Hot-Clock nMOS

Charles L Seitz
Alexander H Frey
Sven Mattisson
Steve D Rabin
Don A Speck
Jan L A van de Snepscheut

Computer Science
California Institute of Technology

5177:TR:85

Hot-Clock nMOS

Charles L Seitz
Alexander H Frey
Sven Mattisson
Steve D Rabin
Don A Speck
Jan L A van de Snepscheut

Computer Science Department
California Institute of Technology

5177:TR:85

The research described in this paper was sponsored by
Defense Advanced Research Projects Agency, ARPA Order No. 3771,
and monitored by the Office of Naval Research
under contract number N00014-79-C-0597

to appear in
Proceedings, 1985 Chapel Hill Conference on VLSI

Hot-Clock *n*MOS

Charles L Seitz, Alexander H Frey[1], Sven Mattisson[2]
Steve D Rabin, Don A Speck, Jan L A van de Snepscheut[3]

Department of Computer Science
California Institute of Technology
Pasadena, CA 91125

1. Also affiliated with IBM Los Angeles Scientific Center.

2. At Caltech 1982-84; now at the Applied Electronics Department, Lund Institute of Technology, Sweden.

3. At Caltech 1983-84; now at the Department of Mathematics and Computing Science, Groningen University, The Netherlands.

Abstract: "Hot-Clock *n*MOS" is a style of design that has advantages in circuit energetics and performance. When the application of this style is carried to its limits, an *n*MOS chip is powered entirely from its clock signals. There are savings in area, delay, and power, even when the bootstrap circuits of this style are used together with conventional circuitry. We have used this technique in numerous small projects and test structures, and in 3 substantial projects fabricated through MOSIS.

1. Energetics

How is the power required by and dissipated on a MOS chip used? Even in CMOS technology, in which the static power is negligible, "dynamic" power is required to charge and discharge capacitances:

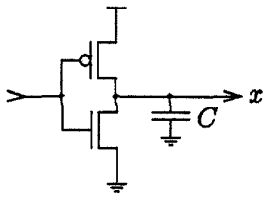

Figure 1

The research described in this paper was sponsored in part by the Defense Advanced Research Projects Agency, ARPA Order number 3771, monitored by the Office of Naval Research under contract number N00014-79-C-0597; and in part by IBM Corporation.

Each time the signal x is to change $0 \rightarrow 1$, the power supply must provide a quantity of charge CV_{dd} at potential V_{dd}, hence energy CV_{dd}^2. Half of this energy ends up stored in the capacitance C, and the other half is dissipated in the p-channel transistor. When x is to change $1 \rightarrow 0$, the charge stored on C is conducted through the n-channel transistor into the ground terminal, and the stored energy is dissipated in the transistor. Thus in a full cycle $0 \rightarrow 1 \rightarrow 0$ of signal x, energy CV_{dd}^2 must be supplied to the chip, and is dissipated in the transistors that drive this signal. The fundamental motivation behind hot-clock nMOS is to get around this "inevitable" dissipation of power on a high-complexity chip.

If one were to try to spot the places on a MOS chip where most of the dynamic power goes, it would be in the drivers of relatively large capacitances – long and/or highly loaded wires – that are driven at relatively high frequencies. Examples of such signals are control lines and data buses in instruction and arithmetic processors; word and bit lines in RAMs; literal and implicant lines in large PLAs, and output pads. By virtue of their capacitance, these are also signals that are difficult to drive with small delay.

Many of these signals are naturally driven in synchrony with one of the clock signals, so another possibility is to drive them through some approximation to an ideal switch:

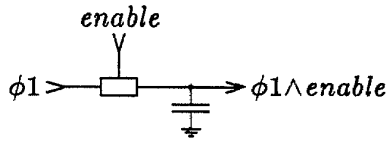

Figure 2

Here we assume that the output is initially 0, and that the *enable* signal changes only during $\phi 2$ in a two-phase non-overlapping clocking scheme. The output is then $\phi 1 \wedge$ *enable*. If the switch were ideal, the circuit would introduce no delay, the output being just a gated replica of the input clock. Also, the switch turns on only when there is no voltage across it, and off when there is no current flowing through it, so even if it did exhibit some non-zero resistance or conductance while switching, it dissipates no power in changing state.

