Low-Power Carry-Select Adder Using Adaptive Supply Voltage Based on Input Vector Patterns

Hiroaki Suzuki[†], Woopyo Jeong, and Kaushik Roy

Department of Electrical and Computer Engineering,
Purdue University, West Lafayette, IN 47907, USA
{jeongw, kaushik}@ecn.purdue.edu

[†]Renesas Technology Corporation
Itami, Hyogo 664-0005, Japan
suzuki.hiroaki@renesas.com

ABSTRACT

Demands for the low power VLSI have been pushing the aggressive design methodologies to reduce the power consumption drastically. To meet the growing demand, we propose Adaptive Supply Voltage Carry-Select Adder (CSA) based on the input vector patterns. A proposed level converter based on the Complementary Pass Transistor Logic (CPL) cancels out the delay penalty of level conversion. We achieved 26% power improvement on a 128-bit CSA prototype over a conventional design with same performance.

Categories and Subject Descriptors

B.7.1 [**Integrated Circuits**]: Types and Design Styles

General Terms

Performance, Design

Keywords

Low Power Adder, Adaptive Supply Voltage, Carry-Select Adder

1. INTRODUCTION

Increasing demand for mobile electronic devices such as cellular phones, Personal Digital Assistances (PDA), and laptop computers requires the use of power efficient VLSI circuits. The power consumption of a CMOS digital circuit can be represented as follows.

$$P = \alpha{\cdot}f{\cdot}C{\cdot}VDD^2 + \alpha{\cdot}f{\cdot}I_{short}{\cdot}\ VDD + I_{leak}{\cdot}\ VDD, \quad \text{............}(1)$$

where α is the activity factor of the circuit, f is the clock frequency, C is the average switched capacitance per clock cycle, VDD is the supply voltage, I_{short} is the short circuit current, and I_{leak} is the off current [1]. Using a lower VDD is an effective way to reduce the dynamic power consumption because all terms of equation (1) strongly depend on VDD. However, the drawback of using a lower VDD is the performance degradation. Adaptive-VDD and Multiple-VDD techniques can reduce the power consumption without performance degradation [2-7].

DSP (Digital Signal Processing) chips and ALUs (Arithmetic Logic Unit) are most commonly used components of mobile products requiring low power dissipation. Addition is the key operation in those components with respect to performance and power consumption. Hence, it is essential to use low power adders for achieving low energy dissipation. We proposed Adaptive-VDD RCA to meet such demands [2]. Depending on the carry propagation length, it determines the supply voltage in every cycle. However, since the Adaptive-VDD RCA is slow, it is not suitable for chip design requiring low power consumption with high performance.

In this paper, we propose Adaptive-VDD Carry-Select Adder (CSA) suitable for higher performance applications. Section 2 explains the basic design methodology for Adaptive-VDD RCA [2]. In Section 3 we implement prototype circuits of 32-bit, 64-bit and 128-bit Adaptive-VDD CSA's, and discuss the CSA optimization procedures and simulation results in detail. We also proposed a new level converter using the Complimentary Pass Transistor Logic (CPL) [8]. This circuit converts the low VDD signals into the standard CMOS output level (VDD) without any delay penalty.

2. SUPPLY VOLTAGE CONTROL BASED ON INPUT VECTOR PATTERN

Our approach utilizes the delay differences depending on the input patterns. A typical example is a Ripple Carry Adder (RCA), which serially propagates the carry signal through the entire bit of the operand width. The serial carry chain is activated by the "propagate" condition, which is $P[i] = A[i] \oplus B[i]$, where i=0, 1, …, n-1 for an n-bit adder. When $P[i]$ = "0", the carry propagation path along the critical path of an adder is divided into two parts: carry propagations start at the 1st stage and the ith stage simultaneously. Hence, in this case, the delay is determined as the longest delay between two carry propagation delays. For example, if P[9] = "0" in a 16-bit RCA, the critical path of the adder is as long as that of a 9-bit adder because the delay of an RCA is determined by the carry propagation length. At a certain lower supply voltage, the delay can be the same as that of the worst delay of 16-bit RCA with VDD. The Adaptive-VDD RCA utilizes this advantage of the delay decrease due to the split critical paths. The mid-points are monitored to adaptively apply the low supply voltage as required.

