Methodology to Generate Approximate Circuits to Reduce Process Induced Degradation in CNFET Based Circuits

Kaship Sheikh, Lan Wei

Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada

***Abstract* – A systematic methodology is presented to generate approximate circuits with fewer nodes and shorter paths to reduce process induced degradation due to imperfect process in emerging technologies such as CNFET. In a 16-bit CNFET adder example, at $PCNT_{open}$ =5%, two resulted approximate adders achieve 80.5% and 90.2% circuit-level pass rate with a penalty of 3.3% and 24.0% in relative logic error, respectively, in comparison with 12.5% pass rate for the precision counterpart. The study paves the path to practically utilize such technology for error-resilient applications.**

I. Introduction

With superior device performance shown at extremely small dimensions, emerging low dimensional materials (LDMs) [1] [2] technologies including Carbon Nanotube Field Effect Transistor (CNFET) have shown the potential to replace Si as channel material for future transistors. Among the LDMs, CNFET seems to be the one close to high volume manufacturing [3] [4] [5], with development of wafer-scale process and experimental demonstration of sub-10nm CNFET being shown having superior device performance in comparison to Si [6]. However, the current material and process quality for CNFETs, like other LDMs, is still far from replacing Si in the near future for applications requiring precision. However, these technologies can be suited for error-tolerant applications such as approximate computing. Approximate computing being error tolerant relax the need for precise circuits, thus approximate circuits can be used in place of precise circuits providing the benefits of energy efficiency, area etc [7] [8]. Furthermore, process induced degradation is less likely to occur in approximate circuits (ACs), which have simpler topologies, reduced number of nodes and stages.

Traditional ACs are designed for silicon technology with mature materials and high-yield process, for the purposes of reducing delay, energy consumption, area, etc [7] [8] [9]. However, for the low yield technologies like CNFETs, ACs can be obtained with the main aim of reduced process induced degradation by having reduced nodes/stages and also reduced capacitances at few nodes.

In this paper, we present a systematic methodology to obtain AC for emerging technology like CNFETs, aiming to reduce circuit-level degradation due to imperfect process and materials. With the example of 16-bit CNFET adder, we show the approximate adders obtained using our methodology have significantly low process-induced degradation at reasonable logic accuracy.

The rest of the paper is organized as follows: In section II, we discuss about the methodology to provide link between process imperfection to circuit-level performance. This is followed by our systematic methodology to generate ACs with reduced process-induced degradation in section III. In section IV, we discuss about the 16-bit approximate adders obtained using the methodology in section III, and also compare approximate adders to the precise adder in terms of circuit-level performance, logic accuracy, area.

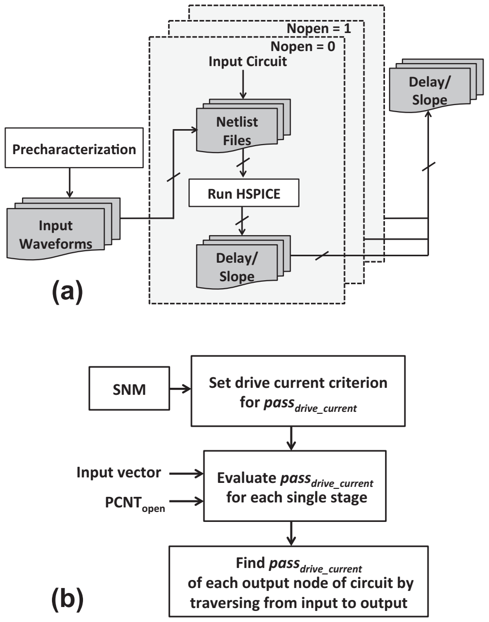
Nopen = 1
Nopen = 0
Input Circuit
Delay/ Slope
Netlist Files
Precharacterization
Run HSPICE
Input Waveforms
Delay/ Slope
(a)
SNM
Set drive current criterion for $pass_{drive\_current}$
Input vector
$PCNT_{open}$
Evaluate $pass_{drive\_current}$ for each single stage
Find $pass_{drive\_current}$ of each output node of circuit by traversing from input to output
(b)

Fig 1. (From Top to Bottom) (a) Steps to obtain Delay/Slope LUTs for an input circuit for one arc at given $PCNT_{open}$ (b) Steps to calculate $pass_{drive\_current}$ for each output node of the circuit at given $PCNT_{open}$ [10].

