Design of High Performance FPGA Based Face Recognition System

I. Sajid, **M. M. Ahmed**, **I. Taj**, **M. Humayun**, **and F. Hameed**
Department of Electronic Engineering
Mohammad Ali Jinnah University, Islamabad, Pakistan

Abstract— A number of defense, security and commercial applications demand real time face recognition systems, especially when other biometric techniques are not feasible. Eigen values are widely used in engineering problems and particularly in face recognition algorithms. Floating point operations are used in Eigen values algorithms because Eigen values are highly sensitive for precision. Floating point operations are costly and complex in terms of hardware. Whereas fixed point technique with software hardware co-design (SHcoD) methodology reduces machine cycles and provides the flexibility in face recognition systems. It has been demonstrated that SHcoD concept can be used with dynamic partial reconfigurability to improve the conventional face recognition systems. An FPGA based novel design has been developed for efficient face recognition system which provides SHcoD, customization of algorithm and adaptability in the system. It has been shown that the proposed system is reasonably power efficient than floating point architecture and can be employed for portable applications.

1. INTRODUCTION

A number of defense, security and commercial applications demand real time face recognition systems [1], especially when other biometric techniques are not feasible. Finding a face from a video frame is one of the situations where face recognition may help reasonably. Human can easily and quickly identify this variance while machine is slower and error prone.

Now-a-days research has been focused the design and development of machine face recognition algorithms and their embedded implementation.

Mapping of matlab or C algorithm without modification in the software code on hardware, results may not be efficient or expected. Most of engineering applications uses Eigen values for projection of input data. Usually floating point operations are used in Eigen values algorithms, but they are costly and complex in terms of hardware [2]. Fixed-point implementation of floating-point operations is one of the classical techniques which may speed up the algorithm [3] with marginally lose in precision. Eigen values are highly depending upon precision of intermediate values. Trade-off between precision and efficiency would be analyzed for better use.

In the proposed system, RISC μ-processor decides of downloading the intellectual property IPs/functional units (FUs) at run time by observing instructions level demand. Downloading the memory FUs will enhance the on-chip memory, while arithmetic operation units may boost up the computing power.

Therefore a novel technique is being proposed which provides software-hardware co-design partitioning, customization of algorithm and adaptability in the system. In this technique energy, FPGA resources would be saved. The high level view of proposed system is presented in Fig. 2.

2. BACKGROUND

Biometric techniques/algorithms are not new methods for verification. Babylonian kings used clay finger prints for authenticity several year ago [4]. Egyptian used anatomical features like length of hand or half arm for biometric identification. Algorithm for face recognition can be broadly divided into holistic, feature base, and hybrid types. Holistic algorithms of face recognition address the global structure because it computes only the co-relation within the images [5]. Principal component analysis (PCA) is one of the holistic algorithm performs unsatisfactory when local details of images are vital [6].

Real time adaptive face recognition needs efficient software implementation of algorithms which would be designed as per the target reconfigurable hardware (FPGA) structure. Earlier designed systems [7, 8] did not produced acceptable results because they lacked customization of algorithms according to the target hardware resources. Therefore a decade earlier, conventional face recognition systems were not answering the real life challenges because they have limitation like producing power dissipation, poor response time and lack of adaptability.

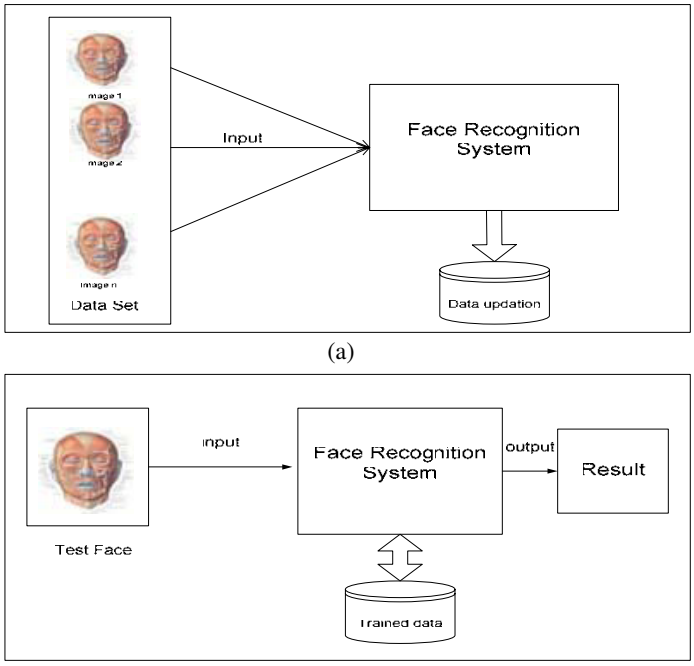

Figure 1: Overview of a general face recognition system (a) Training Phase (b) After the completion of training.

