High Speed CMOS Circuit Design

Lecture 3: Dynamic Circuits

(c) 1997 David Harris

1.0 Introduction to Dynamic Gates

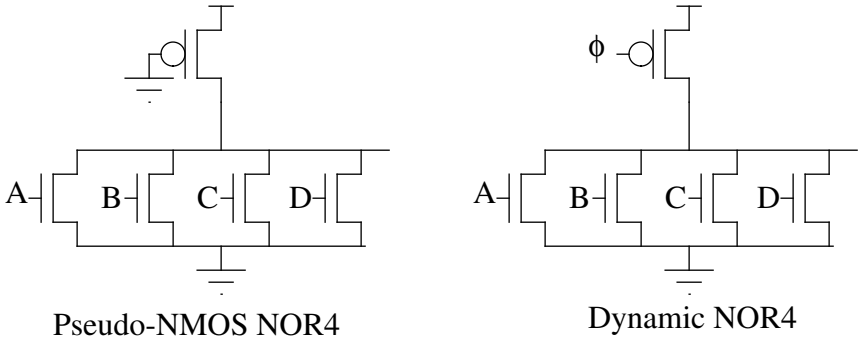
Static logic is great for its robustness. However, it is too slow to meet requirements for high speed processor critical paths. Thus, designers are increasingly turning to dynamic logic in hopes of achieving a 1.5 - 2x speedup over static gates.

The main drawback of static gates is the fact that inputs must drive both NMOS and PMOS transistors. Only one of the two transistors is ever on, meaning that the input capacitance of the other transistor loads the critical path without increasing the current drive of the gate. Moreover, the PMOS transistors must be large and thus add much capacitance. If we could construct gates which only have NMOS transistors in the critical path, circuits would run much faster.

Pseudo-NMOS logic achieves this goal by replacing the PMOS stack with a single grounded PMOS transistor serving as a resistive pullup. Thus, the NMOS pulldowns can be very fast. Unfortunately, the PMOS transistor fights against the NMOS during a falling transition, slowing the fall time. Also, it must be weaker than the NMOS, so the rise time is not very good. Finally, when the output is low, there is a path from VDD to GND wasting power.

An alternative scheme is to connect the PMOS transistor to a clock, instead of ground. This topology is called a dynamic gate. Figure 1 shows the similarity between dynamic logic and pseudo-NMOS gates.

FIGURE 1. Pseudo-NMOS and Dynamic gates

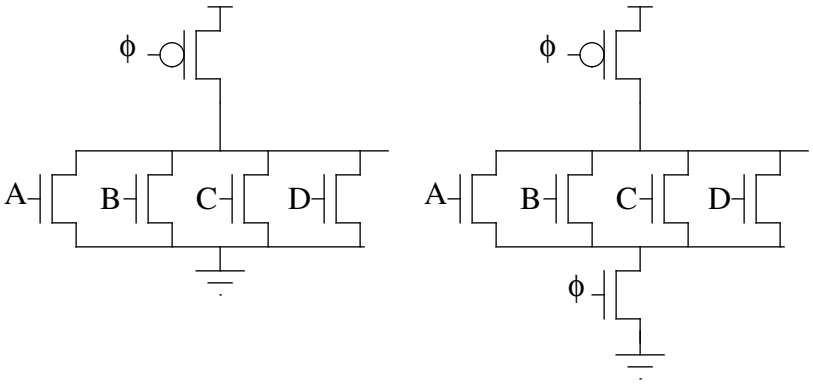


Pseudo-NMOS NOR4 Dynamic NOR4

Dynamic gates operate in two phases: precharge and evaluation. During the precharge phase, the clock is low, turning on the PMOS device and pulling the output high. During evaluation, the clock is high, turning off the PMOS device. The output may “evaluate” low through the NMOS transistor stack.

The **PRECHARGE** rule of dynamic gates states that *there should be no active path to ground during precharge*, so that the PMOS transistor can fully precharge the output high without contention with the NMOS pulldowns. Sometimes this can be achieved by guaranteeing that some inputs are low. For example, with a NOR4 gate, all four inputs must be low. For a NAND4 gate, only one of the series pulldown transistors must be low. It is not always possible to guarantee this condition, so often an extra clocked evaluation transistor is placed at the bottom of the pulldown stack, as shown in Figure 2.

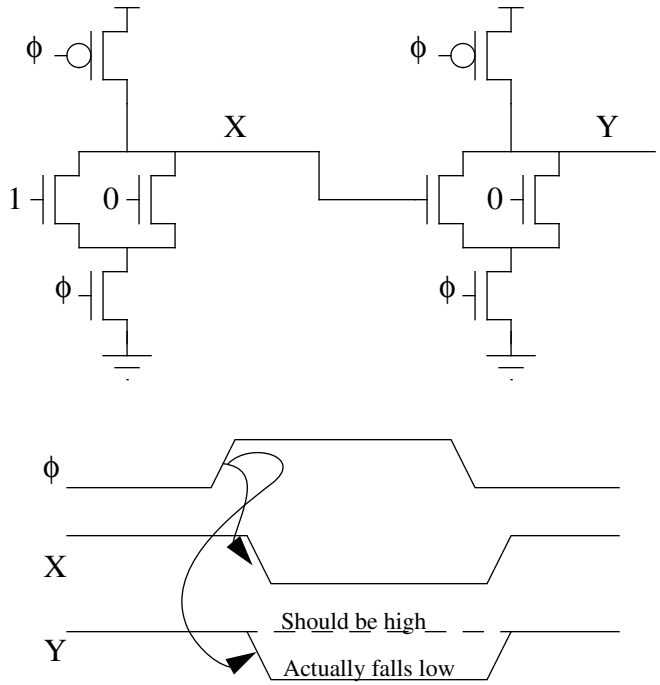
FIGURE 2. Dynamic gates with and without clocked evaluation transistors



No Clocked Evaluation Transistor Clocked Evaluation Transistor

Another limitation of dynamic gates arises when one dynamic gate directly drives the next, as shown in Figure 3. When ϕ is low, both gates precharge high. When ϕ rises, both gates begin evaluating. Since the first gate has a high input, its output X falls low. The second NOR gate should therefore produce a high output. Unfortunately, the second NOR gate initially received the high value at X and evaluates low. By the time X falls, Y may have become corrupted. Since the precharge transistors are off, Y has no way of recovering to the correct high value during the evaluation phase. The circuit produces an incorrect result.

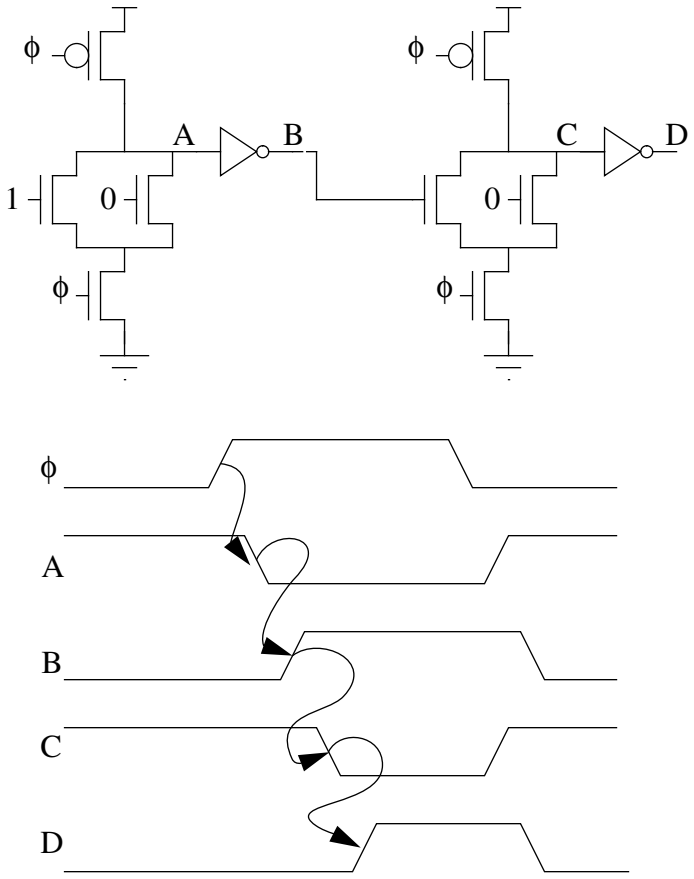
FIGURE 3. Incorrect operation of cascaded dynamic gates