II. Evaluate process-induced circuit-level degradation

Among the emerging LDM based technologies, CNFET was the first studied and by far the closest to high volume manufacturing [11] [3]. The popular separation-placement process has regularly reported removing >99.9% metallic CNTs causing unwanted short circuits [3] [12]. However, unwanted open circuit remains a major issue due to missing CNTs in the channel trenches (thus no effective channel connecting source and drain). The probability of having a trench not covered by a CNT ($PCNT_{open}$) is in the range of 10% (wide channel) to > 30% (narrow channel) in the recent reports [11]. High $PCNT_{open}$ means reduction in drive current, resulting in two major issues: (1) failure to meet

978-1-5386-6790-3/18/$31.00 ©2018 IEEE

minimum frequency requirement due to increased critical path delay, and (2) degradation in Static Noise Margin (SNM).

In [10], we develop a methodology to link process imperfection including $PCNT_{open}$ with circuit level degradation. Delay at particular $PCNT_{open}$ is obtained by combining delay look-up table (LUT) results from HSPICE simulations (using Stanford VS-CNFET model [13]) under different input transitions (rising/falling) and input slopes (Fig. 1(a)). For SNM, simulation shows a roughly linear relationship between reduced drive current (increasing $PCNT_{open}$) and reduced SNM. We define a drive current criterion based on the simulation: for a certain stage, the drive current with $PCNT_{open} > 0$ in both pull-up or pull-down paths have to be at least 70% of those with $PCNT_{open}$=0, to ensure SNM of at least $0.25V_{DD.}$ We define the circuit level pass rate ($pass_{drive\_current}$) as the probability of that all the stages along the path from input to output nodes meeting the drive current criteria. For each $PCNT_{open}$ and input vector, $pass_{drive\_current}$ at the worst- case output is obtained (Fig. 1(b)). The results are then averaged for all 100 random input vectors.

III. Obtain ACs to reduce process induced degradation

Both delay and $pass_{drive\_current}$ are closely impacted by the number of nodes and stages to reach an output. Hence, it is critical to have short paths from all inputs to all outputs to reduce process induced degradation. We define "*Linked Node Number*" (*LN#*) as the total number of nodes encountered along the paths from all contributing inputs to an output.

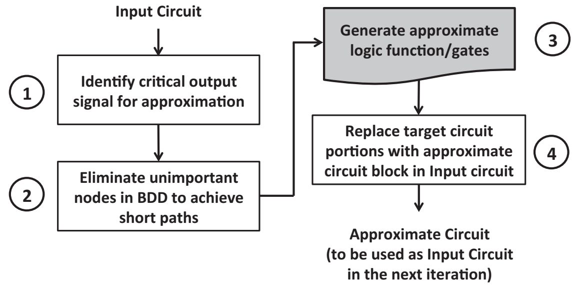
Input Circuit
1
Identify critical output signal for approximation
2
Eliminate unimportant nodes in BDD to achieve short paths
3
Generate approximate logic function/gates
4
Replace target circuit portions with approximate circuit block in Input circuit
Approximate Circuit (to be used as Input Circuit in the next iteration)

Fig 2. Steps to obtain approximate circuit by replacing the circuit portions contributing to the critical output in the input circuit. Steps 1 to 4 are to be repeated to obtain the final approximate circuit.

