Sub-500-ps 64-b ALUs in 0.18-μm SOI/Bulk CMOS: Design and Scaling Trends

Sanu K. Mathew, *Member, IEEE*, Ram K. Krishnamurthy, *Member, IEEE*, Mark A. Anders, *Member, IEEE*, Rafael Rios, *Member, IEEE*, Kaizad R. Mistry, *Member, IEEE*, and K. Soumyanath, *Member, IEEE*

***Abstract*—In this paper, we present: 1) design of a single-rail energy-efficient 64-b Han–Carlson ALU, operating at 482 ps in 1.5 V, 0.18-μm bulk CMOS; 2) direct port of this ALU to 0.18-μm partially depleted SOI process; 3) SOI-optimal redesign of the ALU using a novel deep-stack quaternary-tree architecture; 4) margining for max-delay pushout due to reverse body bias in SOI designs; and 5) performance scaling trends of the ALU designs in 0.13-μm generation. We show that a direct port of the Han–Carlson ALU to 0.18-μm SOI offers 14% performance improvement after margining. A redesign of the ALU, using an SOI-favored deep-stack architecture improves the margined speedup to 19%. A 10% margin was required for the SOI designs, to account for reverse body-bias-induced max-delay pushout. Preconditioning the intermediate stack nodes in the dynamic ALU designs reduced this margin to 2%. Scaling the ALUs to 0.13-μm generation reduces the overall SOI speedup for both architectures to 9% and 16%, respectively, confirming the trend that speedup offered by SOI technology decreases with scaling.**

***Index Terms*—High-performance adders, high-performance and low-power CMOS design, silicon-on-insulator technology.**

I. Introduction

THE requirements of high-throughput Internet servers necessitate the use of multiple ALUs in high-performance 64-b execution cores. Consequently, each ALU demands a compact energy-efficient 64-b adder core with single-cycle latency. The resultant critical path, which is a balanced mix of interconnect, diffusion, and gate loads, forms a representative test bed for evaluating competing circuit techniques and process technologies.

Partially depleted SOI technology has been shown to provide performance advantages over bulk in 0.18-μm generation [1]–[3]. These advantages stem from reduced diffusion capacitance, the absence of body effect, and dynamic lowering of device threshold voltage due to the floating-body effect [1].

In this paper, we quantify the costs and benefits of designing high-performance datapath circuits in SOI. We begin with an energy-efficient adder architecture in bulk CMOS and develop novel circuits that enable a compact single-rail implementation. In the migration of our designs to SOI, we present two options:

Manuscript received March 21, 2001; revised June 15, 2001.

S. K. Mathew, R. K. Krishnamurthy, M. A. Anders, and K. Soumyanath are with the Circuit Research Lab., Intel Corporation, Hillsboro, OR 97124 USA (e-mail: sanu.k.mathew@intel.com; ram.krishnamurthy@intel.com; mark.a.anders@intel.com; krishnamurthy.soumyanath@intel.com).

R. Rios is with Technology-CAD, Intel Corporation, Hillsboro, OR 97124 USA (e-mail: rafael.rios@intel.com).

K. R. Mistry is with Portland Technology Development, Intel Corporation, Hillsboro, OR 97124 USA (e-mail: kaizad.mistry@intel.com).

Publisher Item Identifier S 0018-9200(01)08217-8.

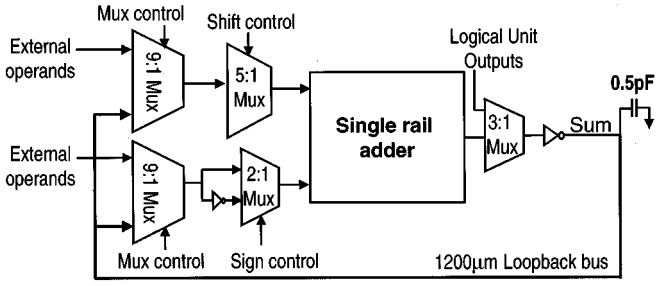


Fig. 1. 64-b ALU architecture.

1) direct port of the baseline bulk design to a comparable SOI technology;
2) SOI-optimal redesign of the adder core.

We describe the optimal design approach for developing adder architectures in SOI by leveraging some of the key advantages offered by SOI technology. Further, we describe design margining required for the SOI implementations and report the results of shrinking the two architectures to 0.13-μm bulk/SOI. In both cases, a sophisticated SOI compact model that incorporates features to effectively model the SOI floating-body effect is used.

The remainder of this paper is organized as follows. Section II describes the 64-b ALU architecture used as the evaluation test bed. In Section III, we describe an energy-efficient adder architecture used as the baseline bulk design. We also present two novel circuits that enabled an energy-efficient single-rail implementation. Section IV describes our SOI process and model characteristics and reports measured history-effect data in 0.18-μm SOI. In Section V, we present the "direct port" option of migrating bulk designs to SOI. The second migration option, of redesigning the ALU using an SOI-optimal adder core, using a novel deep-stack architecture, is presented in Section VI. Section VII deals with the margining issues involved in an SOI design. Section VIII presents the effects of scaling in both SOI and bulk technologies. Finally, in Section IX, we summarize the results and conclude the paper.

II. 64-b ALU Architecture

The 64-b ALU architecture (Fig. 1) is designed to accommodate six ALUs in the execution core of a microprocessor. A 9:1 multiplexer in the first stage of the ALU selects from among six ALU operands and three register file operands to deliver the first operand to the adder core. (One of the ALU operands is the ALU output looping back to its own input through a 1200-μm loopback bus.) The first operand goes into a 5:1 shifter multiplexer,

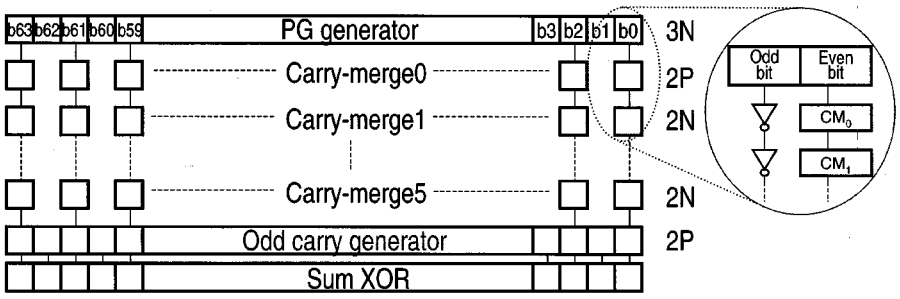


Fig. 2. 64-b Han–Carlson adder core.

where a 5-b shift operation may be performed. The second adder operand is provided by the 2:1 multiplexer that selects between the true and complementary values of the second operand. This enables subtract operations in the ALU. A 64-b single-rail adder forms the core of the ALU. The output of this adder goes into a 3:1 result multiplexer that selects between the adder and logical unit outputs. Finally, we have a bus driver driving a 1200-μm loopback bus and a 0.5-pF load to the L0 cache.

The above ALU is a good mix of different types of circuits, ranging from diffusion-dominated transmission gate multiplexers to load-dominated static and dynamic gates within the adder core. It also has long interconnects, both in the adder core (up to 750 μm in length), and in the 1200-μm ALU loopback bus. Hence, it forms an ideal representative test bed for evaluating the performance of different process technologies. The various ALU configurations which we will explore in this paper focus on the implementation of the adder core. The peripheral circuits remain unchanged in the different ALU designs presented in this paper.

III. Single-Rail 64-b Han–Carlson ALU

The 64-b Han–Carlson adder core [4] designed in bulk CMOS technology forms our baseline design. While the Han–Carlson adder (Fig. 2) employs a logarithmic binary carry–merge scheme, similar to a Kogge–Stone approach, the key difference between the two lies in the implementation of the carry–merge tree. As shown in Fig. 2, a full carry–merge implementation is avoided by skipping alternate bitslices in the tree and performing the carry–merge operation on even bitslices $(b_0, b_2, \ldots b_{62})$ only. In the odd bitslices, propagate (P_i) and generate (G_i) signals from the first stage are sent down the adder core through a chain of minimum-sized inverters. Thus, at the end of six stages of carry–merge (Carry–merge0 ...Carry–merge5), 32 bits of carry $(C_0, C_2 \ldots C_{62})$ are generated. The missing odd carries $(C_1, C_3 \ldots C_{63})$ are generated in an extra stage of carry–merge logic, called the odd carry generator (Fig. 2).