The problem arose because the second dynamic gate had an input switch from high to low while the gate was evaluating. The output had pulled low while the input was high, then cannot recover to a correct high value. To avoid this problem, dynamic gates must obey the **MONOTONICITY** rule: *all inputs to dynamic gates should make only low to high transitions while the gates are evaluating*. Inputs must be “monotonically rising,” meaning they can stay low, stay high, or may rise, but may not fall.

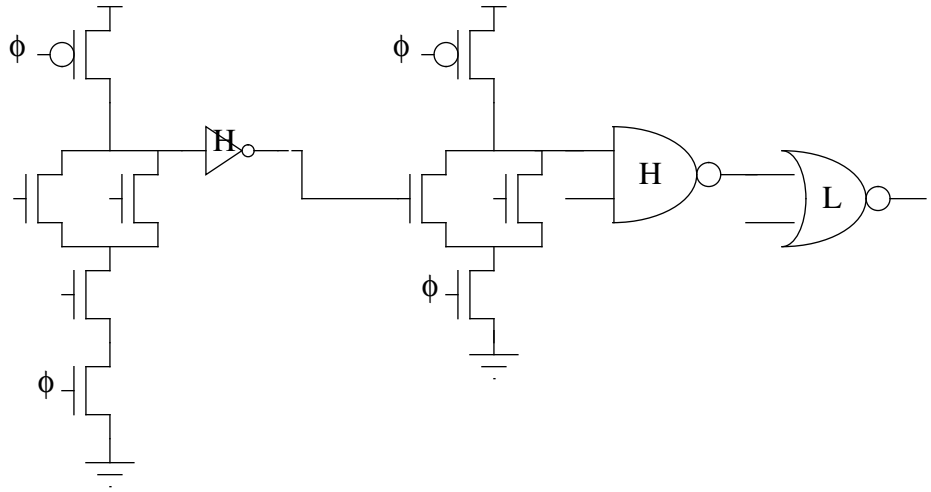
An easy way to achieve this condition is to insert an inverting static gate between each dynamic gate, as shown in Figure 4. Now the gate operates correctly. The dynamic/static gate pair is called a domino gate. All domino gates in a cycle are precharged simultaneously, like dominos being set up, then one may trigger the next which triggers the next, like a chain of dominos falling.

FIGURE 4. Correct operation with domino gates



In general, any inverting static gate, not just an inverter, may be used between dynamic gates. Domino gates precharge simultaneously, but evaluate sequentially. Therefore, the evaluation transition is generally more critical than the precharge transition. For best speed, the static gate should be skewed high to favor its critical rising output. If the static gate then drives another static gate instead of a dynamic gate, the next static gate can be skewed low, as shown in Figure 5. However, if too many static gates are included in series, precharge may become critical.

FIGURE 5. Dynamic and skewed static gates

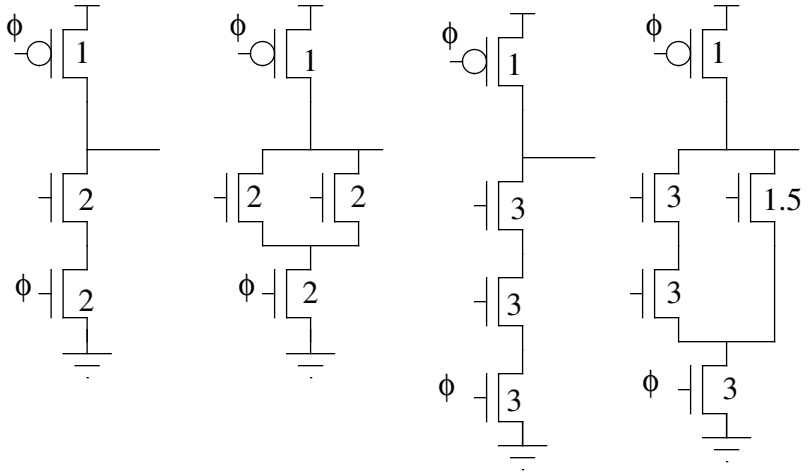


2.0 Logical Effort of Dynamic Gates

The logical effort of a dynamic gate is the ratio of its input capacitance to a static inverter with the same pulldown resistance. Since precharge is assumed to be less critical, the size of the precharge transistor does not impact logical effort. Generally, sizing the precharge device to be half the strength of the pulldown path gives sufficiently fast precharge while keeping the clock loading and diffusion parasitics of the precharge device reasonably low. A few examples of logical effort of dynamic gates is shown in Figure 1.

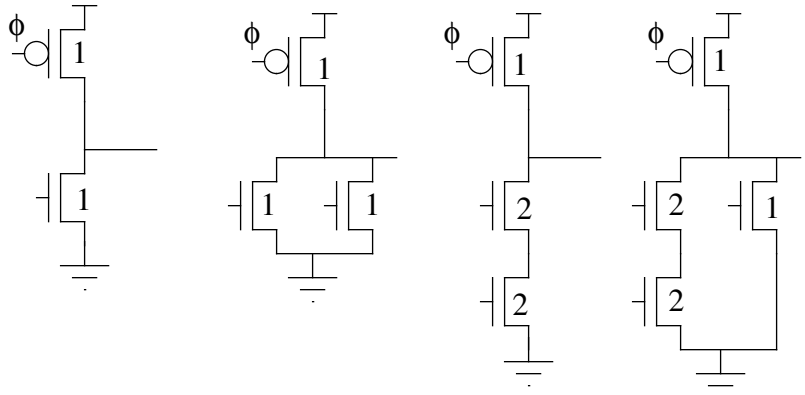
FIGURE 6. Logical effort of dynamic gates

Dynamic gates with clocked pulldowns



INV: LE=2/3 NOR: LE=2/3 NAND: LE=1 AOI: LE=1 or 1/2

Dynamic gates with no clocked pulldowns



INV: LE=1/3 NOR: LE=1/3 NAND: LE=2/3 AOI: LE=2/3 or 1/3

Removing the clocked pulldown transistor reduces logical effort. When the clocked pulldown is necessary, some designers double its width to reduce the effective resistance at the expense of larger clock loading.

Designing with dynamic gates has an interesting effect on the optimal gain per stage. Remember that the calculation of optimal gain per stage driving heavy loads were based on optimal fanout per stage of about 4. Since the buffer element in static circuits is an inverter, the logical effort is 1 and the optimal gain per stage equals the optimal fanout per stage. Dynamic circuits can buffer heavy loads with a domino buffer built from a dynamic inverter followed by a high skew static inverter. These gates have lower logical effort. Therefore, paths built from dynamic circuits should be expected to have a lower optimal gain per stage for quickly driving heavy loads.

