

β -Driven Threshold Elements

Victor I. Varshavsky
the University of Aizu, Japan
victor@u-aizu.ac.jp

Abstract

Circuits on threshold elements have aroused considerable interest in recent years. One of the possible approaches of their implementation is using output wired CMOS invertors [3,4,5]. The model of such an element is a CMOS pair with variable β of fully open p - and n -transistors. This model is specified by the ratio form of threshold function. It has been proved that any threshold function can be rewritten in ratio form. This gives us an evident way of β -driven implementation of threshold functions. It has the following differences from implementation on output wired CMOS invertors:

- β DTE requires one transistor per weight unit rather than two;

- the implementability of β DTE depends only on threshold value, not on the input weights sum.

The analysis of β DTE implementability, examples of circuits and results of their SPICE simulation are given.

1 Introduction

During the last 40 years, the tides of interest to threshold and majority logics rose and waned periodically. This was caused, on the one hand, by new circuit elements of threshold nature (like transfluxors and parametrons in 50's) and, on the other hand, by the tasks which used threshold functions to specify functional blocks. There is a lot of such tasks, for example, those of reliability, threshold coding, AD/DA-conversion, filtration, etc. Of special note are neural networks, the behavior of formal neurons in which is described by threshold functions, starting from the classic work by McCulloch and Pitts.

One of the central questions here is hardware implementation of threshold functions. Since threshold functions are a subclass of Boolean functions, any threshold function can be naturally represented as a superposition of operations (circuit elements) of any functionally full basis. The question is whether we can build circuits that would implement threshold functions or some of their types in a simpler way than the traditional circuits do. If the answer is positive, we probably can simplify the implementations of arbitrary logic functions having a bigger number of basic elements.

A threshold function is defined as

$$y = \text{Sign}(\sum_{i=0}^{n-1} w_i x_i - T) =$$

$$\begin{cases} 1 & \text{if } \sum_{i=0}^{n-1} w_i x_i - T \geq 0 \\ 0 & \text{if } \sum_{i=0}^{n-1} w_i x_i - T < 0 \end{cases} \quad (1)$$

where w_i is the weight of the i -th input and T is the threshold. Thus, a threshold element should consist of an adder and threshold comparator. Hence, a threshold element is an analog-discrete or, at least, multivalued-binary element. So, all the problems we have in multivalued logics (accuracy, parametrical and noise stability, etc.) evidently spread to threshold logics.

It is naturally to define the complexity of a threshold function as

$$R(f) = k_1 \sum_{i=0}^{n-1} w_i + k_2 T \quad (2)$$

where k_1 and k_2 are coefficients depending on particular implementation.

Let us consider Boolean and threshold representations of carry-look-ahead gates for 4 digits:

$$C_4 = x_3 y_3 + (x_3 + y_3)(x_2 y_2 + (x_2 + y_2)(x_1 y_1 + (x_1 + y_1)(x_0 y_0 + (x_0 + y_0) c_0)))$$

$$C_4 = \text{Sign}(8x_3 + 8y_3 + 4x_2 + 4y_2 + 2x_1 + 2y_1 + x_0 + y_0 + c_0 - 16) \quad (3)$$

The complexity of a Boolean function is usually determined by the number of character entries in the minimum representation.¹ In CMOS-implementation, every character entry corresponds to two (p - and n -) transistors. Then both representations in (3) have close complexity and CMOS-implementation is more preferable by many reasons. Thus, considering possible implementations of threshold elements we should answer the question of their application area.

Recently, new circuits of CMOS threshold elements have been suggested and studied. These are threshold elements on the base of CMOS-invertor with floating gate (v CMOS) and capacity inputs [1,2] as well as threshold elements implemented by output wired

¹ If this representation can be implemented by one complex gate.

CMOS inverters [3,4,5]. The latter are the subject of this work.

Let us consider the circuit in Fig.1a.

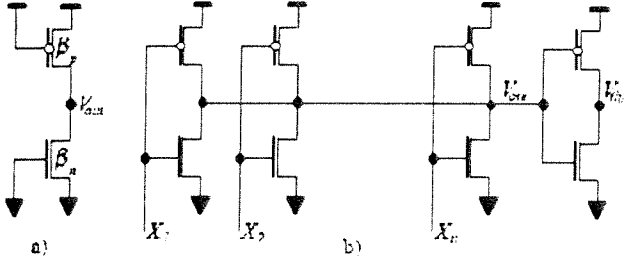


Figure 1: Output wired CMOS inverters.

In the static mode when both transistors are completely open, the output voltage V_{out} is determined by the ratio $\alpha = \beta_n/\beta_p$ and by thresholds of the transistors. V_{out} drops when α increases and grows when α declines. This trivial thing is the base for the idea of a threshold element implemented by output-wired CMOS inverters (Fig.1b). The class of threshold circuits produced by the CMOS couple in Fig.1a will be referred to as beta-driven threshold elements (β DTE).

