

Pseudo Dynamic Logic (SDL): A High-Speed and Low-Power Dynamic Logic Family

G. R. Chaji, S. M. Fakhraie, K. C. Smith*

VLSI Circuits and Systems Laboratory

ECE Department, University of Tehran, Tehran, IRAN

*ECE Department, University of Toronto, Toronto, Canada

Email: rzchaji@yahoo.com, mfakhraie@yahoo.com, lcfujino@cs.com

ABSTRACT

In this paper, a new logic-design style called Pseudo Dynamic Logic (SDL) is introduced. In this logic-design style, the internal nodes of the logic circuits are not precharged to high or low values, rather the initial charges on nodes are shared to yield an intermediate precharge value for faster evaluation. A 32-bit adder has been designed and simulated using HSPICE Level-49 parameters of a 0.6 μ m CMOS process. Simulated measurements on this adder show that the worst-case delay is 1.56ns. This demonstrates 2.1 times speed improvement in comparison to a domino dynamic logic design implemented with the same technology.

1 INTRODUCTION

Dynamic circuits have become a necessity for designing high speed and compact circuits [1], [4]. Dynamic logic circuits use only the PDN (Pull-Down Network) or PUN (Pull Up Network) of a comparable CMOS circuit, so their input capacitances and areas are reduced. However, there are three disadvantages associated with dynamic circuits. First, simple dynamic circuits cannot chain together easily. If similar dynamic circuit (e.g. N-Domino circuit) is to be connected in a chain, an inverter must be inserted between each stage [1], [4], [6]. The second problem is clock power [6]. The clock capacitance is large and the clock activity is high, so the clock power consumption is very large. The third problem is that in the precharge phase, charges of some internal nodes are destroyed. This means that they are precharged to predetermined values regardless of their final states.

In the following section, a new logic style is proposed that uses a self-timing concept and has smaller clock capacitance than other dynamic circuits. Therefore, succeeding stages can be chained together easily, and during the precharge phase, the charges of the internal nodes are reused.

2 SDL LOGIC

In this section, the basic operation of SDL logic is described. The operation of SDL is based on Pass-Gate Tree (PG) concept [2], [3], [7].

Figure 1 shows the SDL gate performing an AND function: $q = a.b$. Input variables are implemented using dual-rail signaling style. The ah and bh signals are equal to the input variables a and

b respectively. al and b are complimentary signals of input variables a and b . The $Clkb$ is compliment of the main system clock (Clk).

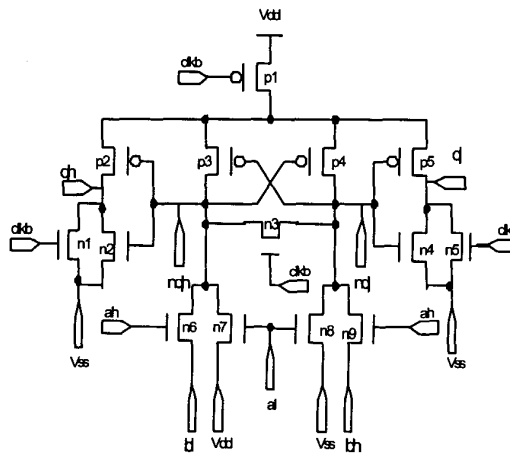


Figure 1: AND gate in SDL logic style.

2.1 PRECHARGE

The precharge phase begins when $Clkb$ goes high. In precharge, all inputs and outputs are forced low. Transistor $N6, N7, N8, N9$ and $P1$ are OFF and transistor $N1, N3$ and $N5$ are ON (Figure 1).

The charges of qh and ql nodes are discharged from $N1$ or $N5$ respectively. The charges of nqh and nql are shared and voltages of nqh and nql nodes become about $Vdd/2$ [8]. Transistors $N2, N4, P2$, and $P5$ are weakly active. $N2$ and $N4$ help $N1$ and $N5$, so $N1, N2, N4, N5$ are weak transistors. However, the sources of $P2$ and $P5$ are floating. Therefore, these transistors do not perform any function, except partly depleting the capacitive node connected to their sources.