Consider a buffer chain built from dynamic inverters alternating with high-skewed inverters. To provide a fanout of 4 per stage, the domino gate (dynamic + static gate) must have

a fanout of 16. The static inverter has a logical effort of 5/6. The dynamic gate has a logical effort of 2/3 with a clocked pulldown or 1/3 without. Thus, the total gain of the two stages is 16*5/6*(2/3 or 1/3) = 8.9 or 4.5, corresponding to an optimal gain per stage of about 3 when dynamic gates have clocked pulldowns or 2.1 with no clocked pulldown.

The conclusion of this analysis is that paths with dynamic gates should be sized with a gain of about 3 per stage instead of 4 per stage used for static paths.

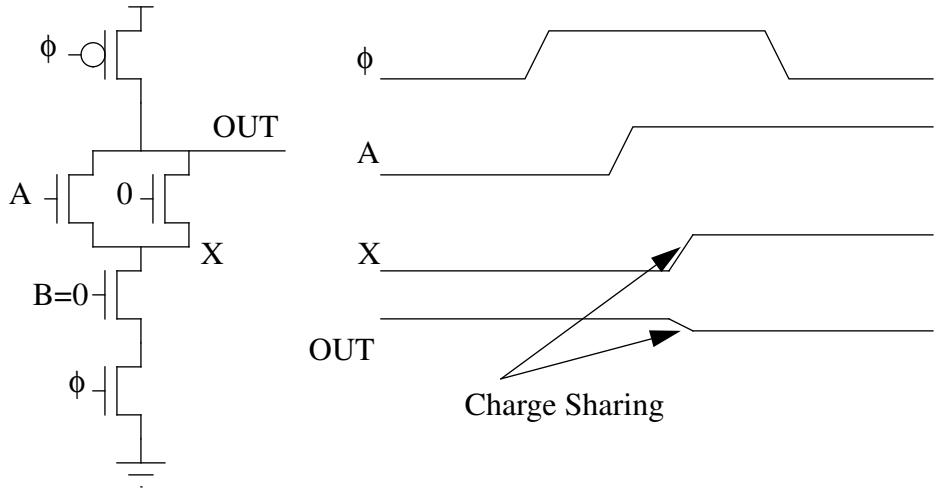
3.0 Domino Robustness Rules

Domino gates which obey the precharge and monotonicity rules will be logically correct. However, domino circuits are much more sensitive to noise than static circuits and therefore additional electrical rules are requires to ensure correct operation. If static gates receive too much noise, they can glitch, but eventually settle to the correct output. If dynamic gates receive too much noise, they can incorrectly evaluate and never recover.

3.1 Charge Sharing

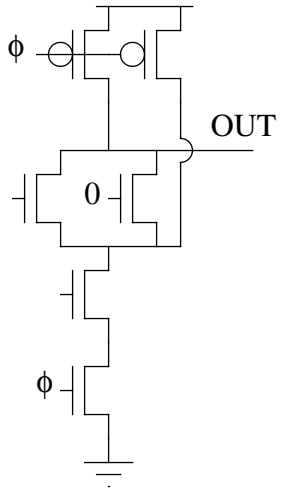
Charge sharing is one important dynamic gate failure mode. When a dynamic gate drives a small load, the internal diffusion capacitances may become comparable to the load capacitance. If the diffusion capacitances are low when evaluation begins, they may share charge with the load capacitance, causing the output voltage to droop from the capacitive voltage divider. For example, consider the dynamic gate in Figure 6. Suppose node X is initially low, perhaps from operation in the previous cycle. Suppose after evaluation begins, input A rises, but input B stays low. The capacitance on node X will share charge with the output capacitance, resulting in a dip on the output voltage. If the ratio of charge is too large, the output will fall by more than the noise margin of the next gate and produce an incorrect result.

FIGURE 7. Charge sharing



There are several ways to keep charge sharing to acceptable levels. One is to precharge internal diffusion nodes so that they will not disturb the output. This can be done with small “secondary” precharge transistors connected to the internal nodes, as shown in Figure 7. Each precharge device adds extra diffusion capacitance which slightly slows the gate, so it is best to only precharge the smallest number of internal nodes necessary to satisfy noise margins. A rule of thumb is that precharging every other node is sufficient for most gates.

FIGURE 8. Secondary precharge transistor



Another way to limit charge sharing noise is to make the load capacitance large, i.e. to have the dynamic gate drive a large fanout. Some dynamic libraries set minimum fanouts for the dynamic gates to limit charge sharing.

Of course, only certain dynamic gates are prone to charge sharing. A dynamic NOR gate can never experience charge sharing because it has no internal nodes. Conversely, a complex AND-OR-invert structure is most susceptible because it has large amounts of internal diffusion.

3.2 Coupling

A second common problem with dynamic circuits is coupling noise. The output node of a dynamic gate is sensitive to coupling because there is no active transistor holding the output high to fight against downward coupling from adjacent switching lines. Moreover, since the receiver is a high skew gate, the noise margin is lower. The input of a dynamic gate is also sensitive because although there is an active gate driving the line, the noise margin is only V_t and a small amount of coupling is sufficient to turn on the dynamic gate and cause it to incorrectly evaluate low.

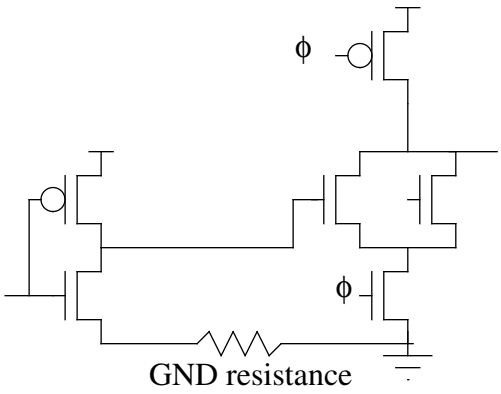
Every dynamic input and output therefore has a budget for coupling noise. To keep this coupling low, the designer can use a variety of tactics. If it is possible, a simple solution is to make adjacent lines only switch while the dynamic gate is in precharge and hence insensitive to noise. Keeping the wires short keeps the ratio of coupling capacitance to

load capacitance small and reduces noise problems. Increasing the spacing between wires is also helpful.

3.3 Power Supply Noise

Another source of noise in dynamic gates comes from power supply variation across the die. Suppose a static inverter drives a dynamic input and that the ground potential of the inverter is higher than that at the dynamic gate due to IR drops across the ground network. If the ground difference exceeds Vt, the dynamic gate will turn on and incorrectly evaluate.

FIGURE 9. Power supply noise between static and dynamic gates



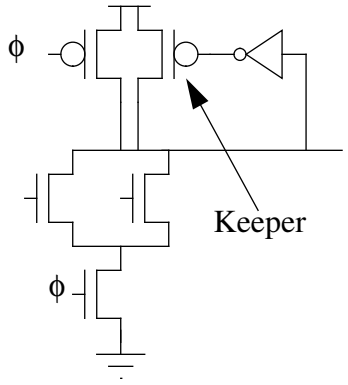
3.4 Subthreshold Leakage

When a dynamic gate is in evaluation with the inputs low and the output high, the output remains high due to charge stored on the load capacitance. All transistors leak slightly due to subthreshold conduction; such leakage may gradually drain charge off the load capacitance and slowly pull the output low. While this is not a problem at normal operating frequencies because evaluation will end long before much leakage occurs, it does present a problem for chips with clocks that may stop. Such cases arise if the clocks are stopped for scan, when a portion of the chip like the floating point unit is shut down to save power, or when the entire chip is placed in a sleep mode. It is worst for gates with wide NOR structures with many parallel leakage paths. It is also a growing problem because threshold voltages are scaling.

