SORTING NETWORKS IMPLEMENTED AS νMOS CIRCUITS[1]

Esther Rodriguez, Jose M. Quintana, Maria J. Avedillo and Adoracion Rueda

Instituto de Microelectrónica de Sevilla, IMSE-CNM, Universidad de Sevilla.

Edif. CICA, Avda. Reina Mercedes s/n, 41012 Sevilla, SPAIN.

[1]This effort was partially supported by the spanish CICYT under Projects TIC95-0094 and TIC97-0648.

Indexing Terms: Digital design, Sorting.

***Abstract**:*

This letter proposes a new realization for *n*-input sorters. Resorting to the neuron-MOS (νMOS) concept and to an adequate electrical scheme, a compact and efficient implementation is obtained

Introduction:

This letter presents a new hardware realization for the problem of building binary sorting networks (SN). An *n*-input SN is a switching network with *n* outputs that generates an output which is a sorted (non increasing order) permutation of inputs. Binary SN's are built from comparator cells, which have two inputs and two outputs: one of them provides the maximum of both inputs and the other the minimum. The internal structure of the comparator depends on the application. Inputs can be binary numbers and the comparator is a complex element, or binary signals and then, maximum and minimum become OR and AND operations respectively. Figure 1a shows the behavior of a *n*-input sorter with *k* inputs equal to 1's, and Figure 1b the logic implementation of the comparator cell. A lot of attention has been for many years devoted to the problem of efficient SN design [1], being a milestone the constructive method proposed by Batcher [2]. Figure 1c shows a 4-input SN implemented following Batcher's method.

In this letter, a different approach to implement a sorter is presented, based on the fact that each output of an *n*-input sorter depends only on the number of inputs equal to 1. The output *i*-

th will be 1 if and only if at least *i* of the *n* inputs are 1. So, an *n*-input sorter can be seen as a cascaded two-block circuit. The first block provides an output which depends linearly on the number of 1's in the applied inputs. The second block takes this output signal and compares it with a set of *n* fixed values by means of a battery of comparators, thus providing the set of *n* output functions of an *n*-input sorter.

Recently high-functional νMOS transistors have been developed which can perform weighted summation of multiple input signals at the gate levels [3]. νMOS transistors have a buried floating polysilicon gate and a number of input polysilicon gates that couple capacitively to the floating gate. The voltage of the floating gate becomes a weighted sum of the voltages in the input gates, and hence, it is this sum which controls the current in the transistor channel. A schematic of this transistor is shown in Fig. 2. There is a floating gate and a number of input gates $x_1, x_2, \ldots, x_n$. Weights for every input are proportional to the ratio of the corresponding input capacitance, C_i, between the floating gate and each of the input gates, to the total capacitance, including the transistor channel capacitance, C_{chan}, between the floating gate and the substrate.

An 8-input sorter has been designed exploiting the above two-block approach and resorting to the νMOS transistor principle for its implementation, as it requires counting the number of 1's in the inputs, or equivalently performing an arithmetic addition of the inputs. The compact architecture and the efficient physical implementation we propose compare very favorably with the traditional solution.

Electrical realization of a sorter circuit

Figure 3 shows the two-stage schematic diagram of the proposed *n*-input sorter. The implementation of the first block resorts to the νMOS principle and to current mirroring to provide an analog output voltage, V_1, which increases proportionally, in a staircase shape, to the number of binary inputs equal to 1. This operation is performed by transistor M_1-M_4 in their saturation regions. Transistors M_2 and M_4 are equally sized *n*-channel νMOS transistors. M_1 and M_3 area equal PMOS transistors. The sorter inputs are the M_2 input gates capacitively coupled to its floating gate with identical coupling capacitances, C_u, which produces a floating gate voltage, V_{FG}, linearly dependent of the sum of the inputs. However, with this circuit several input combinations with different number of 1's can give floating gate voltages below the threshold

