California State University

EGCP 450

Spring 2023

Lab 3

Professor Michael Turi

Zaid frayeh

Question1

```
#include <stdint.h>
#include "SysTick.h"
#include "msp432p401r.h"
struct State {
    uint32_t Out;
    uint32_t Outw;
    uint32_t Time; // 10 ms units
    const struct State *Next[20];
            };
typedef const struct State styp;
#define GOS     &FSM[0]
#define WAITS   &FSM[1]
#define GOW     &FSM[2]
#define WAITW   &FSM[3]
#define WALK    &FSM[4]
#define HON1    &FSM[5]
#define HOFF2   &FSM[6]
#define HON3    &FSM[7]
#define HOFF4   &FSM[8]
#define HON5    &FSM[9]
#define HOFF6   &FSM[10]
#define HON7    &FSM[11]
#define HOFF8   &FSM[12]
#define HON9    &FSM[13]
#define HON     &FSM[14]
#define HOFF    &FSM[15]
#define WAITWW  &FSM[16]
#define WAITWS  &FSM[17]
#define REQW    &FSM[18]
#define REQS    &FSM[19]

styp FSM[20] = {
 {0x21,0x01,100,{GOS,WAITS,GOS,WAITS,REQS,REQS,REQS,REQS}},  //1 gos
 {0x22,0x01,100,{GOW,GOW,GOW,GOW,REQS,REQS,REQS,REQS}},      //2 waitS
 {0x0C,0x01,100,{GOW,GOW,WAITW,WAITW,REQW,REQW,REQW,REQW}},  //3 goW
 {0x14,0x01,100,{GOS,GOS,GOS,GOS,REQW,REQW,REQW,REQW}},      //4 waitW
 {0x24,0x02,100,{WALK,HON1,HON1,HON1,WALK,HON1,HON1,HON1}},   //WALK
 {0x24,0x01,100,{HOFF2,HOFF2,HOFF2,HOFF2,HOFF2,HOFF2,HOFF2,HOFF2}},
//HURRYON1
 {0x24,0x00,100,{HON3,HON3,HON3,HON3,HON3,HON3,HON3,HON3}},
//HURRYOFF2
 {0x24,0x01,100,{HOFF4,HOFF4,HOFF4,HOFF4,HOFF4,HOFF4,HOFF4,HOFF4}},
//HURRYON3
 {0x24,0x00,100,{HON5,HON5,HON5,HON5,HON5,HON5,HON5,HON5}},
//HURRYOFF4
 {0x24,0x01,50,{HOFF6,HOFF6,HOFF6,HOFF6,HOFF6,HOFF6,HOFF6,HOFF6}},
//HURRYON5
 {0x24,0x00,50,{HON7,HON7,HON7,HON7,HON7,HON7,HON7,HON7}},
//HURRYOFF6
```

