This article was downloaded by: *[Cinvestav del IPN]*
On: *26 October 2010*
Access details: *Access Details: [subscription number 790035130]*
Publisher *Taylor & Francis*
Informa Ltd Registered in England and Wales Registered Number: 1072954 Registered office: Mortimer House, 37-41 Mortimer Street, London W1T 3JH, UK

International Journal of Electronics
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713599654

Methodology for the design of a 4-bit soft-hardware-logic circuit based on neuron MOS transistors

M. A. Reyes-Barranca[a]; A. Medina-Santiago[a]
[a] Department of Electrical Engineering, CINVESTAV-IPN, Mexico

To cite this Article Reyes-Barranca, M. A. and Medina-Santiago, A.(2008) 'Methodology for the design of a 4-bit soft-hardware-logic circuit based on neuron MOS transistors', International Journal of Electronics, 95: 6, 517 — 530
To link to this Article: DOI: 10.1080/00207210802015434
URL: http://dx.doi.org/10.1080/00207210802015434

PLEASE SCROLL DOWN FOR ARTICLE

Full terms and conditions of use: http://www.informaworld.com/terms-and-conditions-of-access.pdf

This article may be used for research, teaching and private study purposes. Any substantial or systematic reproduction, re-distribution, re-selling, loan or sub-licensing, systematic supply or distribution in any form to anyone is expressly forbidden.

The publisher does not give any warranty express or implied or make any representation that the contents will be complete or accurate or up to date. The accuracy of any instructions, formulae and drug doses should be independently verified with primary sources. The publisher shall not be liable for any loss, actions, claims, proceedings, demand or costs or damages whatsoever or howsoever caused arising directly or indirectly in connection with or arising out of the use of this material.

International Journal of Electronics
Vol. 95, No. 6, June 2008, 517–530

Methodology for the design of a 4-bit soft-hardware-logic circuit based on neuron MOS transistors

M.A. Reyes-Barranca* and A. Medina-Santiago

Department of Electrical Engineering, CINVESTAV-IPN, D.F., Mexico

(*Received 19 February 2007; final version received 19 January 2008*)

As soft-hardware-logic circuits had been proposed in the literature as an alternative for digital circuits taking advantage the fact that any Boolean function could be implemented with the same cell, just configuring external signals, this work shows a methodology that could be followed particularly for the design of a four bits logic gate, using the so-called neuron MOS transistor (ν-MOS). Simulation results show the feasibility of the design for performing as XNOR, NOR, OR, XOR, AND or NAND logic gates, for instance. In order to extrapolate the design to a higher number of bits, the key issue is to properly consider the weight of the input capacitances in correlation with the number of input bits. A D/A converter can be used as the input stage of the configuration. This design considers the D/A converter-less version, since it helps to increase device integration as the number of transistors used is reduced with no difference in its performance. The design should be based on the theoretical floating potential diagram (FPD) of the desired logic gate.

Keywords: floating-gate transistor; soft-hardware-logic; floating-gate potential diagram; CMOS; neu-MOS

Downloaded By: [Cinvestav del IPN] At: 14:35 26 October 2010

I. Introduction

First formally conceived in 1967, the MOS floating gate transistor is formed by an electrically isolated gate and a control gate. Through time, this first device has been well studied, such that new and optimum designs were reported in literature and implemented in practical systems. Such is the case of the so called neu-MOS (or νMOS) since it resembles the behaviour of a biological neuron, as it has more than one control gate, then functionally representing a weighted sum at the floating gate, of all the input signals connected to the control gates. As a methodology, it presents a means for implementing a non-volatile memory element in silicon CMOS technology when the charge on the floating gate is to be modified when storing information. When performing as a non-volatile memory, the charge on the floating gate can be modified, having a charge loss of about 0.1% in 10 years. This is not the case in circuits like those presented in this work, since the condition needed for the operation of the gates is to have no charge at all upon the floating gate. Although floating gate transistors are primarily used as storage devices in digital systems, there has been a trend in research and development over the last 15 years to use them as an analogue circuit element (Schwartz, Howard and Hubbard 1989;

*Corresponding author. Email: mreyes@cinvestav.mx

ISSN 0020–7217 print/ISSN 1362–3060 online
© 2008 Taylor & Francis
DOI: 10.1080/00207210802015434
http://www.informaworld.com

Kub, Moon, Mack and Long 1990; Shibata and Ohmi 1992; Ramirez-Angulo, Choi and Gonzalez Altmirano 1995; Minch, Diorio, Hasler and Mead 1996; Hasler and Lande 2001; Ramirez-Angulo and Lopez 2001). Such is the case reported in (Shibata and Ohmi 1993a,b), where soft-hardware logic (SHL) circuits based on neu-MOS transistor are presented. The concept of SHL circuit proposed by Shibata and Ohmi using a multi-input floating gate MOS transistor (MIFGMT) brings out the opportunity to increase the integration of more transistors in a small area. This is such a versatile configuration as any Boolean function can be implemented with the same circuit. Another advantage of this configuration, over the conventional gates, is the great reduction in the number of transistors, which can be as high as 90%. On the other hand, the concern is directed to technological issues, such as device parameter spreading, since multiple input capacitances are used in the neuMOS, and operating frequency, which is also reduced due to the lower transconductance of the MIFGMT.

It is interesting to have a background in designing these SHL configurations, as different functions and systems could be designed and simulated, so performance of circuits as A/D converters, summers or logic gates, can be evaluated in advance of their physical implementation with a silicon foundry. The circuits can be designed using a D/A converter as the input stage of the basic cell, but with the inconvenience that the threshold voltages of the PMOS and NMOS transistors, used in a complementary νMOS source follower configuration, should be positive and negative, respectively. This imposes the use of ion implantation to adjust the threshold voltages, and this represents an extra technological step not available in commercial technology. Anyway, a version of the gate without this stage is possible with no problem, reducing even the number of transistors used and increasing the speed. The four bit circuit designed in the present work, is followed with the later approximation and can be fabricated with available commercial technologies having two polisilicon layers.

