Low-Power Logic Styles: CMOS Versus Pass-Transistor Logic

Reto Zimmermann and Wolfgang Fichtner, *Fellow, IEEE*

Abstract— **Recently reported logic style comparisons based on full-adder circuits claimed complementary pass-transistor logic (CPL) to be much more power-efficient than complementary CMOS. However, new comparisons performed on more efficient CMOS circuit realizations and a wider range of different logic cells, as well as the use of realistic circuit arrangements demonstrate CMOS to be superior to CPL in most cases with respect to speed, area, power dissipation, and power-delay products. An implemented 32-bit adder using complementary CMOS has a power-delay product of less than half that of the CPL version. Robustness with respect to voltage scaling and transistor sizing, as well as generality and ease-of-use, are additional advantages of CMOS logic gates, especially when cell-based design and logic synthesis are targeted. This paper shows that complementary CMOS is the logic style of choice for the implementation of arbitrary combinational circuits, if low voltage, low power, and small power-delay products are of concern.**

Index Terms— **Adder circuits, CPL, complementary CMOS, low-voltage low-power logic styles, pass-transistor logic, VLSI circuit design.**

I. Introduction

THE increasing demand for low-power very large scale integration (VLSI) can be addressed at different design levels, such as the architectural, circuit, layout, and the process technology level [1]. At the circuit design level, considerable potential for power savings exists by means of proper choice of a logic style for implementing combinational circuits. This is because all the important parameters governing power dissipation—switching capacitance, transition activity, and short-circuit currents—are strongly influenced by the chosen logic style. Depending on the application, the kind of circuit to be implemented, and the design technique used, different performance aspects become important, disallowing the formulation of universal rules for optimal logic styles. Investigations of low-power logic styles reported in the literature so far, however, have mainly focused on particular logic cells, namely full-adders, used in some arithmetic circuits. In this paper, these investigations are extended to a much wider set of logic gates, and with that, to arbitrary combinational circuits. The power dissipation characteristics of various existing logic styles are compared qualitatively and quantitatively by actual logic gate implementations and simulations under realistic circuit arrangements and operating conditions [2]. Investigations of sequential elements, such as latches and flip-flops, were not included in this work, but can be found elsewhere in the literature [3].

Manuscript received November 20, 1996; revised January 29, 1997.

The authors are with the Integrated Systems Laboratory, Swiss Federal Institute of Technology (ETH), CH-8092 Zurich, Switzerland.

Publisher Item Identifier S 0018-9200(97)04363-1.

Section II gives a short introduction to the most important existing static logic styles and compares them qualitatively. Results of quantitative comparisons based on simulations of different logic gates as well as of a 32-b adder implementation are given in Sections III and IV, respectively. Some conclusions are finally drawn in Section V.

II. Logic Styles

A. Impact of Logic Style

The logic style used in logic gates basically influences the speed, size, power dissipation, and the wiring complexity of a circuit. The circuit *delay* is determined by the number of inversion levels, the number of transistors in series, transistor sizes (i.e., channel widths), and intra- and inter-cell wiring capacitances. Circuit *size* depends on the number of transistors and their sizes and on the wiring complexity. *Power dissipation* is determined by the switching activity and the node capacitances (made up of gate, diffusion, and wire capacitances), the latter of which in turn is a function of the same parameters that also control circuit size. Finally, the *wiring complexity* is determined by the number of connections and their lengths and by whether single-rail or dual-rail logic is used. All these characteristics may vary considerably from one logic style to another and thus make the proper choice of logic style crucial for circuit performance.

As far as cell-based design techniques (e.g., standard-cells) and logic synthesis are concerned, *ease-of-use* and *generality* of logic gates is of importance as well. *Robustness*[1] with respect to voltage and transistor scaling as well as varying process and working conditions, and *compatibility* with surrounding circuitries are important aspects influenced by the implemented logic style.

B. Logic Style Requirements for Low Power

According to the formula

$$P_{dyn} = V_{dd}^2 \cdot f_{clk} \cdot \sum_n \alpha_n \cdot c_n + V_{dd} \cdot \sum_n i_{sc_n}$$

the dynamic power dissipation of a digital CMOS circuit depends on the supply voltage V_{dd}, the clock frequency f_{clk}, the node switching activities α_n, the node capacitances c_n, the node short-circuit currents i_{sc_n}, and the number of nodes n. A reduction of each of these parameters results in a reduction of dissipated

[1] A robust circuit guarantees correct functioning under a wide range of certain conditions.

power. However, clock frequency reduction is only feasible at the architecture level, whereas at the circuit level frequency f_{clk} is usually regarded as constant in order to fulfill some given throughput requirement. All the other parameters are influenced to some degree by the logic style applied. Thus, some general logic style requirements for low-power circuit implementation can be stated at this point.

1) Switched capacitance reduction: Capacitive load, originating from transistor capacitances (gate and diffusion) and interconnect wiring, is to be minimized. This is achieved by having as few transistors and circuit nodes as possible, and by reducing transistor sizes to a minimum. In particular, the number of (high-capacitive) inter-cell connections and their length (influenced by the circuit size) should be kept minimal. Another source for capacitance reduction is found at the layout level [4], which, however, is not discussed in this paper. Transistor downsizing is an effective way to reduce switched capacitance of logic gates on noncritical signal paths [5]. For that purpose, a logic style should be robust against transistor downsizing, i.e., correct functioning of logic gates with minimal or near-minimal transistor sizes must be guaranteed (*ratioless* logic).

2) Supply voltage reduction: The supply voltage and the choice of logic style are indirectly related through delay-driven voltage scaling. That is, a logic style providing fast logic gates to speed up critical signal paths allows a reduction of the supply voltage in order to achieve a given throughput. For that purpose, a logic style must be robust against supply voltage reduction, i.e., performance and correct functioning of gates must be guaranteed at low voltages as well. This becomes a severe problem at very low voltages of around 1 V and lower, where noise margins become critical [6], [7].

3) Switching activity reduction: Switching activity of a circuit is predominantly controlled at the architectural and register transfer level (RTL). At the circuit level, large differences are primarily observed between static and dynamic logic styles. On the other hand, only minor transition activity variations are observed among different static logic styles and among logic gates of different complexity, also if glitching is concerned.

4) Short-circuit current reduction: Short-circuit currents (also called dynamic leakage currents or overlap currents) may vary by a considerable amount between different logic styles. They also strongly depend on input signal slopes (i.e., steep and balanced signal slopes are better) and thus on transistor sizing. Their contribution to the overall power consumption is rather limited but still not negligible ($\approx$ 10–30%), except for very low voltages $V_{dd} \leq V_{tn} + |V_{tp}|$, where the short-circuit currents disappear. A low-power logic style should have minimal short-circuit currents and, of course, no static currents besides the inherent CMOS leakage currents.

C. Logic Style Requirements for Ease-of-Use

For ease-of-use and generality of gates, a logic style should be highly robust and have friendly electrical characteristics, that is, *decoupling* of gate inputs and outputs (i.e., at least one inverter stage per gate) as well as good *driving capabilities* and full *signal swings* at the gate outputs, so that logic gates can be cascaded arbitrarily and work reliably in any circuit configuration. These properties are prerequisites for cell-based design and logic synthesis, and they also allow for efficient gate modeling and gate-level simulation. Furthermore, a logic style should allow the efficient implementation of arbitrary logic functions and provide some regularity with respect to circuit and layout realization. Both low-power and high-speed versions of logic cells (e.g., by way of transistor sizing) should be supported in order to allow flexible power-delay tuning by the designer or the synthesis tool.

D. Static Versus Dynamic Logic Styles

A major distinction, also with respect to power dissipation, must be made between static and dynamic logic styles. As opposed to static gates, dynamic gates are clocked and work in two phases, a precharge and an evaluation phase. The logic function is realized in a single NMOS pull-down or PMOS pull-up network, resulting in small input capacitances and fast evaluation times. This makes dynamic logic attractive for high-speed applications. However, the large clock loads and the high signal transition activities due to the precharging mechanism result in an excessive high power dissipation. Also, the usage of dynamic gates is not as straightforward and universal as it is for static gates, and robustness is considerably degraded. With the exception of some very special circuit applications, dynamic logic is no viable candidate for low-power circuit design [1], [8], [9] and was therefore not considered any further in this study.