2.2 EVALUATE

This phase begins when $Clkb$ goes low. In the evaluate phase transistor $N1, N3$, and $N5$ are OFF and $P1$ is ON (Figure 1).

If both ah and bh are high, then the charge on node nqh is discharged through $N6$ (from $V_{dd}/2$ to V_{ss}) and node nql is charged by $N9$ (from $V_{dd}/2$ to $V_{dd} - V_{thn}$) and the swing is completed to V_{dd} through $P4$ (whose gate is kept at low voltage by the node nqh) and $P1$. Then qh goes high and ql goes low. If ah is low, then nqh is charged by $N7$, and nql is discharged by $N8$. If bh is low and ah is high, then nqh is charged through $N6$ and nql is discharged by $N9$. Therefore, qh goes low and ql goes high.

3 COMPLIMENT SIGNAL (CMP)

If we chain this gate with another gate, a problem arises unless, the second gate is held off until the evaluate phase of the first gate is completed. The *CMP* signal shows this completion.

The *CMP* signal is active when one of the complementary rails goes high. Figure 2 shows a simple circuit that can generate this signal. When *Clk* is low *CMP* goes high. When *Clk* is high, *CMP* goes low when one of $N2$ or $N3$ is *ON*. A pair of complementary signals (e.g. qh and ql) is connected to these two transistors ($N2$ and $N3$). The next stage uses this *CMP* signal as its *Clkb*.

To obtain better performance, the latest occurring inputs should be connected to the nodes that are closest to the output.

We have proposed two chaining schemes for SDL logic. In the first scheme, the gates having the same order of inputs are combined into one group. For each group there is one *CMPG* that corresponds to the slowest input of this group. In the second scheme, for each gate there is a *CMPG* that corresponds to the slowest input to this gate. The first scheme has an area saving in comparison to the second, but the second scheme is easier to design (Figure 3).

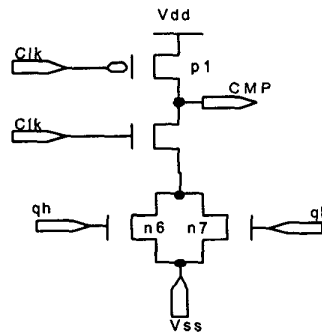


Figure 2: CMP Generator (CMPG).

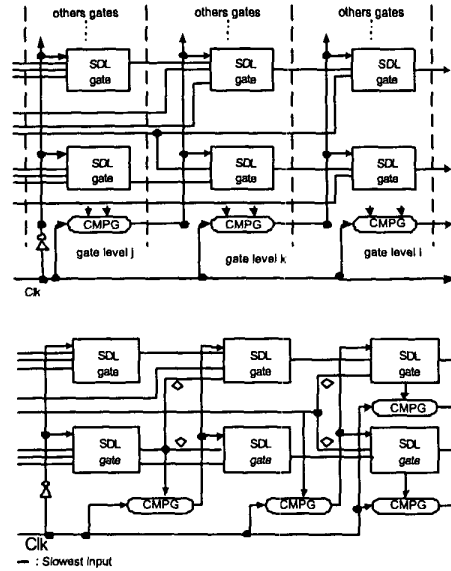


Figure 3. Chaining schemes.

