

Lecture 13

Why dynamic Logic

- It occupies less area.....compare to static CMOS
- It has higher speedcompare to static CMOS
- It has less powercompare to static CMOS

Why not

- Requires a clock
- Can not operate at low speed
- It is affected by charge sharing
- Circuits are more sensitive to timing errors and noise
- Design is more difficult

An introduction to dynamic logic

1. Dynamic logic

In static logic families the pull up and pull down networks operate concurrently. Dynamic logic on the other hand uses a sequence of precharge and conditional evaluation phases governed by the clock to realize complex logic functions.

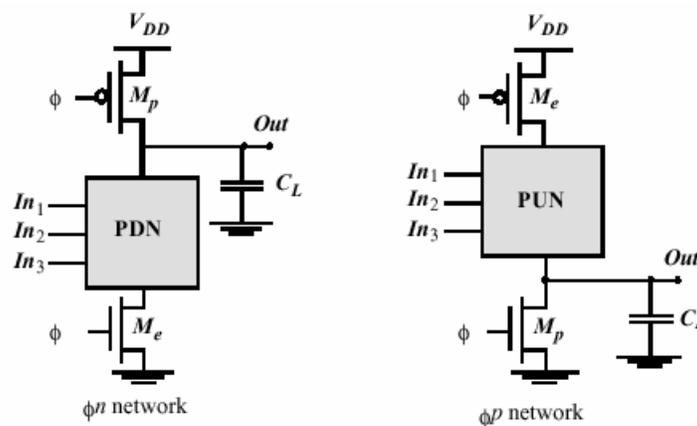


Figure 1 Dynamic logic

A dynamic logic block is shown in Fig. 1. Both forms of Fig.1 can be used.

In our analysis we will concentrate on Fig.1 n-logic network. The operation of the pull-down network (PUN) can be divided into two major phases. The precharge and the evaluation phase. In what mode the circuit is operating is determined by the signal ϕ , the “clock” signal.

Let us take an example of either network:

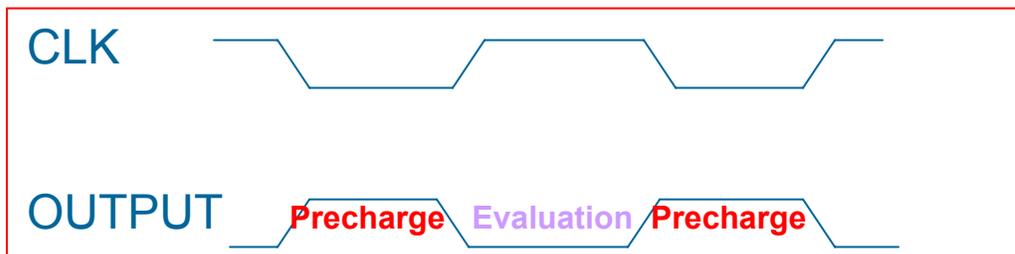
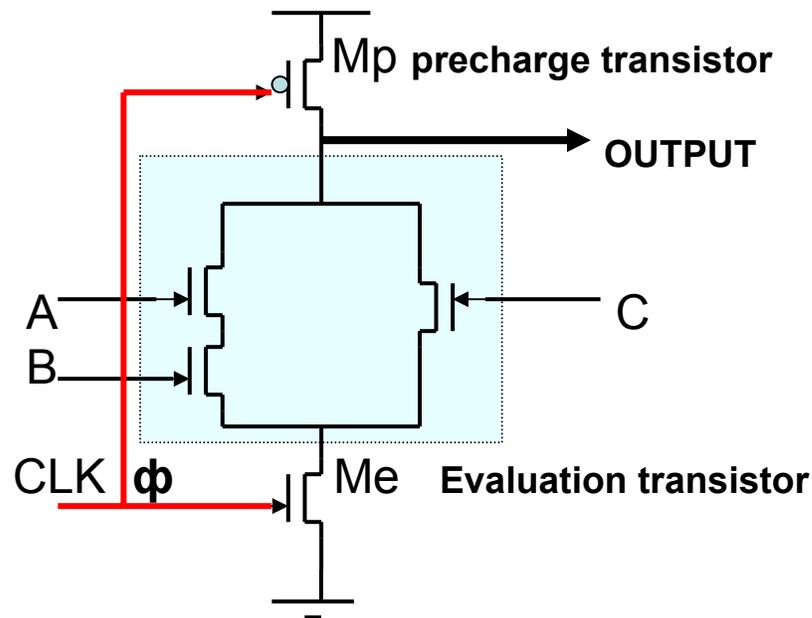


Fig. 2 Example of nmos block For $OUTPUT = (A.B + C)'$

Precharge

When $\phi = 0$, the output node "OUT" is precharged to V_{DD} by the PMOS transistor. During that time, the nmos evaluation transistor is off, so the nmos logic network is isolated from ground by a series of nmos transistors and hence no dc current flows regardless of the values of the input signal. Input signals can change with no effect to the output (except

for charge sharing to be discussed later). It is also difficult to control the change of variables during precharge.

Evaluation

When $\phi=1$, the precharge pull up transistor is off, and the evaluation transistor is turned on. Depending on the values of the inputs and the composition of the PDN, a conditional path between OUT and (through the nmos transistors) GND is created. If such a path exists, OUT is discharged and a low output signal is obtained. If not, the precharge value remains stored on the output capacitor C_L , (C_L is a combination of the diffusion capacitors, the wiring capacitance and the input capacitance of the fan-out gates) and a high output value is obtained during the evaluation phase. The only possible path between the output node and a supply rail is to GND, consequently, once “OUT” is discharged, it cannot be charged again. This is in contrast with the static circuit, where the output node is low-impedance under all possible circumstances. The inputs to the circuit can therefore make at most one transition during evaluation. The action of charging and discharging is shown in the following diagram

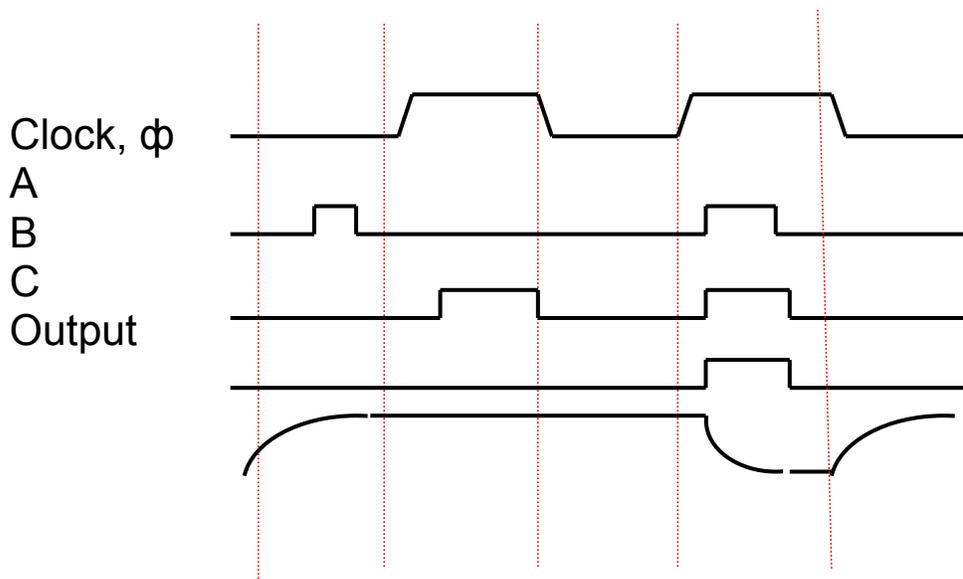


Fig. 3 Input/Output relations of Fig. 2

Current leakage will limit the time that the charge may be stored on the output node. Clock frequency must be sufficient not to allow the charge to degrade.

