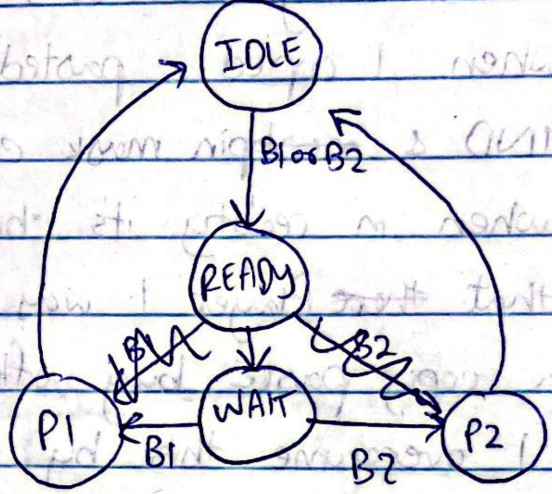
State Diagram



Description

IDLE = flash rapid

Ready = ~~steady~~ 3 flash wait

WAIT → steady

P1 = steady on LED1, LED2 = off, timed exit to IDLE

(or)

P2 = steady on LED2, LED1 = off, ~~timed exit to idle~~

Psuedocode

1. state is set to idle
2. check if button is pressed
   → if either button is pressed and state is idle set state to ready
   → if state is ~~ready~~ wait and button is pressed, ~~turn on respective leds~~ & set state to player
3. After ~~a~~ steady flash of winner's LED in player state, return to Idle state & ~~repeat~~ start from step 2

Problems ~~→ wrong winner~~

- ~~turn off winner light~~
- ~~only 1 player working~~

The bottom of my previous page shows the problems I faced when testing my code. Initially only one player was working. This was due to a typo in my code when I copied & pasted a section (I was checking if PIND & ~~mask~~ pin mask equalled to 1 for a button press, when in reality it's button not pressed. Then I noticed that ~~there~~ Player 1 was always winning. This too was a copy paste bug after I copied a section of code. I overcame this by making Player 2's button press go to Player 2 state instead of Player 1 state. Lastly, I forgot to clear bits B0 & B2 when starting a new game. This left the winner's light turned on. I fixed this by clearing both bits when entering the IDLE state (~~to~~ which starts a new game).

My ~~machine~~ code compiled to 630 bytes.

PreLab 2

a. R_{pu} has a range of 50-20 $k\Omega$. V_{IH} has a range of $0.7V_{cc}$ ~~thn~~ $V_{cc}+0.5$ for V_{cc} = 1.8-2.4V and a range of $0.6V_{cc}$ - $V_{cc}+0.5$ for V_{cc} = 2.4-~~3~~5.5 V.

(Ignore scratch.)

Prelab 2

a. These values are found in the DC Characteristics table on page 313. This is in chapter 26, Electrical Characteristics.

b. $R_{PU} = 20\ k\Omega$ $\qquad V_C = \frac{Q}{C}$ 0

$V_{DD} = 5V$

$I = \frac{V_{DD} - V_C}{R_{PU}} = \frac{V_{DD} - 0}{R_{PU}} = \frac{8}{\frac{20 \times 10^3}{4}} = 0.00025\ A$

c. $V_c(t) = V_{IH} = V_{dd}\left(1 - e^{-\frac{t}{R_{PU}C}}\right)$ for $t > 0$

$\frac{V_{IH}}{V_{dd}} = 1 - e^{-\frac{t}{R_{PU}C}}$

$e^{\frac{-t}{R_{PU}C}} = 1 - \frac{V_{IH}}{V_{dd}}$

$\frac{-t}{R_{PU}C} = \log_e\left(1 - \frac{V_{IH}}{V_{dd}}\right)$

since time can't be negative

$\boxed{t = -R_{PU}C \times \log_e\left(1 - \frac{V_{IH}}{V_{dd}}\right)}$

d. If a larger capacitor is used, t would go up because the two are directly related. This makes sense physically because the capacitor with greater capacitance would take longer to charge.

e. My oscilloscope showed a bounce time of $40\,\mu s$. Since we need a t longer than bounce time, t in the equation has to be atleast $41\,\mu s$.

$$t = -R_{PU}C \times \log_e\left(1 - \frac{V_{IH}}{V_{DD}}\right)$$

~~for min V_{IH}~~
$$C = \frac{t}{-R_{PU} \times \log_e\left(1 - \frac{V_{IH}}{V_{DD}}\right)}$$

For min V_{IH} of $0.6\,V_{DD}$:

$V_{IH} = 3V$

$V_{DD} = 5V$

$R_{PU} = 20\,k\Omega$

$t = 41\,\mu s$.

$$C = \frac{41 \times \cancel{10^{-6}}}{20 \times 10^{9} \times \log_e\left(1 - \frac{3}{5}\right)}$$

$$C = \frac{41}{+1.83 \times 10^{10}} = +2.24 \times 10^{-9}\,F$$

For max V_{IH} of $V_{DD} + 0.5$:-

$V_{IH} = 5.5\,V$

$V_{DD} = 5V$

$R_{PU} = 20\,k\Omega$

$t = 41\,\mu s$

$$C = \frac{41 \times \cancel{10^{-6}}}{20 \times 10^{9} \times \log_e\left(1 - \frac{5.5}{5}\right)}$$

$C =$

f. All of these capacitors didn't show the switch bouncing. This makes sense because all of these have a bigger capacitance than the one we used in class, which was small enough to prevent button bounces.

g. 20 kΩ is the worst case because it allows maximum current to flow. With maximum current flowing, the wires would have a lot of ~~energy~~ current which increases the chances of the button bouncing. With a 50 kΩ resistor, there is less current flowing in the circuit and the chances of electricity jumping between switch plates is less. Basically, the chances of button bouncing is reduced.

Design Challenges

1. Staff initials that your pushbutton-controlled counter works well: DM.

2. Staff initials that your reaction-time game works well: DM.

Reaction-time game size (bytes), from avrdude: 630.

In addition to this solution sheet and your final, well organized and documented C code, include your map of the game logic, pseudo-code, and a diary of your design and debugging process.