The adder core consists of nine gate stages (five dynamic gates interspersed with four static gates) that perform a radix-2 carry–merge operation $(G = G_i + P_i \cdot G_{i-1}; P = P_i \cdot P_{i-1})$ in both the dynamic and static gates of the carry–merge tree. Consequently, the worst-case evaluation path is a 3-nMOS pulldown followed by a 2-pMOS pullup, and so on. (Hence, a critical path of 3N-2P-2N-2P-2N-2P-2N-2P-XOR, as shown in Fig. 2). The 3-nMOS pulldown path in the first stage is due to the extra clocked evaluation transistor required at that stage. This architecture enabled 50% reduction in gate count in the carry–merge tree, resulting in a compact energy-efficient design. The adder core may be upsized to account for the extra stage of carry–merge logic and arrive at the same performance as a Kogge–Stone adder. At equal performance, the Han–Carlson adder consumes 43% less energy compared to an equivalent Kogge–Stone implementation (see Table I).

TABLE I
Energy/Transition of Single-Rail 64-b Adders

Adder architecture	Energy/transition
Kogge-Stone	120pJ
Han-Carlson	68pJ

TABLE II
Han-Carlson Carry–Merge Tree Logic

PG Generator :

$$P = \overline{\overline{A_i} \bullet \overline{B_i}}, \quad G = \overline{\overline{A_i} + \overline{B_i}} \quad i = 0 \cdots 63$$

CM0, CM2, CM4 :

$$\overline{P} = \overline{P_i \bullet P_{i-1}}, \quad \overline{G} = \overline{G_i + P_i \bullet G_{i-1}} \quad i = 0, 2 \cdots 60, 62$$

$$\overline{\overline{P}} = \overline{P_i}, \quad \overline{\overline{G}} = \overline{G_i} \quad i = 1, 3 \cdots 61, 63$$

CM1,CM3 :

$$P = \overline{\overline{P_i} + \overline{P_{i-1}}}, \quad G = \overline{\overline{G_i} \bullet \left(\overline{P_i} + \overline{G_{i-1}}\right)} \quad i = 0, 2 \cdots 60, 62$$

$$\overline{P} = \overline{\overline{P_i}}, \quad \overline{G} = \overline{\overline{G_i}} \quad i = 1, 3 \cdots 61, 63$$

CM5 :

$$Carry_{even} = \overline{\overline{G_i} \bullet \left(\overline{P_i} + \overline{G_{i-1}}\right)} \quad i = 0, 2 \cdots 60, 62$$

$$G = \overline{\overline{G_i}} \quad i = 1, 3 \cdots 61, 63$$

Odd Carry Generator :

$$\overline{Carry_{odd}} = \overline{G_i + P_i \bullet Carry_{even}} \quad i = 1, 3 \cdots 61, 63$$

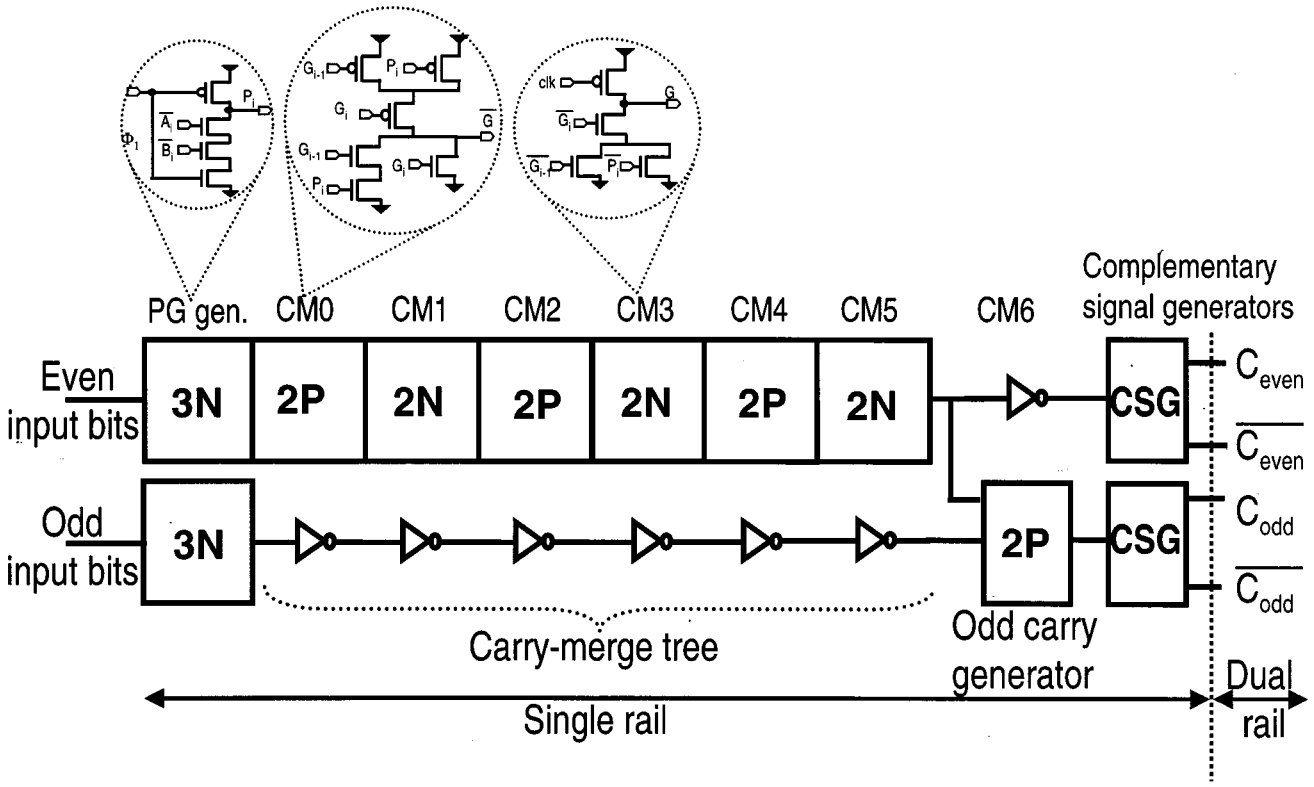


Fig. 3. Han–Carlson carry–merge tree.

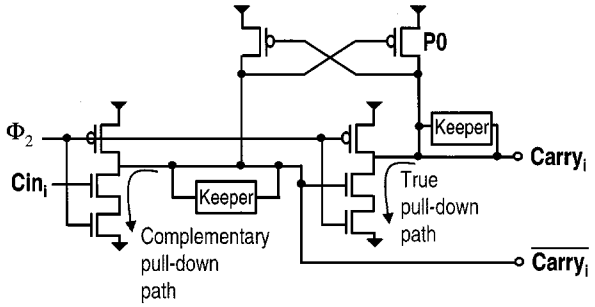


Fig. 4. Complementary signal generator.

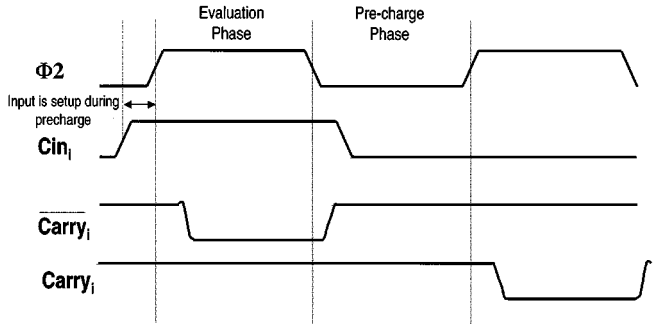


Fig. 5. Timing diagram of complementary signal generator.

