Designing Low-Power Digital CMOS

Gerard M Blair

The increasing levels of circuit integration are leading to the implementation of highly sophisticated algorithms. Many of the commercial application areas have a requirement for portability which leads to the need for low-power design. This article considers the issues and design solutions for complex low-power digital CMOS IC design.

INTRODUCTION

Advances in VLSI fabrication in recent years have greatly increased the levels of integration making possible the implementation of highly complex algorithms such as Viterbi decoders, discrete cosine transforms etc. Smaller integrated circuit device and feature sizes have lead naturally to increased speeds. Expectations and demand have grown for the continuing development of both speed and functionality. In particular, the communications market is expanding rapidly with each provider seeking advantage in enhanced system performance. However, this and other markets also have a requirement for low power consumption since many products are portable and so battery operated. Unfortunately, the rapid development in VLSI has not been reflected in developments in battery technology and so the impetus is upon VLSI designer to adapt emerging technologies to provide complex high-throughput low-power systems.

A further motivation for low-power design is simply that there are limits to how much power a component can dissipate without the need for special cooling[1]. High performance CMOS designs are already exceeding these limits and, for these, the costs of the ultimate system are dominated by the costs of the cooling apparatus. Significant market advantage can be gained by using low-power ICs which avoid the need for this additional hardware.

Fortunately, many aspects of the emerging integrated circuit technologies lend themselves to low-power design. Smaller feature size (developed for increased integration) also provide faster devices and smaller capacitances. Lower supply voltages (being developed to allow even more integration) also have a significant impact upon power consumption and speed. Increased levels of integration in themselves allow the designer to use the vacated area for extra circuits to compensate for the device speed reduction due to lower supply voltages.

The overall power consumption of an integrated circuit can be influenced at all levels of its design[2-4]: fabrication technology, circuit optimisation, logic design, control and clocking strategies, architectural partitioning and layout, and the underlying system's algorithm. This article will look at the power consumption of an integrated circuit and at how strategies for low-power can be found at each of these levels.

SOURCES OF POWER CONSUMPTION

There are two types of power consumption in digital CMOS. The first may be thought of as *useful* in that it establishes information by charging and discharging signal lines; the second type is *waste* and comes from short-circuit currents which flows directly from the power supply to ground.

The useful power dissipation is illustrated in figure 1 for a simple CMOS inverter. When the input signal is low, the p-type transistor is ON (and the n-type OFF) allowing the output capacitor to charge up to the supply voltage; when the input goes high, the n-type is switched ON (and the p-type OFF) and the output discharges to ground. The power dissipated in this manner is known as *dynamic* power since it only occurs which the output switches. Starting from school physics, power is the product of voltage (V) and current. In this case, the voltage is the supply voltage (V_{dd}) and the current is the rate at which charge moves from the power rail to ground. The charge moves by charging and discharging the output; if this has capacitance C then the amount of charge is (CV_{dd}) and the current is that charge multiplied by how often the output switches. Thus dynamic power dissipation is:

$$V_{dd} \cdot \sum_i (C_i \cdot V_{dd} \cdot A_i) \qquad (1)$$

where the summation is over each gate, and A_i is the *average* rate at which the output of gate i charges and discharges.

The power due to short-circuit current has the simple equation:

$$V_{dd} \cdot \sum_i I_{i\,sc} \qquad (2)$$

where the summation is over each gate and $I_{i\,sc}$ is the *average* short-circuit current flowing through gate i.

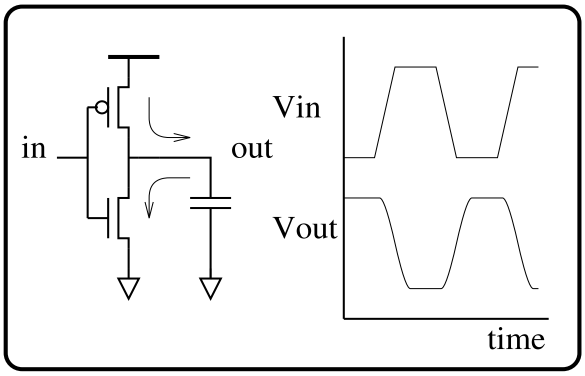

Figure 1: dynamic power dissipation

Thus to design components with low power consumption, we must consider how to reduce the values of V_{dd}, C, A and I_{sc}.

REDUCE V_{dd}

The first strategy is to reduce V_{dd} and this seems the best place to start since it appears as a squared term in equation 1 (although the relationship is complicated by the fact that A is also voltage dependent). Clearly the design engineer does not normally have control over V_{dd}. However, developments in fabrication are already moving from the existing standard of 5V towards a new level of 3.3V and experimental processes are looking at even lower voltages. It is worth considering the issues in this development[5].

