Design of a Branch-Based 64-bit Carry-Select Adder in 0.18 µm Partially Depleted SOI CMOS

Amaury Nève, Denis Flandre
Laboratoire de Microélectronique
Université Catholique de Louvain
Place du Levant 3
B-1348 Louvain-la-Neuve, Belgium
++ 32 10 47 25 81
neve@dice.ucl.ac.be

Helmut Schettler, Thomas Ludwig, Gerhard Hellner
IBM Entwicklung GmbH
Schönaicherstrasse 220
D-71032 Böblingen, Germany
++ 49 7031 16 3007
shettler@de.ibm.com

ABSTRACT
The paper presents the design of a 64-bit carry-select adder in Branch-Based Logic, a static design style that minimizes the internal node capacitances. This feature is used to lower the dynamic power dissipation, while maintaining good speed performances. The experimental realization of the adder demonstrates an overall delay of 720 ps while only dissipating 96 mW at 1 GHz. The fabrication is based on the 0.18 µm IBM CMOS8S2 SOI technology, which uses partially depleted transistors and copper metallization.

Categories and Subject Descriptors
B.6.1 [**Logic Design**]: Design Styles – *Combinational Logic.*

General Terms
Performance, Design.

Keywords
Circuit Design, SOI technology, Logic design styles.

1. INTRODUCTION
With the continuous rise of operating frequencies and integration density, dynamic power dissipation becomes a serious concern in microprocessors. Many low-power techniques are proposed in the literature, but are often inappropriate because the power reduction is achieved at the expense of considerable speed loss or reduced noise margins. The renewed interest in design styles like pseudo-NMOS [1] and ratioed CMOS [2] shows that alternative design styles are investigated in order to reduce the power dissipation while still maintaining high-speed performances. The 64-bit adder reported here demonstrates that Silicon-On-Insulator (SOI) technology and Branch-Based Logic (BBL) can be efficiently combined to significantly lower both delay and dynamic power consumption when compared to previous results. Our experimental results are supported by an in-depth study of BBL design and comparisons with conventional static CMOS logic.
In section 2 we briefly introduce the branch-based logic design style and its essential features. Section 3 describes the design of the adder, which is used as a test vehicle to evaluate the potential of branch-based logic. The experimental results are detailed in section 4. Section 5 gives the main conclusions of this work.

ISLPED '02, August 12-14, 2002, Monterey, California, USA.

2. BRANCH-BASED LOGIC
Branch-Based Logic (BBL) is a static design style that is intended for low-power and high-speed circuits [3]. In BBL, the logic function is written as a sum of products so that the corresponding logic cell is implemented exclusively with branches. Each branch is made of a stack of NMOS or PMOS transistors, connected between supply and the output node. CMOS NAND and NOR gates can be considered as elementary branch-based circuits. But the advantages of BBL arise with the implementation of more complex logic functions. Figure 1 shows a circuit block of the 64-bit adder implemented in BBL. Figure 2 shows the same function in conventional CMOS logic. It has been reported that the main advantage of BBL is the reduction of the internal node capacitances of the cell, thanks to the absence of connections between branches other than at the common output node. Moreover, in many cases, two cascaded CMOS logic stages can be designed as a single stage in BBL, which further reduces the parasitic capacitances. Since a lower capacitance is switched during each transition, this results in less dynamic power consumption and lower delays. Finally, it is worthwhile to mention that the static nature of BBL makes it very appropriate for design in Partially Depleted (PD) SOI, since the parasitic effects associated with the floating body are greatly reduced, on the contrary to dynamic circuits [4], [5].

3. DESIGN OF THE 64-BIT ADDER
This section details the design of the 64-bit adder used as a test vehicle to evaluate the performances of the BBL design style. The first sub-section explains the global architecture. Sub-sections 2 and 3 show the implementation of the basic adder cells with the branch-based logic style and compare them with conventional CMOS gates. Sub-section 4 relates the simulation results of the complete adder.

