adequately model delay. In addition, accuracy is also essential at $V_{GS} = 0$ to ensure that leakage currents are modeled correctly. A convenient starting point is the onset of inversion, $V_{GS} = V_T$, where the current can be expressed as:

$$I_S = 2 n C_{ox} \frac{W}{L} (n_t)^2. \quad (1)$$

The model in (1) is based on the EKV formulas [7], with the subthreshold slope factor, mobility, oxide capacitance C_{ox}, and thermal voltage $n_t = kT/q$ as parameters. The current in the vicinity of V_T can often be modeled as:

$$I_{DS} = \frac{I_S \cdot IC}{k_{fit}}. \quad (2)$$

Here, IC represents the inversion coefficient, and k_{fit} is a model-fitting parameter. The inversion coefficient expresses the degree of inversion of the transistor, and covers both the subthreshold ($IC < 1$) and above-threshold ($IC > 1$) regions. While the introduction of the IC parameter leads to simple current expressions, the link to supply voltage is somewhat lost, as IC is a strongly nonlinear function of V_{DD}, as described in (3):

$$IC = \ln\left(e^{\frac{1 \cdot V_{DD} - V_T}{2 n_t}} - 1\right)^2; \text{ or}$$
$$V_{DD} = \frac{V_T - 2 n_t \ln\left(\sqrt{IC} - 1\right)}{1}. \quad (3)$$

in which η represents the DIBL factor.

The leakage current at $V_{GS} = 0$ can be expressed using (4), based on the EKV formulas:

$$I_{Leakage} = I_S e^{\frac{-V_{DD} - V_T}{n_t}}. \quad (4)$$

Finally, we must ensure that the models around threshold (2) and at the cutoff point (4) are based on the same set of technology parameters, which in our case will be accomplished by curve-fitting to transistor I–V simulations. Generally, such curve-fitting approaches may have inaccuracies to predict scaling trends, but the presented modeling approach can be used to quickly estimate fitting parameters for any technology. The objective is, thus, to develop compact and accurate models for design optimizations.

For a 65-nm CMOS technology, we sweep V_{DD} from 0.1 V to 0.6 V to extract model parameters for n- and p-devices with high- and low V_T, are considered to derive general process parameters for the technology and to compare different device options for ultralow-power design. We restrict the model

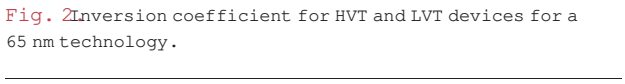

Fig. 2. Inversion coefficient for HVT and LVT devices for a 65 nm technology.

set that $IC = 1$ corresponds to $V_{GS} = V_T$. This means that $n_t \ln\left(e^{\frac{V_T - 2 n_t}{2 n_t}} - 1\right)$, as derived from (3).

We performed a simultaneous fitting of the parameters to both low- (V_T) (LVT) and high- (V_T) (HVT) transistors, as shown in Fig. 3. Since a single set of fitting parameters is used for both types of transistors, the mean-square error increased from 0.5% to around 1.5%, but the model is still very accurate. This allows us to evaluate V_T optimization in the energy–delay space. The current model will be used next as a baseline for the derivation of delay and energy models.

B. Delay Model

Based on the current model from the previous section, the model for delay analysis can be derived [12]. Substituting (1) and (2) into the alpha-power law model for delay, the gate delay can be expressed as:

$$t_p = \frac{k_{tp} C_L V_{DD}}{2 n C_{ox} \frac{W}{L} (n_t)^2} \frac{k_{fit}}{IC} \quad (5)$$

where k_{tp} will be the delay-fitting parameter. A first-order analysis of (5) is helpful to make the impact of design and operating variables explicit (as needed for design optimization). The C_L parameter is the load capacitance of the driving stage and includes the intrinsic capacitance of the fanout gates and the interconnect capacitance. The W in the denominator of (5) stands for the width of the transistors in the driving stage. For path-delay analysis, we annotate the W of the individual gates as stage i, respectively. Thus, C_L is proportional to $C_{ox} \gamma_i W_i (W_{i+1} + C_{parasitic,gate})$, where γ_i is the ratio of gate parasitic to intrinsic capacitance. L is the channel length, and C_{ox} and n_t are technology parameters. In this work, the denominator of (5)

