

Research Article

Architectures and Arithmetic for Low Static Power Consumption in Nanoscale CMOS

Peter Nilsson

Department of Electrical and Information Technology, Faculty of Engineering, Lund University, Box 118, 22100 Lund, Sweden

Correspondence should be addressed to Peter Nilsson, peter.nilsson@eit.lth.se

Received 9 July 2009; Revised 23 September 2009; Accepted 9 October 2009

Recommended by Israel Koren

This paper focuses on leakage reduction at architecture and arithmetic level. A methodology for considerable reduction of the static power consumption is shown. Simulations are done in a typical 130 nm CMOS technology. Based on the simulation results, the static power consumption is estimated and compared for different filter architectures. Substantial power reductions are shown in both FIR-filters and IIR-filters. Three different types of architectures, namely, bit-parallel, digit-serial, and bit-serial structures are used to demonstrate the methodology. The paper also shows that the relative power ratio is strongly dependent on the used word length; that is, the gain in power ratio is larger for longer word lengths. A static power ratio at 0.48 is shown for the bit-serial FIR-filter and a power ratio at 0.11 is shown in the arithmetic part of the FIR-filter. The static power ratio in the IIR-filter is 0.36 in the bit-serial filter and 0.06 in the arithmetic part of the filter. It is also shown that the use of storage, such as registers, relatively the arithmetic part, affects the power ratio. The relatively lower power consumption in the IIR-filter compared to the FIR-filter is due to the lower use of registers.

Copyright © 2009 Peter Nilsson. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. Introduction

The power consumption is becoming a major obstacle in future circuit design. Referring to Moore's law, by adding more functionality in an exponential way, we will also increase the power consumption in the same pace. From the power consumption perspective, the dynamic power consumption has been the major concern, for a long time, when integrating digital CMOS circuits.

However, in today's technologies it can be noticed that the static power consumption is an important factor for the total power consumption. In fact, in ITRS predictions [1] the static power will dominate below the 65 nm technology. A number of methods to limit the static power consumption on device level as well as logic level have been shown in the literature. Examples are dual threshold design, respectively, self reverse biasing of transistor stacks [2]. However, to reduce the total power consumption, all abstraction levels must be considered. That is, system, algorithm, architecture, arithmetic, as well as logic and device level are all important to reduce the total power consumption. The focus for this

paper is mainly static but also dynamic power reduction at arithmetic and architecture level, an area where relatively few papers are published. Results on architectural and arithmetic level are presented in [3, 4]. Especially, [3] shows how to trade off circuitry towards transitions for an optimum power consumption in bit-parallel architectures.

This paper focuses on architectures using bit-serial, digit-serial, and bit-parallel arithmetic in order to make the best choice in reducing the static power consumption, in regions where the dynamic power consumption can be neglected, that is, in low data rate applications. Such applications can be wireless and battery operated medical applications or sensor network applications.

The idea of bit-serial arithmetic was invented long time ago. An early paper that describes the arithmetic is [5]. Another publication related to information technology is [6]. More recent, bit-serial architectures in the area of filters and signal processing related to VLSI can be found in [7]. Digit-serial arithmetic is often an alternative to bit-serial arithmetic. The arithmetic is based on the same idea as bit-serial arithmetic; compare Figure 1(b) and 1(c). One good

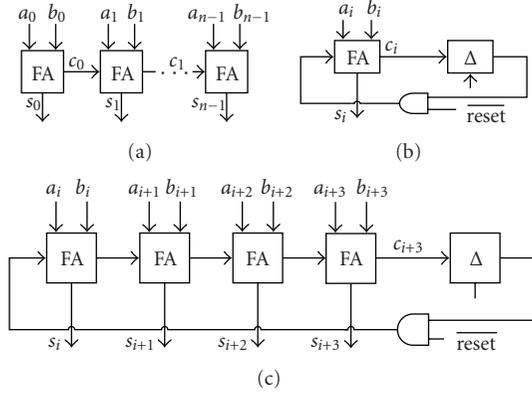


FIGURE 1: A bit-parallel ripple-carry adder in (a), a bit-serial adder in (b), and a 4-bit digit-serial adder in (c).

reason to choose digit-serial arithmetic is that the needed clock frequency is lower for the same throughput. In [8–10] there are various aspects on digit-serial architectures.

This paper is organized as follows. Section 2 gives a short background to static power consumption and Section 3 presents the methodology for arithmetic power reduction of the static power consumption. Section 4 describes the circuitry used in the simulations. In Section 5 the simulation results are presented, which is the base for Section 6, where the results are applied on FIR-filter structures as well as in Section 7 where the methodology is applied on IIR-filters.

2. Background

The transistors leakage is the main source for the static power consumption, which consists of many fractions particularly in technologies below 50 nm [2]. Examples are gate oxide tunneling, junction band-to-band-tunneling (BTBT), drain induced barrier lowering (DIBL), and the gate induced drain leakage (GIDL) [2]:

$$I_{\text{off}} = I_0 \times e^{(V_{\text{GS}} - V_T)/(m \times V_T)} \times (1 - e^{-V_{\text{DS}}/V_T}), \quad (1)$$

$$I_0 = \mu_0 \times C_{\text{OX}} \times \frac{W}{L} \times v_T^2 \times (m - 1), \quad v_T = \frac{K \times T}{q}.$$

However, in present technology generations, for example, 65 nm, the main contributor to the leakage is the subthreshold current, which often is described by (1) [2].

From the equation, it can be seen that the leakage is dependent on the transistor width W . The total static power is thus dependent on the amount of transistors on the chip; that is, it increases with the number of arithmetic blocks that are used. A good concept for reducing the static power would therefore be to use arithmetic that requires a low number of arithmetic blocks. Bit-serial arithmetic is an arithmetic with very few arithmetic blocks, which instead are reused many times.

3. Arithmetic for Static Power Reduction

Figure 1(a) illustrates the concept of bit-parallel arithmetic, using a carry ripple adder. In the parallel adder all bits, a_i and b_i , are available to the adder at the same time. To get the bits s_i in the output sum the carry, c_i , has to ripple through all the adder cells in the adder.

