ELECTRONICS AND ELECTRICAL ENGINEERING
ISSN 1392 – 1215 2007. No. 3(75)
ELEKTRONIKA IR ELEKTROTECHNIKA

T 171 MICROELECTRONICS
MIKROELEKTRONIKA

Synthesizing CMOS Circuits for Speed-Power Product Minimization

P. Vijayakumar, K. Gunavathi
Department o f Electrical and Electronics Engineering, PSG College of Technology, Coimbatore-641004, India, tel.: -0091-9894994643, e-mail: vijay_p6@hotmail.com

Introduction

In the modern world portable systems are becoming increasingly popular. Hence the need for devices that dissipate lower power arises. Power dissipation in CMOS circuits can be classified into two types, namely static and dynamic power dissipation. Of these, dynamic power dissipation is the chief contributor [1,2]. Hence efforts at power optimization will be aimed at reducing dynamic power dissipation. However, power optimization quite often results in an unavoidable reduction in the speed of the devices. As reduction in speed is not affordable, ways to optimize the speed of the devices will also have to be looked upon. In short, Speed-Power optimization is the need of the hour, which is precisely the aim of this paper.

Power optimization techniques currently in use are such as voltage scaling, gate resizing, precomputation, etc. All these techniques optimize power at the expense of the speed of the circuit [3,4,5]. Stepwise charging results only in a small decrease in the speed of the circuit and that too can be compensated by pipelining the circuit [6,7,8].

Pipelining is a technique of decomposing a circuit into a number of segments, with each segment operating concurrently with all other segments thus increasing the speed of the circuit. Intermittent latches separate the segments from one another. Retiming algorithm is used to place these latches judiciously [9,10]. In stepwise charging the single supply voltage V of a conventional driver is replaced with a bank of N supplies with evenly distributed voltages $V_i = (i/N)V$. In most practical cases, N lies between 2 and 8 [8].

Using a combination of Retiming algorithm & Stepwise charging an efficient algorithm for minimization of power dissipation as well as delay has been evolved. The above principles have been implemented and validated by applying it to a number of benchmark circuits. Experimental results have shown a reduction of around 80% in power-delay product.

Principles of retiming

Retiming is an algorithm to pipeline the given circuit into number of stages so that the overall glitching activity in the circuit is reduced.

Pipelining is a technique by which a given circuit is divided into a number of segments separated by means of latches, so that each of these segments functions independently. Thus the throughput and hence the speed of the circuit is increased considerably. It is characteristic of pipelines that several computations can be in progress in different segments of the circuit at the same time. The concurrency of operation is made possible by associating a latch with each segment in the pipeline. These latches provide isolation between segments in the pipeline.

The simplest way of visualizing a pipeline structure is to consider that each segment consists of an input latch followed by a combinational circuit. The input latch holds the input data and the combinational circuit performs the sub-operation in that particular segment. The output of that particular segment is given as input to the input latch of the next segment. A clock is applied to all latches after enough time has elapsed to perform sub-operations in all the segments. In this way information flows through the pipeline one step at a time. As shown in Fig. 1, the given circuit is divided into a number of combinational logic blocks (C/L) with latches separating them. Figure1 shows a pipelined circuit of N stages.

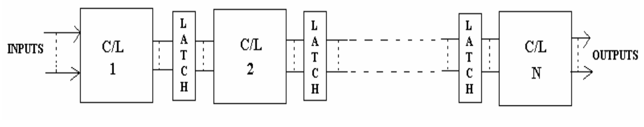

Fig. 1. Pipelined Circuit

Glitches are known to contribute more than 60% of the total power dissipation in CMOS circuits, even with glitch free inputs [10,11]. As a result, glitch reduction is of prime importance in low power design. There are various techniques for reducing glitches, retiming being one among them [12,13,14]. Retiming involves repositioning of latches in a pipelined circuit so that the glitches are eliminated as far as possible. In other words retiming helps in placing the intermittent latches of the pipelined circuit judiciously such that maximum glitch reduction is achieved.