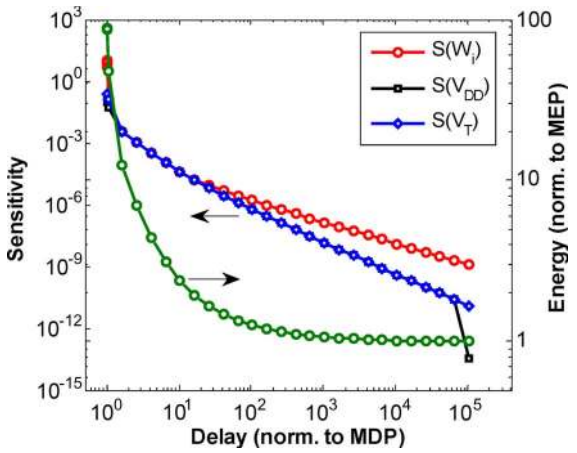

Fig. 4. ***Energy-delay sensitivity*** $S(x) = (\partial E/\partial x)/(\partial D/\partial x)$ ***to sizing*** (W_i), supply (V_{DD}) and threshold (V_T) voltage (left *y*-axis), ***and energy-delay trade-off (right*** *y*-***axis) for a 32-bit carry look-ahead adder.***

In (10) and (11), α represents the activity factor of the datapath, or the average activity for all gates. K_{sw} and K_{lk} represent technology (and fitting) constants. Note that the energy-per-operation E_{op} is the path energy E divided by the activity factor α, $E_{\text{op}} = E/\alpha$. As activity approaches zero, E_{op} would approach infinity. This may seem counterintuitive at first, but makes sense because no operation is performed at zero activity yet (leakage) energy is being dissipated. Separation of voltage- and size-dependent parameters in (11) will prove useful in the derivative analysis, discussed next.

III. SENSITIVITY ANALYSIS

In this section, we present a framework to analyze the impact of gate sizing, supply and threshold voltage on energy-delay trade-offs. The energy-delay trade-offs via voltage and gate sizing will be quantified using the concept of energy-delay sensitivity. The sensitivity to a parameter x represents a percent reduction in energy for a percent increase in delay, $S(x) = (\partial E/\partial x)/(\partial D/\partial x)$, [1], [31], [32]. Previous work [1], [32] has shown that sizing was the most effective around MDP. Here, the emphasis will be placed on the trade-offs around MEP. Let's examine the sensitivities of the optimization parameters along the optimal energy-delay (E-D) curve.

Fig. 4 shows simulated energy-delay sensitivity for an adder as well as optimal E-D trade-off when gate sizing, supply and threshold voltage are varied. Fig. 5 shows a closer look into areas around MDP [Fig. 5(a)] and MEP [Fig. 5(b)] to compare techniques for high-performance and low-energy design optimization. On the optimal E-D curve, the sensitivities of the active parameters are equal. Lower sensitivity represents more delay reduction for a fixed energy increase or less increase in energy for a fixed delay reduction. When the sensitivity to a parameter deviates from the lowest curve, such parameter has reached its constraint limit, and is no longer active to support further energy reduction. This is the case with V_T and sizing (W_i) at MEP [Fig. 5(b)], and V_T and V_{DD} at MDP [Fig. 5(a)]. As expected, near MEP, V_{DD} adjustment has the lowest sensitivity (it has least increase in energy for a given delay reduction), and thus the most effective parameter in delay reduction. Notice that we are looking at energy-delay sensitivity. Delay-energy sensitivity (as a measure of delay improvement for a given energy increase) to V_{DD} would be the highest, just like E-D sensitivity to sizing is the highest around MDP. As we traverse up the E-D curve, from Fig. 5(b) to Fig. 5(a), V_T also becomes significant, while sizing becomes significant only for high-V_{DD} and low-V_T scenarios, as we move towards high-performance regime in Fig. 5(a).

Sensitivity formulas (12)–(14), obtained from the delay and energy models from Section II, can be used to analytically calculate results from Figs. 4 and 5. Partial