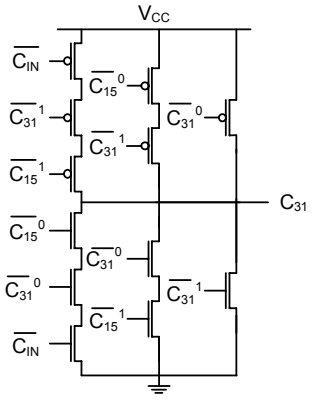

Figure 1. Carry-Select (CS) Box CS-C1 implemented in BBL.

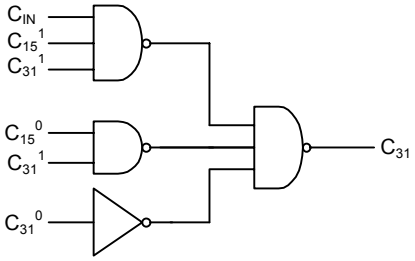

Figure 2. Carry-Select (CS) Box CS-C1 implemented with conventional CMOS gates.

3.1 Architecture of the 64-bit adder

In order to test the potential of BBL, a 64-bit adder is used as a benchmark circuit. The adder has a carry-select structure, since this is very efficient for achieving high-speed addition and carry calculation. The 64-bit adder is divided into four sections (Figure 5). Each section is composed of two 16-bit adders, the first one having a carry-in fixed at "0", the second one having the carry-in signal at "1". The appropriate sum is selected through a multiplexer, whose command is generated by a specific circuit from the previous section, called "Carry Selection Box" or "CS-box". The great advantage of the carry-select structure is that all the carry-signals from the 16-bit adders are computed in parallel and arrive approximately at the same time. They are then fed into the CS-boxes to generate the control signals for the multiplexers and the final carry-out. No time is lost waiting for a carry signal to ripple through the whole circuit.

The carry-select architecture is reproduced at three different levels, i.e. in the 64-bit, the 16-bit and the 4-bit adders. This increases the die area, but enables a very fast carry generation. The CS-boxes of the 16-bit and of the 4-bit adders have just the same circuit topology as those for the 64-bit adder, but the optimization is different because the output loads are not the same.

All the cells are implemented using BBL, with the exception of the multiplexers, which are transmission-gate multiplexers.

3.2 Implementation of the basic building blocks

The CS-boxes generate the control signals for the multiplexers as well as the final carry-out. The BBL CS-boxes are very compact for two reasons. Firstly, the branches corresponding to specific "never happen" input combinations are removed from the schematic. Secondly, one or two stages are sufficient to realize the function. In BBL, CS-C0 (Figure 3) and CS-C1 (Figure 1) are realized in one stage instead of two in conventional static CMOS. CS-C2 (Figure 4) is composed of two stages linked with an inverter. A conventional CMOS equivalent circuit requires the use of four successive stages. Moreover the BBL implementation requires fewer transistors than the implementation in conventional CMOS (table I). These features contribute to the reduction of the dynamic power consumption.

Table I. Number of transistors in BBL CS-boxes and conventional CMOS CS-boxes.

