Chapter 10
SOLUTIONS

1. [C, None, 9.2] For the circuit in Figure 0.1, assume a unit delay through the Register and Logic blocks (i.e., $t_R = t_L = 1$). Assume that the registers, which are positive edge-triggered, have a set-up time t_S of 1. The delay through the multiplexer t_M equals 2 t_R.

a. Determine the minimum clock period. Disregard clock skew.

Solution

The circuit and paths of interest has been reproduced for convenience in Figure 0.1

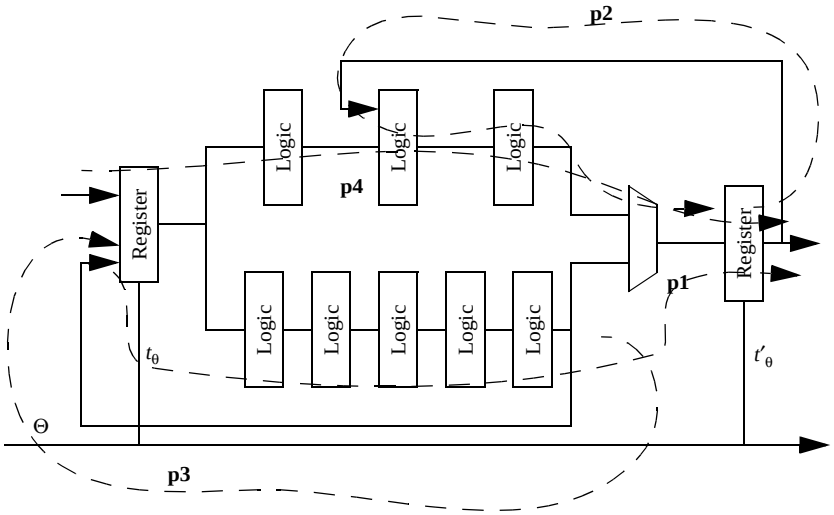


Figure 0.1 Sequential circuit.

Out of the 4 paths shown in the figure, p1 is the critical one and determines the lower bound on the clock period. Using $T \geq t_{reg} + t_{logic} + t_{setup} - \delta$, we get $\boldsymbol{T_{min}}$ **= 1+7+1 = 9**.

b. Repeat part a, factoring in a nonzero clock skew: $\delta = t'_\theta - t_\theta = 1$.

Solution

With finite clock skew, the time periods for different paths are as follows : T_{min}(p1) = 9 - 1 = 8, T_{min}(p2) = 6, T_{min}(p3) = 7, T_{min}(p4) = 7 - 1 = 6 (Note that the clock skew is 0 for paths p2 and p3). Therefore the minimum clock period is $\boldsymbol{T_{min}}$ **= 8**.

c. Repeat part a, factoring in a non-zero clock skew: $\delta = t'_\theta - t_\theta = 4$.

Solution

As the clock skew increases, the most significant path changes. Repeating the calculations in part (b) we get : T_{min}(p1) = 9 - 4 = 5, T_{min}(p2) = 6, T_{min}(p3) = 7, T_{min}(p4) = 7 - 4 = 3(Note that the clock skew is 0 for paths p2 and p3). Therefore the minimum clock period is $\boldsymbol{T_{min}}$ **= 7**.

d. Derive the maximum positive clock skew that can be tolerated before the circuit fails.

Solution

The maximum positive clock skew is determined by the inequality $\delta \leq t_{cd,reg} + t_{cd,logic}$. Assuming that the contamination delay is same as the propagation delay, we get. $\delta_{max} = 1 + 3 + 2 = 6$. Note that p4 determines the maximum tolerable skew (the fastest path will produce the earliest contamination). Paths p3 and p2 do not matter since there is no skew involved.

e. Derive the maximum negative clock skew that can be tolerated before the circuit fails.

Solution

The maximum positive negative skew has no bound since the clock period has no upper bound.

2. This problem examines sources of skew and jitter.

a. A balanced clock distribution scheme is shown in Figure 0.2. For each source of variation, identify if it contributes to skew or jitter. Circle your answer in Table 0.1

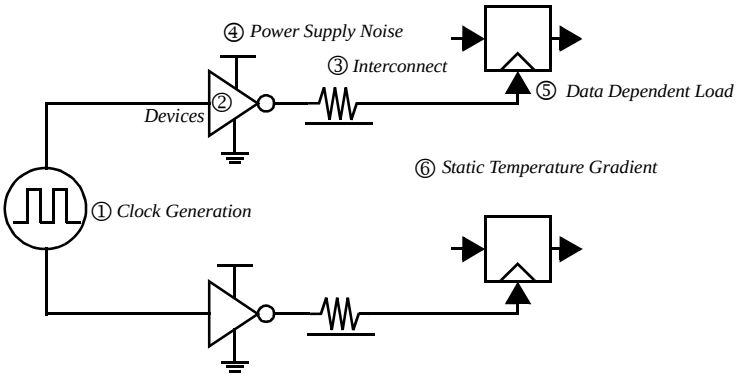


Figure 0.2 Sources of Skew and Jitter in Clock Distribution.

1) Uncertainty in the clock generation circuit	Skew	(Jitter)
2) Process variation in devices	(Skew)	Jitter
3) Interconnect variation	(Skew)	Jitter
4) Power Supply Noise	Skew	(Jitter)
5) Data Dependent Load Capacitance	Skew	(Jitter)
6) Static Temperature Gradient	(Skew)	Jitter

Table 0.1 Sources os Skew and Jitter

b. Consider a Gated Clock implementation where the clock to various logical modules can be individually turned off as shown in Figure 0.3. (i.e., $Enable_1$,..., $Enable_N$ can take on dif-

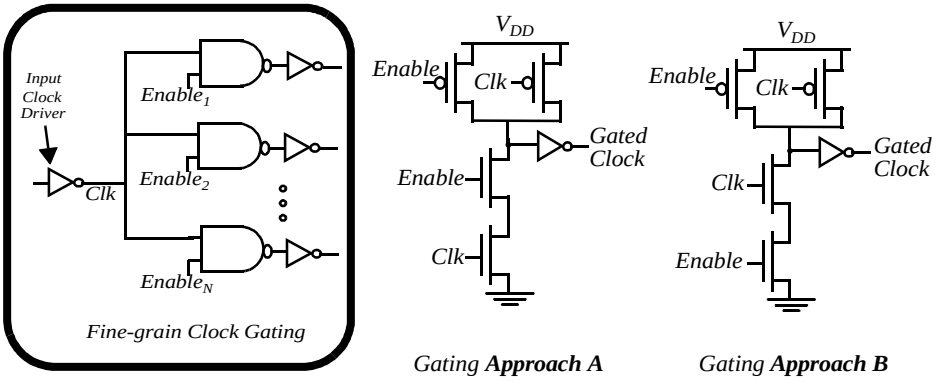


Figure 0.3 Jitter in clock gating

ferent values on a cycle by cycle basis). Which approach (*A* or *B*) results in lower *jitter* at the output of the input clock driver? (hint: consider gate capacitance) Explain.

Solution

Approach **A** results in lower jitter. For Approach A, the capacitance seen by CLK is independent of data (the Enable signals) to first order.

3. Figure 0.4 shows a latch based pipeline with two combinational logic units.

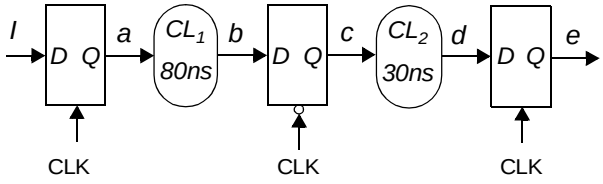


Figure 0.4 Latch Based Pipeline

Recall that the timing diagram of a combinational logic block and a latch can be drawn as follows, where the shaded region represents that the data is not ready yet.

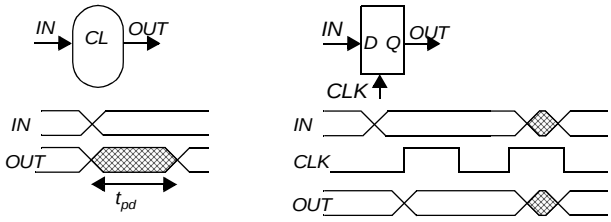


Figure 0.5 Timing diagrams of combinational logic and latch

Assume that the contamination delay t_{cd} of the combinational logic block is zero, and the $t_{clk\text{-}q}$ of the latch is zero too.

a. Assume the following timing for the input *I*. Draw the timing diagram for the signals *a*, *b*, *c*, *d* and *e*. Include the clock in your drawing.

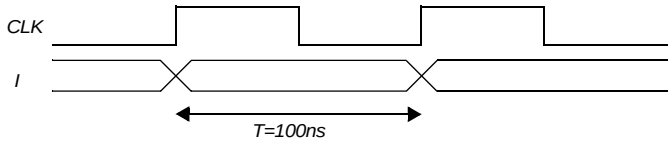
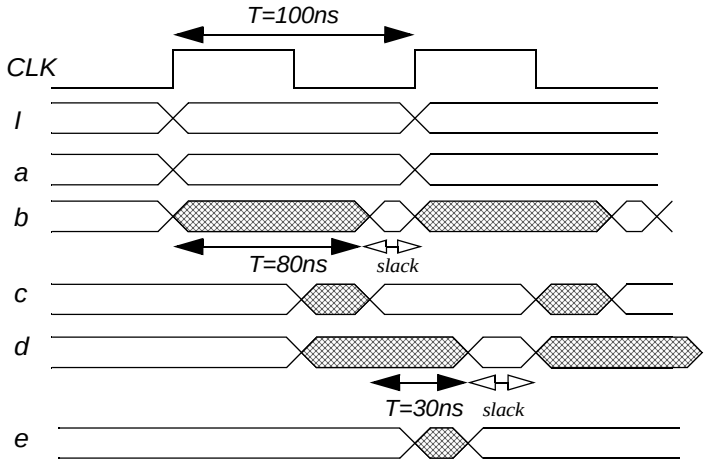


Figure 0.6 Input timing

Solution



b. State the deadline for the computation of the signal *b* and *d*, i.e. when is the latest time they can be computed, relative to the clock edges. In your diagram for part (a), label with a "<-->" the "slack time" that the signals *b* and *d* are ready before the latest time they must be ready.

