image or video even at low band width of the system. This pre coder will make the layers of input data whether video or image and after synchronization send it to the output of pre coder on two different layers at the same time. Prior to understand image compression issue it is more important to become familiar with different image standard formats under usage for certain application. Mainly they include JPEG, GIF, and TIFF etc. Image compression scenario is the main entity to be included in the dissertation as per our project requirement. A new idea for scalable concurrent image compression is introduced which gives superior image reconstruction performance as compare to existing techniques. The verification can be done by calculating gray level and PSNR of reconstructed image. The bit stream is required to be compressed for image data transfer if the main system requirement is the memory saving and fast transformation with little sacrifice in the quality of image for lossy compression scheme. A valuable study is accomplished by K Shen, 1997 for parallel implementation of image and video compression. It is suggested that an ideal algorithm should have a low compressed data rate, high visual quality of the decoded image/video and low computational complexity. In hardware approaches special parallel architectures can be design to accelerate computation suggested by R. J. Gove(1994) and Shinji Komori (1988) et al. Parallel video compression algorithms can be implemented using either hardware or software approaches as proved by V. Bhaskaran (1995). These techniques provided the guidelines to deal with digital image compression schemes fro speed and complexity point of view. For video compression, motion estimation fenomenan has its own importance and different techniques are already presented to have motion estimation to get good quality image. Decoding is considered as first step of compression followed by encoding at receiving end of image and reconstruction side. Intermediate step in data/image and video compression is the transform. Different transform techniques have been used depending upon application.

2. LSCIC Architecture

In order to describe complete working of LSCIC image/video compression pre coder, different steps are defined starting with the elaboration of LSCIC architecture. Fig .1 is architecture initially considered followed by Fig.2 which an optimal modified design.

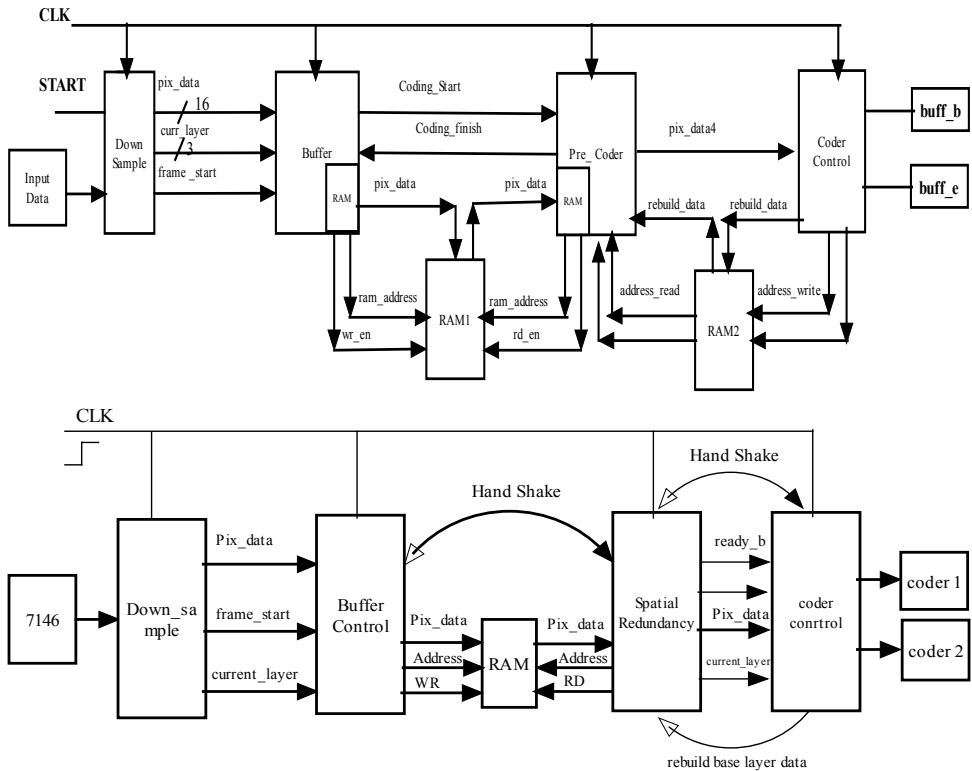


Fig. 1. and Fig. 2. Initially proposed and modified pre coder design

Initially proposed design is quite complicated which includes 16 frames RAM with lot of handshaking signals. It was investigated later on that the design can be simplified by proposing a PING PONG RAM and reducing handshaking signals.