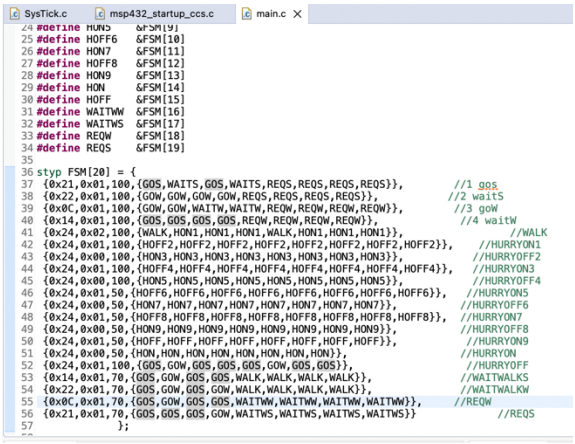
```
 {0x24,0x01,50,{HOFF8,HOFF8,HOFF8,HOFF8,HOFF8,HOFF8,HOFF8,HOFF8}},
//HURRYON7
 {0x24,0x00,50,{HON9,HON9,HON9,HON9,HON9,HON9,HON9,HON9}},   //HURRYOFF8
 {0x24,0x01,50,{HOFF,HOFF,HOFF,HOFF,HOFF,HOFF,HOFF,HOFF}},    //HURRYON9
 {0x24,0x00,50,{HON,HON,HON,HON,HON,HON,HON,HON}},            //HURRYON
 {0x24,0x01,100,{GOS,GOW,GOS,GOS,GOS,GOW,GOS,GOS}},          //HURRYOFF
 {0x14,0x01,70,{GOS,GOW,GOS,GOS,WALK,WALK,WALK,WALK}},      //WAITWALKS
 {0x22,0x01,70,{GOS,GOW,GOS,GOW,WALK,WALK,WALK,WALK}},     //WAITWALKW
 {0x0C,0x01,70,{GOS,GOW,GOS,GOS,WAITWW,WAITWW,WAITWW,WAITWW}},    //REQW
 {0x21,0x01,70,{GOS,GOS,GOS,GOW,WAITWS,WAITWS,WAITWS,WAITWS}}     //REQS
            };

void ports(void){

   //output pins
        P4SEL0 &= ~0x3F;
        P4SEL1 &= ~0x3F;
        P4DIR  |= 0x3F;
   // output led
        P2SEL0 &= ~0x03;
        P2SEL1 &= ~0x03;
        P2DS   |= 0x03;
        P2DIR  |= 0x03;
        P2OUT  |= 0x03;
   //input buttons
        P1SEL0 = 0x00;
        P1SEL1 = 0x00;
        P1DS   = 0x00;
        P1DIR  = 0x00;
        P1REN  = 0X02;
        P1OUT  = 0x02;
   //input buttons
        P5SEL0 &= ~0x06;
        P5SEL1 &= ~0x06;
        P5DIR  &= ~0x06;
}
void main(void){
    uint32_t Input;
    uint32_t Input2;
    styp *Pt;
    SysTick_Init();
    ports();
    Pt = GOS; // start state
while(1){
            P4OUT=(Pt -> Out);
            P2OUT=((Pt -> Outw)&0x03);
            SysTick_Wait10ms(Pt->Time);
             Input = P5IN&06;
            Input = Input/2;
            Input2 = ((~P1IN&0x02) <<1 );
            Pt = Pt->Next[Input2 + Input];
        }
             }
```

Question 2

In my Finite State Machine design there are 20 states there are 4 main states for the two traffic lights which I started my first program with after that I added 2 requests for the states if a pedestrian wants to request to walk and after that another 2 states which waits for the button to still be pressed which provides the 2 second delay for the press, the other states are basically walk for the pedestrians as well as hurry for the pedestrians which turns on and off the light and speeds up at the end.
This makes up a design of 20 states, In my design I used pointers and a linked data structure to achieve my finite state machine.

Question 3

For a given state there are always 4 possible transitions in the first part and 8 possible transitions in the second part, this part is determined by the number of input buttons we have which is 2^number of buttons we have. We have a south traffic light and west traffic light and one for pedestrians



This part shows how I defined the next state arrow transitions as we can see there is 20 states and 8 arrow transitions

Question 4

Only sites that I have used are the websites provided in the file,
As well as some student help from David Mouser and Moniel flores. They helped me understand the way to do the code and how to approach it.

Screenshots that have helped me build my design for both parts :

South Traffic Lights

P 4.0 G
P 4.1 y
P 4.2 R
] output

P 5.2 Button Input

40 - 60

Input	W	S	out
	0	0	X
	0	1	W
	1	0	N
	1	1	T

Bits	7 6 5 4 3 2 1 0
Go S	1 0 0 0 0 1
Wait S	1 0 0 0 1 0
Go W	0 0 1 1 0 0
WaitW	0 1 0 1 0 0