Fig. 1 shows a block diagram of the 16-bit Adaptive Supply Voltage RCA, which is one of the sub-blocks of Adaptive Supply Voltage CSA. The VDD Control circuit (VDD Cont) block turns on one of the PMOS switches that supply the VDD or VDDlow to the Virtual VDD (VVDD) of the target 16-bit RCA. If P[9:6] are all high, SW0 turns on, and supplies VDD level to VVDD node. Otherwise, SW1 supplies VDDlow. At a certain low supply voltage (VDDlow), the worst delay of the split carry propagation paths with VDDlow is the same as that of the 16-bit adder with standard VDD. Since VDDlow is expected to be much lower than VDD, the circuit consumes power less than the conventional RCA that uses a fixed value of VDD. To minimize the delay due to the control circuit, it is operated at the fixed VDD. In addition, N-wells are connected to VDD to reduce the total capacitance of the VVDD node.

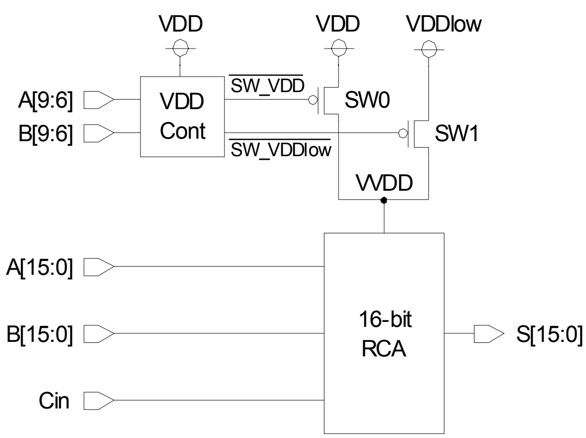


Fig. 1. Block diagram of the 16-bit Ripple Carry Adder (RCA) with adaptively controlled supply voltage (the 16-bit Adaptive-VDD RCA).

3. CARRY SELECT ADDER WITH ADAPTIVE VDD

We can extend the Adaptive VDD technique applied for RCA to Carry Select Adder (CSA) because a CSA is composed of the RCA sub-blocks. Fig. 2 shows a block diagram of the 32-bit CSA. Here, the VDD Cont blocks and PMOS switches are not shown to simplify the diagram. In a CSA, the control circuit overhead of the proposed method can be reduced to half of that of RCA because the carry-one and the carry-zero blocks of the carry selection can share one VDD control circuit. On the other hand, the 32-bit carry chain is split into several parts with the smaller size adders towards the LSB. We used conventional RCA for the sub-blocks that are smaller than 6-bit because simulation results show no advantage of using the Adaptive VDD technique for such small-sized adders. For example, in Fig. 2, Block 0 and Block 1 consist of conventional 4-bit RCA's.

In the Adaptive-VDD RCA's used for sub-blocks of CSA, delay penalty due to charging/discharging VVDD nodes of each RCA can become significant. To compensate this delay penalty we used the Carry-Skip technique in Section 3.1 for the worst delay propagation path of each RCA. The Adaptive-VDD CSA also needs converting a signal level from VDDlow to VDD at the interface between a multiplexer (MUX) and an RCA block. Otherwise, a MUX consumes short-circuit current during VDDlow operation. The level converter would have a delay overhead. To avoid this delay penalty, we propose Complimentary Pass Transistor Logic (CPL) MUX with dual VDD in Sections 3.2. The conventional CSA is assumed to use the CMOS transmission gate MUX.

