Post Silicon Power/Performance Optimization in the Presence of Process Variations Using Individual Well Adaptive Body Biasing (IWABB)

Justin Gregg, *Student member, IEEE,* and Tom W. Chen, *Senior member, IEEE*‡

***Abstract*— The economics of continued scaling of silicon process technologies beyond the 90nm node will face significant challenges due to variability. The increasing relative magnitude of within die process variations will cause power-frequency distributions to widen, thus reducing manufacturing yields. Mitigating the effects of these process variations can be done by using the proposed IWABB scheme of locally-generated body biases. IWABB allows for highly localized circuit optimizations with very little overhead in silicon area and routing resources. We present two algorithms to find near-optimal configurations of these biases which can be applied as post-silicon tuning. The proposed IWABB scheme can improve an initial yield of 12% to 73%.**

I. Introduction

PROCESS variations are a growing problem for leading edge semiconductor manufacturers. Although the absolute magnitude of variations in the critical dimension (CD) is shrinking, its rate hasn't kept pace with CD scaling. This means the relative magnitude of variations in the CD is increasing. Most sources expect variations in transistor gate lengths (L_{eff}) to be 10%–20% of the CD by the 65nm technology node [1], [2]. Variations in device threshold voltages due to random discrete dopant placement and other phenomenon are also increasing.

Process variations can be divided into lot-to-lot (L2L), wafer-to-wafer (W2W), die-to-die (D2D) and within die (WID) variations. In general, process variations cause the operating power and maximum frequency of manufactured dies to vary [1], [3], [4]. The dies produced by a manufacturing process have a distribution in power and frequency. The increasing relative magnitude of process variations causes a widening of these distributions. Some manufacturers use product binning in an effort to boost yields in light of these distributions. However, this is only side-stepping the problem of wide operating parameter distributions caused by process variations. The effects of process variations need to be mitigated in order to tighten the operating parameter distributions.

Adjusting the body bias applied to FETs on a die is a very effective way of modifying its operating parameters. The body bias can be optimized during post-fabrication test or dynamically adjusted during runtime. Such adaptive body biasing (ABB) methods have been employed in circuits to reduce power consumption [5], increase performance [6], [7], and improve manufacturing yields [8], [9].

‡Justin Gregg and Tom Chen are with the Department of Electrical and Computer Engineering, Colorado State University.

The same concept can be applied to supply voltages. Adaptive supply voltage scaling (ASV) schemes are an effective way to perform power-performance tradeoffs. The use of ASV schemes has gained more momentum recently [10], [11]. It has been shown that ASV can be used during post-fabrication testing to improve product binning yields with comparable effectiveness as ABB schemes [12]. ASV only requires changing the fixed supply voltage to an adjustable one, and the addition of optimization time during post-fabrication testing. Combing ASV and ABB can also be very effective [13] at increasing yields.

Most of the existing ABB methods a single body bias for NFETs (V_{b_n}) and PFETs (V_{b_p}), and a uniform supply voltage (V_{dd}) across an entire die. In this way they are able to address L2L, W2W, and D2D variations. However, WID variations are becoming more significant than the others, making such schemes less effective [1].

To address WID variations, optimal body biases must be applied on a smaller scale within each die. The WID-ABB scheme presented in [14] applies optimal body biases at a sub-circuit level. It requires a replica critical path, phase detector, counter, R-2R ladder D/A converter and op-amp for each sub-circuit controlled by the scheme. All this additional circuitry makes for a large increase in silicon area compared to a chip without this scheme. Also, the spatial granularity of WID variations addressed by this scheme is limited by the size of this circuitry; e.g., variations that occur within the true critical path but not in the replica critical path cannot be corrected. The scheme's effectiveness could also be influenced by the location and orientation of the sub-circuit and its WID-ABB circuitry.

We present a locally-generated body biasing scheme that requires much less overhead than the WID-ABB scheme presented in [14]. Different biases can be applied on a well-by-well basis to reduce the impact of WID variations. Two implementation options are presented to produce local well biases: one using floating wells, and the other using a small voltage divider to locally generate well bias voltages. Our simulation results show that the most effective individual well adaptive body biasing (IWABB) scheme is capable of tightening frequency and power distributions in two test circuits, thus, increasing the binning yield from 12% to 73%.

Even though the 2003 International Technology Roadmap for Semiconductors projects the use of ultra-thin body, fully depleted silicon-on-insulator (SOI) processes by 2008 [15], the IWABB scheme is not suitable for such processes due to the

lack of body contacts. It is well known, however, that such SOI processes involve additional manufacturing complexity which can adversely effect manufacturing yields [15]. Interim solutions, such as IWABB, can be used to extend the useful lifetime of bulk CMOS processes by mitigating the effects of process variations until these problems with SOI processes are resolved.

Section II describes the two different IWABB circuits for producing locally-generated body biases. The sizing of this IWABB circuitry is crucial to its effectiveness in mitigating WID variations. We present methods for determining proper sizing for the IWABB circuitry in Section III. By adding either circuit implementation to each well, near-optimal body biases can be applied to individual wells. Optimization of these biases is done during post-manufacturing test using an external search algorithm. Two such algorithms are presented in Section IV. Two test circuits and a simulated manufacturing process used to test IWABB are shown in Section V. Yield improvement results for the two implementations and two algorithms from the simulated manufacturing process are shown in Section VI. Conclusions on the effectiveness of the IWABB scheme and the two algorithms is in Section VII.

II. Circuitry

High performance digital circuits usually employ standard dynamic and/or static logic blocks as well as memory and analog blocks. Normally the well voltages for the logic blocks are connected directly to power supply voltages—nwells to V_{dd} and the pwell substrate to V_{ss}. With ABB schemes, FET bodies are instead attached to a separate external or on-chip power supply distributed via a grid. That is, nwells are tied to a separate V_{b_p} grid and the p-type substrate is tied to a V_{b_n} grid. Since very little current is needed in these supply grids, they can be constructed of minimum sized wires. Nonetheless, adding these grids can be costly in routing resources. Another problem is that the functionality of the analog and memory blocks (e.g. SRAM cells) are sensitive to substrate biasing. Therefore, using V_{b_n} biasing would require isolating such elements in separate pwells. This requires a triple well process. Due to the expense involved with a triple well process, we only considered V_{b_n} biasing for comparison to other ABB schemes.

The simple circuits shown in Figs. 1–2 allow each nwell bias voltage to be a chip-wide bias (e.g. V_{dd}, dynamic V_{b_p}) or a locally-generated bias. There are two methods to generate this local bias. The floating well method (Fig. 1) uses only a pullup PFET to provide the choice between a chip-wide distributed bias and a floating nwell. The voltage divider controlled well method (Fig. 2) adds the option of applying a predefined locally-generated voltage to each well. The choice between these methods must be done at design time. These two methods are detailed in the following sections.

A. Floating well method

The floating well method is the simpler IWABB implementation. A single control signal (*pull_cntl*) controls the pullup PFET. When the pullup is conducting, the well is tied to the chip-wide distributed bias V_{b_p}. This voltage can be the same as

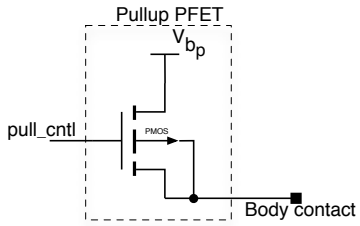


Fig. 1. Well voltage control structure for floating well IWABB method.