Consider a gate G as shown in Fig. 2a. Let the average switching activity at the output of the gate G be N_G and the load capacitance be C_{load}. The power dissipated at the output of the gate is proportional to N_G C_{load}. Now, consider the scenario of placing a latch F at the output of gate G, as shown in Fig. 2b. The capacitance seen at the input of the latch F is C_F. Let the average switching activity at the output of the latch F be N_F. Now, the power dissipated at the output of this gate is proportional to N_G C_F + N_F C_{load}.

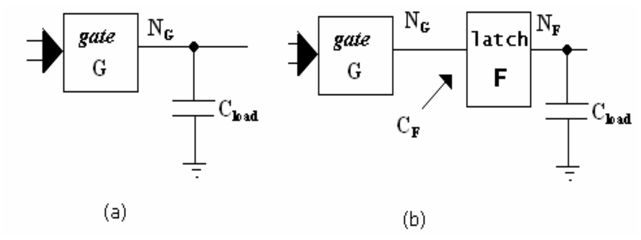

Fig. 2. Adding a flip flop to a gate

It is possible for P_2 to be less than P_1 if $N_F << N_G$ and $C_F << C_{load}$. The first condition ($N_F << N_G$) is obvious as a latch can make only one transition in a clock cycle. To satisfy the second condition, the nodes having larger number of switched capacitance must be chosen. Hence, if such nodes, ones with greater probability of glitches present are selected and place latches it is possible to save power by reducing the glitch power dissipation.

Algorithm

Let us now see a working algorithm to assign weight or cost to each node in the circuit depending upon the glitching, fanin, fanout and probability of transition at a gate propagating through its transitive fanout. We then trace paths from each primary input to each primary output and place one latch in every path at the output of the node with maximum weight or cost.

1) Find power, capacitance, switching probability of each node both for the unit delay model (P_1,C_1,S_1) and for the zero delay model (P_0,C_0,S_0).

2) We know that power dissipated at any node is given by

$$P = {}^1/_2\, C.V^2 f_{clock}\, N, \qquad (1)$$

where

C = load capacitance,
V = supply voltage,
f_{clock} = clock frequency,
N = switching activity.

From this we find the switching activity for unit delay model as well as zero delay model as given by

$$N_1=P_1/({}^1/_2V^2f_{clock}\,C_1), \qquad (2)$$
$$N_0= P_0/({}^1/_2V^2f_{clock}\,C_0). \qquad (3)$$

3) The amount of glitching is the difference in switching activity of the circuit for unit delay model and for zero delay model. Hence

$$N_{glitch}=N_1-N_0 \qquad (4)$$

4) Let fanin, fanout of the node x be fi(x) and fo(x) and the weight function at the node W(x) is given by

$$W(x)=(fi(x)+fo(x))\cdot(N_{glitch})\cdot\left[C_1(x)+\sum_{y\in fo(x)}(C_1(y)\cdot S_1(y))\right] \qquad (5)$$

here $fi(x)$ – fanin of x; $fo(x)$ – fanout of x; $C_1(x)$ – capacitance of x; $C_1(y)$ – capacitance of y; $S_1(y)$ – sensitivity of y to glitch at x.

5) Trace all paths from every primary input to every primary output.

6) If weights of all nodes are zero then place no latches and go to Step 10; else go to Step 7

7) Place a latch at the output of the node having maximum weight among all nodes in the circuit.

8) Make the weight of all nodes that fanin to the chosen node or fanout from the chosen node as zero. This is to make sure that each path gets only one latch and not more than one.

9) Check whether all paths have one latch. If not go to step 6, else go to Step 10.

10). Stop the process.