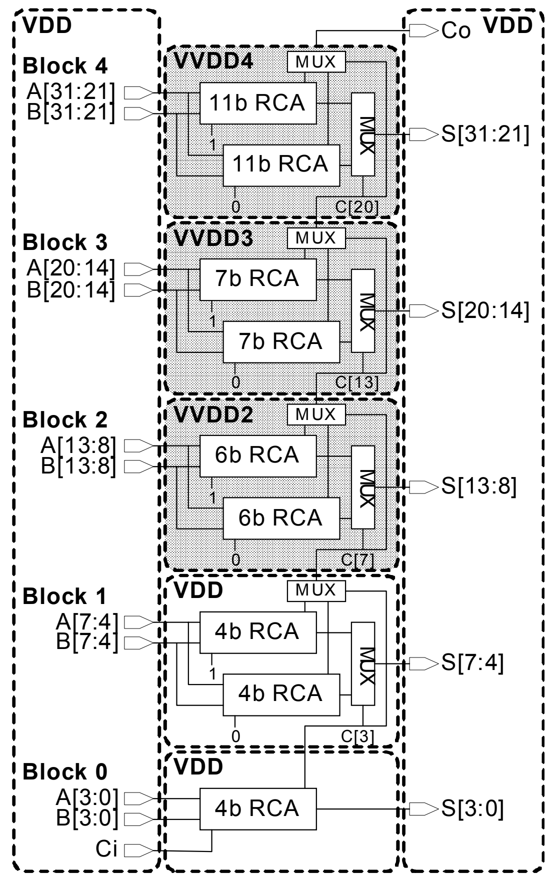


Fig. 2. Block diagram of the 32-bit CSA. Here, the VDD Cont blocks and PMOS switches are not shown to simplify a diagram.

3.1 Carry-Skip Scheme Compensating Delay Overhead

The transition of the VVDD node causes a delay penalty due to the control circuit[2]. Carry-Skip scheme is used for the RCA's, which consist of sub-blocks of CSA, to compensate the delay penalty caused by lowering effective supply voltage during the VVDD transition.

Fig. 3 shows examples of waveforms from SPICE simulation for a 16-bit Adaptive-VDD and a conventional RCA sub-block with a 0.18μm CMOS technology and 1.80V of supply voltage. So[15] is the Sum-out of the most significant bit (MSB) when VVDD changes from VDDlow to VDD. Co[*i*] represents the intermediate carry signals along the critical path of Adaptive-VDD RCA. Sr[15] is the waveform of Sum-out of the MSB of the conventional RCA where VVDD does not change and always stays at 1.80V. Cr[*i*]'s represent the intermediate signals along the critical path of the conventional RCA. Cin is the carry input for the first stage of the 16-bit RCA, and A[*i*]'s and B[*i*]'s are the inputs for the *i*th stage of the adder, where Cin is 1, and A[*i*] is different from B[*i*] for all i ($0 \le i \le 15$). With such an input vector, the entire 16-bit length of the carry propagation path is activated, and the Adaptive-VDD RCA should operate at 1.80 V. However, during the first 0.5 ns (1.0ns ~1.5ns), the circuit operates at lower supply voltage than 1.80 V due to the delay associated with the VDD Control circuit. Hence, Co[1] becomes slower than Cr[1] by 126ps, and finally, So[15] becomes slower than Sr[15] by 137ps. The increased delay is due to the transition of signals while the VVDD node changes from 1.46 V to 1.80 V.

To cancel out this delay penalty, a carry-skip feature is applied to the check-bit operand field. Fig. 4 shows the block diagram of the Adaptive-VDD RCA with a carry-skip feature. The signal, "all propagate", from the VDD Control circuit feeds into the additional multiplexer, and makes it possible to bypass the carry through the check-bit fields, i.e., it makes the lowest carry signal of the check-bit operand field propagate to the highest bit of the check-bit operand field directly. Here, the "all propagate" signal is determined by AND of "propagate" signals at that field.

The Adaptive-VDD RCA has the "all propagate" signal for the check-bit field to control the PMOS switches to supply VVDD. In the specific design shown in Fig. 4, C[5] is forwarded to C[9] if SW0 is activated, that is, if propagate signals, P[9:6], are all "1". The reduced delay due to carry-skip scheme is enough to compensate the delay penalty caused by VDD control circuit. For example, when A[9:6]="1111" and B[9:6]="0000", P[9:6]="1111", the carry-in C[5] propagates to the carry-out C[9] directly. On the other hand, if A[9:6]="1111" and B[9:6]="0100", the carry-out C[8] is generated by A[8] and B[8] and propagates to C[9] regardless of the state of C[5]. Hence, when P[9:6]="1111", we can skip 4-bit RCA having inputs A[9:6] and B[9:6], thereby, we can reduce the worst propagation delay.