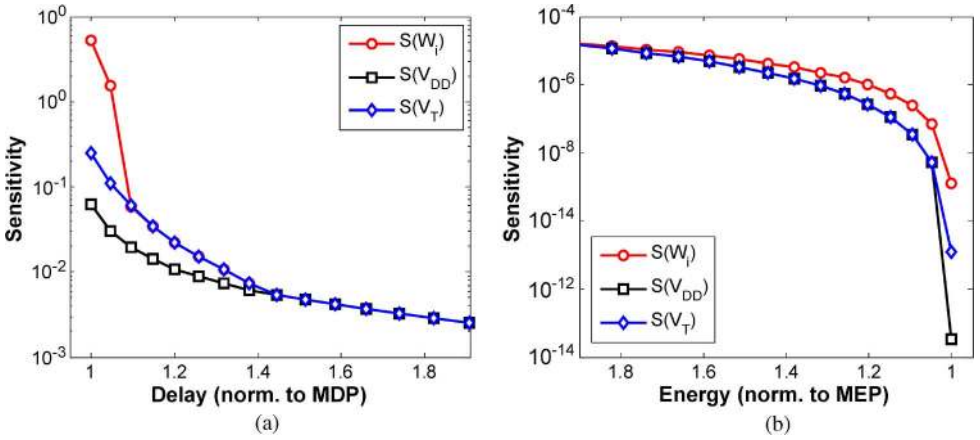

Fig. 5. ***Energy-delay sensitivity*** $S(x) = (\partial E/\partial x)/(\partial D/\partial x)$ ***near (a) MDP and (b) MEP for a 32-bit carry look-ahead adder from Fig. 4.***

derivatives with respect to V_{DD}, V_T, and W_i lead to the following sensitivity results:

$$S_{V_{DD}} = \frac{\partial E/\partial V_{DD}}{\partial D/\partial V_{DD}} = \frac{E_{sw}}{D}\cdot\frac{2}{1-N_0} + \frac{E_{lk}}{D} \cdot \frac{2+\sigma\cdot\frac{V_{DD}}{n\cdot\phi_t}-N_0}{1-N_0}$$

$$N_0 = \frac{1+\sigma}{\sqrt{IC}}\cdot\frac{V_{DD}}{n\cdot\phi_t} \quad (12)$$

$$S_{V_T} = \frac{\partial E/\partial V_T}{\partial D/\partial V_T} = \frac{E_{lk}}{D}\cdot(1-\sqrt{IC}) \quad (13)$$

$$S_{W_i} = \frac{\partial E/\partial W_i}{\partial D/\partial W_i} = \frac{ec_i}{K_d\cdot\frac{V_{DD}}{IC}\cdot(f_{i-1}-f_i)} + \frac{E_{lk}}{D} + \frac{E_{lk}}{K_d\cdot\frac{V_{DD}}{IC}\cdot(f_{i-1}-f_i)} \quad (14)$$

where f represents the effective fanout $f = g \cdot h$ for a gate.

To demonstrate the sensitivity of V_{DD} and W_i in the E-D space, Fig. 6 plots energy-delay optimization space when V_{DD} and W_i are individually tuned, starting from MEP. As predicted, scaling V_{DD} is much more effective than using W_i around MEP, because more delay improvement is possible for a given increase in energy. Actually, sizing is hardly effective until we get close to MDP.

Therefore, unlike MDP where sizing was the most dominant optimization variable, supply voltage should be used around MEP. This is because at MEP leakage current/energy is linear function of W_i and so is performance, while V_{DD} is more effective for performance increase than sizing because V_{DD} exponentially affects performance. Given the large disparity in sizing and supply sensitivities, we may reduce sizing (if possible) around MEP to create energy slack that can be utilized by a small increase in V_{DD} for overall performance increase. This is similar, albeit in different order of adjusting variables, to increasing V_{DD} around MDP to create timing slack that can be utilized by sizing for overall energy reduction [1]. These trade-offs are generally not possible at MEP/MDP since the sizing and supply variables reach their bounds at these extreme points, so the use of sizing (MDP) or V_{DD} (MEP) is the most optimal. Indeed, this is really good news for MEP region, because supply adjustment is easier to do than to adjust gate sizing. Gate sizing involves many more variables than simple V_{DD} scaling. Besides, global V_{DD} scaling does not require any layout changes and could be done after chip fabrication.

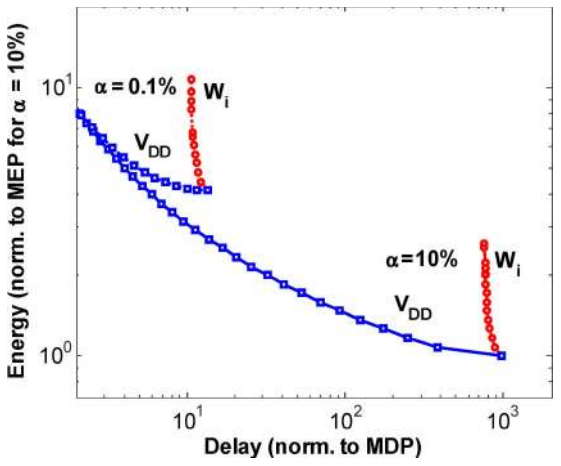

Fig. 6. ***Energy-delay trade-off after gate sizing*** (W_i) ***and voltage scaling*** (V_{DD}) ***for different activity levels for a 32-bit carry look-ahead adder from Fig. 4.***

