

Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science

6.012
Microelectronic Devices and Circuits
Spring 2009

Design Project

This project may be worked on 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 The Optical Converter

Information can be encoded and transmitted optically. Such a data transmission system requires a decoding receiver that converts the information contained in the intensity of transmitted photons into useful electrical signals understandable by digital processors. You will be designing a converter from photons to a pulse width modulated signal to be used as the first stage of a four-state optical receiver.

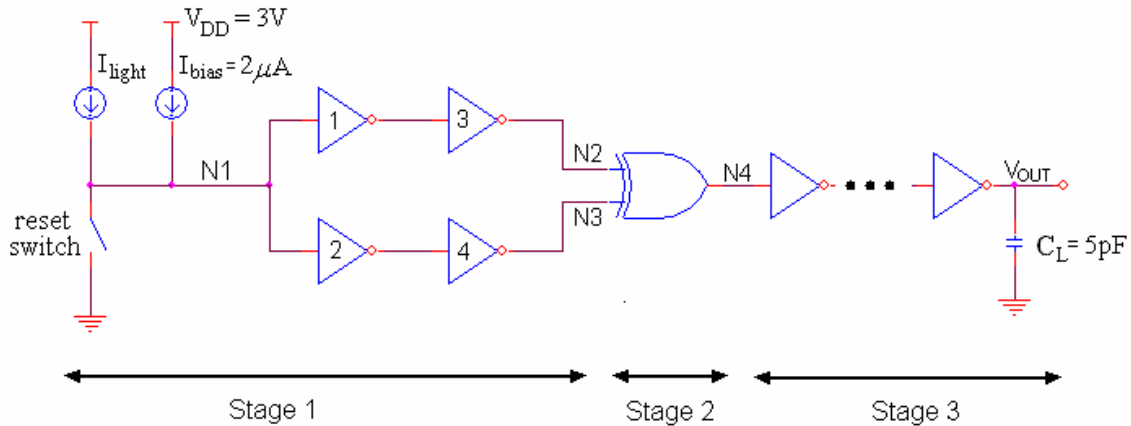
Our converter will generate pulses of four different widths in time, corresponding to the four light intensity levels the receiver detects. We will design for a 200MHz receiver system clock speed. The four pulses must be wide enough to allow for several clock cycles to complete in one pulse, and sufficiently distinct in width from each other for each of the four pulses to be differentiated by the next stage of the receiver, which counts the number of clock cycles within a pulse.

The converter should be designed in three stages:

1. **Stage 1** detects the incoming photon intensity and converts it into a current signal. A photodiode is a device that passes a current that is directly proportional to the detected light intensity. This current is used in the generation of two voltage steps, where the time delay between the two steps is a function of the photodiode current.
2. **Stage 2** converts the time delayed voltage steps to a single pulse, whose pulse width is the time delay between the two outputs from Stage 1.
3. **Stage 3** is an output buffer that is designed to drive a 5pF load capacitance.

As electronics trend ever smaller and more portable, die area has become an increasingly important design parameter for circuit designers. In this project, we will focus our design efforts on minimizing total gate area while preserving desired functionality.

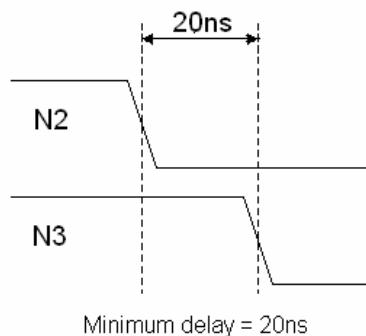
2 Implementation



2.1 STAGE 1

We will model the photodiode in our circuit as an ideal four level current source that sources $I_{\text{light}} = 0\mu\text{A}$, $1\mu\text{A}$, $2\mu\text{A}$, and $3\mu\text{A}$, with $0\mu\text{A}$ corresponding to a completely dark input. A bias current source of $I_{\text{bias}} = 2\mu\text{A}$ is placed in parallel with the photodiode. The two current sources sum and drive the capacitance seen at node N1, which will be the sum of gate capacitances of inverters 1 and 2. An ideal voltage controlled switch connects N1 to ground. When the switch is closed, node N1 sees a direct path to ground. When the switch is open, the current sources charge the capacitances at N1, producing a voltage ramp at that node. The difference in V_M between inverters 1 and 2 produce the time delay between the steps at N2 and N3. Inverters 3 and 4 clean up the output signals from inverters 1 and 2.