Fig. 2. represents LSCIC pre coder architecture. This pre coder is comprised of 5 modules which are integrated after complete verification of design with respect to their operation.

3. LSCIC Phase-I

LSCIC architecture is divided into two sub phases for the design and testing convenience and also to become acquainted with hurdles encountered during algorithmic design and architecture implementation.

LSCIC phase-I addresses a problem of large data to be processed through RAM in proposed design. As image data is large in size and randomly extracted from image, the requirement of system is to place and temporarily hold the data in large size RAM prior to its transmission to next module for further processing. RAM with conventional design is not able to complete simulation process in desired time and unwanted delay is introduced. Prior to realize the design it is important to circumvent this problem of large data handling and inclusion of huge hardware components in design.

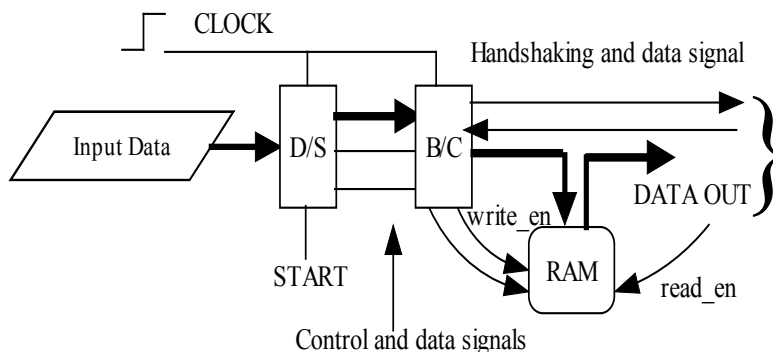


Fig. 3. (Phase-I) Module 1, Module 2 and RAM unit for Data pixel recognition

Figure 3 is the phase-I of LSCIC pre coder describing operation of first 3 units of proposed design with all inevitable control and data signals. It mainly emphasizes the issue to include large RAM unit into design with all constraints with ample solution. Directional bold arrows represent the data path while thin lines indicate the control and hand shaking signals.

3.1 LSCIC Phase-I (Circuit operation) and Mathematical Model

For image compression process, designed circuit will perform different useful tasks. One of them is to get output data concurrently from two independent channels and secondly, circuit may be adaptive to different band widths to capture reasonably good quality image. For MPEG applications, if load on the network is changing causing variations in the system band width may cause video disturbance. The resulting design can handle the situation and provides good compression even when net work is over loaded. After obtaining the solution of large input data for the simulation through external file, next step is to place it for certain operation like down sampling, buffering and proper recognition of pixels.

First module works to "Down Sample" image data to give four image layers B1, E2, E3, E1 initially and fifth layer B2 is extracted afterwards from one of the available enhanced layers E1, E2 or E3. This multilayer scenario, as discussed before is called Multi description scheme as each layer describes its own characteristics and behavior. All layers are of same size except B2 which is $\frac{1}{4}$ of the size of any other pixel layer. These layers are required to be placed in PING PONG RAM to make one frame with a unique starting address.

The design was initially proposed with a RAM placed after down sample and Buffer control module with 16 frames. But after careful investigation, it has been concluded that only two frames are sufficient in address RAM for data handling on the bases of concurrent writing and reading data process, CWCR. This characteristic of CWCR made it to work as PING PONG RAM i.e. concurrent Read and Write operation.

