High Performance Current-Mode Differential Logic

Ling Zhang[1], Jianhua Liu[2], Haikun Zhu[3], Chung-Kuan Cheng[1], Masanori Hashimoto[4]

[1]Dept. Computer Science & Engineering, Univ. of California, San Diego, La Jolla, CA 92093
[2]Altera Corp., San Jose, CA 95134, [3]Qualcomm Inc., San Diego, CA 92121
[4]Dept. Information Systems Engineering, Osaka University, Osaka, 565-0871 Japan
[1,2,3]{lizhang,hazhu,jhliu,ckcheng}@cs.ucsd.edu, [4]hasimoto@ist.osaka-u.ac.jp

Abstract— **This paper presents a new logic style, named Current-Mode Differential logic (CMDL), that achieves both high operating speed and low power consumption. Inspired by the low-voltage swing (LVS) logic, CMDL uses a shunt resistor at the differential output to obtain constant low swing signal without the need to reset low. Furthermore, conditional shunt transistors are used for the internal nodes to prevent high-voltage swing, thus entirely eliminate the power-hungry clocked reset network in LVS circuits. We show that the CMDL is suitable for high-end microprocessor integer core by providing three datapath modules implemented in CMDL. Our simulation results indicate that, operating at comparable speed with LVS logic, CMDL circuits can achieve up to 50% reduction of delay-power product compared to CMOS logic and LVS logic. In addition, CMDL reduces the power consumption of LVS by up to 40%.**

I. Introduction

With the continuous scaling of semiconductor technology, high performance circuit design has become increasingly difficult. As clock frequency increases, clock skew, pipeline overhead, large wire delay are eating a significant portion of the cycle time, leaving very stringent room for logic operations. This scenario is especially true for designing high-end microprocessor integer cores, which are required to operate at twice of the system clock rate [1], [2]. Moreover, high clock frequency imposes great challenges on power budgeting, as power increases linearly with the clock rate.

Various differential logic styles were proposed in the last 20 years aiming at achieving low latency, most of which fell into the category of voltage-mode logic style. One well-known approach is the Cascode Voltage Switch Logic (CVSL) developed by [7]. It is composed of a NMOS logic tree, which has complemental inputs on transistor gates, and a load circuit to provide fast pull-up and output restoration. Robustness and power improvement were made by [9]–[11].

Another approach is the Complemntary Pass-Transistor Logic (CPL) described in [12], which has both complementary gate inputs and complementary drain inputs at the same time. Optional PMOS latch could be added to reduce power consumption.

A more recent example is the Low-Voltage Swing (LVS) logic used in Intel Pentium 4 processor. Instead of taking full swings along the logic path, LVS logic takes advantage of differential small signals as propagate through a *large Diffusion Connected Network (DCN)*. The low-swing output voltage is then restored to full-voltage swing by a sense-amplifier. Similarly to CVSL, LVS logic requires a reset operation to clean the charges of all the internal nodes of the DCN before each cycle of evaluation. Without a reset network, the DCN output may reach full-voltage swing if the input values keep unchanged for a few clock cycles. This full-swing output will in turn destroy the evaluation in the following cycle, necessitating the reset logic. Fig. 2 shows a LVS circuit block with reset logic. The four NMOS transistors in the middle are functional DCN. The rest nine NMOS transistors controlled by the clock form the reset logic. The overhead associated with the reset logic is significant: 1) The number of transistors doubles; 2) The load on clock increases; 3) One extra stage of logic (Thru-gate) is added to DCN; 4) A reset phase is inserted to each clock cycle. All the above overhead greatly raises the power consumption of LVS logic style.

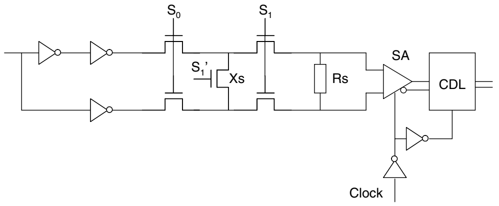

Fig. 1. CMDL circuit block diagram.

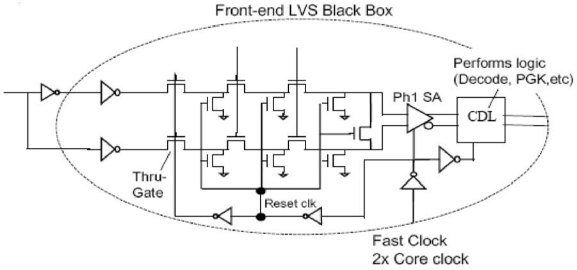

Fig. 2. LVS circuit block diagram [2].

Allam and Elmasry [13] proposed a current-mode logic (DyCML) using a dynamic logic style. They used current source for two differential paths, and the output logic is based on the current difference between the two branches. The dynamic logic has pre-charge and evaluation stages. The cross-coupled PMOSes pull-up the outputs to be full-swing signal. To reduce the power consumption, they introduced a virtual ground to eliminate the static current.