Your task is to design the four inverters 1–4. To maintain compatibility with the 200MHz system clock, the time delay between voltage steps generated in this stage should be no less than 20ns. The minimum difference between time delays generated by different light intensities should also be 20ns. That is, if at some I_{light} the generated delay is 20ns, then at the next I_{light} level, the time delay should be no less than 40ns, and so on.



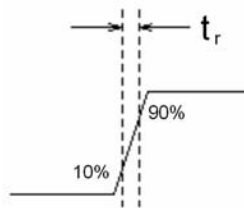
High gain inverters 1 and 2 should be sized to achieve the appropriate V_M and gate capacitance, chosen to achieve the delay specifications while trying to minimize the total gate area of Stage 1. Your project will be evaluated based on how low gate area you achieve, which should not exceed $400\mu\text{m}^2$.

2.2 STAGE 2

Stage 2 takes as inputs the time delayed steps from Stage 1, and outputs a single pulse whose width is the time delay between the two Stage 1 steps. A device that can accomplish this is an XOR gate. Show using a truth table why this is true. Your task is to design an XOR gate. The XOR gate should be composed of AND, OR gates, or their complements as well as inverters if needed.

2.3 STAGE 3

The output buffer is a chain of inverters designed to drive 5pF . Your task is to determine how many and what size inverters are necessary to drive 5pF and generate a positive going pulse at V_{OUT} . The inverters should be individually sized to drive the increasing load capacitance seen at the output of each subsequent inverter. Rise and fall times at V_{OUT} should each not exceed 3ns , measured from the 10% to 90% points.



The total gate area for stage 2 and 3 should not exceed $600\mu\text{m}^2$, and as with Stage 1, you should try to achieve as low gate area as you can design.

2.4 SUMMARY OF SPECIFICATIONS

Supply	$V_{\text{DD}} = 3\text{V}$		
Stage 1	Maximum gate area	400	μm^2
	Minimum pulse width	20	ns
	Minimum output step size	20	ns
Stage 2&3	Maximum total gate area	600	μm^2
Stage 3	Maximum t_r, t_f	3	ns
	Load capacitance	5	pF

3 Calculations

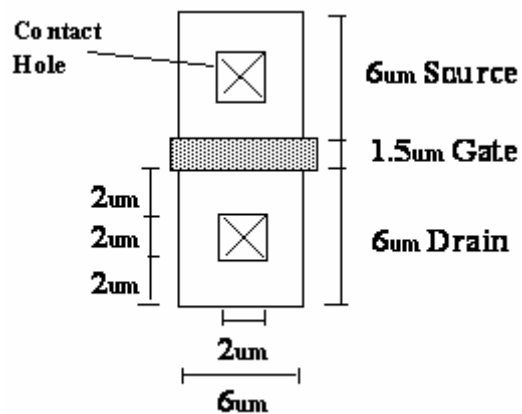
Before simulating, you should make sufficient calculations to choose appropriate transistor sizes to meet specifications. You should get a good feel for the tradeoffs you will need to make and which performance parameters are affected by your sizing decisions. Be prepared to explain your calculations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