4 ADDER DESIGN WITH SDL

A sample Adder was designed in Manchester-carry-chain (*MCC*) style. It has five stages (Figure 4). At the first stage, it generates 32 propagate (p_i) and generate (g_i) signals.

$$p_i = a_i \oplus b_i, \quad i=1, \dots, 32$$

$$g_i = a_i \cdot b_i, \quad i=1, \dots, 32$$

At the next stage, eight group-generate (gg) and eight group-propagate (gp) signals are produced for each of eight four-bit-wide sets of signals. gg shows that at least one carry is generated and propagated to the output (e.g. C_4) while gp shows that carry-in has been propagated to the output (e.g. C_4):

$$gg = g_{i-4} \cdot p_{i-1} \cdot p_{i-2} \cdot p_{i-3} + g_{i-3} \cdot p_{i-2} \cdot p_{i-3} + g_{i-2} \cdot p_{i-3} + g_{i-1}$$

$$gp_i = C_{in} \cdot p_i \cdot p_{i-1} \cdot p_{i-2} \cdot p_{i-3}$$

At the third stage, modulus-four carries are produced to be used in the next stage:

$$C_0 = C_{in}$$

$$C_{4^i} = C_{4^{i-1}} \cdot gp_{i-1} + gg_{i-1}$$

At the fourth stage, all other carries are produced using Manchester-Carry-Chain (*MCC*) logic. At the fifth stage, the sum finally results are produced:

$$S_i = p_i \oplus C_i$$

5 ADDER SIMULATION RESULTS

We have simulated this Adder using HSPICE Level-49 models of a 0.6 μ m CMOS process, and have measured the delay of the critical path (in which C_i is generated and propagated to S_{31}), power consumption of internal logic, input nodes, and clock tree. For a fair comparison, an optimized dynamic adder with similar blocks has been designed, optimized, and simulated. The following tables show the results of our comparative study.

5.1 DELAY

The critical path in this adder occurs when a carry is generated at the first bit and propagated to the last bit (32nd bit), while intermediate bits have produced no carry.

Table 1: Tp for the critical path.

Logic Style	Tp (ns)
Dual Rail Dynamic	3.4
New SDL	1.56

5.2 POWER CONSUMPTION

Three components of the total power consumption are important, namely the input-node power, internal-logic power, and clock-distribution power consumption. Input power is determined by input capacitance. Internal power is determined by two factors: the internal node capacitance of the gate, and short-circuit current flow. The loading of the clock-distribution tree and the level of clock activity decide the clock power consumption.

Table 2: Power Consumption

Logic Style	Input Power (mW)	Internal Power (mW)	Clock Power (mW)	Total Power (mW)
Dual Rail Dynamic	5.15	23.30	7.4	35.5
New SDL	6.20	24.13	4.4	34.73

5.3 AREA

Generally, the summation of the widths of all of the transistor employed for each circuit in each logic style gives an indication of total device area. Accordingly, we use this method for area measurement level. As well, categorized total transistor counts are given with NMOS and PMOS counts specified separately:

Table 3: Total width of the employed transistors.

Logic style	XOR (μ m)	AND (μ m)	OR (μ m)
Dual Rail Dynamic	37	32.2	32.2
New SDL	27.9	27.9	27.9

Table 4: Number of the transistors for each adder.

Logic	PMOS		NMOS		Total
	Weak	Strong	Weak	Strong	
Dual rail dynamic	288	576	--	1232	2096
New SDL	288	432	720	704	2144

6 CONCLUSION

A new family of dynamic logic has been introduced. In this style, charge is shared between respective internal nodes in the precharge phase. In the evaluate phase, if the node should go high, it must be charged from $V_{dd}/2$ to V_{dd} , while if it should go low, it must charge from $V_{dd}/2$ to V_{ss} . Correspondingly, the speed of SDL is 2.1 times faster than dual-rail dynamic logic. As well, SDL shows measurable improvements in power consumed and area used in comparison with standard dual-rail dynamic logic circuits.