In this paper we propose a new logic style, *Current-Mode Differential Logic (CMDL)*. As illustrated in Fig. 1, CMDL uses a shunt resistor to connect the two rails of each differential output. Therefore, there is always current flowing through the shunt resistor, in one way or another, giving rise to the name *current mode*. CMDL uses low-voltage swing at the differential output, which enhances the operation speed and saves power as well. In addition, unlike LVS, CMDL eliminates the need of clocked reset networks and has no reset stage. Our simulation results indicate that the CMDL can achieve better delay-power product compared to LVS and CMOS standard cell. Furthermore, CMDL has high noise immunity because the noise is reduced according to the ratio of shunt resistor value and equivalent loop resistance value.

The rest of the paper is organized as follows. We first discuss the basic concepts of CMDL in Section II. The generalized Elmore delay is presented in Section III. Section IV shows three example core designs using CMDL, namely, a 32-bit alignment MUX, a 16-bit carry-skip adder, and a 8-bit shifter/rotator. Experimental results are listed and discussed in Section V. Section VI concludes the paper.

II. Basic concepts of CMDL

In this section, we present the motivation behind CMDL first, and explain the structure and advantages. We show some CMDL examples at the end.

978-1-4244-1922-7/08/$25.00 ©2008 IEEE

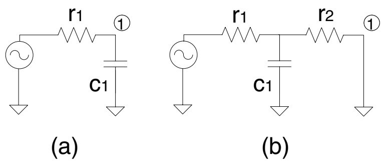

Fig. 3. Single branch RC trees: (a) without shunt resistor; (b) with shunt resistor.

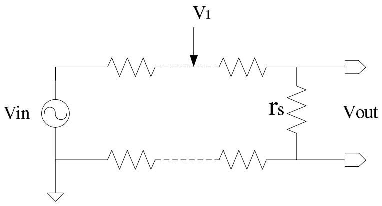

Fig. 4. Noise immunity of CMDL

A. Motivation

The CMDL distinguishes itself from other previous work by using a shunt resistor at the end of output. There are several reasons for using the shunt: 1) To maintain the output low swing, 2) To improve the circuit response time, 3) To reduce the noise effect.

For voltage-mode logic to maintain a low-swing output, the reset or pre-charge network is most commonly adopted. It not only increases the activity of the DCN, but also adds significant load to the clock tree, which is a full-swing net switching in every cycle. In contrast, current mode logic inherently enables low swing operation without extra overhead. To demonstrate the concept, Fig. 3 shows two single branch RC trees. Without shunt resistor (Fig. 3(a)), node 1 will eventually reach supply voltage in an enough long period. However, if a shunt resistor is added (Fig. 3(b)), the output voltage at node 1 will be capped at low voltage determined by the ration of r_2 and r_1, and a constant current is flowing through r_1 and r_2.

In addition to enabling low-voltage swing, adding shunt resistors also improves the response time of the circuit. In this example, the time constant of the circuit in Fig. 3(a) is simply r_1c_1, while the time constant of the circuit in Fig. 3(b) is $\frac{r_2}{r_1+r_2}r_1c_1$.

Fig. 4 illustrates the concept of how current-mode logic reduces the noise effect. When the DCN transistors of a path are switched to ON, the input-output propagation in steady state can be modeled as a resistor loop. Assume the total resistance of the loop is r_{total}, and the shunt resistance is r_s, the nominal value of differential output is $V_{out} = \frac{r_s}{r_{total}}V_{in}$. If an internal node has a noise, which makes the voltage change from V_1 to $(V_1+\Delta V)$, the change of the current on this loop will be $\Delta I = \frac{\Delta V}{r_{total}}$. As a result, the change of output voltage will be $\Delta V_{out} = r_s \cdot \Delta I = \frac{r_s}{r_{total}} \cdot \Delta V$, which means the noise is reduced.

B. Structure and advantages of CMDL

Fig. 5 gives a list of one to one mapping between basic CMDL and LVS logic components. The first stage of LVS logic is the thru-gate controlled by clock, which serves as an interface between static input data and clock-gated signals, while the first stage of CMDL implements functionality. Middle stage of LVS logic contains two clock controlled reset transistors, while the CMDL only uses one shunt transistor controlled by the complement of control signal. The last stage of LVS logic consists three transistors, while CMD logic only uses a resistor to convert the current-mode signal back to the voltage-mode signal, which can be restored by the sense-amplifier.

Similar to LVS logic, the CMDL has some design constraints on the DCN: 1) The DCN can have multiple inputs and multiple outputs; 2) Any path from the input to output has at most six stages of logic; 3) The differential signal generated at each output must be greater than 0.1V at the end of each evaluation stage (VCC=1.0V).

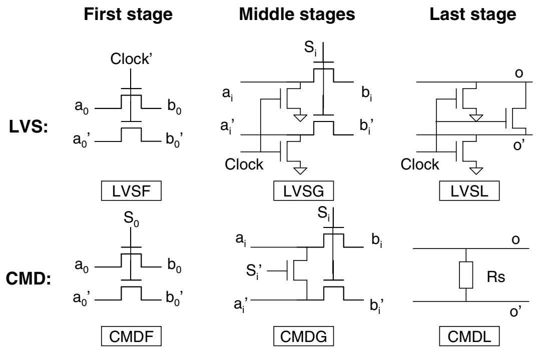

Fig. 5. Basic LVS and CMDL gates.