3.1 MOSFET PARAMETERS

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

1. Threshold voltages are fixed at $V_{Tn} = 0.5$ [V], $V_{Tp} = -0.5$ [V].
2. Oxide thicknesses are fixed at $t_{OX} = 15$ [nm].
3. Mobility: $\mu_n = 220$ [cm^2/Vs] for the NMOS, $\mu_p = 110$ [cm^2/Vs] for the PMOS.
4. Junction capacitance (CJ) = 1×10^{-4} [F/m^2] NMOS, 3×10^{-4} [F/m^2] PMOS.
5. Sidewall junction capacitance (CJSW) = 5×10^{-10} [F/m] (NMOS), 3.5×10^{-10} [F/m] (PMOS)
6. Built in potential ϕ_B (PB) = 0.95 [V] (NMOS), 0.9 [V] (PMOS)
7. The parameter $LAMBDA = 67 \text{ mV}^{-1}$ is known for $1.5 \mu\text{m}$ long devices. When using lengths other than $1.5 \mu\text{m}$, it is necessary to adjust the value of λ . For a MOSFET of length L , the new λ value is $(1.5/L) * 67 \text{ mV}^{-1}$. This is because λ is proportional to $(1/L)$. For your convenience, we have created sub-circuits which are simply a MOSFET but have the λ value and geometries computed for you.
8. The backgate is tied to GND for N-MOSFETS, and V_{DD} for P-MOSFETS.

3.2 MINIMUM TRANSISTOR SIZES



Minimum Size Transistor: Top View

A minimum size transistor diagram is shown above. The minimum gate length is $1.5 \mu\text{m}$ and minimum gate width is $6 \mu\text{m}$. 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, to ensure that the contact hole is over only the diffused area, and not the surrounding bulk silicon. Any of these dimensions may be bigger, but it is not advantageous to make the source or

drain bigger, since it only adds to the parasitic capacitance. We ask you to minimize gate area, so there is no need to calculate the complete area of your circuit, however knowing the geometry of the transistor is important in calculating parasitic device capacitances.

3.3 CAPACITANCE 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/m²] 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 description 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 description 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 μm² = 36x10⁻¹²m². The area of the gate region is 1.5 x 6 = 9x10⁻¹²m². The total area of the transistor is thus 36 + 36 + 9 = 81 μm².

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, and on the ramp generated at node N1. To simplify your hand calculations, assume that the capacitors are linear with value equal to the zero voltage capacitance. This will produce a conservative design, as the average capacitance is less than the zero voltage capacitance.

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.

4.1 LTSPICE

The 6.012 staff strongly encourages you to use LTSpice as we expect it to save you significant netlist debugging time.

Linear Technology Corporation provides a widely used, free graphical SPICE simulator known as LTSpice. Download and install LTSpice IV from Stellar or the Linear Technology website, <http://www.linear.com>. LTSpice is only available for Windows. If you do not own a PC, you may consider downloading the free 30 day trial of Crossover, a windows compatibility layer, at <http://www.codeweavers.com/products> for Mac (Intel processor) or Linux. Alternatively, there are Windows clusters in:

37-312 (faster, newer machines)
Student Center

If you install LTSpice in a Windows cluster, make sure to install it in your winathena directory, not the computer's local directory.

4.1.1 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). You should tie backgates to GND for NFETs, and V_{DD} for PFETs in your schematic.

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 into the first Value line. Parameter values for lg and wg must be entered in the Value2 line.

We also provide a model for the ideal voltage controlled switch. You will not need to change any parameters in the default component attribute editor, however you do need to provide the voltage that controls the switch. This voltage should be a square wave voltage source swinging 0-3V that rises and falls 1 V/ns. You should choose the period of your square wave to accommodate your unique design parameters.

Save the following as a .sub file, or download the model file directly from Stellar, and call it in your schematic with the SPICE directive:

```
.inc [filename].sub
```

```
.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

.model SW Vswitch RON=100MEG ROFF=1 VON=3 VOFF=0

```

4.1.2 Helpful LTSpice Hints:

LTSpice (and most commonly used simulators) is generally much faster and friendlier to use if you get to know keyboard shortcuts well. Visit Tools → Control Panel → Drafting Options → Hot Keys to either get to know default keyboard shortcuts or set up a set of shortcuts that are intuitive to you.

To exit a placement mode hit ESC.

To view your netlist, visit View → SPICE netlist. This will be useful should you need to debug your circuit.

Any SPICE directive you place on the schematic will be incorporated into the netlist.

