

| Termin zajęć | Godzina | Imię i Nazwisko | Nr albumu |
|--------------|------------|----------------------|---------------|
| Czwartek TN | 7:30-10:30 | Łukasz Sztuka | 243168 |

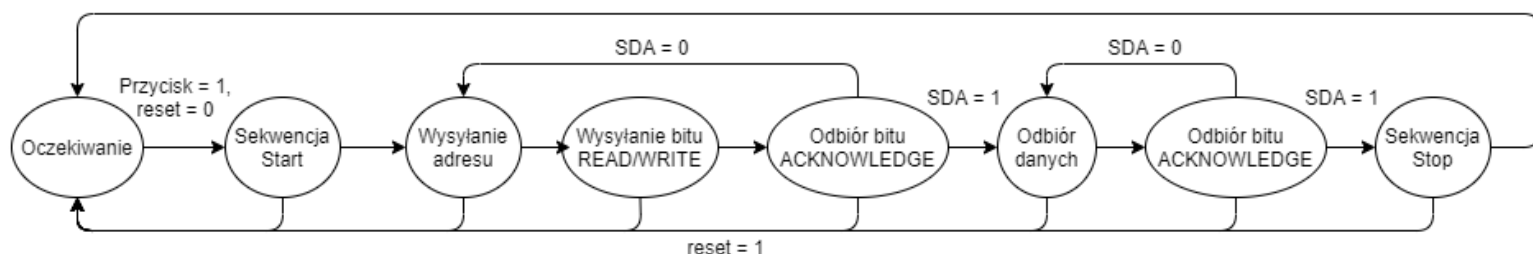
Temat: Komunikacja z modułem czasu rzeczywistego z pomocą magistrali I²C

1. Specyfikacja projektu

Celem projektu było opisanie układu w języku VHDL, realizującego połączenie z niskobudżetowym modułem czasu rzeczywistego MCP7940M przy pomocy magistrali szeregowej I²C. Moduł ten znajduje się na makiecie dydaktycznej kompatybilnej z programowanym przeze mnie układem Artix-7. Dla poprawnej pracy modułu, należało go skonfigurować w tryb master receiver.

2. Opis jednostki głównej

Zaprojektowany układ swoje działanie opiera głównie na automacie Mealy'ego oraz liczniku pomocniczym. Układ po uruchomieniu znajduje się w stanie "oczekiwanie", a sekwencja obioru informacji z modułu RTC, wywoływana jest kiedy na wejściu "przycisk" pojawi się stan logiczny "1". Pierwszym stanem jest "Sekwencja Start" w której układ zgodnie z standardem magistrali I²C ustawia stan niski na linii SDA kiedy na linii SCL wciąż jest stan wysoki. Następnie rozpoczyna się wysyłanie 7-bitowego adresu modułu (adres ten można znaleźć w notce katalogowej [1]). Adres pobierany jest z wejścia o tej samej nazwie, a następnie jest serializowany i zgodnie z taktami zegara przepisywany na linię SDA. W międzyczasie linia SCL podłączona zostaje do wejścia zegara. Kolejnym stanem jest wysłanie bitu READ, który definiuje czy moduł będzie wysyłał czy odbierał dane. Kolejnym stanem jest obiór bitu ACK, oznaczającego poprawność wykonanej transmisji danych. Wejście SDA ustawiane jest w stan wysokiej impedancji, aby możliwy był odbiór danych. Następnie przesyłana jest 8-bitowa ramka danych. Transmisja kończy się ponownym odbiorem bitu ACK i sekwencją końca transmisji zgodną z standardem magistrali I²C, czyli ustawieniem stanu wysokiego na linii SDA kiedy na linii SCL również jest stan wysoki. Po zakończeniu transmisji obie linie przesyłania danych są ustawione na wartość 1, a automat powraca do stanu "oczekiwanie". Algorytm automatu zaprezentowany został na Rys.1.



Rys. 1. Algorytm komunikacji

Lista wejść i wyjść przedstawiona została w Tab. 1. Konieczne było również dodanie trzech sygnałów pomocniczych, których dokładny opis znajduje się w Tab. 2.

| Nazwa | Rodzaj | | Opis |
|----------|--------|------------------|--|
| reset | in | std_logic | Pin restartu układu |
| clk | in | std_logic | Wejście zegara |
| adres | in | std_logic_vector | Wejście adresowe |
| przycisk | in | std_logic | Wejście wywołujące sekwencje komunikacji |
| scl | out | std_logic | Wyjście magistrali I2C (SCL) |
| sda | inout | std_logic | Wyjście magistrali I2C (SDA) |