Solution

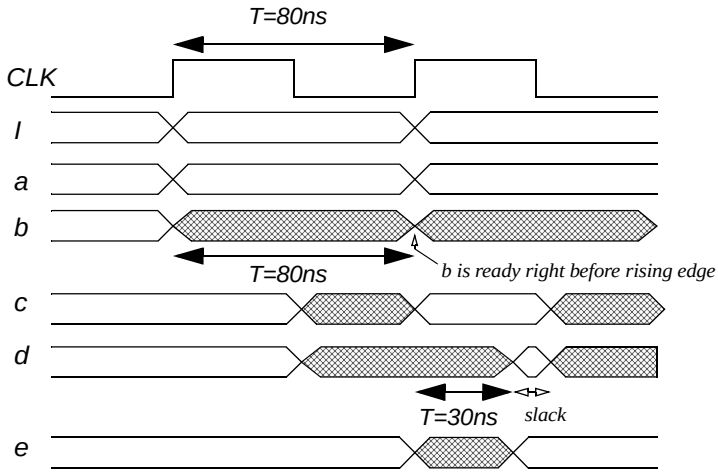
b should be ready before the rising edge of *CLK* for the negative latch to latch and hold its value. *d* should be ready before the falling edge of *CLK* for the second positive latch to latch and hold its value.

c. Hence deduce how much the clock period can be reduced for this shortened pipeline. Draw the modified timing diagram for the signals *a*, *b*, *c*, *d*, and *e*. Include the clock in your drawing.

Solution

The clock can be reduced by 20 ns.

In general, it may be difficult to identify how much slack can be removed from the



clock because it depends on the length of the pipeline too.

4. Consider the circuit shown in Figure 0.7.

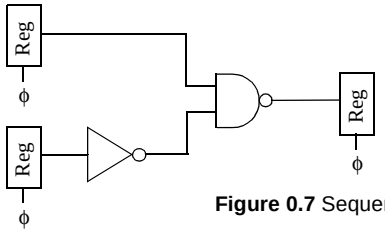


Figure 0.7 Sequential Circuit

a. Use SPICE to measure t_{max} and t_{min}. Use a minimum-size NAND gate and inverter. Assume no skew and a zero rise/fall time. For the registers, use the following:

- A TSPC Register.
- A C^2MOS Register.

Solution

From Figure 0.8 and Figure 0.9 we can see that for the TSPC Register:

$t_{r,max}$=175ps, $t_{r,min}$=94ps, $t_{and,max}$=90ps, $t_{and,min}$=83ps. (Note that we don't need the inverter to implement the logic when we use TSPC Registers).

$$\text{So } T > t_{r,max} + t_{and,max} = 265\text{ps}.$$

From Figure 0.10 and Figure 0.11we can see that for the C^2MOS Register:

$t_{r,max}$=82ps, $t_{r,min}$=49ps,$t_{inv,max}$=40ps, $t_{inv,min}$=33ps.

$$\text{So } T > t_{r,max} + t_{inv,max} + t_{and,max} = 212.$$

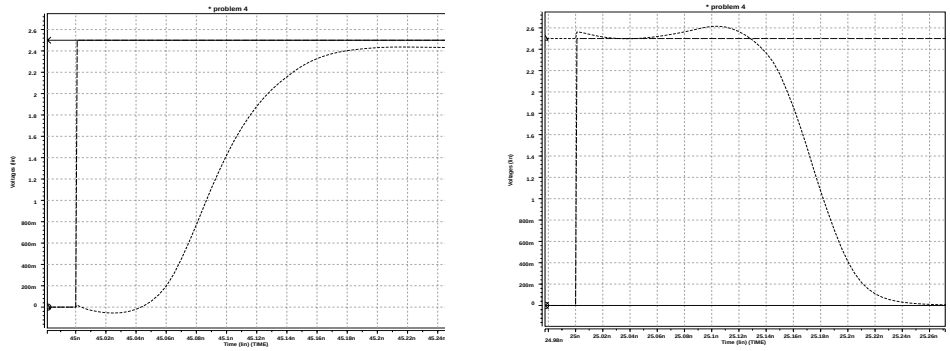


Figure 0.8 $t_{r,max}$ and $t_{r,min}$

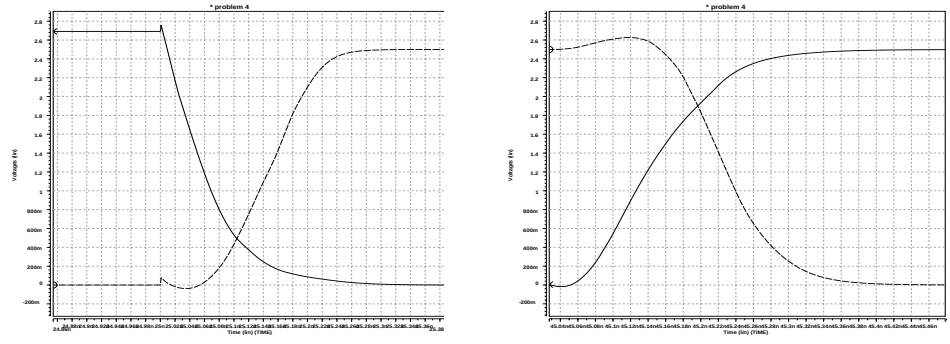


Figure 0.9 $t_{and,max}$ and $t_{and,min}$

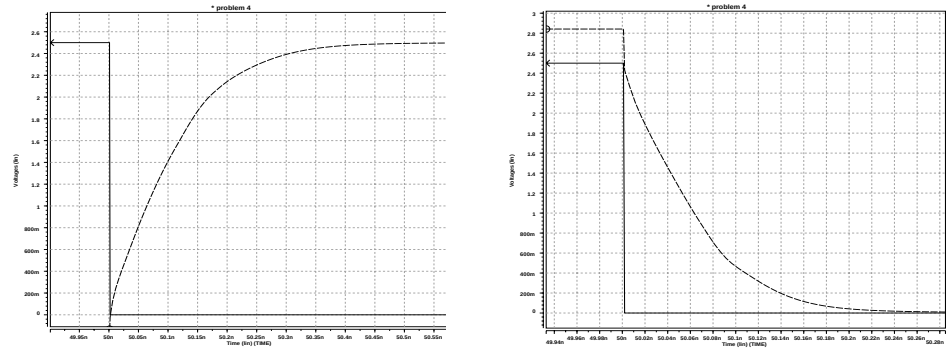


Figure 0.10 $t_{r,max}$ and $t_{r,min}$

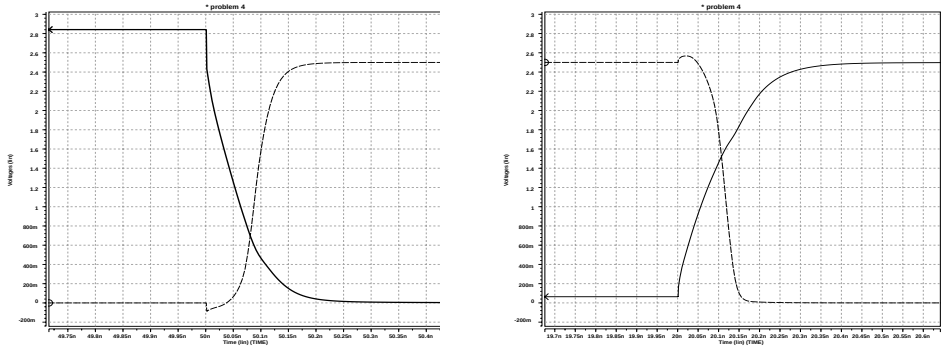


Figure 0.11 $t_{inv,max}$ and $t_{inv,min}$

b. Introduce clock skew, both positive and negative. How much skew can the circuit tolerate and still function correctly?

Solution

We will examine the case with the TSPC Register.

The maximum positive skew that the circuit can tolerate is 100ps. Figure 0.12 show the correct operation with no skew. The next two figures show the cases with skew of 100ps and 110ps. It is obvious that in the second case there is some corruption.

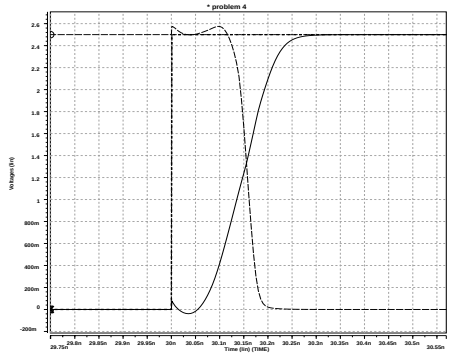


Figure 0.12 No skew

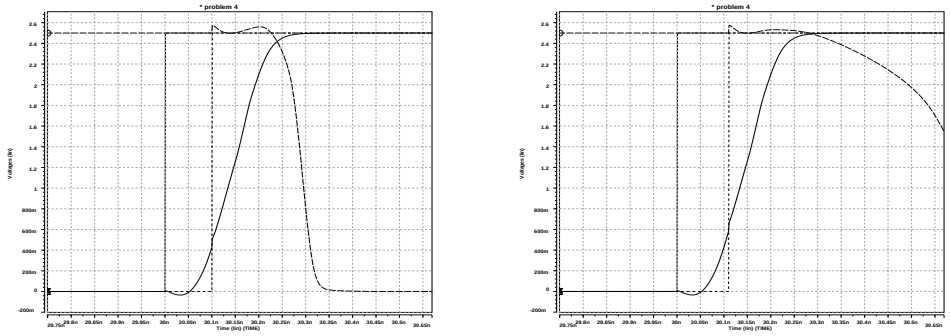


Figure 0.13 100ps skew and 110ps skew

When the clock is routed in the opposite direction of the data (negative skew) the circuit operates correctly, with a negative impact on the circuit performance.

c. Introduce finite rise and fall time to the clocks. Show what can occur and describe why.

Solution

As the rise and fall times of the clock increase, both chains of the C^2MOS chains are on simultaneously. The first graph whow the correct operation of the register, while in the next two graphs rise and fall times of 2ns and 3ns respectively is introduced.

