Gate Input Reconfiguration for Combating Soft Errors in Combinational Circuits

Warin Sootkaneung and Kewal K. Saluja

Department of Electrical and Computer Engineering
University of Wisconsin-Madison

Motivations

- Soft errors are:
  - dependent on circuit inputs.
  - dependent on the signal values on inputs to logic gates – for example input of 01 has different probability of soft error relative to input of 10 to any of two- input AND/OR/NAND/NOR gates.

How to reconfigure gate input pins to harden the gate/circuit against soft error?

Outline of Presentation

- Introduction
- Contributions
- Soft Error Models
- The Gate Input Reconfiguration Technique
- Evaluation
- Conclusion
- Remarks

Introduction (1 of 2)

- Soft error characteristics:
  - Soft error is a transient error
    - Induces transient glitches at primary outputs.
    - Potentially causes permanent bit flips in memory element(s).
  - Soft error is caused by particle strikes near strong reverse-biased junction of a device.
  - Types of particles:
    - Alpha particles from package impurities
      → improved well by package technologies
    - Neutrons carried by cosmic rays
      → cannot be shielded easily

Introduction (2 of 2)

- Circuit design approaches can combat soft errors due to neutrons
- Smaller technology nodes are more vulnerable
  - To even very low-energy neutron strikes.
  - To performance degradation when soft error immunity features are added.

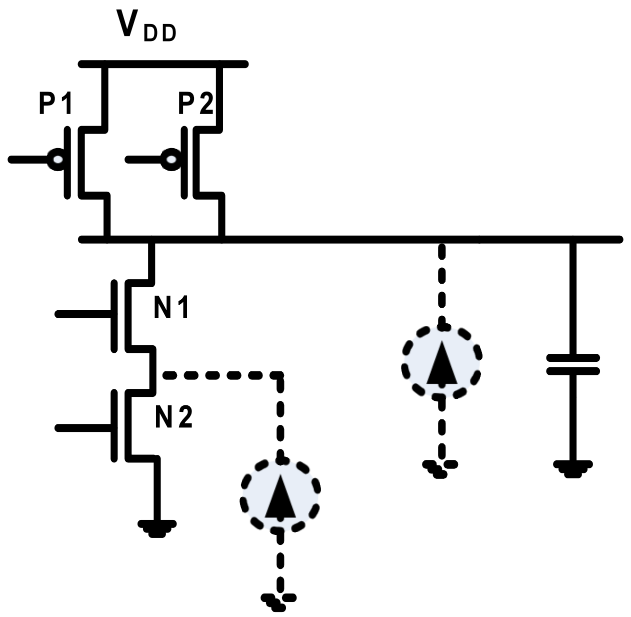
Contributions

- Explore in depth the input dependence of soft errors using our simulator
  - For all gate types in the library
  - For various benchmark circuits
  - Under technology node variations
- Introduce a gate input reconfiguration approach to reduce soft errors
  - Almost overhead-free

Soft Error Models

(1 of 6)

- Neutron hit is modeled as a current source



$$I(t) = \frac{Q}{T}\sqrt{\frac{t}{T}}e^{-t/T}$$

– Q = amount of charge deposition caused by a strike
  - Critical charge (Q_{crit}) is Q that causes gate output to change more than $Vdd/2$ (the gate fails)
  - Q can be (+) or (-) depending on the hit occurring on PMOS or NMOS

– T = time constant

Soft Error Models (2 of 6)

- Transistor Level Simulation:
  - Determination of Q_{crit} for each gate type in the library for each possible input combination to the gate
  - Determination of Q_{crit} under technology node variations
- Energy transfer from particle to silicon
  - Determination of the strike energy producing Q_{crit}

Soft Error Models (3 of 6)

- Mapping neutron energy to neutron flux
  - Use the *JEDEC89A* standard to obtain $P_{i(t,j)}$ - total neutron flux above the energy producing Q_{crit} of a gate i, transistor t, and gate input j
    - The larger the value of Q_{crit}, the smaller the amount of neutron flux
    - $P_{i(t,j)}$ is in the unit of the strike rate per unit area

Soft Error Models (4 of 6)

- Probability of Failure Estimation (step1)
  - Determination of logical masking probability through logic simulation/fault injection
    - Provide 100,000 random inputs to each circuit
    - Soft error injection: complement each gate output and record the *Error Count*, $E_{i(j)}$ corresponding to gate *i* and gate input vector *j*
      - $E_{i(j)}$ is updated if this error injection can propagate to primary outputs

Soft Error Models (5 of 6)