If, in the simplest case, all the transistors except for those in the output inverter have the same steepness β_0 , it is easy to see that

$$\alpha = \frac{\beta_n}{\beta_p} = \frac{\sum_{i=1}^n x_i}{\sum_{i=1}^n \bar{x}_i} \quad (4)$$

and, if the output inverter threshold is properly selected, the circuit implements a majoritary function of n variables $\text{Sign}(\alpha - 1)$. Indeed:

$$\begin{aligned} \text{Sign}(\alpha - 1) &= \text{Sign}\left(\sum_{i=1}^n x_i - \sum_{i=1}^n \bar{x}_i\right) = \\ \text{Sign}\left(\sum_{i=1}^n x_i - \sum_{i=1}^n (1 - x_i)\right) &= \text{Sign}\left(\sum_{i=1}^n x_i - \frac{n}{2}\right) \end{aligned} \quad (5)$$

The input weights for an arbitrary threshold function can be introduced by multiple magnification of the transistors widths in the respective inverter. The threshold can be changed with respect to average level by incorporating permanently open transistors of appropriate width.

This seemingly exhausts the task of β DTE logical synthesis. But expressions (4.5) make us think a little deeper.

2 Ratio form of threshold functions.

Theorem: Any threshold function $F(X) = \text{Sign}\left(\sum_{j=0}^{n-1} w_j x_j - T\right)$ can be represented as $F(X) = \text{Sign}\left(\frac{\sum_{j \in S} w_j x_j}{\sum_{j \in \bar{S}} w_j \bar{x}_j} - 1\right)$, where S is a certain subset of indexes, such that $\sum_{j \in S} w_j = T$.

Proof:

1. Among the combinations of variables X (hypercube vertices), there is at least one combination A ($x_j = a_j$) such that $\sum_{j=0}^{n-1} w_j a_j - T = 0$ (vertex A lies in the separating hyperplane). If there is no such a combination (vertex), the vertex closest to the separating hyperplane can be introduced into it by a shift and/or rotation of the hyperplane. Furthermore, if there is no such a combination (vertex), the representation of the threshold function is not minimum and the solution of the respective integer programming task with a linear criterion lies along the border of permissible solutions. If A is one of the vertices lying in the separating hyperplane, the index $j \in S$ if $a_j = 1$.

2. $\sum_{j=0}^{n-1} w_j x_j - T$ can be rearranged to give:

$$\begin{aligned} \sum_{j=0}^{n-1} w_j x_j - T &= \sum_{j \notin S} w_j x_j - (T - \sum_{j \in S} w_j x_j) = \\ \sum_{j \notin S} w_j x_j - \sum_{j \in S} w_j (1 - x_j) \end{aligned}$$

$$\begin{aligned} \text{Sign}\left(\sum_{j=0}^{n-1} w_j x_j - T\right) &= \text{Sign}\left(\sum_{j \notin S} w_j x_j - \sum_{j \in S} w_j \bar{x}_j\right) \\ &= \text{Sign}\left(\frac{\sum_{j \notin S} w_j x_j}{\sum_{j \in S} w_j \bar{x}_j} - 1\right) \end{aligned} \quad (6)$$

because $\text{Sign}\left(\sum_{j \notin S} w_j x_j - \sum_{j \in S} w_j \bar{x}_j\right) =$

$$\begin{cases} 1 & \text{if } \sum_{j \notin S} w_j x_j \geq \sum_{j \in S} w_j \bar{x}_j \\ 0 & \text{if } \sum_{j \notin S} w_j x_j < \sum_{j \in S} w_j \bar{x}_j \end{cases}$$

that proves the theorem.

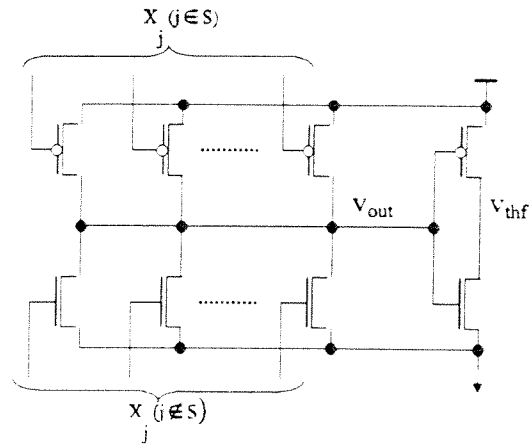


Figure 2: β -driven threshold element.