Power supply reduction

One of the main motivations in technology development has been to increase the levels of integration by reducing feature sizes. However, as gate lengths are reduced (without reducing voltage levels) the electric field strength increases in the gate region. This leads to reliability problems as the high electric field strengths accelerate the conducting electrons to such speeds that they cause substrate current (by dislodging holes on impact in the drain area) and actually penetrate the gate oxide. The latter effect gradually alters the characteristics of the device leading eventually to latch-up and so to destruction. There are three approaches to enabling further feature size reduction. The first is drain engineering in which the doping profile is crafted in the channel region to reduce the degradation due to hot-electrons; the lightly doped drain (LDD) technique allows the smallest gate length[6]. The second approach is to use new circuit techniques which avoid the high electric fields across individual transistors. The third approach is to reduce the supply voltage; this solution is much the simplest for circuit designers but acceptance has been delayed as the industry wished to maintain compatibility with existing products.

The reduction in V_{dd} does not lead to a quadratic reduction in power as might be thought from equation 1 since some the other terms are dependent upon the supply voltage. To understand the actual effect, consider the activity level of each gate (A). This can be re-expressed as the product of the frequency (f) with which new inputs are presented to a whole circuit (for synchronous circuits, the clocking frequency) and a probability for each node (pr_i) that it will change on any given cycle. The maximum possible frequency of a circuit ($f_{\max}$) represents the fastest throughput of data and this is limited by its critical path or longest delay; thus $f_{\max}$ is inversely proportional to circuit delay. This brings us to a common measure of circuit quality: the power-delay product. By re-arranging equation 1 we have:

$$power\ .\ delay = \frac{P}{f_{\max}} = V_{dd}^2\ .\ \sum_i (\ C_i\ .\ pr_i) \quad (3)$$

Thus variation in V_{dd} actually leads to a quadratic change in the power-delay product.

Variation of the threshold voltage

From the standard transistor current equations, the speed of a circuit is a function of $(V_{gs} - V_T)$ where V_{gs} is the gate-source voltage (limited by V_{dd}) and V_T is the threshold voltage. Thus it is also desirable to reduce the magnitude of the threshold voltages[7] either to minimise the reduction in speed or to allow further reduction in V_{dd}.

There are other reasons for reducing the threshold voltages. Rather than thinking about reducing one parameter at a time it is instructive to consider how to improve a technology as a whole. The work of Dennard[8] in 1974 promised that if the voltage levels (power and threshold) were scaled by the same amount as feature sizes then delay would be reduced by the same factor and power consumption (for the same circuit) by its square. This principle avoids the high electric fields which lead to the hot-electron effect because the voltage levels are reduced also; it is known as *constant electric-field scaling*. For example, a circuit designed in a technology of $V_{dd} = 5V$, $|V_T| = 1V$, and gate length $= 1\mu$ could be re-implemented in one of $V_{dd} = 2.5V$, $|V_T| = 0.5V$, and gate length = 0.5μ with twice the speed and a quarter the area and power consumption.

Of course, all good things come to an end. One feature which does not scale is the roll-off rate of sub-threshold current as the following explains. In the weak inversion region (where V_{gs} is below V_T) there is no *drift* current; however, there is *diffusion* current which has the form:

$$I_{ds(sub-threshold)} \propto exp(V_{gs} - V_X) \quad (4)$$

where V_X is the lowest gate voltage for the weak inversion region. The important point is that this exponential roll-off rate is not effected by voltage scaling. For silicon it is in the region of 70-90mV per decade of current. Figure 2 shows how *threshold voltage* can be defined as the intersection between the linear current and the axis, and how the sub-threshold current rolls-off. If the threshold voltage is reduced (achieved by changing the substrate and channel dopant concentrations) then the whole curve moves towards the left. Thus for low threshold voltages, the device cannot be properly switched OFF (when $V_{gs} = 0V$) and there is significant short-circuit current.

