

Novel Low-Overhead Operand Isolation Techniques for Low-Power Datapath Synthesis

N. Banerjee, A. Raychowdhury, S. Bhunia, H. Mahmoodi, and K. Roy
School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN
Email: {nbanerje, araycho, bhunias, mahmoodi, Kaushik}@ecn.purdue.edu

Abstract: Power consumption in datapath modules due to redundant switching is an important design concern for high-performance applications. Operand isolation schemes are adopted to reduce redundant switching in datapaths. However, they incur considerable overhead in terms of delay, power, and area. This paper presents novel operand isolation techniques based on supply gating that reduce the overheads associated with isolating circuitry. The proposed schemes also target leakage minimization and application of operand isolation at the internal logic of datapath to further reduce power consumption. We integrate the proposed techniques and power/delay models to develop a complete flow for low-power datapath synthesis. Simulation results show that the proposed operand isolation techniques can achieve at least 40% reduction in power consumption compared to the original circuit with minimal area overhead (5%) and small delay penalty (0.15%).

1. INTRODUCTION

Present day circuit designs contain many datapath modules which occasionally perform useful computations but spend a considerable amount of time in the idle states. However, the switching activity at their inputs in their idle states causes redundant computations which are not used by the downstream circuit. This redundant switching significantly increases the power consumption. Operand isolation is an effective technique that prevents unnecessary switching in a module by utilizing isolation circuitry at the input of the module. Enabling the isolation circuitry forces the modules in their idle states. Leakage power, however, becomes an important issue in this idle state of the module. Therefore, it should be ensured that the isolation circuitry be designed in a way that the isolated module consumes minimal leakage power as well.

Operand isolation was first introduced in IBM PowerPC 4xx-based controllers [8] where it was applied manually within a local boundary for isolation of multiplexer steered modules using the multiplexer select signal as the controlling signal. Precomputation-based methods have been applied to turn off sequential circuits based on their precomputed value from a certain number of input bits [10]. However, these methods not only require extra area to route the bits to the pre-computation logic but also duplication of the target circuit for multi-output circuits. Moreover, for modules like adder and multipliers this method requires utilization of all the bits to compute the pre-computation signal and hence the pre-computation logic might consume an area equivalent to the pre-existing logic in those modules. Tiwari et al. proposed an operand isolation methodology at the RT-level named “guarded evaluation” [9]. This method isolates a circuit by introducing transparent latches on the inputs to the arithmetic blocks and utilizes existing signals in the circuit as control signals for activating the latches. However, the applicability of this technique is limited by the existence of such signals. Furthermore, it is difficult to implement the logic for automatic selection of latches for large designs and there is also significant area overhead for placing latches in wide datapaths. Kapadia et al. presented a technique for saving power dissipation in large datapath buses by preventing switching-activity in the bus driver modules [1]. In this scheme, insertion of extra latches to block transition activity was avoided by utilizing the enable signals of the steering

modules (registers, multiplexers, tri state buffers) as the isolating signals. However, this method is unable to provide optimal isolation in multiple fan-out steering modules. For instance, in case of a multiplexer based isolation this method always has to select one of the two inputs for any switching activity in them even when the computation is redundant. This method also does not provide power savings in combinational blocks that are directly connected to primary inputs.

Munch et al. [2] addressed some of the previous limitations and presented a low overhead AND/OR based isolation technique, which could also be applied to obtain power savings from blocks that are fed from the primary inputs. However, leakage power reduction is not addressed while designing their isolation circuitry. In this paper, we make the following contributions:

- Novel isolation circuitry based on the concept of supply gating that incurs significantly lower design overhead compared to existing implementations.
- Application of operand isolation at finer circuit-level granularity to prevent redundant switching inside datapath modules such as comparators and carry-select adders that suffer from redundant computations.
- Isolation techniques for cases where control signal gating is non-optimal for operand isolation (e.g. multiplexer logic where one of the operands has to be chosen during computation).
- Minimization of leakage power consumption in the idle state.
- Integration of the proposed operand isolation techniques to derive a complete synthesis flow at RT-level with power and delay calculation models.

2. NOVEL LOW-OVERHEAD ISOLATION CIRCUITRY

As mentioned in Section I, operand isolation techniques may pose significant area overhead, power consumption and delay penalty. Therefore, it is extremely important to design low-overhead circuits to effectively provide operand isolation so that the savings obtained through active power reduction is not offset by the other factors (area, delay, leakage power). In this section, we present a set of novel isolation techniques which have been designed to show significantly less overhead in terms of die-area, delay and power compared to the existing schemes.

2.1. Input-Isolating Multiplexer (I^2 -MUX)

Multiplexers (MUX) before datapath modules are common due to resource sharing of complex and large modules. Multiplexers can be effectively utilized to prevent redundant switching on the datapath modules (functional blocks) by isolation (blocking) of operands from inputs of the functional blocks.

