5

VLSI Design of Sorting Networks in CMOS Technology

Víctor M. Jiménez-Fernández et al.*
Universidad Veracruzana/Facultad de Instrumentación Electrónica, México

1. Introduction

Although sorting networks have extensively been reported in literature (Batcher, 1962), there are a few references that cover a detailed explanation about their VLSI (Very Large Scale of Integration) realization in CMOS (Complementary Metal-Oxide-Semiconductor) technology (Turan et al., 2003). From an algorithmic point of view, a sorting network is defined as a sequence of compare and interchange operations depending only on the number of elements to be sorted. From a hardware perspective, sorting networks can be visualized as combinatorial circuits where a set of denoted compare-swap (CS) circuits can be connected in accordance to a specific network topology (Knuth, 1997). In this chapter, the design of sorting networks in CMOS technology with applicability to VLSI design is approached at block, transistor, and layout levels. Special attention has been placed to show the hierarchical structure observed in sorting schemes where the so called CS circuit constitutes the fundamental standard cell. The CS circuit is characterized through SPICE simulation making a particular emphasis in the silicon area and delay time parameters. In order to illustrate the inclusion of sorting networks into specific applications, like signal processing and nonlinear function evaluation, two already reported examples of integrated circuit designs are provided (Agustin et al., 2011; Jimenez et al., 2011).

2. Compare-swap block design in CMOS technology

In an algorithmic context, the CS element is conceived as an ideal operator which is free of the inherent delay time presented when a signal propagates through it. It can be seen as a trivial two-input/two-output component with a general two number sorting capability. Also, it is considered that the CS element works taking in two numbers and, simultaneously, placing the minimum of them at the bottom output, and the maximum at the top output by performing a swap, if necessary (Pursley, 2008). Figure 1 shows the typical Knuth diagram for a CS operator. In this pictorial representation, at the input, the horizontal lines describe

* Ana D. Martínez[1], Joel Ramírez[1], Jesús S. Orea[1], Omar Alba[2], Pedro Julián[3], Juan A. Rodríguez[3], Osvaldo Agamennoni[3] and Omar D. Lifschitz[3]
[1]*Universidad Veracruzana/Facultad de Instrumentación Electrónica, México*
[2]*Instituto Tecnológico Superior de Xalapa/ Departamento de Electrónica, México*
[3]*Universidad Nacional del Sur/ Departamento de Ingeniería Eléctrica y de Computadoras, Argentina*

the two numbers to be sorted (A and B) and, at the output, max(A,B) and min(A,B) denote the maximum and minimum numbers, respectively. In turn, the vertical connector line represents the element dedicated to compare and interchange (swap) data.

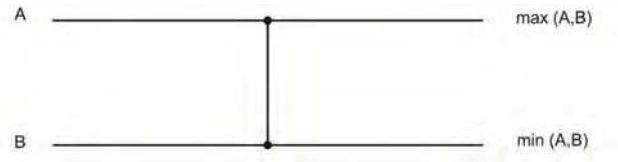


Fig. 1. Knuth diagram for a compare-swap element

However, this is only a theoretical viewpoint, because when the CS element is carried out to a level of silicon realization it is affected by parasitic elements, presenting a different time delay for each output. Due to the fact that in a sorting network the main structural element is the CS circuit, a special attention is given to describe in detail its internal design. In this section, the CS circuit design is covered at schematic transistor and at layout levels; furthermore, the area and delay time are estimated by considering a given 0.5 microns process technology.

2.1 Design at transistor level for a compare-swap standard cell

The CS element is a combinatorial circuit that accepts as input two binary signals (numbers), compares their magnitude, and outputs the maximum in the max(A,B) bus line, whereas the minimum is output in the min(A,B) bus line. This block is integrated by one full-adder and two multiplexers, as shown in Fig. 2.

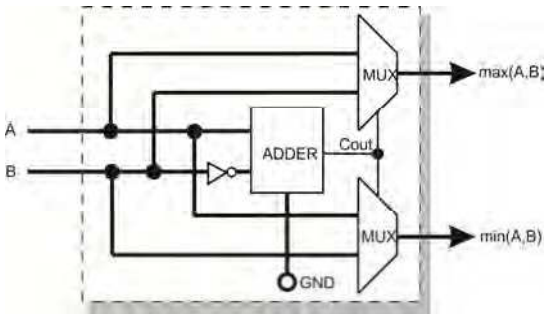


Fig. 2. Block level diagram for the CS circuit

Notice that due to one input is complemented, the full-adder is in fact configured as a subtractor. The most significant bit resulting from the subtraction, carry out (C_{out}), is used to make the selection in the multiplexer. If a greater number is subtracted from a lesser one, then the result is a negative number what, in binary terms, can be identified because the generated C_{out} will be in high ("1" logic). When the C_{out} signal is in high state, a swap data will be performed; otherwise, the input data will not be interchanged.

For translating the diagram of the CS block, in Fig. 2, to a transistor level circuit description, the two well known standard cells for the full-adder and for the multiplexer (Kang &

Leblebici, 2003), are used. Figure 3, shows the transistor level schematic for the one-bit full-adder and for the one-bit multiplexer. The full-adder is composed by 24 MOS transistors, topologically connected in a CMOS configuration, where 12 PMOS transistors (M1…M12) belong to the pull up network and 12 NMOS transistors (M13…M24) are associated to the pull down network. The multiplexor is integrated by two transmission gates (composed by transistors M25-M28) and one NOT-gate with two inputs (In0, In1), one selector (Sw), and one output (Out). The multiplexer output depends only of the C_{out} of the full-adder, since C_{out} is assigned to the selector that will activate one pair of transistors. If the selector is low ("0" logic), the transmission gate at the top (integrated by M25 and M26) switches ON, so In0 becomes the output; otherwise the transmission gate at the bottom (composed by M27 and M28) is ON, hence In1 is the output.

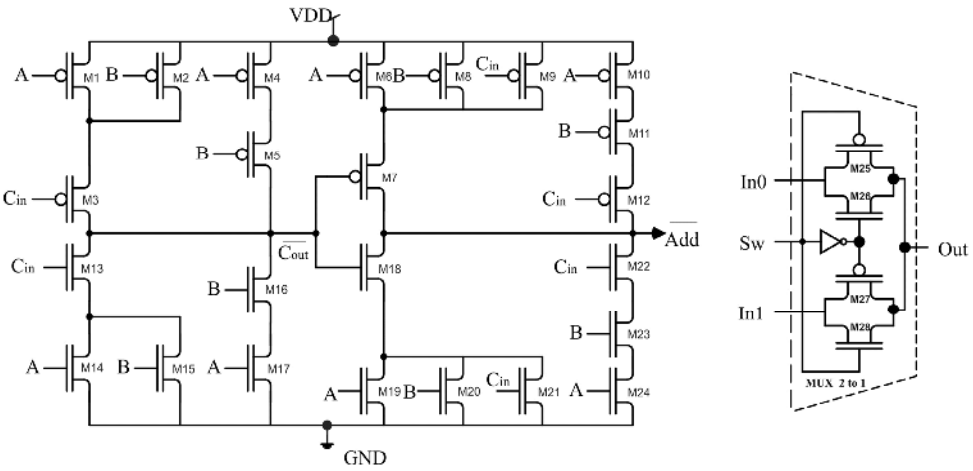


Fig. 3. Transistor level schematic of the one-bit full-adder (on left) and one-bit multiplexer (on right)