7 REFERENCES

- [1] J. M. Rabaey, *Digital Integrated Circuits*, Prentice Hall, 1996.
- [2] Fang-shi Lai, and Wei Hwang, "Design and implementation of differential cascode voltage switch with pass-gate (DCVSPG) logic for high-performance digital systems," *IEEE J. of Solid-State Circuit*, vol. 32, NO. 4, April 1997.
- [3] F. S. Lai, and W. Hwang, "Differential cascode switch pass-gate (DCVSPG) logic tree for high performance CMOS digital systems", *International Symposium on VLSI Technology, Systems, and Applications*, 1993
- [4] Gin Yee, and Carl Sechen, "Clock-delayed domino for dynamic circuit design," *IEEE Transactions On Very Large Scale Integration (VLSI) Systems*, vol. 8, NO. 4, August 2000.
- [5] Uming Ko, Poras T. Balsara, and Wai Lee, "Low-power design techniques for high-performance CMOS adders," *IEEE Transactions On Very Large Scale Integration (VLSI) Systems*, vol. 3, No. 2, June 1995.
- [6] R. Rafati, S. M. Fakhraie, and K. C. Smith, "Low-power data-driven dynamic logic (D³L)", *ISCAS '2000*, vol.1, pp. 752-755.
- [7] Reto Zimmerman and Wolfgang Fichtner, "Low-Power Logic Style: CMOS Versus Pass-Transistor Logic", *IEEE J. of solid-state Circuits*, vol. 32, NO. 7, July 1997.
- [8] A.J. Acosta, M. Valencia, A. Barriga, M.J. Bellido and J.L. Huertas, "SOSD: A New CMOS Differential-Type Structure", *IEEE J. of solid-state circuits*, vol. 30, NO. 7, July 1995.

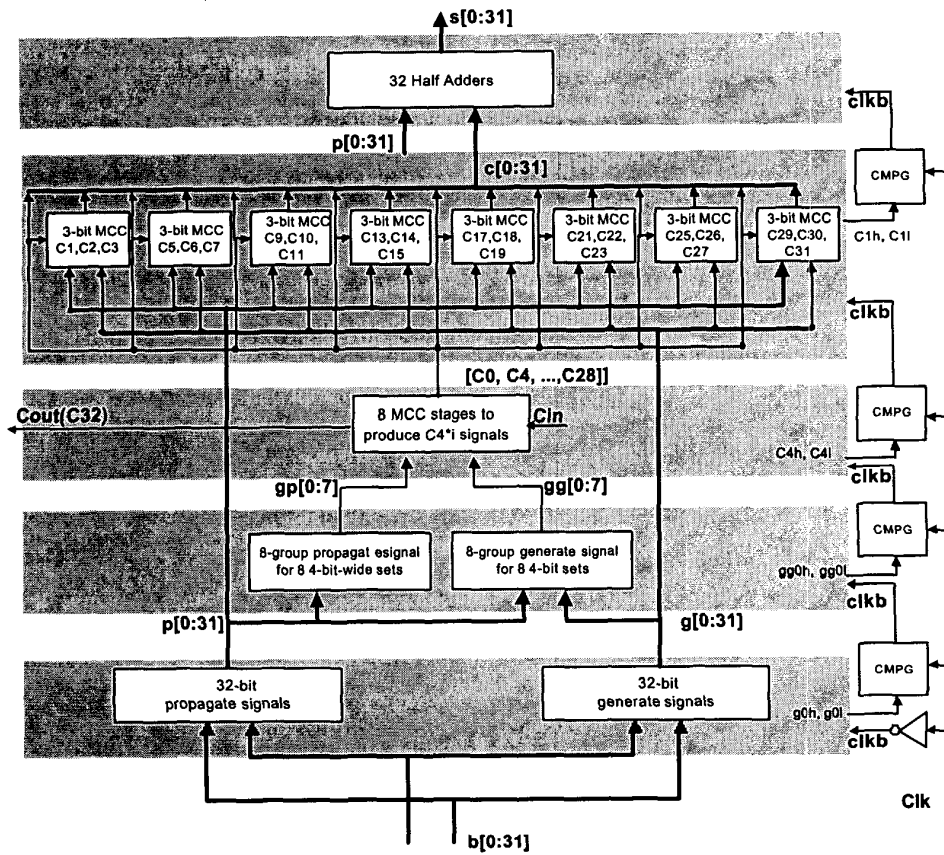


Figure 4: 32-bit Adder structure.