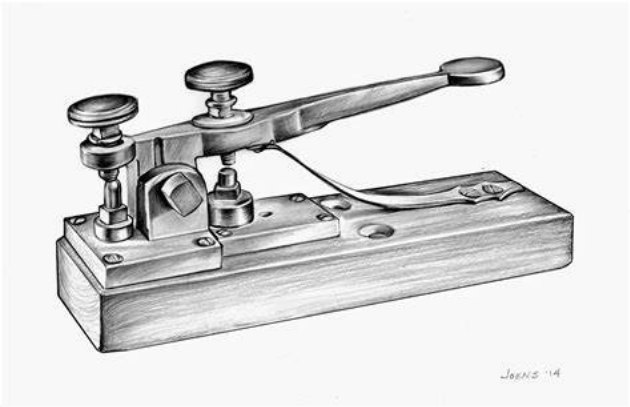
SISTEMA MORSE SU FPGA:
PROGETTO DI CODIFICA E DECODIFCA

INGEGNERIA INFORMATICA E ROBOTICA
SISTEMI ELETTRONICI EMBEDDED

A.A 2024/2025

TABLE OF CONTENTS

Introduction: Morse code



Developed in the 19th century by Samuel Morse and Alfred Vail. It was mainly used in the **telegraph** for long-distance communications.

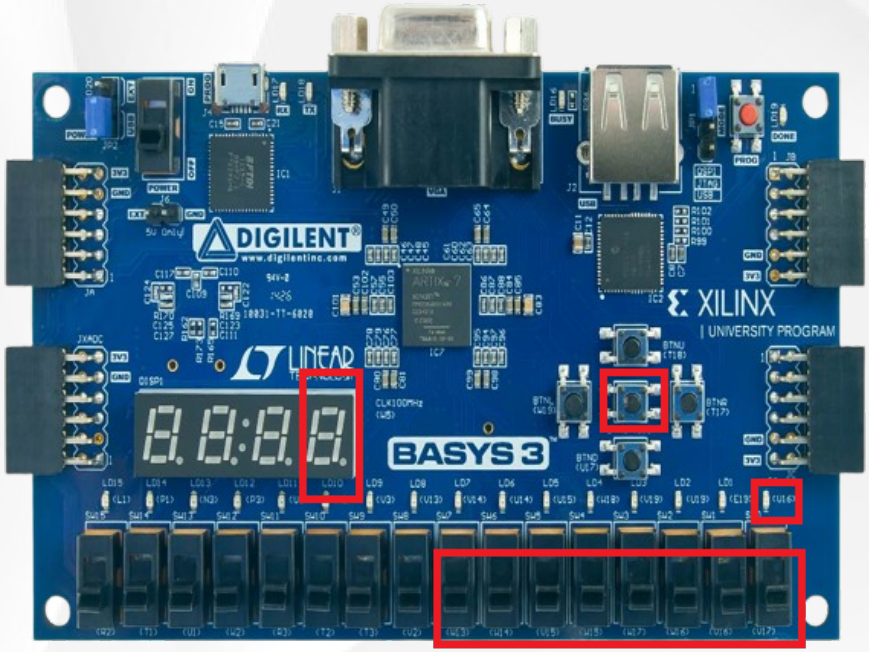
Purposes of use:

- Military field
- Maritime field
- Today it is useful in specific contexts such as the medical sector

A ●−	J ●−−−	S ●●●
B −●●●	K −●−	T −
C −●−●	L ●−●●	U ●●−
D −●●	M −−	V ●●●−
E ●	N −●	W ●−−
F ●●−●	O −−−	X −●●−
G −−●	P ●−−●	Y −●−−
H ●●●●	Q −−●−	Z −−●●
I ●●	R ●−●	

Project goal

Implementation of a Morse code encoding and decoding system using an **FPGA Artix-7 by Xilinx**. The resources provided by the **Basys3** board will be used as input/output. The Hardware is programmed using **Vivado** Software.