The ϕ_p network block is very similar with a ϕ_n network block. It consists of an NMOS precharge transistor and a PMOS pull-up network, (PUN). The PUN block is precharged (to GND) during $\phi=1$ and evaluations during the $\phi=0$ phase. Due to the lower mobility of the PMOS devices, a ϕ_p block is slower than a ϕ_n block.

Dynamic circuits have a variety of properties for example for the n-network:

- Dynamic logic has higher speed than equivalent static family.
- It occupies less area. The number of transistors is lower than in the static case: $N+2$ versus $2N$.
- The NMOS pull-down network implements the logic function. The construction of the PDN proceeds just as it does for static CMOS and pseudo-NMOS.
- It is non-ratioed. The noise margin does not depend on transistor ratios, as is the case in the pseudo-NMOS family.
- It has low power dissipation. It only consumes dynamic power. No static current path ever exists between V_{DD} and GND.
- Due to the reduced number of transistors in the circuit and the single transistor load per fan-in, the load capacitance for the circuit is substantially lower than for static CMOS. This results in faster switching speeds.
- Dynamic logic always require clock.
- It is impossible to operate the Dynamic logic at low speeds.
- Dynamic circuits are more sensitive to noise and timing errors.
- Dynamic logic is affected by charge sharing.
- Design is more difficult.

2. Steady-State Behavior of Dynamic Logic

In our discussion we are concentrating on the n-logic block. Similar considerations hold for the P-logic block. The low and high output levels V_{OL} and V_{OH} are easily identified as GND and V_{DD} and do not depend upon the transistor sizes. The voltage transfer characteristic parameters are dramatically different from the static circuit. Noise margins and switching thresholds have defined static quantities, which are not influenced by time. To be functional, a dynamic circuit requires to be refreshed. Pure static analysis, therefore, does not apply. Quantitative analysis can be made for example the value of the noise margins is a function of the length of the evaluation period. If the clock period is too long, the high output level is severely affected by charge leakage. On the other hand extending the evaluation period results in a lower value of V_{OL} .

The pull-down network of a dynamic circuit starts to conduct when the input signal exceeds the threshold voltage of the NMOS pull-down transistor. If one waits long enough, the output eventually reaches GND.

In the high output state, the output impedance of the gate is very high, since the output node is floating. Hence the output level is sensitive to noise and disturbances.

3 Performance of Dynamic Logic

The most attractive property of dynamic gate is its high switching speed. The reduced number of transistors result in a small value for load capacitance C_L . After the precharge phase, the output is high. For a low input signal, no additional switching occurs. The high to low transition requires the discharging of the output capacitance through the pull-down network. Therefore, t_{pLH} is proportional to C_L and the p transistor size and t_{pHL} is proportional to the current-sinking capabilities of the PDN. The presence of the evaluation transistor is an added resistance that slows the gate somewhat. During the precharge the logic in the gate cannot be utilized.

The length of precharge time can be adjusted by changing the size of the PMOS precharge transistor. By making the PMOS transistor larger the required precharge time

can be decreased. But making the PMOS transistor too large should be avoided because it both slows down the gate and increases the capacitor's load on the clock line.

4. Noise Considerations in Dynamic Design

The dynamic circuit concept results in simple and fast structure at the expense of a reduced robustness with regards to noise. The logic has a number of deficiencies that must be dealt with to guarantee functional operation.

- **Charge leakage**

The operation of the dynamic logic depends on the principles of dynamically storing a charge on the output node (capacitor). Due to leakage currents, this charge gradually leaks away, resulting eventually in malfunctioning of the gate.

Charge leakage causes degradation in the logic high level. Dynamic circuits require a minimal clock rate which is typically between 250Hz and 1 kHz.

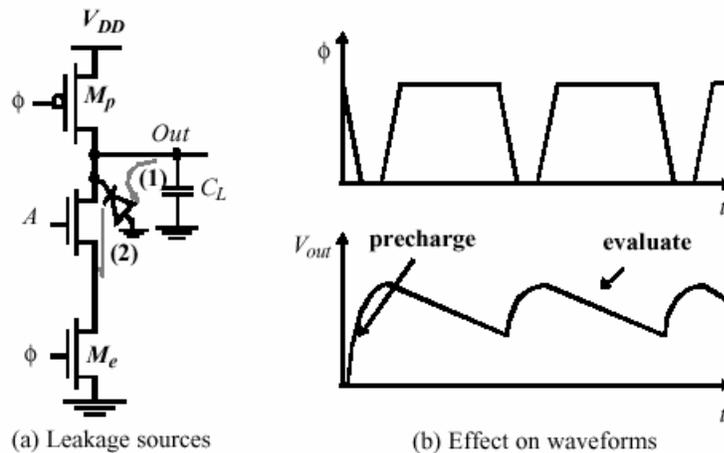


Figure 4 Charge leakage

Charge sharing

Consider the circuit of Figure 5. During the precharge phase, the output node is precharged to V_{DD} . Capacitors C_a and C_b represent the parasitic capacitances of the internal nodes of the circuit. Assume now that during precharge all inputs are set 0 and

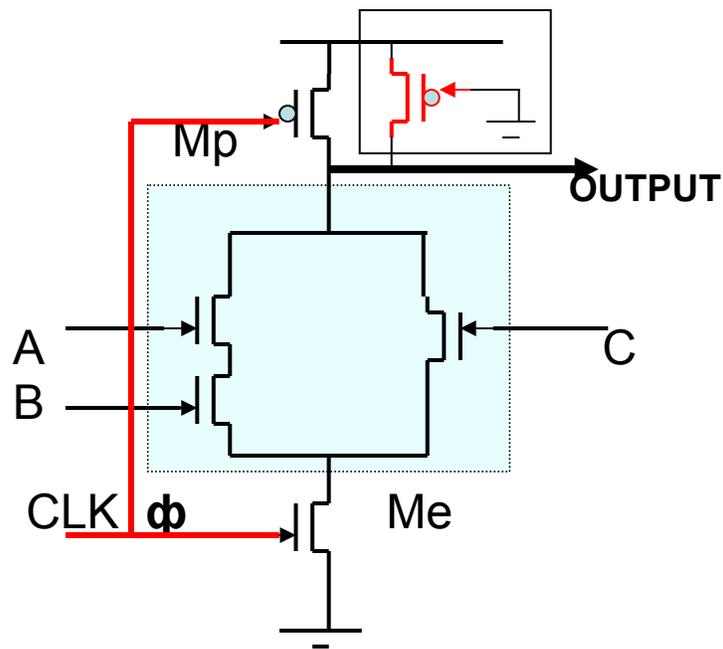


Fig. 6 Addition of level restoring transistor

- **Clock feedthrough**

The clock signal is coupled to the storage node by the gate-source capacitance and the gate-overlap capacitance of the precharge device as shown in Fig. 7. The fast rising and

