Unified Logical Effort- Delay Minimization Method in Logic Paths with RC Interconnect

Bujjibabu. P[1], Satyanarayana. V[2], GNPK Mahalakshmi. E[3]

1. Associate Professor ECE Department, AEC, Surampalem
2. Associate Professor ECE Department, AEC, Surampalem
3. PG Student, Al-AMEER College of Engineering & Tech

Abstract

In modern applications fast processors are needed to avoid wasting time on waiting. It needs to talk about fast processors build with fast logic gates, which introduces the main task in designing CMOS circuits: how to get fast logic gates or how should the logic gates´ transistors be designed to achieve the greatest speed or to obtain the least delay? In this paper a method will be introduced to get the answer of these questions. It is called ***Unified Logical Effort (ULE)****[1] The Unified Logical Effort is an easy way of delay evaluation and minimization in CMOS circuits. It is an extension of the Logical Effort model, which was first introduced by Sutherland [2],[3]. This method considers only the delay caused by the logic gates and neglect on-chip wires. However the circuits continue to scale, so that the delay of wires becomes not negligible anymore and the Logical Effort needs improvement. With the method of Unified Logical Effort the logic gates as well as the wires are taken into consideration to evaluate the delay and then to minimize it.*

1. Introduction

Nowadays time is very valuable. Every second costs money. Everything is getting faster and faster: trains, cars, mobile phones… and mainly processors. Fast processors are needed to avoid wasting time on waiting for loading an internet page or the execution of an instruction. Talking about fast processors means talking about fast logic gates, which introduces the main task in designing CMOS circuits: how to get fast logic gates? Or how should the logic gates´ transistors be designed to achieve the greatest speed or to obtain the least delay?

The Unified Logical Effort method comprises two steps:

Delay evaluation
Delay optimization

2. Delay evaluation of a logic gate

Model of a logic gate: The inner structure of the inverter is shown figure.The inverter is composed of two transistors:

- **p-mos** transistor
- **n-mos** transistor

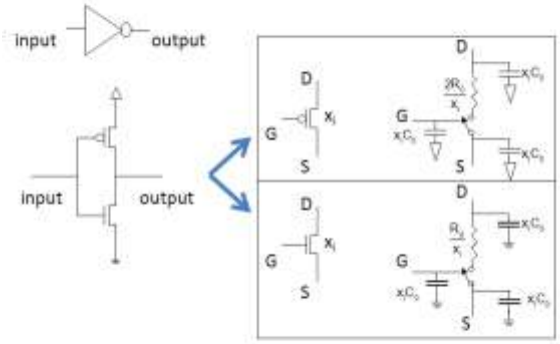


Fig 1: Evaluating the delay by considering the wires

Each transistor can be modelled with 3 capacitances (a gate capacitance, a drain capacitance and a source capacitance) and 1 resistance. The values of these parameters depend on transistor´s width. If an n-mos transistor has the width W = xi.W_0, its capacitances are equal to ***xi .C_0*** and its resistance to ***R_0/ xi.*** C_0 and R_0 are the capacitor and resistor values of the minimum sized inverter ($W = W_0$ and $xi = 1$). With the same width $W = xi.W_0$ the p-mos transistor has the same capacitances but the double resistance as the n-mos transistor because the holes are twice as slow as the electrons. Both models connected together present the following inverter model, which is called **The General RC Inverter Model.** In general each logic gate has

- **an input capacitance *Ci* :** the capacitance of the transistor gates connected to the input
- **an output resistance *Ri* :** pull-down resistance *Rdi* or pull-up resistance *Rui* depending on which switch conducts
- **a parasitic capacitance *Cpi* :** due to the inner capacitances
- **a load Capacitance *Cout* :** the capacitance that the gate has to drive