	BBL	CMOS
CS-C0	6	10
CS-C1	12	18
CS-C2	23	30

3.3 Basic building block simulation results in BBL and conventional CMOS

To allow a fair and meaningful comparison between BBL and conventional CMOS logic, each cell has been optimized by considering the input combinations that give the largest delay. The W/L-ratios of the transistors are then further tuned to reduce the internal node capacitances when applying other input patterns. We compared the performances of the basic building blocks of the adder in BBL and conventional CMOS by using circuit simulations with a 0.18 μm SOI PD CMOS technology. The device models include the parasitic capacitances of source, drain and gate. Each cell is loaded with one CMOS inverter. The detailed delay results are presented in table II. BBL cells with few inputs and few branches (like CS-C0 and the half-adder) are as fast or even faster than equivalent conventional CMOS cells. But larger BBL cells are slower. The bad speed performances of the latter cells can be explained by the combination of two factors. Firstly, the worst-case pattern activates a 3-transistor PMOS stack in the BBL cells. In case of the conventional CMOS cells, NMOS-stacks are activated in the worst case. Moreover, the high number of branches connected at the output increases the parasitic capacitance at the output node, even if the total parasitic capacitance of the cell is reduced. On the contrary to the delay, the results related to dynamic power dissipation are always in favor of the branch-based cells. The simulation results are presented in table III. The use of a branch-based architecture to implement the cells results in a 22 % to 50 % reduction of dynamic power dissipation compared to conventional CMOS.

Table II: Simulated delays for the BBL and conventional CMOS implementations of the basic building blocks of the carry-select adder. V_{DD} = 1.5 V; T= 25 °C; load = 1 CMOS inverter.

	BBL	CMOS	% delay reduction
Half-adder	53.1 ps	66.1 ps	24.5 %
CS-C0	40.4 ps	44.1 ps	9.1 %
CS-C1	68.0 ps	53.0 ps	-22.1 %
CS-C2	104.8 ps	96.8 ps	-7.6 %

Table III: Simulated dynamic power dissipation for the BBL and conventional CMOS implementations of the basic building blocks of the carry-select adder. V_{DD} = 1.5 V; T= 25 °C; Bit rate = 1 GHz.

	BBL	CMOS	% power reduction
Half-adder	55.1 µW	70.5 µW	21.8 %
CS-C0	30.5 µW	60.9 µW	49.9 %
CS-C1	65.4 µW	106.5 µW	38.6 %
CS-C2	108.8 µW	172.5 µW	36.9 %

3.4 Simulation of the 64-bit adder

Schematic simulations of the BBL and CMOS 64-bit carry-select adders confirm the trend observed at the cell level. With random input patterns applied at a rate of 1 GHz, the BBL adder features a dynamic power consumption, which is reduced by 10 % compared to the conventional CMOS adder. The delay increase associated with BBL is less than 2 % for supply voltages lower than 1.5 V. For higher supply voltages, the BBL vs. CMOS speed difference increases slightly, but remains lower than 5 %. For worst-case input patterns, the peak dynamic power consumption is reduced by 16 % in BBL compared to CMOS. The overall reduction of dynamic power dissipation is lower than for the single cells, because the complete adder also involves inverters and multiplexers, which are similar in both realizations. The differences result only from the choice of the design style of the half-adders and the carry-select boxes, the BBL and conventional CMOS adders having the same global architecture. For the simulations, the capacitive output load was 150 fF. These results are coherent with previous results obtained for a 16-bit carry-select adder on other SOI CMOS processes: 0.25 µm Fully-Depleted (FD) SOI and 0.25 µm Partially-Depleted (PD) SOI [6]. For the 16-bit adder in PD SOI, the peak dynamic power consumption is reduced by 20 % in comparison with a conventional CMOS implementation. When moving the 16-bit adder to FD SOI, the power reduction reaches 35 %.

4. EXPERIMENTAL RESULTS

An experimental 64-bit BBL carry-select adder has been realized on the IBM CMOS8S2 SOI technology. It is composed of 18k devices and occupies an area of 735 µm x 280 µm. The IBM CMOS8S2 SOI technology is a 0.18 µm Partially Depleted SOI CMOS process [7]. Key features include: 7 layer copper metallization with 2 reverse scaled layers to improve RC delay. In addition, a local interconnect layer based on tungsten is used to improve circuit level density. On transistor level, a dual threshold feature is offered (regular V_T and low V_T). This adder uses only regular V_T devices in order to maintain low static power dissipation. Gate oxide thickness is 3.3 nm in conjunction with a typical operating voltage of 1.5 V.