Figure 1(b) shows the corresponding bit-serial adder [11]. The signals, a_i and b_i , are provided serially, with the LSB bits first, to form the corresponding output sum. The output carry c_i is delayed one clock cycle to be added with next higher significant input bits. In Figure 1(c), 1(a) 4-bit digit-serial adder is shown as an example. In this adder the word that should be processed is divided in digits. It basically has the same function as the bit-serial adder but here every fourth carry is stored.

3.1. Conceptual Discussion. In general, bit-parallel arithmetic uses n one-bit arithmetic units, adder cells, to process n -bit words, where bit-serial arithmetic only requires a single one-bit arithmetic unit to process n -bit words. The reduction in number of arithmetic units is thus traded off to a higher number of clock cycles.

Bit-serial arithmetic shows essentially the same properties as the bit-parallel, for the dynamic power consumption, which is illustrated by the left term in (2) and (3):

$$P_{\text{bit-par}} = f \times n \cdot C_{\text{add}} \times V_{\text{DD}}^2 + n \cdot I_{\text{add}} \times V_{\text{DD}}, \quad (2)$$

$$P_{\text{bit-ser}} = n \cdot f \times C_{\text{add}} \times V_{\text{DD}}^2 + I_{\text{add}} \times V_{\text{DD}}. \quad (3)$$

In the left term of (2), n one-bit units are used once, but in (3), only one single unit is used n times. The clock frequency is thus increased n times, for the same throughput. Both circuits will thus charge and discharge the same amount of capacitance per addition. However, it shall be noted that (2) and (3) are conceptual. The bit-parallel arithmetic often suffers from glitching which increases the dynamic power, sometimes substantially. In the case of bit-serial arithmetic, the register and the AND gate are neglected for the moment. However, it will be taken into account from Section 4 and forward.

The big difference is when comparing the static power consumption in the two adders. The static power consumption is dependent on the amount of transistors as shown in (1), that is, the number of adder cells in this case. As shown in the right term of (2), the bit-parallel adder dissipates n equivalent currents, I_{add} , whereas the bit-serial arithmetic that only uses one single unit, only dissipates one equivalent current, as shown in (3). The bit-serial arithmetic has thus the potential to have n times lower static power consumption; for example, a 12-bit combinatorial bit-parallel datapath will have a static power consumption that is 12 times higher than the bit-serial datapath.

As shown in (2) and (3) the bit-serial arithmetic needs a clock frequency at bit-level for the same throughput as the bit-parallel. However, that does not mean that the bit-serial adder cannot reach the same maximum throughput. The carry has to go through the same number of adder cells in both cases which means that the total delay for an adder

TABLE 1: Transistor widths in the simulated circuitry.

W in nm	1-stacked	2-stacked	3-stacked
n -channel	280	560	840
p -channel	560	1120	1680

TABLE 2: The simulated static power consumption.

FA (nW)	Reg. (nW)	AND (nW)	HA (nW)	Inv. (nW)
$P_{FA} = 11.2$	$P_{reg} = 7.1$	$P_{and} = 1.9$	$P_{HA} = 3.7$	$P_{inv} = 0.9$

operation will be about the same. In the used cells described later in Section 5, the bit-serial addition has a simulated maximum throughput that is about half the bit-parallel. The highest throughput for the two adders is thus comparable.

If the bit-serial arithmetic still gives an inconveniently high clock frequency, an alternative is digit-serial arithmetic, where the word is divided in digits instead of single bits [12, 13]; see Figure 1(c). The digit-serial arithmetic is thus a trade-off; it gives more leakage than the bit-serial but less than the bit-parallel.

4. Circuitry

A Cadence environment with Spectre as the circuit simulator has been used for the circuit simulations. Five cells are designed, a full-adder (FA), a half-adder (HA), a register cell, an AND-gate, and an inverter cell. The full adder and the AND-gate will be used in the simulations in Section 5 and the three others will also be used later in Sections 6 and 7. The same adder cell is used in the bit-parallel and the bit-serial design. A mirror-adder cell, containing 28 transistors, is used for the full adders; see [14, page 567]. A transmission-gate register cell, using 16 transistors is used for the registers; see [14, page 335].

All transistor lengths are minimum sized, that is, 120 nm. The used transistor widths, W , in the cells are shown in Table 1. The widths are made wider depending on the number of stacked transistors. Furthermore, due to the difference in mobility between n -channel and p -channel transistors, the p -type transistors are made wider. The widths are multiplied with the number of serial transistors stacked on top of each other. In addition, the minimum width $W = 280$ nm is doubled if it is a p -type transistor. Table 2 shows the simulated static power consumption in all cells, when a supply voltage at 1.2 V is used.

5. Simulation Results

A typical 130 nm transistor model, provided by UMC, has been used in the simulations. Both bit-parallel and bit-serial adders have been simulated to show the difference in power consumption. Two different cases with the word lengths 12 and 24 bits are used as examples to illustrate the total power consumption. The simulations also show that the dynamic and static power consumption can be extracted separately

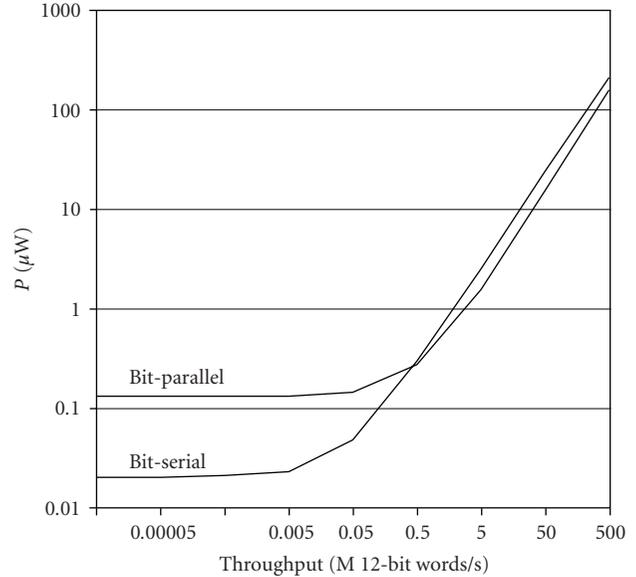


FIGURE 2: Bit-serial versus bit-parallel power consumption for addition of 12-bit words.

from the simulations. The simulations are performed at maximal switching activity, in all cases; however, without glitching, which often occurs in bit-parallel structures.

