

Projekt : Terminál pre odomykanie / zamykanie dverí pomocou PIN kódu

Členovia tímu

Baránek Michal

Bartoň Jan

Bařina Tadeáš

Bekeč Alexander

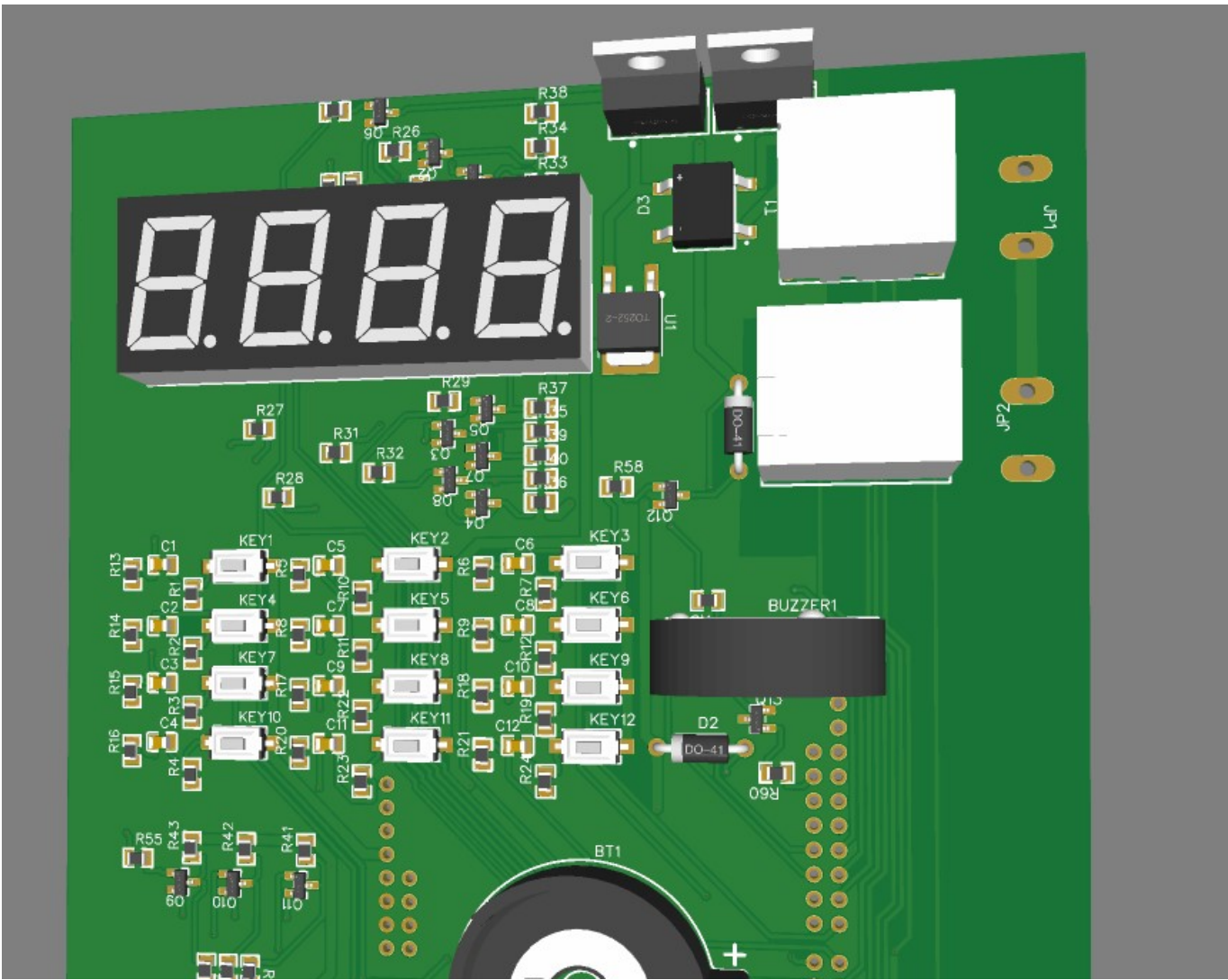
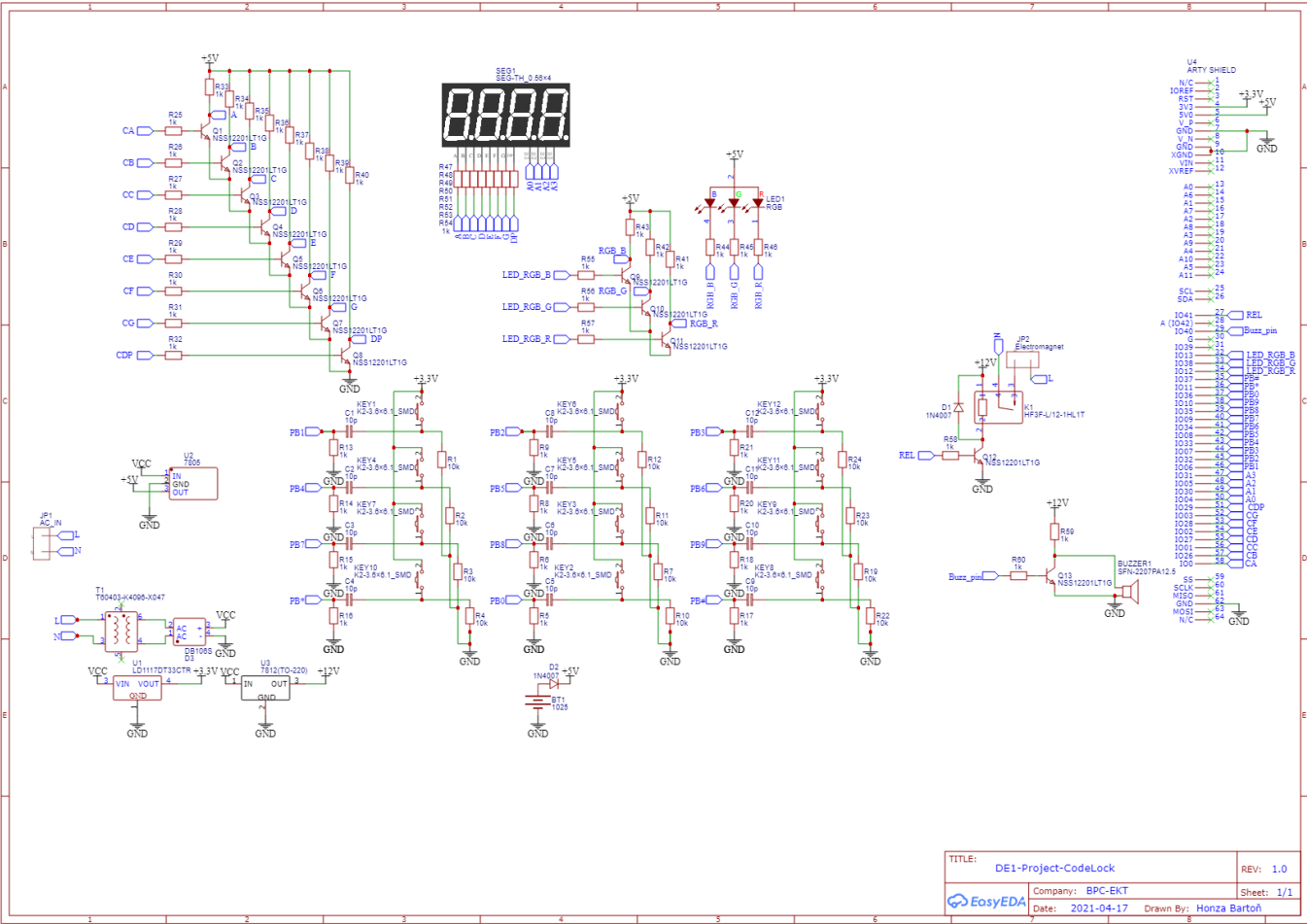
Odkaz na zložku s projektom: https://github.com/xbarto0c/Digital-electronics-1/tree/main/Labs/Project_code_lock