- Probability of Failure Estimation (step2)
  - Calculate the *probability of failure, POF,* of transistor *t* and gate input vector *j* for a gate *i*

$$Tr\,POF_{i(t,j)} = \frac{1}{k} \bullet P_{i(t,j)} \bullet Ad_{i(t)} \bullet w_{i(t)} \bullet E_{i(j)}$$

» *k* = total number of simulated input vectors

» *Ad*$_{i(t)}$ = *active area = drain area of sensitive transistor*

» *w*$_{i(t)}$ = *weighting factor = active area / circuit area*

» *POF* is in the unit of *1/s*

Soft Error Models (6 of 6)

- Probability of Failure Estimation (step3-4)
  - Sum *Tr POFi(t,j)* values to obtain the *POF* of each Gate

$$GatePOF_i = \sum_{i(j)} \sum_{i(t)} TrPOF_{i(t,j)}$$

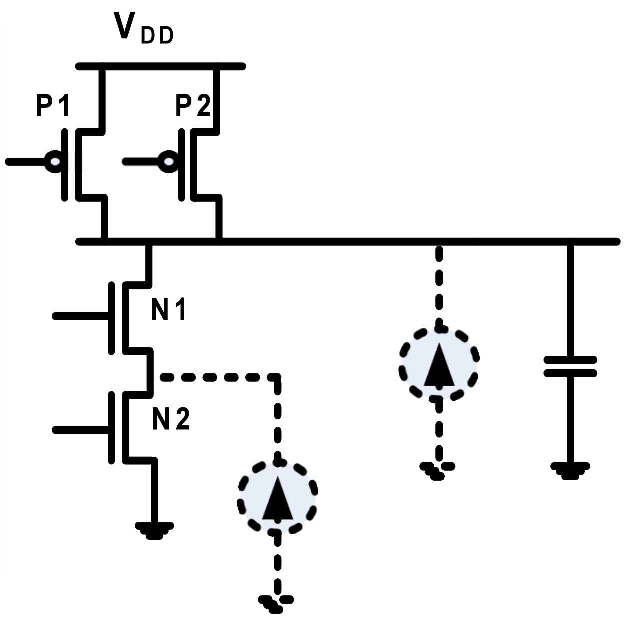
  - Sum *POF* of each gate *i* to obtain the *POF* of the circuit

$$CircuitPOF = \sum_i GatePOF_i$$

The Gate Input Reconfiguration Technique (1 of 4)

- Q_{crit} of a two-input NAND gate with 90 nm

Input	P1	P2	N1	N2
00	Not sensitive	Not sensitive	42.4	Not sensitive
01	Not sensitive	Not sensitive	21.5	<u>117</u>
10	Not sensitive	Not sensitive	22.2	21.7
11	29.8	29.8	Not sensitive	Not sensitive



"01" potentially has lower soft error rate than "10"

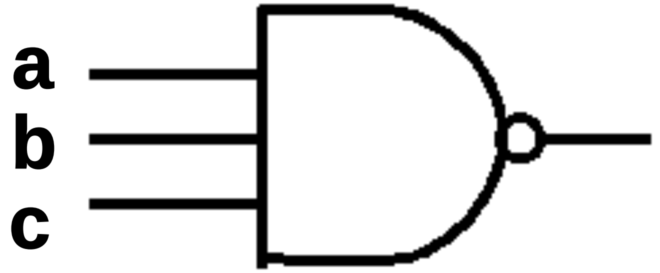
The Gate Input Reconfiguration Technique (2 of 4)

- Basic concept
  - For all input configurations of each gate, at least one configuration gives minimum *GatePOF*
  - For a gate *i*
    - with *n* number of input pins
    - Containing each possible configuration $config_l$
    - The optimal value of a configuration, $config_{optim}$, for a gate *i* is chosen

$$GatePOF_{i(config_{optim})} = \min\left[GatePOF_{i(config_l)}\right], l \in \{1, .., n!\}$$

The Gate Input Reconfiguration Technique (3 of 4)

- Example for a three-input NAND gate
  - Original input pin order: a-b-c
    - For each possible input to the circuit, the possible input to this gate is one of NAND3(j) where j= 000, 001, ..., 111
  - There are 6 input pin configurations:



$config_1$ = a-b-c	$config_2$ = a-c-b	$config_3$ = b-a-c
$config_4$ = b-c-a	$config_5$ = c-a-b	$config_6$ = c-b-a

  - Calculate each $GatePOF_{NAND3(config_i)}$ by swapping the original $E_{NAND3(j)}$ to the E of new input position, e.g.
    - For the $config_2$ in which b and c are swapped,

original $E_{NAND3(001)} \rightarrow E_{NAND3(010)}$

The Gate Input Reconfiguration Technique (4 of 4)