This algorithm divides a given circuit into two segments A and B. Now segment A can further be subdivided by applying the above algorithm. This will give a pipelined circuit of three stages. This can be repeated until the algorithm says that there is no need to place latches, which means that the segment A cannot be subdivided any further. Thus repeated application of this algorithm gives multistage pipelined circuit.

Stepwise charging

Stepwise charging is a power optimization technique, wherein power saving is obtained by charging the load with supply voltage which increases in a series of steps from 0 to V_{dd}. To make a complete understanding of the concept of stepwise charging, let us compare it with a conventional CMOS driver. For conventional driver, the load is charged to the supply voltage V by connecting it to the power rail through a switch with a certain on-resistance. During charging, the voltage drop across the switch varies from V to 0. Hence the average drop is given by

$$V_{avg} = (V-0)/2 = V/2 \qquad (6)$$

The energy dissipated during charging is given by

$$E_{con} = Q \times V_{avg}. \quad (7)$$

Substituting $Q = C \times V$

$$E_{con} = CV \times V/2 = ½\, CV^2. \quad (8)$$

Note that the same amount of energy will be dissipated during discharge phase also.

Next, consider the stepwise driver. The single supply voltage V is now replaced by a bank of N supplies each with voltage V/N. These supplies are connected in series and switches S_1 to S_N are used to connect these supplies to the power rail as shown in Fig. 3. To charge the load, switches $S_1$to S_N are turned ON and OFF in succession i.e. (S_1 ON), (S_2 ON & S_1 OFF), (S_3 ON & S_2 OFF), and so on up to (S_N ON & S_{N-1} OFF) and finally (S_1 ON & S_N OFF). Note that when switch S_j is turned ON and all other switches are OFF, voltages V_1 to V_j are connected in series to the power rail, making the voltage available at the power rail to be $j \times V/N$. Thus the load is charged in N steps

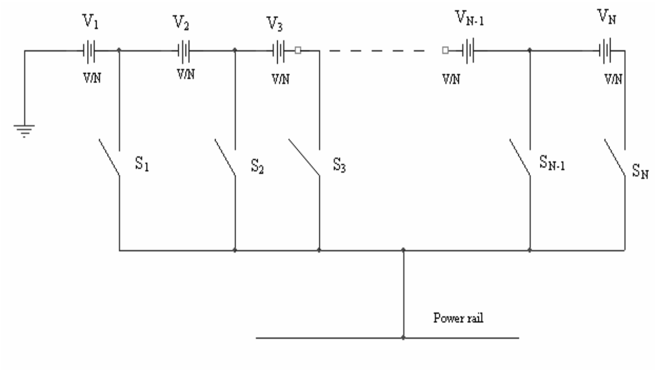

Fig. 3. Stepwise Charging Model

The energy dissipation per step is given by

$$E_{per\ step} = Q \times V_{mean}, \quad (9)$$

here

$$V_{mean} = ((V/N)-0)/2 = V/(2). \quad (10)$$

We know that

$$Q = C \times (V/N), \quad (11)$$

$$E_{per\ step} = Q \times V_{mean} = (CV/N) \times V/(2N) = ½\,(C \times V^2/N^2). \quad (12)$$

There are N steps .So the dissipation for the whole transition is given by

$$E_{stepwise} = N \times E_{perstep}$$

$$= N \times ½\,(CV^2/N^2) = ½\,(CV^2/N). \quad (13)$$

Hence, the gain factor for stepwise charging over the conventional method is given by

$$\text{Gain} = (E_{con}/E_{stepwise}) = (½\, CV^2)/(½\, CV^2/N) = N. \quad (14)$$

It appears from the above discussion that the number of steps N should be made as large as possible to get the lowest dissipation. However, N cannot usefully be made arbitrarily large because each step requires that a switch be turned ON and OFF, which itself causes dissipation. The optimal number of steps for any given circuit depends on the intrinsic speed of switching devices but is in independent of the load capacitance. In most practical cases it lies between 2 and 8. The optimal number of steps of the stepwise driver is different for different circuits and has to be found out on a case-by-case basis. The stepwise driver produces a voltage waveform as shown in Fig. 4 to drive the circuit.