Let us now consider Fig.2. To provide that V_{inv} is a threshold function, it is necessary that the inverter threshold is appropriately selected and that V_{out} represents $\alpha = \sum_{j \in S} w_j x_j / \sum_{j \in \bar{S}} w_j \bar{x}_j$ for the threshold function which is inverse to the given one. This

condition is met if S is such a subset of indexes that

$$\sum_{j \in S} w_j = T - 1$$

As is easy to see, the circuits in Fig.1a and Fig.2 have two considerable differences:

- a reduced β DTE contains twice less transistors per an input weight unit than an output-wired CMOS inverter;
- the value $\alpha=1$ is determined not by the number of input variables but by the threshold of element firing.

3 Implementability of β -Driven Threshold Elements

Let us return to the question about the difference between β DTE and output-wired CMOS invertors. First, β DTE implementations require one transistor per an input weight unit, unlike a CMOS couple (two transistors) in output-wired CMOS inverter implementations. Second, the latter has $\sum_{j=0}^{n-1} w_j/2$ open p - and n -transistors in the working point (when $\alpha=1$, i.e. when $\sum_{j=0}^{n-1} w_j x_j = \sum_{j=0}^{n-1} w_j/2$) and the minimum $\Delta\alpha = \frac{4}{2 + \sum_{j=0}^{n-1} w_j}$; while β DTE implementation has $\min(T-1, \sum_{j=0}^{n-1} w_j - T + 1)$ open p - and n -transistors (also when $\alpha=1$) and the minimum $\Delta\alpha = \frac{1}{\min(T-1, \sum_{j=0}^{n-1} w_j - T + 1) + 1}$. It follows from the above that the implementability area for output-wired CMOS invertors depends on $\sum_{j=0}^{n-1} w_j$, i.e. ultimately on the number of variables, while that for β DTE depends only on $\min(T-1, \sum_{j=0}^{n-1} w_j - T + 1)$.² This warrants the implementability of β DTE to be specially discussed.

Since β DTE is an analogous element, it is essential to analyze its accuracy and influence of parametrical instability. We will start from a pretty rough estimation. Let β_0 have a relative error $\Delta\beta/\beta$, then the absolute error of α caused by variations of β when $\alpha = 1$ is $\Delta\beta\alpha = 2\Delta\beta/\beta$. Then the output voltage change caused by variations of β should be less than the minimum functional change of the output voltage, i.e. $\Delta_f \alpha \cdot \frac{dV_{out}}{d\alpha} \Big|_{\alpha=1} > \Delta\beta\alpha \cdot \frac{dV_{out}}{d\alpha} \Big|_{\alpha=1}$ or $\min(T-1, \sum_{j=0}^{n-1} w_j - T + 1) \leq 4$. This constraint is essential because it does not depend on $\frac{dV_{out}}{d\alpha}$. For output-wired CMOS invertors, this constraint looks like $\sum_{j=0}^{n-1} w_j < 18$, not depending on the nature of the output circuits. Therefore, the assertion made in [5, p.146, table 3] about getting $\sum_{j=0}^{n-1} w_j \leq 67$ at the cost of two output invertors when $\Delta\beta/\beta=10\%$ looks very doubtful.

²The values T and $\sum w_j$ are estimated in the appendix.

Fig.3 will help us estimate the implementability area more precisely. First of all, let us find out the value of $\frac{dV_{out}}{d\alpha} \Big|_{\alpha=1}$. When $V_{tp} \leq V_{out} \leq V_{dd} - V_{tn}$ in the steady mode, n -transistors have $V_{gs} = V_{dd}$ and $V_{ds} = V_{out} \leq V_{gs} - V_{tn}$. Hence, the n -transistors are not saturated. The same is true for the p -transistors. Then:

$$I_n = \beta_n \left[(V_{dd} - V_{tn})V_{out} - \frac{V_{out}^2}{2} \right];$$

$$I_p = -\beta_p \left[(V_{dd} - V_{tp})(V_{dd} - V_{out}) - \frac{(V_{dd} - V_{out})^2}{2} \right];$$

$$I_n + I_p = 0; \quad \beta_n/\beta_p = \alpha;$$

Let $V_{tn} = V_{tp} = V_{th}$ and $V_{dd} - V_{th} = \Delta V$;

Then, if $\alpha = 1$, we have $V_{out} = V_{dd}/2$ and

$$\alpha = \frac{2\Delta V(V_{dd} - V_{out}) - (V_{dd} - V_{out})^2}{2\Delta V V_{out} - V_{out}^2};$$

$$\frac{d\alpha}{dV_{out}} = -\frac{2\Delta V - 2V_{dd} + 2V_{out}}{2\Delta V V_{out} - V_{out}^2} - \alpha \frac{2\Delta V - 2V_{out}}{2\Delta V V_{out} - V_{out}^2};$$

$$\frac{d\alpha}{dV_{out}} \Big|_{\alpha=1} = -2 \frac{2\Delta V - V_{dd}}{2\Delta V V_{out} - V_{out}^2};$$

$$\frac{dV_{out}}{d\alpha} \Big|_{\alpha=1} = -\frac{2\Delta V V_{out} - V_{out}^2}{4\Delta V - 2V_{dd}} = \frac{4\Delta V V_{dd} - V_{dd}^2}{16\Delta V - 8V_{dd}} \approx 2.3;$$

$$\Delta\alpha = \frac{2\Delta\beta}{\beta}; \quad \Delta\alpha V_{out} \approx \pm 4.6 \frac{\Delta\beta}{\beta}$$