If a conventional MUX is used for operand isolation, the MUX has to select the input that does not switch over a period of time [1]. This limits the possibility of isolation for MUX driven functional units, because the switching of an input operand of the MUX depends on the functional block generating it. Insertion of gating logic (such as latch and OR gates) at the interface from the MUX to a functional block provides more opportunities for operand isolation. That is because in this case the gating logic can prevent the switching at the input of the functional block irrespective of the state of the operands at the inputs of

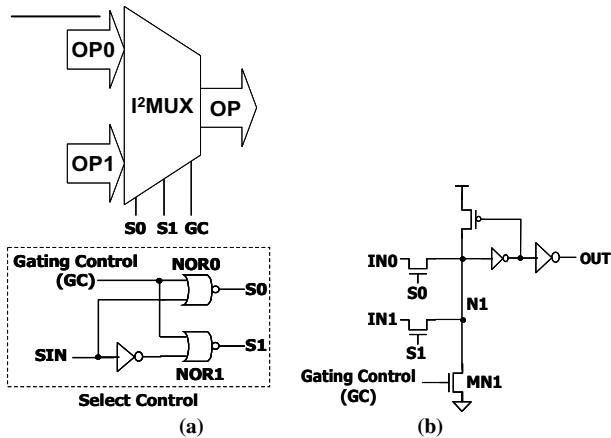


Fig. 1. Schematic of 2-to-1 I²-MUX : (a) MUX and select control; (b) bit slice of I²-MUX unit (I²F-MUX with output forced to '0')

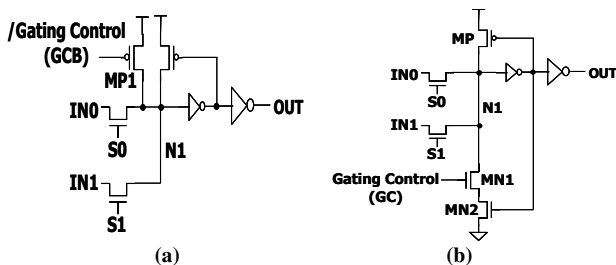


Fig. 2. (a) I²F-MUX with output forced to 1 (b) I²H-MUX with output hold the MUX. However, the extra gating logic can have significant area, power, and delay penalty in the normal mode of operation.

We propose a new MUX circuit, called Input-Isolating Multiplexer (I²-MUX), that provides a gating state in which none of the inputs are directed to its output and the state of the output is forced either to '0' or '1' (I²F-MUX), or the output is held at its previous value before going to the gated mode (I²H-MUX). The gated mode provides a new state for the I²-MUX resulting in three states for a 2-to-1 multiplexer. The schematic of the I²-MUX is shown in Fig. 1. The inverter buffers of the conventional select control (decoder) are replaced by the NOR gates (Fig. 1(a)). The second set of inputs of the added NOR gates are driven by the Gating Control (GC) signal. If the GC signal is low, the MUX operates similar to a conventional MUX.

If the GC signal is high, both select signals (S0, S1) are forced to zero irrespective of the state of the select input (SIN). This state is not allowed in a conventional MUX. However, in the I²-MUX, this state isolates the output of the MUX from its inputs. If the GC signal is '1', S0 and S1 are both low and transistor MN1 is ON pulling the node N1 to low, forcing the state of the output of the MUX to '0'. A minimum sized transistor for MN1 is large enough to be able to pull N1 to a low value in the gated mode. Therefore, the state of the output of the MUX is isolated from the operands (OP0 and OP1). This scheme does not need any extra control signals and also does not add any gating logic in the datapath. The only delay overhead can be due to slight increase in the capacitance of the MUX internal node (N1) by the diffusion capacitance of the minimum sized transistor MN1, and replacement of the inverters in the select unit with the NOR gates (NOR0 and NOR1 in Fig. 1(a)). This delay and power overhead in this method is definitely much smaller than the overhead due to insertion of extra gating logic on all the output bits of the MUX. Moreover, the added transistor (MN1) is a minimum sized transistor and the gating logic (NOR0 and NOR1) is in the select control path and shared by all bit slices of the MUX unit further minimizing the area and power overhead.

In the I²F-MUX shown in Fig. 1(b), the state of the output of the MUX is forced to '0' in the gated mode. The I²F-MUX can also be designed to force the state of its output to '1' (Fig. 2(a)) in the gated mode. In this case, the inverted gating control (GCB) signal is applied to a pull-up PMOS (MP1) to pull up the internal node (N1), and therefore the output (OUT), to '1' in the gated mode. Fig. 2(b) shows the I²-MUX with hold capability at the output (I²H-MUX). I²H-MUX holds the state of its output in the gated mode. In the gated mode when GC is high (MN1 is ON), transistors MN2 and MP1 form an inverter holding the state of the internal node N1 by a cross coupled inverter action. The added transistors (MN2, MN1, and MP1) are all minimum sized transistors, resulting in minimal area, power, and delay overhead.

