

Termin zajęć Wtorek NP 7:30 – 11:00	Układy cyfrowe i systemy wbudowane	
Osoby wykonujące ćwiczenie: Jakub Suski 264028, Adam Czekalski 264488		Grupa: D
Tytuł ćwiczenia: Detektor Typu Znak i Licznik Bitów w Spartanie		Laboratorium nr: 8
Data wykonania ćwiczenia	16.01.2024	Ocena:
Data oddania sprawozdania	19.01.2024	


```

76   if IN_I >= "01100001" and IN_I <= "01111010" and is_clicked = '0' then
77       --CHAR_TYPE <= "01000001";
78       CHAR_TYPE <= "00001010";
79       is_clicked <= '1';
80       -- count ones and zeros
81   elsif IN_I >= "01000001" and IN_I <= "01011010" and is_clicked = '0' then
82       --CHAR_TYPE <= "01000010";
83       CHAR_TYPE <= "00001011";
84       is_clicked <= '1';
85   elsif IN_I >= "00110000" and IN_I <= "00111001" and is_clicked = '0' then
86       --CHAR_TYPE <= "01000011";
87       CHAR_TYPE <= "00001100";
88       is_clicked <= '1';
89   elsif IN_I >= "00111010" and IN_I <= "01000000" and is_clicked = '0' then
90       --CHAR_TYPE <= "01000100";
91       CHAR_TYPE <= "00001101";
92       is_clicked <= '1';
93   else
94       is_clicked <= '0';
95   end if;
96
97   slow_counter <= 0;
98
99   else
100       slow_counter <= slow_counter + 1;
101
102   end if;
103 end if;
104
105 ONES_COUNT <= ones(3 downto 0);
106 ZEROS_COUNT <= zeros(3 downto 0);
107 ones <= (others => '0');
108 zeros <= (others => '0');
109 end process;

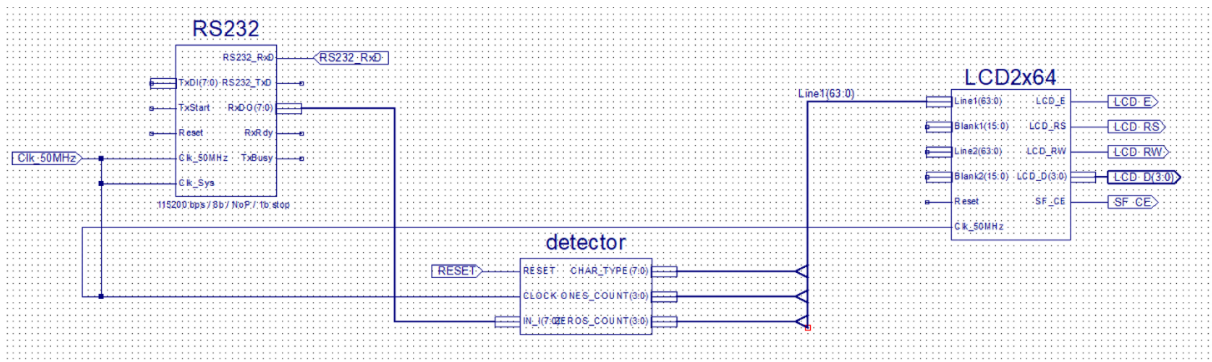
```

Z programu z poprzedniego laboratorium został jedynie dzielnik spowalniający licznik, który otoczył zupełnie nową logikę. Po dopuszczeniu do logiki, iterujemy przez cały wektor *IN_I* wprowadzonego znaku ASCII w systemie binarnym w celu zliczenia ilości '1'. Następnie wykrywamy:

- Jeśli wprowadzony znak ASCII jest z zakresu [97;122], wtedy przypisujemy do zmiennej *CHAR_TYPE* liczbę 10 w systemie dziesiętnym, czyli A w systemie szesnastkowym, która zostanie przekazana na wyświetlacz
- Jeśli z zakresu [65;90], wtedy zmienna *CHAR_TYPE* przyjmie liczbę B w systemie szesnastkowym
- Jeśli z zakresu [48;57], wtedy zmienna *CHAR_TYPE* przyjmie liczbę C w systemie szesnastkowym
- Dla pozostałych wartości – *CHAR_TYPE* = D

Dodatkowym zabezpieczeniem jest zmienna *is_clicked*, która wykrywa czy został załadowany znak ASCII z klawiatury. Przyjmuje wartość '0' gdy nic nie jest wprowadzane, '1' gdy jest wprowadzana wartość. Pod koniec procesu przekazujemy ilość zliczonych '0' i '1' oraz znak A/B/C/D na wyjście.

2. Podłączenie klawiatury i wyświetlacza



Podłączenie na schemacie odbywa się podobnie jak na poprzednim laboratorium, jedyne to co się zmienia to magistrala przekazywania danych na wyświetlacz. Docelowo na wyświetlaczu pokazują się dane w następującym formacie:

"05030A"

Gdzie „05” oznacza liczbę wykrytych ‘0’ w ciągu ASCII, „03” liczbę wykrytych ‘1’, a „A” – typ wykrytego znaku, w tym przypadku małej litery.

3. Pliki z rozszerzeniem .ucf

Pliki .ucf nie uległy wielkim zmianom, jedynie zakomentowano linijki odpowiadające za sterowanie licznikiem, został przypisany tylko przycisk RESET. Linijki dotyczące wyświetlacza i portu RS232 – bez zmian.

