

New Approaches to Total Power Reduction Including Runtime Leakage

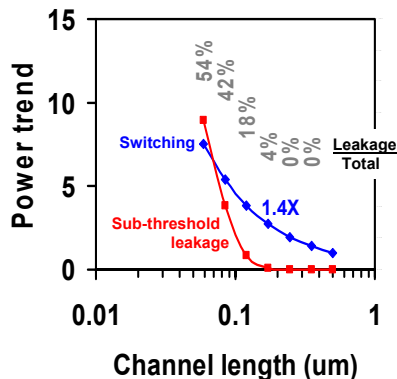
Dennis Sylvester

University of Michigan, Ann Arbor
Electrical Engineering and Computer Science
<http://vlsida.eecs.umich.edu>
dennis@eecs.umich.edu

March 1, 2004

Colleagues on this work: Prof. David Blaauw, Ashish Srivastava, Dongwoo Lee, Harmander Deogun, Rajeev Rao, Saumil Shah

Components of power dissipation



- Increasing contribution of static (leakage) power
- Leakage is significant in both standby mode (mobile apps) and runtime (high-performance non-mobile parts)

Figure source: Intel

Reducing Power Dissipation

- Pressing need to reduce power dissipation
 - High-performance designs
 - Packaging / cooling costs
 - Power supply integrity
 - Reliability (temperature)
 - Mobile applications
 - In addition to above: Battery life
- Circuit performance is generally determined by a small fraction of the gates
 - Requires the availability of very high performance devices
 - Higher Vdd
 - Lower threshold voltage
 - Aggressive gate length
- All gates in the design contribute to power dissipation
 - Would like to use slower devices whenever possible (higher Vth, lower Vdd, possibly longer gate lengths)

Multiple Vth

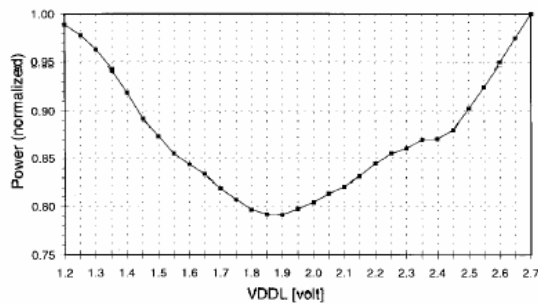
- Exponential reduction in leakage power
- Cost : Additional masks
- Value of higher threshold
 - Tradeoff: Delay penalty ↔ Leakage reduction
- Can be easily incorporated into standard design flows
 - Multi-threshold library
 - Tradeoff: Library size ↔ runtime
 - Generally threshold selection is done at gate level
 - 2X library size
- Provides runtime leakage power reduction
 - Contrary to standby mode based approaches

Multiple Vdd

- Quadratic reduction in switching power

$$P_{\text{switching}} \sim \alpha_{\text{sw}} \cdot C_L \cdot V_{\text{DD}}^2 \cdot f$$

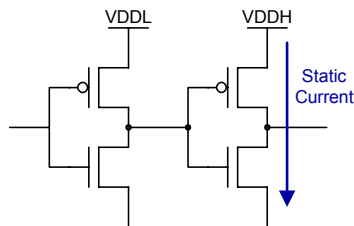
- Roughly cubic reduction in leakage power (DIBL, $V \cdot I_{\text{off}}$)
- Value of lower Vdd
 - 0.6 – 0.7 times $V_{\text{DD}_{\text{high}}}$
 - $0.5 \cdot V_{\text{DD}_{\text{high}}}$ in dual-Vth processes



Ref: Usami

Multiple Vdd - Topological Constraint

- $V_{\text{DD}_{\text{low}}}$ cells cannot be directly connected to $V_{\text{DD}_{\text{high}}}$ cells
 - PMOS does not turn off
 - Results in static current