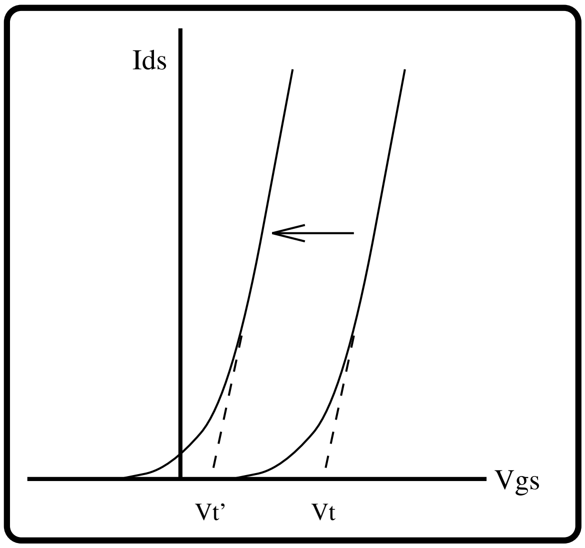

Figure 2: threshold voltage roll-off

As an example, a minimum sized gate for a "typical" 1.5μ CMOS process with $V_{dd} = 3V$ has $I_{ds} = 30\mu A$ at $V_{gs} = V_T$ with a roll-off rate of about 80mV/dec. Thus if the threshold voltage was set at 0.15V, a component of one million such devices would have a power consumption of about 1W due to subthreshold currents alone. The implication is that the threshold voltage must be kept high to prevent significant power consumption due to sub-threshold currents at $V_{gs} = 0V$ and this imposes a practical minimum of about half a volt[8].

An alternative perspective is that by reducing V_T to lower levels it would be possible to reduce V_{dd} even further. Thus it may be possible to tolerate some short-circuit current because of the resulting reduction in dynamic power consumption. This has yet to be demonstrated.

Optimal power voltage

The hot-electron effect establishes an upper limit on power supply voltage due to reliability criteria but, as suggested above, the low-power designer would prefer a lower limit. Here are two suggestions to be applied to any given technology.

To optimise the power-delay product, it has been found that the optimal power supply voltage for a given technology is three times its threshold voltage[3, 9]. This seems intuitively reasonable in that it allows one threshold for each device type and one extra for noise margin. To avoid significant sub-threshold current, the minimum V_T should be at least 0.5V, giving an optimal power supply voltage of 1.5V.

A second approach considers a phenomenon, known as *velocity saturation*, in which the velocity of the charge carriers reaches a maximum with increasing electric field strengths. In other words, an increase in voltage does not increase the current and so does not improve the device speed. This then sets a limit above which it is unproductive to raise the supply voltage and this limit depends upon the fabrication technology and the effective channel length[5].

Compensating for lower speed

As the industry moves from the standard 5V processes to ones with lower supply voltages, design engineers need to compensate for the loss of performance if they wish to achieve the same throughput. There are two architectural approaches: first, apply the standard speed optimisation techniques only more so; second, use parallelism.

Pipelining is a standard technique for increasing the overall speed of a circuit by introducing clocked latches into sequences of combinatorial logic so that the data flow is staggard and controlled by a clock signal. Data may then be processed at a frequency (f) which is the inverse of the longest delay between any two adjacent latches. A designer will typically insert as many latches as are necessary to make the critical-path delay (T) low enough to allow the desired frequency. If the target supply voltage is then lowered, then the circuit speed will be reduced and the critical-path delay will be increased. To compensate for this, a designer would then have to insert or redistribute latches so that the desired frequency is again attained. Figure 3 illustrates how inserting extra latches in the midst of a delay path in a circuit going at half its original speed allows a designer to maintain throughput. Of course this will only be possible if the original circuit had not already not "engineered" to peak performance.

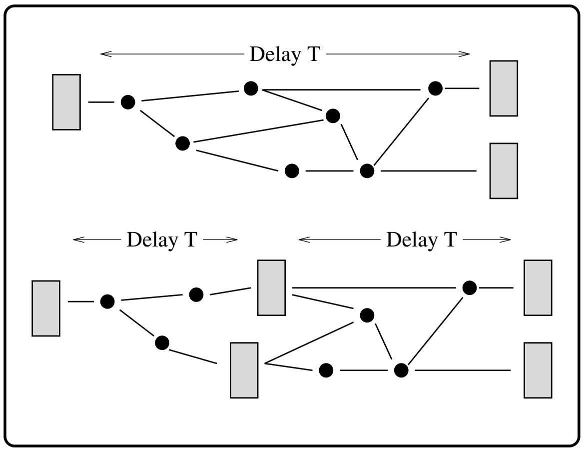

Figure 3: pipelining

The idea of using parallelism is simply to have more operations being conducted at the slower speed to achieve the same overall performance. This is essentially a trade-off between circuit area and throughput. The use of parallelism is illustrated in figure 4. Here we assume that the critical path delay (T) through the combinatorial logic block has (nearly) doubled due to a reduction in the power supply voltage. To achieve the same throughput, the data is interleaved so that new data is presented to one block while the previous data is still being processed by the other. The outputs of the two blocks are selected by a multiplexor so that the valid data is latched at the original frequency. Notice that although the total capacitance of the circuit has been (approximately) doubled, the term A (in equation 1) has been halved because of the speed reduction: these two effects compensate for each other in the dynamic power equation.