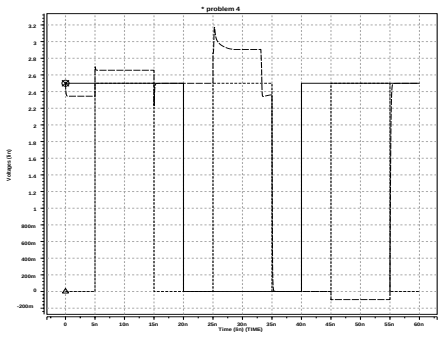


Figure 0.14 0ns rise and fall times

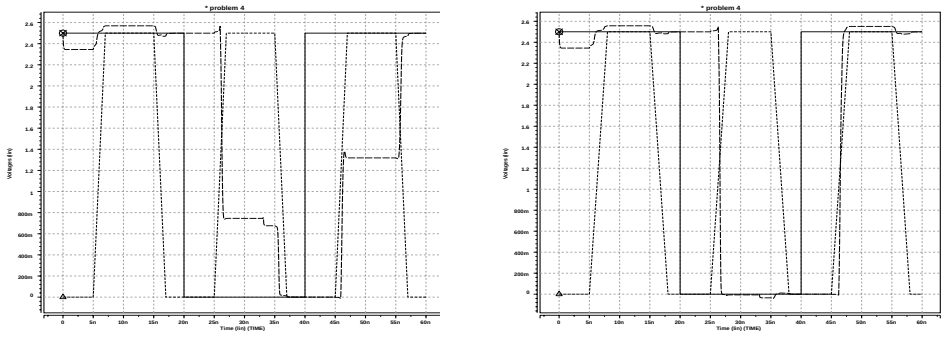


Figure 0.15 2ns and 3ns rise and fall times

5. Consider the following latch based pipeline circuit shown in Figure 0.16.

Assume that the input, *IN*, is valid (i.e., set up) 2ns before the falling edge of *CLK* and is held till the falling edge of *CLK* (there is no guarantee on the value of *IN* at other times). Determine the maximum *positive* and *negative* skew on *CLK'* for correct functionality.

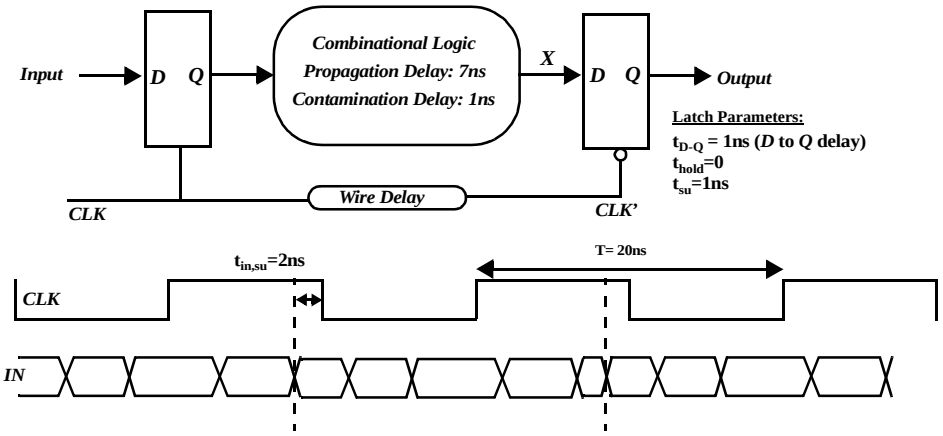
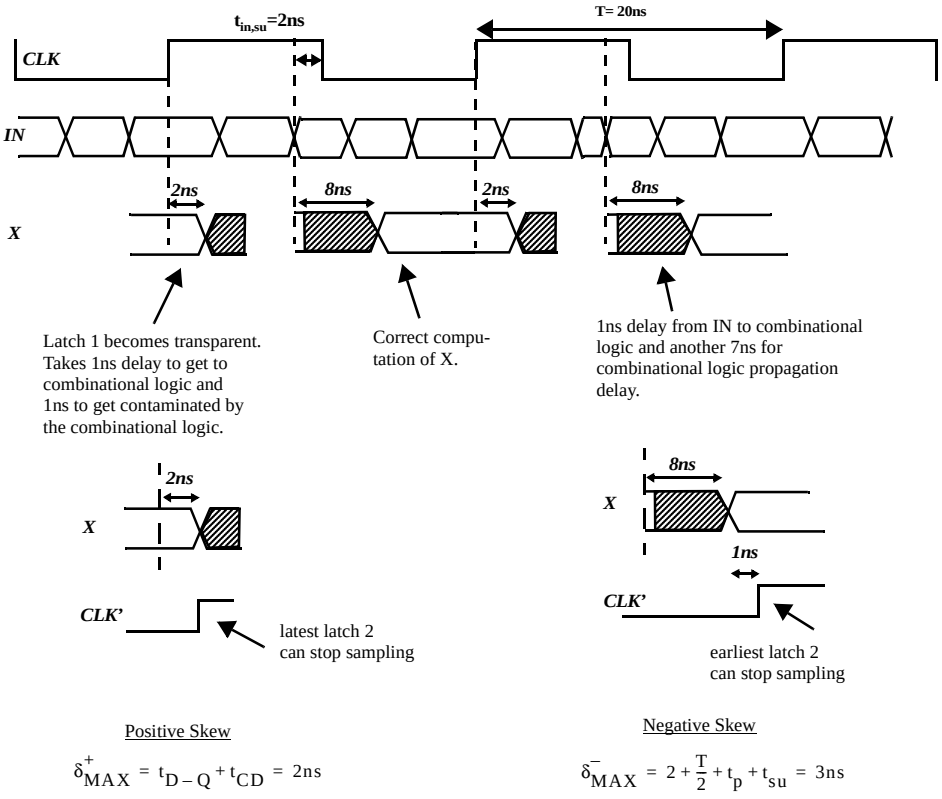


Figure 0.16 Latch based

Solution

The positive and negative skew are given after the analysis below.



Positive Skew

$$\delta^{+}_{MAX} = t_{D-Q} + t_{CD} = 2\text{ns}$$

Negative Skew

$$\delta^{-}_{MAX} = 2 + \frac{T}{2} + t_p + t_{su} = 3\text{ns}$$

6. For the L1-L2 latch based system from Figure 0.17**,** with two overlapping clocks derive all the necessary constraints for proper operation of the logic. The latches have setup times T_{SU1} and T_{SU2}, data-to-output delays $T_{D\text{-}Q1}$ and $T_{D\text{-}Q2}$, clock-to-output delays $T_{Clk\text{-}Q1}$ and $T_{Clk\text{-}Q2}$, and hold times T_{H1} and T_{H2}, respectively. Relevant clock parameters are also illustrated in Figure 0.17. The constraints should relate the logic delays, clock period, overlap time T_{OV}, pulse widths *PW*1 and *PW*2 to latch parameters and skews.

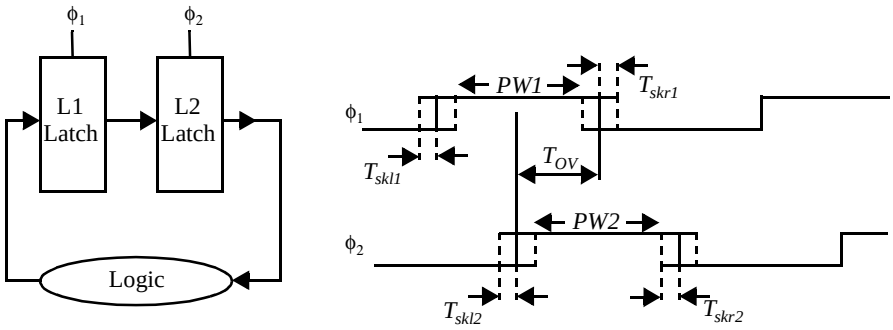


Figure 0.17 Timing constraints

Solution

Latest arrival of the D2 signal in the current clock cycle (“Setup 2”)

$$PW2 \geq T_{ov} + T_{sw2} - T_{sw1} + T_{D-Q1} + T_{skr2} - T_{skr1}$$

$$PW1 + PW2 \geq T_{ov} + T_{sw2} + T_{C-Q1} + T_{skl1} - T_{skr2}$$

Latest arrival of the D1 signal in the next clock cycle (“Setup 1”)

$$P \geq T_{D-Q1} + T_{D-Q2} + T_{gates}$$

$$PW1 \geq -P + T_{C-Q1} + T_{D-Q1} + T_{sw1} + T_{gates} + T_{skl2} + T_{skr2}$$

$$P \geq -T_{ov} + T_{C-Q2} + T_{sw1} + T_{gates} + T_{skl1} + T_{skl2}$$

Earliest changes of D1 signal (“Hold 1”)

$$T_{d,\,logic} > T_{ov} + T_{H1} + T_{skr1} + T_{skl2} - T_{C-Q2}$$

$$T_{d,\,logic} > PW1 + T_{H1} + T_{skr1} + T_{skl1} - T_{D-Q1} - T_{C-Q2}$$

Earliest changes of D2 signal (“Hold 2”)

$$PW2 < T_{H1} - T_{H2} + T_{D-Q1} + T_{ov} + T_{skr1} - T_{skr2}$$

$$PW1 + PW2 \geq T_{ov} + P + T_{C-Q1} - T_{H2} - T_{skr1} - T_{skr2}$$

7. For the self-timed circuit shown in Figure 0.18, make the following assumptions. The propagation through the NAND gate can be 5 nsec, 10 nsec, or 20 nsec with equal probability. The logic in the succeeding stages is such that the second stage is always ready for data from the first.

a. Calculate the average propagation delay with t_{hs} = 6 nsec.

Solution

$$t_p = \frac{5 + 10 + 20}{3} + 6 = 17.67\text{ns}$$

$$f = 56.6\text{MHz}$$

b. Calculate the average propagation delay with t_{hs}=12 nsec.

Solution

$$t_p = \frac{5 + 10 + 20}{3} + 12 = 23.67\text{ns}$$

$$f = 42.2\text{MHz}$$

c. If the handshaking circuitry is replaced by a synchronous clock, what is the smallest possible clock frequency?

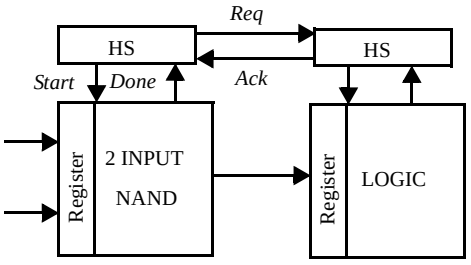


Figure 0.18 Self-timed circuit.

Solution

In setting clock frequency, we account for the longest delay:

$$f = \frac{1}{20\text{ns}} = 50\text{MHz}$$

Note that the delay in the handshaking circuit can be a strong factor in choosing clocking strategies.

8. Lisa and Marcus Allen have a luxurious symphony hall date. After pulling out of their driveway, they pull up to a four-way stop sign. They pulled up to the sign at the same time as a car on the cross-street. The other car, being on the right, had the right-of-way and proceeded first. On the way they also have to stop at traffic signals. There is so much traffic on the freeway, the metering lights are on. Metering lights regulate the flow of merging traffic by allowing only one lane of traffic to proceed at a time. With all the traffic, they arrive late for the symphony and miss the beginning. The usher does not allow them to enter until after the first movement.

