2010 IEEE 26-th Convention of Electrical and Electronics Engineers in Israel

Adaptive Clock Gating for Shift Register Based Circuits

Shmuel Wimer
Eng. School Bar-Ilan University
Ramat-Gan 52900, Israel

Israel Koren
ECE Dept. University of Massachusetts
Amherst, MA 01003, USA

Itamar Cohen
Eng. School Bar-Ilan University
Ramat-Gan 52900, Israel

Abstract - Clock gating is a widely used technique for dynamic power reduction in VLSI design. In its most straightforward application it allows disabling the clock signal of a flip-flop once its state is no longer subject to changes. This paper extends this technique one step further and proposes a systematic way to achieve additional dynamic power savings based on the correlation of flip-flops' activities. Circuits based on shift registers are widely used in digital systems and we selected them to demonstrate the effectiveness of the proposed method. The best, worst and average cases for dynamic power savings tare analyzed.

I. INTRODUCTION

Market demand for low power mobile computing and consumer electronics products has refocused VLSI design in the last decade on lowering power and increasing energy efficiency. The clock signal of a digital systems is a major dynamic power consumer; typically responsible for 50% of the total dynamic power consumed. Many design methodologies and techniques to reduce the dynamic power have been developed, of which clock gating is the most popular and well established in the design community. Clock gating is employed in all levels: system architecture, block design [1], logic design and local circuits [2]. Clock enabling signals are usually introduced by designers at the system and block design phases, where the interdependencies of the various functions are well understood. Going down to the circuit implementation, it is very difficult to define such signals since the interdependencies of the state of various flip-flops depend on the gate-level implementation which is usually automatically synthesized.

This paper presents an approach to maximize the clock gating at the circuit level, where the clock signal that is driving a given flip-flop is disabled (gated) once the flip-flop's state is not subject to a change in the next clock cycle. Clock gating does not come for free. Extra circuits are required for its implementation, and therefore, its potential benefit should be predicted beforehand. The data sequences, and hence the implied state transitions of flip-flops in digital systems like microprocessors, controllers, DSP and other applications, depend on the typical data processed by those. Assessing the effectiveness of clock gating may require therefore, extensive simulations, trace derivation and statistical analysis of sequential elements activity. As shown later in this paper, analyzing this data for further exploitation of clock gating beyond what is done for individual flip-flops requires more effort. It is therefore beneficial to study digital circuits whose state sequence can be derived directly from their structure. Shift register based circuits have this property and they include, for example, counters, Linear Feedback Shift Registers (LFSRs) that are used for pseudo-random number generation, serial adders, and Tapped Delay Lines (TDL) that are used in signal processing applications [3]. The simple structure of these shift register based circuits enable accurate prediction of clock gating power savings.

978-1-4244-8682-3/10/$26.00 ©2010 IEEE

II. CLOCK GATING OF FLIP-FLOPS

Fig. 1 shows how a sequential element can find out that its clock can be disabled in the next cycle. A XOR gate compares its output at the present cycle with the present data input, that is supposed to appear at the output in the next cycle. The *clk_enable* output of the XOR indicates whether or not a clock signal will be required in next cycle. The clock driver shown in Fig. 1(a) is then replaced by a 2-way AND gate where the clock signal is enabled. We will use the symbol in Fig. 1(b) to represent sequential elements that incorporate generation of *clk_enable*.

Controlling the clock in each flip-flop by a dedicated gater using the flip-flop's *clk_enable* signal was studied in [4]. An implementation for LFSR has been discussed recently in [5], where after taking into account the power consumed by the extra circuitry, 10% net power reduction was reported. Additional power reduction can be achieved by reducing the number of clock gaters. We could control several flip-flops with a common gater if we knew that they are toggling simultaneously most of the time, thus achieving almost the same power reduction, but with fewer gaters. The grouping may contain up to several dozens of flip-flops in a single group, and is usually done in the physical VLSI design phase by clock tree synthesizers [6]. Such tools are focusing on skew, power and area minimization, and are not aware of the toggling correlations of the underlying flip-flops.