3. MOTIVATION

Intrusive and non-intrusive are main categorization of biometric techniques. Researcher are more seriously thinking about facial recognition as one of the suitable non-intrusive biometric especially after 9/11 accident.

Face recognition is one of the non-intrusive biometric techniques. Most of the face recognition algorithms produce burst of data flow between processor and memory [9, 10]. To run these algorithms

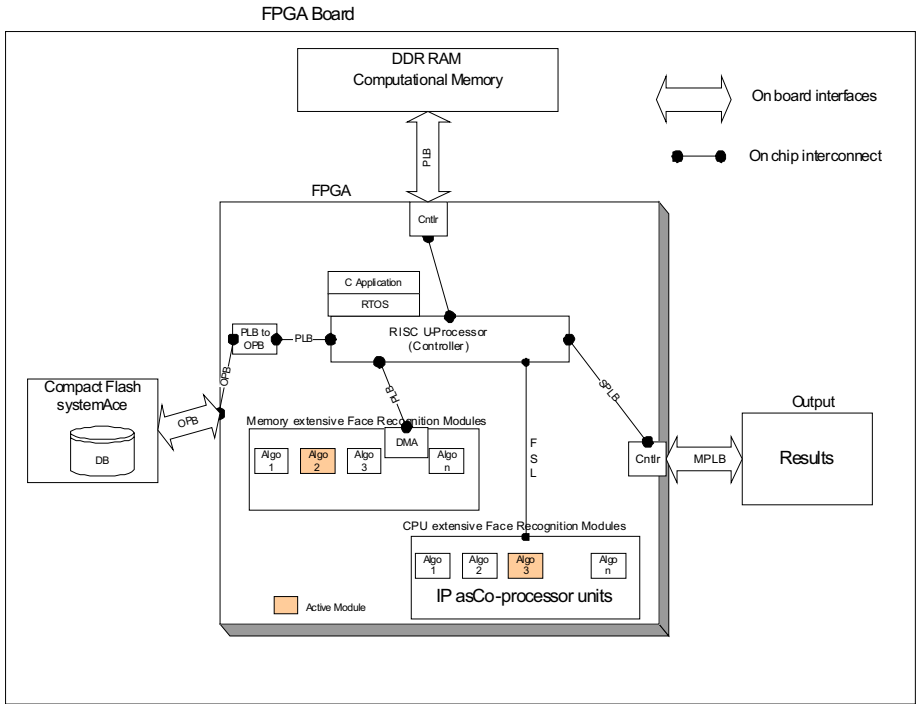

Figure 2: High level description Proposed system.

in general purpose machine will choke the processor performance because processor architecture may not be designed for such type of image processing algorithms. Specialized general processor based systems give acceptable performance however they are not flexible.

The design of hardware system in field programmable devices (FPDs) has many benefits over application specific integrated circuits (ASIC), like low cost rapid prototyping (LCRP) [11]. During the development cycle of new algorithms in hardware, LCRP has the key advantage. High logic capacity of field programmable gate array (FPGA) among FPDs makes possible designing of complex system on-chip (SoC). The application is slower and power hunger when algorithm is implemented in FPGAs as compared with the same applications implemented in custom ASICs [12]. To address with these two issues while maintaining LCRP and flexibility. Customization, software hardware co-designs (SHcoD), and dynamic partial reconfigurability (DPR) are introduced in face recognition algorithms.

4. CUSTOMIZATION/PARTITIONING OF ALGORITHM

Mapping of matlab or C algorithm without modification in the software code according to reconfigurable hardware, results may not be efficient or expected [12]. SHcoD means some part of the system works like hardware and other runs as software over the hardware. Manual partitioning of code according to the devices is not only laborious but also error prone. Single tool from any vendors does not provide efficient and full automatic partitioning for SHcoD [13], but integration of more than one tool may facilitate some process of SHcoD [12].