Since the CMDL does not have reset logic, low voltage swing on each internal node has to be maintained by shunt resistor/transistor. Hence two more design rules are required: 1) For any input stimulus, the differential inputs must be connected through a shunt resistor or a closed transistor; 2) For each pair of differential output, there shall be no other shunt resistors or closed transistors on the active path. The first rule guarantees that there will not be full-voltage swing on any internal node, while the second rule enables the 0.1V differential signal at the outputs.

The first advantage of CMDL is less number of transistors. In the worst case, one third of the total transistors are shunt transistors, while in LVS logic the reset network could account for half the total transistors. The second advantage, compared to LVS logic, is more headroom of the logic depth. The first stage (i.e. Thru-gate stage) in LVS logic serves to control the switching between evaluation phase and reset phase, and this stage is removed in CMDL. Thus CMDL can achieve one more logic depth than LVS logic. Thirdly and very importantly, CMDL achieves low power consumption because the clocked reset network is no longer needed. Note the reset network in LVS logic has to toggle as a whole in every clock cycle, regardless of the input switching patterns, which is clearly power inefficient. On the contrary, the switching activity in CMDL only depends on the toggle rate of the inputs. The power consumption of CMDL is self-adjusted according to the input toggle rate. Finally, as mentioned in Section II-A, CMDL has the ability of reducing noise effect and hence enhances the circuit reliability.

C. CMDL examples

Fig. 6 presents the block diagrams and circuit schematic of 2-to-1 multiplexer, NAND, NOR and XOR gates implemented in CMDL. In each instance, the two differential path are controlled by complement signals, which guarantees that one and only one path is active at one time.

Fig. 7 gives the schematic and block diagram of a 4-to-1 MUX in CMDL logic, which consists of two CMDL stages. Two types of CMDL gates are used in the design. The CMDF gates are used in the first stage, while the CMDG gates are used in the second stage. The conditional shunt transistors in the CMDG gates are necessary to maintain low-voltage swing at the internal nodes. For example, when S_1 is 0, the differential node pair b_0/b_0' is disconnected from the differential output out/out'. The shunt transistor controlled by S_1' turns on to keep the source region of the series transistors in low-swing state. Finally, at the output of the 4:1 MUX, a shunt resistor is added to maintain the low-voltage swing on the active path.

Fig. 8 shows a 2-bit ripple carry adder using CMDL. The 2-bit adder contains two full adders in series, whose primary inputs are carry-in, carry-propagation signal P_i, carry-generation signal G_i and carry kill signal K_i. Since P_i, G_i, K_i can not be zero simultaneously, there is

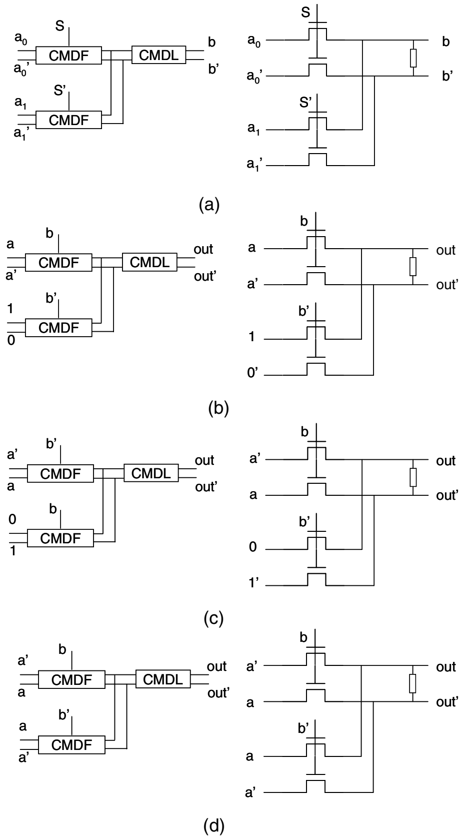

Fig. 6. small examples of CMDL: (a) 2 to 1 MUX (b) 2-input NAND gate (c) 2-input NOR gate (d) 2-input XOR gate

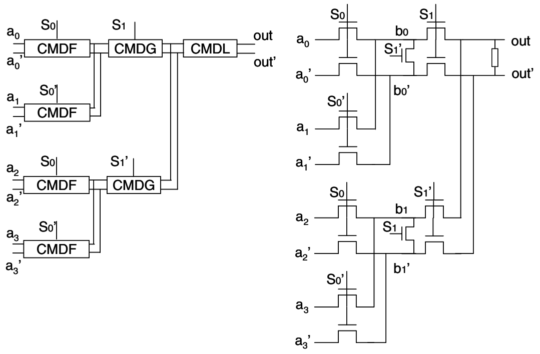

Fig. 7. A 4 to 1 MUX in CMDL

always a current through the shunt resistor at C_{out} and C'_{out}. However, the path for generating sum output is different from carry-out, since all the sum output are driven by the carry-in signal. If every sum output pair has a static resistor (i.e. the CMDL component), there will be multiple shunt resistors along the path from carry-in to carry-out, which violates our design rule. To address this issue, we use a controlled shunt resistor (i.e. the CMDCL component) at each sum output. The transistor in series with the resistor is controlled by the complement of carry-propagation signal P'_i, which enables the shunt only when the P_i signal is zero, and that means the path starting from carry-in terminates at S_i and S'_i.