Downloaded By: [Cinvestav del IPN] At: 14:35 26 October 2010

The concept of theoretical floating-gate potential diagram (FPD) of the logical functions is presented and simulated using PSpice, corresponding to the equivalent expected response of the logic function considered. A circuit that represents logic functions, such as XOR, NAND, NOR, AND, OR and XNOR, etc., can be implemented by adjusting external control signals without any modifications in the circuit configuration and a detailed design methodology is presented. The FPD is used for the gate performance prediction, applied to the floating gate of the νMOS inverter, shown in Figure 1. First, an equivalent representation of the desired logic output, should be made with the help of an FPD. Then the threshold voltages of the programmable inverters $(A, B, C \ldots)$ are read from the abscissa of the FPD, and the values of coupling capacitances of the output stage (neuMOS circuit) are determined from the ordinate of this same graph. One main objective is to reduce the number of transistors through the elimination of the D/A converter shown in Figure 1, using the FPD. In this case, the design is done for a gate with a 4-b input, establishing the magnitude of coupling capacitances and external voltages necessary for each of the studied gates.

In §2 of this paper, the design methodology for a circuit with a logical external configuration is presented, as well as the standards of design for the coupling capacitances used in the neuron circuit, programmable inverters and pre-input-gate inverter. Also, the FPD is described in detail. Section 3 will present the simulation for one of the gates considered, simulated both in DC and transient behaviours, together with the table of external voltages that should be applied to the programmable inverters, in order to obtain

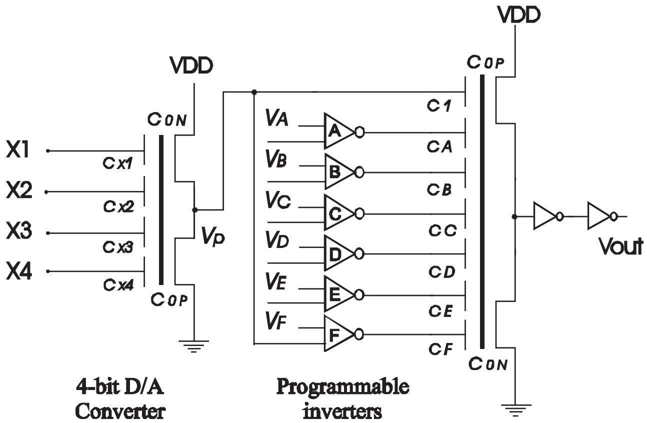

Figure 1. Configuration of a 4-b νMOS SHL circuit, with a D/A converter as the input stage.

Downloaded By: [Cinvestav del IPN] At: 14:35 26 October 2010

the respective logic functions. Section 4 presents the conclusions of this work, and a suggested application for the designed circuit.

2. Methodology

The configuration of an SHL circuit with a D/A converter at the input stage is presented in Figure 1. The circuit receives binary signals X1, X2, X3 and X4 as the input and gives a binary signal output, VOUT. This particular circuit, given as an example, will show the XOR function of X1, X2, X3 and X4. The output stage of the circuit is a neuron circuit and is comprises a 7 input-gate complementary neuMOS inverter and a conventional inverter, whose function is to give logic 0s and 1s, as with a conventional digital circuit (Shibata and Ohmi 1991, 1992). The input stage of the circuit is a complementary νMOS source follower which serves as a single-stage D/A converter. This circuit converts a 4-b binary input signal, X1, X2, X3 and X4, into a sixteen-level analogue signal, Vp, which is named as a principal variable. This variable, Vp, is also fed to inverters A, B, C, D, E and F (Figure 1). The simulations of the circuit in Figure 1 can be seen in Medina-Santiago and Reyes-Barranca (2003, 2005).

The D/A converter can be eliminated if the principal variable is fed properly through a pre-input-gate inverter to the programmable inverters (intermediate stage) as will be shown later. First, consider a design example using the D/A converter, in order to explain how to use the FPD diagram. The FPD of the output stage in Figure 1 is presented in Figure 2, where the y-axis is the floating-gate potential of the neuronMOS circuit (ϕ_F) and the x-axis corresponds either to the analogue value, Vp, from 0 Volts to V_{DD} (base line, Figure 2), or to its digital equivalence for each of the sixteen possible combinations with 4-b.

From Figure 1, the capacitances considered are the following: C_{X1}, C_{X2}, C_{X3} and C_{X4}, are the coupling capacitances for the D/A converter input signals, with C_{X1} for the LSB. C_1 corresponds to the coupling capacitance of the neu-MOS circuit connected to the D/A output. C_A, C_B, C_C, C_D, C_E and C_F are the coupling capacitances connecting each programmable inverter's output, to the neu-MOS circuit. The number of coupling capacitances used for the circuit is determined precisely from the FPD.

