

Physics 364, Fall 2014, Lab #19 **Name:** _____
(*Digital Logic Introduction*)

Wednesday, November 5 (section 401); Thursday, November 6 (section 402)

Course materials and schedule are at positron.hep.upenn.edu/p364

Today, we finally begin the digital segment of the course! In the first part of today's lab, borrowed from Penn's ESE111, you will use mechanical switches and LEDs to implement a simple form of logic gate. In Part 2, you will observe the behavior of a commercially-made CMOS Integrated Circuit that performs the NAND function; and in Part 3 you will use several of these NAND gates to implement AND and OR functions. Finally, in Part 4, you will build up your own inverter ("NOT" gate) and NAND gate from individual MOSFETs.

Next week and the week after, the focus will be on programming tiny "Arduino" computers to carry out various tasks. Then in the final two weeks of the course, we'll return to digital logic to see how, in principle, a computer can be built up from logic gates.

This weekend's reading is not finished yet (sorry!), but will be posted on the course web page (and linked from Canvas) by Friday. It will discuss both digital logic and Arduino programming. Please stay tuned!

Part 1

Start Time: _____

ESE111 Lab 2, §1–2

(time estimate: 45 minutes)

Go through sections 1 and 2 of the ESE111 Lab 2 (Intro to Digital Logic), included on the following pages. You don't need to write anything down as you go through this first part. The main idea here is to demonstrate to yourself in a very concrete way, using mechanical switches and LEDs, how logic gates work in principle. It then becomes easy to see how MOSFET switches can accomplish the same task.

There is one very important way in which these mechanical-switch-based logic gates — as implemented here in Part 1 — are not realistic: they signal their HIGH/LOW status by the presence/absence of a current through an LED. A more realistic logic gate would supply a low-impedance "+5 V" voltage output for the HIGH state and a low-impedance "0 V" voltage output for the LOW state.

Lab 2 – Intro to Digital Logic and Transistors

Introduction:

Up until now, everything that you have done has been in the analog realm. By changing the resistance of your simple LED circuit, you have been able to sweep through a continuous range of light intensity. However, in the world of digital electronics, a signal can have only one of two values: ON (HIGH, 1) or OFF (LOW, 0). In this lab, you will become familiar with basic digital logic, and you will implement basic logic functions using pushbutton switches and integrated circuit (IC) chips. You will also learn how to use a transistor as a switch, and in the process, you will learn how to use the function generator to supply a voltage signal and the oscilloscope to look at voltage signals.

Goals:

- Learn how to implement logic gates with switches
- Learn how to read pinout diagrams
- Understand the operation and importance of the half-adder circuit
- Learn how to use the transistor as a switch
- Learn how to use the function generator and oscilloscope

Procedure:

1. Turn an LED on/off using a pushbutton

- Obtain a four-terminal pushbutton. Figure 1 shows a diagram of a pushbutton and its internal connections. When the button is not being pressed, terminals 1 and 2 are electrically connected to each other (shown in red), and terminals 3 and 4 are electrically connected to each other (also shown in red). When the button is pressed, all four terminals become electrically connected to each other (shown in green).

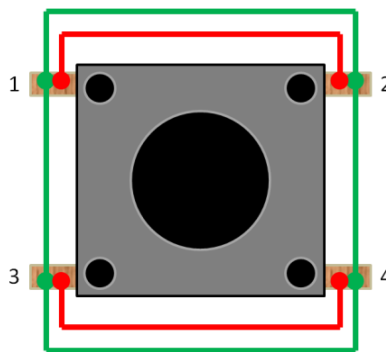


Figure 1: Pushbutton connections; red = not pressed, green = pressed

- Turn the output of the power supply off and build the circuit represented in Figure 2. Use Figure 3 as a reference. Note that the four-terminal switch that we are using has been abstracted to a two-terminal switch for simplicity.

Created by Nick Howarth (EE '13), Noam Eisen (EE '14), and Sam Wolfson (EE '13)
Last updated: September 12, 2012