Variations of β affect not only the threshold circuit itself. They also shift the threshold of the output forming inverter. In the singular point of inverter switching curve both the transistors are saturated and

$$V_{in} = \frac{V_{dd} - V_{tp} + V_{tn}\sqrt{\alpha}}{1 + \sqrt{\alpha}} \quad [6, (2.24 \text{ p.66})]. \quad \text{Then}$$

$$\frac{d\alpha}{d\beta_n} = \frac{1}{\beta_p}; \quad \frac{d\alpha}{d\beta_p} = \frac{\alpha}{\beta_p}; \quad \Delta\beta_n = \alpha\Delta\beta_p;$$

$$\frac{dV_{in}}{d\alpha} = -\frac{V_{dd} - V_{tp} - V_{tn}}{2(1 + \sqrt{\alpha})^2 \sqrt{\alpha}} \quad \text{and, hence,}$$

$$\Delta\beta V_{in} = \frac{dV_{in}}{d\alpha} \left(\frac{\alpha\Delta\beta_p}{\beta_p} + \frac{\alpha\Delta\beta_p}{\beta_p} \right) =$$

$$(V_{dd} - V_{tp} - V_{tn}) \frac{\sqrt{\alpha}}{(1 + \sqrt{\alpha})^2} \frac{\Delta\beta}{\beta} \quad (7)$$

As you can see from Fig.3, we are interested in the value of inverter threshold shifted about V_{out} ($\alpha = 1$). When $\alpha=4$, $V_{in} \approx 2v$ and $\Delta\beta V_{in} = 0.66 \frac{\Delta\beta}{\beta}$; when $\alpha=0.25$, $V_{in} \approx 3v$ and $\Delta\beta V_{in} = 0.66 \frac{\Delta\beta}{\beta}$.

Two inequations follow from Fig.3 that determine the relative shifts of V_{out} ($\alpha=1$), inverter threshold

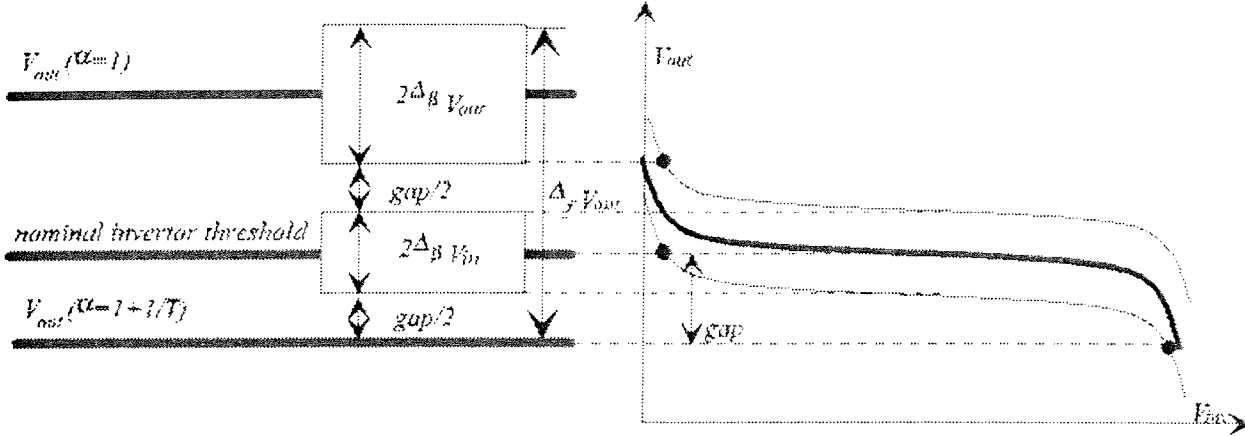


Figure 3: Signal relations on β DTE.

and minimum functional increment α ($\Delta_f V_{out}$) necessary for correct functioning of the threshold element itself and output forming inverter.

$$V_{out}(\alpha = 1) - V_{in} \geq \Delta_\beta V_{out} + \Delta_\beta V_{in} + \text{gap}/2;$$

$$V_{out}(\alpha = 1) + \Delta_\beta V_{out} - \Delta_f V_{out} \leq V_{in} - \Delta_\beta V_{in} - \text{gap}/2$$

Solving this system of inequalities gives the following implementability area:

$$\Delta_f V_{out} \geq 2(\Delta_\beta V_{out} + \Delta_\beta V_{in}) + \text{gap};$$

$$V_{out}(\alpha = 1) - V_{in} \geq \Delta_\beta V_{out} + \Delta_\beta V_{in} + \text{gap}/2 \quad (8)$$

