Efficient techniques in the sizing and constrained optimisation of CMOS combinational logic circuits

J.-S. Hwang
C.-Y. Wu

Indexing terms: Optimisation, Logic

Abstract: Two techniques are proposed which enhance the optimisation efficiency of CMOS combinational logic circuits. One uses transition times (rise and fall times) of each gate as variables of the optimisation process. The other technique uses the optimal characteristic waveform synthesising method (OCWSM) to obtain the initial guess for the optimisation process. The optimisation process, with these two techniques, can perform sizing and optimisation for circuits with a smaller fixed-delay specification than other sizing and optimisation algorithms. The circuits sized using the proposed algorithm have shown a smaller power dissipation, especially when the delay specification is small. The CPU time consumed is reasonable. High-speed low-power circuits are thus more realisable using the proposed algorithm.

List of symbols

$C_{bdp(n)f1}$ = drain-bulk junction capacitance of PMOSFET(NMOSFET) in CMOS inverter, estimated at output voltage $V_o = (V_{DD} + V_{DSATN})/2$

$C_{bdp(n)f2}$ = drain-bulk junction capacitance of PMOSFET(NMOSFET) in CMOS inverter, estimated at output voltage $V_o = V_{DSATN}/2$

$C_{bdp(n)r1}$ = drain-bulk junction capacitance of PMOSFET(NMOSFET) in CMOS inverter, estimated at output voltage $V_o = (V_{DD} - V_{DSATP})/2$

$C_{bdp(n)r2}$ = drain-bulk junction capacitance of PMOSFET(NMOSFET) in CMOS inverter, estimated at output voltage $V_o = (V_{DD} + V_{DSATP})/2$

$C_{gdp(n)f1}$ = gate-drain overlap capacitance of PMOSFET(NMOSFET) in CMOS inverter, estimated when output voltage V_o lowers from V_{DD} to V_{DSATN} and NMOSFET is operated in saturation region

$C_{gdp(n)f2}$ = gate-drain overlap capacitance of PMOSFET(NMOSFET) in CMOS inverter, estimated when output voltage V_o lowers from V_{DSATN} to 0 V and NMOSFET is operated in linear region

$C_{gdp(n)r1}$ = gate-drain overlap capacitance of PMOSFET(NMOSFET) in CMOS inverter, estimated when output voltage V_o raises from 0 V to $V_{DD}-V_{DSATP}$ and PMOSFET is operated in saturation region

$C_{gdp(n)r2}$ = gate-drain overlap capacitance of PMOSFET(NMOSFET) in CMOS inverter, estimated when output voltage V_o raises from $V_{DD}-V_{DSATP}$ to V_{DD} and PMOSFET is operated in linear region

$C_{gsp(n)f1}$ = gate-source overlap capacitance of PMOSFET(NMOSFET) in CMOS inverter, estimated when output voltage V_o lowers from V_{DD} to V_{DSATN} and NMOSFET is operated in saturation region

$C_{gsp(n)f2}$ = gate-source overlap capacitance of PMOSFET(NMOSFET) in CMOS inverter, estimated when output voltage V_o lowers from V_{DSATN} to 0 V and NMOSFET is operated in linear region

$C_{gsp(n)r1}$ = gate-source overlap capacitance of PMOSFET(NMOSFET) in CMOS inverter, estimated when output voltage V_o raises from 0 V to $V_{DD} - V_{DSATP}$ and PMOSFET is operated in saturation region

$C_{gsp(n)r2}$ = gate-source overlap capacitance of PMOSFET(NMOSFET) in CMOS inverter, estimated when output voltage V_o raises from $V_{DD}-V_{DSATP}$ to V_{DD} and PMOSFET is operated in linear region

C_L = output capacitive loads

$GAMMA$ = bulk threshold parameter (SPICE device parameter)

$L_{effp(n)}$ = effective channel length of PMOSFET(NMOSFET)

P_{fi} = equivalent fall pole of input waveform defined as $P_{fi} = (\ln 2)/T_{Fi}$

P_{ri} = equivalent rise pole of input waveform defined as $P_{ri} = (\ln 2)/T_{Ri}$

T_F = fall time which is time interval within which output voltage lowers from 0.9 V_{DD} to 0.1 V_{DD}

T_{Fi} = fall time which is time interval within which input voltage lowers from 0.9 V_{DD} to 0.3 V_{DD}

T_{PHL} = fall delay time which is time interval between input voltage $V_i = 0.5\ V_{DD}$ and output voltage $V_o = 0.5\ V_{DD}$

T_{PLH} = rise delay time which is time interval between input voltage $V_i = 0.5\ V_{DD}$ and output voltage $V_o = 5\ V_{DD}$

T_R = rise time which is time interval within which output voltage raises from 0.1 V_{DD} to 0.9 V_{DD}

Paper 7946E (C2, E10), first received 18th June and in revised form 12th November 1990

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

T_{Ri} = rise time which is time interval within which input voltage raises from 0.1 V_{DD} to 0.9 V_{DD}
T_{ox} = channel oxide thickness
V_{DSATN} = velocity saturation voltage of NMOSFET
V_{DSATP} = velocity saturation voltage of PMOSFET
V_{TNF} = final threshold voltage of n-channel MOSFET under the condition that gate voltage $V_G = V_{DD}$ and source voltage $V_S = V_{TNF}$
V_{TO} = zero-bias threshold voltage of MOSFET(SPICE device parameter)
$W_{N(P)}$ = channel width of $n(p)$-channel MOSFET
$\varepsilon_{Si(SiO_2)}$ = permittivity of Si semiconductor (silicon dioxide)
$\lambda_{n(p)}$ = effective channel length modulation parameter (SPICE device parameter)
$\eta_{n(p)}$ = $n(p)$-channel parameter η
$\gamma_{sn(p)}$ = effective short-channel N(P)MOSFET GAMMA
ϕ_F = Fermi potential
$\phi_{Fn(p)}$ = Fermi potential of $n(p)$-type silicon
μ_s = surface mobility of carriers
$\mu_{n(p)}$ = effective $n(p)$-channel μ_s

1 Introduction

Sizing the transistors of a CMOS digital IC when optimising various circuit performance parameters such as delay time, power dissipation and chip area has been an important and challenging area of work. There have been many publications [1–14] which have discussed sizing and optimisation algorithms for MOS logic circuits. Some consider the transistor sizing along the critical delay path (the selected path) only [3–9, 11, 12]. This lacks a global view of the whole circuit. Others [1, 2, 10, 13] apply mathematical optimisation techniques to solve the transistor sizing problem. The design object is formulated as a nonlinear mathematical equation, with the desired specifications as constraints, and then optimised through a timing analyser [10, 13] or a circuit simulator [1, 2]. For the sizing and optimisation using general-purpose circuit simulators [1, 2], the manageable circuits are typically restricted to those with at most thirty design parameters [10]. For the sizing and optimisation using timing analysers, no such restrictions exist.

In all the sizing and optimisation algorithms proposed [1–14], the device sizes have been chosen as independent variables of the optimisation process because they are the desired solution. Since the relation between delay time and device sizes is very complex, the efficiency in the optimisation using device sizes as optimisation variables may be degraded. The initial guess (initial device sizes) of the optimisation process are either arbitrarily chosen by the user [10] or obtained from the heuristic approach [13]. The required circuit performance is usually different from that achieved when using the user-assigned device sizes (usually the minimum allowable device sizes). This may cause difficulties in the optimisation process because large iteration number and large CPU time consumption may result. The convergence speed can be greatly enhanced by using the heuristic approach to obtain the initial guess for the optimisation process. As the specified delay time approaches the optimal value, the required CPU time in the heuristic approach is intolerable even for a medium-size circuit [13].

This paper aims to solve the above mentioned problems. Two techniques in sizing CMOS combinational logic circuits are proposed. The output transition times (rise and fall times) of each gate, instead of the device sizes, are chosen as independent variables of the optimisation process. The optimised transition times are then used to calculate the required device sizes using the physical timing models [15–18]. The initial guess of transition times in the optimisation process is obtained from a simple and quick pre-optimiser which uses the optimal characteristic waveform-synthesising method (OCWSM) [19, 20] to obtain near optimal transition times with much less CPU time.