Figure 2 shows the power consumption for 12-bit additions in logarithmic scale for both bit-parallel and bit-serial adders. The figure shows the total power consumption for throughputs up to 500 MWords/s.

To the right, at high throughputs, the diagram shows a linear increase in power consumption. The power consumption is thus dominated by the dynamic power consumption, which has a linear dependence to frequency, as expected from (2) and (3).

In Figure 2, the total power consumption dominated by the static power consumption is shown to the left in the diagram. The flat part of the diagram shows that the power is more or less constant; that is, the power consumption is dominated by the static power consumption and hence the dynamic power consumption can be neglected.

The power consumption for 24-bit additions is, in the same manner, shown in Figure 3. The total power consumption for throughputs up to 250 MWords/s is shown in the figure. Also in this case, the static and dynamic power consumption can be seen separately in the two regions.

Another good measure is to describe the energy per switching event, that is, the power delay products (PDP), defined by $P \times t_p$, where P is the power consumption and t_p is the propagation delay. The PDP is an invariable measure, when the dynamic power consumption dominates, since the PDP is proportional to the power consumption and to the inverse of the throughput. In Table 3, the PDP is shown for higher throughputs. It can be seen that the PDP is about double for the bit-serial arithmetic in that region and it can also be seen that it is about two times higher when the word length is doubled.

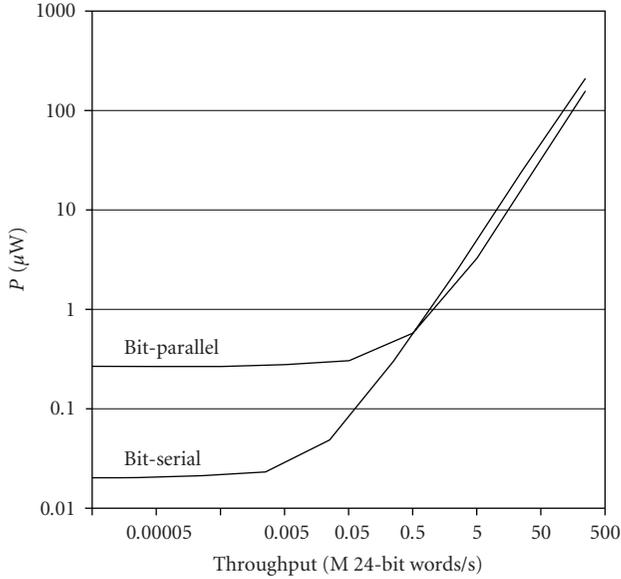


FIGURE 3: Bit-serial versus bit-parallel power consumption for addition of 24-bit words.

TABLE 3: The power delay product.

12-bit serial	12-bit parallel	24-bit serial	24-bit parallel
278 fJ	140 fJ	576 fJ	302 fJ

For the lower throughputs, where the static power consumption dominates, the PDP decreases with increasing throughput. The reason is that the power consumption is constant in this case but not the throughput. The difference in PDP is thus not that illustrated in that region.

When examining the two diagrams, in Figures 2 and 3, some important observations can be made.

5.1. Higher Throughputs. It can be noted that the power consumption for the bit-serial arithmetic is slightly higher than that for the bit-parallel, for higher throughputs, 1.95 and 1.91 times for the 12-bit and the 24-bit additions, respectively. This higher dynamic power consumption is caused by the extra circuitry in form of one register cell and one AND-gate, in the bit-serial case. Hence, the conceptual discussion from (2) and (3) is a too positive estimate, but it is still reasonable assumption.

5.2. Lower Throughputs. For lower throughputs, 500 kWords/s and less, the static power consumption starts to dominate. It can be noted that the bit-parallel power consumption is much higher than the bit-serial. For the 12-bit case, the static power consumption is simulated to 20.3 nW and 134 nW for the bit-serial and bit-parallel adders, respectively. The bit-parallel power consumption is thus $134/20.3 = 6.6$ times higher. In the 24-bit additions the corresponding power consumption is 13.4 times higher for bit-parallel additions. An important observation is thus that it becomes more and more attractive to use bit-serial

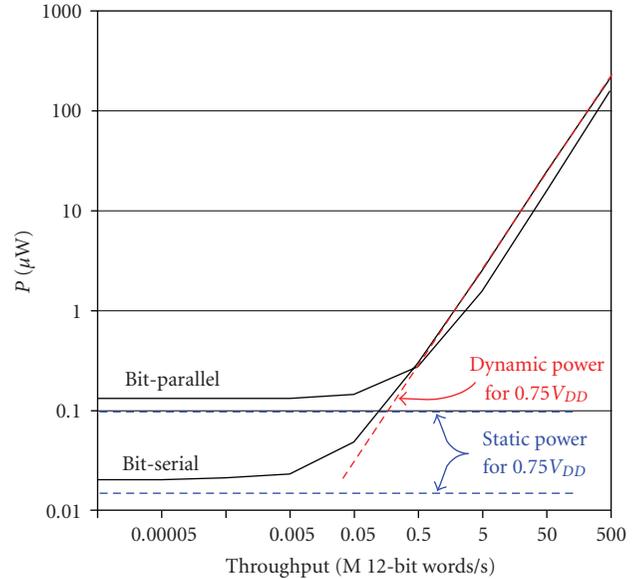


FIGURE 4: Effects on lowering the supply voltage for bit-serial versus bit-parallel architectures for addition of 12-bit words.

arithmetic when the word length increases. It can also be noted that the power reduction is not so high as the expectations from (2) and (3). The reason is, again, that the register cell and the AND-gate were neglected in that section. However, the power reduction is still very high.

5.3. The Crossover. There is a crossover in the middle, where the dynamic and static power consumption becomes equal. There are two aspects on where this crossover will appear.

5.3.1. Switching Activity. The simulations are done with maximum switching activity. In a general application, the switching activity is often much lower, for example, down to 10% or less [4]. The crossover will in that case move up to higher throughputs, at about 1-2 orders of magnitude. It will thus be more and more important to design for low static power consumption.