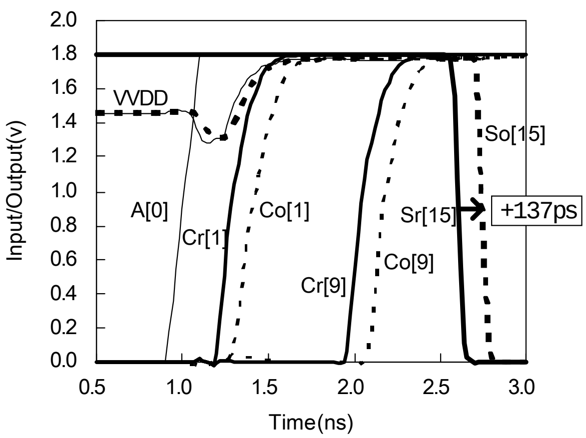


Fig. 3. Waveforms of the Adaptive-VDD and the conventional RCA's. (0.18μm CMOS Technology, VDD=1.80V, VDDlow = 1.46V, Temp = 25C)

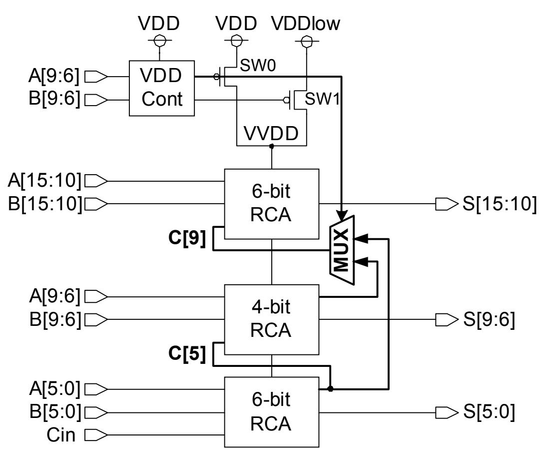


Fig. 4. Block diagram of the Adaptive-VDD RCA with the carry-skip feature for the check-bit field.

Fig. 5 shows waveforms of the Adaptive-VDD RCA with the carry-skip scheme. The suffix *s* represents the improved Adaptive-VDD RCA. During the first 0.5 ns (1.0ns ~1.5ns), the circuit operates at lower supply voltage than 1.80 V due to the delay associated with the VDD Control circuit. Hence, Cs[1] is 126ps slower than Cr[1] like Co[1] in Fig. 3. However, the carry-skip scheme recovers the delay at Cs[9], and finally, Ss[15] becomes faster than Sr[15] by 88ps. Thus, the proposed Adaptive-VDD RCA does not have any delay penalty.

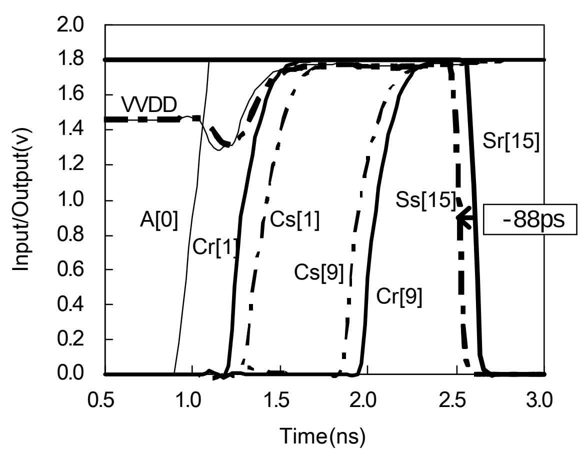


Fig. 5. Waveforms of the Adaptive-VDD with the carry-skip scheme and the conventional RCA's. (0.18μm CMOS Technology, VDD=1.80 V, VDDlow = 1.46 V, Temp = 25C)

3.2 Output Level Converter Using Complimentary Pass Transistor Logic with Dual VDD

The overhead due to the level conversion from VDDlow to VDD is a common problem among the multiple and locally adaptive VDD techniques. In this paper, using Complementary Pass Transistor Logic (CPL) circuit [8] for the level conversion, we reduce both delay and power consumption overhead. Multiplexer circuits drive the outputs of the CSA's. Fig. 6 shows the CPL level shifting multiplexer that uses dual supply voltages: the adaptively controlled VVDD and the fixed VDD. The multiplexer circuit carries out this functionality and outputs the selected sum into the following block. SL0 and SL1 control the NMOS switches to select between the two inputs, Di0 and Di1. Here, SL0 is the inversion of the SL1. Then, if SL1="1", i.e. SL0="0", $\overline{\text{Di1}}$ transfers to $\overline{\text{OUT}}$. Likewise, if SL0="1", i.e. SL1="0", $\overline{\text{Di0}}$ transfers to $\overline{\text{OUT}}$. By driving the cross-coupled PMOS with VDD, the output node always swings between VDD-GND regardless of the VVDD level driving the NMOS transmission gates. Some level converter circuits can cause DC current to flow due to different supplies. However, the proposed CPL level shifting multiplexer does not have the DC current since none of the NMOS have a current from VDD to VVDD via PMOS transistors. One of NMOS's turns on and drives high level of $\overline{\text{OUT}}$ or its complementary node. The turned-on NMOS can have voltage drop between the drain at VDD and the source at VDDlow when VVDD is connected to VDDlow. However, there is no DC current through the NMOS's because the gate voltage is the same as the source voltage, VDDlow, as long as all input nodes of SL0, SL1, Di0 and Di1 are driven by the same Adaptive VVDD node. Otherwise, the level conversion by the CPL with the dual supply voltage can cause DC current for a certain combination of input patterns.