These two techniques are applied to the minimisation of power dissipation with fixed-delay specifications for the sizing of CMOS combinational logic circuits. Many optimisation methods [21, 23] can solve this constrained optimisation problem. Minimising the augmented Lagrangian function of the equality constrained problem combines the advantages of the penalty and the primal-dual approaches [21]. It is therefore adopted in the minimisation of power dissipation with fixed-delay specifications. The Davidon-Fletcher-Powell method [21, 23] is an optimisation method with quadratic convergence rates. This method uses only the functional values and gradient vectors in generating mutually conjugate search directions. It is very suitable for the sizing of CMOS combinational logic circuits from the computation point of view. However, the optimisation result may converge to a nonoptimal point because of the round-off errors, numerical errors, inaccurate line searches and nonquadratic terms in the objective function [23]. The self-scaling and restarting quasi-Newton method [23] is thus adopted in solving the above mentioned optimisation problem.

The adopted timing models must be accurate when using timing analysers for the sizing and optimisation. Otherwise, the results of sizing, optimisation, and timing verification would have an unbearable error. Many timing models [4, 11, 12, 15–18, 24–32] and timing simulators [33] have been developed. The physical timing models developed by the present authors [15–18] show a satisfactory accuracy for CMOS combinational gates with wide ranges of device sizes, capacitive loads, device parameter variations and input excitation waveforms. Although the models have complicated equations of gate delay times as a function of device parameters, the required CPU time is still much smaller than that in conventional circuit simulators.

The physical timing models of CMOS static logic gates are described. The power dissipation model of CMOS static logic gates are given. The details of the proposed techniques are given and some experimental results to verify the proposed techniques presented. Finally, conclusions and discussions are given.

2 Physical timing models of CMOS combinational gates

The timing models [15–18] adopted in the sizing and optimisation process are developed by deriving, region by region, the analytical formulas of the rise/fall times (T_R/T_F) from the linearised large-signal equivalent circuit of a CMOS logic gate, under the characteristic-waveform consideration. Tables 1 and 2 list the expressions of rise and fall times for short-channel CMOS inverters, respectively. The rise and fall delay times T_{PLH} and T_{PHL} are semi-empirically expressed in terms of the calculated rise/

fall times as

$$T_{PLH} = a_{rr} T_R + a_{rf} T_{Fi} + \frac{\ln 2}{\ln 9} T_R - \frac{\ln 2}{\ln 9} T_{Fi} \quad (1)$$

$$T_{PHL} = a_{fr} T_{Ri} + a_{ff} T_F + \frac{\ln 2}{\ln 9} T_F - \frac{\ln 2}{\ln 9} T_{Ri} \quad (2)$$

where a_{rr}, a_{rf}, a_{fr}, and a_{ff} are universal empirical constants for the initial delay. The above equations are universal for different device and circuit parameters of the logic gate. Applying this modelling technique, timing models have been derived for CMOS inverters [15, 17], multi-input NAND and NOR gates [15, 17], AOI and

Table 1: Rise time equations for short-channel CMOS inverters

$$T_R = \begin{cases} \frac{1}{P_{rs}} \ln\left(\frac{0.9\,V_{DD} + 1/\lambda_p}{0.1\,V_{DD} + 1/\lambda_p}\right) & V_{DSATP} < 0.1\,V_{DD} \\ \frac{1}{P_{rs}} \ln\left(\frac{0.9\,V_{DD} + 1/\lambda_p}{V_{DSATP} + 1/\lambda_p}\right) + \frac{1}{P_{r1}} \ln\left(\frac{V_{DSATP}}{0.1\,V_{DD}}\right) & V_{DSATP} > 0.1\,V_{DD} \end{cases}$$

where

$$P_{rs} = (\lambda_p I_{DPO})/C_{R1}$$

$$P_{r1} = G_p/C_{R2}$$

$$C_{R1} = C_{gdp1} + C_{gdn1} + C_{bdp1} + C_{bdn1} + C_L + C_{next,\,gate}$$

$$C_{R2} = C_{gdp2} + C_{gdn2} + C_{bdp2} + C_{bdn2} + C_L + C_{next,\,gate}$$

$$I_{DPO} = \frac{W_P}{L_{effp}} \frac{\varepsilon_{SiO_2}}{T_{ox}} \frac{\mu_p}{1 + \lambda_p V_{DSATP}} Rx$$

$$G_p = \frac{W_p}{L_{effp}} \frac{\varepsilon_{SiO_2}}{T_{ox}} \mu_p Ry$$

$$Rx = \left[V_{DD} - R1 - V_{binp} - \frac{\eta}{2} pV_{DSATP}\right] V_{DSATP} - \tfrac{2}{3}\gamma_{sp}[(2\phi_{Fn} + V_{DSATP})^{1.5} - (2\phi_{Fn})^{1.5}$$

$$Ry = V_{DD} - R1' - V_{binp} - \frac{\eta}{4} pV_{DSATP} - \tfrac{2}{3}\gamma_{sp}\left(2\phi_{Fn} + \frac{V_{DSATP}}{2.0}\right)0.5$$

$$V_{binp} = V_{TOp} - GAMMA(2\phi_{Fn})^{0.5} + (\eta - 1)(2\phi_{Fn})$$

$$R1 = V_{DD} \exp(-0.4)\left(\frac{V_{DD} + 1/\lambda_p}{(V_{DD} + V_{DSATP})/2 + 1/\lambda_p}\right)^{-(P_{ri}/P_{rs})}$$

$$R1' = V_{DD} \exp(-0.4)\left(\frac{V_{DD} + 1/\lambda_p}{V_{DSATP} + 1/\lambda_p}\right)^{-(P_{ri}/P_{rs})} 2.0^{-(P_{ri}/P_{r1})}$$

Table 2: Fall time equations for short-channel CMOS inverters

$$T_F = \begin{cases} \frac{1}{P_{fs}} \ln\left(\frac{0.9\,V_{DD} + 1/\lambda_n}{0.1\,V_{DD} + 1/\lambda_n}\right) & V_{DSATN} < 0.1\,V_{DD} \\ \frac{1}{P_{fs}} \ln\left(\frac{0.9\,V_{DD} + 1/\lambda_n}{V_{DSATN} + 1/\lambda_n}\right) + \frac{1}{P_{r1}} \ln\left(\frac{V_{DSATN}}{0.1\,V_{DD}}\right) & V_{DSATN} > 0.1\,V_{DD} \end{cases}$$

where

$$P_{fs} = (\lambda_p I_{DPO})/C_{F1}$$

$$P_{f1} = G_p/C_{F2}$$

$$C_{F1} = C_{gdp1} + C_{gdn1} + C_{bdp1} + C_{bdn1} + C_L + C_{next,\,gate}$$

$$C_{F2} = C_{gdp2} + C_{gdn2} + C_{bdp2} + C_{bdn2} + C_L + C_{next,\,gate}$$

$$I_{DNO} = \frac{W_N}{L_{effn}} \frac{\varepsilon_{SiO_2}}{T_{ox}} \frac{\mu_n}{1 + \lambda_n V_{DSATN}} Fx$$

$$G_p = \frac{W_p}{L_{effp}} \frac{\varepsilon_{SiO_2}}{T_{ox}} \mu_p Fy$$

$$Fx = \left[V_{DD} - F1 - V_{binn} - \frac{\eta_n}{2} V_{DSATN}\right] V_{DSATN} - \tfrac{2}{3}\gamma_{sn}[(2\phi_{Fp} + V_{DSATN})^{1.5} - (2\phi_{Fp})^{1.5}]$$

$$Fy = V_{DD} - F1' - V_{binn} - \frac{\eta_n}{4} V_{DSATN} - \tfrac{2}{3}\gamma_{sn}\left(2\phi_{Fp} + \frac{V_{DSATN}}{2.0}\right)^{0.5}$$

$$V_{binn} = V_{TOn} - GAMMA(2\phi_{Fp})^{0.5} + (\eta - 1)(2\phi_{Fp})$$

$$F1 = V_{DD} \exp(-0.6)\left(\frac{V_{DD} + 1/\lambda_n}{(V_{DD} + V_{DSATN})/2 + 1/\lambda_n}\right)^{-(P_{ri}/P_{fs})}$$

$$F1' = V_{DD} \exp(-0.6)\left(\frac{V_{DD} + 1/\lambda_n}{V_{DSATN} + 1/\lambda_n}\right)^{-(P_{ri}/P_{fs})} 2.0^{-(P_{ri}/P_{fi})}$$

OAI gates [18], and static flip-flops [16] with MOS device channel length (mask) down to 1.5 μm.