5.3.2. Technology. The simulations are done for a typical 130 nm technology. However, the technology choice is not important for the results presented in this paper. Denser technologies will show the same behavior with one flat part, one increasing linear part, and a crossover in between. The difference is that the crossover will move up in throughput since more dense technologies will have more and more leakage. In [1], the dynamic and static power consumption is predicted to be equal in the 65 nm node. The static power consumption will thus become more and more important.

5.4. Discussion on Lowering the Supply Voltage. This paper suggests a methodology to reduce the static power consumption on the arithmetic and architectural level. The focus is on low and medium throughputs since it is at those data rates where the static power consumption dominates.

However, the dominant methodology for reducing the power consumption, the last decades, has been to lowering the supply voltage, V_{DD} . Often the designer does not have the possibility to choose the supply voltage [3]. Fixed supply voltages and threshold voltages are often assigned at the overall system level and in those cases the discussion in Sections 5.3.1 and 5.3.2 is valid. However, for the case where the supply voltage can be chosen freely it is important to examine how lowering the supply affects the methodology, proposed in this paper.

5.4.1. Higher Throughputs. As described in Section 5.1, the power consumption is about 2 times higher for the bit-serial architectures, in the high-throughput region. The simulations also show that the maximum throughput is about 2 times higher for bit-parallel architecture. By lowering the bit-parallel supply voltage to approximately $0.75 V_{DD}$, the dynamic power consumption will be equal to the bit-serial, at the same throughput. This is illustrated in Figure 4, where we can see the extrapolated dynamic power consumption for the bit-parallel adder when it is on top of the bit-serial curve; see the “diagonal” dotted line. The maximum throughputs will be comparable as well. However, the obvious choice, in the region where the dynamic power consumption dominates, is to use bit-parallel arithmetic, since it in general is an advantage to run at lower supply voltages.

5.4.2. Lower Throughputs. As described in Section 5.2, the static power consumption dominates in this region. With the same operation, lowering the bit-parallel supply voltage to approximately $0.75 V_{DD}$, the static power consumption will decrease to about $0.75 P$, according to (1). The static power consumption will thus decrease with reduced supply voltage. This is also illustrated in Figure 4, where we can see the extrapolated static power consumption for the bit-parallel adder as the upper dotted horizontal line. The effect of lowering the bit-parallel supply voltage is thus that the relative gain in reduced static power consumption is lower. However, in the lower throughput region, the same operation can be applied to the bit-serial arithmetic as well. By lowering the supply for the bit-serial adder, the static power consumption will decrease in the same pace as well. This is shown in Figure 4 as well, where we can see the extrapolated static power consumption for the bit-serial adder as the lower dotted horizontal line. For lower throughputs, the relative difference in static power consumption will thus be kept, if the supply voltage is reduced by the same amount. The supply voltage reduction can continue down to voltages where the reliability, the noise margin, sets the limit for a proper function, more or less without changing the power ratio between the bit-parallel and the bit-serial arithmetic.

6. The Concept Applied On a Hilbert Transformer

The simulation results from Sections 4 and 5 can be used to estimate the static power consumption on larger structures. An 11-tap linear phase FIR-filter, a Hilbert transformer

TABLE 4: The filter coefficients.

Coefficient	Decimal value	Binary value
$C_{10} = -C_0$	0.203125 ₁₀	00001101 ₂
$C_8 = -C_2$	0.328125 ₁₀	00010101 ₂
$C_6 = -C_4$	1.0 ₁₀	01000000 ₂

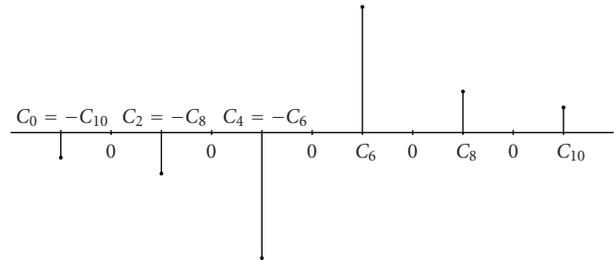


FIGURE 5: The impulse response for the Hilbert transformer.

[7], is, in this section, used as an example to show the methodology of reducing the static power consumption. In Section 7, the methodology will be shown on an IIR-filter as well.

One characteristic for Hilbert filters is that every second coefficient is zero. They are also antisymmetric; that is, the same coefficient appears two times, where one is positive and the other one is negative. The impulse response for the Hilbert transformer is shown in Figure 5.

In Figure 6, the linear-phase FIR-filter corresponding to the impulse response in Figure 5 is shown. In the figure, two samples are subtracted before each coefficient multiplication to gain one multiplication, which is illustrated with the signal S in Figure 6. Six of the used 8-bit coefficient values are given in Table 4. All other coefficients are zero.

Three different filter architectures will be used to show the power reduction methodology, one bit-parallel, one digit-serial, and one bit-serial.

6.1. The Bit-Parallel Filter Architecture. The filter has two fixed coefficient multiplications, that is, the $C_{10}(X_{n+10} - X_n)$ and $C_8(X_{n+8} - X_{n+2})$ signals. The multiplication $C_6(X_{n+6} - X_{n+4})$ is a multiplication with “1”, which thus is trivial. Figure 7 shows a 12-bit fixed multiplier for the C_{10} coefficient. The bits for $A_{10} = X_{n+10} - X_n$ are denoted by a_i and the result by s_i in the figure. Note that the upper-case letters denote word-level data and lower-case letters represent bit-level data.

The coefficient $C_{10} = 00001101$ contains three “1 : s” at the positions c_3 , c_2 , and c_0 . The multiplier can be realized by using two adders, where the upper adder is used for adding A_{10} for c_0 together with the two-step left-shifted A_{10} for c_2 . A three-step left-shifted A_{10} is, after that, added to the result in the lower adder. Figure 8 shows the corresponding methodology to realize the multiplier for the coefficient $C_8 = 00010101$.