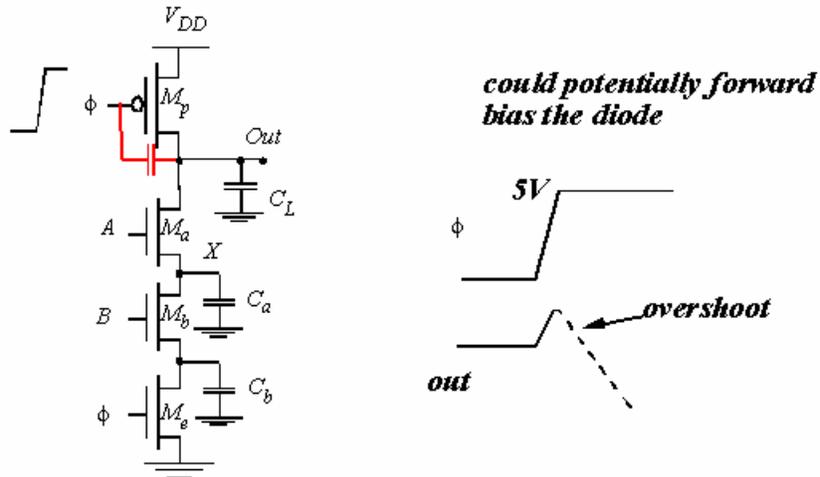


Figure 7 Clock feed-through

falling edges of the clock couple into the signal node. The danger of clock feed-through is that it causes the signal level to rise sufficiently above the supply voltage as to forward-bias the junction diode. This causes electron injection into the substrate and eventually resulting in faulty operation. This can be avoided by providing a sufficient number of well contacts close to the precharge device to collect the injected current.

- **Cascading dynamic gates**

Cascade dynamic gates of the type described will result in malfunction of the circuit. Consider two simple inverters connected in series, as shown in Figure 8. The problem is that during precharge all outputs are being precharged to 1. The PDN of the second gate is thus in a conducting state at the onset of the evaluation phase. Suppose now that “In” makes a 0 to 1 transition. At the onset of the evaluation period ($\phi=1$), output “Out1” starts to discharge. When “Out1” exceeds the switching threshold of the second gate, a conducting path exists between “Out2” and GND. “Out2” therefore discharges as well, and wrongly so, as the correct output of the gate equals 1. This conducting path is only turned off when “Out1” reaches V_{Tn} and shuts off the NMOS pull-down transistor. This leaves “Out2” at an intermediate voltage level. The charge loss leads to reduce noise

margins and eventual malfunction. It is obvious that the cascading problem arise because the output is precharged to “1”, so correct operation is guaranteed as long as the inputs can only make a signal 0 to 1 transition during the evaluation period.

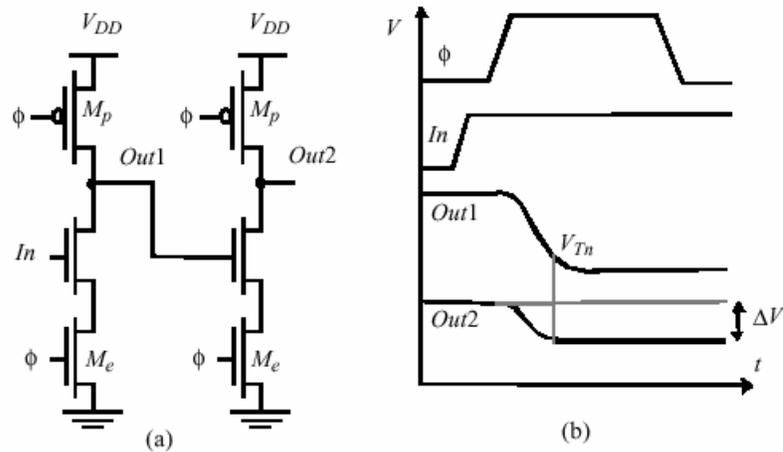


Figure 8 Cascading dynamic gates

The Domino and np-CMOS logic families solve the above problems.

- **Dynamic power consumption**

The average power consumption of a circuit is defined as:

$$P_{avg} = \frac{V_{DD}}{T} \int_0^T i_{DD}(t) dt$$

T is the period of interest, and V_{DD} and i_{DD} the supply voltage and current respectively.

We can calculate P_{avg} by HSPICE.

- **Current spiking during transients**

In reality, the assumption of the zero rise and fall times of the input wave form is not correct. As a result, a direct current between V_{DD} and GND exists for a short period of time during switching, while the NMOS and PMOS transistors are conducting

simultaneously. It is determined by the saturation current of the devices and is hence directly proportional to the size of the transistors. To minimize it, try to reduce t_f and t_r .

5. Domino logic

Domino logic offers a simple technique to eliminate the need for complex clocking scheme, by utilizing a single phase clock. A Domino logic module consists of a ϕ_n block followed by a static inverter as shown in Fig. 9. This ensures that all inputs to the next logic block are set to 0 after the precharge periods. Hence, the only possible transition

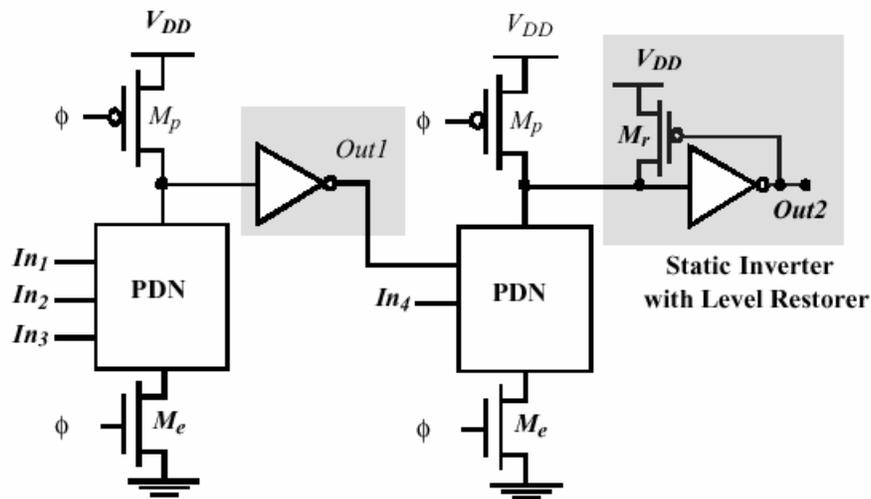


Figure 9 The block of Domino logic

during the evaluation period is 0 to 1 transition. The introduction of the static inverter has the additional advantage of the output having a low-impedance output, which increases noise immunity and drives the fan-out of the gate. The buffer furthermore reduces the capacitance of the dynamic output node by separating internal and load capacitance. The buffer itself can be optimized to drive the fan-out in an optimal way for high speed. The inverter will introduce another problem that this type of logic family is non inverting.

