

Prof. Mark D Shattuck
Physics 39907 Computational Physics
August 26, 2024

Problem Set 1

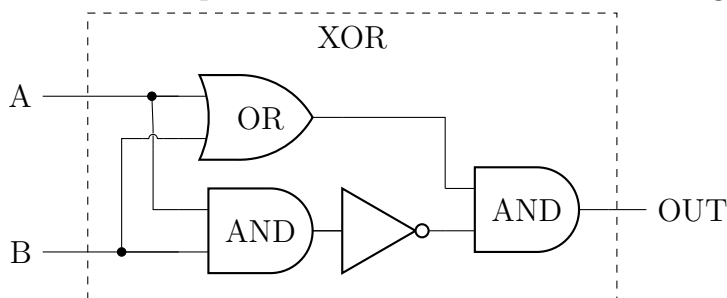
Question 1. *LaTeX and Overleaf:* Use the *Learn LaTeX in 30 minutes* tutorial found here: https://www.overleaf.com/learn/latex/Learn_LaTeX_in_30_minutes along with the matlab code below to reproduce the document here: [Monte_Carlo.pdf](#). Replace "Your Name" in the matlab code and the final document with your own name. This will produce data for that is specific to you. The matlab code can also be download as a file from here [circle.m](#). When you execute the code in circle.m in matlab it produces a figure file which you will need to upload to Overleaf and estimates the value of π . Use this data in your version of Mote_Carlo.pdf. Upload both the pdf and the .tex versions along with your version of circle.m (with your name in the correct spot) to blackboard. There are also nice video tutorials for LaTeX. Here is a short [one](#) and a longer [one](#). There is also a playlist with a number of good [videos](#).

```
1 %% Setup
2 N=1e4;           % Number of points
3 t=0:.01:2*pi;   % helper list
4
5 rng(sum('Your Name'),'twister'); % reset random number generator
6
7 %% Monte Carlo integration for the area of a circle
8
9 u=2*rand(2,N)-1; % Choose random point in the 2D plane [-1 1]x[-1 1]
10 x=u(1,:);       % Nicer names
11 y=u(2,:);
12
13 ii=(x.^2+y.^2<=1); % True for points inside circle of radius 1
14
15 %% Plot points
16 plot(x(ii),y(ii),'b.',x(~ii),y(~ii),'r.',sin(t),cos(t),'k');
17 axis('square')
18 set(gca,'fontsize',20);
19 xlabel('x-axis');
20 ylabel('y-axis');
21 title('Random Points');
22 print -depsc2 random.eps
23
24 %% Calculate and display Pi
25
26 myPI=4*sum(ii)/N;
27 disp(myPI)
```

Question 2. *Binary and Hexadecimal* For the these values $A = 47$, $B = 0b11011001$, $C = 0x5$, $D = 0b110$, $E = 0xe4$ find and express the following as binary and hexadecimal and decimal:

- (1) A (e.g., $A = 47 = 0b1011111 = 0x2F$)
- (2) B
- (3) C
- (4) D
- (5) E
- (6) $A + C$
- (7) $A + B$
- (8) $E - C$
- (9) $D \times B$

Question 3. *Universal Logic* Build a NOT gate and an OR gate from only NAND gates. To add your solution to your LaTeX (Overleaf) document you could use a package called CircuiTikZ that is part of a general drawing package called TikZ. It produces nice results like the following:



However there is more to learn. If you want to go this way, then there is a nice tutorial on Tikz and CircuiTikZ [here](#) and a good video on TikZ [here](#) and a related package pgfplots for plotting [here](#). Here is the CircuiTikZ code to make the diagram above for an XOR gate:

```
\documentclass{article}
\usepackage{tikz}
\usepackage{circuitikz}
\ctikzset{logic ports=ieee}

...

\begin{center}
\begin{circuitikz}

\draw
(0,0)      node (myor)   [or port]           {OR}
(0,-1.5)   node (myand1) [and port]          {AND}
(myand1.out) node (mynot) [anchor=west,not port] {}
(mynot.out) node (myand2) [anchor=in 2,and port] {AND};

\draw (myor.in 1)  -- ++(-1.8,0) node[left] (A)  {A};
\draw (myand1.in 2) -- ++(-1.8,0) node[left] (B)  {B};
\draw (myand2.out) -- ++(.5,0)  node[right] (out) {OUT};
```

```

\draw (myor.in 2) -- ++(-.2,0) node[left,jump crossing] (C1) {};
\draw (myor.in 1) -- ++(-.3,0) node[left,circ] (B1) {};
\draw (myand1.in 2) -- ++(-.7,0) node[left,circ] (B2) {};

\draw (C1.north) -- (B1);
\draw (C1.south) |- (myand1.in 1);
\draw (C1.west) -| (B2);
\draw (myor.out) -| (myand2.in 1);

\draw [dashed] (out) ++(-8,-1.2) rectangle ++(7,3.8) node[midway,yshift=45] {XOR};

\end{circuitikz}
\end{center}

```

Another option is to include a photo of a hand drawn picture:

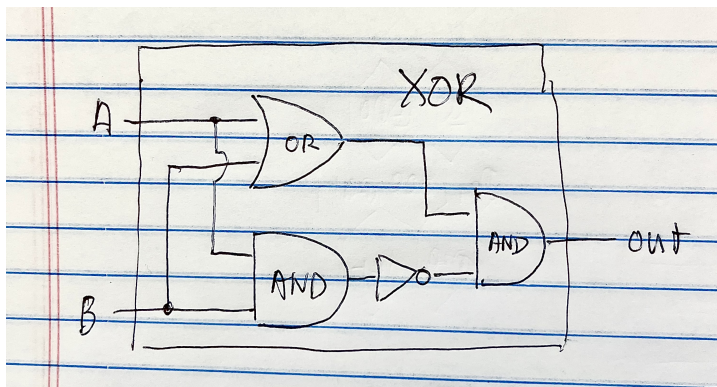


FIGURE 1. XOR from TikZ.

You could also draw it in a drawing program:

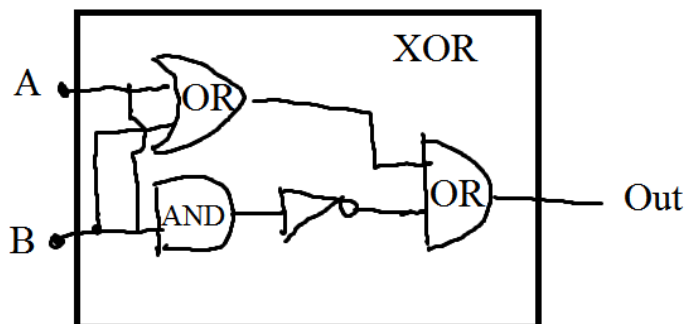


FIGURE 2. XOR from drawing program.

I also found several circuit drawing programs online. This one, [circuitlab](#), makes nice drawings but you have to pay to save it. So I used screen capture to make this:

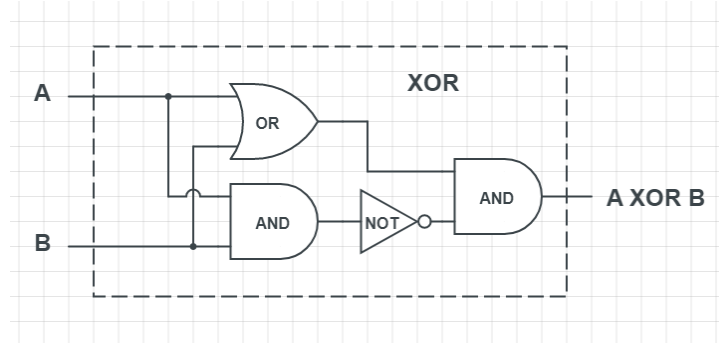


FIGURE 3. XOR screen capture from circuitlab.