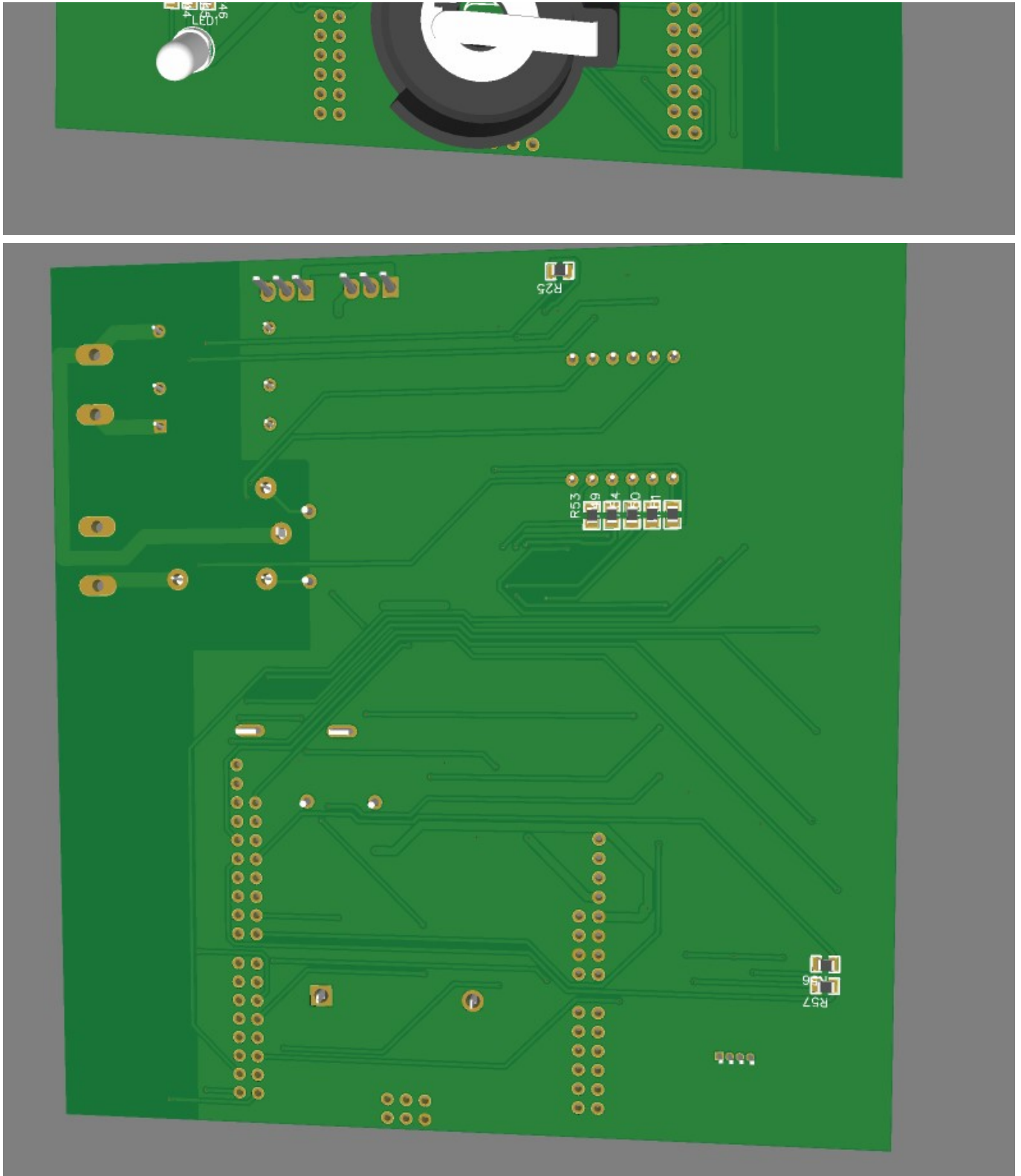
Ciele projektu

Cieľom projektu bolo navrhnuť terminál pre odomykanie dverí pomocou štvormiestneho PIN kódu. Projekt mal obsahovať tlačidlá pre zadávanie PIN kódu, štyri sedemsegmentové diode pre zobrazenie hesla a relé pre ovládanie zámku.