To counter leakage on nodes that may float high for extended periods, a dynamic gate may use a weak PMOS transistor called a keeper, as shown in Figure 9. The keeper should be sized as small as possible because when the dynamic output begins switching low, the keeper fights the NMOS evaluation transistors. The keeper should be at least a factor of 4-10 weaker than the pulldown transistors; such keepers typically increase the delay of the gate by 4-6%. After the dynamic output falls far enough, the inverter rises and turns off the keeper to avoid unnecessary power dissipation. If the static gate following a dynamic

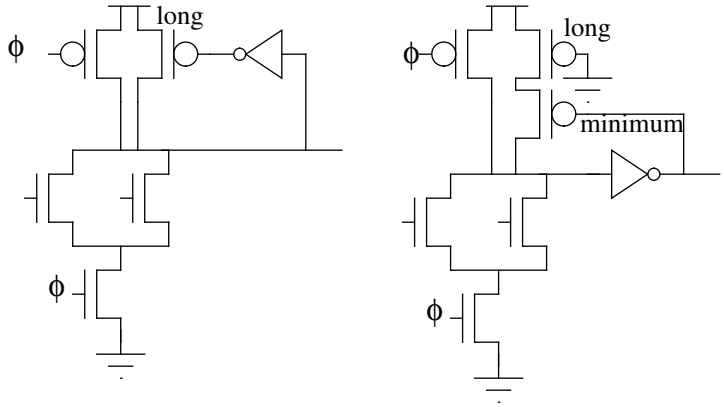
gate is an inverter, it may be used to control the keeper; otherwise an additional small inverter is necessary.

FIGURE 10. Keeper to counter subthreshold leakage



This keeper arrangement works well for dynamic gates with medium or large transistors. For dynamic gates with small transistors, even a minimum width PMOS transistor is stronger than desired for the keeper. Thus, the channel length of the PMOS transistor may have to be increased. However, if the keeper is driven by the same inverter which is driving the next stage, the loading of the long-channel keeper slows the inverter. To prevent this the keeper can be broken into two transistors, one being a long channel device that is always on and the other being a minimum size device controlled by the inverter. These variations are shown in Figure 10.

FIGURE 11. Keepers for small dynamic gates



Long-channel keeper Split keeper

Some designers use keepers to reduce coupling noise or charge sharing. Unfortunately, small keepers are too weak to help very much in these cases. Such noise occurs rapidly, causing sudden dips in the output voltage. The keeper will slowly pull the output back to the correct high value, but by the time a weak keeper can help very much, the next stage has probably been tripped and will send a glitch through subsequent stages of logic. Therefore, do not expect much noise margin benefit from weak keepers.

In very noisy environments, it may be worthwhile to increase the size of the keeper substantially. This offers the possibility of trading off lower speed for better noise margins. For example, if a dynamic gate suffers noise on its input, a moderate sized keeper is sufficient to hold the output high when the NMOS devices have barely turned on due to noise.

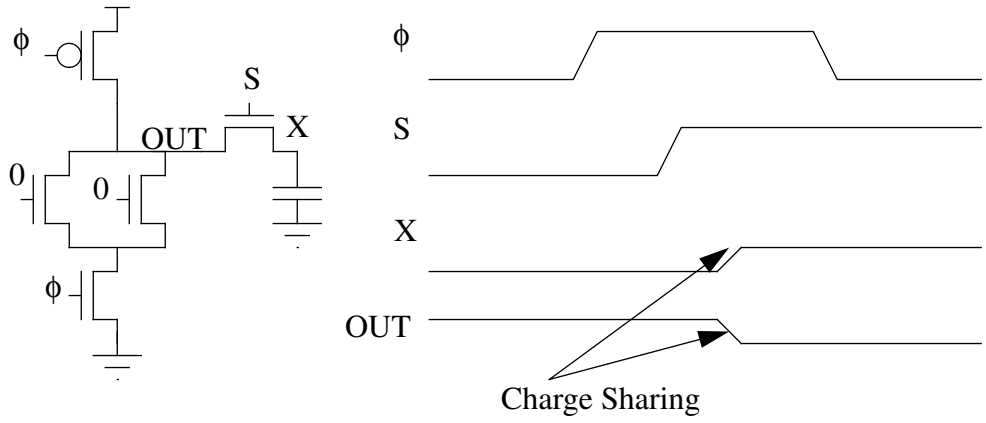
DEC's design methodology for the Alpha 21164 sets a minimum operating frequency of 1/10 the nominal frequency. At this minimum frequency, leakage is only an issue on wide NOR gates; therefore, keepers are only used on wide NORs.

3.5 Pass Transistor / Dynamic Gate interactions

Combining dynamic gates with pass transistors leads to a number of electrical problems. Backdriving and charge sharing are the major concerns.

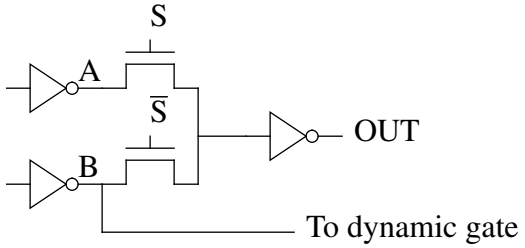
Charge sharing can occur if a dynamic gate drives the diffusion input of a pass transistor, as shown in Figure 11. Suppose the pass transistor is off during precharge and that the far side of the pass transistor is at zero volts. If the pass transistor then turns on during evaluation, the charge on the dynamic output will share with the charge on the far side of the pass transistor, corrupting the result.

FIGURE 12. Charge sharing with pass transistors



As discussed in the last lecture, back-driving of pass transistors can occur when two select signals are both momentarily high. If S and S are both high, the inverters driving A and B will fight. If A is high and B is low, the fighting will cause A to dip and B to rise slightly. This noise on B can falsely cause a dynamic gate also driven by B to evaluate.

FIGURE 13. Back driving pass transistors



The moral of these problems is that neither the inputs nor outputs of dynamic gates should ever be connected to nodes which also connect to the diffusion terminal of a pass transistor.

3.6 Charge Injection into Substrate

Certain circuits such as I/O drivers powering highly inductive loads are prone to ringing which may send the output below GND. In such a case, the drain-substrate junction becomes forward biased and injects charge into the substrate. This charge may be collected on nearby dynamic nodes, corrupting the value. To prevent this, circuits prone to charge injection should be placed far away from dynamic logic. They should also be surrounded by guard rings. Charge injection is only a concern with special-purpose circuits and therefore is not part of the noise budget of most dynamic gates.

3.7 Summary of Domino Electrical Rules

In summary, domino gates must be checked for the following electrical rules:

- Neither inputs nor outputs of dynamic gates should connect to diffusion terminals of pass transistors.
- Nodes which may float indefinitely must be held by keepers.
- Noise margins on dynamic inputs and outputs must be checked.

The noise margin rules can be broken down into dynamic inputs and outputs. Dynamic inputs have a noise margin of only Vt. Major noise sources are power supply variation and coupling. Dynamic outputs have a noise margin set by the high-skewed gate following the dynamic gates. Major noise sources are charge sharing, coupling, and power supply variation.

Noise margins on outputs can be increased by reducing the skew of the subsequent gate. Noise margins on inputs can be increased with a moderate strength keeper.

A detailed noise budget prevents the need for overdesign. For example, power supply variation is large across a die, but small within a local area. Charge sharing is large for some gates with tall stacks and few secondary precharge devices, but zero for wide NOR gates. If worst-case power supply and charge sharing noise were assumed for all gates, coupling noise requirements would be very stringent and would lead to heavy shielding or wide spacing of lines. If actual noise is computed given the gate topology and power supply routing, coupling noise limits may be relaxed.