Operand isolation using I²F-MUX with output forced to a '0' or '1' value results in an extra switching on the functional block for switching from the previous state to the forced state (similar to operand isolation with AND/OR gates). Therefore, no energy savings is obtained if the gated mode is applied for only one clock cycle. The advantage of I²H-MUX is that it blocks the input switching without forcing inputs to any particular state. Therefore, there can be energy savings even if it is applied for one clock cycle.

2.2. Using supply gating for operand isolation

For circuits where the inputs of a datapath module are not provided through multiplexers or latches, extra masking logic (latch or AND/OR) is added for operand isolation [2]. This extra logic creates significant area overhead, delay and power penalty in the normal (non-gated) mode. To reduce the overhead, we propose the use of supply gating in a way suitable for operand isolation.

2.2.1. First Level Supply-Gating (FLS) Operand Isolation Scheme

First Level Supply gating (FLS) insertion technique is originally proposed in [4], where only the first level logic gates are gated using supply gating transistors. Insertion of the gating transistor in the first level logic screens the rest of the combinational logic from the input transitions, and therefore provides operand isolation. In [4], FLS is used as a low power scan test technique. Adding an extra transistor at only one logic level renders significant advantages with respect to area, delay and power overhead compared to previous methods, which use gating logic at each of the inputs.

Among the various FLS schemes, the gated-GND scheme proposed in [4] is most suitable for supply gating due to smaller area overhead and less delay and power penalties. *In this paper, we have used the first level supply gating strategy for operand isolation.* Fig. 3 shows the proposed FLS based operand isolation technique applied to a general datapath module. For the implementation of the gating transistors, all first level gates share a single gating transistor through a virtual ground node. By sharing the supply gating transistor, area overhead can be reduced because a shared supply gating transistor can have less size

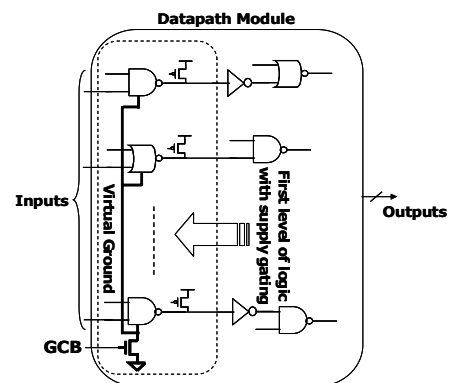


Fig. 3. FLS operand isolation scheme

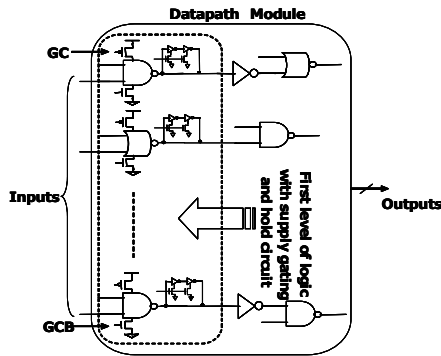


Fig. 4. FLH operand isolation scheme

than the sum of the sizes of all supply gating transistors in the unshared case.

2.2.2. First Level Hold (FLH) Operand Isolation Scheme

Similar to OR/AND based isolation techniques, FLS based operand isolation cannot prevent one redundant switching at the input of the datapath module when switching to the gated mode. In the FLS scheme, the states of outputs of first level gates are forced to '0' or '1'. Applying a fixed state causes a redundant switching on every transition to the gated mode due to which there will not be any energy savings if the isolation is applied for a period of only one clock cycle. *In this section, we develop an operand isolation technique based on supply gating that prevents extra switching by holding the state of the first level gates in the supply gated mode.* To implement this technique, the output of the first level gates needs to be held at their initial values when applying this method. This can be achieved by adding a latch element (cross-coupled inverters) at the output node. The latch element needs to be enabled only in the gated mode to hold the output state of the first level gate. This scheme is called First Level Hold (FLH) and is used in [5] for low power delay fault testing as an alternative to the enhanced scan based delay fault testing, with significantly less design overhead. Fig. 4 shows the above method applied to a general circuit for isolation. In this scheme, the sharing of supply gating transistors is not possible because the outputs of first level gates may store different values.