III. Generalized Elmore Delay

In voltage mode logic, the time constant is calculated by the conventional Elmore delay, which assumes that there is no DC path to the ground. For the RC ladder circuit shown in Fig. 9, the time constant for node i as predicted by the well-known Elmore delay formula [3] is given by:

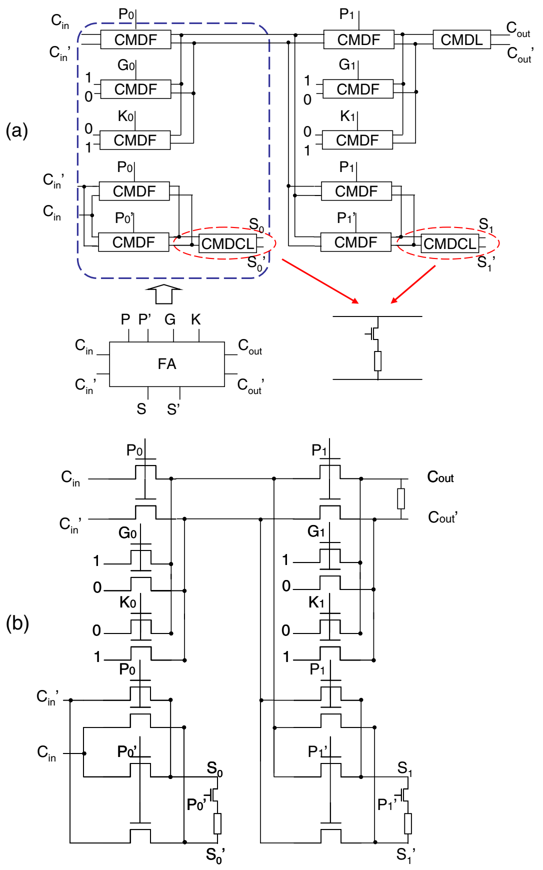

Fig. 8. A 2-bit ripple carry adder in CMDL

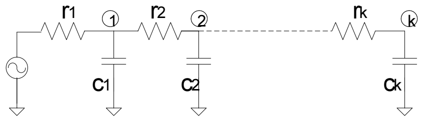

Fig. 9. RC ladder without shunt resistors to ground.

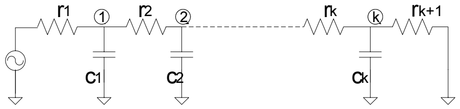

Fig. 10. RC ladder with shunt resistors to ground.

$$\tau_i = \sum_{j=1}^{i} r_j \sum_{l=j}^{k} c_l \qquad (1)$$

which, however, is not applicable to the case with shunt resistors.

To calculate the time constant for current mode logic, we employ the generalized first-order time constant as defined in the following equation [4]:

$$\tau_i = \frac{1}{V_i(\infty) - V_i(0^+)} \int_{0^+}^{\infty} (V_i(\infty) - V_i(t))dt \qquad (2)$$

Fig. 10 presents a RC ladder in current mode logic. For any pair of nodes i and j, assuming $i \le j$, we define their common path resistance and common shunt resistance as follows:

$$R_{i,j}^{\text{path}} = \sum_{t=1}^{i} r_t \qquad (3)$$

$$R_{i,j}^{\text{shunt}} = \sum_{t=j}^{k} r_{t+1} \qquad (4)$$

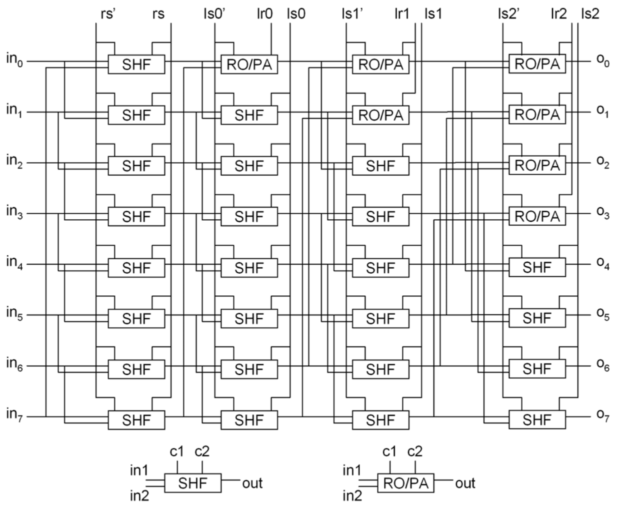

Fig. 11. An 8-bit barrel shifter/rotator.