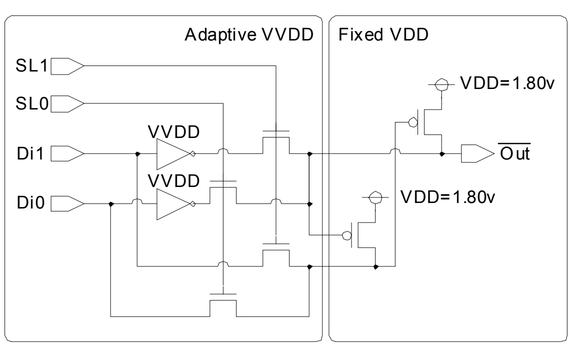


Fig. 6. Complementary Pass transistor Logic (CPL) level shifting multiplexer that uses dual supply voltages: the adaptively controlled VVDD and the fixed VDD.

Fig. 7 shows delay comparison of the CPL multiplexer with the standard CMOS transmission-gate (TG) multiplexer. Fig. 8 shows their power consumptions. "CPL LVC (Level Converter)" represents the proposed circuit with dual supply voltage; "TG LVC" is the standard CMOS transmission gate with the conventional level converter. "CPL" and "TG" denote the CPL and the standard CMOS TG multiplexer with single supply voltage (VVDD). The performance of CPL is very close to TG in terms of both delay and power consumption. Obviously, TG LVC has extra delay and power consumption due to the serially connected components of the level converter. On the other hand, CPL LVC does not have any penalty at 1.80V since it is exactly the same as the conventional CPL. In the low-power operation range, it has a delay penalty of 54ps at 1.46V. The 56ps delay corresponds to the delay improvement of the 32-bit CSA with 0.01 V higher VDDlow value. Hence, the increased VDDlow and power consumption overhead by level conversion would be insignificant in the practical design.

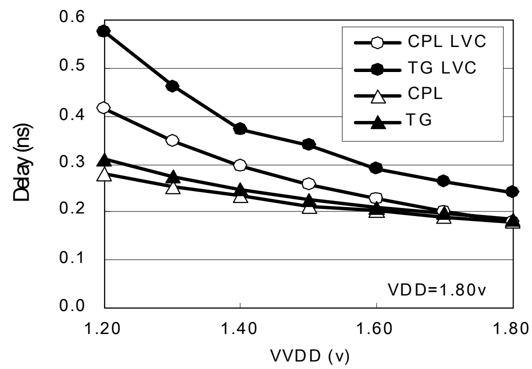


Fig. 7. Delay comparison of the CPL multiplexer with the standard CMOS transmission gate multiplexer.

