Homework 6 - Solutions

1 Problem 9-2 (Dally and Poulton)

Bundled Signaling Without a Clock: Events involving an alphabet of seven symbols can be encoded onto three lines without a separate clock line. The three lines encode both the value of the current symbol and the time at which each symbol starts (even if it is the same as the last symbol). Describe a signal encoding that meets this specification. What are the constraints on relative timing between the three lines?

Let the alphabet of seven symbols be a, b, c, d, e, f, g. Let the 3 lines used to transmit the symbols be A_0, A_1, A_2.

The following table shows the symbols and the corresponding values on each of the lines.

Current Symbol	A_2 A_1 A_0
a	0 0 1
b	0 1 0
c	0 1 1
d	1 0 0
e	1 0 1
f	1 1 0
g	1 1 1

First we consider the case when **the current symbol is different from the previous symbol**. In this case, the time of the current symbol is recognized as a transition on one of the three lines, and the value of the new symbol is decoded from the state of the lines after 1 bit cell time. Thus, if the current symbol is b and the next symbol is g, the lines ($A_2A_1A_0$) go from 010→111. Only the first transition (A_2 or A_0) indicates that a new symbol has started. (The transitions may be skewed due to timing uncertainty.) All other transitions are ignored for one bit time. The value of the line (111) at the end of the one bit time determines the value of the symbol.

The other case to consider is when **the next symbol has the same value as the current symbol**. For example, we transmit the symbols "ff" consecutively. In this case all 3 lines go to the 000 state after the first f. Since 000 has not been chosen to signify any of the 7 seven symbols, this guarantees that there will be a transition indicating that a new symbol (the next f) has started. Since the new symbol has the same value as the previous symbol, we ignore the 000 value on the line and use the previous line values (110) as the symbol value. If a third f is transmitted we transition from 000 back to 110 (the value of f) and proceed decoding mormally at the other end. So at the receiver, whenever a 000 is received the value of that symbol is interpreted as the value of the previous symbol.

For example, we transmit the sequence: [e d d f f a b]

We start the line with 000. Line values ($A_2A_1A_0$) in time are as follows: 000→101→100→000→110→000→001→010. The decoder at the receiver end detects 7 symbols corresponding to 7 transitions separated by 7 bit times. The two transitions to 000 indicate that the symbol d (100) and the symbol f (110) are repeated.

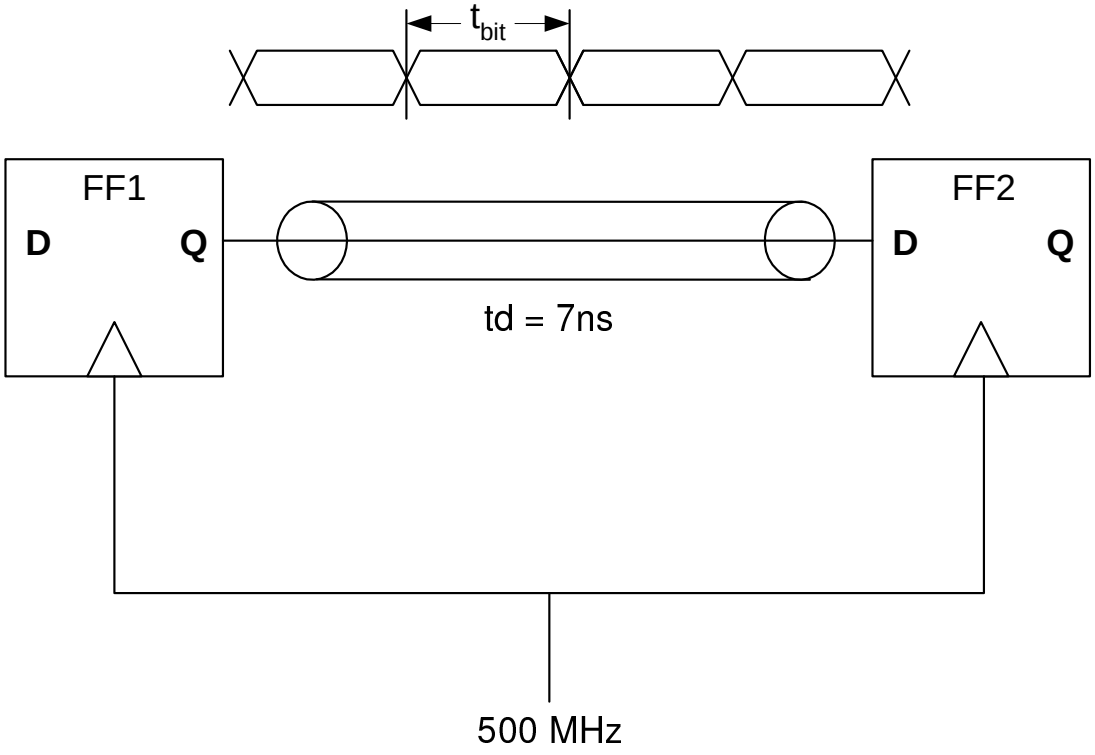
The constraints on the relative timing between these three lines is that the skew between the lines, A_0, A_1 and A_2, cannot exceed the bit time. Thus the transition from d to g (100→111) involves transitions on both A_1 and A_0 which have to take place 'simultaneously' - i.e. within the spacing of a bit cell. If A_1 wiggles first then it will indicate the start of a new symbol. The wiggle on A_0 has to happen within a bit cell time, otherwise the current symbol will be incorrectly interpreted as 110. However, as mentioned earlier, the