- Algorithm

```
// Start at gate $i$ = 1
For ($i$ = 1; $i$ <= $total\ number\ of\ gates$; $i$++)
    {
    // Start at input configuration $l$ = 1.
    /* Note for a gate $i$ with $n_i$ input pins, there are $n_i$!
    configurations. */
    For ($l$ = 1; $l$ <= $n_i$! $l$++)
        {
        Calculate $GatePOF_i$ for each $config_l$ ;
        }
    $GatePOF_{i(config_{optim})} = \min[GatePOF_{i(config_l)}]$ ;
    }
Calculate $CircuitPOF$ ;
```

Evaluation

- Experimental benchmark circuits information
  - Various ISCAS'85/'89 and ITC suit circuits were evaluated
  - Device information
    - Operating temperature: 25^{O} C
    - 65 and 90 nm predictive technology nodes
    - Cell library consists of 2-, 3-, and 4-input NAND, NOR gates, and Inverters

Evaluation (2 of 7)

- Experimental benchmark circuit information

Circuit	Circuit Information		
	#PIs	**#POs**	**#Gates**
C432	36	7	159
C1196	32	31	472
C6288	32	32	2672
i6	138	67	340
i7	199	67	512
i8	133	81	1685
S13207	700	790	9577
S15850	611	684	12101

Evaluation (3 of 7)

- Reported Results
  - All *POF* values are normalized with respect to the original circuit layout
  - The smaller the value, the better it is

Evaluation (4 of 7)

- Gate input reconfiguration vs. upsizing technique for 90 nm benchmark circuits

Circuit	Gate Input Reconfiguration	Percent Upsize		
		5%	10%	15%
c432	0.82	0.89	0.53	0.48
c1196	0.88	0.89	0.53	0.48
c6288	0.97	0.86	0.10	0.09
i6	1	0.87	0.62	0.55
i7	0.72	0.88	0.38	0.34
i8	0.58	0.87	0.33	0.30
s13207	0.96	0.88	0.40	0.36
s15850	0.92	0.87	0.56	0.50

Evaluation (5 of 7)

- Gate input reconfiguration vs. upsizing technique for 65 nm benchmark circuits

Circuit	Gate Input Reconfiguration	Percent Upsize		
		5%	10%	15%
c432	0.81	0.94	0.87	0.82
c1196	0.88	0.93	0.83	0.77
c6288	0.97	0.94	0.87	0.78
i6	1	0.94	0.87	0.82
i7	0.70	0.94	0.87	0.82
i8	0.55	0.94	0.86	0.81
s13207	0.96	0.94	0.88	0.83
s15850	0.93	0.92	0.83	0.76

Evaluation (6 of 7)

- Gate input reconfiguration for 65 and 90 nm benchmark circuits

Circuit	Gate Input Reconfiguration	
	65nm	90nm
c432	0.81	0.82
c1196	0.88	0.88
c6288	0.97	0.97
i6	1	1
i7	0.70	0.72
i8	0.55	0.58
s13207	0.96	0.96
s15850	0.93	0.92

Evaluation (7 of 7)

