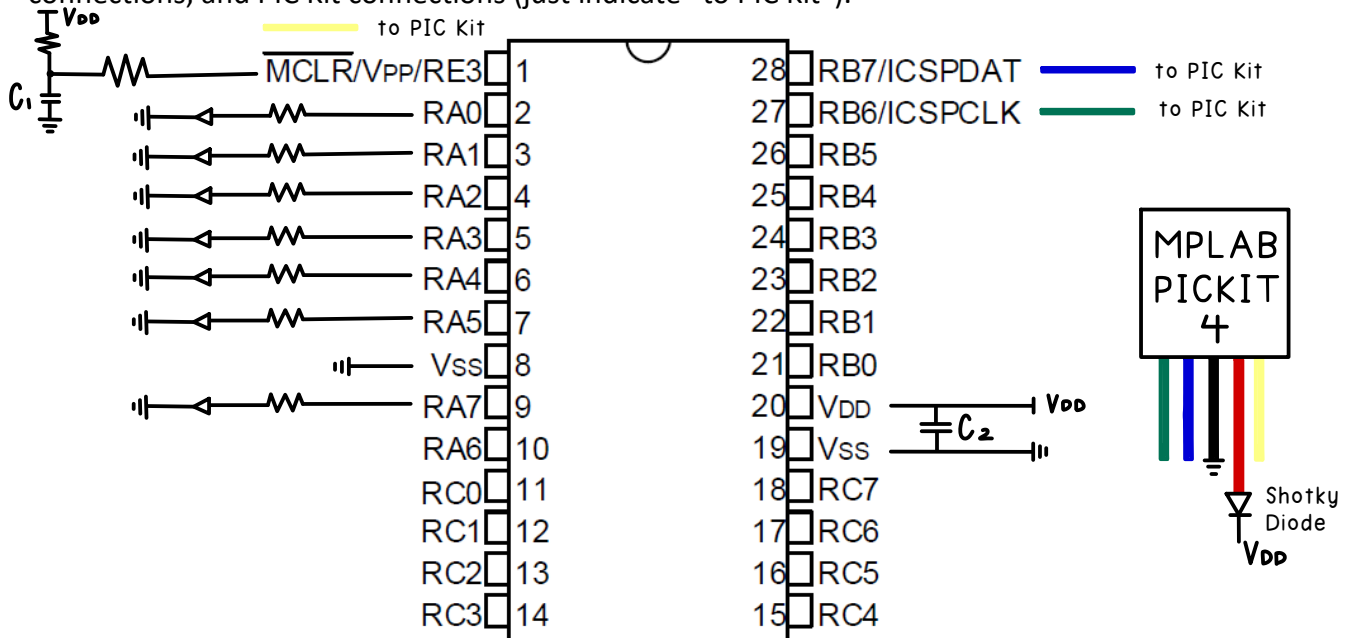


Think about what the LEDs would look like as the counter goes from 0 to 16.

Now, build this circuit on your breadboard. Use RA0 for bit 0, RA1 for bit 1, ..., RA5 for bit 5. You've already done RA0 in PIC Activity 2. The only trick is figuring out how to physically get everything on your breadboard that will cause the least amount of trouble for wiring. Make sure, whatever you do, that the LEDs are lined up beside each other so the counter will look decent.

Add one more LED on RA7. This LED should be initialized ON when you set up the registers in your code and should never turn off as your counter runs. This is your indicator that you are writing only the counter bits in the output register.

Draw in the circuit components on the image below to create your circuit schematic. **This image is one of the deliverables for this activity**, so be neat. Label the parts with values so someone looking at your schematic could build the circuit. Don't forget power and ground connections, and PIC Kit connections (just indicate "to PIC Kit").



```
void GpioSetup(void) {  
    Initialization (this only runs once when the code starts)  
        - Configure TRISA, ANSELA, and LATA (or PORTA)  
          registers  
          TRISA: set to 0x00  
          ANSELA: set to 0x00  
          PORTA: set to 0x00  
}
```

```
void UserAppRun(void) {  
    Initialization  
        - LATA: set to 0x80  
  
    for loop  
        (counter starts at zero and counts to maximum value of  
         64 and increments)  
        - set LATA to increment  
        - 2 Hz blink rate  
        - delay function to hold for 250 ms  
}
```