Optimization Of Burst-Mode Circuits Using Logical Effort Theory

Charles Eric LaForest

April 2006
University of Waterloo
Independent Studies

Abstract

The logical effort and parasitic delay of static CMOS logic gates are determined through simple simulation experiments based on Logical Effort theory. This calibrated model is then used to calculate near-optimum gate sizes and path delays for an active/passive pair of Burst-Mode circuits under load. Finally, an extension of Logical Effort theory is applied to the circuits in order to determine their critical delay under an equal gate delay model and compare them to other asynchronous control circuits.

Contents

List of Figures

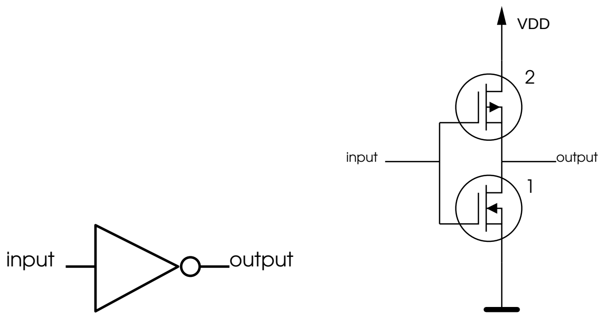
List of Tables

1 Logical Effort Theory (in a very small nutshell)

Logical Effort theory is a linear model of the delay of a gate, or of a path of gates, which unifies the electrical and logical aspects of circuit design, allowing decisions about transistor sizing and circuit topology to be made with simple manual calculations. Although the results calculated in such a manner are necessarily approximate, since a number of second-order effects are ignored, they usually fall within 10% of the achievable optima [SSH99, p. 24].

1.1 Fundamentals (Inverter)

The form of the Logical Effort expressions for static CMOS (Complementary Metal Oxide Semiconductor) gates can be derived almost by inspection. A more formal and in-depth derivation of Logical Effort can be found in [SSH99, Ch. 3], along with a dense summary in [SL01, Appendix A].



(a) Symbol (b) Schematic

Figure 1: Inverter Schematic

Figure 1b shows the schematic for an inverter, the simplest static CMOS gate. The input signal drives the gates of the pMOS (upper) and nMOS (lower) transistors, which connect to the output via their drain terminals. A pMOS transistor conducts when its gate is driven low (to ground), and vice-versa (to vdd) for the nMOS transistor. In the configuration shown, a given input signal will connect the opposite supply line to the output, thus inverting the logic value (intermediate values are disallowed). Since pMOS transistors carry current about half as well as their nMOS counterparts, they must be made approximately twice as wide in order to obtain a symmetrical output drive strength. The width of the transistors is denoted by an adjacent number[1].

1.1.1 Logical Effort

The underlying physics of a CMOS transistor result in a drive strength that is directly proportional to the capacitance of its gate, and thus to its width. Since the pMOS and nMOS transistors are sized so as to have similar drive strengths, and since an input always sees the capacitance of both transistors, a measure of the drive strength controlled by that input is the sum of the width, the total input capacitance, of all its transistors.

The logical effort of an input is defined as the ratio of the total capacitance of that input with that of an inverter *of equal drive strength*. The logical effort of an inverter is therefore 1, and the reference for all other gates.

$$g = \frac{C_{input}}{C_{inverter}} \tag{1}$$

Logical effort is a measure of how much worse than an inverter a given input is at driving the gate's output, *independent of transistor size*. A higher logical effort means a proportionately greater delay when the driven load is increased. By extension, the path logical effort through a circuit is the product of the logical effort at each stage.

$$G = \prod g_i \tag{2}$$

[1] The gate length is always kept at the minimum allowable by the design rules, barring exceptional circumstances not encountered here.

1.1.2 Electrical Effort

But size does matter. A larger gate has more drive strength than a smaller one and conversely, presents a larger load to the gate driving it. Therefore, another component of the signal delay through a gate must be its size relative to its load. The electrical effort of a gate is defined as that ratio.

$$h = \frac{C_{load}}{C_{input}} \tag{3}$$

The path electrical effort through a circuit is defined in the same manner.

$$H = \frac{C_{pathload}}{C_{pathinput}} \tag{4}$$

1.1.3 Stage Effort

The electrical and logical efforts are both proportionalities which affect each other: for a given electrical effort, a larger logical effort will result in a larger delay. The stage effort is thus the product of the two.

$$f = gh \tag{5}$$

The stage effort is a measure of delay, and so the path delay is the sum of stage efforts.

$$D_F = \sum f_i \tag{6}$$

1.1.4 Branching Effort

The effort of a path is affected by its distributaries, other paths that branch away from the main one under examination. The effect of such branches is to take away current, proportionately to their capacitance. Each such branch introduces a branching effort.

$$b = \frac{C_{onpath} - C_{offpath}}{C_{onpath}} = \frac{C_{total}}{C_{useful}} \tag{7}$$

Since this loss of current acts as an additional virtual load, it multiplies the electrical effort of a stage. The path branching effort is thus the product of the branching efforts along that path.

$$B = \prod b_i \tag{8}$$

1.1.5 Path Effort

The path effort is the product of the stage efforts along that path, or put differently, as the product of the path logical, branching, and electrical efforts.

$$F = \prod f_i \tag{9}$$

$$F = GBH \tag{10}$$

Equation 10 is used in Section 4 as part of the path optimization method.