The values of these parameters depend on transistor´s width. Every logic gate is defined as a scaled version of a *template circuit*, which is the minimum sized symmetric inverter with the minimal width $W = W_0$, input capacitance $Ci = C_0$, output resistance $Ri = Rui = Rdi = R_0$ and parasitic capacitance $Cpi = Cp_0$. Thus the quantities of each logic gate are related to the template parameters and the scaling factor *xi* as

$$Ri = Rui = Rdi = R_0 xi$$

$$C_i \propto C_0\, xi$$

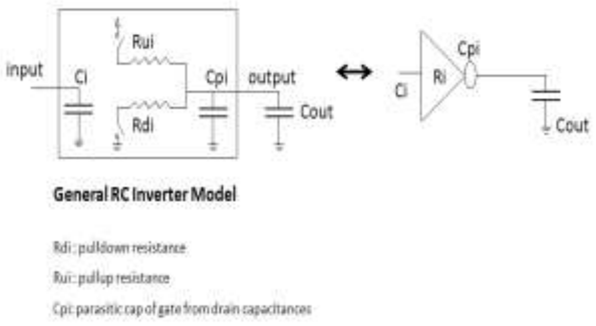


Fig 2: RC-Inverter Circuit Model

Scaling the template means scaling the transistors ´widths by the factor *xi*. As shown in the Figure 2 the capacitances and the resistance of the transistor are respectively proportional and inversely proportional to the transistor width. The input capacitance *Ci* of a logic gate is driven by the previous logic gate. Depending on its load the load capacitance *Cout* and the parasitic capacitance *Cpi* may be respectively charged or discharged through the pull-up or the pull-down resistance. Charging and discharging capacitors through resistors take time, which represents the time delay. That means the delay depends on the output capacitance, the parasitic capacitance, the output resistance and the input capacitance. The delay is comprised of two components:

□ a fixed part caused by the parasitic capacitance called *the parasitic delay p* [4]

□ a part caused by the output capacitance, resistance and the input capacitance called *the effort delay f*

The sum of the two parts gives the total delay:

$$d=f+p$$

The effort delay is also comprised of two components:

□ a part caused by the load capacitance called *the electrical effort h* [5]

□ a part caused by the input capacitance and the output resistance called *the logical effort g*[6]

The effort delay of the logic gate is the product of these two factors [4]

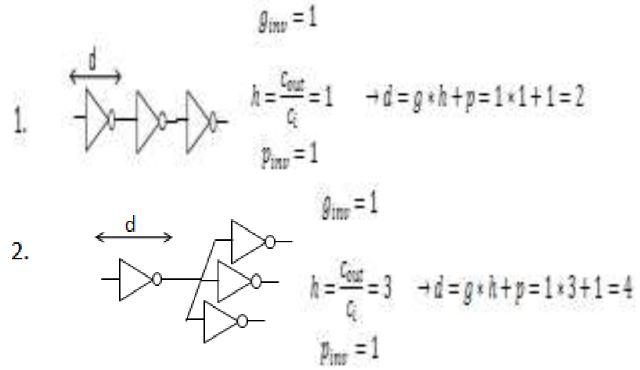
$$f=g*h$$

The logical effort g quantifies the contribution of the logic gate´s topology to the delay. It is independent of the transistors´ size in the circuit. Because the inverter is the simplest logic gate, it drives loads best. The other logic gates have more transistors, some of which are connected in series, increasing the output resistance and hence the delay. The electrical effort h captures the effect of the load capacitance on the delay considering he ratio of driving capabilities and leads to drive the input capacitance. It is defined by:

$$h = Cout/Ci$$

So the basic equation of the total delay through a single logic gate is $d=g*h+p$

Examples:


$g_{inv}=1$
1.
d
$h=\frac{C_{out}}{C_i}=1$
$\rightarrow d=g*h+p=1*1+1=2$
$p_{inv}=1$
$g_{inv}=1$
2.
d
$h=\frac{C_{out}}{C_i}=3$
$\rightarrow d=g*h+p=1*3+1=4$
$p_{inv}=1$

However, as the logic gates are getting smaller and smaller, the contribution of the on-chip wires can´t be neglected anymore.

