

# Optimized design of a 4-bits absolute-value detector based on linear programming

**Zifeng Wang**

Department of Electrical Engineering and Automation, East China Jiaotong University, Nanchang, 33000, China

2021021001000528@ecjtu.edu.cn

**Abstract.** Nowadays, neural signal acquisition systems are constantly developing, and spike classification algorithms have been widely studied and concerned. This paper designed a practical spike detection circuit, named as the absolute value detector. Project is committed to adopting only simple gate circuits, by using Morgan's theorem to optimize the circuit structure, so that we can ensure that the detector has the following several advantages, simple and beautiful, easy to understand, powerful performance. In addition, this paper also considered the optimization of performance including latency and energy consumption. The increase of Vdd will increase the energy consumption and reduce the delay, while the increase of size will reduce the energy consumption and increase the delay. Using MATLAB software for linear programming, Under the condition of a 1.5-fold increase in the delay, and then adjusting Vdd and size, energy consumption was down by 78 percent.

**Keywords:** Absolute-Value Detector, Linear Programming, Energy Consumption Optimization.

## 1. Introduction

With the development of electronic technology, all electronic products are rapidly updated while the demand grows rapidly, increasing the circuit speed and reducing the circuit energy consumption has become a hot topic of research [1]. Nowadays, due to the limitations of Moore's law, researchers used to rely on reducing the size of the chip to obtain smaller energy, and the use of the delayed method is no longer effective. Therefore, the use of better circuit design to achieve this goal has become the main research direction of today's researchers [2]. The 4-bit absolute value detector is the main basic unit of the neural signal acquisition system, it is also widely applied in the chip's ALU unit, so the optimization of the 4-bit absolute value detector is of great practical value, which will greatly improve the chip's operational efficiency. Although this is only a small part of the chip, the hard work of researchers can create a higher speed computing chip [3]. This paper focuses on the optimization of the circuit topology, the calculation of delay and energy, and the optimization of performance.

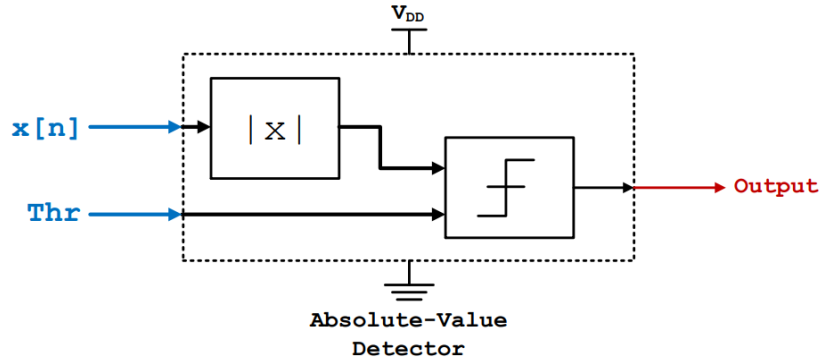
In the next session, it will present the design and optimization of the circuit topology, calculation of delays, sizing, and power supply (Vdd) optimization. Specifically, the second part is about the design of the circuit topology, which consists of a magnitude calculator and a comparator. The third part is about the calculation of delay. First, the circuit topology of the second part is utilized to determine its critical path delay, and then the logic effort formula is used to calculate its circuit delay and the detailed size of each logic gate. The last part is about energy optimization. In total, there are three different

approaches, size only optimization, voltage only optimization and size plus voltage simultaneous optimization to achieve reduced energy consumption.

## 2. Circuit model and the design process

### 2.1. General idea

This project ultimately needs to design a 4-bit “absolute value detector” with a minimum delay energy 50% (1.5 times) longer than the minimum delay [4]. Because we can have a better energy performance while the delay is not significantly increased as a static logic circuit [5][6]. This is an important characteristic to consider for its use in reversible neural signal processing arithmetic and logic units (ALUs) [7]. Where “delay” refers to the propagation delay on the critical path, because the delay of other paths is less important, and “energy” refers to the total energy obtained from the VDD under a given input probability distribution [8]. In this paper, gate size and supply voltage scaling are used as variables to optimize the energy consumption. The final designed circuit is divided into two main parts, the first part derives the magnitude value of the 4-bit random variable input and the second part compares the magnitude value with the threshold value and outputs a high level if the threshold value is exceeded, otherwise a low level is output. The basic framework diagram is shown in figure 1 [9].