E. Complementary CMOS Logic Style

Logic gates in *conventional* or *complementary CMOS* (also simply referred to as *CMOS* in the sequel) are built from an NMOS pull-down and a dual PMOS pull-up logic network. In addition, *pass-gates* or *transmission gates* (i.e., the combination of an NMOS and a PMOS pass-transistor) are often used for implementing multiplexers, XOR-gates, and flip-flops efficiently (CMOS with pass-gates will be denoted as *CMOS+*). Any logic function can be realized by NMOS pull-down and PMOS pull-up networks connected between the gate output and the power lines. Figs. 1(a) and (b) depicts a two-input multiplexer gate (MUX2) in pure CMOS (using tristate inverters) and CMOS with pass-gates, respectively. Simple monotonic gates, such as NAND/NOR and AOI/OAI, can be realized very efficiently with only a few transistors (A⇓, P⇓)[2], one signal inversion level (T⇓), and a few circuit nodes (P⇓). Non-monotonic gates, such as XOR and multiplexer, require more complex circuit realizations but are still quite efficient.

Other advantages of the CMOS logic style are its robustness against voltage scaling and transistor sizing (high noise margins) and thus reliable operation at low voltages and arbitrary (even minimal) transistor sizes (ratioless logic). Input signals are connected to transistor gates only, which facilitates the usage and characterization of logic cells. The layout of CMOS gates is straightforward and efficient due to the complementary transistor pairs. Basically, CMOS fulfills all the requirements regarding the ease-of-use of logic gates. An often mentioned disadvantage of complementary CMOS is the substantial number of large PMOS transistors, resulting in high input loads (P⇑, T⇑, A⇑). However, the best gate performance is achieved with a

[2]This notation documents the tendency whether circuit area (A), delay (T), and power (P) are increased (⇑) or decreased (⇓) by the mentioned property.

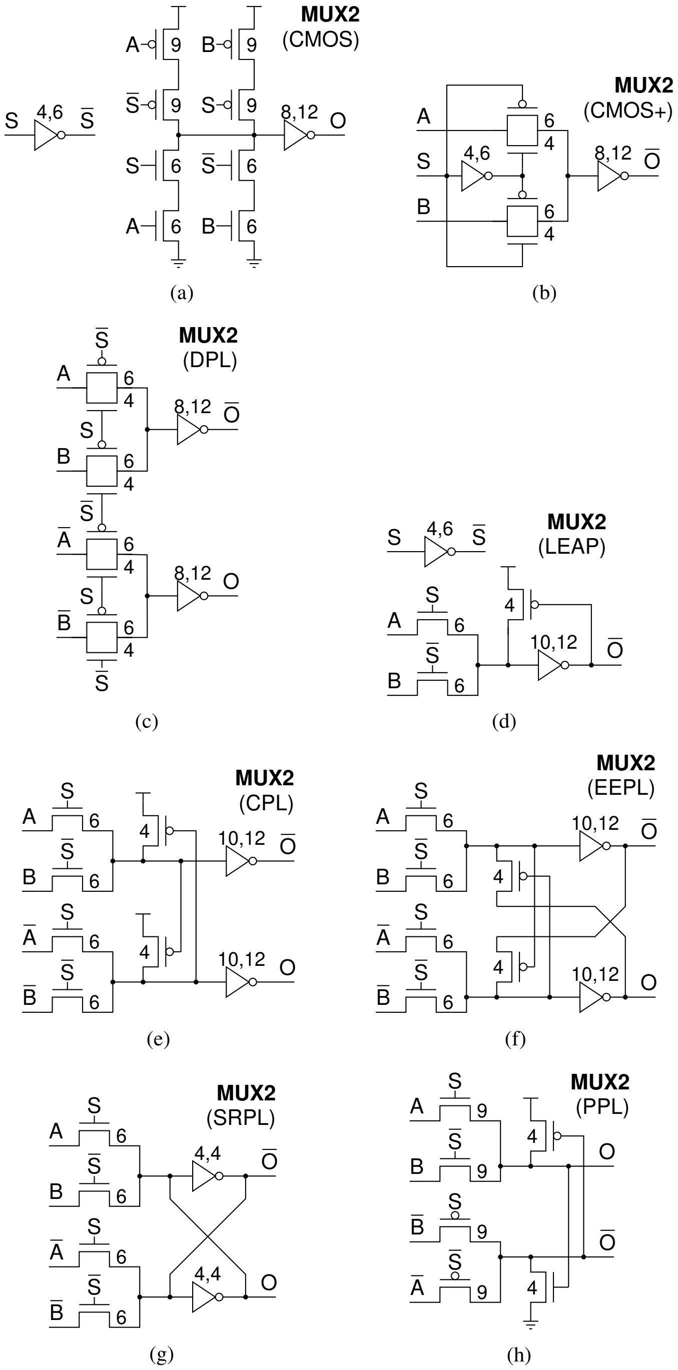

Fig. 1. Two-input multiplexer in (a) CMOS, (b) CMOS with pass-gates, (c) DPL, (d) LEAP, (e) CPL, (f) EEPL, (g) SRPL, and (h) PPL logic style.

PMOS/NMOS width ratio of only about 1.5 (= $\sqrt{\mu_n/\mu_p}$, [10]), and this ratio will decrease even further in deep-submicron technologies, where the carrier drift velocities in NMOS and PMOS transistors become almost equal due to velocity saturation [11]. Another drawback of CMOS is the relatively weak output driving capability due to series transistors in the output stage (T⇑). This, however, can be corrected by additional output buffers/inverters which are inherent in other logic styles.

A more restrictive approach was taken for the design of low-power low-voltage cells using CMOS *branch-based logic* in [4], [6]. Here, the transistor networks consist only of branches (i.e., a series of up to three transistors between power line and gate output), thus disallowing the usage of pass-gates. The advantages of transistor branches are higher layout regularity (i.e., smaller diffusion capacitances) and simpler characterization (i.e. branch instead of gate modeling). Other aspects, such as the design of delay-independent flip-flops, were addressed in order to face the massively increasing effects of process, temperature, voltage, and transistor size variations at very low voltages.

F. Pass-Transistor Logic Styles

The basic difference of pass-transistor logic compared to the CMOS logic style is that the source side of the logic transistor networks is connected to some input signals instead of the power lines. The advantage is that one pass-transistor network (either NMOS or PMOS) is sufficient to perform the logic operation, which results in a smaller number of transistors and smaller input loads, especially when NMOS networks are used (A⇓, T⇓, P⇓). However, the threshold voltage drop ($V_{out} = V_{dd} - V_{tn}$) through the NMOS transistors while passing a logic "1" makes swing (or level) restoration at the gate outputs necessary in order to avoid static currents at the subsequent output inverters or logic gates. Adjusting the threshold voltages (i.e., $V_{tp} > V_{tn}$) as a solution at the process technology level is usually not feasible for other reasons. In order to decouple gate inputs and outputs and to provide acceptable output driving capabilities, inverters are usually attached to the gate outputs (A⇑, T⇑, P⇑). Because the MOS networks are connected to variable gate inputs rather than constant power lines, only one signal path through each network must be active at a time in order to avoid shorts between inputs. Therefore, each pass-transistor network must realize a multiplexer structure, which limits the number of logic functions that can be implemented efficiently.[3] Because these pass-transistor multiplexer structures require complementary control signals, dual-rail logic is usually used in order to provide all signals in complementary form. As a consequence, two MOS networks are again required in addition to the swing restoration and output buffering circuitry (A⇑, T⇑, P⇑), which all in all annihilates the advantage of low transistor count and small input loads of pass-transistor logic. Also, the required double inter-cell wiring increases wiring complexity and capacitance by a considerable amount (A⇑, P⇑). A small advantage of dual-rail logic is that inverted signals are for free. Layout of pass-transistor cells is not as straightforward and efficient due to rather irregular transistor arrangements and high wiring requirements. Finally, pass-transistor logic with swing restoration circuitry is sensitive to voltage scaling [12] and transistor sizing with respect to circuit robustness (reduced noise margins), i.e., efficient or reliable operation of logic gates is not necessarily guaranteed at low voltages or small transistor sizes. In other words, transistor sizing is crucial for correct gate operation and therefore more difficult (ratioed logic). Short-circuit currents are rather large due to competing signals in the swing restoration circuitry.

Many different pass-transistor logic styles have been proposed recently. The most important ones are now briefly summarized.

