# X-MatchPRO: A ProASIC-Based 200 Mbytes/s Full-Duplex Lossless Data Compressor

José Luis Núñez, Claudia Feregrino, Simon Jones, Stephen Bateman*

Electronic Systems Design Group, Loughborough University,
Loughborough, Leicestershire LE11 3TU. England.
*BridgeWave Communications, Inc.3350 Thomas Rd, Santa Clara, CA 95954, USA.

J.L.Nunez-Yanez@lboro.ac.uk, C.Feregrino-uribe@lboro.ac.uk,
S.R.Jones@lboro.ac.uk, SBateman@Bridgewave.com

**Abstract.** This paper presents the full-duplex architecture of the X-MatchPRO lossless data compressor and its highly integrated implementation in a non-volatile reprogrammable ProASIC FPGA. The X-MatchPRO architecture offers a data independent throughput of 100 Mbytes/s and simultaneous compression/decompression for a combine full-duplex performance of 200 Mbytes/s clocking at 25 MHz. Both compression and decompression channels fit into a single A500K130 ProASIC FPGA with a typical compression ratio that halves the original uncompressed data. The device is specially targeted to enhance the performance of Gbit/s data networks and storage applications where it can double the performance of the original system.

## 1 Introduction

Lossless data compression [1], where the original data is reconstructed exactly after decompression is accepted as a tool that can bring important benefits to an electronic system. Its applications have been increasing over the past years thanks to the arrival of compression standards and a combination of pressure for more bandwidth and storage capacity while still reducing power consumption. Lossless data compression has been successfully applied to storage systems (tapes, hard disk drives, solid state storage, file servers) and communication networks (LAN, WAN, wireless).
The remainder of this paper is organized as follows. Section 2 describes the basic characteristics of the X-MatchPRO algorithm. Section 3 depicts the X-MatchPRO full-duplex architecture. Section 4 compares our device with other high-performance lossless data compressors. Finally section 5 concludes this paper.

## 2 The X-MatchPRO Algorithm

The X-MatchPRO algorithm [2-4] uses a dictionary of previously seen data and attempts to match or partially match the current data element with an entry in the dictionary. Each entry is 4 bytes wide and several types of matches are possible where all or some of the bytes at different positions inside the tuple match. Those

G. Brebner and R. Woods (Eds.): FPL 2001, LNCS 2147, pp. 613-617, 2001.

bytes that do not match are transmitted literally. This partial match concept gives the name to the procedure- the X referring to 'don't care'. At least 2 bytes have to match and when no valid match is generated a miss is codified adding a single bit to the literal. The dictionary is maintained using a move to front (MTF) strategy [5] whereby a new tuple is placed at the front of the dictionary while the rest move down one position. When the dictionary becomes full the tuple placed in the last position is discarded leaving space for a new one. X-MatchPRO reserves one location in the dictionary to code internal runs of full matches at location zero. This Run-Length-Internal (RLI) technique is used to efficiently code any 32-bit repeating pattern. Additionally an Out-of-Date-Adaptation (ODA) policy is used in X-MatchPRO for throughput purposes. This means that adaptation at time t+2 takes place using the adaptation vector generated at time t.

## 3 The X-MatchPRO Hardware

The architecture is depicted in Fig. 1. The full-duplex architecture has 5 major components, namely: the Model, the Coder, the Decoder, the Packer and the Unpacker.

