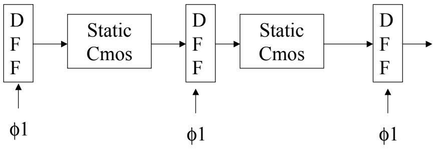
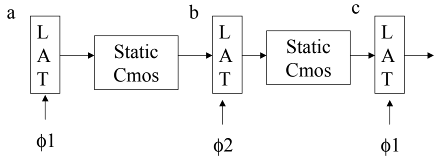
Pipelined DFF System (pulsed Latches) with Static CMOS



Clock period = Tcq + Tcl (max)

if Tsu (negative) > Tskew

Pipelined Latch System with Static CMOS

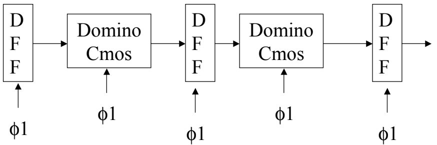


Clock Period = 2* Tc2q + Tcl (max path)

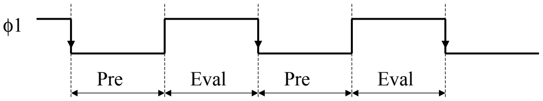
If clock phase overlap, then:

Tcl(min path) > Tskew – Tcq – Tsu (to avoid racethru)

Domino Logic + FF system

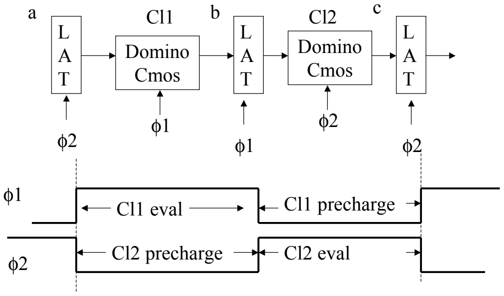


Assume falling edge triggered, pulsed latches



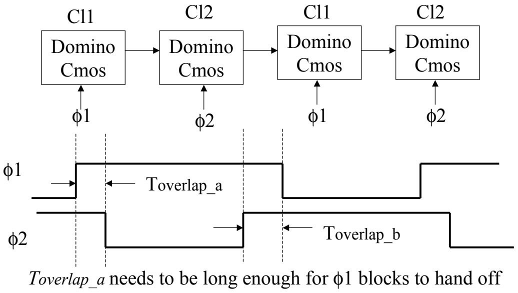
A poor match, we are wasting time doing precharge. If Domino block same evaluation time as Static block, then slower than Static CMOS. Precharge time adds to clock period.

Domino Logic + Latch system



Clock period = 2* Tc2q + Tcl (max path), same delay as static CMOS system. Notice that precharge time is hidden.

How can we improve this? Remove[†] Latches!!!



Toverlap_a needs to be long enough for $\phi 1$ blocks to hand off results to $\phi 2$ blocks ($\phi 2$ blocks consume $\phi 1$ results) and also to hide skew.

[†] Harris & Horowitz, "Skew-Tolerant CMOS Circuits", JSSC Nov 1997

Two Phase Overlapping Domino [†]

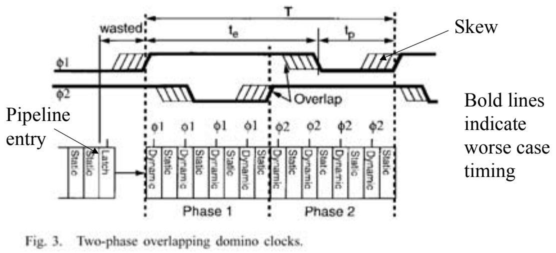

wasted
T
te
tp
Skew
φ1
φ2
Overlap
Pipeline entry
Bold lines indicate worse case timing
φ1 φ1 φ1 φ1 φ2 φ2 φ2 φ2
Static Static Latch
Dynamic Static Dynamic Static Dynamic Static Dynamic Static
Dynamic Static Dynamic Static Dynamic Static Dynamic Static
Phase 1
Phase 2

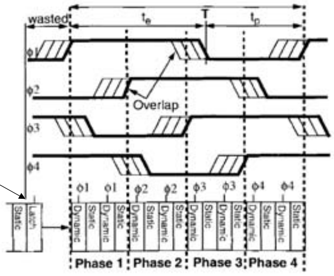
Fig. 3. Two-phase overlapping domino clocks.

