Analysis and Reduction of Glitches in Synchronous Networks

Jeroen Leijten[1,2], Jef van Meerbergen[1] and Jochen Jess[2]

[1] Philips Research Laboratories, WAY 4.47, Prof. Holstlaan 4, 5656 AA Eindhoven, The Netherlands
[2] Department of Electrical Engineering, Eindhoven University of Technology, The Netherlands

Abstract

The influence of transition activity on dynamic power dissipation is analysed by examining three components: dissipation in combinational logic, flipflops and clock line. Transition activity is analysed by making a distinction between useful transitions and glitches (useless transitions). A transition counting and parity evaluation method is used for this.

Most glitches can be eliminated by introducing flipflops using retiming and pipelining and/or by choosing different architectures. In this way an optimal level for pipelining can be found.

1 Introduction

As battery powered products using integrated circuits become more important, severe constraints are imposed on the power that may be consumed by these circuits. Apart from this, the integration of an increasing number of transistors on a single chip can lead to areas of excessive power dissipation which can cause reliability problems. Furthermore excessive power dissipation leads to IC-packaging problems. These considerations reveal the need for a reduction in power dissipation of integrated circuits.
Several actions at different levels of the design can be taken to reduce power dissipation in CMOS integrated circuits [1]. Our research focuses on gate level architectures for synchronous static CMOS circuits. Glitches can occur in these circuits, leading to unnecessary power dissipation. Other research [2][3][4] shows that the number of transitions can be counted and used to obtain a circuit performance measure regarding power dissipation. It further shows that the reduction of glitches helps to obtain a significant decrease in power dissipation.

In our work a new transition counting method is used in which a distinction is made between useful transitions and glitches. First we analysed the large impact of glitches on transition activity using probability calculations on a ripple carry adder. Next we investigated the influence of delay imbalance in circuit architectures on the number of glitches. This was done using unit delay gate level simulations on different multiplier architectures.

Layout level simulations were carried out on a typical video processing unit. The effect of introducing flipflops (by using retiming) to reduce the amount of glitches was investigated using these simulations. For this purpose dynamic power dissipation was examined by dividing it in three components: dissipation in the combinational logic, in the flipflops and in the clock line.

After some preliminary remarks regarding transition activity and power dissipation our methods and results are presented. Finally our conclusions will be given.

2 Transition activity and power dissipation

We will refer to a signal change in a circuit node from logic level 0 to logic level 1 or vice versa as a signal *transition.* A signal change from 0 to 1 will be termed a *power-consuming transition*, because in this case the loading capacitance will be charged directly from the supply and therefore power from the supply is consumed by the circuit. This can be seen from Figure 1 where the effect of a transition in an arbitrary circuit node is depicted.

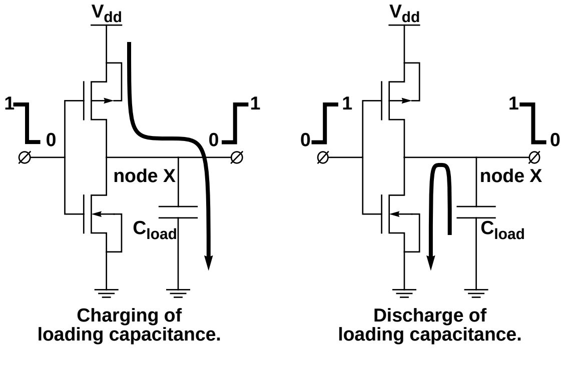
V_{dd}
1
0
∅
node X
C_{load}
0
1
V_{dd}
0
1
∅
1
0
node X
C_{load}
Charging of loading capacitance.
Discharge of loading capacitance.

Figure 1 Dynamic power dissipation caused by signal transitions in node X.

A transition can be either *useful* or *useless*. A transition is termed useful when it is needed to ensure the correct functional behaviour of the architecture being used. Otherwise it is termed useless. Whether a transition is useful or useless depends on other possible transitions in the same clock cycle. This will be further explained in section 3.3. Two consecutive useless transitions constitute a *glitch*. Figure 2 shows an arbitrary binary signal exhibiting these different forms of transitions.