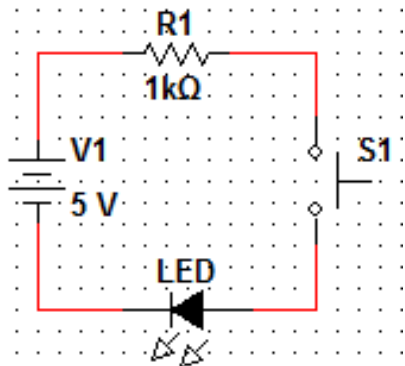


Figure 2: Simple circuit schematic using switch to turn on/off LED

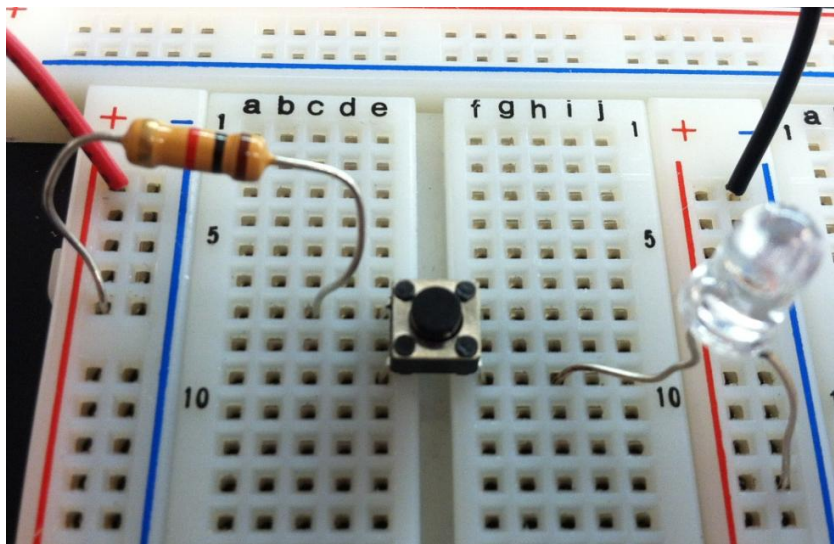


Figure 3: Simple circuit using pushbutton to turn on/off LED

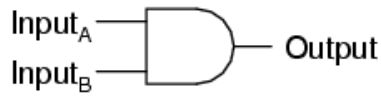
- Turn the power supply on. The LED should now only turn on when you press the button. In the digital world, the output is either a 1 (the LED is on) or a 0 (the LED is off).

2. Build logic gates using pushbuttons

Logic gates are the basic building blocks of all digital electronics. Logic gates have some number of binary inputs, usually two, and one output. You will now build two basic logic gates, the AND gate and the OR gate, using pushbuttons.

The symbol and truth table for the two-input AND gate are shown in Figure 4.

2-input AND gate



A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1

Figure 4: Symbol and truth table for two-input AND gate

In a truth table, a '0' represents a low voltage (0V) and a '1' represents a high voltage (in our case, 5V). The output of the AND gate is only '1' when both of the inputs are '1'; otherwise, the output is '0.' This AND gate can easily be constructed by placing two switches in **series**.

- Turn the output of the power supply off and build the circuit represented in Figure 5. Use Figure 6 as a reference.

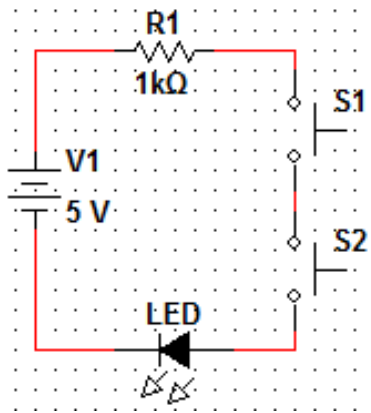


Figure 5: Schematic of AND gate implemented with pushbuttons

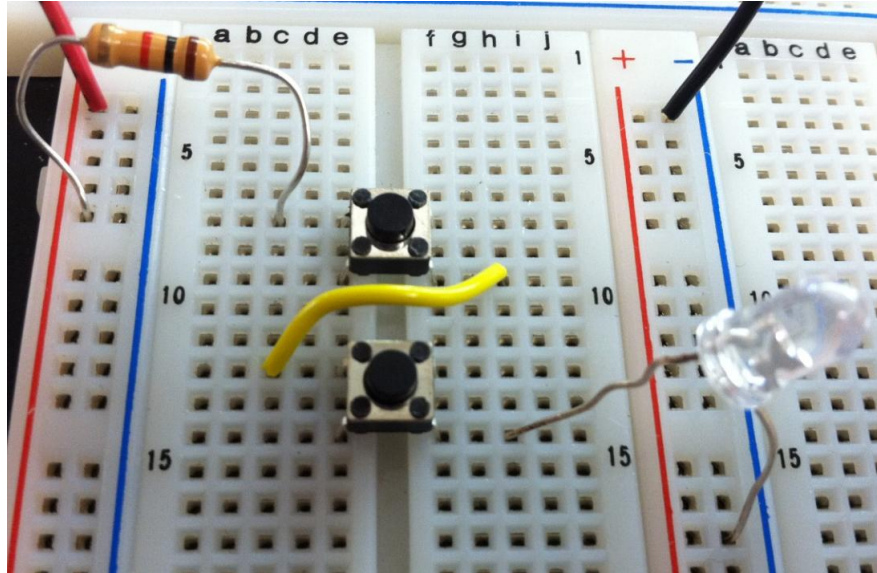
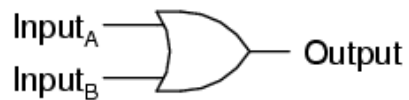


Figure 6: Implementation of AND gate with pushbuttons

- Turn the power supply on. The LED should now only turn on when **both** buttons are pressed. Since the pushbuttons are placed in **series**, current will only be able to flow from 5V to ground through the LED if both buttons are pressed; otherwise, the circuit is open and current does not flow, so the LED is off.