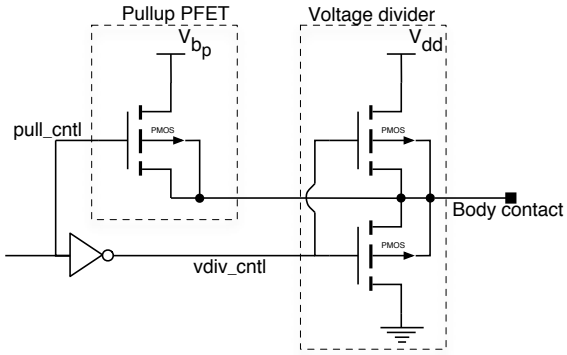


Fig. 2. Well voltage control structure for voltage divider IWABB method.

V_{dd}, or a separate static or scalable voltage. When the pullup is nonconducting, the well floats. Floating a well causes the well voltage to be determined by the capacitive coupling between all the PFETs in the nwell. The actual voltage is dependent of the switching conditions and outputs of the PFETs in the well. This also allows for significant voltage bounces at the body of PFETs in the well. Accurate simulation of this method (and the voltage divider method) requires modeling the well parasitics to account for the effects of these body bounces.

1) Well parasitics: Well parasitic resistance (R) and capacitance (C) are usually ignored until final layout verifications. However, these properties can play an important role in circuit operation. Most of the time it is assumed during circuit design that PFET bodies are tied directly to an ideal voltage source. This makes for very consistent transistor operation since the body voltage does not bounce when the transistor or one of its neighbors switches. However, the actual body voltage contact is not ideal and is located a finite distance from the transistor. The connection to the voltage supply is shielded by the parasitic R and C of the nwell. This shielding allows transistor body voltages to bounce significantly when a transition occurs. When a well is allowed to float, its voltage is heavily dependent on these parasitics. To exemplify this, a simple inverter chain (Fig. 3) is simulated through two transitions in a commercial 90nm CMOS process. The well parasitics are modeled with a distributed RC equivalent to a well contact located half the maximum allowed distance from the transistor bodies. Fig. 4 shows the voltage at the PFET bodies ($V_{bp1,2}$) as the transitions occur. The well contact is tied to an ideal 0.8V supply which is approximately equivalent to the voltage on a floating well. To model an nwell with several gates switching nearly simultaneously, both PFETs are 1μm

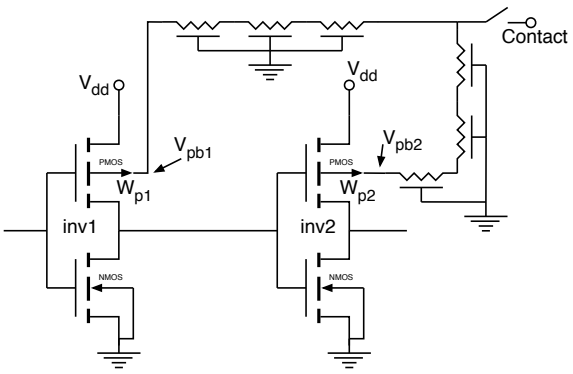


Fig. 3. Inverter chain test circuit to investigate body voltage bounce when nwell parasitic resistance and capacitance are included.

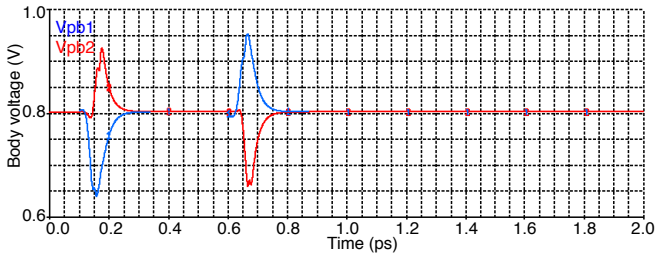


Fig. 4. Body voltage in the inverter chain shown in Fig. 3 with the nwell biased to 0.8V by an ideal source, $W_{p1} = W_{p2} = 1\mu$m and the NFETs appropriately sized to balance the inverters.

wide.

There is about 150mV bounce on V_{b_p} as each gate switches. The recovery time from such bounces is simply the time it takes for current from the well contact to charge the parasitic well capacitance through the parasitic well resistance. If there is another transition before the body voltage is fully recovered, body effect causes a shift in threshold voltage (V_{th}) of the transistor that tends to favor a change in the output. By taking advantage of the self biasing nature of the nwell caused by the capacitive coupling between transistors, the body bias can be dynamically adjusted to always favor the next transition.

This effect is exactly what is referred to in silicon-on-insulator (SOI) technology as floating body transient V_{bs} effects [16]. In general, these effects account for a significant part of the speed advantages SOI processes have over bulk CMOS [17]. Floating an nwell in a bulk process is slightly different from SOI in that the well voltage is controlled by the current and previous states of *all* the transistors in the same well instead of just a single transistor. This fact actually gives added control over the body bias since the probabilistic ratio of up switching to down switching transistors can be adjusted during design. Even without this well switching analysis, the performance advantage of floating wells exists.

B. Voltage divider method

By adding a voltage divider to the simple pullup circuit (Fig. 2, the choice of a locally-generated bias is added. Now the well bias is controlled by the *pull_cntl* and *vdiv_cntl* signal pair, where *pull_cntl* = *vdiv_cntl*′. The signals control the pullup PFET and the voltage divider. As before, the pullup PFET connects the well to a chip-wide distributed bias which can be a scalable or fixed V_{dd} or a scalable V_{b_p} from an on-chip or external source. The voltage divider supplies a predefined, locally-generated bias determined by the relative sizing of its PFETs. The sizing can be chosen to be an optimal bias based on an energy-delay-product (EDP) analysis detailed in Section III-B.

TABLE I
IWABB BIASING MODES

VDIV+NWB+VDD	Voltage divider controlled nwells with adjustable V_{b_p} and V_{dd} biasing
VDIV+VDD	Voltage divider controlled nwells with adjustable V_{dd} biasing
VDIV+NWB	Voltage divider controlled nwells with adjustable V_{b_p} biasing
VDIV	Voltage divider controlled nwells
NWB+VDD	Adjustable V_{b_p} and V_{dd} biasing
DWB	Adjustable V_{b_n} and V_{b_p} biasing, for comparison to other research
VDD	Adjustable V_{dd} biasing, for comparison to other research

Combining a scalable supply voltage with this simple circuit makes for a versatile biasing system. We have defined several different IWABB biasing modes shown in Table I. Modes with labeled with VDIV refer to using either of the locally-generated well biasing methods (voltage divider or fully floating). The last three IWABB modes do not use the previously detailed IWABB circuitry and are included for comparison to ABB and ASV schemes. Each different mode requires a different amount of overhead. The first overhead, discussed previously, is the additional silicon area needed for the voltage divider and pullup PFETs in each IWABB controlled nwell (Fig. 2). Since this circuit encompasses both well control methods and the total added area is small, it can be added regardless of the method used. This area overhead occurs for all modes with VDIV. Next, each adjustable bias requires an on-chip or external power supply. An additional power supply grid is also need to distribute V_{b_p} in modes using NWB. There is no addition grid routing overhead for V_{dd} since its grid is already present. The final overhead for IWABB is the additional post-manufacturing test time needed to find near-optimal biasing configurations. Modes with more “knobs" require more test time. This final overhead can be the most significant factor in total IWABB cost and is explained further in Section IV.