**Figure 1.** Basic framework diagram [9].

### 2.2. Schematic of magnitude calculator circuit and the design process

First, I list the binary original code corresponding to 4-bits randomly input binary complement in the truth table, and then simplify the simplest expression with Karnaugh map. The truth table and the simplest expression are as follows in table 1.

**Table 1.** Truth table.

A3A2A1A0	Y2Y1Y0	A3A2A1A0	Y2Y1Y0
0000	000	1000	000
0001	001	1001	111
0010	010	1010	110
0011	011	1011	101
0100	100	1100	100
0101	101	1101	011
0110	110	1110	010
0111	111	1111	001

$$Y_2 = \overline{A_3}A_2 + A_2\overline{A_1}A_0 + A_3\overline{A_2}A_1 + A_3\overline{A_2}A_0 \quad (1)$$

$$Y_1 = A_1\overline{A_0} + \overline{A_3}A_1A_0 + A_3\overline{A_1}A_0 \quad (2)$$

$$Y_0 = A_0 \quad (3)$$

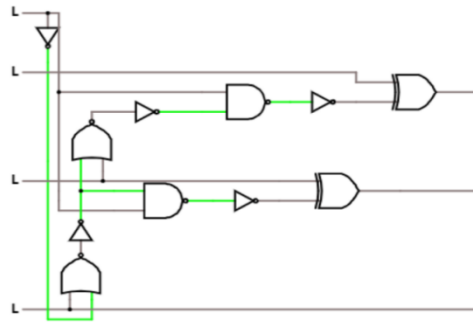
In order to improve the performance of the circuit, first, the goal of my design is to use fewer gate circuits and share more common parts that can be used together. My method is to change the expression by using Morgan's theorem to get the desired expression; Second, it been known that AND, OR and XOR are not easy to be generate, so this paper transforms them into NAND, NOR which are easy to get [10]. Therefore, it get the final expression as follows:

$$Y_2 = ((\overline{A_0} + A_3 + A_2)A_0) \oplus A_1 \quad (4)$$

$$Y_1 = ((\overline{A_0} + A_3)A_0) \oplus A_2 \quad (5)$$

$$Y_0 = A_0 \quad (6)$$

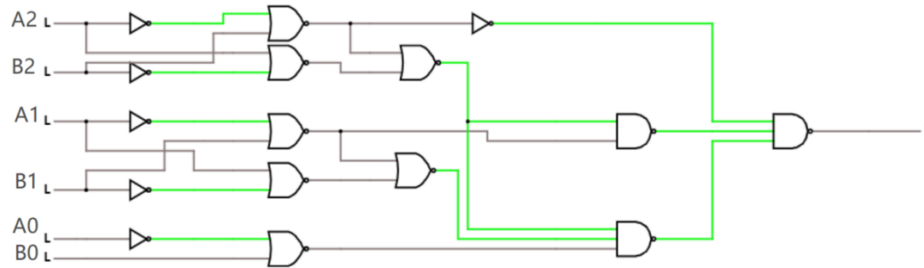
According to the expression, the magnitude calculator circuit is shown in Figure.2. This magnitude calculator includes several NOR, NAND, XOR and inverter.



**Figure 2.** Schematic of magnitude calculator circuit.

### 2.3. Schematic of 3-bits comparator circuit and the design process

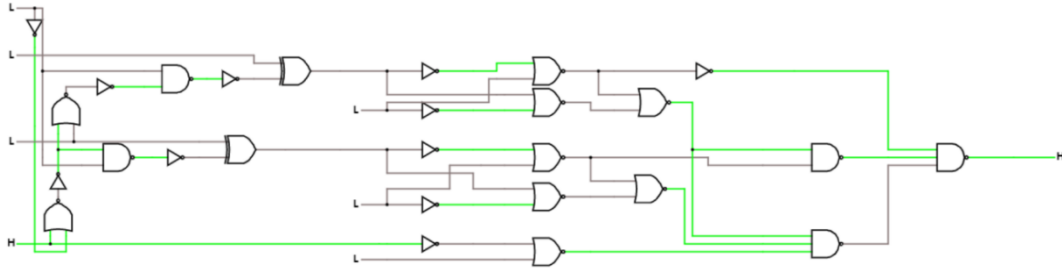
In this module, I have designed a three-bit comparator which needs to compare the amplitude calculated(A2A1A0) in the previous stage with the threshold(B2B1B0) value and outputs a high level if it is greater than the threshold value, otherwise it outputs a low level. I used the same method by first listing the output expression and building it out in a circuit diagram with the following expression. Finally, we can get the following comparator circuit diagram Figure.3. This is a 3-bit comparator circuit with a critical path number of 5 stages from A2 to Out.