Popis hardvéru

Základným hardvérom projektu je doska Arty A7-35T. Nakoľko táto doska neobsahuje niektoré súčasti, ktoré boli potrebné pri návrhu projektu, bola navrhnutá periféria - DPS obsahujúca potrebné súčasti (sedemsegmentový displej, tlačidlá pre zadávanie PIN kódu s CR článkami pre vytvorenie pulzu pri stlačení tlačidla, piezo bzučiak, relé...):





Popis a simulácia VHDL modulov

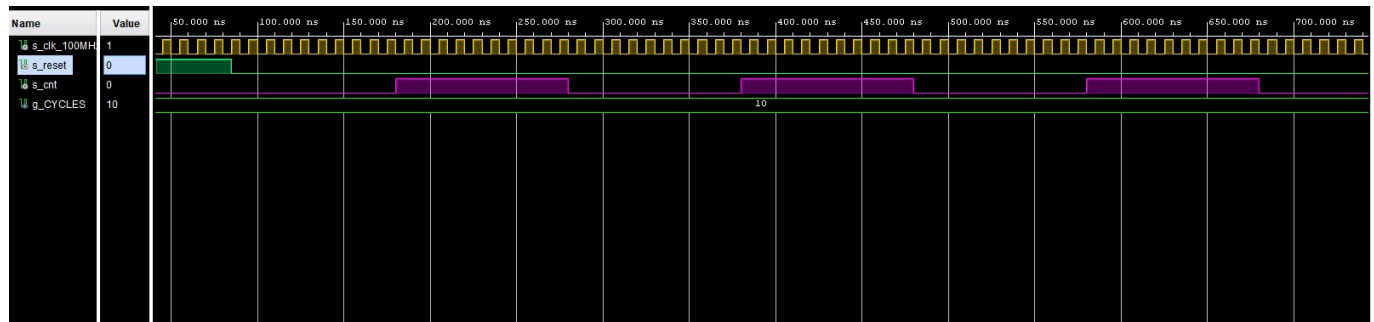
Upozornenie: Všetky moduly majú upravené pomery a trvania pre účely simulácie, preto simulácie ukazujú hlavný princíp a nie presný beh projektu a modulov.

clock_divider

Odkaz na zdrojový kód: https://github.com/xbarto0c/Digital-electronics-1/blob/main/Labs/Project_code_lock/DE1_project_code_lock/DE1_project_code_lock.srscs/sources_1/new/clock_divider.vhd

Popis: Modul clock_divider delí frekvenciu hodinového signálu, teda vytvára signál s menšou frekvenciou ako samotný hodinový signál.

Simulácia:

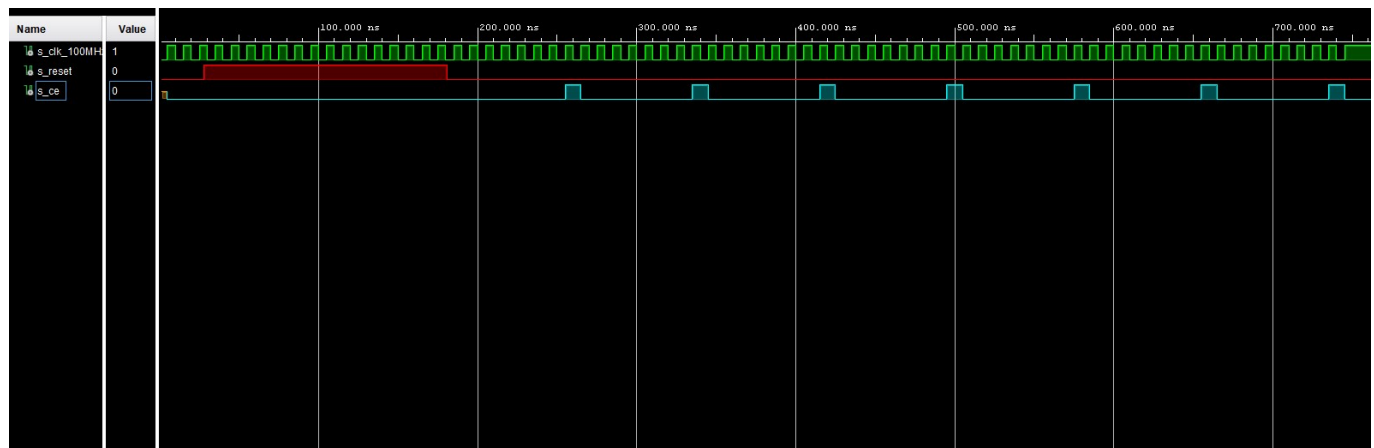


clock_enable

Odkaz na zdrojový kód: https://github.com/xbarto0c/Digital-electronics-1/blob/main/Labs/Project_code_lock/DE1_project_code_lock/DE1_project_code_lock.srscs/sources_1/new/clock_enable.vhd

Popis: Modul clock_enable vytvára pulz po priebehu určitého počtu períód hodinového signálu.

Simulácia:



cnt_up_down

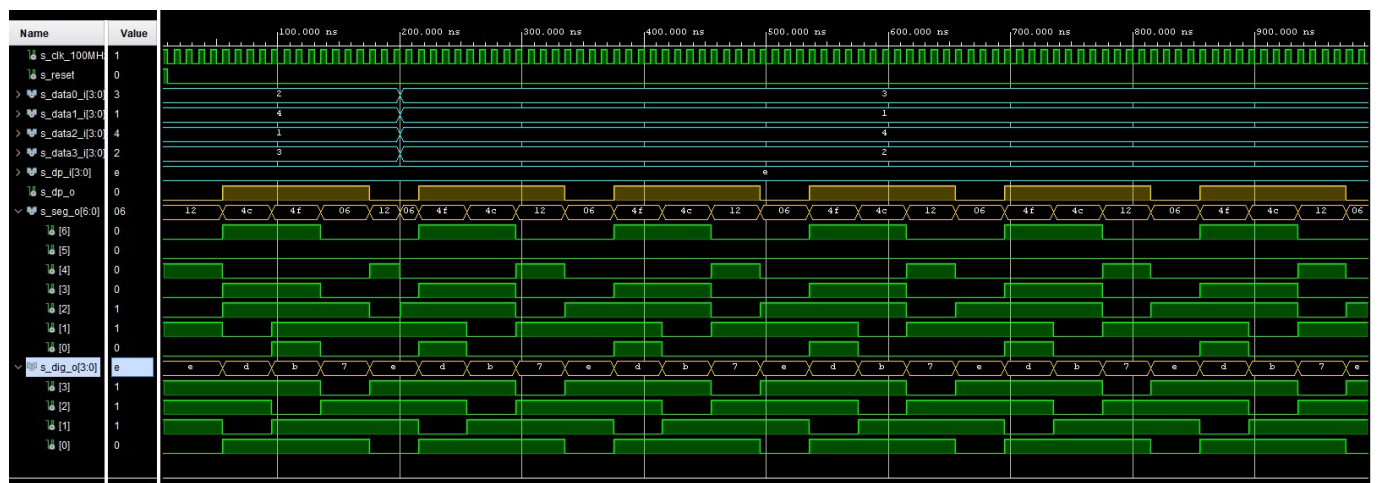
Odkaz na zdrojový kód: https://github.com/xbarto0c/Digital-electronics-1/blob/main/Labs/Project_code_lock/DE1_project_code_lock/DE1_project_code_lock.srscs/sources_1/new/cnt_up_down.vhd

Popis: Modul cnt_up_down je up/down bitový čítač.