III. Circuit Sizing

Correctly sizing the FETs in the IWABB circuitry is crucial since they determine the effectiveness of IWABB. Sizing of these FETs is dependent on the overall size of the PFETs in the nwell, their switching polarity and activity factor, as well as the process technology used. Overall, the additional circuitry requires less than a 0.01% increase in silicon area, including the added inverters and scan-latches. This small area overhead allows the IWABB circuitry to be added to every nwell on a chip, giving IWABB the maximum control over

process variations. However, if desired, the IWABB circuitry could be added to just the nwells on major critical paths. Doing so reduces the silicon area and routing overhead as well as reduces the post-manufacturing test time needed to find near-optimal biasing configurations. Determining the correct size for each FET in the IWABB circuit is described in the following sections.

A. Chip-wide well voltage

The sizing of the pullup transistor is crucial since it supplies the nwell bias from the chip-wide source. It needs to be wide enough to effectively tie the well voltage to the applied bias even when transistors in the well switch, causing a bounce on the well voltage. However, each transistor is shielded by the well parasitics between the location of the pullup and the location of the transistor. Using the circuit in Fig. 3 again, Fig. 5 compares the bounce in V_{b_p} at inv1 as the width of PFETs in the inverters vary with three different well biasing options: connected to an ideal 0.8V source; connected to a 1.0μ wide, minimum length pullup PFET; floating. The floating well has a steady-state voltage of approximately 0.8V. As the PFET widths are varied the NFETs in the inverters are scaled to keep each gate balanced, the parasitic well C is increased, and parasitic well R is decreased, proportionally. In Fig. 5a, the nwell is connected to an ideal 0.8V source. The V_{b_p} bounce is only strapped to within several hundred mV. Fig. 5b shows a 1.0μm wide, minimum length pullup transistor is able to accomplish nearly the same strapping. Based on Fig. 5, a 1.0μm minimum width pullup FET can act as switch between a chip-wide bias and a locally-generated (self-regulated or voltage divider supplied) well bias.

Finally, Fig. 5c shows the fully floating well exhibits approximately a 30% larger bounce compared to the ideally connected well in the worst cases. While this added bounce may seem rather significant (and frightening), it can be useful in terms of added performance. Fig. 6 shows the inverter chain delay benefits from the additional bounce. The floating nwell configuration yields about a 2% decrease in delay due to floating body transient effects over a similarly forward-biased nwell configuration. These effects can only be realized when the body of FETs bounce due to switching events.

Combining the results from Fig. 5 and 6, the smallest benefit to delay comes when the ratio of up to down switch PFETs in the well is close to 1. The floating body transient effects are greatest when the bounce in V_{b_p} is greatest, which occurs when all the PFETs in the well switch at the same time. Having all the PFETs in a well switch at the same time is not feasible, though. For random logic averaged over time, the switching ratio will be near one. However, grouping critical path logic in such a way to maximize coincident switching during crucial events may be a way to close timing on trouble paths. One could also group critical path PFETs with large dummy PFETs that switch at the same time in order to realize the transient speed effects. The designer would need to consider how much V_{b_p} bounce is acceptable for the design, though.

Another consideration in the sizing of the pullup FET is the leakage through the pullup when it is turned off to allow

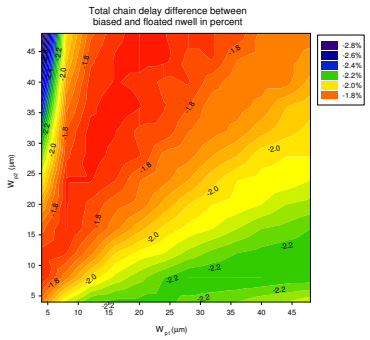


Fig. 6. Change in total inverter chain (Fig. 3) delay from the ideal 0.8 V source biased nwell to floated nwell ($(d_{floated} - d_{biased})/\text{mean}(d_{floated})$) for varying W_{p1} and W_{p2}.

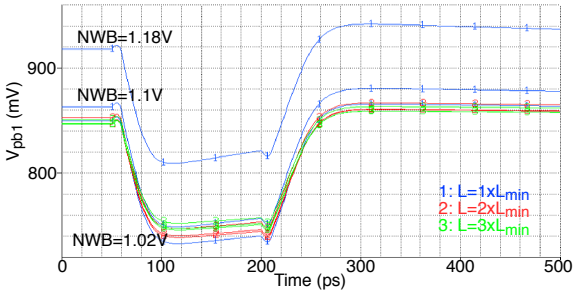
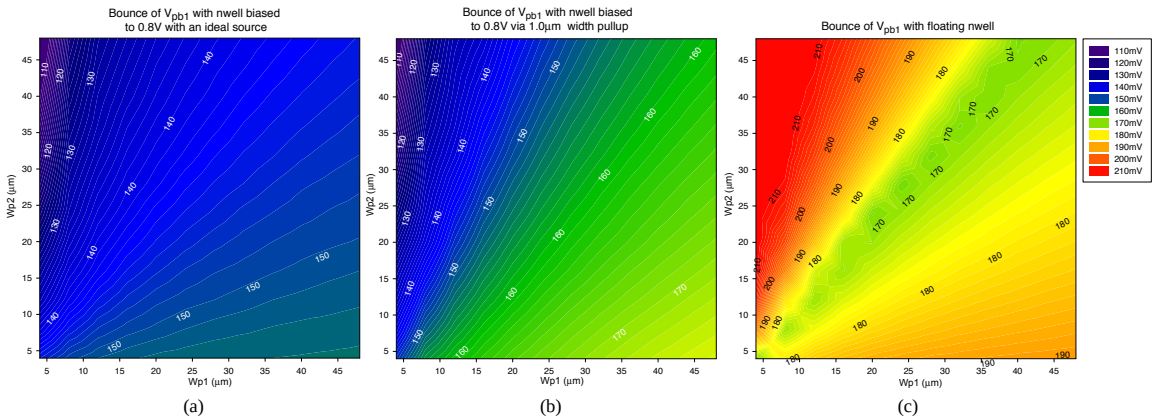


Fig. 7. Floating PFET body voltage at inv1 of Fig. 3 with a 1.0μm wide pullup PFET. The length of the pullup varies from one to three times L_{min}, while the nwell bias voltage varies from 1.08V to 1.18V.

the well to float or be controlled by the voltage divider. This leakage current is primarily determined by the pullup transistor length (L_{eff}), the pullup transistor width (W), the nwell bias voltage applied to the source of the pullup (V_s), and the well voltage at the body contact. If the pullup transistor is too short, the leakage through it can actually charge the floating or voltage divider controlled well. The leakage current can be substantial since increasing V_s increases V_{sd} and V_{sg} making the PFET more conductive. As the nwell bias voltage increases, current leaking through the off pullup will increase the well voltage. This increase causes a reduction in the static power consumption of the transistors in the well. However, the increase in pullup leakage power can be more than the static power decrease in the well, causing a net increase in power consumption.