Figs. 2 and 3 demonstrate the trade-off between the number of disabled clock pulses and the amount of hardware needed for the gaters' implementation. Fig. 2 shows how to join k *clk_enable* signals generated by distinct flip-flops into one gating signal. It saves the individual clock gaters used for every flip-flop in the expense of introducing an OR gate and a latch required to captures the enable signal and holds it until the next clock cycle. Clearly, the hardware save increases

with k, but the amount of disabled clock pulses is decreasing. It is therefore required for gating scheme proposed in Fig. 2 to be highly beneficial that the clock enabling signals of the grouped flip-flops to be highly correlated. This paper develops later a technique for obtaining the flip-flops groups yielding maximal activity correlation. A simple example is shown in Fig. 3(a) where each clock cycle produces the appropriate value of *clk_enable* for a period of 10 cycles. Clocking each flip-flop separately, 14 clock pulses are disabled out of the total 20, yielding 70% savings in the cost of two AND gaters. Fig. 3(b) generates a joint gated clock with the aid of the gater in Fig. 2. As shown, while a single gater is used, the clock toggling power saving reduced to 60%. Obviously, a key factor in obtaining effective joint clock gating is in finding large groups of flip-flops having similar toggling. Accurate formulation of flip-flop grouping problem supplemented with optimal algorithm is presented later.

Joining enabling signals of individual flip-flops suits very well clock-tree distribution network commonly used in digital circuits. A typical topological structure of such a tree is shown in Fig. 4. The clock signal enters the block at a pin called root, and then it is driven downstream to sequential circuits trough chains of drivers connected in a tree topology. It is possible to replace the drivers of the tree in Fig. 5 by k-way gaters proposed in Fig.2. A gater of the tree receives the enabling signals of its k children and then delivers the clock signal downstream accordingly.

III. LFSR CLOCK GATING

LFSR (Linear Feedback Shift Register) is a well-known circuit for pseudo-random number generation. Since the register flip-flops are toggling in a pseudo-random manner, it is interesting to investigate the potential power savings that can be achieved by disabling clock pulses to the flip-flops comprising the register, when those clock pulses are unnecessary. Its analysis can serve other shift register based circuits whose toggling is known a priori (e.g., counters) or statistically (e.g., serial adder). Due to the random nature of the flip-flops' toggling we would expect only very limited power savings. We show however, that for a maximum period N-bit LFSR (that cycles through 2^N-1 states), in the best case $(N-1)(2^{N-1})$ out of the total $N(2^{N-1})$ clock pulses can be disabled, while in the worst case only $\alpha(2^{N-1})$ clock pulses are disabled, where α is some constant. It is further shown that the average number of disabled clock pulses is $(N/2)(2^{N-1})$, half way between the worst and best cases. Since clock disabling requires extra circuitry in the register and clock network, the paper explores grouping the flip-flops of the LFSR to be driven by common clock gaters in an attempt to reduce circuit overhead. We show that in a binary clock-tree (fan-out of 2) the average number of saved clock pulses is $1/4$ of the total, while in a k-fan-out clock-tree the average number is $1/2^k$. The paper shows the circuit implementation and discusses the timing implications and circuit overhead.

A. Counting the number of disabled clock pulses

Let the LFSR contain N D-flip-flops and have a maximal period, thus visiting 2^N-1 distinct states, i.e., all possible states except the 00…0 state. Let us denote by w_i, $1 \le i \le 2^N-1$ the cycle states, and tabulate those in a $(2^N-1)\times N$ matrix. An entry m_{ij} of $\mathbf{M}$, $1 \le i \le 2^N$, $0 \le j \le N-1$ represents the output Q the flip-flop j at state i. Therefore, column c_j of $\mathbf{M}$, $0 \le j \le N$ shows the toggling of flip-flop j during the entire cycle.