1.1.6 Parasitic Delay

Physical gates are not ideal devices. They have some internal capacitance, and thus a delay component that is independent of the load being driven. This component, greatly simplified, is proportional to the area of the transistors connected to the output of a gate relative to an inverter *of equal drive strength*, times the actual parasitic delay of an inverter, which is provisionally defined as 1. For this, and other reasons, the parasitic delay is practically independent of the total size of the transistors in a gate.

$$p = \frac{C_{output}}{C_{inverter}} p_{inv} \tag{11}$$

The path parasitic delay is of course, the sum of all the parasitic delays along it.

$$P = \sum p_i \tag{12}$$

1.1.7 Delay Expression

Putting all the above together, the total delay of a gate and of a path can be expressed in the same manner.

$$d = f + p \tag{13}$$

$$D = D_F + P \tag{14}$$

This delay is a dimensionless value which must be somehow related to the actual initial inverter from which is is derived. An ideal inverter without parasitic delay, driving another identical inverter, would have a gate delay of 1. If this ideal inverter was a physical device, this delay would be the fundamental time unit, denoted by τ[2]. This leads to the expression of the absolute delay of a gate and a path.

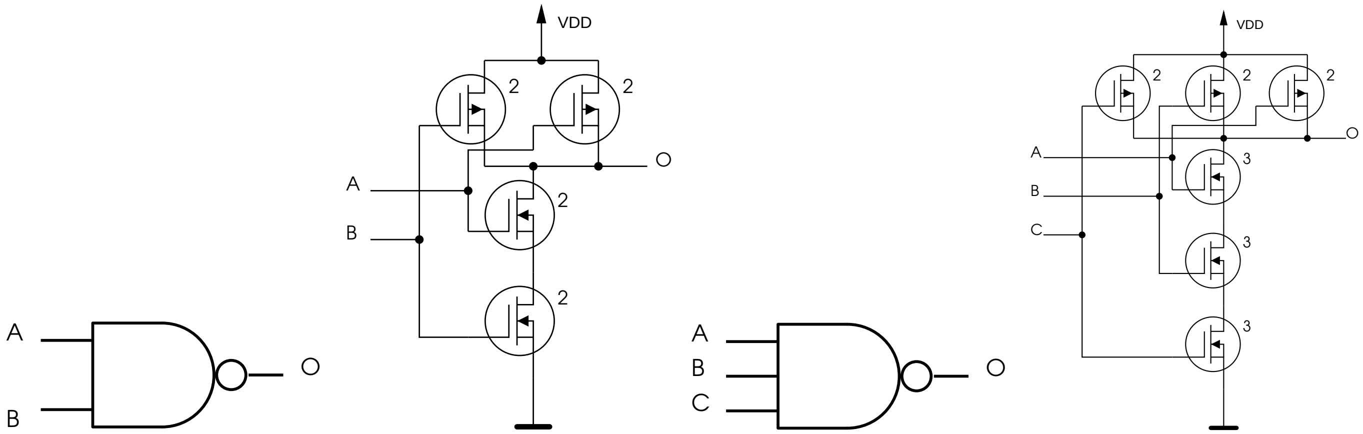
$$d_{abs} = \tau(f + p) \tag{15}$$

$$D_{abs} = \tau(D_F + P) \tag{16}$$

Section 2 will introduce a method for measuring τ and p from test circuits.

1.2 NAND Gates

From the deductions made about the inverter, it is easy to create a Logical Effort model of two and three input NAND gates. Figure 2 shows the construction of these gates. The output O goes low only when all the inputs are high. The order and the labelling of the inputs is significant, and will be discussed in Section 2.2. Note that the series nMOS transistors are scaled up to provide the same drive strength as the one in the original inverter. Table 2.2 shows the derived model parameters.



(a) NAND2 Symbol (b) NAND2 Schematic (c) NAND3 Symbol (d) NAND3 Schematic

Figure 2: NAND2 and NAND3 Schematics

Gate	NAND2	NAND3
Logical Effort	$\frac{4}{3}$	$\frac{5}{3}$
Parasitic Delay	$\frac{6}{3}$	$\frac{9}{3}$

Table 1: NAND Model (Per Input)

[2] This τ is a different quantity than the one established in [MC79].

2 Model Calibration

Logical Effort Theory is a linear model, as seen by Equation 15, and can thus be calibrated by solving a system of linear equations derived from measurements from actual circuits or simulations. In this case, the circuits are drawn up in Electric and simulated using the IRSIM switch-model simulator, which uses an RC time constant model. The capacitance and resistance of the wires are ignored for simulations of schematics.

2.1 Inverter

The inverter allows us to begin the calibration by eliminating g from consideration, since by definition it must always be 1, leaving p and τ to be determined, and h as the independent variable.

Figure 3 shows a Fanout-Of-Four (FO4) test circuit, which provides a system where $h = 4$. The gate inside the dashed box is the one under test, which drives the four identical inverters that follow. The remaining inverters form input and output buffers to provide realistic signals strengths and electrical loads, since the simulator's inputs and outputs are idealized and would affect the circuit under test.

Rising and falling transitions are sent through the gate under test and the propagation delay is measured. These are shown in Table 2 along with the delays for a similar circuit with a fanout of eight (F08). The averages are used to mitigate the inequality in delay between the rising and falling transitions, due the the pMOS and nMOS relative widths not yielding identical drive strengths. This is sufficient to optimize the average delay through a path, but not its worst or best delay [SSH99, 7.1].

2.1.1 Sources of Error

This difference between the rise and fall times is the major source of error between the calculations and the simulations. This can be seen by comparing the error on rising and falling transitions on the same signal: One is much lower than the other. This is clearly visible in Tables 8 and 10. The effect worsens in proportion to the complexity of the gate, as seen in Tables 4 and 5.

While the errors should cancel out when going through two consecutive identical gates, since CMOS gates are all inverting and the errors lie on either side of the average, if a complex gate is preceded (or followed) by a simple one, then the error from the complex gate will be much greater than the opposite one from the simpler gate, and the difference will remain.