AMD

Vivado

VHDL

Very High Speed Integrated Circuit Hardware Description Language

Project goal

ENCODING

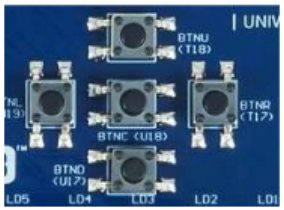
insert char in binary

Translate into Morse code

Blinking led

DECODING

Insert the Morse code of a char/numer



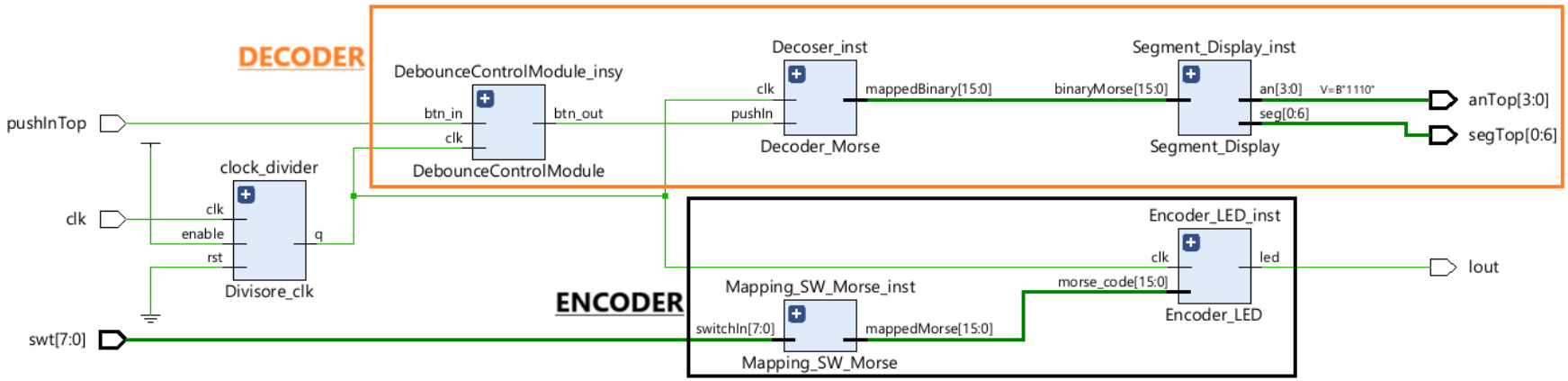
Translate dots and dashes into 7 segment rappresentation

Show the char/numer

Project Structure- RTL

Register Transfer Level:

- How data moves between registers;
- What logical or arithmetic operations are performed on the data during transfers;



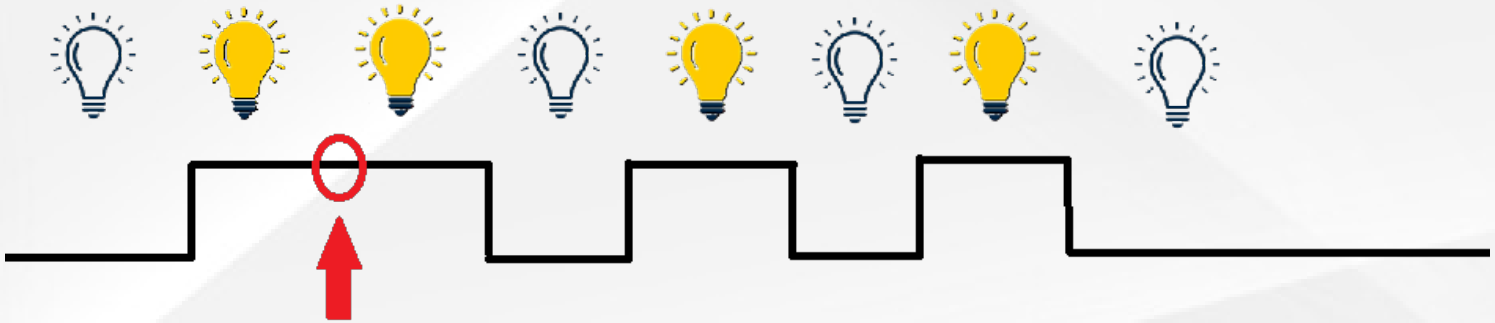
Project Structure - Encoding

Two modules:

- **Mapping_SW_Morse**, It converts the binary input provided through the switches into an identifying Morse code encoding.

  e.g. $D \rightarrow 01000100 \rightarrow 1101010000000000 \rightarrow$ - . .

- **Encoder_LED**,It makes the LED blink based on the value obtained from the mapping module mentioned above.



Project Structure -Encoder_LED Code

```
58  if clk'event and clk = '1' then
59          if i > 0 then
60              if morse_code(i) = '1' then
61                  led <= '1';
62                  test_on <= test_on + 1;
63                  wait_timer_on := wait_timer_on  + 1;
64              else
65                  led <= '0';
66                  test_off <= test_off + 1;
67                  wait_timer_off := wait_timer_off  + 1;
68              end if;
69
70              if wait_timer_on = 5000000 then
71                  led <= '0';
72                  wait_timer_on := 0;
73                  i := i - 1;
74              end if;
75
76              if wait_timer_off = 3000000 then
77                  wait_timer_off := 0;
78                  i := i - 1;
79              end if;
80
81
82          else
83              i :=15 ;
84              wait_timer_on := 0;
85              wait_timer_off := 0;
86          end if;
87
88  end if;
```

Project Structure - Decoding

Three modules:

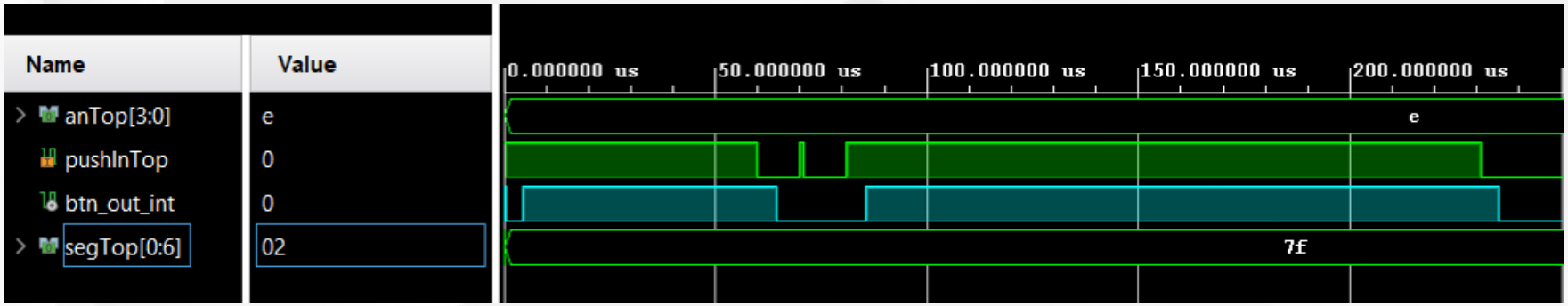
- **DebounceControlModule**, manages the unwanted mechanical bounce of the pushbuttons;
- **Decoder_Morse**, interpret the input signals from the pushbuttons and convert them into binary strings (encoding as seen in the previous slides);
- **Segment_Display**, performs the mapping from the Morse code entered using the pushbuttons into a binary string for turning on the 7-segment display.

Project Structure - DebounceControlModule code

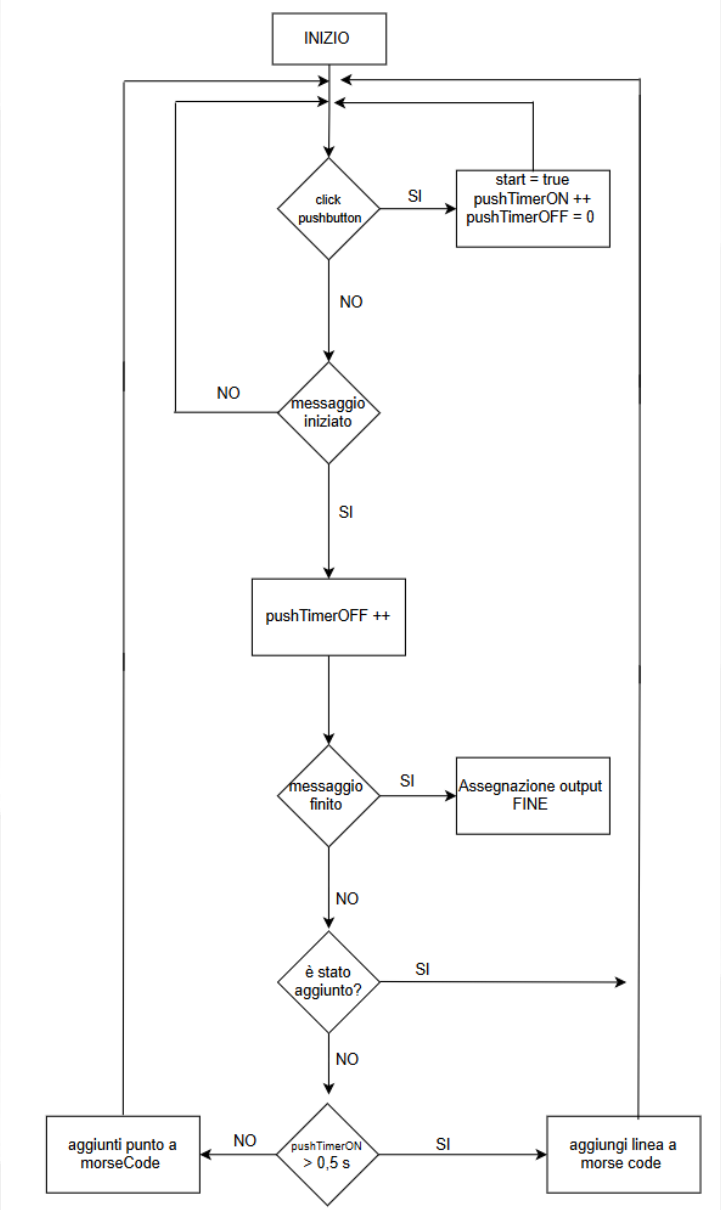
```
process(clk)
begin
    if clk'event and clk = '1' then
        -- Se il pulsante è cambiato rispetto allo stato precedente
        if btn_in /= btn_past then
            -- Se il pulsante è cambiato, resetta il contatore
            counter <= 0;
        else
            -- Se il pulsante è stabile, incrementa il contatore
            if counter < DEBOUNCE_COUNT_MAX then
                counter <= counter + 1;
            else
                -- Dopo il tempo di debounce, aggiorna lo stato finale
                btn_reg <= btn_in;
            end if;
        end if;
        -- Memorizza lo stato precedente del pulsante
        btn_past <= btn_in;
    end if;
end process;

-- Uscita finale del pulsante stabilizzato
btn_out <= btn_reg;
```