The procedure to obtain approximate circuit is carried in following steps (Fig. 2) with a case study of a 16-bit Han Carlson tree adder (Fig. 3). We first explain the procedure where approximations are done for the entire circuit, i.e., the entire precision circuit is considered as "Input Circuit" in the first iteration in Fig 2. (1) Every iteration starts with the selection of the critical output signal, the one with the highest *LN#*. (e.g., S_{15} in precise Han-Carlson adder in Fig. 3 in the first iteration.) (2) For the signal identified in step 1, approximate Binary Decision Diagram (BDD) is obtained by applying Cudd_SubsetShortPaths algorithm in [14]. Here, the unimportant nodes are removed, while retaining the short paths in the BDD, critical for logic accuracy consideration. (3) The approximate logic function (gates) for the critical signal is derived based on the approximate BDD. (4) Next, the circuit portions in the input circuit between inputs and the critical output signal are replaced by the approximate circuit block from step 3, to obtain the overall approximate circuit. In the next iteration, the approximate circuit obtained from step 4 of the previous iteration is now considered as "Input Circuit". This is followed by selection of next critical output (e.g after the approximation of S_{15}, the next critical signal will be S_{14} in Fig. 3), and followed by steps 2 to 4. This procedure is repeated till we reach the outputs in the circuit that have lesser or equal number of *LN#* than the approximate circuits for the previously selected outputs.

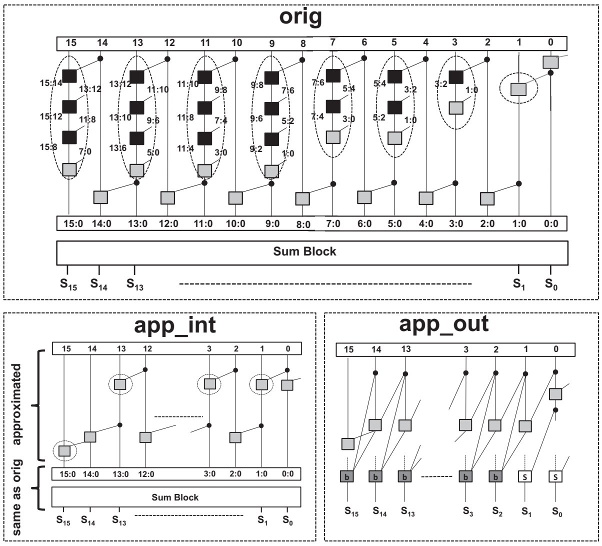

Fig 3. Schematics of 16-bit adders: precise Han Carlson Tree adder (orig), approximate adders with partial-circuit approximation (app_int), and approximate adder with whole-circuit approximation (app_out). For app_out, blocks s and b are described in Fig 4. The vertical broken line on top of s and b in app_out (full connection not shown to avoid congestion) represents the bit wise propagate signal (P_i). For the orig circuit, the dotted circles illustrate the circuit blocks, which are replaced by approximate circuit block to obtain app_int.

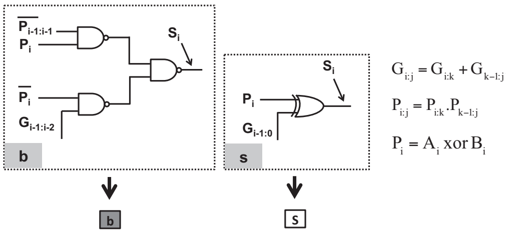
$\overline{P_{i-1:i-1}}$
P_i
S_i
$\overline{P_i}$
$G_{i-1:i-2}$
b
S_i
P_i
$G_{i-1:0}$
s
$G_{i:j} = G_{i:k} + G_{k-1:j}.P_{i:k}$
$P_{i:j} = P_{i:k}.P_{k-1:j}$
$P_i = A_i \text{ xor } B_i$
b
s

Fig 4. Schematics of 1-bit approximate adder modules ('b', 's') used in app_out. $G_{i:j}$, $P_{i:j}$, P_i are group generate, group propagate, and bitwise propagate signal respectively. In the paper, module 'b', 's' are either represented by the box symbols or by letters 'b', 's'.