wiggle on A_0 is ignored as far as timing is concerned since the wiggle on A_1 all ready indicated the start of a new symbol.

This is one possibility of encoding 7 symbols on 3 signal lines. Other algorithms are possible.

2 Problem 9-3 (Dally and Poulton)

Operating Range for Synchronous Timing: Consider a system that operates at 500Mb/s using synchronous timing, as shown in Figure 9-58. A communications link in this system includes a 7-ns (3 1/2-bit-long) transmission line. The link has a rise-time of $t_r = 1ns$, an aperture time, $t_a = 300ps$, and a timing uncertainty (skew + jitter) of $t_u = 300ps$. Over what ranges of clock frequencies will the link operate with correct timing?



The data travels on the link from FF1 to FF2. The delay of the wire is $t_{wire} = 7ns$.

If there were no timing uncertainty and no required aperture time and rise time, the maximum delay of the wire would be:

$$t_{wire} \leq 4t_{bit}$$

I.e. as long as FF2 were clocking at some time less than the transition of the fifth bit, the fourth bit would get clocked correctly through FF2.

However, we do have timing uncertainty in our system and a required aperture time and rise time. So, our maximum wire delay is actually:

$$\boxed{t_{wire} \leq 4t_{bit} - [0.5(t_r + t_a) + t_u]}$$

Manipulating this equation, we solve for t_{bit} and get:

$$t_{bit} \geq \frac{7ns + 0.5(1ns + 300ps) + 300ps}{4} = 1.99ns$$

$$ClockFrequency \leq 503MHz$$

For the worst-case minimum delay we perform a similar analysis.

Again, if there were no timing uncertainty and no required aperture time and rise time, the minimum delay of the wire would be:

$$t_{wire} \geq 3t_{bit}$$

I.e. as long as FF2 were clocking some time after the transition of the 4th bit, $\geq 3t_{bit}$, the fourth bit would get clocked correctly through FF2.

However, again, we do have timing uncertainty in our system and a designated aperture time and rise time. So, our minimum wire delay is actually:

$$\boxed{t_{wire} \geq 3t_{bit} + [0.5(t_r + t_a) + t_u]}$$

Manipulating this equation, we solve for t_{bit} and get:

$$t_{bit} \leq \frac{7ns - 0.5(1ns + 300ps) - 300ps}{3} = 2.02ns$$

$$ClockFrequency \geq 496MHz$$

Thus, the link works for the following clock frequency range:

$$\boxed{496MHz \leq ClockFrequency \leq 503MHz}$$

At clock frequencies outside of this range, the above constraints will not be satisfied and the system will fail.