Assume that when this switch is implemented with MOS transistors, it can be modeled as an ideal switch in series with an effective resistance

R, and that the clock transition can be modeled as a ramp from 0V to V_c in time t_r, and from V_c to 0V in time t_f:

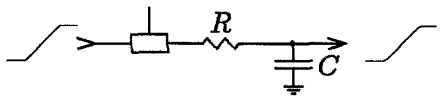

Figure 3

Let the switch be on. If t_r were 0, the full clock voltage would appear initially across the resistance, and the output node would be charged conventionally with the CV_c^2 energy from the clock being split between the output node capacitance and dissipation in the resistance. The case of interest to us is where t_r is some normal and achievable value – say a few ns –, and the sizes of the transistors that form the switch are selected so that R is small enough that the delay through the circuit, approximately RC, is much less than t_r. In this case in which $RC \ll t_r$, the switch is also sized to dissipate little power. The energy dissipated per switching event is closely approximated by:

$$E \approx \frac{RC}{t_r} CV_c^2.$$

Presume that the value of RC is fixed in the design, that is, that the sizes of the switch transistors are determined according to a given $RC \ll t_r, t_f$. Presume also that t_r and t_f are a fixed fraction of the clock period T. The same chip may then be operated at a longer clock period T, resulting in slower operation but also in a smaller *proportion* of the energy supplied in each clock transition being dissipated on-chip. In other words, these circuits exhibit a characteristic in which the total energy required to perform a computation varies as $1/T$. The computation is less costly if one is not in a hurry. This characteristic is at odds with complexity arguments that assert that the cost be expressed in E_{sw} units (§9.10 in [5]), and is interestingly similar to the AT^2 invariance exhibited by many algorithms [8], in which the cost AT of performing a given computation also varies as $1/T$.

This scheme does not really "solve" the power and speed problems of driving the capacitance C. Rather, we have *exported* the problem* elsewhere, namely, to whatever circuit drives the signal $\phi 1$.

* One might compare this technique with a country banning certain polluting industries, but still purchasing the products of those industries from other countries. The country may then be accused of "exporting pollution".

In hot-clock nMOS we export the problem off the chip. The off-chip clock driver can then use a technology that is better suited to driving the capacitive clock load than is the high complexity MOS technology. For example, bipolar transistors have much higher transconductance than MOS devices at present feature size, and make excellent clock drivers. A more interesting possibility is to use clock driver circuits that employ inductances. A resonant driver allows an almost lossless transfer of charge from the power supply to the clock capacitance, and then from the clock capacitance back into the power supply. Figure 4 presents an idealized circuit to implement this scheme of saving power in driving capacitive loads:

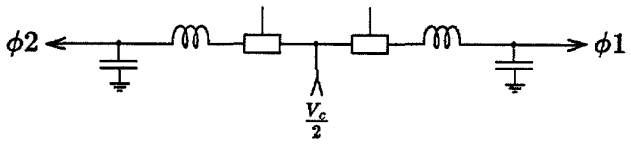

Figure 4

A practical circuit is rather more complicated. This trick is somewhat like "LC logic" (§9.1.3 in [5]), with the L's brought off the chip, and is subject to the same inevitability of some loss in the switching process. However, such techniques could allow VLSI systems – even in nMOS technology – to operate at dramatically lower overall dissipation levels than present CMOS circuits, let alone nMOS circuits.

2. The Elementary nMOS Clock-AND

This technique of exporting the problem of driving large capacitive loads can be applied either to CMOS or to nMOS designs. However, it is nMOS technology that benefits most from a better way to drive signals to 1 quickly, and that lends itself most readily to a set of elegantly simple clock-powered bootstrap circuits. An nMOS circuit that serves as a good approximation of the behavior idealized in Figure 2 is the "Clock-AND":

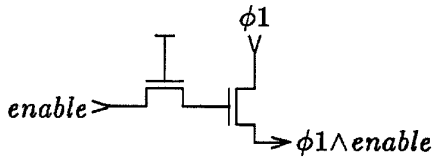

Figure 5

The transistor whose gate is connected to V_{dd} is called the *isolation transistor*, and the transistor that passes the clock signal is called the *clock-pass transistor*. The *enable* signal is assumed to change only during $\phi 2$. The origin of this circuit is probably as old as MOS technology. The earliest reference that we know of that describes the circuit carefully [3] is from 1972, and includes an analysis of a *p*MOS version of the clock-AND.

