

New fast fixed-delay sizing algorithm for high-performance CMOS combinational logic circuits and its applications

Jen-Sheng Hwang
Chung-Yu Wu

Indexing terms: Algorithms, CMOS, Combinational logic

Abstract: A sizing methodology called the near-characteristic waveform-synthesising method (NCWSM) is proposed to determine the device sizes of CMOS combinational logic circuits under a fixed delay specification. By using accurate physical timing models and the NCWSM, a fixed-delay sizing algorithm is developed and implemented, which sizes circuits quickly and globally. It can handle CMOS inverters, multi-input NAND/NOR gates, and AOI/OAI gates, all with device channel lengths down to 1.5 μm . It is shown through experimental verifications that the proposed algorithm can size a circuit with much smaller CPU time than that for the heuristic approach, and the resultant circuit power dissipations are nearly the same. As the circuit complexity increases, the above advantageous feature becomes more significant and the minimum realisable delay is even smaller than that of the heuristic approach. With high efficiency and delay accuracy, the proposed sizing algorithm and methodology can handle large-scale circuits with less design time. It can also serve to provide a good initial guess for more advanced sizing operations.

1 Introduction

Realising various circuit parameters to meet a delay time requirement is one of the most important issues in computer-aided integrated circuit (IC) design. Recently, there have been several publications [1-3] that have discussed such a fixed-delay sizing problem for MOS logic circuits. These approaches might provide an optimal sizing through an iterative process, but the required CPU time is increased considerably to an intolerable level. Sometimes, the sizing approaches cannot even get convergent results.

In certain IC design cases where the design turn-around time is critical or the circuit scale is quite large, a fast sizing operation is valuable because of the restricted design time and the large manageable circuit size to be dealt with within a fixed amount of CPU time. Although the optimal realisation for the given specifications may

not be exactly achieved, the fast sizing operation can still realise a circuit with a better performance than that achieved in manual design. Moreover, the results of a fast sizing operation can serve as a good initial guess for other more advanced sizing operations.

It is the aim of this paper to develop an algorithm for fast transistor sizing of CMOS combinational logic circuits subjected to fixed-delay specifications. A sizing methodology called the near-characteristic waveform-synthesising method (NCWSM) is proposed. Using the NCWSM, the output rise and fall times in each gate except the first gate, whose size is fixed, can be obtained from the given rise (or fall) time, the chosen rise-time/fall-time ratio, and the fan-in number of the gate [4]. The device sizes of the gate can then be calculated through suitable timing equations which relate device sizes to delay, rise, or fall times. After the device sizes are obtained, the maximum circuit delay is calculated by using the timing equations and by comparison with the specified value. From the comparison results, the previous rise (and fall) time is adjusted and the sizing operation and the delay calculation are performed again. This process is done subsequently in every signal path simultaneously until the specified delay time is achieved. In comparison with the conventional fixed-delay sizing algorithm [1, 3], the new sizing algorithm consumes much smaller CPU time, and the sized circuits have nearly the same power dissipation. As the circuit becomes complex, the proposed fixed-delay algorithm can size the circuit with a smaller fixed-delay specification.

In the NCWSM, accurate, efficient, and versatile models for the delay times of logic gates are required. The timing models developed by the present authors [5-8] have a reasonable accuracy and a wide applicability range for different CMOS logic gates with short device channel lengths down to 1.5 μm , different device parameters, capacitive loads, fan-out numbers, and input excitation waveforms. Hence these models are adopted in the sizing operations using the NCWSM.

2 Physical timing models of CMOS combinational gates

Generally, the rise and fall and delay times of an MOS logic gate are determined by (i) the device currents; (ii) the internal device capacitances; (iii) the load capacitance or resistance contributed by the loading gates; (iv) the load capacitance or resistance contributed by the interconnection line or off-chip fixed capacitive loads; (v) the rise and fall times of the input waveforms, and (vi) the excited input nodes or the input excitation patterns. Of these, the

Paper 8978E (C3, E10), received 10th June 1991

The authors are with the Department of Electronics Engineering and Institute of Electronics, National Chiao Tung University, 75 Po-Ai Street, Hsin-Chu, Taiwan 30039, Republic of China

first two factors are related to the device sizes and the last two factors are associated with the input excitations.

Ideally, an accurate, efficient, and versatile timing model formulates the rise and fall and delay times of a logic gate in terms of the above six factors. The formulas have an analytical form and a good accuracy under variations of device and circuit parameters. Such a model, then, can be applied to the sizing process and timing verification with good accuracy and efficiency.