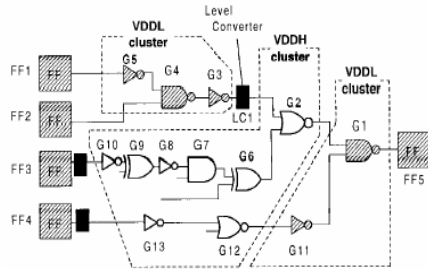
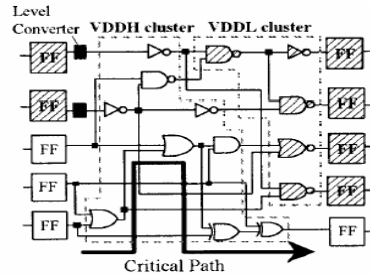


- Level converters (LCs) are used to up-convert a low Vdd signal to a high Vdd signal
 - Incurs delay and energy overhead

Multiple Vdd – 2 General Approaches

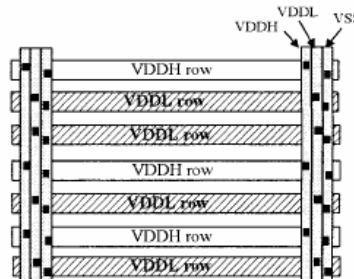
- Clustered Voltage Scaling (CVS)
 - Only one voltage transition along a path
 - Level conversion only at flip-flops

- Extended CVS (ECVS)
 - Multiple voltage transitions along a path
 - Level conversion using asynchronous LC's
 - 40 - 50% improvement in power observed



Other Issues in Multi-Vdd

- Generation of additional voltage supplies
- Impact on power grid design
- Hard to use standard design tools
 - Simple Power Compiler based approach found to provide only a 6% power reduction
 - Cell layout must change
 - Increase in routing costs



Outline

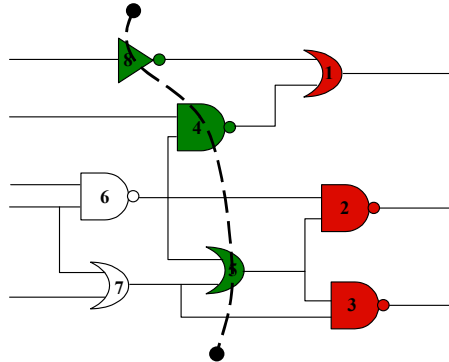
- **Concurrent Vdd/Vth assignment and sizing algorithm**
- Standby mode leakage reduction using state, Vth, and Tox assignment
- Runtime leakage reduction with bus encoding + novel Vth assignment strategies

Our Approach: Overview

- Seek to maximize total power reduction in a dual Vdd/Vth design
- Uses Vdd, Vth, and sizing: **VVS**
- VVS is a two-pass approach
 - Uses sensitivity metrics to minimize power in each pass
- 1st pass: CVS with concurrent up-sizing
 - Generates slack and allows for a larger fraction of gates to be set to low Vdd
- 2nd pass: Move back towards primary outputs (POs), setting gates to high Vth and re-setting gates to high Vdd or resizing to recover slack
 - Continue while total power dissipation is found to decrease

Gate Level Vdd/Vth Assignment

- Perform timing analysis and begin CVS
 - Initial circuit synthesized at $V_{dd_{high}}$ and $V_{th_{low}}$
- Obtain the candidate set of gates (front)
 - Do not serve as input to any high Vdd gate
 - If set to low Vdd will violate timing



Backward Pass

- Order candidates based on a metric
 - Slack, capacitance, etc.
- To meet timing size up gates
 - Gates to be sized up are obtained based on sensitivity
 - Size up until timing is again met
- Sensitivity = $\Delta D / \Delta \text{Area}$
 - $\Delta D = \sum_{\text{arcs}} \{ \Delta \text{delay}_{\text{arc}}(t) * 1 / (k + \text{Min}(\text{slack}) - \text{slack}_{\text{arc}}) \}$
 - k is a small positive number
 - Weights arcs that impact critical paths

Backward Pass, cont.