The circuit obtained above might not meet the logic accuracy requirement. In this case, only partial circuit is used as "Input Circuit", approximated following the procedure from step (1) to (4) iteratively. Some intermediate nodes are chosen that the paths after these nodes are fixed and not included in the approximation procedure. Instead of just affecting the 1 primary critical output, these intermediate nodes affect two or more outputs (e.g. signal $G_{13:0}$ of orig in the Fig. 3 affects two sum outputs S_{14}, S_{15}). The circuit obtained this way will be logically more accurate. However, it might suffer from increased *LN#* in comparison to the case where approximation is done for the whole circuit including paths all the way till the output nodes.

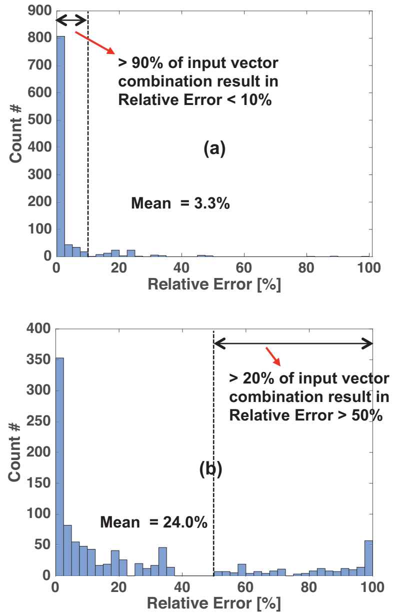

Fig 5. (Starting from top) Histogram shows *%Relative Error* for approximate circuit (a) app_int and (b) app_out. For app_int, > 90% of the input vector combinations have *Relative Error* < 10%.

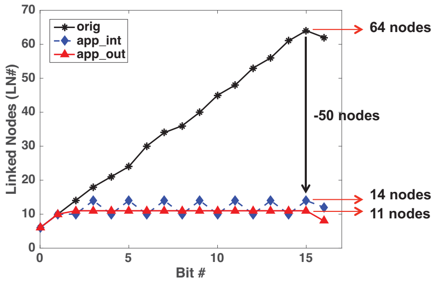

Fig 6. *LN#* for each sum output from (S_0) LSB to (S_{15}) MSB including C_{out}. The x-axis is the bit position. Bit # (0, …,15, 16) represent outputs (S_0, …, S_{15}, C_{out}) respectively. Using approximate circuits (app_int, app_out) significantly reduces the number of *LN#*. The critical outputs (with the highest *LN#*) of orig, app_int and app_out have *LN#* of 64, 14 and 11, respectively.

Regardless whether the approximations are done at intermediate or output nodes, the approximate circuit obtained from the proposed procedure would have reduced number of *LN#* for the critical output, hence enhanced $pass_{drive\_current}$. Moreover, with less number of stages in the critical path and reduced capacitances at nodes due to simpler topologies, the critical path delay and circuit area would also be reduced.

IV. Example of CNFET 16-bit Han-Carlson Adder

We have taken 16-bit Han Carlson tree adder for case study. Without the loss in generality, we apply the proposed methodology to the whole and partial circuit to construct two 16-bit approximate adders (app_out and app_int, respectively) (Fig. 3). app_int has all the approximations in the internal tree structure, without any change in the sum block which remains the same like the precise version (orig). The sum block in both orig and app_int is formed by XOR gates [15] towards their output for generating the sum signals from S_0, S_1, … S_{15}. The solid black and grey blocks in Fig. 3 follow the conventional design for group propagate and group generate as in [15]. In comparison to the precise version (orig), app_int has lesser number of group generate/propagate cells. app_out has the approximations for each output signal from S_2 to S_{15} (sum outputs) including C_{out} (Carry out). app_out is composed of 1-bit adder modules 'b' (Fig. 4) for the sum output S_2 to S_{15}, while sum outputs S_0, S_1 are composed of 's' block (similar to the precise version). Table I reports key comparison of the precise adder and the two approximate adders.