Obviously, each row of $\mathbf{M}'$ must contain at least a single 1 as otherwise there would have been two successive identical states in a cycle, which is impossible. Among all maximum period LFSRs, there is one that follows a Gray code for which successive states differ only in one bit. For this LFSR, there is a single 1 in every row of $\mathbf{M}'$ with one exception - the transition from $w_{2^N-1} = 10\cdots00$ back to $w_1 = 00\cdots01$, resulting in two 1s in $\mathbf{M}'$. We conclude that out of the $N(2^{N-1})$ total entries of $\mathbf{M}'$ for the Gray code LFSR, there are $N(2^{N-1})-2^N$ 0s, for which the clock pulse can be disabled. We next summarize the above observations.

Proposition 1: Given an N-bit LFSR with a period of 2^N-1 states, the maximal number of clock pulses that can be disabled out of the $N(2^{N-1})$ clock pulses in the cycle is $N(2^{N-1})-2^N$ and the disabling ratio approaches 1 with increasing N.

Consider the "time derivative" matrix $\mathbf{M}'$ defined as follows:

(1.1) $$m'_{ij} = \begin{cases} 0 & \text{if } m_{i,j} = m_{i+1,j} \\ 1 & \text{otherwise} \end{cases}, \quad 1 \le i \le 2^N-1,\ 0 \le j \le N-1.$$

The row index in (1.1) is taken cyclically, i.e., row i is followed by $i+1 \bmod N$. An entry $m'_{ij} = 0$ means that flip-flop j does not change from state i to state $i+1$, hence it need not be clocked for this state transition and the clock pulse driving that flip-flop can be disabled. In the following we find the maximum, minimum and expected number of disabled clock pulses.

We next search for the maximum period LFSR that yields the smallest number of disabled clock pulses. We denote a pair of successive states of $\mathbf{M}$ by $\langle w_i, w_{i+1} \rangle$ and the number of bits they have in common by $a(\langle w_i, w_{i+1} \rangle)$. Therefore, $a(\langle w_i, w_{i+1} \rangle)$ equals N minus the Hamming distance between the two states, and it is the number of clock pulses required for the transition from w_i to w_{i+1}. To ease the search for the desired LFSR we assume for the moment that the word $00\cdots0$ is

included in $\mathbf{M}$ (we will drop it later). To find the worst cycle we wish to minimize $\sum_{i=1}^{2^N} a(\langle w_i, w_{i+1}\rangle)$, where the sum is taken cyclically. A pair $\langle w', w''\rangle$ comprising of two complementary codes clearly satisfies $a(\langle w', w''\rangle) = 0$. Therefore, if there exists an $\mathbf{M}$ that includes 2^{N-1} such pairs, no clock pulse could be disabled during the 2^{N-1} state transitions and $\mathbf{M}$ will correspond to the worst cycle. The order of the complementary pairs $\langle w', w''\rangle$ in the worst cycle

2. $\mathbf{M}_1$ and $\mathbf{S}_1$ are concatenated and then a Most Significant Bit (MSB) with value 0 is added to every odd indexed row and a MSB of value 1 to every even indexed row, yielding $\mathbf{M}_2 = [\langle 00,11\rangle, \langle 01,10\rangle]$.

Notice that $\mathbf{M}_2$ includes all distinct 2-bit states. The addition of an MSB in an alternating manner ensures that the pair $\langle w_i, w_{i+1}\rangle$, for i odd, consists of complementary states, hence $a(\langle w_i, w_{i+1}\rangle) = 0$ is invariant of the construction. For i even we have $a(\langle w_i, w_{i+1}\rangle) = 1$.