Consider now the operation of a chain of Domino gates. During precharge, all inputs are set to 0. During the evaluation, the output of the first Domino block either stays at 0 or makes a 0 to 1 transition, affecting the second Domino. This effect might ripple through the whole chain, one after the other, as with a line of falling dominoes. Domino CMOS has the following properties:

- Each gate requires $N+4$ transistors
- Logic evaluation propagates as falling dominoes hence minimum evaluation period is determined by the logic depth.
- The nodes must be precharged during the precharge period. Total precharge time depends on size of pmos.
- Inputs must be stable (only one rising transition) during the evaluation period.
- Gates are ratioless and are non inverting.

Domino gates can be made more immune to parasitic effects by adding a level-restoring transistor to the static CMOS inverter.

P-E logic

The problem of non inversion logic and the static inverter with Domino logic can be solved by using alternate logic. That is the combinational circuits are constructed from

alternating nmos and pmos blocks. This ensures that 0 to 1 transition occur during precharge.

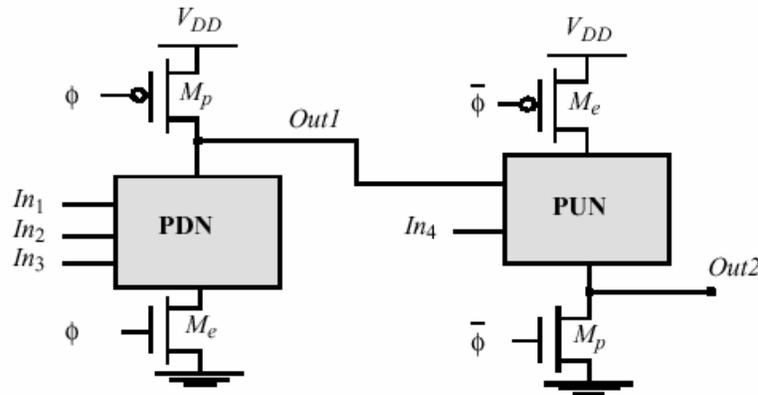


Figure 10 The block of P-E logic

The precharge output value of ϕ_n block equals 1, which is the correct value for the input of a ϕ_p block during precharge. All PMOS transistors of the Pull-Up Network (PUN) are turned off thus an erroneous discharge at the onset of the evaluation phase is prevented. In similar way, a ϕ_n block can follow a ϕ_p block without any problem, as the precharge value of inputs equals 0. To make the evaluation and precharge times of the ϕ_n and ϕ_p block coincide, one has to clock the ϕ_p block with an invert clock ϕ' . The block of P-E logic is shown in Figure 10.

The use of the pmos gates will slow down the logic due to lower mobility of the pmos. Or alternatively the pmos transistors have to be larger than the nmos. The clock also must be routed with its complement. Compared to Domino logic, P-E CMOS is more than 20% faster due to the elimination of the static invert and the smaller load capacitance.

8 NORA Logic

Combining C²MOS pipeline register and P-E CMOS dynamic logic function block, NORA-CMOS (mean NO-Race) is obtained. The logic family targets the implementation of, pipelined datapaths. Each module consists of a block of combinational logic that can

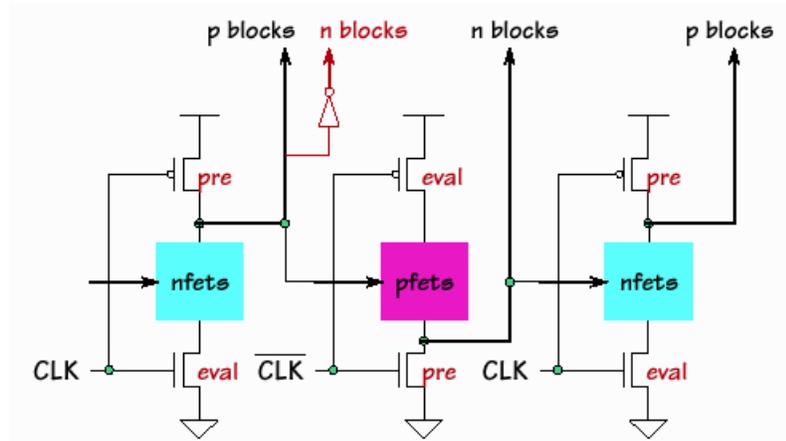


Figure 11 The block of NORA logic

be a mixture of static and dynamic logic, followed by a C²MOS latch. Logic and latch are clocked in such a way that both are simultaneously in either evaluation, or hold (precharge) mode. A block that is in evaluation during $\phi = 1$ is called a ϕ --module, while the inverse is called ϕ' module. A NORA datapath consists of a chain of alternating ϕ and ϕ' modules. While one class of modules is precharging with its output latch in hold mode, preserving the previous output value, the other class is evaluating. Data is passed in a pipelined fashion from models to module. The resulting datapath combines high performance with high layout density.

9. 2-Phase Logic

Two-phase registers are usually replications of single-phase structures with ϕ_1 feeding the master and ϕ_2 feeding the slave of the register. A reduction in the number of clock lines can be accommodated if only n-MOS design is used. Two effects occur in the configuration. First, the '1' level transferred to the input of the inverter is degraded to approximately $V_{dd} - V_{tn}$. This has the effect of slowing down the low transition of the inverter. Furthermore, the high-noise margin of the inverter is degraded.

The operation is described as follows: the first stage is precharged during ϕ_1 and evaluated during ϕ_2 . While the first stage is evaluated, the second stage is precharged and the first-stage outputs are stored on the second-stage inputs. During ϕ_1 , the second stage is evaluated and latched in a succeeding ϕ_1 stage. Fig. 12 shows the architecture of this logic family.