Fig. 3 shows two adjacent bitslices of the adder core. The propagate–generate stage is followed by six stages of carry–merge terminating in the odd carry generator. The logic implemented at each stage is shown in Table II. The PG generator, implemented in dynamic logic, generates the P_i and G_i signals from the inputs ($\overline{A_i}$ and $\overline{B_i}$). The carry–merge tree is implemented in six stages (CM0–CM5), with the static gates (CM0, CM2, CM4) interspersed with the dynamic carry–merge gates (CM1, CM3). In the odd bitslices ($i = 1, 3, \ldots 61, 63$), minimum-sized inverters send the P_i and G_i signals to the next logic stage. After six stages of carry–merge, the even carries ($\mathrm{Carry}_{\mathrm{even}}$) are generated by CM5. In the odd bitslices, the P_i and G_i signals are sent to the odd carry generator, where they are merged with the even carries to generate the missing odd carries ($\mathrm{Carry}_{\mathrm{odd}}$). Thus, the 64 carry bits, generated after eight stages of logic, may now be XORed with the partial sum to generate the final sum.

A. Single-Rail Implementation of Han–Carlson Adder Core

This being a domino design, the XOR operation would require a dual-rail domino compatible carry signal. A full dual-rail domino design was avoided by keeping the whole adder core single-rail, and using a complementary signal generator to generate the dual-rail carry signals. The single-rail/dual-rail boundary is shown in Fig. 3 by the dotted line. The complementary signal generator and a single-ended dynamic XNOR gate are two novel circuits that enabled a single-rail implementation of the adder core.

B. Complementary Signal Generator

The complementary signal generator (Fig. 4) takes in a single-ended signal (C_{in}) and generates a domino-compatible dual-rail output (Carry_i and $\overline{\mathrm{Carry}_i}$). This circuit has two pulldown paths, both of which are precharged with the same clock Φ_2:

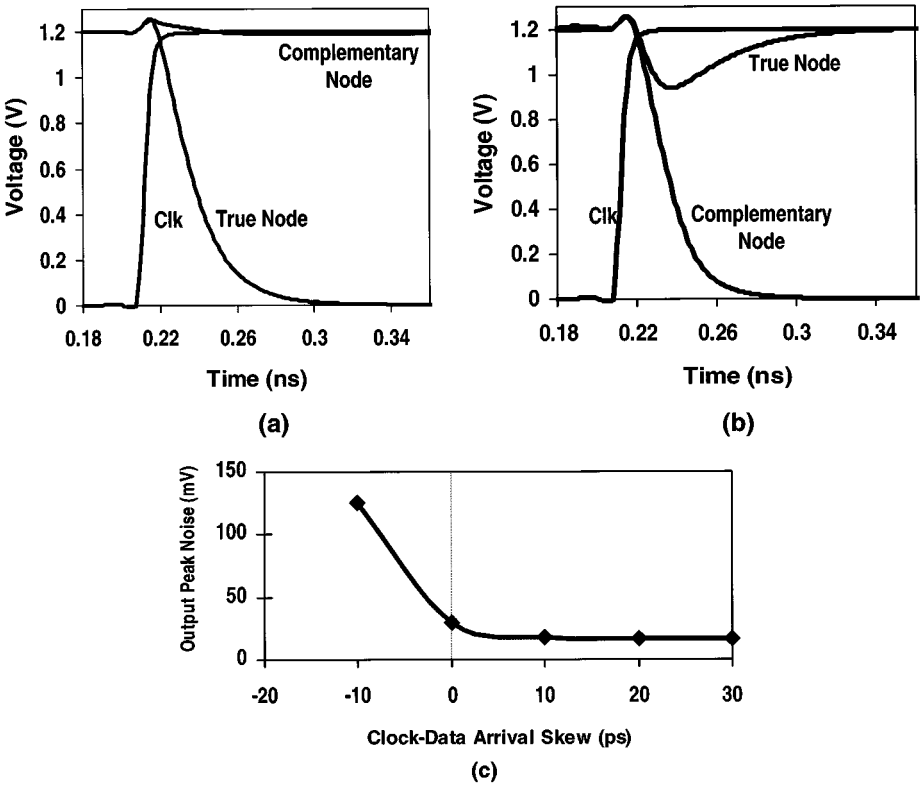


Fig. 6. CSG simulation waveforms. (a) $C_{\rm in} = 0$. (b) $C_{\rm in} = 1$. (c) Sensitivity of CSG to clock-input data skew.

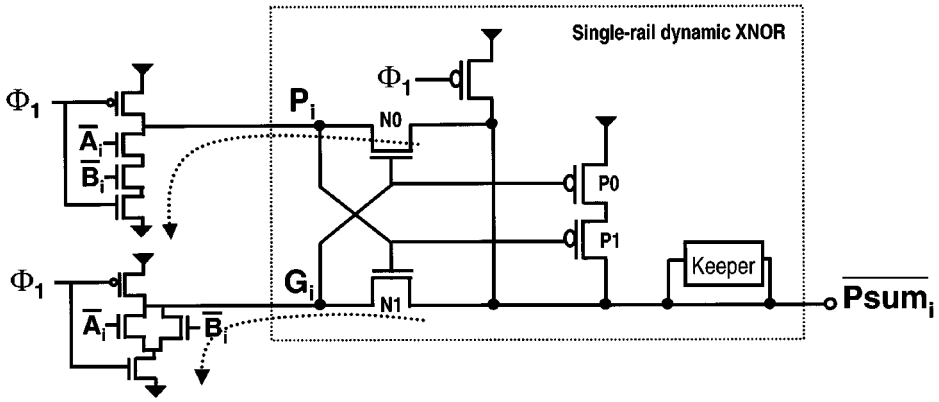


Fig. 7. Partial-sum generator circuit.

1) the complementary pulldown path, gated with the input signal;
2) the true pulldown path, gated with the complementary signal node.

Depending on the logic state on the input, either of these two paths will discharge to Gnd, holding the other path high through the cross-coupled pMOS network. In this way, Carry_i and $\overline{\text{Carry}_i}$ signals are both precharged high and will settle at complementary logic levels during evaluation (Fig. 5).

Fig. 6(a) shows the simulation waveform for the case when $C_{\rm in} = 0$. (The signal $C_{\rm in}$, originating from the static portion of a domino gate, is precharged low). In this case, the complementary node remains high, resulting in the discharge of the true node. When $C_{\rm in} = 1$ [Fig. 6(b)], the rising edge of the clock triggers the discharge of the complementary node. The finite discharge time of this node, causes a noise droop on the true node. The magnitude of this noise pulse is limited by the cascode pMOS device P0 (Fig. 4), which is turned on by the discharge of the complementary node. Thus, the true node is statically held by the complementary node.

A race between the clock and the input signal can increase the noise droop [Fig. 6(b)] on the true output node. In the extreme case, this can result in a false evaluation. The sensitivity of the true node noise to input data skew is shown in Fig. 6(c). To limit the noise magnitude to 20 mV, the input signal must be set up before the clock arrives, necessitating a setup margin at the clock edge. To avoid this 'nontime borrowability' penalty, we place the gate at the Phase-2 clock boundary, thereby absorbing this penalty in the phase jitter margin that is normally applied at all phase boundaries.

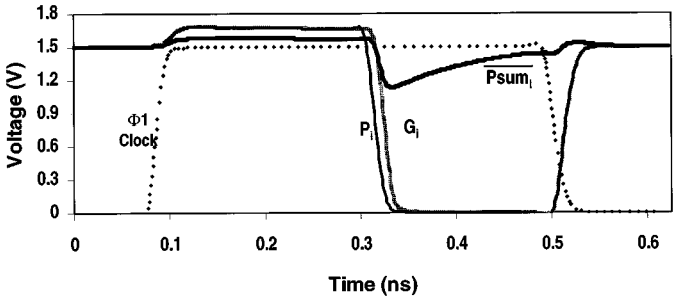


Fig. 8. Single-rail dynamic XNOR simulation waveforms.