T_e = evaluation time, T_p = precharge time

"Static" refers to the static inverter (or logic) in Domino gate

[†] Harris & Horowitz, "Skew-Tolerant CMOS Circuits", JSSC Nov 1997

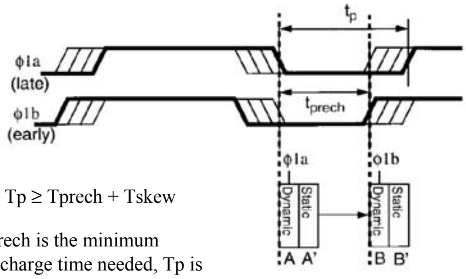
Four Phase Overlapping Domino †



Pipeline entry

†Harris & Horowitz, "Skew-Tolerant CMOS Circuits", JSSC Nov 1997

Precharge Constraint †

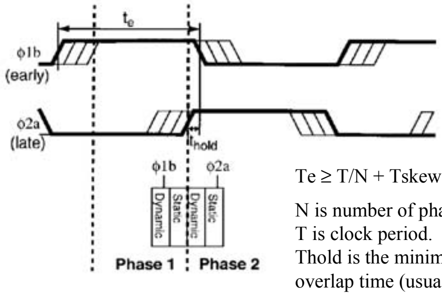


Tp ≥ Tprech + Tskew

Tprech is the minimum precharge time needed, Tp is actual time needed with skew consideration.

†Harris & Horowitz, "Skew-Tolerant CMOS Circuits", JSSC Nov 1997

Evaluation Constraint †



Te ≥ T/N + Tskew + Thold

N is number of phases.
T is clock period.
Thold is the minimum overlap time (usually a small negative number).

†Harris & Horowitz, "Skew-Tolerant CMOS Circuits", JSSC Nov 1997

Maximum Allowable Skew †

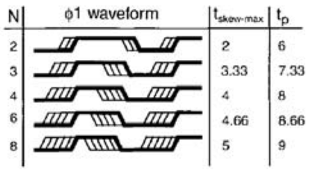
Know that T = Tp + Te, so can combine the two constraints and solve for Tskew max:

$$\text{Tskew\_max} = \frac{(N-1)/N * T - \text{Thold} - \text{Tprech}}{2}$$

Thold is typically a small negative number since 1st gate of next stage evaluates immediately, while precharge of last gate must ripple thru gate and the static inverter.

†Harris & Horowitz, "Skew-Tolerant CMOS Circuits", JSSC Nov 1997

Maximum Allowable Skew †



N	ϕ1 waveform	$t_{skew-max}$	t_p
2		2	6
3		3.33	7.33
4		4	8
6		4.66	8.66
8		5	9

Fig. 7. Basic skew tolerance (in FO4 inverter delays).

Assumes Thold = 0, T = 16 FO4 delays. Note that skew tolerance goes up as number of phases increases.

†Harris & Horowitz, "Skew-Tolerant CMOS Circuits", JSSC Nov 1997

Global versus Local Skew

Local skew (skew within a region, between nearby elements) is more tightly bounded than global skew (skew between arbitrary elements)

Designate two skews as Tskew_global, Tskew_local.

Precharge only depends upon Tskew_local because precharge only must complete before the next gate in the same phase resumes evaluation.

Evaluation dependent on global skew since clocks from different domains must overlap.

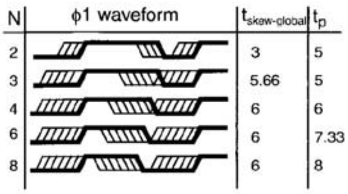
Tskew_max_global

Use Tskew_local in Tp equation, Tskew_max_global in Tp equation, solve for Tskew_max_global.