IV. ENERGY–DELAY OPTIMIZATION

Most practical systems involve supply and sizing optimization, while threshold is selected from the available discrete values. This section explores supply and sizing optimizations for low- and high-V_T devices to compare options offered by the two thresholds. The optimization will then be expanded to include V_T, which can be performed at the device level (e.g., body-bias) and at the circuit level (e.g., type of logic family).

We start the optimization from MEP as a reference. Unlike MDP, which is a fixed point in the E-D space, MEP depends on circuit activity. Let's then first examine MEP as a function of activity factor and V_T. The discussion below is based on the 32-bit carry look-ahead adder example.

Plots in Fig. 7 show MEP and *IC* versus activity for high- and low-V_T designs. Since MEP is leakage-limited, HVT will always yield lower energy at the same activity. Under a very low activity factor, total energy of the circuit is dominated by its leakage energy, therefore the high-V_T cells gain significant advantage for low activity factors. For activity factor of 0.01%, for example, MEP of the HVT design achieves a 10-times lower leakage energy compared to the LVT design. Even under a high-activity factor of 10%, MEP of the HVT design is still lower in energy than that of the LVT design. It is also interesting to observe that *IC* corresponding to MEP greatly varies with the activity factor. For $\alpha = 0.1\%$, $IC = 5$ minimizes energy for low-V_T devices, while for $\alpha = 10\%$, MEP occurs around $IC = 0.03$ [Fig. 7(b)]. MEP is important, because it is the starting point in our optimizations. The plots in Fig. 7 do not indicate performance, which must be considered for a complete E-D comparison.

Optimal energy-performance trade-off of the same adder is shown in Fig. 8, along with the corresponding *IC* and V_{DD} curves in Fig. 9. From the E-D plot in Fig. 8, it is evident that although high-V_T cells achieve lower energy-per-operation than low-V_T cells, HVT has 10- to 100-times lower performance than LVT. Such large performance penalty for marginal energy reduction is highly undesirable in ULP design. For performance-constrained low-power

Fig. 9. Plot of (a) C_{load} and (b) V_{DD} vs. delay for a 32-bit carry look-ahead adder.

if V_T can be further lowered without increasing the subthreshold leakage current, $I_{Leakage}$. This is not possible in typical complementary static CMOS circuits, where $I_{Leakage}$ is tightly coupled to V_T. But it is definitely an option in circuits without gain, such as pass-transistor networks that can be designed to have no V_{DD}-to-GND subthreshold leakage paths.

One logic style that falls into this class is sense amplifier-based pass-transistor logic (SAPTL) [14], which attempts to decouple V_T from $I_{Leakage}$ by using pass-transistor (PT) networks to perform logic functions. The needed gain is provided using sense amplifiers and delay will as illsutrated in Fig. 12.

The SAPTL is composed of: a) a PT network called the stack; b) a root driver; and c) a sense amplifier (SA). SAPTL can operate synchronously using a clock, or asynchronously using additional hand-shaking circuitry. The stack has a single root node energized by the driver to ensure feedforward-only operation. The function inputs

Fig. 10. Energy vs. delay for a 32-bit carry look-ahead adder for various V_T adjustment options (LVT, HVT, variable).

steering the root current to either of the two pseudo-differential output nodes to signal either a logic 1 or a logic 0. This will produce a small voltage difference, V_{stack}, at the output of the stack. This voltage is then detected and restored to full-rail by the sense amplifier.

Since the pass-transistor stack has no V_{DD}-to-GND connections, the only effect of subthreshold leakage in the pass-transistors is a detrimental reduction of the pseudo-differential output nodes, Fig. 12. This also implies that the stack threshold voltage, $V_{T,stack}$, reduces the stack delay without any subthreshold leakage penalty as only V_{DD}-to-GND leakage paths appear in the sense amplifier and the driver.

This separation of concerns allows for simultaneous optimization of logic performance and static power dissipation. To maximize the logic performance, the thresholds of the pass-transistors can be lowered

Fig. 11. Optimal supply and threshold voltage vs. delay for a 32-bit carry look-ahead adder after sizing, supply and threshold voltage optimizations in Fig. 10. Lower activity dictates higher voltage.