If *enable* is 0 during $\phi 1$, the isolation transistor holds the gate of the clock-pass transistor at 0, the clock-pass transistor is off, and $\phi 1$ does not pass to the output. If *enable* is at the logic-1 voltage, nominally V_{dd}, the voltage on the gate of the clock-pass transistor just before the beginning of $\phi 1$ is $V_{dd} - V_T$. Then when $\phi 1 : 0 \rightarrow V_c$, the gate of the clock-pass transistor is "bootstrapped" to a voltage more positive than V_c by the coupling of the gate-to-channel capacitance of the clock-pass transistor. The isolation transistor fulfills the role suggested by its name by remaining off while the gate of the clock-pass transistor, also called the bootstrap node, is at a voltage in excess of $V_{dd} - V_T$. The clock voltage, V_c, whether larger or smaller than V_{dd}, is passed to the output. When $\phi 1 : V_c \rightarrow 0$, the output is discharged through the clock-pass transistor back into the clock signal.

The switch-level simulator MOSSIM II [1] models the behavior of these circuits correctly, and we believe that other switch-level simulators can be adapted similarly. However, the bootstrap action and the drive capability of the clock-pass transistor depend on proper transistor sizing.

Although there is no absolute criterion for what is satisfactory bootstrap voltage, this elementary clock-AND is almost foolproof. The capacitance between the gate and channel of the clock-pass transistor gives adequate bootstrap voltage, so long as the parasitic capacitance of the bootstrap node is reasonably minimized in layout. If the parasitic capacitance is too large, the capacitive divider formed between the gate-to-channel capacitance and parasitic capacitances yields too small a V_{gs} on the clock-pass transistor for good bootstrap performance.

For the elementary clock-AND we have used "Speck's rule", which states that the bootstrap action is satisfactory as long as the gate capacitance of the clock-pass transistor is at least 4 times the parasitic capacitance on the bootstrap node, each calculated using typical process parameters. This rule of thumb is based on SPICE simulations with a wide variation of process parameters around those observed for typical MOSIS *n*MOS runs, as well as test chips and projects fabricated on every run since mid-1982.

The rules for selecting the size of the clock-pass transistor are less empirical, and exhibit an interesting speed-power tradeoff. As indicated

in the previous section, the case of interest is when RC, the approximate delay of the clock-AND, is much less than the rise- or fall-time, t_r or t_f. In a typical example from one of our designs, we might assume that $t_r, t_f \geq 5$ns, and we choose the size of the clock-pass transistors to assure that $RC \leq 0.5$ns. In order to drive 0.5 pF of capacitance – 32 minimum geometry transistors and 800μm of poly wire in a 3μm nMOS process (a select line in a RAM) –, R must be less than $1K\Omega$. The voltage across the transistor is small, and the analysis is approximate, so let us use a simple resistive model for the transistor: $1/R = \frac{W}{L}\mu C_{ox}(V_{gs} - V_T)$. For $\mu C_{ox} \approx 50\mu\text{A/V}^2$ and $(V_{gs} - V_T) \approx 2\text{V}^*$, the transistor is about $10K\Omega/\square$, and $\frac{W}{L} = 10$ will serve. This choice results in quite a small area, energy, and delay for "driving" such a load. One is reminded, however, that the clock-AND circuit is not properly called a "driver". It does not provide any power amplification of the clock input.

Having selected the size of the clock-pass transistors in a design to a given limit on RC, *eg* $RC \leq 0.5$ns, one may still trade speed and power dissipation by the choice of t_r, t_f and clock period. The target value of RC determines also the tolerable resistance of the clock distribution conductors.

One caution about using the clock-AND circuit is that when the *enable* signal is 0, the output is floating, and is thus susceptible to charge sharing and spurious capacitive coupling. What is more, unless *enable* is 1 at some regular interval, leakage currents may cause the floating output node to drift to some voltage significantly above 0V. Thus the clock-AND is usually augmented with a *keeper* transistor that keeps the output at ground, such as:

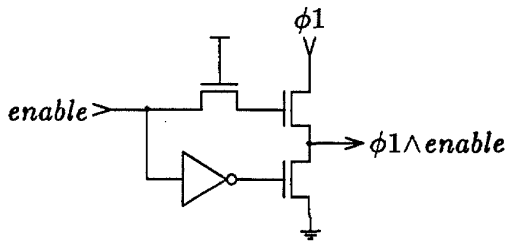

Figure 6

In this circuit the output is always driven. However, this property is not always desirable, and the circuit requires a static inverter, so one

* Body effect increases V_T at the elevated source potential, thus this conservative estimate.