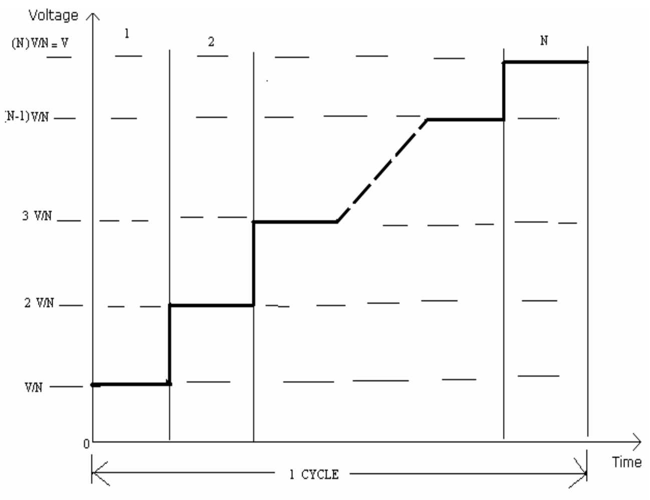

Fig. 4. Output of Stepwise Driver

Simulation results

The given circuit is converted into a pipelined circuit of varying stages using the retiming algorithm, softwares SIS and C being used for synthesis. Stepwise charging technique is now applied to the pipelined circuits, simulation being done in Tspice. Having found the power dissipation and the circuit delay in the above manner, the power-delay product and subsequently the gain are computed. Implementing it on a number of benchmark circuits has validated the above method.

Conclusion

In this paper, a novel method to reduce the power-delay product using a combination of Retiming algorithm and Stepwise charging principle has been proposed. Thus it has been made possible to achieve a lesser power-delay product.

Experimental results have shown a reduction of about 80% in power-delay product. This method has a small disadvantage that the area increases with the addition of latches and overhead for the special driver. Thus an effective method for Speed-Power optimization has been devised.

References

1. **Chandrakasan A., Sheng S., and Brodersen R. W.** Low-power CMOS design // IEEE Journal of Solid-State Circuits, April 1992. – P. 472–484.
2. **Weste H. E.** Principles of CMOS VLSI Design, Pearson Education Pvt. Ltd., Singapore, 2002.
3. Web resource: http://www.ece.umd.edu/courses/papers.

4. **Burah M., Owens R. M., and Irwin M. J.** Transistor Sizing for low power CMOS circuits // IEEE Transaction on Computer Aided Design, June 1996. – P. 665–677.
5. Web resource: http://csdl.computer.org/comp/proceedings/ /arvlsi/1995.
6. **Chandra Kasan A. P., and W. Brodersen.** Low Power Digital CMOS Design. – Norwell: Kluwer, 1995.
7. **Giriad P., Landraut C., Pravossoudovitch S., and Severat D.** A Gate Resizing Technique for High Reduction in Power Consuming // In Proceedings of International Symposium on Low Power design, August 1987. – P. 281–286.
8. Web resource: http://www.isi.edu/acmos/stepwise.
9. **Monteiro J., Devadas S., and Ghosh A.** Retiming Sequential Circuits for Low Power // In Proceedings of the Int'l Conference on Computer Aided Design, November 1993. – P. 398–402.
10. **Malik S., Sentovich E., Brayton R. K., and Sangiovanni-Vincentelli A.** Retiming and Resynthesis: Optimization of Sequential Networks with Combinational Techniques // In Proceedings of the Hawaii International Conference on System Sciences, January 1990. – P. 397–406.
11. **Shen A., Ghosh A., Devadas S., and Keutzer K.** On average power dissipation and random pattern testability // In Proc. 32nd Design Automation Conf., June 1995. – P. 402–407.
12. **Favalli M., and Benini L.** Analysis of glitch power dissipation in CMOS ICs // In Proc. 1995 Inter. Symp. on Low Power Design, Apr. 1995. – P. 123–128.
13. **Raghunathan A., Dey S., and Jha N. K.** Register-transfer level estimation techniques for switching activity and power consumption // In Proc. of IEEE Inter. Conf. on Computer Aided Design, Nov 1996.
14. **Raghunathan A., Dey S., and Jha N. K.** Register transfer level power optimization with emphasis on glitch analysis and reduction // In IEEE Trans. on CAD, Aug. 1999. – P. 1114–1131.