It is suggested that design should be made for complete data processing with minimum possible time. The RAM discussed above is designed for the purpose of data storage with 12 address lines and 4096 unique addresses which gives output with considerable long time delay and sticks to infinite time when synthesis of design is carried out. This problem during behavioral design implementation is well addressed in this chapter and results are obtained by incorporating the co-design methodology which causes simulation to be completed in reasonable short time. According to proposed design which is extendable to large scale, one pixel is comprised of 16 bits and there are 256X128 pixels in one layer. As there are 5 layers in each frame, a large data is to be handled and placed properly in designed RAM prior to coder operation proposed by Kamran and Shi in 2006. The READ operation is kept fast as compare to WRITE in order to keep the stability of circuit high. High stability means, during transmission of data in given unit, minimum data loss is observed and almost all pixels reached the receiving end. Prior to proposing pseudo code of phase-I of LSCIC pre processor design, it is regarded as more important to describe mathematical model to get preliminary information about different signals and sequences of operation.

For the verification of proposed algorithm, a mathematical model is presented to clarify the pixels processing with respect to timing and control signals. The design of LSCIC phase-I is described comprehensively by adding all required signals along with data flow path. As described earlier, given model explains the operations of first three modules with mathematical notations explaining the design operating sequence.

Figure 4 gives mathematical representation of all input and processing signals with components encountered in LSCIC-phase-I architecture. Image is characterized as one dimension column matrix containing pixels, P_1 to P_n . Logic value of "Start" signal decides whether pixels are required to be transmitted or not. Down sample module will divide the image into number of layers with addresses decided by a special control signal "Current Layer". It is 3 bit signal needed to represent addresses of 5 possible image pixel layers formed in module 1(4 initial and one extracted layers after wards). Buffer control just controls the sequence of pixel stream and generates WRITE address in RAM to store pixel information. The objectives of design are described as under in two steps;

- (1) Question was to generate large data automatically, instead of doing manual labor which wastes considerable design simulation time.
- (2) Secondly, the problem of large size component inclusion fails the synthesis operation, which ultimately causes the failure of design.

Explaining the mathematical model of Figure 4, it is mentioned that input video/image data is sent to the down sample module, which divides this data initially into 4 layers. 5th layer b2 is extracted from 'e1' whose size is $\frac{1}{4}$ of the size of e1. Buffer control module just calculates the addresses of layers to be placed into specific locations in RAM. RAM is designed such that READ process is faster as compare to WRITE for more efficient data handling. Despite of all these observations, input signal "START" should be kept high for all operations to be processed.

$\partial \rightarrow$ Re presents Downsample Pr ocess;

$\alpha \rightarrow$ Denotes addresses in RAM for Different layers;

$$\text{Video Input data} = \begin{bmatrix} p_0 \\ p_1 \\ \dots \\ p_n \end{bmatrix} ; \quad \text{Start(Signal)} = \{0,1\}$$

$1 - D / \text{Sample}(\partial(\text{Video_data}); b1, e2, e3, e1 \& b2);$

$b1, e2, e3 \& e1 \in \text{Video Input data} ;$

$\Rightarrow \partial(\text{Video_data}) \rightarrow b1, e2, e3 \& e1 (\text{All same number of pixels})$

$\Rightarrow b2 (\text{Extracted layer}) = \partial(e1)$

$\text{Size of } b2 = (\text{size of any layer}) / 4;$

$2 - \text{Buffer_Control}(\partial(\text{Video}(b1, e2, e3 \& e1), \text{Coding_finish}; (\alpha(\text{RAM_address}), \text{wr_en}, \text{Coding_start}))$

$\alpha 1 \rightarrow$ address for $b1,$

$\alpha 2 \rightarrow$ address for $e2,$

$\alpha 3 \rightarrow$ address for $e3,$

$\alpha 4 \rightarrow$ address for $e1,$

$\alpha 5 \rightarrow$ address for $b2,$

All addresses are generated when coding_finish is high;

Followed by Coding_start = '1';

$3 - \text{RAM}(\text{write / read}, \text{Pixel_data_in}, \alpha; \text{pixel_data_out})$

Number of Frames = 2;

Each frame contains all layers($b1, e2, e3, e1 \& b2$);

Condition :