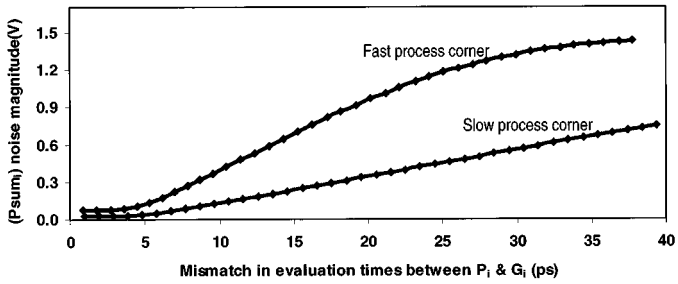


Fig. 9. Sensitivity of dynamic XNOR node to input evaluation mismatch.

C. Single-Rail Dynamic XNOR

Another gate that enabled a single-rail implementation is a single-ended dynamic XNOR circuit. This gate is used in the partial sum generator (Fig. 7) to deliver a domino compatible partial sum signal. In this gate, propagate (P_i) and generate (G_i) signals from the PG-generator are XORed to generate the partial sum. P_i and G_i are single-ended dynamic signals, precharged high with the clock Φ_1. The output of the XNOR gate is also precharged high by Φ_1. The operation of the XNOR gate is as follows:

- If either P_i or G_i discharge, the output will also discharge through either of the two discharge paths through N0 or N1.
- If both P_i & G_i discharge, the NMOS transistors N0 and N1 turn off, cutting off both discharge paths. The 2-pMOS stack formed by P0 and P1 will turn on, holding the output node high (Fig. 8)
- If neither P_i nor G_i discharge, once again, there is no discharge path, and the output is held high by the keeper.

Mismatches in the discharge times of P_i and G_i (due to history effect/process variations) can cause the output node $(\overline{\text{Psum}}_i)$ to discharging partially (before the 2-pMOS stack restores it to V_{cc}), resulting in a noise glitch at the output (Fig. 8). The magnitude of noise on the dynamic node, depends on the mismatch between discharge times of P_i and G_i (Fig. 9). To minimize the noise magnitude to below 30 mV, the evaluation times of P_i and G_i should be synchronized. We ensure this by placing the partial sum generator at the Φ_1 clock boundary. Since we do not time-borrow on the phase boundary, we guarantee that the inputs $\overline{A_i}$ and $\overline{B_i}$ are setup before the clock fires limiting the dynamic node noise to below 30 mV (Fig. 9).

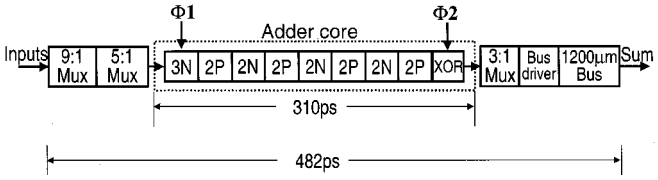


Fig. 10. Han–Carlson ALU critical path and simulation results in 0.18-μm bulk.

TABLE III
PROCESS PARAMETERS: $V_{cc} = 1.5$ V, $T = 30$ °C, 0.18-μm TECHNOLOGIES

	Ioff(nA/μm)	Idsat(μA/μm)
NMOS-Bulk	3.3	1070
NMOS-SOI	3.3	1050
	Ioff(nA/μm)	Idsat(μA/μm)
PMOS-Bulk	0.7	445
PMOS-SOI	0.7	441

Fig. 10 shows the critical path of the Han–Carlson ALU, with the single-ended dynamic XNOR (Fig. 7) placed at the Φ_1 clock boundary and the complementary signal generator (Fig. 4) folded in with the sum XOR and placed in the Φ_2 clock boundary. The ALU operates with a delay of 482 ps in 0.18-μm bulk CMOS technology [5] with a supply voltage of 1.5 V. The adder core, being a smaller portion of the ALU, operates at 310 ps. These are the baseline numbers against which the performance of the different flavors of the 64-b ALU will be compared.

IV. MIGRATION FROM BULK CMOS TO SOI

The potential performance improvements offered by SOI technology over bulk CMOS [1]–[3] motivate a migration of our bulk designs to SOI. Here, we have two options.

1) The first option is a direct port of the bulk design to a comparable SOI process. The designs issues here relate to managing the reduced noise tolerance of dynamic gates due to lowered dynamic V_t [6], increased power supply noise due to the lack of n-well decoupling capacitances [7] and the impact of the history effect on min-delay paths [8], [9].
2) The second option is to leverage some of the key advantages offered by SOI technology and do an SOI-favored redesign of the ALU, to arrive at an SOI-optimal design.

A. SOI Process Characteristics

Floating-body effects in SOI forward bias the body of the device at dc conditions, lowering its V_t, resulting in an increase in subthreshold leakage current (I_{off}) [1]. To keep the comparisons between the bulk and SOI designs fair, we stipulate that the two technologies should have equal I_{off} at dc conditions. The SOI device was therefore engineered to have the same I_{off} at 30 °C dc equilibrium conditions, as a bulk device would (Table III). Consequently, the I_{dsat} of the SOI device is between 1%–2% lower than its bulk counterpart. We believe that this is a conservative stipulation, since the V_t of the SOI device will be lowered

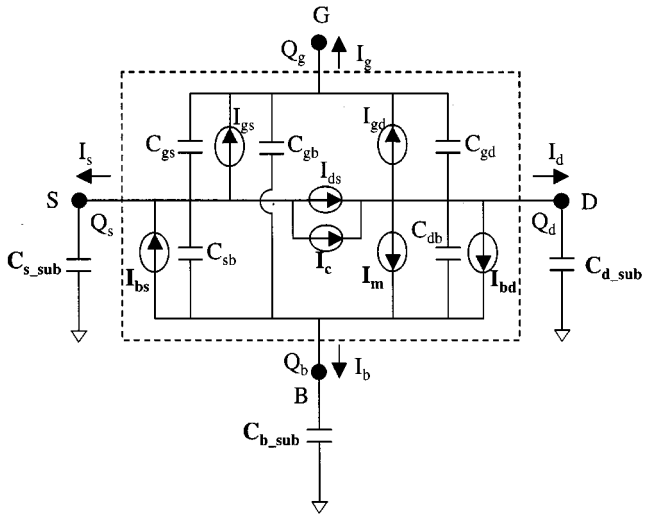


Fig. 11. Equivalent circuit for partially depleted SOI devices. The core device model is highlighted by the dashed box. All intrinsic model capacitance components are not included here for simplicity.

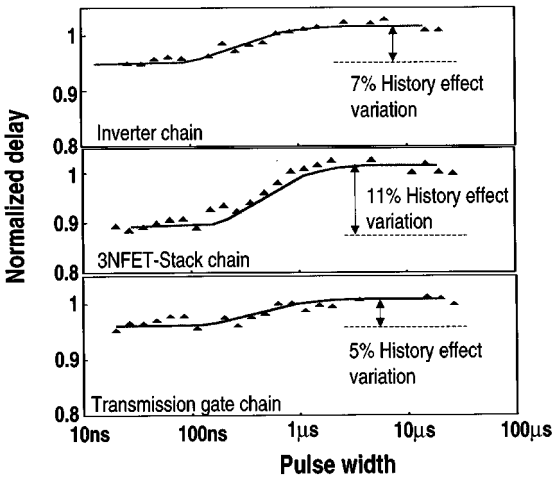


Fig. 12. History-effect measurements in 0.18-μm SOI technology.

further during transient operation due to the ac floating-body effects [6], [8].

