

ulation condition as that of Fig. 6 in [3]. The PDIC detector shows the same performance in both of our simulations and Fig. 6 in [3]. The PDWIC detector provides  $\sim 1.8$  dB gain in SNR at  $\text{BER} = 10^{-3}$  compared with that of the previous PDIC detector. The PDWIC detector outperforms the PDIC detector by one order of magnitude of BER in the high SNR region. In Fig. 2b, the BER performance of the proposed PDWIC scheme is obtained by simulations for a five-user system with  $N = 16$  over a frequency-nonsselective Rayleigh fading channel. Our proposed scheme results in  $\sim 0.8$  dB gain in SNR at  $\text{BER} = 10^{-3}$  compared with that of the previous PDIC detector. The error rate of the PDWIC detector is less than half that of the previous PDIC detector in the high SNR region. Thus, we can see that the proposed PDWIC scheme works significantly better than the previous PDIC detector under interference limited environments.

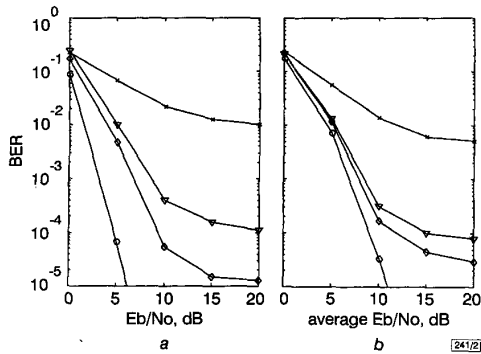


Fig. 2 BER performance

- a Coded eight-user system with  $N = 24$  over AWGN channel  
 b Coded five-user system with  $N = 16$  over frequency-nonsselective Rayleigh fading channel  
 — conventional matched filter  
 ▽ PDIC  
 ◇ PDWIC (proposed scheme)  
 ○ single user bound

**Conclusions:** We have proposed a weighted interference cancellation scheme for improving the performance of post-decoding interference cancellation detectors in a convolutionally coded CDMA system. A weight determination method for partial cancellation is proposed in which the information bit error probability from the soft-output Viterbi algorithm is used.

© IEE 1999

Electronics Letters Online No: 19991134  
 DOI: 10.1049/el:19991134

13 July 1999

Weon Yong Joo and Hwang Soo Lee (Department of Electrical Engineering, Korea Advanced Institute of Science and Technology, 373-1 Kusong-Dong, Yusong-Gu, Taejeon 305-701, Korea)

E-mail: wyjoo@rcunix.kotel.co.kr

Soon Young Yoon (Telecommunication R&D Center, Samsung Electronics Co., Ltd., Korea)

Hwang Soo Lee: Also with Central R&D Laboratory, SK Telecom, Korea

## References

- VARANASI, M.K., and AAZHANG, B.: 'Near-optimum detection in synchronous code-division multiple-access systems', *IEEE Trans.*, 1991, COM-39, (5), pp. 725-736
- ELEZABI, A., and DUEL-HALLEN, A.: 'Combined error correction and multiuser detection for Rayleigh fading synchronous CDMA channels'. Proc. 33rd Annual Allerton Conf. Communication, Control, and Computing, 1995, pp. 1-10
- ELEZABI, A., and DUEL-HALLEN, A.: 'Two-stage receiver structures for coded CDMA systems'. Proc. VTC '99, 1999, pp. 1425-1429
- KIM, S.R., CHOI, I.K., KANG, S.B., and LEE, J.G.: 'Adaptive weighted parallel interference cancellation for CDMA systems', *Electron. Lett.*, 1998, 34, (22), pp. 2085-2086
- DIVSALAR, D., SIMON, M.K., and RAPHAELI, D.: 'Improved parallel interference cancellation for CDMA', *IEEE Trans.*, 1998, COM-46, (2), pp. 258-268
- HAGENAUER, J., and HOEHER, P.: 'A Viterbi algorithm with soft-decision outputs and its applications'. Proc. GLOBECOM '89, 1989, pp. 47.1.1-47.1.7

## FPGA based runtime configurable clause evaluator for SAT problems

P.H.W. Leong and C.K. Chung

An FPGA based clause evaluator for Boolean satisfiability problems is presented in which a customised bitstream is directly generated from the problem specification, avoiding the need for resynthesis. A three orders of magnitude improvement in reconfiguration time was seen over the standard approach for a 50 variable, 80 clause problem.

