EE330 Homework 2 Fall 2018 TA: Robert Buckley

Problem 1:

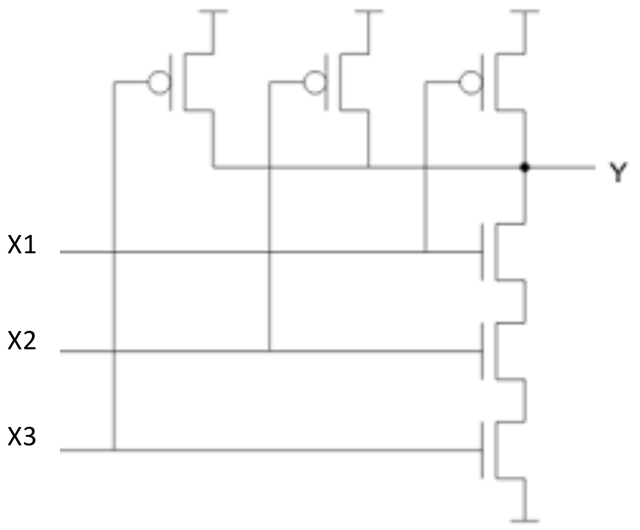
In figure 1.4 we can find the count of transistors in the processor increases by about ~10 times per seven years. In 2002 the count was about 0.1 Billion, so we can predict the count in 2020 to be about

0.1 billion * 10^(18/7) = 37.28 billion.

Problem 2:

The transistor level schematic for a CMOS 3-input NAND gate can be made like this (other ways are possible)

Y = $\overline{X1 \cdot X2 \cdot X3}$



X1, X2, and X3 are inputs

Y is the output

Problem 3:

Area of die = $A_{dir} = (5mm)^2 = 25\ mm^2 0.25cm^2$

Area of wafer = $A_{wafer} = \left(\frac{300}{2}\right)^2 * \pi = 70685mm^2 = 706.85\ cm^2$

Hard yield of die $Y_H = e^{-Adie*d} = e^{-.25*1.2} = 0.7408$

The total yield $Y = Y_H * Y_S = 0.7408 * 1 = 0.7408$

The cost per good die $C_{good} = \frac{C_{wafer}}{A_{wafer}/A_{die}} * \frac{1}{Y} = \frac{\$3200}{\frac{70685}{25}} * \frac{1}{0.7408} =$ \$1.529

Problem 4:

$$Y = \frac{x-u}{g} = \frac{3mv - 0mv}{1.5\ mv} = \frac{3}{2} = 1.5$$

$P_{Soft\_Amp} = \int_{-1.5}^{1.5} f(x)\,dx = 2F_N(1.5) - 1 = 0.866$

Two Op Amps per chip $P_{Soft\_chip} = 0.866^{\,2} = 0.750$

Problem 5:

Since the input offset voltage follow Gaussian distribution we have the soft yield θ:

$Y_s = P_{soft} = 2F_N(y) - 1 = 0.98 \Rightarrow F_N(y) = 0.990$

$$y = 2.33 = \frac{x-u}{g} = \frac{1.5mv - 0}{\left(\frac{20mV\mu m}{\sqrt{A}}\right)} \Rightarrow A = 965.14\ \mu m^2$$

Problem 6:

The area of a die in new process = $A_{die} = \left(\frac{7}{65}\right)^2 * 0.5cm^2 = 5.7988 * 10^{-3}cm^2$

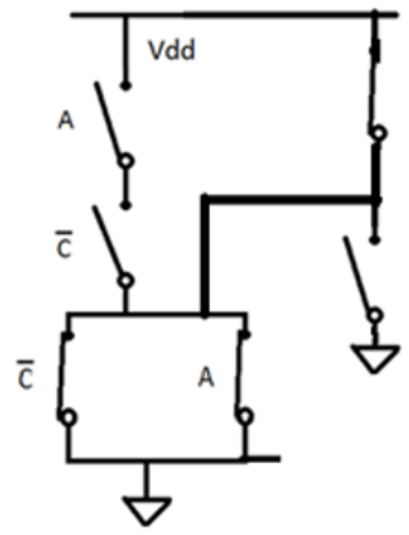
The hard yield = $Y_H = e^{-A_{die}*d} = e^{-0.00696} = 0.9931$

