Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science

6.012
Microelectronic Devices and Circuits
Fall 2016

Design Project
Assigned: October 20, 2016
Due: November 10, 2016, 4:00pm

This project may be done in groups of two. If you choose to work with a partner, turn in one report for both partners. The design process is long – start working early.

1 Background and Motivation

People today are familiar with the term "Internet of things" (IoT) as more and more of daily life becomes infused with "smart" technology. Buildings and other items are being embedded with electronics, software and sensors that allow them to communicate and to be controlled over a network. The addition of these electronics has significant allure: with it, for example, the various aspects of a home (lighting, heating, appliances, etc.) can be controlled remotely, easily, and more efficiently. This is but one set of applications and with the advent of the IoT comes potential revolutions in medicine, environmental monitoring, manufacturing, energy management and many other fields.

The more ubiquitous these technologies become, the more people must concern themselves with the power each device or each sensor consumes. A low-power solution for each device is thus desired such that in aggregate these many devices do not themselves become a significant energy drain.

Another consideration is that the devices need to also be able to communicate at all times over the network. Wireless communication is often implemented using electromagnetic waves. By changing the waves' characteristics, information can be encoded within the wave. One often used characteristic to encode data is the frequency of the electromagnetic wave. Frequency shift keying, FSK, is a digital communication scheme where a frequency represents a digital code word. With two frequencies, F_0 and F_1, a 1-bit code word can be transmitted – to send a '0' frequency F_0 would be transmitted and F_1 transmitted for a '1'.

For this design project you will be designing the frequency generator for a distributed sensor network. At any given time there may be more than one pair of sensors trying to communicate. If multiple sensors were to use the same frequencies to communicate, then there would be interference. One way to avoid this is to have multiple frequencies a sensor can choose to use to communicate. This is the approach that you will be designing for. You will need to design a frequency generator capable of generating eight different frequencies.

2 The Frequency Generator

Using a 1.5µm MOSFET technology, you will be designing a frequency generator that will output a square wave. The frequency of the square wave will need to be controllable and produce **8 frequencies: 10, 20, 30, 40, 50, 60, 70, and 80 MHz**.

For the frequency generation we will use a current starved ring oscillator. The "starved" part refers to the fact that we limit the amount of current the inverter can use to switch states. By changing how much current we allow the inverter to switch, we can change the frequency of the oscillator.

The frequency generator will have three building blocks:

Block 1 – The current-starved inverter is a CMOS inverter with its current supplied limited by a current mirror. A greater current leads to less inverter propagation delay, but at the cost of greater power dissipation.

Block 2 – The ring oscillator is a loop of an odd number of current-starved inverters that produces an oscillation that is our generated frequency. The oscillation frequency is inversely proportional to the inverter propagation delay. Therefore, control of the inverter's mirror current allows the speed of the ring oscillator to be tuned.

Block 3 – The Digital-to-Analog Converter, DAC serves to digitally change the amount of mirrored current in the current starved inverters thus changing the frequency of oscillation of the ring oscillator.

As electronics become ever smaller and more portable, power has become a crucial design constraint for circuit designers. In this project, you are asked to **minimize total power dissipation while meeting the design specifications**.

3 Implementation

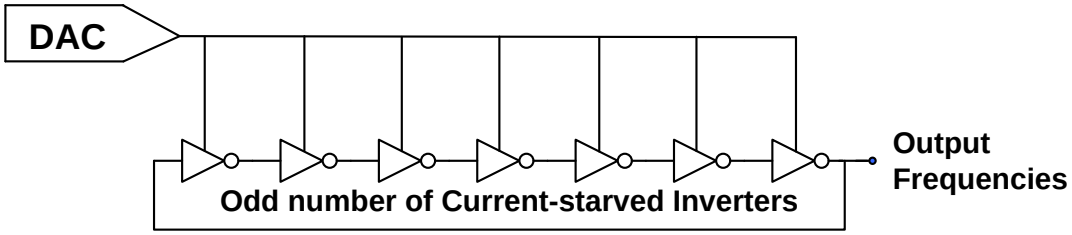

Figure 1: The frequency generator

The DAC will set the bias current that starves the inverters which in turn set the output frequency. An odd number of inverters will be chained together to form a ring oscillator.

