question | Like **19** views

lab10 still not working - tests 1-3 pass, test 4 fails - not sure what is wrong?

Here is a code snippet and the command window output of the tests that both pass and fail.
It appears that the code works fine for test 1 where all 3 sensors are set, test 2 where the the
walk light should come on, test 3 where the west rd green light should come on, but it fails (times out)
for test 4 where the south rd green light should come on. I've double checked my FSM and the code
but not seeing my error. Any help appreciated! I'm continuing on for now to the week c11 videos - trying
to catch up this week a bit! Thanks.

```
Running with Code Size Limit: 32K
Load "C:\\Keil\\Labware\\Lab10_TrafficLight\\Lab10.axf"


*** Restricted Version with 32768 Byte Code Size Limit
*** Currently used: 18844 Bytes (57%)


Start grading
Clock rate appears to be : 80 MHz
Running 5 tests


0) Initialization tests :


- (PE2-0) have been selected as the input pins
- Verifying input configuration...
Pass: PORTE DEN bits (2-0) are high
Pass: PORTE DIR bits (2-0) are low
Pass: PORTE AFSEL bits (2-0) are low
Pass: PORTE AMSEL bits (2-0) are low
Pass: PORTE PCTL bits (11-0) are low


- (PB5-0) have been selected as the output pins
- Verifying output configuration...
Pass: PORTB DEN bits (5-0) are high
Pass: PORTB DIR bits (5-0) are high
Pass: PORTB AFSEL bits (5-0) are low
Pass: PORTB AMSEL bits (5-0) are low
Pass: PORTB PCTL bits (23-0) are low


1) Servicing all 3 requests, lights should make a complete cycle
FSM: Transition: 2 to 3
FSM: Transition: 3 to 4
FSM: Transition: 4 to 5
FSM: Transition: 5 to 1
FSM: Transition: 1 to 6
FSM: Transition: 6 to 1
FSM: Transition: 1 to 7
FSM: Transition: 7 to 1
FSM: Transition: 1 to 7
FSM: Transition: 7 to 9
FSM: Transition: 9 to 2
Pass: All requests were serviced
Pass: South request was serviced
Pass: West request was serviced
Pass: Walk request was serviced
- Test PASSED


2) Servicing walk button, walk light should come on
FSM: Bad Transition: 2 to 1
FSM: Transition: 1 to 6
Pass: All requests were serviced
Pass: Walk request was serviced
```

```
- Test PASSED


3) Servicing west button, west green light should come on
FSM: Transition: 6 to 1
FSM: Transition: 1 to 7
FSM: Transition: 7 to 1
FSM: Transition: 1 to 7
FSM: Transition: 7 to 9
FSM: Transition: 9 to 2
Pass: All requests were serviced
Pass: West request was serviced
- Test PASSED


4) Servicing south button, south green light should come on
* FAIL: Did not service all requests before the timeout
* FAIL: South request was not serviced
- Test FAILED


Done grading. Score is 65
```

Code snippet: (Note: plan to rename the states according to what was in the lab description - somehow I missed the suggestions on what they should be called when I first wrote it!

```
STyp FSM[9]={
 {0x31,50,{goWest, waitWest, goWest, waitWest, goWalk, waitWest, goWalk, waitWest }},
 {0x51,50,{goSouth, goSouth, goSouth, goSouth, goSouth, goSouth, goSouth, goSouth }},
 {0x85,50,{goSouth, goSouth, waitSouth, waitSouth, goWalk, goWalk, waitSouth, waitSouth }},
 {0x89,50,{goWest, goWest, goWest, goWest, goWest, goWest, goWest, goWalk }},
 {0x92,50,{goWest, hurryWalk, hurryWalk, hurryWalk, goWalk, hurryWalk, hurryWalk, hurryWalk }},
 {0x91,50,{noWalkNoCars1, noWalkNoCars1, noWalkNoCars1, noWalkNoCars1, noWalkNoCars1, noWalkNoCars1, noWalkNoCars1, noWalkNoCars1  }},
 {0x90,50,{dontWalkNoCars, dontWalkNoCars, dontWalkNoCars, dontWalkNoCars, dontWalkNoCars, dontWalkNoCars, dontWalkNoCars, dontWalkNoCars }},

 {0x91,50,{noWalkNoCars2, noWalkNoCars2, noWalkNoCars2, noWalkNoCars2, noWalkNoCars2, noWalkNoCars2, noWalkNoCars2, noWalkNoCars2 }},
 {0x90,50,{goWest, goWest, goWest, goWest, goWest, goWest, goWest, goWest }}};
#define SENSOR                  (*((volatile unsigned long *)0x4002401C)) //bits 2-0 port E
#define LIGHT                  (*((volatile unsigned long *)0x400050FC)) //bits 5-0 port B
  while(1){
        //Moore machine - output based on current state
    LIGHT = FSM[S].Out >> 2;  // set west and south road traffic LED lights (PB5-0)
        GPIO_PORTF_DATA_R = ((FSM[S].Out & 0x2) << 2) | ((FSM[S].Out & 0x1) << 1); // set walk/don't walk leds (PF3 and PF1)
      //wait for time relevant to state
    SysTick_Wait10ms(FSM[S].Time);
        //get input sensors for cars (one for west rd, one for south rd) and one for pedestrian
    Input = SENSOR;     // read sensors (SENSOR defines to read bits PE2-0 ---no need to shift right 2 bits defined this way)
        //Moore machine - next state based on Input and current state
        S = FSM[S].Next[Input];
  }
```

lab10

5 hours ago by Karen West

the students' answer, *where students collectively construct a single answer*

Click to start off the wiki answer

followup discussions *for lingering questions and comments*

Resolved [✓ Unresolved]

john long 4 hours ago

would love to help but just cant grasp the way you programmed your lights .. eg **{0x31,50,{goWes**t persuming this is the code for your green light on west red on south and red on walk ... but to me that turns on a yellow light and red light on west PB4 and PB5 ... 3 being 0011 and also 1 being 0001 ... PB0 ... turning on a red light on south ...