3. COMPARISON WITH EXISTING OPERAND ISOLATION TECHNIQUES

To estimate the effectiveness of the proposed operand isolation schemes, we simulated a set of datapath benchmark circuits using BPTM 70nm models (to observe sub-100nm effects) and obtained power and performance in normal mode of operations and area overhead due to operand isolation circuit. The gate-level netlists were first technology-mapped to a LEDA 0.25 μ standard cell library [6] using Synopsys design compiler with the mapping effort at medium. The benchmark circuits are then translated to Hspice and scaled to 70nm. Power is measured in NanoSim by applying 200 random vectors to the inputs and delay is measured by Hspice simulation of the critical paths of a circuit.

We consider two scenarios for our comparisons: (a) if datapath modules are preceded by multiplexers, the conventional OR and latch (inserted between multiplexer and datapath module) based operand isolation techniques are compared with the proposed I²F-MUX technique (Fig. 1, 2), and (b) if datapath modules are not preceded by multiplexers, the conventional OR and latch (inserted at the inputs of datapath modules) techniques are compared with the proposed FLS and FLH operand isolation schemes (Fig. 3, 4). Table I to III show the results of comparisons of the various techniques (area, delay, power).

Table I compares of these techniques in terms of area overhead. Since the layout rules for the 70nm node are not available, the measure used for area is the total transistor active area ($W \times L$ for a transistor) for the

different implementations. The proposed I²F-MUX technique exhibits the smallest area overhead for all datapath circuits. It shows 92.8% reduction in area overhead as compared to the existing OR-based gating technique, which has the least area penalty among the conventional techniques. I²H-MUX also shows significant reduction in area overhead compared to the latch-based gating. It can be noted that, for the OR or latch-based method, area overhead is proportional to the number of inputs of the datapath module. However, in FLS, gating logic is inserted in all first level gates (Fig. 3), the number of which depends on the number of first level gates of the datapath module. Therefore, for a datapath module with large number of first level gates, such as multiplier, there will be additional area overhead when implementing operand isolation utilizing FLS/FLH schemes (Table I (b)).

Table II shows comparative impact of the existing and proposed operand isolation techniques on circuit delay for different benchmarks. It is observed that the OR-based gating has the largest increase in delay. Compared to the OR-based gating, I²F-MUX exhibits delay overhead reduction of up to 47%, when compared to the latch-based gating. As observed from table II, the delay overhead of the FLS technique is less than 0.4% for all the benchmark circuits. Compared to the OR and latch-based gating, FLS and FLH techniques exhibit significant delay overhead reduction.

Table III compares the power consumption for the various implementations in normal mode of operation. The I²-MUX techniques have considerable (>90%) reduction in power overhead compared to the conventional techniques. The power dissipation of the FLS circuits is very close to the power dissipation of the original combinational circuit without any gating technique because in FLS the gating transistor and the pull-up PMOS do not switch in the active (normal) mode. Interestingly, for large benchmark circuits such as multiplier, the power of the FLS circuit is even less than the power of the original circuit (negative overhead or gain) because the gating transistor results in leakage reduction (due to stacking effect [3]) for the idle gates. This leakage is called *active leakage* since it occurs in the active mode for the idle gates and it is a significant part of the overall active power in the 70nm technology node. Reducing the active leakage on the first level gates can result in overall power reduction for large circuits. FLS shows overall power reduction of up to 127% compared to the OR-based technique.

4. LEAKAGE REDUCTION IN OPERAND-ISOLATED MODE

Leakage reduction during active mode is a concern for modern day designs. In this section, we explain how our operand isolation techniques can be used for significant savings of active leakage power compared to existing implementations. In the gated mode, the functional block does not switch; however, it still dissipates power due to standby leakage which becomes significant in scaled technologies [3]. Leakage of a combinational circuit is a strong function of the state of its inputs [7]. Therefore, by selecting the best input vector for a combinational circuit in standby mode, its leakage power can be significantly reduced. In this section, we show that leakage reduction is an additional advantage of the I²F-MUX and FLS operand isolation techniques on top of the benefits in terms of area, delay, and power.

4.1 Mixed I²F-MUX Operand Isolation Scheme

By selective use of I²F-MUX with output forced to '0' (Fig. 1(b)) and I²F-MUX with output forced to '1' (Fig. 2(a)) on individual inputs, the best input vector for leakage minimization in the gated mode can be applied to the datapath modules that are preceded by multiplexers. We refer to this isolation method as mixed I²F-MUX. If the leakage energy dissipation during the gated (standby) mode is larger than the energy associated with one extra switching associated with the use of I²F-MUX, it would make sense to apply mixed I²F-MUX to save leakage by applying the best vector to the functional block in the gated mode. The