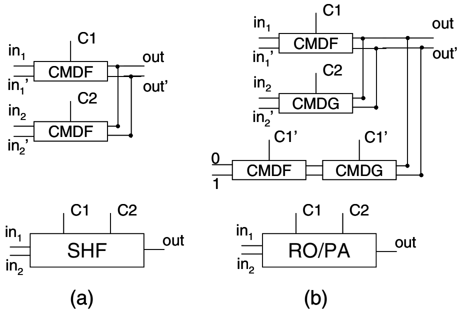

Fig. 12. (a) SHF cell; and (b) RO/PA cell.

TABLE I
FUNCTION TABLE OF THE RO/PA CELL

Operation	c1	c2	out
No shift	1	0	in1
Rotate	0	1	in2
Padding 0	0	0	0

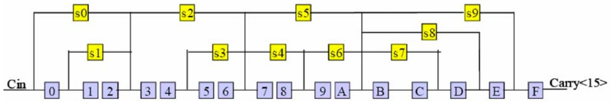

Fig. 13. A 16-bit carry-skip adder [2].

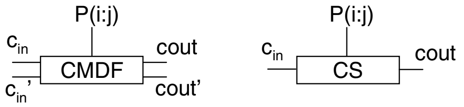

Fig. 14. Carry-skip (CS) cell

where k is the last node before shunt. For any node i, the time constant as defined in equation (2) can be derived as

$$\tau_i = \frac{\sum_{j=1}^{k}(c_j R_{i,j}^{\text{path}} R_{i,j}^{\text{shunt}} \Delta V_j)}{\Delta V_i \sum_{j=1}^{k+1} r_j} \qquad (5)$$

where $\Delta V_i = V_i(\infty) - V_i(0^+)$ is the initial voltage on node i.

IV. DESIGN EXAMPLES USING CMDL

In this section, we present three CMDL circuit designs that can be used in the microprocessor integer cores. Following the LVS circuits proposed in [2], the arithmetic/logic blocks implemented by CMDL include a 32:1 alignment multiplexer, a 8-bit rotator/shifter and a 16-bit carry-skip adder.

A. Alignment MUX

Alignment MUX is used to fetch data from L0-cache to integer core. According the integer core of Intel Pentium 4 processor, we implement 32:1 alignment MUX by CMDL.

To build the 32:1 alignment MUX, we use the 4:1 MUX shown in Section II-C as the building block (Fig. 7). We then build a 16:1 MUX using five 4:1 MUXes, and the 32:1 MUX is in turn constructed by two 16:1 MUXes and one 2:1 MUX. Therefore, there are five levels of 2:1 MUX in total. The first level contains CMDF gates only, while CMDG gates are used for level 2 to level 5. One shunt resistor is placed at the primary output.

To analyze the functionality of the 32:1 MUX, we consider all the paths from a differential input to the differential output, each path consisting of five logic stages. For any combination of the select signals, one of the 32 differential inputs is passed to the output, and all the other differential inputs are killed by a shunt transistor. Therefore, this design satisfies the CMDL design rules.

B. Rotator/Shifter

The 8-bit rotator/shifter adopts the barrel shifter structure proposed in [5], as shown in Fig. 11. This structure can left rotate or shift the operand by 0 to 7 bits. There are two kinds of basic cells in this shifter network: SHF and RO/PA. The SHF cell can be considered as a 2:1 MUX, since the two control signals to SHF are always complementary. The RO/PA cell has three operation modes, which are listed in Table I.

When the unit performs rotate or no-shift operation, each differential input will drive one output. Thus, no shunt transistor is needed for SHF cells. On the other hand, when the unit performs logic shift, some outputs may be padded with constant zeros, and this shall always happen in the RO/PA cells. Therefore, shunt transistors are necessary in RO/PA cells. Fig. 12 shows the SHF and RO/PA cells implemented in CMD logic. It can be observed that the SHF cell contains only one logic stage, while the RO/PA cell either passes a differential input to output in one logic stage, or produces a zero at the output through two logic stages.

The longest path of the rotator/shifter goes through four levels of SHF or RO/PA cells. There is no path longer than five logic stages. Each differential input is either shifted to an output, or removed by the CMDG gate within a RO/PA cell at some point. Thus the rotator/shifter design satisfies the CMD design rules.

C. Carry-Skip Adder

Fig. 13 shows the topology of the 16-bit carry-skip adder. This topology guarantees that any path from input to output consists of less than six cells. There are two kinds of cells: full adder (FA) cell and carry-skip (CS) cell. Similar to the 2-bit ripple carry adder mentioned in Section II, the carry propagation signal P_i, carry generation signal G_i, carry kill signal K_i, and carry-skip control signals $P(i:j)$ are all produced from outside. The CMDL FA and CS cells use these signals directly as primary inputs. The function of the CS cell is to simply pass the carry-in to the carry-out when the carry-skip control $P(i:j)$ is 1. The function of the FA cell is to generate the carry-out and sum signals at each bit position. Due to the XOR operation, the sum bit is always driven by the carry-in signal, while the carry-out is either driven by the carry-in, or produced by the local voltage sources.