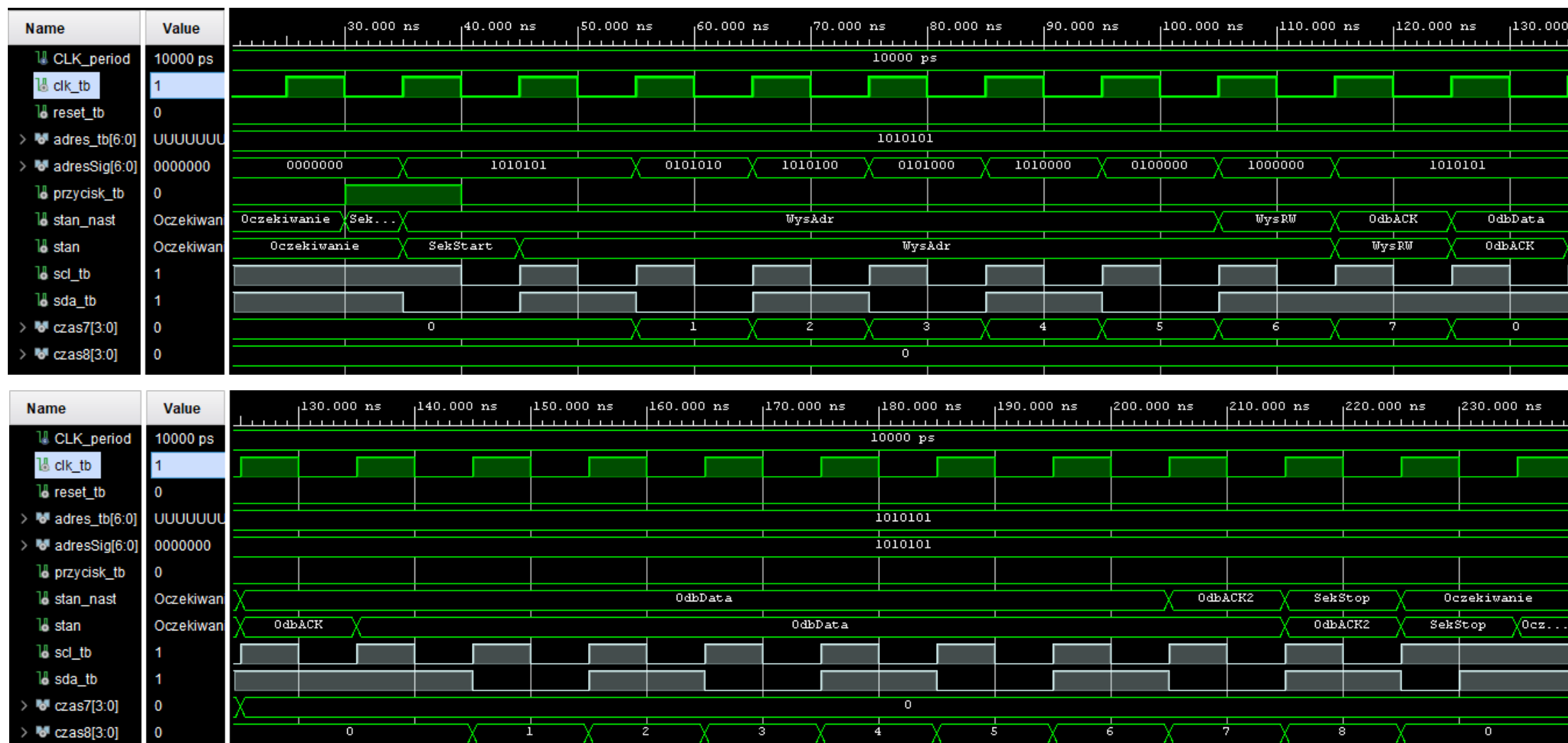
Tab. 1. Wejścia i wyjścia układu

| Nazwa | Rodzaj | Ilość bitów | Opis |
|----------|------------------|-------------|--|
| Czas7 | std_logic_vector | 4 | Przechowuje stan licznika wysyłania adresu |
| Czas8 | std_logic_vector | 4 | Przechowuje stan licznika odbioru danych |
| adresSig | std_logic_vector | 7 | Przechowuje wysyłany adres modułu RTC |