Table I. Comparisons of area overhead of operand isolation techniques

Datapath module	(1) Datapath module preceded by MUX						(2) Datapath module not preceded by MUX					
	Conventional		Proposed				Conventional		Proposed			
	OR	Latch	I ² F-MUX		I ² H-MUX		OR	Latch	FLS		FLH	
			Area	%improv. over OR	Area	%improv. over latch			Area	%improv. over OR	Area	%improv. over latch
Comparator	25.8	47.3	1.9	92.6	2.8	94.0	45.2	82.9	18.7	58.6	56.2	32.2
Adder	12.6	23.1	0.9	92.8	1.4	93.9	16.0	29.3	13.3	16.9	40.1	-36.8
Multiplier	0.4	0.8	0.01	97.5	0.01	98.7	0.4	0.8	3.1	-675.0	9.2	-1050.0

Table II. Comparisons of delay overhead of operand isolation techniques

Datapath module	(1) Datapath module preceded by MUX						(2) Datapath module not preceded by MUX					
	Conventional		Proposed				Conventional		Proposed			
	OR	Latch	I ² F-MUX		I ² H-MUX		OR	Latch	FLS		FLH	
			Delay	%reduction from OR	Delay	%reduction from latch			Delay	%reduction from OR	Delay	%reduction from latch
Comparator	4.1	2.4	1.2	71.5	1.4	41.8	3.8	2.1	0.4	89.3	0.8	60.6
Adder	2.4	2.0	0.4	82.1	1.1	47.3	2.2	1.7	0.0	100.0	0.0	102.7
Multiplier	1.4	1.0	0.2	82.6	0.7	25.6	1.9	0.9	0.3	86.9	0.0	97.2

Table III. Comparisons of power overhead (normal mode) of operand isolation techniques

Datapath module	(1) Datapath module preceded by MUX						(2) Datapath module not preceded by MUX					
	Conventional		Proposed				Conventional		Proposed			
	OR	Latch	I ² F-MUX		I ² H-MUX		OR	Latch	FLS		FLH	
			Pow.	%reduction from OR	Pow.	%reduction from latch			Pow.	%reduction from OR	Pow.	%reduction from latch
Comparator	41.9	52.0	3.0	92.7	0.6	98.9	117.2	201.5	3.3	97.2	38.6	80.8
Adder	16.8	29.6	3.5	79.3	0.2	99.3	44.1	71.0	1.2	97.3	32.8	53.8
Multiplier	27.8	28.2	0.4	98.6	0.1	99.6	6.1	6.6	-1.7	127.5	3.9	40.0

Table IV. Comparisons of leakage power (μ W) for different operand isolation schemes in gated mode

Datapath module	(1) Datapath module preceded by MUX						(2) Datapath module not preceded by MUX					
	Conventional		Proposed				Conventional		Proposed			
	OR	Mixed AND/OR	I ² F-MUX (out='0')		Mixed I ² F-MUX		OR	Mixed AND/OR	FLS (Gated GND)		Mixed FLS	
				%reduction from OR		%reduction from mixed AND/OR				%reduction from OR		%reduction from mixed AND/OR
Comparator	46.7	45.4	18.7	60.0	13.9	69.4	9.1	7.8	6.4	30.0	5.0	35.4
Adder	58.1	56.9	33.8	41.8	24.8	56.5	20.5	19.3	18.7	8.6	13.8	28.3
Multiplier	648.5	638.9	574.0	11.5	401.8	37.1	611	601.2	560.0	8.3	555.1	7.7

decision whether to use I²H-MUX or mixed I²F-MUX with output forced to best vector depends on the relative magnitude of leakage power with respect to the switching component of power and also the cycle time. It is worth noting that longer the cycle time, the larger the ratio of leakage power to the switching power.

4.2 Mixed FLS Operand Isolation Scheme

The OR-based operand isolation technique fixes the state of all inputs to '1' in the isolated mode, which might not be the best input vector that minimizes overall leakage power for the module. For latch-based operand isolation, the state of the inputs cannot be set to the best vector since the inputs are fixed at their right state before going to the gated mode. However, AND and OR gating together can provide the best input vector for the datapath module by OR masking the inputs that are to be at logic state of '1', and AND masking the inputs that are to be at the logic state of '0'. However, even though the mixed AND-OR

system forces the inputs to be in minimum-leakage states, the blocking gates (AND, OR) themselves dissipate considerable leakage power.

In the proposed FLS operand isolation scheme, the outputs of all first level gates are forced to logic level '1' or '0', respectively. However, this state of inputs may not correspond to the best input vector for minimum leakage. By selective use of gated-GND or gated-VDD [4] for individual inputs, the state of the datapath module can be assigned to the best input vector during operand isolation to minimize leakage. This scheme is called mixed FLS operand isolation scheme.