The 64-bit adder has been tested under different voltage and temperature conditions and operates successfully. Worst-case input patterns have been applied and the propagation delays to the sum and carry outputs measured. In the worst-case situation and at 1.5 V, the final carry-out is produced after 600 ps. Thanks to the independent carry-network composed of the CS-boxes, the carry-out arrives earlier than the last sum outputs, corresponding to bit positions 32 to 47 (S32..47), which arrive after 720 ps in the worst case. The sum outputs corresponding to bit positions 48 to 63 (S48..63) arrive earlier, after 640 ps, because C47 is ready before C31 (Figure 5). C47 intervenes in the computation of the carry-out and also commands the multiplexer selecting S48..63. C31 commands the multiplexer selecting S32..47. The late completion of SUM32-47 is associated with our choice to favor fast carry generation in two ways. Firstly, since the delay in a CS-C2 cell is the largest, the intermediate carry signals C_{15}^0, C_{15}^1, C_{31}^0, C_{31}^1 are fed directly into CS-C2 for the generation of C47. To compute the carry-out C_{63}, only one additional stage is necessary i.e. CS-C0. Secondly, buffers are added on the signal path to the inputs of CS-C1 in order to reduce the capacitive load seen by the outputs of the cells generating the signals C_{15}^0, C_{15}^1, C_{31}^0, C_{31}^1. By this way, these signals do not see the high capacitive charge of the CS-C1 cell. Figure 6 shows the measured delay for supply voltages from 1.3 V up to 2.55 V. The dynamic power dissipation when applying random inputs at a rate of 1 GHz is 96 mW with a supply voltage of 1.5 V and at 25 °C.

5. CONCLUSION

A 64-bit carry-select adder has been designed and implemented using Branch-Based Logic. The simulation results show that this design technique allows a reduction of the dynamic power consumption without degrading the speed performances compared to a conventional CMOS implementation. The design has been validated on a 0.18 µm SOI Partially Depleted technology and operates successfully at different temperatures and different supply voltages. Our work demonstrates the benefits of branch-based logic combined with SOI for low power applications.

6. ACKNOWLEDGMENTS

The authors thank Mr. J. Appinger for his helpful assistance in obtaining the experimental data. The authors thank also Mr. J. Keinert and Mr. U. Krauch for the help they provided when using the simulation and extraction tools.

7. REFERENCES

[1] Subba N., Salman A., Mitra S., Ioannou D.E. and Tretz C., "Pseudo-nNMOS Revisited: Impact of SOI on Low Power, High Speed Circuit Design", 2000 IEEE Int. SOI Conf., Oct. 2000, pp. 26-27.

[2] Tretz C.R., Montoye R.K. and Reohr W., "Ratioed CMOS : a Low Power High Speed Design Choice in SOI Technologies", 2000 IEEE Int. SOI Conf., Oct. 2000, pp. 28-29.

[3] Masgonty J.M., Arm C. and Piguet C., ″Technology- and Power Supply-Independent Cell Library″, IEEE Custom Integrated Circuits Conference, 1991, pp. 25.5.1-25.5.4.

[4] Das K. and Brown R., '' Evaluation of Circuit Approaches in Partially Depleted SOI-CMOS'', 2000 IEEE Int. SOI Conference, pp. 98-99.

[5] Canada M., Akrout C., Cawthron D., Corr J., Geissler S., Houle R., Kartschoke P., Kramer D., McCormick P., Rohrer N., Salem G. and Warriner L., "A 580 MHz RISC Microprocessor in SOI", ISSCC 1999, pp. 430-431.

[6] Nève A. and Flandre D., "Branch-Based Logic for High Performance Carry-Select Adders in 0.25 um Bulk and Silicon-On-Insulator CMOS Technologies", PATMOS 2001, pp. 8.2.1-8.2.10.

