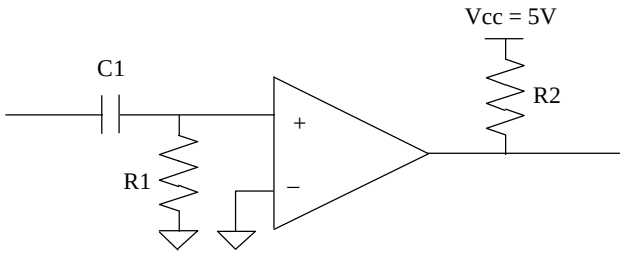
EECE 690/890
Digital Radio Hardware Design
HW #4
Due Thursday 9/24/98

*** indicates a problem that is optional for 690 students

NOTE: The problems in this assignment are somewhat "open-ended" and are designed to be similar to those you might be required to solve in a hardware design job. In most cases, they derive from issues discussed in class, but the solution methods needed must be recalled from earlier cources (e.g. EECE526). You are encouraged to help each other out in understanding the problem and the solution processes needed. HOWEVER your final solutions must be worked on your own, and explained in your own words.

This is a good time to get the phone number of your team/company colleagues so that you can work together on some of this!

1. As discussed in class, the raw data waveform coming from our receiver will be only a few hundred mV in amplitude, and will be "riding" on a DC offset which is not well defined. Hence a threshold adjustment and "data-slicer" circuit will be needed between the receiver and the digital PLD circuitry. In this problem, you will design such a circuit. One possible architecture for the circuit is shown below.



Assume the comparator is an LM2903[1], for which the output is an open-collector type and the input bias current is 25 nA. Values are needed for the resistors and capacitor.

a) Select a suitable value for the output pull-up resistor R2. Assume that our serial data stream's rate is 100 k b/s, that the output is connected to a 10 pF load (the input to the PLD), that you want the rounding of the waveform to be minimal (e.g. rise/fall times on the order of 20 ns or less), and that you want to draw the least current possible.

HINT: Sketch the RC circuit composed of the pull-up resistor and load capacitor, and think about what waveform will appear when the output transitions from low to high (open-collector output transistor goes from conducting to an open-circuit). How fast can

---

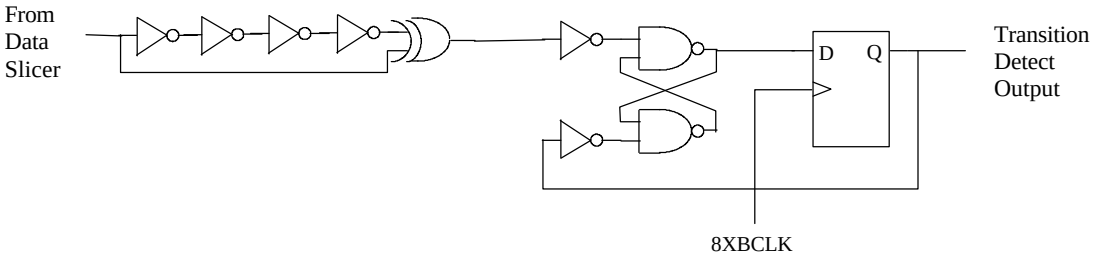
[1] Although this is a single-supply device, the input circuitry allows the input signals to be as low as -500 mV without problem. Hence, it is OK for the signal on R1 to go negative.

the output voltage change? (Ignore the fall-time requirement. Assume the transistor can discharge the capacitive load faster than it can charge it.)

b) Pick a value for the input resistor R1 (which is used to provide a DC path to ground for the bias current leaving the comparator input pin). Select a value that is reasonable, given the size of the input signal and justify your definition of "reasonable". The other (competing) requirement here is that the resistor value should not be too small, since that could cause the C value solved for below to be too large.

c) Pick a value for the capacitor C1, assuming that the input to the circuit is an ideal voltage source, that the initial bias error[2] is on the order of 500 mV, and that the capacitor should "acquire" the voltage needed to bring the average voltage input to the comparator to "near" zero within 100 ms or less. How near is near enough? Note that your C value cannot be arbitrarily small since you must make sure that the capacitor voltage will not change significantly if a long string of zeros or ones are received. (What is the longest string of all zeros or all ones that you could get with our packet design?)