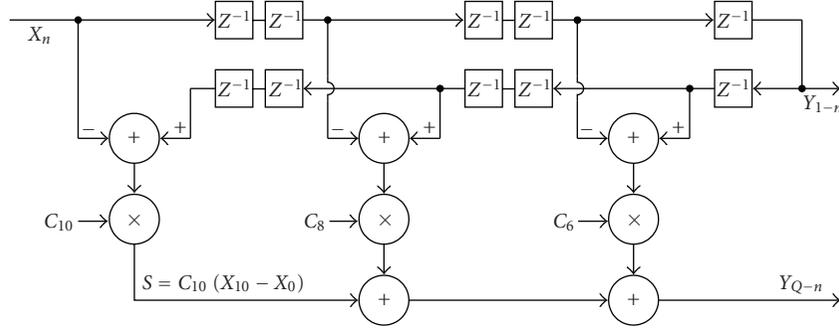


FIGURE 6: An 11-tap linear-phase Hilbert transformer.

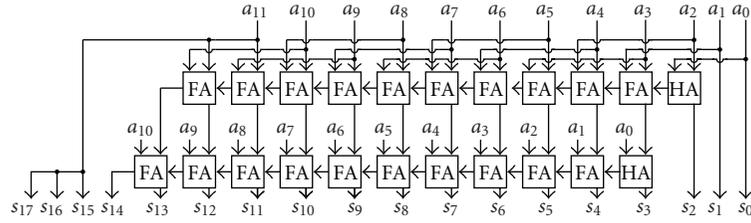


FIGURE 7: The 12-bit fixed coefficient C10 bit-parallel multiplier.

Both the C_{10} and the C_8 multipliers contain 20 full adders (FAs) and 2 half adders (HAs) each if the 12-bit case is considered. Furthermore, two adders containing 26 FAs and 2 HAs are needed to sum up the results from the multiplications, see Figure 6. In addition, 3 subtractors with 36 FAs are used. The subtractors also need 36 inverters for the two's complement conversion. The arithmetic in the parallel filter will contain 102 FAs, 6 HAs, and 36 inverters, in total.

6.2. The Bit-Serial Filter Architecture. The arithmetic part of the bit-serial filter is shown in Figure 9. To the left, the three bit-serial subtractors are shown. In the middle, there are three multipliers, for the C_{10} , C_8 , and C_6 coefficients. Finally, the two adders that sum up the result are placed to the right.

Counting the leaf cells, the bit-serial arithmetic gives 9 FAs, 29 register cells, 9 AND-gates, and 3 inverters, in total. The architecture is independent of the word length. However, more clock cycles are needed to process more bits.

6.3. Digit-Serial Filter Architecture. Figure 10 shows a 2-bit digit-serial architecture. The structure is similar to the structure in Figure 9. Accordingly, the flow through the architecture can be compared to the flow through the bit-serial architecture but here are two bits processed each clock cycle. The digit-serial arithmetic contains 18 FAs, 49 register cells, 9 AND-gates, and 3 inverters.

The cell count for the 12-bit structures is shown in Table 5, divided on the different cells. Note that the AND-gates are only needed together with the registers that are used in the bit- and digit-serial adders; see Figure 1.

From Table 5 and the simulated power consumption in Table 2, we can estimate the power consumption in the filter

TABLE 5: The number of cells in the three 12-bit FIR-filters designs.

Cell	k_{FA}	k_{HA}	k_{Reg}	k_{AND}	k_{Inv}
Bit-Serial	9	0	29	9	3
2-Bit Digit-Serial	18	0	49	9	3
Parallel	102	6	0	0	36

arithmetic. The power P_{stat} in (4) shows the total sum of the currents in the adders, registers, and the AND-gates:

$$P_{stat} = k_{FA} \times P_{FA} + k_{HA} \times P_{HA} + k_{Reg} \times P_{Reg} + k_{AND} \times P_{AND} + k_{Inv} \times P_{Inv}. \quad (4)$$

The following two subsections will discuss the relative power reduction for the arithmetic part of the FIR-filter, in Section 6.4 and for the complete FIR-filters, in Section 6.5.

6.4. The FIR-Filter Arithmetic. The power consumption in the arithmetic can thus be estimated separately by excluding the n -bit registers, as shown in (5):

$$\begin{aligned} P_{Ser} &= 9 \times 11.2 + 29 \times 7.1 + 9 \times 1.9 + 3 \times 0.9 = 326 \text{ nW}, \\ P_{Dig} &= 18 \times 11.2 + 49 \times 7.1 + 9 \times 1.9 + 3 \times 0.9 = 569 \text{ nW}, \\ P_{Par} &= 102 \times 11.2 + 6 \times 3.7 + 36 \times 0.9 = 1165 \text{ nW}. \end{aligned} \quad (5)$$

Table 6 shows a substantial power reduction for the 12-bit arithmetic in the filter, where the bit-serial arithmetic only consumes 28% of the static power compared to the bit-parallel consumption. The 2-bit digit-serial arithmetic only dissipates about half the static power, compared to the bit-parallel.

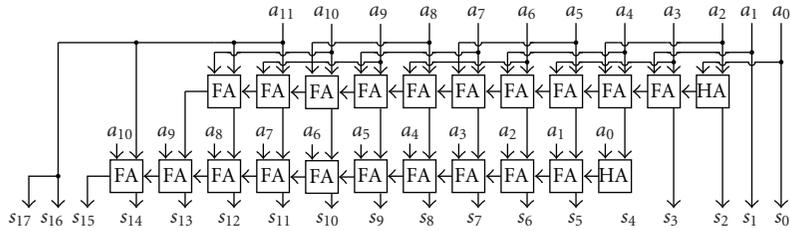


FIGURE 8: The 12-bit fixed coefficient C8 bit-parallel multiplier.

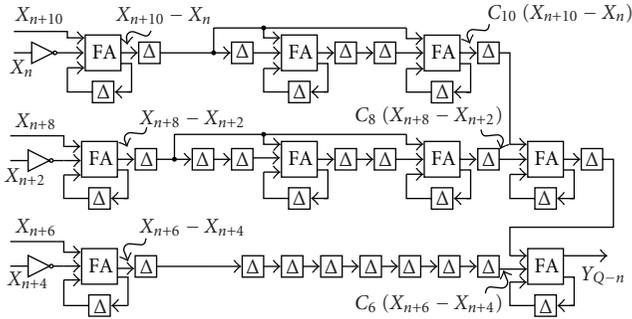


FIGURE 9: The arithmetic in the bit-serial architecture.