4.0 Monotonicity Issues

Standard domino gates can only implement non-inverting functions because every gate contains two inversions. Unfortunately, most interesting circuits require inversion somewhere. In particular, non-monotonic functions such as XOR cannot be implemented with a

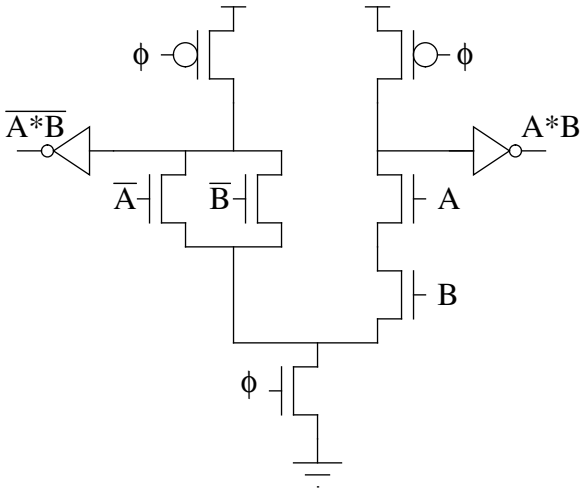
domino gate because the output might rise or fall in response to one input rising depending on the state of the other input.

4.1 Dual-rail Domino

A popular way to solve this problem is to use both true and complementary inputs and outputs, labeled _h and _l, respectively. The _h signal is asserted to mean a signal is 1. The _l signal is asserted to indicate a signal is 0. Neither _h nor _l are asserted when the gate is precharged and has not yet evaluated. If both _h and _l are asserted, an error has occurred. Such circuits are called Differential Cascode Voltage Switch (DCVS) or dual-rail domino.

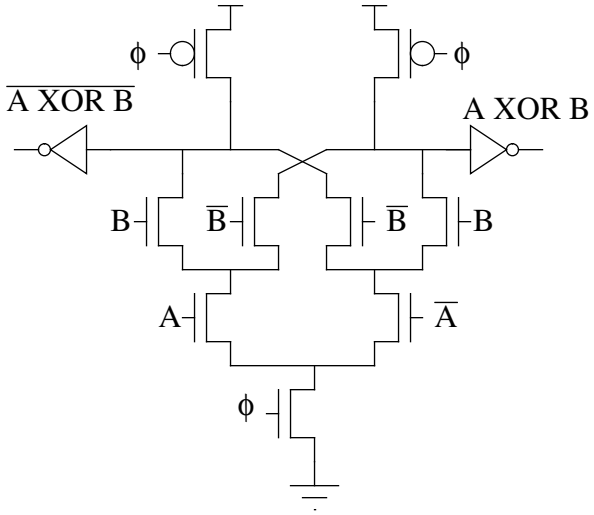
For example, a dual-rail AND/NAND gate is shown in Figure 13. It accepts inputs a_h, a_l, b_h, and b_l and produces out_h and out_l:

FIGURE 14. Dual-rail AND/NAND gate



The complementary output is built using DeMorgan's law, so series transistors in one stack become parallel transistors in the other stack and vice versa. Notice how the clocked pulldown device can be shared between the two stacks. In more complex gates, additional transistors can be shared. For example, a dual-rail XOR/XNOR gate is shown in Figure 14. If sharing was not done, nine transistors would be required in the pulldown network. Since A and $\overline{A}$ are shared between the true and complementary paths, only seven transistors are necessary.

FIGURE 15. Dual-rail XOR/XNOR gate

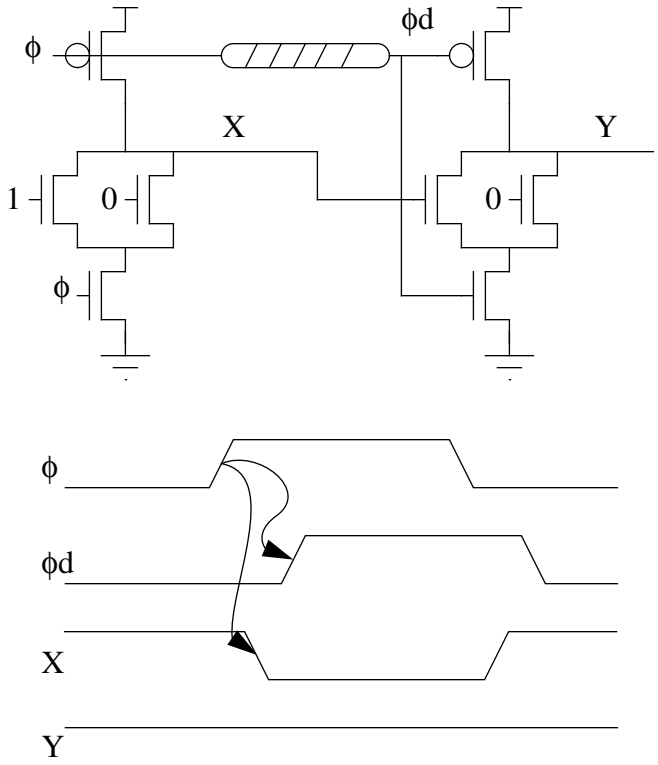


With dual-rail logic, all logic functions can be implemented. The drawback of any flavor of dual-rail circuit, dynamic or static, is the requirement of twice as many wires and almost twice as many transistors. As more transistors become available, transistor count becomes less of an issue, but wiring density is always a problem. Moreover, since dual-rail circuits use both complementary pulldown networks, wide gates cannot be efficiently implemented. In particular, wide NORs can be built well with conventional single-rail domino, but require tall pulldown stacks to produce complementary dual-rail outputs.

4.2 Self-timed Domino

An alternative to dual-rail domino is self-timed logic. Remember that the monotonicity rule only requires monotonic inputs while a dynamic gate is in evaluation. If we delay the evaluation clock to a dynamic gate until the inputs have settled, we can allow non-monotonic inputs. For example, consider a modification of the problem circuit from Figure 3, now shown in Figure 15:

FIGURE 16. ISelf-timed dynamic gates



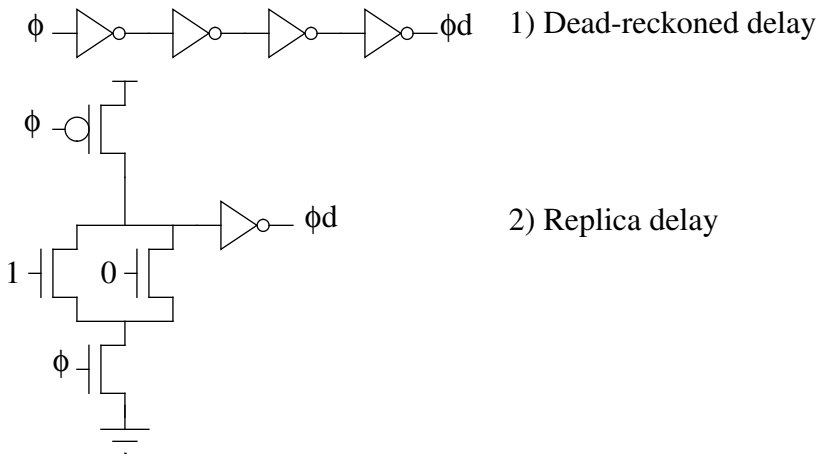
In this circuit, a magic delay element produces a self-timed clock ϕd which does not rise until node X has had time to evaluate. Thus, the second gate obeys the monotonicity rule and node Y correctly remains high. One danger of such circuits is that there is gate-source and gate-drain overlap capacitance on the second dynamic gate. If the gate pulls low on account of anther input, it may pull floating node X with it. If X fans out to other gates, the logic may fail.