Name	Value	
s_clk_100MHz	0	
s_reset	0	
s_en	0	
s_cnt_up	0	
s_cnt[3:0]	8	

Odkaz na zdrojový kód: https://github.com/xbarto0c/Digital-electronics-1/blob/main/Labs/Project_code_lock/DE1_project_code_lock/DE1_project_code_lock.srsrcs/sources_1/new/display_driver7seg_digit.vhd

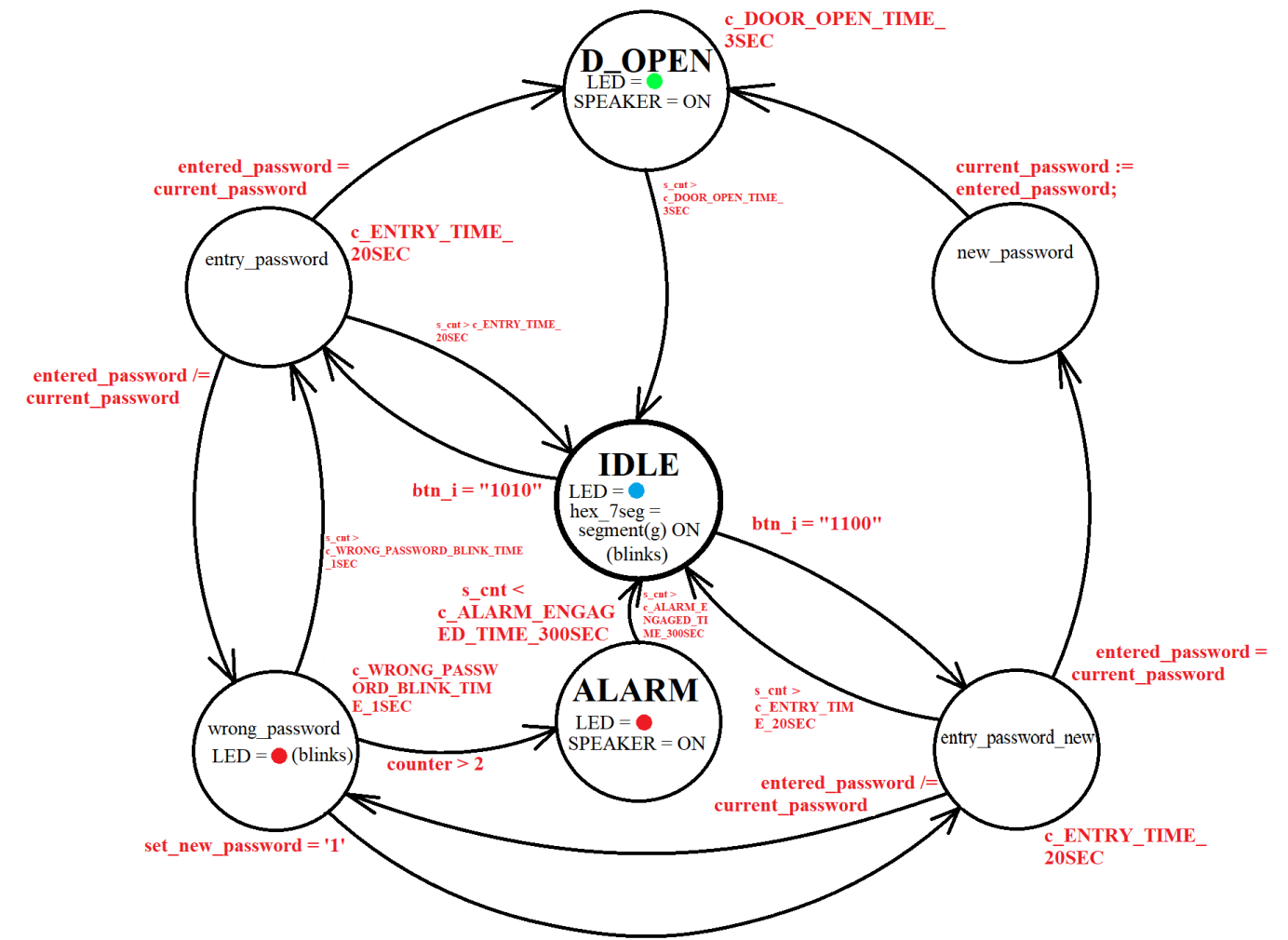
Simulácia:



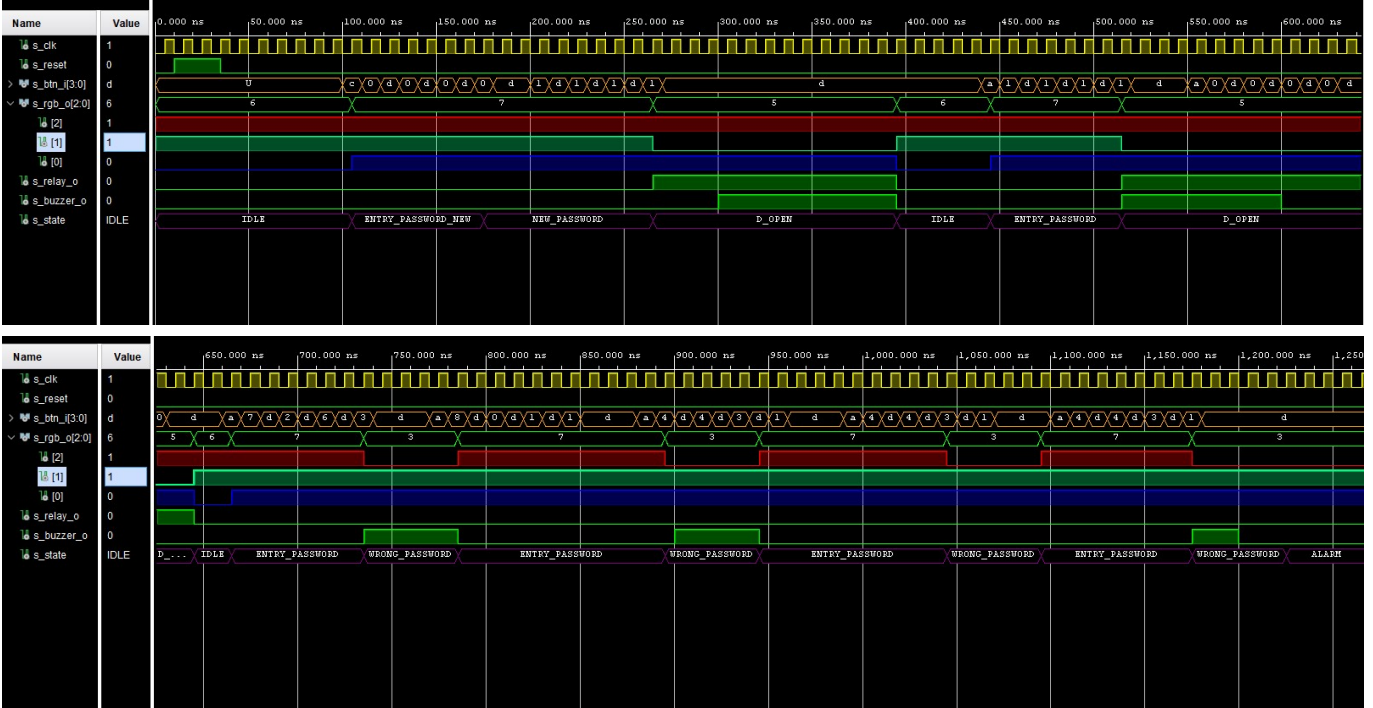
Odkaz na zdrojový kód: https://github.com/xbarto0c/Digital-electronics-1/blob/main/Labs/Project_code_lock/DE1_project_code_lock/DE1_project_code_lock.srcs/sources_1/new/door_lock_core.vhd