2.2 Design at layout level for a compare-swap standard cell

Masks of the CMOS full-adder and multiplexer circuits using minimum size transistor are depicted in Fig. 4 and Fig. 5. It is important to point out that the (W/L) ratio for all the NMOS and PMOS transistors in this layout were computed to optimize the transient performance of the circuit, specifically a balance between the high-to-low and low-to-high propagation times. For PMOS transistors a (10λ/2λ) ratio was considered while a (6λ/2λ) ratio was used to NMOS transistors. Since a 0.5 microns process technology is included, the physical dimension of lambda for this design technology is λ=0.35 microns. The well-known layout style based on a "line of diffusion" rule that is commonly used for standard cells in automated layout systems (Weste & Eshraghian, 1993) is employed in this layout. In this style, four horizontal strips can be identified: a metal ground at the bottom of the cell (GND or VSS), n-diffusion for all the NMOS transistors, a n-well with a corresponding p-diffusion for all the PMOS transistors, and a metal power at the top (VDD). A set of vertical lines of poly-silicon are also used to connect the transistor gates while within the cell metal layers connect the transistors in accordance with a schematic diagram.

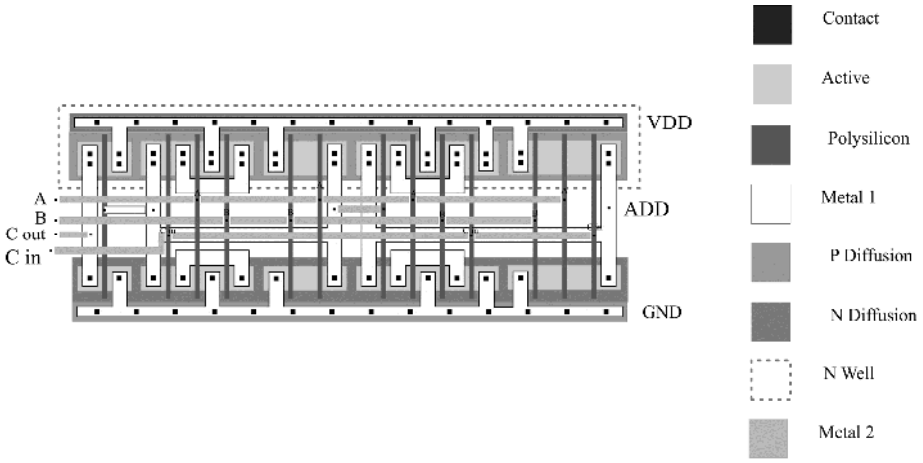


Fig. 4. Mask layout of the CMOS one-bit full-adder circuit

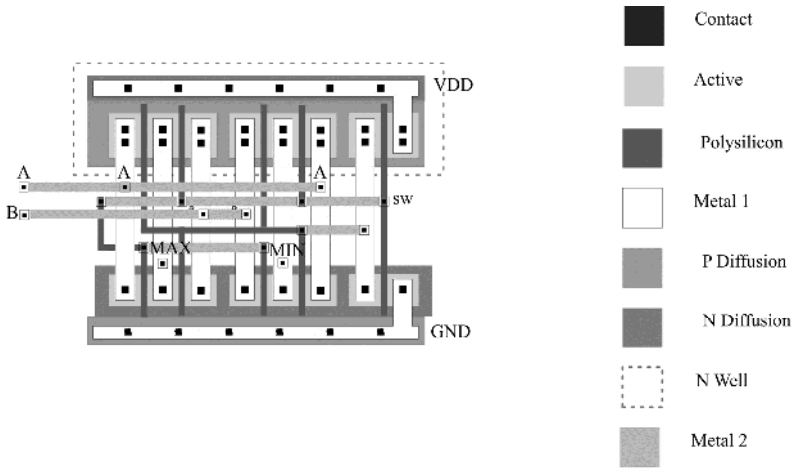


Fig. 5. Mask layout of the one-bit multiplexer circuit

2.3 Delay time and area estimations for the compare-swap cell

In order to have a reference of the switching speed for the one-bit CS circuit, an empirical delay time estimation supported by SPICE simulations is performed. Due to the speed in a CMOS gate is limited by the time taken to charge load capacitances toward VDD and discharge toward GND (Rabaey, 2003), the parasitic capacitances induced by the layout structure are considered. In this sense, a parasitic extractor software (e.g., L-Edit extractor of Tanner EDA) can be used to obtain a circuit netlist file in which all these elements be incorporated. By using SPICE simulation and including the proper test-data fabrication model parameters (AMIS 0.5 microns), an accurate transient response is achieved. The resulting transient responses are analyzed to estimate the switching speed through the delay

time (difference between input transition at 50% and the 50% output level). The simulated output voltage obtained for the one-bit CS circuit is shown in Fig. 6. In this simulation, the voltage supply of 5V (VDD) and the overall frequency of 5MHz are considered. Also, the simplest representation 0 or 1 will be hereafter used instead of the "1" logic or the "0" logic notations. After running the SPICE simulation, it can be observed the outputs MAX(A,B)={0,1,1,1} and MIN(A,B)={0,0,0,1} when the inputs A and B are given by A={0,0,1,1} and B={0,1,0,1}. It is important to notice that the signal CARRY_OUT (C_{out}) is only in high when A=0 and B=1 (the unique case where a swap is needed).

