PURDUE UNIVERSITY DEPARTMENT OF PHYSICS

Physics 53600

Electronics Techniques for Research

Now in PowerPoint!

Spring 2020 Semester

Prof. Matthew Jones

ANNOUNCEMENT

- Obvious changes to the course:
  - No in-person lectures: you'll have to read the lecture notes yourself
  - No more labs: don't worry about it – your grade will be based on work done so far
  - Remaining assignments will try to cover topics that would have been explored in the lab
  - Second mid-term: simplest to cancel it
  - Final exam: I'm not sure what to do about this yet, but I'll figure something out.
- Changes to grading scheme:
  - Old scheme: Assignments (30%) exams (40%) lab (30%)
  - New scheme: Assignments (50%) exams (25%) lab (25%)

ANNOUNCEMENT

- Because there won't be any in-person lectures, you will have to read the lecture notes yourself.
- To demonstrate that you have read them, you will be required to answer *one or two simple questions* before the next lecture is posted.
- The question will be somewhere (like maybe at the end?) and you just have to e-mail me the answer

  mjones@physics.purdue.edu
- To make this easy, please make your subject look like this:

  "PHYS53600 Lecture xx questions Your Name"
- These will be part of your assignment grade, maybe contributing 10% of your total grade.

ANNOUNCEMENT

- Remember that you have Assignment #4 due on Thursday, March 26th.
- Please scan, or somehow generate a PDF file of your solutions and e-mail them to the grader

lim185@purdue.edu

- It might be useful to use a standard subject:

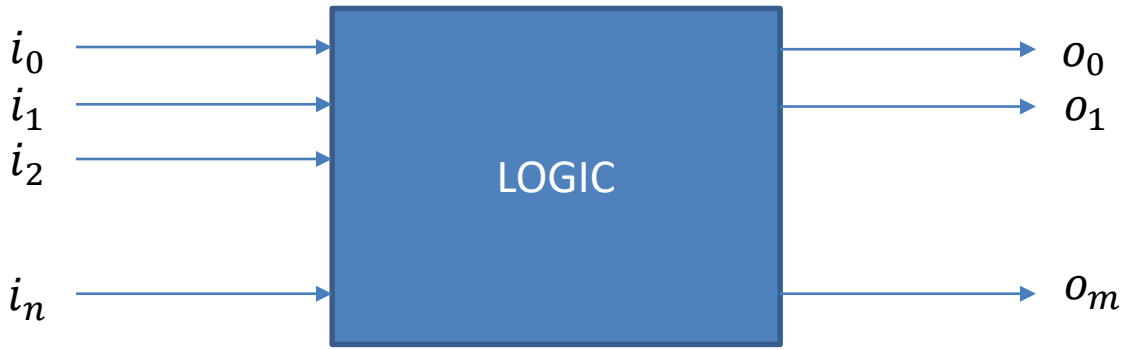
"PHYS53600 Assignment #4 Your Name"

LECTURE 18 QUESTION #1

Briefly summarize some of the similarities and differences of combinatorial and sequential logic

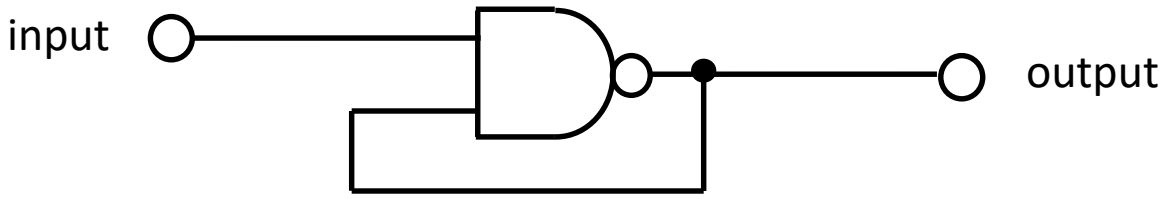
Combinatorial Logic

- So far we have considered combinatorial logic circuits. These could (in principle) be constructed from elementary logic gates (inverters, AND, OR, NAND, NOR, etc…)
- Essentially, these simply evaluate Boolean algebraic expressions:



Combinatorial Logic

- A characteristic feature of combinatorial logic is that the outputs depend ONLY on the inputs.
- This property is easy to break…
- Here is a simple example where this is not the case:



- Question: what is the output, as a function of the input?