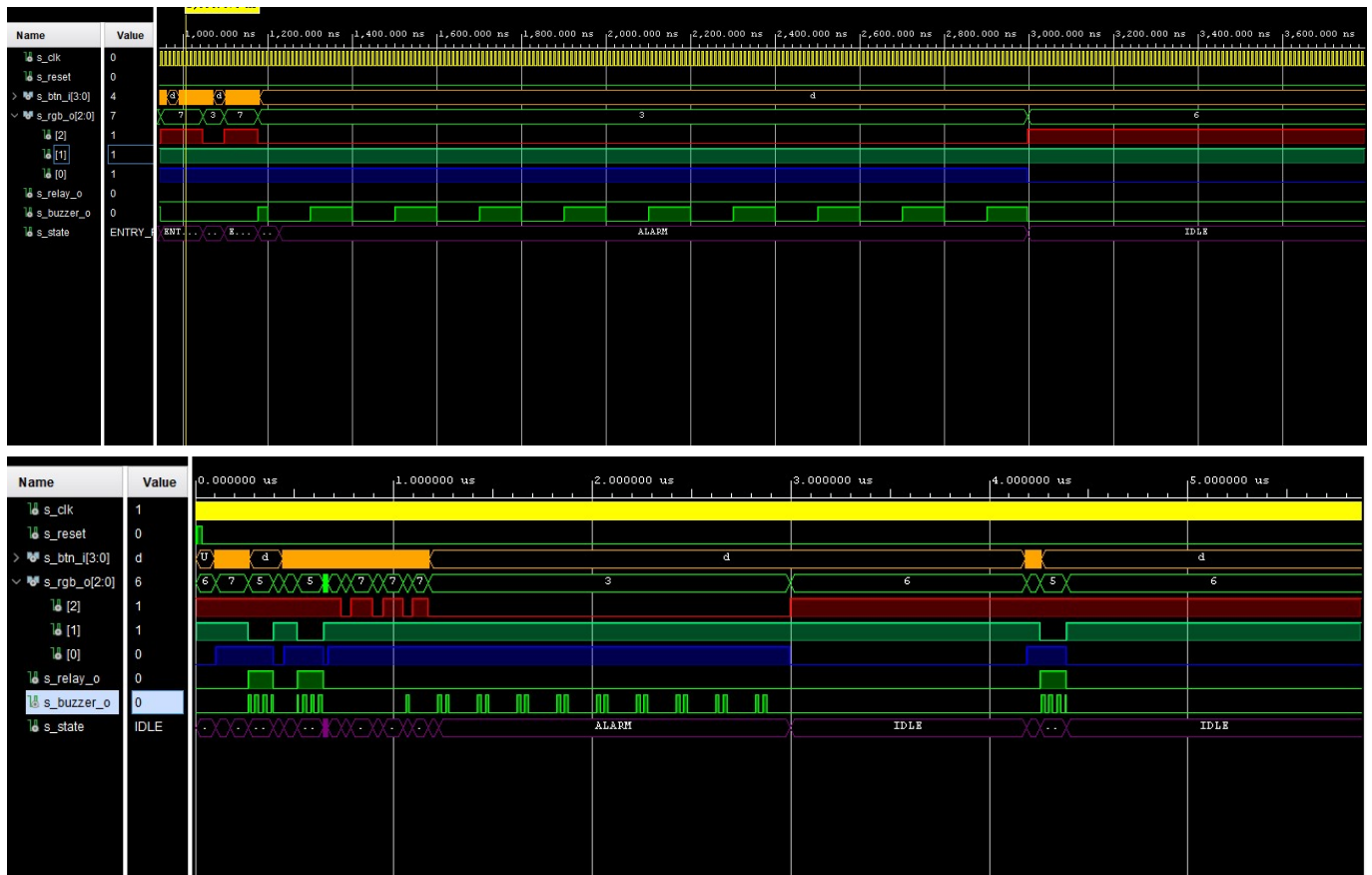
5 / 12

jadro tvorí stavový automat:



Simulácia:





hex_7_seg

Odkaz na zdrojový kód: https://github.com/xbarto0c/Digital-electronics-1/blob/main/Labs/Project_code_lock/DE1_project_code_lock/DE1_project_code_lock.srscs/sources_1/new/hex_7_seg.vhd

Popis: Modul hex_7_seg je dekodérom štvorbitového výstupu na výstup pre ovládanie sedemsegmentového displeja. Pre účely použitia v tomto projekte bol výstup pre A zmenený na - (pomíčku) a B na prázdny displej.