- Stopping criterion
 - When a gate is set to low Vdd only a fixed number of gates are upsized
 - The total power dissipation measure is not used in the hope to get out of local minimas
- The end of the pass is signaled when no candidate gates can be set to low Vdd
- The best seen solution is stored and is restored at the end of the pass

Forward Pass

- Now candidate gates which define the front are
 - Operating at low Vdd
 - Have all high Vdd as inputs
- Select gates on the front and set them to high Vdd/upsized
 - Select gates to be set to high Vt
 - Commit these changes if total power is found to decrease
 - Stop when no available options for gate upsizing/high Vdd assignment
- The gates are set to high Vth based on their sensitivity
 - Sensitivities of the form $\Delta\text{Power}/\Delta\text{Delay}$
 - Weighted by slack
 - All gates are candidates to be set to high Vth (no topological constraints)

Results

- 0.13 μ m process, timing constraint is 20% slower than absolute fastest design point (optimally sized, all Vdd_{high} and Vth_{low})
- Vdd_{high}=1.2V, Vth_{high}=0.23V
- Vdd_{low}=0.6V, Vth_{low}=0.12V

Circuit	Initial Power (μ W)			% Savings compared to initial design								
	Leakage	Switching	Total	CVS only			Backward Pass			VVS		
				Leakage	Switching	Total	Leakage	Switching	Total	Leakage	Switching	Total
c432	35.4	81.7	117.1	0.5%	1.9%	1.5%	0.5%	1.9%	1.5%	57.8%	6.0%	21.7%
c880	48.9	140.1	188.9	20.6%	19.8%	20.0%	20.6%	19.8%	20.0%	44.0%	22.9%	28.4%
c1908	75.3	202.7	278.0	5.4%	5.6%	5.5%	5.4%	5.6%	5.5%	44.1%	7.4%	17.4%
c2670	100.0	248.9	349.0	20.3%	21.4%	21.1%	20.2%	37.8%	32.7%	20.2%	37.8%	32.7%
c3540	131.6	302.6	434.2	3.4%	6.5%	5.6%	2.8%	26.4%	19.2%	49.4%	26.1%	33.2%
c5315	210.9	413.8	624.7	21.2%	25.4%	23.9%	18.9%	50.5%	39.9%	19.0%	50.7%	40.0%
c6288	544.3	1716.2	2260.5	1.1%	15.7%	12.2%	1.0%	15.8%	12.2%	20.3%	19.4%	19.6%
c7552	214.9	521.4	736.3	30.2%	32.7%	32.0%	36.4%	50.8%	46.6%	36.6%	51.2%	46.9%
Huffman	60.2	144.8	205.0	9.1%	9.3%	9.2%	20.9%	27.2%	25.4%	35.6%	27.0%	29.5%
SOVA1	1483.2	3270.1	4753.3	42.7%	45.3%	44.5%	50.7%	57.0%	55.0%	83.3%	58.6%	66.3%
SOVA2	3481.5	8016.7	11498.2	4.9%	5.1%	5.1%	41.5%	69.0%	60.7%	49.0%	69.8%	63.5%
Average	290.5	704.2	994.7	15.4%	18.4%	17.6%	17.7%	29.3%	25.8%	41.0%	30.7%	33.6%

High switching activity at primary inputs

CVS+sizing (backward pass) does much better than just CVS

Impact of Circuit Activity

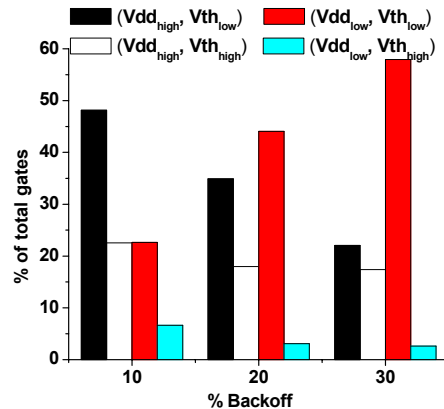
- For low activities the algorithm successfully steers toward a better solution by attacking leakage power more directly
 - In some benchmarks switching power is increased to minimize total power
 - Low activities \rightarrow converges dual-Vth + sizing
- VVS provides a single cohesive algorithm that seeks out best power reduction over a range of switching activities
 - Ex: across functional units in a design

Average power reduction by component across switching activities

Activity	Static	Dynamic	Total
High (3)	41%	31%	34%
Nominal (1)	69%	16%	45%
Low (1/3)	73%	7%	59%

Other results ...