On this trip, Lisa and Marcus proceeded through both synchronizers and arbiters. Please list all and explain your answer.

Solution

At the stop sign, the law of “right of way” is the **arbiter** of two cars arriving at the same time.

The stop light may also be considered an **arbiter** as it ensured that two cars don’t try to merge simultaneously, however it is more like a **synchronizer** as it allows traffic into an intersection only at specific times.

The metering lights are **synchronizers** as they allow cars to enter the freeway at distinct times.

The user is a **synchronizer** making sure that people go in and out of the concert at the proper times.

9. Design a self-timed FIFO. It should be six stages deep and have a two phase handshakin with the outside world. The black-box view of the FIFO is given in Figure 0.19.

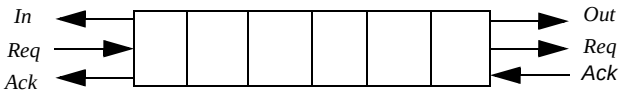
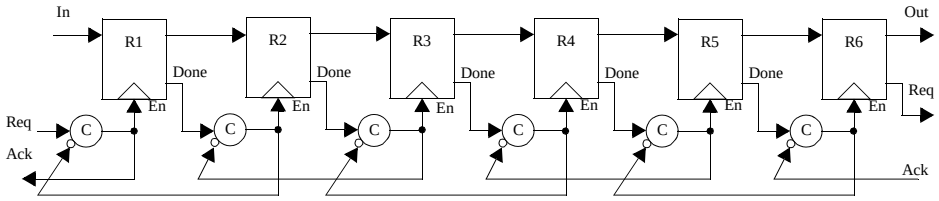


Figure 0.19 Overall structure of FIFO.

Solution

The block diagram of the FIFO is given in the next figure.



The registers are dual-edge triggered on the enable signal, and Done is just a delayed version of enable. The FIFO is full if the Enable Signals alternate between 0's and 1's. On the other hand, the FIFo is empty if all enable signals are equal (either 0 or 1).

10. System Design issues in self-timed logic

One of the benefits of using self-timed logic is that it delivers average-case of performance rather than the worst-case performance that must be assumed when designing synchronous circuits. In some applications where the average and worst cases differ significantly you can have significant improvements in terms of performance. Here we consider the case of ripple carry addition. In a synchronous design the ripple carry adder is assumed to have a worst case performance which means a carry-propagation chain of length N for an N-bit adder. However, as we will prove during the course of this problem the average length of the carry-propagation chain assuming uniformly distributed input values is in fact O(log N)!

a. Given that $p_n(v)$ = Pr(carry-chain of an n-bit addition is ≥ v bits), what is the probability that the carry chain is of length *k* for an *n*-bit addition?

Solution

The Pr(carry-chain = k bits) = Pr(carry-chain is ≥ k bits) - Pr(carry-chain is ≥ k+1 bits), which is:

$$P_k = p_n(v) - p_n(v+1)$$

b. Given your answer to part (a), what is the average length of the carry chain (i.e., a_n)? Simplify your answer as much as possible.

Now $p_n(v)$ can be decomposed into two mutually-exclusive events, A and B. Where A represents that a carry chain of length ≥ v occurs in the first *n*-1 bits, and B represents that a carry chain of length *v* ends on the *n*th bit

Solution

The average length of the carry chain is simply the expected value of P_k, which is:

$$a_n = E[P_k] = \sum_{i=0}^{n} i \cdot \left(p_n(i) - p_n(i+1)\right)$$

$$= p_n(1) - p_n(2) + 2p_n(2) - 2p_n(3) + \ldots$$

$$= p_n(1) + p_n(2) + \ldots + p_n(n)$$

$$= \sum_{i=1}^{n} p_n(i)$$

c. Derive an expression for Pr(A).

Solution

Pr(A) is simply $p_{n-1}(v)$.

d. Derive an expression for Pr(B). (HINT: a carry bit i is propagated only if $a_i \neq b_i$, and a carry chain begins only if $a_i = b_i = 1$).

Solution

For B to occur a carry must be generated in bit $(n - v)$ and then propagated all the way to bit n. In addition we must ensure that no carry chain of length v occurs in the initial $(n - v)$ bits. The probability of a carry being generated is $Pr(A = B = 1) = (1/2)^2 = 1/4$, and the probability of this carry being propagated until bit n is $Pr(A \neq B)^{v-1} = (1/2)^{v-1}$. The probability of a carry chain of length v not occurring in the first $(n-v)$ bits is $(1 - p_{n-v}(v))$. Hence the probability of event B occurring is:

$$Pr(B) = (1 - p_{n-v}(v)) \cdot \frac{1}{4} \cdot \frac{1}{2^{v-1}} = \frac{1 - p_{n-v}(v)}{2^{v+1}}$$

e. Combine your results from (c) and (d) to derive an expression for $p_n(v) - p_{n-1}(v)$ and then bound this result from above to yield an expression in terms of only the length of the carry chain (i.e., v).

Solution

From the question we are given that:

$$p_n(v) = Pr(A) + PrB = p_{n-1}(v) + \frac{1 - p_{n-v}(v)}{2^{v+1}}$$

So all we have to do is substitute in our values of Pr(A) and Pr(B) and then rearrange the equation to yield the required expression:

$$p_n(v) = p_{n-1}(v) + \frac{1 - p_{n-v}(v)}{2^{v+1}}$$

$$\Rightarrow p_n(v) - p_{n-1}(v) = \frac{1 - p_{n-v}(v)}{2^{v+1}}$$

Since $p_{n-v}(v)$ is a probability it is non-negative and hence we can bound $(1 - p_{n-v}(v))$ from above by 1, thus:

$$p_n(v) - p_{n-1}(v) \le \frac{1}{2^{v+1}}$$

f. Using what you've shown thus far, derive an upper bound for the expression:

$$\sum_{i=v}^{n} (p_i(v) - p_{i-1}(v))$$

Use this result, coupled with the fact that $p_n(v)$ is a probability (i.e., it's bounded from above by 1), to determine a two-part upper bound for $p_n(v)$.

Solution

To derive the first upper bound we expand the given summation and collect terms:

$$\sum_{i=v}^{n} (p_i(v) - p_{i-1}(v)) = p_v(v) - p_{v-1}(v) + p_{v+1}(v) - p_v(v) + \ldots + p_n(v) - p_{n-1}v = p_n(v)$$

where $p_n(v)$ can be bounded from above by 1.

The second upper bound is calculated using the expression that we derived in (e), and substituting it into the given summation. Note that the expression derived in (e) is independent of the summation variable and hence the result is simply $(n - v + 1)$ times the bound given in (e):

$$\sum_{i=v}^{n} (p_n(v) - p_{n-1}(v)) \le \frac{n-v+1}{2^{v+1}}$$

Combining the two results we get the final, dual-valued upper bound on $p_n(v)$:

$$p_n(v) \le \min\left\{1, \frac{n-v+1}{2^{v+1}}\right\}$$

g. (The magic step!) Bound n by a clever choice of k such that $2^k \le n \le 2^{k+1}$ and exploit the fact that $log_2 x$ is concave down on $(0, \infty)$ to ultimately derive that $a_n \le log_2 n$, which concludes your proof!

Solution

Go back to our original derivation of the average carry chain length (i.e., $E[P_k]$), and split the summation into two parts: those terms from 1 to (k-1), and those terms from k to n.

$$E[P_k] = \sum_{i=1}^{k-1} p_n(v) + \sum_{k}^{n} p_n(v)$$

Now utilize your two upper bounds from (f) to bound the above expression:

$$E[P_k] \le \sum_{i=1}^{k-1} 1 + \sum_{i=k}^{n} \frac{n-i+1}{2^{i+1}}$$

$$= (k-1) + \sum_{i=k}^{n} \frac{n-i+1}{2^{i+1}}$$

$$\le (k-1) + \sum_{i=k}^{n} \frac{n}{2^{i+1}}$$

$$= (k-1) + \frac{n}{2}\sum_{i=k}^{n} 2^{i}$$

$$= (k-1) + \frac{n}{2}\left(\frac{1}{2^k} - \frac{1}{2^n}\right)$$

$$\le (k-1) + \frac{n}{2^k}$$

which is a linear function of *n*. At the limits defined for *n* we have:

$$n = 2^k \rightarrow E[P_k] = (k-1) + \frac{2^k}{2^k} = k = \log n$$

$$n = 2^{k+1} \rightarrow E[P_k] = (k-1) + \frac{2^{k+1}}{2^k} = k+1 = \log n$$

Since $\log_2 n$ is concave down on $(0,\infty)$ we have that $\log_2 n$ is an upper bounds of the linear function of n (e.g., Figure 0.20) derived above. Hence $E[P_k] \leq \log_2 n$ and we are finished.

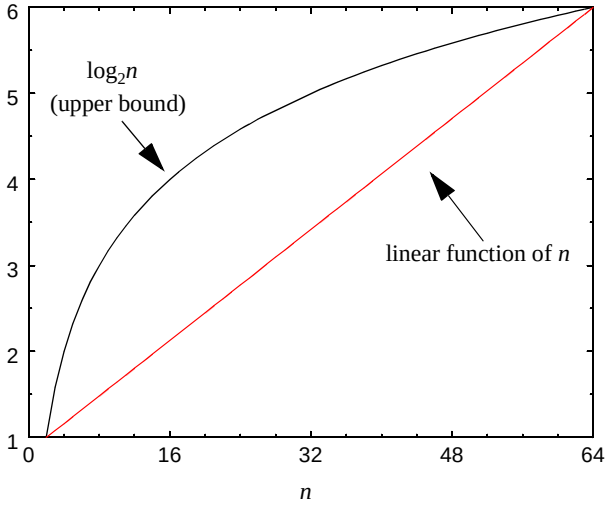


Figure 0.20 Comparison of $\log_2 n$ and a Linear Function of n

h. Theoretically speaking, how much faster would a self-timed 64-bit ripple carry adder be than its synchronous counterpart? (You may assume that the overhead costs of using self-timed logic are negligible).

Solution

Given that a self-timed ripple-carry adder requires a delay on the order of log_2n, while a synchronous version requires a delay on order n, then the improvement is:

$$\text{Speedup} = \frac{\text{Speed}_{\text{self-timed}}}{\text{Speed}_{\text{synchronous}}} = \frac{1/(\log 64)}{1/64} = \frac{64}{\log 64} = \frac{32}{3}$$

11. Figure 0.23 shows a simple synchronizer. Assume that the asynchronous input switches at a rate of approximately 10 MHz and that t_r = 2 nsec, f_ϕ = 50 MHz, $V_{IH} - V_{IL}$=0.5 V, and V_{DD} = 2.5 V.

a. If all NMOS devices are minimum-size, find $(W/L)p$ required to achieve V_{MS} = 1.25 V. Verify with SPICE.