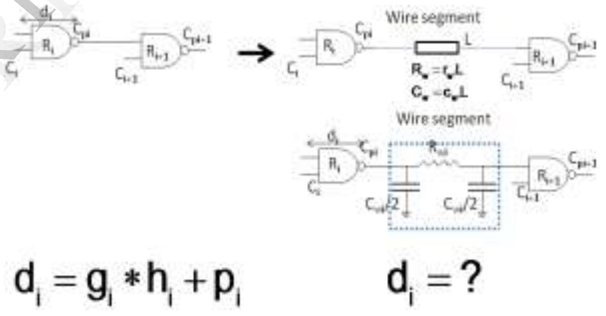


Fig 3: wires delay evaluation

3. Delay evaluation of logic gate with interconnect

Thanks to the Elmore delay model [7] the delay of a circuit comprising logic gates and wires can be easily calculated.

The Elmore delay of the above RC-circuit is defined by:

$$D=R_1\,(C_1+C_2+C_3)+R_2\,(C_2+C_3)+R_3C_3$$

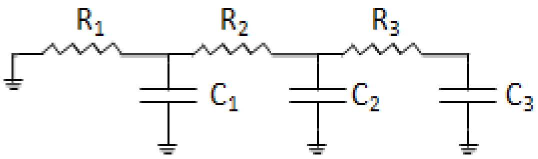


Fig: 4: RC Circuit Model

Analogous to the Elmore delay the absolute delay expression of the following first logic gate is

$$D_i = R_i(C_{pi}+C_{wi}+C_{i+1}) + R_{wi}\ (0.5\ C_{wi}+C_{i+1})$$

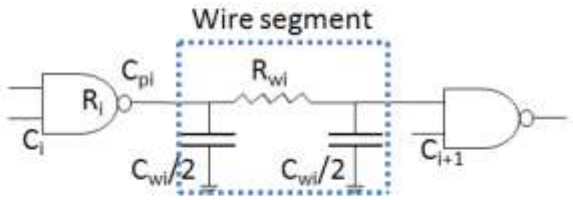


Fig 5: Cascaded logic gates with RC-interconnect

This expression can be rewritten in function of the delay of a minimum sized inverter τ =R$_0$.C$_0$, where R$_0$ and C$_0$ are the output resistance and input capacitance of a minimum sized inverter:

$$D_i = \tau.d_i = \tau\left[\frac{R_i}{R_0}.\frac{(\mathrm{C}_{\mathrm{w_i}}+C_{i+1}+C_{pi})}{C_0}+\frac{R_{wi}}{R_0C_0}(0.5C_{wi}+C_{i+1})\right]$$

The delay *di* normalized with respect to a minimum sized inverter delay τ is defined by:

$$d_i = g_i(h_i+\frac{C_{w_i}}{C_i})+\frac{R_{w_i}.(0.5C_{w_i}+C_{i+1})}{\tau}+p_i$$

where $g_i = \frac{R_i.C_i}{R_o.C_o}$ is the logical effort,

$h_i = \frac{C_{i+1}}{C_i}$ is the electrical effort

and $p_i = \frac{R_i.C_{pi}}{R_o.C_o}$ is the parasitic delay.

The capacitive interconnect effort h_w and the resistive interconnect effort p_w are, respectively,

$$h_{w_i} = \frac{C_{w_i}}{C_i}\ ,\ p_{w_i} = \frac{R_{w_i}.(0.5C_{w_t}+C_{i+1})}{\tau}$$

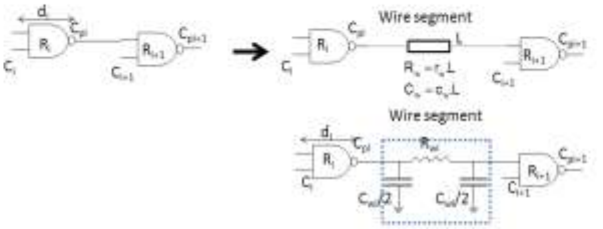
The wire influences the electrical effort of the logic gate with h_w and contributes more delay to the total delay with p_w.