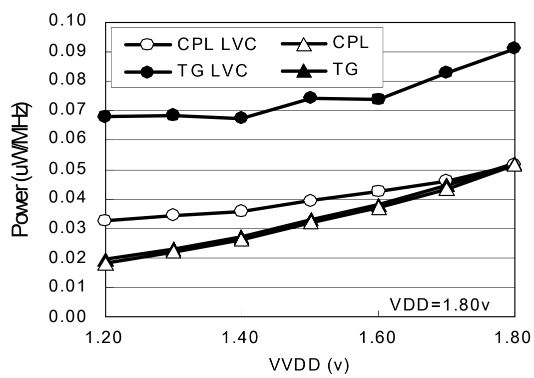


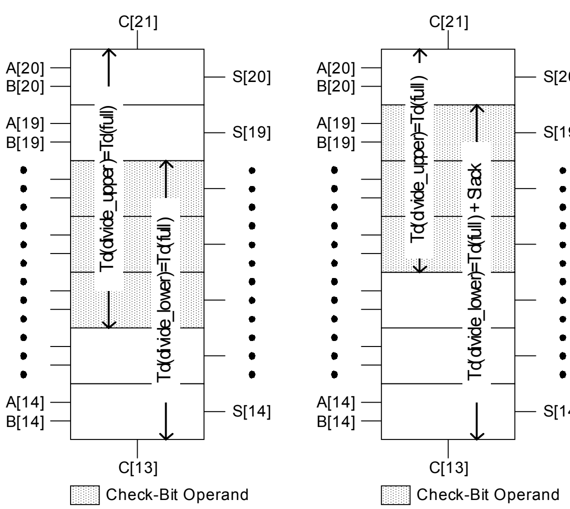
Fig. 8. Power comparison of the CPL multiplexer with the standard CMOS transmission gate multiplexer.

3.3 Optimization for CSA architecture

In a well-optimized CSA, each sub-block has the different operand width. The locally optimal VDDlow levels are different depending on the operand widths. First of all, we designed the sub-blocks, using the same procedure as the Adaptive-VDD RCA [2]. Second, we optimized the VDDlow for the CSA architecture.

There are delay slacks between the delay for generating the sums of sub-blocks and that of the MSB block of the CSA. The worst delays of sub-blocks are smaller than that of the MSB block, which is the same as that of the CSA. Those differences between delays of sub-blocks and that of the MSB block of the delay are the slacks. The definition of the slack is as follows. If we define the largest delay of Block *i* as Td,max(Block *i*), the slack of Block *i* is the delay difference between Td,max (MSB_Block) and Td,max(Block *i*). Here, Td,max (MSB_Block) is equal to the delay of the CSA. For example, the worst delay of the MSB block (Block 4 in Fig.2) is equal to that of the 32-bit CSA, 1.232 ns. Hence, there is no slack. The worst delay of Block 3, 0.903ns, is smaller than 1.232ns. Hence, the slack of Block 3 becomes 0.329ns. Likewise, the lower Blocks, Block 0-2, show the smaller delays, and hence, results in the larger slacks.

By utilizing the slacks, we can move the position of the check-bit fields, and thereby, can reduce VDDlow for Adaptive-VDD sub-blocks in the CSA. Fig. 9 shows optimization of check-bit-field position of Block 3 in the 32-bit CSA. Fig. 9(a) is the case when check-bit field is at center ([18:16]), and Fig. 9(b) is when check bit field is moved towards MSB by 1 bit ([19:17]). In the standard RCA equalization process, Td(divide) is equalized to Td(full) [2]. In Fig. 9(a), Td(divide_upper) and Td(divide_lower) are the same as Td(divide), and VDDlow is determined by the condition of Td(divide_upper) = Td(divide_lower) = Td(full), i.e., Td(5b,VDDlow) = Td(7b,VDD).



(a) Before (b) After

Fig. 9. Optimization of check-bit-field position of Block 3 of the 32-bit CSA to reduce the VDDlow.

When carry-propagation path in the CSA sub-block is divided, the upper stages should generate carry out for next sub-block. Hence, Td(divide_upper) should not be lager than Td(full) because increasing Td(divide_upper) affects to the total delay of CSA due to carry propagating through sub-blocks. On the other hand, the lower stages only need to generate sums of that sub-block. Since the lower stages have slack in the sum output, we can increase the delay constraint of lower stages. Hence, Td(divide_lower) = Td(full)+Slack while Td(divide_upper) = Td(full). In order to satisfy the previous constraints, we shift check-bit field, and reduce the VDDlow. For example, in Fig. 9(b), we shift the check-bit operand from [18:16] to [19:17]. By shifting, Td(divide_upper) becomes Td(4b,VDDlow'), and Td(divide_lower) becomes Td(6b,VDDlow'). In this case, we choose VDDlow' satisfying Td(4b,VDDlow') = Td(7b,VDD), where VDDlow' is lower than VDDlow because Td(4b,VDDlow') = Td(5b,VDDlow). Then, we should check if the delay constraint of the lower stages is satisfied at VDDlow'. If the delay constraint of the lower stages is satisfied, i.e., Td(6b,VDDlow') ≤ Td(7b,VDD) + Slack, we shift the check-bit field in the direction of MSB by one bit, and we repeat the procedure mentioned above. If the constraint is not satisfied, we should find another VDDlow, VDDlow'', satisfying Td(6b,VDDlow'') = Td(7b,VDD) +Slack, where VDDlow'' is larger than VDDlow'.