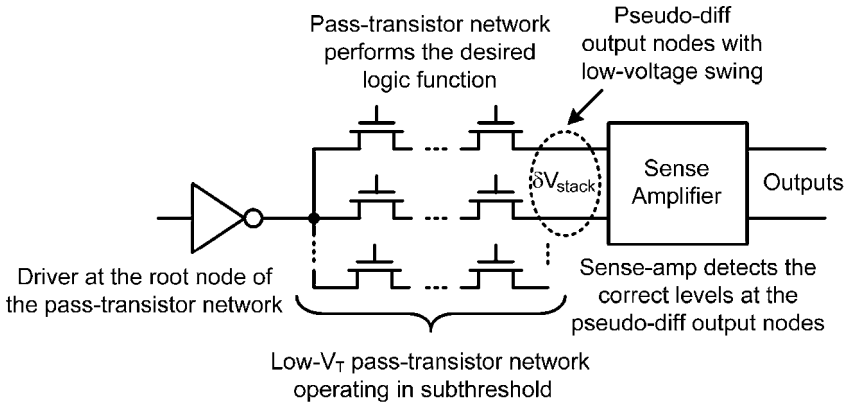

Fig. 12. ***Sense-amplifier-based pass-transistor logic (SAPTL) basic architecture.***

the energy needed by the sense amplifier to resolve the correct stack output becomes too large. Typically, a $\delta V_{\text{stack}} > 100$ mV is easily achievable at $V_{T,\text{stack}} \approx 100$ mV and can be detected with reasonable sense-amplifier energies, allowing the pass-transistors to operate comfortably in the $V_{T,\text{stack}} + \Delta V$ region.

Since $V_{T,\text{stack}}$ is different from the sense amplifier and driver threshold voltages, where leakage dominates at very low energy levels, operation in the near- or below-V_T region is desirable. One possible relation between threshold and supply voltages for the different components of the SAPTL is illustrated in Fig. 13. The pass-transistor stack has a threshold voltage $V_{T,\text{stack}}$ below the nominal V_T of logic. Stacking is the key factor for leakage control thus allowing for this configuration of logic gates.

The SAPTL delay can be expressed as the sum of the sense amplifier and driver delays, D_{active}, and the stack delay, D_{stack}. Assuming a simple dominant-pole model for the pass-transistor network, D_{stack} can be expressed as:

$$D_{\text{stack}} = \frac{k_1 \cdot n_{\text{depth}}^2}{V_{DD} - V_{T,\text{stack}}} \tag{15}$$

where n_{depth} is the depth of the pass-transistor network, i.e., the number of transistors traversed by the signal injected from the root to the output, and k_1 is a constant.

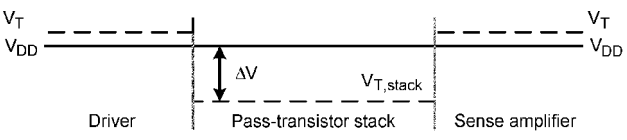

Fig. 13. ***One possible SAPTL supply and threshold voltage scenario showing subthreshold operation in the sense amplifier and driver and above-threshold operation in the pass-transistor stack.***

Thus, we can express the total SAPTL delay over M identical stages as:

$$D_{\text{SAPTL}} = M \cdot D_{\text{active}} + M \cdot \frac{k_1 \cdot n_{\text{depth}}^2}{V_{DD} - V_{T,\text{stack}}}. \tag{16}$$

Note that if the delay of the stack dominates, then reducing $V_{T,\text{stack}}$ is an effective way of reducing the delay.

The energy required by the SAPTL for a single operation is thus:

$$E_{\text{SAPTL}} = M \cdot C \cdot V_{DD}^2 + M \cdot V_{DD} \cdot \sum_{i=1}^{n_{\text{depth}}} V_i \cdot C_i + V_{DD} \cdot I_{\text{leak}} \cdot M^2 \cdot \left(D_{\text{active}} + \frac{k_1 \cdot n_{\text{depth}}^2}{V_{DD} - V_{T,\text{stack}}} \right). \tag{17}$$

The first two terms of (17) represent the active energy used by the sense amplifier and driver. Note that the voltage swing of the internal stack nodes can be kept well below V_{DD}. The last term represents the leakage energy due to both the driver and sense amplifier.