$\text{Speed of Write operation} < \text{Speed of Re ad}$

Fig. 4. Mathematical Model description of Phase-I

To attain the first objective of LSCIC phase-I that is automatic data transfer for simulation which can be accomplished by creating an external data ".dat" file giving rise to hardware/software co design approach. This idea is quite successful for simulation, but synthesis does not allow such external file additions into design as synthesis tool does not have option to add such files in design by Kamran. After proposing solution of first constraint in design by adding external data file to verify simulation, second point was concentrated to find the way to add large size hardware components like, RAM, ROM, Buffers, Multipliers etc., in design. It is advised for designers, if overall digital system is a big scenario and some hardware component as described above is a part of it, IP core is recommended to be placed. It will cause fast simulation, synthesis and verification of design on behavioral and on circuit level with minimum time. For the purpose of LSCIC-Phase-I verification, IP core RAM is used. The procedure to append RAM into design is given below;

Single port RAM is selected with the maximum capacity of 32768 pixels location for 30,000 gates device under operation. While appending the core into design, designer should have to get the core component and port map information from automatically generated *.vho file. Figure 5 represents block diagram of CORE RAM wrapped in VHDL source file. Component and port map is copied from *.vho and paste them in *.vhd RAM file should present in the project. Lastly in wrapper file we make connections of core signals and wrapper inputs and outputs. This combination of *.vhd and *.vhofile describing components becomes IP CORE which becomes part of design instead of module placed with the help of conventional VHDL code. It is to be noted here that data transfer is successfully achieved during our research by conventional RAM design but it costs more time as compare to IP Core.

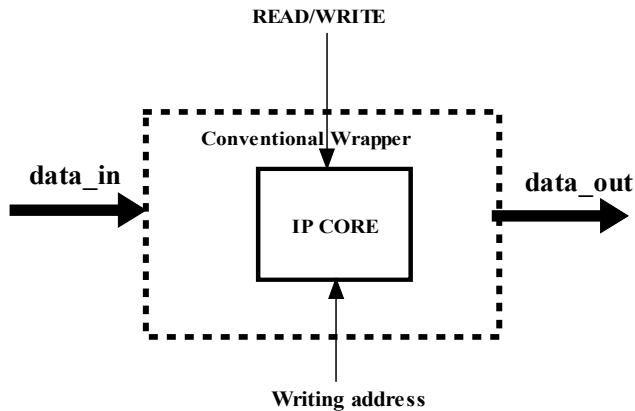


Fig. 5. IP CORE Wrapped in VHDL Source File

Following is the pseudo code description of appending IP RAM into conventional VHDL RAM design;

Code Description

Defining IEEE library

Entity ur_ram is

Entity portion to define our wrapper file of RAM

Architecture Behavioral of ur_ram is

Component first_ram

IP Core is acting as component of ur_ram.

End component;

Define all component parameters as signal

begin

u1: first_ram

port map (Generated core)

Assigning the core signals to the wrapper file signals to act as complete unit.

End Behavioral;

3.2 LSCIC-Phase-I (Results)

Last portion of LSCIC phase-I is to present results after successful simulation and synthesis. Figure 6 gives the simulation results after completion of PINGPONG RAM processing. It is important to note that same data is used throughout the testing of different aspects and characteristics of LSCIC pre coder.

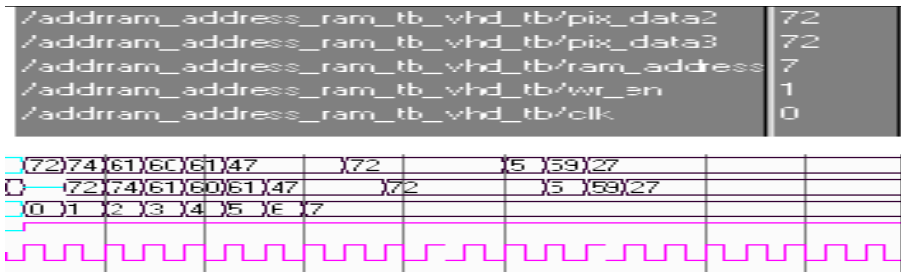


Fig. 6. RAM IP CORE operation

Figure 7. provides the results after joining first two defined modules DOWN SAMPLE and BUFFER CONTROL in proposed design.