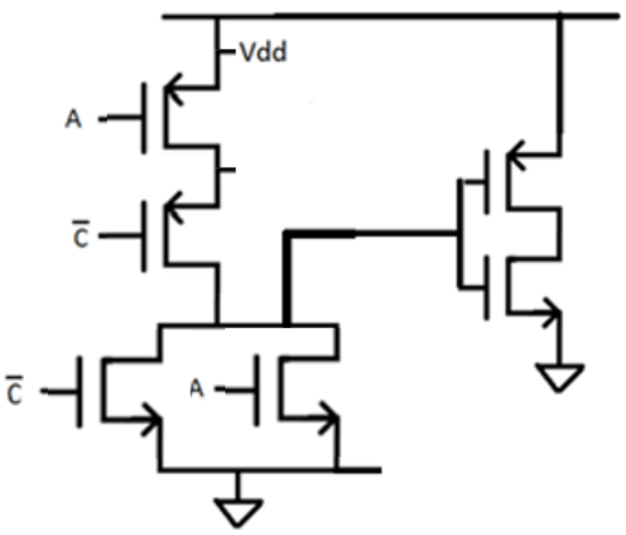
The yield $Y = Y_H * Y_S = 0.9931$

$$C_{good} = \frac{C_{wafer}}{\left(A_{wafer}/A_{die}\right)} * \frac{1}{Y} = \frac{\$10000}{\frac{1590.4}{0.005799}} * \frac{1}{0.9931} = \$0.0367$$

Problem 7:

This switch level model assumes A =1 and $\bar{C}$ = 1.





Problem 8:

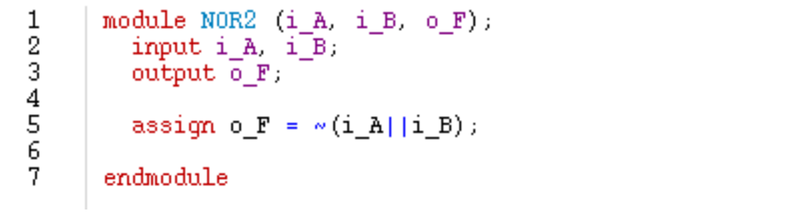
When both inputs are 0V the output is VDD = 3V
When one input is 0V and the other is 3V the output is VDD = 0V
When both inputs are 3V the output is VSS = 0V

Problem 9:

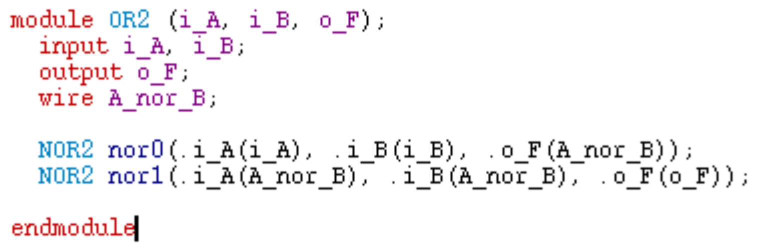
a- The master is addressing a slave and requesting data. Data is transferred to the master.
b- The master is sending data to slave and receiving acknowledgement again just like part a.
c- Open drain refers to the exposed drain of a transistor typically used as the output.
d- Logic contention refers to data being driven at any time from several devices. I2C follows a protocol so that the contention doesn't appear.
e- Due to the I2C open drain protocols the bus remains on a low threshold voltage. Devices can drive past the threshold so that the different logic levels are apparent

Problem 10:

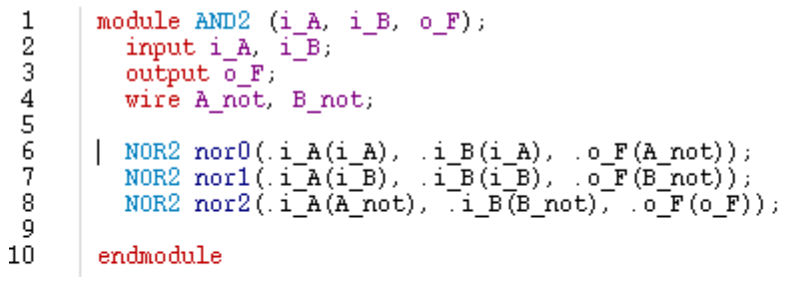
Here is one solution to this problem



```
module NOR2 (i_A, i_B, o_F);
  input i_A, i_B;
  output o_F;

  assign o_F = ~(i_A||i_B);

endmodule
```



```
module OR2 (i_A, i_B, o_F);
  input i_A, i_B;
  output o_F;
  wire A_nor_B;

  NOR2 nor0(.i_A(i_A), .i_B(i_B), .o_F(A_nor_B));
  NOR2 nor1(.i_A(A_nor_B), .i_B(A_nor_B), .o_F(o_F));

endmodule
```

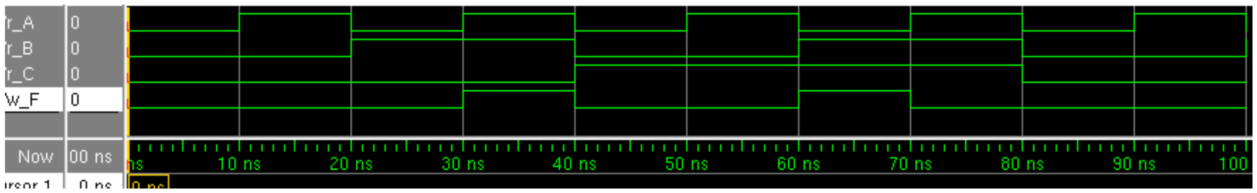


```
module AND2 (i_A, i_B, o_F);
  input i_A, i_B;
  output o_F;
  wire A_not, B_not;

  NOR2 nor0(.i_A(i_A), .i_B(i_A), .o_F(A_not));
  NOR2 nor1(.i_A(i_B), .i_B(i_B), .o_F(B_not));
  NOR2 nor2(.i_A(A_not), .i_B(B_not), .o_F(o_F));

endmodule
```

EE330 Homework 2 Fall 2018 TA: Robert Buckley

```
module HW2_Problem_TB();
  reg r_A, r_B, r_C;
  wire w_F;

  initial
  begin
    r_A = 1'b0;
    r_B = 1'b0;
    r_C = 1'b0;
  end

  always
    #10 r_A = ~r_A;
  always
    #20 r_B = ~r_B;
  always
    #40 r_C = ~r_C;

  HW2_Problem name(.i_A(r_A), .i_B(r_B), .i_C(r_C), .o_F(w_F));

endmodule
```

```
module HW2_Problem (i_A, i_B, i_C, o_F);
  input i_A, i_B, i_C;
  output o_F;
  wire A_not, C_not;
  wire nA_B, nA_B_C, B_nC, A_B_nC;

  NOR2 nor0(.i_A(i_A), .i_B(i_A), .o_F(A_not));
  NOR2 nor1(.i_A(i_C), .i_B(i_C), .o_F(C_not));
  AND2 and0(.i_A(A_not), .i_B(i_B), .o_F(nA_B));
  AND2 and1(.i_A(nA_B), .i_B(i_C), .o_F(nA_B_C));
  AND2 and2(.i_A(i_B), .i_B(C_not), .o_F(B_nC));
  AND2 and3(.i_A(i_A), .i_B(B_nC), .o_F(A_B_nC));
  OR2  or0(.i_A(nA_B_C), .i_B(A_B_nC), .o_F(o_F));

endmodule
```


r_A
0
r_B
0
r_C
0
w_F
0
Now
00 ns
ursor 1
0 ns
ns
10 ns
20 ns
30 ns
40 ns
50 ns
60 ns
70 ns
80 ns
90 ns
100
0 ns