The solution to this leakage is to reduce the pullup transistor's I_{off} by increasing its L_{eff}. Fig. 7 shows the floating PFET body voltage at inv1 (V_{pb1}) of Fig. 3 with a 1.0μm wide pullup PFET. The length of the pullup varies from one to three times the minimum (L_{min}) and the nwell bias voltage varies from 1.08V to 1.18V. When the pullup is L_{min} and the nwell bias is 1.18V, the floating body voltage is almost 70mV above the normal 850mV. This takes the well voltage out of the EDP-optimal biasing range discussed later. Increasing the pullup length reduces this leakage-induced V_{b_p} increase and only marginally reduces the pullup's voltage strapping effectiveness. To compensate, the width of the pullup can be increased, but this was determined to be unnecessary.



(a) (b) (c)

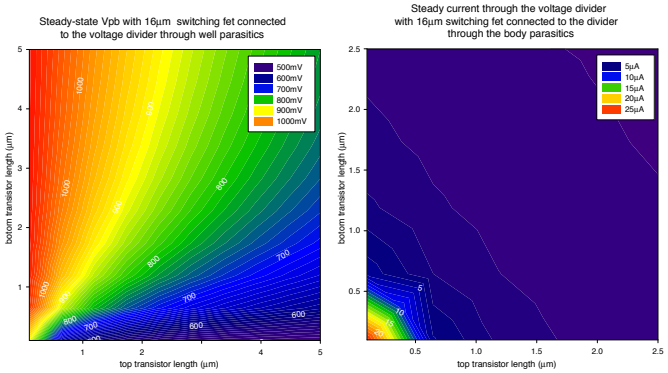
Fig. 5. Comparison of body voltage bounce V_{pb1} in the inverter chain shown in Fig. 3 for varying W_{p1} and W_{p2} with the nwell (a) biased to 0.8V with an ideal source (b) biased to 0.8V with a 1.0μm wide, minimum length pullup transistor (c) floating.

B. Locally-generated well voltage

When using the voltage divider controlled well method, one of the most important parameters in IWABB is the floating or voltage-divider-controlled well voltage. The optimal well voltage is process and circuit dependent, and can be determined by examining the total power and delay of the circuit with varying well voltages. Fig. 9b shows the circuit delay, P_{op}, power-delay product (PDP) and (EDP) as a function of V_{b_p} for the circuit in Fig. 3. The minimum in the EDP curve at about 850mV represents the optimal nwell biasing point for this circuit and process [18]. This biasing point gives the optimal trade-off between performance and power. Biasing at any other point would diminish the effectiveness of IWABB. Biasing at this voltage can be generalized to be near-optimal for other circuits. Circuits with process variations causing generally longer (shorter) FETs may have lower (higher) optimal bias voltages.

For the floating well method, the floating well voltage is determined primarily by the polarity of the transistors in the well. If a majority of the PFETs in a well are conducting (weighted by their widths), the well will tend to float at higher voltage than a well with more non-conducting PFET width. In order to achieve optimal floating well voltage, one could group PFETs into nwells such that the most common well polarity gives a voltage near the optimal point on the EDP curve. This can prove to be a difficult task since it depends on the number of PFETs in each nwell, the most common output of each PFET, each PFET's location in each nwell, the separation distance between each PFET, the width of each PFET. However, even a grouping based on proximity (the normal method) can have a significant effect on circuit operating parameters.

For the voltage-divider controlled well method, the well voltage is determined by the ratio of the lengths of the transistors in the voltage divider. This makes it possible to tune the well voltage to the optimal point without having to carefully group the PFETs based on switching polarity. The total length of the transistors in the voltage divider determines the static short-circuit current when the voltage divider is on. This power would need to be budgeted for the chip. This circuit



(a) (b)

Fig. 8. Voltage divider sizing tests based on (a) steady-state V_{b_p} and (b) voltage-divider static current with 16μm switching inverter in the nwell for varying voltage divider component lengths.

structure is intended for use in high performance microprocessor designs. The static power dissipated by each conducting voltage divider represents a power-performance trade off. The additional power makes using this circuit extremely low power designs undesirable. In general, the voltage divider power dissipation can be on the order of 2μW, and therefore not be a major concern in overall chip power. For a large chip with 10^7 PFETs divided into 10^4 nwells this amounts to about 0.02W. Given the power budget of modern microprocessors beyond 100W, this is only 0.02% of the overall power budget. Of course this is an absolute worst case with all voltage dividers on. In a real implementation, less than half of the voltage dividers would be on at any time.

The width of the voltage divider FETs can be at the minimum as long as the total length is adjusted to allow significant current to flow in the divider. This static current needs to remain above a certain level in order to charge the well capacitance during a switching event. To determine the correct lengths of the voltage divider FETs, Fig. 8 shows the well voltage and voltage-divider current with respect to the two lengths. Lengths that achieve a well voltage of about 850mV and a divider current of 2μA were chosen.

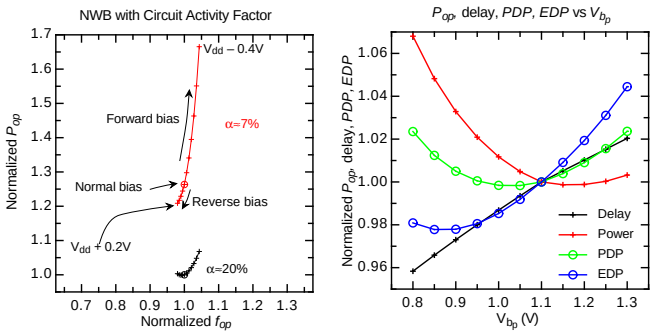


Fig. 9. (a) P_{op} and f_{op} versus V_{b_p} with activity factors of about 7% and 20%. (b) Circuit delay, P_{op}, PDP and EDP with varying V_{b_p}.

C. Well activity factor

The total power change in the well caused by a change in the nwell bias voltage is determined by several factors including the overall size of the well, the voltage at the body of the transistors in the well, and the activity factor in the well (α). The final factor in this is very important since it can govern whether individual well biasing is beneficial. Since body biasing mainly changes the static power of FETs, if α of a well is too low (there are too few switching FETs in the well for a given time period) then forward biasing the nwell will cause a large increase in the static power. In such a case, the static power dominates the total power, making the increase due to forward biasing very pronounced. However, since I_{off} is exponentially decaying with respect to a reverse bias in V_{b_p}, increasing V_{b_p} causes a progressively smaller decrease in the static power. These trends can be seen in the low activity curve in Fig. 9a. With a larger α, the total power is increasingly dominated by the dynamic power component. Therefore, forward biasing a well only marginally increases the total power. By observing the curves in Fig. 9a, the slopes represent this concept in terms of power-performance trade-off. The $\alpha = 7\%$ curve has a significantly steeper slope than the $\alpha = 20\%$ curve. This shows that forward and reverse biasing have a better power-performance trade-off with higher activity factors. The activity factor for a well could be controlled by intelligently grouping FETs, but again, this would be difficult. The α of a well just needs to be considered for small wells or wells where many of the FETs are inactive. Neither of these cases should exist in most critical paths.