In the timing models [5-8] to be adopted in the sizing algorithm, the general modelling approach is to derive, section-by-section, the analytical formulas of the rise of fall times T_R and T_F from the linearised large-signal equivalent circuit of a CMOS logic gate under the characteristic-waveform [5-10] considerations. Then the rise and fall propagation delay times T_{PLH} and T_{PHL} are semi-empirically expressed in terms of the calculated rise and fall times as

$$T_{PLH} = a_{rr} T_R + a_{rf} T_F + \frac{\ln 2.0}{\ln 9.0} T_R - \frac{\ln 2.0}{\ln 9.0} T_F \quad (1)$$

and

$$T_{PHL} = a_{fr} T_R + a_{ff} T_F + \frac{\ln 2.0}{\ln 9.0} T_F - \frac{\ln 2.0}{\ln 9.0} T_R \quad (2)$$

where a_{rr} , a_{rf} , a_{fr} , and a_{ff} are empirical constants for the initial delay times. On applying this modelling technique, timing models have been derived for CMOS inverters [5, 7], multi-input NAND and NOR gates [5, 7], static flip-flops [6], and AOI/OAI gates [8] with an MOS device channel length (mask) down to 1.5 μm . It is found that eqns. 1 and 2 are universal for all types of combinational logic gate each of which has its own values of a_{rr} , a_{rf} , a_{fr} , and a_{ff} , of course. It is also believed that universal initial-delay formulas can only be obtained by incorporating the calculated T_R and T_F into them.

The accuracy of the timing models for CMOS inverters has been widely verified through extensive comparisons between model calculation and SPICE simulation. Some of the comparisons are shown in Fig. 1 for 2 μm CMOS inverters with $C_L = 0$ pF and under exponential input excitations with time constants from 0.2-

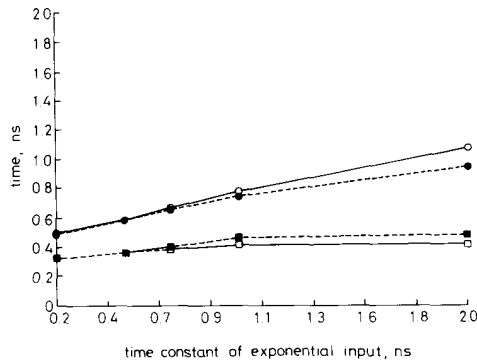


Fig. 1 Comparisons between model calculation and SPICE simulation for 2.0 μm CMOS inverters

$W_p = 2.0 \mu\text{m}$; $W_n = 2.0 \mu\text{m}$; $C_L = 0$ pF
Exponential input excitations with time constants from 0.2-2.0 ns
—○— rise delay (SPICE)
—●— rise delay (theory)
—□— fall delay (SPICE)
—■— fall delay (theory)

2.0 ns. It is found through accuracy verification that the maximum error is under 15% for inverters with different input waveforms, device parameters, and circuit parameters [5-8].

The developed timing models can also characterise the timing of multi-input logic gates excited at any input node. As an illustrative example, a 3-input CMOS NAND gate shown in Fig. 2 is considered. The timing is

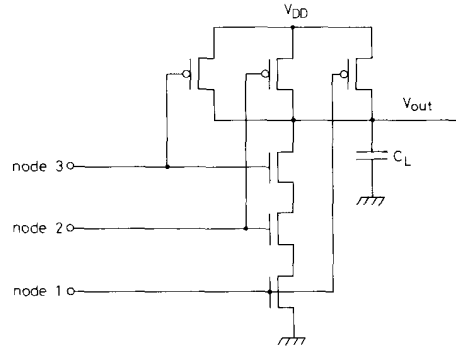


Fig. 2 CMOS 3-input NAND gate

of the worst-case type if only the input node 1 is excited while other input nodes are kept at V_{DD} . But if the node 2 or the node 3 is excited while other nodes are at V_{DD} , the timing is of the non-worst-case type. According to our observations, the longest rise delay of a 3-input CMOS NAND gate with one fan-out gate is about 58% longer than the corresponding shortest delay, thus it cannot be overlooked. The timing models [5-8] developed by the present authors considered all the triggering cases. Some of the comparisons between SPICE simulation and model calculation are listed in Table 1 for 2 μm 3-input

Table 1: Comparison of model calculation and SPICE simulation for 2.0 μm 3-input CMOS NAND gates with $W_p = 2.0 \mu\text{m}$, $W_n = 2.0 \mu\text{m}$, and $C_L = 0$ pF for different input triggering cases and characteristic-waveform consideration

Triggered node	Signal timing	SPICE	Theory	Error %
		ns	ns	%
3*	Rise time	1.407	1.469	4.4
	Fall time	1.553	1.454	-6.4
	Rise delay	0.812	0.736	-9.3
	Fall delay	0.800	0.745	-6.8
2	Rise time	1.792	1.939	8.2
	Fall time	1.593	1.647	3.4
	Rise delay	1.099	1.020	-7.2
	Fall delay	1.019	0.979	-3.9
1†	Rise time	2.182	2.456	12.6
	Fall time	1.568	1.728	10.2
	Rise delay	1.288	1.315	2.1
	Fall delay	1.170	1.015	-13.2

* Node 3 is the input node nearest to the output node
† Node 1 is the input node furthest from the output node

CMOS NAND gates in different triggering cases and under characteristic-waveform considerations. It is found that the maximum error is 15% for 3-input CMOS NAND gates excited at different input nodes and with different device and circuit parameters.

