Lab 3: LCD Counter in Assembly

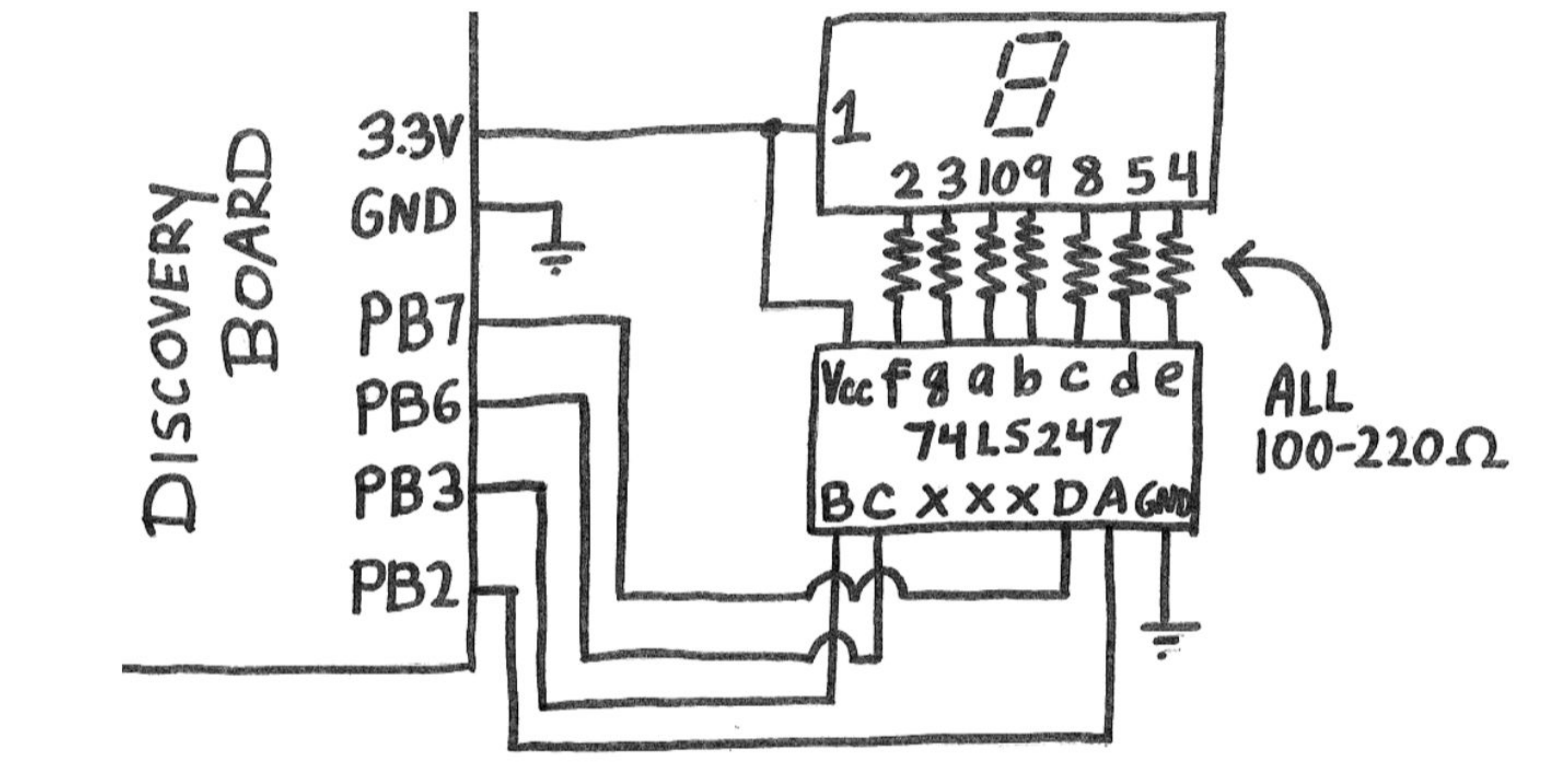
ECE 2020

Lab Outline

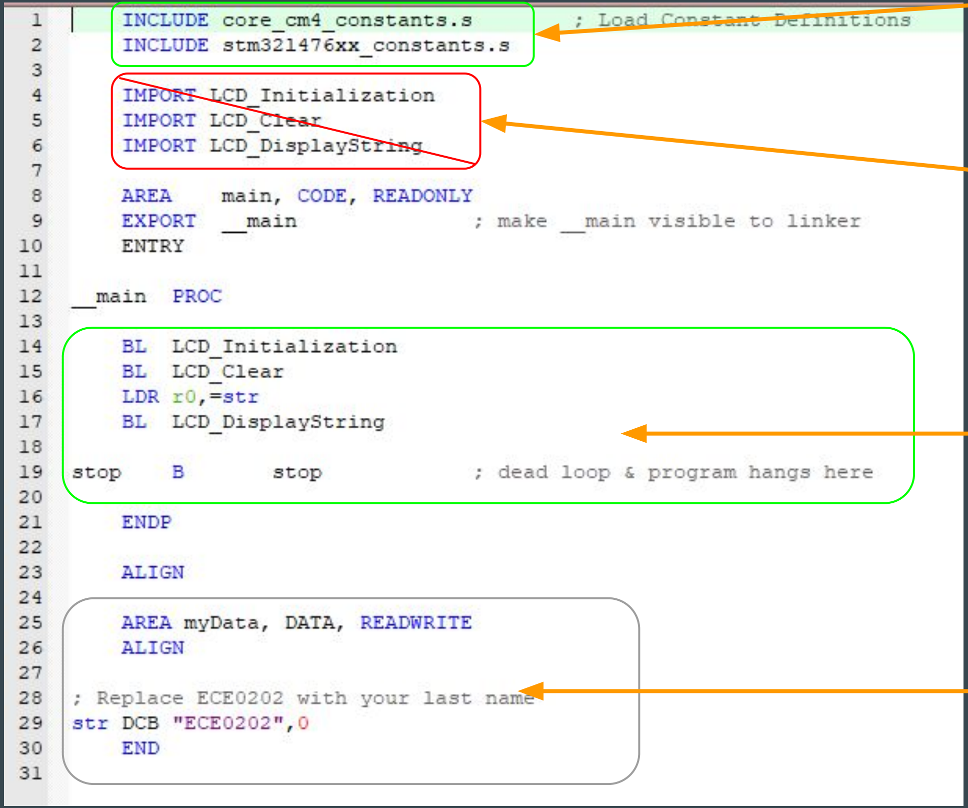
1. Download assembly template project from courseweb (same as lab 1)
2. Enable peripheral GPIO pins (4 output, 1 input)
3. Write a loop that updates output pins after a delay
4. Write logic within that loop to check if reset has been pressed

- ➢ Show us counting 0-9 on the LCD (checkoff)
- ➢ Show us that you have a reset button (checkoff)
- ➢ Submit code (e.g. ‘main.s’) on courseweb

The Circuit


DISCOVERY BOARD
3.3V
GND
PB7
PB6
PB3
PB2
1
2 3 10 9 8 5 4
Vcc f g a b c d e
74LS247
B C X X X D A GND
ALL
100-220Ω

Editing the “main.s” file



```
    INCLUDE core_cm4_constants.s        ; Load Constant Definitions
    INCLUDE stm32l476xx_constants.s

    ~~IMPORT LCD_Initialization~~
    ~~IMPORT LCD_Clear~~
    ~~IMPORT LCD_DisplayString~~

    AREA    main, CODE, READONLY
    EXPORT  __main              ; make __main visible to linker
    ENTRY

__main  PROC

    BL  LCD_Initialization
    BL  LCD_Clear
    LDR r0,=str
    BL  LCD_DisplayString

stop    B       stop            ; dead loop & program hangs here