Fig. 14 illustrates the CS cells designed in CMDL, and the FA cell is shown in Fig. 8. As explained in Section II, the shunt transistors are not used in this design, and a CMDCL gate is placed at each sum output, and transistor in CMDCL gate at S_i is controlled by complement of the signal P_i. It can be proved that if P_i is 0, $P(i:j)$ is 0 for any j. This guarantees that when a controlled shunt resistor is enabled, no other shunt resistors will present on this path. For Carry$_{15}$, because it

TABLE II

PERFORMANCE COMPARISON OF CMOS, CMDL AND LVS CIRCUITS

	32-bit MUX			**8-bit Shifter**			**16-bit Adder**		
	CMOS	LVS	CMDL	CMOS	LVS	CMDL	CMOS	LVS	CMDL
Cycle time(ps)	200	215	180	200	210	180	800	350	380
Delay(ps)	195.6	153.8	118.7	165.3	148.4	120.7	709.6	251.5	286.6
Norm. Delay	1.00	0.79	0.61	1.00	0.90	0.73	1.00	0.35	0.40
Avg/Peak power(mW)	0.38/5.69	0.45/3.63	0.38/3.10	0.36/2.81	0.48/3.23	0.41/2.34	0.26/2.37	0.53/6.29	0.32/2.70
Norm. Avg Power	1.00	1.18	1.00	1.00	1.33	1.14	1.00	2.04	1.23
Delay × Power(fJ)	74.33	69.21	45.11	59.51	71.23	49.49	184.50	133.30	91.71
Norm. Delay × Power	1.00	0.93	0.61	1.00	1.20	0.83	1.00	0.72	0.50
Delay$^2\times$ Power(pJ$\times$ps)	14.54	10.64	5.35	9.84	10.57	5.97	130.9	33.5	26.28
Norm. Delay$^2\times$ Power	1.00	0.73	0.37	1.00	1.07	0.61	1.00	0.26	0.20
Input Power(mW)	0.15	0.17	0.16	0.08	0.09	0.09	0.03	0.04	0.04
Load Power(mw)	0.004	0.001	0.004	0.01	0.04	0.04	0	0.04	0.04
Sense Amp Power(mW)	-	0.002	0.002	-	0.12	0.09	-	0.17	0.13
Logic Power(mW)	0.23	0.28	0.21	0.27	0.23	0.20	0.23	0.28	0.11
Total Transistor Count	312	322	162	392	316	226	393	450	315
Transistor Overhead	0	145.8%	23.7%	0	49.1%	6.6%	0	50.5%	5.4%

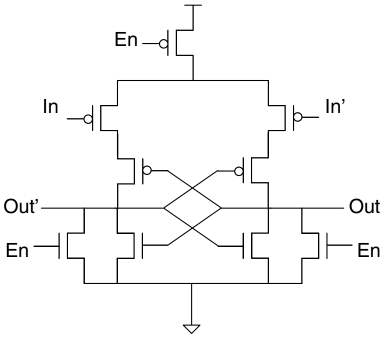

Fig. 15. sense amplifier

is the last CMD stage, a static shunt resistor is placed here.

D. Sense Amplifier

The sense amplifier is needed for both LVS logic and CMDL to restore the small differential output signals, and Fig. 15 shows the schematic of sense amplifier adopted in our experiment. When En signal is high, the sense amp outputs are pre-charged to low. The differential input should be ready before the falling edge of En signal, which triggers the sense amp. The cross-coupled PMOS pair and NMOS pair provide positive feed-back loop so that the full-swing differential outputs reach the steady state with one high and one low very quickly.

V. SIMULATION RESULTS

To quantify the performance and power consumption of the proposed technique, we construct the net-list of the 32-bit alignment MUX, 8-bit rotator/shifter and the 16-bit carry-skip adder in TSMC 90nm technology using CMDL and LVS logic. We also use standard cell library of TSMC90nm as basic building blocks to construct the net-list of the CMOS version of these three designs. For the 32-bit MUX and 8-bit rotator/shifter, we use only minimum size NMOS transistors in CMDL circuits, but use 6x NMOS transistors in the first stage of the LVS circuits. In the 16-bit carry-skip adder, gate sizing is performed, and is the same for both CMDL and LVS circuits. In any case, the reset logic in LVS circuits is composed of minimum size NMOS transistors only. We use the CMOS inverters with size 4 to drive all the data inputs and control inputs, and the logic outputs or sense amp outputs are loaded with CMOS inverter with minimum size. For CMOS logic, the loads directly connect to logic outputs, and for LVS and CMDL, the loads connect to sense amp outputs. The schematic of sense amp for LVS and CMDL is shown in Fig. 15. Since differential inputs and outputs are used in CMDL and LVS logic, the number of input inverters and load inverters used in these two logics are twice as that in CMOS logic.

The cycle time for each logic is determined by the circuit worst case delay from the inputs to the outputs. For LVS and CMDL, the high voltage of sense amp is greater than 0.8v under the chosen cycle time for better signal quality. To measure the power dissipations, we use Hspice simulation as evaluation tool, and 100 randomly generated input patterns are fed into each circuit.