In summary, it has been shown that the maximum error of the developed timing models is under 15% for

the CMOS combinational logic gates with a wide range of device sizes, capacitive loads, device parameter variations, input excitation nodes, and input voltage waveforms not deviating much from characteristic waveforms [7]. Fine tuning for the gates with commonly used dimensions can further reduce the maximum error to less than 10%.

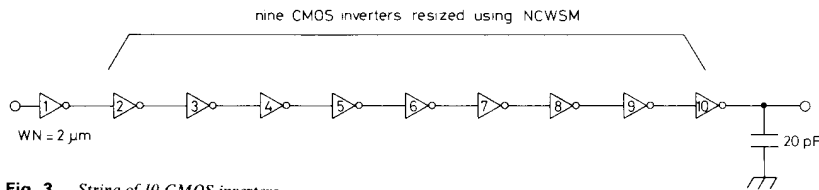


Fig. 3 String of 10 CMOS inverters

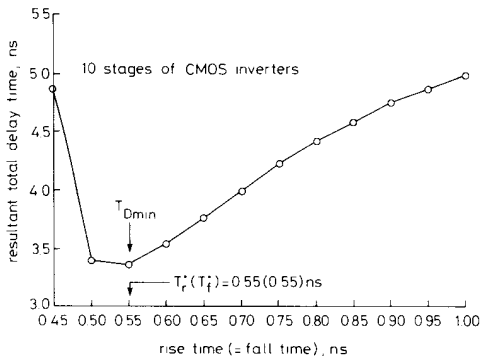


Fig. 4 Resultant total delay time under different rise (or fall) times

3 Sizing methodology

In sizing an MOS digital IC, the logic structure, the input waveforms to the circuit, the output off-chip loading, and the technology and device parameters are known. In combinational logic circuits, the output loading of the last stage in each signal path is the output off-chip loading and the loading of the internal device capacitances of the last stage, which depend on the device sizes. If the input excitation patterns and the input and output rise and fall times of the last stage are given, its device sizes are the only unknown factors in the rise-time and fall-time equations [5–8] of the timing models. Then the device sizes of the last stage can be calculated from those timing equations. Having obtained the device sizes of the last stage, the output loading of the stage preceding the last stage can be determined. Thus its device sizes can also be calculated similarly by solving its timing equations. This implies that the sizing operation of combinational logic circuits can be done sequentially from the last stages of all signal paths towards the first stages of the paths. In this way, the sizing is done globally.

In the circuit to be sized by using the fixed-delay sizing, the size of the first gate in each signal path should be fixed and the sizes of the other gates have to be found to reduce the circuit maximum delay below the specified value. In this Section, the general sizing methodology is described first, and then it is applied to the fixed-delay sizing. The associated algorithm will be presented in the following Section.

As discussed previously in this Section, the input excitation patterns and the input and output rise and fall times of a gate must be given, and the sizing can then be performed. To determine the input and output rise and

fall times, a commonly used CMOS design strategy is adopted. In this strategy, the rise-time/fall-time ratio in a gate is fixed. In the general sizing methodology, such a ratio can be arbitrarily chosen by the designer. The ratio can be unity for symmetrical transition. Having determined the ratio, the rise (fall) time can be calculated from the given fall (rise) time.

In the design of taper buffer [9, 10], we found that, as the minimum-delay design is achieved, the internal voltage waveform at each output node becomes the same, being the characteristic waveform [5–10]. For a combinational logic circuit with different kinds of gate, it is found from extensive SPICE simulations on some special testing circuits that the minimum delay can be reached if the output rise or fall time of a gate is nearly proportional to its fan-in number. The resultant waveforms are still closed to characteristic waveforms. Such behaviours have also been explored [4]. Based on the above considerations, a sizing methodology called the near-characteristic waveform-synthesising method (NCWSM) is proposed to determine the rise and fall times of each gate. In the NCWSM, a suitable time is chosen, and the output rise and fall times of each logic gate except the first gate in each signal path can be calculated from the chosen rise-time/fall-time ratio and the fan-in number N , i.e. the maximum number N of the series NMOS or PMOS in the gate.