To illustrate the general rule for $a(\langle w_i, w_{i+1}\rangle)$ for i even, consider $\mathbf{M}_3$, which based on the above construction procedure is: $\mathbf{M}_3 = [\langle \bullet\bullet\bullet, 111\rangle, \langle \bullet\bullet 1, 11\bullet\rangle, \langle \bullet 11, 1\bullet\bullet\rangle, \langle \bullet 1\bullet, 1\bullet 1\rangle]$. Due to the alternating MSBs $a(\langle w_i, w_{i+1}\rangle)$ does not change in the upper half of $\mathbf{M}_3$, obtained from $\mathbf{M}_2$. In the lower half obtained from $\mathbf{S}_2$, $a(\langle 00,10\rangle)$ turns into $a(\langle 100,010\rangle)$ in $\mathbf{M}_3$, which also preserves the number of identical bits. The only increase in the count of bit agreements can occur at the two transitions between $\mathbf{M}_2$ (upper half of $\mathbf{M}_3$) and $\mathbf{S}_2$ (lower half of $\mathbf{M}_3$). In that case, there exists $a(\langle 110,011\rangle) = a(\langle 101,000\rangle) = 1$. Applying the construction procedure for $\mathbf{M}_4$, state transitions between $\mathbf{M}_3$ and $\mathbf{S}_3$ satisfy $a(\langle 1101,0111\rangle) = a(\langle 1010,0000\rangle) = 2$. This bit agreement is governed by the following two invariants of extending $\mathbf{M}_{K-1}$ into $\mathbf{M}_K$:

So far we have included the $00\cdots0$ state, but since LFSRs exclude it, we need to remove it from $\mathbf{M}_N$. This turns the 3-state sequence $\cdots0101\cdots$, $\bullet\bullet\cdots\bullet$ and $11\cdots1$, into the 2-state sequence $\cdots0101\cdots$ and $11\cdots1$, which adds $\lceil N/2\rceil$ bit agreements between successive states, but does not change the complexity of T_N obtained in (1.5). We summarize the above as follows:

Proposition 2: Given an N-bit LFSR with a period of $2^N - 1$ states, the minimal number of clock pulses that can be disabled out of $N(2^N - 1)$ clock pulses in the cycle is $O(2^N)$ and the disabling ratio approaches 0 with increasing N.

$\mathbf{M}$ can be obtained by the inductive construction described below.

For the basis of the induction consider $N = 1$ and $\mathbf{M}_1 = [\langle \bullet, 1\rangle]$. Clearly, $a(\langle \bullet, 1\rangle) = a(\langle 1, \bullet\rangle) = \bullet$. The matrix $\mathbf{M}_2$ is obtained as follows:

1. From $\mathbf{M}_1$ we generate a new matrix, denoted by $\mathbf{S}_1$, where the states of every pair are swapped, thus, $\mathbf{S}_1 = [\langle 1,0\rangle]$.
1. The first and second states of $\mathbf{M}_K$ are $00\cdots0$ and $11\cdots1$, respectively, and
2. The last two states of $\mathbf{M}_K$ consist of alternating bits; one has a Least Significant Bit (LSB) of 0 and the other a LSB of 1, or vise versa, depending on the evenness of K.

We conclude that upon constructing $\mathbf{M}_K$, the number of additional bit agreements between successive states involving $\mathbf{M}_{K-1}$ and $\mathbf{S}_{K-1}$ is:

(1.2) $$\left[a(\langle w_{2^{K-1}}, w_{2^{K-1}+1}\rangle) + a(\langle w_{2^K}, w_1\rangle)\right] - \left[a(\langle w_{2^{K-1}}, w_1\rangle) + a(\langle w_{2^K}, w_{2^{K-1}+1}\rangle)\right].$$

The subtrahend in the above equation follows from the disappearance of the transitions from $w_{2^{K-1}}$ to w_1 in $\mathbf{M}_{K-1}$ and from w_{2^K} to $w_{2^{K-1}+1}$ in $\mathbf{S}_{K-1}$. Notice that in the state pairs of (1.2) one is either $00\cdots0$ or $11\cdots1$, while the other has alternating 0 and 1 bits. Taking the evenness of K into account, the net increase in bit agreement is:

(1.3) $$2\left\lfloor \frac{K}{2}\right\rfloor - 2\left\lfloor \frac{K-1}{2}\right\rfloor = \begin{cases} 0 & K \text{ odd} \\ 2 & K \text{ even} \end{cases}.$$

We denote by T_N the total number of bit agreements resulting from repetitive application of matrix duplication. It then satisfies the following recursive equation:

(1.4) $$T_N \le 2T_{N-1} + 2,$$

whose solution is:

(1.5) $$T_N = \alpha 2^N,$$

where α is some constant.