The pessimistic estimation for the gap is 750mV³ and, taking into account what we said above, it follows from (9) that for 10% variation of β

$$\min(T - 1, \sum_{j=0}^{n-1} w_j - T + 1) \leq 1 \quad (9)$$

The result has probably discouraged you. However, we are going to show in the next section that even under this constraint some useful and interesting circuits can be suggested. An essential contribution to (9) is made by the gap, whose impact can be considerably decreased by introducing another forming inverter, as it is done in [5]. The gap estimation in [5,6] assumed that the maximum noise margin for the inverter is attained when the gap is placed between two points where the switching curve derivative is equal to one. Note also that these estimations are made for a symmetrical inverter. The shift of inverter switching threshold changes its characteristics a little. All the analytic estimations obtained above perfectly meet the results of SPICE simulation. So, we can use them with a clear conscience for finding the gap value. The results of SPICE simulation for an inverter with switching threshold $\approx 2v$ ($\alpha \cong 4$) give a gap equal to

³In [6], it is even more, 1v.

360mV (the range of the inverter output voltages is $V_{in} \div V_{dd} - V_{tp}$).

Besides, it is important that the estimation (10) was made for the worst case. It is safe to assume that closely placed transistors have such parameters as carriers mobility, TOX, ΔL , ΔW , etc. deflected with the same sign. There is reason to hope that practically, even when the parameters dispersion is 10%, we can provide $\min(T - 1, \sum_{j=0}^{n-1} w_j - T + 1) \leq 2$.⁴ Works on new super precision technologies [7] give us still more hopes.

4 Some circuit examples

Generally speaking, β DTE is not something absolutely new. Well-known pseudo-NMOS NOR-gate [6] and pseudo-PMOS NAND-gate are nothing more nor less than β DTE. They have the following ratio forms

of their threshold functions: $y = \text{Sign} \left(\frac{\epsilon}{\sum_{j=0}^{n-1} x_j} - 1 \right)$

and $y = \text{Sign} \left(\frac{\sum_{j=0}^{n-1} \bar{x}_j}{\epsilon} - 1 \right)$ respectively. The part of ϵ in the numerator/denominator is played by the weak transistor.⁵

Below we will give a number of circuits obviously useful in practical applications, restricting ourselves by $\min(T - 1, \sum_{j=0}^{n-1} w_j - T + 1) \leq 2$.

4.1 Majority elements

Majoritary elements are very popular due to their broad application in fault-tolerant systems and the possibility of simplifying the circuit implementations by introducing these elements into the functional basis of the synthesis.

⁴Our estimations are considerably more pessimistic than those in [3,4,5]. However, we have made just general estimations in this section and some problems should be solved at the level of physical design.

⁵As far as we know, the implementability of these elements has not been in doubt.

In the notation we have accepted, a 3-input and 5-input majority elements have the following representations:⁶

$$\overline{\text{maj}(x_0, x_1, x_2)} = \text{Sign} \left(\frac{\overline{x_0} + \varepsilon}{x_1 + x_2} - 1 \right);$$

$$\overline{\text{maj}(x_0, x_1, x_2, x_3, x_4)} = \text{Sign} \left(\frac{\overline{x_0} + \overline{x_1} + \varepsilon}{x_2 + x_3 + x_4} - 1 \right);$$

Fig.4 contains the results of SPICE simulation⁷ for a 5-input majority element.

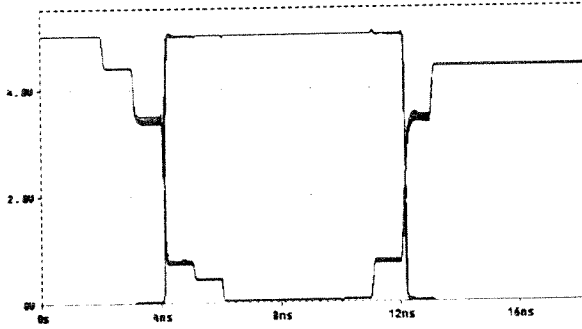


Figure 4: 5-input majority element SPICE simulation.