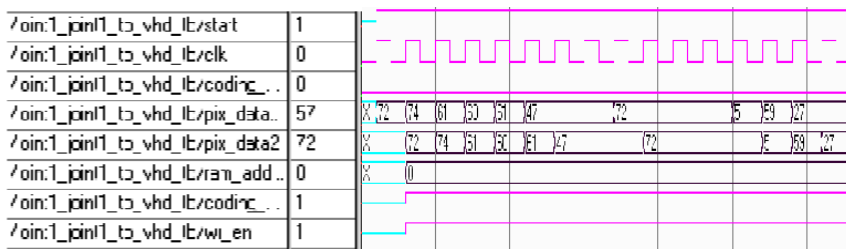


Fig. 7. Simulation Results of DOWN SAMPLE and BUFFER CONTROL connection

After acquiring the pixel data from BUFFER CONTROL, RAM comes in action and picks the pixels one by one into their respective addresses defined by Current-layer signal to perform WRITE operation. The two simulation results show complete coordination of data with 0% loss of data pixel till RAM module. But during post simulation it is found that some anonymous pixels due to circuit constraints are introduced but they seldom affect the quality of image. The relation between expected final result and experimental result is shown in Figure 8.

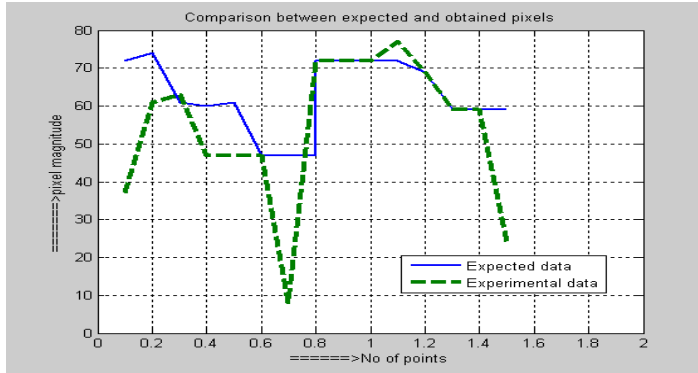


Fig. 8. Comparison of experimental and expected data results

4. Resource Allocation Results

After simulation and synthesis results, it is feasible to include hardware resource allocation results which design occupies on selected FPGA. For proposed design verification, Spartan 2-E, xc2s300e-6fg456 with 30,000 gates internally is utilized. Table 1 gives final module resource utilization information on target FPGA proved by Kamran 2006. It is already proved by Shinji Komori, in 1988 that for data driven processors, elastic pipelined causes high processing rate and smooth data stream concurrently. Our design is also meant to get concurrent data for processing for fast and efficient operation.

Logic Utilization	Used	Available	Utilization
Number of Slices	1372	3072	44%
Number of Slice Flip Flops	1379	6144	22%
Number of 4 input LUTs	2489	6144	40%
Number of bonded IOBs	49	329	14%
Number of TBUFs	7	3072	0%
Number of BRAMs	1	16	6%
Number of GCLKs	1	4	25%

Table 1. LSCIC (Stage 4) Resource Allocation Table

Table 1 provides the estimated device utilization summary of all modules implemented. Similarly data is collected for other subtasks and evaluation of resource utilization is made to become acquainted with the module complexity. Table 2 is the comparison of all sub modules with respect to resource utilization. It is required to be mentioned that stage 1 is comprised of down sample and buffer control module combination, stage 2 is formed by integrating stage 1 and RAM, stage 3 is organized by joining stage 2 and spatial redundancy module while stage 4 represents LSCIC pre coder by combining stage 3 and coder control module which causes the concurrent data to be extracted for coder and compression

process. Figure 9 gives the graph for the resource utilization versus module addition in the design. This graph also provides us the information about the complexity of the module, i.e., more complex the module is, more utilization of slices, flip flops and other resources available on destination FPGA device is found. Moreover, it gives negligible difference between %age resources utilization in stage 1 and stage 2 as these two stages are approximately equally complex in configuration.