- For high switching activities, VVS assigns many gates to low Vdd and low Vth combination to attack dynamic power
- Exhaustive cutset enumeration was performed to find optimal results
 - VVS performs close to optimal
 - Least effective when optimal front lies in middle of circuit (more possibilities)



Backoff	Initial Power (uW)	Final Power using VVS (uW)	Final Power using cutset enumeration (uW)	% Difference
1.2	117.10	91.70	91.70	0.00%
1.3	95.70	74.27	73.90	0.38%
1.4	78.60	57.84	56.90	1.20%
1.5	72.60	56.12	51.50	6.37%
1.6	66.70	48.40	46.80	2.40%

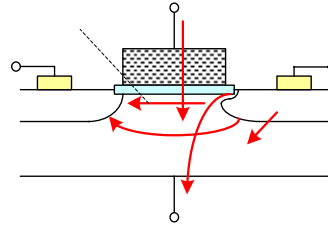
Outline

- Concurrent Vdd/Vth assignment and sizing algorithm
- **Standby mode leakage reduction using state, Vth, and Tox assignment**
- Runtime leakage reduction with bus encoding + novel Vth assignment strategies

Leakage Current Components

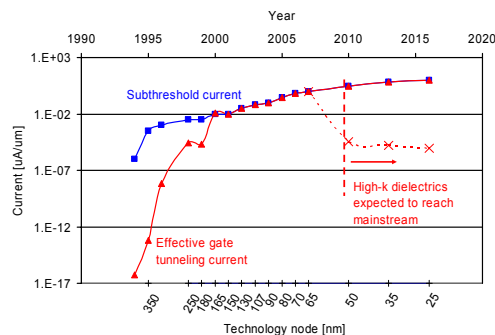
- Subthreshold leakage (I_{sub})

- Dominant when device is OFF
- Enhanced by reduced V_t from process scaling



- Gate tunneling leakage (I_{gate})

- Due to aggressive scaling of gate oxide thickness (T_{ox})
- A super-exponential function of T_{ox}
- Comparable to I_{sub} in 90nm technologies



Low Power Standby Mode

- Previous approaches to put a circuit into standby mode

- State assignment [Halter, CICC1997]
- Multi-threshold CMOS (MTCMOS) [Mutoh, JSSC1995]
- Dual- V_t assignment [Wei, DAC1998]
- Simultaneous state and V_t assignment [Lee, DAC2003]

- Only for subthreshold leakage reduction

- Proposed work

- Leakage current reduction in standby mode
- Minimize both I_{sub} and I_{gate}
- Simultaneous state, V_t and T_{ox} assignment
- Gate leakage for PMOS
 - One order of magnitude smaller than NMOS
 - PMOS I_{gate} is considered negligible in current analysis

Introduction – Dual V_t and Dual T_{ox}

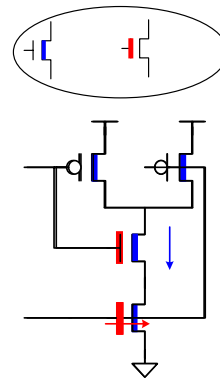
- Exploit dual oxide thickness technologies (becoming available)
 - Dual T_{ox} – for I_{gate} minimization
 - Dual V_t – for I_{sub} minimization

Assignment		Normalized values	
V_t	Oxide thickness	Leakage	Delay
Low	Thin	1.00	1.00
High	Thin	0.31	1.33
Low	Thick	0.51	1.26
High	Thick	0.05	1.69

- $\Delta T_{ox} \sim 3A$, $\Delta V_t \sim 120mV$, $I_{gate}/I_{leak} = 36\%$
- Both high V_t and thick T_{ox} : very large performance impact