The symbol and truth table for the two-input OR gate are shown in Figure 7. The output of the OR gate is '1' when *either* of the inputs is '1'; the output is '0' if *neither* input is '1.' As you might have guess, the OR gate can be constructed by placing two switches in **parallel**.

2-input OR gate



A	B	Output
0	0	0
0	1	1
1	0	1
1	1	1

Figure 7: Symbol and truth table for two-input OR gate

- Turn the output of the power supply off and build the circuit represented in Figure 8. Use Figure 9 as a reference.

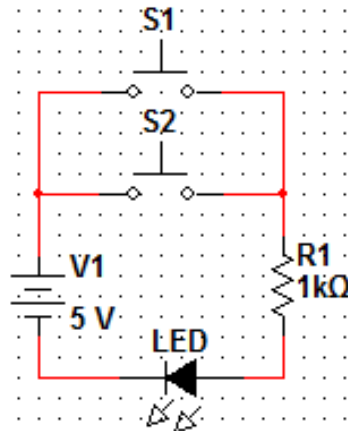


Figure 8: Schematic of OR gate implemented with pushbuttons

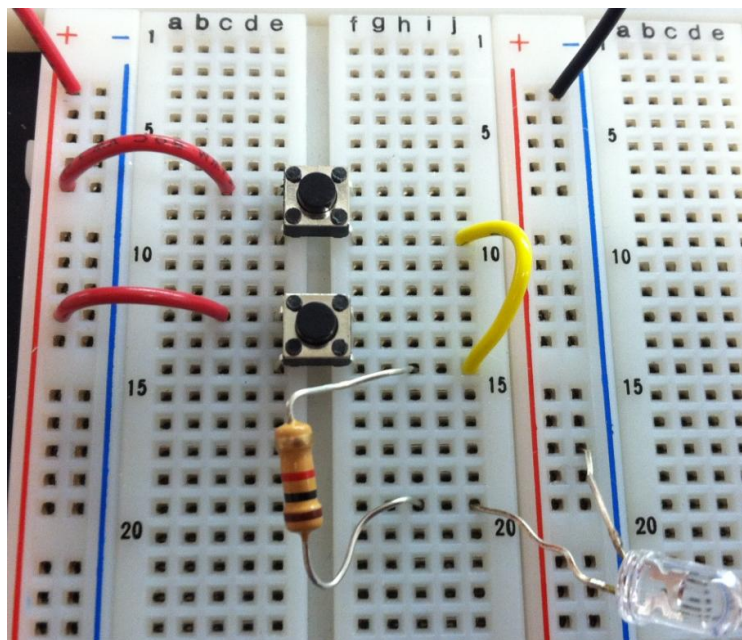


Figure 9: Implementation of OR gate with pushbuttons

- Turn the power supply on. The LED should now only turn on when *either* button is pressed. Since the pushbuttons are placed in **parallel**, current can travel through either or both pushbuttons if they are pressed, thus lighting the LED; otherwise, the circuit is open and current does not flow, so the LED is off.

Using combinations of logic gates, complex operations can be performed. In this lab, you used electromechanical switches to implement simple logic gates. However, in modern electronics, logic gates are implemented with **transistors**, which can be used as electrically activated switches. Anywhere from a few to thousands of transistors can be combined on a single chip to create **integrated circuits (ICs)**.

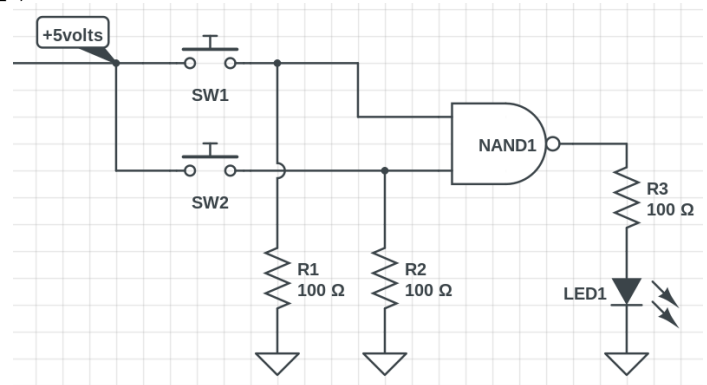
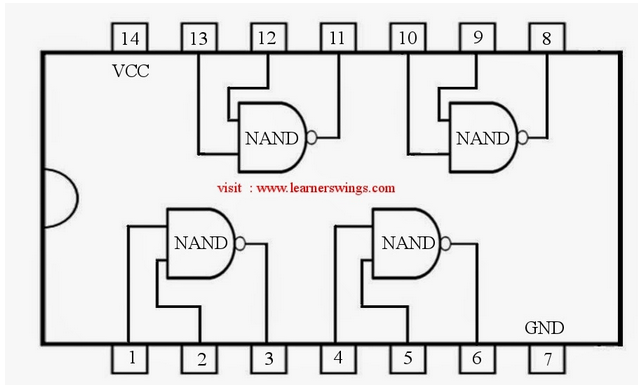
Part 2