3.1 BLOCK 1: THE CURRENT-STARVED INVERTER

We will use a current mirror to starve the current that will pull the inverter high. A current mirror is a circuit block that provides an accurate and robust method to copy (or "mirror") a reference current I_{REF} to other parts of the circuit. As shown below, both transistors M1 (reference) and M2 have the same V_{SG} and W/L. M1 is always in saturation because its $V_{SD1} = V_{SG1} > (V_{SG1}+V_{Tp})$. When M2 is also in saturation, M2 will produce the same current as M1 (neglecting channel-length modulation). This is demonstrated below:

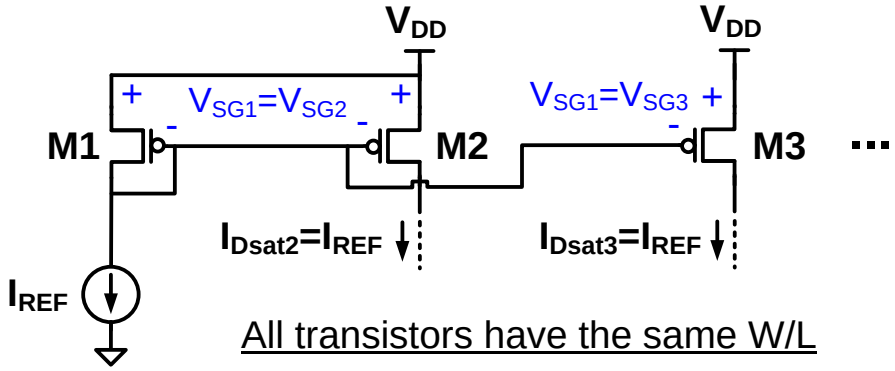

Figure 2: Current mirror explanation

$$I_{Dsat1} = I_{REF} = W/(2L)*\mu_p C_{ox}*(V_{SG1}+V_{tp})^2$$
$$I_{Dsat2} = W/(2L)*\mu_p C_{ox}*(V_{SG2}+V_{tp})^2$$

Because of identical W/L and $V_{SG1} = V_{SG2} \rightarrow I_{Dsat2} = I_{Dsat1} = I_{REF}$

The current mirror is not limited to only one branch. In figure 2, the same principle applies to M3, which also mirrors I_{REF}. Because of this, **there will be only one reference current branch for all your inverters**.

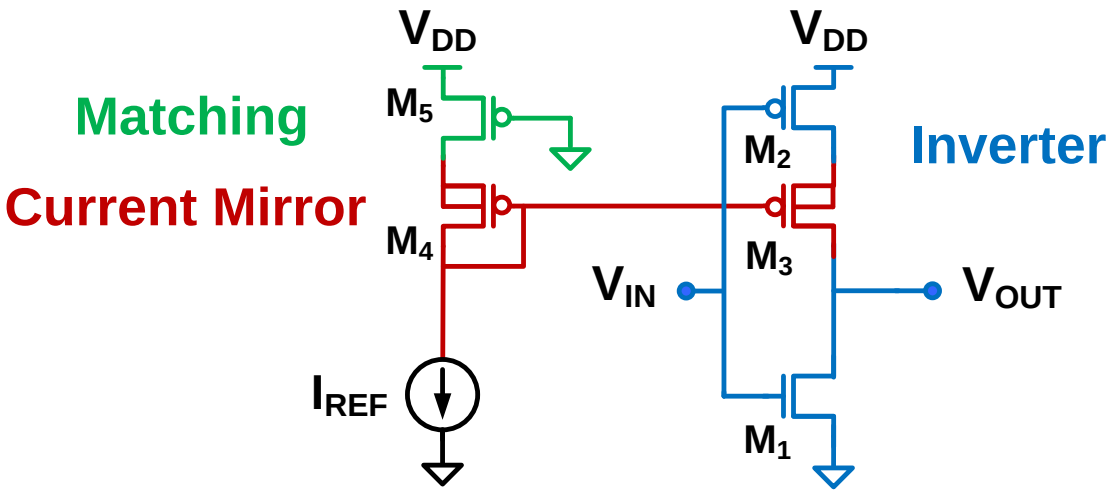

Figure 3: The current starved inverter

We will use the current-starved inverter for our ring oscillator seen in figure 3. The current-starved inverter consists of a CMOS inverter (M1 and M2 in blue) and a current mirror device M3 (in red). When the inverter's input is high M1 is turned on and will pull the output low. When the input is low, M1 is turned off and M2 is turned on. With M2 on, the current pulling the output high will be limited by M3.