Few chunks of code in the algorithm should be identified, which are computational extensive and they can work independently. These codes are designed as hardware units while remaining algorithm works as high level C code. Few new tools facilitate to convert user defined high level code to HDL [12], however advanced architectural techniques like loop unrolling are necessary to optimization or efficient performance.

These converted HDL code is treated as intellectual property (IPs) soft-cores and necessary to integrate with high speed bus like FSL in the proposed architecture. The limitation of power, size and time would be addressed in embedded system with SHcoD methodology [14].

4.1. Precision Vs Efficiency

Most of the face recognition algorithm need for Eigen values. Eigen values and vectors are highly sensitive to precision of intermediate calculations. Software algorithms use double or float data type to avoid precision errors. Floating point computations in hardware are very expensive in terms of machine cycles. Therefore most of the real time systems do not like float operations.

Floating point supported u-processor PPC 405D5 or PPC 405F6 [15] core would be simpler solution which is efficient but not energy saver. Float point instructions are also consuming more micro-processor machine cycles.

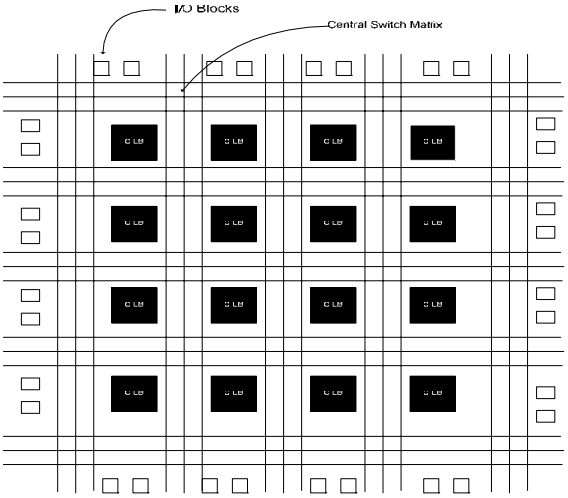

Figure 3: Basic architecture of an FPGA.

Fix-point arithmetic is a programmatic technique which preserves real values within certain limits depending upon fractional bits [3], while saves tremendously machine cycles for same oper-

ation. Fix-point is attractive for efficient and energy conscious hardware system, but it can not work such algorithm whose minimum and maximum value ranges fluctuate widely. To achieve acceptable accuracy of Eigen values by using fix format technique is a challenging task because simple fix format can not calculate numerically convergent Eigen values. Adaptive fix-point format is designed which adjust itself during the intermediate values. The Section 7 depicts the successful effort in this regard.

5. ADAPTABILITY

Various areas like control and automobile have taken the benefits of dynamic partial reconfigurability [16, 17]. Automated dynamic partial reconfiguration has achieved with the help of ROCCC compiler and partial bitstream (PARBIT) tool [24]. Dynamic reconfiguration is proposed for the face recognition in the context as shown in Fig. 3.

Adaptive systems may reduce the size, space and power of the system, because in this pattern physical resources of FPGA use same while system may have more computing power. Dynamic reconfiguration or adaptable system [10] may adjust itself at run time according to the requirement.

In dynamic reconfiguration or reconfigurable computing (RC), certain area of the device can be reconfigured while system is running on FPGA device [18]. RC is a technique through which the free part of FPGA can be reconfigured/converted into computing units or on-chip cache [9, 10] as per demand, while considering inter-communication constrains [19]. Basic architecture of FPGA illustrates that huge logic gates and registers may be configured by programmatically. Configurable logic block (CLB) contains four slices which are responsible for combinational, sequential logic and storage elements in FPGA as shown in Fig. 3 [20]. By considering FPGA architectural aspects, the proposed system may use optimal number of CLBs for the required functionality. This means less hardware resources are required to implement the algorithm as compare to implement the same algorithm without proposed technique. This will reduce power dissipation and other related VLSI issues in the system.