In equation 1 the dynamic power dissipation in CMOS is given as a function of the probability of a power-consuming transition p_t , load capacitance C_{load} , supply voltage V_{dd} and clock frequency *f*.

$$P_{dyn} = p_t C_{load} V_{dd}^2 f \qquad \textbf{(1)}$$

In static CMOS the dynamic power dissipation dominates the total power dissipation. From 1 it is clear that the dynamic power dissipation is linearly dependent on the number of power-consuming signal transitions and consequently on the number of glitches.

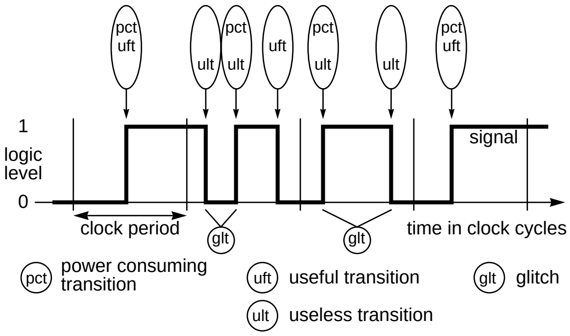

Figure 2 Possible forms of signal transitions.

3 Probability analysis of transition activity

To gain a first insight in the problem of glitches in logic circuits, transition activity in a ripple carry adder was analysed using probability calculations.

An *N*-bit ripple carry adder consists of *N* cascaded full adders (see Figure 3). Here we assume a unit delay model for every full adder stage. We further assume that new input bits A_i and B_i of the two input operands $A_{N-1}...A_0$ and $B_{N-1}...B_0$ always arrive at the beginning of a clock cycle.

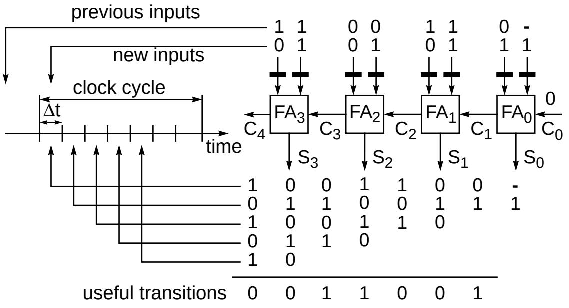

Figure 3 Worst case number of transitions in an N=4 bit ripple carry adder.

When the inputs change, full adder FA_0 will introduce a delay in the calculation of sum bit S_0 and carry out bit C_1. Thus full adder FA_1 will first compute its sum S_1 and carry out C_2 using the new input bits A_1 and B_1 as well as the old carry input C_1 of the previous addition. After this C_1 will reach its new value and full adder FA_1 will perform another calculation with this new value. This means that two transitions can occur in S_1 and C_2 within a single clock cycle.

Similarly it can be found that in total three transitions can occur in S_2 and C_3. In general we find that *i+1* transitions can occur in S_i and C_{i+1} of full adder FA_i within a single clock period. This means that in a worst case situation *N* transitions occur in S_{N-1} and C_N.

3.1 Worst case number of transitions

The worst case situation is only possible if C_N has the values *0,1,0,1,...* or *1,0,1,0,...* for the delta time moments $\Delta t_0, \Delta t_1, \Delta t_2, ..., \Delta t_{N-1}$ within a single clock period. This can only occur if two conditions are met:

1. After the completion of the previous addition the carry bits of the full adders have the values $(C_N, C_{N-1}, C_{N-2}, C_{N-3}) = (0,1,0,1,...)$ or $(C_N, C_{N-1}, C_{N-2}, C_{N-3}, ...) = (1,0,1,0,...)$.
2. The carry must be able to ripple through all *N* full adder stages.

It can be shown [5] that the probability that both conditions are met is equal to $3 \cdot (1/8)^N$ for random inputs. Already for relatively small word sizes *N* this probability will be negligible. More interest lies in the average number of transitions appearing in a circuit.