The accuracy of the timing models has been widely verified through extensive comparisons between model calculation and SPICE simulation. Part of the comparisons are shown in Fig. 1 for 2 μm CMOS inverters with $C_L = 0$ pf (only one fanout) under characteristic-waveform consideration. Fig. 2 shows the comparisons of 2 μm CMOS inverters with $C_L = 0$ pf (only one fanout) under the exponential input excitations with time constants from 0.4 to 4.0 ns. It is found through the accuracy verification that the maximum error is under 15% for inverters with different device dimensions, capacitive loads, input waveforms, device parameters and temperatures. Fine tuning can further decrease the error to below 7% [17].

The timing models should be able to characterise the timing of multi-input logic gates excited at any input node to accurately calculate the delay time of a logic circuit. A CMOS three-input NAND gate, as shown in Fig. 3, is considered. The timing is of the worst-case type if only the input node 1 is excited. The other inputs nodes are stabilised at V_{DD}. If the node 2 or the node 3 is excited and the other nodes are stabilised at V_{DD}, the timing is not the worst case type. According to our observations, the longest rise delay of a CMOS three-input NAND gate with one fanout gate is about 58% longer than the corresponding shortest one. The timing models [15, 18] developed by the authors considered all the triggering cases. Part of the comparisons between the SPICE simulation and the model calculation are listed in Table 3 for 2 μm CMOS three-input NAND gates in different triggering cases. It is shown through the accuracy verification that the maximum error of the developed timing models is under 15% for CMOS multi-input NAND/NOR gates with wide ranges of device sizes, capacitive loads and device parameter variations. The input voltage waveforms were not deviating much from characteristic waveforms. Fine tuning can further reduce the maximum error to below 10% [17].

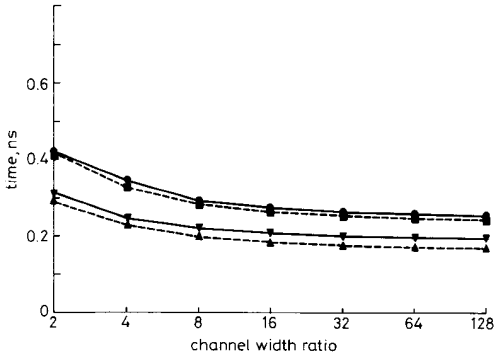

Fig. 1 *Rise and fall delays against channel width ratio*
2.0 μm CMOS inverters
Characteristic waveform consideration $C_L = 0$ pF; $W_P = W_N$; $L_{effp}/L_{effn} = 0.9/1.1$ μm
—●— SPICE rise delay
--■-- theory rise delay
—▼— SPICE fall delay
—▲— theory fall delay

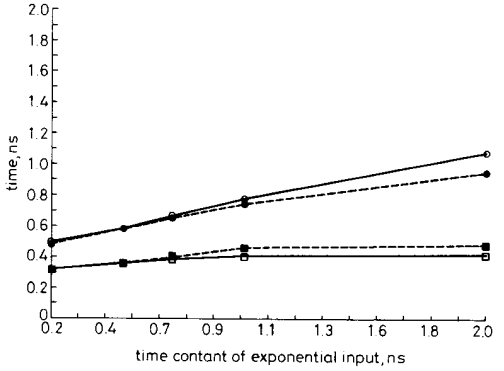

Fig. 2 *Rise and fall delays against time constants*
2.0 μm CMOS inverters
Exponential input excitations
$C_L = 0$ pF; $W_P = W_N = 2$ μm
—○— SPICE rise delay
--●-- theory rise delay
—□— SPICE fall delay
--■-- theory fall delay

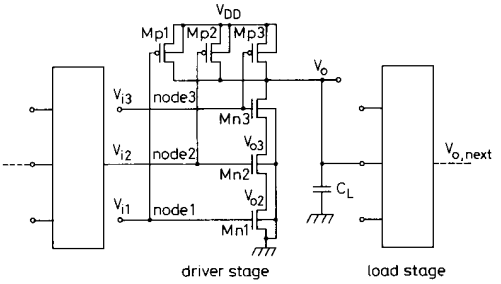

Fig. 3 *CMOS three input NAND gate*
Triggering at node 2

Table 3: Signal timing of 2.0μm CMOS three input NAND gates

Triggered node	Signal timing	SPICE ns	Theory ns	Error %
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

$W_P = 2.0$ μm; $W_N = 2.0$ μm; $C_L = 0$ pF.
* Node nearest output node
† Node furthest from output

A similar error characteristic can be obtained for the timing models of small-geometry CMOS AOI/OAI gates [18].

3 General power dissipation models of CMOS combinational gates

Consider a string of CMOS three-input NAND gates excited at the node 2 as shown in Fig. 3. Typical output voltage characteristic waveforms of V_{i2}, V_o, V_{o3} and V_{o2} are shown in Fig. 4. As the input voltage V_{i2} rises from 0 V to V_{DD}, the output voltages V_o and V_{o3} fall from V_{DD} to 0 V and $(V_{DD} - V_{TNF})$ to 0 V, respectively. V_{TNF} is the

threshold voltage of an n-channel MOSFET with the source-bulk voltage $V_{SB} = V_{DD} - V_{TNF}$. The output voltage V_{o2} first raises from 0 V to V_p and then falls from V_p to 0 V.

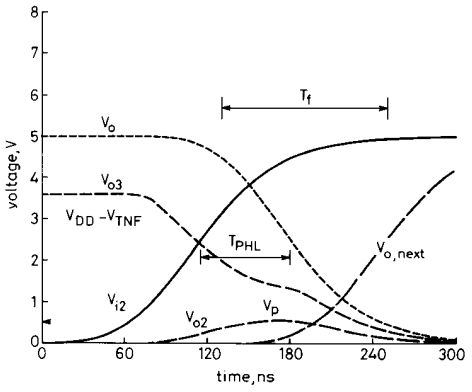

Fig. 4 *Typical fall characteristic of CMOS three input NAND gate*
Triggering at node 2
C_L = 5.0 pF

To neglect the short-circuit power dissipation is an acceptable approximation as long as the short-circuit power dissipation is small compared with the dynamic power dissipation needed to charge the capacitor [11]. According to our observations, the energy loss of a CMOS logic circuit is mainly caused by the charging and discharging of device capacitances during the output rising/falling transition periods when the CMOS logic circuits are operated under the characteristic-waveform consideration. Only the dynamic power dissipation is considered.

There are four different types of device capacitances: voltage-dependent drain (source)-bulk junction capacitance, voltage-dependent gate-drain (source, bulk) capacitance, voltage-independent gate-drain (source) overlap capacitance, and external voltage-independent capacitances. The energy loss during the charging/discharging cycle of a capacitor is equal to the change of the energy stored on the capacitor [34]. To characterise the change of the energy stored on the voltage-dependent pn junction capacitances, the case of the drain-bulk junction capacitance is considered. The change of stored energy from the drain-bulk voltage $V_{DB} = 0$ V to $V_{DB} = V_{bias}$ is derived in the Appendix. For CMOS three-input NAND gates excited at the node 2, the change of the energy stored on voltage-dependent drain-bulk junction capacitances can be further divided into three subgroups.