The main challenge of self-timed circuits is to produce clocks which arrive at exactly the right time. If ϕd arrived too early, Y would incorrectly pull low. If ϕd arrived too late, the circuit will run more slowly. Since it is impossible to exactly match two delays, the designer must always make ϕd nominally arrive slightly late so that if the magic delay turns out to be shorter than expected, the circuit will still work. Unfortunately, if ϕd then arrives later than expected, the circuit will be slowed further. The difference between the delay to X and the worst case delay to ϕd is called the self-timing margin.

Self-timing margin depends on how well magic delays can be made to track with the actual gate delay. Two magic delay generators are shown in Figure 16. The first one uses inverters, which may change speed much differently than dynamic NOR gates over process and environmental corners. Self-timed margins of 30-50% are usually necessary; insufficient margin results in chip failure at any clock speed. The second delay generator uses a replica of the circuit it is attempting to match. This generator will track much better and can use a reduced margin, such as 10-20% plus the delay of the inverter. Replica

delays should always use the worst-case input patterns (in this case, only one of the NOR inputs being high), and should match the layout and loading of the circuit they must track.

FIGURE 17. Delay generator circuits



Self-timed circuits are especially popular in dynamic PLAs. A NOR-NOR PLA consists of an AND plane and an OR plane constructed from wide NOR gates. The OR plane must not begin evaluating until the AND plane has finished. This can be done by adding an extra row to the AND plane with worst-case loading; it's completion can kick off the self-timed start of the OR plane.

Self-timed circuits are also used in SRAM sense amplifiers and in certain other special-purpose non-monotonic dynamic logic. They are best to use when the advantages of wide NOR structures and compact gates outweigh the cost of the self-timing margin. Wide-scale self-timing has not yet been proved on large chips and is somewhat frightening because insufficient self-timing margin leads to irrecoverable chip failure.

4.3 Application: Integer Execution Unit

Lets consider how monotonicity issues impact a real circuit, such as an integer execution unit. Consider a six-way superscalar processor with four integer pipes. Each integer pipe contains a 64-bit adder which must run as fast as possible.

Fast adders are usually constructed with logarithmic carry select structures which are monotonic until the last gate, which is a multiplexor requiring true and complementary select lines. When designers used fully static circuits, the adders have no monotonicity issues. When designers first adopted dynamic circuits, they often built single-rail dynamic logic to implement all but the final stage, then used a static inverter and static multiplexor to select the output. Unfortunately, the transition back from static to dynamic logic, needed to use the result in the next cycle of computation, adds a budget for clock skew to the critical path, as we will learn in a future lecture. Therefore, there is strong incentive to build the adder entirely with dynamic logic.

The fastest reported adders, such as HP's Ling adder, are now built with fully dual-rail domino. The adders take true and complementary inputs and produce true and complementary outputs. Such adders have latencies of around 7 FO4 inverter delays.

Unfortunately for our super-scalar processor, each integer pipe requires two source busses and a result bus, for a total of 12 busses in the entire integer execution unit. If each bus is constructed with dual-rail logic, 24 metal lines are required in each bitslice. To reduce coupling, these lines may be run with greater than minimum spacing or even with shielding. Wire count on such datapaths is becoming a serious issue.

An alternative is to build single-rail logic with a self-timed clock on the final multiplexor which enables it when the non-monotonic inputs have settled. If the single-rail logic is just one half of the dual-rail design, the delay will increase significantly because the single-rail and dual-rail logic delays are almost the same but the single-rail circuits add a hefty self-timing margin. Instead, the single-rail logic may be redesigned to employ several stages of wide self-timed dynamic NOR functions which implement the carry chain taking advantage of the speed of dynamic NORs.

5.0 Circuit Pitfalls

We have seen numerous circuit pitfalls in the past lectures. In this section we'll collect them into a single list. We'll then apply the list to understand weaknesses of many other circuit families.

5.1 Ratioed Circuits

Ratioed circuits are any circuits that depend on the relative strength of PMOS and NMOS transistors. Process variation causes this ratio to change. Thus, the ratio must be selected so that even under worst case process variation, one transistor is sufficiently stronger than the other to produce the correct result. Pseudo-NMOS circuits are a classic example of ratioed logic.

5.2 Charge Sharing

Charge sharing occurs anytime charge may be transferred onto a floating node. It can lead to data-dependent failures when particular input patterns deposit the wrong charges which get shared with the floating node. Dynamic gates exhibit the classic example of charge sharing.

5.3 Voltage Drops

Certain circuits exhibit voltage drops which prevent their output from swinging rail to rail. Classic examples are NMOS pass transistors (which only swing up to V_{DD}-V_t) and bipolar transistors (which only swing up to V_{DD}-V_{BE}). If no restoring devices are included to pull the output to the rail, the PMOS transistors of the next stage may turn slightly on and

dissipate power. If the drop is too large a fraction of the power supply, the next stage may actually sense an incorrect value. This is especially severe for low-voltage circuits. V_{BE} is constant at 0.6-0.7 volts, so it becomes a larger fraction of the supply. Similarly, if a circuit designed in a high-voltage process with large values of V_t is then used on a low-voltage part, V_t drops become more important.

5.4 Power Supply Noise

The power supply VDD and GND voltages vary across the chip and in time. Operating frequency is set by the average value of the supply over the entire cycle; if VDD droops at the beginning of the cycle but charges back up later, the first gates will run slower than average and the last gates faster, for a net average delay that sets cycle time. Unfortunately, noise margins are impacted by instantaneous power supply variation.

Since the inductance of power pins is too large to supply significant current at extremely high frequency, most gates get their instantaneous switching current from nearby on-chip bypass capacitance. For example, suppose at the beginning of a cycle, a bank of flip-flops toggle and send new values to large bus drivers. The bus drivers pull their current from the nearby power network. This causes an IR drop through the nearby power lines. The current prefers to come from nearby sources with lower series resistance to minimize this drop. Thus, only nearby capacitors can supply much of the current. These nearby capacitors include both explicit bypass capacitance and implicit capacitance provided by all the nearby gates which are not switching. Even if the overall power supply resistance is low for the chip, a high supply resistance caused by a minimum width supply line driving many simultaneously switching gates will lead to large instantaneous IR drop and high noise. This must come out of the noise margin of the next gate.

Another bad bug from power supply noise occurs when a signal from far away drives the diffusion input of a transmission gate latch. If the signal GND is lower than the latch GND by a threshold, a low value of the signal will cause the source of the latch NMOS transistor to fall enough below the gate to turn on the transistor and discharge the value stored in the latch. This is avoided by either buffering the input to the latch with an inverter with larger noise margins or by using a strong enough feedback inverter on the latch node to hold the value even when the input transistor turns partway on.

Over a longer time, such as an entire cycle, average currents flow through the power pins to recharge the on-chip capacitance to an average value.

5.5 Coupling Noise

As we have seen in detail, coupling between adjacent signals is a major source of noise.

5.6 Leakage

Transistors leak even when off due to subthreshold conduction. As long as circuits are actively driven sufficiently often, leakage is not a problem. However, if the clock is

stopped indefinitely, nodes which are not driven may leak away to the wrong value. The rate of leakage increases exponentially with reduced threshold voltages.