voltage of the NMOS transistor, so not been distinguished. This offset is avoided injecting an initial charge in the M_2 floating gate. For this purpose, inverter I_1 has been included as well as two aditional inputs to transistor M_2 with coupling capacitances $C_u/2$ and C_0. With $\Phi_R = 1$ (initialization mode) switches controlled by this phase short circuit the M_2 floating gate and the output and input of I_1, and the input terminals $x_1, x_2, \ldots, x_n$ are connected to ground (input switches not shown in Figure 3). After initialization, when $\Phi_R = 0$, (processing mode), $V_{FG} = \left(\sum_{i=1}^{n} x_i \right) \cdot V_{DD} \cdot C_u / C_{tot} + V_{I1}^* - V_{DD} \cdot (C_u/2)/C_{tot}$, where V_{I1}^* is the threshold voltage of inverter I_1, $C_{tot} = (n + 1/2)C_u + C_{chan} + C_o$. Capacitance C_o is introduced by the extra grounded input in order to maintain M_2 satured, even when the n inputs of the sorter are at logical 1. This V_{FG} controls the current through M_1 and M_3. Since M_4 is made equal to M_2 this circuit produces a voltage at the M_4 drain terminal, $V_1 = V_{FG}$.The purpose of using this scheme to obtain the analog output voltage V_1 is twofold. First, to make operation insensitive to the parasitic charges in the floating gate, thus avoiding the need of post fabrication UV erasure. Secondly, to make the resulting staircase shape voltage robust versus process parameter variations.

The second block is constituted by the set of comparators which have been implemented as inverters. Each inverter is sized so that its threshold voltage is between two given consecutive steps of the staircase mentioned above. For example, the output O_1 must be a logical one if there is at least an input at logical one and so the threshold voltage of inverter I_{O1} is fixed to $(V_1(0) + V_1(1))/2$, where $V_1(0)$ stands for the voltage at node V_1 when the all zero input vector is applied and $V_1(1)$ corresponds to the voltage at node V_1 when an input vector with only one 1 is applied.

Due to process parameter variations the voltages $V_1(i), i = 0, \ldots, n$ as well as the threshold voltage of the comparator inverters can change from their nominal value. In order to reduce this sensitivity I_1 has been made identical to I_{O1}. The role of capacitor $C_u/2$ becomes now clear; it assures that with all inputs at logical zero, the V_1 voltage is under V_{IO1}^*.

Design and Evaluation of an 8-input sorter

An 8-input νMOS sorter has been designed and laid out in a 0.8 μm double poly CMOS process. In order to minimize excessive load of the comparators over the first block, the M_3-M_4 branch has been replicated. Chains of inverters have been used for the comparators to regenerate the output signal to full logic swing. Correct operation under process and ambient parameter variations have been validated through extensive HSPICE simulations of the extracted circuit including Monte Carlo simulations and simulations using different standard worst case device parameters. Figure 4 plots these simulation results for nodes V_0 and V_1 as functions of the number of inputs equal to 1. As it can be seen, the changes produced at V_0 are significantly reduced at V_1.

For the purpose of comparison, we have designed and laid out also an 8-input sorter following the Batcher's conventional approach consisting in a network of comparator cells. There are 23 of such cells in an 8-input SN. Table 1 compares the area, the time performance and the power consumption of both sorters. Time characteristics and average power have been measured on post-layout simulation results using typical device parameters at a supply voltage of 5V. The worst case delay time corresponds to situations where the inputs or the input sequence are such that the circuit operation is slowest. In the conventional design an input vector exciting the true longest path has been used to measure that delay. In the νMOS counterpart an input vector consisting on only 1's followed by an input vector consisting on only 0's has been employed. The power has been measured using a random generated input sequence with 100 vectors.

Conclusions

A new νMOS based realization for *n*-input sorters have been proposed, and its feasibility illustrated with an 8-input sorter. Compared to a conventional gate-based implementation, the νMOS design is very efficient in terms of area. It occupies an area that is nearly an order of magnitude smaller than its conventional counterpart while exhibiting better time performance. Concerning to the power consumption, it has been observed that it is very dependent of the frecuency for the conventional approach unlike to the νMOS design. At a frequency of 170MHz and above, the νMOS sorter consumes less power than the conventional one.

References

1 D.E. Knuth, *The Art of Computer Programming, Vol. III, Sorting and Searching*, 2nd ed. Reading, MA: Addison-Wesley, 1973, ch. 5.

2 K.E. Batcher, “Sorting Networks and their Applications”, in *Proc. 1968 SICC*, AFIPS, vol. 32, 1968, pp. 307-314.

3 T. Shibata, T. Ohmi, “A functional MOS transistor featuring gate level weighted sum and threshold operations”, *IEEE Trans. on Electron Devices*, **39**, (6): 1444-1445, 1990.

Captions to the figures:

Figure 1: a) Sorting Network with *k* binary signal inputs equal to 1.

b) Logic gate implementation of the comparator cell (2-input SN).

c) Batcher's implementation of a 4-input SN.