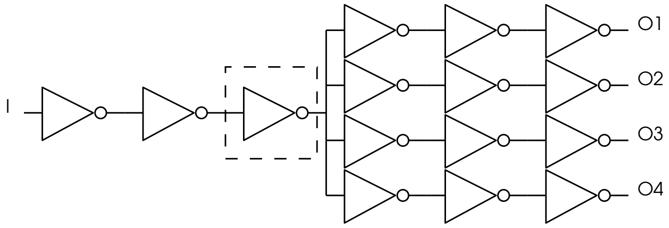


Figure 3: Inverter FO4 Test Circuit Schematic

FO4 Inverter Delay (ps)		FO8 Inverter Delay (ps)	
Transition	Delay	Transition	Delay
L to H	196.17	L to H	351.06
H to L	199.07	H to L	358.79
Average	197.62	Average	354.93

Table 2: Inverter Delays

From these results, and the defined properties of the inverter, we get a pair of equations (17 and 18) from which we can determine p and τ, shown in Table 3[3].

$$354.93ps = 8\tau + \tau p_{inv} \tag{17}$$

$$197.62ps = 4\tau + \tau p_{inv} \tag{18}$$

Inverter Parameters	
τ	39.33ps
p_{inv}	1.025

Table 3: Inverter Parameters

2.2 NAND Gates

Using the value of τ from the measurements performed on the inverter, the parameters g and p of more complex gates, implemented in the same technology or simulation, can be determined in the same manner. Figure 4 shows an FO4 test circuit for the A input of a NAND2 gate.

This test must be repeated for each input, since the parasitic delay seen by each is different. Referring back to Figure 2d, the nMOS transistor for input A only has to discharge its own parasitic capacitance, while the nMOS transistor for input C has to do so for A and B also, and thus should have a higher value of p.

Tables 4, 5, 6, and 7 show the measured delays and the derived parameters. The logical effort values effectively match those of the model in Table 1, but the parasitic delays are lower and increasing as predicted.

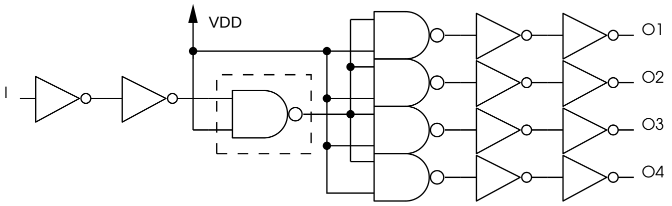


Figure 4: NAND2 FO4 Test Circuit Schematic

FO4 NAND2 Delay (ps)			FO8 NAND2 Delay (ps)		
Transition	Input A	Input B	Transition	Input A	Input B
L to H	249.28	259.49	L to H	454.90	463.66
H to L	272.55	291.01	H to L	484.42	503.36
Average	260.92	275.25	Average	469.66	483.51

Table 4: NAND2 Delays

FO4 NAND3 Delay (ps)				FO8 NAND3 Delay (ps)			
Transition	Input A	Input B	Input C	Transition	Input A	Input B	Input C
L to H	308.33	326.56	334.83	L to H	565.01	582.74	589.75
H to L	344.06	371.16	399.04	H to L	607.85	634.52	664.58
Average	326.95	348.86	366.94	Average	586.43	608.63	627.17

Table 5: NAND3 Delays

[3] An interesting and useful relationship: 1 FO4 delay $\approx 5\tau$. The FO4 delay is a commonly used reference for comparing circuit technologies.

NAND2 Parameters		
Parameter	Input A	Input B
g	1.33	1.32
p	1.30	1.73

Table 6: NAND2 Parameters

NAND3 Parameters			
Parameter	Input A	Input B	Input C
g	1.65	1.65	1.65
p	1.71	2.27	2.75

Table 7: NAND3 Parameters

3 Circuit Descriptions

The control circuits are built up from the previously described gates. The details of their use falls outside the scope of this report and are described in [LaF05].

3.1 Passive Element

Figure 5 shows the schematic for the passive element with the gates labelled. A particular input is identified by the gate label subscripted with its input label from Figure 2. For example, the input signal F enters inputs $\mathsf{a}_\mathsf{A}$and b_A.

The passive element is an implementation of the common Muller C-element circuit. The output A goes high when both inputs F and R do, and does not return low until both inputs have done so.

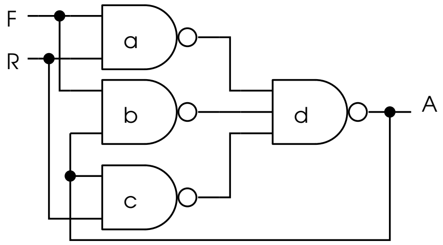


Figure 5: Passive Element Schematic

3.1.1 Delay Calculations

The calculated delay of a particular path through a circuit is the sum of the delay of each stage according to Equation 15. For example, if the F input goes high, followed by R, then the path through the circuit to the output A is $(\mathsf{a}_\mathsf{B}, \mathsf{d}_\mathsf{A})$. This set of transitions is denoted $\mathsf{F}^+\mathsf{R}^+ \rightarrow \mathsf{A}^+$.

From this information the proper values of g and p and h can be determined once the load capacitance driven by A, expressed in transistor gate widths, is known. From [LaF05] and [LaF06] it can be determined that a passive element used in an active/passive synchronizer pair driving flip-flops, sees a load of 30, plus that of the feedback going to b_B and c_A (4 each)[4], plus an extra 3 from an optimization that will be described in Section 4.3.2, for a total of 41.