(*a*) Drain-bulk junction capacitances at the output node: In a typical CMOS logic gate, the substrate of the PMOS is connected to the positive power supply V_{DD}. When the output voltage is V_{DD}, the voltage across the drain-bulk junction capacitance of the PMOSFET is 0 V. There is no charge stored on this capacitor. The voltage across this capacitance is V_{DD} when the output voltage is 0 V. The change of the energy during each transition period can then be expressed as

$$\mathscr{E}_p = CJ_p A_{D,p} F_{area,p}(V_{DD}) + CJSW_p P_{D,p} F_{peri,p}(V_{DD}) \quad (3)$$

where CJ_p is the zero-bias bulk capacitance of the PMOSFET, $CJSW_p$ is the zero-bias perimeter capacitance of the PMOSFET, $A_{D,p}$ is the drain area of the PMOSFET, $P_{D,p}$ is the drain perimeter of the PMOSFET, $F_{area,p}(V_{DD})$ and $F_{peri,p}(V_{DD})$ can be found from the Appendix for the PMOSFET.

The change of the energy stored on the drain-bulk junction capacitance of the NMOSFET can be also expressed by using the same derivation technique.

(*b*) Drain-bulk junction capacitances of internal series NMOSFETs: The voltage swing of V_{o3} at the node 3 is $V_{DD} - V_{TNF}$. Thus, the change of the energy on the drain-bulk junction capacitor at this node is

$$\mathscr{E}_n = CJ_n A_{D,n} F_{area,n}(V_{DD} - V_{TNF}) + CJSW_n P_{D,n} F_{peri,n}(V_{DD} - V_{TNF}) \quad (4)$$

where CJ_n is the zero-bias bulk capacitance of the NMOSFET, $CJSW_n$ is the zero-bias perimeter capacitance of the NMOSFET, $A_{D,n}$ is the drain area of the NMOSFET, $P_{D,n}$ is the drain perimeter of the NMOSFET, $F_{area,n}(V_{DD} - V_{TNF})$ and $F_{peri,n}(V_{DD} - V_{TNF})$ can be found from the Appendix for the NMOSFET.

(*c*) Internal inactive drain-bulk junction capacitances: From Fig. 4, it is found that the voltage at the node 2 (V_{o2}) before the transition period is 0 V, so is the voltage after the transition period. The bias voltage V_{bias} is 0 V. From the Appendix, the change of the energy on such an inactive drain-bulk junction capacitor is 0, as is the power dissipation.

The change of the energy stored on the voltage-dependent source-bulk junction capacitor can also be formulated. For the voltage-dependent gate-drain (source, bulk) capacitance, its energy change is calculated region by region with suitable capacitance values determined by the device operating regions. For the voltage-independent gate-drain (source) overlap capacitance and external voltage-independent capacitances, the energy change can be easily characterised. The total energy loss during the rising/falling transition period can be determined by summing up the changes of the energy in a logic gate. The average power consumption of a logic circuit can be calculated by using the definition

$$\mathscr{P} \equiv \left(\sum_{i=0}^{N} \mathscr{E}_i\right) \Big/ T_{D,max} \quad (5)$$

where N is the total gate number in the circuit, $\mathscr{E}_i$ is the energy loss of the gate i, and $T_{D,max}$ is the critical maximum delay time of the logic circuit under operation, which is the delay time of the critical path.

It should be emphasised that the energy losses of a logic gate are different for different excitation inputs and so are the average power dissipations of a logic circuit. The power dissipation models developed can characterise those different power dissipations.

4 Sizing and constrained optimisation

As an example of the proposed techniques in the sizing and constrained optimisation, the minimisation of power dissipation with fixed-delay specifications in the sizing of CMOS combinational logic circuits is considered. The same techniques can also be applied to improve the optimisation efficiency for other optimisation problems.

Minimising the power dissipation, $\mathscr{P}$, with fixed-delay specifications is a nonlinear constrained optimisation problem which is defined as

$$\text{minimise } \mathscr{P} \quad (6)$$

such that

$$\boldsymbol{T}_D - \boldsymbol{T}_{SPEC} = \boldsymbol{0}$$

where

$$T_D = [T_{d1}, T_{d2}, \ldots, T_{dm}]'$$

$$T_{SPEC} = [T_{s1}, T_{s2}, \ldots, T_{sm}]'$$

In the above equations, m is the number of delay constraints. T_{d1}, T_{d2}, ..., T_{dm} are delay times of nodes 1, 2, ..., m, respectively. T_{s1}, T_{s2}, ..., T_{sm} are the specified delay times at nodes 1, 2, ..., m, respectively.

4.1 Using transition times as independent optimisation variables

Transistor sizes are conventionally chosen as optimisation variables in the sizing and constrained optimisation. They are manipulated to obtain optimisation directions and steps toward a given delay in the constrained optimisation with specified delay times as design constraints. The delay time is a very complicated function of transistor sizes as may be seen from the physical timing models [15–18]. It is found that such a complex function leads to many difficulties in mathematical treatment.

The rise/fall and delay times of a MOS logic gate are generally determined by

(i) driving capability

(ii) internal gate capacitances

(iii) load capacitance or resistance contributed by the loading gates

(iv) load capacitance or resistance contributed by the interconnection line or the on-chip or off-chip fixed capacitive loads

(v) rise/fall times of the input waveforms

(vi) excited input nodes or the input excitation patterns.

The first two factors are related to device sizes. The last two factors are associated with input excitations. If the output loading of a logic gate and the input excitations are known, the output transition times can be determined from the device sizes. The device sizes of a logic gate can be also determined from its output transition times if the output loading of a logic gate and the input excitations are known.

In sizing a MOS digital IC, the logic structure, the input waveforms to the circuit, the output off-chip loading, and technology and device parameters are known. In combinational logic circuits, the output off-chip loading becomes the loading of the last stage in each of the signal paths. If input excitation patterns and output rise and fall times of the last stage are given, their device sizes are the only unknown factors in the rise-time and fall-time equations [15–18] of the timing models. They can then be calculated from these timing equations. Having obtained the device sizes of the last stage, the output loading contributed by the last stage to the stage preceding the last stage can be determined. If input excitation patterns and output rise and fall times of those stages are given, their device sizes can be also calculated by solving their timing equations. This implies that if the output rise and fall times of each logic gate are known, the timing synthesis of combinational logic circuits can be achieved using the last stage of each signal path to the first stage of the path simultaneously. The sizing can be performed simultaneously and globally from all the output stages to all the input stages. It is therefore feasible to treat rise and fall times of the gates in a circuit as independent variables in the sizing and optimisation process. In each optimisation step, the corresponding device sizes in each gate can be calculated from the rise/fall times by using the timing equations. Since the delay time of a CMOS logic gate is approximately a linear function of the rise and fall times as described earlier, it is expected that the optimisation using rise and fall times as optimisation variables is more optimal and/or has faster convergent rate than that using device sizes as optimisation variables.

In the synthesising process, the resultant rise and fall times may be larger or smaller than those in practical circuits. The synthesised device sizes are thus smaller than the user-specified minimium allowable channel widths or larger than the user-specified maximum allowable channel widths. The device sizes are reset to the minimum or maximum allowable values and the transition times are reset to the corresponding values to solve this problem.

In the optimisation, the ratio of rise time to fall time for all the logic gates can be defined by users. Symmetrical rise and fall transitions is one of the most important design issues so the ratio of rise time to fall time for all the logic gates is considered to be unity. All MOSFETs in series are designed with equal channel widths as are all MOSFETs in parallel.

4.2 Using the OCWSM to obtain a set of device sizes as the initial guess

In the design of a tapered buffer, the minimum total delay can be obtained by equalising the delay in each stage [3, 15–18, 28]. The resultant rising or falling waveform in each stage is the same, being the characteristic waveform. The characteristic waveform appears in any minimum delay path of identical logic gates. In a minimum delay signal path with different types of logic gates, although the exact characteristic waveform does not appear, the deviation of the actual waveform from the characteristic waveform is not so significant because of the similarities among these inverting logic gates. It is expected that actual waveforms in an optimally designed chip are close to the characteristic waveforms. Based on these considerations, a quick sizing method was developed called the optimal characteristic waveform synthesising method (OCWSM) [19, 20].