Table II compares the Hspice simulation results of CMOS, LVS and CMDL circuits in terms of delay, average/peak power, delay-power product and delay^2-power product. The absolute values are listed in Row 4, 6, 8 and 10, and the normalized values are given in Row 5, 7, 9 and 11. We list in percentage the input power (the power consumed by input inverters), the load power (the power consumed by load inverters), the sense amp power, and the logic power (the power consumed by logic circuits) in Row 12 to Row 15. For LVS logic, the power of reset network is included in the logic power. The transistor count is also compared (Row 16 and 17). The transistor overhead is calculated as the ratio of extra transistor count to the summation of DCN transistor count and sense-amplifier transistor count. Extra transistors are defined as the reset network and the first stage of the DCN network (thru gates) of LVS, and the conditional shunt transistors for CMDL. Transistor count due to input and load inverters are not included. For instance, the CMOS 32-bit Multiplexer can operate at 200ps cycle time, and has a delay of 195.6ps. The average total power is 0.38mW, and the peak total power is 5.69mW. The delay-power product is 74.33fJ, and the delay^2-power product is 14.54pJ$\times$ps. The input power, load power and logic power are 0.15mW, 0.004mW and 0.23mW The total transistor count is 312, and the overhead is zero since there are no transistors for reset network, thru-gates or shunt transistors.

In terms of the metrics of delay-power product and delay^2-power product, CMDL is better than CMOS and LVS. Compared to the other two logics, CMDL reduces the delay-power product by up to 50% (in the case of adder). CMDL also reduces the delay^2-power product of CMOS by up to 80%, and reduces the delay^2-power product of LVS by up to 49%.

The second advantage of CMDL is the high operating speed. The speed of CMDL is faster than CMOS because it adopts the DCN network and uses differential small signals. The speed of CMDL is comparable to LVS in the multiplexer and shifter cases, and is slower in adder case by 9% because with the elimination of reset stage, the differential output needs to be charged from the opposite voltage level instead of zero.

The third advantage of CMDL is the significant power efficiency compared to LVS logic: the total power savings for multiplexer, shifter and adder are 15%, 14% and 40% respectively. Since CMDL requires

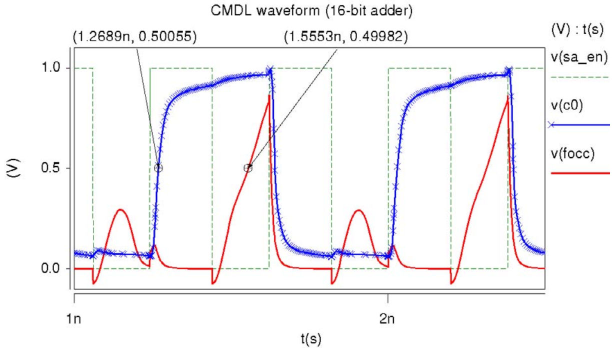

Fig. 16. Simulation waveforms of the CMDL 16-bit carry-skip adder.

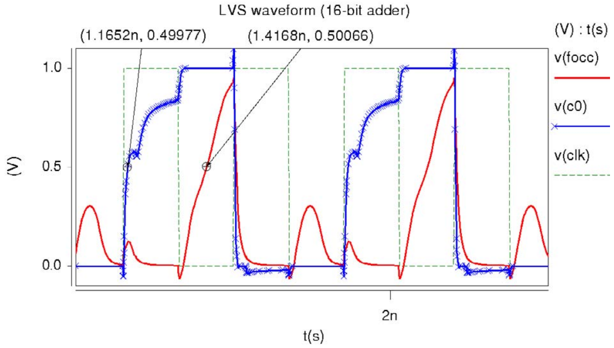

Fig. 17. Simulation waveforms of the LVS 16-bit carry-skip adder.

the same number of input inverters, load inverters and sense amp as LVS logic, the power reduction is mainly comes from the logic power saving, as can be observed from Row 12 to Row 15. The primary reason for the improvement is that CMDL does not use reset networks and therefore reduce the switching of internal nodes and extra transistor counts, which can be seen from the last row.

We can also see from Table II that CMDL dissipates more power than CMOS logic: the power consumption of shifter and adder increase by 14% and 23%. This is understandable since there is a static current flow in CMDL which introduces static power consumption. Another reason for higher power consumption of CMDL is more overhead introduced by complement inputs, loads and sense amp. For instance, the 32-bit MUX in CMOS logic needs 37 input inverters, 1 load inverters and no sense amp, while the 32-bit MUX in CMDL needs 74 input inverters, 2 load inverters and 1 sense amp. Therefore, we can see from Table II that the input power of CMDL is larger than CMOS, the load power and sense amp power are very small and can be ignored, and actually the logic power of CMDL is smaller than that of CMOS. The 8-bit shifter/16-bit adder in CMOS logic needs 15/59 input inverters, 8/17 load inverters, while the CMDL version shifter needs 30/72 input inverters, 16/34 load inverters and 8/17 sense amps, which consume more than half of the total power. The logic power of CMDL shifter/adder is 0.20mW/0.11mW, and the CMOS shifter/adder has a logic power of 0.27mW/0.23mW.

