Hochschule für angewandte Wissenschaften und Kunst

Hardware- & Software Entwurfsmuster

-

Meilenstein 2

Autoren: Kürsat Avci, Ben Schuhknecht
Betreuer: Prof. Dr.-Ing. Steffen Kaufmann
Datum: 26. Dezember 2025

1 Vorbereitung

Taktteiler

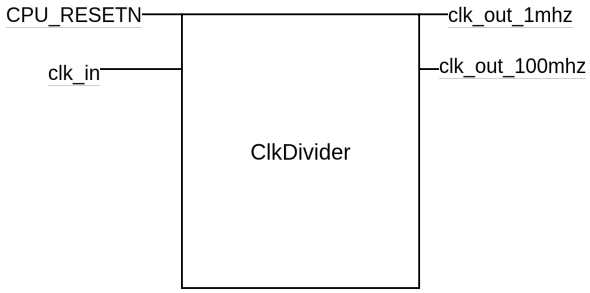

Fig. 1. Blockschaltbild des ClkDivider-Moduls (links in, rechts out)

Das ClkDivider-Modul bekommt 100 MHz als Input und macht daraus einen 1 MHz Output indem es ein signal „clk_div“ high oder low schaltet, wenn der Zähler im fünfzigsten Durchlauf ist. Außerdem wird der Input-Takt durchgeschaltet um „Clock skew“ zu vermeiden.

```
clk_out_100mhz <= clk_in;

    process (clk_in)
    begin
        if rising_edge(clk_in) then
            if CPU_RESETN = '0' then
                count   <= (others => '0');
                clk_div <= '0';
            else
                -- toggle every 50 cycles => 100 MHz / (2*50) = 1 MHz
                if count = 49 then
                    count   <= (others => '0');
                    clk_div <= not clk_div;
                else
                    count <= count + 1;
                end if;
            end if;
        end if;
    end process;

clk_out_1mhz <= clk_div;
```

Sendemodul

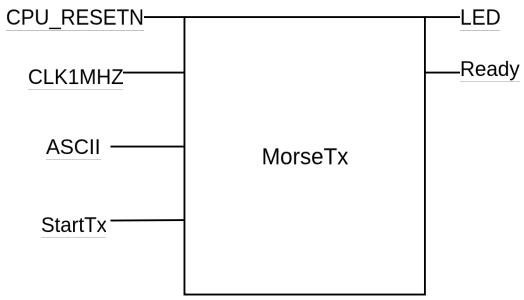

Fig. 2. Blockschaltbild des MorseTx-Moduls (links in, rechts out)

Als Sender wurde ein MorseTx-Modul mittels einer endlichen Zustandsmaschine (Fig. 3) implementiert. MorseTx kann hierbei die Zustände RESET, IDLE, LOAD, MARK oder GAP einnehmen. Beim Anschalten des FPGAs, nach einem Reset oder nach dem Schreiben des Bitstreams wird ein RESET ausgelöst und der IDLE-Zustand eingenommen, alle Werte werden zurückgesetzt. Die LEDs sind aus und das Modul ist bereit Nachrichten zu senden. Über StartTx (in Port) wird das Modul in den LOAD-Zustand gebracht. Das gewünschte ASCII-Zeichen wird geladen. Je nach dem welches Zeichen gerade gesendet werden soll wird in den GAP-Zustand oder in den MARK-Zustand übergegangen. MARK entscheidet hierbei ob es sich um einen „Dash“ oder „Dot“ handelt. Sind alle Zeichen gesendet wird in den Zustand GAP übergegangen. Gap unterscheidet zwischen „WORD_GAP“ und „LETTER_GAP“ und passt jeweils „units_left“ an. Je nach nächstem Zeichen wird in den IDLE oder den MARK Zustand übergengangen.