Table 1(a). 32-bit CSA with 5 sub-blocks.

Block Number	Operand Field of Sub Bock	Ideal VDDlow (V)	Applied VDDlow (V)	Td,max (ns)	Slack at 1.80V (ns)
4	31:21	1.46	1.46	1.232	0.000
3	20:14	1.36	1.46	0.903	0.329
2	13:8	1.35	1.46	0.744	0.488
1	7:4	1.80	1.80	0.601	0.631
0	3:0	1.80	1.80	0.545	0.687

Table 1(b). 64-bit CSA with 8 sub-blocks.

Block Number	Operand Field of Sub Bock	Ideal VDDlow (V)	Applied VDDlow (V)	Td,max (ns)	Slack at 1.80V (ns)
7	63:49	1.44	1.44	1.601	0.000
6	48:37	1.44	1.44	1.357	0.244
5	36:27	1.40	1.44	1.180	0.421
4	26:18	1.33	1.44	1.052	0.549
3	17:11	1.33	1.44	0.862	0.739
2	10:6	1.80	1.80	0.675	0.926
1	5:3	1.80	1.80	0.508	1.093
0	2:0	1.80	1.80	0.395	1.206

Table 1(c). 128-bit CSA with 12 sub-blocks.

Block Number	Operand Field of Sub Bock	Ideal VDDlow (V)	Applied VDDlow (V)	Td,max (ns)	Slack at 1.80V (ns)
11	127:106	1.41	1.41	2.309	0.000
10	106:89	1.36	1.41	2.055	0.254
9	87:72	1.36	1.41	1.852	0.457
8	71:57	1.34	1.41	1.657	0.652
7	56:44	1.27	1.41	1.468	0.841
6	43:33	1.26	1.41	1.288	1.021
5	32:24	1.26	1.41	1.117	1.192
4	23:16	1.32	1.41	0.955	1.354
3	15:10	1.31	1.41	0.792	1.517
2	9:6	1.80	1.80	0.644	1.665
1	5:3	1.80	1.80	0.509	1.800
0	2:0	1.80	1.80	0.395	1.914

Table 1(a), (b), and (c) show the ideal VDDlow and applied VDDlow of (a) the 32-bit, (b) the 64-bit and (c) the 128-bit CSA's, where "ideal VDDlow" is the minimized VDDlow from previous optimization, and "Applied VDDlow" is the VDDlow used for the dual VDD operation in which two voltages, one Applied VDDlow and VDD are used for the Adaptive VDD. As shown in Table 1, the 32-bit Adaptive-VDD CSA ideally requires four supply voltages. However, using four supply voltages on one chip is not practical for LSI implementation. For dual VDD design, we used the largest value among the ideal VDDlow for the Adaptive-VDD blocks and 1.80V fixed voltage for the Non-Adaptive-VDD blocks.

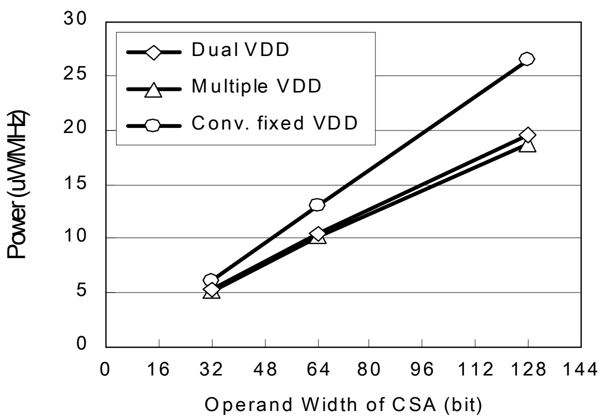


Fig. 10. Power consumption of the 32-bit, 64-bit and 128-bit CSAs.