Solution

Metastability occurs when both inputs to the cross coupled NANDs are high. One NAND of the cross-coupled pair is shown in Figure 0.21 a.

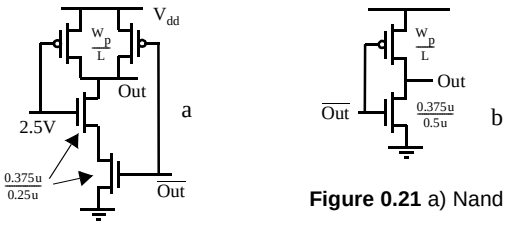


Figure 0.21 a) Nand Gate, b) Simplified Gate

To solve for the metastable point, we can simplify it to the gate shown in Figure 0.21 b, where the NMOS device size has been modified accordingly. Setting Out = $\overline{\text{Out}}$ = 1.25V and assuming that both devises are velocity saturated, I have:

$$k_n'\frac{W_n}{L}V_{DSATn}\left(V_{OUT}-V_{TN}-\frac{V_{DSATn}}{2}\right)+k_p'\frac{W_p}{L}V_{DSATp}\left(V_{OUT}-V_{DD}-V_{TP}-\frac{V_{DSATp}}{2}\right)=0$$

$$115\times10^{-6}\frac{0.375}{0.5}0.63\left(1.25-0.43-\frac{0.63}{2}\right)+(-30\times10^{-6})\frac{W_p}{L}(-1)\left(1.25-2.5-(-0.4)-\frac{(-1)}{2}\right)=0$$

which results in $(W/L)p$=2.6.

Hspice simulation gives V_M=1.23V.

b. Use SPICE to find τ for the resulting circuit.

Solution

The time constants for the metasble point to Vdd and ground are measured in Figure 0.22 and the two vlaues are 237ps and 362ps.

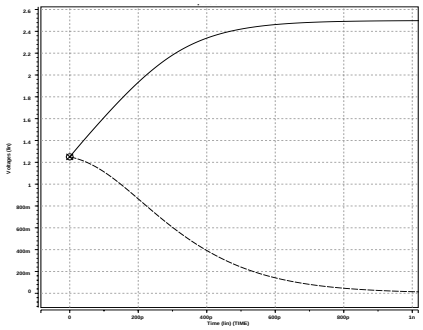


Figure 0.22 Measuring τ

c. What waiting time *T* is required to achieve a MTF of 10 years?

Solution

We can use the following equation, with the largest time constant, to find the waiting time *T.*

$$\frac{1}{\text{MTF}}=\frac{(V_{IH}-V_{IL})e^{-T/\tau}}{V_{SWING}}\frac{t_r}{T_{SIGNAL}T_\phi}$$

$$(315360000\text{s})^{-1}=\frac{(0.5)e^{-T/(362\text{ps})}}{2.5}\frac{2\text{ns}}{\frac{1}{10\text{MHz}}\times\frac{1}{50\text{MHz}}}$$

$$(315360000\text{s})^{-1}=200\times10^{3}e^{-T/(362\text{ps})}$$

from which we get:

$$T=11.5\text{ns}$$

d. Is it possible to achieve an MTF of 1000 years (where $T > T_\phi$)? If so, how?

Solution

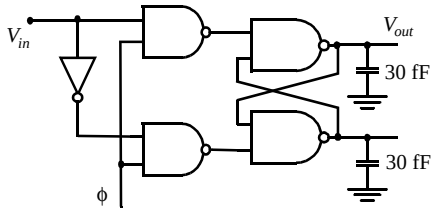


Figure 0.23 Simple synchronizer

We have: $(315360000\text{s})^{-1} = 50 \times 10^{3} e^{-T/(362\text{ps})}$, (T_{Φ}=5ns)
Solving yields: T=9.42ns > T_{Φ}.

12. Explain how the phase-frequency comparator shown in Figure 0.24 works.

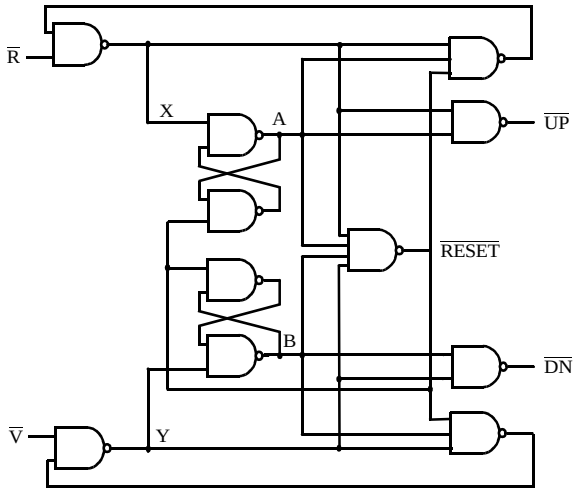
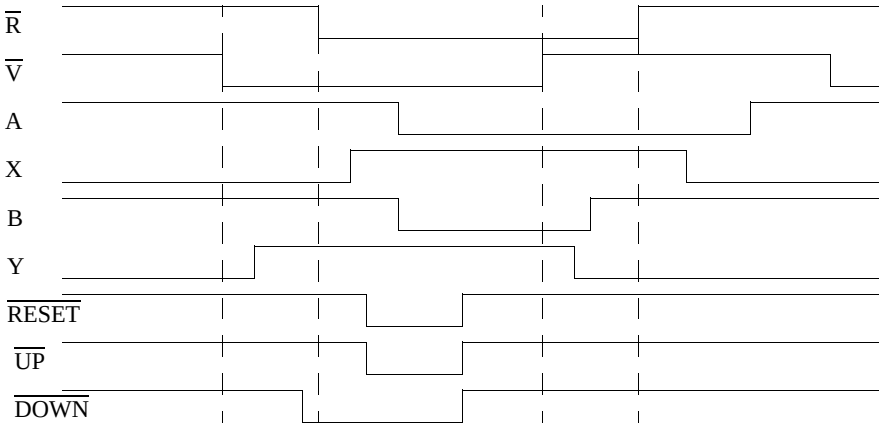


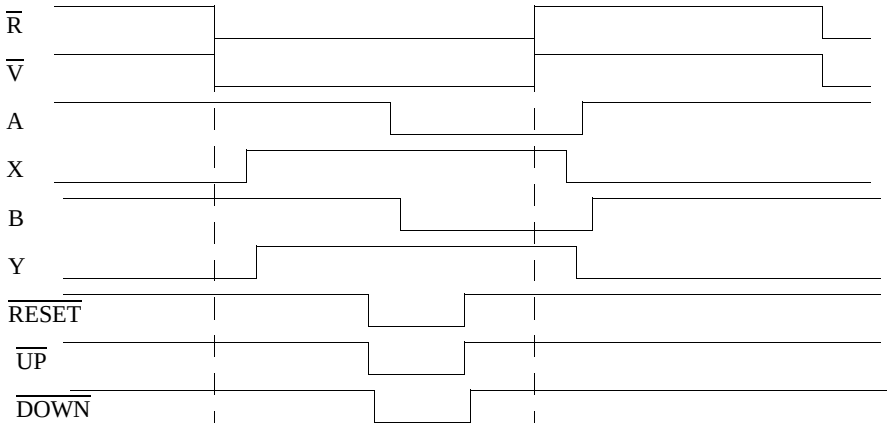
Figure 0.24 Phase-frequency comparator

The operation of the circuit is best explained with the timing diagrams below:



If the VCO clock ($\overline{V}$) leads the reference clock then the $\overline{DOWN}$ pulse is wider than the $\overline{UP}$ pulse. That will eventually shift $\overline{V}$ to the left so that the two clocks are locked.

The locked operation is shown in the next diagram. When the two clocks are locked



then the $\overline{UP}$ and $\overline{DOWN}$ pulses have equal widths.

13. The heart of any static latch is the cross-coupled structure shown in Figure 0.25 (part a).

a. Assuming identical inverters with *Wp/Wn =kn'/kp'*, what is the metastable point of this circuit? Give an expression for the time trajectory of V_Q, assuming a small initial V_{d0} centered around the metastable point of the circuit, V_M.

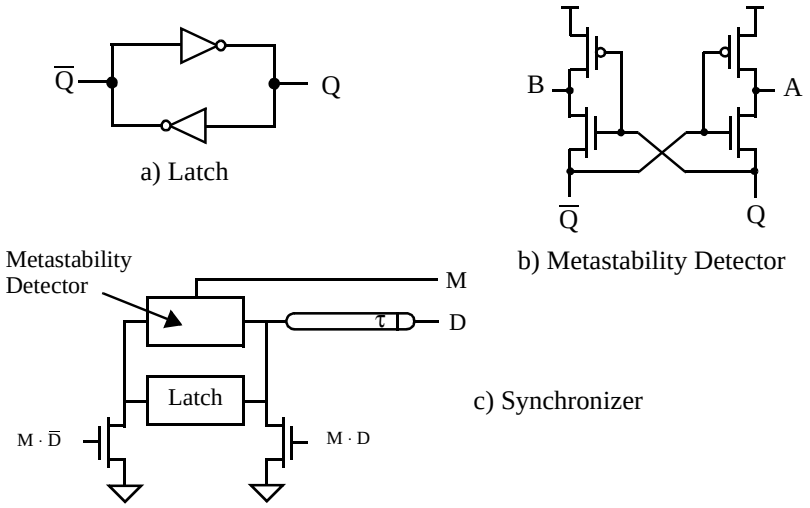


Figure 0.25 Simple synchronizer

Solution

To find the metastability point of the circuit we just need to find the gate voltage of one inverter that gives the same voltage at the inverter output.