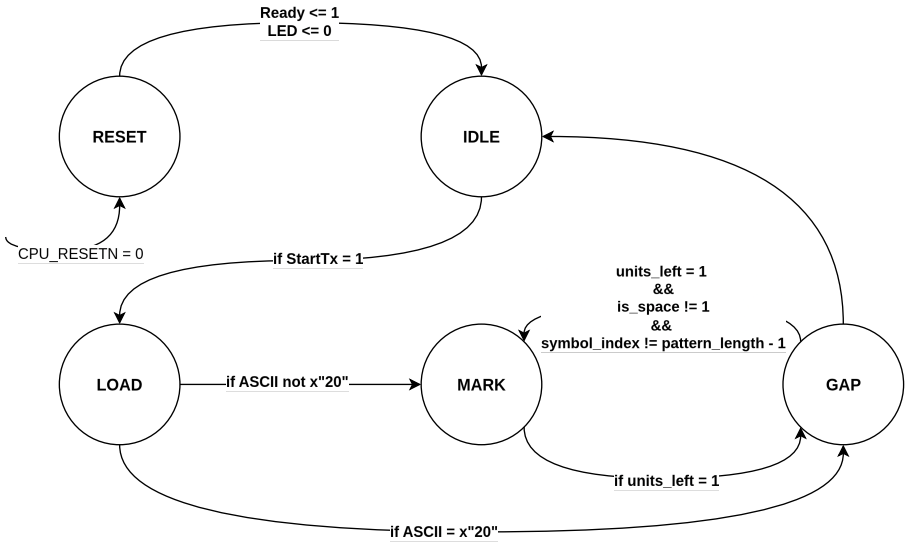

Fig. 3. FSM des MorseTx-Moduls

Empfangsmodul

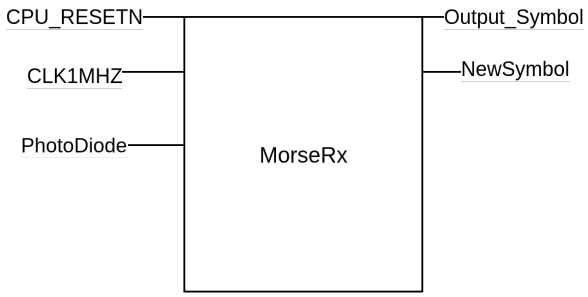

Fig. 4. Blockschaltbild des MorseRx-Moduls (links in, rechts out)

Das MorseRx-Modul zum Empfangen von Morse-Code-Nachrichten wurde, wie das MorseTx-Modul, mittels einer endlichen Zustandsmaschine (Fig. 5) umgesetzt. Im Gegensatz zum Sender sind zum Empfangen nur drei Zustände erforderlich. Die Zustände sind IDLE, MARK und GAP. Wann immer ein Reset ausgelöst wird werden alle Werte zurückgesetzt und die „State machine“ befindet sich die im IDLE. Der Pegel der PhotoDiode wird über den Flip Flop „pd_ff2“ und das signal „pd_prev“ abgetastet. Wenn ein Signal erkannt wird geht das Empfangsmodul in den MARK-Zustand über. Hier werden „high_ticks“ gezählt solange die PhotoDiode einen High-Pegel ausgibt. Ticks werden zu Units umgerechnet und nach einer erkannten Zeichen-Pause wird dem empfangenen Morsecode ein ASCII-Zeichen zugeordnet und es wird auf „Output_Symbol“ ausgegeben. Das Bestimmen der Länge der Pausen passiert im Zustand GAP. Zusätzlich gibt diese bei einer erkannten Wort-Lücke ein Leerzeichen aus. Wird erneut ein Signal auf der PhotoDiode erkannt springt das Empfangsmodul von GAP zu MARK.

