

# FPGA INTERCONNECT DESIGN USING LOGICAL EFFORT

Haile Yu, Yuk Hei Chan, Philip H. W. Leong

Department of Computer Science and Engineering  
The Chinese University of Hong Kong  
{hlyu,yhchan,phwl}@cse.cuhk.edu.hk

## ABSTRACT

Logical effort (LE) is a linear technique for modelling the delay of a circuit in a technology independent manner. It offers the potential to simplify delay models for FPGAs and gain more insight into how the parameters affect the result. In this paper, the LE model will be introduced and an application to FPGA interconnect driver sizing described. Simple closed form equations are given for delay, sensitivity of delay to driver size and optimal delay. The results are shown to closely agree with Spice simulations.

## 1. INTRODUCTION

The design of FPGA circuits is often aided by analogue circuit simulation program such as Spice. This allows simulation of the internal delays associated with the device and gives accurate results when the primitive elements and parasitics are correctly modeled. Unfortunately, such simulations often do not offer much intuition into dominant sources of delays and minimal achievable delay, nor do they help in sizing if multiple transistors are considered.

To address this problem, we introduce logical effort (LE) based models for FPGA interconnect. Using this approach, arbitrary circuits can be modeled and closed form analytic equations that model delay can be obtained. Such models can be further used for device sizing, to compare different circuit configurations, determine parameters and study sensitivity.

The logical effort technique was developed by Sutherland and is widely used to size transistors and for comparative circuit studies [1]. Dao et al. used LE to study adder topologies [2]. Hu et al. proposed a family of via-programmable gain-based logic blocks (GLB) which are optimized for performance by choosing the appropriate fabric using LE theory [3]. Keane et al. described how the LE framework could be adapted to size subthreshold circuits [4]. To the best of our knowledge, no previous work in applying LE to interconnect modelling has been reported.

The rest of this paper is organized as follows, in Section 2, we review the logical effort model. Section 3 describes the approach used to calibrate the LE model used in

this paper. In Section 4, the application of LE to FPGA interconnect sizing is given and in Section 5, conclusions are drawn.

## 2. LOGIC EFFORT

A brief review of logical effort, following that of Sutherland et al. [1] is presented in this section. In LE, delay incurred in a logic gate is modelled as being comprised of two components, the fixed *intrinsic delay*  $p$  together with the *effort delay*  $f$  which is proportional to the output load. The total delay  $d$ , is a normalized value and is given by  $d = p + f$ .

The effort delay  $f$  depends on both the topology and load of the logic gate. These are represented by the *logical effort*  $g$  and the *electrical effort*  $h$  respectively, the effort delay being the product of these two factors, i.e.  $f = gh$ .  $g$  represents the ability of a gate to produce output current compared with an inverter, given that the input capacitance is the same as an inverter. Increased loads lead to increased delays and this is represented by  $h$ , defined as  $h = C_{out}/C_{in}$ .

Combining all of the effects described, the normalized delay of a logic gate is given by

$$d = gh + p \quad (1)$$

where  $g$ ,  $h$  and  $p$  are all normalized numbers and relatively independent of technology. To obtain absolute delay,  $d$  is multiplied by  $\tau_{inv}$ , the delay of an inverter driving an identical inverter with no parasitics.

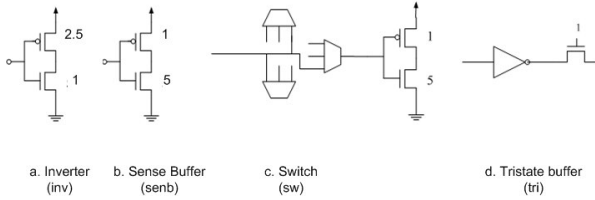
For an arbitrary gate with equivalent resistance, input capacitance and output capacitance parameters  $R_t$ ,  $C_t$  and  $C_{pt}$  respectively, the LE parameters can be directly determined from the layout and transistor sizes:

$$\begin{aligned} d_{abs} &= \tau_{inv}(gh + p) \\ \tau_{inv} &= \kappa R_{inv} C_{inv} \\ g &= \frac{R_t C_t}{R_{inv} C_{inv}} \quad h = \frac{C_{out}}{C_{in}} \quad p = \frac{R_t C_{pt}}{R_{inv} C_{inv}} \end{aligned} \quad (2)$$

where the absolute equivalent resistance and capacitance of an inverter are  $R_{inv}$  and  $C_{inv}$ ,  $\kappa$  is a process dependent proportionality constant and  $d_{abs}$  is the absolute delay.

**Table 1.** Summary of extracted LE parameters. Numbers in parenthesis are for double sized drive strength as used in Section 4.3.