1) Complementary pass-transistor logic (CPL): A CPL gate [1], [13] consists of two NMOS logic networks (one for each sig-

[3]Note that each logic function can be realized in a multiplexer structure, but often at a lower circuit efficiency.

nal rail), two small pull-up PMOS transistors for swing restoration, and two output inverters for the complementary output signals. Fig. 1(e) depicts a two-input multiplexer which represents the basic and minimal CPL gate structure (ten transistors). All two-input functions (e.g. AND, OR, XOR, ...) can be implemented by this basic gate structure, which is relatively expensive for simple monotonic gates such as NAND and NOR. The advantages of the CPL style are the small input loads (P⇓, T⇓), the efficient XOR and multiplexer gate implementations, the good output driving capability due to the output inverters (T⇓), and the fast differential stage due to the cross-coupled PMOS pull-up transistors (T⇓). This differential stage, on the other hand, leads to considerably larger short-circuit currents (P⇑). Other disadvantages of CPL are the substantial number of nodes and high wiring overhead due to the dual-rail signals (P⇑, A⇑) and the inefficient realization of simple gates (i.e., high transistor count, two signal inversion levels).

2) Swing restored pass-transistor logic (SRPL): The SRPL style [14] is derived from CPL. Here, the output inverters are cross-coupled to a latch structure which performs swing restoration and output buffering at the same time [Fig. 1(g)]. Note that the pull-up PMOS transistors are not required anymore and that the output nodes of the NMOS network are also the gate outputs. Because the inverters have to drive the outputs and must also be overridden by the NMOS network, transistor sizing becomes very difficult and results in poor output driving capability (T⇑, P⇑), slow switching (T⇑), and large short-circuit currents (P⇑). This becomes even worse when cascading SRPL gates. The resulting series of NMOS networks with competing inverters in between leads to very slow switching and unreliable operation. SRPL gates are highly sensitive to transistor sizing and show acceptable performance only in very special circuit arrangements (e.g., no gates in series, small output loads).

3) Double pass-transistor logic (DPL): In the DPL style [7], [15], [16], both NMOS and PMOS logic networks are used in parallel [Fig. 1(c)]. This provides full swing on the output signals (i.e., no level restoration circuitry is needed), and circuit robustness is therefore high. However, the number of transistors—especially large PMOS transistors—and the number of nodes is quite high (A⇑, P⇑), leading to substantial capacitive loads (T⇑, P⇑). The combination of large PMOS transistors and inefficient dual-rail logic makes DPL not competitive compared to other pass-transistor logic styles and to complementary CMOS. Note that DPL can be regarded as a dual-rail pass-gate logic, while CMOS+ is a single-rail pass-gate logic.

4) Single-rail pass-transistor logic (LEAP): A single-rail pass-transistor logic is proposed in the LEAP logic design scheme [12]. As opposed to the dual-rail logic styles, only single inter-cell wiring and single NMOS networks are required (A⇓, T⇓, P⇓), while the required complementary input signals are generated locally by inverters [Fig. 1(d)]. Swing restoration is realized by a fed back pull-up PMOS transistor which, however, is slower than the cross-coupled PMOS transistors of CPL working in differential mode. Note also that this swing restoration structure only works for $V_{dd} > V_{tn} + |V_{tp}|$, because the threshold voltage drop through the NMOS network for a logic "1" prevents the NMOS of the inverter and with that the pull-up PMOS from turning on. Therefore, robustness at low voltages is only guaranteed if the threshold voltages are appropriately small. On the other hand, ease-of-use of logic gates and compatibility with conventional cell-based design is partly provided in this logic style. The fact that conventional logic networks can be mapped more efficiently onto simple logic gates than on multiplexers is dealt in the LEAP system with a new synthesis approach which exploits the full functionality of multiplexer structures [12].

TABLE I
QUALITATIVE LOGIC STYLE COMPARISONS.

logic style	# MOS networks	output driving	I/O decoupl.	swing restor.	# rails	robust-ness
CMOS	n + p	med.–good	yes	no	single	high
CPL	2n	good	yes	yes	dual	medium
SRPL	2n	poor	no	yes	dual	low
DPL	2n + 2p	good	yes	no	dual	high
LEAP	n	good	yes	yes	single	medium
EEPL	2n	good	yes	yes	dual	medium
PPL	n + p	poor	no	yes	dual	low

5) Other pass-transistor logic styles: Some other pass-transistor logic styles have been proposed. The differential pass-transistor logic (DPTL) in [17] is a generalized dual-rail pass-transistor logic structure. It consists of the NMOS pass-transistor networks and a buffer circuit for level restoration, which can be a clocked precharging buffer (dynamic) or a static buffer (e.g., as in CPL). In the energy economized pass-transistor logic (EEPL) of [18], the sources of the PMOS pull-up transistors of a CPL gate are connected to the complementary output signal instead of V_{dd} [Fig. 1(f)]. The reputed advantage of shorter delay and smaller power dissipation compared to CPL, however, could not be confirmed in this work. The push-pull pass-transistor logic (PPL) of [19] can be regarded as a CPL gate without output inverters and with complementary transistors on one signal rail [i.e., PMOS pass-transistors followed by an NMOS pull-down transistor, Fig. 1(h)]. Besides its attractively low transistor count, switching and output driving characteristics are even worse than in SRPL (see Section III), and it does not work for $V_{dd} < V_{tn} + |V_{tp}|$.

G. Qualitative Comparisons

Some basic logic style characteristics which influence circuit performance and power dissipation are qualitatively compared in Table I. In particular, the number of MOS logic networks, the output driving capabilities, the presence of input/output decoupling, the need for swing restoration circuitry, the number of signal rails, and the robustness with respect to voltage scaling and transistor sizing are given for the logic styles discussed.

III. ANALYSIS OF LOGIC GATES

The efficient implementation of logic gates is a prerequisite for the realization of well-performing combinational circuits. This is especially true for high-speed and low-power applications.

A. Results from the Literature

Various investigations of logic styles with respect to low power dissipation have recently been carried out and reported in the literature [1], [12]–[14], [19]–[23]. In all these publications (except [23]), CPL or related pass-transistor logic styles are propagated as low-power logic styles. This is basically explained by the fact that CPL gates count less transistors, have smaller transistors and smaller capacitances, and are faster than gates in complementary CMOS.

However, some weak points show up in all these investigations. First, all examinations are based only on full-adder circuits. This comparison, however, is not representative because the critical three-input XOR function of the full-adder required for sum bit calculation is perfectly suited for implementation in pass-transistor logic due to its multiplexer structure. On the other hand, the XOR is the logic function with the least efficient implementation in CMOS. Secondly, rather inefficient CMOS full-adder implementations counting 40 transistors were used throughout except for [12]. More efficient CMOS realizations with only 28 transistors exist which perform better with respect to circuit size, speed, and power dissipation.

Furthermore, full-adders have only limited importance even in arithmetic circuits. Full-adders or the related 4-2 compressors are the basic cells in adder arrays (i.e., carry-save adders) used in multipliers and similar components like dividers. In such applications, efficient full-adder circuits are crucial since these building blocks are often the critical ones. However, in simpler arithmetic circuits, such as adders, incrementers/counters, and comparators, full-adders are hardly used. Most fast adder architectures (e.g. carry-lookahead) do not use entire full-adders since their function is broken up in order to speed up carry-propagation. Moreover, the greater part of typical circuit applications is made up of other (nonarithmetic) combinational functions, which require no full-adders at all.

Finally, the simulation conditions and circuit arrangements are often not clearly specified. One has to assume that idealistic and highly specific rather than realistic and more general setups are used in some cases.

B. Improved Investigations

For a more general characterization of logic styles with respect to low-power circuit implementation and standard-cell library development, the investigations have to be extended to a larger set of gates and therefore must include multiplexers and simple gates as well. Realistic circuit and simulation setups have to be chosen in order to capture worst case behavior, which is crucial in synchronous designs. In particular, gate inputs have to be driven by typical gate outputs rather than by the simulator. Similarly, gate outputs have to drive typical gate inputs, thus simulating realistic fan-outs. Several gates have to be cascaded in order to observe their behavior within multilevel logic circuits. A comprehensive set of input stimuli has to be applied during simulation for sensitization of all critical signal paths.

An additional aspect to be considered within pass-gate and pass-transistor circuits is the fact that input signals may connect to transistor gates *and* transistor sources at the same time. Since current is drawn from a logic gate input at the transistor source, switching of that respective signal is slowed down (i.e., flat signal ramp). If the same signal is connected to a transistor gate somewhere else, switching of that transistor and of the corresponding logic gate is slowed down as well. For simulation of this effect (referred to as *source-gate effect*), such worst-case input combinations must be included in the circuit arrangement as well.