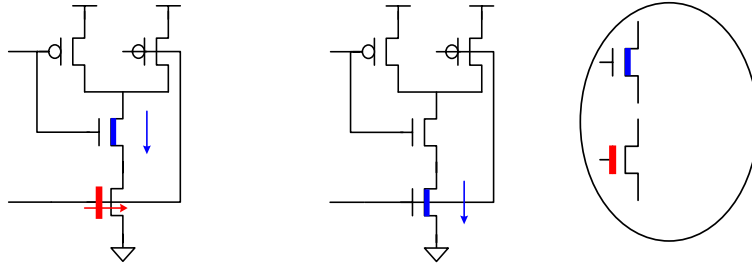
Overview of Approach

- If input state is unknown
 - Cannot be predicted which transistors will be ON or OFF
 - Some transistors must be assigned to both high- V_t and thick oxide
- Given a known input state
 - OFF device: I_{gate} is small
 - Considered only for high- V_t
 - ON device: no impact on I_{sub}
 - Only needs to be considered for thick T_{ox}
- A transistor need not be assigned to both high- V_t and thick T_{ox}
 - Significantly improved leakage/delay trade-off
- Only a subset of transistors need to be considered for high- V_t or thick T_{ox}



Exploit Input Pin Re-ordering

- I_{gate} dependence of input pin ordering [Lee, DAC2003]
 - I_{gate} depends strongly on the position of ON/OFF transistors
 - Place off-transistor at bottom of stack



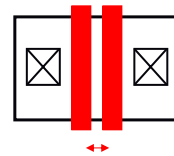
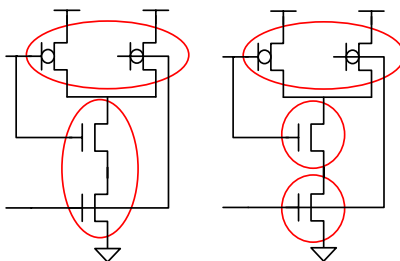
- Reduce performance penalty of thick-oxide transistors

0 i_1 p_1 p_2

Cell Library Options

- Library options
 - Trade-off points for a given gate
 - 4 vs. 2
 - Details in the paper (DATE04)
 - V_t or T_{ox} assignment control in a stack
 - individual-based vs. stack-based
 - Both libraries have the same number of cells

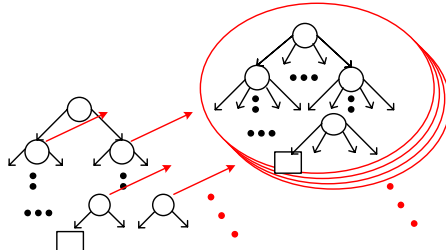
		# of trade-off points	
		2	4
Stack control	Individually	✓	✓
	Uniform	✓	✓



Design rule constraint for different V_t and T_{ox} assignment

Heuristics

- Exact solution has search space size of 2^{n+2m} (where n is # of PIs and m is # of gates)
- Branch and bound approach used
- Heuristic 1
 - Both state & gate tree: only one downward traversal
 - Gate tree: pre-sorted by leakage
 - Tends to produce a fast high quality solution
- Heuristic 2
 - Gate tree: only one downward traversal
 - State tree: search w/time limit
- Results indicate
 - Heuristic 1: fast runtime
 - Heuristic 2: better results



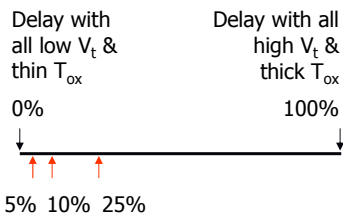
Results

Leakage current comparison between heuristics

- 5% of maximum delay penalty
- Baseline is avg of 10K random input vectors

Current: (uA), time: (sec)