5.7 Alpha Particle Hits

Alpha particles are helium nuclei (two protons and two neutrons) which hurtle through space at high speed. They occasionally crash into chips and dump their energy onto the node they hit. If they hit floating dynamic nodes, such as RAM cells or dynamic gate outputs, they can cause the voltage to change. To minimize this change, the energy held on the capacitance of the floating node must be much greater than the energy in the alpha particle. Typical design rules call for a minimum of several fF of capacitance on any dynamic node.

5.8 Contention

Whenever two drivers connect to a single node, contention can occur. This contention could cause glitches on sensitive inputs attached to the node. It can also cause failure in circuits like latches where feedback allows the latch to be backdriven.

5.9 Races

Whenever the ordering of two signals matters, it is important to guarantee the correct order will occur under any processing or environmental variation. A classic race problem arises constructing matched delays for dynamic logic, as was discussed in a previous lecture.

Extra margin is added to the delay which is supposed to be longer to make sure the signal always arrives last. This margin may be 50% of the nominal delay if layout information is not available. With extracted layout information, the margin can be relaxed to around 30%. These margins are very large and significantly impact chip performance, so often it is best to avoid races entirely.

Races are crucial to operation of a chip. Since races are based on relative delays instead of on a clock, a failed race will force the chip to be scrapped, rather than just run more slowly.

5.10 Noise Margins

Many of these effects described in this section contribute noise to a system. The combined noise of all the effects may cause circuits to fail even when any single source is too small. Thus, noise budgets are used to divide the total noise margin of a circuit among the various sources of noise. The chance of failure can never be totally eliminated, but it can be reduced to acceptable levels.

Noise margins are largest for static gates with switching thresholds near VDD/2. Skewed static gates have smaller, but still hefty margins. Dynamic gates have input noise margins of only V_t. Sense amplifiers have even smaller noise margins.

Since the noise margins of dynamic gates are so low, being overly conservative about noise budgets results in undesignable circuits. In particular, coupling noise is a function of layout, so decreased coupling noise can be traded for increased layout area. By taking advantage of specific information like the fact that dynamic NOR gates never experience charge sharing, some dynamic gates can allow higher coupling noise and thus easier layout.

5.11 Tool limitations

Microprocessors have become so complex that no human can verify a design. Instead, automatic tools must be used to analyze circuits and check design rules. Hence, special circuits which cannot be handled by available tools must not be used on a wide-spread basis.

6.0 More Circuit Families

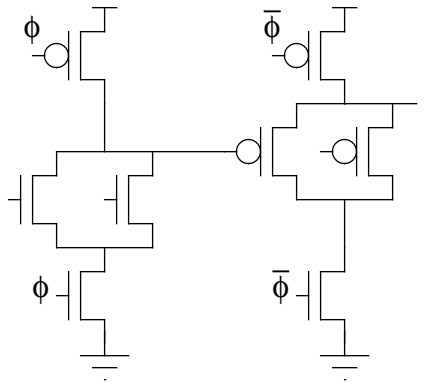
Having discussed a variety of circuit pitfalls, let us move on to examine more circuit families. Keep the pitfalls in mind while analyzing the circuit families.

6.1 NORA

Domino logic is fast because it reduces the input capacitance of dynamic gates by eliminating PMOS transistors and because it responds to smaller input swings. Unfortunately, there is still a static inverter between each dynamic gate.

NORA (NO RAce) logic [1] replaces the static inverter with another precharged gate. The replacement gate is precharged low and evaluates high through PMOS transistors, as shown in Figure 2. Though such PMOS dynamic gates are not as fast as normal NMOS dynamic gates, they are still faster than inverters.

FIGURE 18. NORA Gates



NMOS Block PMOS Block

Unfortunately, now the noise margins of the gates are greatly reduced. The PMOS precharge block only has a noise margin of Vt, like the NMOS precharged block. Unfortunately, its inputs now come from noise-prone dynamic nodes rather than from statically driven gates. DEC tried the technique on the J-11 chip and “it was a disaster.” AT&T tried a related technique, Zipper Domino [2], on their CRISP microprocessor. Zipper swings the clocks less than rail-to-rail so the precharge transistors remain partially on and serve as weak keepers to improve noise margins. Unfortunately, Rev 1 of the CRISP ALU did not work.

NORA and its variants are therefore generally regarded as too risky for general use.

6.2 BiCMOS

BiCMOS logic takes advantage of the excellent current drive characteristics of bipolar transistors to build buffers that can quickly drive large fanouts. A typical BiCMOS inverters shown in Figure 3.

FIGURE 19. BiCMOS Inverter

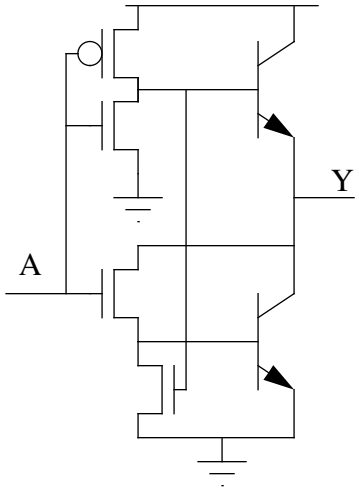
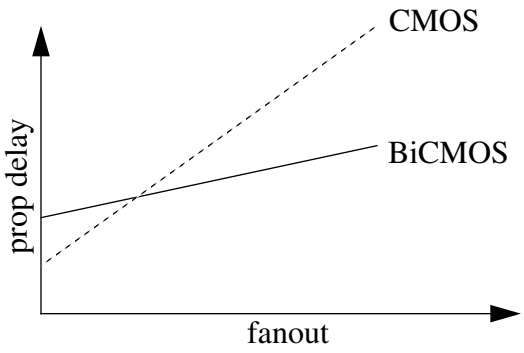


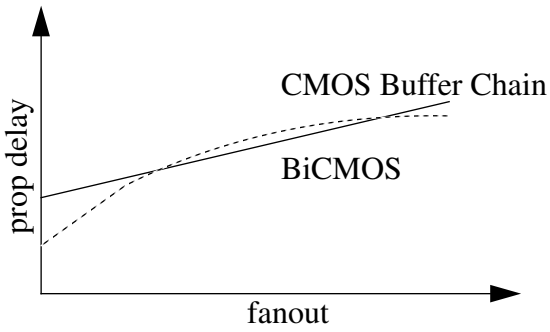
Figure 4 shows the relative delays of BiCMOS and regular CMOS inverters. For the same input capacitance, a BiCMOS driver has a larger intrinsic delay but a smaller fanout-dependent slope. This is the principle argument for using BiCMOS drivers.

FIGURE 20. CMOS and BiCMOS inverter delays



The flaw in this argument is that good CMOS designers would never drive huge fanouts with an inverter anyway. Instead, they would construct an exponential horn of inverters with a fanout of around 4 per stage to drive a large load in time logarithmically related to fanout. Such a delay curve is shown in Figure 5. Note that these delays are not to scale, but are meant to illustrate the general concepts.

FIGURE 21. CMOS and BiCMOS delays with tapered buffer chains



Although BiCMOS is not better than CMOS by a huge factor, it still can be somewhat faster. Bipolar transistors take a large amount of area and require extra processing steps, but the costs can be justified where speed benefits sufficiently. Intel used BiCMOS gates with success on Pentium and Pentium Pro in their 0.8 and 0.6 micron processes, and in a slow version of their 0.35 micron process.

Bipolar transistors finally passed out of style with the scaling of VDD. Power supplies are falling low enough that the V_{BE} drop can no longer be tolerated. A 0.8 volt drop on a 5 volt supply is tolerable, but a 0.8 volt drop on a 1.6 volt supply produces an output that is very hard to use. Intel no longer uses bipolar transistors on its 0.35 micron and below processes intended for low power supplies.