CMOS NAND gate — integrated circuit

Start Time: _____

(time estimate: 45 minutes)

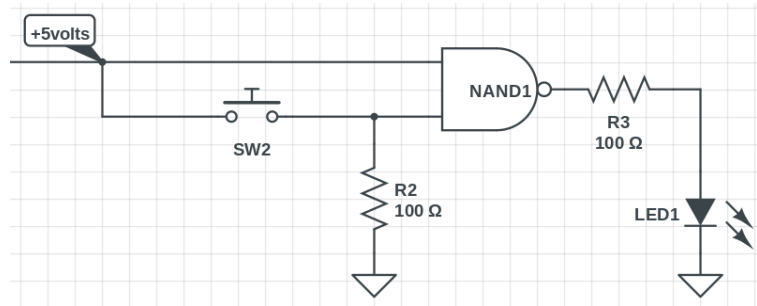
The figure below (left) shows the pin assignments of the 74HC00 CMOS NAND integrated circuit (IC). The 14-pin package contains four separate NAND gates. You need to connect pin 7 to ground and pin 14 to +5 V, to power the chip. CMOS logic gates behave in surprising and intermittent ways when you forget to make the power and ground connections, because “protection diodes” at each logic input can provide an alternative (but flaky) power path. Whenever you use these 14-pin logic chips, connect ground (pin 7) V_{CC} (pin 14) before you wire up anything else, to avoid later debugging! **A few pages of the 74HC00 data sheet are attached at the end of this write-up, for reference.**



2.1 The above-right figure shows a handy trick for wiring up two push-button switches such that each corresponding input sees +5 V when the button is pressed and sees 0 V when the button is not pressed.

What is the role of the two “pulldown” resistors R_1 and R_2 ? Why is resistor R_3 needed in series with the LED? Use the two switches to verify the NAND-gate truth table.

2.2 Next, keep one switch in place, but replace the other switch with a wire to +5 V. Convince yourself that this NAND gate is now basically working as a logical inverter (a NOT gate), and verify this with the push-button switch.

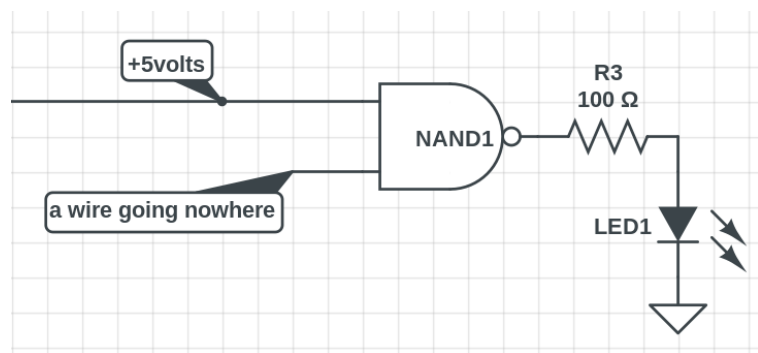


2.3 Now keep one NAND input connected to +5 V as before, but disconnect the one remaining pushbutton switch and replace it with a few inches of wire, one end of which goes nowhere (i.e. it is just dangling in the air).

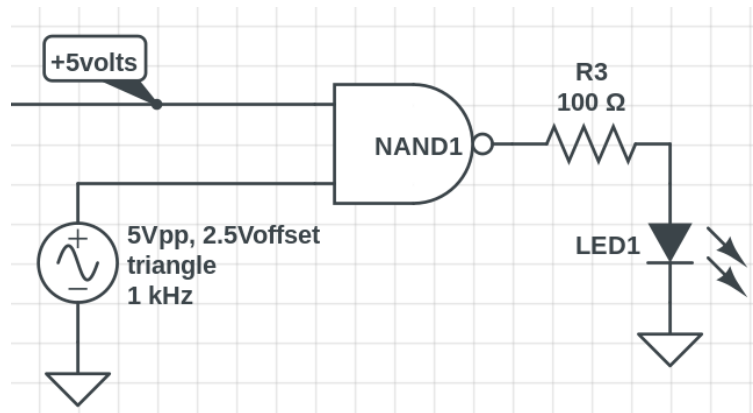
Touch the dangling wire with one hand while you touch your breadboard's ground connection with the other hand, then let go of both. What happens to the LED?

Then touch the dangling wire with one hand while you touch your breadboard's +5 V with the other hand, then let go of both. What happens to the LED now?

How does the fact that these are MOSFET-based logic gates help you to explain this?