Submitted for publication 2006 11 02

P. Vijayakumar, K. Gunavathi. Synthesizing CMOS Circuits for Speed-Power Product Minimization // Electronics and Electrical Engineering. – Kaunas: Technologija, 2007. – No. 3(75). – P. 27–30.

We propose an efficient algorithm to optimize Power-Delay product in CMOS logic circuits. The proposed algorithm aims at reducing both power dissipation as well as the delay, which is achieved by retiming and stepwise charging techniques. The retiming technique divides the circuit into stages with latches interposed between them, which decreases the delay. After retiming, stepwise charging technique is employed, where in the supply voltage itself is applied in a series of steps before reaching the maximum supply voltage, which reduces the power dissipation. The algorithm is tested on ISCAS benchmark circuits. Experimental results have shown a reduction of around 80% in power-delay product with small area overhead. Ill. 4, bibl. 14 (in English; summaries in English, Russian and Lithuanian).

П. Вияйакумар, К. Гунаватхи. Синтезирование CMOS цепей для уменьшения параметров задержки и мощности // Электроника и электротехника. – Каунас: Технология, 2007. – № 2(75). – С. 27–30.

Анализируется алгоритм эффективного оптимизирования параметров мощности и запаздывания в CMOS логических цепях. Данный алгоритм предложен для уменьшения как распределяемой мощности, так и запаздывания. Этого можно достигнуть применив методы ступенчатой зарядки и запаздывания. Применяя временную настройку цепь разложена по отдельным ступеням, запаздывание которых может быть управляемо индивидуально. Дальше применяется метод ступенчатой загрузки, когда напряжение питания изменяется по ступеням, пока достигается максимальное значение. Таким образом, уменьшается распределительная мощность. Алгоритм тестирован применяя ISCAS цепи тестирования. Экспериментальные результаты показали, что единый параметр, характеризующий распределительную мощность и запаздывание, снижается около 80 %. Ил. 4, библ. 14 (на английском языке; рефераты на английском, русском и литовском яз.).

P. Vijayakumar, K. Gunavathi. CMOS grandynų projektavimas siekiant minimizuoti galios ir vėlinimo parametrus // Elektronika ir elektrotechnika – Kaunas: Technologija, 2007. – Nr. 2(75). – P. 27–30.

Siūlomas algoritmas, skirtas galios ir vėlinimo parametrams efektyviai optimizuoti CMOS loginiuose grandynuose. Siūlomas algoritmas, kaip sumažinti tiek išsklaidomąją galią, tiek vėlinimą. Tai pasiekiama taikant pakopinės įkrovos ir vėlinimo derinimo metodus. Taikant laiko derinimą, grandynas skaidomas į atskiras pakopas, kurių vėlinimas gali būti valdomas individualiai. Po to pritaikomas pakopinės įkrovos metodas, kai maitinimo įtampa keičiama pakopomis, kol pasiekia maksimalią vertę. Taip sumažinama išsklaidomoji galia. Algoritmas testuotas naudojant ISCAS testavimo grandynus. Eksperimentų rezultatai rodo, jog jungtinis išsklaidomąją galią ir vėlinimą apibūdinantis parametras sumažėja apie 80 %. Il. 4, bibl. 14 (anglų kalba; santraukos anglų, rusų ir lietuvių k.).