4.2 C-elements

C-elements are widely used in asynchronous (self-timed) circuits as synchronizing devices. A *C*-element is a device with a feedback (memory). A two-input *C*-element is a three-input majority element with one of the inputs fed by the output. It is easy to see that building a two-input *C*-element takes only 6 transistors while the best one of those known before took 8 transistors. The behavior of a 3-input *C*-element is specified by a threshold function

$$C(x_0, x_1, x_2) = \text{Sign}(x_0 + x_1 + x_2 + 2C - 3) = 1 - \text{Sign} \left(\frac{\overline{x_0} + \overline{C}}{x_1 + x_2 + C} - 1 \right) \quad (10)$$

In this case, we need 7 transistors instead of 10. The results of SPICE simulation for 3-input *C*-elements are given in Fig.5.⁸

As n increases, the threshold and maximum weight of a *C*-element input grow linearly. So, it is hardly possible to get a stable implementation of *C*-elements like (10) when $n > 4$. On the other hand, a *C*-element is a hysteresis element and the feedback signal just commutates the threshold: when $C=1$, the condition for switching is $\bigcup_{j=0}^{n-1} x_j = 0$ and when $C=0$, it is $\bigcap_{j=0}^{n-1} x_j = 0$. These switching functions

⁶ ε here also stands for a weak transistor.

⁷ Monte-Carlo, $n=200$, $\Delta\beta/\beta=10\%$, $\Delta V_{th}/v_{th}=10\%$.

⁸ Monte-Carlo, $n=200$, $\Delta\beta/\beta=10\%$, $\Delta V_{th}/v_{th}=10\%$.

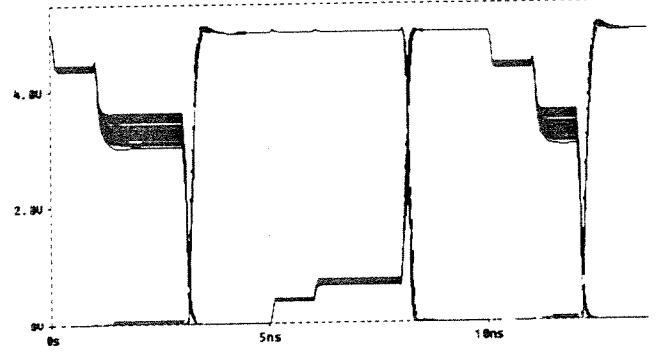


Figure 5: 3-input *C*-element SPICE simulation.

themselves can be implemented by β DTE with unlimited number of inputs that suggests the idea of using the following *C*-element structure: $C(X) = 1 - \text{Sign} \left(\frac{\overline{C} \sum_{j=0}^{n-2} \overline{x_j} + 0.5 \overline{x_{n-1}}}{C \sum_{j=0}^{n-2} x_j + 0.5 x_{n-1}} - 1 \right)$. The corresponding circuit is given in Fig.6. If the commutating transis-

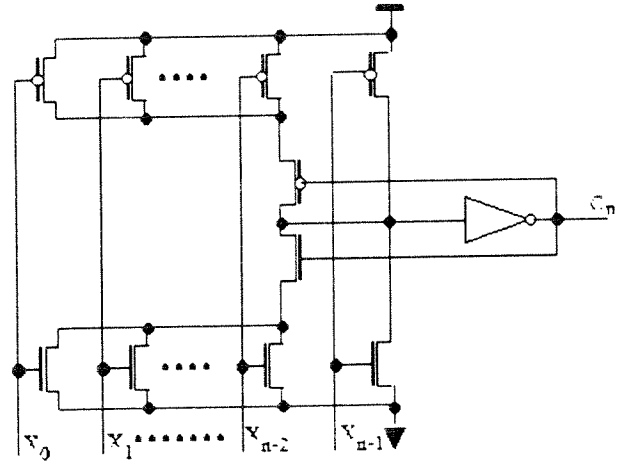


Figure 6: n -input *C*-element.

tors are wide enough, they practically do not affect the circuit functioning. The circuit is workable because the pseudo-NMOS NOR- and pseudo PMOS NAND gates are workable. It requires two transistors per one input. If we increase the number of inputs, only the performance will be affected, because the common capacities will also grow at the sources and drains of the p -transistors.⁹

4.3 Full adder

Threshold representation of a full adder is

$$c_i = \text{maj}(x_i, y_i, c_{i-1}); \quad \sigma_i = \text{Sign}(x_i + y_i + c_{i-1} - 2c_i - 1)$$

⁹ We patented this circuit in 1987 [8]. However, in that time we studied it in terms of electricity, not logics. Its description was published only in Russian.

$$= \text{Sign}(x_i + y_i + c_{i-1} + 2\bar{c}_i - 3);$$

whence it follows that $\bar{c}_i = \text{Sign}\left(\frac{c_{i-1} + \bar{c}_i}{x_i + y_i} - 1\right)$; $\bar{\sigma}_i = \text{Sign}\left(\frac{\bar{c}_{i-1} + \bar{c}_i}{x_i + y_i + \bar{c}_i} - 1\right)$.