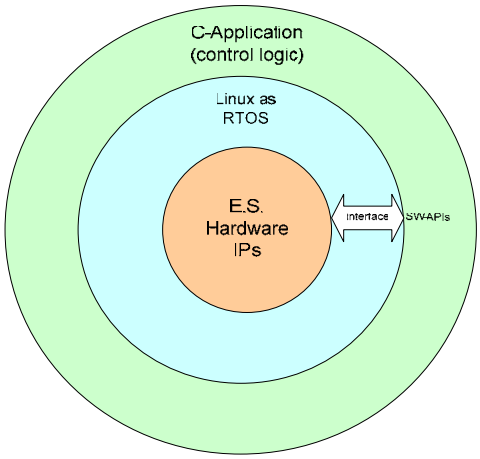

Figure 4: Layer view of system.

6. DESIGN OF PROPOSED SYSTEM

Assembly view in Fig. 5 gives hardware integration view of system. Peripheral local bus (PLB), peripheral on chip bus (OPB), fast simplest link (FSL) and fabric core bus FCB interfaces are proposed in system. PLB is high speed 64-bits address and 128-bits data bus. Processor core and bus controller are integrated with on-chip PLB. PLB masters are attached to PLB through separate address bus, while PLB slaves are connected to PLB through shared bus [21]. Master driven operation is achieved by central arbitration mechanism. High speed devices are connected with direct PLB, like DDR memory. OPB is designed for common devices. Processor core communicates with TFT controller and UART through OPB bus. OPB is 64-bits address and 32 or 64 bits data bus [22]. Intellectual Property Interface (IPIF) is used as sandwich between IP and selected bus [23].

Power PC is used as controlling unit in the proposed architecture. Like processing elements in the architecture of Hau T. Ngo [7], we proposed IPs or patch of algorithm. Layer view (Fig. 4)

of HW/SW design methodology indicates that some computation is part of hardware and other is part of software application.

CPU extensive parts like matrix multiplication or calculation of covariance matrix are implemented as IP core. Thses IPs are implemented as co-processing units [24]. These co-processing units are connected to auxiliary processing unit (APU) through FSL/FCB channels. This hierarchy may distribute/balance the load of embedded system (ES) in reconfigurable media.

The calculation of distance vectors based on neighboring pixel value is used frequently in face recognition algorithms. Such memory extensive modules is proposed to connect with PLB, co-processing unit is proposed to connect with FLS in virtex II or FCB in virtex4 devices. DMA controller drives PLB for rapid memory access.

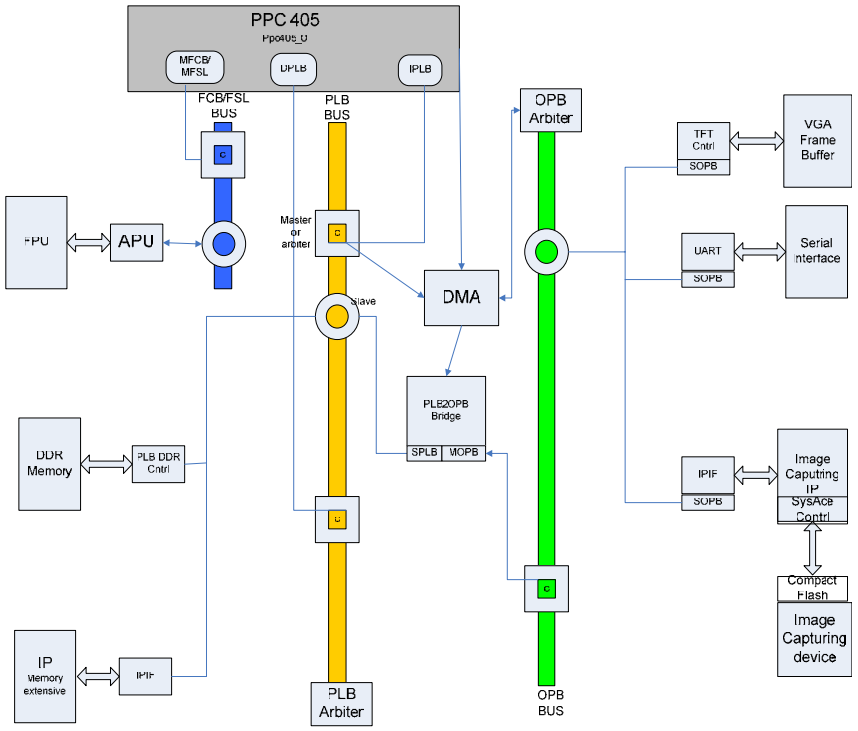

Figure 5: Hardware component integration of the connected to PLB.