Tskew_max_global = (N-1)/N*T – Thold – Tprech – Tskew_local

Assumes Local Skew = 1 FO4

Improved Global Skew results



N	ϕ1 waveform	$t_{skew\text{-}global}$	t_p
2		3	5
3		5.66	5
4		6	6
6		6	7.33
8		6	8

Fig. 8. Global skew tolerance (in FO4 inverter delays).

Time Borrowing

Excess overlap can be used for time borrowing

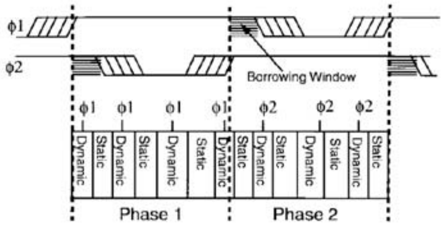
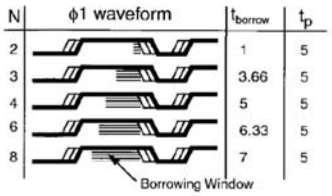


Fig. 9. Time borrowing.

Time Borrowing

Tborrow = Tglobal_skew_max – Tglobal_skew_actual



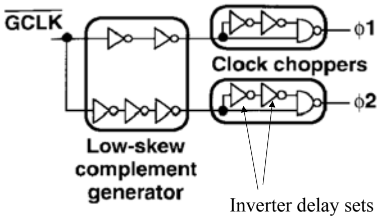
N	ϕ1 waveform	t_{borrow}	t_p
2		1	5
3		3.66	5
4		5	5
6		6.33	5
8		7	5

Borrowing Window

Fig. 10. Time borrowing availability (in FO4 inverter delays).

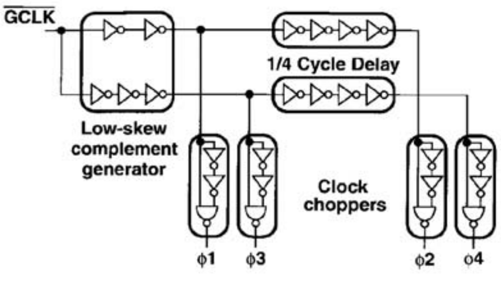
Assumes Acutal Global Skew = 2, Local Skew = 1, Tp = 4, Thold = 0, T = 16. Again, more phases = more borrowing.

Two Phase Clock Generation

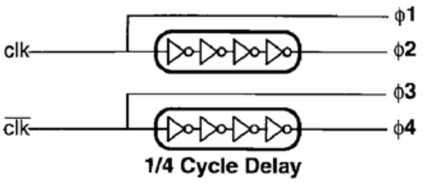


Inverter delay sets asymmetry of clock phases.

Four Phase Clock Generation



Four Phase Clock Generation (Simplified)



Local phase generators should serve a small enough area such that skew from wiring is load (radius < 2 mm).

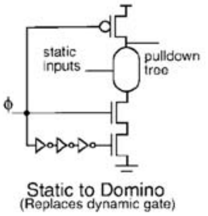
Race-thru (Min-delay failure)

If overlap is too large, or delay through evaluation logic too small, can get a race-thru failure (see slides on Race-thru in overlapping clocks in earlier discussion).

Need at least one gate in each logic block.

Cannot have an input from one phase skipping a phase entirely (ie. an output from $\phi1$ cannot feed a $\phi3$ input).

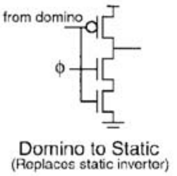
Domino Interfaces



Simple Pulsed Latch

Can place weak cross-coupled inverters on output to support stop-clock operation.

Domino Interfaces



Transparent latch.

Can place weak cross-coupled inverters on output to support stop-clock operation.