The threshold elements themselves can be implemented using 9 transistors. However, there is a question whether it is possible to directly use the signal \bar{c}_i at the input of element $\bar{\sigma}_i$. SPICE simulation of the circuit in Fig.7 lets us give a positive answer to this question. The simulation results are given in Fig.8.¹⁰

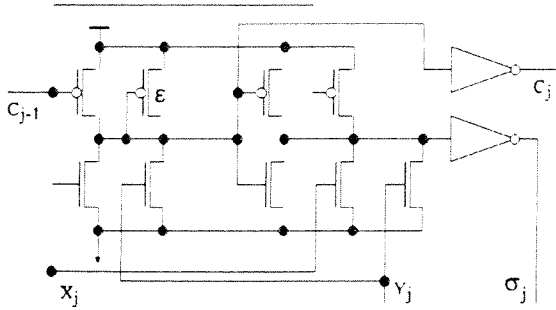


Figure 7: Full adder.

Note that a similar implementation of a full adder in a conventional CMOS circuitry requires 20 transistors (plus two invertors in both cases).

5 Conclusion

We hope we have managed to demonstrate a good logical power of β DTE even with considerable limitations on the threshold value. For example, using 4 transistors and an invertor we can implement any threshold function of 3 variables (xyz , $x(y+z)$, $xy+yz+yz$, $x+yz$, $x+y+z$). Moreover, it is possible to show that any Boolean function of 3 variables can be implemented by a two-cascade circuit using not more than 11 transistors and 2 invertors. In any case, using β DTE considerably expands the basis of logical synthesis. They provide unique possibilities for both low- and high-threshold functions of many variables.

However, nothing is free. When using β DTE instead of CMOS gates, the increase in power consumption is the cost for the extra area. In CMOS gates, power is mostly consumed to charge the capacities under the transistor gate. The contribution of through currents is much smaller and the recharge current from the source is consumed only in one phase. When we use β DTE, everything changes cardinally. Through currents dominate and achieve big values, because n - and p -transistors are fully open in the switching mode. The current is consumed in both phases. The work of through currents can achieve $2.5\tau nJ$, where τ is the switching process duration (when the input variables are changing) in nanoseconds. For a CMOS pair, the

work of recharging the input capacity is $\approx 0.25 nJ$.¹¹ Thus, the loss of power is compensated if using β DTE saves 20τ transistors as compared to CMOS implementation.

One way or the other, if output-wired CMOS invertors as threshold elements are studied and used [3,4,5], studying their modifications certainly makes sense.

6 Appendix

(Threshold function complexity growth estimation)¹²

Any threshold function of $n+1$ variables $\Phi(y, X) = \text{Sign}\left(w_y y + \sum_{j=0}^{n-1} w_j x_j - T\right)$ can be rewritten as $\Phi(y, X) = y\varphi_1(X) + \varphi_2(X)$ where $\varphi_1(X)$ and $\varphi_2(X)$ are threshold functions, such that

$$\varphi_1(X) = \text{Sign}\left(\sum_{j=0}^{n-1} w_j x_j - T_1\right) = \text{Sign}\left(w_y y + \sum_{j=0}^{n-1} w_j x_j - T + w_y\right); \quad (11)$$

$$\varphi_2(X) = \text{Sign}\left(\sum_{j=0}^{n-1} w_j x_j - T\right) \quad (12)$$

and $w_y = T - T_1$.

Let us consider a number of threshold functions which can be represented by Gornier's scheme:

Type 1	Type 2
x_0	x_0
$x_1 + x_0$	$x_0 x_1$
$x_2 + x_1 x_0$	$x_2(x_0 + x_1)$
$x_3 + x_2(x_1 + x_0)$	$x_3(x_2 + x_1 x_0)$
$x_4 + x_3(x_2 + x_1 x_0)$	$x_4(x_3 + x_2(x_1 + x_0))$
.....

It is easy to see that the weights of the inputs, their sums and thresholds form the following Fibonacci rows:

w_{n-1}	1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89 ...
$\sum_{j=0}^{n-1} w_j$	1, 2, 4, 7, 11, 20, 33, 54, 88, 143, 222 ...
T_1	1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89 ...
T_2	1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144 ...

From this it immediately follows that

$$w_{n-1} = \frac{1}{\sqrt{5}} \left[\left(\frac{1+\sqrt{5}}{2} \right)^n - \left(\frac{1-\sqrt{5}}{2} \right)^n \right];$$

$$T_{1,n} = w_{n-1}; \quad T_{2,n} = w_n; \quad \sum_{j=0}^{n-1} w_j = w_{n+1} - 1; \quad (13)$$

¹¹In both cases, the estimations are given by SPICE simulation.

¹²This appendix is important for threshold logics in its own right. Here it helps understand the implementability areas of particular threshold element implementations.

¹⁰Monte-Carlo, $n=200$, $\Delta\beta/\beta=10\%$, $\Delta V_{th}/V_{th}=10\%$.