Face images are treated as matrices in image processing. Matrices are store on non-volatile media Compact flash (CF). Processing of images explode burst of data. Streaming model is proposed for image processing applications due to high speed data rate transfer between cpu and memory. These matrices are used for features extraction in the training process. OPB has sufficient speed for CF disk and it is synchronous with processor core [22]. On board CF is controlled by sysace controller (hardware). Through C level APIs this data is read from CF and would be loaded into DDR memory for further face recognition computation, for example to calculate eigen faces. DDR is fast memory storage device; therefore it is connected to PLB.

Native PPC 405 core does not support floating point instruction [25]. Face recognition algorithms badly need real number arithmetic. PPC 405F6 core is closely interface by auxiliary processor unit (APU) controller [25]. Floating point instructions are executed by FCM. This architecture of PPC 405F6 can only be supported by Virtex-4-Fx family. Matrices multiplication of real numbers is used for precision comparison. In this regard floating point unit (FPU) is connected to micro-processor through fabric co-processor module (FCM).

With this motivation and SHcoD of face recognition algorithm imparts the designing detail of IPs/function units. This design is part of my PhD work. These units will be coded in C/HDL languages. Partial bit-streams are created after synthesis, routing and mapping steps. These bitstreams is used for adaptive nature of the system. In this way an efficient, low power system is designed without compromising on the functionality.

7. RESULTS

PCA needs weighted vector for recognition. First three values contribute 99.43195583 percent of the whole weight, while 99.42413122 percent in case of fixed point implementation. Therefore recognition or reconstruction result does not affect more than 0.0079 percent. On the other hand fixed point operations on vertex device are supported by native hard core. Floating point unit (APU) can be connected through APU in vertex4 FX devices. Fixed point computations are fully compliant with user instruction set architecture of Power PC 405 hard core instance. Floating point MAC operation in PPC 405 instance of vertex4 consume 15 times more cycles. FPU takes 1600 slices, 2 block ram and 4 dsp blocks. This novel implementation of HH algorithm saves power in the cost of losing precision no more than .008 percent in the diagonal values of computed matrix.

Data set images 20

resolution 292x376

Table 1. Simulation results of house holder algorithm

H.H.Floating	H.H Fixed point	Diff(FLP-FP)	% Diff
33547739136	33550653440	-2914304	-0.008687035
1800899968	1800899840	128	7.10756E-06
255354944	250970624	4384320	1.716951288
31969308	22567680	9401628	29.40829373
22742982	22560000	182982	0.804564878
20840046	19640064	1199982	5.758058308
19436386	19608320	-171934	-0.884598608
17626140	19313920	-1687780	-9.57543739
12864361	19285760	-6421399	-49.91619094
12475056	12296448	178608	1.431721028
9660825	10238976	-578151	-5.984488902
8189179.5	8933632	-744452.5	-9.090684848
7029789	7297536	-267747	-3.808748741
6516366	7192320	-675954	-10.37317425
6399316.5	6599680	-200363.5	-3.13101407
6343345.5	6355712	-12366.5	-0.194952332
6315585.5	6252032	63553.5	1.006296249
5948435.5	6220288	-271852.5	-4.570151261
5650524.5	6145536	-495011.5	-8.760452238
3394177.5	5703424	-2309246.5	-68.03552554

Eigen values weight		
	FLP	FP
Total	35807395872	35808735232
First 3	35603994048	35602523904
diff	203401823.5	206211328
%	99.43195583	99.42413122
% diff	0.007825	

8. CONCLUSION/FUTURE WORK

1 FPGA based high level design and component integration view has been demonstrated.

2 Minimum and maximum ranges of intermediates values have been found during iterative part of householder (HH) and QL decomposition algorithms.

3 Fixed point implementation HH algorithm saves thousands of machine cycles in the cost of losing 008% weight in highest three Eigen values.

The following steps would be required for efficient and low power implementation of HH on FPGA. In fact the following steps have been proposed in my PhD synopsis.

a Fix-point dynamic macros that accommodate the step (3) requirement would be designed.

b Modified code which uses dynamic macros would be run on hardware (without floating point unit) and total machine cycles used for such a scenario would be calculated. Time of execution of HH algorithm in terms of nano second and power consumed in term of pico watts would be calculated.

c Convert householder and QL algorithmic part into HDL and then compare the machine cycles and power factor.

d Acceptable results in terms of efficiency and accuracy in step (d) with the floating point co-processor would be appreciable academic contribution in the face recognition particularly and other engineering fields specially.