The final expression of the ULE delay of a single logic gate considering the inter-connect is:

$$d=g*\ (h+h_W)+(p+p_W)$$

For an N stage logic path with interconnect the ULE delay is the sum of each delay of the single stage:

$$d = \sum_{i=1}^{N} g_i^{\ *}(h_i+h_{w_i})+(p_i+p_{w_i})$$



$$d_i = g_i * h_i + p_i \qquad d_i = g_i * \left(h_i + h_{w_i}\right) + \left(p_i + p_{w_i}\right)$$

Fig 6: Logic gate delay with interconnect

4. Delay minimization using Unified Logical Effort

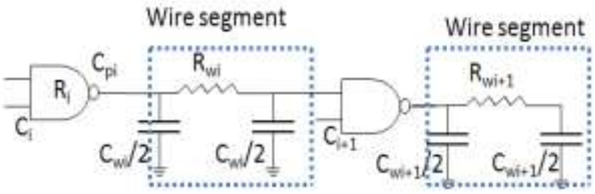


Fig 7: Cascaded logic gates with RC-interconnect

The total delay of the two stages is:

$$d = g_i^{\ *}(h_i+h_{w_i})+(p_i+p_{w_i})+g_{i+1}^{\ *}(h_{i+1}+h_{w_{i+1}})+(p_{i+1}+p_{w_{i+1}})$$

$$d = g_i^{\ *}\left(h_i+\frac{C_{w_i}}{C_i}\right)+p_i+\frac{R_{w_i}.(0,5.C_{w_i}+C_{i+1})}{R_o.C_o}+g_{i+1}\left(\frac{C_{i+2}}{C_{i+1}}+\frac{C_{w_{i+1}}}{C_{i+1}}\right)+p_{i+1}+p_{w_{i+1}}$$

with $C_{i+1} = h_i.C_i$

$$d = g_i^{\ *}\left(h_i+\frac{C_{w_i}}{C_i}\right)+p_i+\frac{R_{w_i}.(0,5.C_{w_i}+C_{i+1})}{R_o.C_o}+g_{i+1}\left(\frac{C_{i+2}}{h_iC_i}+\frac{C_{w_{i+1}}}{C_{i+1}}\right)+p_{i+1}+p_{w_{i+1}}$$

To achieve the least delay the logic gates ´transistors must have the optimal size, that means the derivative of the delay with respect to the logic gate size must be equated to zero,

$$\left(g_i+\frac{R_{w_i}.C_i}{R_o.C_o}\right).h_i = g_{i+1}*\left(h_{i+1}+h_{w_{i+1}}\right)$$

By multiplying by *R0.C0* and using the relationships

$$h_i = \frac{C_{i+1}}{C_i}, C_i = C_o.g_i.x_i \text{ and } R_i = \frac{R_o}{x_i}$$

The optimum condition can be rewritten as following:

$$(R_i+R_{w_i}).C_{i+1} = R_{i+1}.(C_{i+2}+C_{w_{i+1}})$$

That means that the optimum size of gate *i+1* is met when the delay part $R_i + R_{wi} \cdot C_{i+1}$ caused by the logic gate input capacitance is equal to the delay part *Ri*+1 · $C_{i+2} + C_{wi+1}$ caused by the output resistance of the logic gate.

The delay due to the capacitance of gate *i* is defined by:

$$D_{C_i} = (R_{i-1} + R_{w_{i-1}}).C_i = (R_{i-1} + R_{w_{i-1}}).C_o.g_i x_i$$

The delay due to the resistance of gate *i* is defined by:

$$D_{Ri} = R_i.(C_{i+1} + C_{Wi}) = \frac{R_0}{\chi_i}(C_{i+1} + C_{Wi})$$

The total delay of gate *i* is:

$$D_i = D_{Ci} + D_{Ri} + Const$$

So to obtain the least delay the derivatives of the delay components with respect to the logic gate size *xi* have to be equal to 0

$$\frac{\partial D_{Ci}}{\partial \chi_i} = (R_{i-1} + R_{Wi-1}).C_0.g_i$$

$$\frac{\partial D_{Ri}}{\partial \chi_i} = -\frac{R_0}{\chi_i^2}(C_{i+1} + C_{Wi})$$

$$\frac{\partial D_i}{\partial \chi_i} = \frac{\partial D_{Ci}}{\partial \chi_i} + \frac{\partial D_{Ri}}{\partial \chi_i} = 0$$