2.4 Now leave the first NAND input connected to +5 V, but drive the second input (the one that was floating in the air a moment ago) from Channel 1 of the function generator, using 5 V_{pp} amplitude and +2.5 V DC offset, so that the waveform voltage spans the range from 0 V to +5 V. You might want to try a very low frequency first, like 10 Hz, so that you can see the LED blink. Then try a 1 kHz triangle wave and watch both the function-generator waveform and the NAND output with the oscilloscope. At what input voltages do the HIGH→LOW and LOW→HIGH transitions occur?



Now replace the triangle wave with a square wave, and try to estimate the time delay between the LOW→HIGH transition on the input and the corresponding HIGH→LOW transition on the output of the NAND gate. For comparison, the SN74HC00 data sheet specifies a maximum “propagation delay” $t_{pd} < 23$ ns, with a “typical” value around 9 ns.

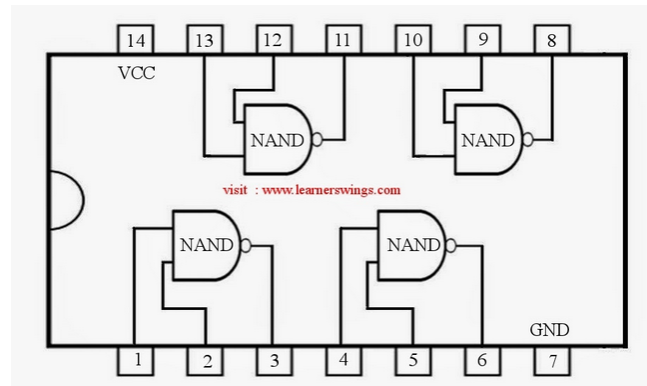
Part 3

Start Time: _____

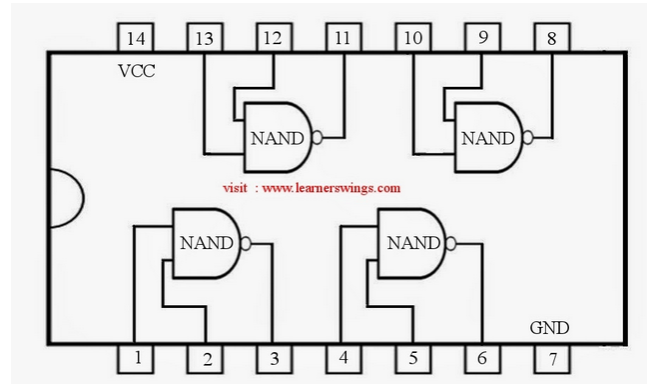
NAND gate applications

(time estimate: 30 minutes)

3.1 Use several NAND gates together (several logic gates from a single 74HC00 chip) to perform the AND function: light the LED if and only if both inputs are HIGH. Use the pushbutton switches from part 2.1 to provide your two test inputs. Draw your schematic below and then test your circuit.



3.2 Use several NAND gates together to perform the OR function: light the LED if either input is HIGH (or if both inputs are high). Use the pushbutton switches from part 2.1 to provide your two test inputs. Draw your schematic below and then test your circuit.