Figure 2: Schematic of the νMOS transistor.

Figure 3: Two-stage schematic of the proposed *n*-input sorter.

Figure 4: Simulated behaviour of nodes V_0 and V_1 as functions of the number of inputs equal to 1showing the stabilizing action of the circuit.

Table I: Area, time performance and power consumption of νMOS and conventional sorters.

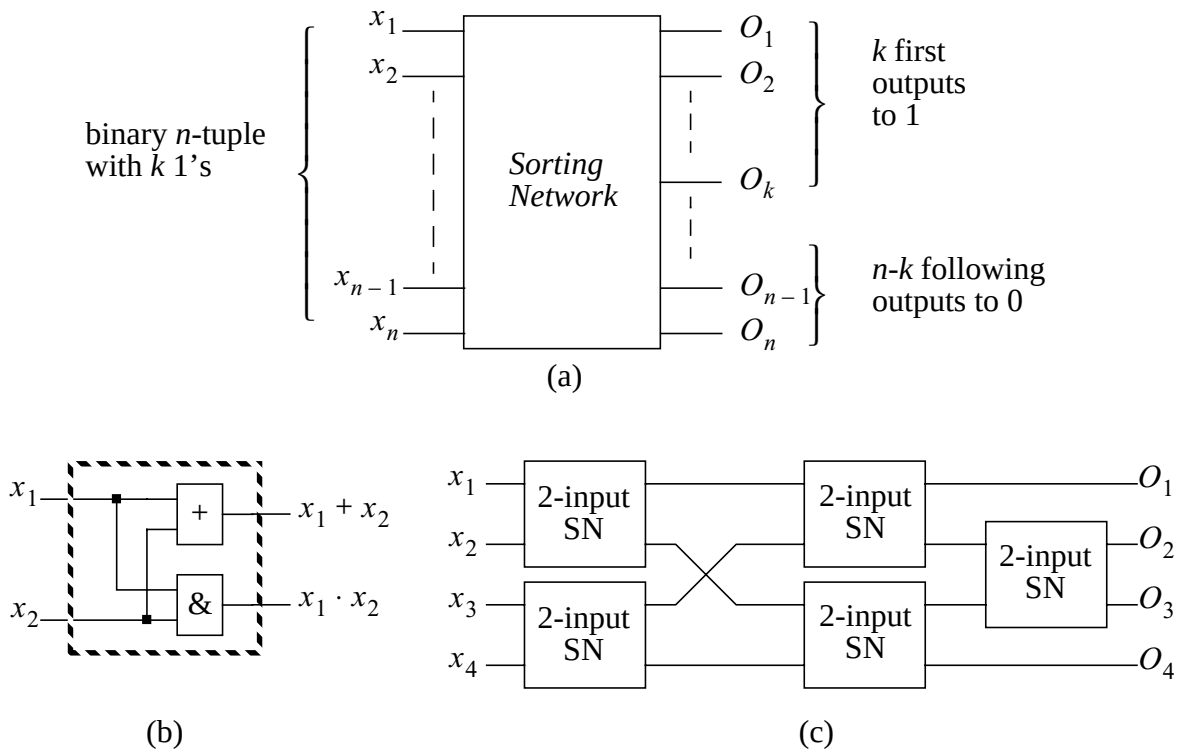


Figure 1

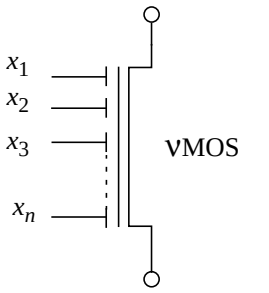


Figure 2

.

Table 1:

	Area	worst case delay	Power comsumption (@175MHz)
νMOS	5625μm^2	4.1 ns	7.8 mw
conventional	45400μm^2	5.2 ns	8.2 mw

