3. Architectural Level

Low power architecture design techniques are important because the design analysis, verification and automated synthesis begin at this level. Typical architecture level design decisions involve the selection and organization of the functional macro of the design. The choice of clocking strategy, parallelism, pipelining, component organization, etc., are issues to be considered at this level.
The suppression of switching activities always involves some trade-off decisions. In general, hardware logic is required to suppress unwanted switching activities and the additional logic itself consumes power. Therefore, it is important to estimate how many switching activities can be eliminated by a particular technique so that it can be justified.

i.) Guarded Evaluation

Guarded evaluation is a technique to reduce switching activities by adding latches or blocking gates at the inputs of a combinational module if the outputs are not used. An example is depicted in Figure in which the result of the multiplication mayor may not be used depending on the condition selection of the multiplexer. To reduce switching activities, latches are added at the inputs of the multiplier. The latches are transparent when the result of the multiplication is to be used. Otherwise, the latches preserve the previous values of the multiplier inputs to suppress activities inside the multiplier because the result will not be used. the latches preserve the previous values of the multiplier inputs to suppress activities inside the multiplier because the result will not be used.

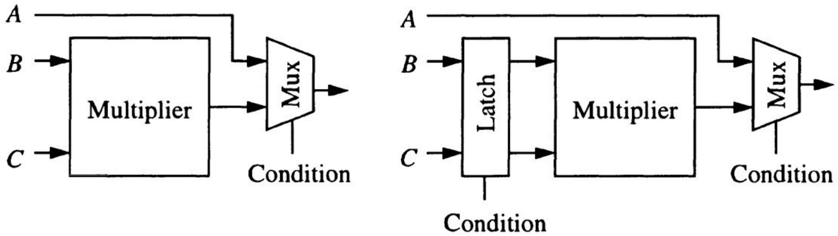

Fig. Guarded evaluation.

ii.) Glitch Reduction

An important architecture-level measure to reduce switching activity is based on delay balancing and the reduction of glitches. In multi-level logic circuits, the propagation delay from one logic block to the next can cause spurious signal transitions, or glitches as a result of critical races or dynamic hazards. In general, if all input signals of a gate change simultaneously, no glitching occurs. But a dynamic hazard or glitch can occur if input signals change at different times. Thus, a node can exhibit multiple transitions in a single clock cycle before settling to the correct logic level. In some cases, the signal glitches are only partial, i.e., the node voltage does not make a full transition between the ground and V_{DD} levels, yet even partial glitches can have a significant contribution to dynamic power dissipation.

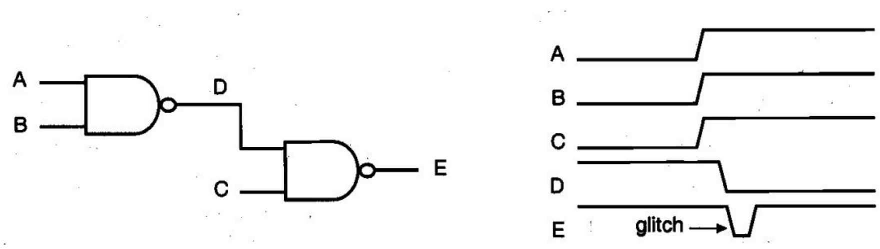

Fig. Signal glitching in multi-level static CMOS circuits.

Glitches of a signal node are dependent on the logic depth of the node, i.e., the number of logic gates from the node to a primary input or a sequential element. Generally, nodes that are logically deeper are more prone to signal glitches. One of the reasons is that the transition arrival times of the gate inputs are spreaded longer due to delay imbalances. Also, a logically deep node is typically affected by more primary input switching and therefore more susceptible to glitches.