Solving this equation provides the optimal sizing factor X*iopt*,

$$\chi_{iopt} = \sqrt{\frac{R_o}{(R_{i-1} + R_{Wi-1})} \frac{(C_{i+1} + C_{Wi})}{C_0.g_i}}.$$

Example: The method of ULE is applied to a logic path with nine identical NAND gates with equal wire segments for various lengths shown in the Figure 8. The input capacitance of the first and the last stage are *10.C₀* and *100.C₀*, respectively. The solution range between two limits:

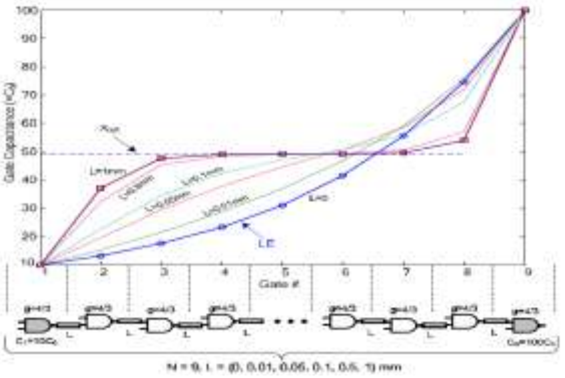


Fig 8: Optimization of ULE sizing (normalized with respect to C$_0$) for a chain of nine NAND gates with equal wire segments for a variety of lengths [8]

- For zero wire lengths the solution converges to LE optimization (delay evaluation and minimization without considering on-chip wires)
- For long wires, the gate size in the middle of the path converges to a fixed value, X*iopt*=50

5. ULE Optimization in paths with branches

The ULE method can be also used in paths including branches or gates with multiple fanout.

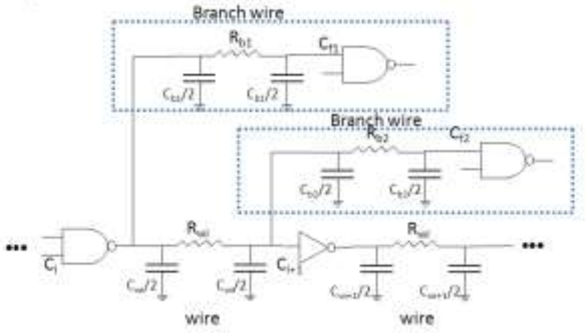


Fig 9: A logic path segment including RC interconnect and two branches

The optimum condition of this case is:

$$(R_{i+1} + R_{Wi-1}).C_i = R_i.(C_{i+1} + C_{Wi} + \underbrace{C_{b1i+1} + C_{f1i+1}}_{\text{Cbf1}} + \underbrace{C_{b2i+1} + C_{f2i+1}}_{\text{Cbf2}})$$

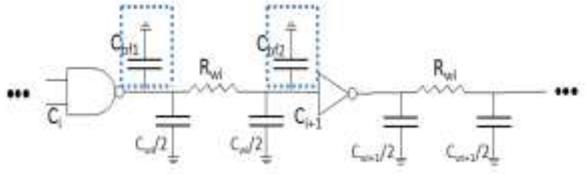


Fig 10: Equivalent circuit with the effective branch and fanout capacitances

Introducing $C_{BF} = \sum_1^n C_{bn} + \sum_1^m C_{fm}$ the optimum condition can be simplified to:

$$(R_{i-1} + R_{wi-1}).C_i = R_i.(C_{i+1} + C_{wi} + C_{BFi})$$

6. Comparison with benchmark circuits

ULE Optimization is compared with the results of Cadence Virtuoso® Analog Optimizer, a numerical optimizer that uses a circuit simulator for delay modeling. The delay of a four-bit carry-lookahead adder is minimized with three methods: LE, ULE and the Analaog Optimizer (AO). All three optimization results are presented in Figure 11.

