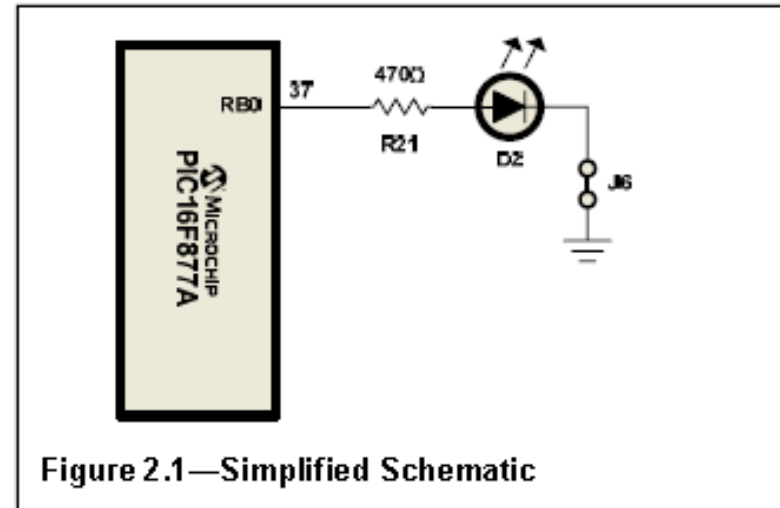


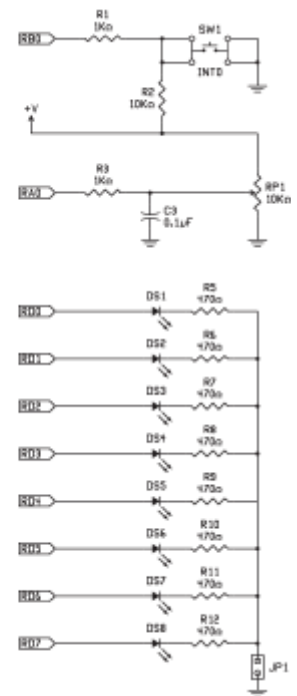
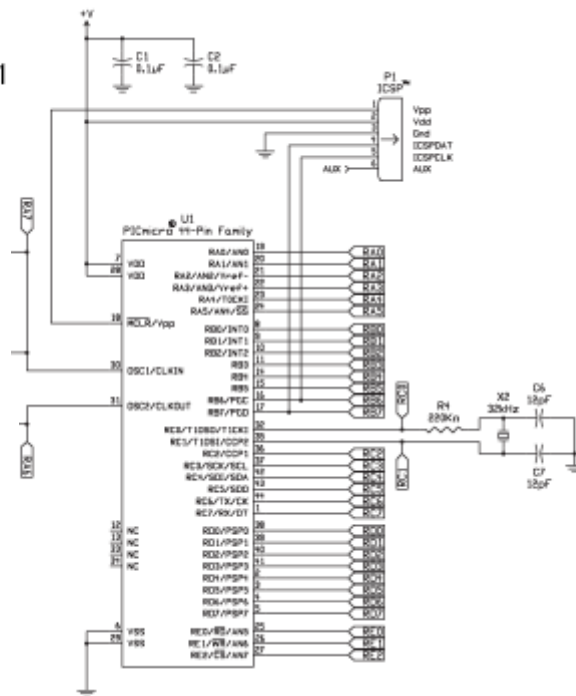
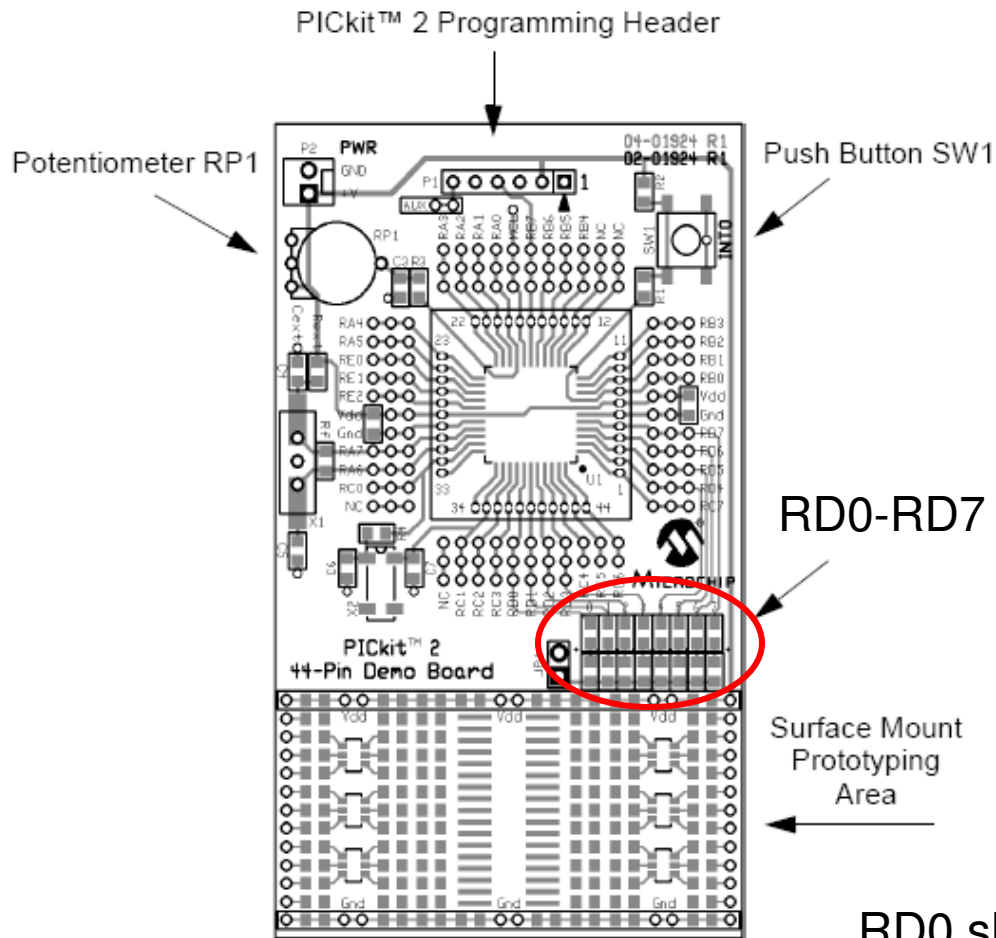
**BLINK an LED**

