

Prosty stoper

Zadanie: Układ ma mierzyć czas z dokładnością do setnych części sekundy. Kolejne naciśnięcia przycisku **BTNC** (start_stop_button_i) (w trybie zdalnym - wirtualny przycisk **BTN0**) wywołują następujące działanie:

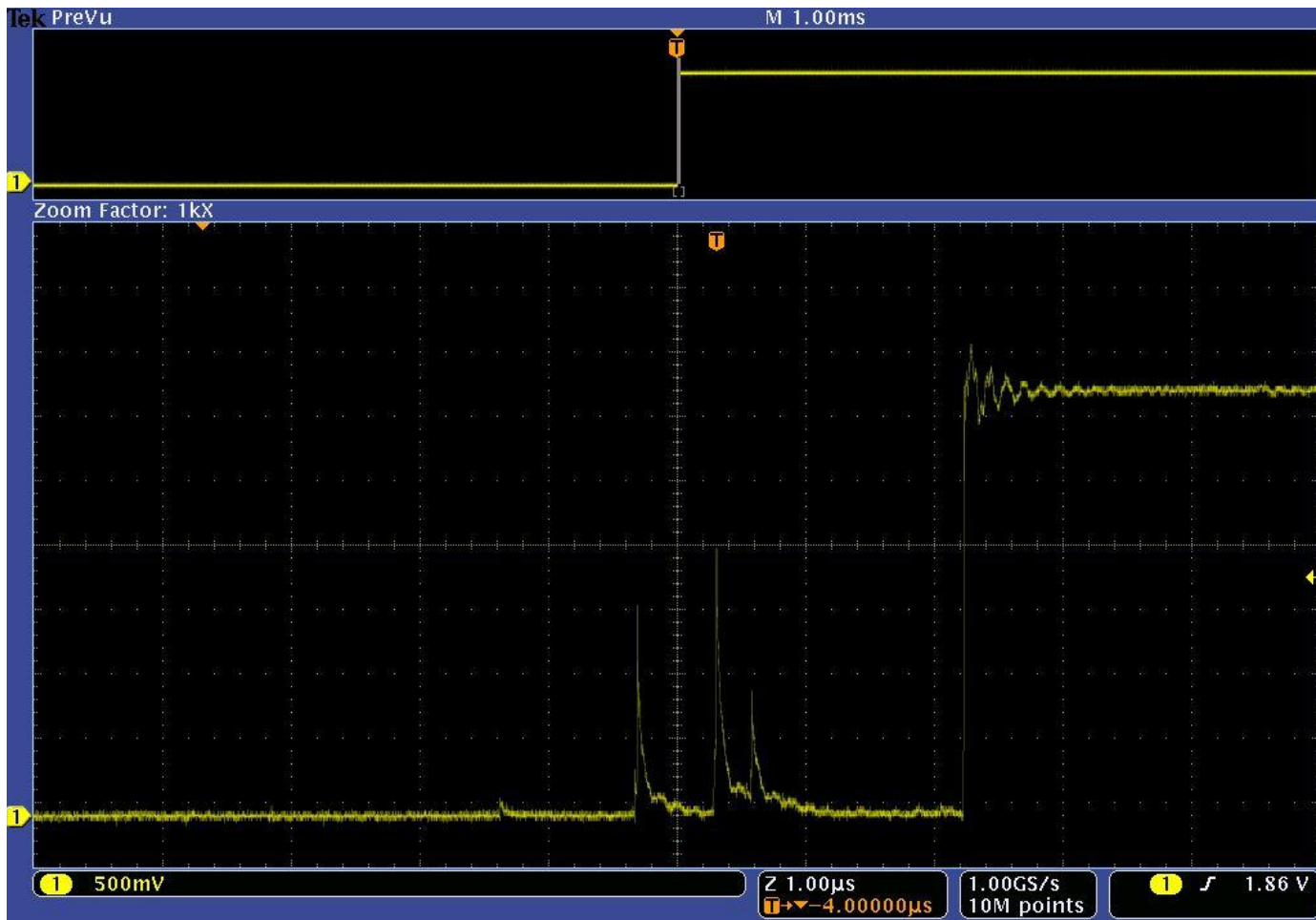


Rys.1 Działanie stopera po kolejnych naciśnięciach przycisku.

Czas wyświetlany ma być na wyświetlaczu LED w postaci: **SS.DD**, gdzie: **SS**=sekundy, **DD**=setne części sekundy. Przekroczenie czasu 59.99 sekund należy zasygnalizować wyświetleniem specjalnego symbolu oznaczającego przepełnienie, np. ("--.--").

Przycisk **BTNR** (rst_i) (w trybie zdalnym - wirtualny przycisk **BTN3**) to dodatkowy reset asynchroniczny (powinien zresetować stoper – wyświetlić „00.00” i oczekiwać na start).

Uwaga: Układ należy zabezpieczyć przed drganiami przycisku **BTNC** (**BTN0**). Dobrze jest założyć, że po naciśnięciu przycisku styki przycisku mogą drgać przez czas aż do 50 ms.



Rys.2 Przykładowy przebieg napięcia podczas naciskania przycisku.

W układzie należy osadzić, jako *component*, blok sterowania wyświetlaczem z poprzedniego ćwiczenia.

Należy wykonać symulację funkcjonalną oraz zweryfikować układ praktycznie poprzez zaprogramowanie płytki testowej.

Minimalne wymagania dotyczące symulacji i weryfikacji na płycie: wykonać symulację i weryfikację tak, aby pokazać start, zatrzymanie (dla weryfikacji po dowolnym czasie – dla symulacji powinien to być czas równy 1,xy sekund; gdzie xy to dwie ostatnie cyfry numeru indeksu studenta lub jednego ze studentów zaliczających ćwiczenie) oraz reset stopera (za pomocą `start_stop_button_i`). Weryfikacja na płycie powinna także zademonstrować przepełnienie i kasowanie przepełnienia (za pomocą `start_stop_button_i`). Na zakończenie weryfikacji pokazać działanie wejścia resetu asynchronicznego (`rst_i`) podczas pracy stopera. Częstotliwość zegara w symulacji – 100 MHz.

Uwaga, w trybie zdalnym podczas weryfikacji stopera na płycie proszę włączyć emulację drgań zestyków (zaznaczyć **SW/BTN bouncing**).

Fragment głównego pliku projektowego VHDL z deklaracją sygnałów:

```
entity top is
  Port ( clk_i : in STD_LOGIC;
        rst_i : in STD_LOGIC;
        start_stop_button_i : in STD_LOGIC;
        led7_an_o : out STD_LOGIC_VECTOR (3 downto 0);
        led7_seg_o : out STD_LOGIC_VECTOR (7 downto 0));
end top;
```

Plik z ograniczeniami projektowymi dla płytki Nexys-A7 (układ FPGA xc7a100tcsg324-1): [iup7s.xdc](#)

Wersja pliku z ograniczeniami projektowymi przeznaczona dla trybu zdalnego (obsługa przycisków wirtualnych): [iup7z.xdc](#)