**Figure 3.** Schematic of 3-bits comparator circuit.

### 2.4. Total circuit schematic

We connect all the circuit topology above to get the final 4-bits absolute value Detector. The total circuit schematic as shown in figure 4 below. This total circuit is in the operating state with the input of 0001.

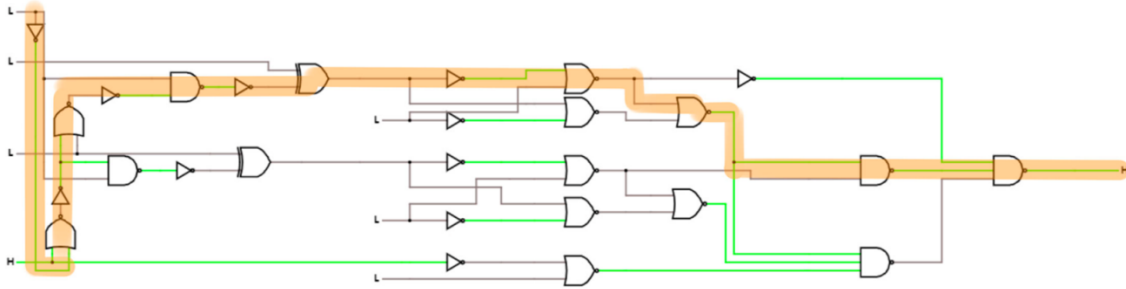


**Figure 4.** Total circuit schematic.

### 3. Critical Path Analysis

#### 3.1. Detect the critical path

The critical path is a complete conduction circuit with the most stages. We can find that the critical path is as shown in the figure 5 below.

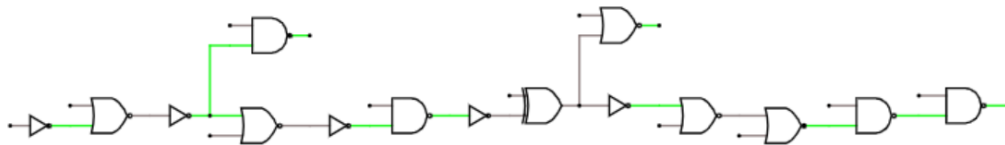


**Figure 5.** Critical path with yellow marked.

Figure 5 shows the overall schematic design of the 4-bits absolute value comparator with the critical path indicated in yellow. The total number of stages in the critical path is 13 stages. We will calculate the critical path delay in the next section.