# BLINK

- This lesson shows how to turn on a LED, and make it blink.
- While this might seem a trivial change from previous lesson, it gives a context to explore several more instructions.



# 44 Pin Demo Schematic and Layout



RD0 should blink!

# Why do we need a delay?

- With the PICmicro running at 4MHz, each instruction takes only  $1\mu\text{s}$  to execute. Because of this, it will be necessary to slow down the main program loop which would only take  $4\mu\text{s}$  to execute without any delays.
- If we don't include the delay, the LED will blink so fast that it will appear to be on constantly, though at a level slightly less than full brightness.
- The delay routine consists of two registers used to implement a 16-bit counter. In the delay loop, this counter will be decremented through each pass until it reaches zero.
- If you are curious as to how long this delay lasts, it is fairly easy to calculate if you remember that all instructions execute in a single cycle ( $1\mu\text{s}$ ) except for program branches which take two cycles ( $2\mu\text{s}$ ).

# Delay Loop

```
const int TwoHundredms = 11500;
```

```
while(i<TwoHundredms)
```

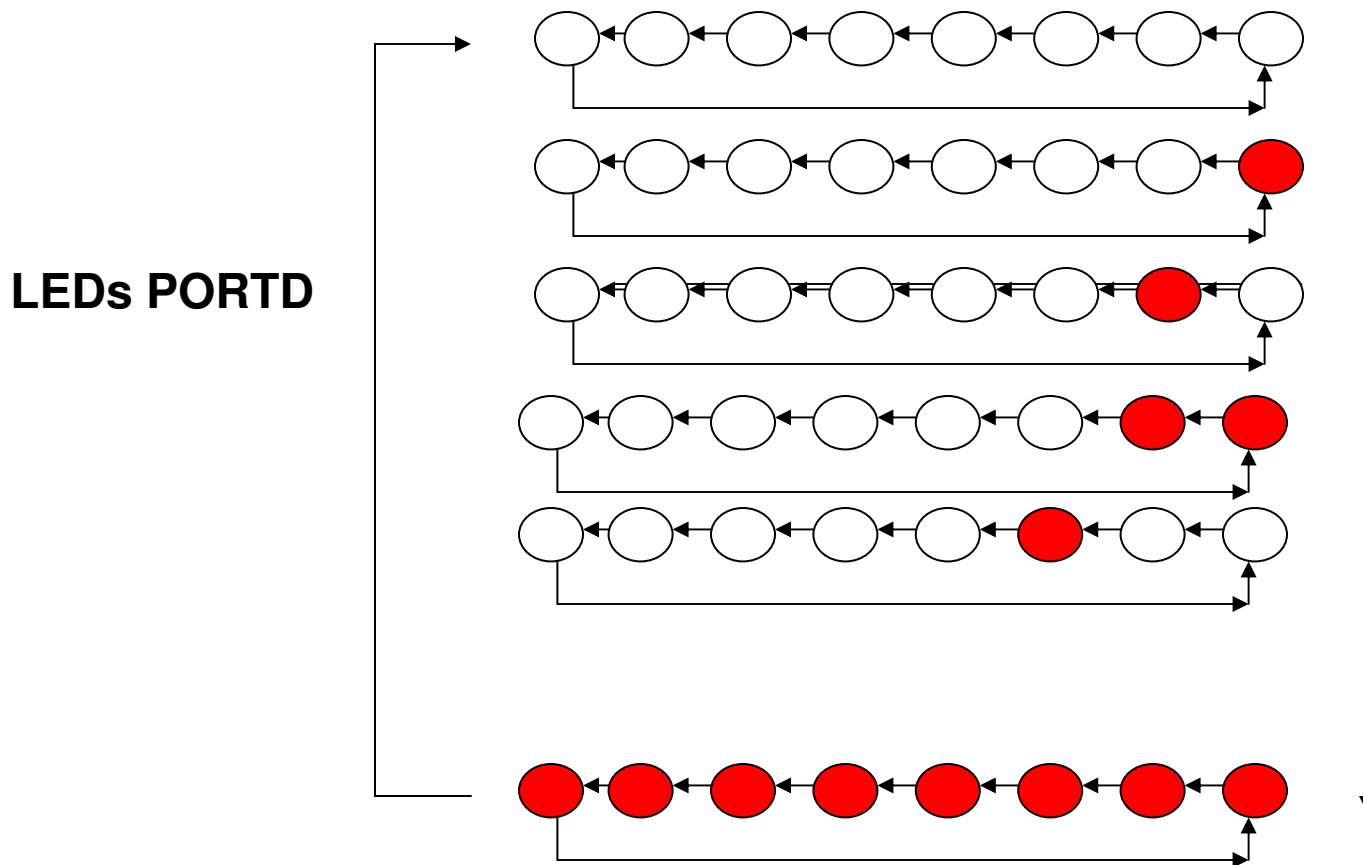
```
{ i++; }
```

# Exercise-Measure Delay

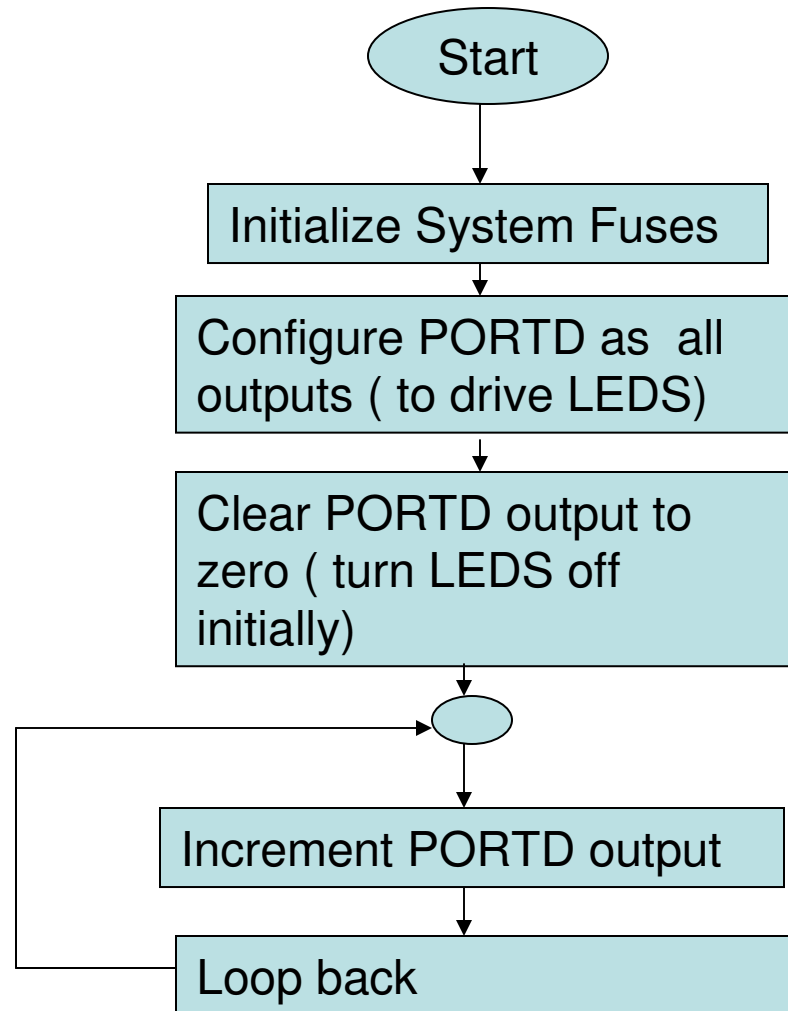
1. Navigate to:C:\EET250\16F887\Lesson 4 Blink with Demo Board\blink
2. Open blink.mcp
3. Examine code what does  $RD0 = RD0 \wedge 1$ ; do?
4. verify RD0 blink operation by download and running demo board using PICKIT2
5. Pause operation once blinking is verified
6. Switch debugger to simulator
7. Make sure simulator is using correct clock value
8. Debugger -> setting set clock to 4 MHz.
9. Set 1<sup>st</sup> breakpoint at code line  $i=0$ ;
10. Set 2<sup>nd</sup> breakpoint at code line  $RD0 = RD0 \wedge 1$ ;
11. Reset processor- run to first breakpoint.
12. Get debugger -> stopwatch. Reset stopwatch
13. Continue execution to second breakpoint.
14. Stop watch should contain the delay value
15. What is it?