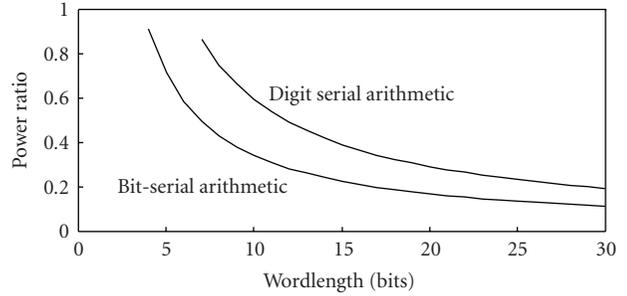


FIGURE 11: Static power ratio for the arithmetic part in the bit-serial and the 2-bit digit-serial filters, respectively.

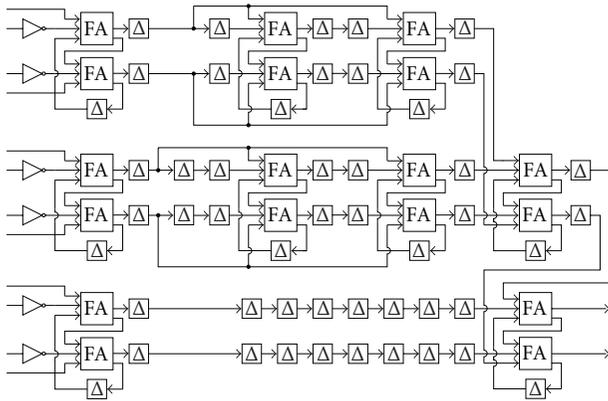


FIGURE 10: The 2-bit digit-serial arithmetic.

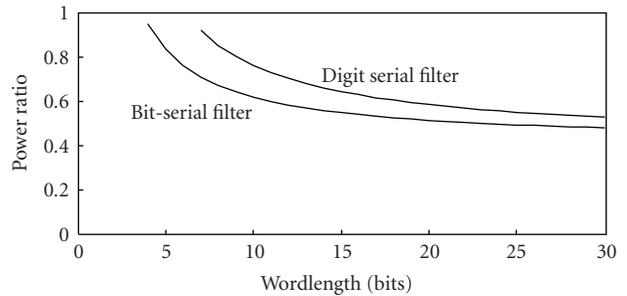


FIGURE 12: Static power ratio for the bit-serial as well as the 2-bit digit-serial arithmetic, in the complete filters.

TABLE 6: The static power ratio in the 12-bit FIR-filter arithmetic.

Bit-Serial Arithmetic	$326/1165 = 0.28$
2-Bit Digit-Serial Arithmetic	$569/1165 = 0.49$

There is a strong dependence on the word length as well. As the word length increases, the relative power reduction will be larger, which could be expected from (2) and (3). The word length dependence is shown in Figure 11, where the word length is varied from 4 bits up to 30 bits.

The lower graph shows the bit-serial arithmetic and the upper shows the digit-serial. The bit-serial arithmetic only consumes 11% and the digit-serial only 19% of the static power consumption compared to the bit-parallel arithmetic, when the word lengths are 30 bits long.

6.5. *The Complete Filter.* Based on (4), we get the static power consumption for the complete filter as shown in (6). It can be noted that the static power in the bit-serial and the digit-serial arithmetic, in (5), is independent of the word length. However, that is not the case when comparing the power for the complete filter, since there are ten n -bit registers, which vary with the word length, as shown in (6) for the 12-bit case:

$$\begin{aligned}
 P_{\text{Filter-Ser}} &= P_{\text{Ser}} + 10 \times 12 \times 7.1 = 1178 \text{ nW}, \\
 P_{\text{Filter-Dig}} &= P_{\text{Dig}} + 10 \times 12 \times 7.1 = 1421 \text{ nW}, \\
 P_{\text{Filter-Par}} &= P_{\text{Par}} + 10 \times 12 \times 7.1 = 2017 \text{ nW}.
 \end{aligned}
 \tag{6}$$

In Table 7, the filters are compared. The table shows a large reduction of the static power consumption as well, even if it is not as large as when the arithmetic part is examined alone. The reason is that the ten n -bit registers do have the same size in both the parallel and the serial cases; see

TABLE 7: Static power ratio in the complete 12-bit filters.

Bit-Serial Filter	1178/2017 = 0.58
2-Bit Digit-Serial Filter	1421/2017 = 0.70

TABLE 8: The number of cells in the filter designs.

Cell	k_{FA}	k_{Reg}	k_{AND}
12-bit Serial	4	40	4
12-bit Parallel	48	36	0
24-bit Serial	4	76	4
24-bit Parallel	96	72	0

Figure 6. There is thus no relative power gain in the n -bit registers, only in the arithmetic.

In Figure 12, the relative power reduction for the complete filter is shown. A power ratio at 0.48 and 0.52 can be noted, respectively. The difference at higher word lengths is thus not so high. The digit-serial filter can thus be worth to consider in order to reduce the clock frequency.

7. The Methodology Applied on a Half-Band Filter

A third-order IIR-filter structure is here used to show the static power reduction methodology. The filter structure, shown in Figure 13, is a third-order bireciprocal lattice wave digital filter [7, 15] also called a half-band filter.

The recursive structure has four adders and three registers. There is also one multiplication with the coefficient 0.5, which corresponds to a shift. By using bit-serial arithmetic, a substantial reduction of the static power consumption can be achieved.

The same number and size of the registers is needed in both the bit-parallel and the bit-serial registers, since the words have to be stored in three parallel registers or fed through three equally sized serial registers. Beside the registers, four single register cells have to be added in the bit-serial case, one for each adder; see Figure 1. The number of adder cells is however very different. In the bit-parallel case the adders have full width, that is, the same as the word length n , but in the bit-serial case only four adder cells are needed.

The cell count for both the 12-bit and the 24-bit structures is shown in Table 8, divided by the number of FAs k_{FA} , register cells k_{Reg} , and AND-cells k_{AND} . Note that the AND-gates are only needed together with the registers that are used in the bit-serial adders, as shown in Figure 1.

From Table 8 and the simulated power consumption in Table 2, we can estimate the static power consumption in the filter arithmetic, which is shown in (8) and the power consumption in the complete filter in (9):

$$P_{stat} = k_{FA} \times P_{FA} + k_{Reg} \times P_{Reg} + k_{AND} \times P_{AND}. \quad (7)$$

TABLE 9: Static power ratio in the arithmetic part of the IIR-filter.