3.2 Average number of transitions for random inputs

We have calculated the average number of transitions per clock period appearing in a signal X_i. We call this figure the average transition ratio $TR(X_i)$.

Arithmetic circuits, like multipliers and adders, are often used in a multiplexed environment. Due to this multiplexing the original signal statistics and correlations are lost, and randomly distributed input signals arrive at the inputs of the arithmetic circuits. Furthermore source coding is often applied to signals that are fed as inputs to arithmetic circuits. Source coding also results in almost complete loss of signal correlation. Under those conditions assuming random inputs gives a good approximation of a practical situation in which these arithmetic circuits are used.

In [5] we show that for random inputs the average transition ratio $TR(C_{i+1})$ for carry out C_{i+1} of full adder FA_i in a ripple carry adder is equal to:

$$TR(C_{i+1}) = \frac{3}{4} - \frac{3}{4} \cdot \left(\frac{1}{2}\right)^{i+1} \qquad \textbf{(2)}$$

The average transition ratio $TR(S_i)$ for sum S_i of full adder FA_i is equal to:

$$TR(S_i) = \frac{5}{4} - \frac{3}{4} \cdot \left(\frac{1}{2}\right)^i \quad \textbf{(3)}$$

3.3 Average number of useful and useless transitions

A useful transition in a signal occurs when its final value which resulted from the previous addition differs from the value resulting from the new addition. In other words, a useful transition occurs if the signal makes transitions *0→1→0→...→1* or *1→0→1→...→0* within a single clock cycle. From this observation the following properties of useful and useless transitions can be derived:

1. If the number of transitions a signal makes within one clock period is an odd number, this signal makes *one* useful transition. All extra transitions it makes are useless transitions.
2. If the number of transitions a signal makes within one clock period is an even number, all these transitions are useless transitions.

An example for this is given in Figure 4.

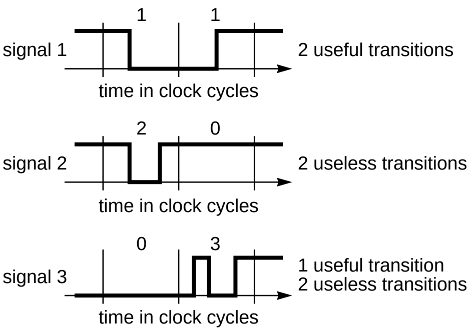

Figure 4 Useful and useless transitions in a single clock cycle.

We used these two properties and probability calculations to determine the average useful transition ratios per clock cycle *(UFTR(X_i))* and average useless transition ratios per clock cycle *(ULTR(X_i))* [5]. The following ratios were found:

$$UFTR(S_i) = \frac{1}{2} \quad \textbf{(4)}$$

$$ULTR(S_i) = \frac{3}{4} - \frac{3}{4} \cdot \left(\frac{1}{2}\right)^i \quad \textbf{(5)}$$

$$UFTR(C_{i+1}) = \frac{1}{2} - \frac{1}{2} \cdot \left(\frac{1}{4}\right)^{i+1} \quad \textbf{(6)}$$

$$ULTR(C_{i+1}) = \frac{1}{2} \cdot \left(\left(\frac{1}{2}\right)^{i+1} - \frac{1}{2}\right) \cdot \left(\left(\frac{1}{2}\right)^{i+1} - 1\right) \quad \textbf{(7)}$$

If we multiply these ratios with the total number of clock cycles (or equivalently the total number of input stimuli) the average number of transitions, useful transitions, and useless transitions can be calculated. This is done in Figure 5 for 4000 random inputs applied to a ripple carry adder.

It is apparent that the number of useless transitions constitutes a large part of the total number of transitions appearing in the circuit. For the ripple carry adder used with 4000 random inputs, a total number of 119002 transitions is found using equations 2 and 3 and summing the results for all sixteen sum and carry bits. Using equations 4, 5, 6 and 7 it can be found that 63334 of these transitions are useful. The remaining 55668 transitions are useless transitions. In other words, the ratio of useless transitions to useful transitions *L/F* is equal to *L/F = 55668/63334 = 0.88*.

