```
;
; ic_test.asm
;
; Created: 10/3/2022 11:38:03 AM
; Author : Jason Chen
;

start:
    ldi r16, 0xFF       ; load r16 with all 1s
    out VPORTD_DIR, r16 ; VPORTD - all pins configured as outputs
    ldi r16, 0x0F       ; load r16 with 0000 1111
    out VPORTA_DIR, r16 ; VPORTA - pins 0-3 as outputs, 4-7 as inputs
    ldi r16, 0x00       ; load r16 with all 0s
    out VPORTC_DIR, r16 ; VPORTC - all pins configured as inputs
    cbi VPORTE_DIR, 0   ; set direction for PE0 as input
    sbi VPORTE_DIR, 1   ; set direction for PE1 as output
    cbi VPORTE_DIR, 2   ; set direction for PE2 as input

post:
    ldi r16, 0xC0
    out VPORTD_OUT, r16 ; turn on all working LEDs
    rjmp one_sec_delay
    ldi r16, 0xFF
    out VPORTD_OUT, r16 ; turn all LEDs OFF

again:
    sbi VPORTE_OUT, 1   ; set PE1 to 1 to "unclear" the DFF
    sbic VPORTE_IN, 0   ; skip if PE0 is 0
    rjmp again

; Wait for the pushbutton to send clock signal to DFF and output to PE00
wait_for_push:
    sbis VPORTE_IN, 0   ; skip if PE0 is 1
    rjmp wait_for_push
    ldi r16, 0xDF       ; load r16 with 1101 1111
    out VPORTD_OUT, r16 ; white LED ON, all other LEDs OFF

test_type:
    in r16, VPORTC_IN   ; load switch positions to r16
    andi r16, 0x07      ; mask for relevant info
    cpi r16, 0x00       ; is it NAND / 74HC00
    breq test_nand
    cpi r16, 0x01       ; is it AND / 74HC08
    breq long_jump_and
    cpi r16, 0x02       ; is it OR / 74HC32
    breq long_jump_or
    cpi r16, 0x03       ; is it XOR / 74HC86
    breq long_jump_xor

test_ls_nand:           ; test 74LS03, default
    ldi r18, 0x08       ; load r18 with 0000 1000
                        ; enable pull-up resistors
    sts PORTA_PIN4CTRL, r18
    sts PORTA_PIN5CTRL, r18
    sts PORTA_PIN6CTRL, r18
    sts PORTA_PIN7CTRL, r18

    ldi r17, 0x00       ; load r17 with all 0s
    out VPORTA_OUT, r17 ; send inputs AB = 00 to device
    nop
    nop
    in r17, VPORTA_IN   ; read VPORTA
    andi r17, 0xF0      ; mask for PA4 - PA7
    cpi r17, 0xF0       ; check if device outputs 1s
    brne test_fail_jump

    ldi r17, 0x04       ; load r17 with 0000 0100
    out VPORTA_OUT, r17 ; send inputs AB = 01 to device
    nop
    nop
    in r17, VPORTA_IN   ; read VPORTA
    andi r17, 0xF0      ; mask for PA4 - PA7
    cpi r17, 0xF0       ; check if device outputs 1s
    brne test_fail_jump

    ldi r17, 0x08       ; load r17 with 0000 1000
    out VPORTA_OUT, r17 ; send inputs AB = 10 to device
    nop
    nop
    in r17, VPORTA_IN   ; read VPORTA
    andi r17, 0xF0      ; mask for PA4 - PA7
    cpi r17, 0xF0       ; check if device outputs 1s
    brne test_fail_jump

    ldi r17, 0x0C       ; load r17 with 0000 1100
    out VPORTA_OUT, r17 ; send inputs AB = 11 to device
    nop
    nop
    in r17, VPORTA_IN   ; read VPORTA
    andi r17, 0xF0      ; mask for PA4 - PA7
    cpi r17, 0x00       ; check if device outputs 0s
    brne test_fail_jump

    rjmp test_pass

long_jump_and:
    rjmp test_and

long_jump_or:
    rjmp test_or

long_jump_xor:
    rjmp test_xor

```

```
test_nand:              ; test 74HC00