A current mirror works because all transistors in the mirror share the same source-to-gate voltage. Because M2 in the inverter is connected to the source of the current mirror M3, the small voltage drop across the transistor could affect the source-to-gate voltage of M3. To keep all source-to-gate voltages the same, the green PMOS M5 is placed at the source of the transistor M4 that is setting the reference current.

The current-starved inverter's operation is described below. The capacitance C_{inv} is representing the input capacitance of the next inverter.

- When the input goes from LOW→HIGH, the NMOS gate turns on while the PMOS gate turns off. Just like a standard CMOS inverter, C_{inv} is quickly discharged through the NMOS gate and the output goes from HIGH→LOW. Because the inverter's PMOS gate is turned off, the current mirror cannot source current from VDD and does not provide current.
- When the input goes from HIGH→LOW, the NMOS gate turns off while the PMOS gate turns on. The current mirror begins supplying current (nominally I_{REF}) through the PMOS gate. This current charges C_{inv} and the output goes from LOW→HIGH.

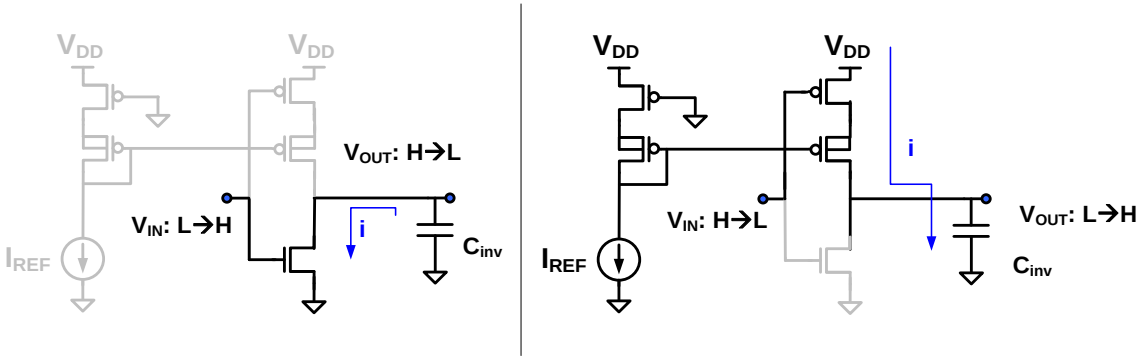

Figure 4: Operation of current starved inverter

As the designer you will have the power to size the current mirror (shown in blue in figure 5). While you have control to size these transistors, they must be the same. The red transistors will be sized for you (W=10um, L = 1.5u). Later you will be designing a DAC that will change the reference current, which will change the frequency of the ring oscillator. The relationship of the frequency of oscillation to the reference current will be explained more in the next section, but for now just know that you will be designing the current mirror to mirror 8 different currents. **You need to design the gate sizes of the current mirror such that the circuit will be able to mirror all 8 currents.**