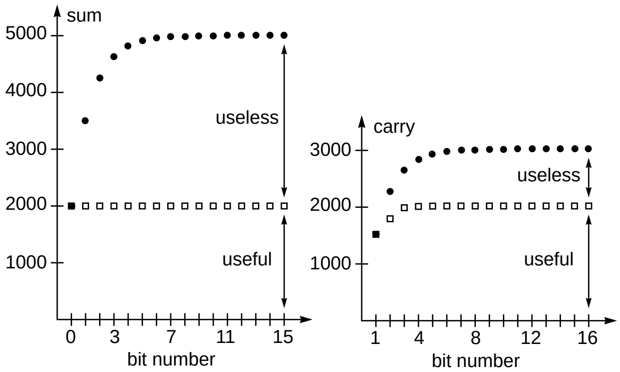

Figure 5 Useful transitions and useless transitions occurring for 4000 random inputs.

4 Influence of delay imbalance

To investigate the effect of delay imbalance on transition activity, gate level simulations were carried out on different architectures. These architectures implement the same functional behaviour, but have more or less unbalanced delay paths in their structures. During the gate level simulation all internal signal nodes are monitored. The number of signal transitions for each monitored signal and for each clock cycle is counted. Using parity evaluation according to the two properties mentioned in section 3.3 a distinction between useful and useless transitions can then be made.

4.1 Multiplier architectures

Multipliers have functional properties that make it easy to introduce more or less delay unbalanced paths in their architecture. For example, a choice between an array architecture and a wallace tree architecture can be made. These types of multiplier architectures were used as test cases to investigate the influence of delay imbalance on transition activity.

Figure 6 shows the basic architecture of an 8x8 array multiplier for positive numbers. It is clear that many unbalanced delay paths exist in this type of architecture. Figure

7 shows the architecture of an 8x8 wallace tree multiplier. This architecture has a much more balanced structure.

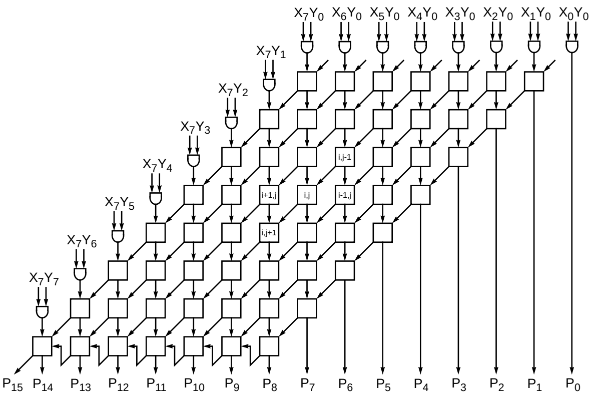

Figure 6 8x8 array multiplier architecture.

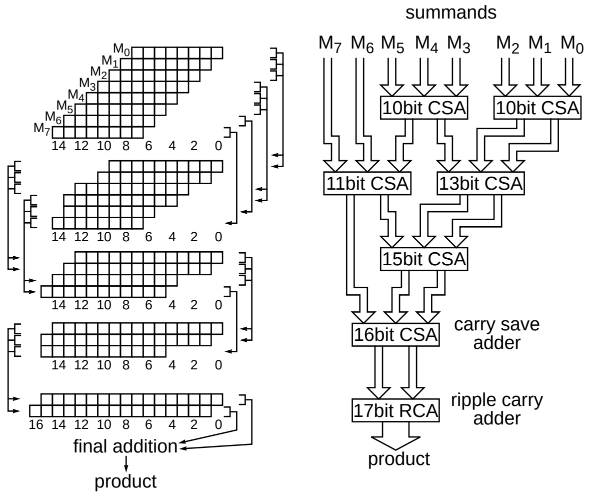

Figure 7 8x8 Wallace tree multiplier architecture.