    ldi r17, 0x00       ; load r17 with all 0s
    out VPORTA_OUT, r17 ; send inputs AB = 00 to device
    nop
    nop
    in r17, VPORTA_IN   ; read VPORTA
    andi r17, 0xF0      ; mask for PA4 - PA7
    cpi r17, 0xF0       ; check if device outputs 1s
    brne test_fail_jump

    ldi r17, 0x04       ; load r17 with 0000 0100
    out VPORTA_OUT, r17 ; send inputs AB = 01 to device
    nop
    nop
    in r17, VPORTA_IN   ; read VPORTA
    andi r17, 0xF0      ; mask for PA4 - PA7
    cpi r17, 0xF0       ; check if device outputs 1s
    brne test_fail_jump

    ldi r17, 0x08       ; load r17 with 0000 1000
    out VPORTA_OUT, r17 ; send inputs AB = 10 to device
    nop
    nop
    in r17, VPORTA_IN   ; read VPORTA
    andi r17, 0xF0      ; mask for PA4 - PA7
    cpi r17, 0xF0       ; check if device outputs 1s
    brne test_fail_jump

    ldi r17, 0x0C       ; load r17 with 0000 1100
    out VPORTA_OUT, r17 ; send inputs AB = 11 to device
    nop
    nop
    in r17, VPORTA_IN   ; read VPORTA
    andi r17, 0xF0      ; mask for PA4 - PA7
    cpi r17, 0x00       ; check if device outputs 0s
    brne test_fail_jump

    rjmp test_pass

test_fail_jump:
    rjmp test_fail

test_and:
    ldi r17, 0x00       ; load r17 with all 0s
    out VPORTA_OUT, r17 ; send inputs AB = 00 to device
    nop
    nop
    in r17, VPORTA_IN   ; read VPORTA
    andi r17, 0xF0      ; mask for PA4 - PA7
    cpi r17, 0x00       ; check if device outputs 0s
    brne test_fail_jump

    ldi r17, 0x04       ; load r17 with 0000 0100
    out VPORTA_OUT, r17 ; send inputs AB = 01 to device
    nop
    nop
    in r17, VPORTA_IN   ; read VPORTA
    andi r17, 0xF0      ; mask for PA4 - PA7
    cpi r17, 0x00       ; check if device outputs 0s
    brne test_fail_jump

    ldi r17, 0x08       ; load r17 with 0000 1000
    out VPORTA_OUT, r17 ; send inputs AB = 10 to device
    nop
    nop
    in r17, VPORTA_IN   ; read VPORTA
    andi r17, 0xF0      ; mask for PA4 - PA7
    cpi r17, 0x00       ; check if device outputs 0s
    brne test_fail_jump

    ldi r17, 0x0C       ; load r17 with 0000 1100
    out VPORTA_OUT, r17 ; send inputs AB = 11 to device
    nop
    nop
    in r17, VPORTA_IN   ; read VPORTA
    andi r17, 0xF0      ; mask for PA4 - PA7
    cpi r17, 0xF0       ; check if device outputs 1s
    brne test_fail_jump

    rjmp test_pass

test_or:
    ldi r17, 0x00       ; load r17 with all 0s
    out VPORTA_OUT, r17 ; send inputs AB = 00 to device
    nop
    nop
    in r17, VPORTA_IN   ; read VPORTA
    andi r17, 0xF0      ; mask for PA4 - PA7
    cpi r17, 0x00       ; check if device outputs 0s
    brne test_fail

    ldi r17, 0x04       ; load r17 with 0000 0100
    out VPORTA_OUT, r17 ; send inputs AB = 01 to device
    nop
    nop
    in r17, VPORTA_IN   ; read VPORTA
    andi r17, 0xF0      ; mask for PA4 - PA7
    cpi r17, 0xF0       ; check if device outputs 1s
    brne test_fail

    ldi r17, 0x08       ; load r17 with 0000 1000
    out VPORTA_OUT, r17 ; send inputs AB = 10 to device
    nop
    nop
    in r17, VPORTA_IN   ; read VPORTA