12-bit filter arithmetic	81/538 = 0.15
24-bit filter arithmetic	81/1075 = 0.075

TABLE 10: Static power ratio in the filters.

12-bit Filter	336/793 = 0.42
24-bit Filter	592/1586 = 0.37

The power P_{stat} in (7) shows the total sum of the different power terms in the adders, registers cells, and the AND-gates. The power, P_{stat} , is presented for each case in (8) and (9):

$$\begin{aligned} P_{Ser-12} &= 4 \times (11.2 + 7.1 + 1.9) = 81 \text{ nW}, \\ P_{Par-12} &= 48 \times 11.2 = 538 \text{ nW}, \\ P_{Ser-24} &= 4 \times (11.2 + 7.1 + 1.9) = 81 \text{ nW}, \\ P_{Par-24} &= 96 \times 11.2 = 1075 \text{ nW}. \end{aligned} \quad (8)$$

The following two subsections discuss the relative power reduction for the filter arithmetic only, in Section 7.1 and for the complete filter, in Section 7.2.

7.1. The Filter Arithmetic. The power consumption in the arithmetic can thus be estimated separately by excluding the n -bit registers, as shown in (8).

Table 9 shows a substantial power reduction for the arithmetic in the filter, especially for the 24-bit case where the bit-serial arithmetic only consumes 7.5% of the static power compared to the bit-parallel consumption. The table also shows that there is a strong dependence on the word length. As the word length increases, the relative power reduction will be larger, which could be expected from (2) and (3).

Figure 14 shows the relative power reduction up to a word length of 30 bits. Based on the same discussion as in Section 6, the power ratio for digit-serial arithmetic is added as well. The diagram shows that the power ratio goes down to 0.06 and 0.16 for the bit-serial and the digit-serial arithmetic, respectively, compared to the bit-parallel arithmetic.

7.2. The Total Filter. Based on (7), we get the static power for the complete filter shown in (9):

$$\begin{aligned} P_{Filter-Ser-12} &= 4 \times 11.2 + 40 \times 7.1 + 4 \times 1.9 = 336 \text{ nW}, \\ P_{Filter-Par-12} &= 48 \times 11.2 + 36 \times 7.1 = 793 \text{ nW}, \\ P_{Filt-Ser-24} &= 4 \times 11.2 + 76 \times 7.1 + 4 \times 1.9 = 592 \text{ nW}, \\ P_{Filt-Par-24} &= 96 \times 11.2 + 72 \times 7.1 = 1586 \text{ nW}. \end{aligned} \quad (9)$$

It can be noted that the static power consumption in the serial arithmetic, P_{Ser-12} and P_{Ser-24} in (5), is independent of the word length, for this filter as well. However as for the FIR-filter, that is not the case when comparing the power reduction for the complete filter, since the n -bit registers vary with the word length; see $P_{Filter-Ser-12}$ and $P_{Filter-Ser-24}$ in (9).

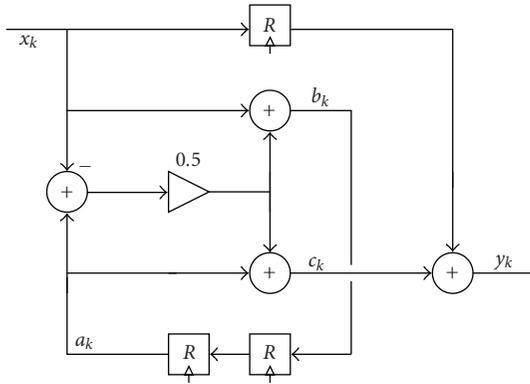


FIGURE 13: A third-order half-band filter.

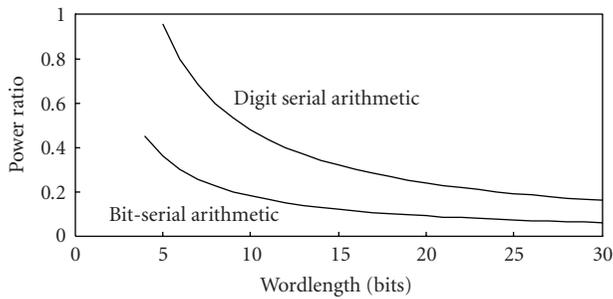


FIGURE 14: Static power ratio for the arithmetic part of the bit-serial as well as the 2-bit digit-serial arithmetic.

In Table 10, the filters are compared. The table shows a large reduction of the static power consumption as well, even if it is not as large as when we are examining the arithmetic part alone. The reason is that the three n -bit registers; see Figure 13, do have the same size in both the bit-serial and the bit-parallel case. There is thus no relative power gain in the registers, only in the arithmetic. However, the resulting gain is still remarkably high, with a reduction down to 37%.

Figure 15 shows a similar diagram as in Figure 14, but for the complete IIR-filter. The power ratio for the complete filter goes down to 0.36 and 0.43, respectively, for the bit-serial and digit-serial filter compared to the bit-parallel filter.

Figures 14 and 15 show the power ratio for the bit-serial and the 2-bit digit-serial architectures. In Figure 16, the power ratio versus the digit-serial bit size is shown for bit sizes up to 24-bits in the 24-bit filter structure. It can be seen that the power ratio increases linearly up to the bit-size 23-bits. After that the digit-serial filter will be equivalent to the bit-parallel filter with the power ratio 1.0. The left part of the diagram when the digit bit-size is 1 corresponds to the bit-serial case.

8. Discussion

It has been shown that a good architectural methodology to reduce the static power consumption is to choose arithmetic that has a lower number of processing elements. Bit-serial and digit-serial arithmetic can be very useful to reduce the

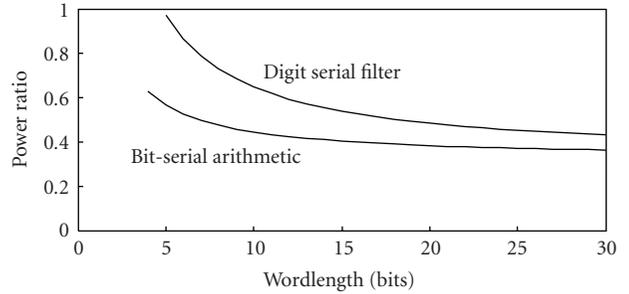


FIGURE 15: Static power ratio for the bit-serial as well as the 2-bit digit-serial arithmetic, in the complete filters.