#### 3.2. Delay calculation

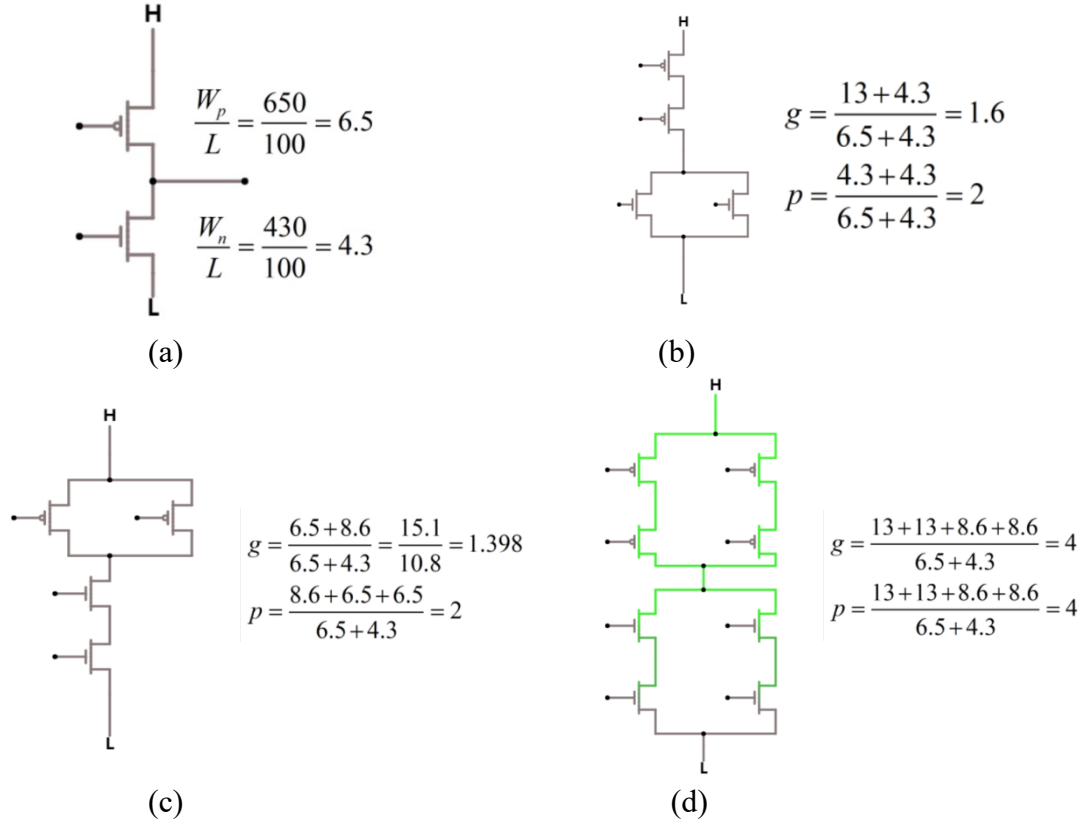
According to Figure 5 in the previous section, there are a total of 13 phases for the critical path from the input port to the output port. We assume that the capacitance of an inverter as a unit of the  $C_{in}$ , and we assume that the output load is a 32  $C_{in}$  load. Critical path circuit, as shown in Figure 6.



**Figure 6.** Critical path circuit schematic.

Before the input port there is a non-gate as an input load with a capacitance of  $C_{in}$  and after the output port there is a capacitive load of 32 times  $C_{in}$ . On this basis, the circuit designed in this project uses a critical path containing 13 stages with an output load that is 32 times the input load. Also to be taken into account during the delay calculation is the fact that there are 2 branches in this critical path. In this paper, it is assumed that  $W_p$  is equal to 650 nm and  $W_n$  is equal to 430 nm for unit size inverter. then this paper can get  $W_p : W_n$  is equal to 1.5. drawing the specific gate circuit diagrams and analyzing

them, it is not difficult to get their parasitic and logical efforts. The gate circuit is analyzed as shown in follow figures 7.



**Figure 7.** Gate circuit.

For the critical path, this paper calculates the overall logical effort of this path in the following way (the unit  $i$  in the following equations, represents the  $i$ th level gate circuit):

The first part is the logical effort generated on the path. The specific calculation formula is as follows:

$$G = \prod g_i = 71.93 \quad (7)$$

The degree of total ladder was calculated using the ratio of the terminal load capacitor size to the front load capacitor size:

$$H = \frac{C_{out}(path)}{C_{in}(path)} = 32 \quad (8)$$

The second part is the logical effort generated on the branch. The specific calculation formula is as follows:

$$B = \prod \frac{g_i(\text{Upper branch}) + g_i(\text{Lower branch})}{g_i(\text{Lower branch})} = 4.88 \quad (9)$$

Ultimately, the logical effort of the entire critical path is as follows:

$$F = GHB = 11232.59 \quad (10)$$

The total number of stages  $N$  is as follows:

$$N = 13 \quad (11)$$

When the logical effort at the same value at each stage, the delay can get minimum value, the logical effort of stage is obtained by dividing the total logical effort by N times:

$$f^* = F^{\frac{1}{N}} = 2.049 \quad (12)$$

Using following formula, the lowest delay for the entire critical path is obtained as follows:

$$D_{\min} = \sum P_i + f^* \times N = 49.64 \quad (13)$$

In addition, we need to calculate the data in the table below, the size, delay, and energy consumption of each stage, for the subsequent optimization work.