LCD.ucf:

```
1 # ==== Character LCD (LCD) ====
2 NET "LCD_E" LOC = "M18" | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
3 NET "LCD_RS" LOC = "L18" | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
4 NET "LCD_RW" LOC = "L17" | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
5 # LCD data connections are shared with StrataFlash connections SF_D<11:8>
6 NET "LCD_D<0>" LOC = "R15" | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
7 NET "LCD_D<1>" LOC = "R16" | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
8 NET "LCD_D<2>" LOC = "P17" | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
9 NET "LCD_D<3>" LOC = "M15" | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
10 NET "SF_CE" LOC = "D16" | IOSTANDARD = LVCMOS33 | DRIVE = 4 | SLEW = SLOW ;
```

genIO.ucf:

```
1  # soldered 50MHz Clock.
2  NET "Clk_50MHz" LOC = "C9" | IOSTANDARD = LVTTTL;
3  NET "Clk_50MHz" PERIOD = 20.0ns HIGH 50%;
4
5  # Ignore cross-clock domain data paths in timing analysis
6  #NET "Clk50" TNM = "TNM_Clk50";
7  #NET "ClkSys" TNM = "TNM_ClkSys";
8  #TIMESPEC "TS_12" = FROM "TNM_ClkSys" TO "TNM_Clk50" TIG;
9  #TIMESPEC "TS_13" = FROM "TNM_Clk50" TO "TNM_ClkSys" TIG;
10
11 #
12 # Push-buttons (Press = Hi)
13 #
14 #NET "btn_north" LOC = "V4" | IOSTANDARD = LVTTTL | PULLDOWN;
15 #NET "btn_east" LOC = "H13" | IOSTANDARD = LVTTTL | PULLDOWN;
16 NET "RESET" LOC = "K17" | IOSTANDARD = LVTTTL | PULLDOWN;
17 #NET "btn_west" LOC = "D18" | IOSTANDARD = LVTTTL | PULLDOWN;
18
19 #
20 # Slide switches (Up = Hi)
21 #
22 #NET "START_STOP" LOC = "L13" | IOSTANDARD = LVTTTL | PULLUP;
23 #NET "REVERSE" LOC = "L14" | IOSTANDARD = LVTTTL | PULLUP;
24 #NET "PAUSE" LOC = "H18" | IOSTANDARD = LVTTTL | PULLUP;
25 #NET "SW<3>" LOC = "N17" | IOSTANDARD = LVTTTL | PULLUP;
26
27 #
28 # Rotary encoder
29 #
30 #NET "ROT_A" LOC = "K18" | IOSTANDARD = LVTTTL | PULLUP;
31 #NET "ROT_B" LOC = "G18" | IOSTANDARD = LVTTTL | PULLUP;
32 #NET "ROT_CENTER" LOC = "V16" | IOSTANDARD = LVTTTL | PULLDOWN;
```

```

35 # Simple LEDs (Hi = On)
36 #
37 #NET "led<0>" LOC = "F12" | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 4;
38 #NET "led<1>" LOC = "E12" | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 4;
39 #NET "led<2>" LOC = "E11" | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 4;
40 #NET "led<3>" LOC = "F11" | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 4;
41 #NET "led<4>" LOC = "C11" | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 4;
42 #NET "led<5>" LOC = "D11" | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 4;
43 #NET "led<6>" LOC = "E9" | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 4;
44 #NET "led<7>" LOC = "F9" | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 4;
45
46 #
47 # PS/2 port
48 #
49 #NET "PS2_Data" LOC = "G13" | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 8;
50 #NET "PS2_Clk" LOC = "G14" | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 8;
51
52 #
53 # VGA
54 #
55 #NET "VGA_R" LOC = "H14" | IOSTANDARD = LVTTTL | SLEW = FAST | DRIVE = 8;
56 #NET "VGA_G" LOC = "H15" | IOSTANDARD = LVTTTL | SLEW = FAST | DRIVE = 8;
57 #NET "VGA_B" LOC = "G15" | IOSTANDARD = LVTTTL | SLEW = FAST | DRIVE = 8;
58 #NET "VGA_HS" LOC = "F15" | IOSTANDARD = LVTTTL | SLEW = FAST | DRIVE = 8;
59 #NET "VGA_VS" LOC = "F14" | IOSTANDARD = LVTTTL | SLEW = FAST | DRIVE = 8;
60
61 # RS-232 Serial Port: DCE
62 NET "RS232_RXD" LOC = "R7" | IOSTANDARD = LVTTTL ;
63 #NET "RS232_TXD" LOC = "M14" | IOSTANDARD = LVTTTL | DRIVE = 8 | SLEW = SLOW ;
64 # RS-232 Serial Port: DTE
65 #NET "RS232_RXD" LOC = "U8" | IOSTANDARD = LVTTTL ;
66 #NET "RS232_TXD" LOC = "M13" | IOSTANDARD = LVTTTL | DRIVE = 8 | SLEW = SLOW ;

```

4. Wnioski

Detektor wprowadzonego znaku ASCII działał poprawnie. Niestety nie udało się zrealizować zliczania '1' i '0' w ciągu przesyłanego znaku.