You can create a symbol of your schematic to instantiate in other schematics. Visit Hierarchy → Open this Sheet's Symbol. This allows you to create hierarchical schematics that are much easier to read and manipulate. For example, you can create an inverter schematic and generate a symbol for it. In the top level schematic you can instantiate that inverter several times such that when you manipulate the inverter schematic, it will change the schematic for all inverters you instantiate at the top level.

The most useful simulation commands you will need are .trans and .step.

LTSpice documentation is available in its help menu F1.

4.2 HSPICE

If you really cannot access a Windows machine, you can complete your simulations using HSPICE, a non-graphical SPICE compiler available on Athena, and plot your output files using Avanwaves.

Using the same device models as given for LTSpice but with the .sp extension, create spice netlists using extension .sp. A minimum size instantiation of the MOSFET models is shown below:

```
X1 OUT IN VDD VDD PFET lg=1.5u wg=6u
X2 OUT IN GND GND NFET lg=1.5u wg=6u
```

To run HSPICE on a netlist 'name.sp' and plot the results, use the following commands:

```
% add hspice
% hspice name.sp > name.out &
% awaves name &
```

Documentation for HSPICE and Awaves can be found at:

/mit/hspice/Z-2007.03-SP1/hspice/docs/hspice_sa.pdf
</mit/hspice/Z-2007.03-SP1/hspice/docs/avanwaves.pdf>

Note: Awaves does not run on Linux. To run plot your waveforms while on a Linux box:

```
% ssh -Y x.dialup.mit.edu
% add hspice
% awaves name &
```


5 Report

The deliverable for this project is a report detailing your design decisions and how you arrived at them. You must turn in a **HARD COPY** of your report, which should include:

1. The summary cover page attached to this assignment.
2. 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 your approach differ between designing the various stages?
3. Summarize your hand calculations. Show equations used and describe their significance in your design. This must include calculations for your node N1 capacitance, node N1 ramp rate, V_M and gain for inverters 1 and 2, load capacitances seen by each of your output stage inverters, t_r and t_f calculations, and gate area calculations.
4. Explanation of any calculated values that did not match simulated values.
5. Screen captures of hierarchical schematics of every circuit you design, 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.
6. Screen captures of simulation output waveforms. Include transient waveforms for node N1, N2, N3, N4, and V_{OUT} at each I_{light} . You are welcome to superimpose plots from all I_{light} , however make it obvious which graphs result from which current steps. Again it might be necessary to change the background color of the waveform window to produce readable plots.
7. SPICE netlist:
 - 7.1 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.
 - 7.2 **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 measured from simulating your uploaded netlist should match those presented in your report.

Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science

6.012 Design Problem: The Optical Converter

REPORT COVER SHEET

Name: _____

Name: _____

Please list the node names you used in your SPICE netlist that correspond to:

N1 _____ N2 _____ N3 _____ N4 _____ V_{OUT} _____

Parameter Name	Specified	Calculated	Simulated
N1 Capacitance (pF)	■		
V _M – Inverter 1 Inverter 2	■	1: 2:	1: 2:
Gain – Inverter 1 Inverter 2	■	1: 2:	1: 2:
Pulse width (ns) I _{light} = 0μA I _{light} = 1μA I _{light} = 2μA I _{light} = 3μA	>20ns per pulse > 20ns difference for different I _{light}	0μA: 1μA: 2μA: 3μA:	0μA: 1μA: 2μA: 3μA:
t _r at V _{OUT} (ns) t _f at V _{OUT} (ns)	t _r : < 3ns t _f : < 3ns	t _r : t _f :	t _r : t _f :
Gate Area (μm ²)	Stage 1: < 400μm ² Stage 2&3: < 600 μm ²	Stage 1: Stage 2&3:	Stage 1: Stage 2&3:
W/L - Inverter 1 Inverter 2 Inverter 3 Inverter 4	■		
W/L of All Output Stage Inverters	■		

MIT OpenCourseWare
<http://ocw.mit.edu>

6.012 Microelectronic Devices and Circuits
Spring 2009

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.