Part 4

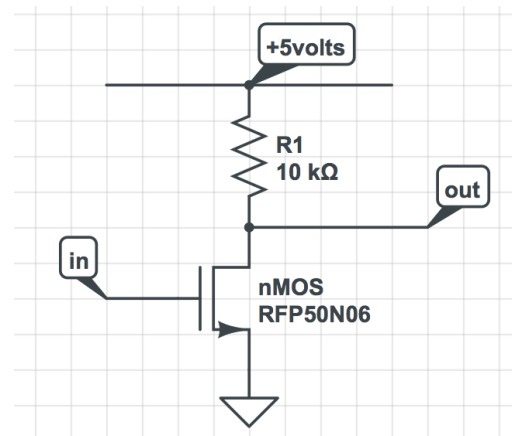
Start Time: _____

building logic gates from MOSFETs

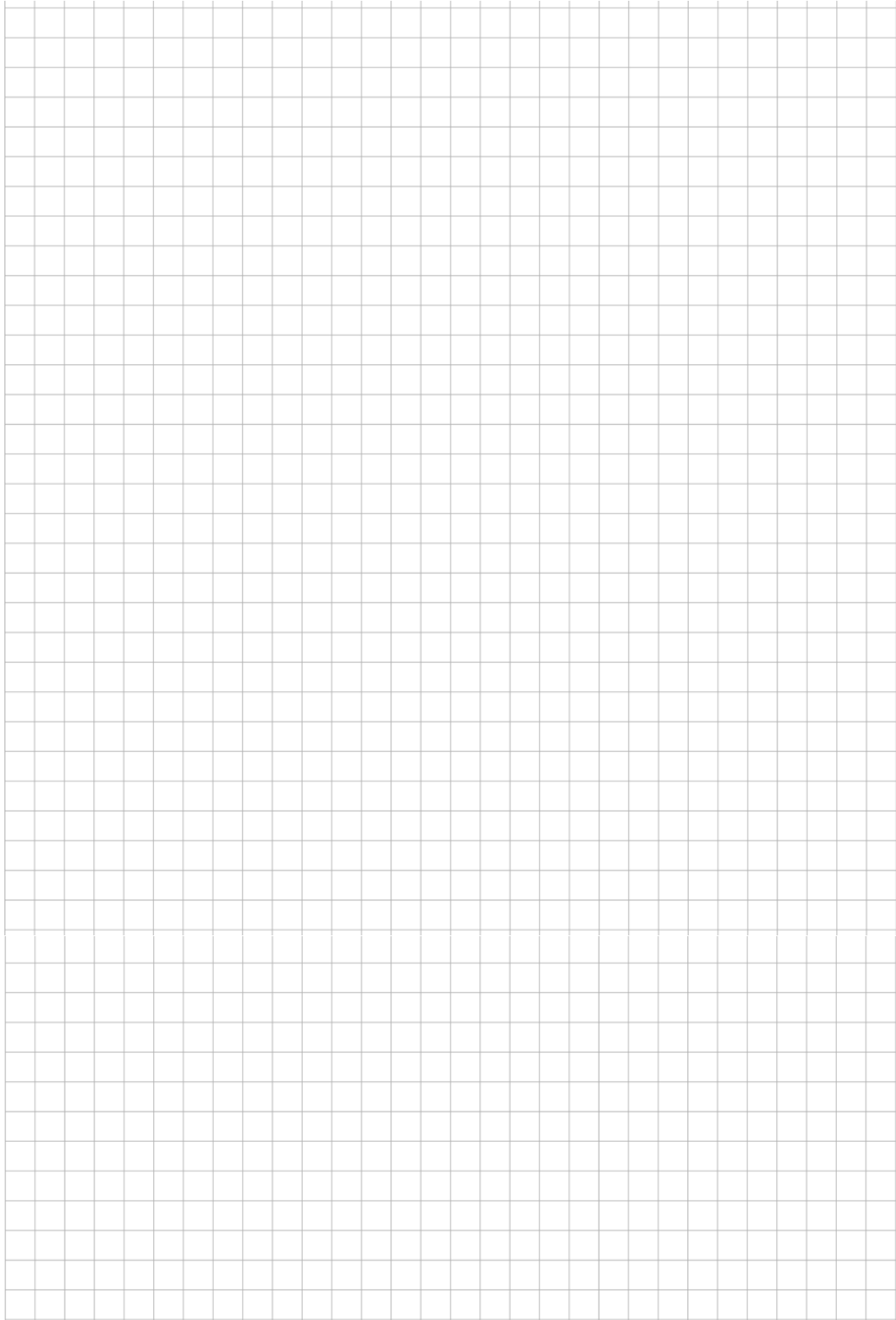
(time estimate: 30 minutes)

4.1 The circuit shown below is an “nMOS” logical inverter. It uses only an n -channel MOSFET, while the logic gates we studied in the notes used complementary nMOS/pMOS pairs. By studying this circuit, you’ll see the advantage of CMOS over nMOS.

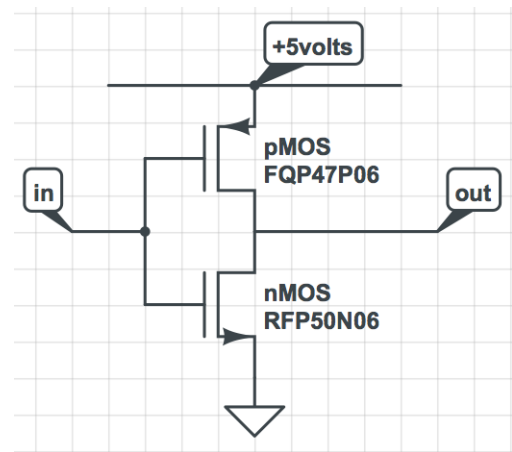
Build this circuit, using a single RFP50N06 n -channel MOSFET and a $10\text{ k}\Omega$ “pullup” resistor. Drive the input with a 1 kHz square wave (from CH1 of your FG) whose LOW value is 0 V and whose HIGH value is $+5\text{ V}$. Watch both $V_{\text{in}}(t)$ and $V_{\text{out}}(t)$ with the oscilloscope, and confirm that this circuit does indeed perform a logical inversion. (Notice that this circuit resembles a MOSFET analogue of the high-gain “grounded-emitter” version of the common-emitter amplifier — which is an inverting amplifier.)



Now increase the frequency until you start to see the inverter fail to do its job properly. Draw the waveforms (on the next page), both for the well-behaved low-frequency case and for the high-frequency case where the behavior is marginal. What do you think is happening?