Continuing with the example, Equation 19 expresses the delay[5] along path $(\mathsf{a}_\mathsf{B}, \mathsf{d}_\mathsf{A})$. Table 8 shows the results for this and other transitions.

$$\tau\left(\frac{4}{3} \times \frac{5}{4} + 1.71\right) + \tau\left(\frac{5}{3} \times \frac{41}{5} + 1.71\right) = 3.38\tau + 15.38\tau = 738ps \tag{19}$$

[4] The optimization method does not change the size of the gates at the root of a path, so this apparent combinational loop can be treated as a simple load. If the gates were in the middle of the path, iteration would be required to converge to the desired delay.

[5] Unfortunately, throughout the calculations I mistakenly used 1.71 instead of 1.73 for the parasitic delay of the A input of a NAND2 gate. Fortunately, the error itself is only slightly above 1%, and is a much smaller fraction in the final results.

Passive Element Unoptimized Delays				
		Delay (ps)		
Transitions	Path	Calculated	Measured	Error (%)
$F^+R^+ \to A^+$	(a_B, d_A)	738	762.59	+3.33
$F^-R^- \to A^-$	(c_B, d_C)	779	775.02	-0.51
$R^+F^+ \to A^+$	(a_A, d_A)	722	748.04	+3.61
$R^-F^- \to A^-$	(b_A, d_B)	744	741.99	-0.27

Table 8: Passive Element Unoptimized Delays

3.2 Active Element

The active element (Figure 6) is the counterpart to the passive one. The output R goes high when the F input does so and drops low when the A input goes high. The circuit is reset once both inputs return to low. The schematic and path labelling conventions are the same as in Section 3.1.

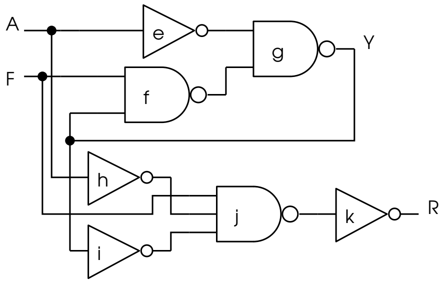


Figure 6: Active Element Schematic

3.2.1 Delay Calculations

The delay calculations are done using the same method as in Section 3.1.1, and are shown in Table 9. The load seen by R is 20. The $Y^+ \to R^-$ transition is not a causal one. It is included to observe the internal difference in delay between the (h, i_B, k) and (e, g_A) paths.

Active Element Unoptimized Delays				
		Delay (ps)		
Transitions	Path	Calculated	Measured	Error (%)
$F^+ \to R^+$	(j_A, k)	409	415.69	+1.64
$A^+ \to R^-$	(h, j_B, k)	537	590.85	+10.0
$F^+A^+ \to Y^+$	(e, g_A)	236	257.60	+9.15
$Y^+ \to R^-$	no path	n/a	332.82	n/a
$F^-A^- \to Y^-$	(e, g_A)	236	239.70	+1.57
$A^-F^- \to Y^-$	(f_A, g_B)	263	279.09	+6.12

Table 9: Active Element Unoptimized Delays

4 Optimizations

4.1 Theory

The minimum delay along a path is achieved when the path effort F from Equation 9 is distributed equally to each of N stages, denoted as $\hat{f}$. This is proven in [SSH99, Ch. 3].

$$\hat{f} = F^{\frac{1}{N}} \tag{20}$$

Placing this optimal effort into Equation 5 and expressing the electrical effort component as in Equation 3 gives an expression of the optimal relationship between the input and output capacitances of a stage, and therefore of the size of the transistors within it.

$$\frac{C_{load}}{C_{input}} = \frac{\hat{f}}{g} \tag{21}$$

$$C_{input} = \frac{gC_{load}}{\hat{f}} \tag{22}$$

Starting at the path output, one can now work backwards to calculate the optimum input capacitance for each stage. The capacitance gets distributed amongst the transistors in a gate in proportion to their relative sizes. For example, for the inverter in Figure 1b the pMOS transistor would get $\frac{2}{3}$ of the total capacitance, and the nMOS would get $\frac{1}{3}$.

4.2 Passive Element

4.2.1 Optimization

The passive element paths for F and R are similar, and so the optimization of one applies to the other. Figure 7 shows the path for F, including the final load driven by A mentioned in Section 3.1.1. The path actually branches into two non-branching, reconvergent paths of equal length: $(\mathsf{a_A}, \mathsf{d_A})$ and $(\mathsf{b_A}, \mathsf{d_B})$. The method for analyzing such paths is presented in [SSH99, 10.1].

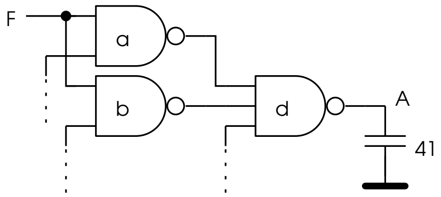


Figure 7: Passive Element Path

The effort of each non-branching path must first be determined. The input capacitance of each path includes all input loads (both $\mathsf{a_A}$ and $\mathsf{b_A}$).

$$G_{(\mathsf{a_A},\mathsf{d_A})} = G_{(\mathsf{b_A},\mathsf{d_B})} = \frac{4}{3} \times \frac{5}{3} = \frac{20}{9} = 2.22 \tag{23}$$

$$H_{(\mathsf{a_A},\mathsf{d_A})} = H_{(\mathsf{b_A},\mathsf{d_B})} = \frac{41}{8} = 5.125 \tag{24}$$

$$F_{(\mathsf{a_A},\mathsf{d_A})} = F_{(\mathsf{b_A},\mathsf{d_B})} = GH = \frac{20}{9} \times \frac{41}{8} = \frac{820}{72} = 11.39 \tag{25}$$