Having determined the input and output rise and fall times, the input excitation pattern to each gate except the first gate has to be determined. In the NCWSM, only the input excitation pattern which leads to the worst-case timing of a logic gate is considered to simplify the computation complexity and to save computer time. Thus the input excitation node of a logic gate is the node farthest away from the output node. This leads to a safe design, in that the actual chip delay is always equal to or smaller than that designed.

Using the determined rise and fall times and input excitation patterns, the next step in the NCWSM is to calculate the device sizes. The device size of each gate except the first gate can be calculated from the suitable rise-time/fall-time equations, and its total delay time can be calculated from eqns. 1 and 2. Here, a commonly used CMOS design strategy is also applied. This strategy is that all MOSFETs in series are designed with equal channel width, and so are all MOSFETs in parallel. Sometimes, the synthesised device sizes are smaller than the user-defined minimum allowable channel width or larger than the user-defined maximum allowable channel width. In the first abnormal case, the device sizes are set to the minimum allowable values in the NCWSM. In the second abnormal case, the device sizes are set to the maximum allowable values. These two reset considerations have also been adopted in the sizing methodology.

4 Fixed-delay sizing algorithm

Before the fixed-delay sizing algorithm is developed, typical characteristics of the total delay time as a function

of rise (fall) time $T_R(T_F)$ in a string of 10 CMOS inverters are investigated. The inverters string is shown in Fig. 3 where the size of the first inverter is fixed. The channel width of the n -channel MOSFET in the first inverter is $2.0 \mu\text{m}$, whereas the channel width of the p -channel MOSFET is determined from the chosen unity rise-time/fall-time ratio. The transistors in other stages are synthesised in sequence using the NCWSM with different given output rise (or fall) times $T_R(=T_F)$. The resultant total delay time as a function of $T_R(=T_F)$ is shown in Fig. 4, where T_{Dmin} is the minimum total delay time obtained using the NCWSM and the corresponding rise (fall) time is denoted by $T_R^*(T_F^*)$. Once the specified rise (fall) time is equal to $T_R^*(T_F^*)$, the resultant output rise (fall) time of the first inverter is also equal to $T_R^*(T_F^*)$, resulting in the well known taper buffer design. The internal output waveforms are characteristic waveforms. If the specified

output rise or fall time is larger or smaller than T_R^* or T_F^* , the resultant output rise or fall time of the first inverter after sizing is smaller or larger than the specified one because the second inverter has a smaller or larger size than that in the taper buffer case. In both cases, the total delay time is increased.

From Fig. 4, it is also found that, if the total delay time is much larger than T_{Dmin} , the resultant total delay time can be approximated by a linear function of the given rise (fall) time. However, if the total delay time is slightly larger than T_{Dmin} , it can be approximated by a quadratic function. These two approximations will be used to adjust the next rise (fall) time in the fixed-delay sizing algorithm. The detailed algorithm is shown in Fig. 5.

Fig. 5 shows the flow diagram of the sizing algorithm. The first step is to calculate the maximum fan-in number