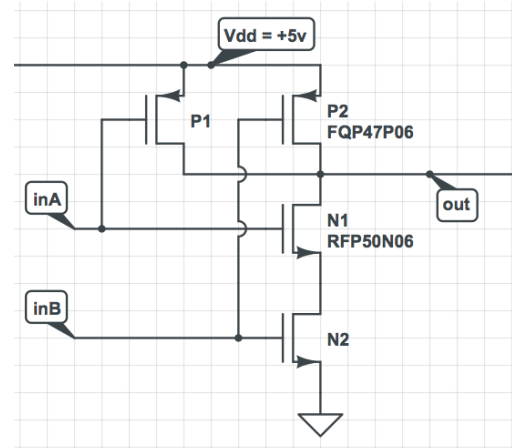


4.2 Now remove the resistor and add a *p*-channel MOSFET (FQP47P06), thereby forming a conventional CMOS inverter, as shown below. Confirm that this circuit indeed is a logical inverter, with a low-frequency (1 kHz) square wave, and then try it at the high frequency at which your nMOS inverter began to fail. How does the CMOS inverter compare? (By the way, this circuit resembles the CMOS push-pull, but here the pMOS transistor is on top, with the two drains connected together at the output, while the push-pull puts the pMOS transistor on the bottom, with the two sources connected together at the output.)



If all goes as planned, you should find that the CMOS inverter is much faster than the nMOS inverter, especially on the LOW→HIGH transition of the output. In the nMOS circuit, stray capacitance forms an RC low-pass filter with the 10 k Ω resistor, slowing down the output transitions.

4.3 Next, use two pMOS FETs (FQP47P06) and two nMOS FETs (RFP50N06) to build the CMOS NAND gate shown below. In whatever way you wish (pushbuttons, LEDs, oscilloscope, function generator), confirm that it indeed performs the NAND logic function. Briefly describe how you did your testing.



Puzzle through how this circuit actually achieves the NAND function, by noticing that the *n*-channel enhancement-mode MOSFETs turn ON when their gates are HIGH (and OFF when LOW), while the *p*-channel enhancement-mode MOSFETs turn ON when their gates are LOW (and OFF when HIGH).

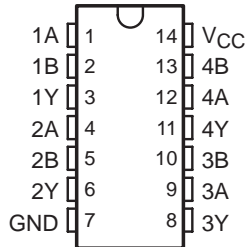
(blank page)

SN54HC00, SN74HC00 QUADRUPLE 2-INPUT POSITIVE-NAND GATES

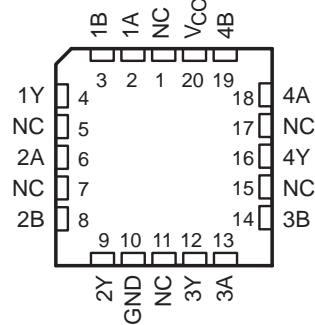
SCLS181E – DECEMBER 1982 – REVISED AUGUST 2003

- Wide Operating Voltage Range of 2 V to 6 V
- Outputs Can Drive Up To 10 LSTTL Loads
- Low Power Consumption, 20- μ A Max I_{CC}
- Typical $t_{pd} = 8$ ns
- ± 4 -mA Output Drive at 5 V
- Low Input Current of 1 μ A Max

SN54HC00 . . . J OR W PACKAGE
SN74HC00 . . . D, DB, N, NS, OR PW PACKAGE
(TOP VIEW)



SN54HC00 . . . FK PACKAGE
(TOP VIEW)



NC – No internal connection

description/ordering information

The 'HC00 devices contain four independent 2-input NAND gates. They perform the Boolean function $Y = A \cdot B$ or $Y = \bar{A} + \bar{B}$ in positive logic.

ORDERING INFORMATION

T_A	PACKAGE†		ORDERABLE PART NUMBER	TOP-SIDE MARKING
-40°C to 85°C	PDIP – N	Tube of 25	SN74HC00N	SN74HC00N
	SOIC – D	Tube of 50	SN74HC00D	HC00
		Reel of 2500	SN74HC00DR	
		Reel of 250	SN74HC00DT	
	SOP – NS	Reel of 2000	SN74HC00NSR	HC00
	SSOP – DB	Reel of 2000	SN74HC00DBR	HC00
	TSSOP – PW	Tube of 90	SN74HC00PW	HC00
Reel of 2000		SN74HC00PWR		
Reel of 250		SN74HC00PWT		
-55°C to 125°C	CDIP – J	Tube of 25	SNJ54HC00J	SNJ54HC00J
	CFP – W	Tube of 150	SNJ54HC00W	SNJ54HC00W
	LCCC – FK	Tube of 55	SNJ54HC00FK	SNJ54HC00FK

† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at www.ti.com/sc/package.



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

 **TEXAS
INSTRUMENTS**

POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

Copyright © 2003, Texas Instruments Incorporated
On products compliant to MIL-PRF-38535, all parameters are tested unless otherwise noted. On all other products, production processing does not necessarily include testing of all parameters.