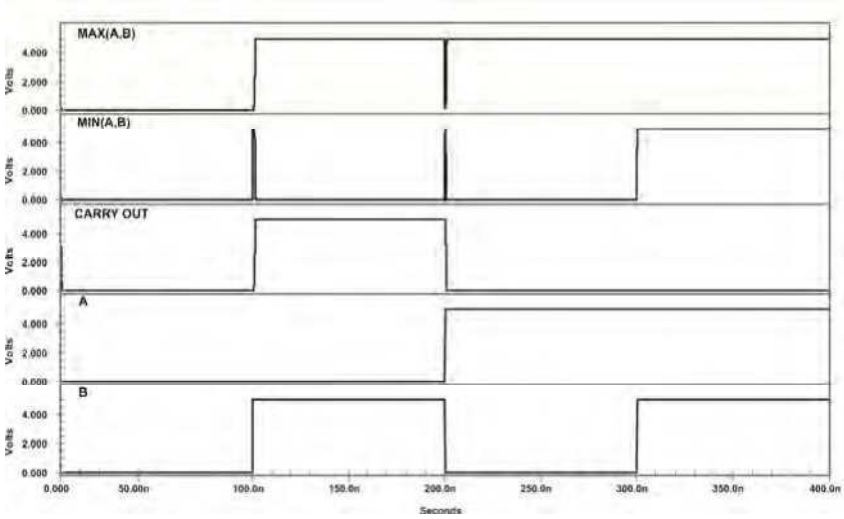


Fig. 6. Simulated output voltage obtained for the one-bit CS circuit

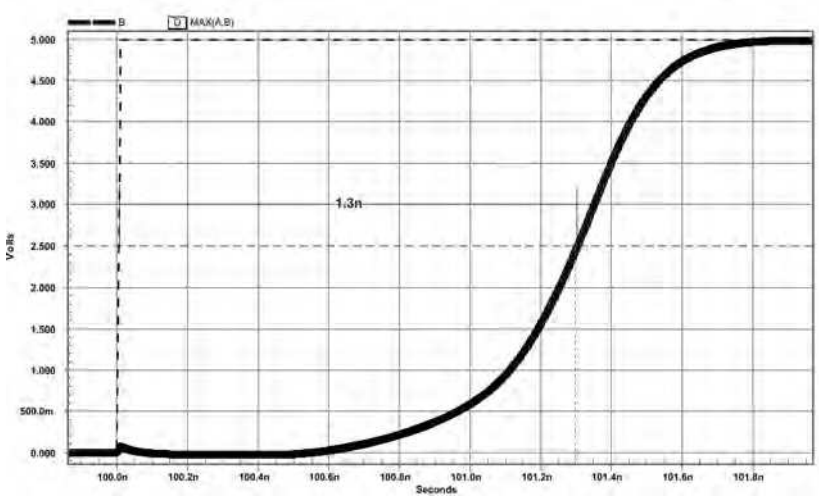


Fig. 7. Worst-case delay time for the one-bit CS circuit

As it was expected, the worst-case of delay time is presented in the swapping case. However, not only the delay time depends on the C_{out} propagation, but also it is related to the delay time added by the transmission gate. In accordance with simulation, a delay of 1.3 ns is exhibited. In Fig. 7 this delay time is showed, the dashed line indicates the input B=1 when A=0 and the solid line represents the propagated B datum after the swap operation.

An accurate silicon area estimation of the CS design can be computed directly from the layout editor by using a ruler tool (usually provided in this software). Figure 8 shows the CS cell layout design that highlights the length and width dimensions expressed in terms of lambda. From this figure the area estimation is given by 30830 λ^2 = 0.0037767 mm^2.

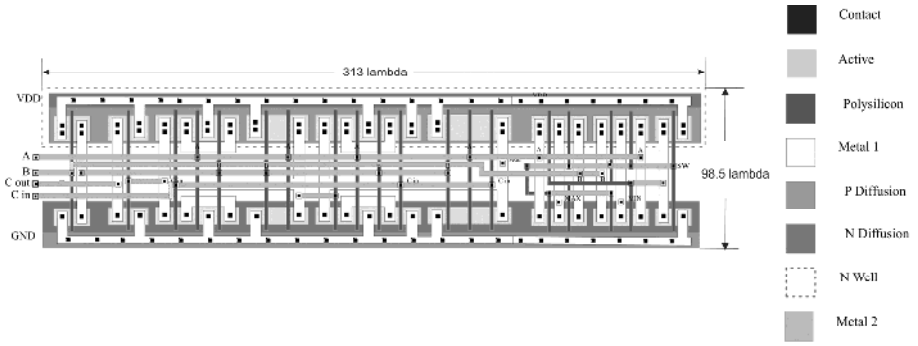


Fig. 8. Silicon area estimation for the one-bit CS layout

2.4 The n-bits compare-swap cell

The one-bit CS circuit in Fig.2 can be easily expanded into an n-bits structure. In order to illustrate how this expansion can be performed, the schematic diagram for a 4-bit CS circuit is shown in Fig. 9. Because of the overall speed of the CS circuit is limited by the delay propagation of the C_{out} bits through the *n*-bits chain, therefore an estimation of this time becomes essential for determining the speed performance. However, besides to the delay produced due to the critical path of C_{out}, the delay time added by the multiplexer block is also taken into account.

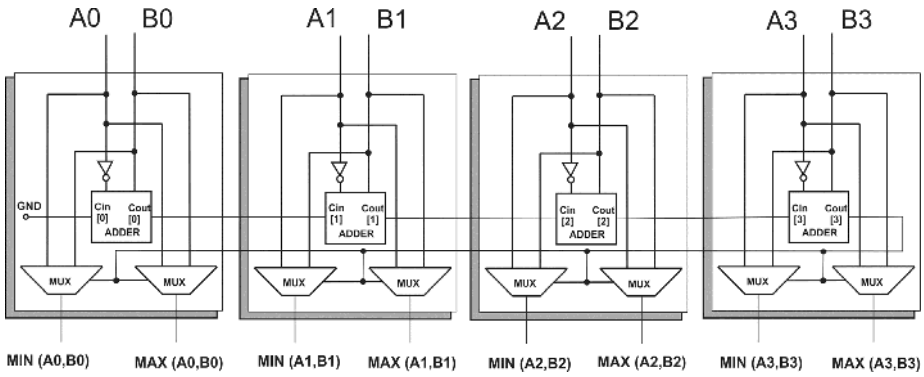


Fig. 9. Block diagram for the 4-bit CS circuit

In Fig. 10, the simulated output voltage obtained for the 4-bit CS circuit is shown when inputs A and B are given by: [A3:A0]={ 0101 (5_{10}), 1001 (9_{10}), 0011 (3_{10}), 0100 (4_{10}) } and [B3:B0]={ 1010 (10_{10}), 0110 (6_{10}), 1100 (12_{10}), 0100 (4_{10}) }.