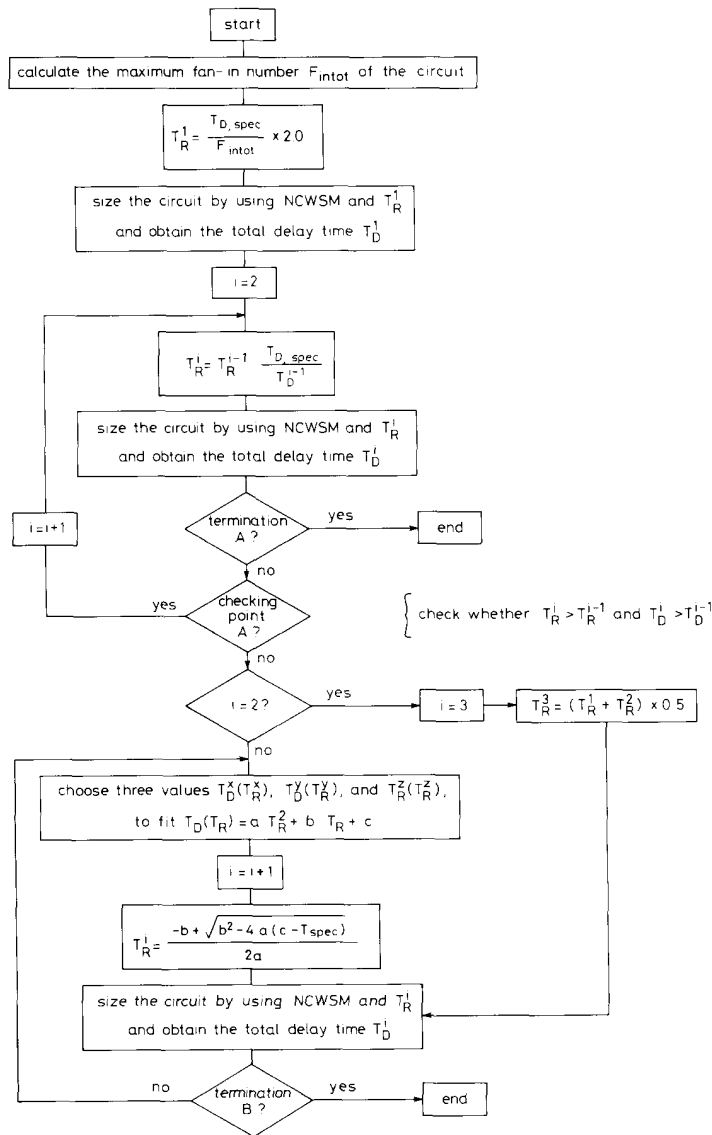


Fig. 5 Proposed algorithm

of the whole circuit. Then the suitable initial value T_R^1 or T_F^1 of the rise or fall time is determined from the specified total delay $T_{D, spec}$ by an empirical equation T_R^1 or $T_F^1 = 2(T_{D, spec}/F_{in, 10})$. It has been shown that the determined initial guessed value leads to a faster convergence than the other values.

In Fig. 5, the check point A is used to check whether the total delay time is a linear or a quadratic function of rise (fall) time. If the smaller rise (fall) time between $T_R^{i-1}(T_F^{i-1})$ and $T_R^i(T_F^i)$ does not lead to the smaller total delay time between T_D^{i-1} and T_D^i , the total delay time as a function of $T_R(T_F)$ is approximated by a quadratic equation. Otherwise, it is approximated by a linear equation. The other two check points, called termination A and termination B in Fig. 5, are used to check whether the termination condition that the error between the resultant and the specified delay times is achieved within 1%. In termination B, whether the specified delay time $T_{D, spec}$ is too small for the circuit is also checked and a message is sent out.

If the delay time is a linear function of the rise (fall) time, the next value of rise (fall) time $T_R^i(T_F^i)$ can be determined by $T_R^i = T_R^{i-1}(T_{D, spec}/T_D^{i-1})$ where T_D^{i-1} is the calculated total delay time with T_R^{i-1} . On the other hand, to determine the coefficients for the quadratic equation, three values are required to fit $T_D(T_R) = aT_R^2 + bT_R + c$. Since the quadratic equation cannot be fitted by two values of rise (fall) time, the third rise and fall times can be determined for $i = 2$. In the case of $i \geq 3$, the fixed-delay algorithm directly selects three delay (rise) times from previous calculations.

From Fig. 4, it is found that, if the specified total delay time is larger than T_{Dmin} , the combinational logic circuits can be sized by two sets of rise or fall times. Since the smaller output rise (fall) time requires a larger channel width, the developed fixed-delay algorithm chooses the larger one to perform the sizing. In general, the required number of iterations using the NCWSM is under 15.

5 Design examples using proposed fixed-delay sizing algorithm

To demonstrate the feasibility of the proposed fixed-delay sizing algorithm, many circuit examples have been tested. Moreover, a comparison between the proposed fixed-delay sizing algorithm and the heuristic approach [1, 3] has been made. In the comparison, the timing models developed by the present authors were used in both algorithms. The increment constant BUMPSIZE [1, 3] used in the heuristic approach was 1.1.

Fig. 6 shows a design example on taper buffers. The comparisons of the minimum realisable delay times using the proposed fixed-delay algorithm and the heuristic approach are listed in Table 2. From Table 2, it is seen

that the difference is only within 2%. To further verify the efficiency of the proposed fixed-delay algorithm, the performance of the sized circuit and the CPU time consumed in both methods are listed in Table 3 for $C_L = 10$ pF and Table 4 for $C_L = 50$ pF. In these data, the power dissipation is defined as the total energy loss

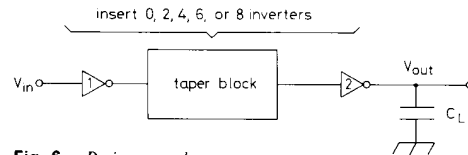


Fig. 6 Design example

Table 2: Comparison of reachable minimum fixed-delay specifications for proposed fixed-delay algorithm and the heuristic approach for design example of Fig. 6 for $C_L = 10$ pF

Number of inserted inverter with $C_L = 10$ pF	Minimum realisable delay time, ns	
	proposed	heuristic
0	6.208	6.174
2	2.624	2.590
4	2.523	2.498
6	2.753	2.727
8	3.078	3.064

during one transition divided by the resultant delay time of the sized circuit [11]. It is found that the power consumption of the circuit sized by using the proposed fixed-delay algorithm and the heuristic approach are of the same order of magnitude. However, the required CPU times and numbers of iterations in the proposed fixed-delay algorithm are much smaller than in the heuristic approach. Generally, the required number of iterations in the proposed fixed-delay sizing algorithm is under 14.