The branching effort of each non-branching path is calculated in a manner similar to Equation 7, by comparing the effort of the path against the sum of all path efforts. In this case the branching efforts happen to be equal, but only because gates a and b are identical.

$$B_{(\mathsf{a_A},\mathsf{d_A})} = B_{(\mathsf{b_A},\mathsf{d_B})} = \frac{F_{(\mathsf{a_A},\mathsf{d_A})} + F_{(\mathsf{b_A},\mathsf{d_B})}}{F_{(\mathsf{a_A},\mathsf{d_A}),(\mathsf{b_A},\mathsf{d_B})}} = 2 \tag{26}$$

The actual path efforts can now be computed using Equation 10. In this case both paths are identical.

$$F = \frac{20}{9} \times 2 \times \frac{41}{8} = \frac{1640}{72} = 22.78 \tag{27}$$

The optimization is now straightforward since we know the number of stages[6] and thus $N = 2$. The correctness of the results is confirmed by finding that the input capacitances remain unchanged.

$$\hat{f} = 22.78^{\frac{1}{2}} = 4.77 \tag{28}$$

[6] The number of stages could have been increased by adding buffers at the output and including them in the calculations, but the path efforts involved are low enough to not require this. See [SSH99, 1.4] for details.

$$C_{\mathsf{d}} = \frac{\frac{5}{3} \times 41}{4.77} = 14.33 \tag{29}$$

$$C_{\mathsf{a,b}} = \frac{\frac{4}{3} \times 14.33}{4.77} = 4.01 \tag{30}$$

4.2.2 Delay Calculations

The delays can now be recalculated using Equation 19 and compared to the unoptimized values from Table 8 in Section 3.1.1.

Passive Element Optimized Delays						
		Delay (ps)			Improvement (%)	
Transitions	Path	Calculated	Measured	Error (%)	Calculated	Measured
$\mathsf{F}^+\mathsf{R}^+ \rightarrow \mathsf{A}^+$	$(\mathsf{a_B}, \mathsf{d_A})$	510	581.18	+14.0	44.7	31.2
$\mathsf{F}^-\mathsf{R}^- \rightarrow \mathsf{A}^-$	$(\mathsf{c_B}, \mathsf{d_C})$	551	561.67	+1.94	41.4	40.0
$\mathsf{R}^+\mathsf{F}^+ \rightarrow \mathsf{A}^+$	$(\mathsf{a_A}, \mathsf{d_A})$	494	566.20	+14.6	46.2	32.1
$\mathsf{R}^-\mathsf{F}^- \rightarrow \mathsf{A}^-$	$(\mathsf{b_A}, \mathsf{d_B})$	516	531.58	+3.02	44.2	39.6

Table 10: Passive Element Optimized Delays

4.3 Active Element

The active element is more complex than the passive one, but breaks down into a number of simpler paths based on the signal transitions from Table 9. The path effort is calculated using Equation 10 and optimized as done in Section 4.1.

4.3.1 $(\mathsf{j_A}, \mathsf{k})$ Path

The $(\mathsf{j_A}, \mathsf{k})$ path (Figure 8) is the shortest path from an input to an output, so it will be optimized first. Starting with a longer path would alter the size of the j gate and thus alter the input capacitance of F, which would affect the load, and thus the speed, of the earlier circuitry driving it.

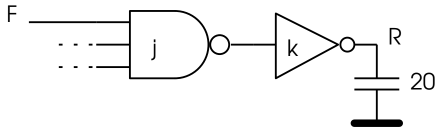


Figure 8: Active Element $(\mathsf{j_A}, \mathsf{k})$ Path

$$C_k = \frac{1 \times 20}{2.58} = 7.75 \tag{31}$$

$$C_j = \frac{\frac{5}{3} \times 7.75}{2.58} = 5.01 \tag{32}$$

Active Element Optimized $(\mathsf{j_A}, \mathsf{k})$ Path Delays						
		Delay (ps)			Improvement (%)	
Transitions	Path	Calculated	Measured	Error (%)	Calculated	Measured
$\mathsf{F}^+ \rightarrow \mathsf{R}^+$	$(\mathsf{j_A}, \mathsf{k})$	311	337.31	+8.46	31.5	23.2
$\mathsf{A}^+ \rightarrow \mathsf{R}^-$	$(\mathsf{h}, \mathsf{j_B}, \mathsf{k})$	439	515.08	+17.3	22.3	14.7
$\mathsf{F}^+\mathsf{A}^+ \rightarrow \mathsf{Y}^+$	$(\mathsf{e}, \mathsf{g_A})$	236	257.26	+9.00	0	0.1
$\mathsf{Y}^+ \rightarrow \mathsf{R}^-$	no path	n/a	258.37	n/a	n/a	28.8
$\mathsf{F}^-\mathsf{A}^- \rightarrow \mathsf{Y}^-$	$(\mathsf{e}, \mathsf{g_A})$	236	240.35	+1.84	0	-0.3
$\mathsf{A}^-\mathsf{F}^- \rightarrow \mathsf{Y}^-$	$(\mathsf{f_A}, \mathsf{g_B})$	263	276.34	+5.83	0	0.3

Table 11: Active Element Optimized $(\mathsf{j_A}, \mathsf{k})$ Path Delays

4.3.2 $(\mathsf{h}, \mathsf{j_B}, \mathsf{k})$ Path

The $(\mathsf{h}, \mathsf{j_B}, \mathsf{k})$ path (Figure 9) includes the $(\mathsf{j_A}, \mathsf{k})$ path and so already benefits from its optimization. However, the h inverter still has an electrical effort of $\frac{5}{3}$. If it is scaled up to 6, its electrical effort will be reduced to $\frac{5}{6}$, which is less than p_{inv}, which will then be the quasi-totality of the delay through h. The trade-off is that the input capacitance of A is increased to 9 and the delay is simply moved to the previous stage, which in this case is the output of the passive element. This small extra load was already taken into account in Section 3.1.1. This change does not alter the size of gate j, and so does not affect the optimization of the $(\mathsf{j_A}, \mathsf{k})$ path.