yellow S
Red W

1 4

22

West Traffic Lights

P.4.3 G
P.4.4 y
P.4.5 R
] output

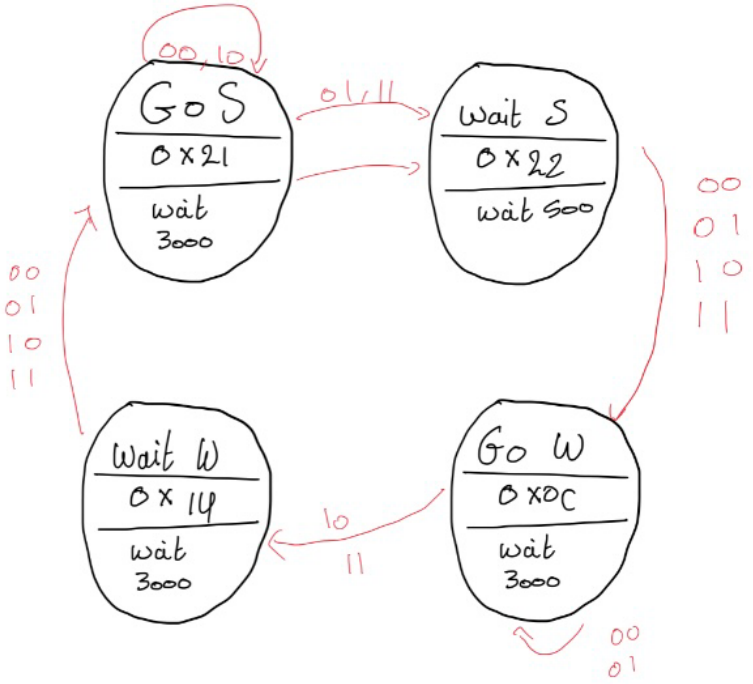
P.5.1 Button Input

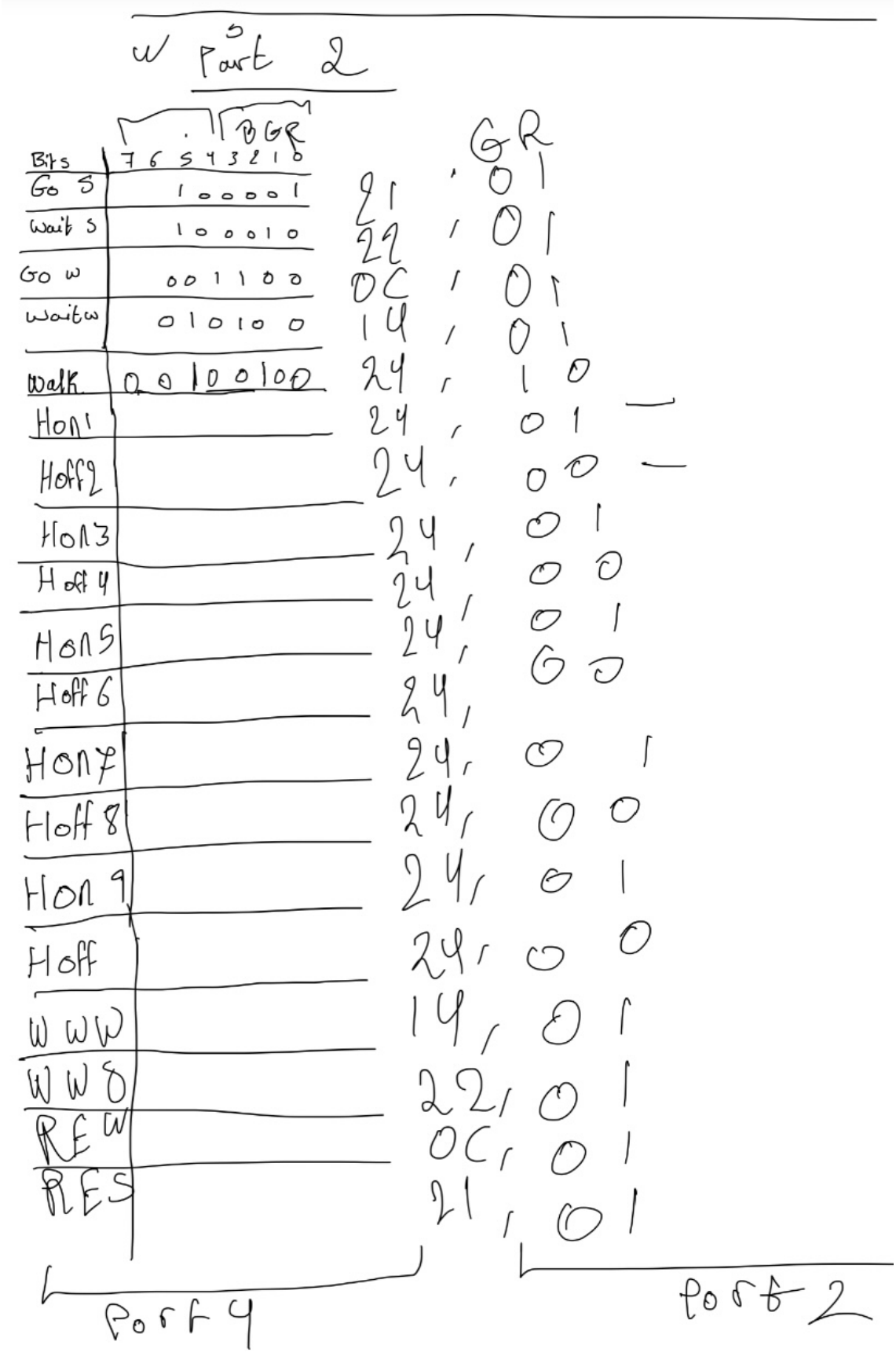
1 - 30

Finite State Machine

Input	No cars 00	west 01	South 10	Both 11
Go S	Go S	wait S	Go S	wait S
wait S	Go W	Go W	Go W	G W
Go W	Go W	Go W	Wait W	waitW
wait W	GO S	Go S	Go S	GoS

Table




W Port 2
BGR
GR
Bits 7 6 5 4 3 2 1 0
Go S 1 0 0 0 0 1 21, 01
Wait S 1 0 0 0 1 0 22, 01
Go W 0 0 1 1 0 0 0C, 01
Wait W 0 1 0 1 0 0 14, 01
Walk 0 0 1 0 0 1 0 0 24, 10
Hon1 24, 01
Hoff2 24, 00
Hon3 24, 01
Hoff 4 24, 00
Hon5 24, 01
Hoff 6 24, 00
Hon7 24, 01
Hoff 8 24, 00
Hon 9 24, 01
Hoff 24, 00
W WW 14, 01
WW 8 22, 01
REW 0C, 01
RES 21, 01
Port 4
Port 2

Finite State Machine

	Input	No cars walk W 000	west button walk S W 001	south button walk SW 010	go south & west walk SW 011	west walk walk SW 100	south walk walk S W 101	south & west walk walk S W 110	walk S W 111