Logic Utilization	Stage-4	Stage-3	Stage-2	Stage-1
Slices	44%	25%	4%	3%
Slice F/F	22%	12%	2%	2%
4 Input LUT	40%	23%	2%	2%
Bonded IOB's	14%	16%	10%	12%

Table 2. Resource Utilization comparison between different stages

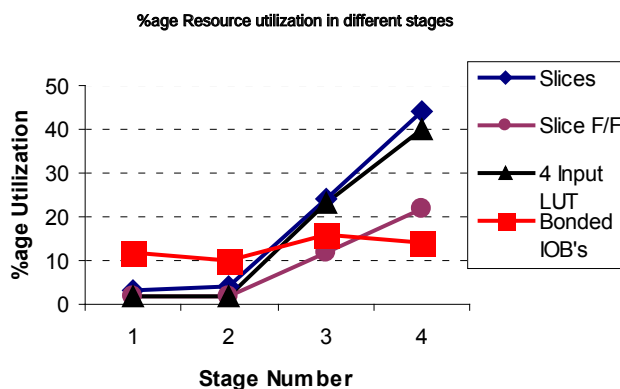


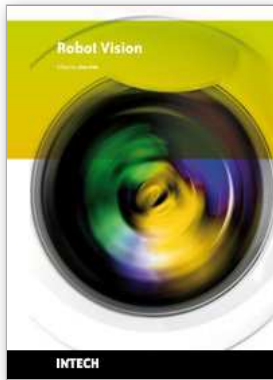
Fig. 9. Graphical Representation of Resource Utilization in different stages

5. Conclusion

The given LSCIC image and video compression is found quite amazing with respect to compression ratio and quality of reconstructed image. LSCIC is also adaptive with respect to band width variations. More experiments are being arranged for video reconstruction using wavelet transform with LSCIC pre coder.

6. References

- Ke Shen (1997). "A study of real time and rate scalable image and video compression", PhD Thesis, Purdue University, pp. 8-9, USA, December 1997
- Muhammad Kamran, Suhail Aftab Qureshi, Shi Feng and A. S. Malik, "Task Partitioning- An Efficient Pipelined Digital Design Scheme", Proceedings of IEEE ICEE2007, April 11-12, 2007, 147~156
- Muhammad Kamran, Shi Feng and Abdul Fattah Chandio, "Hardware Component Inclusion, Synthesis and Realization in Digital Design", Mehran University Research Journal of Engineering and Technology, July 2006, vol.25 issue 3, 223~230
- R. J. Gove, "The MVP: a highly-integrated video compression chip", Proceedings of IEEE Data Compression Conference, March 28-31, 1994, 215~224
- Shinji Komori, Hidehiro Takata, Toshiyuki Tamura, Fumiyasu Asai, Takio Ohno, Osamu Tomisawa, Tetsuo Yamasaki, Kenji Shima, Katsuhiko Asada and Hiroaki Terada, "An Elastic Pipeline Mechanism by Self Timed Circuits", IEEE Journal of Solid State Circuits, February 1988, vol. 23, issue 1, 111~117
- S. M. Akramullah, I. Ahmad, and M. Liou, "A data-parallel approach for real time Mpeg-2 video encoding", Journal of Parallel and Distributed Computing, vol. 30, issue 2, November 1995, 129~146
- V.Bhaskaran and K. Konstantindies, "Image and Video Compression standards algorithms and architectures", Massachusetts, Kluwer Academic Publishers, 1995



Robot Vision

Edited by Ales Ude

ISBN 978-953-307-077-3

Hard cover, 614 pages

Publisher InTech

Published online 01, March, 2010

Published in print edition March, 2010

The purpose of robot vision is to enable robots to perceive the external world in order to perform a large range of tasks such as navigation, visual servoing for object tracking and manipulation, object recognition and categorization, surveillance, and higher-level decision-making. Among different perceptual modalities, vision is arguably the most important one. It is therefore an essential building block of a cognitive robot. This book presents a snapshot of the wide variety of work in robot vision that is currently going on in different parts of the world.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Muhammad Kamran, Shi Feng and Wang YiZhuo (2010). LSCIC Pre Coder for Image and Video Compression, Robot Vision, Ales Ude (Ed.), ISBN: 978-953-307-077-3, InTech, Available from: <http://www.intechopen.com/books/robot-vision/lscic-pre-coder-for-image-and-video-compression>

INTECH

open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.