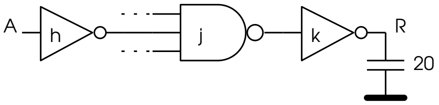


Figure 9: Active Element $(\mathsf{h}, \mathsf{j_B}, \mathsf{k})$ Path

Active Element Optimized $(\mathsf{h}, \mathsf{j_B}, \mathsf{k})$ and $(\mathsf{j_A}, \mathsf{k})$ Path Delays						
		Delay (ps)			Improvement (%)	
Transitions	Path	Calculated	Measured	Error (%)	Calculated	Measured
$\mathsf{F}^+ \rightarrow \mathsf{R}^+$	$(\mathsf{j_A}, \mathsf{k})$	311	336.54	+8.21	31.5	23.5
$\mathsf{A}^+ \rightarrow \mathsf{R}^-$	$(\mathsf{h}, \mathsf{j_B}, \mathsf{k})$	406	473.53	+16.6	32.3	24.8
$\mathsf{F}^+\mathsf{A}^+ \rightarrow \mathsf{Y}^+$	$(\mathsf{e}, \mathsf{g_A})$	236	270.88	+14.8	0	-4.90
$\mathsf{Y}^+ \rightarrow \mathsf{R}^-$	no path	n/a	202.09	n/a	n/a	64.7
$\mathsf{F}^-\mathsf{A}^- \rightarrow \mathsf{Y}^-$	$(\mathsf{e}, \mathsf{g_A})$	236	252.88	+7.15	0	-5.21
$\mathsf{A}^-\mathsf{F}^- \rightarrow \mathsf{Y}^-$	$(\mathsf{f_A}, \mathsf{g_B})$	263	278.88	+6.04	0	0.8

Table 12: Active Element Optimized $(\mathsf{h}, \mathsf{j_B}, \mathsf{k})$ and $(\mathsf{j_A}, \mathsf{k})$ Path Delays

4.3.3 $(\mathsf{e}, \mathsf{g_A})$ and $(\mathsf{f_A}, \mathsf{g_B})$ Paths

The Y signal is an internal signal used to avoid logic hazards on gate j. The active element cannot reliably operate faster than it, thus there would be benefit in optimization. However, the path efforts are very low (and equal by chance), so the improvement is slight. The branching at A and F is not considered in the path effort calculations since the input capacitances are all fixed by previous optimizations, and the transitions on Y are not causal to transitions on R, so the method used in Section 4.2.1 is not applicable.

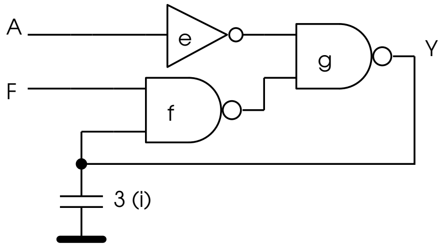


Figure 10: Active Element $(\mathsf{e}, \mathsf{g_A})$ and $(\mathsf{f_A}, \mathsf{g_B})$ Paths

$$C_g = \frac{\frac{4}{3} \times 7}{1.76} = 5.30 \tag{33}$$

$$C_e = \frac{1 \times 5.30}{1.76} = 3.01 \tag{34}$$

$$C_f = \frac{\frac{4}{3} \times 5.30}{1.76} = 4.02 \tag{35}$$

Active Element Optimized $(\mathsf{h}, \mathsf{j_B}, \mathsf{k})$, $(\mathsf{j_A}, \mathsf{k})$, $(\mathsf{e}, \mathsf{g_A})$ and $(\mathsf{f_A}, \mathsf{g_B})$ Path Delays

		Delay (ps)			Improvement (%)	
Transitions	Path	Calculated	Measured	Error (%)	Calculated	Measured
$\mathsf{F}^+ \rightarrow \mathsf{R}^+$	$(\mathsf{j_A}, \mathsf{k})$	311	336.86	+8.32	31.5	23.4
$\mathsf{A}^+ \rightarrow \mathsf{R}^-$	$(\mathsf{h}, \mathsf{j_B}, \mathsf{k})$	406	472.08	+16.3	32.3	25.2
$\mathsf{F}^+\mathsf{A}^+ \rightarrow \mathsf{Y}^+$	$(\mathsf{e}, \mathsf{g_A})$	230	273.98	+19.1	2.61	-6.00
$\mathsf{Y}^+ \rightarrow \mathsf{R}^-$	no path	n/a	198.9	n/a	n/a	67.3
$\mathsf{F}^-\mathsf{A}^- \rightarrow \mathsf{Y}^-$	$(\mathsf{e}, \mathsf{g_A})$	230	253.61	+10.3	2.61	-5.48
$\mathsf{A}^-\mathsf{F}^- \rightarrow \mathsf{Y}^-$	$(\mathsf{f_A}, \mathsf{g_B})$	258	280.09	+8.56	1.94	-0.36

Table 13: Active Element Optimized $(\mathsf{h}, \mathsf{j_B}, \mathsf{k})$, $(\mathsf{j_A}, \mathsf{k})$, $(\mathsf{e}, \mathsf{g_A})$ and $(\mathsf{f_A}, \mathsf{g_B})$ Path Delays

For comparison with Table 9, Table 14 shows the delays from the original active element with only the $(\mathsf{e}, \mathsf{g_A})$ and $(\mathsf{f_A}, \mathsf{g_B})$ path optimizations.