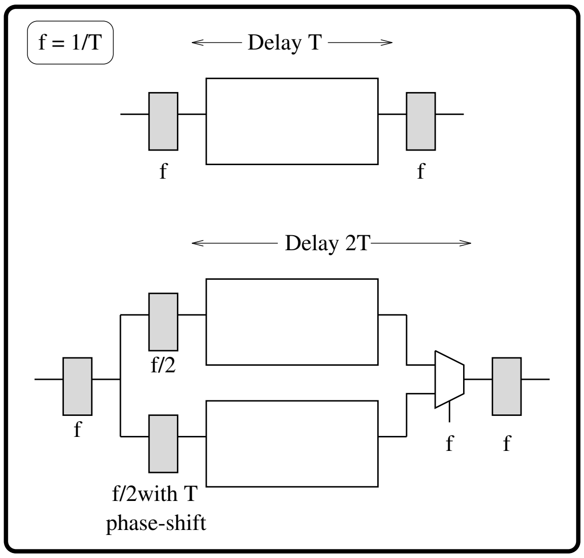

Figure 4: parallelism

Of course, this strategy may sound attractive in the context of rapidly increasing levels of integration, but in terms of commercial viability it must be remembered that doubling the circuit area can have a large impact upon component cost. While many design specifications may demand this approach for the resulting speed, many will also preclude it on the grounds of cost.

In both of these example, the design has been modified to compensate for a halving in circuit speed resulting from a reduction in power supply. To illustrate with a very rough calculation, assume that this is done with a standard 5V process without changing any other fabrication parameters. If speed is taken to be proportional to $(V_{dd} - V_T)$ and $V_T = 1V$, then a halving of speed is achieved by reducing V_{dd} from $5 \rightarrow 3$V. Ignoring the relatively small changes in $\sum_i C_i A_i$ due to the extra circuitry, the dynamic power consumption is reduced to $(3/5)^2 = 0.36$ its original value. If the threshold voltage could also have been reduced to say 0.5V, then the supply voltage could be reduced to 2.5V and the dynamic power to only 0.25 of its original value.

Voltage swing

A final way of reducing power loss connected with the supply voltage is to re-examine equation 1. The second V_{dd} term actually refers to the voltage swing of the internal nodes. If this were reduced, then the total power consumption would also be reduced. One example of this concerns an internal bus architecture[10] which is designed for operation at about 2V with an internally generated supply for the bus itself. Modified thresholds, and special driving and sensing circuitry, allow the bus to swing less that 1V. This not only saves power in itself, but also increases the bus speed making operation at 2V more attractive.

REDUCE *C*

The second strategy is to reduce capacitance. This comes naturally with smaller feature sizes and so a circuit designer will generally wish to use the minimum geometries possible in the given technology.

Partition blocks

As a general rule, it is best to partition large blocks into smaller ones. The design on the left in Figure 5 is a large memory block: the shaded area is the address generation and bit detection circuitry, and the unshaded region is the memory array. The power calculation for each memory access is based upon the capacitances of the bit and word lines which run vertically and horizontally across the whole array. If, instead, the array is broken down into four sub-units (each with its own support circuitry) and only one unit is addressed with each access, then the product of activity and capacitance is reduce by a half.

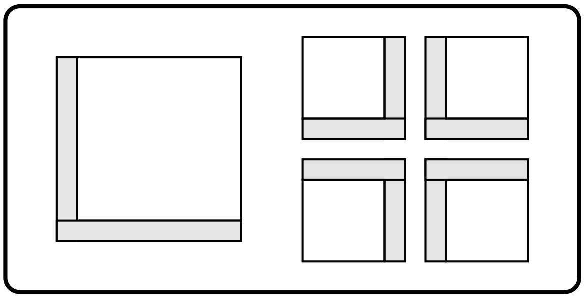

Figure 5: block partioning

Locality of reference

There is another architectural strategy which can significantly reduce the capacitance (or more specifically $\sum_i C_i A_i$) of a design; it can be summarised in the phrase *locality of reference*. This is a design philosophy in which signals are generated and used locally in terms of their physical location on the silicon surface since the further a signal has to travel, the higher is the capacitance of that connection. With signals being processed locally, there is greater opportunity for parallel execution. With parallel execution, there is greater throughput which could be traded-off for a lower supply voltage and so lower power consumption.

Designing with locality of reference is desirable for another reason related to the new fabrication technologies. Communication within a component is achieved using metal interconnect. For large feature sizes, the RC delay on such lines is relatively small compared with the transistor delays in the circuit. However, while transistor delays scale down with feature size, the RC delays actually increase as fringing capacitance begins to dominate the total capacitance (and does not scale) and the resistance increases as the interconnect lines become narrower. Thus, in sub-micron technologies, the communication delays predominate[11]. Thus using locality of reference as a design style avoids the major potential source of delay.