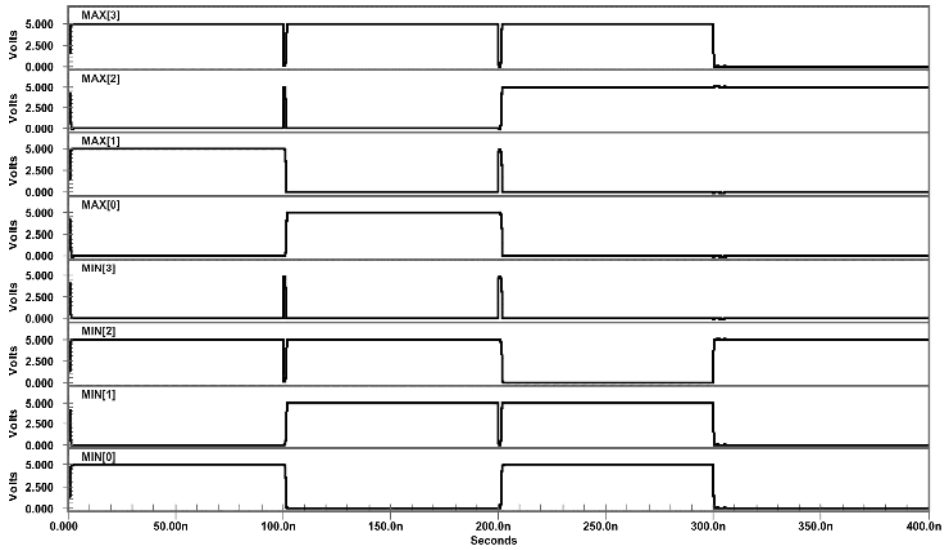


Fig. 10. Simulated output waveforms of the 4-bit CS circuit

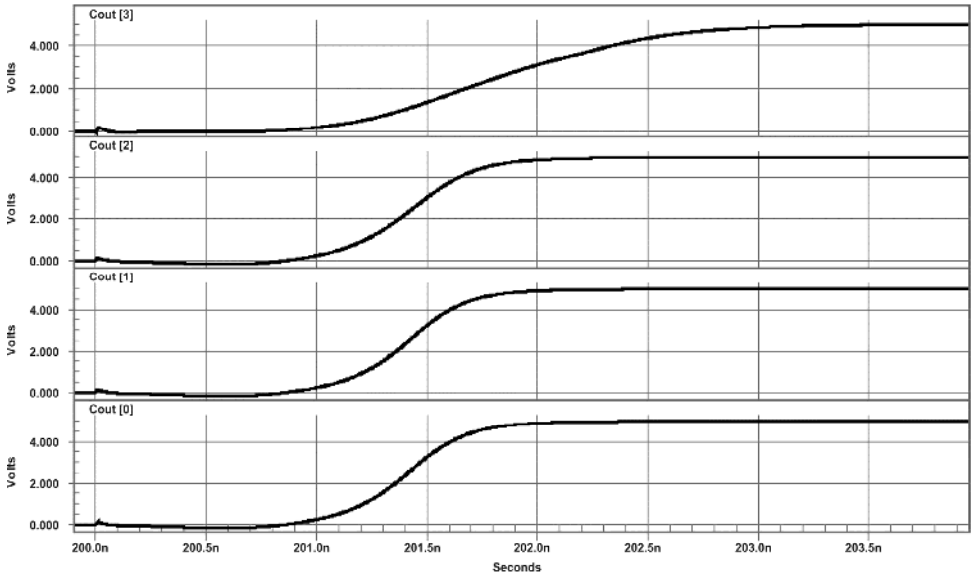


Fig. 11. Carry out (C_{out}) propagation through the 4-bit CS circuit for the C_{out}=[1,1,1,1] case

Figure 11 depicts the C_{out} propagation while Fig.12 indicates the delay time between a signal and its corresponding output after that a swap operation is performed. In these simulations, the delay time was also examined at the overall frequency of 5MHz, VDD=5V, and by considering the worst-case of C_{out} propagation. This case occurs when [A3:A0]=[0000] and [B3:B0]=[1111] what ensures transferring C_{out}= 1 at every one-bit CS basic cell.

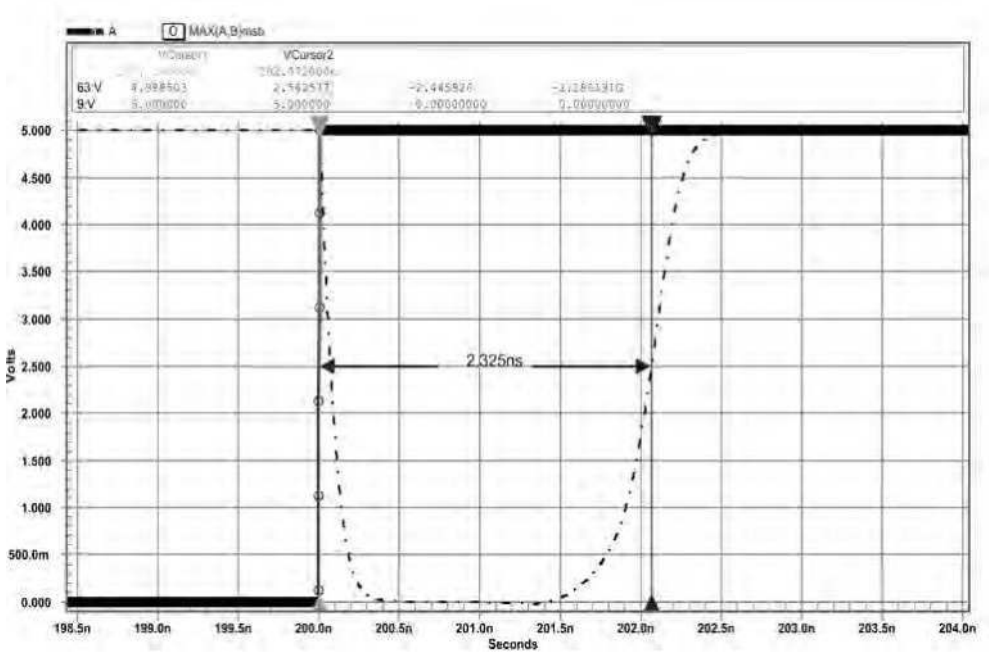


Fig. 12. Delay time between B[0] and MAX[A3,B3]

3. Median filtering for image denoising using sorting network

In order to illustrate the application of the CS circuit to the CMOS design, a digital architecture which is dedicated to median filtering for image denoising, is taken as a reference. This kind of filtering technique is used to reduce impulsive noise in acquired images (Faundez, 2001). Its main advantage consists in diminishing the lossless of information due to the computed pixel values have correspondence to one of the already presented in the image and its main characteristic is the requirement of a sorting operation (Vega et al., 2002).

Before of describing this design, it is important to present a briefly explanation about the algorithm which serves as basis for its digital architecture. The following notational conventions will be used: if $\boldsymbol{I}(x,y)$ is a grayscale image divided in $(m\times n)$ pixels (squares) and also $\boldsymbol{I}(x,y)$ is affected by impulsive noise, then by applying a median filter algorithm, a denoised image $\boldsymbol{IF}$(x,y) can be obtained. In order to achieve $\boldsymbol{IF}(x,y)$, the value of each output pixel must be computed by using iteratively a (3×3) square array (mask) of 9 pixels with center in $\boldsymbol{I}(x,y)$. The position of this mask is shifted along to $\boldsymbol{I}(x,y)$ until the median filtering

process is completed. It is worth to mention that because of the mask operates over the neighbour pixels, then it is needed to add elements (for example zeros) around $\boldsymbol{I}(x,y)$, increasing its dimension as $(m+2)\times(n+2)$. At each one of these pixels, a sorting procedure is performed by following three basic steps into the (3×3) mask: firstly, the pixels of the mask are sorted in a column by column sequence, then row by row, and finally along to the diagonal elements. After the sorting task is achieved, the central element (median) of the mask is picked out of $\boldsymbol{I}(x,y)$ and stored in the $\boldsymbol{IF}(x,y)$ to construct the filtered image. An illustrative description for this median algorithm is depicted in Fig. 13. A more formal description of this algorithm can be found in reference (Jimenez et al., 2011).