The designer chooses the ratio of rise time to fall time in all the gates with the OCWSM. The ratio of the output rise (or fall) time to the fan-in number of a gate is also fixed. Given an initially guessed value of rise or fall time, other rise/fall times of all the gates can be found through the two fixed ratios. If the ratio of the rise time to fall time is unity and the rise time of CMOS inverter is T_x, the fall time of CMOS inverter is T_x and the rise and fall times of two-input CMOS NAND gate are $2T_x$. OCWSM finds a value of T_x to achieve the minimum delay. This is a single variable optimisation problem and the OCWSM can quickly find the optimal value of rise (or fall) time for the minimum delay. The required number of iterations is typically under eight.

Using the solution from the OCWSM as the initial guess in the sizing and optimisation process, it is found that the speed of convergence can be significantly improved compared with that obtained when using the heuristic approach.

4.3 Outline of the augmented Lagrangian function and the self-scaling and restarting quasi-Newton method

The augmented Lagrangian function $L_c(x, \lambda)$ of eqn. 6 is defined as

$$L_c(x, \lambda) = f(x) + \lambda' h(x) + \tfrac{1}{2} c \|h(x)\|^2 \quad (7)$$

where c is the penalty parameter, n is the number of optimisation variables (design parameters), $\boldsymbol{x}$ is a $n \times 1$ vector (the vector of design parameters), λ is a $m \times 1$ vector (multiple vector), and λ' is the transpose of λ.

$$f(\boldsymbol{x}) = \mathscr{P}$$

$$\boldsymbol{h}(\boldsymbol{x}) = [T_{d1}/T_{s1} - 1,\ T_{d2}/T_{s2} - 1,\ \ldots,\ T_{dm}/T_{sm} - 1]'$$

and $\|\boldsymbol{h}(\boldsymbol{x})\|$ is the Euclidean norm of $\boldsymbol{h}(\boldsymbol{x})$.

A sequence of minimisations in the form

$$\begin{aligned} &\text{minimise } L_{c_i}(\boldsymbol{x}, \lambda_j) \\ &\text{subject to } \boldsymbol{x} \in \boldsymbol{X} \end{aligned} \tag{8}$$

is performed, where $\{c_i\}$ and $\{\lambda_j\}$ are the sequence of positive penalty parameters and multiplier vectors, respectively.

The modified self-scaling and starting quasi-Newton method is implemented as shown in Table 4. In Table 4, the vector $\boldsymbol{g}_j$ is the first derivative of the cost function at $\boldsymbol{x}_j$. The vectors $\boldsymbol{g}_j$, $\boldsymbol{d}_j$, $\boldsymbol{p}_j$, and $\boldsymbol{q}_j$ are $n \times 1$ vectors. The vector $\boldsymbol{S}_j$ is a $n \times n$ matrix (inverse Hessian). The value α is the optimal step size along the descent direction $\boldsymbol{d}_j$ for the minimisation of $L_{c_i}(\boldsymbol{x}, \lambda_j)$.

Table 4: Proposed algorithm

Step 1:	Initial guess $\boldsymbol{x}_0$, evaluate c_0, and $i = 0$
Step 2:	$\boldsymbol{S}_0 = \boldsymbol{I}$, calculate $\boldsymbol{g}_0$, $\lambda_0 = 0$, and $j = 0$
Step 3:	$\boldsymbol{d}_j = -\boldsymbol{S}_j \boldsymbol{g}_j$, minimise $L_{c_i}(\boldsymbol{x}_j + \alpha \boldsymbol{d}_j, \lambda_j)$, and $\boldsymbol{x}_{j+1} = \boldsymbol{x}_j + \alpha \boldsymbol{d}_j$
Step 4:	If condition A_1, go to step 6; else $\boldsymbol{p}_j = \alpha \boldsymbol{d}_j$, calculate $\boldsymbol{g}_{j+1}$, $\boldsymbol{q}_j = \boldsymbol{g}_{j+1} - \boldsymbol{g}_j$ $\boldsymbol{S}_{j+1} = \left[\boldsymbol{S}_j - \dfrac{\boldsymbol{S}_j \boldsymbol{q}_j \boldsymbol{q}_j' \boldsymbol{S}_j}{\boldsymbol{q}_j' \boldsymbol{S}_j \boldsymbol{q}_j}\right] \dfrac{\boldsymbol{p}_j \boldsymbol{q}_j'}{\boldsymbol{q}_j' \boldsymbol{S}_j \boldsymbol{q}_j} + \dfrac{\boldsymbol{p}_j \boldsymbol{p}_j'}{\boldsymbol{p}_j' \boldsymbol{q}_j}$ $\lambda_{j+1} = \lambda_j + c_i \boldsymbol{h}(\boldsymbol{x}_j)$, and add one to j
Step 5:	If j is smaller than RN, return to step 3
Step 6:	If condition A_2, go to step 8; else $\boldsymbol{x}_0 = \boldsymbol{x}_j$, $c_{i+1} = F_{inc} \times c_i$, and add one to i
Step 7:	If i is smaller than *iteration*, return to step 2
Step 8:	Obtain the solution

The device sizes of each gate are first calculated from the deviated $\boldsymbol{x}_j$ by using the timing equation to determine the derivative of the cost function. Dynamic power dissipation of the circuit is determined from the calculated device sizes. The derivative of the cost function is approximated by the first finite divided difference of the deviated cost function.

There are two iteration loops as shown in Table 4. The inner loop uses the self-scaling scheme to approximate the inverse Hessian matrix $\boldsymbol{S}$. One complete cycle of an approximation requires at least n steps to approach the result of the conjugate gradient method. Round-off errors, numerical errors, and inaccurate line searches mean that the resultant inverse Hessian matrix may deviate from the actual inverse Hessian matrix and degrade the quadratic convergence rate. A smaller step number RN is assigned to the inner loop. The outer loop is used to construct the restarting scheme of the self-scaling quasi-Newton method. The maximum number of restarting cycles is *iteration* specified by the user.

There are two check points (A_1 and A_2) in the optimisation process. A_1 is used to check whether $L_{c_i}(\boldsymbol{x}_{j+1}, \lambda_j)$ is 0.99 times larger than $L_{c_i}(\boldsymbol{x}_j, \lambda_j)$. If that condition is satisfied, the optimisation process restarts. A_2 is used to check whether $L_{c_i}(\boldsymbol{x}_j, \lambda_j)$ is 0.99 times larger than $L_{c_i}(\boldsymbol{x}_0, \lambda_j)$. If that condition is satisfied, the optimisation process ends and the required timing specifications for each gate is obtained. The device sizes of all the circuit can then be determined by using the timing equations from all the circuit output to all the circuit inputs.

From eqn. 7, it is found that the scale of $f(\boldsymbol{x})$ and $\|\boldsymbol{h}(\boldsymbol{x})\|$ is not compatible. The penalty parameter c_0 is first normalised to balance the scale of $f(\boldsymbol{x})$ and $\|\boldsymbol{h}(\boldsymbol{x})\|$. The subsequent values of c_j are monotonically increased using the equation $c_{j+1} = F_{inc}\, c_j$. The value of F_{inc} is typically larger than 4 and smaller than 10 [21].

The multiplier vector λ_0 is initially chosen to be $\boldsymbol{0}$. The subsequent vectors of λ_j are modified using the equation $\lambda_{j+1} = \lambda_j + c_i \times \boldsymbol{h}(\boldsymbol{x}_j)$. Other good modified equations of the multiplier vectors are described in the nonlinear optimisation text [21–23]. The objective of this paper is only to verify the efficiency of the proposed techniques, so the other equations are not considered.

5 Experimental results

Using Turbo-C on a PC-AT, the above sizing and optimisation techniques have been implemented in an experimental program called the TISA [19, 20] and applied to size many circuits. The memory required for program and dynamic data is 200 Kbyte and 64 Kbyte/100 gate. The required memory increases quadratically as a function of the gate number because the optimisation method contains a two-dimensional matrix $\boldsymbol{S}(\boldsymbol{x})$. The maximum number of gates allowed in the optimisation is 128 under the PC-AT 640 Kbyte real-mode limitation. The conjugate gradient method [23] uses a one-dimensional vector in the optimisation process. Implementing TISA using the conjugate gradient method in PC-AT protection-mode operation (with 16 Mbyte), the maximum number of gates would be expected to be more than 10 000.