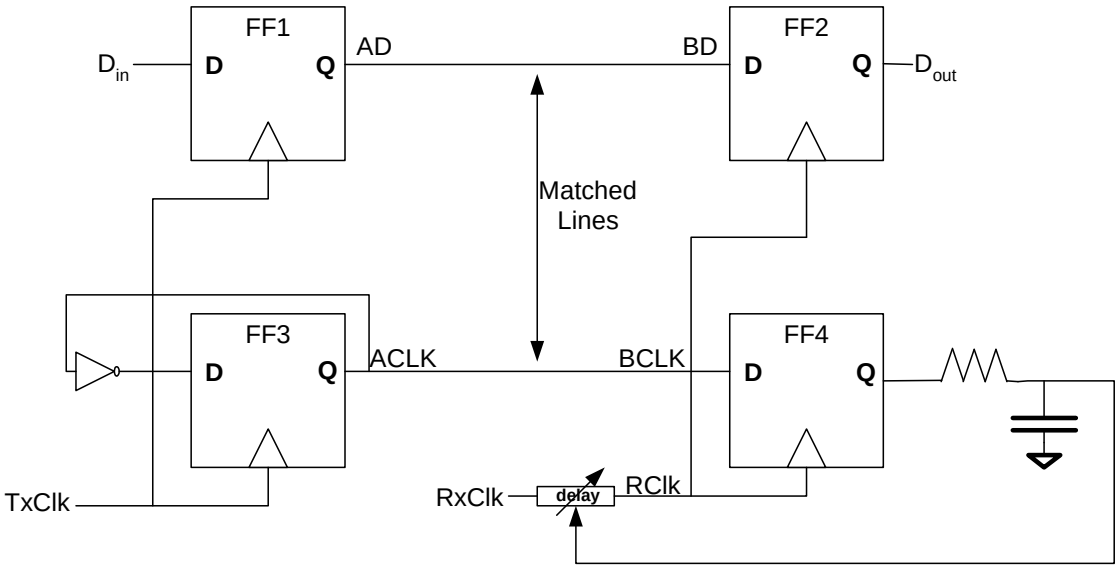
3 Problem 9-5 (Dally and Poulton)

Closed-Loop Timing: A friend is building a closed-looped timing system (like the one in Figure 9-38) for a communications link in his laptop computer. He has wired it up as shown in Figure 9-60. The $TxClk$ and $RxClk$ have exactly the same frequency, 100 MHz, but an unknown, constant phase difference. The

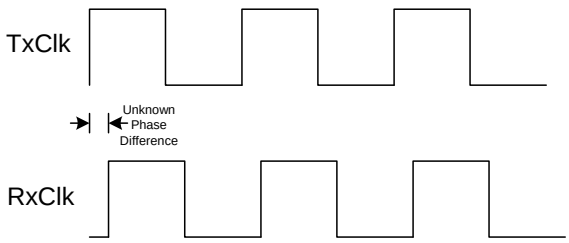
voltage-controlled delay line has a delay range of 0 to 20ns and has a delay that increases with an increasing control voltage. Your friend is puzzled that the system does not operate properly. Draw a timing diagram of this system. What is wrong? How would you fix it? Draw a diagram of a corrected circuit.

The nonideality that closed-loop timing is compensating for is that the receive clock has a fixed unknown offset from the transmitting clock. If there were no offset between the clocks, we could simply add a fixed delay to the receiving clock to ensure that it sampled at the center of the eye of the data. However, there is an unknown fixed delay and we compensate for it with closed-loop timing.

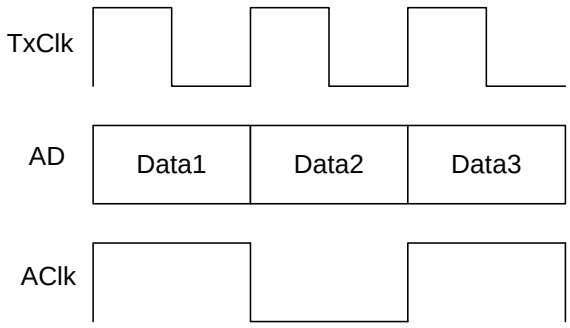
Let's analyze the circuit as it is currently:



The system has a transmit clock and a receive clock with a fixed, unknown phase difference, as shown below.



At the transmitter side, we transmit the data, AD, and AClk according to the transmit clock, as shown below. Note: in the figures below we are ignoring rise time, aperture offset of the flip-flops etc. for simplicity. In actuality, these timing constraints need to be taken into account. Again, for simplicity and clarity of explanation we are ignoring them in the figures.



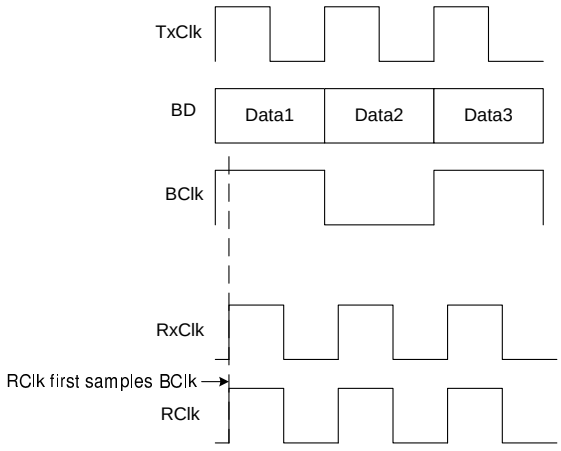
Note that since the flip-flops are single-edge triggered, the data and the transmitted clock, BClk, only transition on the rising edge of TxClk.