Both multiplier architectures were simulated using unit delay modelling for the multiplier cells. This was done both for the 8x8 multipliers and their 16x16 equivalents. The results of our simulations using 500 random inputs are given in Table 1.

Table 1 Transition activity for 500 random inputs.

	array mply		wallace tree mply	
	8x8	16x16	8x8	16x16
total	58858	438575	50824	200380
useful F	23418	102845	39608	173330
useless L	35440	335730	11216	27050
L/F	1.51	3.26	0.28	0.16

From this table it can be seen that the wallace tree multiplier has far less useless transitions and a much better useless/useful ratio than the array multiplier. The 16x16 wallace tree multiplier has more useful transitions than the 16x16 array multiplier due to its larger number of gates. However, the number of useless transitions in the 16x16 wallace tree is much smaller than in the 16x16 array multiplier. It is clear that decreasing the number of unbalanced delay paths in the circuit architecture significantly reduces the number of useless transitions.

In practice the delay of the sum calculation in a full adder is about twice as large as the delay of the carry calculation. A more realistic unit delay modelling of the 8x8 multipliers was therefore applied by defining the delay of the sum generation δ_{sum} in every full adder (in a multiplier cell) to be twice as large as the delay of the carry generation δ_{carry}. The same transition activity simulations were then carried out on these more accurate models and compared to the previously used unit delay models (which used equal sum and carry delays). The results of these simulations are shown in Table 2. As expected because of the increased delay imbalance, the number of useless transitions is higher. This further deteriorates the useless/useful ratios.

The difference in the number of useless transitions appearing in the array multiplier compared to the number appearing in the wallace tree multiplier is still significant.

Table 2 Transition activity for 500 random inputs.

	8x8 array		8x8 wallace tree	
	$\delta_{sum} = \delta_{carry}$	$\delta_{sum} = 2{*}\delta_{carry}$	$\delta_{sum} = \delta_{carry}$	$\delta_{sum} = 2{*}\delta_{carry}$
useful F	23552	23552	38786	38786
useless L	34346	47340	11274	24762
L/F	1.46	2.01	0.29	0.64

4.2 Direction detector architecture

Transition activity simulations were also carried out on a processing unit for Phideo, known as a *direction detector.* A block diagram of a direction detector is shown in Figure 8. The direction detector is used in the implementation of a so-called *progressive scan conversion* algorithm [6].

Random inputs are again a good choice for these transition activity simulations. The original video input signal statistics and correlations are almost completely lost very early in the circuit, immediately after the absolute differences are taken. In the rest of the circuit the signals will be more and more randomly distributed.

The direction detector was simulated using unit delay modelling and 4320 random inputs. The results of the transition activity simulation on the direction detector were as follows:

number of useful transitions:	272842
number of useless transitions:	1033970

ratio useless/useful: 3.79

From this useless/useful ratio it is clear that on average for each useful transition about 3.8 useless transitions occur. Therefore transition activity in the combinational logic of the direction detector can be reduced with a factor of 1 + 3.8 = 4.8 if all delay paths are balanced.

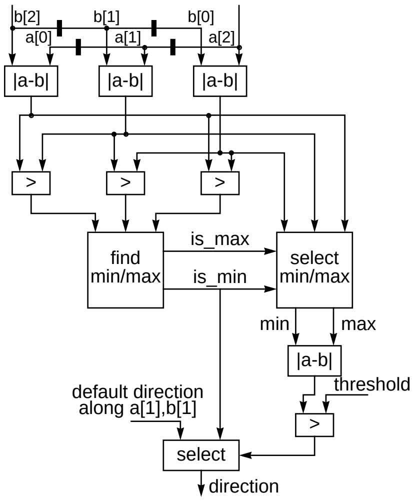

Figure 8 Direction detector.