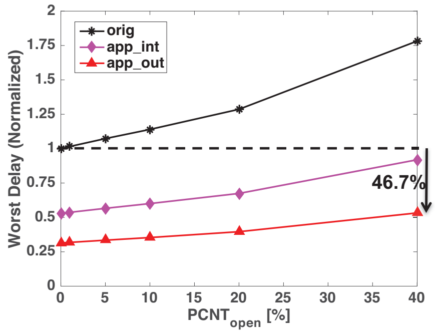

Fig 7. Worst-case Delay (normalized to that of the precise circuit with $PCNT_{open}$ = 0%) as a function of $PCNT_{open.}$ The worst-case Delay for approximate circuit app_out is lower by 46.7% at $PCNT_{open}$ = 40% in comparison to worst-case Delay for precise adder (orig) at $PCNT_{open}$ = 0%. For approximate circuit app_int, the worst Delay at $PCNT_{open}$ = 40% is also lower (less than 8.1%) in comparison to precise adder (orig) at $PCNT_{open}$.

Logic error - We define the term *Relative Error* (= $|S_{approx} – S_{orig}|/ S_{orig}$) to represent the relative logic error, where S_{orig}, S_{approx} are sum value based on the output of precise and approximate adders respectively. Fig. 5 shows *%Relative Error* of app_int and app_out over a set of 1000 random input vectors. app_int (from partial circuit approximation) has lower logic error with > 90% of the input vectors resulting in *%Relative Error* < 10%. In comparison, app_out (from whole circuit approximation) have *%Relative Error* > 50% for > 20% of input vectors. On average, *%Relative Error* of 24.0% and 3.3% are reported for app_out and app_int respectively (Table I).

Improvement in process induced degradation - Fig. 6 shows that *LN#* for the critical output (S_{15}) for the orig is quite high (64 nodes). However, for app_int, *LN# =14* is achieved for critical output, implying significantly less process induced degradation. For app_out, *LN#* are further reduced to 11. Fig. 7 shows that Worst Delay (among set of 100 random vectors) of the critical output for each of the orig, app_int and app_out. Even at $PCNT_{open}$ = 40%, the

Delay for both app_int and app_out are lower than Delay of orig at $PCNT_{open}$ = 0% (lower by 8.1% and 46.7% respectively).

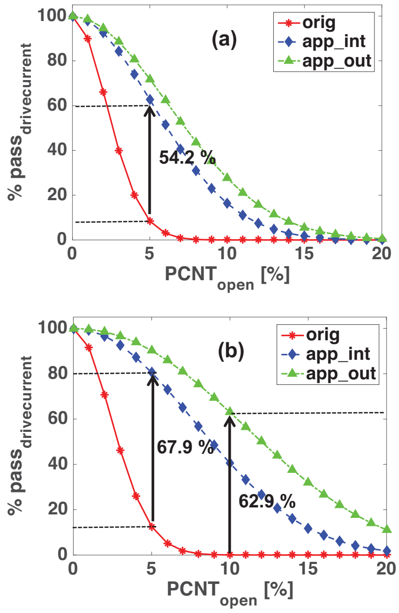

Fig 8. (Starting from top) Mean of *%passdrive_current* plotted for critical output of precise adder (orig) and approximate adders (app_int, app_out) (a) Without transistor upsizing, (b) With transistor upsizing of 10%-15% while still maintaining total area smaller than precise circuit area.

Table I: Comparison of 16-bit precise Adder ('orig') with approximate adders ('app_int' and 'app_out').

CKT	Mean Relative Error [%]	Normalized Delay* [$PCNT_{open}$ = 40%]	%Pass$_{drive\_current}$ (no upsizing) [$PCNT_{open}$ = 5%]	%Pass$_{drive\_current}$ (with upsizing) [$PCNT_{open}$ = 5%]	Normalized Area (no upsizing)
orig	0%	1.78X	8.4%	12.5%	1X
app_out	24.0%	0.53X	71.8%	90.2%	0.78X
app_int	3.3%	0.92X	62.7%	80.5%	0.78X

* Normalized to delay of orig at $PCNT_{open}$ = 0%

Fig. 8 shows the plot for Mean of $\%pass_{drive\_current}$ along the critical output for orig, app_int, and app_out over set of 100 random vectors. At $PCNT_{open}$ = 5%, $\%pass_{drive\_current}$ is improved to 62.7% and 71.8% with app_int and app_out respectively, in comparison with 8.4% in the precision counterpart (Fig. 8(a)). Moreover, with transistor upsizing by 10%-15%, which still keeps the area of ACs smaller than that of the precise one, $\%pass_{drive\_current}$ is above 80% for both approximate adders at $PCNT_{open}$ = 5% (90.2% for app_out and 80.5% for app_int) (Fig. 8(b)). The number is above 50% with app_out at $PCNT_{open}$ = 10%, implying the potential adoption of CNFET technology even at the current process maturity level.