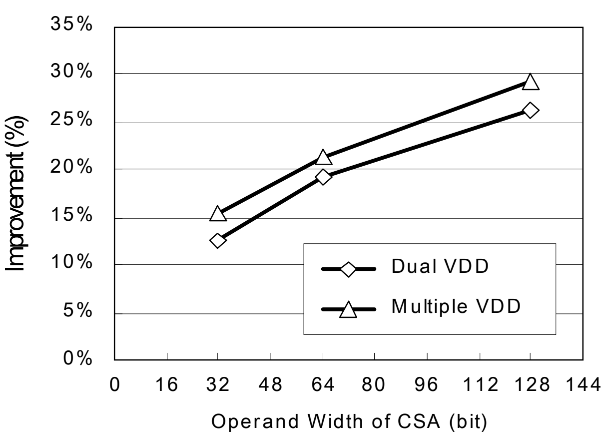


Fig. 11. Power improvement of the 32-bit, 64-bit and 128-bit CSAs.

Using the simulator, NanoSim[9], we simulated the power consumption of the optimized 32-bit, 64-bit and 128-bit CSA's. The results are shown in Fig. 10. "Dual VDD" represents the Adaptive-VDD CSA with dual supply voltages, and "Multiple VDD" is for the ideal Adaptive VDD assuming that there is no restriction on the number of the voltage supplies. The simulation results show that the difference between "Dual VDD" and "Multiple VDD" is less than 6%. Fig. 11 shows the power improvement of Adaptive-VDD CSA over conventional CSA. The optimized 32-bit, 64-bit, and 128-bit CSA's with two VDD's achieve improvement of 12%, 19% and 26%, respectively, in power consumption. The improvement ratios of the dual VDD over the Multiple VDD (ideal case) are 0.82 - 0.90. Hence, in the case of dual VDD, we can achieve comparable power improvement to the Multiple VDD (ideal case).

4. CONCLUSIONS

A very low power Carry-Select Adder (CSA) with adaptive supply voltage has been proposed. The adaptive supply voltage technique for the Ripple Carry Adder (RCA), which applies a lower supply voltage depending on the input vectors, is extended to the CSA design. We utilize the delay slacks inside the CSA architecture for VDD optimization. Using CPL multiplexer with dual supply voltages, the proposed CSA cancels out any delay penalty due to the output level conversion. Simple design of the CSA with our scheme requires several VDD's. However, using too many supply voltages is not practical. For best improvement in the dual supply voltages, we established an optimization process to determine the suitable voltage for the low VDD. The prototype 128-bit CSA shows 26% improvement in power consumption over a conventional CSA with same performance.

5. REFERENCES

[1] A. Chandrakasan, S. Sheng and R. Brodersen, "Low-power CMOS digital design," *IEEE J. Solid-State Circuits*, vol. 27, pp. 473-484, Apr. 1992.

[2] H. Suzuki, W. Jeong and K. Roy, "Low Power Adder with Adaptive Supply Voltage," in Proc. Of Int'l Conf. on Computer Design, pp. 103-106, October 2003.

[3] T. Kuroda, *et al*., "Variable Supply-Voltage Scheme for Low-Power High-Speed CMOS Digital Design," *IEEE J. Solid-State Circuits*, vol. 3, pp. 454-462, March 1998.

[4] T. Burd, T. Pering, A. Stratakos and R. Brodersen, "A Dynamic Voltage Scaled Microprocessor System," *IEEE J. Solid-State Circuits*, vol. 35, pp. 1571-1580, November 2000.

[5] L. Chandrasena and M. Liebelt, "Energy Minimization in Dynamic Supply Voltage Scaling Systems Using Data Dependent Voltage Level Selection," in *Proc. IEEE Int'l Symp. on Circuit and Systems*, May 2000, pp. II-525-528.

[6] T. Pering, T. Burd and R. Brodersen, "Voltage Scheduling in the IpARM Microprocessor System," in *Proc. Int'l Symp. on Low Power Electronics and Design*, July 2000, pp. 96-101.

[7] J. Kao, M. Miyazaki and A. Chandrakasan, "A 175-mW Multiply-Accumulate Unit Using an Adaptive Supply Voltage and Body Bias Architecture," *IEEE J. Solid-State Circuits*, vol. 37, pp. 1545-1554, November 2002.

[8] K. Yano, *etal.*, "A 3.8-ns CMOS 16×16-b multiplier using complementary pass-transistor logic", *IEEE J. Solid-State Circuits*, vol. 25, pp. 388-395, April 1990.

[9] SYNOPSYS Inc., *NanoSim User Manual*, 2002.