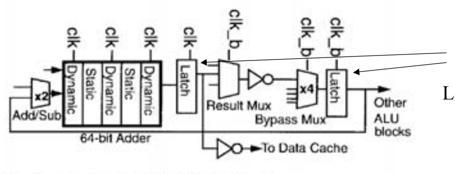
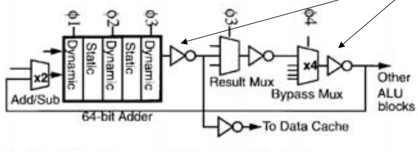


Fig. 16. ALU self-bypass path (textbook domino).

Latches



Inverters

Fig. 17. ALU self-bypass path (skew-tolerant domino).

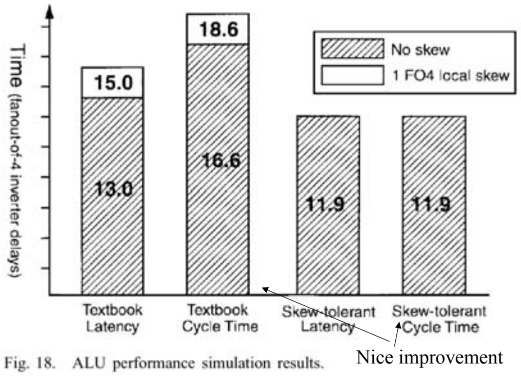
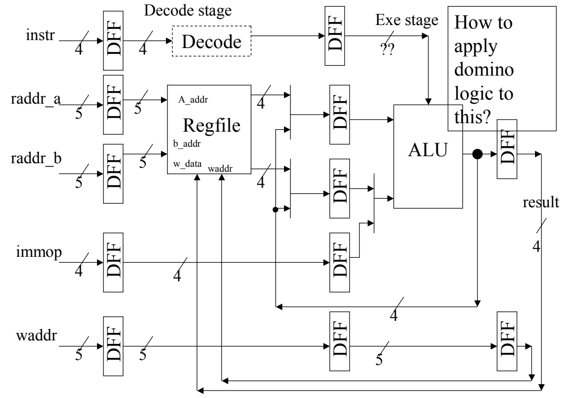
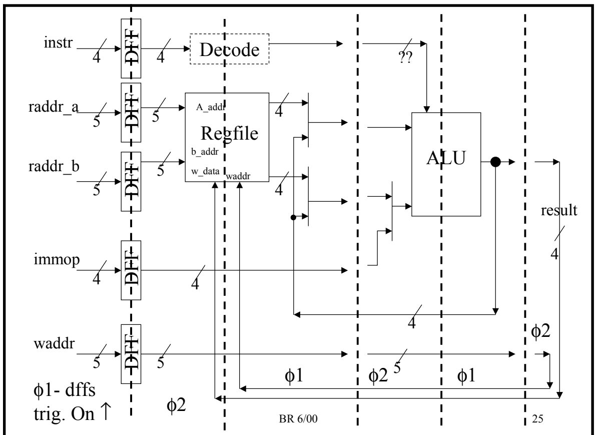


Fig. 18. ALU performance simulation results.

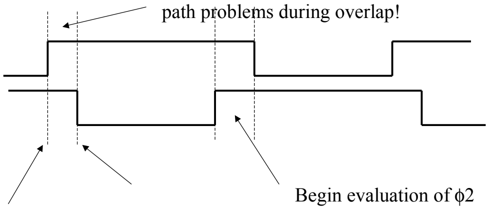
Nice improvement



How to apply domino logic to this?



MUST worry about shortest path problems during overlap!



New values provided by decode stage DFFs

Old results from idecode ϕ2 being latched.

Begin evaluation of ϕ2 idecode. Register file write done during ϕ2 , register file read done during ϕ1 evaluation.

Other Comments on Pipeline ALU example

- Any net connections show going from ϕ1 to ϕ2 (and vice versa) must have logic on it, even if it is a buffer, to prevent shortest path problems.
- ALU 2nd stage is ϕ1 and it must drive the ϕ1 logic blocks of the decode stage for bypass to work correctly (there is no intervening ϕ1 logic)
- The registered output of the ALU is ϕ2 logic, and it drives the ϕ2 logic of the decode stage.