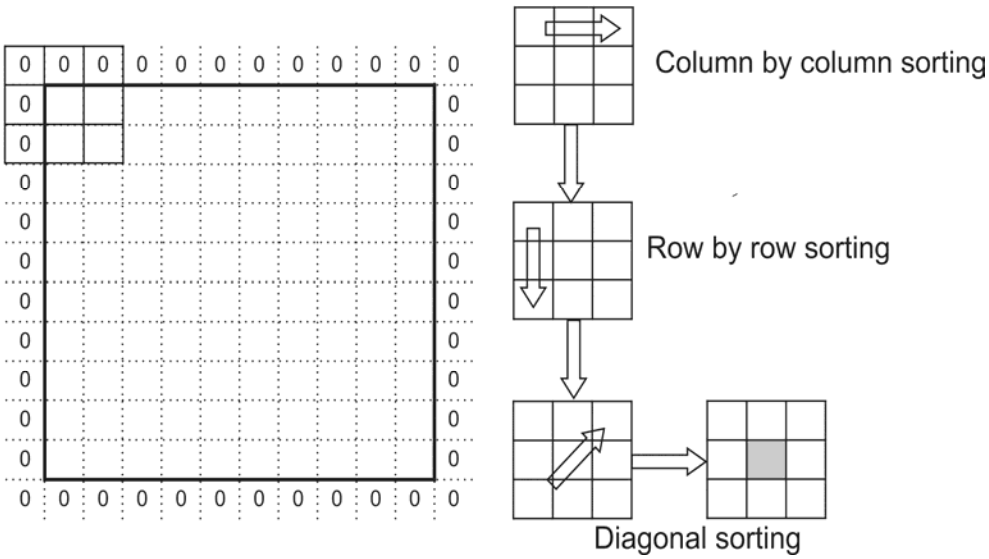


Fig. 13. Graphical description for the median filter algorithm

3.1 The sorting network block in the median filter algorithm

A Knuth diagram for the sorting network procedure which is described in the median filter algorithm is shown in Fig. 14.

Notice that the above sorting network exhibits a very regular structure that is hierarchically partitioned in seven blocks of three-data for median computing. The first stage of three blocks is dedicated to the column by column sorting, the second stage of three blocks is devoted for the row by row sorting, and finally the last block performs the diagonal sorting. It can be also observed that after all data have been propagated through the entire network, the median datum will be appearing in the bus line D4. If the (3×3) mask is defined as follows:

$$\mathrm{MASK}_{I(x,y)} = \begin{vmatrix} D0 & D3 & D6 \\ D1 & D4 & D7 \\ D2 & D5 & D8 \end{vmatrix} \tag{1}$$

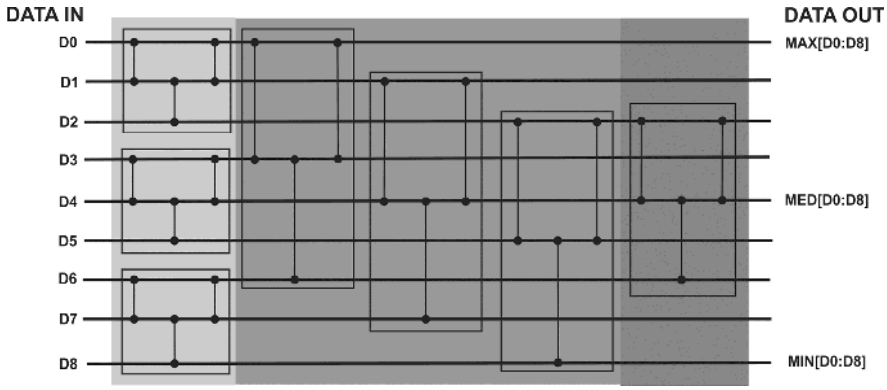


Fig. 14. Knuth diagram for the sorting network included in the median filter algorithm

Then the median datum (collected in D4) is computed trough the next steps:

1. Column by column sorting:

$$SORT\ (D0, D1, D2)$$

$$SORT\ (D3, D4, D5)$$

$$SORT\ (D6, D7, D8)$$

2. Row by Row sorting:

$$SORT\ (D0, D3, D6)$$

$$SORT\ (D1, D4, D7)$$

$$SORT\ (D2, D5, D8)$$

3. Diagonal sorting:

$$SORT\ (D2, D4, D6)$$

3.2 Digital architecture for the image filtering based on sorting network

In reference (Jimenez et al., 2011) a FPGA (Field Programmable Gate Array) implementation for median filtering image based on a sorting algorithm is reported. In such architecture two blocks can be distinguished: a nine-data accumulator and a nine-data sorting network module. The accumulator is a memory register in which the data is received from the (3x3) mask and temporarily stored. The sorting network, which is in fact the kernel of the median filter architecture, is also a nine-inputs/one-output combinational module. It is constituted by an array of seven blocks of three-data comparator modules as corresponds to Fig. 14. This interconnection topology is directly related to the median algorithm because it operates by following the already described three steps: column sorting, row sorting and diagonal sorting. It can be seen that although this block is able to output the nine data in a sorted sequence, only the datum in D4 is collected since it represents the median.

In order to illustrate the correct performance of this architecture, results obtained from the FPGA implementation and from the coded algorithm in Matlab are compared. Figure 15 shows a group of images that have been intentionally corrupted by impulsive noise and then filtered directly by Matlab software and by FPGA hardware.

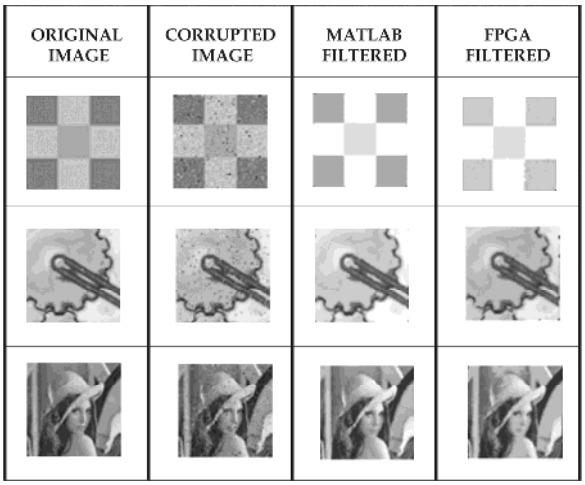


Fig. 15. Collection of images filtered by software (Matlab) and by the FPGA device

3.3 Floor planning and design at layout level

The main structural component in the sorting network which is exposed in section 3.1, is a three-data comparator. As shown in Fig. 14, this element can be constituted by a set of interconnected one-bit CS cells. Three 8-bit word-length inputs described as: A, B and C can be identified. Also, three 8-bit CS blocks make possible to collect the median datum in the middle bus denoted by MED(A,B,C), and the corresponding minimum and maximum data into the external buses described as MIN(A,B,C) and MAX(A,B,C). In order to minimize the layout area, the CS modules have been rotated and placed in the position as illustrates the floorplanning and layout of Fig. 16.