4.3. Results and comparison of power in gated mode

The results of leakage reduction by input vector control using mixed OR/AND, mixed I²F-MUX, and mixed FLS for different benchmark circuits are shown in Table IV. The best input vectors are found using algorithms described in [7]. Depending on the benchmark, significant savings can be achieved by applying the best input vector using mixed I²F-MUX (module preceded by multiplexer) and mixed FLS (module

not preceded by multiplexer). The mixed I²F-MUX operand isolation technique shows improvements of 69%, 56%, and 37% in leakage power compared to the mixed AND/OR based technique for the benchmarks. The mixed FLS technique shows improvements of 35%, 28%, and 7.7% in leakage power compared to the mixed AND/OR based technique. The FLS technique eliminates the extra gating logic circuits (AND/OR) and also reduces the leakage of first level gates due to the stacking effect [3], improving the power dissipation. Due to the exponential increase of leakage with technology scaling and temperature increase, the leakage reductions of the mixed I²F-MUX and mixed FLS become more effective as the technology scales or the temperature increases.

5. OPERAND ISOLATION AT BIT-LEVEL

Operand isolation techniques described in Section II achieve active power reduction by preventing redundant computations in modules, and forcing them to their idle state. However, it is possible to apply certain isolation techniques to achieve further power reduction even while the circuit is doing useful computations. In this section, we introduce a novel methodology for reducing redundant switching in datapath modules (comparator, carry select adder) by efficient supply/GND gating at the bit-level, even when they are performing useful computations for downstream circuits.

5.1 Operand Isolation for Comparator circuit

Consider the design of a 3 bit comparator. The Boolean logic for output Y in SOP (sum-of-product) form (Fig. 5(a)) is:

$$Y = A_2 \overline{B_2} + (A_2 \overline{B_2}) \cdot A_1 \overline{B_1} + (A_2 \overline{B_2}) \cdot (A_1 \overline{B_1}) \cdot A_0 \overline{B_0} \quad (2)$$

When $A_2=1$ and $B_2=0$, the first term of (2) is 1 and hence the computation of the second and third terms are redundant. To avoid this redundant switching, we use GND gating on the NAND gate 8, using

$A_2 \overline{B_2}$ as shown in Fig 5(b). It can be noted that the inputs to the NAND gate have at least a two gate delay whereas the gating transistor has a single gate delay. As a result it can effectively remove redundant switching in the path marked in Fig 5(b), when $A_2=1$ and $B_2=0$. In general, for any comparator a part of the redundant switching in the path where A_n and B_n are compared, can be eliminated by GND gating with $A_{n-2} \overline{B_{n-2}}$. GND gating gives the added advantage of leakage power

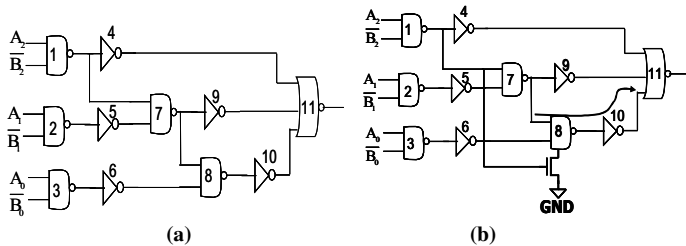


Fig 5: (a) Schematic of a SOP implementation of comparator; (b) Reduction of redundant switching by GND-gating

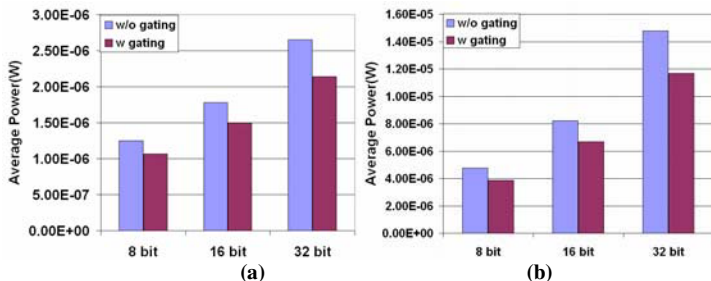


Fig 6: Average power of 8, 16 and 32 bit comparators with and without bit-level gating at clock frequencies (a) 100MHz; (b) 500MHz

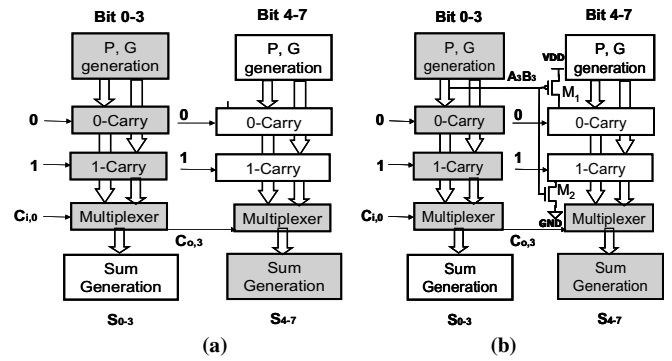


Fig 7: (a) 8-bit carry select adder; (b) Reduction of redundant switching by supply gating