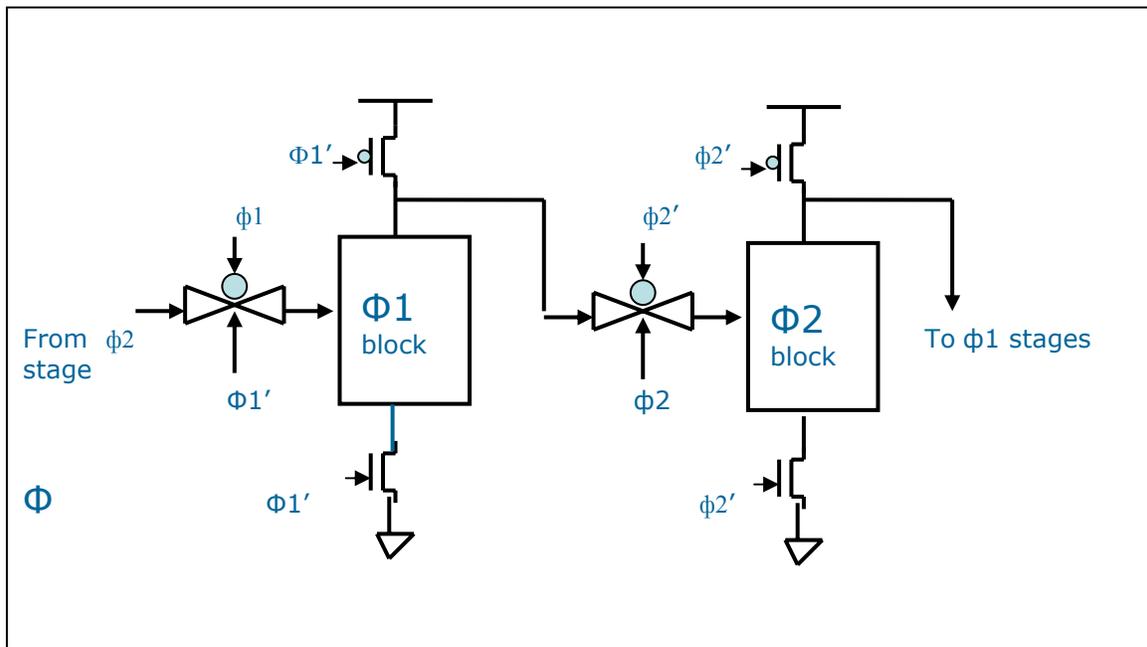


Fig. 12 2-Phase logic

Domino n-MOS gates may also be employed. A typical gate is shown in Figure 10. Here, a single clock ($\phi 1$ or $\phi 2$) is used to precharge and evaluate the logic block. The succeeding stage is operated on the opposite clock phase, as illustrated in Figure 10. The difference between this logic structure and that previously shown in Figure 9 is that in the domino logic, a number of logic stages may be cascaded before latching the result.

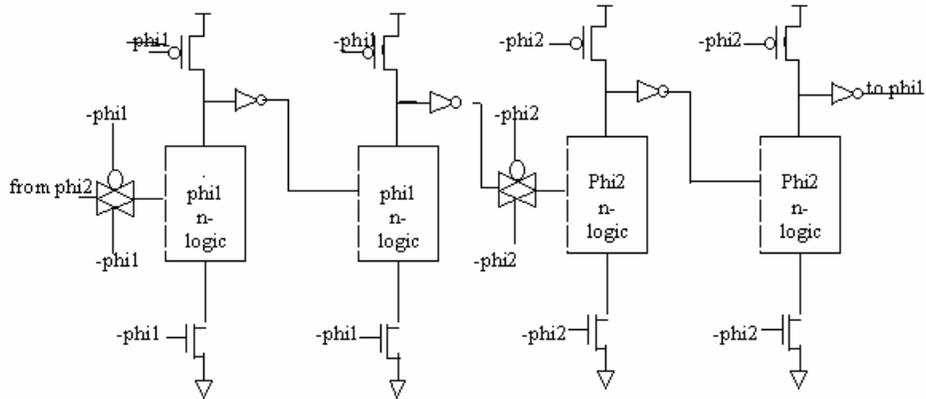


Figure 13 The block of two-phase dynamic logic using domino logic

10. Multiple O/P Domino Logic

Hwang and Fisher formally introduced MODL in 1989, and it has been shown to provide considerable hardware savings. In domino CMOS logic, as well as other non complementary MOS logic styles, there is only one output available from a given logic gate. However, it is a fact that multiple functions are often implemented in the logic tree with one being a sub-function of another. Therefore, if one or more of these sub-functions are needed as separate output signals, they have to be implemented in several additional gates, resulting in a replication of circuitry.

Basically speaking, the main concept behind MODL is the utilization of sub-functions available in the logic tree of domino gates, thus saving replication of circuitry. The additional outputs are obtained by adding precharge devices and static inverters at the corresponding intermediate nodes of the logic tree.

It is apparently area-efficient if we build the several outputs on only one tree; also it will not affect the speed of implementing all the functions.

Figure 11 shows the block of MODL for the following functions:

$$Q3=(C+D)$$

$$Q2=B.(C+D)$$

$$Q1=A.B.(C+D)$$

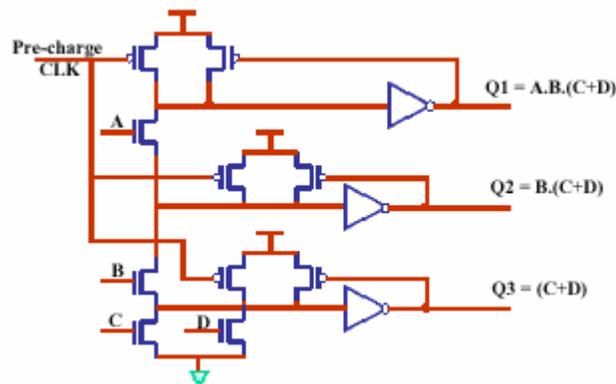


Figure 14 block of MODL

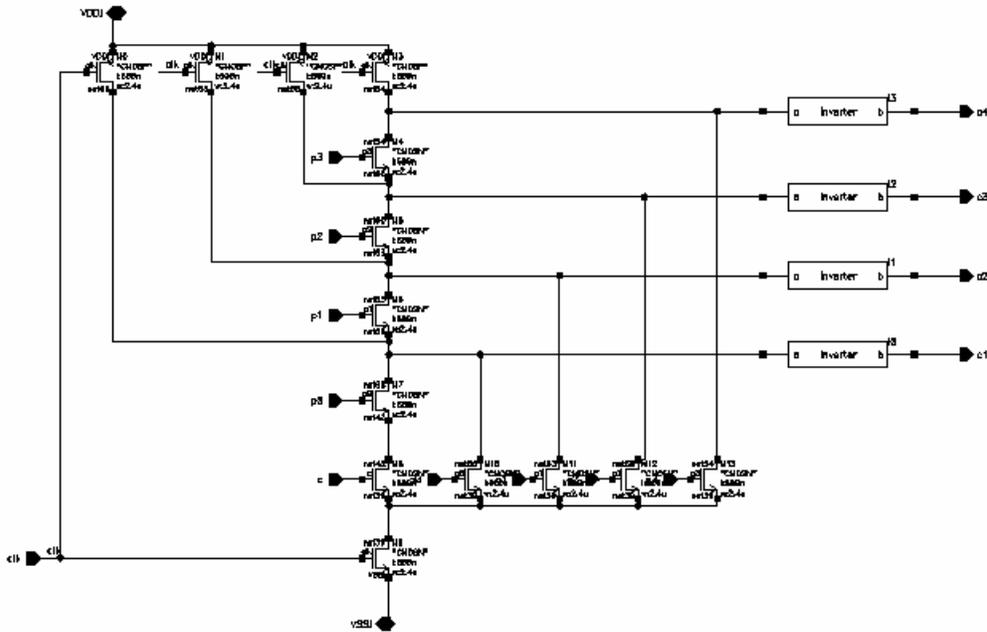


Figure 15 Schematic of Carry Generate Block

For four stages of lookahead, the appropriate terms are

$$C 1 = G 1 + P 1 C 0$$

$$C 2 = G 2 + P 2 G 1 + P 2 P 1 C 0$$

$$C 3 = G 3 + P 3 G 2 + P 3 P 2 G 1 + P 3 P 2 P 1 C 0$$

$$C 4 = G 4 + P 4 G 3 + P 4 P 3 G 2 + P 4 P 3 P 2 G 1 + P 4 P 3 P 2 P 1 C 0$$

11. Cascode Logic family

Further refinement leads to a clocked version of the CVSL gate as shown in Fig. 16. This is really just two “Domino” gates operating on the true and complement inputs with a minimized logic tree. The advantage of this style of logic over domino logic is the ability to generate any logic expression, making it a complete logic family. This is achieved at the expense of the extra routing, active area, and complexity associated with dualing-rail logic. However, the ability to generate any logic function is of advantage where automated logic synthesis is required.

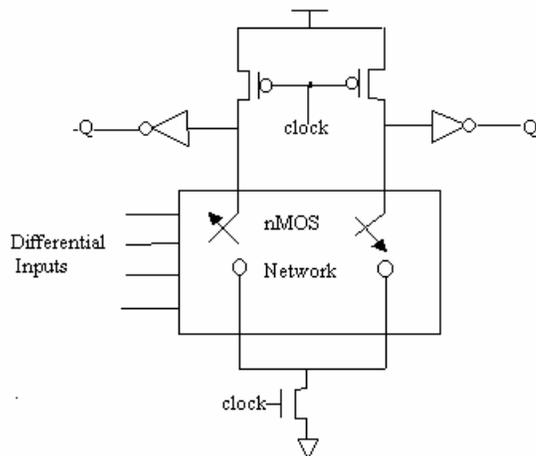


Figure 16 A clocked version of the CVSL gate

12. Conclusion

This logic style, as opposed to static, uses fewer transistors since it effectively replaces the whole PUN with a single precharge transistor, but adds another n-type transistor in the PDN. This would decrease the input capacitances to half, as well as the diffusion capacitance. The operation of the dynamic CMOS largely depends on the internal capacitances where the state of the input and output nodes are determined by the presence of charges on the capacitances of these nodes. This makes the dynamic gate very prone to noise and capacitive coupling during evaluation phase, and may result in a less robust circuit. Aside from these, we must also take into account several issues to insure the correct operation of the dynamic gate, which includes charge sharing, and clock feedthrough. There is also a design overhead due to the necessity of a clock. It is evident that dynamic circuits are relatively harder to design than static circuits.

13. Reference

- Neil H.E. Weste, “Principle of CMOS VLSI Design: A System Perspective”.
- A Tutorial on Using Cadence’s Cmosis5 Technology of Schematic Editor, Analog Artist and Virtuoso Editor”, Dept. of Electrical and Computer Engineering, Concordia University.
- Jan M. Rabaey, “Digital Integrated Circuits, A Design Perspective”
- Michael John Sebastian Smith, Application-Specific Integrated Circuits, Addison Wesley, 1997.
- Zhongde Wang, “Fast Adders Using Enhanced Multiple-Output Domino Logic”, IEEE JOURNAL OF SOLID –STATE CIRCUIT, VOL, 32 NO. 2 FEBRUARY 1997.
- Ahmed M. Shams, “Performance Analysis of Low-Power 1-bit CMOS Full Adder Cells”. IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, VOL. 10 NO.1 FEBRUARY 2002.
- JIREN YUAN, “High-Speed CMOS Circuit Technique”, IEEE JOURNAL OF SOLID –STATE CIRCUIT, VOL, 24 NO. 1 FEBRUARY 1989.
- Gustavo A. Ruiz, “Evaluation of three 32-bit CMOS adders in DCVS logic for self-timed circuits”. IEEE JOURNAL OF SOLID –STATE CIRCUIT, VOL, 33 NO. 4 APRIL 1998.

12. Appendix A

Comparison of Dynamic Logic Families

To compare the performance of the outlined logic families, 8-bit carry ripple adders were designed using nine circuits of the six dynamic logic functions. The designs were simulated. Performance such as power, area, delay, energy and AT^2 are obtained and compared.

- **Power**

The average power consumption of a circuit is defined:

$$P_{avg} = \frac{V_{DD}}{T} \int_0^T i_{DD}(t) dt$$

With T the period of interest, and V_{DD} and i_{DD} the supply voltage and current respectively. HSPICE was used to calculate P_{avg} .

Table 1 shows the power consumption of seven 8-bit adders.

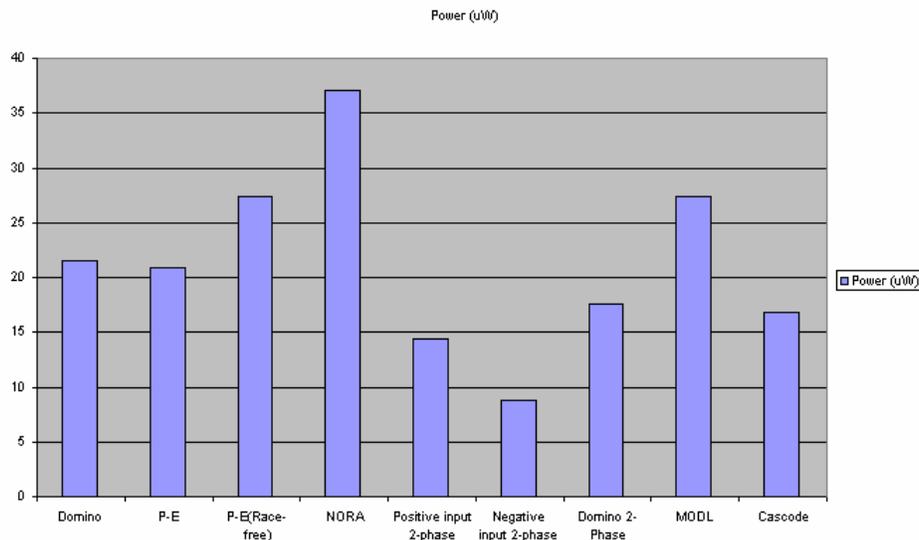


Figure 13 Power consumption of 8-bit Adders

Table 1 Power consumption of 8-bit Adders

	Power (uW)
Domino	21.5
P-E	20.8
P-E(Race-free)	27.3
NORA	37.1
+ve input 2-phase	14.4
-ve input 2-phase	8.7
Domino 2-Phase	17.5
MODL	27.3
Cascode	16.7

Area

In calculating the area, the following assumptions were made:

Transistor's drain and source length $L_s=L_d=1\mu\text{m}$, width $W=W_{\text{min}}$.