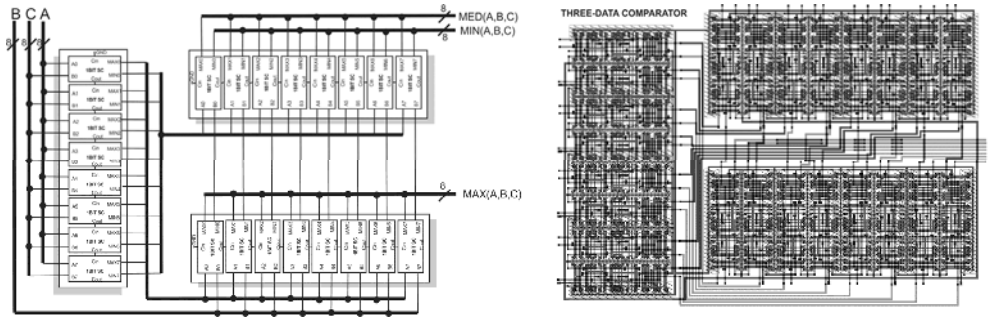


Fig. 16. Floorplanning (on left) and layout (on right) for the 8-bit three-data comparator included in the median filter

A graphical description, about the size and placement of the three-data modules that constitutes the nine-data sorting network is presented in Fig. 17. This floorplanning shows the connectivity between every module without showing internal layout details. It can be observed that some modules should be flipped to improve the routing and also achieving an area minimization. In this layout, two blocks can be recognized: the nine-data serial-in/parallel-out register and the nine-data sorting network. In accordance to the proposed median filter algorithm the unique signal of interest from the sorting network output is the median datum (D4) while the other data (D1,D2,D3,D5,D6,D7, and D8) are discarded.

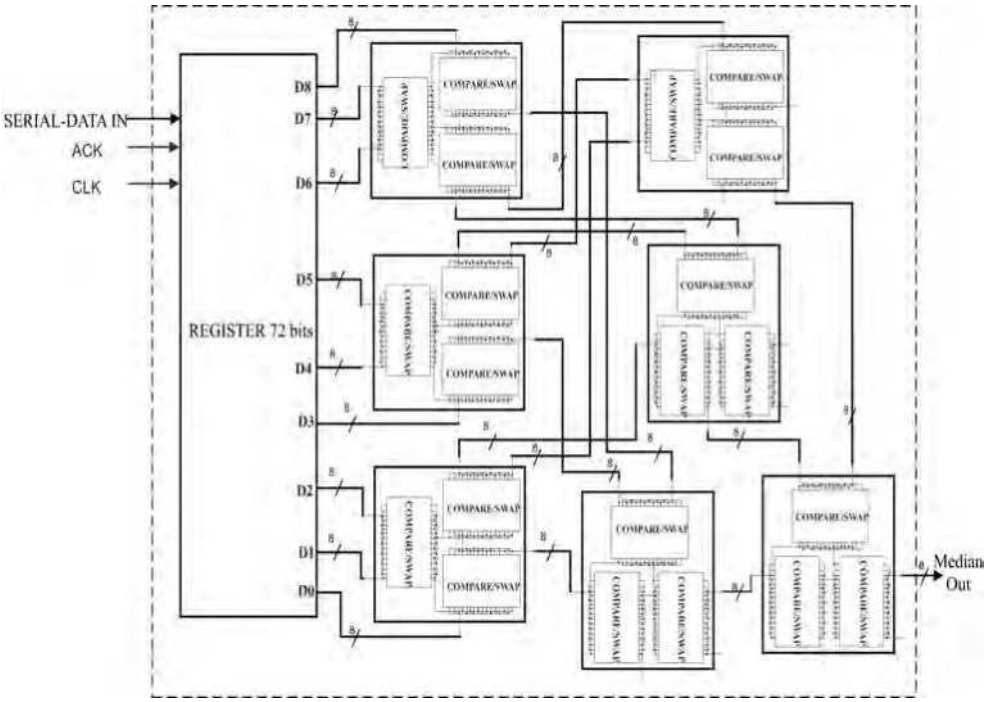


Fig. 17. Floorplanning for the nine-data sorting network in the median filter design

Figure 18 shows the translation to a layout level of the floorplanning design of fig. 17.

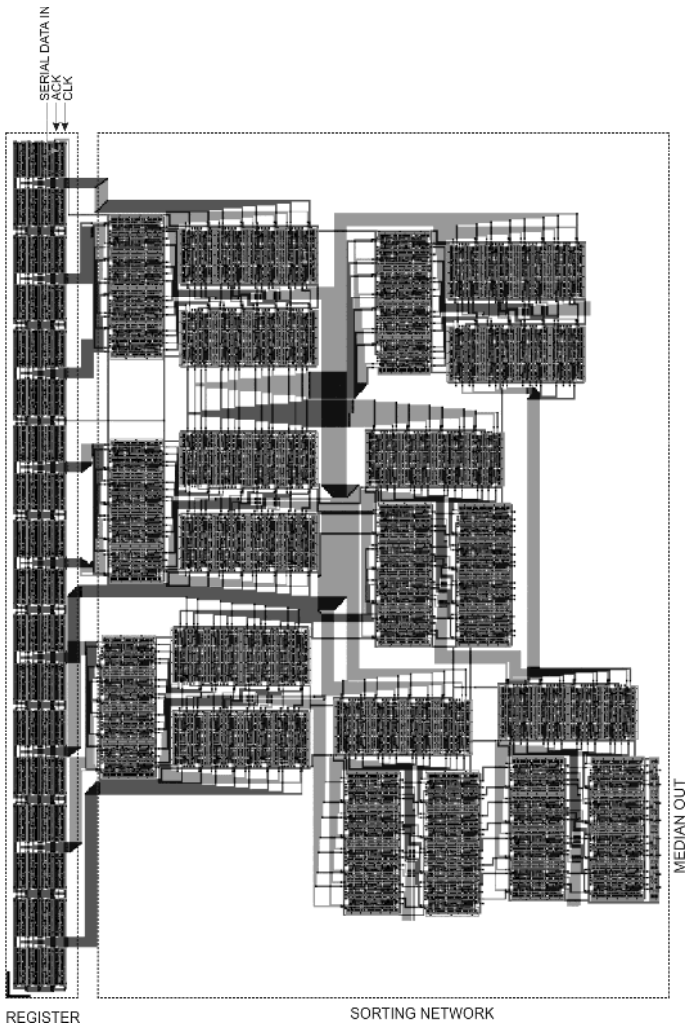


Fig. 18. Layout for the nine-data sorting network in the median filter design

3.4 Spice simulations for delay time estimation and electrical verification

The netlist including parasitic capacitances from the layout of Fig.18 is simulated to verify the circuit operation. The voltage waveforms for the median datum are shown in Fig. 19. The signals S0, S1, ..., and S7, correspond to the 8-bits median datum of the set of inputs {D0, D1, ..., D9} given by: D1=00101011 (212_{10}), D2=01000011(194_{10}), D3=00100010 (69_{10}), D4=00001101 (176_{10}), D5=00011100 (56_{10}), D6=00011110 (120_{10}), D7=00010111 (232_{10}), D8=00001111(240_{10}), and D9=00010101 (168_{10}).

In this simulation, the median for the input described in base-10 { 212_{10}, 194_{10}, 69_{10}, 176_{10}, 56_{10}, 120_{10}, 232_{10}, 240_{10}, 168_{10} } is the datum 176_{10} ,which in a binary base is represented as

D4=0000110110. It is important to clarify that the output median datum will be enabled until the acknowledge signal (ACK) is high and the CLK low.