One way to reduce glitches is to shorten the depth of combinational logic by adding pipeline registers. This is very effective especially for data path components such as multipliers and parity trees. Data path components are more glitch-prone because they have deep logic and the inputs switch simultaneously.

iii.) Parallel Architecture with Voltage Reduction

When the system design of a chip has the luxury of defining the system operating voltage, an interesting low power technique using parallel processing can be used. This stems from the observation that a higher voltage system can be operated at a higher frequency because of shorter average gate propagation delay. the maximum operating frequency of a CMOS gate is inversely proportional to its operating voltage. As the operating voltage approaches the transistor threshold voltage, the gate delay increases, thus effectively limits the maximum operating frequency of the system. However, a low voltage system is attractive for the obvious benefit of low power dissipation. To overcome this problem, parallelism can be applied so that the frequency requirement of the system is reduced, thus allowing the system designer to choose a lower operating voltage.

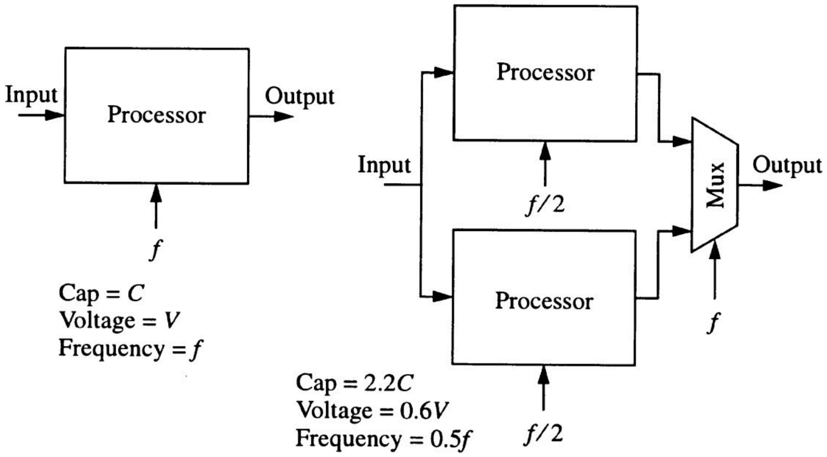

Fig. Power dissipation of uni-processing and parallel processing systems.

If we double the number of processing units, each unit can be operated at half the frequency /12. The desired system throughput is still maintained by multiplexing the outputs of the two processing units as shown in Figure 7.8. Since the operating frequency requirement is now reduced to half, the system operating voltage can be reduced to conserve power. As a thought experiment, let's assume that the average capacitance switched in the parallel system is 2.2C, slightly more than double due to some overhead in multiplexing and de-multiplexing the data. Suppose that the reduced operating frequency allows us to reduce the operating voltage of the parallel' system to 0.6V. The power dissipation parallel system is therefore:

$$P_{par} = (2.2C)\ (0.6V)^2\ (0.5f) = 0.396P_{uni}$$

IV.) Adiabatic Logic Circuits

In conventional level-restoring CMOS logic circuits with rail-to-rail output voltage swing, each switching event causes an energy transfer from the power supply to the output node, or from the output node to the ground. During a O-to-VDD transition of the output, the total output charge Q $=C_d.V_{DD}$ is drawn from the power supply at a constant voltage. Thus, an energy of $E_{suppl} = C_{load}\ V_{DD}^2$ is drawn from the power supply during this transition. Charging the output node capacitance to the voltage level VDD means that at the end of the transition, the amount of stored energy in the output node is E_{stored} = Cad $V_{DD}^2/2$. Thus, half of the injected energy from the power supply is dissipated in the PMOS network while only one half is delivered to the output node. During a subsequent V_{DD} to-0 transition of the output node, no charge is drawn from the power supply and the energy stored in the load capacitance is dissipated in the NMOS network.

To reduce the dissipation, the circuit designer can minimize the switching events, decrease the node capacitance, reduce the voltage swing, or apply a combination of these methods. Yet in all these cases, the energy drawn from the power supply is used only once before being dissipated. To increase the energy efficiency of logic circuits, other measures can be introduced for recycling the energy drawn from the power supply. A novel class of logic circuits called adiabatic logic offers the possibility of further reducing the energy dissipated during switching events, and the possibility of recycling, or reusing, some of the energy drawn from the power supply.