Lab Lecture 1

Friday, January 29, 2021 4:45 AM

1. Lab Procedures
2. Lab Requirements
3. Initialization
4. Negative vs. Positive Logic
5. EOR
6. Algorithm
7. AREA
8. Keil Basics

1. Lab Procedures

- Labs are done by checkout time you signed for in lab
  - ↳ sunk in canvas
- Graded based on:
  - Deliverables — submit on canvas
  - performance
  - coding standard
  - Demonstration — Questions Asked During checkout
    - ↳ ask in OH if not sure

USE PIAZZA AND START LOOKING FOR LAB PARTNERS

TA OH POSTED ON SYLLABUS

2. Lab 1 Requirements

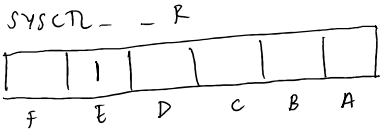
- NO CONDITIONAL BRANCHES, only logical operations: AND, ORR, EOR, LSL/LSR
- Inputs (switches): PE0 - PE2 (negative logic)
  Output (LED): PE4 (positive logic)
- GOAL: even # of switches pressed → LED output 1
  (even # of 0s in inputs)

3. Initialization

A. Turn on the clocks (Port E)

```
LDR  R0, = SYSCTL_RCGCGPIO_R
LDRB R1, [R0]
ORR  R1, #0x10
STRB R1, [R0]
```

read, modify, write



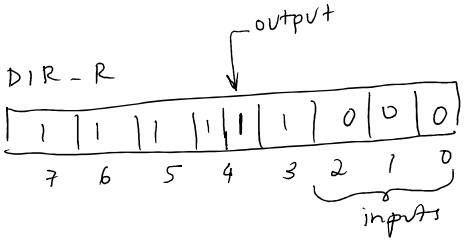
B. Wait for clock to stabilize

```
NOP
NOP
```

C. Define inputs and outputs (DIR)

```
LDR  R0, = GPIO_PORTE_DIR_R
LDRB R1, [R0]
AND  R1, #0xF8
ORR  R1, #0x10
STRB R1, [R0]
```

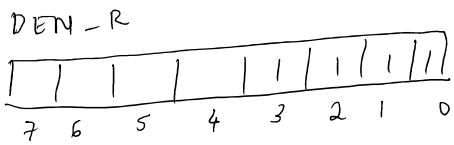
} friendliness: modifying only bits you need to



inputs - 0
outputs - 1

D. Digitally enable pins (DEN)

```
LDR  R0, = GPIO_PORTE_DEN_R
LDRB R1, [R0]
ORR  R1, #0x1F
STRB R1, [R0]
```

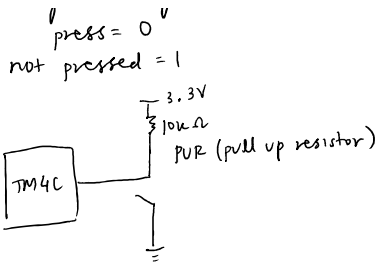


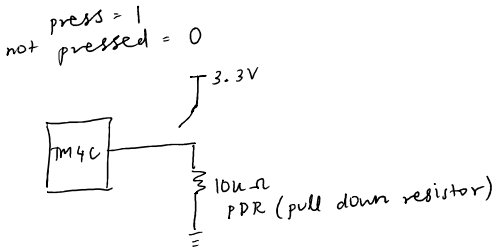
4. Negative vs. Positive Logic

- does not affect initialization

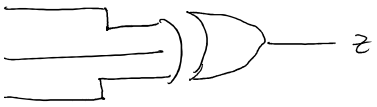
Negative logic
press = 0
not pressed = 1
— 3.3V

Positive logic
press = 1
not pressed = 0
T 3.3V





(5) EOR (exclusive OR)

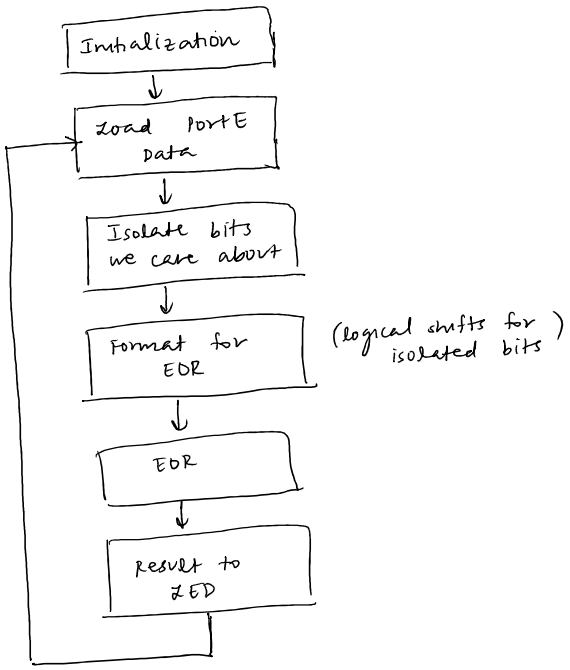


EOR	
0 ^ 0	0
0 ^ 1	1
1 ^ 0	1
1 ^ 1	0

A	B	C	z	
0	0	0	0	
0	0	1	1	✓ even # pressed
0	1	0	1	✓ even # pressed
0	1	1	0	
1	0	0	1	✓ even # pressed
1	0	1	0	
1	1	0	0	
1	1	1	1	✓ even # pressed

} expected output of Lab 1

(6) Algorithm



(7) AREA

```
AREA DATA ; objects in RAM
AREA CODE ; objects in ROM
```

(8) Keil Basics