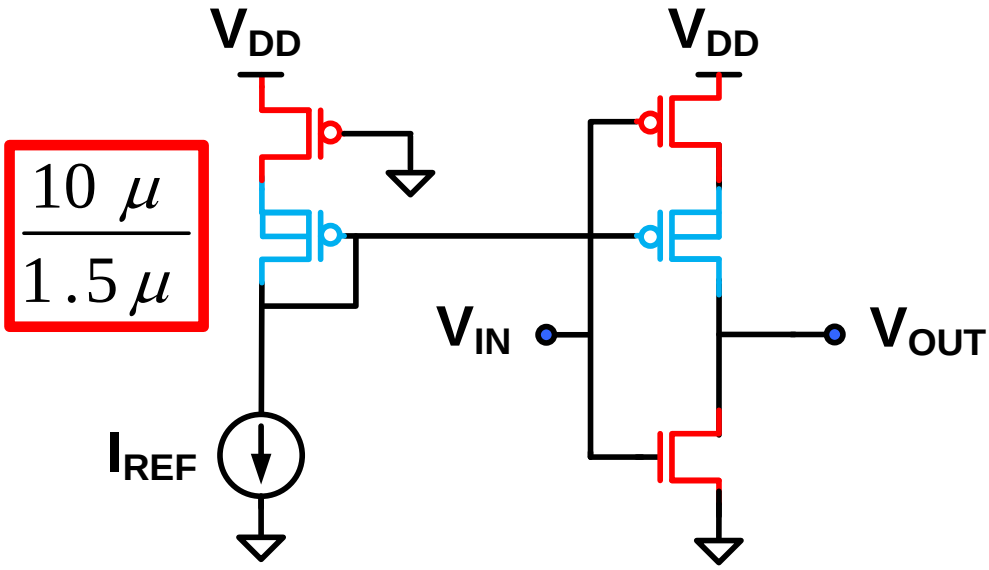

Figure 5: Current starved inverter

3.2 BLOCK 2: THE RING OSCILLATOR

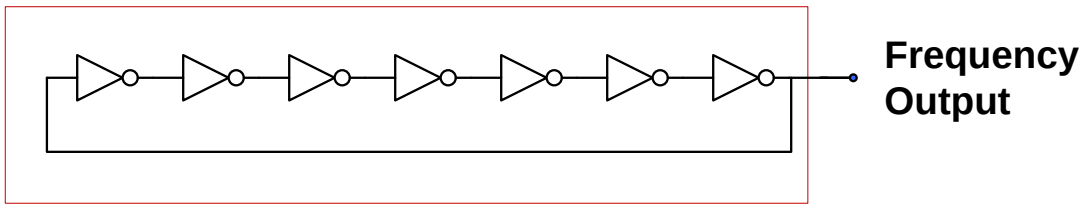

Figure 6: Ring oscillator

The ring oscillator consists of a loop of an odd number of inverters. Because of the odd number of inverters, the output is constantly oscillating between LOW and HIGH. In our design, the inverters are the current-starved inverters designed in Block 1. As mentioned above, **the reference current will be controlled by the DAC.**

The oscillation frequency of the ring oscillator is inversely proportional to the inverter propagation delay by the following relationship:

$$f_{clock} = 1/(2*N*t_P)$$

N = number of inverters = 7
t_P = propagation delay of each inverter = $(t_{PLH} + t_{PHL})/2$

Therefore, frequency depends on the number of current starved inverters and the value of the current in the current mirror device. **You are to design the current-starved inverters such that the ring oscillator oscillates at the required frequencies using minimal power (i.e. minimal I_{REF}). The two free design parameters are the number of inverters and the reference current.**

In addition to the ring oscillator needing to oscillate at all the required frequencies, the output of the ring oscillator needs to be at or above **90% VDD for 20% of the period, of the oscillation, T_{cycle}** and at or below **10% VDD for 20% of the period**; as diagramed in figure 7.

$$T_{high}/T_{cycle} \geq 20\% \text{ and } T_{low}/T_{cycle} \geq 20\%$$

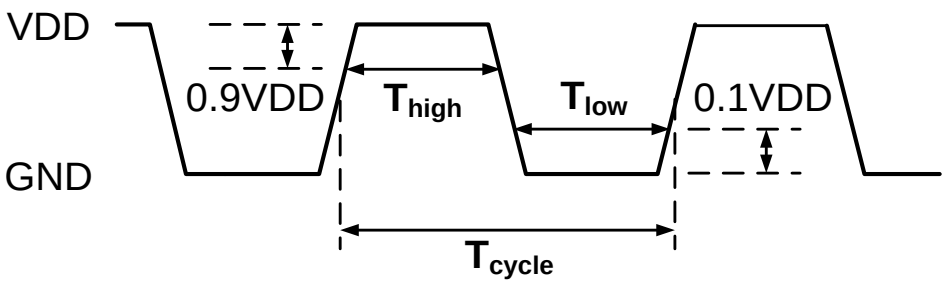

Figure 7: Explanation of high and low time requirements

3.3 BLOCK 3: THE DAC

The digital to analog converter (DAC) will replace the current sources in the figures 3 and 5 and set the reference current for the current-starved ring oscillator. The circuit we will use to do this is shown in figure 8 outlined in blue, connected to the ring oscillator circuit outlined in red.