Architectural strategies based upon this idea may include: processing of data locally to where it is stored, communication only with physically adjacent functional units, and dedication point-to-point buses rather than shared ones. Figure 6 illustrates this idea. The architecture on the left consists of a large number of units connected by a global shared bus; this is not uncommon. Consider the communications bus alone. The architecture on the right has fourteen much smaller buses which together have roughly the same total capacitance as the single bus on the left. If only one was active, then the activity-capacitance product would be 1/14th; if they were all active, then the power consumption would be the same but up to 14 times as much information could be transferred.

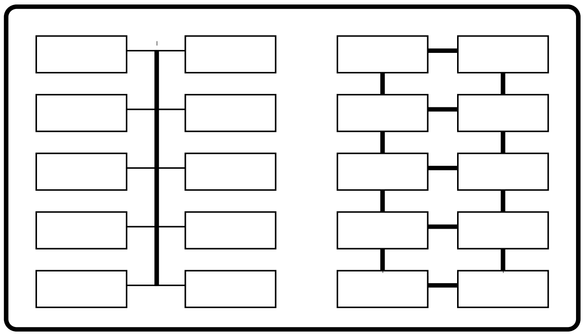

Figure 6: locality of reference

It is clear that low-power systems will include the use of dedicated function-specific circuits which transform particular sections of the total algorithm onto localized areas of silicon – they will not have central processing units communicating by global buses with large memories.

Clocks and control

In architectures with distributed processing, the question arises as to whether there should be global control and clock signals. On the one hand, there needs to be synchronisation between communicating pairs of processors; on the other hand, the global distribution network has a very large capacitance and is switched frequently. There are several possible strategies.

A new latch circuit was introduced in 1987 which allows *true single phase clocking*[12, 13] (TSPC). This implies that only one clock signal needs to be distributed where-as previously designs had relied upon having at least the complement of the clock available (either distributed in parallel or generated locally). Thus by using TSPC, a design can greatly reduce the capacitance of its largest network. TSPC has already been used to implement extremely fast and power-hungry designs (e.g. the DEC Alpha) but, as we have seen above, the speed advantage could be traded-off against power by designing for lower supply voltages.

If the problem is the widely distributed clock signal, then one solution is not to distribute it so widely. In this approach, regions of the component which are not being used have their portion of the clock network gated off. The draw back to this scheme is the need to generate and distribute the clock control signals and the added design complexity in providing a synchronous clock signal on a network which is partitioned by function rather than by equal capacitance. A variation of this approach is to disable sections of the power (rather than the clock) distribution network.

Self-timed circuits dispense with a global clock altogether. In this scheme, units independently store their data, process it and generate a "ready" signal. Communication between units is achieved through a hand-shaking protocol where the next unit will receive the "ready" signal from the previous one, take the data into its local storage and then generate an "acknowledge" signal. The first unit is then free to take in new data itself, if that data is available. In this manner, data is processed by local units and passed throughout the component without a global clock. While this seems to remove the high capacitance-activity of the global clock line, in practice the power dissipated in the ready and acknowledge signals can be of the same order. Low-power designers should not assume that self-timed logic is necessarily preferable to clocked logic.

Logic design

Another approach is to use logic families which feature low capacitance. One promising family is the *complementary pass-transistor logic*[14] (CPL). This uses networks of purely n-type pass-transistors to form logic functions (without any p-types). All signals are generated in complementary values and the outputs from the logic functions drive CMOS inverters. Figure 7 illustrates the CPL implementation of the *sum* function using 12 transistors instead of the 22 needed for a conventional implementation. The power advantage comes mainly from the reduced number of gates and so the lower capacitance.

This technique has been successfully applied to a 4V, 0.5μ CMOS process to implement a 16x16-bit multiplier. However, there are features with this technique of which a circuit designer should be aware. Due to the threshold voltage drop in the pass transistor network, the output high logic level is V_{Tn} below the power supply. This means that the p-type transistors in the inverters are only switched off by ($|V_{Tp}| - |V_{Tn}|$) leading to sub-threshold currents as described in the previous section. This can be reduced by using cross-coupled p-type pull-up transistors on the complementary logic outputs (leading to increased transistor count and reduced speed) or by using a special fabrication technology with a lower threshold voltage for the pass-transistor n-types only (0V was used in 16x16-b multiplier). Thus the best results using this circuit technique for low-power depend upon also matching the fabrication process to it.