Active Element Optimized $(\mathsf{e}, \mathsf{g_A})$ and $(\mathsf{f_A}, \mathsf{g_B})$ Path Delays

		Delay (ps)			Improvement (%)	
Transitions	Path	Calculated	Measured	Error (%)	Calculated	Measured
$\mathsf{F}^+ \rightarrow \mathsf{R}^+$	$(\mathsf{j_A}, \mathsf{k})$	409	416.27	+1.78	0	-0.14
$\mathsf{A}^+ \rightarrow \mathsf{R}^-$	$(\mathsf{h}, \mathsf{j_B}, \mathsf{k})$	537	590.89	+10.0	0	0
$\mathsf{F}^+\mathsf{A}^+ \rightarrow \mathsf{Y}^+$	$(\mathsf{e}, \mathsf{g_A})$	230	258.90	+12.6	2.61	-0.50
$\mathsf{Y}^+ \rightarrow \mathsf{R}^-$	no path	n/a	332.68	n/a	n/a	-0.04
$\mathsf{F}^-\mathsf{A}^- \rightarrow \mathsf{Y}^-$	$(\mathsf{e}, \mathsf{g_A})$	230	238.82	+3.83	2.61	-0.37
$\mathsf{A}^-\mathsf{F}^- \rightarrow \mathsf{Y}^-$	$(\mathsf{f_A}, \mathsf{g_B})$	258	280.05	+8.55	1.94	-0.34

Table 14: Active Element Optimized $(\mathsf{e}, \mathsf{g_A})$ and $(\mathsf{f_A}, \mathsf{g_B})$ Path Delays

5 Critical Delay

5.1 Theoretical Extensions

A recent extension to Logical Effort Theory [EGC04] provides a more formal way of expressing the paths in a circuit and allow for combinational loops. Equation 36 presents a template of the model for a hypothetical gate i in a circuit.

The right-hand side described the total capacitance driven by the gate, where $g_a x_a$ is the product of the logical effort g and the drive strength x of gate a. Recall from Section 1.1.1 that the logical effort of a gate input, relative to an inverter, is the amount of input capacitance per unit of drive strength. Therefore the product of the logical effort and of the drive strength is the capacitance of that input. Similarly, the amount of output capacitance per unit of drive strength is the logical effort of the gate output, which is identical to the parasitic delay denoted by p_i (see Section 1.1.6). Finally, simple capacitive loads such as wires are denoted by L, also expressed units of transistor width.

The left-hand side described the total amount of capacitance that gate i can drive in time period s_i (expressed in units of τ). Drive strength is expressed in units of capacitance per unit time. This makes intuitive sense: for a given load capacitance (right-hand side), a gate must either take longer to drive it, or have its drive strength scaled up to meet the time constraint.

$$s_i * x_i = p_i x_i + g_a x_a + g_b x_b + ... + L \tag{36}$$

Equation 37 shows the general form of this model as a matrix equation, where S is the matrix of delays, x is the vector of all gate drive strengths, T is the matrix of logical efforts, and b is the vector of fixed capacitances. The matrix T effectively describes the connections between the gates.

$$S * x = T * x + b \tag{37}$$

A special case occurs when the delay for all gates is defined as being equal, and thus a scalar, as shown by Equation 38. Given a delay, the vector x will contain the drive strengths (and thus the sizes) of each gate such that they will all have the exact same propagation delay s. This provides more accurate results according to the authors of [EGC04], and simplifies the comparison of path delays to simply counting the gates along them.

$$s * x = T * x + b \tag{38}$$

A property of Equation 38 derived from the theory of non-negative matrices[7] is that the solution x is non-negative if and only if the value of s is greater than the maximum absolute value of all eigenvalues of T. This maximum absolute value is defined as the critical delay of the circuit. It is the lowest possible bound on the delay of the gates, assuming an equal-gate-delay model, regardless of gate size. The critical delay provides a technology-independent way of comparing the performance.

[7] The logical effort and capacitance values provided in T and b are necessarily non-negative.

5.2 Passive Element

Equations 39, 40, 41, and 42 describe the passive element from Figure 5. The theoretical values of g and p determined in Table 1 are used, instead of the measured ones from Section 2.2, so as to match the calculations in [EGC04].

From these we can extract the matrix T, shown in (43)[8], and compute its eigenvalues, shown in (44). The largest absolute eigenvector value, and thus the critical delay, is 4.667. This value is consistent with the fact that the passive element is an implementation of a Muller C-element, and a chain of C-elements was shown to have a critical delay of 4.5616.

$$s_{\mathsf{a}} x_{\mathsf{a}} = 2x_{\mathsf{a}} + \frac{5}{3}x_{\mathsf{d}} \tag{39}$$

$$s_{\mathsf{b}} x_{\mathsf{b}} = 2x_{\mathsf{b}} + \frac{5}{3}x_{\mathsf{b}} \tag{40}$$

$$s_{\mathsf{c}} x_{\mathsf{c}} = 2x_{\mathsf{c}} + \frac{5}{3}x_{\mathsf{d}} \tag{41}$$

$$s_{\mathsf{d}} x_{\mathsf{d}} = \frac{4}{3}x_{\mathsf{b}} + \frac{4}{3}x_{\mathsf{c}} + 3x_{\mathsf{d}} \tag{42}$$

$$T_{passive} = \begin{bmatrix} 2 & 0 & 0 & \frac{5}{3} \\ 0 & 2 & 0 & \frac{5}{3} \\ 0 & 0 & 2 & \frac{5}{3} \\ 0 & \frac{4}{3} & \frac{4}{3} & 3 \end{bmatrix} \tag{43}$$

$$\begin{aligned} \lambda_1 &= 2 \\ \lambda_2 &= 4.667 \\ \lambda_3 &= 2 \\ \lambda_4 &= 0.3333 \end{aligned} \tag{44}$$

[8] Note that the diagonal of T is filled with the parasitic delays of the gates. The other values in each row then describe to which other gates they are connected.