Simulácia:



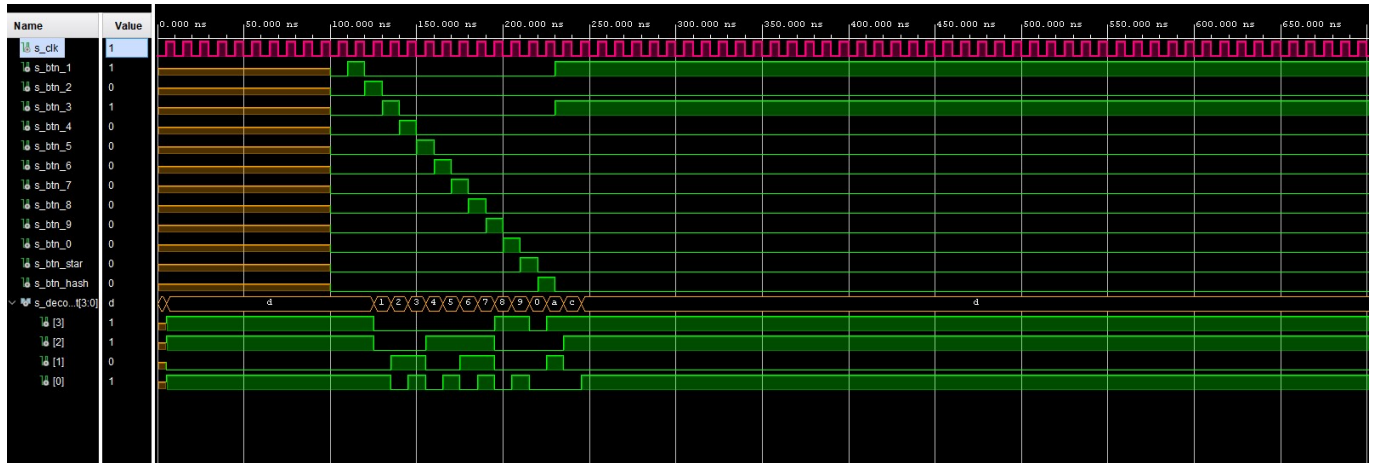
keyboard_decoder

Odkaz na zdrojový kód: https://github.com/xbarto0c/Digital-electronics-1/blob/main/Labs/Project_code_lock/DE1_project_code_lock/DE1_project_code_lock.srscs/sources_1/new/keyboard_decoder.vhd

Popis: Modul keyboard_decoder prijíma výstupy jednotlivých tlačidiel, ktoré následne spojí do jedného signálu, ktorý dekoduje na štvorbitový reťazec pre modul door_lock_core. Modul má "zabudovanú" ochranu

proti stlačeniu dvoch tlačidiel naraz.

Simulácia:

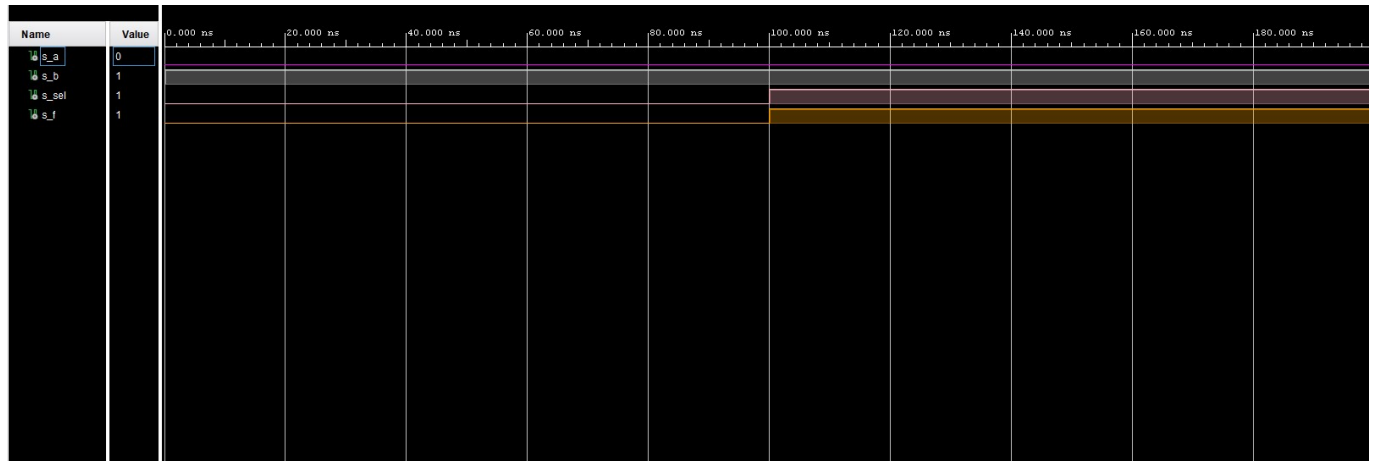


mux_2to1

Odkaz na zdrojový kód: https://github.com/xbarto0c/Digital-electronics-1/blob/main/Labs/Project_code_lock/DE1_project_code_lock/DE1_project_code_lock.srscs/sources_1/new/mux_2to1.vhd

Popis: Modul mux_2to1 je multiplexer s dvoma datovými vstupmi (jednobitovými), jedným vstupom rozhodovacím (tiež jednobitový) a jedným výstupom.

Simulácia:

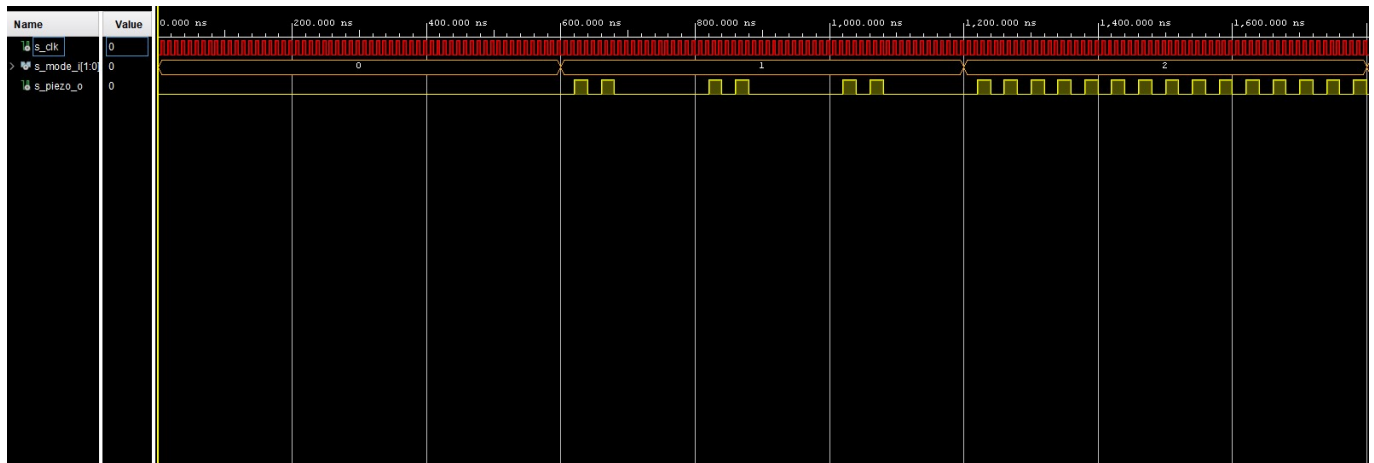


piezo_driver

Odkaz na zdrojový kód: https://github.com/xbarto0c/Digital-electronics-1/blob/main/Labs/Project_code_lock/DE1_project_code_lock/DE1_project_code_lock.srscs/sources_1/new/piezo_driver.vhd