B. SOI Simulation Model

The model used for circuit simulations was specially designed to handle partially depleted SOI structures and was built on top of a compact model for bulk devices. The core intrinsic model is based on an efficient direct solution of the surface potential equation [10] using the charge-sheet approximation [11]. The model features a single equation for the drain current valid for all regions of operations. In addition, single equations for all terminal charges are derived based on the same underlying approximations, resulting in self-consistent dc and ac models. Some of the features of the compact model include advanced short-channel effects [12], poly-depletion effect [13], quantum mechanical effect [14], and direct tunnel gate leakage [15].

Fig. 11 depicts the SOI device equivalent circuit used for circuit simulations. The impact ionization current I_m, the parasitic bipolar current I_c, and the source and drain junction leakage currents $I_{\rm bs}$ and $I_{\rm bd}$, are modeled with special care to capture the dc physical behavior of partially depleted SOI devices. In addition, the parasitic back oxide capacitances from source (Cs_sub), drain (Cd_sub), and body (Cb_sub) to the grounded substrate are calculated at run time based on the active area geometry and included in the circuit simulations.

C. History-Effect Measurements in 0.18-μm SOI

Fig. 12 shows the history-effect measurements obtained for 0.18-μm SOI test circuits that include chains of inverters, 3-nFET stacks and transmission gates (gates used in the ALU design). An input pulse was applied to these chains and the delay through the chain was measured. The graphs in Fig. 12 show the effect of width of the input pulse (x-axis) on the delay of the gates (y-axis).

Floating-body effects in SOI devices cause a history-dependent delay characteristic in SOI gates [8]. Thus, the delay of a gate depends on its activity prior to switching. The first switch delay is the delay measured for a gate that switches after a long period of inactivity (>5 μs). In this condition, the body voltage of the devices (and hence switching delay) is determined by the parasitic junction diode and parasitic bipolar currents. Immediately after the first switch, the body voltage of the device is determined by the magnitude of capacitive coupling from the gate/source/drain terminals. If the gate switches at this point, its delay (referred to as second switch delay) will differ from the first switch delay. The spread between first switch delay and second switch delay is referred to as history-effect induced delay variation.

In Fig. 12, the delay of the chain (y-axis) is normalized to the first switch delay. Measurements in our SOI process show that the second switch is always faster than the first switch. Further, as the input pulse width increases, the second switch delay approaches the first switch delay, confirming previously reported history-effect behavior [18]. History-effect induced delay variation (of 5% to 11%) has been measured in our SOI test circuits. These results correspond well with our simulation results, thereby establishing the validity of our model.

Since the history effect is seen to only cause a gate to speed up, it will not affect performance (max-delay). However, min-delays are affected, requiring a margin to be applied during min-delay timing analysis. (Max-delay and min-delay refer to worst-case and best-case delays through a datapath block). While the ALU is not affected by this min-delay margin; in general, timing tools will have to apply an 11% margin to all min-delay paths.

V. Direct Port to SOI

In our first "migration-to-SOI" option, we directly port the Han–Carlson ALU design from bulk CMOS to the SOI process described earlier. We did not strengthen the keepers on the dynamic nodes to handle the worst-case leakage conditions in SOI. The maximum 'history-effect induced noise' on the dynamic XNOR node was limited to 80 mV by appropriately sizing the 2-PMOS stack (Fig. 8). The directly-ported SOI design offers a 16% speedup (Table IV) over the bulk design, while the speedup of the adder core is 14%. This smaller speedup (compared to the 21% adder core speedup reported in [2]), is attributable to

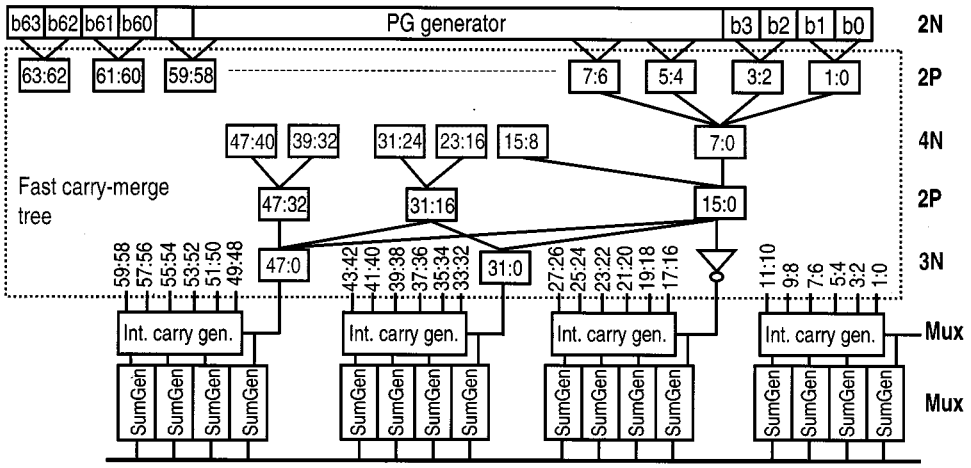


Fig. 13. Deep-stack quaternary-tree adder core: SOI-optimal redesign.

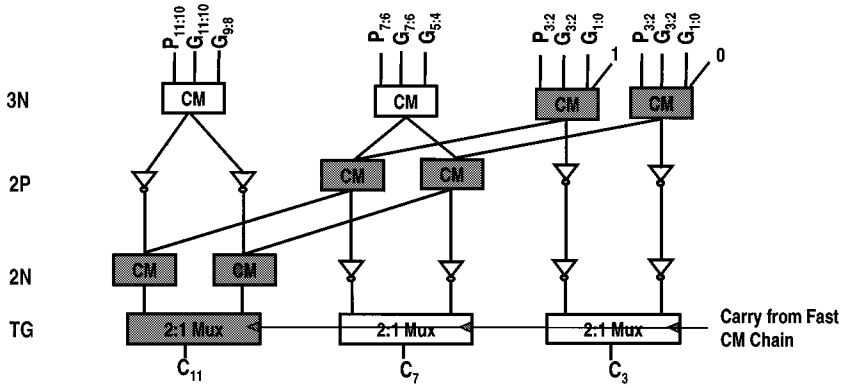


Fig. 14. Intermediate carry generator.

TABLE IV
DIRECT PORT TO SOI: SIMULATION RESULTS, $V_{cc} = 1.5$ V, $T = 110$ °C

64b Han-Carlson ALU delay simulations		Speedup over bulk
Bulk	482ps	16%
Direct-port to SOI	403ps	

TABLE V
BREAKUP OF SPEEDUP IN 0.18-μm TECHNOLOGY

Stage type	Speedup over bulk
Static gates	12-15%
Dynamic gates	2-9%
3:1 TG Mux	20%
5:1 TG Mux	23%
9:1 TG Mux	35%

the aggressive reduction of diffusion capacitances in our bulk process [5]. It should be pointed out that while the ALU is one of the critical paths in the processor, speedup in the ALU is not indicative of overall CPU speedup.

Table V shows how the speedup is distributed among the various components of the ALU. One of the key advantages of SOI technology is the reduction in device diffusion capacitance [1]. Therefore, as expected, we see the maximum speedup (35%) in the diffusion-dominated 9:1 transmission-gate multiplexer. The speedup reduces in the smaller multiplexers, and falls to as low as 2% in the load-dominated dynamic gate in the adder core. The static gates being more diffusion-dominated than the dynamic gates, show a greater speedup in SOI. Overall, these numbers average out to a 16% speedup for the whole ALU. Since the adder contains load dominated gates, the speedup obtained here is lower (14%).

VI. SOI-OPTIMAL REDESIGN

SOI technology offers features that expand the design space with respect to a bulk design, motivating an SOI-optimal redesign of the ALU. In particular, the absence of the classical body effect in SOI results in a lower stack penalty compared to bulk CMOS. Thus, increasing the stack height in SOI gates can offer additional speedup by reducing the total number of stages in the design [1], [6].