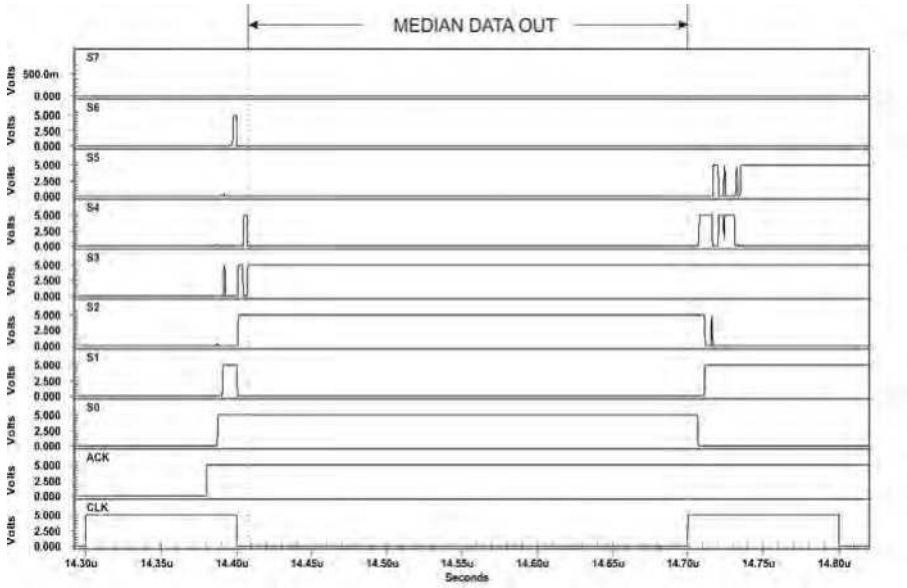


Fig. 19. Simulated output waveforms of the nine data sorting network

4. Piecewise linear function computation using sorting network

As a second example of the application of sorting networks in dedicated VLSI systems, an Application Specific Instruction Processor (ASIP) for piecewise linear function evaluation is described. Piecewise linear (PWL) functions allow the aproximation of multidimensional nonlinear functions or models in a convenient way to be evaluated with computing systems (Julian et al., 1999). The simplicity of the representation and evaluation methods, combined with the scalability in terms of the number of dimensions, impulsed the adoption of PWL functions as the modelling abstraction in a broad spectrum of systems. The first and most traditional area has been the computationally efficient resolution of nonlinear circuits (Chua & Ying, 1983) that require high performace function evaluation in terms of speed, and more recently in communication systems (Kaddoum et al., 2007) and power electronics (Pejovic & Maksimovic, 1995) for nonlinear operations involved in predistortion (Hammi et al., 2005). Motivated by these applications, the ASIP for piecewise linear function evaluation, hereafter denoted as PWLR6-µp, was designed and implemented in CMOS technology to provide a flexible environment for computation of 6-dimentional PWL functions and, due to the fact that PWL evaluation algorithm requires a sorting procedure, sorting networks have been embedded in this design as it will be expossed in this section.

4.1 The sorting networks for nonlinear function evaluation

Given an n-dimensional domain divided into a simplicial partition with a regular grid, a n-dimensional PWL function can be expressed as the weighted sum:

$$F(X) = \sum_{i=0}^{n} \mu_i c_i \tag{2}$$

where $\boldsymbol{X}$= $\{x_1 ... \ x_n\}$ is the point in the n-dimensional domain where the function is evaluated, μ_i are scaling parameters depending on $\boldsymbol{X}$, and c_i are the values of the function at simplex vertices, i=0,...,n. In order to compute the PWL equation c_i and μ_i are required. Usually the c_i parameters are stored in a RAM while μ_i need to be computed. For this operation, the algorihmic procedure defined in (Agustin et al., 2011) is followed what involves the descomposition of the x_i componentens into integer and fractional parts, sorting of the fractional parts and performing a sucesive subtraction of the sorted fractional parts.

4.2 Micro-architecture for the PWL evaluation through a sorting scheme

Micro-architecture of an ASIP strongly depends on its target application. A successful ASIP provides the required hardware to solve the target set of computational problems in an optimized way in terms of execution time, power, or chip resources, while maintaining the flexibility and programmability characteristics of a general purpose microprocessor. A trade-off exists among optimization levels and flexibility levels; thus, an ASIP can be considered as an intermediate point between a general purpose microprocessor and an Application Specific Integrated Circuit (ASIC). Three main architectural blocks of the PWLR6-µP, namely, Data Path, I/O, and Control, were designed taking into account the special operations required to perform the PWL calculation. The result was a nearly basic microprocessor with special features that accelerate the PWL computation. In this seccion, the sorting step of the algorithm and its relationship with the resources provieded by the PWLR6-up is addressed. For further details about the rest of the architecture and its special resources for PWL function evaluation, the reader is referred to (Agustin et al., 2011). Sorting constitutes the second part of the PWL function evaluation algorithm (as it was mentioned, it is required to evaluate the μ_i parameters). In this implementation, the 6-fractional parts are sorted following the comparison sequence of the so called Bose–Nelson sorting network (Knuth, 1997). However, in order to maintain the microprocessor structure and to avoid the area overhead of 12 CS blocks, only one comparator (the one provided by the ALU) was used. Consequently, the Bose-Nelson sorting network is embedded in the PWLR6-µp by combining an apropiate harware-software design.

The hardware resouces provided by the PWLR6-µp architecture are showed in Fig. 20 and they are used during the sorting step as follows: firstly the RF (register file), which is composed by six registers, stores the fractional parts to be sorted. Then, the two bidirectional ports, Port A, which is connected to Register A, and port B connected to Register B, transfer data between them and RF. After that, compare operation is performed from these registers and depending on the result, Register A and Register B values may be written back into the RF by switching sources and destinations.

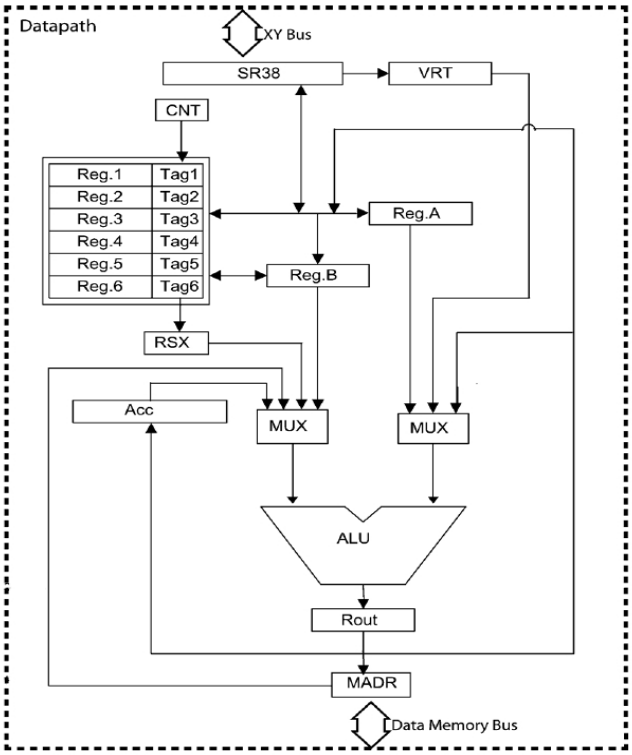


Fig. 20. Architecture for the six-data sorting network embedded in the PWL microprocessor