may prefer to drive the keeper with the alternate clock. The output is then kept at 0V during $\phi 2$, so that leakage currents may not accumulate charge on the output node, and is either floating or driven to V_c during $\phi 1$. Such circuits can be used in "wired-OR" forms, and also, clock-ANDs can be put in series with the output of one driving the clock input of another to form an AND function:

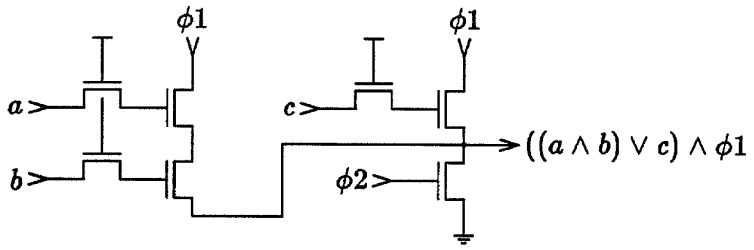

Figure 7

A variant on the clocked keeper is suggested when the ground is not conveniently available in the layout. Since $\phi 1$ is 0 while $\phi 2$ is at V_c, one can as well cause the keeper transistor to sink the output to $\phi 1$, resulting in the following peculiar-looking but perfectly functional circuit:

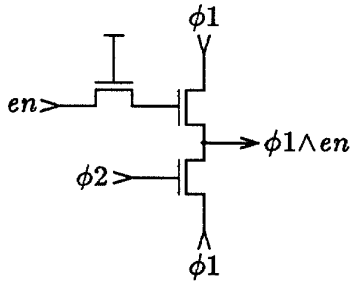

Figure 8

All the work in switching the output in these circuits is done by the clock-pass transistor. The keeper transistor turns on only when the output is already at 0V, and can be of minimum size.

The techniques outlined in this section can be used in conjunction with Mead-Conway style nMOS designs with no known difficulties, even for beginning students. When the students in the Caltech VLSI design course learn about nMOS (following their first project, which is done in CMOS/SOS), they learn these techniques from the start. The resulting

project chips have advantages in (1) higher speed, which in nMOS is otherwise seriously limited by the ability to drive signals to 1 quickly with depletion pullups or "superbuffers", (2) less sensitivity of speed to variations of the depletion transistor threshold, (3) less power dissipation, and (4) less area.

Since the clock signal originates off-chip, it is not strictly necessary that the clock HIGH voltage, V_c, be the same as V_{dd}. There are advantages to making V_c exceed the nominal logic-1 voltage, V_{dd}, by at least the V_T of the enhancement transistors, so that logic-1 voltages are not degraded through pass transistors whose gates are driven by a clock or gated clock. We typically use $V_c = 7\text{V}$ and $V_{dd} = 5\text{V}$, but the circuits also work correctly, although slower and at lower power, with $V_c = 6\text{V}$ and $V_{dd} = 4\text{V}$. Since the saturation current in pass transistors depends on $(V_{gs} - V_T)^2$, a little extra clock voltage also goes a long way to increasing performance. If $V_c = V_{dd}$, the usual ratio rules apply, but if $V_c \geq V_{dd} + V_T$, one can avoid the degraded logic-1 signals that require higher ratios. In addition to passing the V_c voltage, the clock-AND output switches all the way to 0V, a property that is critically important to driving select lines in dynamic RAMs [2].

3. Fancy Circuits

If this hot-clock style of nMOS design – typified by powering circuits from the clock signals – were "carried to its limits", can it also be made universal? That is, can extensions of these circuits be used to implement arbitrary logical functions and reliable clocked storage elements? They can, and we believe to excellent advantage; however, *Fair Warning:* At this point we shall depart from circuits that are reasonably foolproof. Correct operation may depend on capacitance ratios, and charge sharing between nodes of different capacitance may be used determine the direction of signal flow. The tolerance of these circuits to process variation should be checked by circuit simulation.

Let us start with a clocked storage element. Having said so far that the clock-AND *enable* signal changes during $\phi 2$ and remains stable during $\phi 1$, we can make a circuit that latches the control input on $\phi 2$, what we call a *clocked-isolation* clock-AND (Figure 9). This circuit does not require V_{dd}, so this is a good time to dispense with ground as well (except for the substrate), as in the circuit of Figure 8. However, in this or in some of the following circuits, certain clock inputs can be replaced with ground.