The specific calculation formula is as follows (the default Vdd value is equal to 1):

$$size_i = C_i = g \times b \times \frac{C_{i+1}}{f^*} \quad (14)$$

$$delay_i = g_i \times h_i + p_i \quad (15)$$

$$h_i = \frac{C_{i+1}}{C_i} \quad (16)$$

$$E_i = C_i \times Vdd^2 = size \quad (17)$$

Specific data are shown in table 2.

**Table 2.** Detail datas of each stage.

stage	1	2	3	4	5	6	7	8	9	10	11	12	13
Gate	INV	NOR	INV	NOR	INV	NAND	INV	XOR	INV	NOR	NOR	NAND	NAND
g	1	1.6	1	1.6	1	1.4	1	4	1	1.6	1.6	1.4	1.4
p	1	2	1	2	1	2	1	4	1	2	2	2	2
size	1	2.05	2.62	2.87	3.67	7.53	11.01	22.56	4.45	9.11	11.66	14.94	21.86
h	2.05	1.28	1.10	1.28	2.05	1.46	2.04	0.2	2.04	1.28	1.28	1.46	1.46
delay	2.05	4.048	2.1	4.048	3.05	4.044	3.04	4.8	3.04	4.048	4.048	4.044	4.044
energy	1	2.05	2.62	2.87	3.67	7.53	11.01	22.56	4.45	9.11	11.66	14.94	21.86

### 3.3. energy calculation

The energy consumption of the whole critical circuit is the accumulation of the energy consumption of each circuit stage including the energy consumption of the end loads, and it is not difficult for us to calculate the total energy consumption.

$$E = \sum E_i = 147.33 \quad (18)$$

## 4. performance optimization

Based on our calculations above, we got the minimum delay for the circuit design, but we have not optimized the energy yet, we decided to sacrifice our delay to reduce the total energy consumption, the maximum delay we can get is 1.5 times our current minimum delay and we assumed that the maximum supply voltage is 1.

### 4.1. Only Vdd optimization

Using expression shown below, we can obtain the voltage value after only optimized Vdd:

Initial condition:

$$D = K \frac{Vdd}{(Vdd - V_T)^2}, Vdd = 1V, V_T = 0.2V \quad (19)$$

Computational process:

$$\frac{D}{1.5D} = \frac{K \frac{1}{(1-0.2)^2}}{K \frac{Vdd'}{(Vdd'-0.2)^2}} \quad (20)$$

Final result:

$$Vdd' = 0.776, E = CVdd'^2 = 88.72, D = 74.46 \quad (21)$$

With a delay time of 1.5 times its minimum, only optimizing Vdd reduces the energy consumption to 88.72, a reduction of about 39.8%.

#### 4.2. Size optimization

This part of the optimization is to adjust the size in order to reduce the energy use, and the optimized data for the specific dimensions are shown in Table 3 (L will be calculated later).

**Table 3.** Size Optimization data.

stage	1	2	3	4	5	6	7	8	9	10	11	12	13
size	1	1	1	1	1	1	1	1	1	1	1	1	L
g	1	1.6	1	1.6	1	1.4	1	4	1	1.6	1.6	1.4	1.4
h	1	1	1	1	1	1	1	1	1	1	1	L	$\frac{32}{L}$
p	1	2	1	2	1	2	1	4	1	2	2	2	2

The method used is to first set the size of all stages before the last stage to 1, and invert the size of the last stage by calculating the delay as follows:

$$D' = \sum g_i \times h_i + p_i = 38.8 + 1.4L + \frac{44.8}{L} = 1.5D = 74.46 \quad (22)$$

$$L = 24.14 \quad (23)$$

$$E = CVdd^2 = (12 + 24.14 + 32) \times 1 = 68.14 \quad (24)$$

With a delay time of 1.5 times its minimum value, only optimizing the size reduces the energy consumption to 68.14, a reduction of about 46%.

#### 4.3. Vdd and size optimization together

At this part, we need to optimize both Vdd and size, but the increase of Vdd will increase the energy consumption and reduce the delay, while the increase of size will reduce the energy consumption and increase the delay. So we use the linear programming to find the optimal solution. The conditional formulas for the linear programming are as follows:

Objective function:

$$E_{\min} = 32Vdd^2 + \sum_i^N Size_i Vdd^2 \quad (25)$$

Constraint condition:

$$\begin{cases} 1 \leq size \leq \frac{C_{load} = 32}{f^*} \\ 0.776 \leq Vdd \leq 1 \\ k \frac{Vdd}{(Vdd - 0.2)^2} + \sum_{i=1}^N g_i \frac{Size_{i+1}}{Size_i} + p_i \leq 2.5 Delay_{\min} \end{cases} \quad (26)$$

By optimizing both Vdd and size by linear programming using MATLAB software, we can finally get a reduction in energy consumption to 31.94, a reduction of about 78%.