Assuming that both devices are velocity saturated and neglecting channel length modulation, we can add the pmos and nmos currents. The equation looks like:

$$k_n'\frac{W_n}{L}V_{DSATn}\left(V_M - V_{TN} - \frac{V_{DSATn}}{2}\right) + k_p'\frac{W_p}{L}V_{DSATp}\left(V_M - V_{DD} - V_{TP} - \frac{V_{DSATp}}{2}\right) = 0$$

Solving for V_M,

$$V_M = \frac{V_{TN} + \frac{V_{DSATn}}{2} + \frac{V_{DSATp}}{V_{DSATn}}\left(V_{DD} + V_{TP} + \frac{V_{DSATp}}{2}\right)}{1 + \frac{V_{DSATp}}{V_{DSATn}}}$$

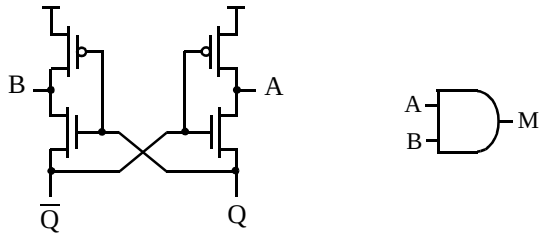
The time trajectory for the output can be modeled by:

$$v(t) = V_{MS} + (V_{d0} - V_{MS})e^{t/\tau}$$

b. The circuit in part b has been proposed to detect metastability. How does it work? How would you generate a signal M that is high when the latch is metastable?

Solution

If Q = $\overline{Q}$, then NMOS are off, so the PMOS devices will pull A and B high. That means that, when M goes high, the latch goes into the metastble state.



c. Consider the circuit of part c. This circuit was designed in an attempt to defeat metastability in a synchronizer. Explain how the circuit works? What ís the function of the delay element?

Solution

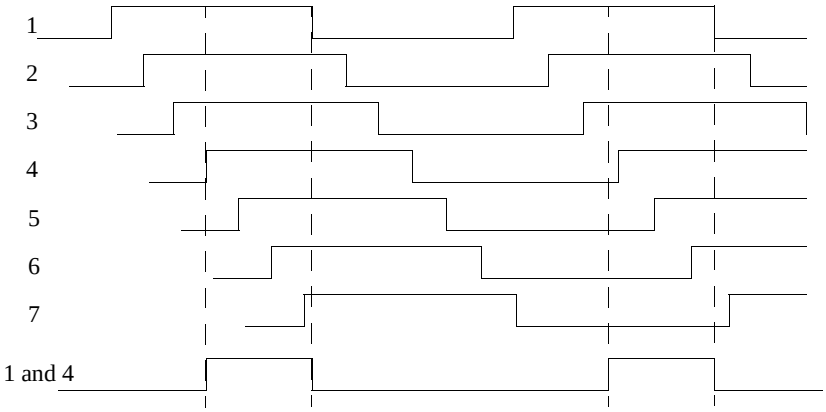
If the latch becomes metastable, then M will go high and turn on the appropriate NMOS pulling the latch out of metastability. The time delay τ gives the Metastability detector and the latch time to pull out of metastability.

14. An adjustable duty-cycle clock generator is shown in Figure 0.26. Assume the delay through the delay element matches the delay of the multiplexer.

a. Describe the operation of this circuit

Solution

The circuit works by using the overlap of two clocks from a ring oscillator to dictate the duty cycle. Longer overlap yields a greater duty cycle.



Clock signal from ANDing 1 + 4 gives 28.6% duty cycle.

b. What is the range of duty-cycles that can be achieved with this circuit.

Solution

The range of duty cycles is: 7-50%.

c. Using an inverter and an additional multiplexer, show how to make this circuit cover the full range of duty cycles.

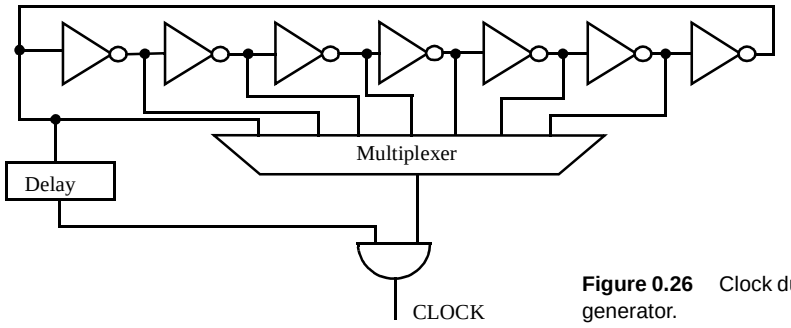
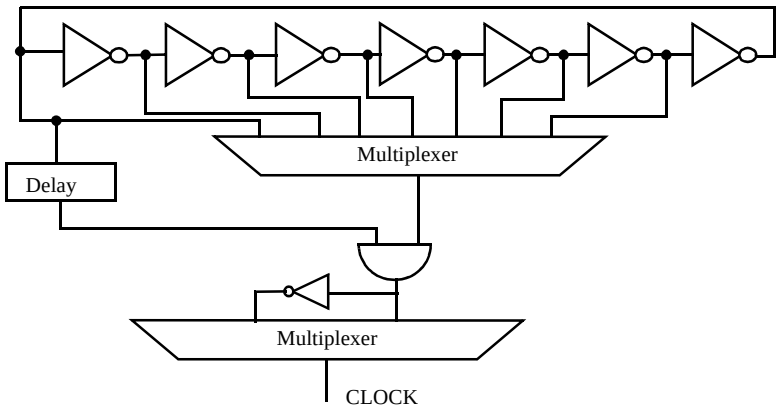


Figure 0.26 Clock duty-cycle generator.

Solution

Inverting the output signal converts a 25% duty cycle to a 75% duty cycle.



15. The circuit style shown in Figure 0.27.a has been proposed by Acosta et. al. as a new self-timed logic style. This structure is known as a Switched Output Differential Structure[1].

a. Describe the operation of the SODS gate in terms of its behavior during the pre-charge phase, and how a valid completion signal can be generated from its outputs.

Solution

Pre-charging is active low, and the inputs must become valid prior to the rising edge of Φ. During pre-charge the outputs are shorted together and at some point one of the pull-down networks will provide a path to ground. This path to ground will turn on one of the two pull-up PMOS transistors, connecting the two outputs to V_{dd}. Hence the outputs are high during the pre-charge phase. During the evaluate phase the outputs will become complimentary so you can use a NAND gate to signal completion when it's output goes high.

b. What are the advantages of using this logic style in comparison to the DCVSL logic style given in the notes?

Solution

The advantage of using SODS is that the delay of the gate is independent of the topology of the pull-down networks, and the gate will be faster due to reduced output capacitance of the switching nodes.

c. What are the disadvantages of using this style in comparison to DCVSL?

Solution

The disadvantage is that it requires the inputs to become valid before the pre-charge phase has ended, the outputs also exhibit reduced noise margins due to the problem discussed in (d). In addition, there can be significant static power dissipation during the evaluate phase if the gate is not designed carefully.

d. Figure 0.27.b shows a 2-input AND gate implemented using a SODS style. Simulate the given circuit using Hspice. Do you notice any problems? Explain the cause of any problems that you may observe and propose a fix. Re-simulate your corrected circuit and verify that you have in fact fixed the problem(s).

---

[1] A.J. Acosta, M. Valencia, M.J. Bellido, J.L. Huertas, "SODS: A New CMOS Differential-type Structure," *IEEE Journal of Solid State Circuits*, vol. 30, no. 7, July 1995, pp. 835-838

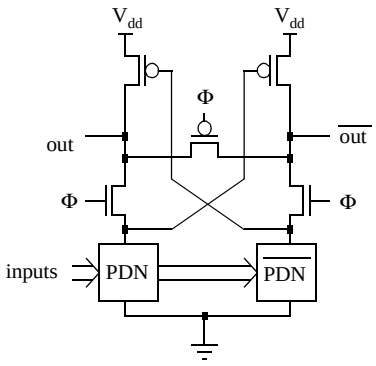


Figure 0.27 a - SODS Logic Style

V_{dd} = 2.5V

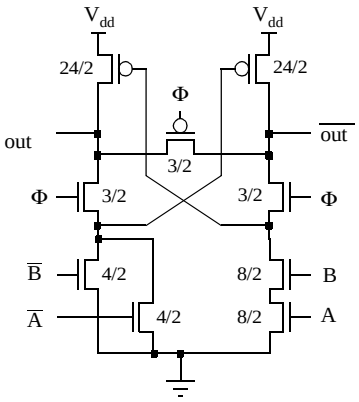
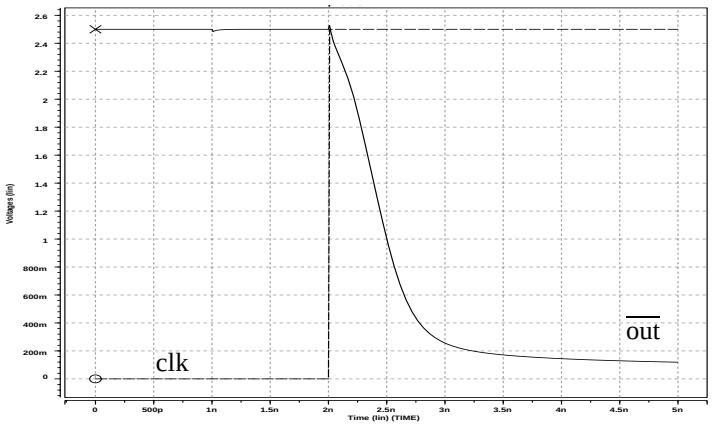


Figure 0.27 b - 2-input And Gate in SODS Style

Solution

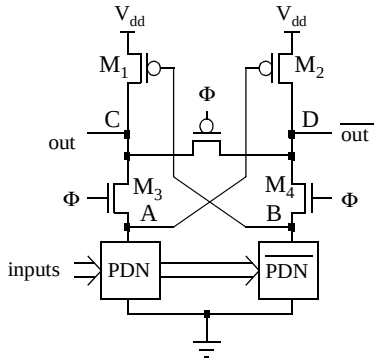
Using an input of A=B = 0, followed by A = B = 1 yields the outputs shown in the next graph. Note that there is a reduced noise margin in the outputs as they cannot be pulled to ground.



The reduced noise margins are caused by the fact that during the evaluate phase either node A or B will be pulled up to V_{dd} through an NMOS pass gate (Figure 8). For example: suppose that during the pre-charge phase node A is pulled low by its PDN and B is left to float, while nodes C and D are pre-charged to V_{dd}.

When Φ goes high the outputs are connected to A and B via the M_3 and M_4. Since A is connected to ground it will discharge C, pulling out low as well. Node D will remain pulled up to V_{dd} via M_2 and hence B will be pulled up to (V_{dd} - V_{tN}) through M4. This leaves the gate of M_1 at (V_{dd} - V_{tN}) as well, which will turn on M1

to some degree (as determined by the ratio of V_{tP} to V_{tN}), causing some static current to flow through M_3, generating some voltage at out given by $I_{leakage}R_{on(M_3)}$.