5 Influence of flipflops

Power dissipation is influenced by transition activity. However, equation 1 shows that loading capacitances of circuit nodes also play an important role. To be able to obtain accurate power dissipation results, circuit level simulations were carried out on extracted netlists of real circuit layouts. The goal of these simulations was to investigate the influence of extra flipflops in the circuit on power dissipation. Flipflops can be used to improve delay balance in a circuit. In this way they help to reduce the amount of glitches and thus power dissipation in combinational logic. Figure 9 shows how the insertion of flipflops by using retiming [7][8] affects the occurrence of glitches in a circuit. The flipflops inserted in the input lines just before the operation node make sure both inputs arrive at the same time. Therefore no glitches will appear at the output of the operation node.

The power dissipation simulation results were divided into three components:

1. dissipation in the combinational logic
2. dissipation in the flipflops
3. dissipation in the clock line

This was done by simulating the circuit with a separate clock power supply and a main power supply feeding the rest of the circuit. Clock power dissipation and main power dissipation were measured separately. Flipflop power dissipation was calculated by multiplying the average power dissipation of a single flipflop, having 50% input transition activity[1], by the total number of flipflops in the circuit. Power dissipation in the combinational logic was then calculated by subtracting the flipflop power from the simulated main power dissipation.

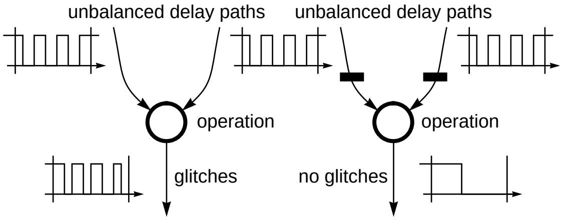

Figure 9 Glitches and retiming.

Four different direction detector layouts were created in 0.8µm, 5V layout technology. Each was retimed for a different clock frequency, resulting in more or less pipeline flipflops in the circuits. Introducing more flipflops increases the delay balance in the circuit and consequently more glitches are eliminated. On the other hand more flipflops result in more flipflop power dissipation and more clock lines, and more clock lines increase the clock capacitance and consequently clock power dissipation.

All circuits were simulated for 20 clock periods using 20 ‘random’ inputs at an equivalent clock frequency of 5MHz. The relatively small number of 20 ‘random’ inputs was used to limit the large amount of simulation time needed for circuit level simulation. However, this limited number of inputs still appeared to give reliable results. The results of the simulations are shown in Table 3 and are graphically displayed in Figure 10.

Table 3 Power dissipation simulation results.

circuit number	**1**	**2**	**3**	**4**
number of flipflops	48	174	218	350
area (mm^2)	0.73	0.99	1.00	1.23
clock capacitance (pF)	3.2	10.5	12.8	19.9
logic power (mW)	21.8	9.7	7.5	6.1
flipflop power (mW)	0.9	3.3	4.1	6.6
clock power (mW)	0.5	1.5	1.8	2.8
total power (mW)	23.2	14.5	13.4	15.5

The power dissipation in the combinational logic reduces as the retiming frequency and consequently the number

[1] It is realistic to assume that on average the input of a flipflop in the circuit is constant for about 50% of the time and changing for the rest of the time under normal circuit operation for random inputs. Using this 50% input transition activity figure the dynamic power dissipation of a single flipflop can be estimated using circuit level simulation.

of flipflops in the circuit increases. A factor of $21.8/6.1 \approx 3.6$ difference in logic power dissipation is found between circuits 1 and 4. This reduction comes from the dramatic decrease in the number of glitches when a large number of pipelining flipflops is inserted in the circuit.

Flipflop power dissipation increases with an increasing number of flipflops. As we stated before, this power dissipation figure was calculated and is linearly dependent on the number of flipflops.

The clock power dissipation is highly dependent on the load capacitance of the clock input. Because extra clock circuitry is necessary when more flipflops are inserted in the circuit, this capacitance will increase. This explains the fact that the clock power dissipation increases with an increasing number of flipflops.