    andi r17, 0xF0      ; mask for PA4 - PA7
    cpi r17, 0xF0       ; check if device outputs 1s
    brne test_fail

    ldi r17, 0x0C       ; load r17 with 0000 1100
    out VPORTA_OUT, r17 ; send inputs AB = 11 to device
    nop
    nop
    in r17, VPORTA_IN   ; read VPORTA
    andi r17, 0xF0      ; mask for PA4 - PA7
    cpi r17, 0xF0       ; check if device outputs 1s
    brne test_fail

    rjmp test_pass

test_xor:
    ldi r17, 0x00       ; load r17 with all 0s
    out VPORTA_OUT, r17 ; send inputs AB = 00 to device
    nop
    nop
    in r17, VPORTA_IN   ; read VPORTA
    andi r17, 0xF0      ; mask for PA4 - PA7
    cpi r17, 0x00       ; check if device outputs 0s
    brne test_fail

    ldi r17, 0x04       ; load r17 with 0000 0100
    out VPORTA_OUT, r17 ; send inputs AB = 01 to device
    nop
    nop
    in r17, VPORTA_IN   ; read VPORTA
    andi r17, 0xF0      ; mask for PA4 - PA7
    cpi r17, 0xF0       ; check if device outputs 1s
    brne test_fail

    ldi r17, 0x08       ; load r17 with 0000 1000
    out VPORTA_OUT, r17 ; send inputs AB = 10 to device
    nop
    nop
    in r17, VPORTA_IN   ; read VPORTA
    andi r17, 0xF0      ; mask for PA4 - PA7
    cpi r17, 0xF0       ; check if device outputs 1s
    brne test_fail

    ldi r17, 0x0C       ; load r17 with 0000 1100
    out VPORTA_OUT, r17 ; send inputs AB = 11 to device
    nop
    nop
    in r17, VPORTA_IN   ; read VPORTA
    andi r17, 0xF0      ; mask for PA4 - PA7
    cpi r17, 0x00       ; check if device outputs 0s
    brne test_fail

    rjmp test_pass

test_fail:
    ldi r16, 0xEF       ; mask for red LED
    out VPORTD_OUT, r16 ; red LED ON, all other LEDs OFF
    rjmp clear_and_reset

test_pass:
    ldi r18, 0x08
    or r16, r18         ; bitwise add r16 and 0000 1000 for green LED
    com r16
    out VPORTD_OUT, r16 ; white and red LEDs OFF, green LED ON and bargraph

clear_and_reset:
    ldi r18, 0x00       ; load r18 with all 0s
                        ; disable pull-up resistors
    sts PORTA_PIN4CTRL, r18
    sts PORTA_PIN5CTRL, r18
    sts PORTA_PIN6CTRL, r18
    sts PORTA_PIN7CTRL, r18
    cbi VPORTE_OUT, 1   ; clear the DFF

wait_for_release:       ; debounce release of pushbutton
    sbic VPORTE_IN, 2   ; skip if PE2 is 0
    rjmp wait_for_release
    rcall var_delay
    sbic VPORTE_IN, 2   ; skip if PE2 is 0
    rjmp wait_for_release
    rjmp again

; Delay r18 * 0.100475 ms
var_delay:
    ldi r18, 0xFF       ; for delay of ~25.6ms
    var_outer_loop:
        ldi r17, 133
    var_inner_loop:
        dec r17
        brne var_inner_loop
        dec r18
        brne var_outer_loop
    ret ; return to caller

; 1.00008575 seconds @ 4 MHz system clock, 192 us resolution
one_sec_delay:
    ldi r30, LOW(5202)  ;outer loop 16- bit iteration count
    ldi r31, HIGH(5202) ;16-bit value in r31:r30
    outer_loop:
        ldi r18, $FF        ;inner loop 8-bit iteration count
    inner_loop:
        dec r18             ;subtract 1 from inner loop count
        brne inner_loop
        sbiw r31:r30, 1     ;subtract 1 from outer loop count
        brne outer_loop
    ret
```