5.3 Active Element

The critical delay of the active element is 3.333, which places it above the 2-4 GasP control (2.4179) and close to the asP*[9] circuit (3.9509) given in [EGC04]. For comparison the absolute lowest critical delay possible, provided by a 'magic control', is 2.

$$s_{\mathsf{e}} x_{\mathsf{e}} = x_{\mathsf{e}} + \frac{4}{3} x_{\mathsf{g}} \tag{45}$$

$$s_{\mathsf{f}} x_{\mathsf{f}} = 2x_{\mathsf{f}} + \frac{4}{3} x_{\mathsf{g}} \tag{46}$$

$$s_{\mathsf{g}} x_{\mathsf{g}} = \frac{4}{3} x_{\mathsf{f}} + 2x_{\mathsf{g}} + x_{\mathsf{i}} \tag{47}$$

$$s_{\mathsf{h}} x_{\mathsf{h}} = x_{\mathsf{h}} + \frac{5}{3} x_{\mathsf{j}} \tag{48}$$

$$s_{\mathsf{i}} x_{\mathsf{i}} = x_{\mathsf{i}} + \frac{5}{3} x_{\mathsf{j}} \tag{49}$$

$$s_{\mathsf{j}} x_{\mathsf{j}} = 3x_{\mathsf{j}} + x_{\mathsf{k}} \tag{50}$$

$$s_{\mathsf{k}} x_{\mathsf{k}} = x_{\mathsf{k}} \tag{51}$$

$$T_{active} = \begin{bmatrix} 1 & 0 & \frac{4}{3} & 0 & 0 & 0 & 0 \\ 0 & 2 & \frac{4}{3} & 0 & 0 & 0 & 0 \\ 0 & \frac{4}{3} & 2 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & \frac{5}{3} & 0 \\ 0 & 0 & 0 & 0 & 1 & \frac{5}{3} & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{52}$$

$$\begin{aligned} \lambda_1 &= 1 \\ \lambda_2 &= 3.3333 \\ \lambda_3 &= 0.6667 \\ \lambda_4 &= 1 \\ \lambda_5 &= 1 \\ \lambda_6 &= 3 \\ \lambda_7 &= 1 \end{aligned} \tag{53}$$

[9] asP* stands for "asynchronous symmetric pulse protocol", a family of circuits developed by the late Charles E. Molnar which are the predecessors of GasP circuits.

5.4 Active/Passive Pair

The active and passive elements are meant to work as a pair, each controlling one latch or register and synchronizing a data transfer between them. The pair circuit can be expressed by connecting the schematics from Figures 5 and 6. The A lines are connected together, as are the R lines. The F lines are ignored since they are solely driven by external controlling circuits.

Equation 54 shows the combined T matrix of the circuit pair. Note that the top-leftmost 4x4 section is $T_{passive}$ (Equation 43) while the bottom-rightmost 7x7 section is T_{active} (Equation 52). The nonzero entries in the remainder of T_{pair} show the interconnections between the two elements.

The resulting critical delay is 4.7467, which is somewhat larger than that of a C-element or of the passive element. However, a pair controls two latches or registers instead of one. How this affects the meaning of the critical delay has not been explored yet.

$$T_{pair} = \begin{bmatrix} 2 & 0 & 0 & \frac{5}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & \frac{5}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & \frac{5}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{4}{3} & \frac{4}{3} & 3 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & \frac{4}{3} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & \frac{4}{3} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{4}{3} & 2 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \frac{5}{3} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \frac{5}{3} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 1 \\ \frac{4}{3} & 0 & \frac{4}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (54)$$

$$\begin{aligned} \lambda_1 &= 4.7467 \\ \lambda_2 &= 0.0575 + 0.5932i \\ \lambda_3 &= 0.0575 - 0.5932i \\ \lambda_4 &= 0.9770 + 0.6532i \\ \lambda_5 &= 09770 - 0.6532i \\ \lambda_6 &= 3.1101 \\ \lambda_7 &= 2.6014 \\ \lambda_8 &= 2.4727 \\ \lambda_9 &= 2 \\ \lambda_{10} &= 2 \\ \lambda_{11} &= 1 \end{aligned} \quad (55)$$

References

[EGC04] J. Ebergen, J. Gainsley, and P. Cunningham, *Transistor sizing: How to control the speed and energy consumption of a circuit*, Proc. International Symposium on Advanced Research in Asynchronous Circuits and Systems, IEEE Computer Society Press, April 2004, pp. 51–61.

[LaF05] Eric LaForest, *Unit 4-5 (IS 302B) Burst-Mode locally-clocked asynchronous implementation of the gullwing stack computer architecture*, IS Unit report, University of Waterloo, Independent Studies Program, December 2005, Otherwise unpublished.

[LaF06] ______, *Unit 5 (IS 307) Optimization of Burst-Mode circuits using Logical Effort Theory*, IS Unit report, University of Waterloo, Independent Studies Program, April 2006, Otherwise unpublished.

[MC79] Carver Mead and Lynn Conway, *Introduction to VLSI systems*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1979, The original and authoritative book on VLSI circuitry.

[SL01] Ivan E. Sutherland and Jon K. Lexau, *Designing fast asynchronous circuits*, ASYNC '01: Proceedings of the 7th International Symposium on Asynchronous Circuits and Systems (Washington, DC, USA), IEEE Computer Society, 2001, p. 184.

[SSH99] Ivan Sutherland, Bob Sproull, and David Harris, *Logical effort: designing fast CMOS circuits*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999.