To demonstrate the efficiency of the proposed techniques in the sizing and optimisation, the conventional optimisation algorithms [10, 13], were also implemented and applied to size the same circuits. In one of the conventional algorithms, the device sizes are used as the variables of the optimisation process and the minimum device sizes are used as the initial guess of the optimisation process [10]. It is then called the minimum-size algorithm for simplicity. In the other algorithm, the device sizes are used as variables of the optimisation process and the heuristic approach is used to obtain the initial guess of the optimisation process [13]. It is called the heuristic algorithm for simplicity. The timing models developed [15–18] are also used in both algorithms for a fair comparison. The increment constant *bumpsize* [13] used in the heuristic approach is 1.1.

Different input excitation nodes lead to different output timing and power consumption. The input excitation node of each gate is considered to be the node furthest away from the output node to simplify the computation complexity and the computer time in sizing. This can lead to a safe design so that the actual chip delay is always equal to or smaller than that designed. Different input excitation nodes are considered in timing verification.

To verify the efficiency of the proposed techniques, the values of F_{inc}, which is the factor associated with the penalty parameters ($c_{j+1} = F_{inc}\, c_j$), must be comprehensively considered. Both the developed and the conventional sizing and optimisation algorithms were applied to size a four-bit even parity checker as shown in Fig. 5. The input voltage was an exponential waveform with a rise/fall time of 0.44 ns. Using device sizes as optimisation minimisation variables and minimum device sizes as the initial guess, the minimum achievable fixed-delay specifi-

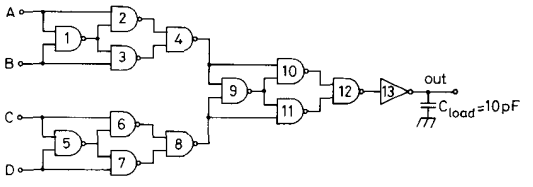

Fig. 5 *Four bit even parity checker*

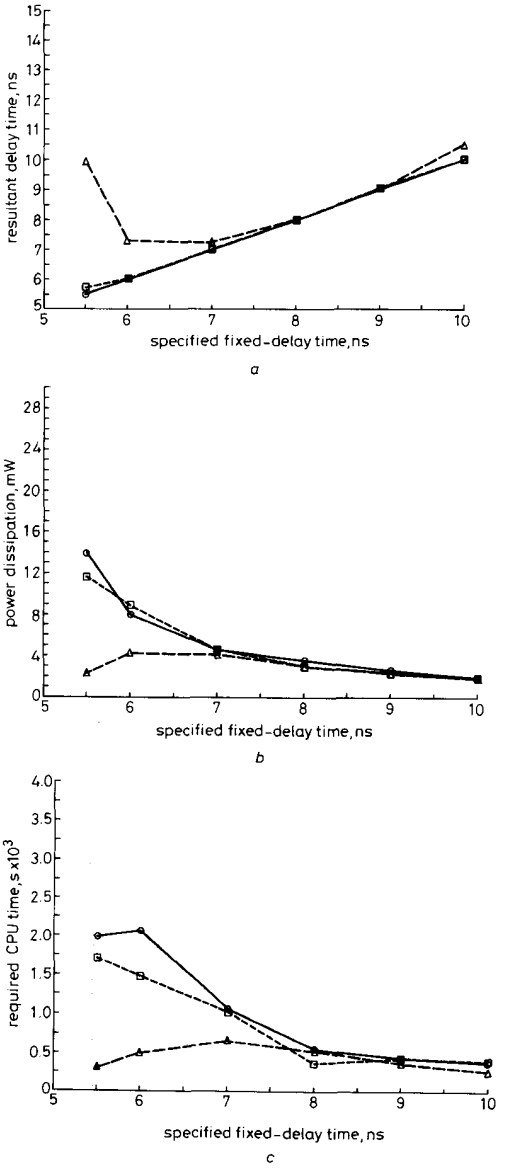

Fig. 6 *Comparison of algorithms*
Four bit even parity checker
F_{inc} = 4.0; Input rise/fall time = 0.44 ns
—○— transition times/OCWSM
--□-- device sizes/heuristic approach
--△-- device sizes/minimum device sizes
a Resultant time delay
b Resultant power dissipation
c CPU time consumption

cation of the optimisation with fixed-delay specification is 7.23 ns. For the optimisations with 5.5, 6, or 7 ns fixed-delay specifications, this algorithm can not respond as can be seen from Fig. 6*a*. Using device sizes as optimisation variables and the heuristic approach to obtain the initial guess, the minimum fixed-delay specification achievable is 6 ns. The fixed-delay specification of the optimisation using transition times as optimisation variables and the OCWSM to obtain the initial guess can be as small as 5.5 ns. These proposed techniques are called the proposed algorithm.

It is also found that the optimisation with 10 ns fixed-delay specification performed by using the minimum-size algorithm has the local minimum problem. The error between the resultant and the specified delay times is greater than 1%. This phenomenon is not seen in the other two algorithms (the proposed algorithm and the heuristic algorithm).

From Fig. 6*b*, it is found that the resultant power dissipation of the circuit optimised with 8 ns and 9 ns fixed-delay specifications and by using the proposed algorithm are greater than that obtained by using the heuristic algorithm. As the fixed-delay specification decreases to 6 ns and 7 ns, the power dissipation of the circuit optimised by using the proposed algorithm becomes smaller than the heuristic algorithm. This means that in high-speed design, the proposed algorithm can perform a more satisfactory optimisation with less resultant power dissipation.

In Fig. 6*c*, it is found that the required CPU time for the optimisations with 7, 8 and 9 ns fixed-delay specifications performed by using the three algorithms are very close. Although the required CPU time for the optimisation with 6 ns fixed-delay specification performed by using the proposed algorithm is 30% greater than that by using the heuristic algorithm, the power dissipation of the circuit optimised by using the proposed algorithm is smaller. The trade-off between CPU time and circuit performance is thus satisfactory in the proposed algorithm.

To further verify the efficiency of the optimisation process by using the proposed algorithms for the complex circuit, a benchmark circuit RD53 [37] shown in Fig. 7 was optimised. It contains CMOS standard static logic gates and AOI gates. The given input rise/fall time is 0.44 ns. The fixed-delay specifications at nodes F0, F1, and F2 are considered to be the same. Fig. 8 shows the comparisons between the optimisation results of the proposed algorithm and the heuristic algorithm. The optimisation using the minimum-size algorithm can not satisfy the specified delay time and so it is not shown in Fig. 8. It is found from Fig. 8*a* that the heuristic algorithm and the proposed algorithm can satisfy the delay specification. As seen from Fig. 8*b*, the power dissipations obtained by using the proposed algorithm are smaller than those obtained by using the heuristic algorithm. This is because the $h(x)$ in the cost function is nearly proportional to the optimisation variable. The relation between the cost function and the optimisation variables of the proposed algorithm is more linear than that of the heuristic algorithm. The proposed algorithm can thus avoid the incorrect optimisation convergence caused by the numerical errors and inaccurate line searches from the first finite divided difference of the deviated cost function. These characteristics also make the CPU time of the proposed algorithm smaller than that of the heuristic algorithm (Fig. 8*c*).