SN54HC00, SN74HC00 QUADRUPLE 2-INPUT POSITIVE-NAND GATES

SCLS181E – DECEMBER 1982 – REVISED AUGUST 2003

FUNCTION TABLE
(each gate)

INPUTS		OUTPUT
A	B	Y
H	H	L
L	X	H
X	L	H

logic diagram (positive logic)



absolute maximum ratings over operating free-air temperature range (unless otherwise noted)†

Supply voltage range, V_{CC}	-0.5 V to 7 V
Input clamp current, I_{IK} ($V_I < 0$ or $V_I > V_{CC}$) (see Note 1)	± 20 mA
Output clamp current, I_{OK} ($V_O < 0$ or $V_O > V_{CC}$) (see Note 1)	± 20 mA
Continuous output current, I_O ($V_O = 0$ to V_{CC})	± 25 mA
Continuous current through V_{CC} or GND	± 50 mA
Package thermal impedance, θ_{JA} (see Note 2):	
D package	86°C/W
DB package	96°C/W
N package	80°C/W
NS package	76°C/W
PW package	113°C/W
Storage temperature range, T_{stg}	-65°C to 150°C

† Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTES: 1. The input and output voltage ratings may be exceeded if the input and output current ratings are observed.
2. The package thermal impedance is calculated in accordance with JESD 51-7.

recommended operating conditions (see Note 3)

		SN54HC00			SN74HC00			UNIT
		MIN	NOM	MAX	MIN	NOM	MAX	
V_{CC}	Supply voltage	2	5	6	2	5	6	V
V_{IH}	High-level input voltage	$V_{CC} = 2$ V		1.5	1.5		V	
		$V_{CC} = 4.5$ V		3.15	3.15			
		$V_{CC} = 6$ V		4.2	4.2			
V_{IL}	Low-level input voltage	$V_{CC} = 2$ V			0.5		V	
		$V_{CC} = 4.5$ V			1.35			
		$V_{CC} = 6$ V			1.8			
V_I	Input voltage	0		V_{CC}	0		V_{CC}	V
V_O	Output voltage	0		V_{CC}	0		V_{CC}	V
$\Delta t/\Delta v$	Input transition rise/fall time	$V_{CC} = 2$ V			1000		ns	
		$V_{CC} = 4.5$ V			500			
		$V_{CC} = 6$ V			400			
T_A	Operating free-air temperature	-55		125	-40		85	°C

NOTE 3: All unused inputs of the device must be held at V_{CC} or GND to ensure proper device operation. Refer to the TI application report, *Implications of Slow or Floating CMOS Inputs*, literature number SCBA004.



POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

SN54HC00, SN74HC00 QUADRUPLE 2-INPUT POSITIVE-NAND GATES

SCLS181E – DECEMBER 1982 – REVISED AUGUST 2003

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER	TEST CONDITIONS		V _{CC}	T _A = 25°C			SN54HC00		SN74HC00		UNIT
				MIN	TYP	MAX	MIN	MAX	MIN	MAX	
V _{OH}	V _I = V _{IH} or V _{IL}	I _{OH} = -20 μA	2 V	1.9	1.998		1.9		1.9	V	
			4.5 V	4.4	4.499		4.4		4.4		
			6 V	5.9	5.999		5.9		5.9		
		I _{OH} = -4 mA	4.5 V	3.98	4.3		3.7		3.84		
		I _{OH} = -5.2 mA	6 V	5.48	5.8		5.2		5.34		
V _{OL}	V _I = V _{IH} or V _{IL}	I _{OL} = 20 μA	2 V		0.002	0.1		0.1		0.1	V
			4.5 V		0.001	0.1		0.1		0.1	
			6 V		0.001	0.1		0.1		0.1	
		I _{OL} = 4 mA	4.5 V		0.17	0.26		0.4		0.33	
		I _{OL} = 5.2 mA	6 V		0.15	0.26		0.4		0.33	
I _I	V _I = V _{CC} or 0		6 V		±0.1	±100		±1000		±1000	nA
I _{CC}	V _I = V _{CC} or 0, I _O = 0		6 V			2		40		20	μA
C _i			2 V to 6 V		3	10		10		10	pF

switching characteristics over recommended operating free-air temperature range, C_L = 50 pF (unless otherwise noted) (see Figure 1)

PARAMETER	FROM (INPUT)	TO (OUTPUT)	V _{CC}	T _A = 25°C			SN54HC00		SN74HC00		UNIT
				MIN	TYP	MAX	MIN	MAX	MIN	MAX	
t _{pd}	A or B	Y	2 V		45	90		135		115	ns
			4.5 V		9	18		27		23	
			6 V		8	15		23		20	
t _t		Y	2 V		38	75		110		95	ns
			4.5 V		8	15		22		19	
			6 V		6	13		19		16	

operating characteristics, T_A = 25°C

PARAMETER	TEST CONDITIONS	TYP	UNIT
C _{pd} Power dissipation capacitance per gate	No load	20	pF



POST OFFICE BOX 655303 • DALLAS, TEXAS 75265