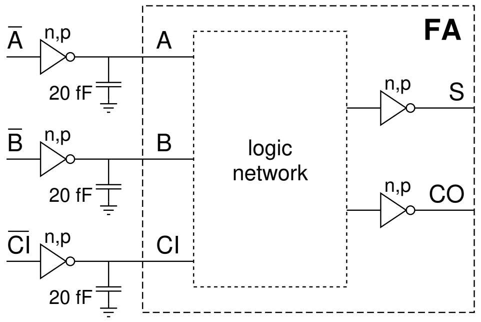

Fig. 2. Circuit arrangement for the simulation of full-adders.

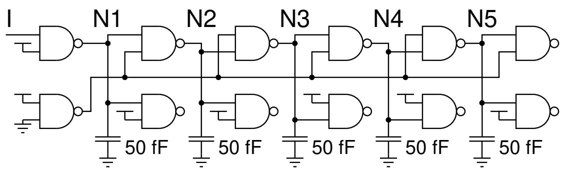

Fig. 3. Circuit arrangement for the simulation of logic gates.

C. Circuit Arrangement and Simulation Conditions

The first set of comparisons was carried out on various simple and complex logic gates. Circuits were designed at the transistor-level in a standard 0.6-μm CMOS process technology (double-metal, $V_{tn} = 0.8$ V, $V_{tp} = -0.8$ V). Layout was carried out for all compared logic gates and for the CMOS and CPL full-adders. It was done in a standard-cell-like manner using symbolic layout and compaction, which allowed for an efficient exploration of layout topologies for the different logic styles. The circuits were simulated using HSPICE at 3.3 V and 1.5 V, 27° C, 20 MHz, with the capacitances extracted from the layout. All possible transition combinations at the gate inputs were simulated. Worst-case gate delays and average power dissipation (including power from short-circuit currents) were obtained from simulation. PT-products are calculated as a quality measure for power efficiency, giving the energy consumed by a gate per switching event. Transistors were sized carefully by hand with the objective of balanced gate performance, low PT-products, and, to some extent, uniform and regular transistor sizes. Most circuits are depicted in Figs. 1 and 4 with the transistor widths (W) given in λ ($\lambda = 0.3\mu$m, $L = 2\lambda$).

Fig. 2 illustrates the circuit arrangement for simulation of the full-adders. Inverters equivalent to the full-adder output inverters are placed at the inputs and wiring capacitances of 20 fF attached in order to simulate two full-adders connected in series with a fan-out of one, which is typical for full-adder applications (e.g., adder arrays, Wallace trees, and ripple-carry adders). This simple circuit setup allows application of arbitrary signal transition combinations to the full-adder inputs, as well as consideration of output driving and fan-out characteristics.

Fig. 3 shows the general circuit arrangement used for all other

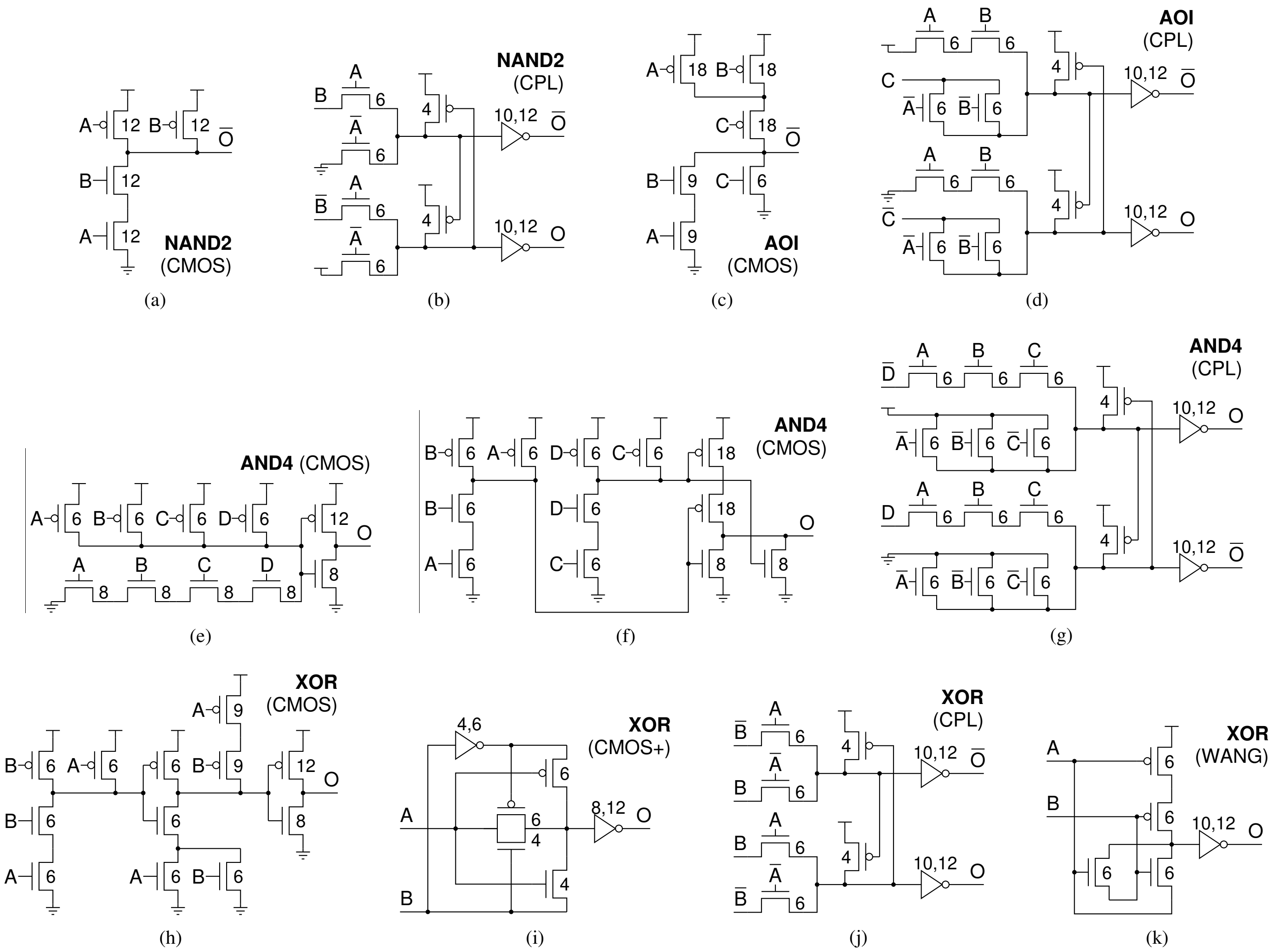

Fig. 4. Simulated gates in (a), (c), (e), (f), (h), (l), (p) CMOS, (i), (m) CMOS with pass-gates, and (b), (d), (g), (j), (n), (o) CPL logic style, (k) Wang's XOR.

logic gates. Several gates of the same type are connected in series with a fan-out of two and with typical interconnect loads attached (50 fF, corresponds to three typical cell pitches [24]). This setup makes sure that all inputs are driven by typical gate outputs and that all possible gate input combinations are simulated (source-gate effect mentioned above).

D. Comparisons and Results

1) Full-adders (FA): Four different CMOS full-adder circuits were implemented: the mentioned 28-transistor version [25] [Fig. 4(p)], the often used 40-transistor version [1], a version using branch-based gates [26], and a pure pass-gate version [25]. Pass-transistor full-adders were realized for CPL [Fig. 4(o)], LEAP, EEPL, and DPL. A comparison based on actual layout and extracted capacitances was done only for the CMOS and CPL full-adders. Their layout is given in Fig. 5. Another set of comparisons comprising all logic styles was done without layout and based on estimated diffusion and wiring capacitances.

The simulation results are given in Table II. The comparisons based on cell layouts basically confirm the better delay and PT-product values of CPL full-adders at 3.3 V due to the efficient three-input XOR pass-transistor implementation, while the power dissipation of CMOS and CPL are comparable. However, CMOS has a shorter carry-in to carry-out delay ($c_{in} \rightarrow c_{out}$) at 3.3 V as well as overall shorter delays and comparable PT-products at 1.5 V. Similar results were reported recently in [23]. Also, the layout size of the CMOS full-adder is considerably smaller due to the smaller number of transistors and, in particular, due to a higher circuit regularity (i.e., complementary transistors are easy to layout) and smaller number of wires (single-rail).