Project Structure - DebounceControlModule Simulation

Simulated Debounce → 1 us

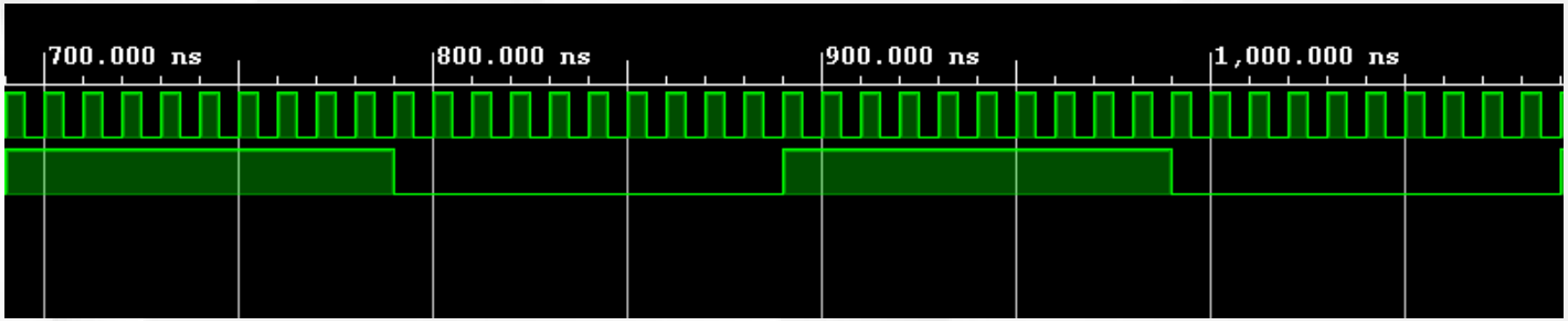


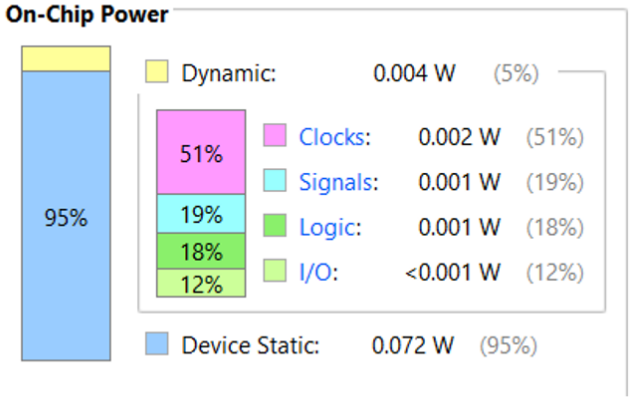
Project Structure - Flow Diagram Decoder_Morse

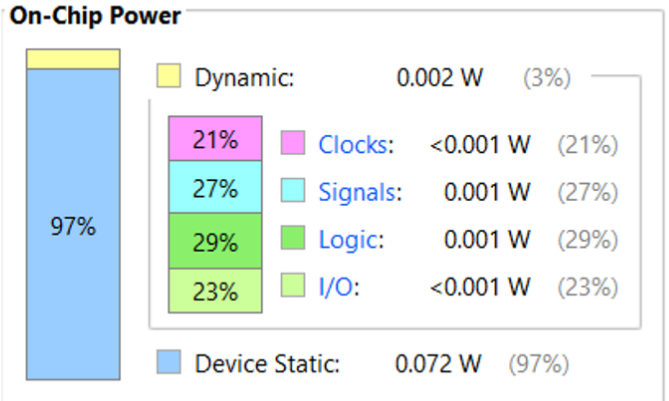


Project Structure - Clock Division

100 MHz → 10 MHz







Conclusions and Future Developments

Possible improvements:

- Development of UART connection for the conversion of multiple characters simultaneously..
- Clock gating to prevent the unused mode from consuming power.

THANK YOU FOR YOUR ATTENTION