- Combination of gate input configuration & upsizing techniques

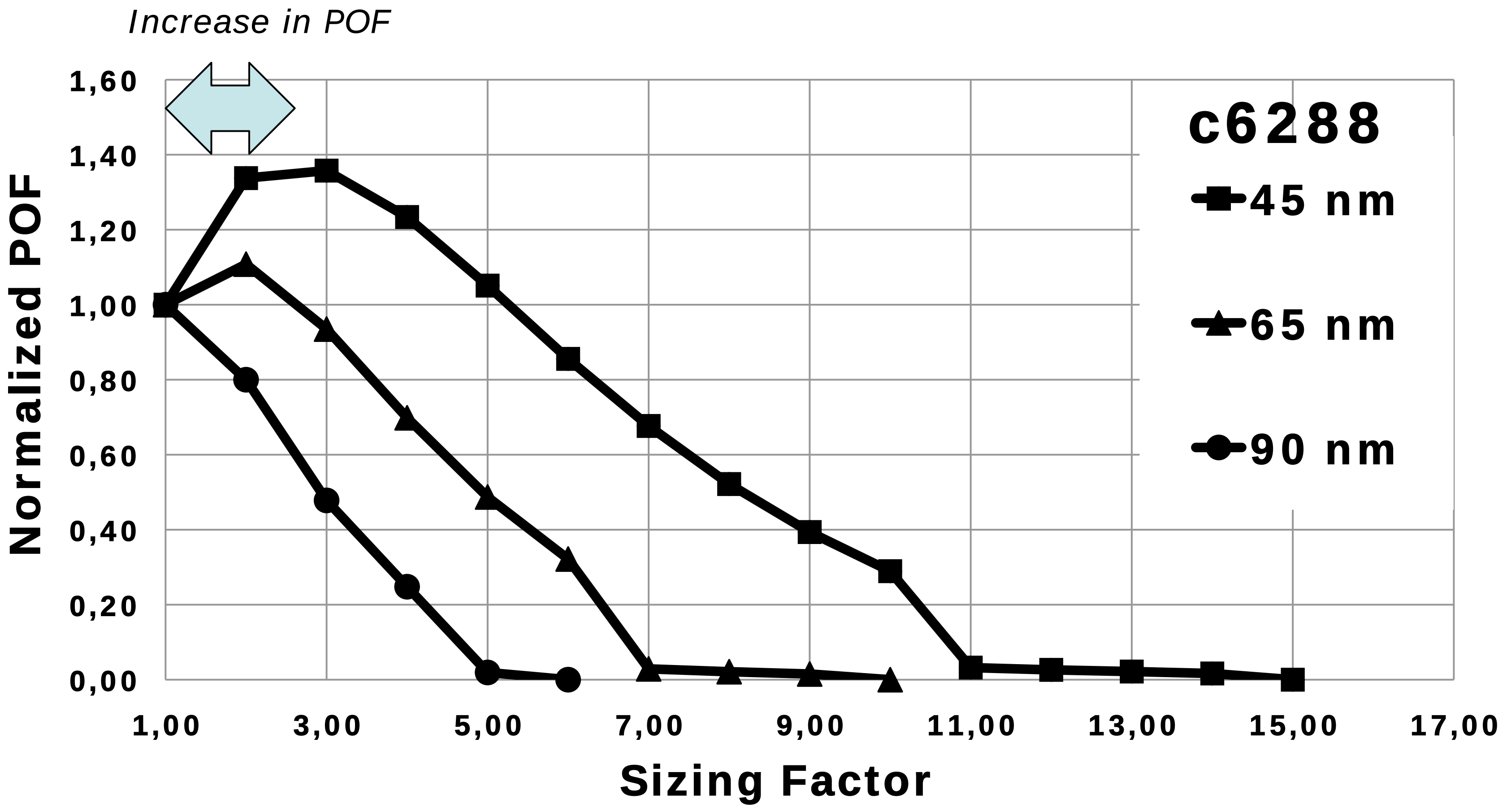
Circuit	Gate Input Reconfiguration and upsizing					
	65nm			90nm		
	5%	10%	15%	5%	10%	15%
c432	0.76	0.69	0.64	0.71	0.45	0.40
c1196	0.45	0.40	0.38	0.77	0.44	0.40
c6288	0.89	0.82	0.75	0.82	0.1	0.09
i6	0.94	0.87	0.82	0.87	0.62	0.55
i7	0.79	0.73	0.69	0.63	0.36	0.32
i8	0.51	0.46	0.41	0.50	0.28	0.24
s13207	0.90	0.83	0.77	0.84	0.38	0.34
s15850	0.86	0.78	0.71	0.81	0.55	0.48

Conclusion

- The gate input reconfiguration technique alone
  - Is almost overhead-free
  - Provides very impressive soft error rate reduction (as much as 45% in some circuits)
- The combination of gate input reconfiguration and upsizing techniques
  - Achieves even larger soft error rate improvement

Remarks (1 of 2)

- The limit of upsizing technique



Remarks (2 of 2)

- Upsizing methods need changes for adapting to sub-micron circuits
  - Optimization formulation $\rightarrow$ takes large CPU time
  - Heuristics $\rightarrow$ fast
    - Uses fault-sensitivity based upsizing techniques
    - Requires fairness of area distribution

The End

- Any Question?

New Results

• Saturation consideration: old vs. new algorithm for 45 and 65 nm benchmark circuits with 5% overhead and the most 2.5% sensitive gates are selected.

Circuit	45nm		65nm	
	Old	New	Old	New
c432	0.91	0.87	0.87	0.85
c1196	0.77	0.75	0.75	0.73
c6288	0.90	0.90	0.88	0.88
i6	1.00	0.92	1.00	0.91
i7	0.87	0.85	0.84	0.82
i8	0.80	0.79	0.78	0.75
s13207	0.86	0.83	0.83	0.81
s15850	0.90	0.86	0.87	0.83