The comparisons without cell layouts show a higher performance advantage of CPL over CMOS full-adders. This again documents the worse layout efficiency of CPL. The 28-transistor CMOS full-adder performs considerably better than the 40-transistor version and the other CMOS implementations in terms of circuit speed, power dissipation, or both. EEPL proves to be comparable, but not better than CPL, from which it is derived. The single-rail pass-transistor logic style used in LEAP does not work at 1.5 V (i.e., $V_{dd} < V_{tn} + |V_{tp}|$, as mentioned earlier), and its superiority over CMOS [12] at higher voltages could not be confirmed. Finally, DPL is not competitive compared to CMOS and CPL due to the very high transistor count. Note that in all these circuit implementations, power and delay can be traded off by a considerable amount through transistor sizing, while the

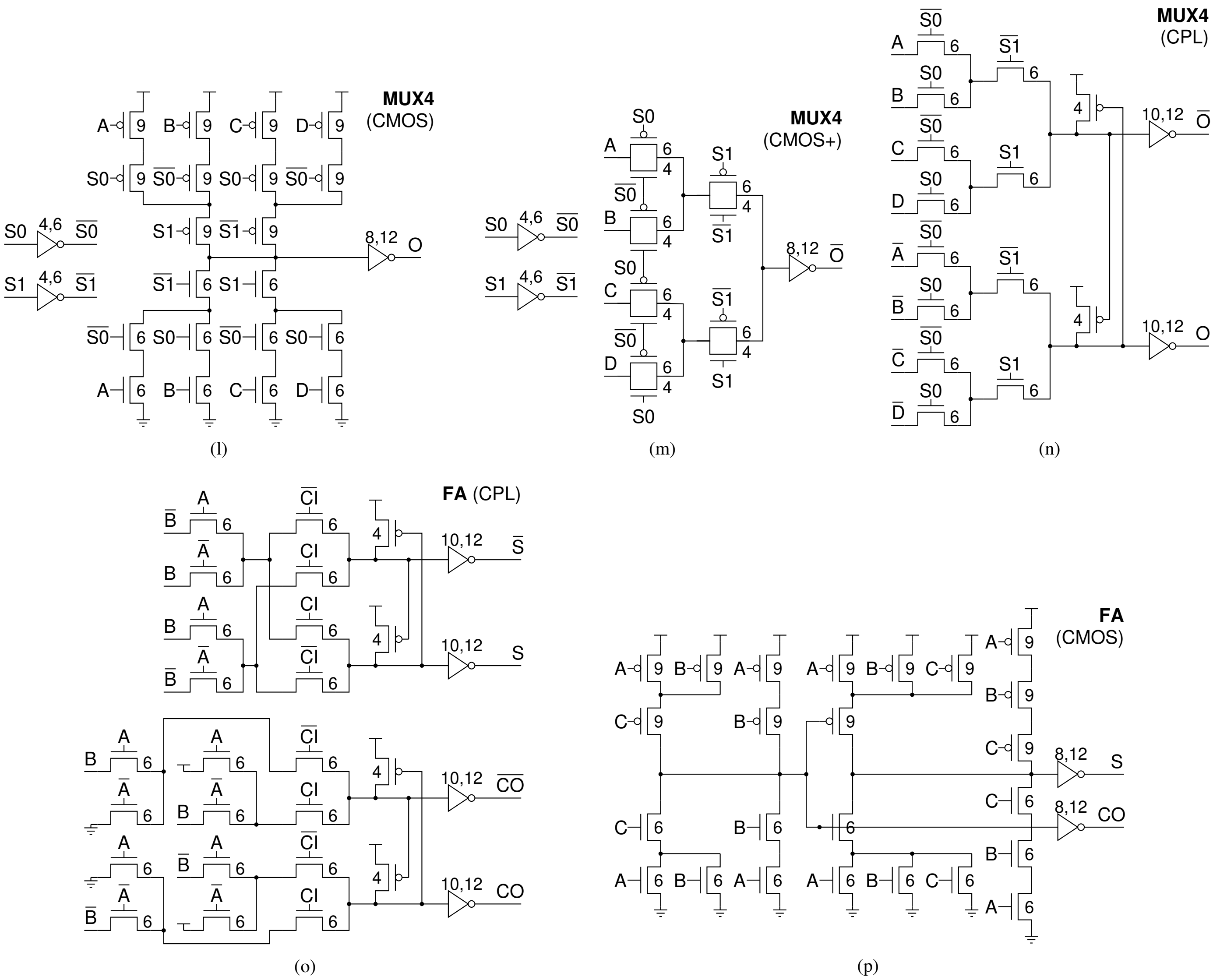

Fig. 4. (*Continued.*) Simulated gates in (a), (c), (e), (f), (h), (l), (p) CMOS, (i), (m) CMOS with pass-gates, and (b), (d), (g), (j), (n), (o) CPL logic style, (k) Wang's XOR.

PT-products remain fairly constant, except for minimum-sized transistors where PT-products become typically larger.

2) Logic gates: Two sets of comparisons on logic gates were carried out based on the cells' layout. The first set includes two-input multiplexers (MUX2) for all different logic styles. The circuits are given in Fig. 1 and the results summarized in Table III. Here, the multiplexer in complementary CMOS outperforms all other implementations with respect to circuit delay, power, PT-product, and layout size, despite the relatively high transistor count. It is far more efficient than any pass-transistor solution, also with respect to layout (Fig. 6). This is remarkable since multiplexers are actually the domain of pass-transistor logic. CPL is the best performing pass-transistor logic style and, in particular, the fastest one. Again, EEPL has worse performance than CPL, and the additional signal connections required in EEPL gates are sometimes difficult to layout. LEAP is quite power-efficient but rather slow. DPL is comparable to CPL in all respects. Finally, SRPL and PPL suffer from the weak output driving capability and the missing input-output decoupling, resulting in increasingly slow signal ramps through a series of gates and, as a consequence, in high short-circuit currents. This is illustrated by the simulated waveforms of Fig. 7 and confirms the well-known fact that gates without input-output decoupling cannot be connected in series to form arbitrary circuits without inserting buffers every few gates. This, however, makes these logic styles difficult to use, and they hardly yield better circuit performance than logic styles with inherent input-output decoupling in each gate.

In the second set of gate investigations, the following logic gates were compared between CMOS and CPL: two-input NAND (NAND2), four-input AND (AND4), three-input and-or-invert/or-and-invert (AOI/OAI), two- and four-input multiplexers (MUX2, MUX4), and two-input XOR [Figs. 4(a)–(n)]. The results are given in Table IV. In most cases, complementary CMOS clearly outperforms CPL with respect to circuit delay, power dissipation, power-delay product, and layout size. This especially holds true for the simple gates (NAND2, AND4, AOI/OAI). The only exception are the MUX4 and XOR gates

TABLE II

FULL-ADDER COMPARISONS.

gate type	logic style	delay (ns) maximal 3.3 V	delay (ns) maximal 1.5 V	delay (ns) $c_{in} \rightarrow c_{out}$ 3.3 V	delay (ns) $c_{in} \rightarrow c_{out}$ 1.5 V	power (μW) 3.3 V	power (μW) 1.5 V	PT (norm.) 3.3 V	PT (norm.) 1.5 V	# trans.	size (λ^2)
with layout (extracted capacitances)											
FA	CMOS	1.89	7.88	1.11	4.87	32.9	6.4	1.00	1.00	28	8 754
	CPL	1.39	8.33	1.23	7.95	34.1	6.0	0.76	0.99	32	14 792
without layout (estimated capacitances)											
FA	CMOS	1.50	6.00	0.85	3.50	29.3	5.7	1.00	1.00	28	–
	CMOS [1]	1.77	6.97	0.79	3.24	32.3	6.2	1.29	1.26	40	–
	CMOS [2]	2.18	9.01	1.59	5.66	31.3	6.3	1.55	1.64	30	–
	TGATE [3]	1.35	6.34	1.00	4.51	33.4	6.7	1.02	1.24	24	–
	CPL	1.02	5.06	0.85	4.54	24.6	4.3	0.57	0.63	32	–
	EEPL	1.11	5.72	0.95	5.28	25.1	4.5	0.63	0.75	32	–
	LEAP	1.73	– [4]	1.06	–	31.0	–	1.22	–	24	–
	DPL	1.31	6.81	0.64	2.89	35.8	6.9	1.07	1.36	48	–

[1] CMOS version used in most comparisons [1]
[2] decomposed, branch-based CMOS version proposed in [26]
[3] pure pass-gate version
[4] does not work for $V_{dd} < V_{tn} + |V_{tp}|$