However, a "deeper-stack" design in adder architectures like Kogge–Stone and Han–Carlson will result in a proportional increase in gate fanouts, offsetting any speedup obtained from stage reduction. A quaternary-tree carry–select adder core [16] (Fig. 13) enables the use of deeper stacks, with a simultaneous reduction in gate fanouts and interconnect loads (by 1/3 and 1/2, respectively). This characteristic results in an SOI-optimal realization of the ALU.

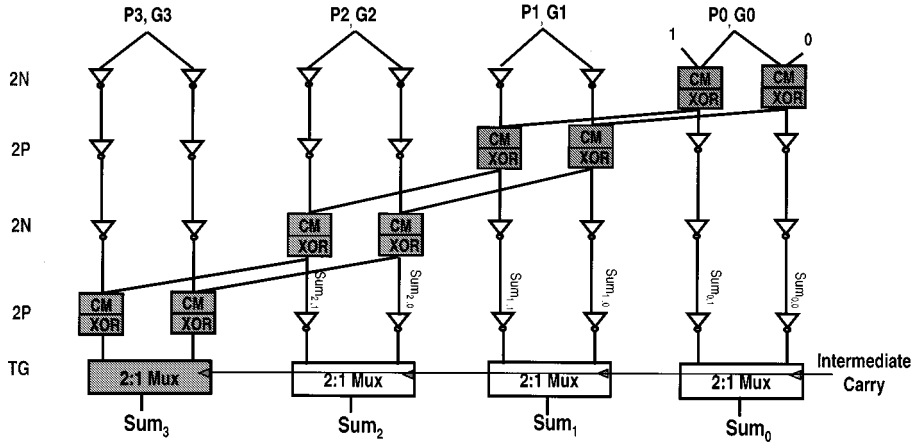


Fig. 15. 4-b conditional sum generator.

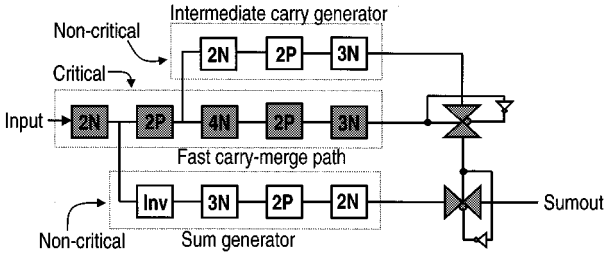


Fig. 16. Critical path of deep-stack quaternary-tree adder.

A. Deep-Stack Quaternary-Tree Architecture

The quaternary-tree architecture breaks up the carry–merge tree into two distinct sections: critical (1 in 16) carries and noncritical (1 in 4) conditional carries (Fig. 13). A sparse carry–merge tree is used to generate the critical carries (C_{15}, C_{31} and C_{47}). The unit fanouts on the generate gates of this tree is a 50% reduction in fanouts over the Han–Carlson implementation. This allows for a high-speed implementation of the fast carry–merge tree. Further, the 80% reduction in wiring density facilitates a compact layout.

Noncritical conditional carries are delivered by the intermediate carry generator (Fig. 14) which operates as a parallel sidepath. This "ripple-carry" block takes its inputs from the first stage of the fast carry–merge tree and delivers conditional carries to a 2:1 multiplexer stage. The output of the fast carry–merge tree serves as the MUX-select input to the intermediate carry generator, thereby generating one in four carries.

The 4-b sum generator (Fig. 15) is another parallel sidepath that generates 4-b conditional sums using a ripple-carry scheme. The output of the intermediate carry generator selects the appropriate 4-b conditional sum at the final 2:1 multiplexer stage.

The critical path of the deep-stack quaternary tree adder core is shown in Fig. 16. Deep stack gates are used in stage 3 (4-nMOS) and stage 5 (3-nMOS stack) of the fast carry–merge tree, resulting in a seven-stage implementation of the adder core. This is a two-stage reduction over the Han–Carlson implementation, with a simultaneous reduction in stage fanouts.

The ALU was redesigned with the deep-stack adder core, replacing the Han–Carlson core. The SOI-optimal deep-stack redesign provides an additional 5% speedup over the baseline bulk design (Table VI) bringing the maximum speedup obtained in the migration from bulk CMOS to SOI in 0.18-μm generation to 21%.

TABLE VI
ALU SIMULATION RESULTS IN 0.18-μm GENERATION, $V_{cc} = 1.5$ V, $T = 110$ °C

64b ALU delay simulations		Speedup over bulk
Bulk	482ps	-
Direct-port SOI	403ps	**16%**
SOI-optimal redesign	380ps	**21%**

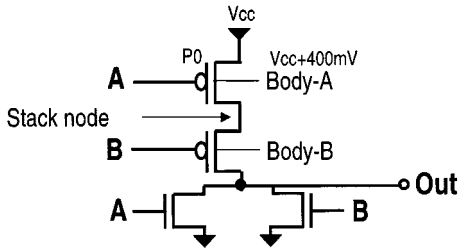


Fig. 17. Reverse-body bias in a static two-input NOR gate.

VII. MARGINING ISSUES IN SOI

While the forward-biased floating body in SOI contributes to a portion of the performance improvement over bulk [1], we observed that the body of an SOI device could also get reverse-biased during normal switching activity. This would increase the V_t of the device, resulting in a delay pushout. For example, it was seen that the input sequence $A = 11001, B = 01\,011$ applied to the inputs of a two-input static NOR gate (Fig. 17) could set up a reverse-bias of 400 mV on the upper pMOS transistor P0.

Figs. 18 and 19 shows the development of a reverse-bias in the two-input static NOR gate. (This gate is representative of the static evaluation path in the carry–merge tree.)

1) In the dc state (time $= 0$):
   - $A = 1, B = 0$.
   - Stack node voltage $= 130$ mV.
   - Body A: 1.35 V (150 mV forward bias).
   - Body B: 0 V (0 bias).

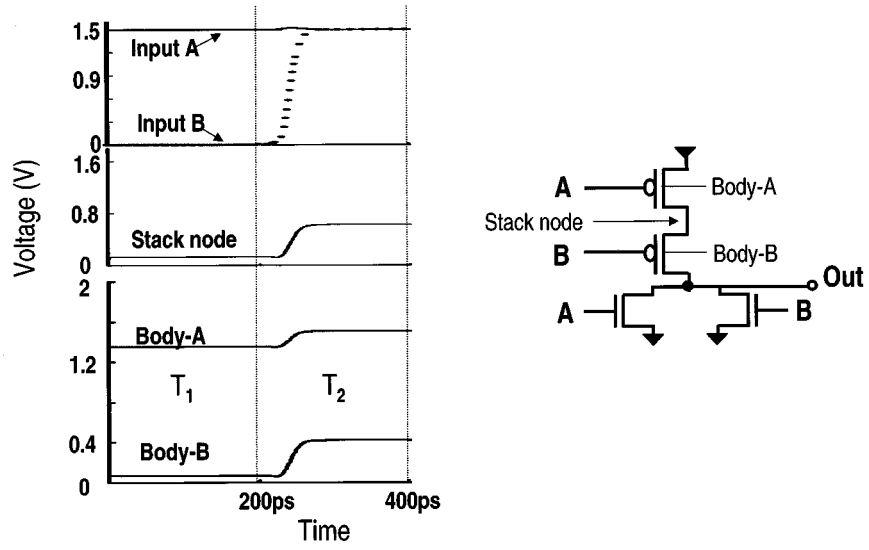


Fig. 18. Waveforms showing development of reverse-body bias in a static two-input NOR gate (Timesteps T1 and T2).