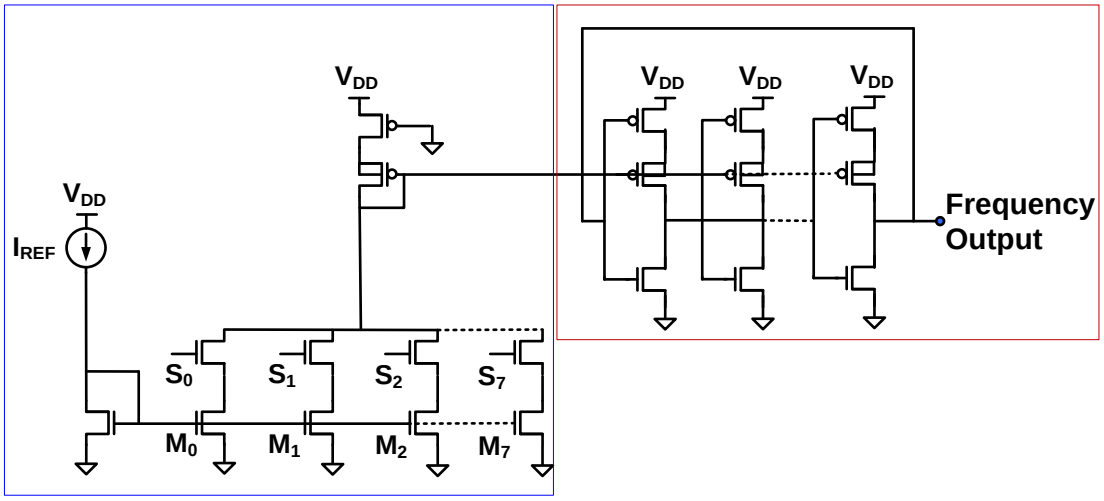

Figure 8: DAC and ring oscillator transistor level

The DACs reference current is mirrored by an NMOS. The other halves of the mirrors are sized such that when the NMOS switch ($S_{0\text{-}7}$) in figure 8 is turned on it will set the current necessary to starve the oscillator the right amount to achieve the required output frequency. Only one switch will be turned on at a time. **To turn on the switches use a 3V gate voltage. To turn the switch off, tie the gate to ground**. As an example of operation: When S_0, and only S_0, is turned on there should be the right amount of current supplied by M_0 such that the current starved oscillator has an output frequency of 10 MHz. When S_1, and only S_1, is turned on there should be the right amount of current supplied by M_1 such that the current starved oscillator has an output frequency of 20

MHz. The following output frequencies for the ring oscillator correspond to each switch being turned on.

- S_0 = 10 MHz
- S_1 = 20 MHz
- S_2 = 30 MHz
- S_3 = 40 MHz
- S_4 = 50 MHz
- S_5 = 60 MHz
- S_6 = 70 MHz
- S_7 = 80 MHz

Your design task for the DAC is to size the NMOS transistors M_0-M_1 and S_0 -S_1 as well as the current mirror PMOS transistor. **All the transistors must be sized such that all transistors except the NMOS switches are in saturation for any value of current being mirrored to the ring oscillator**.

3.4 DESIGN PARAMETERS AND SUMMARY OF SPECIFICATIONS

To meet the specs you are able to choose the reference current and size the black transistors in in the DAC as shown in figure 9. While you size the blue transistors supplying the current to the ring oscillator, all blue transistors must have the same sizing. The red transistors must be sized to have a width of 10µ and a length of 1.5µ. The last design parameter you have control over is the number of current starved inverters in the ring oscillator (remember it has to be an odd number of inverters to work).