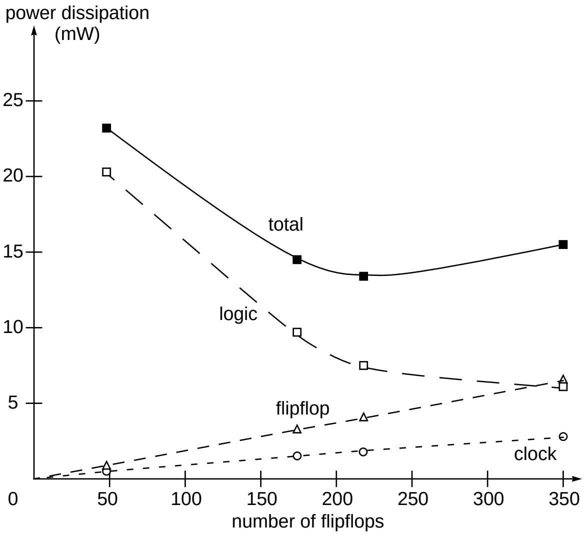

Figure 10 Power dissipation results as a function of the number of flipflops in the circuit.

The total power dissipation plot exhibits a minimum for a certain number of flipflops. Since this number is dependent on the retiming frequency that was used, we find a retiming frequency which is optimal for power dissipation. Suppose the operating point of the original circuit lies to the left of this minimum in Figure 10. Then, retiming the circuit for a higher throughput than necessary, while running it at the original clock frequency will result in lower power dissipation.

On the other hand, consider the case where the operating point of the original circuit lies to the right of the minimum in Figure 10. Then, retiming the circuit for a lower throughput than necessary, lowering the clock frequency accordingly and putting two or more (sub)circuits in parallel may also result in lower power dissipation provided that the power consumed by the necessary extra multiplexing and control circuitry is low enough.

6 Conclusions

Glitches play an important role in excessive useless power dissipation. The ratio of useless transitions to useful transitions can be very large as we showed in our examples of adder, multiplier and processing unit architectures.

A significant reduction in power dissipation can be achieved if the amount of glitches is reduced. This can be done by balancing delay paths and/or by introducing flipflops in the circuit.

Flipflops can be introduced in the circuit by using retiming. By finding the right amount and positions of these pipelining flipflops an optimal total power dissipation can be obtained. This means that an optimum retiming for power dissipation exists.

References

[1] R. Brodersen, A. Chandrakasan and S. Sheng, “Low-Power Signal Processing Systems”, *5th IEEE Workshop on VLSI Signal Processing*, California, USA, October 1992.

[2] T.K. Callaway and E.E. Swartzlander, Jr., “Optimizing Arithmetic Elements for Signal Processing”, *5th IEEE Workshop on VLSI Signal Processing*, California, USA, October 1992.

[3] M.D. Ercegovac and T. Lang, “Reducing Transition Counts in Arithmetic Circuits”, *Digest of Technical Papers IEEE Symposium on Low Power Electronics*, San Diego, USA, October 1994.

[4] Uming Ko, P.T Balsara and Wai Lee, “A Self-timed Method to Minimize Spurious Transitions in Low Power CMOS Circuits”, *Digest of Technical Papers IEEE Symposium on Low Power Electronics*, San Diego, USA, October 1994.

[5] J. Leijten, J. van Meerbergen and J. Jess, “Analysis of Transition Activity and Power Dissipation in Synchronous Logic Circuits”, *Internal Report Eindhoven University of Technology*, Eindhoven, The Netherlands, December 1993.

[6] J. van Meerbergen, et al., “Architectural Strategies for High-Throughput Applications”, *Journal of VLSI Signal Processing*, vol. 5, 1993, p. 201-220.

[7] A. van der Werf, et al., “Efficient Timing Constraint Derivation for Optimally Retiming High Speed Processing Units”, *Proceedings of the 7th International Symposium on High-Level Synthesis*, Ontario, Canada, May 1994.

[8] OPTIMA, *Philips Electronic Design & Tools*, Hilversum, The Netherlands, 1992.