# Dynamic Counting

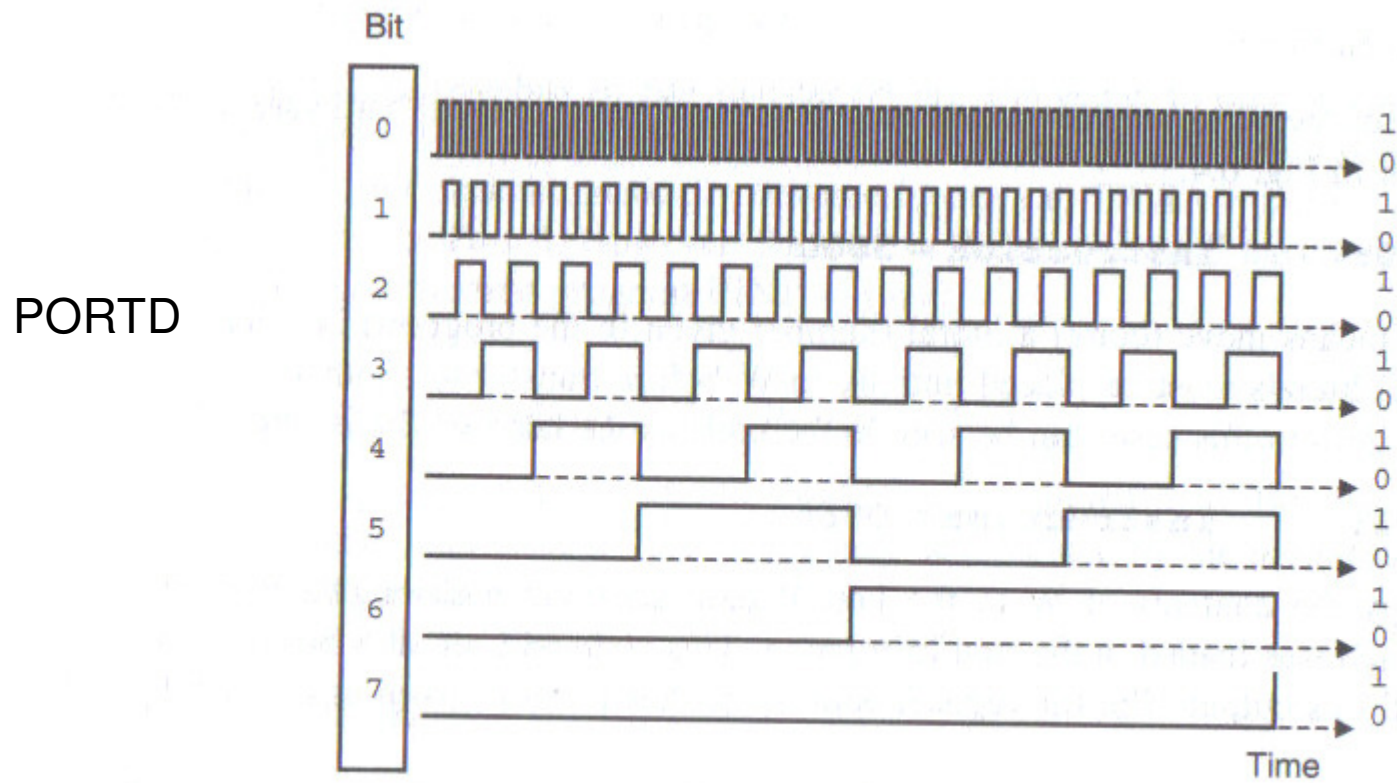
- Building on previous lessons, which showed how to light up a LED as count is incremented in output register



# Program Structure



# Execution Model



**Figure 6.5** Waveforms produced by program BIN at Port D

# Exercise-run visual counter

- Navigate to: C:\EET250\16F887\Lesson 4 Blink with Demo Board\counting
- Open blink.mcp
- Examine source code and insure it follows program structure indicated earlier slide
- Build, download into demo board and run using PICKIT2
- Verify counting operation on LEDs
- Optional –adjust delay count setting and rebuild, reprogram to note effects