First, it should be noticed in Figure 2, that the x-axis is divided into 16 subdivisions for the shown case. This is useful for the determination of the threshold voltage of the programmable inverters and, in general, this is derived from the number of input bits, as follows:

$$\#subdiv.(x\text{-axis}) = 2^N \tag{1}$$

where N is the number of bits considered. Also, the number of subdivisions in the y-axis, should be

$$\#subdiv.(y\text{-axis}) = 2 * 2^N \tag{2}$$

from where the magnitude of the coupling capacitances can be determined in terms of the floating-gate device gain γ, which corresponds to the ratio of the sum of the coupling capacitances to the total capacitance, Equation (3). In this case, the ordinate, ϕ_F, is divided into 32 divisions with γV_{DD} as its maximum. The *y-axis* can be used either to read voltage (floating-gate potential) or capacitance, having in this case, a maximum of γC_{TOT}.

$$\gamma = \frac{C_1 + C_A + C_B + C_C + C_D + C_E + C_F}{C_{TOT}} \tag{3}$$

$$C_{TOT} = C_0 + C_1 + \cdots + C_n \tag{4}$$

$$C_0 = C_{ON} + C_{OP} \tag{5}$$

Downloaded By: [Cinvestav del IPN] At: 14:35 26 October 2010

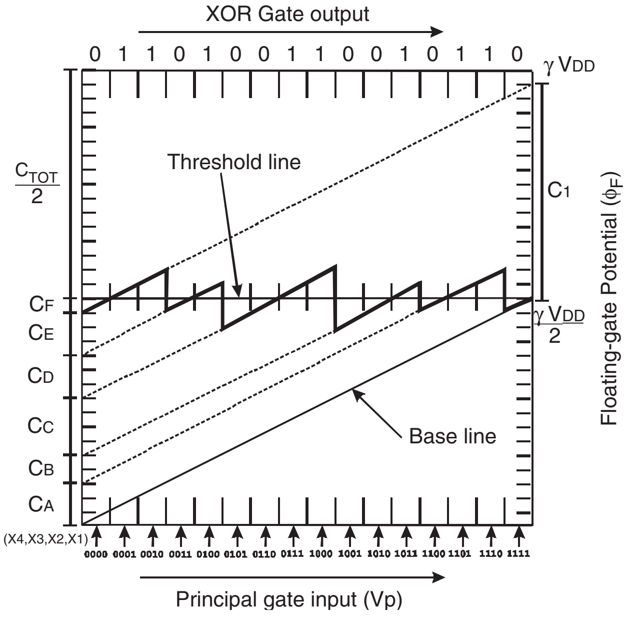

Figure 2. Theoretical floating-gate potential diagram (FPD) for the main νMOS inverter in Figure 1. The case for the XOR gate is illustrated.

$$C_1 = C_{X_1} + C_{X_2} + C_{X_3} + C_{X_4} \tag{6}$$

$$\phi_F = \frac{V_P C_1 + V_A C_A + V_B C_B + V_C C_C + V_D C_D + V_E C_E + V_F C_F}{C_{TOT}} \tag{7}$$

where C_{ON} and C_{OP} are, respectively, the gate capacitances of the NMOS and PMOS transistors of the neuronMOS inverter and will be considered as the minimum unit cell and γ is the floating-gate gain. Since, with this kind of logic function representation, a reference is needed to determine whether the output voltage is a logic '0' or a logic '1', this should be indicated with a line at the middle of the FPD ($\gamma V_{DD}/2$), and labelled as 'threshold line'. Then, each of the subdivisions in the abscissa scale sets the sixteen possible inputs, from '0000' to '1111', correspondingly as Vp increases. A reference slope to trace the equivalent binary function is also needed. This can be done, drawing a line from coordinates $(0, 0)$ to $([2 * 2^N]/2,\ [2 * 2^N]/2)$, and labelled as 'base line'. Analytically, this can be done with the help of Equation (8) when $V_A = V_B = V_C = V_D = V_E = V_F = 0$.

$$\phi_F = \frac{V_P C_1}{C_{TOT}} \tag{8}$$

Following this slope, lines are traced in the FPD over or under the threshold line where a logic '1' or a logic '0', respectively, is dictated by the truth table of the gate, resulting in the piecewise linear signal shown in Figure 2. Table 1 shows the functional table of an XOR gate used in this example, from where the signal was drawn in the FPD.

As mentioned before, the y-axis can be used to read either voltage or capacitance. In the next section it is explained how the coupling capacitances of the neuronMOS circuit can be determined, from where the number of programmable inverters needed in the converter-less version can be known.

Table 1. Functional table of a four bits input XOR gate.

X4	X3	X2	X1	OUT
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Downloaded By: [Cinvestav del IPN] At: 14:35 26 October 2010

2.1. *The design standard for the coupling capacitances of the neuron circuit*

As is shown in Figure 2, lines are extrapolated to the y-axis following the slope of the base line. Then, from the FPD the coupling capacitances of the neuron are graphically determined as follows:

$$\begin{array}{lll} C_A = \frac{3}{32}\gamma C_{TOT} & C_D = \frac{3}{32}\gamma C_{TOT} & \\ C_1 = \frac{15}{32}\gamma C_{TOT} & C_B = \frac{2}{32}\gamma C_{TOT} & C_E = \frac{3}{32}\gamma C_{TOT} \\ C_C = \frac{4}{32}\gamma C_{TOT} & C_F = \frac{1}{32}\gamma C_{TOT} & \end{array}$$

Those capacitances below the threshold line correspond to those connecting the programmable inverters to the neuron circuit and those above the threshold line correspond to the coupling capacitance for Vp, i.e., C_1. Then, the number of programmable inverters is derived from the number of coupling capacitances determined, other than that used to couple Vp. In this case, the programmable inverters needed are six. It is easy to see that

$$C_1 + C_A + C_B + C_C + C_D + C_E + C_F = 31/32\,\gamma C_{TOT} \qquad (9)$$

So

$$C_0 = 1/32\,\gamma C_{TOT} \qquad (10)$$

The weight of each capacitance in Equation (6) can be set as a recommended design standard, following the rule when converting each position of a digital word into its decimal equivalence, i.e., 2^0, 2^1, 2^2, $2^3, \ldots, 2^N$, with the first as the LSB. Then, for the 4-b case

$$\begin{array}{ll} C_{X_1} = \frac{1}{32}\gamma C_{TOT} & C_{X_3} = \frac{4}{32}\gamma C_{TOT} \\ C_{X_2} = \frac{2}{32}\gamma C_{TOT} & C_{X_4} = \frac{8}{32}\gamma C_{TOT} \end{array}$$