To verify the accuracy of the adopted timing models, a one-bit full adder is sized by using the proposed algo-

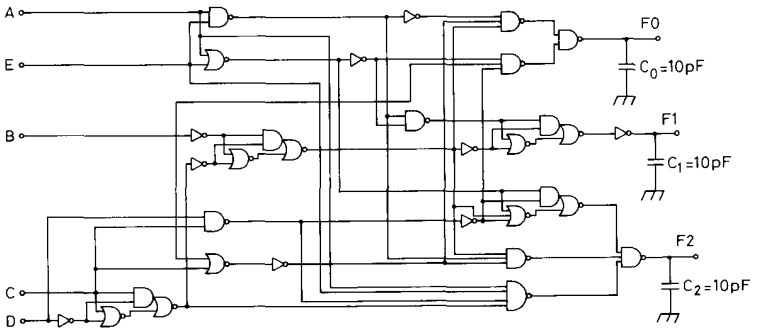

Fig. 7 *Benchmark circuit RD53*

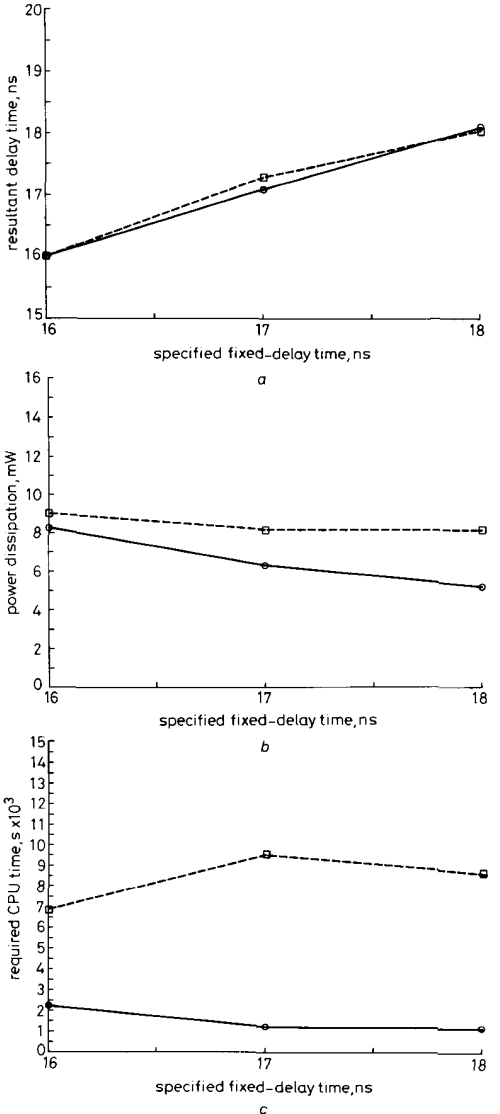

Fig. 8 *Comparison of algorithms*

Benchmark circuit RD53
F_{inc} = 4.0; Input rise/fall time = 0.44 ns
—○— transition times/OCWSM
--□-- device sizes/heuristic approach

a Resultant delay time
b Resultant power dissipation
c CPU time consumption

rithm with different fixed-delay specifications. The SPICE transient simulations of the circuit with the obtained device sizes and the given input excitation is made to obtain the delay time. Table 5 lists the comparisons between the SPICE simulations and the model calculations, assuming that the input modes ABC changes from 000 to 111. The maximum error is 13.29% and 4.43% for the outputs *sum* and *carry*, respectively. The required CPU time of the model calculation is about two orders of magnitude smaller than that of the SPICE simulation. From Table 5, it is also found that because of the consideration of worst-case timing in the sizing and optimisation, the resultant delay times of *sum* and *carry* are smaller than the fixed-delay specifications. This guarantees a safe design.

Table 5: Delay times of one-bit full adder

Specified delay time, ns			SPICE	Model	Error %
6	Delay time, ns	Sum	3.161	2.733	13.29
		Carry	5.015	4.782	4.43
	CPU time (PC/AT), s		500	2	
7	Delay time, ns	sum	4.186	3.720	10.94
		carry	5.598	5.362	4.06
	CPU time (PC/AT), s		399	2	
8	Delay time, ns	sum	4.722	4.306	8.63
		carry	6.228	5.986	3.75
	CPU time (PC/AT), s		372	2	
9	Delay time, ns	sum	6.191	6.112	1.13
		carry	7.206	7.142	0.76
	CPU time (PC/AT), s		384	2	
10	Delay time, ns	sum	6.062	5.937	1.91
		carry	7.874	7.667	2.50
	CPU time (PC/AT), s		387	2	

6 Conclusion and discussion

Two techniques were proposed to enhance the efficiency of the optimisation process. The techniques use the transition times as variables of the optimisation process and the OCWSM to obtain the initial guess of the optimisation process. The proposed techniques can perform the sizing and optimisation of CMOS combinational logic circuits with a smaller delay specification than other sizing and optimisation algorithms. This enhances the speed performance of the sized circuits. The circuit sized by the proposed algorithm has a smaller power dissipation especially when the delay specification is small. The CPU time is of the same order of magnitude as that in other algorithms. It is therefore suitable to use the proposed techniques in the sizing and optimisation of high-performance circuits.

The power dissipation of CMOS logic gates consists of two components: dynamic power dissipation and short-

circuit power dissipation. The power dissipation considered earlier is the dynamic power dissipation. To accurately calculate the power dissipation of a CMOS logic circuit, an accurate model of the short-circuit power dissipation for CMOS logic gates must be constructed. This is the intention of a future study.

There are many design considerations for a CMOS logic gate such as equal rise and fall times, equal channel widths of PMOSFET and NMOSFET, equal high and low noise margin, optimal β ratio of each gate, etc. The design consideration with symmetrical rise and fall times is adopted as used in the above optimisation. All the design considerations can be arranged to use the transition time as the optimisation variables so it is expected that by using the proposed techniques, the above mentioned advantages can be also obtained.

The proposed techniques can be applied to other sizing and constrained optimisation problems. Further generalisation of the proposed algorithm in solving various problems will be performed in the future.

7 References

1 NYE, W., POLAK, E., SANGIOVANNI-VINCENTELLI, A., and TITS, A.: 'DELIGHT: An optimization-based computer-aided design system'. Proc. Int. Symp. Circuits and Systems., 1981, pp. 851–855

2 BRAYTON, R., HACHTEL, G., and SANGIOVANNI-VINCENTELLI, A.: 'A survey of optimization techniques for integrated-circuit design', *Proc. IEEE*, 1981, **69,** pp. 1334–1362

3 KANUMA, A.: 'CMOS circuit optimization', *Solid-State Electron.*, 1983, **26,** pp. 47–58

4 TRIMBERGER, S.: 'Automated performance optimization of custom integrated circuits'. Proc. Int. Symp. Circuit and Systems, 1983, pp. 194–197

5 LEWIS, E.: 'Optimization of device area and overall delay for CMOS VLSI designs', *Proc. IEEE*, 1984, **72,** pp. 670–689

6 GLASSER, L.A., and HOYTE, L.P.J.: 'Delay and Power optimization in VLSI circuits'. Proc. 21st Design Automation Conf., 1984, pp. 781–785

7 VEENDRICK, H.J.M.: 'Short-circuit dissipation of static CMOS circuitry and its impact on the design of buffer circuits', *IEEE Trans.*, 1984, **SC-19,** pp. 468–473

8 LEE, C.M., and SOUKUP, H.: 'An algorithm for CMOS timing and area optimization', *IEEE Trans.*, 1984, **SC-19,** pp. 781–787

9 KAO, W.H., FATHI, N., and LEE, C.H.: 'Algorithms for automatic transistor sizing in CMOS digital circuits'. Proc. 22nd Design Automation Conf., 1985, pp. 781–785

10 MATSON, M.D., and GLASSER, L.A.: 'Macromodeling and optimization of digit MOS VLSI circuits', *IEEE Trans.*, 1986, **CAD-5,** pp. 659–678

11 HEDENSTIERNA, N., and JEPPSON, K.O.: 'CMOS circuit speed and buffer optimization', *IEEE Trans.*, 1967, **CAD-6,** pp. 270–281

12 CIRIT, M.A.: 'Transistor sizing in CMOS circuits'. Proc. 24th Design Automation Conf., June 1987, pp. 121–124

13 SHYU, J.M., SANGIOVANNI-VINCENTELLI, A.J., FISHBURN, P., and DUNLOP, A.E.: 'Optimization-based transistor sizing', *IEEE Trans.*, 1988, **SC-23,** pp. 400–409

14 RICHMAN, B.A., HANSEN, J.E., and CAMERON, K.: 'A deterministic algorithm for automatic CMOS transistor sizing', *IEEE Trans.*, 1988, **SC-23,** pp. 522–526

15 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

16 WU, C.Y., LI, C., and HWANG, J.S.: 'Timing macromodels for CMOS static set/reset latches and their applications', *IEEE Proc. E*, 1988, **135,** pp. 151–160

17 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.