IV. Algorithms

After the IWABB circuitry is added to a design, the chip is manufactured. Then during post-fabrication test, each die can be configured with its own optimal biasing configuration. The number of possible biasing configurations these IWABB modes provide grows exponentially with the number of nwells. It is also a function of the biasing resolution used for the adjustable voltage supplies. With more possible configurations, IWABB can be more effective at mitigating small WID process variations. However, more possible configurations makes for a larger space to search for the optimal biasing configuration. Even with only a small number of nwells, the search space grows large enough to make an exhaustive search intractable. For this method to scale to large circuits, intelligent, non-exhaustive search algorithms must be used to find near-optimal biasing configurations.

Such an algorithm is run on each die during post-manufacturing test, and its output is the near-optimal biasing configuration for that particular die. This configuration is stored and then shifted into a scan chain during startup from an external EEPROM or support processor. Such startup parameter controller are becoming more common in high performance microprocessors.

Design of search algorithms tailored to this relatively predictable problem space is crucial to making the IWABB scheme work. The use of single-objective search algorithms is one way. The objectives of optimization is the overall circuit operating power (P_{op}) and maximum frequency (f_{op}). This two dimensional objective space can be reduced to a single objective by providing target objective points P_{max} and f_{min}. These target points are picked a priori and can be thought of as target yield or product binning points. The search algorithms' objective is to find for each die a biasing configuration such that $P_{op} \leq P_{max}$ and $f_{op} \geq f_{min}$. To accomplish this, the algorithms attempt to minimize the distance in the two dimensional objective space between the points $\{f_{op}, P_{op}\}$ and $\{f_{min}, P_{max}\}$ until such a condition is met.

One algorithm tested is a hybrid single-objective evolutionary algorithm (soIWABB). The second is a gradient-local random hill climber (gIWABB). Both algorithms encode the parameter space with four variables. One is a string of n binary values representing the *vdiv_cntl* signal for each nwell. The others are real valued numbers representing the different possible bias voltages V_{b_n}, V_{b_p}, and V_{dd}. The necessary biasing resolution for the real valued parameters was found to be 30mV. Larger biasing resolutions make the IWABB system significantly less effective. The two search algorithms are detailed in the following sections.

A. soIWABB

The soIWABB algorithm uses a single-objective evolutionary algorithm hybridized with heuristic information on the predictable search space this problem presents. The space is predictable in that increasing the forward biasing of some PFETs on the critical path of a circuit will undoubtedly increase both f_{op} and P_{op}. Similar generalizations can be made about other small biasing changes. Using this information in the search algorithm adds an aspect of parameter-space hill climbing since the algorithm can know which direction to change a parameter to improve the objective.

Pseudo code for the soIWABB algorithm is shown in Table II. Lines 1–4 are for initialization. Lines 5–6 are test evaluations used to generate gradient and sensitivity information to supplement the heuristics discussed previously. This information can determine which bias is most effective at reducing P_{op} or increaseing f_{op}. Lines 7–16 involve the initial population generation. This is where most of the heuristics are employed. soIWABB attempts to generate a population of configurations that is very close the target objective points by

TABLE II
SOIWABB ALGORITHM PSEUDO CODE

```
1   for each die
      initialize individual best ind
      initialize population
      popsize = 0
5     evaluate connected and half vdiv
      evaluate bias tests
      while popsize < maxpopsize
        create new individual ind
        Randomly choose vdiv or bias
10      If bias chosen
          estimate bias needed to meet req.
        If vdiv chosen
          estimate vdivs needed to meet req.
          randomly generate needed vdivs
15      evaluate ind
        add ind to population
      while gen < maxgen
        select two parents from pop
        crossover parents to produce child
20      mutate child
        evaluate child
        reinsert child and parents
        select best maxpopsize inds for new pop
      record best ind
      next die
```

TABLE III
GIWABB ALGORITHM PSEUDO CODE

```
1   for each die
      initialize individuals ind, best ind
      evaluate ind
      for each nwell bit in ind
5       vdiv nwell bit
        evaluate ind
        save gradient info
        connect nwell bit
      if f_op < f_min
10      f_est = f_op
        while f_est < f_min
          add best vdiv
          update f_est
        evaluate ind
15    while step < maxstep
        if f_op < f_min
          randomly choose:
            increase forward bias on NFETs
            increase forward bias on PFETs
20          add best vdiv
        else if P_op > P_max
          randomly choose:
            increase reverse bias on NFETs
            increase reverse bias on PFETs
25          connect worst vdiv
        evaluate ind
        next step
      record best ind
      next die
```

using the gradient and sensitivity information. Lines 17–23 then fine tune configurations in this population with a simple evolutionary algorithm. The **select** function uses a single tournament selector. The **crossover** function uses a weighted uniform crossover for the binary nwell bits, and a intermediate crossover for the real valued parameters. The **mutate** function makes small random changes to the biasing configuration at a rate of about $1/popsize$. The **evaluate** function simulates the configuration using SPICE and returns an objective value based on P_{op}, f_{op}, P_{max} and f_{min}.

B. gIWABB

The gIWABB algorithm makes more use of gradient and sensitivity information by calculating it for each nwell independently. Since the nwells are independent, their effect on the objective parameters is nearly independent. Therefore, the effect of using voltage divider controlled nwells follow vector addition in the two dimensional objective space of $\{f_{op}, P_{op}\}$. This niceness of the parameter-objective space mapping combined with gradient information from each nwell makes finding a near-optimal binary nwell configuration very easy. The gIWABB algorithm then uses a local random walk to tune the real valued components of the biasing configuration. As in soIWABB, the use of basic heuristics allows the random walk to exhibit hill climbing tendencies.

Table III shows the pseudo code for gIWABB. Lines 1–3 are for initialization. Gradient information for each nwell is calculated in lines 4–8. The algorithm moves to the near-optimal binary nwell configuration in lines 9–14. Lines 15–28 are the random walk using heuristic hill climbing. gIWABB uses the same **evaluate** function as soIWABB.

C. Algorithm complexity and test time

In a real implementation, the **evaluate** functions would be done by applying a vector set to the die in order to get a determination of f_{op} and P_{op}. The tester would determine the next bias configuration by running the one of the algorithms

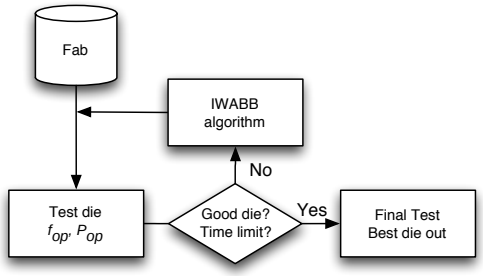


Fig. 10. IWABB post-manufacturing test process.

described. This would be repeated until a predetermined time limit or a good configuration $\{f_{op} \geq f_{min}, P_{op} \leq P_{max}\}$ was reached. This process is depicted in Fig. 10. In order to reduce the amount of test time this requires, several procedures can be modified. First, as discussed previously, the number of wells to which the IWABB circuitry is added can be limited to only those on major critical paths. Doing so would reduce the search space exponentially. Second, the vector set in the **evaluate** function could be a reduced set which provides only an estimated f_{op} and P_{op}. After a near-optimal bias configuration is chosen by the algorithm, a full vector set would be run to verify the die's operating parameters. This would reduce the amount of time each **evaluate** function requires. Finally, the maximum allowable test time for any die can be limited.

