Lab 5 Lecture

Friday, February 26, 2021 9:13 AM

Lab 5 - Traffic Light Controller

★ Written in C and uses real hardware ★

Lecture Plan

0) Demo
1) Systick
2) FSM
3) Structs in C
4) FSM in C
5) Hardware

1) Systick Timer

What is it? System Tick Timer

- 24 bit Down Counter: 0x00ffffff → 0x00fffffE → ... 0x00000000
- Decrements @ Bus Clock frequency
  - 80 MHz ⟷ 12.5 ns

Initialization

Address	31-24	23-17	16	15-3	2	1	0	Name
$E000E010	0	0	COUNT	0	CLK_SRC	INTEN	ENABLE	NVIC_ST_CTRL_R
$E000E014	0	24-bit RELOAD value						NVIC_ST_RELOAD_R
$E000E018	0	24-bit CURRENT value of SysTick counter						NVIC_ST_CURRENT_R

- ❑ Initialization (4 steps)
  - ❖ Step1: Clear ENABLE to stop counter
  - ❖ Step2: Specify the RELOAD value
  - ❖ Step3: Clear the counter via NVIC_ST_CURRENT_R
  - ❖ Step4: Set NVIC_ST_CTRL_R
    - o CLK_SRC = 1 (bus clock is the only option)
    - o INTEN = 0 for no interrupts
    - o ENABLE = 1 to enable

Measure Elapsed Time

★ Need 32 bits to store value ★

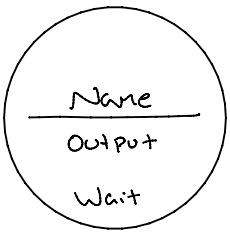
```
Uint32_t Start = NVIC_ST_CURRENT_R;
Uint32_t elapsed = 0;
// Do some fancy coding because we've spent (takes time)
elapsed = (Start - NVIC_ST_CURRENT_R) & 0x00FFFFFF;
```

Why this way? What does this do?

- elapsed = n ⟶ (12.5 n) ns
- Delay: loop until enough elapsed time has passed (One possible method)

(2) FSM

★ Moore FSM ★



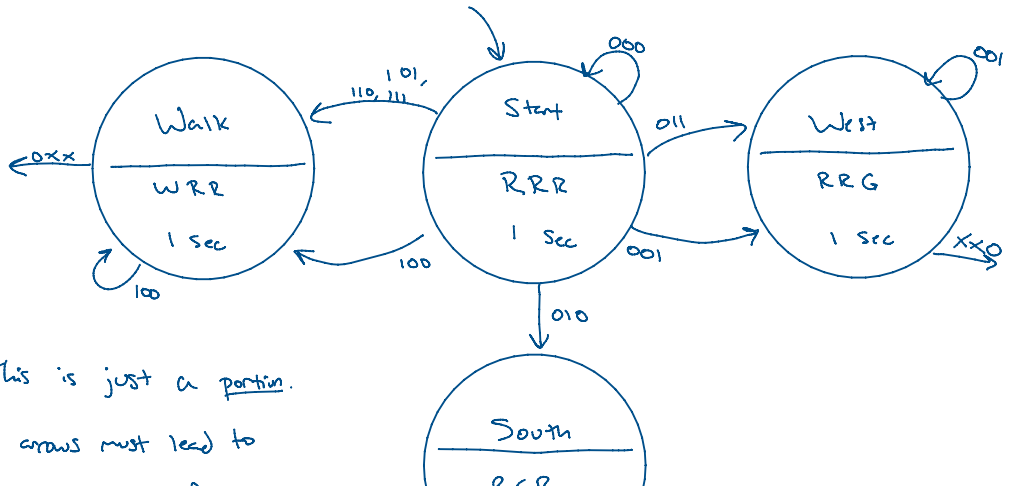
How Many Inputs? 3 switches ⟶ $2^3 = 8$ possibilities ⟶ Arrows

How Many Outputs? 3 sets of lights

* not all are valid and some will be used by multiple states

Inputs: Walk (2) South (1) West (0)

Outputs: Walk (2) South (1) West (0) * not binary

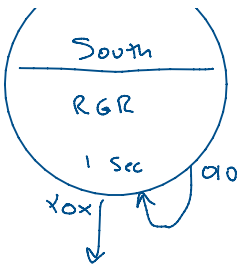


★★ This is just a portion.

All arrows must lead to

All arrows must lead to

Some state in a full STG ✱✱



(3) Structs in C

- physically groups a list of variables
- One block in Memory

```
Struct Point {
    uint8_t x;
    uint8_t y;
};

Struct Line {
    Struct Point p1;
    Struct Point p2;
};

// Declare a line
Struct Point point1;
point1.x = 0;
point1.y = 0;
Struct Point point2;
point2.x = 1;
point2.y = 1;
Struct Line line;
line.p1 = point1;
line.p2 = point2;
```

(4) FSM in C

```
Struct State {
    unsigned long out;
```

Array of the # of the next State indexed by input

```
struct State {
  unsigned long out;
  unsigned long wait;
  uint8_t next[8];
};
```

Array of the # of the next state indexed by input

e.g. Start = 0
West = 1
South = 2
Walk = 3

```
struct State FSM[4] = {
  // State Definitions
};
```

Then FSM[Start].next[1] = West;

```
uint8_t Curr = Start;
while(1) {
  GPIO_PORTA_DATA_R = FSM[Curr].out;  // output
  in = GPIO_PORTB_DATA_R;
  next = FSM[Curr].next[in];
  // Delay w/ Systick
  Curr = next;  // transition
```

☆ Not exact code ☆

(5) Hardware

- 6 LEDs
  - 2 R, 2 Y, 2 G
- 6 470 Ω resistors
  - for LEDs
- 3 switches
- 3 10 kΩ resistors
  - Pull-down for switches
- Onboard LEDs (PF1:3)
  - RGB (all 3 on → White)