Sequential Logic

- We can't analyze this example using Boolean algebra because there is no solution:

$$o = \overline{o \cdot i}$$
$$= \bar{o} + \bar{i}$$

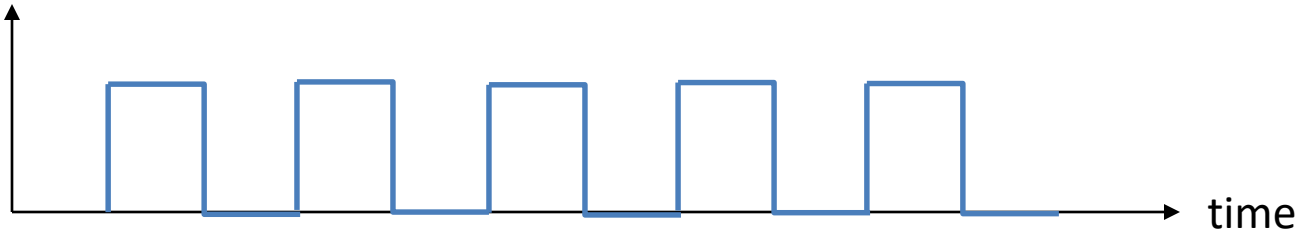
- Consider two cases:
  - If the input is 0, then the output is always 1
  - If the input is 1 then then the algebra simplifies to

$$o = \bar{o}$$

  - This is a contradiction – the output is undefined given the current assumptions.

Sequential Logic

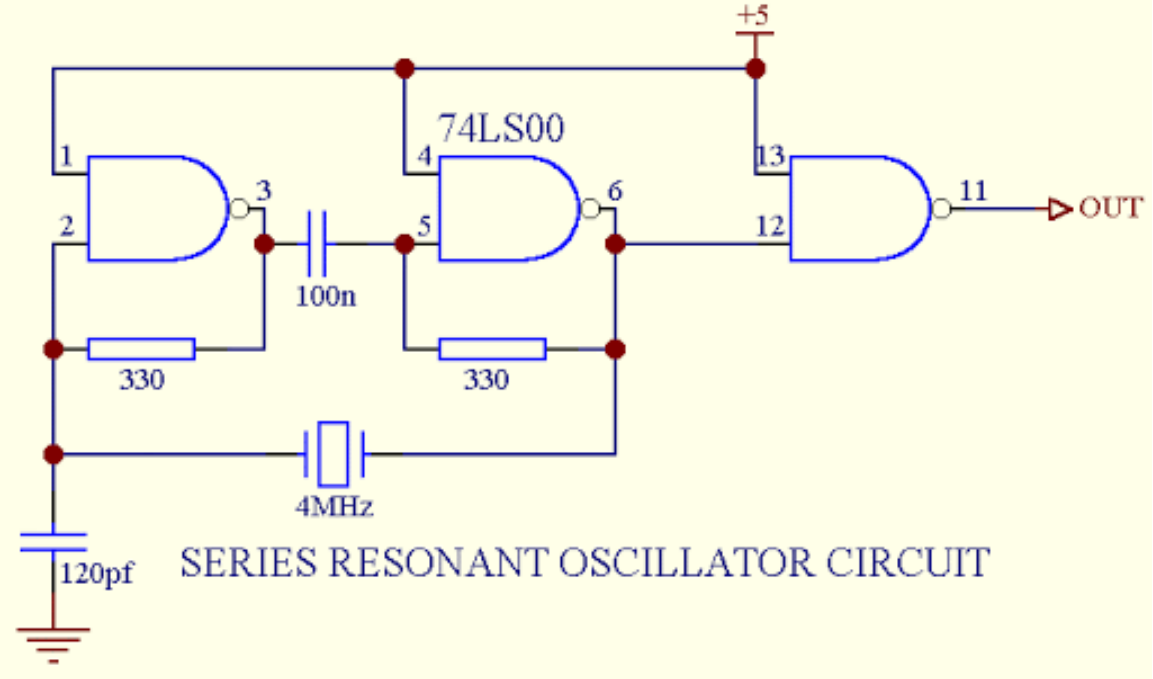
- Practical logic circuits can be very fast, but not infinitely fast.
- It takes time for the input stimulus to propagate through the circuit and to ultimately affect the output.
- In practice, this would form some type of oscillator circuit:



Oscillator Circuits

- In practice, this can be taken advantage of, but the problem is that the frequency can be quite unpredictable.
- You can slow down the response by coupling the output to the input using resistors and capacitors.
- The capacitors need to charge up to the appropriate voltage before the output will switch to the opposite state.

Oscillator Circuits



There are lots of different configurations but they have some common features.

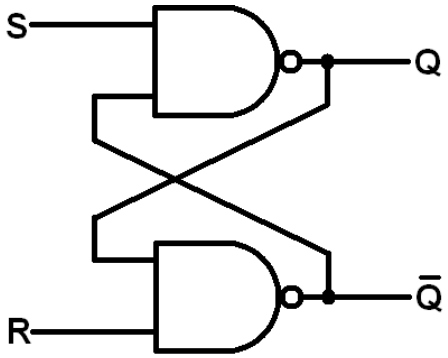
- In this case, both NAND gates are configured as inverters
- The combination of resistors and capacitors slows down the switching frequency
- The crystal acts like a resonant circuit that has a low impedance at a very specific frequency (in this case 4 MHz)
- It might be tricky to find the right component values to make a circuit like this work, but once it does, it can provide a VERY stable 4 MHz square wave output.

Sequential Logic

- By providing feedback between the output an the input of a combinatorial logic circuit, the output will now depend on:
  - The inputs
  - The past history of the output
- We can exploit these features to build some useful elementary sequential logic elements

Sequential Logic Circuits

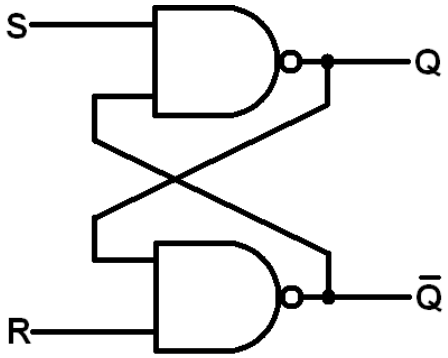
- A simple example is the "Set-Reset Flip-Flop" circuit:



- First suppose that $Q = 1, S = 1, R = 1$
  - Then $\bar{Q} = \overline{R \cdot Q} = 0$
- Next, suppose that $Q = 0, S = 1, R = 1$
  - Then $\bar{Q} = \overline{R \cdot Q} = 1$

Sequential Logic Circuits

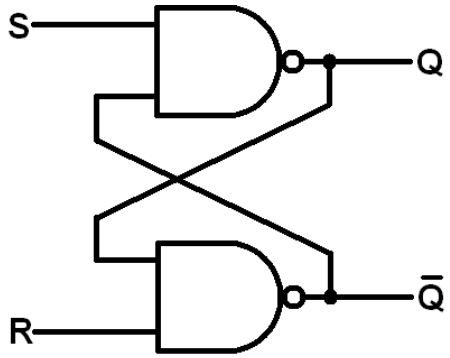
- A simple example is the “Set-Reset Flip-Flop” circuit:



- Now suppose that $S = 0, R = 1$
  - Then $Q' = \overline{S \cdot \bar{Q}} = \bar{S} + Q$
  - If $Q = 0$, then $Q' = 1$
  - If $Q = 1$, then Q' remains unchanged