	All Low V_t & thin T_{ox}	Heu1			Heu2		
		I_{leak}	X	Time	I_{leak}	X	Time
c432	24.5	6.9	3.6	2	3.8	6.5	1800
c499	65.8	24.8	2.7	6	23.4	2.8	1800
c880	50.1	8.7	5.7	7	7.7	6.5	1800
c1355	70.8	15.4	4.6	6	13.1	5.4	1800
c1908	56.7	14.7	3.9	4	13.5	4.2	1800
c2670	104.7	14.7	7.1	75	12.3	8.5	1800
c3540	128.5	21.6	6.0	17	19.9	6.5	1800
c5315	221.2	31.1	7.1	200	30.5	7.3	1800
c6288	346.8	114.7	3.0	63	107.5	3.2	1800
c7552	270.0	32.6	8.3	393	31.3	8.6	1800
alu64	260.0	42.2	6.2	455	40.4	6.4	1800
AVG			5.3		6.0		



Results

- Leakage current comparison vs. previous work
 - At 25% delay penalty

	All low V_t & thin T_{ox}	V_t & State		V_t , T_{ox} & State	
		I_{leak}	X	I_{leak}	X
c432	24.5	8.2	3.0	2.7	9.2
c499	65.8	23.8	2.8	7.5	8.8
c880	50.1	16.2	3.1	7.0	7.1
c1355	70.8	23.9	3.0	7.6	9.3
c1908	56.7	18.2	3.1	6.2	9.2
c2670	104.7	30.0	3.5	11.3	9.2
c3540	128.5	40.3	3.2	13.7	9.4
c5315	221.2	70.6	3.1	24.1	9.2
c6288	346.8	112	3.1	36.8	9.4
c7552	270.0	84.2	3.2	28.3	9.5
alu64	260.0	75.3	3.5	28.0	9.3
AVG			3.1		9.1

Results

- Leakage current comparison between cell library options
 - At 5% delay constraint

	All low V_t & thin T_{ox}	4-option individually		2-option individually		4-option uniform stack		2-option uniform stack	
		I_{leak}	X	I_{leak}	X	I_{leak}	X	I_{leak}	X
c432	24.5	6.9	3.6	7.5	3.3	6.7	3.7	7.8	3.1
c499	65.8	24.8	2.7	27.6	2.4	26.2	2.5	28.6	2.3
c880	50.1	8.7	5.7	9.0	5.6	9.4	5.3	10.3	4.8
c1355	70.8	15.4	4.6	17.0	4.2	22.4	3.2	23.8	3.0
c1908	56.7	14.7	3.9	15.2	3.7	15.2	3.7	15.8	3.6
c2670	104.7	14.7	7.1	12.2	8.6	16.2	6.5	14.8	7.1
c3540	128.5	21.6	6.0	23.9	5.4	25.2	5.1	24.7	5.2
c5315	221.2	31.1	7.1	30.7	7.2	32.1	6.9	33.0	6.7
c6288	346.8	114.7	3.0	120.6	2.9	134.0	2.6	149.6	2.3
c7552	270.0	32.6	8.3	31.2	8.7	32.0	8.4	30.6	8.8
alu64	260.0	42.2	6.2	42.3	6.2	42.8	6.1	46.9	5.5
AVG			5.28		5.27		4.91		4.77

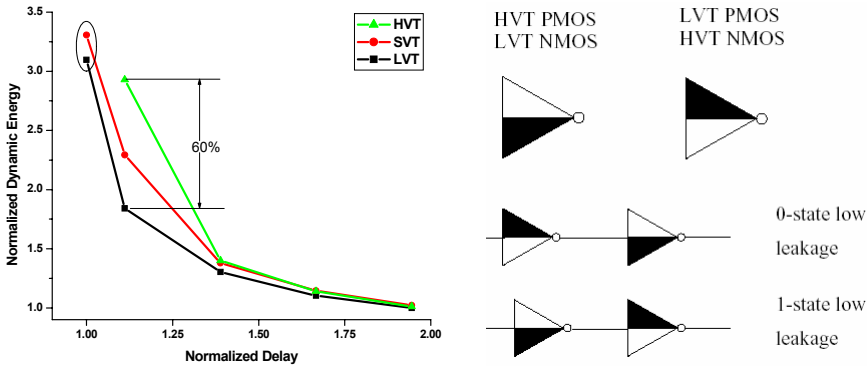
Outline

- Concurrent Vdd/Vth assignment and sizing algorithm
- Standby mode leakage reduction using state, Vth, and Tox assignment
- **Runtime leakage reduction with bus encoding + novel Vth assignment strategies**