2.2. *Threshold voltage of the programmable inverters*

The corresponding inverter for each of the capacitances $C_A, \ldots, C_F$, is used to force ϕ_F to go down through the reference line in the FPD, in order to accomplish with the gate function. The voltage at which each inverter is programmed is defined by the position where the FPD crosses the reference line, $\gamma V_{DD}/2$. For the XOR gate in Figure 2, these threshold voltages are

$$\begin{array}{ll} V_{IA} = \frac{15}{16}V_{DD} & V_{ID} = \frac{5}{16}V_{DD} \\ V_{IB} = \frac{12}{16}V_{DD} & V_{IE} = \frac{3}{16}V_{DD} \\ V_{IC} = \frac{9}{16}V_{DD} & V_{IF} = 0 \end{array}$$

Downloaded By: [Cinvestav del IPN] At: 14:35 26 October 2010

Here, it should be pointed out that each programmable inverter's output will be high until the respective inversion voltage defined by the FPD is reached by the sum of Vp and the external voltage applied to the inverter, which should be determined as the programming voltage. In consequence, the base line in Figure 2 will be shifted upwards following Equation (7), going down when the output voltage of the respective programmable inverter goes low through its inversion voltage, following the method of the theoretical FPD.

Now, in regard to the design of the programmable inverter, it consists of a floating-gate CMOS inverter with two inputs, as shown in Figure 3. One input receives the variable Vp and the other, the external voltage $V_A, \ldots, V_F$ from which the programmed inversion voltage is achieved, and whose magnitude depends on the threshold voltages. A nice design standard that can be applied in order to obtain the magnitude of the respective capacitances, is the following:

$$C_0 = C_{inv} = C_V \tag{11}$$

where $C_0 = C_{0N} + C_{0P}$, C_{inv} is the capacitance connected to Vp and C_V is the capacitance connected to an external voltage. Hence, $C_{TOT(inv)}$ is expressed as follows:

$$C_{TOT(inv)} = C_0 + C_{inv} + C_V \tag{12}$$

and from Equation (11)

$$\frac{C_0}{C_{TOT(inv)}} = \frac{C_{inv}}{C_{TOT(inv)}} = \frac{C_V}{C_{TOT(inv)}} = \frac{1}{3} \tag{13}$$

If this ratio is increased, i.e. to 1/2, the resulting external voltages, $V_A, \ldots, V_F$, can be decreased in consequence, so it can be derived from the V_{DD} circuit bias (Shibata et al. 1993b). Using the ratio expressed in (13), results in the voltages shown in Table 2, which are above V_{DD} and this can lead to the use of dedicated pins in the chip for the application of these voltages. Then, there is a tradeoff between increasing the capacitance ratio, which will affect the integration area and the speed, as well, and the availability of pins for the integrated circuit for particular functions. In order to have a symmetric transfer function,

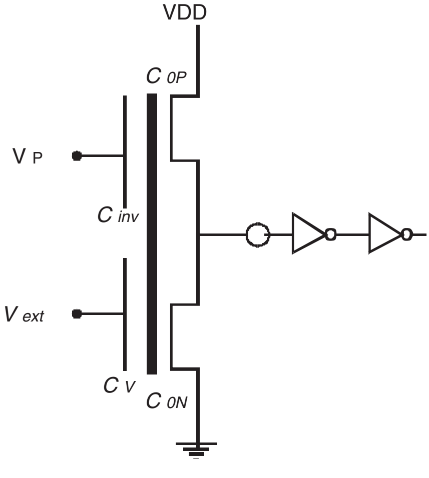

Figure 3. Configuration of one programmable inverter, where *Vp* corresponds to the 4-b input and *Vext* is the external voltage input found with Equation (15).

Downloaded By: [Cinvestav del IPN] At: 14:35 26 October 2010

Downloaded By: [Cinvestav del IPN] At: 14:35 26 October 2010

the voltage in the floating gate of the inverter must be $\phi_{Finv} = V_{DD}/2$, which in turn is the sum of the weighted voltages applied to C_{inv} and C_V. Since Vp is applied to each programmable inverter, the voltage that should be applied to C_V must be calculated depending on the inversion voltage that was read from the FPD. The expression from which this can be known is

$$\phi_{Finv} = \frac{V_{DD}}{2} = \frac{C_{inv} \cdot V_p + C_V \cdot V_{ext}}{C_{TOT(inv)}} = \frac{C_{inv}\left(V_p + V_{ext}\right)}{C_{TOT(inv)}} \tag{14}$$

where V_{ext} is equal to either $V_A, \ldots, V_F$. Solving for V_{ext} from Equation (14)

$$V_{ext} = \frac{C_{TOT(inv)}}{C_{inv}} \cdot \frac{V_{DD}}{2} - V_p \tag{15}$$

and Vp must be substituted by V_{IA}, for the inverter A, V_{IB}, for the inverter B and so on. Figure 4 shows a simulation of the transfer curve of inverter A, with Vp $= V_{IA} = 15/16V_{DD}$,

Table 2. External voltages applied to the programmable inverters.