[7] Leobandung E., Barth E., Sherony M., Lo S.-H., Schulz R., Chu W., Khare M., Sadana D., Schepis D., Boiam R., Sleight I., White F., Assaderaghi F., Moy D., Biery G., Goldblan R., Chen T.-C., Davari B. and Shahidi G., ''High Performance 0.18 μm SOI CMOS Technology'', International Electron Devices Meeting 1999, pp. 679-682.

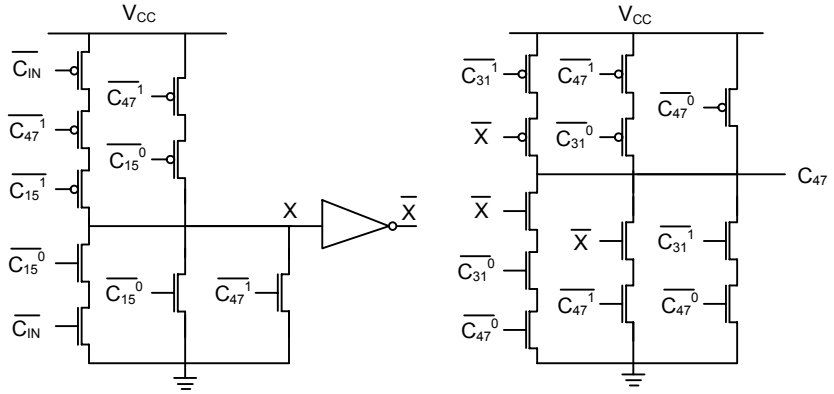

Figure 4. BBL implementation of CS-C2.

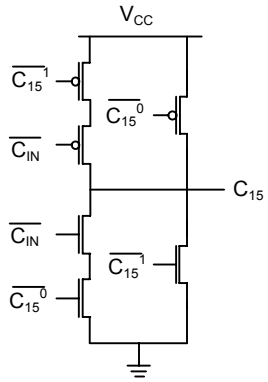

Figure 3. BBL implementation of CS-C0.

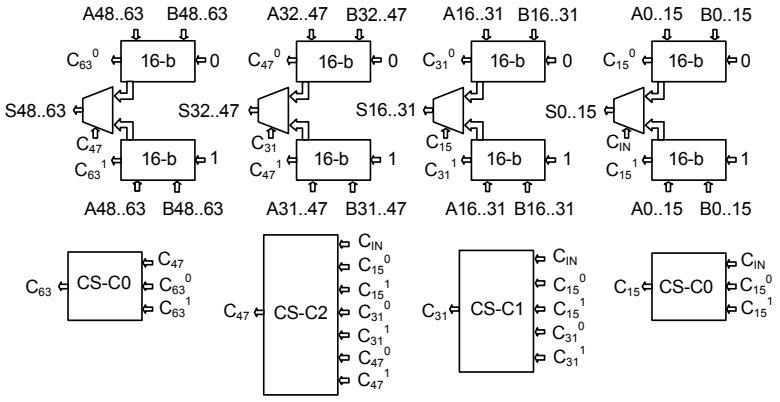

Figure 5. Architecture of the 64-bit adder.

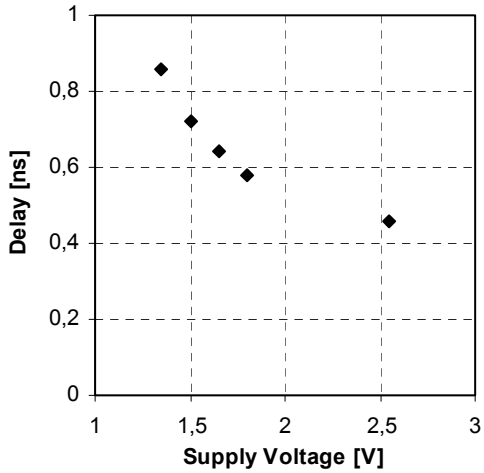

Figure 6. Measured delay vs. supply voltage.