Runtime leakage in buses

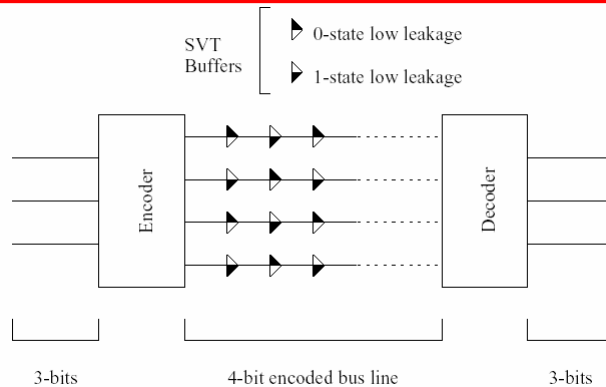
- 50% of total chip leakage in inverters/buffers
 - Much of this in repeaters which are:
 - Very wide
 - Growing in #
 - Heavily speed constrained so often use low Vth
 - Do not experience stack effect as multi-input gates do
- Standby leakage reduction relatively easy compared to runtime
 - We can absorb a delay penalty when we know that no new data is coming
 - In runtime, data can come at any time; must be ready to process as fast as possible
- What can we do besides dual-Vth?

Staggered Vth bus design



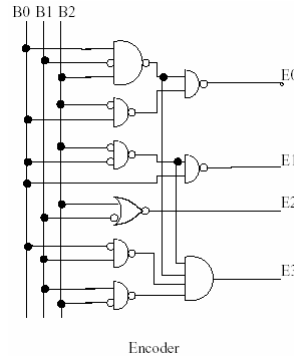
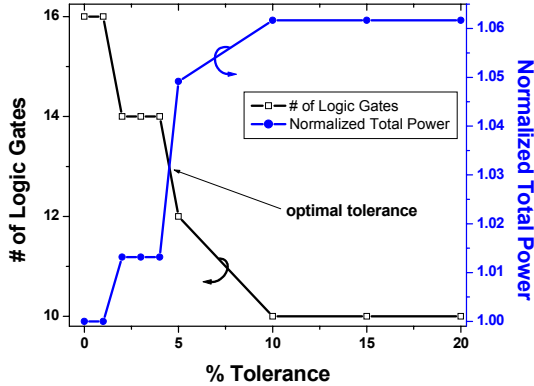
- Selective use of high-Vth devices yields the possibility of low leakage in runtime
 - Stagger them along the wire to create a very low leakage state
 - Delay (or dynamic energy) penalty is much lower than all high-Vth
 - We cannot dictate state in runtime so this does not help in general
 - Unless we **can** dictate state

Encoding to enforce proper state



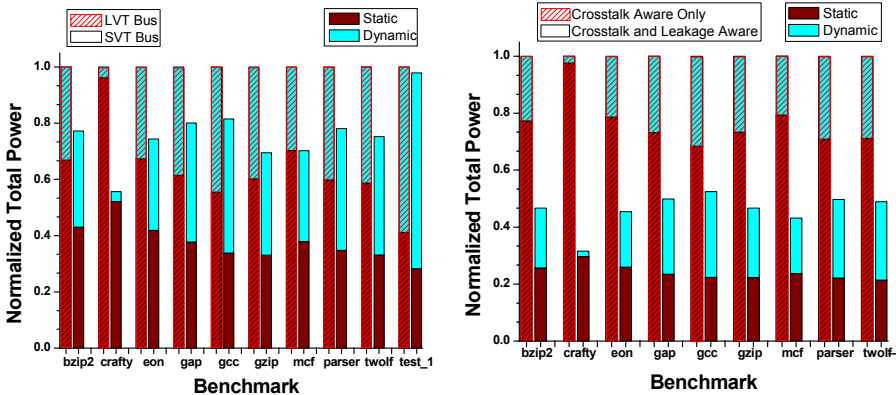
- Choose a 3→4 encoding, also eliminate worst-case crosstalk
- Exact encoding selected to minimize *total power*
 - Requires anticipated state and transition probabilities
 - Ex: what is the most common state, what is the most common transition
 - Also consider the encode/decode logic complexity