Sequential Logic Circuits

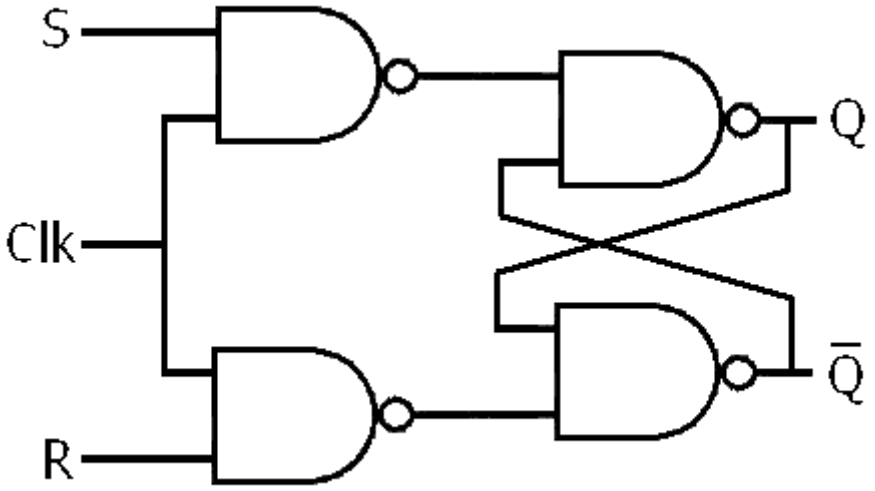
- A simple example is the “Set-Reset Flip-Flop” circuit:



- Next suppose that $S = 1, R = 0$
  - Then $\bar{Q}' = \overline{R \cdot Q} = \bar{R} + \bar{Q}$
  - If $\bar{Q} = 0$, then $\bar{Q}' = 1$
  - If $\bar{Q} = 1$, then $\bar{Q}'$ remains unchanged

Clock Inputs

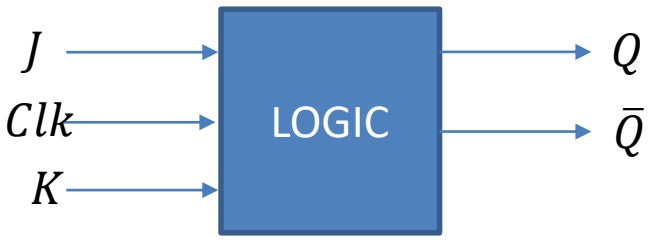
- We often want to sample an input at a specific time.
- This is implemented by adding a “clock” input



- The outputs don’t change until Clk=1

J-K Flip Flop

- It doesn't matter how this circuit is implemented. All that matters is how the outputs respond to the inputs:

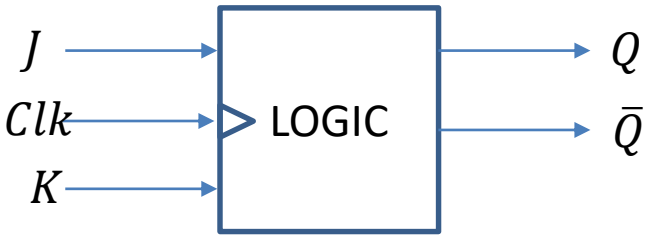


J	K	Clk	Q	Q'
X	X	0	X	Q
0	0	1	X	Q
0	1	1	X	0
1	0	1	X	1
1	1	1	X	$\bar{Q}$

- When J=K=1, the output will toggle when Clk=1

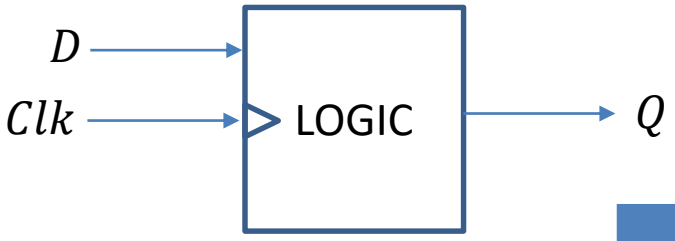
Edge-Sensitive Clock Inputs

- Outputs change only when the clock input makes the transition from 0 to 1.
- This is referred to as the “rising clock edge”
- Nothing changes on the “falling clock edge”
- The edge-sensitive nature of the clock input is indicated by a little wedge thing:



The D Flip-Flop

- Using these concepts, we can describe a very simple sequential logic element:

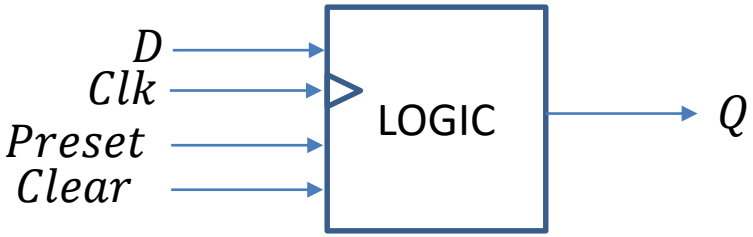


D	Clk	Q	Q'
0	⌐	X	0
1	⌐	X	1
X	⌐	X	Q

- We say that the input is “latched” on the rising edge of the clock input.

Asynchronous Inputs

- Sometimes we need to set the state of the output independent of the clock or D inputs
- Preset and Clear inputs can be asynchronous with the clock input:

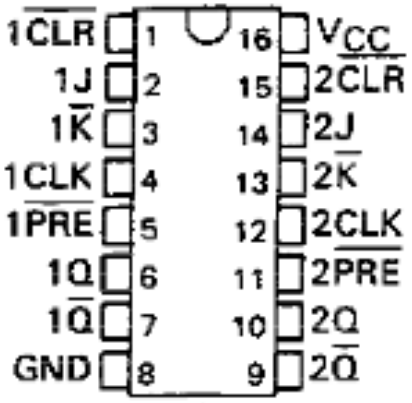


D	Clk	Preset	Clear	Q	Q'
0	⌐	0	0	X	0
1	⌐	0	0	X	1
X	⌙	0	0	X	Q
X	X	1	0	X	1
X	X	0	1	X	0
X	X	1	1	X	invalid

Examples

- 74LS109: Dual J-K edge-triggered flip flop with preset and clear:

SN54109, SN54LS109A . . . J OR W PACKAGE
SN74109 . . . N PACKAGE
SN74LS109A . . . D OR N PACKAGE
(TOP VIEW)



FUNCTION TABLE (each flip-flop)

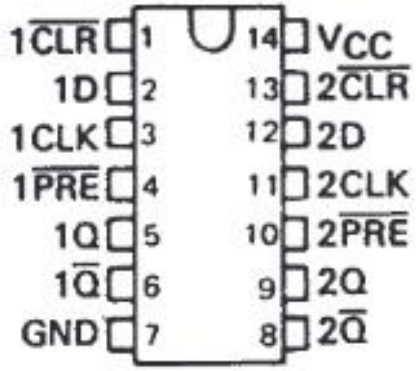
INPUTS					OUTPUTS	
PRE	CLR	CLK	J	K	Q	Q
L	H	X	X	X	H	L
H	L	X	X	X	L	H
L	L	X	X	X	H†	H†
H	H	↓	L	L	L	H
H	H	↓	H	L	TOGGLE	
H	H	↓	L	H	Q_0	$\overline{Q}_0$
H	H	↓	H	H	H	L
H	H	L	X	X	Q_0	$\overline{Q}_0$

† The output levels in this configuration are not guaranteed to meet the minimum levels for V_{OH} if the lows at preset and clear are near V_{IL} maximum. Furthermore, this configuration is nonstable: that is, it will not persist when preset or clear return to their inactive (high) level.

Examples

- 74LS74: Dual D Edge-triggered flip-flop with preset and clear

SN5474 . . . J PACKAGE
SN54LS74A, SN54S74 . . . J OR W PACKAGE
SN7474 . . . N PACKAGE
SN74LS74A, SN74S74 . . . D OR N PACKAGE
(TOP VIEW)