Of most interest is the average case, for which the following can be stated.

Proposition 3: Given an N-bit LFSR with a period of $2^N - 1$ states, the expected number of disabled clock pulses in the cycle is $(N/2)(2^N - 1)$, half way between the worst and best cases.

The proposition follows by assuming that two successive states are random and independent of each other. The probability of a bit to change in a state transition is therefore

0.5, yielding an average of $(N/2)(2^N-1)$ clock pulses that can be disabled.

IV. JOINT CLOCK GATING

As described above, gating the clock pulses of individual flip-flops requires an AND clock gater for each flip-flop, which constitutes a significant circuit overhead. Consider two flip-flops, whose toggling is highly correlated, namely, their outputs are likely to change simultaneously in the same states. In such a case, controlling the two flip-flops by a common clock gater is beneficial, even though we may end up sometimes clocking a flip-flop although it does not change its output while its counterpart does. If this happens only rarely, designing a common clock gater is beneficial. Driving several flip-flops jointly by a single clock buffer is a common practice supported by most commercial clock tree synthesizers.

M					**M′**			
0	**0**	**0**	**1**		**1**	**0**	**0**	**1**
0	**0**	**1**	**1**		**0**	**0**	**1**	**0**
0	**0**	**1**	**0**		**0**	**0**	**0**	**1**
0	**1**	**1**	**0**		**0**	**1**	**0**	**0**
0	**1**	**1**	**1**		**0**	**0**	**0**	**1**
0	**1**	**0**	**1**		**0**	**0**	**1**	**0**
0	**1**	**0**	**0**		**0**	**0**	**0**	**1**
1	**1**	**0**	**0**		**1**	**0**	**0**	**0**
1	**1**	**0**	**1**		**0**	**0**	**0**	**1**
1	**1**	**1**	**1**		**0**	**0**	**1**	**0**
1	**1**	**1**	**0**		**0**	**0**	**0**	**1**
1	**0**	**1**	**0**		**0**	**1**	**0**	**0**
1	**0**	**1**	**1**		**0**	**0**	**0**	**1**
1	**0**	**0**	**1**		**0**	**0**	**1**	**0**
1	**0**	**0**	**0**		**0**	**0**	**0**	**1**

Except the first row which contains two 1s (due to the exclusion of the zero state from the LFSR), each of the other rows contain a single 1. Half of those 1s belong to the LSB column. Therefore, sharing a common gater by the LSB flip-flop and any other flip-flop is undesirable since the other flip-flop will be clocked 2^{N-1} times unnecessarily. In contrast, the first two columns contain only two 1s each, so the clock disabling of the corresponding flip-flops is highly correlated, justifying sharing a common gater by the two MSBs. Though not very likely to occur in an LFSR, the following observation is useful for an N-bit Gray-code counter. Sharing gaters by MSBs is useful allowing the saving of hardware resources in return to a very limited unnecessary clocking of flip-flops. Sharing gaters by LSBs is not as useful, as it produces excessive unnecessary clocking of flip-flops. Generally, since the 1s in the columns of $\mathbf{M}'$ are mutually exclusive (except the first row), any pairing of flip-flops to share a common clock gater doubles the number of flip-flop clock pulses compared to using individual clock gaters. Still, compared to the total $N(2^N-1)$ clock pulses required without gating, this increase is small. On the other hand, the number of clock gaters has been reduced from N to $N/2$. More generally, using joint gaters with k *clk_enable* inputs will cut the number of gaters to $\lceil N/k \rceil$ but will increase the number of clock pulses to $k(2^N-1)$.

In the worst case discussed in proposition 2, joint clock gating is not useful at all since most of the flip-flops are toggling most of the time. More interesting is the average case stated in proposition 3, for which the following result holds.