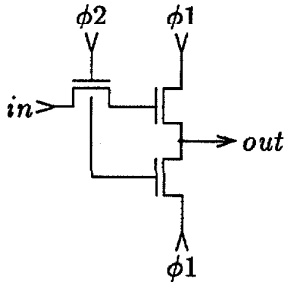

Figure 9

One thing to notice about this circuit is that the isolation transistor can turn on while there is voltage across it, and accordingly, it dissipates power in charging or discharging the bootstrap node. The goal of exporting *all* of the dynamic power is elusive.

When the clocked-isolation clock-AND is used as a clocked storage element, say to make a shift register:

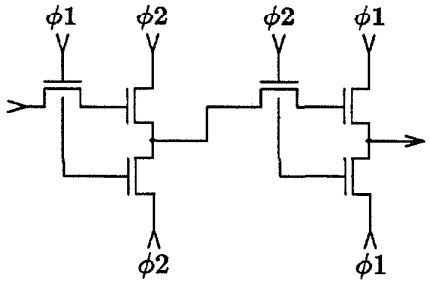

Figure 10

there are several possible modes of misoperation. First of all, charge sharing occurs at the beginning of $\phi 2$ between a floating 0 output of the first stage and the storage/bootstrap node of the second stage. For a typical shift register layout the charge shares in the direction opposite to our intent. Hence this circuit is used only in cases in which the parasitic capacitance between the stages is large. A reliable shift register circuit will be shown after these preliminaries.

Another problem is that a race occurs when the output of the first stage is switching $1 \rightarrow 0$ at the same time as the isolation transistor at the input of the second stage is turning off. The outcome of the race, based on circuit rather than switching considerations, is as we intend,

at least in the absence of clock skew. The skew tolerance of ordinary two-phase clocked circuits is determined by the t_{12} and t_{21} non-overlap periods (§7.2 in [5]). In these hot-clock circuits the skew tolerance is determined also by the relationship between the clock slope, $\frac{V_c}{t_f}$, and the V_T of the isolation transistor. Starting during $\phi 2$ with the output from the first stage equal to V_c, the bootstrap node in the second stage is at a voltage $V_c - V_T$. As $\phi 2$ switches $V_c \rightarrow 0$, the signal at the input to the second stage would have to precede $\phi 2$ by V_T *in voltage* to keep the clocked-isolation transistor on, so that some charge on the bootstrap node could escape. This voltage margin corresponds through the clock slope to a skew margin of $\frac{V_T}{V_c} t_f$ *in time.*

The most difficult problem with the clocked-isolation clock-AND is the case in which the output pulse is not to appear. When the bootstrap node is discharged and – unlike the elementary clock-AND – *isolated* prior to $\phi 1 : 0 \rightarrow V_c$, the clock transition can couple through the gate-drain overlap capacitance of the clock-pass transistor to turn the clock-pass transistor on. The non-linearity in gate-to-channel capacitance is in our favor: when the clock-pass transistor is off, the capacitance from gate to drain is small, but there must be sufficient capacitance on the bootstrap node to absorb the charge coupled from the drain overlap capacitance. We have had some MOSIS circuits work correctly without placing any minimum on the capacitance on the bootstrap node, but only from runs in which the drain overlap capacitance is unusually small. Reliable operation over the range of parameters for MOSIS nMOS runs appears to require a capacitance on the bootstrap node at 0 voltage comparable to the "on" capacitance of the clock-pass transistor. A depletion transistor with its gate at 0 volts and source and drain connected to the bootstrap node is a preferred way to provide this capacitance. At 0 volts one has the benefit of the gate capacitance of the "on" transistor, while at voltages more positive than $-V_T$ of the depletion transistor, the parasitic capacitance is minimized.

This clocked-isolation clock-AND is not used very often in our hot-clock designs. However, it exhibits several of the pitfalls that can appear in the extreme form of the hot-clock style; thus the extended discussion of this 3-transistor circuit. Just to complete the story, so that one might understand that this circuit serves as an non-linear amplifier with respect to the input (necessary for level restoration in digital systems), we display in Figure 11 the experimental transfer characteristic of an elementary clock-AND.