18 WU, C.Y., and SHIAO, M.C.: 'General and efficient timing models for CMOS AND-OR-INVERTERS and OR-AND-INVERTERS gates', *IEEE Trans.*, 1991, **CAD-10**

19 WU, C.Y., and HWANG, J.S.: 'A new autosizing algorithm for CMOS combinational logic circuits'. Int. Symp. VLSI Technology, Systems and Applications, 1989, pp. 242–246

20 WU, C.Y., and HWANG, J.S.: 'A new fast sizing algorithm for high-performance CMOS combinational logic circuits'. 3rd Int. Symp. IC Design and Manufacture, 1989, pp. 165–172

21 BERTSEKAS, D.: 'Constrained optimization and lagrange multiplier methods' (Academic Press, 1982)

22 GILL, P.E., MURRAY, W., and WRIGHT, M.H.: 'Practical optimization' (London, 1981)

23 LUENBERGER, D.: 'Introduction to linear and nonlinear programming' (Addison-Wesley, 1984)

24 SHJOI, M.: 'CMOS digital circuit technology' (Prentice-Hall, Englewood Cliffs, 1988)

25 KANG, S.M.: 'A design of CMOS polycells for LSI circuits', *IEEE Trans.*, 1981, **CAS-28,** pp. 838–843

26 TOKUDA, T., OKAZAKI, K., SAKASHITA, K., OHKURA, I., and ENOMOTO, T.: 'Delay-time modeling for ED MOS logic LSI', *IEEE Trans.*, 1983, **CAD-2,** pp. 129–134

27 ETIEMBLE, D., ADELINE, V., DUYET, N.H., and BALLEGEER, J.C.: 'Micro-computer oriented algorithms for delay evaluation of MOS gates'. Proc. 21st Design Automation Conf., 1984, pp. 358–364

28 BAYRUNS, R.J., JOHNSTON, R.L., FRASER JR., D.L., and FANG, S.-C.: 'Delay analysis of Si NMOS Gbit/s logic circuit', *IEEE Trans.*, 1984, **SC-19,** pp. 755–764

29 SIMONNS, J.G., and TAYLOR, G.W.: 'An analytical treatment of the performance of submicrometer FET logic', *IEEE Trans.*, 1985, **SC-20,** pp. 1242–1251

30 AUVERGNE, D., CAMBON, G., DESCHACHT, D., ROBERT, M., SAGNES, G., and TEMPIER, V.: 'Delay-time evaluation in ED MOS logic LSI', *IEEE Trans.*, 1986, **SC-21,** pp. 337–343

31 DESCHACHT, D., ROBERT, M., and AUVERGNE, D.: 'Explicit formulation of delays in CMOS data paths', *IEEE Trans.*, 1988, **SC-23,** pp. 1257–1264

32 BROCCO, L.M., MCCORMICK, S.P., and ALLEN, J.: 'Macromodeling CMOS Circuits for timing simulation', *IEEE Trans.*, 1988, **CAD-7,** pp. 1237–1249

33 JOUPPI, N.: 'Timing analysis for *n*MOS VLSI'. Proc. 20th Design Automation Conf., 1983, pp. 411–418

34 PLONUS, M.A.: 'Applied electromagnetics' (McGraw-Hill, New York, 1978)

35 DESOER, C.A., and KUH, E.S.: 'Basic circuit theory' (McGraw-Hill, New York, 1978)

36 VLADIMIRESCU, A., and LIU, S.: 'The simulation of MOS Integrated Circuit Using SPICE2'. Memorandum M80/7, 1980

37 MITCHELI, M.D., SANGIOVANNI-VINCENTELLI, A., and ANTOGNETTI, P.: 'Design system for VLSI circuits logic synthesis and silicon compilation' (Netherlands, 1981)

8 Appendix

The change of the energy stored on the drain-bulk junction capacitor is derived. The energy loss of a passive element from time t_0 to t_f is [35]

$$\mathscr{E}(t_0, t_f) \equiv \int_{t_0}^{t_f} V(t')i(t')\,\partial t' \tag{9}$$

Since $i(t')\,\partial t' = \partial q$, the above equation can be rewritten as

$$\mathscr{E}(Q_0, Q_f) \equiv \int_{Q_0}^{Q_f} V(Q')\,\partial Q' \tag{10}$$

The drain-bulk junction capacitance [36] is a voltage-dependent capacitor and can be expressed as

$$C_j = CJ\frac{A_D}{\left(1+\frac{V_{DB}}{PB}\right)^{MJ}} + CJSW\frac{P_D}{\left(1+\frac{V_{DB}}{PB}\right)^{MJSW}} = C_{area} + C_{peri} \tag{11}$$

where CJ is the zero-bias bulk capacitance per square metre, MJ is the bulk-junction grading coefficient, A_D is the drain area, P_D is the drain perimeter, V_{DB} is the voltage across the drain-bulk junction, $CJSW$ is the zero-bias perimeter capacitance per metre, $MJSW$ is the perimeter grading coefficient, C_{area} is the bulk capacitance and C_{peri} is the perimeter capacitance.

The energy loss of a passive element from time t_0 to time t_f (from $V_{DB} = V_0 = 0$ to $V_{DB} = V_f = V_{bias}$) can be

rewritten as

$$\mathscr{E}(Q_0, Q_f) \equiv \int_{Q_0}^{Q_f} V(Q')\,\partial Q'$$
$$= [V(Q')Q']\Big|_{Q_0}^{Q_f} - \int_{V_0}^{V_f} Q(V')\,\partial V'$$
$$= [(C_j V)V]\Big|_{V=V_0}^{V=V_f} - \int_{V_0}^{V_f} (C_j V)\,\partial V' \qquad (12)$$

The energy loss caused by the area capacitance $\mathscr{E}_{area}$ is

$$\mathscr{E}_{area} = \left[CJ \frac{A_D}{\left(1 + \frac{V_{DB}}{PB}\right)^{MJ}} V^2 \right]_{V_0=0}^{V=V_{bias}}$$
$$- \int_0^{V_{bias}} CJ \frac{A_D}{\left(1 + \frac{V_{DB}}{PB}\right)^{MJ}} V\,\partial V$$
$$= CJA_D \left\{ \frac{V_{bias}^2}{\left(1 + \frac{V_{bias}}{PB}\right)^{MJ}} - PB^2 \frac{\left(1 + \frac{V_{bias}}{PB}\right)^{2-MJ} - 1}{2 - MJ} + PB^2 \frac{\left(1 + \frac{V_{bias}}{PB}\right)^{1-MJ} - 1}{1 - MJ} \right\} \qquad (13)$$

Similarly, the energy loss caused by the perimeter capacitance $\mathscr{E}_{peri}$ is

$$\mathscr{E}_{peri} = CJSWP_D \left\{ \frac{V_{bias}^2}{\left(1 + \frac{V_{bias}}{PB}\right)^{MJSW}} - PB^2 \frac{\left(1 + \frac{V_{bias}}{PB}\right)^{2-MJSW} - 1}{2 - MJSW} + PB^2 \frac{\left(1 + \frac{V_{bias}}{PB}\right)^{1-MJSW} - 1}{1 - MJSW} \right\} \qquad (14)$$

Thus, the energy loss of the drain-bulk junction capacitance is expressed as

$$\mathscr{E}_{drain} = CJ\ A_D F_{area}(V_{bias}) + CJSW\ P_D F_{peri}(V_{bias}) \qquad (15)$$

where

$$F_{area}(V_{bias}) = \left\{ \frac{V_{bias}^2}{\left(1 + \frac{V_{bias}}{PD}\right)^{MJ}} - PB^2 \frac{\left(1 + \frac{V_{bias}}{PB}\right)^{2-MJ} - 1}{2 - MJ} + PB^2 \frac{\left(1 + \frac{V_{bias}}{PB}\right)^{1-MJ} - 1}{1 - MJ} \right\}$$

$$F_{peri}(V_{bias}) = \left\{ \frac{V_{bias}^2}{\left(1 + \frac{V_{bias}}{PB}\right)^{MJSW}} - PB^2 \frac{\left(1 + \frac{V_{bias}}{PB}\right)^{2-MJSW} - 1}{2 - MJSW} + PB^2 \frac{\left(1 + \frac{V_{bias}}{PB}\right)^{1-MJSW} - 1}{1 - MJSW} \right\}$$