Proposition 4: Given an N-bit LFSR with a period of 2^N-1 states and N an even number, the expected number of jointly disabled clock pulses is $(N/4)(2^N-1)$, namely, for an arbitrary flip-flop pairing in an LFSR such that every pair shares a common clock gater, one quarter of the entire clock pulses can be disabled.

Proof: The entries of $\mathbf{M}'$ represent the successive bit transitions of the entries in $\mathbf{M}$. Those transitions can be any of 4 possible types (i.e., $0\to0$, $0\to1$, $1\to0$, and $1\to1$), are independent and each of them has the same probability. Consequently, half of $\mathbf{M}'$'s entries are 0 and half are 1. Consider two arbitrary columns of $\mathbf{M}'$. The joint clock gater is disabled for the two corresponding flip-flops only if both columns have 0 in a given row, which has a probability of $1/2\times1/2=1/4$ to occur. ■

A gated clock tree can be constructed from joint gaters illustrated in Fig. 2. In practice, a joint gater will drive multiple flip-flops, but for the sake of analysis we will assume a binary tree. Fig. 5 illustrates how such gaters are connected to create the tree. Clearly, there is no point in enabling the clock buffer at the root. This follows from the LFSR very definition, where all states are distinct, so $\mathbf{M}'$ cannot contain a row whose bits are all 0, implying that the enabling signal at root is always 1.

To asses the benefits of adaptive gating in a binary clock tree, let us return to the best LFSR case for clock gating, i.e., the one that follows a Gray code encoding as discussed in proposition 1. For illustration the corresponding matrices $\mathbf{M}$ and $\mathbf{M}'$ for $N=4$ are shown below.

Similarly to flip-flop pairing at the leaves of a binary clock tree, we can proceed upwards by pairing two gaters and driving those by an upper level gater, thus constructing the binary clock tree bottom-up. The connections of gaters in a binary clock tree are shown in Fig. 5 and imply very strict delay constraints, which are not discussed in this paper. When the underlying circuit occupies significant silicon area (N may reach few tens of thousands in a large block), only the leaves and one or two levels above should be considered for an adaptive clock gating.

V. APPLICATION TO OTHER CIRCUITS

The result of proposition 1 is applicable to Gray code counter a, which from power saving perspective is the best since the disabling ratio approaches 1 with increasing N. A different clock gating approach was proposed in [7]. It is

based on developing the specific Boolean expression for every flip-flop, describing sufficient toggling conditions based on all other flip-flops. This method is robust and has the advantage of not requiring latching disable signals. It looks however, that the amount of logic per flip-flop increases with N which is undesirable. Another shift register application is for tapped delay line (TDL) widely used in DSP. Assuming that the digital signal filtered by the TDL is random with equal probability of 0 and 1, the same power saving as for the LFSR's average case can be expected. This is also true for the shift registers storing the addend and augend in a serial adder.

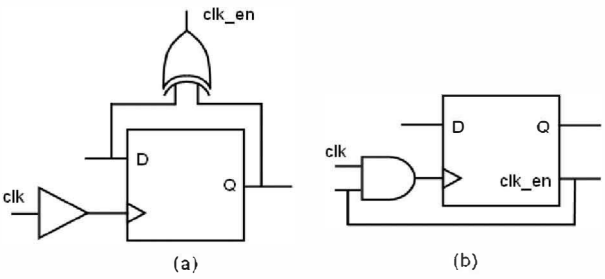


Figure 1: Enabling of the clock signal.

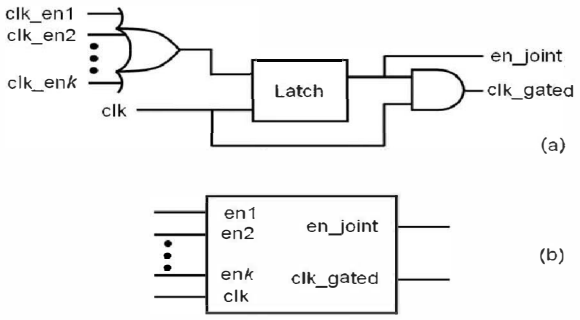


Figure 2: Joining k enabling signals generated by distinct flip-flops into one gating signal.