From (17), we can see that as $V_{T,\text{stack}}$ is reduced, the leakage energy is also reduced. In practice however, this increases the current flow in the off-path stack capacitances, and thus leads to a corresponding increase in off-path node voltages, which tends to cancel-out any energy reduction, but still allowing delay improvement. If we assume that for a certain logical operation, $n_{\text{depth}} \cdot M$ is a constant, i.e., it can be implemented using either many shallow SAPTL stacks or very few but deep stacks, we can then see that stack complexity and gain can be traded off against each other to achieve a desired energy-delay operating point.

In order to understand how various logic functions are implemented, consider the pass-transistor stack that implements a 4-input XOR function as shown in Fig. 14.

Fig. 14. ***A 4-input SAPTL XOR showing the pass transistor stack structure where each circle represents an NMOS transistor controlled by the corresponding input variable.***

Each path from the root of the stack to S represents a minterm and each path from the root to S represents a maxterm. It can be observed from Fig. 14 that the SAPTL implementation of XOR gates is very straightforward. By increasing the complexity of the stack, in this case increasing the number of inputs to the XOR gate, the sense amplifier and driver overhead per input can be reduced, at the expense of decreased performance. This can be seen in Fig. 15, where the energy and delay of a 6-input and 16-input SAPTL XOR gate are compared to their static CMOS equivalents. With the same V_T (equal to low-V_T), SAPTL reduces energy below MEP of CMOS due to longer stacks (higher effective V_T) and lower leakage.

The capability of SAPTL to decouple I_{Leakage} and $V_{T,\text{stack}}$ is illustrated using a self-timed 64-byte parallel CRC16 generator (as used in error detection). The threshold voltages of the pass-transistors (implemented using low-V_T devices) are reduced using varying degrees of forward body biasing. The simulated results are shown in Fig. 16 with supply voltage and activity as independent parameters. The simulation results show that the overall circuit delay can be reduced with almost no impact on energy even at low activity factors such as $\alpha = 1\%$. These results are constrained by the limited effectiveness of body biasing as a means to control $V_{T,\text{stack}}$. The availability of devices with even lower threshold would be desirable as it would increase the effectiveness of SAPTL for energy reduction.

As can be seen in Fig. 16, the performance improvement through body biasing is more prominent at the higher supply voltage $(V_{DD} = 0.5\ \text{V} > V_T)$, at which the delay of the stack dominates the total delay. At lower supply voltages $(V_{DD} = 0.3\ \text{V} \approx V_T)$, the delay of the sense amplifier as well as the hand-shaking circuitry [15] dominates since it is near the edge of subthreshold operation, limiting the performance gains obtainable through reduction of the $V_{T,\text{stack}}$. Circuit- and logic-level techniques are foundation for architecture-level optimizations, which will be next discussed in Section VI.

VI. ARCHITECTURAL OPTIMIZATION

Just as parallelism showed to be effective for energy reduction around MDP, time-multiplexing is best suited for performance increase around MEP. Architectural

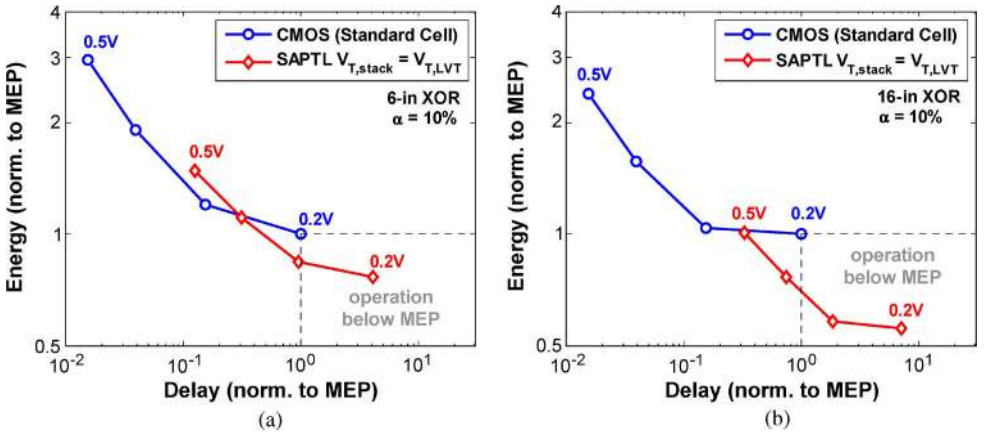

Fig. 15. ***Energy-delay characteristics of SAPTL designs: (a) 6-input XOR, (b) 16-input XOR. The plots show operation below MEP of static CMOS designs.***