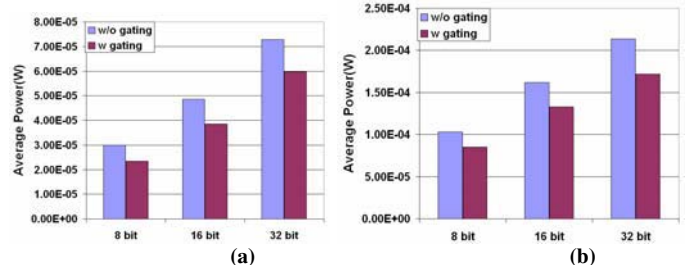


Fig 8: Average power of 8, 16, and 32 bit carry select adders with and without bit-level gating at clock frequencies (a) 100MHz; (b) 500MHz

reduction in the active mode. The improvement in average power for 8, 16 and 32 bit comparators simulated (using BPTM 70nm transistor models) with and without bit-level gating for three different frequencies of operation is shown in Fig. 6. It can be noted that an average power reduction of 18.5% was obtained by efficient GND gating of the comparators. The corresponding average delay increase in the comparators is approximately 4.5%. The methodology is applicable to POS (product-of-sum) implementation also with slight modifications.

5.2 Operand Isolation for Carry-Save adder circuit

Redundant switching can be partly eliminated in a carry select adder (CSA) by selective GND gating. To demonstrate this let us consider an eight-bit CSA which has been split up into two four bit ripple carry full adders (Fig. 7(a)). The topmost block is the propagate (P) and carry generator (G) blocks. The critical path of the circuit has been shaded. When both A_3 and B_3 are '1', carry propagated to the second stage will always be '1' and switching in the 0-carry block for the bits 4 to 7 is redundant. Similarly, if both A_3 and B_3 are '0' then switching in the 1-carry block is redundant. To eliminate this redundant switching we can use NMOS GND gating of the 1-carry block and PMOS supply gating of the 1-carry block by using the propagate signal ($P_3 = A_3 \cdot B_3$). When $P_3 = 1$, the transistor M_1 is turned off thereby eliminating switching in the logic block 3. Similarly when P_3 is '0', it turns off M_2 and eliminates switching in the 1-carry block. If we consider that all the bits can be 0 or 1 with equal probability then, this technique can remove redundant switching in the second stage (bits 4-7) in 50% of the cases. It should be noted that the same technique can be used to supply/GND gate the stage 1 (bits 0-3) with the gating control being the input carry in $C_{i,0}$. Simulations were carried out on 8, 16 and 32 bit carry select adders at three different frequencies (Fig. 10) and results show an average power reduction of 20%. It can be noted that the supply/GND gating transistors of the second and the subsequent stages are added in the non-critical path of the circuit.

Hence, if supply/GND gating is not used in the first stage then there is no performance penalty in our proposed technique. However, if supply/GND gating technique is used in the first stage of the circuit too

(corresponding to the simulation results in Table IV), then there is an overall delay increase of approximately 2%.

6. INTEGRATED SYNTHESIS FLOW

We have developed a complete synthesis methodology for integrating the application of the I²-MUX-based, FLS and FLH-based and bit-level operand isolation techniques at the RT-level. The complete design flow for the insertion of isolation circuitry is shown in Fig. 9. First, we partition the RTL-level circuit into modules based on sequential boundaries and perform isolation on the combinational logic bounded by sequential logic or those connected to the primary inputs. Our assumption in this case is that logic circuits across sequential boundaries do not affect each other. The idle condition for the outputs of each partition is then determined. In the next step, we generate the gating control signals using precomputation logic. We then identify the isolation condition and the best isolating candidate for each circuit formed by partitioning. While application of the gating control signals, the inputs are classified in two categories – i) shared and ii) non-shared. Non-shared inputs are those inputs that are not shared by more than one logic block. Therefore, isolation can be performed on such inputs without affecting the functionality of other blocks. Shared inputs, on the other hand, are simultaneously shared by more than one logic block. Therefore, while isolating these inputs, attention should be taken so that the functionalities of the other blocks sharing that input is not affected. Since the identification of the optimal operand isolation candidates is critical for maximum power savings, we choose the isolation candidate we choose the isolation candidate is as follows:

First, we determine if flip-flops or latches are available for isolation and apply clock gating or control signal gating to them. If latches are not available, we perform isolation by I²-MUX in cases where a multiplexer precedes the datapath. If the switching probability of the multiplexer output is very high (e.g. it switches every clock cycle), we use an I²H-MUX to hold the state of the circuit. Otherwise, we use an I²F-MUX to force it to the minimum leakage state. In absence of multiplexers or latches, our algorithm locates tri-state buffers (e.g., buses have tri-state buffers) and applies control signal gating to the enable signals of the buffers to prevent unnecessary propagation of switching. In case none of these steering modules are available and the logic is connected to primary inputs, we apply FLS or FLH method to isolate them depending on their switching probability as in case of I²-MUX. After performing isolation, we estimate whether the timing constraint is violated after insertion of the isolation circuitry. If the timing constraint for the module is violated, we retain the original non-isolated circuit. Otherwise, if the target delay is met we apply isolation for the circuit. The next step involves optimization of datapath modules with bit-level supply gating for further power reduction. The timing constraint of the respective modules is verified again after applying bit-level operand isolation and the outcome determines whether this optimization is performed on the modules or the design obtained from the previous state is retained. The additional area and the power reduction of the optimized modules (by either operand isolation alone or bit-level isolation only or both) are computed in the final step.

We have applied our operand isolation synthesis flow (along with selection of optimal isolation candidate) to standard benchmark circuits and the results are shown in Table V. The first benchmark is a pipelined complex multiplier where the multiplexer is chosen as the isolation candidate since it chooses either the adder or the multiplier circuit at any instant of time for any valid computation. As observed in Table V, we obtain almost 40% power savings with negligible area overhead (0.95%) for the precomputation logic. The extra delay incurred due to the isolation circuitry is 0.2%. The second benchmark is an ALU core consisting of a datapath module and a logic module. In this case, we apply first level supply gating to isolate the primary inputs for the

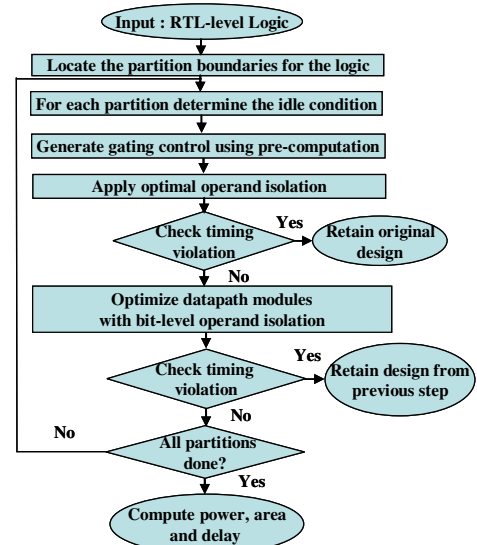


Fig 9: The overall synthesis methodology

Table V. Results of application of our synthesis flow

Design	OP Iso Scheme	Power		Area		Delay	
		[uW]	%red	[um ²]	%inc	[ns]	%inc
Complex Mult.	Mux	980.1	39.1	63999	0.95	42.8	0.2
ALU	FLS	360.1	50	7963	13	9.3	0.1

datapath module when it is not performing useful computations. It can be seen that we obtain around 50% savings in power for this benchmark. The area and the delay increase by insertion of this isolation circuitry are again minimal and around 13% and 0.1%, respectively.

7. CONCLUSION

We have presented novel operand isolation circuits that provide more power savings in datapath with significantly lower design overhead compared to the existing isolation schemes. We have also presented bit-level operand isolation for datapath modules to reduce power consumption while allowing them to perform useful computation for downstream circuits. We have developed an integrated synthesis methodology to automate the application of the proposed operand isolation techniques at the RT-level.

REFERENCES

- [1] H. Kapadia, et. al, Reducing switching activity on datapath buses with control-signal gating, JSSC, Volume: 34, Issue:3, 1999, pp. 405 – 414.
- [2] M. Munch, et. al, Automating RT-level operand isolation to minimize power consumption in Datapaths, DATE, 2000, pp. 624 – 631.
- [3] K. Roy, et. al, Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits, Proceedings of the IEEE, Vol. 91, 2003, pp. 305-327.
- [4] S. Bhunia, et. al, A Novel Low-Power Scan Design Technique Using Supply Gating, ICCD, 2004, pp. 60-65.
- [5] S. Bhunia, et. al, First Level Hold: a novel low-overhead delay fault testing technique, DFT, 2004, pp. 314-315.
- [6] Leda Design Inc., <http://www.leda-design.com>
- [7] M.C. Johnson, et. al, Models and algorithms for bounds on leakage in CMOS circuits, IEEE TCAD, Vol. 18, 1999, pp. 714-725.
- [8] A. Correale, Overview of the Power Minimization Techniques Employed in the IBM PowerPC 4xx Embedded Controllers, ISLPED, 1995, pp. 75-80.
- [9] V. Tiwari, et. al, “Guarded Evaluation: Pushing Power Management to Logic Synthesis/Design”, IEEE TCAD, 17(10), 1999, pp. 1051-1060.
- [10] M. Alidina, et. al, “Precomputation-based sequential logic optimization for low power,” ICCD, Nov. 1994, pp. 74-81.