Figs. 7a and b show the reachable minimum delay times and the power consumptions, respectively, as functions of output load capacitance for a 4 bit even-parity checker shown in Fig. 8, which is sized by using the proposed algorithm and the heuristic approach. It is found from Fig. 7a that the reachable minimum delay times using the two sizing methods have only a 5% difference. As to the power consumption of the sized 4 bit even-parity checker, it is smaller in the proposed algorithm than in the heuristic algorithm for the output load capacitances from 2 to 14 pF. Nevertheless, for the output loading from 18 to 20 pF, the power consumption of the 4 bit even-parity checker sized by using the proposed fixed-delay sizing algorithm is larger than that sized by using the heuristic algorithm. By using the heuristic approach, the resultant power consumption of the sized 4 bit even-parity checker with $C_{out} = 14$ pF is larger than that with $C_{out} = 18$ pF because the increment of the

Table 3: Comparison of resultant power consumptions, resultant total delay times, required CPU times and numbers of iterations for proposed fixed-delay algorithm and heuristic approach for design example of Fig. 6 for $C_L = 10$ pF and with 0, 2, 4, 6, and 8 inserted inverters under 6.20, 2.62, 2.52, 2.75, and 3.08 ns fixed-delay specifications, respectively

Number of inserted inverters ($C_L = 10$ pF)		0	2	4	6	8
Fixed-delay specifications, ns		6.20	2.62	2.52	2.75	3.08
Total delay time of sized circuit, ns	proposed	6.209	2.624	2.523	2.753	3.078
	heuristic	6.207	2.621	2.524	2.756	3.079
Power consumption of sized circuit, ns	proposed	1.64	25.99	55.02	74.96	88.17
	heuristic	1.68	26.14	53.68	73.88	83.62
Required number of iterations to achieve specification	proposed	14	10	8	6	5
	heuristic	34	106	176	241	306
Required CPU time consumption to achieve the specification, s	proposed	116.2	129.8	141.3	132.0	139.8
	heuristic	134.5	583.2	1311.1	2352.0	3815.1

Table 4: Comparison of resultant power consumption, resultant total delay time, required CPU time consumption and number of iterations for proposed fixed-delay algorithm and heuristic approach for the design example for Fig. 6 for $C_L = 50$ pF and with 0, 2, 4, 6, and 8 inserted inverters under 13.40, 3.60, 3.02, 3.09, and 3.34 ns fixed-delay specifications, respectively

Number of inserted inverters ($C_L = 50$ pF)		0	2	4	6	8
Fixed-delay specifications, ns		13.4	3.60	3.02	3.09	3.34
Total delay time of sized circuit, ns	proposed	13.43	3.611	3.024	3.097	3.345
	heuristic	13.41	3.602	3.026	3.096	3.344
Power consumption of sized circuit, mW	proposed	1.60	57.98	154.46	248.05	317.60
	heuristic	1.62	61.32	154.34	266.96	335.26
Required number of iterations to achieve specification	proposed	13	10	8	7	6
	heuristic	42	132	219	305	388
Required CPU time consumption to achieve specification, s	proposed	99.8	133.9	141.1	154.7	159.7
	heuristic	142.9	708.9	1578.2	2755.3	4800.3

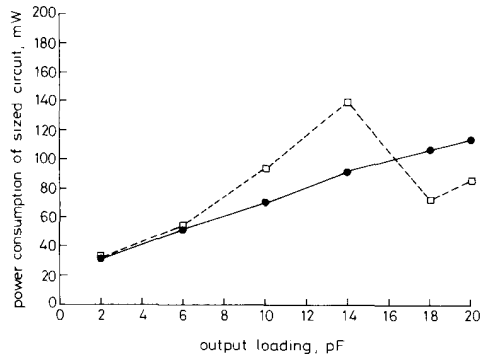
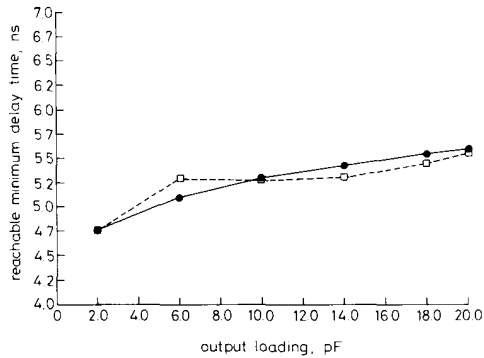