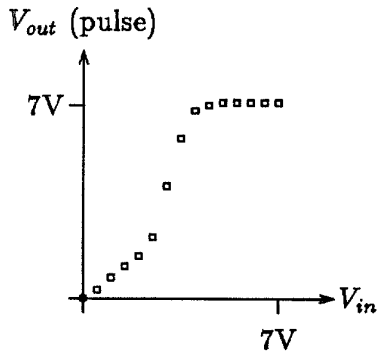

Figure 11

Given the clocked-isolation clock-AND as a clocked storage element and restoring amplifier, it is possible to implement combinational logic by pass networks, such as:

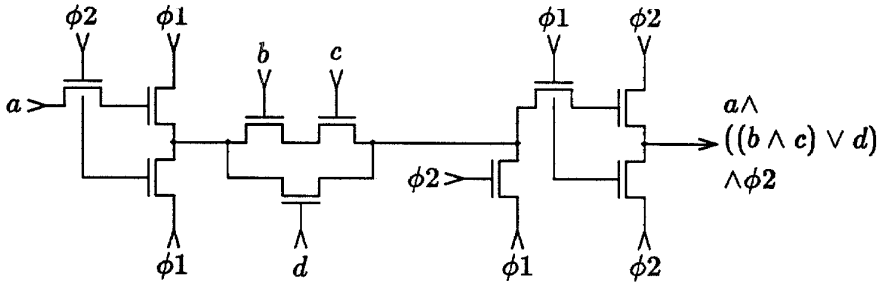

Figure 12

The signals b, c, d may either be stable during $\phi 1$, or may be gated $\phi 1$ pulses. Internal circuit nodes in the pass network capable of significant charge sharing must be pre-discharged during $\phi 2$. A circuit in this form was used, for example, to rotate a quaternary (1-of-4) coded pass input according to a quaternary control input to perform a fast quaternary addition in a multiplier chip.

A pass network demonstrates the ability to compute $\wedge$–$\vee$ expressions, but not complements. It is no great trick to compute the complement of a signal by precharging a node, then discharging it conditional on a clock signal on the following phase:

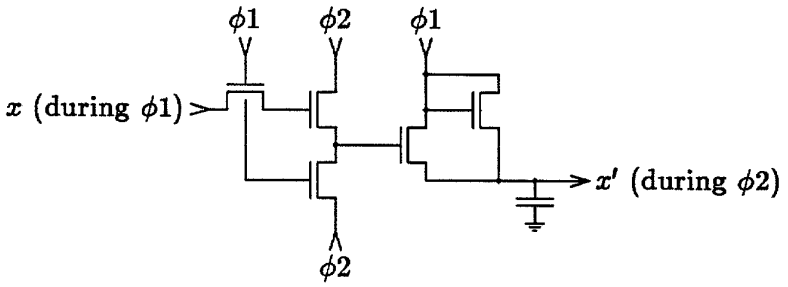

Figure 13

We call the transistor whose gate and drain are both connected to $\phi 1$ a *diode precharger.* It is a pass precharging device that dissipates power only because of its V_T "forward drop". However, the energy stored during $\phi 1$ on the output load capacitance by this diode precharger is (horrors!) *dissipated* in the *discharge transistor* during $\phi 2$ if the output is to be 0. This approach is obviously not very nice if C is large and we are serious about saving dynamic power. Also, the complement is delayed.

The *inverting* clock-AND is based on this same precharge and discharge trick, but more elegantly applied:

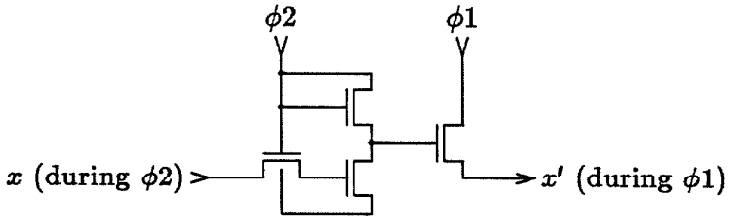

Figure 14

The bootstrap node is precharged by the diode during $\phi 2$. Thus the clock-pass transistor actively sinks the output to $\phi 1$ during $\phi 2$. It is doing double-duty as the keeper (*cf* Figures 8 & 9). The computation of the complement is fairly easy to understand if you first appreciate that it is accomplished in the non-overlap period between $\phi 2$ and $\phi 1$. If the input latched at the end of $\phi 2$ is 0, the bootstrap node remains charged through the next $\phi 1$ epoch, and a $\phi 1$ pulse is produced at the output. If the input latched is 1, the bootstrap node is discharged as $\phi 2$ switches to 0, and no output pulse is produced during $\phi 1$.

Both of the transistors that drive the bootstrap node act as pass devices, and hence dissipate power only due to their V_T drop. The input pass transistor, like the isolation transistor (Fig 9), may switch with voltage across it, but in this circuit the capacitive load may be made smaller than that of the clock-pass transistor. (However, the capacitance of this storage node may need to be augmented since one is trying to store charge on the gate of a transistor while the source and drain are both at V_c, and to retain the charge after the source and drain switch to 0.) Another piece of good news about this circuit is that – unlike the clocked-isolation clock-AND – the bootstrap node is actively held at 0V during $\phi 1$ when the output pulse is not to appear. We need have no anxiety that some process parameter variation will allow the output pulse to appear when we do not mean for it to.