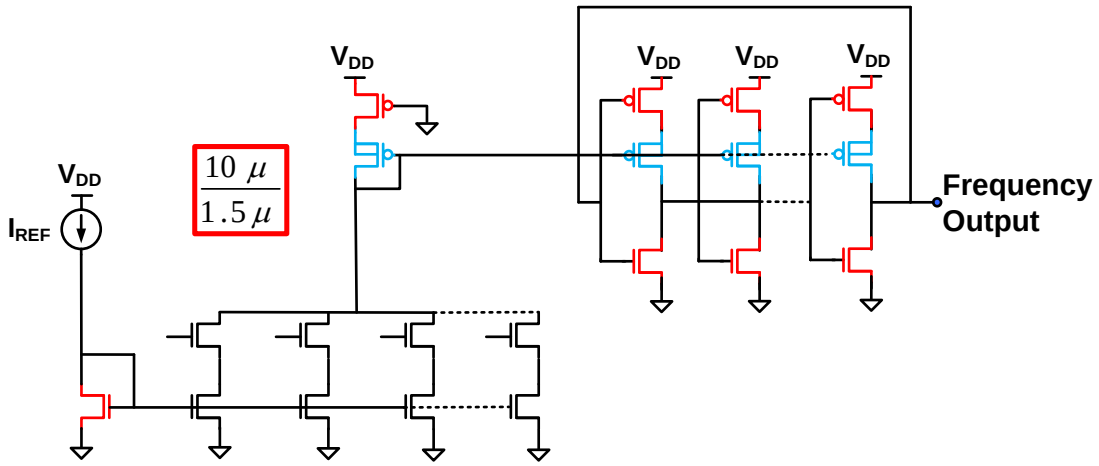

Figure 9: Sizing Restrictions

Table 1: Summary of Specifications

Voltage supply	• V_{DD} = 3V
Total average power dissipation	• P_{avg} < 3.3 mW include CV^2f power
Largest width transistor	• $W \leq 500\mu m$
Block 1: The current-starved mirrors	• Size W_p and L_p

Block 2: The ring oscillator	• Choose an odd number of current-starved inverters
	• Output frequency must be within 1% of:
	  ○ 10 MHz, 20 MHz, 30 MHz, 40 MHz, 50 MHz, 60 MHz, 70 MHz, and 80 MHz
	• Output voltage must be
	  ○ 90% VDD for 20% of T
	  ○ 10% VDD for 20% of T

Block 3: The DAC	• Minimize the reference current to keep the power dissipation low. • Provides V_{SG} necessary to set the current in the current-starved inverters so that the ring oscillator can output the 8 required frequencies • Transistors in the DAC are sized so that all transistors except the switches (S_0-S_7) are in saturation

3 Calculations

Before simulating, you should make sufficient calculations to choose appropriate transistor sizes to meet specifications. You should get a good feel for the tradeoffs you will need to make and which performance parameters are affected by your sizing decisions. Be prepared to explain your calculations and tradeoff decisions in your design report. Each of the calculations should be followed by a simulation to verify that your design actually meets the specification.

3.1 MOSFET PARAMETERS

The MOSFETS we have available in this process have the following properties:

1. Threshold voltages are V_{Tn} = 0.5V, V_{Tp}= -0.5V.
2. Oxide thickness is t_{ox} = 15nm.
3. Mobility: μ_n=220cm^2/Vs for the NMOS, μ_p=110cm^2/Vs for the PMOS.
4. Junction capacitance (CJ) = 1x10^{-4} F/m^2 NMOS, 3x10^{-4} F/m^2 PMOS.
5. Sidewall junction capacitance (CJSW) = 5x10^{-10} F/m NMOS, 3.5x10^{-10} F/m PMOS.
6. Built in potential Φ_B (PB) = 0.95V NMOS, 0.9V PMOS
7. The parameter LAMBDA=100 mV^{-1} is known for 1.5µm long devices. We will only use L=1.5µm devices in our design.

3.2 MINIMUM TRANSISTOR SIZE

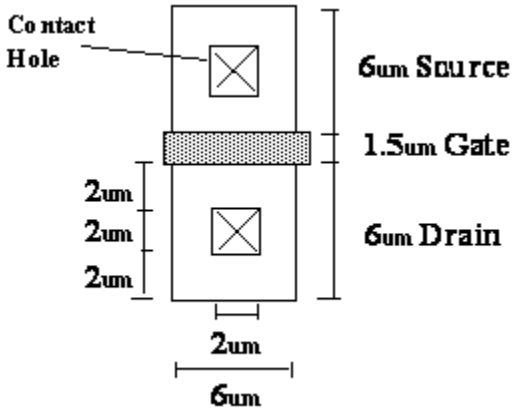

Figure 10: Minimum transistor

Minimum Size Transistor: Top View

A minimum size transistor diagram is shown above. The minimum gate length is 1.5 µm and minimum gate width is 6µm. In order to contact the source and drain, we must leave a 2 micron space for the contact itself and 2 microns on either side as overlap. This ensures that the contact hole is over only the diffused area, and not the surrounding bulk silicon. Knowing the geometry of the transistor is important in calculating parasitic device capacitances.

3.3 FREQUENCY CALCULATION

Parasitic capacitances arise because there is a pn junction between the source or drain region and the substrate. This junction should remain reverse biased, so we need only worry about depletion capacitance. There are two capacitance parameters in the model statements above, CJ and CJSW. CJ [F/m^2] is the capacitance per unit area of the bottom of the source or drain region. You specify the area of the source or drain in the dimension of the individual transistor. CJSW [F/m] is the capacitance of the "sidewall," or perimeter, of the source or drain region. It is specified as a capacitance per unit length because the other dimension, depth, is a parameter of the process, not controllable by the circuit designer. Again, you specify the perimeter of the source and drains in the dimension of the transistor. The capacitances calculated from CJ and CJSW are the worst-case (highest) zero voltage capacitances of the p-n junctions.