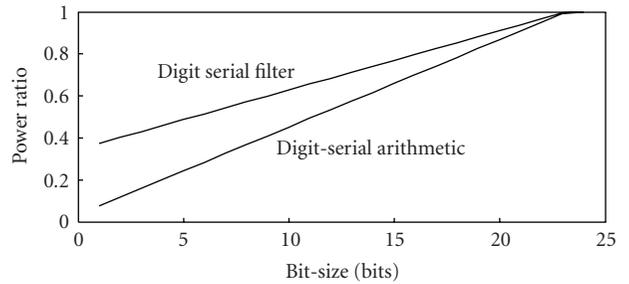


FIGURE 16: Static power ratio for the digit-serial structures when the bit-size is varied from 1 to 24 in the 24 bit filter.

number of units in a VLSI design, which thus also reduces the static power consumption. The methodology is especially useful for low and medium data rates where the static power consumption dominates.

The methodology has been tested on two different filter structures, one FIR-filter and one IIR-filter. Beside the power reduction, two other conclusions can be drawn.

- (1) The proposed methodology gives a higher power ratio on architectures with long word lengths. The reason is that the number of cells in bit-parallel arithmetic increases linearly with the word length, whereas the number of cells in bit-serial and digit-serial arithmetic is constant for all word lengths.
- (2) A good strategy when considering algorithms and architectures is to choose those, which have a minimum of storage elements, such as registers. The size of the registers follows the word lengths and is thus independent if bit-parallel, bit-serial, or digit-serial arithmetic is used. This has been shown by the two filters. In this particular FIR-filter the ratio between the registers and the arithmetic is rather high. A power ratio is in that case relatively high, that is, down to 0.48. On the other hand, in the used IIR-filter, the ratio between the registers and the arithmetic is rather low. Here we can see a power ratio at down to 0.36, which is much lower.

9. Conclusions

Based on simulation results, the static power in a Hilbert transformer and a half band filter is investigated and compared for three different architecture structures, namely, bit-parallel, digit-serial and bit-serial arithmetic, respectively. The paper shows that a substantial reduction of the static power consumption can be achieved when bit-serial and digit-serial arithmetic is used. The paper also shows that the power ratio is strongly dependent on the used word length; that is, the reduction is larger for longer word lengths. The ratio is also dependent on the ratio between the arithmetic and the storage, for example, the registers. Architectures where the arithmetic dominates will show a larger reduction in static power consumption. A static power reduction down to half is shown for the bit-serial Hilbert transformer and almost down to a third in the half band filter. Looking at the arithmetic part only, a power ratio down to 0.11 and 0.06 is shown, respectively. The overall conclusion is that it is a good idea to switch to serial arithmetic for low- and medium-speed architectures when technologies, such as the 130 nm technology in this paper, are used. For denser technologies such as 65 nm and below, it will be well worth to switch to serial arithmetic in architectures at higher speed as well, as long as the dynamic power consumption is not dominating.

References

- [1] IEEE ITRS Technology Roadmap, <http://public.itrs.net>.
- [2] A. Agarwal, S. Mukhopadhyay, A. Raychowdhury, K. Roy, and C. H. Kim, "Leakage power analysis and reduction for nanoscale circuits," *IEEE Micro*, vol. 26, no. 2, pp. 68–80, 2006.
- [3] C. Schuster, C. Pigué, J.-L. Nagel, and P.-A. Farine, "An architecture design methodology for minimal total power consumption at fixed V_{dd} and V_{th}," *Journal of Low-Power Electronics*, vol. 1, no. 1, pp. 1–8, 2005.
- [4] C. Pigué, C. Schuster, and J.-L. Nagel, "Static and dynamic power reduction by architecture selection," in *Proceedings of the 16th International Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS '06)*, vol. 4148 of *Lecture Notes in Computer Science*, pp. 659–668, Montpellier, France, September 2006.
- [5] R. F. Lyon, "Two's complement pipeline multipliers," *IEEE Transactions on Communications*, vol. 24, no. 4, pp. 418–425, 1976.
- [6] E. R. Berlekamp, "Bit-serial Reed-Solomon encoders," *IEEE Transactions on Information Theory*, vol. 28, no. 6, pp. 869–874, 1982.
- [7] L. Wanhammar, *DSP Integrated Circuits*, Academic Press, New York, NY, USA, 1999.
- [8] M. J. Irwin and R. M. Owens, "A case for digit serial VLSI signal processors," *Journal of VLSI Signal Processing*, vol. 1, no. 4, pp. 321–334, 1990.
- [9] R. I. Hartley and K. K. Parhi, *Digit-Serial Computation*, Kluwer Academic Publishers, Boston, Mass, USA, 1995.
- [10] S. G. Smith and P. B. Denyer, *Serial Data Computation*, Kluwer Academic Publishers, Boston, Mass, USA, 1988.
- [11] P. Nilsson and M. Torkelson, "A custom digital intermediate frequency filter for the American mobile telephone system," *IEEE Journal of Solid-State Circuits*, vol. 32, no. 6, pp. 806–815, 1997.
- [12] K. Johansson, O. Gustafsson, and L. Wanhammar, "Multiple constant multiplication for digit-serial implementation of low power FIR filters," *WSEAS Transactions on Circuits and Systems*, vol. 5, no. 7, pp. 1001–1008, 2006.
- [13] P. Nilsson, "Arithmetic and architectural design to reduce leakage in nano-scale digital circuits," in *Proceedings of the 18th European Conference on Circuit Theory and Design (ECCTD '07)*, pp. 372–375, Seville, Spain, August 2007.
- [14] J. M. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits: A Design Perspective*, Prentice-Hall, Englewood Cliffs, NJ, USA, 2003.
- [15] P. Nilsson and M. Torkelson, "Method to save silicon area by increasing the filter order," *Electronics Letters*, vol. 31, no. 6, pp. 439–441, 1995.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