	Corresponding external voltages of the programmable inverters						
Logic function	V_A	V_B	V_C	V_D	V_E	V_F	V_{C2}
XNOR	7.18 V	6.25 V	5.31 V	4.06 V	3.43 V	0 V	0 V
NOR	7.18 V	7.18 V	7.18 V	7.18 V	7.18 V	5 V	5 V
AND	7.46 V	7.46 V	7.46 V	7.46 V	7.46 V	0 V	0 V
NAND	2.81 V	2.81 V	2.81 V	2.81 V	2.81 V	0 V	0.64 V
OR	0 V	0 V	0 V	0 V	0 V	5 V	5.5 V
XOR	2.81 V	3.75 V	4.68 V	5.93 V	6.56 V	5 V	5 V

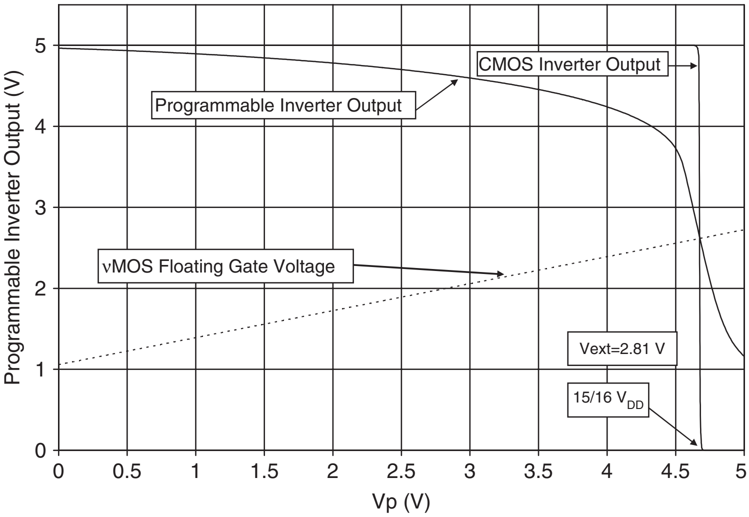

Figure 4. Simulation of the transfer curve of a programmable inverter with $V_I = 15/16V_{DD}$, with an external voltage of 2.81 V, calculated from (15).

$V_{DD} = 5\,\text{V}$ resulting in $V_{ext} = 2.81\,\text{V}$ using Equation (15). It can be seen from this figure that the output of the conventional CMOS inverters connected to the programmable inverter's output changes from a high output to a low output just in 15/16 V_{DD} (4.6875 V). Also, the value of the floating gate potential (dotted line), ϕ_F, is approximately $\phi_{Finv} = V_{DD}/2$, as desired for a symmetric transfer function. This simulation can be done with each of the crossing points specified in the FPD of Figure 2, in the case of the XOR gate, or any other gate considered. Proceeding this way and using (15), the external voltages listed in Table 2 were determined for the six gates analysed.

3. The converter-less gate design

The D/A converter-less version of the exclusive OR (XOR) circuit is shown in Figure 5, in which the configuration of the pre-input-gate inverter (with input signals Vp, V_0 and V_F) is explicitly shown (to be compared with the circuit diagram in Figure 1). The four input signals X1, X2, X3 and X4 are directly coupled via C_{X1}, C_{X2}, C_{X3} and C_{X4}, to the floating gates of the two νMOS inverters, having a weight ratio of 1 : 2 : 4 : 8, respectively, following the same design standards outlined before. For simplicity, these four inputs are indicated as one capacitance, named C_1, that equals their sum. Eliminating the D/A converter stage, simplifies the circuit configuration, therefore improving the integration density as well as the speed performance, with a slight penalty of an increased number of interconnects.

Based on the FPD, as was mentioned above (Figure 2), the coupling capacitances of the neuronMOS circuit and the threshold voltages for each of the programmable inverters used can be determined, as well as the quantity of programmable inverters needed to configure the logic function desired; the same way it was done with the configuration of Figure 1. Following the same example for the XOR gate, the threshold voltages, capacitance values and number of programmable inverters, are the same for

Downloaded By: [Cinvestav del IPN] At: 14:35 26 October 2010

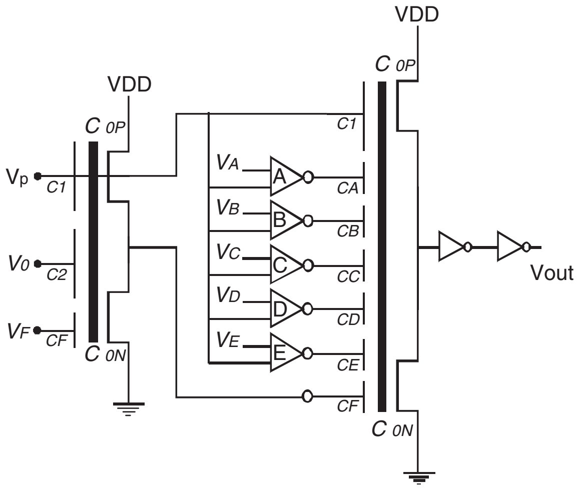

Figure 5. Converter-less version of the exclusive-OR circuit with four input variables X1, X2, X3 and X4.

Downloaded By: [Cinvestav del IPN] At: 14:35 26 October 2010

the converter-less version, because the FPD holds, even in this case, for the neuronMOS circuit. Since, finally, the floating gate of the νMOS inverters is a summing point, one can take advantage of this to eliminate the D/A converter and merge one of the programmable inverters with the input bits, represented with the νMOS inverter on the left in Figure 5. So, consider the design outline that can be suggested for this first stage. Once again, starting from the capacitance unit cell

$$C_0 = C_{0N} + C_{0P} = C_{X1} \tag{16}$$

$$C_{X1} = C_F \tag{17}$$

$$C_{X2} = 2 \cdot C_{X1} \tag{18}$$

$$C_{X3} = 4 \cdot C_{X1} \tag{19}$$

$$C_{X4} = 8 \cdot C_{X1} \tag{20}$$

$$C_2 = C_{X1} + C_{X2} + C_{X3} + C_{X4} \tag{21}$$

$$C_{TOT(pre)} = C_0 + C_{X1} + C_{X2} + C_{X3} + C_{X4} + C_2 + C_F = 32 \cdot C_{X1} \tag{22}$$