TABLE III

MULTIPLEXER COMPARISONS (ALL LOGIC STYLES).

gate type	logic style	delay (ns) 3.3 V	delay (ns) 1.5 V	power (μW) 3.3 V	power (μW) 1.5 V	PT (norm.) 3.3 V	PT (norm.) 1.5 V	# trans.	size (λ^2)
MUX2	CMOS	1.15	4.44	10.4	2.0	1.00	1.00	12	4 111
	CMOS[1]	1.19	4.94	10.4	1.9	1.03	1.07	10	3 969
	CMOS+	1.59	6.50	10.3	1.9	1.37	1.43	8	4 455
	CPL	1.28	6.21	19.0	3.4	2.03	2.42	10	5 528
	EEPL	2.02	10.27	23.0	4.9	3.88	5.72	10	6 328
	SRPL	5.86	29.75	26.2	3.7	12.81	12.52	8	6 009
	PPL	7.77	– [2]	32.7	–	21.16	–	6	4 301
	LEAP	2.07	– [2]	12.6	–	2.18	–	7	4 333
	DPL	1.34	5.33	17.3	3.3	1.93	1.98	12	6 133

[1] without output inverter
[2] does not work for $V_{dd} < V_{tn} + |V_{tp}|$

TABLE IV

LOGIC GATES COMPARISONS (CMOS AND CPL).

gate type	logic style	delay (ns) 3.3 V	delay (ns) 1.5 V	power (μW) 3.3 V	power (μW) 1.5 V	PT (norm.) 3.3 V	PT (norm.) 1.5 V	# trans.	size (λ^2)
NAND2	CMOS	0.91	3.20	7.3	1.3	1.00	1.00	4	2 098
	CPL	1.28	6.12	18.9	3.5	3.67	4.93	10	5 477
AND4	CMOS	1.30	5.28	10.2	1.9	1.00	1.00	10	3 897
	CMOS[1]	1.15	4.81	10.2	1.9	0.88	0.91	12	4 669
	CPL	2.30	11.58	26.9	4.6	4.63	5.25	18	9 580
AOI/OAI	CMOS	1.12	4.40	9.3	1.7	1.00	1.00	6	2 778
	CPL	1.47	7.43	22.0	4.1	3.09	4.15	14	7 211
MUX2	CMOS	1.13	4.17	10.5	2.0	1.00	1.00	12	4 111
	CMOS+	1.59	6.50	10.3	1.9	1.37	1.50	8	4 455
	CPL	1.28	6.21	19.0	3.4	2.03	2.54	10	5 528
MUX4	CMOS	2.03	7.56	14.5	2.6	1.00	1.00	26	10 481
	CMOS+	2.33	10.17	14.4	2.5	1.14	1.31	18	8 112
	CPL	1.76	8.51	23.5	4.0	1.41	1.77	18	7 728 [2]
XOR	CMOS	1.43	5.51	11.2	2.1	1.00	1.00	12	4 523
	CMOS+	1.82	7.94	10.5	2.0	1.19	1.38	8	4 455
	CPL	1.35	6.21	19.3	3.5	1.62	1.90	10	5 069
	WANG	1.45	– [3]	27.1	–	2.45	–	6	3 190

[1] two-input NAND/NOR combination (decomposition)
[2] relaxed cell layout rules due to large number of (otherwise area dominating) input/output wires
[3] does not work for $V_{dd} < V_{tn} + |V_{tp}|$

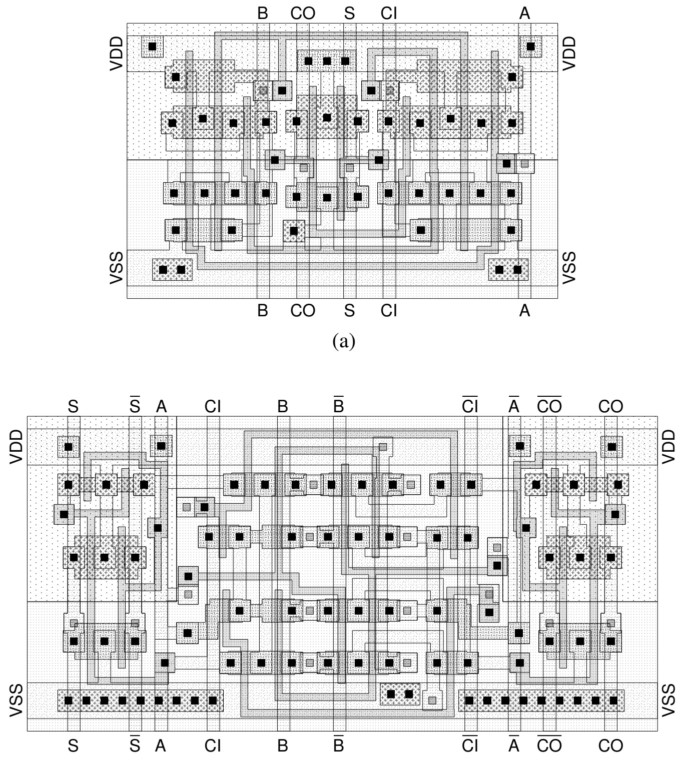

Fig. 5. Layout of (a) CMOS and (b) CPL full-adder.

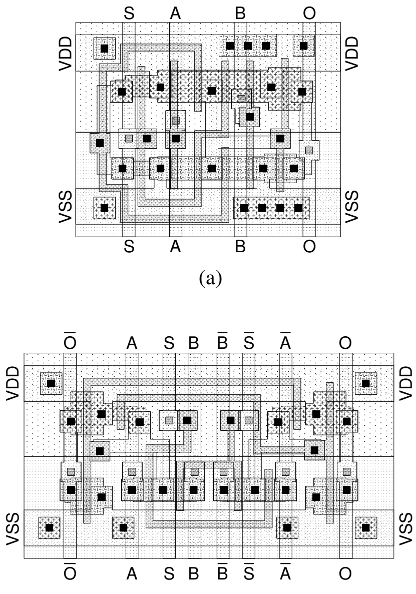

Fig. 6. Layout of (a) CMOS and (b) CPL two-input multiplexer.

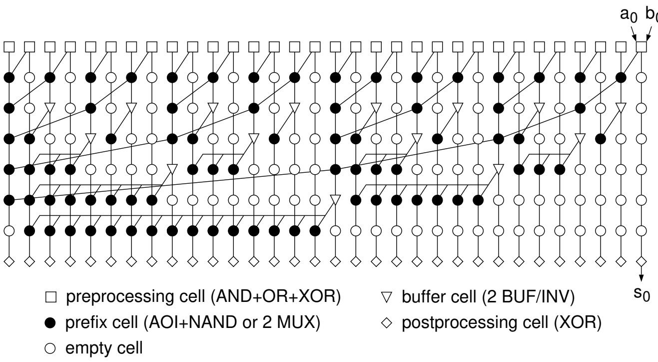

Fig. 8. Buffered parallel-prefix adder structure.

where CPL is faster at 3.3 V. The small layout area of MUX4 in CPL was only achieved by relaxing the cell layout rules (i.e., all input metal-2 wires lead only to one side of the cell). Otherwise, its layout size would have been dominated by the large number of input/output wires and thus have been much larger. CMOS also proves to be less sensitive to voltage scaling than CPL. The delays increase by a smaller amount and the PT-product ratios get better for CMOS when scaling down to 1.5 V. Finally, pure CMOS also performs better than the combination of CMOS and pass-gates (CMOS+), which is one basic advantage of branch-based logic [4]. Also, a reduction of short-circuit currents in CMOS compared to pass-gate logic was reported in [23], when comparing tristate inverter selectors (CMOS) with pass-gate selectors (CMOS+). The two CMOS implementations of AND4 further demonstrate that the decomposition of complex gates into simpler ones often improves performance [4], but not always (see CMOS implementations of full-adder). Complex gate decomposition minimizes the number of series transistors (i.e., simpler gates)—an important aspect at low supply voltages—at the cost of additional signal inversion levels (i.e., more gates).

E. Discussion

Among the pass-transistor logic styles, CPL proves to have the best performance values and lowest power-delay products. Only the single-rail style of LEAP is a viable alternative if lower power and compatibility with cell-based design are of concern.