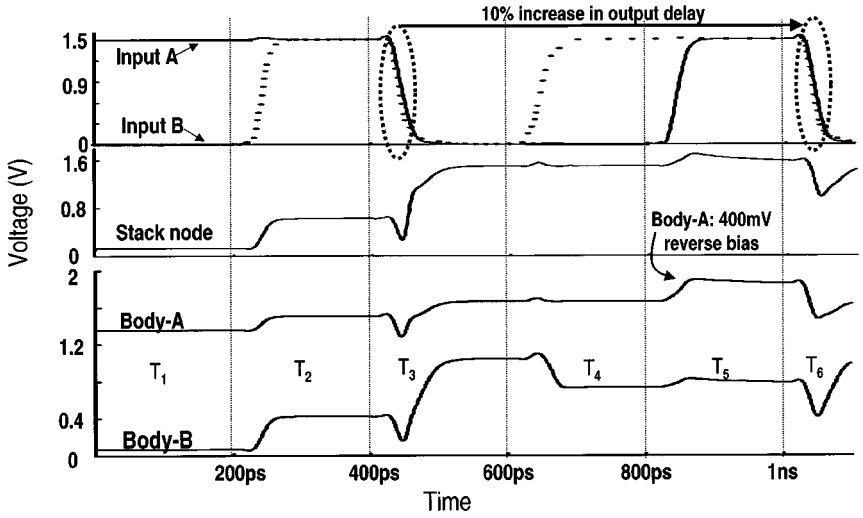


Fig. 19. Waveforms showing development of reverse-body bias in a static two-input NOR gate (Timesteps T1–T6).

2) In timestep T2, (400 ps $>$ time $>$ 200 ps), B transitions high. This rising transition couples onto body A and body B through stack node. This removes the forward bias that existed on body A and introduces a forward bias of 200 mV on body B.
3) In the next timestep T3, both A and B go low (Fig. 19), causing the output to transition high. The rising output transition couples onto body A and body B, creating a reverse bias of 170 mV on the upper transistor.
4) In timestep T4, B goes high. This increases the reverse bias on body B by 20 mV.
5) In timestep T5, A transitions high, coupling onto body A, and setting up a reverse body bias of 400 mV on the upper pMOS transistor.
6) Finally, when A and B go low, the output transitions high once again, as in timestep T3. However, the 400 mV of reverse body bias causes a delay pushout of 10%.

In a large design, it is difficult to determine the input sequences that may set up reverse-biases on various devices in the circuit. Therefore, we apply a 10% margin to all max-delay paths in SOI designs, reducing the overall SOI speedup from 21% to 11%.

A. Reducing Reverse-Bias Penalty

Reverse-bias penalty can be reduced in dynamic SOI gates by precharging the stack nodes using a clock transistor M1 (Fig. 20). This reduces the reverse-bias penalty from 10% to 2%. However, there is a 5% increase in clock energy due to the extra clock load. It should be noted that this is a point solution for dynamic gates only. While static gates would require the full 10% reverse body-bias margin, the max-delay margin on dynamic gates is reduced to 2%.

B. Sensitivity of Reverse Body-Bias to Model Calibration

As seen in Figs. 18 and 19, the primary cause of the reverse body-bias effect in SOI devices is due to the coupling between the drain/source and body (Timestep T3 & T5 in Fig. 19). Since the drain-bulk and source-bulk capacitances can only be indirectly calibrated from measured data, the sensitivity of the reverse-body bias to these capacitances must be quantified.

Fig. 21 shows that 20% variation in drain/source-bulk coupling capacitance results in at most 20-mV change in reverse body bias. Even a 60% error in ac model calibration would result in maximum error of 100 mV in simulated reverse body

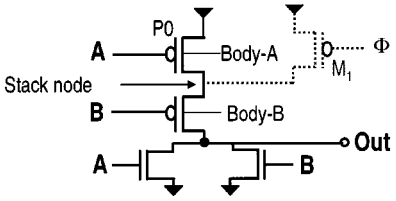


Fig. 20. Reducing reverse body-bias by preconditioning stack nodes in domino gates.

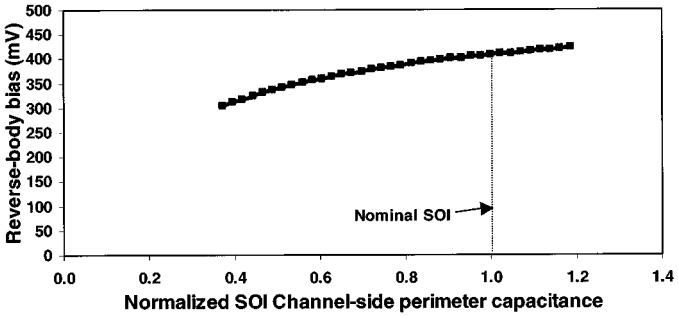


Fig. 21. Sensitivity of reverse body bias to AC model parameters.

TABLE VII
SOI SPEEDUP IN 0.18-μm TECHNOLOGY AFTER REVERSE-BIAS MARGINING
$V_{cc} = 1.5$ V, $T = 110$ °C

64b ALU delay simulations		Speedup over bulk	Speedup after margining
Bulk	482ps	-	-
Direct-port SOI	403ps	16%	**14%**
SOI-Optimal redesign	380ps	21%	**19%**

bias. This confirms the robustness of our reverse body-bias observations.

C. ALU Speedup After Reverse Body-Bias Margining

Applying the 2% margin to the SOI designs (Table VII), the speedup from the direct port to SOI falls from 16% to 14%. The speedup from the SOI-optimal redesign falls from 21% to 19%. Thus, the maximum SOI speedup in 0.18-μm generation is 19% for dynamic designs.

VIII. SCALING TO 0.13-μm TECHNOLOGIES

To quantify the scaling trends of ALU circuits in SOI both ALU designs were ported to 0.13-μm bulk/SOI technologies. As in the case of the 0.18-μm generation, $I_{\text{off-DC}}$ at room temperature for both processes were matched. Key MOSFET parameters and impact ionization data for the SOI model were obtained from 0.13-μm bulk measurements. Further, model fitting techniques for the SOI parasitic bipolar junction transistor (BJT) characteristics and junction diode characteristics were kept unchanged from the 0.18-μm SOI fitting.

The speedup obtained with a direct port of the Han–Carlson ALU from a 0.13-μm bulk technology to a comparable SOI process reduces to 11% (Table VIII). In the case of the SOI-optimal deep-stack ALU, speedup over bulk falls to 18%. This being a dynamic design, we apply the 2% reverse-bias margin

TABLE VIII
SOI SPEEDUP IN 0.13-μm TECHNOLOGY, $V_{cc} = 1.2$ V, $T = 110$ °C

64b ALU delay simulations		Speedup over bulk	Speedup after margining
Bulk	351ps	-	-
Direct-port SOI	312ps	11%	**9%**
SOI-Optimal redesign	286ps	18%	**16%**

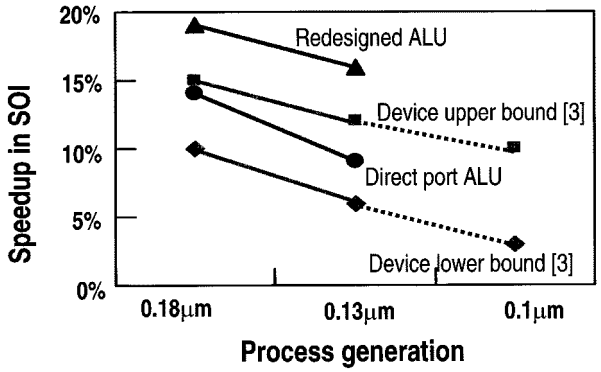


Fig. 22. Scaling trends of SOI speedup.

to the SOI designs, reducing the overall SOI speedup for the two architectures to 9% and 16%, respectively.

The ALU speedup offered by SOI reduces with scaling since the contribution of the diffusion capacitance as a percentage of total load capacitance decreases with scaling. The reduced diffusion capacitance of SOI devices has been shown to be the main factor contributing to speedup in SOI designs [17]. Reduction in the influence of this key advantage with scaling results in a diminishing trend in SOI speedup.