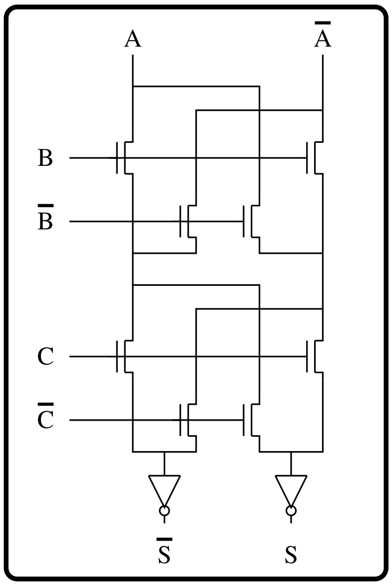

Figure 7: sum function in CPL

Two further design styles related to CPL have also been reported. One overcomes the problem of the threshold voltage drop by using full CMOS pass transistors[15]. This still has better speed performance than conventional CMOS and so would achieve a given throughput at a lower voltage (and so power dissipation). The second related design style uses threshold adjustment on the p- as well as the n-type transistors[16].

Buffer design

One recurrent problem is the design of circuitry to drive a relatively large capacitance (particularly external loads). The basic solution is a sequence of buffers with increasing gate widths; the design issue is what should be the size ratio (α) of each successive buffer.

With speed as the main consideration, the classical answer[17] is $\alpha = e$. With intrinsic output capacitance of the CMOS buffer included this value is known to be layout dependent[18] and in the region of 5-6. However, if power is the main issue and the overhead in charging and discharging intermediate nodes is considered, the optimal ratio is layout and process dependent[19] and is about 11-12. The following table shows the example of different ratios for a buffer chain driving an 11pF load from an initial buffer with input capacitance of 0.1pF; "useful power" is that expended in charging the 11pF capacitance itself, and "other power" is that expended on intermediate nodes.

ratio (α)	e	11.5
# inverters	5	2
useful power	2.5mW	2.5mW
other power	5.4mW	1.5mW
total power	8.1mW	4.0mW
delay	5.5nS	6.5nS

Thus there is over a 50% reduction in power dissipation, and a similar reduction in layout area, at the price of only an 18% increase in delay.

REDUCE *A*

The third strategy is to reduce *A*: the average activity on each gate. Power is only expended when a node is switched; if switching is to restricted to when information changes then power is minimised. This can be summarized by the phrase *transition avoidance*. As a first observation, this argues against the use of circuit styles which involve precharging and discharging as part of logic evaluation.

Glitch avoidance

With some digital logic, there are spurious transitions (known as *glitches*) which occur due to partially resolved functions; figure 8 shows an example. If there is a unit delay through both of these gates then when the inputs both change from 1 to 0, the output will change to a 1 as the logic is resolving before returning to a a final value of 0. This wastes power. The problem is reduced in general by designing circuits so that there are equal delay paths between all of the gate inputs and the system inputs, thus equalising arrival times of changing signals. Of course, this is hard to achieve in practice and impossible if there is feedback in the circuit.

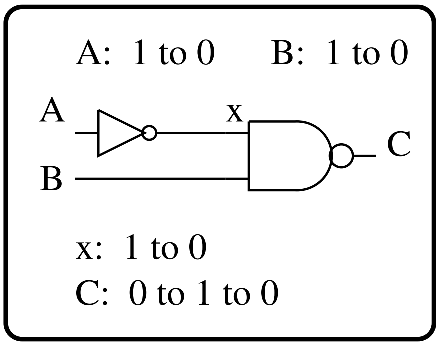

Figure 8: glitches

A more important example of power loss through spurious transitions is the ripple adder. In this logic design, each bit-adder unit passes its carry to the next unit in a carry-chain; the value of its own input is not, however, valid until all the less significant bits have been resolved; thus each carry-bit in the chain may change (along with the corresponding sum outputs) as the valid carry signal propagates along the chain. To avoid the associated loss of power, a different adder design should be used.

Point-to-point buses

This concept of transition avoidance can be viewed at the architectural level also. Suppose there are two independent slowly-varying digital signals within a component. If these are distributed on independent data buses, then transitions only occur when information changes. If, instead, the two signals are combined by a multiplexor onto a single bus for distribution, there is also likely to be a transition when the multiplexor is switched (i.e. when the control signal changes). Although point-to-point buses incur an area cost due to the extra interconnect routing, they save significant power by avoiding transitions which occur when mixing independent signals.

Reviewing the algorithm

The power consumption of a complex system can be greatly influenced at the algorithmic level. Normally, component power consumption corresponds to the usual algorithmic performance criterion of speed since algorithmic speed is a function of the number of operations and this translates onto the component as the amount of switching. Thus the programmer's desire to reduce the number of steps in a computation will naturally reduce the power consumption of its implementation.