Some delay later, the delay of the matched transmission lines, the data, BD, and BClk reach the receiving flip flops, FF2 and FF4.

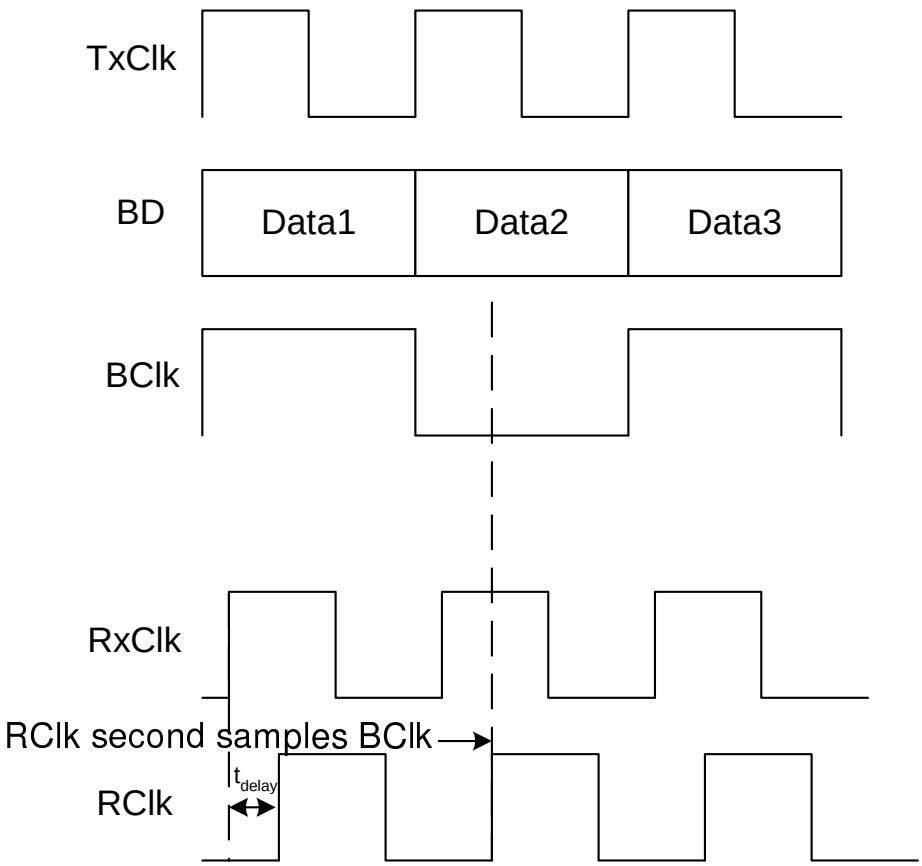
For simplicity, let's say the delay of each of the matched lines is a multiple of the bit time. (Note that the delay of the matched lines could be anything and the following analysis would be the same. We choose an arbitrary delay, an integral number of bit times.)

Let's assume that the delay line is initially 0ns. In other words RxClk and RClk are exactly synchronized to begin with.

Thus, initially the receive clock, RClk, will sample BClk and the incoming data as shown below.

