Judah Ben-Eliezer

112352727
3/5/2021

Lab 4:

Identification Feature for Quadruple Two-Input Logic Gate IC Test System

Questions:

1. The maximum V IHmin for the input to the IC's is 3.5 V. The AVR128DB48 drives a minimum of 4.5 volts out of the GPIO pins. This is sufficient for powering each IC.
2. Yes, my system correctly distinguishes between the 74HC00 and the 74LS03. It does this by first testing for nand logic without internal pullups enabled for the DUT outputs. If a 74LS03 is being tested, this will cause it to fail, but for the 74HC00, it will pass. After testing for the other types of IC's, my system enables pullups and again tests for nand logic. This will pass for the 74LS03. It then disables the internal pullups for the DUT outputs. This method requires no additional external components.

```c
/*
 * ic_test_ident.c
 *
 * Created: 3/4/2021 6:59:47 PM
 * Author : Judah Ben-Eliezer
 */

#define  F_CPU 4000000UL

#include <avr/io.h>
#include <util/delay.h>

#define IN_COMB_00 0x00
#define IN_COMB_01 0x55
#define IN_COMB_10 0xAA
#define IN_COMB_11 0xFF
#define DIP_SW_gm 0xE0
#define START_PB_bm 0x08
#define BARGRAPH_gm 0x07
#define TIP_bm 0x04
#define PASS_bm 0x01
#define FAIL_bm 0x02
#define A_GATES_OUT_gm 0x1F
#define B_GATES_OUT_gm 0x30
#define PA_setup_gm 0x07
#define PB_setup_gm 0xCF
#define PC_setup_gm 0xFF
#define PD_setup_gm 0x07
#define PE_setup_gm 0x0F
#define PF_setup_gm 0xF7

const uint8_t stimulus[] = {IN_COMB_00, IN_COMB_01, IN_COMB_10, IN_COMB_11};

const uint8_t verify[5][4] = {{0x0F, 0x0F, 0x0F, 0x00},
                              {0x00, 0x00, 0x00, 0x0F},
                              {0x00, 0x0F, 0x0F, 0x0F},
                              {0x00, 0x0F, 0x0F, 0x00},
                              {0x0F, 0x0F, 0x0F, 0x00}};

uint8_t gate_type;

uint8_t i, j;

uint8_t test(uint8_t gate) {
    //turn DUT pin 14 on
    PORTE_OUT |= PIN3_bm;

    for (i = 0; i < 4; ++i) {
        PORTC_OUT = stimulus[i];
```

```
50
51          _delay_loop_1(2);
52
53          uint8_t a_in = (PORTA_IN & A_GATES_OUT_gm) >> 3 & 0x03;
54          uint8_t b_in = (PORTB_IN & B_GATES_OUT_gm) >> 2 & 0x0C;
55
56          if (!((a_in | b_in) == verify[gate][i])) break;
57
58      }
59
60      if (i == 4) return 1;
61      else return 0;
62
63
64  }
65
66  uint8_t identify() {
67
68      for (j = 0; j < 4; ++j) {
69          if (test(j) == 1) return j;
70      }
71      //enable pullups
72      PORTA_PIN4CTRL |= PORT_PULLUPEN_bm;
73      PORTA_PIN3CTRL |= PORT_PULLUPEN_bm;
74      PORTB_PIN5CTRL |= PORT_PULLUPEN_bm;
75      PORTB_PIN4CTRL |= PORT_PULLUPEN_bm;
76      if (test(4) == 1)  {
77          PORTA_PIN4CTRL &= ~PORT_PULLUPEN_bm;
78          PORTA_PIN3CTRL &= ~PORT_PULLUPEN_bm;
79          PORTB_PIN5CTRL &= ~PORT_PULLUPEN_bm;
80          PORTB_PIN4CTRL &= ~PORT_PULLUPEN_bm;
81          return 4;
82      }
83
84
85      return 7;
86  }
87
88  int main(void)
89  {
90      PORTA_DIRSET &= PA_setup_gm;
91      PORTB_DIRSET &= PB_setup_gm;
92      PORTC_DIRSET |= PC_setup_gm;
93      PORTD_DIRSET |= PD_setup_gm;
94      PORTE_DIRSET |= PE_setup_gm;
95      PORTF_DIRSET &= PF_setup_gm;
96
97      PORTA_PIN7CTRL |= PORT_PULLUPEN_bm;
98      PORTA_PIN6CTRL |= PORT_PULLUPEN_bm;
```

```
 99      PORTA_PIN5CTRL |= PORT_PULLUPEN_bm;
100      PORTA_PIN4CTRL |= PORT_PULLUPEN_bm;
101      PORTA_PIN3CTRL |= PORT_PULLUPEN_bm;
102      PORTF_PIN3CTRL |= PORT_PULLUPEN_bm;
103
104      PORTD_OUT = ~BARGRAPH_gm;
105      PORTE_OUT &= ~(TIP_bm | PASS_bm | FAIL_bm);
106      _delay_ms(1000);
107      PORTD_OUT |= BARGRAPH_gm;
108      PORTE_OUT |= TIP_bm | PASS_bm | FAIL_bm;
109
110      while (1)
111      {
112          if ((PORTF_IN & START_PB_bm) != 0) continue;
113
114          PORTD_OUT = BARGRAPH_gm;
115
116          PORTE_OUT = ~TIP_bm;
117
118          gate_type = ~(PORTA_IN | 0x1F) >> 5 & 0x07;
119
120          PORTD_OUT &= ~(gate_type & BARGRAPH_gm);
121
122          if (gate_type == 0x04) {
123              //enable pullups
124              PORTA_PIN4CTRL |= PORT_PULLUPEN_bm;
125              PORTA_PIN3CTRL |= PORT_PULLUPEN_bm;
126              PORTB_PIN5CTRL |= PORT_PULLUPEN_bm;
127              PORTB_PIN4CTRL |= PORT_PULLUPEN_bm;
128              if (test(gate_type) == 1) PORTE_OUT &= ~PASS_bm;
129              else PORTE_OUT &= ~FAIL_bm;
130              PORTA_PIN4CTRL &= ~PORT_PULLUPEN_bm;
131              PORTA_PIN3CTRL &= ~PORT_PULLUPEN_bm;
132              PORTB_PIN5CTRL &= ~PORT_PULLUPEN_bm;
133              PORTB_PIN4CTRL &= ~PORT_PULLUPEN_bm;
134          } else if (gate_type == 0x07) {
135              gate_type = identify();
136          } else {
137              if (test(gate_type) == 1) PORTE_OUT &= ~PASS_bm;
138              else PORTE_OUT &= ~FAIL_bm;
139          }
140
141          PORTC_OUT = stimulus[0];
142
143          PORTE_OUT |= TIP_bm;
144
145          PORTD_OUT = ~(gate_type & BARGRAPH_gm);
146
147          //turn DUT pin 14 off
```

```
148             PORTE_OUT &= ~PIN3_bm;
149         }
150
151         return 0;
152     }
153
154
```