The only possible pitfall in using this circuit is that the output is not actively driven to 0 during $\phi 1$. Thus if there is a potential for charge sharing of a 0 output during $\phi 1$, one may add a pulldown network to the circuit in Figure 14:

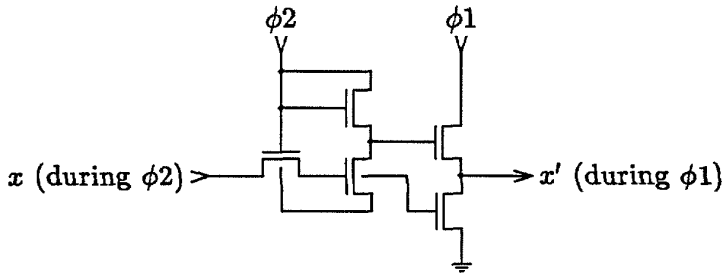

Figure 15

If ground is available on the chip, it should be used as shown in order to augment the storage capacitance. Otherwise, since the clock-pass transistor provides a path to ground during $\phi 2$, a path to ground during $\phi 1$ can be provided on the drain of an extra transistor whose gate is connected to $\phi 1$ and whose source is connected to $\phi 2$. In the form shown in Figure 15, the circuit output is always driven, and the circuit can be composed safely with itself. The same technique producing a complement by diode-charging on one clock phase, with a conditional discharge during the non-overlap period, can also be used to drive the pulldown network. Thus a non-inverting clocked-isolation clock-AND can be made to have this same property of the output being always driven:

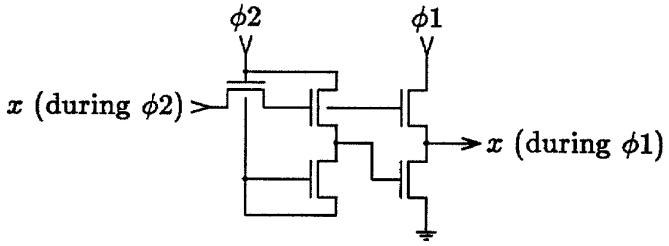

Figure 16

Here again, ground may be used if available, or a path to ground can be generated in a variety of ways from the clock signals.

The discovery of progressively more devious hot-clock nMOS circuits goes on and on, but this paper does not. The interested reader will have no difficulty, and perhaps even some fun, inventing more. Consider, for example, the following final fancy hot-clock circuit, an elaboration on the inverting clock-AND to compute not just the complement, but any switching expression that is negative in all of its inputs:

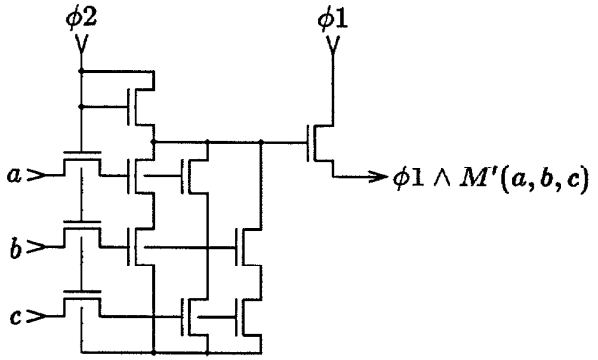

Figure 17

How might the output of this circuit be made to provide an active path to 0 during $\phi 1$? This switching element, while it provides both storage and logic functions, is not (quite) universal by itself, because the functional dependence on an input that enters the network on, say, $\phi 2$ is always negative on $\phi 1$ outputs and positive on $\phi 2$ outputs. For inputs a, b that enter in the same phase, it would not be possible to compute a switching function that is positive in one variable and negative in the other, such as $a \wedge b'$. The combination of inverting and non-inverting clocked-isolation

clock-ANDs is formally universal, and in combination with pass networks has proved to be an effective design paradigm.