**Introduction:** There has been considerable recent interest in the application of field programmable gate array devices (FPGAs) as accelerators for solving constraint satisfaction problems (CSPs) and, in particular, the Boolean satisfiability (SAT) problem. The Boolean SAT problem is a CSP in which the constraints are represented by a Boolean function of  $m$  binary variables ( $F(x_0, x_1, \dots, x_{m-1})$ ) in a product of sums form. Each sum term is a clause,  $C_i$ , and is the sum of single literals, where a literal is a variable or its negation. The Boolean satisfiability problem (SAT) is concerned with finding a variable assignment that makes  $F = 1$  (satisfiable) or proving that  $F = 0$  (unsatisfiable). An important component of any SAT machine is a clause evaluator which evaluates the clauses  $C_0, \dots, C_{n-1}$  with different variable assignments. Inputs to the clause evaluator are the assignment of the variables and the outputs are the evaluations of the clauses. SAT solving systems can be designed with a fixed circuit to perform the search and the clause evaluator customised for different sets of constraints [1].

In most previous implementations of SAT the problem was approached by using a computer program to generate a customised clause evaluation circuit for a particular SAT problem [2, 3]. The design is then synthesised, placed and routed (P&R) to produce a bitstream which is downloaded to the FPGA, which in turn is used to search for a solution to the original SAT problem. This approach requires a complete iteration of the synthesis, P&R cycle for each new set of constraints, and can take several hours for a large design, precluding its use in real-time systems. Recently, runtime reconfigurable systems have been employed to address this problem, modifying the bitstream in a problem specific fashion without requiring resynthesis [1, 4]. To the best of our knowledge, all prior runtime configurable systems have used Xilinx XC6200 series devices which document the manner in which the bitstream relates to the hardware of the device. However, XC6200 devices have been discontinued by Xilinx and also have very small logic capacity (the largest reported runtime reconfigurable system only supporting 13 variables and 29 clauses [4]).

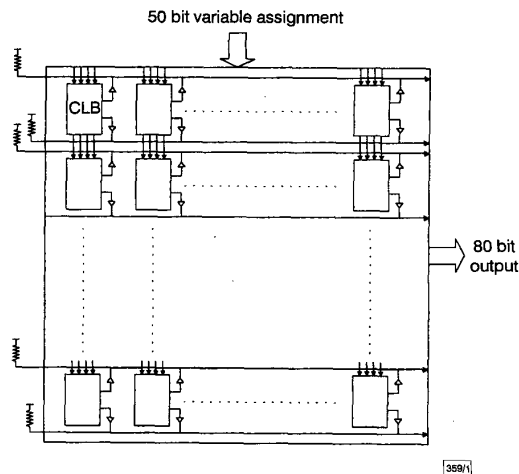


Fig. 1 Block diagram of clause evaluator

In this Letter, we present an architecture for a clause evaluator using industry standard Xilinx XC4000 series devices [4] in which the bitstream is directly generated from the constraint problem. It has the advantage over previous designs in that it (i) does not require the synthesis and P&R steps and (ii) supports XC4000 series devices. The clause evaluator does not place any restrictions

on the number of literals in a clause. Furthermore, the same architecture can be used for the recently announced Xilinx Virtex devices which have much larger capacity and a documented bitstream format [6].

**Clause evaluator:** Fig. 1 shows a block diagram of the clause evaluator. It contains an array of configurable logic blocks (CLBs), the logic primitives of Xilinx XC4000 devices [5]. Each CLB is configured as two  $16 \times 1$  RAM memories and produces two outputs on different rows as illustrated in the Figure. The inputs to the clause evaluator are 50 bits corresponding to the variables and the outputs are the 80 clause evaluations.

Each row of the array in Fig. 1 corresponds to two clauses, the outputs appearing in the two wires immediately above and below the CLB. Each 1/2 CLB in the row has its address lines connected to four consecutive inputs of the variable to be evaluated. The output of the CLB is the evaluation of the sum terms for the input variables to which it is connected. The RAM outputs are connected to the row line through an open drain buffer, implementing the sum terms as a wired-AND (which is equivalent to an active low wired-OR operation). Note also that a pullup resistor is connected to each row.

As an example, for the clause  $C_0 = \bar{x}_0 + x_2 + x_5$ , the first column CLB of Fig. 1 implements  $\bar{x}_0 + x_2$  (as a lookup table) and the second column CLB implements  $x_5$ . If one or more literals is evaluated to be a logical true (in our example, this corresponds to  $x_0$  being false or  $x_2$  being true or  $x_5$  being true), its CLB will drive the row low, asserting the (active low) output.

All the components and routing were placed into predefined locations and routed automatically by the Xilinx Epic Editor from a script created by a C program. The interconnect for the inputs and outputs of the clause evaluator are implemented using longlines [5] which are intended for high fanouts that are distributed over long distances.

As the bitstream format for XC4000 series devices is not documented, the mapping between RAM contents and the bitstream was determined by using a program to produce designs with known patterns in each RAM, compiling the design to a bitstream using the standard Xilinx tools and then finding the pattern in the resulting bitstream. A table of the starting positions of all the RAMs in the FPGA's bitstream was thus compiled. Using this table, another C program can configure the contents of the memories in the bitstream directly from a SAT problem specification in the standard DIMACS benchmark format [7].