Thus, the ratio of the coupling capacitances of the pre-input-gate inverter to the total capacitance can be expressed as

$$\frac{C_{X1}}{C_{TOT(pre)}} = \frac{1}{32} \quad \frac{C_{X4}}{C_{TOT(pre)}} = \frac{8}{32}$$
$$\frac{C_{X2}}{C_{TOT(pre)}} = \frac{2}{32} \quad \frac{C_2}{C_{TOT(pre)}} = \frac{15}{32}$$
$$\frac{C_{X3}}{C_{TOT(pre)}} = \frac{4}{32} \quad \frac{C_F}{C_{TOT(pre)}} = \frac{1}{32}$$

As mentioned before, the pre-input-gate inverter merges the 4-bits input with one programmable inverter, so it is necessary to calculate the external voltage, V_0 and V_F, that should be applied either to C_2 or C_F, respectively. First, the voltage on the floating gate of the pre-input inverter, $\phi_{F(pre)}$ must be expressed as a function of the signals applied to each coupling capacitance.

$$\phi_{F(pre)} = \frac{C_{X1} \cdot V_1 + C_{X2} \cdot V_2 + C_{X3} \cdot V_3 + C_{X4} \cdot V_4 + C_2 \cdot V_0 + C_F \cdot V_F}{C_{TOT(pre)}} \tag{23}$$

Here, it should be pointed out that Vp is now completely digital but being coupled to the floating gate of the pre-input-gate-inverter via C_{X1}, C_{X2}, C_{X3} and C_{X4}. This is important to remember in order to make the next considerations, made for the calculation of the external voltage that must be applied to C_2 and C_F, comprehensive. Since we are considering that the programmable inverter, F is configured in the pre-input-gate-inverter, its threshold voltage was determined before as $V_P = V_{IF} = 0$ for the XOR gate case. Using Equations (16)–(23), the expression from where the external voltages applied to

C_2 and C_F can be calculated is

$$\phi_{F(pre)} = \frac{V_{DD}}{2} = \frac{15 \cdot C_{X1} \cdot V_P + 15 \cdot C_{X1} \cdot V_0 + C_{X1} \cdot V_F}{C_{TOT(pre)}} \tag{24}$$

If $V_P = V_{IF} = 0$, then

$$\phi_{F(pre)} = \frac{V_{DD}}{2} = \frac{15 \cdot C_{X1} \cdot V_0 + C_{X1} \cdot V_F}{C_{TOT(pre)}} = \frac{C_{X1}}{C_{TOT(pre)}} \cdot (15 \cdot V_0 + V_F) \tag{25}$$

It can easily be shown that if $V_0 = V_F = V_{DD}$ is chosen and knowing that $C_{X1}/C_{TOT(pre)} = 1/32$, making substitutions in Equation (25), results effectively in $V_{DD}/2$. In practice, this applied voltages result in a low output of the programmable inverter F, all over the range of V_P, as dictated by the FPD. The design considerations for the rest of the programmable inverters (A, B, C, D and E) are the same as those used for the configuration using a D/A converter.

Downloaded By: [Cinvestav del IPN] At: 14:35 26 October 2010

4. Simulations

Simulations were carried out using PSpice, with level 7 model for MOS transistors, using the technological parameters of the 1.2 μm AMIS technology. Figure 6a–f demonstrates the simulation results for the six functions considered in this methodology. The graphs on the left show the DC simulation and the graphs on the right show the transient simulation with satisfactory results in each logic gate. Thus, the methodology outlined here has been demonstrated also to apply to a converter-less version, and with a 4-b input. Besides, as mentioned in the first section, the number of transistors needed for implementing the logic function, is greatly reduced, optimising hence, the integration area. Also, care should be taken with the external voltage of the programmable inverters when configuring gates as NOR, AND, NAND and OR. The reason for the advice can be justified by their corresponding FPD. As can be seen in Figure 6a–d, these four gates have only one crossing of ϕ_F through the threshold line. Therefore, a simple analysis of the appropriate output of the six inverters should be carried out to predict if it must be high or low to shift the FPD in the correct direction. For example, the NOR gate must have all the inverters with a high output (low input) until $1/16V_{DD}$, where, once again, all of them must have a low output (high input). Using Equation (15), the value that must be applied to $V_A, \ldots, V_E$ is 7.18 V (see Table 2) and V_F and V_0 can be 5 V, and using Equation (25), $\phi_{F(pre)} = 2.5$ V. Hence, as Vp increases, the output of all the inverters is low, following the base line from $Vp = 1/16V_{DD}$ until V_{DD}. For the AND gate, the programmable inverters A, B, C, D and E must have a low output all over the Vp range compared with the pre-input-gate inverter, that must shift ϕ_F up in the FPD by an amount of 1/32 γV_{DD}. The external voltages applied to the programmable inverters A through E, are calculated with Equation (15), with $Vp = 0$ V. The case of the NAND gate is the opposite of the AND gate, this is, the output of the programmable inverters must be high, and the voltages are calculated with $Vp = 15/16V_{DD}$ and $V_0 = 0$ V and $V_F = 0.64$ V. Finally, for the OR gate, programmable inverters must have a high output and the pre-input-gate inverter, a low output all over the Vp range. During simulation with PSpice and depending on the parameters used in the model for MOS transistors, which may depend on the technology that will be used to fabricate the circuits, some