Even without all of these test time limiting procedures, the impact to the test time is small. Both algorithms described are able to consistently converge to a near-optimal solution in a number of iterations on the order of the number of nwells in the circuit. Note that it is well accepted that evolutionary algorithms like the one in soIWABB scale nearly linearly with problem size[19]. It can be easily shown that the gIWABB algorithm scales linearly at best. Therefore, a design with 10^4

nwells, with a test suite of 10^6 vectors running at 1GHz would require approximately 10 additional seconds on the test bench. If a reduced set of vectors were used, and a subset of wells optimized, this could easily be reduced to well under 1 second.

Finally, tuning of the parameters used in these algorithms could change their effectiveness and speed. E.g. the population size and mutation rate in soIWABB, and the weighting of the random choices in gIWABB. However, doing such testing with simulations has proven too time consuming. Experiments with tuning will be done when the algorithms can be run with real silicon.

V. Experimental Setup

Two test circuits from a microprocessor design were used in our experiments. Both are designed using a 90nm commercial nwell CMOS process. First, *test_cir1*, is a pure static logic data path. Second, *test_cir2*, contains an interface logic block between the integer execution unit and cache from a high-end 64-bit microprocessor. It consists of mostly dynamic logic. Nwells for each circuit are constructed based on groupings of functional blocks. Geographic proximity of nwells is based on the logical separation in a functional flow diagram.

For both circuits, the IWABB circuitry was added to each nwell. A distributed RC model was placed between each PFET body and the IWABB circuitry to model parasitic nwell capacitance. The size of the parasitic RC was based on the assumption that the PFETs were uniformly distributed across the nwell up to half the maximum allowed distance from the body contact on the IWABB circuitry.

A simulated manufacturing process was developed in order to produce the well accepted geographically correlated random variations in L_{eff}. The values of L_{eff} for all the dies forms a p-variate normal distribution centered on each FET's drawn length (L'), with a covariance Σ, where p is the number of FETs and Σ is based on geographic proximity.

$$L_{\mathit{eff}} = N_p(L', \Sigma) = L' + \sigma \cdot N_p(\vec{0}, \rho) \tag{1}$$

This can be reduced, as shown in (1), to a p-variate normal distribution centered on 0 with a correlation ρ based on proximity. The structure of ρ is such that variations are highly correlated for geographically close FETs in the same logic block, while variations of widely separated FETs are not correlated. Based on this distribution, 100 unique dies were produced for each test circuit using a σ value consistent with the available process information. Fig. 11 shows the initial operating parameter distributions for each circuit. The figure also shows the target objective or yield points chosen to model a product binning structure.

After the simulated manufacturing process, each algorithm was run in all the biasing modes for each yield point and for both locally-generated well biasing methods. The algorithms' output is the best configuration found for each die for the given mode and initial yield. Based on this output, a final yield can be calculated by counting the number of dies in the acceptable region defined by the yield point.

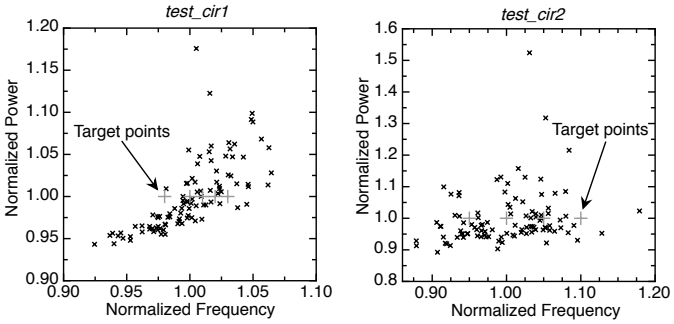


Fig. 11. Pre-IWABB die distributions and product binning target yield points.

VI. Experimental Results

In addition to the 100 dies produced for yield improvement analysis, four test dies were made that had simple patterns of variations. The patterned variations made it possible to immediately identify the optimal nwell binary configuration. These test dies were used to test each algorithm's ability to find this known optimal solution. Given a suitable yield point, both algorithms were able to find the optimal solution. This showed that both algorithms can find near-optimal solutions.

In order to compare the effectiveness of all the different modes, algorithms and methods, the metric of yield improvement was used. Yield improvement is the difference between the yield with no IWABB modifications and the yield after near-optimal IWABB biasing configurations are found by one of the algorithms. First, we summarize the comparison between the different locally-generated well biasing methods. Next, the yield improvement is summarized for each algorithm.

A. Locally-generated well biasing methods

Fig. 12 shows the yield improvement results for the floating and voltage divider controlled well biasing methods. For the sake of clarity, only the results from soIWABB in the VDIV mode for *test_cir1* are shown. The floating well method only produced a small improvement for all initial yields. However, the voltage divider controlled well method effected significant yield improvements across the higher initial yields. It was able to improve the 15% initial yield to 45%. The results from *test_cir2* were nearly indistinguishable between the two methods. Neither method was able to produce a significant improvement without additional adjustable biases on *test_cir2*.

It is clear from Fig. 12 that the floating well method is inferior to the voltage divider method. This is due to the complete lack of well bias optimizations for the floating well method. As discussed, the switching ratio of the PFETs in each well could have been intelligently chosen through grouping or adding dummy PFETs. This would have allowed for statistically choosing the floating well bias. Instead, the nwell grouping was left to only those presented by logical-block divisions and nwell sizing. This caused the normal floating nwell voltage to fall around 900mV. Being far above the EDP optimum bias, this bias caused the PFETs in floating wells to have increase leakage, without a significant improvement in performance. Therefore, the algorithms tended to not float

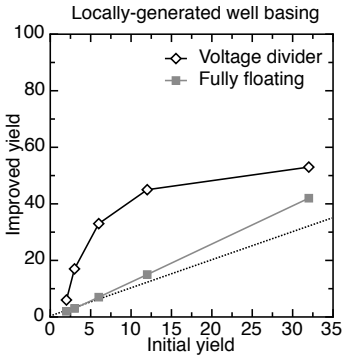


Fig. 12. Comparison of yield improvement from fully floating and voltage divider controlled nwell IWABB methods on *test_cir1*.

very many wells, thus never improving the yield. Based on this, only results from the voltage divider method are shown in the following analysis. The floating well method still hold merit, but the well optimizations needed during design time may prove to be too costly for an effective implementation.