Complementary CMOS, however, proves to be superior to all pass-transistor logic styles in performance for all logic gates, with the exception of the full-adder at higher supply voltages. The advantages of efficient circuit and layout implementation of simple gates, the absence of swing restoration circuitry, and the single-rail logic property are predominant in most circuit applications. CMOS also shows the highest robustness and smallest sensitivity to transistor and voltage scaling, which was also documented in [23].

IV. Analysis of Adders

Binary adders are good examples for circuit performance comparisons because they include a balanced combination of different logic gates and make up the crucial building blocks in many circuit applications.

A. Adder Architecture and Implementation

Adder architecture investigations carried out on cell-based designs showed the best circuit performance measures for the class of parallel-prefix adders (carry-lookahead adders), with the one using the parallel-prefix structure by Sklansky [27] resulting in the fastest adder circuit implementations [28], [29]. This seems also to hold true for transistor-level circuits, since the

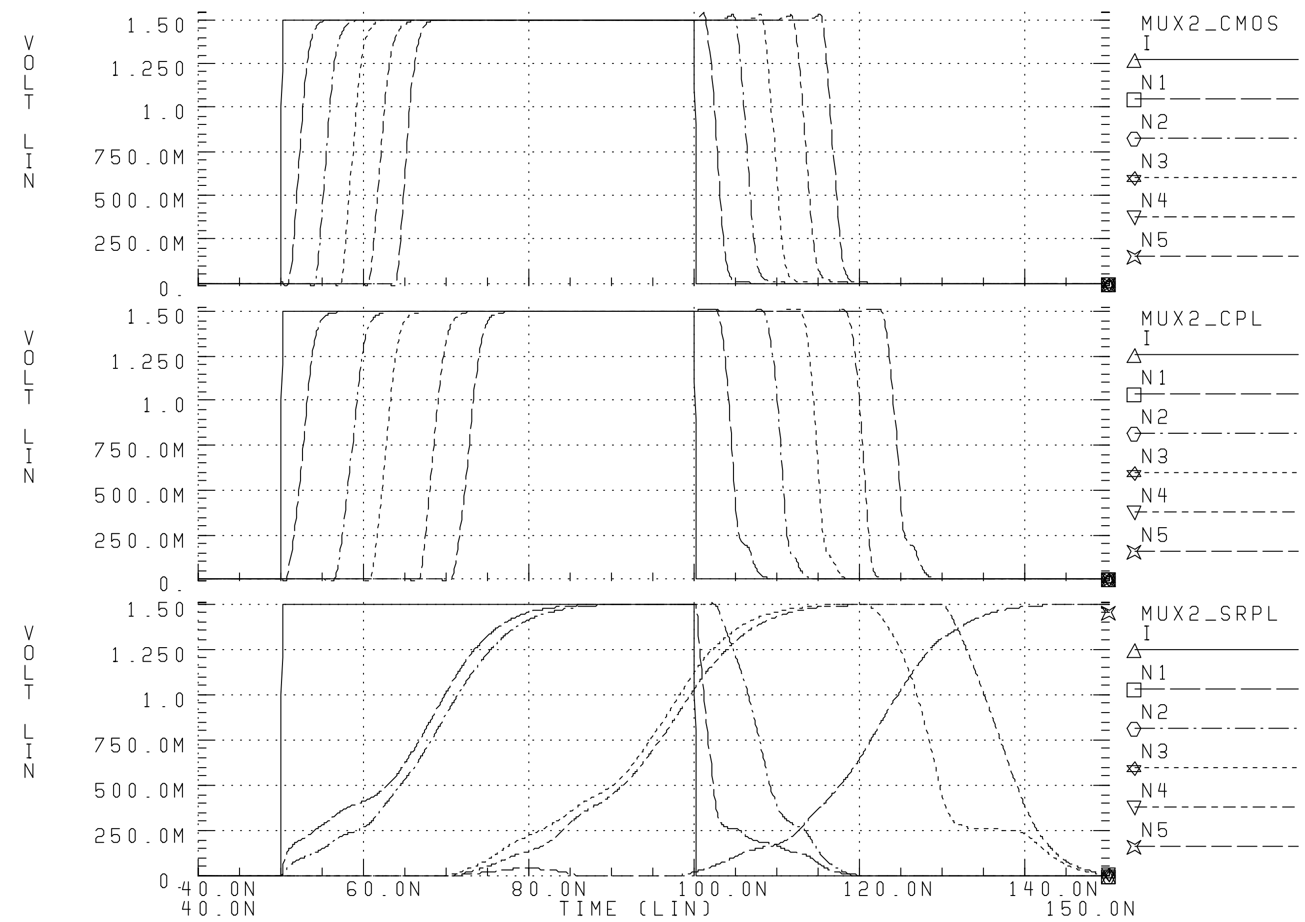

Fig. 7. Simulation waveforms for two-input multiplexer in CMOS, CPL, and SRPL logic style (@ 1.5 V).

TABLE V

32-B ADDER COMPARISONS.

logic style	delay (ns)	power (mW)	PT (norm.)	I_{sc}/I_{tot} (%)	# trans.	# nodes	switching activity	voltage (V)	process technology
CMOS	4.14	7.50	1.00	23.6	1 607	869	0.36	2.8	0.5 μm
CPL	3.47	25.90	2.89	31.2	2 774	1 228	0.49	2.8	0.5 μm
CPL[1]	4.73	16.80	2.56	27.2	2 774	1 228	0.50	2.8	0.5 μm
DPL[2]	5.00	15.00	2.42	–	–	–	–	3.3	0.5 μm

[1] down-sized transistors
[2] conditional-sum adder in DPL from literature [30]

area-efficient but slower Manchester chains as a transistor-level alternative do not fit well into the parallel-prefix adder structure.

A 32-b adder was realized in a 0.5-μm CMOS process using the unbounded fan-out parallel-prefix adder structure of Fig. 8. One level of buffers was inserted for driving the nodes with large fan-outs and thus for fan-out decoupling on the critical paths (i.e., speed-up). Since the prefix carry-propagation can be realized using AOI/OAI-gates or multiplexers, the more efficient variant was chosen for each logic style. That is, the CMOS implementation makes use of the efficient AOI/OAI-gates while the CPL solution uses two-input multiplexers. Transistors were sized for high speed. Note that these adder architectures do not contain any full-adder circuits, and that the three-input XOR's are split into two two-input XOR's, one in the preprocessing and one in the postprocessing stage. The adders were simulated at 2.8 V, 110° C, and 100 MHz with estimated wiring capacitances (layout topology taken into account). The worst-case delay on the critical path as well as average power dissipation on a set of random data was measured.

B. Results and Discussion

Table V gives the comparison results. The CMOS solution is about 20% slower than the CPL version, but has a much smaller transistor count and dissipates less than 1/3 the power. A CPL version with down-sized transistors still consumes twice as much power and is slower than CMOS. The CMOS adder has 41% fewer transistors and 29% fewer circuit nodes than the CPL version. The reasons for the greater power dissipation of the CPL adder are basically the larger switched capacitance (more transistors, dual-rail wiring), larger short-circuit currents I_{sc} (differential swing-restoration circuitry), and a higher average switching activity than was observed in the CMOS version. On the other hand, the CMOS adder takes advantage of the efficient implementation of the simple AOI/OAI-gates used for carry-propagation and of the single-rail interconnects. Note that the inaccuracies from wiring estimation can be regarded as considerably smaller than the observed differences in circuit performance.

For comparison, the performance figures of a low-power high-performance 32-b conditional-sum adder implementation using the DPL style are given from the literature [30].

V. Conclusions

In our investigations, CPL was found to be the most efficient pass-transistor logic style. Complementary CMOS, however, proves to be superior to CPL in all respects with only few exceptions. An interesting alternative is represented by the single-rail pass-transistor logic and the proposed synthesis approach used in LEAP in order to better exploit the multiplexer structure of pass-transistor logic.

The advantages of high functionality with few pass-transistors and of small input capacitances in the CPL style are partially undone by the need for swing restoration circuitry, dual-rail encoding, and the resulting wiring overhead. The investigation results presented show that—for all simple and complex logic gates except the full-adder, and under realistic circuit conditions—-complementary static CMOS performs much better than CPL and other pass-transistor logic styles if low power is of concern. CMOS also compares favorably with regard to circuit speed and layout efficiency. Its single-rail property is crucial for saving routing resources, which is an important issue in submicron VLSI. Its robustness against transistor downsizing and voltage scaling allows the efficient power optimization of noncritical signal nets and of entire circuit components. As a matter of fact, circuit robustness is becoming a key aspect in deep-submicron VLSI, where variation ranges of many process and environment parameters will increase massively [24]. This, together with its ease-of-use, makes complementary CMOS the logic style of choice for low-power, low-voltage implementation of arbitrary combinational circuits and for design automation—i.e., low-power synthesis and cell-based design—, also and particularly in the future [10]. However, other logic styles, such as CPL, may still be viable candidates for low-power high-speed implementation of dedicated circuit applications like multipliers.