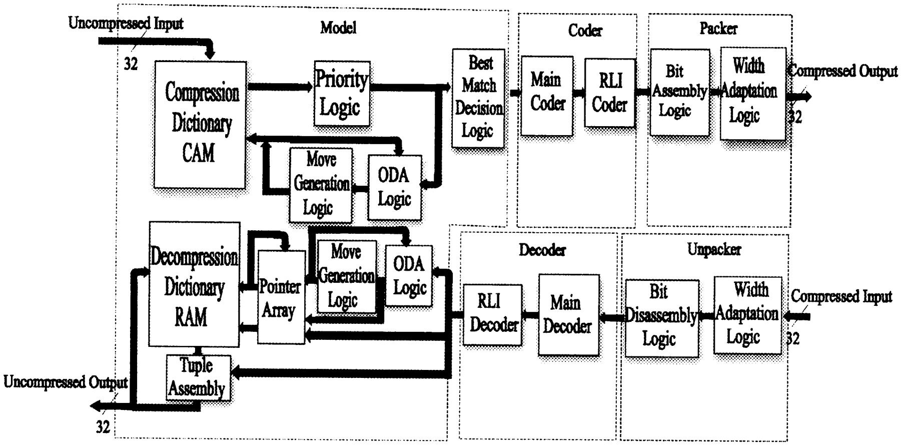

**Fig. 1.** X-MatchPRO full-duplex architecture.

The model is the section of the compressor whose function is to identify where the redundancy is located in a block of data and signal repetitive data sequences to the coder. The coder function is to use the information provided by the model to produce a minimum output of bits and obtain compression. The decoder function is to decode the compressed input stream and provided the model with a combination of dictionary address plus literal data so the model can reproduce the original uncompressed data. The packer function is to pack the variable length codewords output from the coder into fixed-length codewords of 32 bits. Finally, the unpacker function is to break the fixed-length codewords input from the compressed bus into variable length codewords to be processed in the decoder. The initialization of the compression CAM sets all words to zero. This means that a possible input word formed by zeros will generate multiple full matches in different locations but in this case the algorithm simply selects the full match closer to the top. This operational model initializes the dictionary to a state where all the words with location address bigger than zero are declared invalid without the need for extra logic because location x can not generate a match until location x-1 has been updated. X-MatchPRO uses a simple coprocessor style interface to communicate with the rest of the system. Compression and decompression commands are issued through a common 16 bit control data port. A 3-bit address is used to access the internal registers that store the commands plus information related to compressed and uncompressed block sizes for reading or writing. A total of 6 registers form the register bank. 3 registers are used to control the compression channel and the other 3 for the decompression channel. The first bit in the address line indicates if the read/write operation accesses compression or decompression registers. The chip is designed to compress any block size ranging from 8 bytes to 32 Kbytes. A decompression operation can be requested in the middle of a compression operation and vice versa. The full-duplex architecture using a 16-word dictionary has been implemented in a A500K130 ProASIC FPGA [6].

## 4 Results

Table 1 shows a comparison of the FPGA-Based X-MatchPRO implementation against several popular high-performance ASIC compressors. The selection includes:

1. The the ALDC1-40S [7] (IBM) and the AHA3521 [8] (AHA) that implement the ALDC [9] (Adaptive Lossless Data Compression) algorithm. This algorithm is a LZ1 derivative developed by IBM.
2. The AHA3211 [10] that implements the DCLZ [11] (Data Compression Lempel Ziv) algorithm. This algorithm is a LZ2 derivative developed by Hewlett/packer and AHA.
3. The Hi/fn 9600 [12] that implements the LZS [13] (Lempel-Ziv Stac) algorithm . This algorithm is another LZ1 derivative developed by STAC/Hifn.

Table 1 reports the complexity of the X-MatchPRO design in ProASIC tile's. Tile is the basic logic unit in the architecture of the ProASIC technology. Actel ProASIC tiles are simple blocks that can implement a logic function with 3 inputs and 1 output such as an AND gate or a flip-flop. Each tile can be configured to implement one of these simple functions using the internal non-volatile FLASH-based switches. Actel ProASIC architecture is fine-granularity and flat so the simple tiles are repeated across the device forming a matrix of identical logic elements. Dedicated memory blocks are group in the north side of the device. There are a total of 20 memory blocks in a A500K130 and each of them can implement 2304 bits of fully-synchronous dual port RAM. The design uses 70% of the device logic that is approximately equivalent to 30 Kgates and the 20 blocks of embedded RAM available (5 Kbytes). The total gate count equivalent of logic plus memory is 210 K gates. Table 1 shows that X-MatchPRO can achieve higher performance throughput than the ASIC compressors with a lower clock ratio and this is due to its optimal parallel architecture. The compression ratio figure in the last row is a ratio of output bits to input bits (output_bits/input_bits) and it is based on a data set formed by 100 Mbytes of data found in the main memory of a UNIX workstation compressing data in 4 Kbytes blocks. The FPGA-based X-MatchPRO uses a very small dictionary of only 16 locations and that limits its compression performance. The ASIC compressors use dictionary sizes from 512 to 2048 positions.