To reduce the problems the designer can either minimize the width of $M_{1\text{-}2}$ (to reduce $I_{leakage}$), or increase the width of $M_{3\text{-}4}$ (to reduce R_{on}). Figure 9 shows the effects of having the size of $M_{1\text{-}2}$, while doubling the size of $M_{3\text{-}4}$.

16. Voltage Control Ring Oscillator.

In this problem, we will explore a voltage controlled-oscillator that is based upon John G. Maneatis' paper in Nov. 1996, entitled "Low Jitter Process-Independent DLL and PLL Based on Self-Biased Techniques," appeared in the Journal of Solid-State Circuits. We will focus on a critical component of the PLL design: the voltage-controlled ring oscillator. Figure 0.28 shows the block diagram of a voltage controlled ring oscillator:

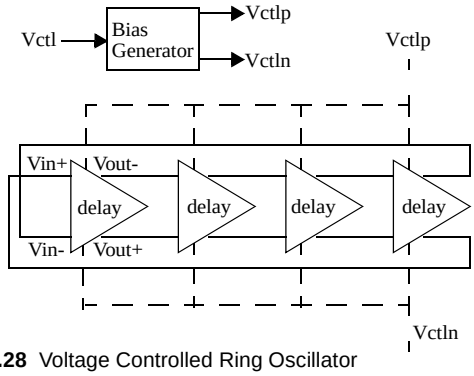


Figure 0.28 Voltage Controlled Ring Oscillator

The control voltage, Vctl, is sent to a bias generator that generates two voltages used to properly bias each delay cell equally, so that equal delay (assuming no process variations) appear across each delay cell. The delay cells are simple, "low-gain" fully differential input and output operational amplifiers that are connected in such a way that oscillations will occur at any one of the outputs with a frequency of 1/(4*delay). Each delay is modeled as an RC time constant; C comes from parasitic capacitances at the output nodes of the delay element,

and R comes from the variable resistor that is the load for the delay cell. Below is a circuit schematic of a typical delay cell.

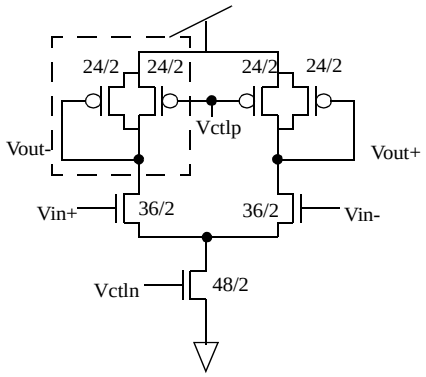


Figure 0.29 One delay Cell

As mentioned before, the value of R is set by a variable resistor. How can one make a variable resistor? The object in the delay cell that is surrounded by a dotted line is called a "symmetric load," and provides the answer to a voltage-controlled variable resistor. R should be linear so that the differential structure cancels power supply noise. We will begin our analysis with the symmetric load.

a. In Hspice, input the circuit below and plot Vres on the X axis and Ires on the Y axis, for the following values of Vctlp: 0.5, 0.75, 1.0, 1.25, 1.5, 1.75, and 2.0 volts, by varying Vtest from Vctlp to Vdd, all on the same graph. For each curve, plot Vres from 0 volts to Vdd-Vctlp. When specifying the Hspice file, be sure to estimate area and perimeter of drains/sources.

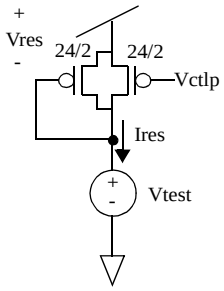
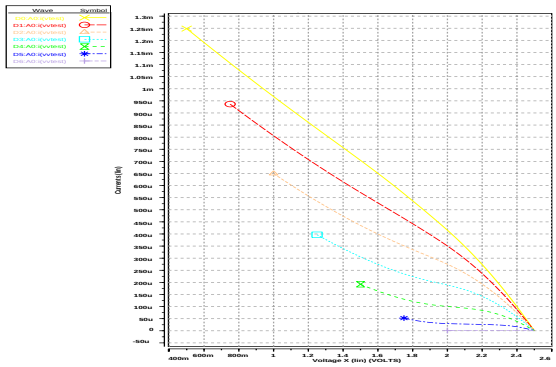


Figure 0.30 :Symmetric Load Test Circuit

After you have plotted the data and printed it out, use a straight edge to connect the end points for each curve. What do you notice about intersection points between the line you drew over each curve, and the curves themselves? Describe any symmetries you see.

Solution

The lines intersect curves at the "point of inflection" of the curves. At these points, the point of symmetry is x-y symmetric, if we take the drawn line as the x axis, and the y axis as a line drawn perpendicular to the x axis and intersecting the "point of inflection." Ideally, for symmetric loads, we should notice that the "point of inflection" should occur at 1/2 the voltage sweep range and 1/2 the current output range; but due to non-linear effects, the "point of inflection" is shifted towards the power rail.



b. For each Vctlp curve that you obtained in a), extract the points of symmetries (Vres, Ires), and find the slope of the line around these points of symmetry. These are the effective resistances of the resistors. Also, for each Vctl curve, state the maximum amplitude the output swing can be, without running into asymmetries. Put all of this data in an worksheet format.

Solution

Although the problem did not ask for 1/gm of the symmetric load when Vres=Vctl, it is good to look at them, because it is good to compare which 'effective resistance' to use as an estimation: the slope at the "point of inflection", or the effective resistance the symmetric load offers when it is in the lowest impedance state (when Vres=Vctl... that gives the lowest gm).

gm=W/L*k'*(Vdd-Vctl-Vt)
where W=sum of both transistor widths in a symmetric load
k'=30e-6
W=48
L=2
Vt=.4
(Vctl, Vres, Ires, Slopes, 1/gm of the symmetric load when Vres=Vctl)
(.5, 1.0, 500u, 2.26K, 868)
(.75, 0.7, 300u, 3K, 1.028K)
(1, 0.6, 200u, 4.76K, 1.262K)
(1.25, 0.5, 120u, 9K, 1.633K)
(1.5, 0.4, 80u, 22.3K, 2.314K)
(1.75, 0.25, 25u, 69K, 3.968K)
(2, 0.22, 0.6u, 2.2M, 13.888K)

For each symmetric load Vctl setting, the theoretical upper swing limit is Vdd, and lower swing limit is Vctl. Thus, the total swing is Vdd-Vctl. We will be using the bias generator to "set" the Vctln for a given Vctl (Vctlp is essentially Vctl) such that half the current runs through each of the delay cell legs. This theoretically biases the delay cell common mode outputs to Vdd-Vctl/2, which is supposed to be the point of inflection for a given Vctl voltage; however, since the symmetric loads are not perfectly symmetric, we will analyze and see how well the assumption holds. A way to correct this so that the symmetry point occurs

where we expect it to, would be relative sizing between the symmetric load transistors; but keep in mind that this actually invalidates the symmetry altogether, for certain ranges of bias currents (try it, you will see).

c. Using the estimations you made for area and perimeter of drain and source that you put in your Hspice file, calculate the effective capacitance. (Just multiply area and perimeter by CJ and CJSW from the spice deck). Since we are placing these delay elements in a cascaded fashion, remember to INCLUDE THE GATE CAPACITANCE of the following stage. Each delay element is identical to one another. Now, calculate the delay in each cell, according to each setting of Vctlp that you found in a): delay=0.69*R*C. Then, write a general equation, in terms of R and C, for the frequency value that will appear at each delay output. Why is it necessary to cross the feedback lines for the ring oscillator in the first figure? Finally, draw a timing/transient analysis of each output node of the delay lines. How many phases of the base frequency are there?

Solution

Capacitance Estimations: (lambda is .125e-6)
Area of drain/source pmos in symmetric: 24*lambda*.625e-6=1.875e-12
Perimiter of drain/source pmos in symmetric: 24*lambda+1.5e-6=4.5e-6
Area of drain/source nmos input: 36*lambda*.625e-6=2.8125e-12
Perimeter of drain/source nmos input: 36*lambda+1.5e-6=6e-6
Cgdon 3.1e-10
Cgdop=2.7e-10
Cjn=2e-3
Cjp=1.9e-3
Cjswn=2.75e-10
Cjswp=2.232e-10
Cox=6e-3

diode connected pmos contributes:
Cgp=Cgdop*Wp+Cox*Wp*Lp=2.7e-10*24*lambda+6e-3*24*2*lambda^2=5.31e-15
Cdb=Cjp*ADp+Cjswp*PDp=1.9e-3*1.875e-12+2.232e-10*4.5e-6=4.566e-15

current source pmos contributes:
Cgd=Cgdop*Wp=2.7e-10*24*lambda=8.1e-16
Cdb=Cjp*ADp+Cjswp*PDp=1.9e-3*1.875e-12+2.232e-10*4.5e-6=4.566e-15

Input gate transistor contributes:
Cgd=Cgdon*Wn=3.1e-10*36*lambda=1.395e-15
Cdb=Cjn*ADn+Cjswn*PDn=2e-3*2.8125e-12+2.75e-10*6e-6=7.275e-15

load capacitance presented by gate capacitance of following stage:
Cg=(2*Cgdon+Cgson)*Wn+Cox*Wn*Ln=(3*3.1e-10)*36*lambda+
+6e-3*lambda^2*36*2 = 10.9e-15

Total load capacitance: 34.817e-15 farads

In spice, the actual capacitance is 27.379e-15 farads. Pretty good estimation!

Here, we are also adding the analysis of Rgm (1/gm of symmetric load when Vctl=Vres of symmetric load) on delay:

(vctl, 0.69*Rslope*C, 0.69*Rgm*C)
(.5, 5.457e-11, 1.79e-11)

(.75, 7.24e-11, 2.127e-11)
(1, 1.1e-10, 2.61e-11)
(1.25, 2.17e-10, 3.38e-11)
(1.5, 5.38e-10, 4.78e-11)
(1.75, 1.666e-9, 8.2e-11)
(2, 5.3e-8, 2.87e-10)

delay=0.69*R*C

The frequency will be 1/(2*4*delay), because a “high” and “low” level output on the ring oscillator will be valid for 4 delay times, equivalently. Thus, it will take two times the four delay blocks to form 1 frequency. It’s necessary to cross the lines so we can get an odd number of inversions, while exceeding the “hold time” of the “first” delay block when we feedback the inverted signal.