4. Applications and Conclusions

MOSIS has fabricated about two hundred Mosaic processors [4], in fact, MOSIS people have used layouts we have assembled for them over a range of feature sizes for yield characterizations of different processes. These processors use the elementary clock-AND circuits extensively; there is hardly a control signal to be found in the Mosaic processor that is not driven by a clock-AND. The observed yields on these processors are indistinguishable from those of other chips of comparable complexity and design refinement. Our students use them with similar results. They have been used successfully in the Pixel-Planes chips [6]. Thus we regard the elementary clock-AND as fully qualified for MOSIS *n*MOS processes.

The Mosaic processors fabricated on 3μm MOSIS runs operate at a clock frequency of about 18 MHz, apparently limited by the pad frame and test jig. One clock period in this machine includes, in parallel, one storage cycle and one microcode cycle, while the datapath performs sequentially an arithmetic operation and a bus transfer. Thus we regard this performance as quite respectable for the technology used. We have observed little variation in performance in the chips from one MOSIS run to another. In designs that use depletion pullups in critical paths we observe the usual sensitivity of speed to the depletion threshold variations.

A array multiplier that uses a quaternary number representation internally was the first of our chips to be designed entirely without depletion devices, and entirely clock-powered. Circuit simulations of the critical paths predicted a 10 MHz throughput, but a complete chip has not yet been fabricated.

The Mosaic RAM, which is described briefly in [4], was the next chip to apply this technique to the limits, indeed, to limits that at the time of its design exceeded our understanding of the tolerance of these circuits to variations in process parameters. This chip includes instances of all of the "fancy" circuits described in section 3. We did, however, receive working silicon for this RAM from a MOSIS run with a fairly large t_{ox} of 65 nm and a very large V_T of 1.2V. Earlier we had received one run of chips that could read but not write, and from SPICE simulations and from studying the misbehavior had isolated the problem to three different clocked-isolation clock-ANDs (figure 9) that were producing an output pulse with a 0 input. Additional capacitance on the bootstrap node is expected to adjust the input switching threshold on these circuits to

provide a tolerance to variation in the ratio of gate capacitance to drain overlap capacitance over about a 2.5 : 1 range.

SPICE simulations of the critical paths of the Mosaic RAM predict operation at 3μm feature size in excess of 20 MHz, and at an on-chip dissipation level of a few milliwatts per 4K-bit section. However, the chips from the one "successful" 4μm run operate only up to 7 MHz, a result that we would like to attribute to marginal operation of certain circuits.

We can readily envision a future *n*MOS process, simplified by the omission of depletion transistors, augmented with additional metal layers, and perhaps including zero-threshold transistors, that would give CMOS a good run for its money. Such a technology used together with resonant clock drivers would provide a substantial gain in the relationship between performance and power dissipation. We believe that such a technology and the hot-clock design style would be particularly suitable for memories, computational arrays, microcomputer arrays, and other structures based on repetitions of one complex chip type (§III.B in [7]).

5. Acknowledgments

We have enjoyed and benefitted from many interesting discussions about "hot-clocking" with our Caltech colleagues Alain J Martin and Richard P Feynman. Special thanks to TEXperts Calvin Jackson and Wen-king Su for their help and nifty systems.

6. References

[1] Randal E Bryant, "A Switch-Level Model and Simulator for MOS Digital Systems", *IEEE TC*, vol C-33, no 2, pp 160-177, February 1984.

[2] James J Cherry, Gerald L Roylance, "A One Transistor RAM for MPC Projects", *Proc Second Caltech Conf on VLSI*, pp 329-341, Caltech Computer Science, January 1981.

[3] Reuben E Joynson, Joseph L Mundy, James F Burgess, Constantine Neugebauer, "Eliminating Threshold Losses in MOS Circuits by Bootstrapping Using Varactor Coupling", *IEEE JSSC*, vol SC-7, pp 217-224, June 1972.

[4] Chris Lutz, Steve Rabin, Chuck Seitz, Don Speck, "Design of the Mosaic Element", *Proc MIT Conf Advanced Research in VLSI*, pp 1-10, Artech Books, 1984.

[5] Carver A Mead, Lynn A Conway, *Introduction to VLSI Systems*, Addison-Wesley, 1980.

[6] John Poulton et al, "Implementation of a Full Scale Pixel-Planes System", *Proc 1985 Chapel Hill Conf on VLSI*, Computer Science Press, 1985, (this volume).

[7] Charles L Seitz, "Concurrent VLSI Architectures", *IEEE TC*, vol C-33, no 12, pp 1247-1265, December 1984.

[8] Clark D Thompson, "A complexity theory for VLSI", Dept of Computer Science, Carnegie-Mellon University, Technical Report CMU-CS-80-140, August 1980.