| DEVELOPERS | | IBM | Advance Hardware Architectures (AHA) | | STAC Electronics | System Design Group Loughborough University |
|---|---|---|---|---|---|---|
| CHIP* | | ALDC1-40S | AHA3521 | AHA3231 | Hi/fn 9600 | X-MatchPRO |
| TECHNOLOGY DETAILS | PROCESS | IBM CMOS 0.8 micron triple-level gate array/std cell | 0.5 micron CMOS | 0.5 micron CMOS | 0.35 micron gate array/std cell | 0.25 micron FLASH-CMOS FPGA Actel A500K ProASIC |
| | COMPLEXITY | 70 Kgates | Not Stated | Not Stated | 100 Kgates | 9039 TILE's 70% of a A500K130-BG456 |
| | CLOCK SPEED | 40 MHZ | 40 MHZ | 40 MHZ | 80 MHZ | 25 MHz |
| THROUGHPUT | | 40 Mbytes/s | 20 Mbytes/s | 20 Mbytes/s | 80 Mbytes/s | 100 Mbytes/s |
| FULL-DUPLEX PERFORMANCE | | N/A | N/A | N/A | 160 Mbytes/s | 200 Mbytes/s |
| ALGORITHM | | ALDC | ALDC | DCZL | LZS | X-MatchPRO |
| EXTERNAL RAM REQUIRED | | NO | NO | NO | NO | NO |
| COMPRESSION RATIO | | 0.44 | 0.44 | 0.52 | 0.44 | 0.58 |

**Table 1.** X-MatchPRO comparison.

# 5 Conclusions

X-MatchPRO offers unprecedented level of compression/decompression throughput in a FPGA implementation of a lossless data compression algorithm for general application. The full-duplex implementation effectively uses the resources available in the FPGA to simultaneously handle a compressed and uncompressed data stream. The use of a fine granularity device like the ProASIC where each block defines a very simple logic function, has proven to be well suited to implement the CAM-based dictionary that represents most of the logic present in the device. Other FPGA architectures where the building blocks implement mixed combinatorial and sequential functions offer poorer utilization ratios

Acknowledgements: We acknowledge with gratitude the support donated by Actel/Gatefield corporation.

## References

[1] M. Nelson, 'The Data Compression Book' , Prentice Hall, 1991.

[2] M.Kjelso, M.Gooch, S.Jones, 'Design & Performance of a Main Memory Hardware Data Compressor', Proceedings 22nd EuroMicro Conference, pp. 423-430, September 1996, Prague, Czech Republic.

[3] J.Nuñez, C. Feregrino, S.Bateman, S.Jones, 'The X-MatchLITE FPGA-based Data Compressor', Proceedings 25th EuroMicro Conference, pp126-133, September 1999.

[4] José Luis Núñez,, Simon Jones,'The X-MatchPRO 100 Mbytes/second FPGA-Based Lossless Data Compressor', proceedings of Design, Automation and Test in Europe, DATE Conference 2000, pp. 139-142, March, 2000.

[5] J. L. Bentley, D. D. Sleator, R. E. Tarjan, V. K. Wei, 'A Locally Adaptive Data Compression Scheme', Communications of the ACM, Vol. 29, No. 4, pp. 320-330, April 1986.

[6] 'ProASIC™ 500K Family', Data sheet, Actel corporation, 955 East Arques Avenue, Sunnyvale, CA, 2000.

[7] 'ALDC1-40S-M', Data sheet, IBM Microelectronics Division, 15080 Route 52, Bldg 504 Hopewell Junction, NY, 1994.

[8] 'AHA3521 40 Mbytes/s ALDC Data Compression Coprocessor IC', Product Brief, Advanced Hardware Architectures Inc, 2635 Hopkins Court, Pullman, WA, 1997.

[9] J.M.Cheng and L.M.Duyanovich, 'Fast and Highly Reliable IBMLZ1 Compression Chip and Algorithm for Storage',Hot Chips VII Symposium, August 14-15, pp. 155-165, 1995.

[10] 'AHA3211 20 Mbytes/s DCLZ Data Compression Coprocessor IC', Product Brief, Advanced Hardware Architectures Inc, 2635 Hopkins Court, Pullman, WA, 1997.

[11] 'Primer: Data Compression (DCLZ)', Application Note, Advanced Hardware Architectures Inc, 2635 Hopkins Court, Pullman, WA, 1996.

[12] '9600 Data Compression Processor', Data Sheet, Hi/fn Inc, 750 University Avenue, Los Gatos, CA, 1999.

[13] 'How LZS Data Compression Works', Application Note, Hi/fn Inc, 750 University Avenue, Los Gatos, CA, 1996.