Downloaded By: [Cinvestav del IPN] At: 14:35 26 October 2010

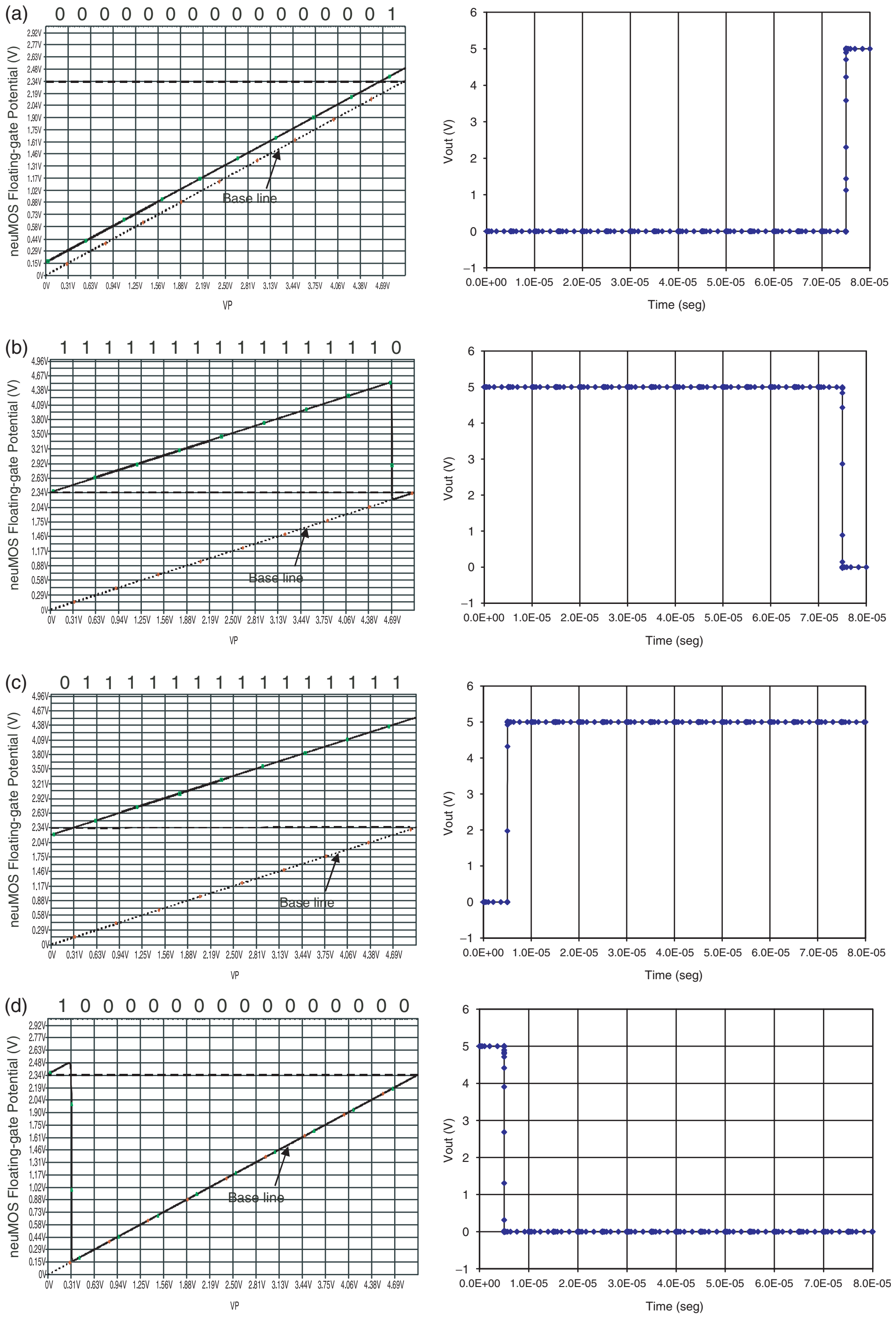

Figure 6. Simulation with PSpice of SHL gates with 4-b input and without D/A converter: (a) AND; (b) NAND; (c) OR; (d) NOR; (e) XNOR; (f) XOR. Note that each subdivision of the *x*-axis is $V_{DD}/16$ and that the threshold line is at $y = \gamma V_{DD}/2$, for $V_{DD} = 5\,\text{V}$.

Downloaded By: [Cinvestav del IPN] At: 14:35 26 October 2010

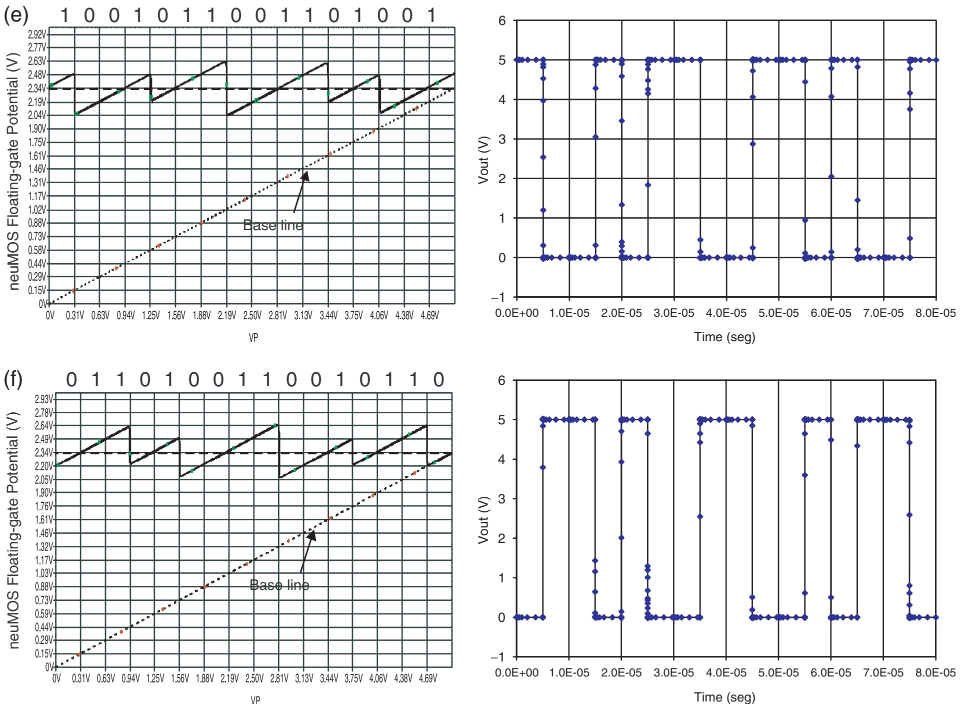

Figure 6. Continued.