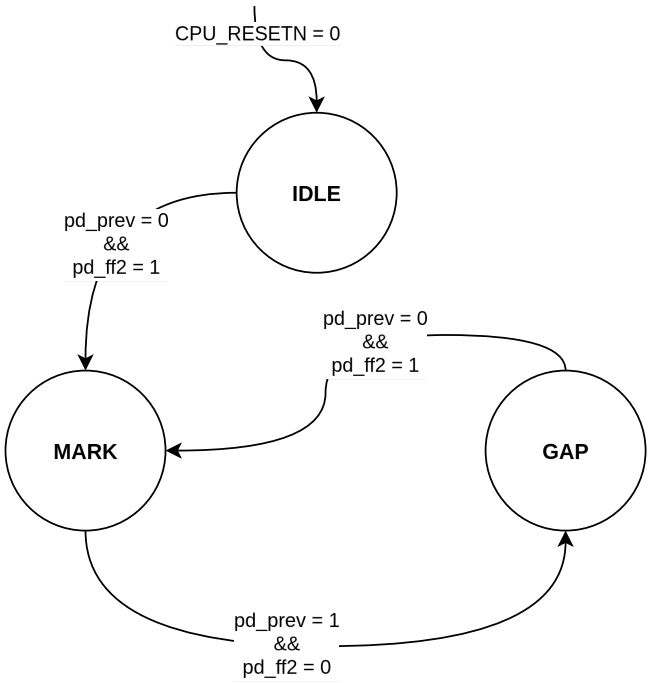

Fig. 5. FSM des MorseRx-Moduls

2 Umsetzung

Loopback_phys

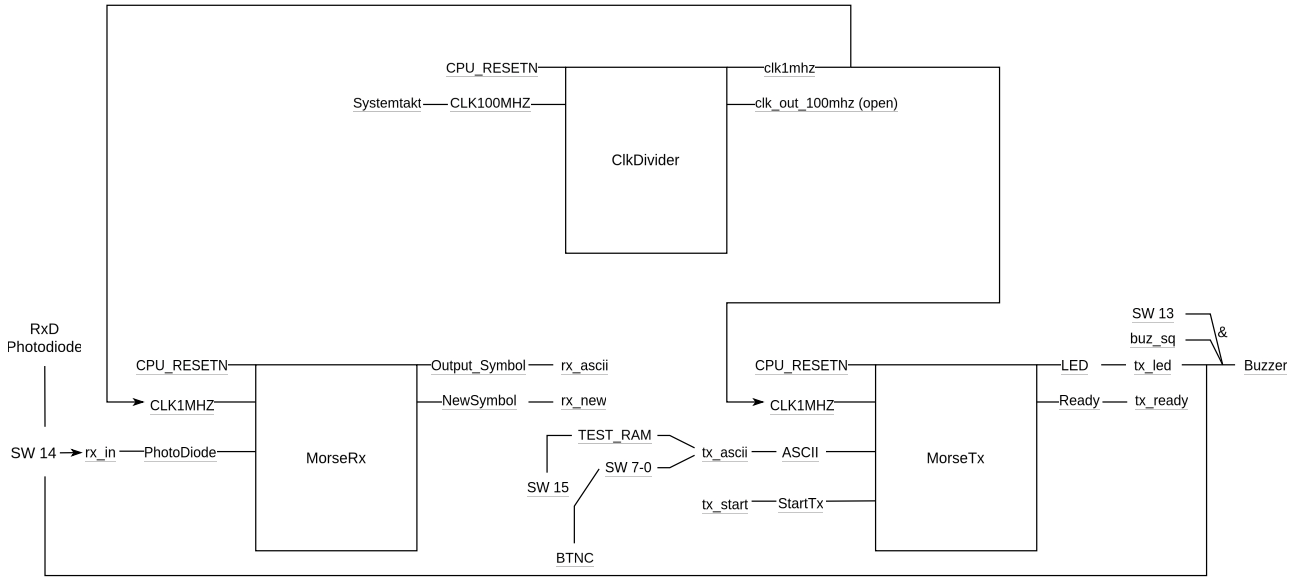

Fig. 6. Blockschaltbild Loopback_phys

Das Loopback_phys-Modul vereint die Funktionalitäten vom Taktteiler, MorseTx und MorseRx. Nach einem Reset werden alle Werte zurückgesetzt. Der Taktteiler erzeugt einen 1 MHz Takt aus dem 100 MHz Systemtakt. MorseTx und MorseRx arbeiten ausschließlich mit dem erzeugten 1 MHz Takt. Erkennt Loopback_phys, dass Schalter 15 auf dem FPGA eingeschaltet ist, wird als ASCII-Output das TEST_RAM Array periodisch ausgegeben. Ist Schalter 15 nicht betätigt kann man über die Schalter 7 bis 0 einen binären Wert einstellen und ihn als Morsecode senden, indem man die Taste BTNC betätigt. Mit Schalter 14 kann festgelegt werden, ob der Input von MorseRx über die LED auf dem PCB oder über den Output von MorseTx kommen soll. Das Modul speichert außerdem gesendete und empfangene Zeichen zwischen um diese auf der 7-Segment-Anzeige auszugeben. Schalter 13 erzeugt ein hörbares Summersignal, welches dem Output an der Morse-LED entspricht. LEDs 7 bis 0 zeigen außerdem empfangene ASCII-Zeichen in binär an. LED 8 zeigt tx_led, LED 9 zeigt tx_ready, LED 10 zeigt tx_start, LED 11 zeigt rx_new und LED 15 bis 12 zeigen Bits 7 bis 4 von tx_ascii an.

tb_Loopback_phys

In der Testbench des Loopback_phys-Moduls wird 200 ns gewartet. Dann wird CPU_RESETN auf 1 gesetzt um das Board zu starten. Auf Schalter 15 wird eine 1 geschrieben um, wie in „Loopback_phys“ beschrieben, das TEST_RAM Array periodisch ausgeben zu lassen. Optional kann auch Schalter 13 simuliert werden um den Buzzer zu simulieren.

Erklärung

Für die Ausarbeitung wurde generative KI genutzt. OPENAI's ChatGPT wurde genutzt, um Quellcode und Stichpunkte für die Ausformulierung von Sätzen zu generieren. KI wurde auch genutzt um sich Vorgänge aus dem Quellcode oder auf dem FPGA erklären zu lassen. Ergebnisse haben wir steuernd bearbeitet und übernehmen die volle Verantwortung für das Endergebnis.