0	Go S	goS	waitS	goS	ReqS	ReqS	ReqS	ReqS	ReqS
1	wait S	goW	goW	goW	goW	ReqW	ReqW	ReqW	ReqW
2	Go W	goW	goW	waitW	waitW	ReqW	ReqW	ReqS	ReqS
3	wait W	goS	goS	goS	goS	ReqS	ReqS	ReqS	ReqS
4	Walk	walk	hurryon1	hurryon1	hurryon1	walk	hurryon1	hurryon1	hurryon1
5	hurryon1	hurryoff2	hurryoff2	hurryoff2	hurryoff2	hurryoff2	hurryoff2	hurryoff2	hurryoff2
6	hurryoff2	on3							
7	hurryon3	off4							
8	hurryoff4	on5							
9	hurryon5	off6							
10	hurryoff6	on7							
11	hurryon7	off8							
12	hurryoff8	on9							
13	hurryon9	on							
	hurryon	off	off	off					
	hurryoff	goS	goW	goS	goS	goS	goW	goS	goS
	waitwalkS	goS	goW	goS	goS	walk	walk	walk	walk
	waitwalkW	goS	goW	goS	goW	walk	walk	walk	walk
	ReqW	goS	goW	goS	goS	waitW	waitW	waitW	waitW
	ReqS	goS	goS	goS	goW	waitS	waitS	waitwalkS	waitWS