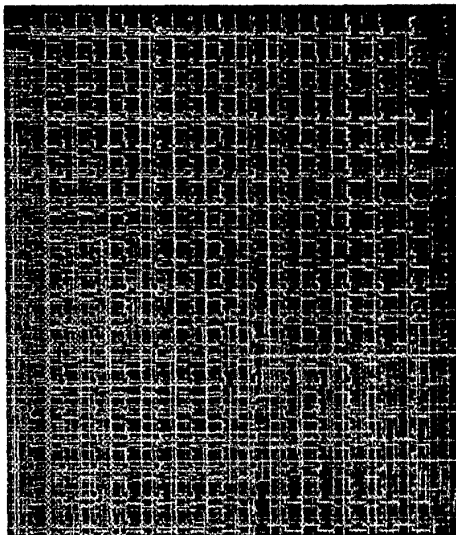


Fig. 2 Layout of clause evaluator

**Results:** The clause evaluator was tested on a DIMACS 3-SAT (i.e. each clause has three literals) benchmark problem (aim-50-1\_6-yes1-1) [7] with 50 variables and 80 clauses. On a Sun Ultra 5/10 UPA/PCI (UltraSPARC-IIi 270MHz), the time required to generate the bitstream for this problem was 0.7s. Using the same UltraSPARC-IIi 270MHz machine, we also

produced a VHDL description of the clause evaluator and performed synthesis (407s) and place and route (660s), giving a total implementation time of 1067s. Thus the runtime reconfigurable version realises a three order magnitude improvement over the resynthesis approach.

The resulting runtime configurable implementation (shown in Fig. 2) required 520 CLBs, approximately 1/4 of the resources of a Xilinx XC4062XL device. The Xilinx implementation tools report a worst case delay of 40ns. This implementation was successfully tested at 25MHz on a single XC4062XL chip of an Annapolis Micro Systems Wildforce board. A profiling analysis of a software implementation of GSAT (version 41) by Selman and Kautz [8] showed that it required  $3.1\mu\text{s}$  per clause evaluation. Note that in this program, variable flips are performed in an intelligent fashion, only with the clauses affected by a variable flip being recomputed. Thus the FPGA implementation of the clause evaluator is 77 times faster than the software implementation.

**Conclusion:** An architecture for a runtime reconfigurable clause evaluator which generates a customised circuit for a particular problem instance has been reported. Distributed RAM devices in a field programmable gate array (FPGA) were utilised to customise the circuit by directly changing the bitstream of the FPGA. This approach showed a 1500 times increase in speed over resynthesis from an HDL and a 77 times improvement in execution speed over an optimised software implementation. We envisage that this technique could be used in hardware based real time constraint solving systems and may have applications in signal processing, robotics and control.

© IEE 1999

Electronics Letters Online No: 19991132

DOI: 10.1049/el:19991132

P.H.W. Leong and C.K. Chung (Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin NT, Hong Kong)

E-mail: phwl@cse.cuhk.edu.hk

16 July 1999

## References

- 1 WONG, H.Y., YUEN, W.S., LEE, K.H., and LEONG, P.H.W.: 'A runtime reconfigurable implementation of the gsat algorithm'. Proc. Field Programmable Logic and Applications Workshop (FPL'99), Scotland, 1999
- 2 ZHONG, P., MARTONOSI, M., ASHAR, P., and MALIK, S.: 'Accelerating boolean satisfiability with configurable hardware'. Proc. IEEE Symp. Field-Programmable Custom Computing Machines, 1998, pp. 186-195
- 3 SUYAMA, T., YOKOO, M., and SAWADA, H.: 'Solving satisfiability problems using logic synthesis and reconfigurable hardware'. Proc. 31st Annual Hawaii Int. Conf. System Sciences, 1998, pp. 179-186
- 4 ABRAMOVICI, M., SOUSA, J., and SAAB, D.: 'A massively-parallel easily-scalable satisfiability solver using reconfigurable hardware'. Proc. ACM/IEEE Design Automation Conf., 1999, pp. 684-690
- 5 Xilinx Inc., Xilinx data book, 1999
- 6 KELEM, S.: 'Xilinx Virtex configuration architecture advanced user's guide (XAPP151)'. 1999
- 7 Dimacs challenge benchmarks: ftp://dimacs.rutgers.edu/pub/challenge
- 8 SELMAN, B., LEVESQUE, H., and MITCHELL, D.: 'A new method for solving hard satisfiability problems'. Proc. Tenth National Conf. Artificial Intelligence (AAAI-92), San Jose, CA, 1992, pp. 440-446

## Analogue median/average image filter based on cellular neural network paradigm

K. Ślot, J. Kowalski, A. Napieralski and T. Kacprzak

A cellular neural network-based image filter is presented, which allows for median and mean image filtering. The circuit implements an array of  $3 \times 64$  analogue processing elements (cells) and appropriate additional circuitry. Images are loaded into the circuit, are read out of the circuit serially, and are processed in real-time.