As an example, let us find the areas and perimeters for the minimum size transistor. The gate length is 1.5 µm and the gate width is 6 µm. The perimeters of the source (and drain) are 6 + 6 + 6 = 18 µm, since the side near the gate is accounted for in the gate capacitance. The areas are 6 x 6 = $36\mu m^2$. The area of the gate region is 1.5 x 6 = $9\mu m^2$. The total area of the transistor is thus 36 + 36 + 9 = $81\mu m^2$.

You must hand calculate the parasitic drain-substrate, source-substrate, and gate-source capacitance for your transistors, as they can have a significant effect on the speed and power dissipation of the circuit. To simplify your hand calculations, assume that the capacitors are equal to the zero voltage capacitance. This will produce a conservative design, as the average capacitance is less than the zero voltage capacitance.

Once you have calculated the parasitic capacitance values, use these values to determine the reference current (I_{REF}) you will need to obtain the 8 required frequencies. To do this, calculate the output capacitance (C_{inv}) of the inverter in the ring oscillator, which is simply the input capacitance of the next inverter. t_{PLH} can be calculated using C_{inv} and I_{REF} using the following equation:

$$i = C*dV/dt \rightarrow I_{REF} = C_{inv}*\Delta V/t_{PLH} \text{ , where } \Delta V = V_{DD}/2$$

With t_{PLH} known, you can calculate the frequency of oscillation (assuming $t_{PLH} >> t_{PHL}$):

$$f_{clock} \approx 1/(N*t_{PLH})$$

3.4 POWER CALCULATION

You should also use your capacitance calculations to estimate the power consumption of your entire circuit by determining how much current will be flowing to charge the next stage's input capacitance and for how long.

$$P_{avg} = \frac{\int_{T0} VIdt}{T_0}$$

Power dissipation is one of the most important parameters to optimize in circuit design. As a result, **roughly 15% of the final grade on the design project will be based on how far below the target 10 mW is your design while meeting all of the other specifications.**

To determine the total power dissipation in LTspice, you should run a transient simulation and measure the current flowing into the positive terminal of the voltage source you use for V_{DD}. After running the simulation, you can plot the current in this node by moving your cursor over the positive terminal of the voltage supply and clicking. A waveform labeled "I(VDD)" should appear. You can then export this waveform to TXT format by going to **File → Export** in the waveform view. You can open this TXT file in your spreadsheet program, and calculate the average I_{DD} over one period of oscillation ($T = 1/f_{clock}$). Using this, you can calculate the total average power dissipation as follows:

$$P_{avg} = V_{DD}*(\text{average } I_{DD} \text{ over one period})$$

4 Simulation

After you have calculated a preliminary design, you must simulate your design to verify its performance and modify your design parameters based on simulated results. Please refer to the SPICE help file available on Stellar for tips on running your simulations.

4.1 DEVICE MODELS

For your convenience, the staff has created subcircuits to model the MOSFET with parameters given in Section 3.1. The subcircuits will calculate λ and device geometries (refer to Section 3.2) for you given the gate length (lg) and gate width (wg).

When instantiating a MOSFET, identify the MOSFET as a subcircuit in the component attribute editor by using the 'x' prefix, and type the subcircuit name (NFET or PFET) into the first Value line. Parameter values for lg and wg must be entered in the Value2 line.

You can save the following as mosfets.sub or download the .sub file directly from Stellar:

```
.subckt NFET D G S GND
.model NCH NMOS LEVEL=1 VTO=0.5 TOX=1.5E-8 U0=220
+ LAMBDA='(1.5u/lg)*1.0E-1' CJ=1.0E-4
+ CJSW=5.0E-10 PB=0.95 GAMMA=0.6
M1 D G S GND NCH l='lg' w='wg' ps='12u+wg'
+ pd='12u+wg' as='6u*wg' ad='6u*wg'
.ends NFET

.subckt PFET D G S VDD
.model PCH PMOS LEVEL=1 VTO=-0.5 TOX=1.5E-8 U0=110
+ LAMBDA='(1.5u/lg)*1.0E-1' CJ=3.0E-4
+ CJSW=3.5E-10 PB=0.9 GAMMA=0.6
M1 D G S VDD PCH l='lg' w='wg' ps='12u+wg'
+ pd='12u+wg' as='6u*wg' ad='6u*wg'
.ends PFET
```