#### 4.4. Summary of the optimization section

It is not difficult to find that the best effect is to optimize both Vdd and size at the same time. We will show the results of the above three parts together in Table 4.

**Table 4.** Optimization detail data.

VDD	Critical path delay	Total energy
1V(Dmin)	tp_IN→OUT = [49.64] FO4(1V)	E = [147.33] Eu(1V)
0.776V(VDD only)	tp_IN→OUT = [74.46] FO4(1V)	E = [88.72] Eu(1V)
1 V(Sizing only)	tp_IN→OUT = [74.46] FO4(1V)	E = [68.14] Eu(1V)
0.776V(Both)	tp_IN→OUT = [74.46] FO4(1V)	E = [31.94] Eu(1V)

It can be seen from the above table that it sacrifices 50% of the minimum delay, and simultaneously optimize Vdd and size, which can effectively reduce the energy consumption by 78%. This is very worthwhile.

## 5. Conclusion

The absolute value detector designed in this paper is divided into absolute value acquisition module and comparator module. In this project, only simple NOR gates, NAND gates, and XOR gates are used to implement the circuit function and optimization of expression and circuit structure is done. In addition, the optimization of performance (delay and energy consumption) is also considered in this paper. In this paper, When calculating the delay of its entire circuit, we only consider the delay of its critical path and ignore the delay of other paths, because the delay of other paths is less important. After rigorous calculations, we found that using MATLAB software for linear programming, the energy consumption decreased by 78% with a 1.5 times increase in delay while adjusting Vdd and size. With the above calculations we can also summarize the following features that allow us to create a static logic circuit that consumes less energy without a significant increase in delay, a feature that can be fully used in reversible neural signal processing algorithms and logic units. Overall, the absolute value detector designed in this project performs well and is optimized beyond expectations.

## References

- [1] Dreslinski R G, Wieckowski M, Blaauw D, et al. Near-threshold computing: Reclaiming moore's law through energy efficient integrated circuits. *Proceedings of the IEEE*, 2010, 98(2): 253-266.
- [2] Yuan J, Svensson C. High-speed CMOS circuit technique. *IEEE journal of solid-state circuits*, 1989, 24(1): 62-70.
- [3] Cirit M A. Transistor sizing in CMOS circuits. *Proceedings of the 24th ACM/IEEE Design Automation Conference*. 1987: 121-124.
- [4] Yuan M. An Absolute-value Detector with Threshold Comparing for Spike Detection in Brain-machine Interface. *Journal of Physics: Conference Series*. IOP Publishing, 2021, 2113(1): 012038.
- [5] Dao H Q, Zeydel B R, Oklobdzija V G. Energy minimization method for optimal energy-delay extraction. *ESSCIRC 2004-29th European Solid-State Circuits Conference (IEEE Cat. No. 03EX705)*. IEEE, 2003: 177-180.
- [6] Subramanyam K, Shaik S, Vaddi R. Tunnel FET based low voltage static vs dynamic logic families for energy efficiency. *18th International Symposium on VLSI Design and Test*. IEEE, 2014: 1-2.



- [7] Chacko J B, Whig P. Low delay based full adder/subtractor by MIG and COG reversible logic gate. 2016 8th International Conference on Computational Intelligence and Communication Networks (CICN). IEEE, 2016: 585-589.
- [8] Dong X, Jing B, Yang X. Improved Design of a 4-bit Absolute-Value Detector Using Simplified Chain Carry Adder. Journal of Physics: Conference Series. IOP Publishing, 2021, 2113(1): 012043.
- [9] Yang G. Optimized Design of a 4-bits Absolute-Value Detector. Highlights in Science, Engineering and Technology, 2023, 31: 224-231.
- [10] Wang Q. A 93.56 FO4 (1V), 141.36 Eu (1V) 4-bit Absolute-Value Detector. Highlights in Science, Engineering and Technology, 2022, 27: 457-464.