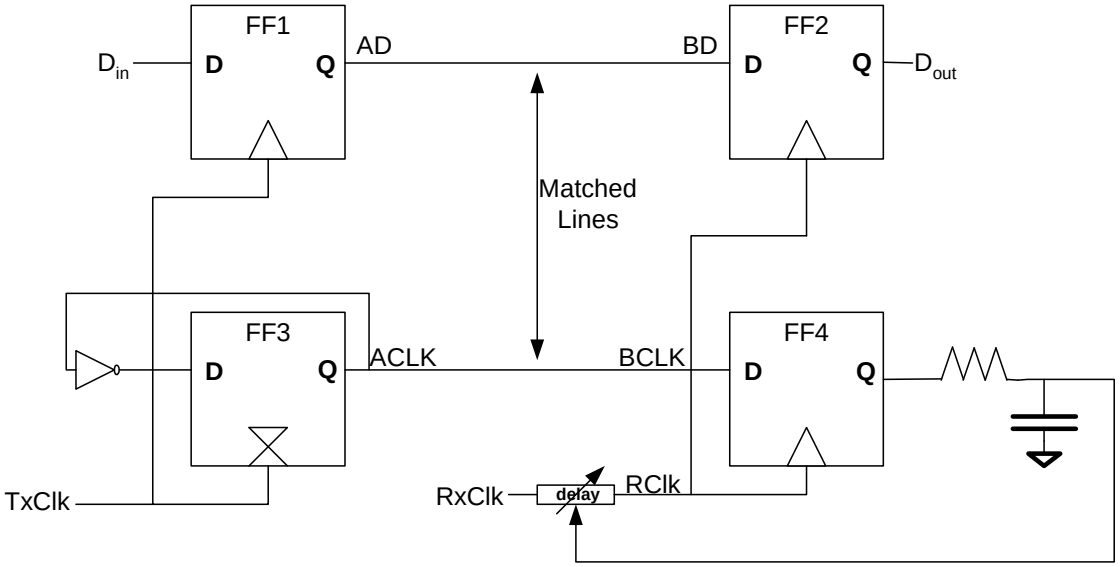


Since BClk is high, the RC filter, acting as an integrator, will increase the delay between RxClk and RClk by an amount, t_{delay}, as noted in the diagram below.

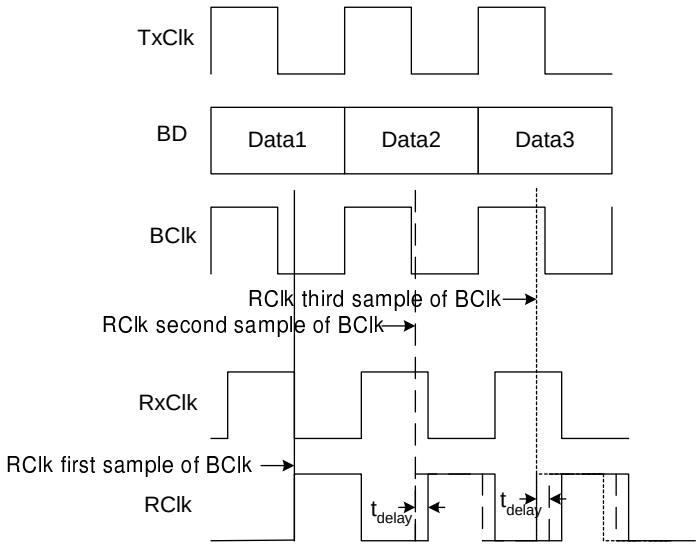


Since BClk is low on the second sampling - as shown below - the RC filter, again acting as an integrator, will decrease the delay between RxClk and RClk. This decrease in delay will be less than the initial increase in delay (from the first sampling) since $\Delta Q = C\Delta V$. And since ΔV is smaller, so will the change in charge be smaller. Remember, the RC filter is acting as an integrator. Thus, the net effect of the first and second sampling is a slight increase in the delay between RxClk and RClk. Extrapolating this result, RClk will continue being delayed until the voltage on the capacitor equals Vdd/2. At this point the positive sample and negative sample will exactly cancel (since $\Delta Q = C\Delta V$ for both the high and the low sample), and the delay will jitter between these two points.

There are several options we could take to fix this circuit. We choose to change the FF3 flip-flop to dual-edge triggered and initialize the delay line to an initial delay of 5ns. If the delay line were at 0 ns at the initial start-up, if BClk is slow compared to RClk (i.e. RClk samples BClk as low on the first sample), the delay line would not be able to adjust - i.e. become less than 0ns delay.



Now our timing diagram looks as shown below.



Upon the first sample, RClk samples BClk as low. This decreases the delay of RClk with respect to RxClk by an amount t_{delay}. This RClk, that has a smaller delay from RxClk than before, is shown in the dashed lines.

Upon the second sample, RClk again samples BClk as low. This again decreases the delay of RClk with respect to RxClk by approximately the same amount, t_{delay}. This new, lesser-delayed version is shown in the dotted lines on the graph. At the third sample, as shown in the figure, the value of BClk is high and it increases the delay between RxClk and RClk by approximately the amount t_{delay}.

Thus, in subsequent steps, RClk will dither about these last two delay points (the RClk shown in the dashed lines and the RClk shown in the dotted lines). This approximately centers the sampling clock, RClk, on the eye of the incoming data BD.

Note that in the above diagrams, the offset caused by flip-flop delays, rise time delay, aperture delay and any timing uncertainty were not taken into account. This was done for clarity in explaining the functionality of closed-loop timing. In practice, these non-idealities need to be taken into account.