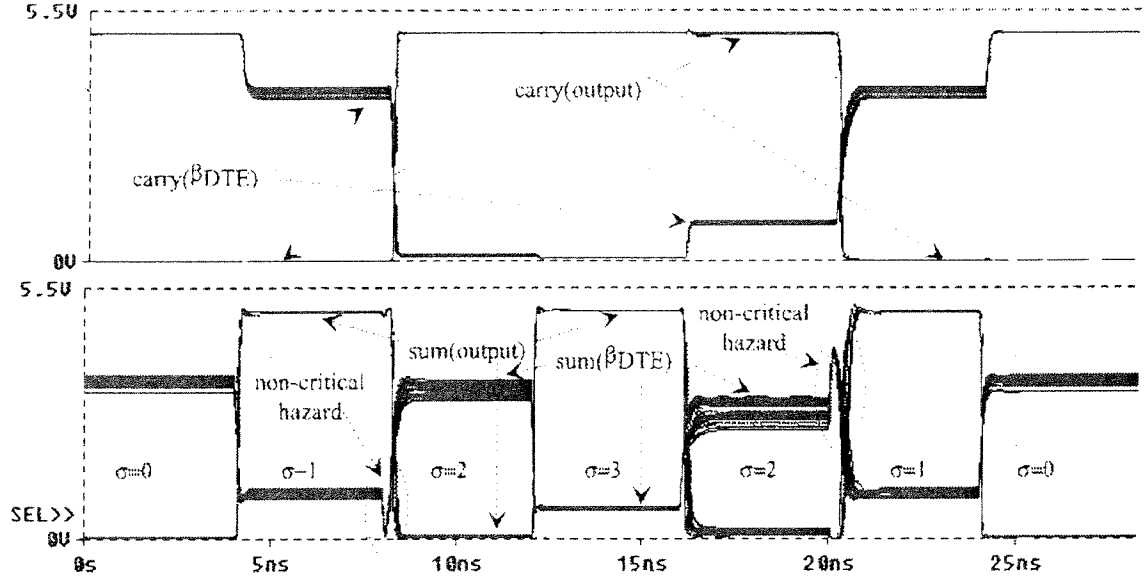


Figure 8: Full adder SPICE simulation.

$$\min \left(T_{2,n} - 1, \sum_{j=0}^{n-1} w_j - T_{2,n} + 1 \right) = w_n$$

The expressions (13) give us the lower border for the upper estimation of complexity of threshold functions of n variables.

On the first glance, it seems that threshold functions determining the threshold properties of binary numbers like

$$F(X) = \text{Sign} \left(\sum_{j=0}^{n-1} 2^j x_j - T \right) \quad (14)$$

have complexity higher than (13). However, (14) is not a minimum expression. Indeed, if $T > 2^{n-1}$, then $F(X) = x_{n-1} F(X|x_{n-1} = 1)$, while if $T \leq 2^{n-1}$, then $F(X) = x_{n-1} + F(X|x_{n-1} = 0)$. So, we get functions which can be represented by Gornier's scheme, i.e. the minimum representation is no worse than (12). This remark is important for threshold functions used in various types of binary transformations.

The question of the upper border is still open. In 60's, an example of a threshold function was found, the optimal implementation of which had minimum input weight more than one¹³. For a big number of variables, such functions can probably produce sequences which majorate those given here.

In any case, the complexity of threshold functions grows exponentially and therefore the possibilities of their technical implementation, in particular creating

universal threshold elements for neural networks, are limited.

References

- [1] Shibata T., Ohmi T. "Neuron MOS binary-logic integrated circuits: Part 1, Design fundamentals and soft-hardware logic circuit implementation." IEEE Trans. Electron Devices, Vol.40, No.5, pp.974-979 (1993).
- [2] Ohmi T., Shibata T., Kotani K. "Four-Terminal Device Concept for Intelligence Soft Computing on Silicon Integrated Circuits." Proc. of IIZUKA'96, pp.49-58.
- [3] Bellabes N. et al. "Ratioed voter circuit for testing and in VLSI processing arrays." Proc. ISCAS'92 (1992).
- [4] Lee Ch.L., Jeh Ch.-W. "CMOS threshold gates and networks for order statistic filtering." IEEE Trans. on Circuits and Systems, Vol.41, No.6, pp.453-456 (1994).
- [5] Bellabes N. et al. "Ratioed voter circuit for testing and in VLSI processing arrays." IEEE Trans. on Circuits and Systems, Vol.43, No.2, pp.143-152 (1996).
- [6] Weste N., Eshraghian K. "Principles of CMOS VLSI Design." Addison-Wesley (1985).
- [7] Ohmi T. "VLSI reliability through ultraclean processing." Proc. of the IEEE, Vol.81, No.5, pp.716-729 (1993).
- [8] Varshavsky et al. "H-latch". USSR Authorship Certificate No.1562964 cl.H 03 K 3/353, priority 16.12.1987, registration 8.01.1990.

¹³Unfortunately, I could not find out that publication. As far as I remember, it was written by Robert Winder, but I may be wrong.