Fig. 16-18 shows the Hspice simulation waveform of 16-bit carry-skip adder implemented in CMDL, LVS and CMOS logic. For each logic style, the carry-in, carry-out and clock/enable signals are shown, and the input and output arrival times are labelled as well.

VI. Conclusions and future work

We proposed a current-mode differential logic (CMDL) style. By adding shunt resistors and transistors, CMDL removes the large reset network in LVS logic. Low-voltage swing in CMDL is maintained by the static current through the output shunt resistor. We demonstrated the effectiveness of CMDL by three arithmetic/logic block designs

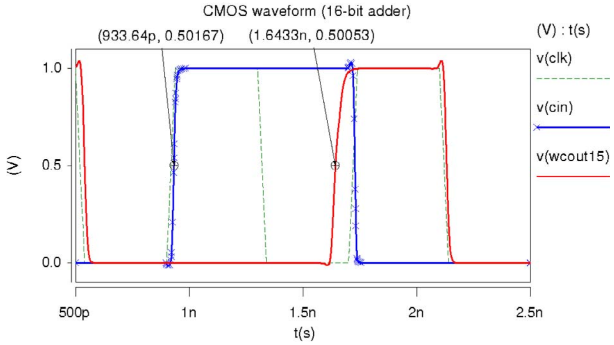

Fig. 18. Simulation waveforms of the CMOS 16-bit carry-skip adder.

including a 32-bit alignment MUX, a 8-bit barrel shifter and a 16-bit carry skip adder. Compared with CMOS and LVS implementations, CMDL achieves much better delay-power product and delay^2-power product. Future work includes the detailed experiments of the energy overhead of CMDL on small circuits, noise immunity and technology scaling. Fabrication is also expected.

VII. Acknowledgement

The authors would like to acknowledge the support of NSF CCF-0618163 and California MICRO Program.

References

[1] D. J. Deleganes, M. Barany, G. Geannopoulo, et al., "Low-Voltage swing logic circuits for a pentium 4 processor integer core," *IEEE Journal of Solid-State Circuits (JSSC)*, Vol. 40, No. 1, pp. 36-43, Jan. 2005.

[2] D. J. Deleganes, M. Barany, G. Geannopoulo, K. Kreitzer, A. Singh, S. Wijeratne, "Low-Voltage-Swing logic circuits for a 7Ghz x86 integer core," in *Digest of Technical Papers, IEEE Int. Solid-State Circuits Conference (ISSCC)*, pp. 154-163, Feb. 2004.

[3] J. Rubinstein, P. Penfield, M. Horowitz, "Signal delay in RC tree networks," *IEEE Transaction on CAD*, Vol. 2, No. 3, pp. 202-211, Jul. 1983.

[4] T. M Lin, C. A. Mead, "Signal delay in general RC networks," *IEEE Transactions on CAD*, Vol. 3, No. 4, pp. 331-349, Oct. 1984.

[5] R. Pereira, J. Mitchell, J. Solana, "Fully pipelined TSPC barrel shifter for high speed applications," *IEEE Journal of Solid-State Circuits*, Vol. 30, No. 6, pp. 686-690, Jun. 1995.

[6] R. Zimmermann, W. Fichtner "Low-power logic styles: CMOS versus pass-transistor logic," *IEEE J. Solid-State Circuits*, Vol. 32, No. 7, pp. 1079-1090, Jul, 1997.

[7] L.G. Heller, W.R.Griffin, "Cascode voltage switch logic: A differential CMOS logic family," *IEEE Int'l Solid-State Circuits Conf. Dig. of Tech. Paper* pp. 16-17. 1984.

[8] K.M. Chu, D.L. Pulfrey, "A comparison of CMOS circuit techniques: Differential cascode voltage switch logic versus conventinal logic," *IEEE J. Solid-State Circuits*, Vol. SC-20, No. 4, pp.528-532, Aug. 1987.

[9] A.J. Acosta, M. Valencia, A. Barriga, M.J. Bellido, J.L. Huertas, "SODS: A new CMOS differential-type structure," *IEEE J. Solid-State Circuits*, Vol. 30, No. 7, pp. 835-838, Jul, 1995.

[10] D. Somasekhar, K. Roy, "Differential current switch logic: A low power DCVS logic family," *IEEE J. Solid-State Circuits*, Vol. 31, No. 7, pp.981-991, Jul, 1996.

[11] J. Park, J. Lee W. Kim, "Current sensing differential logic: A CMOS logic for high reliability and flexibility," *IEEE J. Solid-State Circuits*, Vol. 34, No. 6, pp.904-908, Jun, 1999.

[12] K. Yano, T. Yamanaka, T. Nishida, M. Saito, K. Shimohigashi, A. Shimizu, "A 3.8-ns CMOS 16x16-b multiplier using compementary pass-transistor logic,",*IEEE J. Solid-State Circuits*,Vol. 25, No. 2, pp. 388-395, Apr. 1990.

[13] M. W. Allam, M. I. Elmasry, "Dynamic current mode logic (DyCML): A new low-power high-perforamnce logic style," *IEEE J. Solid-State Circuits*, Vol. 36, No. 3, pp. 550-558, Mar. 2001