and cant see how this is turning on any lights on walk lights

so would need a explanation on how you are working your lights as i and any other FSM i have seen posted use this format **{0x0C,0x02,100,{GreenW .**... 0x0C is for the lights for the cars being that 0x0**C** is 1100 which is PB2 and PB3 which is red south and green west ...... then the **0x02** is for the walk lights which is 0010 which is for PF1 which is red on walk

if you could explain your light set up might be able to help out

Karen West 3 hours ago GoWest (state 0):

PB5-PB3 = 001, WestGreen with PB3 = 1 = Green LED for breadboard in simulator (and I thought I saw that correctly)

PB2-PB0 = 100, SouthRed with PB2 = 1 = Red LED for breadboard in simulator

PF3 and PF1 = 01, PF3 = 0, PF1 = 1 = Don't Walk Red LED for LaunchPad

Put PB5-3, PB2-0, PF3 and PF1 in one 8 bit character and you get: 00110001 = 0x31

This requires to shift the bits when you do the outputting to the breadboard and launchpad LEDs in 2 steps.
So this line sets PB5-0 breadboard LEDs (and in the above it sets West Rd = Green, and South Rd = Red):
LIGHT = FSM[S].Out >> 2; // set west and south road traffic LED lights (PB5-0)
noting that Port B bits 5-0 do not need to shift the bits left by 2 once you output it to the LIGHT--the right shift from the Out puts them in the correct position, since PF3 and PF2 are at bits 1 and 0 in the Out word.
This was some trick they showed us in how the address is defined so that you don't have to shift the bits left for Port B, I think??
#define GPIO_PORTB_OUT (*((volatile unsigned long *)0x400050FC)) // bits 5-0
#define LIGHT (*((volatile unsigned long *)0x400050FC)) //bits 5-0 port B

and this line sets PF3 and PF1 LaunchPad LEDs:
GPIO_PORTF_DATA_R = ((FSM[S].Out & 0x2) << 2) | ((FSM[S].Out & 0x1) << 1); // set walk/don't walk leds (PF3 and PF1)

and Port F bit 3 is in position 3 and bit 1 is in position 1, so Out bit 1 = port F bit 3, so and it with 0x2 and shift it left 2 bits to position 3 for port F
and Out bit 0 = port F bit 1, so AND it with 0x1, and then shift it 1 bit to the left to Port F bit 1 position
So in the above code, PF3 and PF1 = 01 = So Walk Light = Red

How did everyone else do their FSM?
0x0C corresponds to PB0-5: 001100 (the reverse order of the way I defined them? and I also added PF3 and PF1 to the end of the 8 bit?)
So in this way, your go west road is green (001 = PB2 = 1) and the south rd is red (100 = PB3 = 1)

I could be mistaken but I thought that I had the same definition for the outputs as you did? If you see it some other way, please help me clear up my confusion. I thought I had viewed my LEDs (when stepping through my code last Friday) doing the correct thing as far as what color I thought I was outputting at what time. So I thought that it was working when the 1st 3 tests passed, and then found that test 4 had failed as described above so I guess there must be a problem somewhere but I have not been able to track it down. Thanks!

john long 1 hour ago define SENSOR (*((volatile unsigned long *)0x4002401C)) PortE
#define LIGHT (*((volatile unsigned long *)0x400050FC)) // PortB
#define WALKLIGHT (*((volatile unsigned long *)0x40025028)) //Portf with 1,3 added

not getting the lights but lets look at it from another angle

ok give me your state list

eg {**0x31,50**,{goWest is this green on West
is this {0x51,50 waitwest ..... if so logical next states are either to go to **green light on either South or to green light on walk** and you have all goSouth which is mostly right but you have no logical path to green light on walk from waitWest

in the very first line of your FSM you have ... {0x31,50,{goWest, waitWest, goWest, waitWest, **goWalk**, waitWest, **goWalk**, waitWest }}, this is im persuming green on west ... and if a switch is pressed you go to waitWest which is yellow light on West but you have two switches which pressed would bring you too **Green on Walk (goWalk)** without having gone through waitWest so would go like green light on west red on walk then red light on west green on walk and skipped the yellow on west ...

same here {0x85,50,{goSouth, goSouth, waitSouth, waitSouth, **goWalk, goWalk**, waitSouth, waitSouth }},
persuming this is green light on South switch then pressed should go to waitSouth (yellow on South) but you have two switches which would bring straight to green on walk without going through the waitSouth (yellow light on South)

it gets harder to understand then after that would need to know what code matches what state

eg {0x89,50, ... what state is this for ... if you could just list out the state code and then beside it write the name of it i could go through it better
 eg like this .... {0x31,50 .. is for goWest do that for all nine so i can be sure which is for which

keep posting ill keep trying to help and im sure others will too ... late here so wont be on till some time tomorrow but will star this so i can keep checking back ... if i get the chance ill try and run your code through my lab10 just going to take me a while to change mine to yours and run it.