Parameter	$t_r(ps)$	$t_f(ps)$	$t_{av}(ps)$	Normalized
$T_{inv}$			$= g_{inv}$	
$g_{inv}$	18.3	14.3	16.3	(1.15) 1.00
$p_{inv}$	15.9	15.6	15.8	(1.04) 0.967
$g_{senb}$	42.2	16.1	29.1	1.79
$p_{senb}$	69.2	50.8	60.0	3.68
$g_{sw}$	166	28.9	97.6	5.99
$p_{sw}$	116	60.0	88.0	5.40
$g_{tri}$	10.4	15.7	13.0	(0.993) 0.80
$p_{tri}$	27.8	33.0	30.4	(1.96) 1.87



**Fig. 1.** Logical effort primitives.

### 3. CALIBRATION

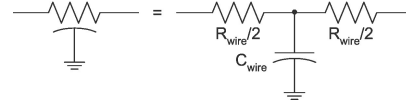
To make the delay estimation accurate, we have the option to calibrate the  $g$  and  $p$  for each primitive gate type individually.

Table 1 summarizes our calibration results for all of the primitives blocks used in this paper. These are: inverter (inv), sense buffer (senb), switch (sw) and tristate buffer (tri). Their circuits are shown in Figure 1. The multiplexers (MUXes) are made of minimum-sized NMOS pass transistors organized in a tree structure. In the experiment 4:1 MUXes are used.

For TSMC 0.18  $\mu\text{m}$  technology, we assume that the interconnect wires are in metal 3, and estimate that a wire spanning one tile is 120  $\mu\text{m}$  in length. Input capacitance is measured by simulating the delay of an inverter driving that gate using HSPICE. A summary of the values thus extracted is given in table 2.

**Table 2.** Table of extracted capacitance and resistance values

Parameter	Value	Normalized
$C_{inv}$	3.43 fF	1.00
$C_{off}$	-	0.167
$C_{senb}$	-	1.54
$C_{sw}$	-	3.71
$C_{wire}$	13.8 fF	3.98
$R_{wire}$	46.6 $\Omega$	



**Fig. 2.** Wire model



**Fig. 3.** Effect of ignoring wire resistance on delay

## 4. SIZING, SENSITIVITY AND OPTIMAL DELAY

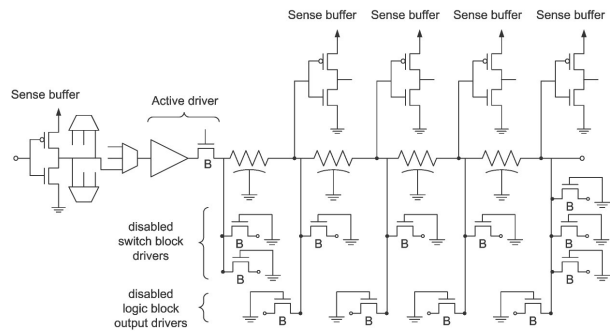
The wire is modeled as the shown in Figure 2, where  $R_{wire}$  and  $C_{wire}$  represent the resistance and capacitance of interconnect wires of one tile.

We simulated the circuit with and without the wire resistance to observe its effect. From Figure 3, we can see that the wire resistance has little effect on delay and thus it is ignored in this study.

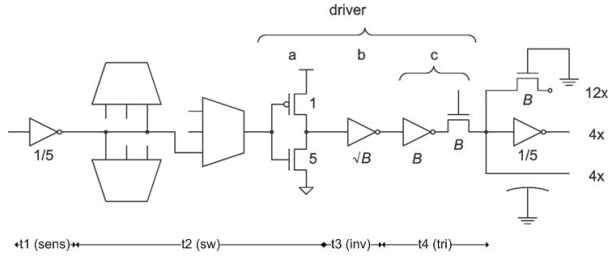
### 4.1. Tristate Interconnect

Figure 4 shows the transistor-level model for the tristate-driver interconnect in reference [5].

In our model, for simplicity, we omit the level restorer of the original design. As a result the circuit for LE modelling is as shown in Figure 5.