FUNCTION TABLE

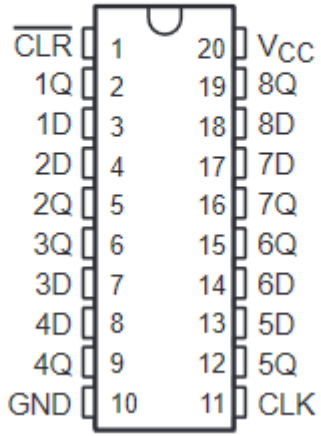
INPUTS				OUTPUTS	
PRE	CLR	CLK	D	Q	Q
L	H	X	X	H	L
H	L	X	X	L	H
L	L	X	X	H†	H†
H	H	↑	H	H	L
H	H	↑	L	L	H
H	H	L	X	Q_0	$\overline{Q}_0$

The output levels in this configuration are not guaranteed to meet the minimum levels in V_{OH} if the lows at preset and clear are near V_{IL} maximum. Furthermore, this configuration is nonstable; that is, it will not persist when either preset or clear returns to its inactive (high) level.

Examples

- Sometimes it is useful to latch many inputs on the same clock edge.
- 74LS273: Octal D-type flip-flop with clear

SN54273, SN74LS273 . . . J OR W PACKAGE
SN74273 . . . N PACKAGE
SN74LS273 . . . DW OR N PACKAGE
(TOP VIEW)



FUNCTION TABLE
(each flip-flop)

INPUTS			OUTPUT
CLEAR	CLOCK	D	Q
L	X	X	L
H	↑	H	H
H	↑	L	L
H	L	X	Q_0

Practical Limitations

- For the D input to be sampled correctly, it must be stable for a minimum period of time *before* the clock edge:

  Setup time, t_{SU}

- For the D input to be sampled correctly, it must remain stable for a minimum period of time after the clock edge:

  Hold time, t_H

Setup and Hold Time

recommended operating conditions

		SN54LS273			SN74LS273			UNIT
		MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, V_{CC}		4.5	5	5.5	4.75	5	5.25	V
High-level output current, I_{OH}				−400			−400	μA
Low-level output current, I_{OL}				4			8	mA
Clock frequency, f_{clock}		0		30	0		30	MHz
Width of clock or clear pulse, t_w		20			20			ns
Setup time, t_{su}	Data input	20↑			20↑			ns
	Clear inactive state	25↑			25↑			
Data hold time, t_h		5↑			5↑			ns
Operating free-air temperature, T_A		−55		125	0		70	°C

↑ The arrow indicates that the rising edge of the clock pulse is used for reference.

- Setup time: 20 ns
- Hold time: 5 ns
- These are very old devices that are used only for illustration.

Modern Example (10 GHz)

NB7V52M

1.8V / 2.5V Differential D Flip-Flop w/ Reset and CML Outputs

Multi−Level Inputs w/ Internal Termination

Table 1. INPUT/OUTPUT SELECT TRUTH TABLE

R	D	CLK	Q
H	x	x	L
L	L	Z	L
L	H	Z	H

Z = LOW to HIGH Transition
x = Don't care

ON Semiconductor®

http://onsemi.com

MARKING DIAGRAM*

1

QFN−16
MN SUFFIX
CASE 485G

16
1
NB7V
52M
ALYW▪
▪

A = Assembly Location
L = Wafer Lot
Y = Year
W = Work Week
▪ = Pb−Free Package

(Note: Microdot may be in either location)

*For additional marking information, refer to Application Note AND8002/D.

Table 5. AC CHARACTERISTICS V_{CC} = 1.71 V to 2.625 V; V_{EE} = 0 V; T_A = −40°C to 85°C (Note 10)

Symbol	Characteristic		Min	Typ	Max	Unit
f_{MAX}	Maximum Input Clock Frequency		10	12		GHz
$f_{DATA\ MAX}$	Maximum Input Data Rate (PRBS23)		10	12		Gbps
t_{PLH}, t_{PHL}	Propagation Delay to Differential Outputs, @ 1 GHz, Measured at Differential Cross−point	CLK/$\overline{CLK}$ to Q/$\overline{Q}$ R/$\overline{R}$ to Q/$\overline{Q}$		200 300	350 600	ps
t_S	Setup Time (D to CLK)		40	15		ps
t_H	Hold Time (D to CLK)		50	20		ps