Reducing encoding complexity



- We consider all possible encodings (mappings from input states to actual transmitted encoded states) within T% of minimal
- Then use logic complexity as tiebreaker
- Results in 1-2% power penalty with 13% fewer gates/area overhead

Results (includes delay overhead)

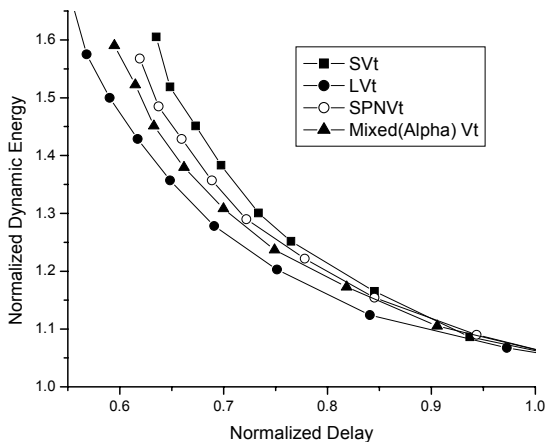


- 0.13um CMOS at 105C, 64-bit Alpha architecture running 9 applications (address bus)
- 26% total power savings on average, 42% leakage reduction
- Maximal switching activity case (Test_1), total power still reduced
- Compared to previous crosstalk-aware approaches, we save 54% total power (nearly all of it in leakage)

Alternate Repeater Vth Assignments

- Other possibilities of Vth assignment in repeaters can help reduce leakage in runtime
- **Separate NMOS/PMOS Vth (SPNVt)**
 - All PMOS are low-Vth, all NMOS are high-Vth
 - Advantages: predictable leakage (state independent), balances fast/slow paths through the repeater chain, easy to manufacture
- **Mixed Vth**
 - Wide devices such as in repeaters are split into parallel fingers, separated by a contacted pitch
 - Assign a fraction, α , of total width to low-Vth ($1 - \alpha$ is then high-Vth)
 - Effectively a third Vth with speed and leakage behavior intermediate to high/low Vth
 - No manufacturing costs for this 3rd Vth, no area penalties since parallel fingers are spaced out significantly already

Vth assignment scheme results



Hybrid approaches are possible; upper bits in 64-bit address buses are usually zeroes → Stagger to favor low-leakage 0s

- Mixed config: $\alpha = 0.3$
- Achievable speed is best for mixed, also good for SPNVt
- Runtime leakage of $\alpha = 0.3$ is 54% lower than low-Vth with small dynamic energy penalty
- Total average power reduction is 14%
 - Switching behavior taken from 11 benchmark applications, address bus
 - Strongly depends on ratio of static to dynamic power

Conclusions

- Need to leverage “multi-everything” to address the power management gap
 - EDA must enable simultaneous sizing, Vdd, and Vth assignment; the 3 major knobs in power reduction
 - Total power reductions on the order of 35-60% are achievable
- Standby mode leakage can be effectively reduced by combining state assignment with Vth and Tox assignment
 - Sizable leakage reductions (5-9X) with modest delay penalties (3-15% vs. all low Vt and thin Tox)
 - Much less overhead than MTCMOS, body biasing
- Runtime leakage in global interconnect repeaters can be addressed using Vth assignment schemes (sometimes with encoding)
 - 40-54% leakage reductions with small dynamic power penalty
 - Total power savings depends heavily on static/dynamic ratio
 - Implies these techniques improve with scaling
 - Mixed Vth provides pseudo-continuous Vth assignment, opening up a range of new optimizations in the energy/delay design space