V. Conclusions

Aiming at reducing process-induced degradation for CNFET circuits, we propose and demonstrate a systematic methodology to generate approximate circuits which could greatly reduce number of nodes and lengths of paths at a tolerable relative logic error. Significant improvement in process induced degradation has been observed in the 16-bit adder example, implying the great potential to practically utilize CNFET technology for error-resilient applications. The methodology can be adapted to other emerging technology with imperfect process.

Acknowledgement

We would like to thank Dr. Shu-Jen Han for valuable discussions. This work is supported by NSERC Discovery Grant.

References

[1] B. Radisavljevic et al., "Single-layer MoS2 transistors," *Nature nanotechnology*, vol. 6, no. 3, p. 147, 2011.

[2] L. Li et al., "High-performance p-type black phosphorus transistor with scandium contact," *ACS nano*, vol. 10, no. 4, pp. 4672-4677, 2016.

[3] S. J. Han et al., "High-speed logic integrated circuits with solution-processed self-assembled carbon nanotubes," *Nature nanotechnology*, vol. 12, no. 9, p. 861, 2017.

[4] M. M. Shulaker et al., "Carbon nanotube computer," in *Nature*, vol. 501, 2013, pp. 526-530.

[5] T.F. Wu et al., "Brain-inspired computing exploiting carbon nanotube FETs and resistive RAM: Hyperdimensional computing case study," in *Solid-State Circuits Conference-(ISSCC), 2018 IEEE International*, 2018, pp. 492-494.

[6] A.D. Franklin et al., "Sub-10 nm carbon nanotube transistor," *Nano letters*, vol. 12, no. 2, pp. 758-762, 2012.

[7] V. Gupta et al., "Low-power digital signal processing using approximate adders," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 32, no. 1, pp. 124-137, 2013.

[8] S. Venkataramani et al., "SALSA: systematic logic synthesis of approximate circuits," in *Proceedings of the 49th Annual Design Automation Conference*, 2012, pp. 796-801.

[9] M. Soeken et al., "BDD minimization for approximate computing," in *Design Automation Conference (ASP-DAC), 2016 21st Asia and South Pacific*, 2016, pp. 474-479.

[10] K. Sheikh, and L. Wei, "Evaluation of Circuit Performance Degradation due to CNT Process Imperfection," in *VLSI Technology, Systems, and Applications (VLSI-TSA), 2018 International Symposium on*, 2018.

[11] B. Kumar et al., "Spatially Selective, High-Density Placement of Polyfluorene-Sorted Semiconducting Carbon Nanotubes in Organic Solvents," *ACS nano*, vol. 11, no. 8, pp. 7697-7701, 2017.

[12] D. Zhong et al., "Solution-processed carbon nanotubes based transistors with current density of 1.7 mA/um and peak transconductance of 0.8 mS/um," in *Electron Devices Meeting (IEDM), 2017 IEEE International*, 2017, pp. 5.6.1-5.6.4.

[13] C.-S. Lee and H.-S. P. Wong. (2015) Stanford Virtual-Source Carbon Nanotube Field-Effect Transistors Model. nanoHUB. doi:10.4231/D3BK16Q68.

[14] F. Somenzi, "CUDD: CU decision diagram package release 3.0. 0," *University of Colorado at Boulder*, 2015.

[15] D. Harris and I. Sutherland, "Logical effort of carry propagate adders," in *Signals, Systems and Computers, 2004. Conference Record of the Thirty-Seventh Asilomar Conference on*, 2003, pp. 873-878.