The idea of locality of reference may be mapped directly into algorithmic design through the use of certain programming languages. Concurrent object-orientated languages (e.g. ADA and VHDL) allow the creation of software modules which mostly use locally declared memory and which allow communication between modules as an alternative to parameter passing by function call. It could be most effective for low-power system design if the underlying algorithms were developed in such a language from the very beginning.

REDUCE I_{sc}

A designer needs to consider short-circuit current in two ways: first, how to minimise what is unavoidable, second how to avoid what is unnecessary.

Resistive networks

Firstly, some logic styles deliberately use resistive networks formed from transistors to establish the value of the output signal (e.g. pseudo-NMOS). These styles cannot be used for low-power design. Secondly, some strategies for avoiding power loss involve generating multiple voltage levels using resistive networks either on-chip or at the system level. This static power loss must be carefully included in the evaluation of such strategies.

Switching current

However, even conventional static CMOS has a source of short-circuit currents. Consider figure 9. As the input to a CMOS inverter changes, there is a period during which both transistors are switched ON that is when the input voltage is between $(V_{dd} - V_{Tp})$ and V_{Tn}. During this period, there is a short-circuit current and so power dissipation. This is clearly dependent upon the rise time of the input signal. For poorly designed circuits, this power loss can be about 20% of the total power dissipation. A simple rule-of-thumb for designers is to size the transistors so that the delay in the output signal is the same as that of the input; with this strategy, the short-circuit power loss is reduced to 1-2% of the dynamic power dissipation[19].

Glitch propagation

The example concerning spurious transitions in figure 8 above was explained in terms of unit time delays. In fact the problem is compounded in that the output glitch propagates on to other

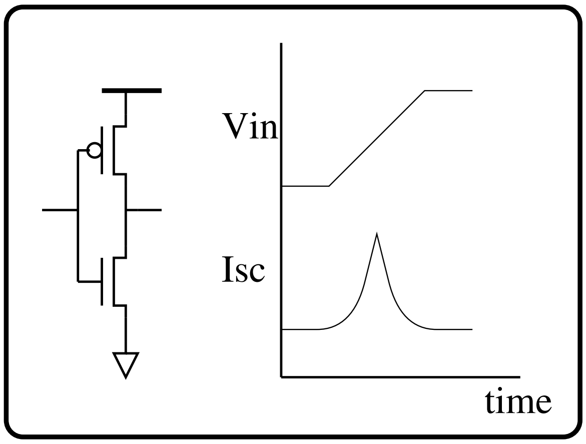

Figure 9: short-circuit current on switching

stages. In practice this signal often takes the form of a slowly varying voltage which hovers in the centre of its range causing short-circuit currents in the next gate. This is another source of power dissipation and a further reason to avoid logic glitches.

APPROACHES TO LOW-POWER DESIGN

The first decision the IC designer needs to make is the choice of fabrication technology. Low supply voltage is good. Small feature size is very good because of the low capacitance and increased device speed due to the short channel length. Fortunately, the first is tending to follow the second because of the hot-electron effect. The simple rule-of-thumb is to use the process with the smallest feature size that the project can afford.

If there is access to a fabrication technology with multiple (and specifiable) threshold voltages, then this might be chosen to support the CPL design style or one of its derivatives.

The next decision is the design partitioning. A designer should avoid architectures based on central processing units, and always review the specified function. The aim should be to partition the function into small independent units (avoiding high capacitance interconnects) operating in parallel (raising throughput). The method is to apply the principle of locality of reference even if this means returning to the algorithmic development level.

The final decision must be to re-evaluate the standard techniques for logic and circuit design. Designing for low power means that many of the old stand-bys are redundant and that new approaches must be developed. It requires a strong but subtle change in emphasis.

For instance, in one sense the designer must abandon the imperative for speed which in itself leads to high power consumption. On the other hand, if a reduced voltage level is the main mechanism for power reduction, then all the old tricks for enhancing speed may be needed to compensate for the reduced drive capability.

A second change in emphasis is that area is no longer as limited a resource as it used to be. Thus, with low power as the main criterion, techniques which require extra area are not unattractive. In particular, resource sharing is less important particularly when it leads to additional switching.

The fascinating opportunity is that since power has become the main design cost, the designer can now explore radical options in algorithmic, architectural, logic and circuit design. The challenge is here, the fun is just beginning.