Basically you will have 4 phases of a clock, and for each phase, you will also have the inverted phase. 8 signals total. Some of them are overlapping one another.

d. Now, we will look at the bias generator. The circuit for the bias generator is as follows:

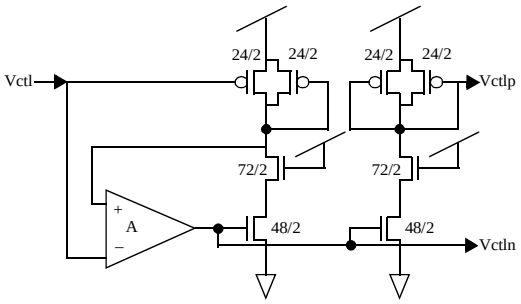
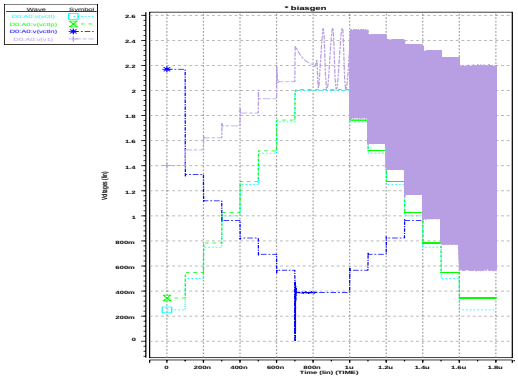


Figure 0.31 :Bias Generator

Implement this circuit in Hspice, and use the ideal voltage controlled voltage source for your amplifier. Use a value of 20 for A. This circuit automatically sets the Vctln and Vctlp voltages to the buffer delays to set the DC operating points of the delay cells such that the symmetric load is swinging reflected around its point of symmetry for a given Vctl voltage. Also, it is important to note that Vctl is the same as Vctlp. It must go through this business to obtain Vctln (which sets the bias current to the correct value, which sets the DC operating point of the buffer). Do a transient run in Hspice to verify that Vctlp is indeed very close to Vctl over a range of inputs for Vctl. Show a Spice transient simulation that goes for 1uS, and switches Vctl in a pwl waveform across a range of inputs between 0.5V and 2.0V. For extra points, explain how this circuit works.

Solution

See the following figure.



We will refer the two legs of current that contain 1 symmetric load in each leg of the bias generator to be “delay cell replicas.” These replicas serve the purpose so that we can put one of them in a feedback loop such that we can set Vctlp equal to Vctl (thereby setting the symmetric load to the lowest swing point for the given Vctl voltage). Through this process, it also generates the correct Vctln, which gives rise to a certain current ‘I’, will produce the desired voltage for Vres such that Vres=Vctl. Note the sizes of the transistors; the lowest NMOS device has the same width as an actual delay cell’s NMOS current sink device. Thus, both the delay cell replica and actual delay cell’s NMOS current source sink the same current. However, since there are two incoming current legs to the NMOS current sink of the delay cell, the current that the symmetric loads in the delay cell each see I/2, this automatically biasing the dc operating point of the delay cell to the symmetric load’s point of symmetry (theoretically).

e. Now, hook up the bias generator you just built with 4 delay cells, as shown in the first figure. For each control voltageVctlp from part c), verify your hand calculations with spice simulations. Show a spreadsheet of obtained frequencies vs. hand-calculation predictions, and in a separate column, calculate % error. Give a brief analysis of what you see. Print out all of the phases (4) of the clock, for a Vctl value of your choice.

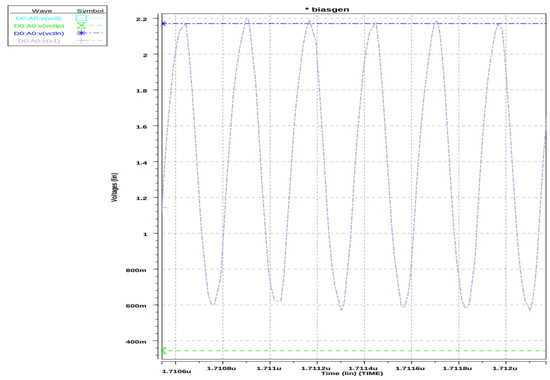
Solution

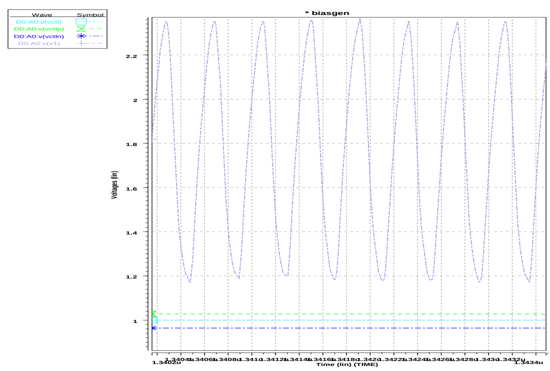
The spreadsheet is given here.

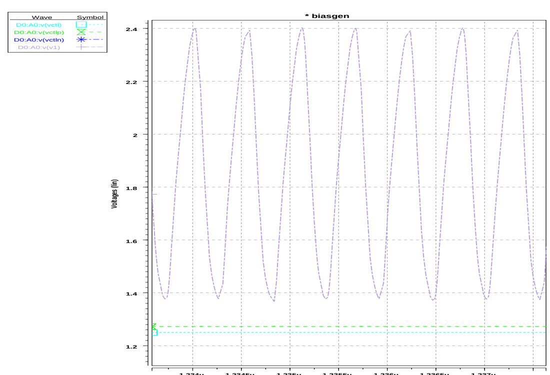
(Vctl, measured, calculated w/Rslope, calculated w/Rgm, % error from Rslope, % err from Rgm)

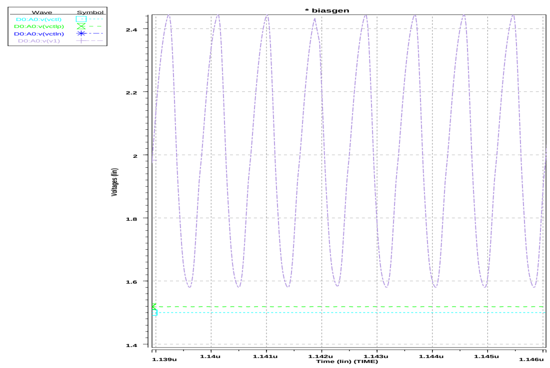
(2,	3.95MHz,	2.3MHz,	435MHz,	41%,	11000%)
(1.75,	384MHz,	75MHz,	1.5GHz,	80%,	390%)
(1.5,	1.6GHz,	232MHz,	2.6GHz,	85%,	162.5%)
(1.25,	2GHz,	576MHz,	3.6GHz,	72%,	180%)
(1,	2.5GHz,	1.13GHz,	4.7GHz,	54%,	188%)
(.75,	3.7GHz,	1.72GHz,	5.8GHz,	53%,	156%)
(.5,	4.2GHz,	2.29GHz,	6.9GHz,	45%,	164%)

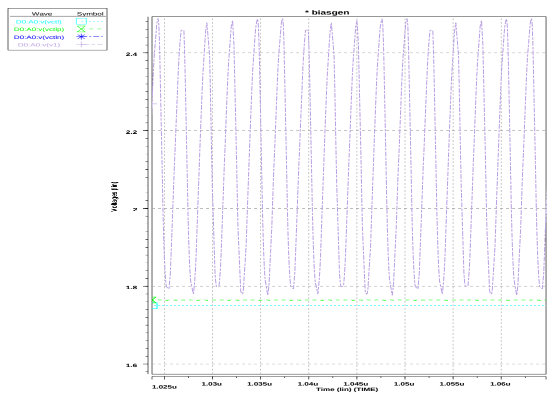
See the following figures.

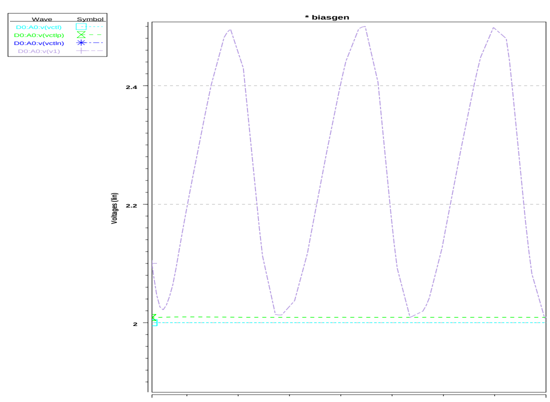

* biasgen
Wave
Symbol
Voltages (lin)
Time (lin) (TIME)


* biasgen
Wave
Symbol
Voltages (lin)
Time (lin) (TIME)


* biasgen
Wave
Symbol
Voltages (lin)
Time (lin) (TIME)


Wave
Symbol
D0:A0:v(vctl)
D0:A0:v(vctlp)
D0:A0:v(vctln)
D0:A0:v(v1)
* biasgen
Voltages (lin)
2.4
2.2
2
1.8
1.6
1.4
1.139u
1.14u
1.141u
1.142u
1.143u
1.144u
1.145u
1.146u
Time (lin) (TIME)


Wave
Symbol
D0:A0:v(vctl)
D0:A0:v(vctlp)
D0:A0:v(vctln)
D0:A0:v(v1)
* biasgen
Voltages (lin)
2.4
2.2
2
1.8
1.6
1.025u
1.03u
1.035u
1.04u
1.045u
1.05u
1.055u
1.06u
Time (lin) (TIME)


Wave
Symbol
D0:A0:v(vctl)
D0:A0:v(vctlp)
D0:A0:v(vctln)
D0:A0:v(v1)
* biasgen
Voltages (lin)
2.4
2.2
2
Time (lin) (TIME)

This begs the question: why are these OFF by so much?

The delay equation of 0.69*R*C is what we used. However, in Maneatis' paper, he calculates delay using just R*C, where R is Rgm that we have included in this problem set solution. If we go back and calculate our estimations using Maneatis' estimation of delay, we come up with less % error in the % err from Rgm column. However, it still does not explain the still glaringly large %error. If we look again at the figure in part (d), we can see that the lower swing limit of the buffers never reach the lowest point, Vctlp. This is due to the fact that we are not putting in enough delay elements so that the overall frequency is slow enough, so that the delay cells can input and output the full swing range. Thus, our estimation of R using the slope and 1/gm is innaccurate. The overall delay only allows "limited swinging." Another effect that may be appearing, is the much degraded gds output resistance of short channel devices. if gds begins to appear in the range of gm, then we will see a reduction in measured frequency vs. calculated frequency. In any case, the VCO does not need to be characterized in an absolute voltage to frequency relation; only that the transfer from voltage to frequency is linear, or at least the slope of the voltage to frequency curve has the same polarity at all times. When placed in a feedback loop, the non-linearities of the voltage to frequency curve of the VCO will be compensated for.