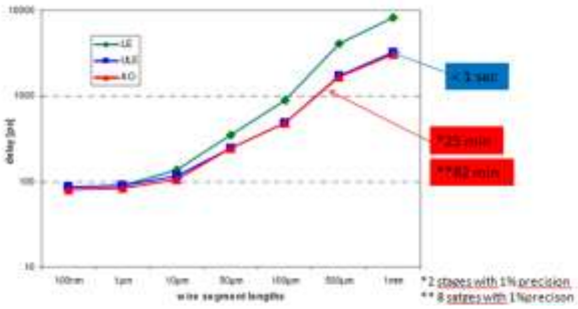


Fig 11: Delay of a carry-lookahead adder for various wire segment lengths after gate size optimization by LE, ULE and Analog Optimizer (AO) [9]

The results of the ULE optimization are very close to the results of the numerical optimizer. But the LE method becomes more and more inaccurate with the increasing wire lengths. Comparing the runtimes the ULE is the fastest method for delay evaluation and minimization.

7. Conclusion

Delay minimization through logic gate sizing is a main task in integrated circuit design. Due to the continuous scaling of the integrated circuits, the inter-connect has to be taken into consideration to get the greatest speed. Thus the Logical Effort model can´t achieve the desired optimization anymore. The Unified Logical Effort has been introduced as an extension of the Logical Effort method solving the problem and considering not only the logic gates but also the on-chip wires to get the least delay. The ULE provides optimum conditions to achieve the optimal gate sizing in logical paths with wires. The delay component caused by the gate capacitance has to be equal to the delay caused by the gate resistance. If the wires´ lengths are negligible the ULE solutions converge to the LE solutions. Compared with the industrial Analogue Optimizer tool the ULE optimization shows in much shorter runtime close results in terms of delay with same accuracy. The ULE has a high potential to be integrated into EDA tools.

References:

[1] Arkadiy Morgenshtein, Eby G. Friedman, Ran Ginosar and Avinoam Kolodny. *Unified Logical Effort- A Method for Delay Evaluation and Minimization in Logic Paths with RC Interconnect*

[2] Ivan Sutherland, Bob Sproull and David Harris, *Logical Effort*: *Designing Fast CMOS Circuits*, Morgan Kaufmann Publishers, 1999.

[3] I.E.Sutherland and R.F.Sproull, "Logical Effort: Designing for Speed on the Back of an envelope, "*Proceeding of the university of California/Santa Cruz Conference on Advanced Research in VLSI (ARVLSI)*,pp. 1-16, 1991.

[4] Ivan Sutherland, Bob Sproull and David Harris, *Logical Effort*: *Designing Fast CMOS Circuits*, Morgan Kaufmann Publishers, pp. 80-82, 1999.

[5] Ivan Sutherland, Bob Sproull and David Harris, *Logical Effort*: *Designing Fast CMOS Circuits*, Morgan Kaufmann Publishers, pp. 5-12, 1999.

[6] Ivan Sutherland, Bob Sproull and David Harris, *Logical Effort*: *Designing Fast CMOS Circuits*, Morgan Kaufmann Publishers, pp. 5-12, 63-69, 1999.

[7] W.C.Elmore, "The Transient Response of Damped Linear Networks with Particular Regard to Wide Band Amplifiers,"*Journal of Applied Physics*, vol. 19, no. 1, pp.55-63, January 1948.

[8] Arkadiy Morgenshtein, Eby G. Friedman, Ran Ginosar and Avinoam Kolodny. *Unified Logical Effort- A Method for Delay Evaluation and Minimization in Logic Paths with RC Interconnect*, p 9.

[9] Arkadiy Morgenshtein, Eby G. Friedman, Ran Ginosar and Avinoam Kolodny. *Unified Logical Effort- A Method for Delay Evaluation and Minimization in Logic Paths with RC Interconnect*, p 12.