5. Conclusion

In this chapter, sorting networks have been addressed since a physical CMOS realization perspective with applicability to VLSI design. The CS circuit, analyzed at the beginning of this chapter, was introduced as the fundamental cell from which more complex sorting topologies could emerge. It must be pointed that because the speed in the CS design is limited by the delay of the *n*-bits carry out critical path, and by the transmission gates delay, a future research proposal for this work must be aimed to achieve higher overall frequencies. The two provided examples: the median filter architecture and the PWL evaluation scheme, allow to show the inclusion of sorting networks, into these specific applications. In these sense, about these examples the following particular conclusions must be observed: firstly, in the sorting network inmerse in the median filter, the main advantage consists in its regular structure beacuse although it is not optimal in the number of comparisons (21), the execution of several CS elements is done in parallel, and finally, the choice of an embeded sorting strategy in the PWL ASIP was due to the simplicity that allows the PWLR6-μP architecture (compared to other sorting algorithms like bubble sort or quick sort, designed to sort bigger datasets) and because of the small size of the input, this strategy is efficient in terms of hardware resources and code length.

6. Acknowledgment

This work has been partially supported by Universidad Veracruzana and by the CB-SEP-CONACyT Project No.102669 of Instituto Tecnológico Superior de Xalapa, México. Some partial research results from the PROMEP/103.5/09/4482 project of Universidad Veracruzana, Mexico, and PICT 2003 No. 13468 of Universidad Nacional del Sur, Argentina, have been referred in this chapter.

7. References

Agustin Rodriguez J., Lifschitz Omar D., Jimenez-Fernandez Victor M., Julian Pedro. (2011). "Application Specific Processor for Piecewise Linear Functions Computation". *IEEE Transactions on circuits and systems*, Vol. 58, No. 5, May 2011, pp. 971-981, 1459-8328

Batcher K. E.. (1962). "Sorting networks and their applications", *Proceedings of AFIPS Spring Joint Computer Conference,* ISBN: n.d., April 1962

Chua L. & R. Ying. (1983). "Canonical piecewise-linear analysis," *IEEE Transactions on Circuits Systems*, Vol. 30, No. 3, Mar 1983, pp. 125–140, 0098-4094

Faundez Zanuy M. (2001). "*Digital voice and image processing with multimedia application*", First Edition, Alfaomega-Marcombo, 97-88-42671244-8, Barcelona, Spain

Hammi O., S. Boumaiza, M. Jaidane-Saidane, and F. Ghannouchi. (2005). "Digital subband filtering predistorter architecture for wireless transmitters," *IEEE Transactions Microwave Theory Tech.*, Vol. 53, No. 5, May 2005 , pp. 1643–1652, 0018-9480

Jimenez F. Victor, Martinez-Navarrete Denisse, Ventura-Arizmendi Carlos, Hernandez-Paxtian Zulma, Ramirez-Rodriguez Joel. (2011) *International Journal of Circuits, Systems and Signal Processing,* Vol. 5, No. 3, Apr. 2011, pp. 297-304, 1998-4464

Julian P., Desages A., and Agamennoni O.,(1999), "High-level canonical piecewise linear representation using a simplicial partition," *IEEE Transactions on Circuits and Systems-I, Fundam. Theory Appl.*, Vol. 46, No. 4, Apr. 1999 , pp. 463–480, 1057-7122

Kaddoum, G.; Roviras, D.; Charge, P.; Fournier-Prunaret, D. (2007). "Analytical calculation of BER in communication systems using a piecewise linear chaotic map". *Circuit Theory and Design, 2007. ECCTD 2007. 18th European Conference on* . 978-1-4244-1341-6 ,Toulouse, France, Aug. 2007

Knuth E. Donald. (1997)."*The Art of Computer Programming,*" Third Edition. Addison-Wesley, 1997. 0-201-89685-0, Massachussets, USA

Kang S. M. and Yusuf Leblebici, (2003). "*CMOS Digital Integrated Circuits,*" Third Edition, Mc Graw Hill, 0-07-246053-9, New York, N.Y.

Pejovic, P. Maksimovic, D. (2002). A new algorithm for simulation of power electronic systems using piecewise-linear device models, *IEEE Transactions on Power Electronics,* Vol.10 , No. 3, May 1995, pp. 340-348, 0885-8993

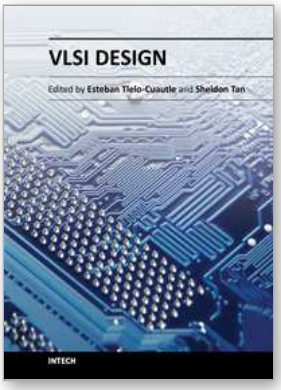
Pursley Bryan, (2008), Sorting Networks, *n.d,* Aug 09,2011, URL: http:// brianpursley.com/Files/CSC204_FinalProject_BrianPursley.pdf

Rabaey J. M., Chandrakasan A. P., and Nikolic B. (2003). "*Digital Integrated Circuits: A Design Perspective*", Second Edition., Prentice Hall Electronics and VLSI Series, Upper Saddle River, NJ: Prentice Hall/Pearson Education, 0-13-0909960-3,n.d.

Turan Demirci, Ilhan Hatirnaz, Yusuf Leblebici. (2003)."Full-custom CMOS realization of a high-performance binary sorting engine with linear area-time complexity". *"In Proceedings of ISCAS (5)'2003"*. ISBN: 0-7803-7761-3, Bangkok, Thailand, May 2003

Vega M., Sanchez J., Gomez J. (2002). "An FPGA-based implementation for median filter meeting the real-time requierements of automated visual inspection systems", *Proceedings of the 10th Mediterranean Conference on Control and Automation-MED2002*, 972-9027-03-X, Lisbon, Portugal, July 2002.

Weste N. H.E., and Eshraghian K., (1993),*"Principles of CMOS VLSI desing"*, Second Edition, Addison-Wesley, 0-201-53376-6



VLSI Design

Edited by Dr. Esteban Tlelo-Cuautle

ISBN 978-953-307-884-7
Hard cover, 290 pages
Publisher InTech
Published online 20, January, 2012
Published in print edition January, 2012

This book provides some recent advances in design nanometer VLSI chips. The selected topics try to present some open problems and challenges with important topics ranging from design tools, new post-silicon devices, GPU-based parallel computing, emerging 3D integration, and antenna design. The book consists of two parts, with chapters such as: VLSI design for multi-sensor smart systems on a chip, Three-dimensional integrated circuits design for thousand-core processors, Parallel symbolic analysis of large analog circuits on GPU platforms, Algorithms for CAD tools VLSI design, A multilevel memetic algorithm for large SAT-encoded problems, etc.

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