Transistor's area = Area_{gate}+Area_{drain}+Area_{source}

Table 2 shows the area of seven 8-bit Adders

Table 2 Area of 8-bit Adders

	Area(μm^2)
Domino	518.4
P-E	622.1
P-E(Race-free)	737.2
NORA	702.7
P-type 2-phase	748.8
N-type 2-phase	794.8
Domino 2-Phase	921.6
MODL	1195.2
Cascode	1774.1

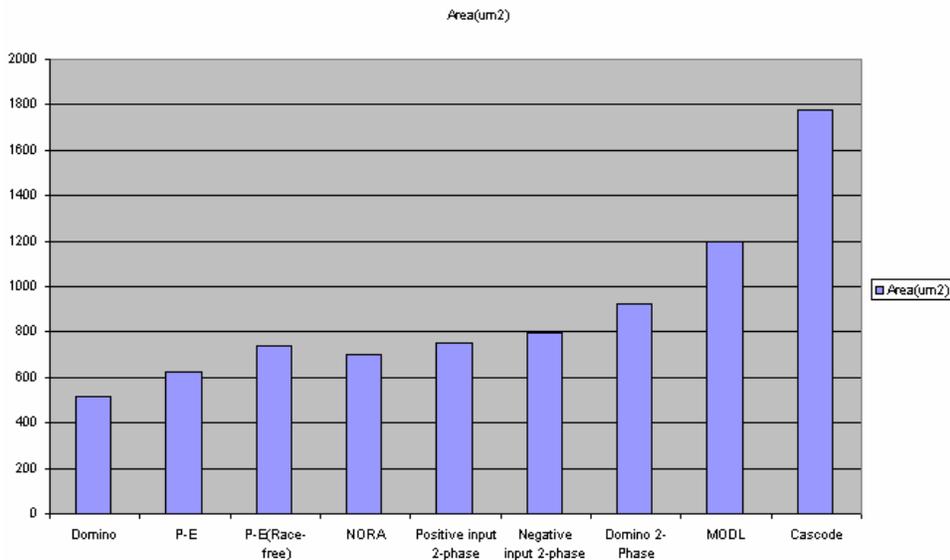


Figure 14 Area of 8-bit Adders

Delay

The Propagation delay was measured from the input application time to the steady output time. Table 3 shows delay of 8-bit Adders

Table 3 Delay of 8-bit Adders

	Delay(ns)
Domino	17.1
P-E	11.1
P-E(Race-free)	11.1
NORA	15.3
P-type 2-phase	9.1
N-type 2-phase	11.4
Domino 2-Phase	10.1
MODL	3.4
Cascode	11.1

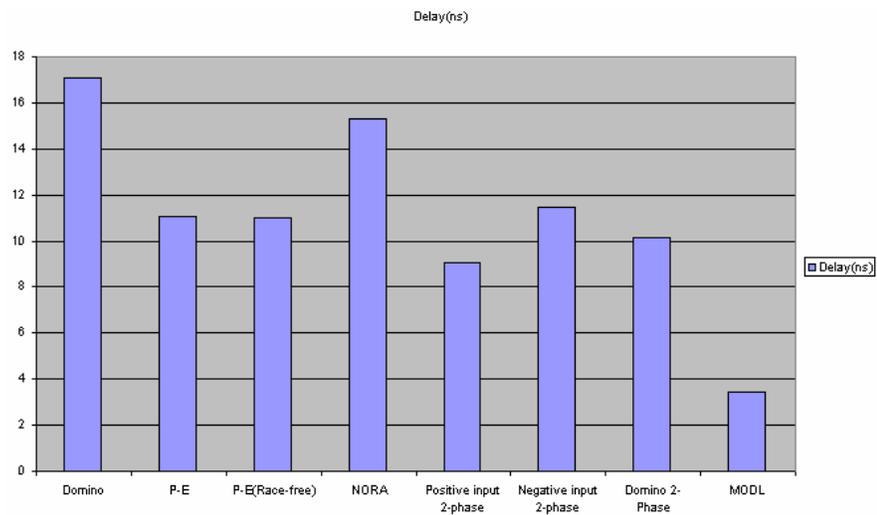


Figure 15 Delay of 8-bit Adders

- DP and AT^2

Table 4 shows the DP and AT^2 of 8-bit adders.

Table 4 DP and AT^2 of 8-bit Adders

	DP	AT^2
Domino	367.2	150938
P-E	230.6	75864
P-E(Race-free)	300.9	89224
NORA	568.1	164869
P-type 2-phase	130.7	156080
N-type 2-phase	100.4	61110.
Domino 2-Phase	178.2	94889.
MODL	94.6	14288.
Cascode	118.6	215620

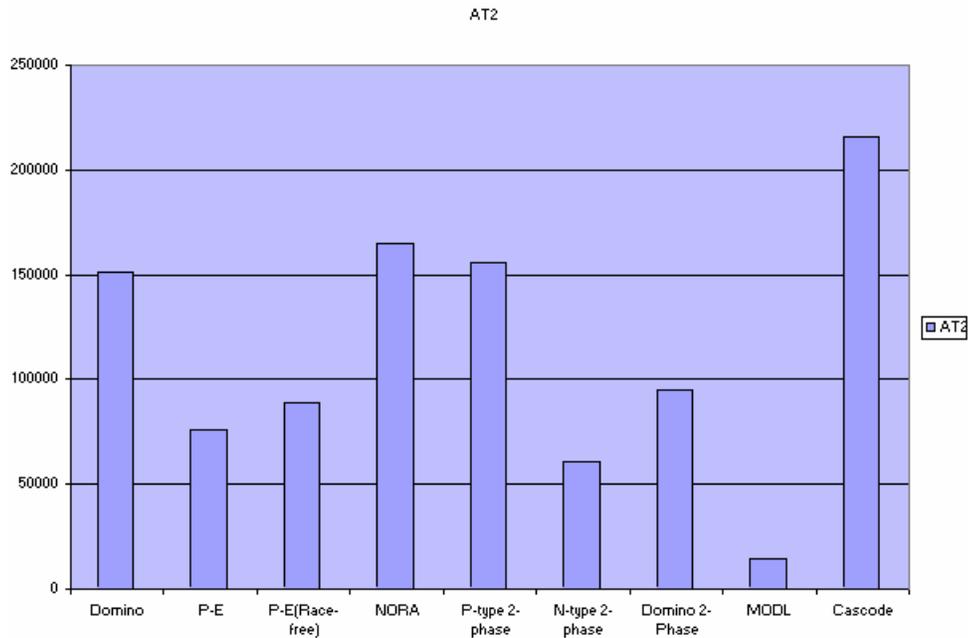


Figure 15 AT^2 of 8-bit Adders

Conclusion

Six logic structures, Domino, P-E, NORA, 2-Phase, MODL and Cascode, are used to design 8-bit dynamic adders. Their performance such as power, area, delay, energy and AT^2 are compared by simulation. The following conclusions can be derived from simulation results:

- Domino Logic: It has minimum area. The power consumption is lower, and the delay is the longest. The DP and AT^2 are average. If the design goal is minimum area and number of the transistors and the speed can be sacrificed a bit, then Domino logic is the best structure for Ripple Carry Adder.
- P-E Logic: It has a smaller area. The power consumption is lower, and the delay is shorter. It has the lower DP and AT^2 for Ripple Carry Adder. If the logic has no inherent race problem, it will be the better choice for Ripple Carry Adder.
- P-E (race-free) Logic: In order to avoid the race condition of P-E Logic, the P-E (race-free) Logic is introduced. It has a small area. The area is larger than P-E logic. The power consumption is average. The delay is shorter. It has lower DP and AT^2 for Ripple Carry Adder. For synthesis, it is the better choice for Ripple Carry Adder.
- NORA Logic: The power consumption is the highest. The area is small. The delay is longer. The DP is highest and AT^2 is higher. The performance of this dynamic logic is so average!.
- 2-Phase Logic
 - (1) P-type 2-Phase Logic: The area is small. The delay is shortest in Ripple Carry structure. The power consumption is lower than other logic structure. DP is lower and AT^2 is high. If the design goal is high speed, this logic structure for designing Ripple Carry Adder is a good choice.
 - (2) N-type 2-Phase Logic: The area is small. The delay is average. The power consumption is lowest. DP and AT^2 are the lowest than other logic structure of Ripple Carry Adders. If the design goal is DP and AT^2 , this structure is the best choice.

(3) Domino 2-Phase Logic: The area is large. The delay is short. The power consumption is lower. DP is lower and AT^2 is same as P-E logic. The performance of this logic is good except for having large area.

- MODL Logic: The adder that has been designed by this logic is Carry-Look-ahead Adder. So, compare with Ripple Carry Adder, Carry-Look-ahead Adder has better performance of delay, DP and AT^2 . Especially, the delay and AT^2 are 60% and 80% less than Ripple Carry Adder.
- Cascode Logic: It has the lower power consumption and the largest area. The delay is shorter. DP is less than other Ripple Carry Adders except N-type 2-phase logic. But AT^2 is the highest. So, if the design goal is a high speed and low power with little attention to area then Cascode is a better choice.

APPENDIX B

Technology parameters used in the simulation

1. Specification of CMOS0.5 μ m technology

All designs are implemented with CMOSIS0.5um technology. The simulation model is Hspice .

The model parameters of PMOS and NMOS transistor which are used in our analysis and calculation are listed below.

Model cmos NMOS level3:

$$V_{to}=0.6566V, k_n=196.47\mu A/V^2, \mu_n=546.2cm^2/V\cdot s,$$

$$C_{ox}=3.6e-03F/m^2, C_j=5.62e-04F/m^2,$$

$$C_{jsw}=5.0e-12F/m^2, C_{jgate}=5.0e-12F/m^2$$

$$C_{gbo}=4.0239e-10F/m^2, C_{gdo}=3.0515e-10F/m^2$$

$$C_{gso}=3.0515e-10F/m^2$$

Model cmosp PMOS level3:

$$V_{to}=-0.9213V, k_p=48.74\mu A/V^2, \mu_p=135.52cm^2/V\cdot s,$$

$$C_{ox}=3.6e-03F/m^2, C_j=9.35e-04F/m^2,$$

$$C_{jsw}=289.00e-12F/m^2, C_{jgate}=289.00e-12F/m^2$$

$$C_{gbo}=3.7579e-10F/m^2, C_{gdo}=2.3922e-10F/m^2$$

$$C_{gso}=2.3922e-10F/m^2$$

Power supply: 3.3 voltage

Delay: Input D has delay time of 500ps; The clock pulse, delay is set to 500p.

APPENDIX C

Equations and metrics used

The switching speed of a CMOS gate is limited by the time taken to charge and discharge the load capacitance C_L . The time delay equals to the time difference between input transition (50%) and the 50% output level.

Assumptions:

- (1) V_i is a perfect input without delay;
- (2) PMOS and NMOS are perfect that turn on/off instantaneously;
- (3). The channel length modification factor $\lambda=0$;

$$\text{Time delay from low to high, } T_{plh} = C_L \frac{V_{DD/2}}{i_{avp}}$$

$$\text{Time delay from high to low, } T_{phl} = C_L \frac{V_{DD/2}}{i_{avn}}$$

The i_{avp} and i_{avn} are calculated by following formulas:

$$i_{avp} = \frac{1}{2} [i_{t=0} + i_{i=T_{plh}}] = \frac{1}{2} \left\{ \frac{1}{2} \beta_p (V_{DD} - |V_{tp}|)^2 + \beta_p \left[(V_{DD} - |V_{tp}|) \times \frac{1}{2} \left(\frac{V_{DD}}{2} \right)^2 \right] \right\};$$

$$i_{avn} = \frac{1}{2} [i_{t=0} + i_{i=T_{phl}}] = \frac{1}{2} \left\{ \frac{1}{2} \beta_n (V_{DD} - V_{tn})^2 + \beta_n \left[(V_{DD} - V_{tn}) \times \frac{1}{2} \left(\frac{V_{DD}}{2} \right)^2 \right] \right\};$$

Where $\beta_p = \mu_p C_{ox} (W/L)_p$ and $\beta_n = \mu_n C_{ox} (W/L)_n$;

From the listed equations, we can see that T_{plh} and T_{phl} are inversely proportional to current i_{avp} and i_{avn} , which is proportional to value β_p and β_n .

For symmetrical inverter $T_{plh} = T_{phl}$,

$$\frac{\beta_n}{\beta_p} \approx 1, \text{ therefore } \frac{(W/L)_p}{(W/L)_n} = \frac{\mu_n}{\mu_p}.$$

According to CMOS0.5 technology model, $\frac{(W/L)_p}{(W/L)_n} \approx 4$.

But for multi-level inverters in cascade, when we consider optimizing the size of transistor of second level and so on, we must think of contribution of size change to the load capacitance of previous level:

$$C_L = C_{\text{gaten}} + C_{\text{gatep}} + C_{\text{dbp}} + C_{\text{dbn}} + C_{\text{gdp}} + C_{\text{gdn}}$$

Where C_{gaten} , C_{gatep} , C_{gdn} , C_{gdp} are gate capacitance and gate overlap capacitance of next level;

$$C_{\text{gaten}} = (W \times L)_n \times C_{\text{ox}}$$

$$C_{\text{gatep}} = (W \times L)_p \times C_{\text{ox}}$$

$$C_{\text{gd}} = C_{\text{gbo}} * 2L + C_{\text{gdo}} * W + C_{\text{gso}} * W$$

C_{dbn} , C_{dbp} are junction capacitance of present level.

$$C_{\text{db}} = C_j * \text{Area} + C_{\text{jsw}} * (W + 2L) + C_{\text{jgate}} * W$$