Fig. 7 Comparison of the reachable minimum delay times and power consumptions of developed fixed-delay algorithm and heuristic approach for sizing a 4 bit even-parity checker with different output loading C_{LOAD}

a Reachable minimum delay time
 b Power consumption
 ●— proposed sizing algorithm
 □-- heuristic approach

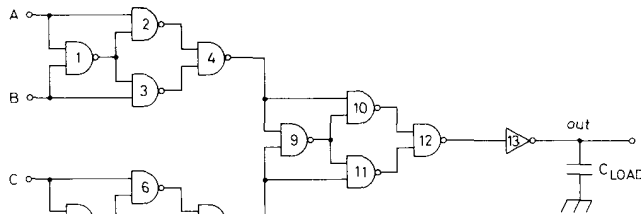


Fig. 8 4 bit even-parity checker

device size in the heuristic approach is discrete and controlled by BUMPSIZE. Although a smaller value of BUMPSIZE could be chosen to reduce the increment of device size and the performance discontinuities, the required CPU time would become intolerably large. This phenomenon is not seen in the sizing operation using the proposed fixed-delay algorithm. Table 5 lists the required

Table 5: Comparison of required CPU time consumption and number of iterations for proposed fixed-delay algorithm and heuristic approach for a 4 bit even-parity checker with different output loading C_{LOAD} for reachable minimum delay time

Output loading C_{LOAD} , pF	Required CPU time, (PC/AT)		Required number of iterations	
	proposed	heuristic	proposed	heuristic
	s	s		
2.0	38.0	2022.3	6	239
6.0	31.2	2121.7	5	257
10.0	31.3	2701.9	5	310
14.0	41.0	2955.6	6	364
18.0	41.0	2442.4	6	293
20.0	41.1	2426.3	6	299

CPU times and numbers of iterations in both methods. It is found that the required CPU time and number of iterations in the proposed fixed-delay algorithm are much smaller than in the heuristic approach.

Fig. 9 shows a benchmark circuit RD53 [12], which contains AOI gates. On sizing the circuit using the heuristic approach, the minimum realisable delay time is 18.8 ns. However, it is only 14 ns on using the proposed fixed-delay algorithm. Table 6 lists the results of the sizing. It is found that the required CPU times and number of iterations in the proposed fixed-delay algorithm are much smaller than in the heuristic approach.

Since the adopted timing models have complicated delay equations, the required calculation time is slightly longer than that of the conventional delay models [9–10]. However, the adopted timing models show a satisfactory accuracy with a wide range of device sizes,

capacitive loads, device parameter variations, input excitation nodes, and input waveforms not deviating much from characteristic waveforms [8]. This greatly enhances the sizing accuracy. To demonstrate this, delay comparisons between calculation and SPICE simulation on the

error of the delay times between SPICE simulation and model calculation is also under 15%.

As seen in Table 7, the resultant delay times of the sized circuit are smaller than the specified delay time. This is because the worst-case delay consideration for the

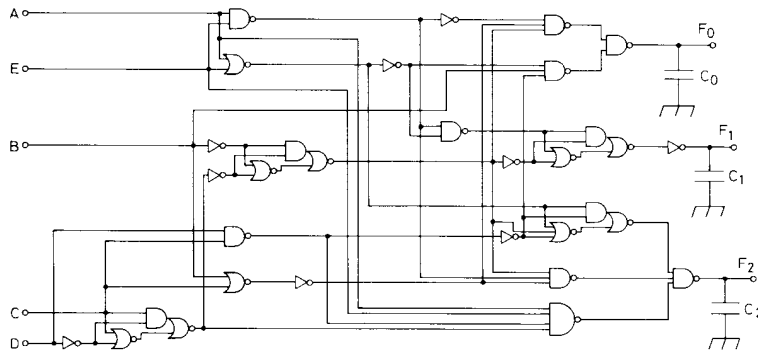


Fig. 9 Benchmark circuit RD53

Table 6: Comparison of required CPU time consumption and number of iterations for the developed fixed-delay algorithm and the heuristic approach for sizing the benchmark circuit RD53 under different fixed-delay specifications