Dr Gerard M Blair is a lecturer in VLSI Design in The Department of Electrical Engineering, The University of Edinburgh, The King's Buildings, Edinburgh EN9 3JL; email: gerard@ee.ed.ac.uk

References

1. Minoru Nagata, "Limitations, Innovations, and Challenges of Circuits and Devices into a Half Micrometer and Beyond", *IEEE Journal of Solid-State Circuits*, vol. 27, no. 4, pp. 465-472, April 1992.
2. Anathna P Chandrakasan, Samuel Sheng, and Robert W Brodersen, "Low-Power CMOS Digital Design", *IEEE Journal of Solid-State Circuits*, vol. 27, no. 4, pp. 473-484, April 1992.
3. Katsuhiro Shimohigashi and Koichi Seki, "Low-Voltage ULSI Design", *IEEE Journal of Solid-State Circuits*, vol. SC-28, no. 4, pp. 408-413, April 1993.
4. Richard F Lyon, "Cost, Power amd Parallelism in Speech Signal Processing", *IEEE Custom Integrated Circuits Conference*, pp. 15.1.1-15.1.9, 1993.
5. Masakazu Kakumu and Masaaki Kinugawa, "Power-Supply Voltage Impact on Circuit Performance for Half and Lower Submicron CMOS LSI", *IEEE Transactions on Electron Devices*, vol. 37, no. 8, pp. 1902-1908, Aug 1990.
6. Hiroaki Mikoshiba, Tadahiko Horiuchi, and Kuniyuki Hamano, "Comparison of Drain Structures in n-Channel MOSFETs", *IEEE Transactions on Electron Devices*, vol. ED-33, no. 1, pp. 140-144, Jan 1986.
7. Dake Liu and Christer Svensson, "Trading Speed for Low Power by Choice of Supply and Threshold Voltages", *IEEE Journal of Solid-State Circuits*, vol. 28, no. 1, pp. 10-17, Jan 1993.
8. Robert H Dennard, Fritz H Gaensslen, Hwa-Nien Yu, V Leo Rideout, Ernest Bassous, and Andre R LeBlanc, "Design of Ion-Implanted MOSFETs with Very Small Physical Dimensions", *IEEE Journal of Solid-State Circuits*, vol. SC-9, no. 5, pp. 256-267, Oct 1974.
9. J Burr and A Peterson, "Ultra Low Power CMOS Technology", *3rd NASA Symposium on VLSI Design*, pp. 4.2.1-4.2.13, Oct 1991.
10. Yoshinobu Nakagome, Kiyoo Itoh, Masanori Isoda, Kan Takeuchi, and Masakazu Aoki, "Sub-1-V Swing Internal Bus Architecture for Future Low-Power ULSIs", *IEEE Journal of Solid-State Circuits*, vol. 28, no. 4, pp. 414-419, April 1993.
11. Arnold Reisman, "Device, Circuit and Technology Scaling to Micron and Submicron Dimensions", *Proceedings of the IEEE*, vol. 71, no. 5, pp. 550-565, May 1983.
12. Y Ji-ren, I Karlsson, and C Svensson, "A true single phase clock dynamic CMOS circuit technique", *IEEE Journal of Solid-State Circuits*, vol. SC-22, pp. 899-901, 1987.
13. Morteza Afghahi and Christer Svensson, "A Unified Single-Phase Clocking Scheme for VLSI Systems", *IEEE Journal of Solid-State Circuits*, vol. SC-25, pp. 225-233, Feb 1990.
14. Kazuo Yano, Toshiaki Yamanaka, Tashiaki Nishida, Masayoshi Saito, Katsuhiro Shimohigashi, and Akihiro Shimizu, "A 3.8-ns CMOS 16x16-b Multiplier Using Complementary Pass-Transistor Logic", *IEEE Journal of Solid-State Circuits*, vol. 25, no. 2, pp. 388-395, April 1990.

15. Makoto Suzuki, Norio Ohkubo, Toshinobu Shinbo, Toshiaki Yamanaka, Akihiro Shimizu, Katsuro Sasaki, and Yoshinobu Nakagome, ‘‘A 1.5-ns 332-m CMOS ALU in Double Pass-Transistor Logic’’, *IEEE Journal of Solid-State Circuits*, vol. SC-28, no. 11, pp. 1145-1151, Nov 1993.

16. Menahem Lowy, Chi-Yuan Chin, and Jerome J Tiemann, ‘‘Low Power Consumption Communication Systems’’, *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, pp. 807-810, May 1991.

17. Carver Mead and Lynn Conway, *Introduction to VLSI systems,* Addison-Wesley, 1980. ISBN 0-201-04358

18. Nils Hedenstierna and Kjell O Jeppson, ‘‘CMOS Circuit Speed and Buffer Optimization’’, *IEEE Transactions on Computer-Aided Design*, vol. CAD-6, no. 2, pp. 270-281, March 1987.

19. Harry J M Veendrick, ‘‘Short-Circuit Dissipation of Static CMOS Circuitry and its Impact on the Design of Buffer Circuits’’, *IEEE Journal of Solid-State Circuits*, vol. SC-19, no. 4, pp. 468-473, Aug 1984.