B. soIWABB

Fig. 13 shows the *test_cir1* yield improvement results from the soIWABB algorithm. The VDIV+NWB+VDD and VDIV+VDD modes are basically even for all the yields, but VDIV+NWB+VDD is slightly better. NWB+VDD performed very well when the initial yield was high, improving the yield from 32% to 83%, though it performs worse than DWB in other initial yields. VDIV+NWB and VDIV both give a moderate improvement for the better initial yield points. The VDD mode isn't able to improve any die beyond the initial yield. In general, the performance of all the modes are similar on the lowest initial yield, but the more effective modes quickly improve with increasing initial yield.

The *test_cir2* yield improvement results from soIWABB are shown in Fig. 14. The mostly dynamic structures used in *test_cir2* mean most of the circuit is made of NFETs. This can make V_{b_p} biasing significantly less effective at changing the operating parameters of the circuit. The general ordering of the different modes is similar to the *test_cir1* results. Even though the yield improvements seem lower than those achieved on *test_cir1*, they are still quite significant. The VDIV+NWB+VDD mode is able to double the 46% initial yield, and is able to increase the 28% initial yield to 68%.

C. gIWABB

The *test_cir1* yield improvement results from gIWABB are shown in Fig. 15. The VDIV+NWB+VDD mode outperformed all other modes, improving the 12% initial yield to 73%. VDIV+VDD, NWB+VDD, DWB all performed about the same, but DWB performance flattened out in the best initial yield. VDIV+NWB and VDIV are about even in third place, with VDD performing the worst.

The gIWABB yield improvement results for *test_cir2* are shown in Fig. 16. Again the general ordering of the different modes is similar to the *test_cir1* results. While there is a significant yield improvement, the results are lower than those

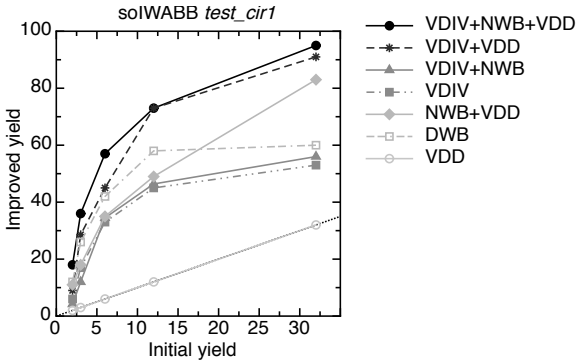


Fig. 13. soIWABB *test_cir1* yield improvement plot.

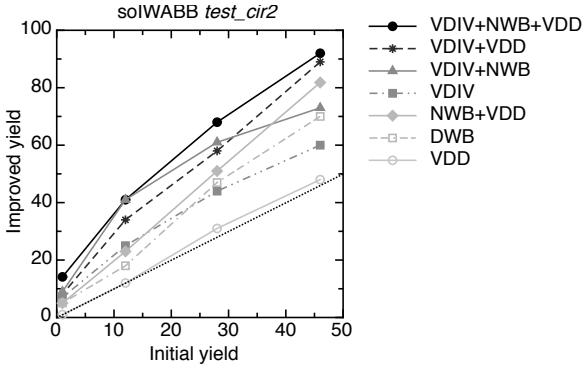


Fig. 14. soIWABB *test_cir2* yield improvement plot.

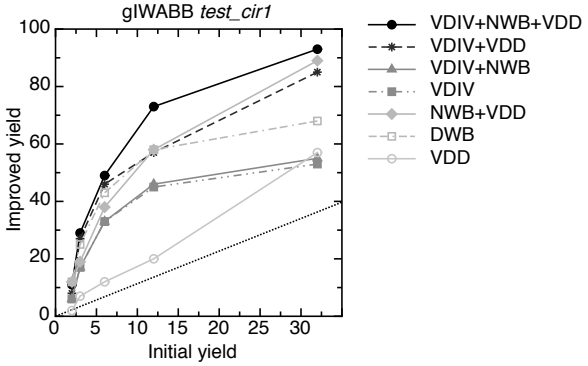


Fig. 15. gIWABB *test_cir1* yield improvement plot.

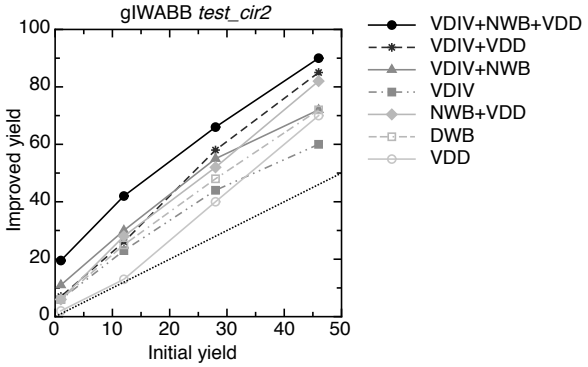


Fig. 16. gIWABB *test_cir2* yield improvement plot.

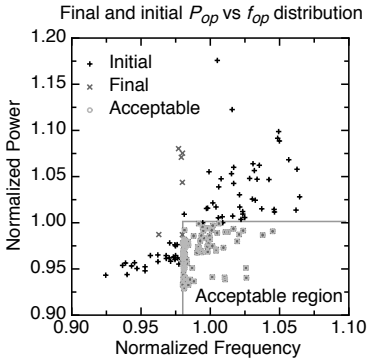


Fig. 17. Initial and final *test_cir1* die distributions.

achieved on *test_cir1*. The VDIV+NWB+VDD mode is able to nearly double the 46% initial yield, and is able to increase the 12% initial yield to 42%.

These yield improvements happen because the near-optimal biasing results for each die tend to cluster the distribution of operating parameters. I.e., the algorithms' outputs reduce the spread of the die distributions by countering the effects of D2D and WID process variations. Fig. 17 shows an example of the final objective space distribution of *test_cir1* dies from gIWABB using the VDIV+NWB+VDD mode. It shows IWABB's ability to reduce P_{op} on some dies and raise f_{op} on other dies. Plots of resulting distributions using the soIWABB and other modes or yields strongly parallel this one.

VII. Conclusions

By comparing IWABB's general ability to improve yield between the two circuits, one can see that IWABB is less effective on *test_cir2*. This can be explained by three factors. First, the inclusion of large driver FETs in some of the nwells makes forward biasing those wells cause a substantial increase in P_{op}. Other PFETs in the same nwells are in the critical path, so the performance of the circuit would greatly benefit from such a forward bias, yet the power-performance trade-off is unfavorable. This causes the algorithms to tend to do nothing to these nwells, which does not improve the yield. Second, *test_cir2* has over four times the number of PFETs, but only one additional nwell. This gives IWABB significantly less granularity in addressing WID variations. Finally, the dynamic structures in *test_cir2* generally have a lower stacking factor than the static structures in *test_cir1*. This makes the increased leakage current caused by forward biasing have a greater effect on P_{op}. This final problem is inherent in dynamic circuits, and makes the value of dynamic circuit structures in sub-90nm processes questionable as a candidate for use with ABB schemes.

Since the general ordering of the effectiveness of the IWABB modes is similar between the circuits, we will focus that discussion on *test_cir1*. To ease comparison of the different algorithms and modes, Table IV shows the yield improvement each algorithm produces averaged across all yield points on *test_cir1*. The best performing mode for each algorithm is shown in bold. It is clear that both algorithms produce

TABLE IV

Average yield improvement for each biasing mode.