REFERENCES

1. Zhao, W., R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *ACM Computing Surveys*, Vol. 35, No. 4, Dec. 2003.
2. Kang, J. and W. Sung, "Fixed-point C compiler for Tms320c50 digital signal processor," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 1, 707–710, Apr. 1997.
3. Kum, K.-I., J. Kang, and W. Sung, "A floating-point to fixed-point C converter for fixed-point digital signal processors," *Proc. 2nd SUIF Compiler Workshop*, August 1997.
4. Ashbourn, J., *Biometrics: Advanced Identity Veri_cation*, Springer, 2002.
5. He, X., S. Yan, Y. Hu, P. Niyogi, and H. Zhang, "Face recognition using laplacianfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27, No. 3, 328–340, Mar. 2005.
6. Belhumeur, P. N., J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 7, 711–720, Jul. 1997.
7. Ngo, H. T., R. Gottumukkal, and V. K. Asari, "A flexible and efficient hardware architecture for real-time face recognition based on eigenface," *IEEE Computer Society Annual Symposium on VLSI: New Frontiers in VLSI Design (ISVLSI'05)*, 280–281, 2005.
8. Zaki, G. F., R. A. Girgis, W. W. Moussa, and W. R. Gobran, "Using HW/SW Co-design to implement an embedded face recognition/verification system on an FPGA".
9. Sangireddy, R., H. Kim, and A. K. Somani, "Low-power high-performance reconfigurable computing cache architectures," *IEEE Transactions on Computers*, Vol. 53, No. 10, 1274–1290, Oct. 2004.
10. Kim, H., "Towards adaptive balanced computing (ABC) using reconfigurable functional caches (Rfcs)," PhD Thesis at Lowa State University, 2001.
11. Brown, S. and J. Rose, "Architecture of FPGAs and CPLDs: A tutorial," *IEEE Computer Society Press*, Vol 13, Issue 2, Los Alamitos, CA, USA, Jun. 1996.
12. Pellerin, D. and S. Thibault, *Practical FPGA programming in C*, Prentice Hall PTR, ISBN: 0-13-154318-0.
13. Saha, P., "Automatic software hardware co-design for reconfigurable computing systems," *17th International Conference on Field Programmable Logic and Applications*, 27–29, Amsterdam, Netherlands, Aug. 2000.
14. Cox, J., J. Ghosn, and P. N. Yianilos, "Feature-based face recognition using mixture-distance," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'96)*, 209, 1996.
15. PowerPC 405 Processor Block Reference Guide, Embedded Development Kit, UG018 (v2.1), Jul. 20, 2005.
16. Toscher, S., T. Reinemann, and R. Kasper, "An adaptive FPGA mechatronic control system supporting partial reconfiguration of controller functionalities," *Nas/Esa Conference on Adaptive Hardware and Systems*, 2006.
17. Uiimann, M. and M. Hubner, "An FPGA run-time system for dynamical on-demand reconfiguration," *18th International Parallel and Distributed Processing Symposium IEEE*, 2004.
18. Two Flows for Partial Reconfiguration: Module Based or Difference Based, Xilinx, Sep. 2004.
19. Robertson, J. I., "A design flow for partially reconfigurable hardware," *ACM Transactions on Embedded Computing Systems*, Vol. 3, No. 2, 257–283, May 2004.
20. Virtex-II Pro and Virtex-II Pro X Platform FPGAs: Complete Data Sheet DS083, Xilinx, Oct.
21. Virtex-II Pro and Virtex-II Pro X Platform FPGAs: Complete Data Sheet DS083, Xilinx, Oct. 2005.
22. IBM On-Chip Peripheral Bus Architecture Specifications Version 2.1, Apr. 2001.
23. OPB IPIF architecture, Xilinx, DS414, Aug. 2004.
24. Mitra, Z. G., A. Baner, and W. Najjar, "Dynamic co-processor architecture for software acceleration on csoCs," *IEE Int. Conf. on Computer Design (ICCD)*, Oct. 2006.
25. UG018 (v2.1), PowerPC 405 Processor Block Reference Guide, July 20, 2005.