Fixed-delay specification	Required CPU time, (PC/AT)		Required number of iterations	
	proposed	heuristic	proposed	heuristic
ns	s	s		
14.0	20.4	x	2	x
15.0	40.6	x	4	x
16.0	52.8	x	5	x
17.0	63.2	x	6	x
18.0	61.7	x	6	x
18.8	51.2	1825.3	5	86

sized circuits are performed. First, a string of 40 CMOS inverters with a fixed-delay specification of 20 ns were sized by the proposed fixed-delay sizing algorithm. The total delay time of the sized circuit obtained from SPICE simulations is 17.3 ns. The error of 13.5% is within the maximum error of the developed timing models. A 4 bit even-parity checker was also sized under different delay specifications and with the input A excited by an exponential input rising waveform ($\tau = 0.3$ ns) and the other nodes (B, C, and D) maintained at V_{DD} . The comparisons of the model calculation and SPICE simulation are given in Table 7. In Table 7, it is found that the maximum

multi-input logic gates is used in the proposed fixed-delay algorithm. Without this consideration, the delay time of the sized circuit could not be under the specified value and the safe design would not be achieved.

6 Conclusions

Through extensive comparison and experimental verification, it has been proved that the developed sizing algorithm for CMOS combinational logic circuits (inverters, multi-input NAND and NOR, AOI, and OAI) has the advantageous features of high computation efficiency and low CPU-time consumption. Compared with the heuristic approach [1, 3], the proposed fixed-delay algorithm can size the circuits with nearly the same power dissipation but less CPU time and fewer iterations, especially when the circuits are complex. Moreover, it can also size the circuits with smaller reachable minimum delay. Owing to the adopted accurate physical timing models, the sized circuit has a good accuracy in delay when compared to SPICE simulation results. These advantages make the proposed new sizing algorithm attractive and versatile in many VLSI CAD applications.

7 References

Table 7: Comparison of SPICE simulation and model calculation for a 4 bit even-parity checker sized by using the proposed fixed-delay algorithm and under different fixed-delay specifications

Fixed-delay specification	Required CPU time (PC/AT)		Resultant delay time		
	SPICE	model	SPICE	model	error
ns	s	s	ns	ns	%
6.5	405.6	2	6.02	5.17	14.1
7.0	405.4	2	6.40	5.58	12.8
8.0	405.0	2	6.54	5.74	12.2
9.0	405.4	2	6.67	5.87	11.9
10.0	405.2	2	7.14	6.29	11.9
11.0	405.6	2	7.70	6.78	11.9
13.0	404.6	2	8.43	7.44	11.7
15.0	404.5	2	8.87	7.78	12.2

- FISHBURN, J.P., and DUNLOP, A.E.: 'TILOS: A polynomial programming approach to transistor sizing'. Proceedings of International Conference on Computer aided design, 1985, pp. 326-328
- KAO, W.H., FATHI, N., and LEE, C.H.: 'Algorithm for automatic transistor sizing in CMOS digital circuits'. Proceedings of 22nd Design Automation Conference, 1985, pp. 781-785
- SHYU, J.M., SANGIOVANNI-VINCENTELLI, A., FISHBURN, J.P., and DUNLOP, A.E.: 'Optimization-based transistor sizing', *IEEE Trans.*, 1988, SC-23, pp. 400-409
- LEE, C.M., and SOUKUP, H.: 'An algorithm for CMOS timing and area optimization', *IEEE Trans.*, 1984, SC-19, pp. 781-787
- WU, C.Y., HWANG, J.S., CHANG, C., and CHANG, C.C.: 'An efficient timing model for CMOS combinational logic gates', *IEEE Trans.*, 1985, CAD-4, pp. 636-650
- WU, C.Y., LI, C., and HWANG, J.S.: 'Timing macromodels for CMOS static set/reset latches and their applications', *IEE Proc. E*, 1988, 135, pp. 151-160
- WU, C.Y., and HWANG, J.S.: 'Physical timing models of small-geometry CMOS inverters and multi-input NAND/NOR gates and their applications', *Solid-State Electron.*, 1989, 32, (6), pp. 449-467

- 8 WU, C.Y., and SHIAO, M.C.: 'General and efficient timing models for CMOS AND-OR-inverter and OR-AND-inverters gates', *IEEE Trans.*, 1991, **CAD-10**, pp. 1002-1009
- 9 BURNS, J.R.: 'Switching response of complementary-symmetry MOS transistor logic circuits', *RCA Rev.*, 1964, pp. 627-661
- 10 KANUMA, A.: 'CMOS circuit optimization', *Solid-State Electron.*, 1983, **26**, (1), pp. 47-58
- 11 HWANG, J.S., and WU, C.Y.: 'Efficient techniques in the sizing and constrained optimisation of CMOS combinational logic circuits', *IEE Proc. E*, to be published.
- 12 MICHELI, M.D., SANGIOVANNI-VINCENTELLI, A., and ANTOGNETTI, P.: 'Design system for VLSI circuits logic synthesis and silicon compilation' (Netherlands, 1981).