**Fig. 4.** Tristate-driver model



**Fig. 5.** Simplified tristate-driver circuit for LE modelling

The equation for the tristate-driver interconnect is similar to that for a single-driver interconnect, the difference being that we have another type of load (the disabled driver with capacitance  $C_{off}$ ) and are driving the wire through a tristate buffer. Here we model the tristate-driver as a normal inverter in series with a NMOS pass transistor, as shown in driver  $c$  of Figure 5. The LE equations are as below.

$$t_1 = g_{senb} \frac{C_{sw}}{C_{senb}} + p_{senb} \quad t_2 = g_{sw} \frac{\sqrt{B}}{C_{sw}} + p_{sw}$$

$$t_3 = g_{inv} \sqrt{B} + p_{inv} \quad t_4 = g_{tri} \frac{C_l}{B} + p_{tri}$$

$$t_{total} = t_1 + t_2 + t_3 + t_4$$

$$= \left( \frac{g_{sw}}{C_{sw}} + g_{inv} \right) \sqrt{B} + \frac{g_{tri} C_l}{B} + g_{senb} \frac{C_{sw}}{C_{senb}} + p_{senb} + p_{sw} + p_{inv} + p_{tri}$$

where  $C_l = 12BC_{off} + 4C_{senb} + 4C_{wire}$  is the output load capacitance.

The minimum is found by making the derivative zero as follows.

$$\frac{dt_{total}}{dB} = 0 \Rightarrow \frac{\left( \frac{g_{sw}}{C_{sw}} + g_{inv} \right)}{2\sqrt{B}} - \frac{g_{tri}(4C_{senb} + 4C_{wire})}{B^2} = 0$$

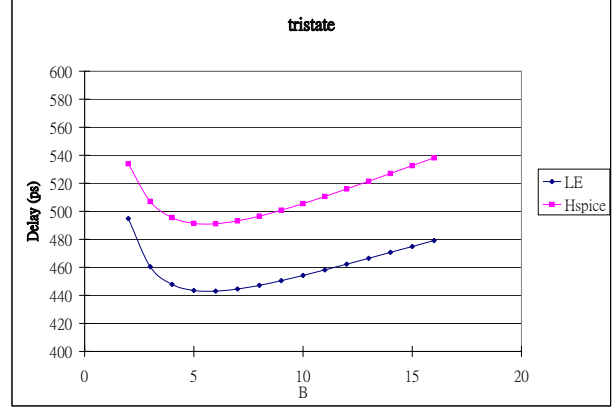
Hence

$$B = \left( \frac{2g_{tri}(4C_{senb} + 4C_{wire})}{\left( \frac{g_{sw}}{C_{sw}} + g_{inv} \right)} \right)^{\frac{2}{3}} \quad (3)$$

The result for the tristate-driver interconnect is shown in Figure 6. The general shape of both plots are similar but their absolute delay has a maximum error of 10%. Equation 3 gives an optimal value of  $B = 5.7$ , while the simulation suggests  $B \approx 5.5$ .

## 4.2. Delay Bound

Logical effort can also give a lower bound for the delay achievable. For a multistage network with stages  $i$  having



**Fig. 6.** Simulated and modeled result for tristate-driver interconnect delay

LE parameters  $g_i$  and  $p_i$ , we define [1]

$$P = \sum p_i \quad G = \prod g_i \quad B = 1 \quad H = \frac{C_{out}}{C_{in}} \quad F = GBH$$

We can then obtain a formula for the minimum delay achievable for an N stage network:

$$D = NF^{1/N} + P.$$

Although this may not be practical in a real-FPGA as area is also a consideration, knowing the lower bound gives an indication of how closely the design approaches optimality. Furthermore, for the optimal delay size, the transistors are sized so that

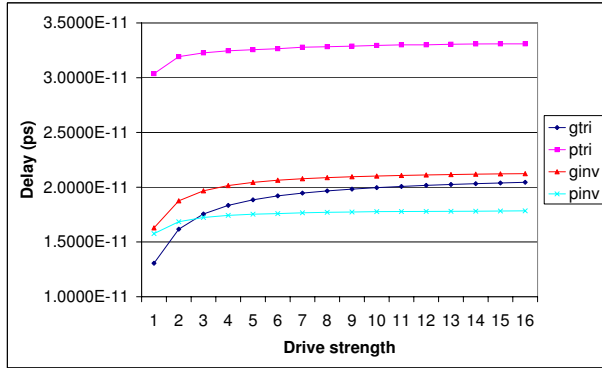
$$g_i h_i = F^{1/N}$$

in each stage.