You need to call the model file in your schematic with the following SPICE directive:

```
.inc mosfets.sub
```

5 Report

The deliverable for this project is a report detailing your design decisions and how you arrived at them. Please budget enough time to complete and revise your report, part of your grade will be based on the quality of your report. You must turn in a **HARD COPY** of your report, which should be **FIVE PAGES in length with 12 pt. font** (not including the cover page, schematics, waveforms, and netlists). The report should include:

- The summary cover page attached to this assignment.
- A detailed explanation of the approach you took to arrive at your final design for each stage of the circuit. This section is very important. What parameters did you consider the most constraining and what steps did you take to design for those parameters? How did you minimize power dissipation? **(~ 3 pages)**
- Summarize your hand calculations. Show equations used and describe their significance in your design. **(~ 1 page)**
- Explanation of any calculated values that did not match simulated values. **(~ 1 page)**
- A table showing for each required frequency your: achieved frequency; frequency error; t_{high} %; t_{low} %; the average power dissipation of the entire circuit; and the drain current, Vgs, and Vds of the NMOS underneath the switch that is mirroring the reference current (ie: referring to figure 8, when S1 is on the transistor of interest is M1).
- Screen captures of hierarchical schematics of every circuit you design, from top level down to transistor level. Be sure your final **device sizes are clearly labeled** at the transistor level. You may find it necessary to change the schematic background color for your schematics to be readable.
- **Screen captures of simulation output waveforms.** Include transient waveforms for the outputs of the ring oscillator. Using clearly labeled cursors, show the ring oscillator's frequency and high and low voltage times for all required frequencies. Include a transient waveform plot of I(VDD) over several periods. For readability, do not use the default black background color of the waveform window.
- Attach to your report a printed copy of your final top level netlist generated by LTspice. This is the .net file generated in the same folder where your top level schematic is saved, and also bears the same filename as your top level schematic. This netlist should include all final simulation commands used.
- **Upload the netlist file onto Stellar**. If you are working as partners, submit the netlist on both partner's Stellar accounts. The report cover sheet attached asks you for node names in your netlist that enables us to simulate your netlist easily. All values simulated using your uploaded netlist should match those presented in your report.

THIS PAGE INTENTIONALLY BLANK

Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science

6.012 Design Problem: Low-power Clock Generator
Due April 14, 2010, 4:00pm

REPORT COVER SHEET

Name: _____

Name: _____

Parameter Name	Specified	Calculated	Simulated
Total average power dissipated (mW)	$P_{avg} < 3.3mW$[1]	Worst Case frequency $f_? =$ $P_{avg} =$	Worst Case frequency $f_? =$ $P_{avg} =$
I_{REF} (µA)	●	$I_{REF} =$	$I_{REF} =$
Current-starved inverter sizing (µm)	●	$W_n =$ $L_n =$	$W_n =$ $L_n =$
Frequency of Oscillation for S_n (MHz)	$S_0 = 10 \pm 1\%$ $S_1 = 20 \pm 1\%$ $S_2 = 30 \pm 1\%$ $S_3 = 40 \pm 1\%$ $S_4 = 50 \pm 1\%$ $S_5 = 60 \pm 1\%$ $S_6 = 70 \pm 1\%$ $S_7 = 80 \pm 1\%$	Worst Case frequency Desired $f_? =$ Actual $f_? =$	Worst Case frequency Desired $f_? =$ Actual $f_? =$
T_{high} and T_{low} (ns)	$t_{low/high0} > 20$ $t_{low/high\ 1} > 10$ $t_{low/high\ 2} > 6.6$ $t_{low/high\ 3} > 5$ $t_{low/high\ 4} > 4$ $t_{low/high\ 5} > 3.3$ $t_{low/high\ 6} > 2.8$ $t_{low/high\ 7} > 2.5$	Worst Case frequency $t_{low} =$ $t_{high} =$ $f_? =$	Worst Case frequency $t_{low} =$ $t_{high} =$ @ $f_? =$

[1] Part of the final grade will be based on how much you surpass the 10mW specification.