2. One possible design for the "transition detector" circuit needed in our bit-sync is shown below:



a) Assume a serial data stream composed of the bits 1 0 1 1 1 0 emerges from data slicer of problem 1 and is supplied as input to this circuit (the "1" is the first bit to arrive). Sketch the input waveform and the waveform at the output of the XOR gate. Assume that the data rate is 100 k b/s, and that the delay time through each gate/inverter is 5 ns. Draw your waveforms approximately to scale.

b) Find the output waveform from the full circuit, assuming that the signal "8XBCLK" is a squarewave clock signal at 8 times the bit rate. HINT: Notice that the circuit following the XOR gate is an SR latch with active high set and reset.

---

[2] The purpose of C1 is to block DC. In operation, it charges to the voltage needed to drop the DC bias voltage out of the demodulator down to zero at the comparator terminal. The "bias error" is the difference between the voltage across the cap at the start of signal reception and the voltage needed to achieve approximately zero at the comparator terminal for proper data slicing.

3. Recall the state diagram for the bit synchronizer discussed in class, which takes the output of the circuit of problem 2 and creates a "regenerated clock" in phase with the received data stream. In this problem you will work out some of the details of a state machine that might be used to implement the bit-sync.

a) Create a state transition table in which the first column lists the current state, the second column lists the possible input values that determines what state transition is taken at the next clock, and the third column lists the next state. Show all values in the table in binary form (e.g. state 8 is 1000). The first line of the state table is shown below to help get you started:

Current State	Input	Next State
0000	X	0001

Note that state tables such as this can be used with many digital circuit design tools to generate the state machine automatically. I.e. If you had such a design tool, your design would be done at this point!

b) *** Assume you do not have an automatic state machine synthesis tools like that described above and that you must synthesize the logic yourself. Write a boolean expression for the combinational logic needed to create the D register input for bit 0 (the LSB), given the state table of part (a). In writing your expression, use D0 for the combinational logic output variable, Y0 ..Y3 for the current state bits, and A for the input from the bit sync.

c) *** Draw a schematic for the logic specified by the equation found in part (b). You do not need to reduce your solution to minimal terms. Just draw the direct AND/OR gate implementation. Then convert your design to one that uses NAND gates only.

d) *** Oops. In testing the circuit, you have discovered that the overall design has a problem. If a data stream transition occurs at state 3, then the output of the transition detector of problem 2 will be gone by the time a decision is made at state 6. Hence, if the incoming bit rate is almost identical to 8XBCLK / 8, then the circuit will take a very long time to achieve lock (theoretically it may never reach lock!). Fix the circuit, and explain how your solution solves the problem.

4. A significant problem with using packets in digital radio designs is the amount of "overhead" involved in sending items like sync patterns, IDs, and CRCs along with the data. This causes the transmitted bit rate to be higher than that of the data, and hence, the signal bandwidth is larger than it would be if only the data were sent. In turn this reduces the number of channels available for use within a fixed spectrum allocation.

   Find the transmitted bit rate for our system and compare it with the bit rate from the ADPCM codec.

5. The cost of digital PLDs is directly related to the size of the digital circuit that is being implemented. For our application, which is "register intensive", a good estimate of size is the number of registers needed.

a) Draw a block diagram for the PLD circuitry for both the transmitter and receiver interfaces and then estimate the number of registers needed. Think through this carefully. There may be registers needed within blocks such as the "MUX" shown in the class diagram. You are not expected to get an exact count (that would require doing the full design). However, you should get close, and you must explain or tabulate where the registers are needed and how many are needed in each location.

b) The Altera chip we plan to use has 128 registers in it. Is it potentially feasible to get all the circuitry into one chip?

6. *** The final design requirement in the PLD is to generate the necessary clock signals to be supplied to the ADPCM codec. This turns out to be a somewhat "non-trivial" problem.

   Download the ADPCM data sheet from the web and study the clock signals required. Then determine a strategy for where the bit clocks are generated. Pick from one of the following three choices of architectures and justify your answer (or come up with a better idea!):

   - The BCLK and X/RSYNC signals are derived from the clock signal generated by the bit sync. This is done in both the transmitter and receiver.
   - The BCLK and X/RSYNC signals are derived from the clock signal generated by the bit sync for the receiver path, but for the transmitter path, the BCLK and XSYNC signals are derived by dividing down the 10.368 MHz crystal reference of the codec chip.
   - The BCLK and X/RSYNC signals are derived from the clock signal generated by the bit sync. This is true for both the transmitter and receiver of the mobile unit, but at the base unit, the clocks are derived from the crystal reference.