question | Like **0** views

moving on from lab 10 for now unless anyone has any more recommendations on what is wrong?

Hi,

Does anyone have any final recommendations on lab 10? I re-attempted to debug it yesterday but did not find my error. I will paste my code below. I know most people are already done with this class and I was one of those grateful for the extension until May 21st! I'm moving on now to finish up labs 11-14, and lab 15 too, although I know there is no grader for that I can attempt it even after May 21st if I find the time. I was able to catch up and finish every lecture video and quiz, so now I will just do the best I can to finish up the rest of your labs before May 21st. This has been a great review of somewhat similar things I've done before in past lives, and I thought it was well done and was unaware the MOOCs did things like this, since of course when I did these types of courses it was in person.

Anyway, here is a cut and paste of my fsm code, in case anyone has any recommendations on what I've done wrong. Would adding more states help here? I did also watch the extra lecture posted yesterday to help with those having trouble with lab 10. I'm moving on to finish up the rest of the labs now before this online MOOC ends on May 21st! Thanks. Here is my lab 10 code - any help / recommendations appreciated! It passed tests 1-3, but on test 4 it said the south state did not finish all tests before time was up. There was not enough information given in the test window to give a clue as to what was wrong.

```
struct State {
  unsigned long Out;
  unsigned long Time;
  unsigned long Next[8];};

typedef const struct State STyp;

#define goWest   0
#define waitWest 1
#define goSouth 2
#define waitSouth 3
#define goWalk   4
#define dontWalk1 5
#define walkOff1  6
#define dontWalk2  7
#define walkOff2  8

STyp FSM[9]={
 {0x31,50,{goWest, waitWest, goWest, waitWest, waitWest, waitWest, waitWest, waitWest }}, //0
 {0x51,50,{goSouth, goSouth, goSouth, goSouth, goWalk, goSouth, goWalk, goSouth }}, //1
 {0x85,50,{goSouth, goSouth, waitSouth, waitSouth, waitSouth, waitSouth, waitSouth, waitSouth }}, //2
 {0x89,50,{goWest, goWalk, goWest, goWest, goWalk, goWalk, goWalk, goWalk }}, //3
 {0x92,50,{goWest, dontWalk1, dontWalk1, dontWalk1, goWalk, dontWalk1, dontWalk1, dontWalk1 }}, //4
 {0x91,50,{walkOff1, walkOff1, walkOff1, walkOff1, walkOff1, walkOff1, walkOff1, walkOff1  }}, //5
 {0x90,50,{dontWalk2, dontWalk2, dontWalk2, dontWalk2, dontWalk2, dontWalk2, dontWalk2, dontWalk2 }}, //6
 {0x91,50,{walkOff2, walkOff2, walkOff2, walkOff2, walkOff2, walkOff2, walkOff2, walkOff2 }}, //7
 {0x90,50,{goWest, goWest, goWest, goWest, goWest, goWest, goWest, goWest }}}; //8

unsigned long S;  // index to the current state
unsigned long Input;

int main(void){
   volatile unsigned long delay;

  TExaS_Init(SW_PIN_PE210, LED_PIN_PB543210); // activate grader and set system clock to 80 MHz
  PLL_Init();       // 80 MHz, Program 10.1
  SysTick_Init();   // Program 10.2
  SYSCTL_RCGC2_R |= 0x32;      // 1) F B E
  delay = SYSCTL_RCGC2_R;      // 2) no need to unlock
  GPIO_PORTE_AMSEL_R &= ~0x07; // 3) disable analog function on PE2-0
  GPIO_PORTE_PCTL_R &= ~0x000000FF; // 4) enable regular GPIO
  GPIO_PORTE_DIR_R &= ~0x07;   // 5) inputs on PE2-0
  GPIO_PORTE_AFSEL_R &= ~0x07; // 6) regular function on PE2-0
  GPIO_PORTE_DEN_R |= 0x07;    // 7) enable digital on PE2-0
  GPIO_PORTB_AMSEL_R &= ~0x3F; // 3) disable analog function on PB5-0
  GPIO_PORTB_PCTL_R &= ~0x00FFFFFF; // 4) enable regular GPIO
  GPIO_PORTB_DIR_R |= 0x3F;    // 5) outputs on PB5-0
  GPIO_PORTB_AFSEL_R &= ~0x3F; // 6) regular function on PB5-0
  GPIO_PORTB_DEN_R |= 0x3F;    // 7) enable digital on PB5-0

  GPIO_PORTF_LOCK_R = 0x4C4F434B;   // 8) unlock GPIO Port F
  GPIO_PORTF_CR_R = 0x0A;           // allow changes to PF3 and PF1
```

```
  // only PF0 needs to be unlocked, other bits can't be locked
  GPIO_PORTF_AMSEL_R &= ~0x0A;            // 9) disable analog function on PF3 abd PF1
  GPIO_PORTF_PCTL_R = 0x00000000;   // 10) PCTL GPIO on PF4-0
  GPIO_PORTF_DIR_R |= 0x0A;         // 11) PF3 and PF1 outputs
  GPIO_PORTF_AFSEL_R &= ~0x0A;            // 12) regular function on PF3 and PF1
  //GPIO_PORTF_PUR_R = 0x11;          // enable pull-up on PF0 and PF4
  GPIO_PORTF_DEN_R |= 0x0A;               // 13) enable digital on PF3 and PF1

  S = goWest; //initial state
  EnableInterrupts();
  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];
  }

}

void SysTick_Init(void){
  NVIC_ST_CTRL_R = 0;               // disable SysTick during setup
  NVIC_ST_CTRL_R = 0x00000005;      // enable SysTick with core clock
}
// The delay parameter is in units of the 80 MHz core clock. (12.5 ns)
void SysTick_Wait(unsigned long delay){
  NVIC_ST_RELOAD_R = delay-1;  // number of counts to wait
  NVIC_ST_CURRENT_R = 0;       // any value written to CURRENT clears
  while((NVIC_ST_CTRL_R&0x00010000)==0){ // wait for count flag
  }
}
// 10000us equals 10ms
void SysTick_Wait10ms(unsigned long delay){
  unsigned long i;
  for(i=0; i<delay; i++){
    SysTick_Wait(800000);  // wait 10ms
  }
}
```

lab10

Just now 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*

Home

Click or ask a question to get started.