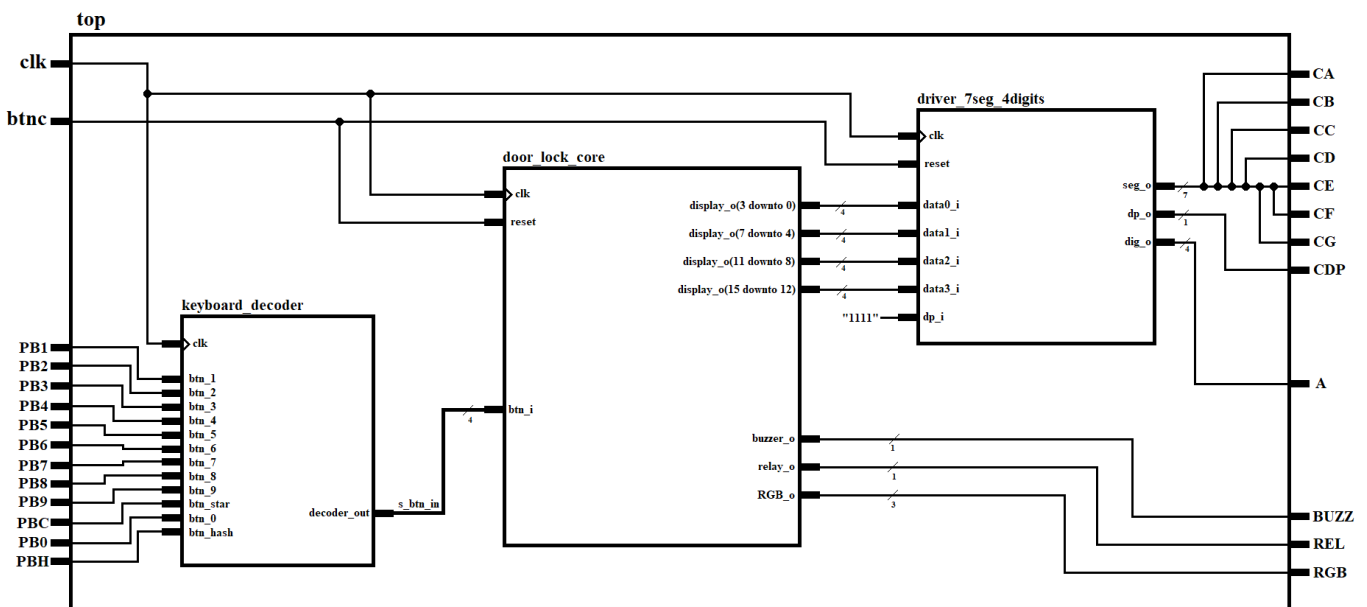
Popis: Modul piezo_driver je ovládač piezo bzučiaka. Generuje obdĺžnikový signál s frekvenciou 1000 Hz a zároveň podľa dvojbítového vstupu (mode_i) rozhoduje, či má byť výstup nulový, konštantný tón alebo pípanie (prepínanie medzi stavom vypnutý - konštantný tón)

Simulácia:



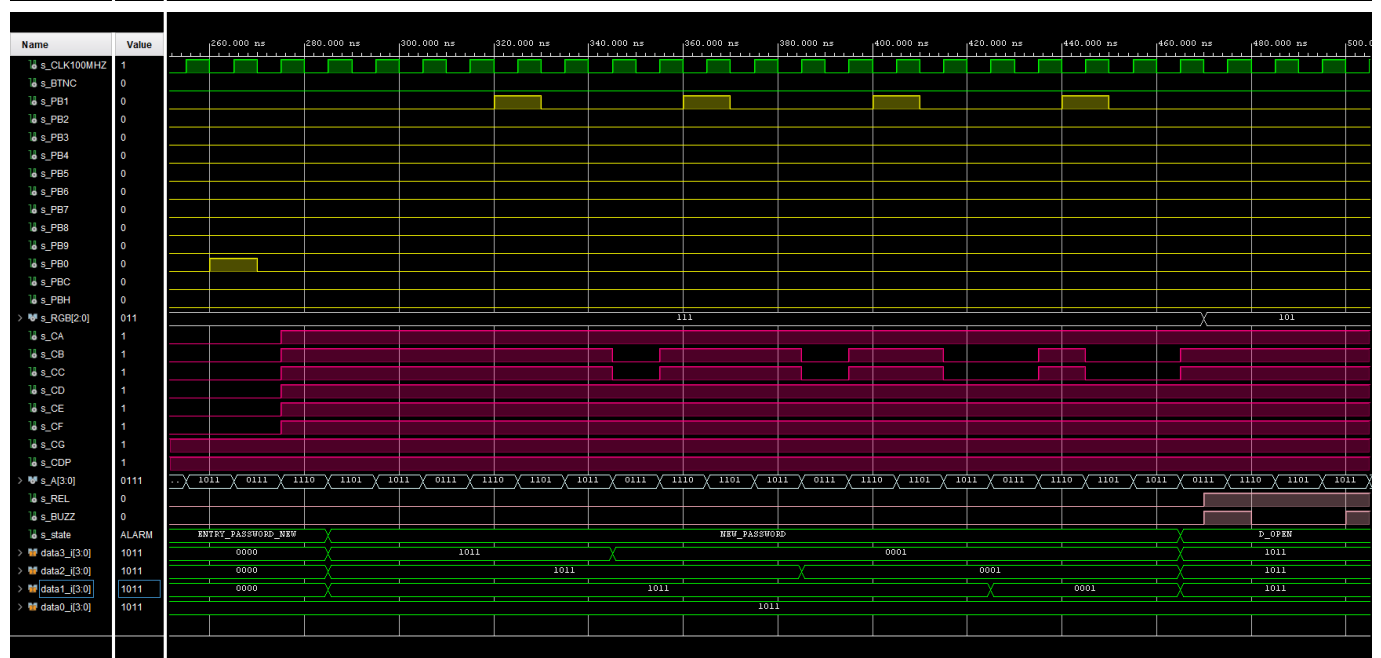
Popis a simulácia TOP modulu

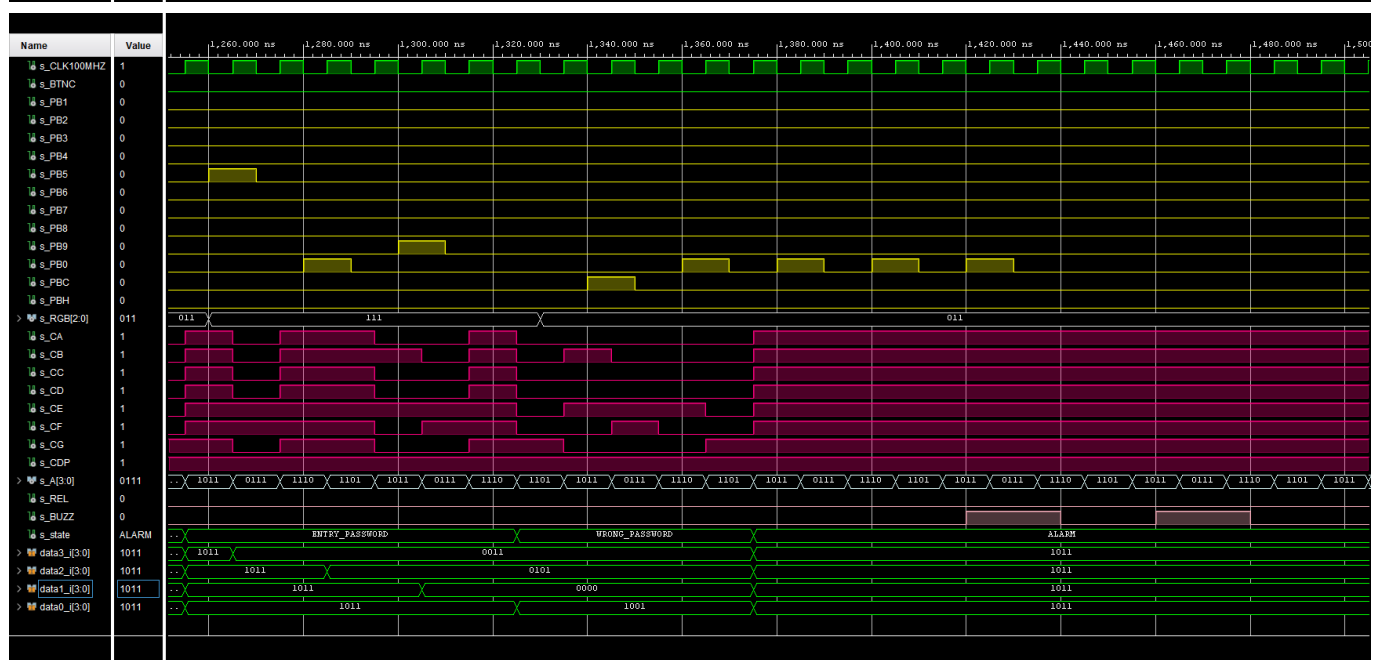
Odkaz na zdrojový kód: https://github.com/xbarto0c/Digital-electronics-1/blob/main/Labs/Project_code_lock/DE1_project_code_lock/DE1_project_code_lock.srcs/sources_1/new/top.v
hd



V rámci top modulu sa prijímajú vstupy z periférie (hlavne výstupy tlačidiel) a hodinový signál z hlavnej dosky. Pulzy z tlačidiel sú spracované dekodérom keyboard_decoder na štvorbityvý výstup, ktorý je pripojený na vstup modulu door_lock_core, kde sa tento vstup postupne zapisuje na jednotlivé pozície zadávaného hesla, overí sa platnosť a na základe toho sa presúva do požadovaného stavu. Postupným chodom medzi stavami sú určené aj jednotlivé výstupy, ktoré sú pripojené na príslušné výstupy top modulu (RGB LED dióda, piezo bzučiak, relé), prípadne na dekodér a ovladač štvormiestneho sedemsegmentového displeja, ktorého výstup je následne pripojený na výstupy top modulu pre displej.

Name	Value	0.000 ns	20.000 ns	40.000 ns	60.000 ns	80.000 ns	100.000 ns	120.000 ns	140.000 ns	160.000 ns	180.000 ns	200.000 ns	220.000 ns	240.000 ns	
🔍 s_CLK100MHZ	1	[Signal waveform]													
🔍 s_BTNC	0	[Signal waveform]													
🔍 s_PB1	0	[Signal waveform]													
🔍 s_PB2	0	[Signal waveform]													
🔍 s_PB3	0	[Signal waveform]													
🔍 s_PB4	0	[Signal waveform]													
🔍 s_PB5	0	[Signal waveform]													
🔍 s_PB6	0	[Signal waveform]													
🔍 s_PB7	0	[Signal waveform]													
🔍 s_PB8	0	[Signal waveform]													
🔍 s_PB9	0	[Signal waveform]													
🔍 s_PB0	0	[Signal waveform]													
🔍 s_PBC	0	[Signal waveform]													
🔍 s_PBH	0	[Signal waveform]													
> 🔍 s_RGB[2:0]	011	[Signal waveform]													
🔍 s_CA	1	[Signal waveform]													
🔍 s_CB	1	[Signal waveform]													
🔍 s_CC	1	[Signal waveform]													
🔍 s_CD	1	[Signal waveform]													
🔍 s_CE	1	[Signal waveform]													
🔍 s_CF	1	[Signal waveform]													
🔍 s_CG	1	[Signal waveform]													
🔍 s_CDP	1	[Signal waveform]													
> 🔍 s_AR[0]	0111	[Signal waveform]													
🔍 s_REL	0	[Signal waveform]													
🔍 s_BUZZ	0	[Signal waveform]													
🔍 s_state	ALARM	[Signal waveform]													
> 🔍 data3_i[3:0]	1011	[Signal waveform]													
> 🔍 data2_i[3:0]	1011	[Signal waveform]													
> 🔍 data1_i[3:0]	1011	[Signal waveform]													
> 🔍 data0_i[3:0]	1011	[Signal waveform]													







Video

Link na video: <https://www.youtube.com/watch?v=K0PdgdODcqE>

Referencie

Informácie a nápady boli čerpané prevažne z predošlých počítačových cvičení.