Sergio FRASCA

# THEORICAL DETERMINATION OF THE WEIGHTS OF A THRESHOLD ELEMENT IN THE CASE OF NON LINEARLY SEPARABLE FUNCTIONS

ABSTRACT: A simple procedure to introduce supplementary inputs in a threshold element in order to "separate" non linearly separable boolean functions is given:the weights relative to the inputs so obtained are calculated.

## I-INTRODUCTION

A threshold element is a system with n binary inputs (i.e. I or -I) $v_I, v_2, \ldots, v_n$ and an output, also binary, connected with them by the relation

$$\text{I)} \qquad U = \begin{cases} -I & \text{if } \sum_0^n{}_i\, w_i * v_i < 0 \\ I & \text{if } \sum_0^n{}_i\, w_i * v_i \geqslant 0 \end{cases}$$

where $v_0=-I$, the coefficients $w_i$ are named weights, the real number $w_0$ is named threshold.

It is easy to see that a threshold element, according to the weights and the threshold, can separate various boolean functions; they are only a subset of all possible $2^{2^n}$ boolean functions of n variables. Nevertheless particular techniques have been studied in order to separate all boolean functions (I).

One of these techniques consists in realizing networks of threshold elements. Since the AND and the OR functions are linearly separable and since for inverting a variable it is sufficient to change the sign of the corresponding weight, two-level networks are enough.

Another technique, which is that with which we'll occupy ourselves, is based on the use of only one threshold element, but with a greater number of inputs. In order to realize whichever function, $2^n$-n-I supplementary inputs are necessary. New inputs are particular combinations of the m normal inputs. In these way we make "non linear" the ipersurface of separation.

In this work a procedure is presented for introducing these supplementary inputs and for calculating theorically the weights relative to them.

## 2-WALSH FUNCTIONS AND WALSH TRANSFORMS

It is given here a particular definition of Walsh functions and transforms. It doesn't correspond to those normally used.

Given a domain D of $2^n$ ordered points $p_0, p_I, \ldots, p_{2^n-I}$, let us define on them the set of $2^n$ functions $W_0, W_I, \ldots, W_{2^n-I}$, which we call "Walsh functions" and which have the following properties:

a) they take only the values I and -I

b) they are orthonormal, i.e.

2) $$\sum_{0\,k}^{2^n-1} W_i(p_k) * W_j(p_k) = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases}$$

c) they form a complete set, i.e., however given a finite function $f(p)$ on the points of D, we can write

3) $$f(p_k) = \sum_{0\,i}^{2^n-1} t_i * W_i(p_k)$$

d) being $t_i$ a succession of suitable real numbers which we call "Walsh transform of $f(p)$". In order to obtain $t_i$, we have to put

4) $$t_i = \sum_{0\,k}^{2^n-1} f(p_k) * W_i(p_k)$$

d) the function $W_i$ changes its sign i times in the domain D; one says that $W_i$ has "sequency" i.

We generate the Walsh functions by a procedure derived from that of Lackey and Meltzer (2) and proved by Davies (3).

Let us define in D the n Rademacher functions $R_I, R_2, \ldots, R_n$: $R_i$ shares D in $2^i$ subsequent domains of equal measure in which it takes alternately the values -I and +I.

Let $g_I^{(i)}\ g_2^{(i)} \ldots g_n^{(i)}$ be the representation of i $(0 \leqslant i \leqslant 2^n-I)$

in the Gray code. For semplicity we take the bits of this code from (-I,I) instead of (0,I).

The i-th Walsh function in the point $p_k$ is given by

$$5)\qquad W_i(p_k) = g_I^{(i)} \cdot R_I(p_k) \oplus g_2^{(i)} \cdot R_2(p_k) \oplus \dots \oplus g_n^{(i)} \cdot R_n(p_k)$$

being $\cdot$ the AND and $\oplus$ the EXCLUSIVE OR.

Let us note that Rademacher functions are a subset of Walsh functions; in fact we have

$$6)\qquad R_j = W_{2^j - I}$$

3-DETERMINATION OF SUPPLEMENTARY INPUTS AND CALCULATION OF THE RELATIVE WEIGHTS.

Let $f(v_I, v_2, \dots, v_n)$ be the boolean function (generally not linear) which we have to separate by the threshold element. At any instant the binary inputs $v_I, v_2, \dots, v_n$ form one "disposition" of the $2^n$ possible dispositions. We can characterize this particular disposition by the number that it forms in binary. Let, in general, k be such a number $(0 \leqslant k \leqslant 2^n - I)$ and let $v_I^{(k)}, v_2^{(k)}, \dots, v_n^{(k)}$ be the representation of k. As in § 2, let $g_I^{(i)}, g_2^{(i)}, \dots, g_n^{(i)}$ be the representation of i in the Gray code. As usually, for semplicity, let the v and the g be in (-I,I).

As it was seen in § I, in order to realize whichever function it is necessary to add to the threshold element $2^n - n - I$ supplementary inputs: in all, then, the threshold element must have $2^n - I$ inputs. Let us call these inputs $e_I^{(k)}, e_2^{(k)}, \dots, e_{2^n-I}^{(k)}$ and let us define them in the following way:

$$7)\qquad e_i^{(k)} = v_I^{(k)} \cdot g_I^{(i)} \oplus v_2^{(k)} \cdot g_2^{(i)} \oplus \dots \oplus v_n^{(k)} \cdot g_n^{(i)}$$

Let us observe that the $v_i$ are a subset of $e_i$ and we have

$$8)\qquad e_{2^j - I} = v_j$$

$e_0$ is always -I (and then it is not an actual input).

Let us represent the f in a domain D of $2^n$ points $p_0, p_I, \dots, p_{2^n - I}$, giving at the generic point $p_k$ the k-th value of the truth table of f (it can be I or -I). The three forms

$$f(v_I^{(k)}, v_2^{(k)}, \dots, v_n^{(k)})$$
$$f(k)$$
$$f(p_k)$$

are equivalent.

The weight relative to the input $e_i$ is

$$9) \quad t_i = \sum_{0\,K}^{2^n-1} f(p_K) * W_i(p_K)$$

The threshold is

$$I0) \quad t_0 = \sum_{0\,K}^{2^n-1} f(p_K) * W_0(p_K)$$

In fact, from 3) and 5) we have

$$II) \quad f(p_k) = \sum_{0\,i}^{2^n-1} t_i * [g_1^{(i)} \cdot R_1(p_K) \oplus g_2^{(i)} \cdot R_2(p_K) \oplus \dots \oplus g_n^{(i)} \cdot R_n(p_K)]$$

Since

$$I2) \quad R_i(p_k) = v_i^{(k)}$$

from 7) and II) we have, being $e_0 = -I$,

$$I3) \quad f(p_k) = \sum_{0\,i}^{2^n-1} t_i * e_i^{(K)}$$

which can be I or -I. Then we see that the $t_i$ are just the weights and $t_0$ is just the threshold, q.e.d..

BIBLIOGRAPHY

(I) G. V. Pallottino "Optimum Tolerance Adaptive Threshold Logic as Logic Circuits and Pattern Classifiers", CNEN, RT/EL(68) I3

(2) R. B. Lackey, D. Meltzer "A Simplified Definition of Walsh Functions", IEEE Trans. on Computers, February I97I

(3) A. C. Davies "On the Definition and Generation of Walsh Functions", IEEE Trans. on Computers, February I972