Acknowledgment

The authors would like to thank Dr. H. Kaeslin for his encouragement, valuable suggestions, and careful reviewing. They would also like to thank Dr. N. Felber for his support and the reviewers for their constructive comments. This work was partly done in collaboration with R. Gupta and D. Fisher from the DSP Device Design Group of Rockwell Semiconductor Systems in Newport Beach, CA, USA.

References

[1] A. P. Chandrakasan and R. W. Brodersen, *Low Power Digital CMOS Design*, Kluwer, Norwell, MA, 1995.

[2] R. Zimmermann and R. Gupta, "Low-power logic styles : CMOS versus CPL", in *Proc. 22nd European Solid-State Circuits Conf.*, Neuchâtel, Switzerland, Sept. 1996, pp. 112–115.

[3] J. Yuan and C. Svensson, "New single-clock CMOS latches and flipflops with improved speed and power savings", *IEEE J. Solid-State Circuits*, vol. 32, pp. 62–69, Jan. 1997.

[4] C. Piguet, J.-M. Masgonty, P. Mosch, C. Arm, and V. von Kaenel, "Low-power low-voltage standard cell libraries", in *Proc. Low Voltage – Low Power Workshop, ESSCIRC'95*, Lille, France, Sept. 1995.

[5] R. Rogenmoser, H. Kaeslin, and N. Felber, "The impact of transistor sizing on power efficiency in submicron CMOS circuits", in *Proc. 22nd European Solid-State Circuits Conf.*, Neuchâtel, Switzerland, Sept. 1996, pp. 124–127.

[6] C. Piguet, J.-M. Masgonty, S. Cserveny, and E. Dijkstra, "Low-power low-voltage digital CMOS cell design", in *Proc. PATMOS'94*, Barcelona, Spain, Oct. 1994, pp. 132–139.

[7] N. Ohkubo *et al.*, "A 4.4 ns CMOS 54 $\times$ 54-b multiplier using pass-transistor multiplexer", *IEEE J. Solid-State Circuits*, vol. 30, pp. 251–257, Mar. 1995.

[8] P. Ng, P. T. Balsara, and D. Steiss, "Performance of CMOS differential circuits", *IEEE J. Solid-State Circuits*, vol. 31, pp. 841–846, June 1996.

[9] K. Chu and D. Pulfrey, "A comparison of CMOS circuit techniques: Differential cascode voltage switch logic versus conventional logic", *IEEE J. Solid-State Circuits*, vol. 22, pp. 528–532, Aug. 1987.

[10] J. M. Rabaey, *Digital Integrated Circuits*, Prentice Hall, Englewood Cliffs, NJ, 1996.

[11] N. Arora, *MOSFET Models for VLSI Circuit Simulation*, Springer-Verlag, Wien, Austria, 1993.

[12] K. Yano, Y. Sasaki, K. Rikino, and K. Seki, "Top-down pass-transistor logic design", *IEEE J. Solid-State Circuits*, vol. 31, pp. 792–803, June 1996.

[13] K. Yano *et al.*, "A 3.8-ns CMOS 16$\times$16-b multiplier using complementary pass-transistor logic", *IEEE J. Solid-State Circuits*, vol. 25, pp. 388–393, Apr. 1990.

[14] A. Parameswar, H. Hara, and T. Sakurai, "A swing restored pass-transistor logic-based multiply and accumulate circuit for multimedia applications", *IEEE J. Solid-State Circuits*, vol. 31, pp. 805–809, June 1996.

[15] M. Suzuki, N. Ohkubo, T. Yamanaka, A. Shimizu, and K. Sasaki, "A 1.5ns 32b CMOS ALU in double pass-transistor logic", in *Proc. 1993 IEEE Int. Solid-State Circuits Conf.*, Feb. 1993, pp. 90–91.

[16] A. Bellaouar and M. I. Elmasry, *Low-Power Digital VLSI Design: Circuits and Systems*, Kluwer, Norwell, MA, 1995.

[17] J. H. Pasternak and C. A. T. Salama, "Differential pass-transistor logic", *IEEE Circuits & Devices*, pp. 23–28, July 1993.

[18] M. Song, G. Kang, S. Kim, and B. Kang, "Design methodology for high speed and low power digital circuits with energy economized pass-transistor logic (EEPL)", in *Proc. 22nd European Solid-State Circuits Conf.*, Neuchâtel, Switzerland, Sept. 1996, pp. 120–123.

[19] W.-H. Paik, H.-J. Ki, and S.-W. Kim, "Push-pull pass-transistor logic family for low-voltage and low-power", in *Proc. 22nd European Solid-State Circuits Conf.*, Neuchâtel, Switzerland, Sept. 1996, pp. 116–119.

[20] T. Kuroda and T. Sakurai, "Overview of low-power ULSI circuit techniques", *IEICE Trans. Electron.*, vol. E78-C, pp. 334–344, Apr. 1995.

[21] K. Shimohigashi and K. Seki, "Low-voltage ULSI design", *IEEE J. Solid-State Circuits*, vol. 28, pp. 408–413, Apr. 1993.

[22] I. S. Abu-Khater, A. Bellaouar, and M. I. Elmasry, "Circuit techniques for CMOS low-power high-performance multipliers", *IEEE J. Solid-State Circuits*, vol. 31, pp. 1535–1546, Oct. 1996.

[23] M. Izumikawa *et al.*, "A 0.25-μm CMOS 0.9-V 100-MHz DSP core", *IEEE J. Solid-State Circuits*, vol. 32, pp. 52–61, Jan. 1997.

[24] J. D. Meindl, "Gigascale integration: Is the sky the limit?", *IEEE Circuits & Devices*, vol. 12, pp. 19–32, Nov. 1996.

[25] N. H. E. Weste and K. Eshraghian, *Principles of CMOS VLSI Design*, Addison-Wesley, Reading, MA, 1985.

[26] J.-M. Masgonty, C. Arm, and C. Piguet, "Technology- and power-supply-independent cell library", in *Proc. IEEE Custom Integrated Circuits Conf.*, San Diego, CA, May 1991, pp. 25.5/1–4.

[27] J. Sklansky, "Conditional sum addition logic", *IRE Trans. Electron. Comput.*, vol. EC-9, pp. 226–231, June 1960.

[28] R. Zimmermann and H. Kaeslin, "Cell-based multilevel carry-increment adders with minimal AT- and PT-products", submitted to *IEEE Trans. VLSI Syst.*

[29] R. Zimmermann, "Non-heuristic optimization and synthesis of parallel-prefix adders", in *Proc. Int. Workshop on Logic and Architecture Synthesis*, Grenoble, France, Dec. 1996, pp. 123–132.

[30] I. S. Abu-Khater and R. H. Yan, "A 1-V low-power high-performance 32-bit conditional sum adder", in *Proc. 1994 IEEE Symp. Low Power Electron.*, San Diego, Oct. 1994, pp. 66–67.

Reto Zimmermann received the Dipl. Ing. degree in computer science from the Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, in 1991. He is currently working toward the Ph.D. degree in electrical engineering.

He joined the Integrated Systems Laboratory of ETH in 1991 as Research and Teaching Assistant. He was involved in the implementation of VLSI components for cryptographic and spread-spectrum systems and in the design and synthesis of arithmetic units for cell-based VLSI. His research interests include digital VLSI design and synthesis, high-speed and low-power circuit techniques, computer arithmetic, computer-aided design, and artificial intelligence.

Wolfgang Fichtner (M'79–SM'84–F'90) received the Dipl. Ing. degree in physics and the Ph.D. degree in electrical engineering from the Technical University of Vienna, Austria, in 1974 and 1978, respectively.

From 1975 to 1978, he was an Assistant Professor in the Department of Electrical Engineering, Technical University of Vienna. From 1979 through 1985, he worked at AT&T Bell Laboratories, Murray Hill, NJ. Since 1985 he is Professor and Head of the Integrated Systems Laboratory at the Swiss Federal Institute of Technology (ETH). In 1993, he founded ISE Integrated Systems Engineering AG, a company in the field of technology CAD.

Dr. Fichtner is a member of the Swiss National Academy of Engineering.