fitting may be done, for instance, in the geometries of the devices to adjust either the external voltages or the transfer curve of the inverters.

Table 2 shows all the applied voltages to the programmable inverters, corresponding to the respective inversion thresholds of the programmable inverters for the six logic functions considered. Here, the fact that the magnitude of some voltages are higher than V_{DD} can be highlighted. This was the result of the standards followed in the design of the programmable inverters. Voltages in the order of V_{DD} or lower can be used if the ratio of the coupling capacitances is increased, for instance to 1/2 instead of 1/3, as was used in the example. On the other hand, doing this will result in an increase in the area of the coupling capacitance.

5. Conclusions

The concept for logic external configuration circuits based upon a theoretical FPD for logical functions was presented. It is possible to simulate with PSpice, the behaviour of a soft-hardware logic gate, with design standards derived form the so called FPD graph. The results obtained correspond to the expected response for the logic functions considered, in this case, for a 4-b input signal. The same circuit can perform several logic functions such as XOR, NAND, NOR, AND, OR and XNOR, by just adjusting external control signals without any modifications to the circuit configuration and good performance was obtained for all of them.

From a practical point of view, the original FPD representation, like the one shown in Figure 2, is much easier to use in designing this kind of logic circuits. Therefore, the presented procedure can be applied. First, draw an FPD pattern that matches the target function, then derive the coupling capacitances from the FPD graph axes; calculate the external voltages applied to the programmable inverters, depending on the logic function desired, and finally simulate the circuit, observing their behaviour.

The number of programmable inverters necessary to implement the circuit, is directly proportional to the times that Φ_F crosses the threshold line ($\gamma V_{DD}/2$) present in an FPD graph, as well as the number of coupling capacitances determined from the y-axis of the FPD, independently from those used for the input signals. For the case studied in this work, it was determined that six programmable inverters were needed. When the external voltages applied to them are adjusted, several logic functions can be performed, without altering the circuit configuration. A future application of this circuit is the development of arithmetic circuits, i.e. adders and multipliers. This could lead to the design of an ALU with parallel processing, for instance.

Downloaded By: [Cinvestav del IPN] At: 14:35 26 October 2010

References

Hasler, P., and Lande, T.S. (2001), "Overview of Floating-gate Devices, Circuits, and Systems," *IEEE Transactions on Circuits and Systems—II: Analogue and Digital Signal Processing*, 48, 1–3.

Kub, F.J., Moon, K.K., Mack, I.A., and Long, F.M. (1990), "Programmable Analogue Vector Matrix multipliers," *IEEE Journal of Solid-State Circuits*, 25, 207–214.

Medina-Santiago, A., and Reyes-Barranca, M.A. (2003), "Convertidor D/A con dispositivos de Compuerta flotante", *9a. Conferencia de Ingeniería Eléctrica*, México D.F., pp. 127–130.

Medina-Santiago A., and Reyes-Barranca, M.A. (2005), "Circuit for Logical-binary Functions using MOS floating-gate Devices," *2nd International Conference on Electrical and Electronics Engineering and XI Conference on Electrical Engineering*, Mexico City, pp. 211–214.

Minch, B.A., Diorio, C., Hasler, P., and Mead, C.A. (1996), "Translinear Circuits using Sub-threshold floating-gate MOS Transistors," *Analogue Integrated Circuits and Signal Processing*, 9, 167–179.

Ramirez-Angulo, J., Choi, S.C., and Gonzalez Altamirano, G. (1995), "Low-voltage Circuits Building-blocks using Multiple-input Floating-gate Transistor," *IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications*, 42, 971–974.

Ramirez-Angulo, J., and Lopez, A.J. (2001), "Mite Circuits: The Continuous-time Counterpart to Switched-Capacitor Circuits," *IEEE Transactions on Circuits and Systems II-Analogue and Signal Processing*, 48, 45–55.

Schwartz, D.B., Howard, R.E., and Hubbard, W.E. (1989), "A programmable Analogue Neural Network Chip," *IEEE Journal of Solid-State Circuits*, 24, 313–319.

Shibata, T., and Ohmi, T. (1991), "An Intelligent MOS Transistor Featuring Gate Level Weight Sum and Threshold Operations," *IEDM Tech. Dig*, December, 919–922.

Shibata, T., and Ohmi, T. (1992), "A Functional MOS Transistor Featuring Gate Level Weight Sum and Threshold Operations" *IEEE Transactions on Electron Devices*, 39, 1444–1455.

Shibata, T., and Ohmi, T. (1993a), "Neuron MOS Binary-Logic Integrated Circuits – Part I: Design Fundamentals and Soft-Hardware-Logic Circuit Implementation," *IEEE Transactions on Electron Devices*, 40, 570–576.

Shibata, T., and Ohmi, T. (1993b), "Neuron MOS Binary-logic Integrated Circuits – Part II: Simplifying Techniques of Circuit Configuration and their Practical Applications," *IEEE Transactions on Electron Devices*, 40, 974–979.