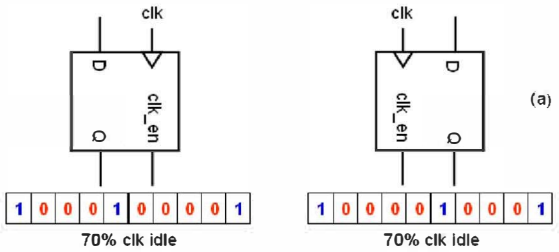
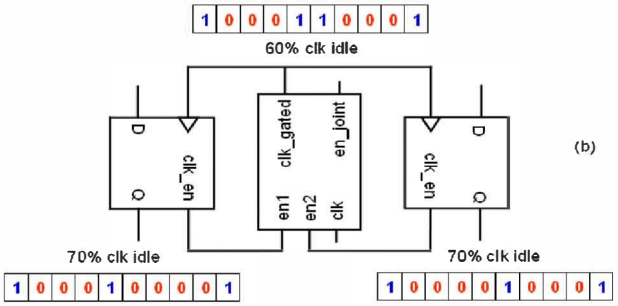


Figure 3: Example of joint gated clock.



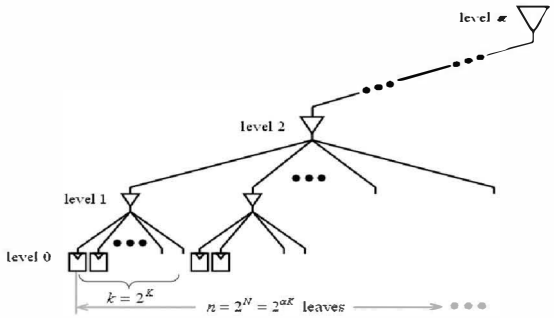


Figure 4: Clock tree distribution.

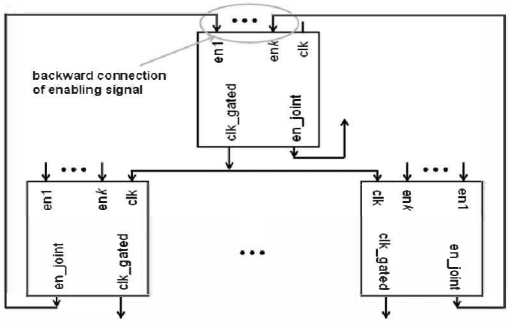


Figure 5: Replacing the drivers of a clock tree by k -way gaters. The k input of the internal OR gate are propagating bottom-up.

REFERENCES

[1] Benini L., Bogliolo A and De Micheli G, "A survey on design techniques for system-level dynamic power management," *IEEE Trans. on VLSI Systems*, Vol. 8, No. 3, June 2000, pp. 299-316.

[2] Hosny M. S. and Yuejian W, "Low power clocking strategies in deep submicron technologies," *IEEE Intl. Conf. on Integrated Circuit Design and Technology*, ICICDT 2008, pp. 143-146.

[3] Weste N. H. E. and Harris D., *CMOS VLSI Design – a Circuit and System Perspective*, Addison-Wesley, 2005.

[4] Lang T., Musoll E. and Cortadella J., "Individual flip-flops with gated clocks for low power datapaths," *IEEE Trans. on Circuits and Systems-II*, Vol. 44, No. 6, June 1997, pp. 507-516.

[5] Aloisi W. and Mita R., "Gated-clock design of linear-feedback shift registers," *IEEE Trans. on Circuits and Systems-II*, Vol. 55, No. 5, June 2008, pp. 546-550.

[6] "Low skew – low power CTS methodology in SOC Encounter for ARM processor cores," http://www.cadence.com/cdnlive/library/Documents/2009/EMEA/DI10_DaveKinjal_ARM_FINAL.pdf

[7] Wu Q., Pedram M. and Wu X., "Clock-gating and its application to low power design of sequential circuits," *IEEE Trans. on Circuits and Systems-I*, Vol. 47, No. 103, March 2000, pp. 415-420.