Exponential made a valiant attempt to use bipolar transistors in a PowerPC implementation [3]. Using bipolar ECL logic and BiCMOS caches, they originally targeted 533 MHz operation at under 85 watts of power. Unfortunately, they never managed to ship parts above mid-400 MHz. Since bipolar logic is less dense than CMOS, they were also forced to use a simpler microarchitecture delivering fewer SpecInt/MHz, so the overall performance was scarcely better than Motorola/IBM PowerPC chips and the price was not competitive.

6.3 Low-Threshold Circuits

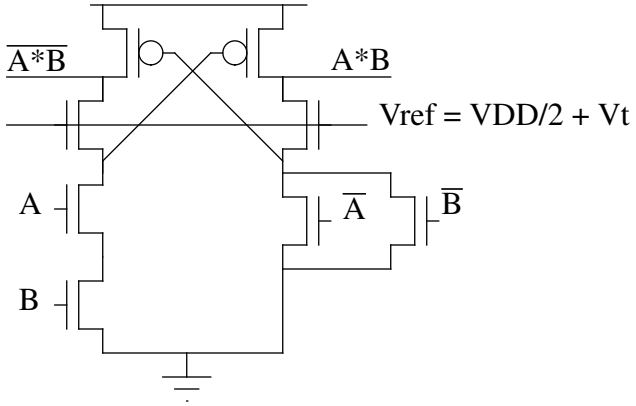
Some processes feature both high and low threshold devices. These can be used to trade between speed and power consumption. For example, NMOS-only pass-transistor circuits can be built with low-threshold transistors. If they drive gates with high thresholds, the threshold drop from the NMOS pass-transistor may be small enough not to turn on the next gate. Thus, level-restoring circuits are not needed. Another example is to build the evaluation transistor of dynamic gates with a high threshold device, but other logic transistors with a low threshold. This reduces subthreshold conduction and power consumption during precharge because the clocked evaluation transistor is off, while preserving the speed benefits of the low-threshold devices in the remainder of the pulldown stack. Unfortunately, noise margins of the dynamic gate are just the threshold of the low-threshold transistor, which can be very small. Also, a keeper is still necessary to fight leakage through the low-threshold transistors while the gate is in evaluation.

6.4 Partial Swing Circuits

The time required for a gate to charge a load capacitance is $\Delta t = C/I\ \Delta v$. Sizing transistors optimizes the C/I ratio of the load to the driver to minimize dt. However, reducing the output swing Δv can also reduce Δt. Circuits which capitalize on this idea fit in the class of partial swing circuits.

One example of partial swing logic is DSL, Differential Split-Level logic (DSL). DSL clamps the internal swings of gates to between 0 and VDD/2 by using a cascoded NMOS transistor tied to VDD/2 + Vt. The output voltages can pull up to VDD but not quite all the way down to 0. As a result, there is some DC power consumption.

FIGURE 22. DSL AND/OR gate



The original paper on DSL [4] obtained good results because it built the pulldown NMOS trees with 1 micron long transistors in a 2 micron process! The paper claimed that the reduced V_{DS} of the pulldown transistors allowed reliable operation. This reduction in channel length more than compensated for the increase in delay due to the cascode transistor. In general, reducing transistor length below the minimum for the process is not wise and/or possible.

DSL also suffers from several other problems. One is that if the gate idles for long enough, subthreshold conduction may pull internal nodes above VDD/2. This will slow the gate the next time it switches because the internal voltage swing is large. Another problem is generating the reference voltage and routing it around the chip without much noise.

6.5 Sense Amplifier Circuits

An extension of the partial swing circuit idea is to build a sense amplifier into a gate which detects a small swing and quickly amplifies it. A classic example of a good application of sense amplifiers is in SRAM arrays. Amplifiers have also been used in a variety of esoteric differential dynamic logic families, mostly with poor results.

In a SRAM array such as in a cache, a small SRAM cell activates when its word line is asserted and must read out onto differential precharged bitlines. Since there may be 127 or more other SRAM cells on the same line and since the SRAM transistors are sized very small for good density, the delay for one bitline to drop is very large. Instead of waiting for the bitlines to take on valid logic levels, a sense amplifier is used to amplify small differences between the two bitline voltages. Self-timed delays may be used to activate the sense amplifier when sufficient there is sufficient voltage difference. Examples of such circuits will come in a future SRAM lecture.

Three versions of dynamic differential logic using amplifiers are Sample-Set Differential Logic (SSDL) [5], Enabled/disabled CMOS Differential Logic (ECDL) [6], and Differential Current Switch Logic (DCSL) [7]. All are prone to charge sharing noise getting amplified and causing incorrect evaluation. The performance is often no better, or even worse, than normal dual-rail domino, so such logic families are not widely used.

6.6 Conclusions

As design requirements and tools change and implementation technologies evolve, the set of useful circuit families also evolves. Thus, it is somewhat dangerous to make blanket recommendations of circuit families. Nevertheless, there are clear trends that just a few circuit families are used for the great majority of designs. Let us summarize these trends, keeping in mind the reasons as well as the results.

Most circuits are built using static CMOS, transmission gates, and dynamic logic. Dynamic logic is used for speed. Transmission gates may be used for multiplexors or XOR structures. Static CMOS is used everywhere else for simplicity and robustness. Some circuit styles allow occasional use of pseudo-NMOS gates when DC power consumption can be tolerated and wide NOR structures are needed. Specialized circuits are used in certain well-understood custom blocks, such as small-swing sense amplifiers in SRAM and special drivers on low-voltage I/O pads.

BiCMOS has been popular in the Intel Pentium and Pentium Pro processors, but is no longer used in Pentium II because the power supply is low enough that the VBE drop cannot be tolerated. Exponential made a valiant attempt to use bipolar transistors in their high-speed PowerPC implementation, but sadly failed to meet performance targets. As power supplies continue to fall, bipolar transistors will probably vanish from the digital design scene.

Most other circuit families suffer from pitfalls that doom them to remain academic curiosities. When evaluating proposed new circuit families, keep a critical eye toward potential pitfalls.

7.0 References

[1] N. Gonclaves and H. De Man, “NORA: A Racefree Dynamic CMOS Technique for Pipelined Logic Structures,” *J. Solid State Circuits*, June 1983, pp 261-266.

[2] Zipper Domino: *IEEE J. Circuits & Devices*, May 1986, pp. 10-16.

[3] E. Cohen, et. al, "A 533 MHz BiCMOS Superscalar Microprocessor," *IEEE Intl. Solid State Ckts. Conf.*, Feb. 1997, pp. 164-5.

[4] L. Pfennings et. al., "Differential Split-Level CMOS Logic for Sub-Nanosecond Speeds," *IEEE Intl. Solid State Ckts. Conf.*, Feb. 1985, pp. 212-213.

[5] T. Groijohn and B. Hoefflinger, "Sample-set differential logic (SSDL) for complex high-speed VLSI," *IEEE J. Solid-State Circuits*, vol. 21, no. 2, pp. 367-369, Apr. 1986.

[6] S. Lu, "Implementation of iterative networks with CMOS differential logic," *IEEE J. Solid-State Circuits*, vol. 23, no. 4, pp. 1013-1017, Aug. 1988.

[7] D. Somasekhar and K. Roy, "Differential Current Switch Logic: A Low Power DCVS Logic Family," *IEEE J. Solid-State Circuits*, Vol. 31, No. 7, pp. 981-991, July 1996.