Mode	$\overline{y_f - y_i}$ soIWABB	gIWABB
VDIV+NWB+VDD	**44.8%**	**45.0%**
VDIV+VDD	38.3%	33.6%
VDIV+NWB	17.8%	20.4%
VDIV	19.8%	19.8%
NWB+VDD	28.2%	32.2%
DWB	28.6%	30.2%
VDD	0%	8.6%

similar yield improvements for each mode. This means that the algorithms are able to find biasing configurations of similar quality regardless of initial yield or mode.

Table IV shows that the VDIV+NWB+VDD mode is the most effective at improving yield and VDIV+VDD is close behind. However, the cost associated with implementing these modes is quite different. Both methods require the silicon area overhead of the IWABB circuitry, the routing overhead of its control signals and scan chain, and an adjustable supply voltage. On top of that, VDIV+NWB+VDD requires a second adjustable power supply along with another distribution grid. The small increase in yield improvement offered by VDIV+NWB+VDD does not outweigh this additional cost of implementing it. For this reason, VDIV+VDD is the best IWABB mode regardless of the search algorithm used to implement it.

Choosing between the algorithms would depend heavily on the implementation in the post-manufacturing test suite, the number of wells IWABB is used in, and the amount of additional test time which is deemed acceptable. Further optimization of the algorithms may result in one scaling to larger problems significantly better than another. Also, use of a high level learning algorithm in addition to the search algorithm could dramatically improve the efficiency of both algorithms. Choosing these options presents a yeild-test cost trade-off.

Overall, the VDIV+VDD mode in IWABB is able to improve an initial yield of 12% to 73%. It is able to do this with less than a 0.01% increase in silicon area, a small increase in routed nets, and a limited increase in test time. As the relative magnitude of process variations continues to increase, to the detriment to manufacturing yields, schemes like IWABB will be invaluable for mitigating their effects and improving yield.

References

[1] K. A. Bowman, S. G. Duvall, and J. D. Meindl, "Impact of die-to-die and within-die parameter fluctuations on the maximum clock frequency distribution for gigascale integration," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 2, February 2002.

[2] A. Srivastava, R. Bai, D. Blaauw, and D. Sylvester, "Modeling and analysis of leakage power considering within-die process variations," *International Symposium on Low Power Electronic Desgin*, pp. 64–67, 2002.

[3] W. Riordan, R. Miller, J. Sherman, and J. Hicks, "Microprocessor reliability performance as a function of die location for a 0.25μ, five layer metal CMOS logic process," *1999 IEEE International Reliability Physics Symposium Proceedings*, pp. 1–11, March 1999.

[4] S. Nassif, "Delay variability: sources, impacts and trends," *2000 IEEE International Solid-state circuits conference, Digest of technical papers*, pp. 268–369, February 2000.

[5] H. Mizuno, K. Ishibashi, and T. Shimura, "An 18-μa standby current 1.8-v, 200-mhz microprocessor with self-substrate-biased data-retention mode," *IEEE Journal of Solid-State Circuits*, vol. 34, no. 11, November 1999.

[6] H. Wann, C. Hu, K. Noda, D. Sinitsky, F. Assaderaghi, and J. Bokor, "Channel doping engineering of mosfet with adaptable threshold voltage using body effect for low voltage and low power applications," *Proceedings of Technical Papers from 1995 International Symposium on VLSI Technology, Systems, and Applications*, pp. 159–163, May 1995.

[7] T. Kuroda, T. Fujita, S. Mita, T. Nagamatu, S. Yoshioka, F. Sano, M. Norishima, M. Murota, M. Kako, M. Kinugawa, M. Kakumu, and T. Sakurai, "A 0.9 V 150 MHz 10 mW 4 mm^2 2-D discrete cosine transform core processor with variable-threshold-voltage scheme," *Digest of Technical Papers from 1996 IEEE International Solid-State Circuits Conference*, pp. 166–167, 437, February 1996.

[8] M. Miyazaki, G. Ono, and K. Ishibashi, "A 1.2-gips/w microprocessor using speed-adaptive threshold-voltage cmos with forward bias," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 2, February 2002.

[9] M. Miyazaki, H. Mizuno, and K. Ishibashi, "A delay distribution squeezing scheme with speed-adaptive threshold-voltage CMOS (SA-Vt CMOS) for low voltage LSIs," *1998 International Symposium on Low Power Electronics and Design Proceedings*, pp. 48–53, August 1998.

[10] T. Burd, T. Pering, A. Stratakos, and R. Brodersen, "A dynamic voltage scaled microprocessor system," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 11, November 2000.

[11] S. M. Martin, K. Flautner, T. Mudge, and D. Blaauw, "Combined dynamic voltage scaling and adaptive body biasing for lower power microprocessors under dynamic workloads," *IEEE/ACM International Conference on Computer Aided Design*, pp. 721–725, 2002.

[12] T. Chen and S. Naffziger, "Comparison of adaptive body bias (ABB) and adaptive supply voltage (ASV) for improving delay and leakage under the presence of process variation," *To appear in: IEEE Transactions on VLSI Systems*, December 2003.

[13] J. Tschanz, S. Narendra, R. Nair, and V. De, "Effectiveness of adaptive supply voltage and body bias for reducing impact of parameter variations in low power and high performance microprocessors," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 5, pp. 826–829, May 2003.

[14] J. W. Tschanz, J. T. Kao, S. G. Narenda, R. Nair, D. A. Antoniadis, A. P. Chandrakasan, and V. De, "Adaptive body bias for reducing impacts of die-to-die and within-die parameter variations on microprocessor frequency and leakage," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 11, November 2002.

[15] *International Technology Roadmap for Semiconductors*. ITRS, 2003.

[16] Y. Zhang, D. K. Schroder, H. Shin, S. Hong, T. Wetteroth, and S. Wilson, "Abnormal transconductance and transient effects in partially depleted SOI MOSFETs," *Solid-State Electronics*, vol. 43, pp. 51–56, January 1999.

[17] G. G. Shahidi and F. Assaderaghi, "Soi technology and circuts," in *Design of High-Performance Microprocessor Circuits*, A. Chandrakasan, W. J. Bowhill, and F. Fox, Eds. IEEE Press, 2001, ch. 5.

[18] M. Horowitz, T. Indermaur, and R. Gonzalez, "Low-power digital design," *Digest of Technical Papers from 1994 IEEE Symposium on Low Power Electronics*, pp. 8–11, October 1994.

[19] F. G. Lobo, D. E. Goldberg, and M. Pelikan, "Time complexity of genetic algorithms on exponentially scaled problems," University of Illinois at Urbana-Champaign, Tech. Rep. 2000015, 2000.

[20] C.-H. Choi, K.-Y. Nam, Z. Yu, and R. W. Dutton, "Impact of gate direct tunneling current on circuit performance: A simulation study," *IEEE Transactions on Electron Devices*, vol. 48, no. 12, pp. 2823–2829, December 2001.