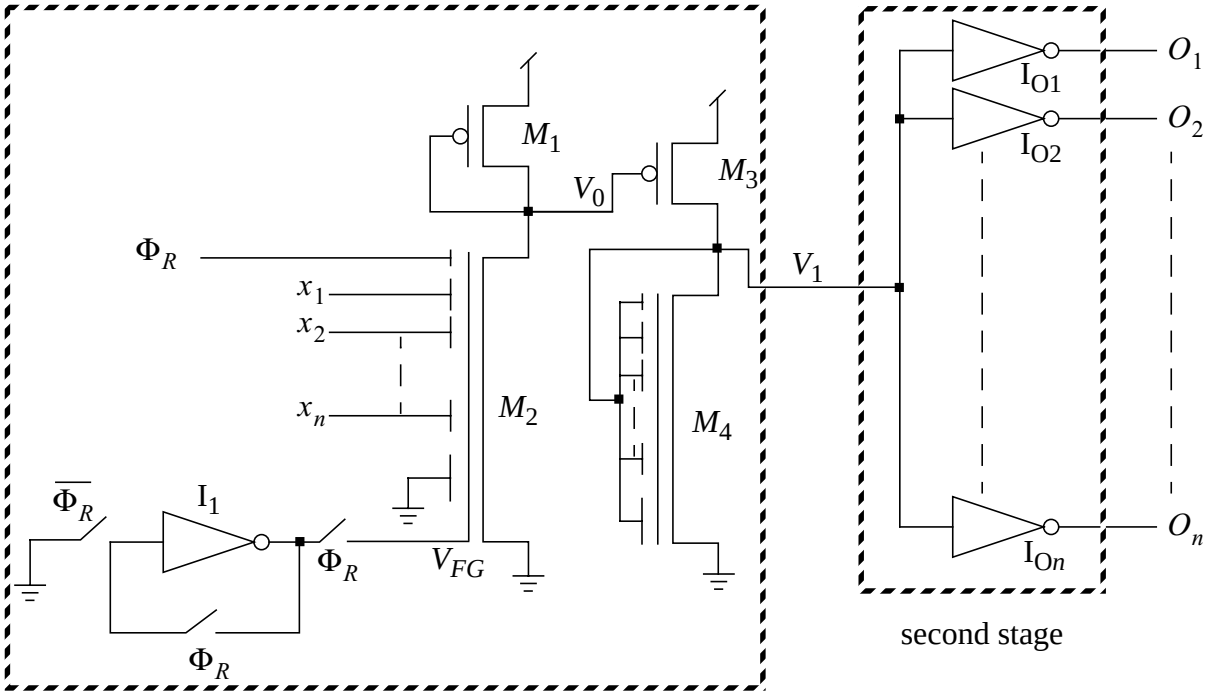
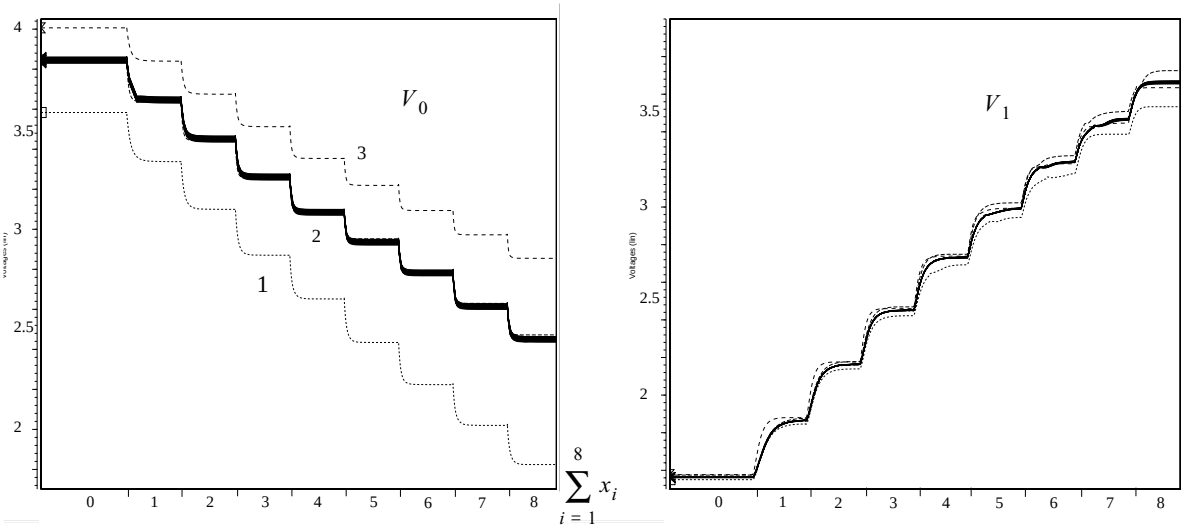


Figure 3



1 worst case (zero)
2 Monte Carlo
3 worst case (one)

Figure 4