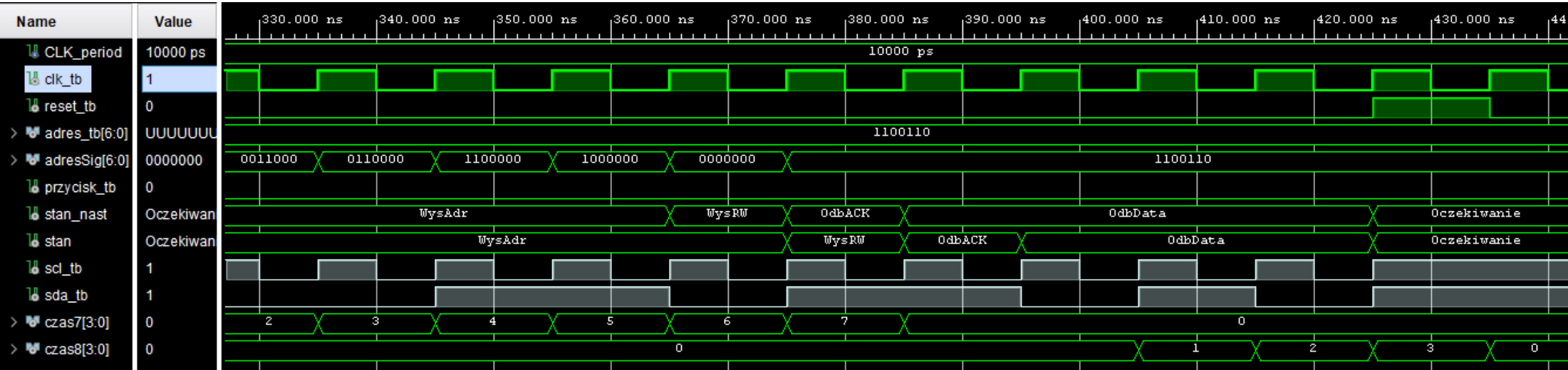
Tab. 2. Lista sygnałów pomocniczych

3. Opis jednostki testującej

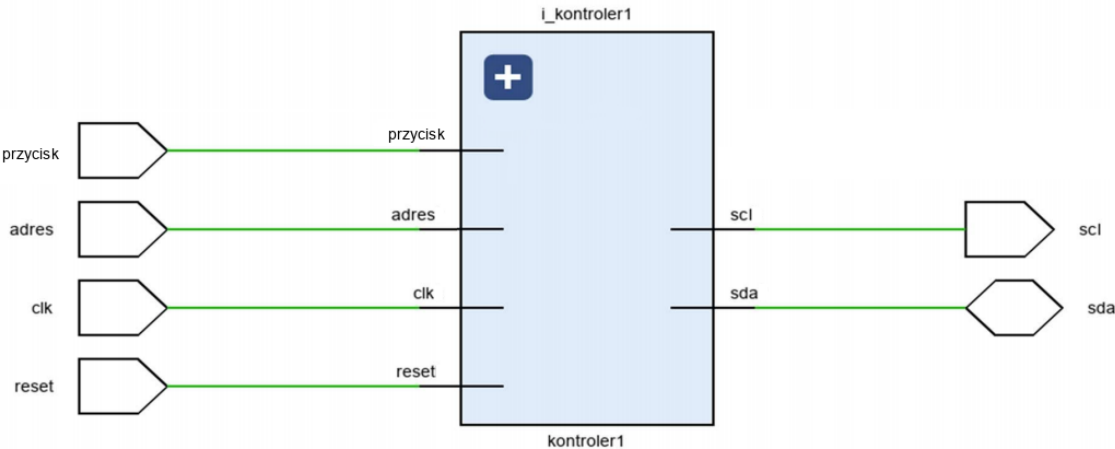
Test układu rozpoczyna się od jego zrestartowania, a następnie wywoływany jest główny algorytm podaniem wysokiego stanu logicznego na wejście "przycisk". Kolejnym krokiem jest odczekanie odpowiedniej ilości czasu do momentu aż układ wejdzie w stany w których odbiera dane. Następnie dane te są manualnie ustawiane na linii SDA, symuluje to działanie rzeczywistego połączenia z modułem. Na przebiegach wygenerowanych przez środowisko vivado możemy zaobserwować czy zachowanie układu jest poprawne. W kolejnej iteracji działania układu testowa jest możliwość wprowadzenia innego adresu układu RTC oraz poprawność resetu w trakcie działania układu. Później testom poddana zostaje możliwość odbioru innych danych. Podczas testów kluczowa jest synchronizacja czasowa układu testującego z jednostką testowaną. Przykładowe przebiegi sygnałów zaprezentowane zostały na rysunkach 2 i 3.



Rys. 2. Przebiegi obrazujące działanie układu



Rys. 3. Przebiegi obrazujące test restartu układu w trakcie działania



Rys. 4. Schemat ideowy układu