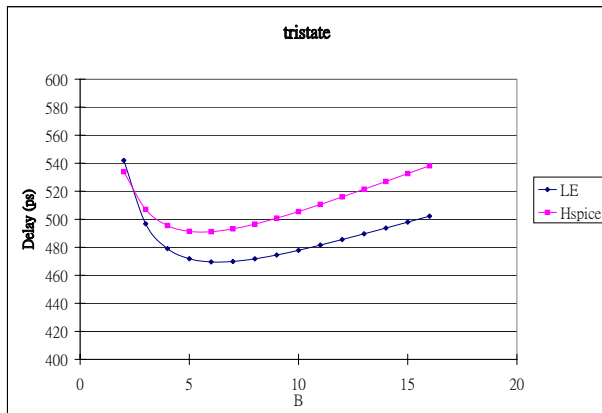
For the tristate-driver interconnect with optimal  $B$ ,  $P = p_{senb} + p_{sw} + p_{inv} + p_{tri} = 11.9$ ,  $G = g_{senb} g_{sw} g_{inv} g_{tri} = 8.58$ , and  $H = C_l / C_{senb} = 21.7$  so  $F = 186$ . Since  $N=4$ , the normalized minimum delay is  $D = 14.7 + 11.9$  which is 434 ps and achieved with a stage effort of  $F^{1/N} = 3.69$ .

## 4.3. Drive Strength Dependency

Inverters with different driving strength were simulated using HSPICE, scaling the load and inverter input capacitance by the same constant. Since  $gh$  is constant, in the LE model, no change in delay should be observed according to equation 2. As can be seen in Figure 7, this is not the case and for both the tristate buffer and inverter,  $g$  is quite a strong function of its drive strength. This is a major source of error in our LE models as the parameters are calibrated for an inverter with driving strength of 1 but used with much larger values. For the single driver interconnect, the inverter is sized according to the parameter  $B$ . If instead of measuring  $g$  for a normal inverter, we use a double size inverter, the



**Fig. 7.** LE parameters for an inverter and tristate buffer as a function of driving strength with constant  $h$ . According to the LE model, the delay should be constant.



**Fig. 8.** Tristate driver comparison using LE parameters extracted from a  $2\times$  inverter and  $2\times$  tristate buffer.

extracted  $g$  and  $p$  parameters are roughly the average value across Figure 7.

The tristate-interconnect uses sized inverters and tristate buffers. We use  $g$  and  $p$  parameters for a  $2\times$  tristate driver (0.993 and 1.96 respectively) as well as the  $2\times$  inverter. The resulting LE delay is shown in Figure 8, with a maximum error of less than 7%.

We thus observe that the LE model is a first order approximation which can be improved if necessary with little additional effort. Further refinements could be to allow  $g$  to be a function of driving strength rather than a constant.

## 5. CONCLUSION

The method of logical effort was applied to evaluate the delay of FPGA circuits and closed form expressions for optimal transistor sizing, sensitivity and optimal delay derived. The method is simple and gives a first order approximation to delay for the interconnect models studied. Although accuracy is not as high as Spice, it can be directly used for

relative circuit comparisons and its simplicity makes it useful for higher level modelling. Correction for changing LE parameters with driving strength greatly improves the accuracy of the models tested.

The LE model allows us to obtain relatively technology independent direct form equations to compare delays of different FPGA circuits and these can be used to gain intuition into the major sources of delay, optimized, and used within other CAD tools. We thus believe that the LE technique is a powerful tool for the design and optimization of FPGAs. Our future work will include developing techniques to deal with pass transistors in a simpler way, studying yield using LE, applying linear programming to optimize circuits and developing generalized simplified models for FPGA interconnect.

## Acknowledgements

The authors gratefully acknowledge support from the Research Grants Council of the Hong Kong Special Administrative Region, China (Earmarked grant CUHK413707).

## 6. REFERENCES

- [1] I. Sutherland, B. Sproull, and D. Harris, *Logical effort: designing fast CMOS circuits*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999.
- [2] H. Q. Dao and V. G. Oklobdzija, "Performance comparison of VLSI adders using logical effort," in *PATMOS '02: Proceedings of the 12th International Workshop on Integrated Circuit Design. Power and Timing Modeling, Optimization and Simulation*. London, UK: Springer-Verlag, 2002, pp. 25–34.
- [3] B. Hu, H. Jiang, Q. Liu, and M. Marek-Sadowska, "Synthesis and placement flow for gain-based programmable regular fabrics," in *ISPD '03: Proceedings of the 2003 International Symposium on Physical Design*. New York, NY, USA: ACM Press, 2003, pp. 197–203.
- [4] J. Keane, H. Eom, T.-H. Kim, S. Sapatnekar, and C. Kim, "Subthreshold logical effort: a systematic framework for optimal subthreshold device sizing," in *DAC '06: Proceedings of the 43rd Annual Conference on Design Automation*. New York, NY, USA: ACM Press, 2006, pp. 425–428.
- [5] G. Lemieux, E. Lee, M. Tom, and A. Yu, "Directional and single-driver wires in FPGA interconnect," in *IEEE International Conference on Field-Programmable Technology*, 2004, pp. 41–48.