    ENDP

    ALIGN

    AREA myData, DATA, READWRITE
    ALIGN

; Replace ECE0202 with your last name
str DCB "ECE0202",0
    END

```

This gives us the memory locations of our registers

We don’t need these since we aren’t using the internal LCD

Replace this with your own code

Doesn’t matter if you keep this stuff or not

Enabling GPIO pins in Assembly

We need to do something called **pre-indexing**

1. Load the base register address
2. Load the value located in base register + offset
3. Change that value with bitwise operations
4. Store the value in the address of base register + offset

```
LDR r0, =GPIOE_BASE
LDR r1, [ro, #GPIO_PUPDR]
ORR r1, #0x1
STR r1, [ro, #GPIO_PUPDR]
```

Control Logic

```
// My Pseudocode

while(1){

    delay = 100;
    while (delay !=0){
        delay -=1;
    }
    count++;
    display(count);
    check(reset);
}
```

Our main loop (after pin initialization)

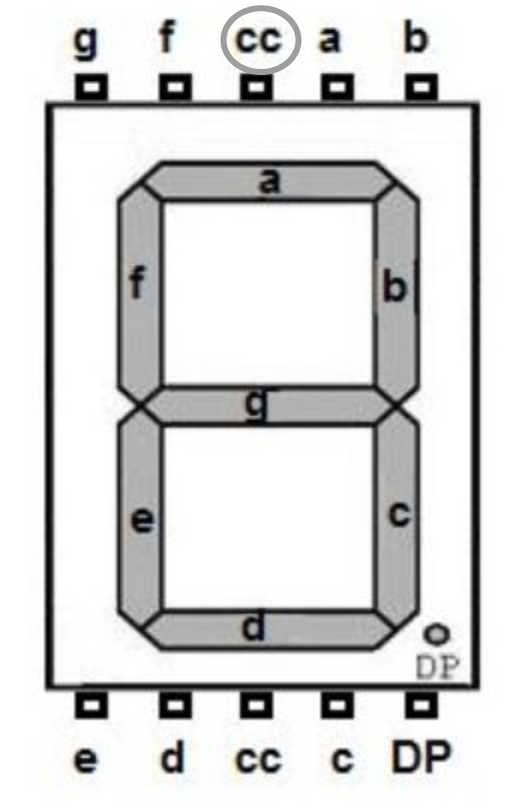
Some delay operation

After the delay, we update our counter

Once per loop we check our reset button

LCD Pinout

3.3V


g
f
cc
a
b
a
f
b
g
e
c
d
DP
e
d
cc
c
DP