IX. SUMMARY AND CONCLUSION

We have presented an energy-efficient dynamic 64-b ALU operating at 482 ps in 0.18-μm bulk CMOS technology, with an adder core running at 310 ps. Novel circuits were developed to enable a single-rail implementation of the adder. A direct port of this design to 0.18-μm partially depleted SOI offered 14% speedup after margining. An SOI-optimal redesign of this ALU using a novel deep-stack quaternary-tree architecture increased the SOI speedup to 19% after margining. Margining was required, because we observed that the floating body of an SOI device could get reverse-biased in the course of normal switching activity. In the case of dynamic gates only, preconditioning the intermediate stack node reduced the reverse-bias margin from 10% to 2% (Static gates would require a 10% reverse body-bias margin). Scaling these designs to 0.13-μm generation, the speedup offered by SOI decreases. The maximum SOI speedup in 0.13-μm generation falls to 16%.

In conclusion, the speedup offered by partially depleted SOI technology, while significant in datapath circuits, shows a declining trend with scaling (Fig. 22). The best-case ALU SOI speedup falls from 19% in 0.18-μm to 16% in 0.13-μm generation. This correlates well with the device-level scaling trends forecast in [3].

ACKNOWLEDGMENT

The authors wish to thank S. Narendra and A. Keshavarzi for assistance with measured results. They also thank D. Ayers for organizing this effort. They also wish to acknowledge S. Borkar, J. Rattner, F. Pollack, W. Holt, S. Rusu and G. Singer for their encouragement and support.

REFERENCES

[1] G. G. Shahidi *et al.*, "Partially depleted SOI technology for digital logic," in *ISSCC Dig. Tech. Papers*, 1999, pp. 426–427.
[2] D. Stasiak *et al.*, "A second-generation 440-ps SOI 64-b adder," in *ISSCC Dig. Tech. Papers*, 2000, pp. 288–289.
[3] K. Mistry *et al.*, "Scalability revisited: 100-nm PD-SOI transistors and implications for 50-nm devices," in *Proc. Tech. Papers Int. Symp. VLSI Technology, Systems, and Applications*, 2000, pp. 204–205.
[4] T. Han *et al.*, "Fast area-efficient VLSI adders," in *Proc. 8th Symp. Computer Arithmetic*, Sept. 1987, pp. 49–56.
[5] T. Ghani *et al.*, "100-nm gate length high-performance/low power CMOS transistor structures," in *IEDM Tech. Dig.*, Dec. 1999, pp. 415–418.
[6] D. Allen, D. Behrends, and B. Stanisic, "Converting a 64-b power PC processor from CMOS bulk to SOI technology," in *Proc. 36th Design Automation Conf.*, 1999, pp. 892–897.
[7] L. Wang and H. Chen, "The conversion of bulk CMOS circuits to SOI technology and its noise impact," in *Proc. Tech. Papers Int. Symp. VLSI Technology, Systems, and Applications*, 1999, pp. 282–285.
[8] A. Wei, M. Sherony, and A. A. Antoniadis, "Effect of floating-body charge on SOI MOSFET design," *IEEE Trans. Electron Devices*, vol. 45, pp. 430–438, Feb. 1998.
[9] P. Lu *et al.*, "Floating-body effects in partially depleted SOI CMOS circuits," *IEEE J. Solid-State Circuits*, vol. 32, pp. 1241–1253, Aug. 1997.
[10] A. Boothroyd *et al.*, "MISNAN—A physically based continuous MOSFET model for CAD applications," *IEEE Trans. Computer-Aided Design*, vol. 10, pp. 1512–1529, Dec. 1991.
[11] J. R. Brews, *Physics of the MOS Transistor*, ser. Applied Solid State Science, Supplement 2. New York: Academic, 1981.
[12] Z. Liu *et al.*, "Threshold voltage model for deep-submicrometer MOSFETs," *IEEE Trans. Electron Devices*, vol. ED-40, pp. 86–95, 1993.
[13] R. Rios *et al.*, "An analytic polysilicon depletion effect model for MOSFETs," *IEEE Electron Device Letters*, vol. EDL-15, pp. 129–131, 1994.
[14] R. Rios *et al.*, "A physical compact model, including quantum mechanical effects, for statistical circuit design applications," in *IEDM Tech. Dig.*, 1995, pp. 937–940.
[15] K. M. Cao *et al.*, "BSIM4 gate lekage model including source-drain partition," in *IEDM Tech. Dig.*, 2000, pp. 815–818.
[16] R. Woo, S. Lee, and H. Yoo, "A 670-ps 64-b dynamic low-power adder design," in *Proc. Int. Symp. Circuits and Systems*, vol. 1, May. 2000, pp. 28–31.
[17] S. Narendra, J. Tschanz, A. Keshavarzi, K. Mistry, T. Ghani, S. Borkar, and V. De, "Comparative performance, leakage power and switching power of circuits in 150-nm PD-SOI and bulk technologies including impact of SOI history effect," in *Proc. Tech. Papers Int. Symp. VLSI Circuits*, 2001, pp. 217–218.
[18] N. Rohrer and K. Bernstein, "SOI for today's integrated circuits," *Proc. 13th Annu. IEEE Intl. ASIC/SOC Conf.*, p. 409, Sept. 2000.

Sanu Mathew (M'00) received the B.Tech degree from the College of Engineering, Trivandrum, India, in 1993 and the Ph.D. degree in electrical engineering from the State University of New York, Buffalo, in 1999. His Ph.D. dissertation was focused on asynchronous circuit design.

He is currently part of the high-performance circuits research group at Intel Corporation's Microprocessor Research Labs, Hillsboro, OR.

Ram K. Krishnamurthy (S'92–M'98) received the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, in 1998, where his dissertation research was focused on low-power DSP circuit design.

Since graduation, he has been with Intel Corporation's Microprocessor Research Labs, Hillsboro, Oregon, where he is currently leading the high-performance circuits research group. He is an Adjunct Faculty of Department of Electrical and Computer Engineering, Oregon State University, Corvallis, where he teaches VLSI system design. He serves on Intel's SRC review committee and the ASIC, CICC, and ISCAS conference program committees. He holds six patents and has published 15 papers.

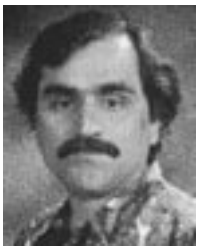
Mark A. Anders (M'99) received the B.S. degree in 1998 and the M.S. degree in 1999, both in electrical engineering from the University of Illinois, Urbana-Champaign.

Since graduation, he has been with Intel Corporation's Microprocessor Research Labs, Hillsboro, Oregon, where he is working on high-performance circuits research.

Rafael Rios (M'85) received the Ph.D. degree in electrical engineering from Drexel University, Philadelphia, PA, in 1990.

From 1986 to 1992, he was with the David Sarnoff Research Center, Princeton, NJ, where he was involved in process and device modeling and simulation for radiation-hardened devices. From 1992 to 1996, he was with Digital Equipment Corporation, Hudson, MA, working in the area of device modeling for digital circuit design. He joined Intel Corporation, Hillsboro, OR, in 1996, where he is currently working on the development of compact device models for microprocessor design.

Kaizad R. Mistry (M'84), photograph and biography not available at time of publication.



K. Soumyanath (M'93) received the B.E. degree in electronics and communication engineering from the Regional Engineering College, Tiruchirappalli, India, in 1979, the M.S. degree in electronics from the Indian Institute of Science, Bangalore, in 1985, and the Ph.D. degree in computer science from the University of Nebraska, Lincoln, in 1993.

He was a Member of the faculty with Tufts University, Medford, MA, until 1995, where he served as the Director of the ARPA supported program in Mixed Signal IC Design for the Department of Defense. Since 1996, he has been with Intel Corporation, Hillsboro, OR, where he is a Principal Engineer and currently leads the CMOS communications circuits program for Intel's Internet Systems and Circuits Research Labs. His previous responsibilities included leading several high-speed digital research projects for the Circuits Research Laboratory. In 1998, he served as the Chair of the Design Sciences task force for the Semiconductor Research Corporation and currently serves on the technical program committee for ICCD. He has published over 15 papers on VLSI, and has six patents issued.