

Implementacja interfejsu 1wire w VHDL przy użyciu Spartan-3E oraz DS18S20

Krzysztof Cabała 210047

Kinga Wilczek 210063

4 maja 2016

Spis treści

1	Założenia projektowe	4
2	Wstęp teoretyczny	4
3	Podstawowe operacje	4
3.1	Inicjalizacja i reset	4
3.2	Zapis i odczyt	4
3.2.1	Zapis bitu	4
3.2.2	Odczyt bitu	5
4	Implementacja podstawowych operacji	5
5	Transmisja bajtu	6
6	Implementacja transmisji bajtu	6
7	Sekwencja konwersji i odczytu temperatury	7
8	Implementacja konwersji i odczytu temperatury	7
9	Algorytm double dabble	7
10	Implementacja double dabble	7

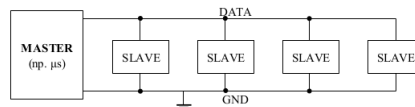
Spis rysunków

1	Przykładowe połączenie urządzeń	4
2	Diagram czasowy dla procedury inicjalizacji	4
3	Diagramy czasowe dla procedury zapisu i odczytu	5
4	Moduł obsługi podstawowych operacji bitowych	5
5	Maszyna stanów BusController	6
6	Symbol IOBuf	6
7	Moduł transmisji bajtu	7
8	Maszyna stanów ByteModule	7

1 Założenia projektowe

2 Wstęp teoretyczny

Interfejs 1wire jest interfejsem opracowanym przez firmę Dallas Semiconductor do komunikacji między dwoma lub większą liczbą urządzeń przy wykorzystaniu zaledwie jednej linii danych, linii GND (konieczne odniesienie dla poprawnego rozpoznawania stanów logicznych) oraz zasilania gVcc. W ramach oszczędności przewodów ogranicza się połączenia do dwóch linii, wtedy układ zasilany jest pasożytniczo z linii danych. Przykładowe połączenie przedstawia schemat:



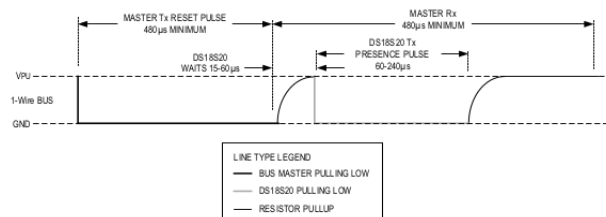
Rysunek 1: Przykładowe połączenie urządzeń

Wyróżnia się urządzenia typu Master (najczęściej mikrokontroler) oraz Slave (peryferia).

3 Podstawowe operacje

3.1 Inicjalizacja i reset

Każda próba komunikacji urządzeń master i slave musi zacząć się od sekwencji składającej się z sygnału reset, wysyłanego przez master, po którym następuje sygnał obecności układu slave. Sygnał reset to wymuszony stan 0 trwający przynajmniej $480\mu s$. Następnie master oczekuje na sygnał obecności innego urządzenia na linii. Następuje wówczas zwolnienie magistrali, co powoduje podciągnięcie jej do stanu wysokiego przez rezystor pull-up. Urządzenie slave wykrywa wówczas narastające zbocze lini i po upływie $15\mu s$ - $60\mu s$ sygnalizuje swoją obecność poprzez wymuszenie stanu niskiego na okres $60\mu s$ - $240\mu s$.



Rysunek 2: Diagram czasowy dla procedury inicjalizacji

3.2 Zapis i odczyt

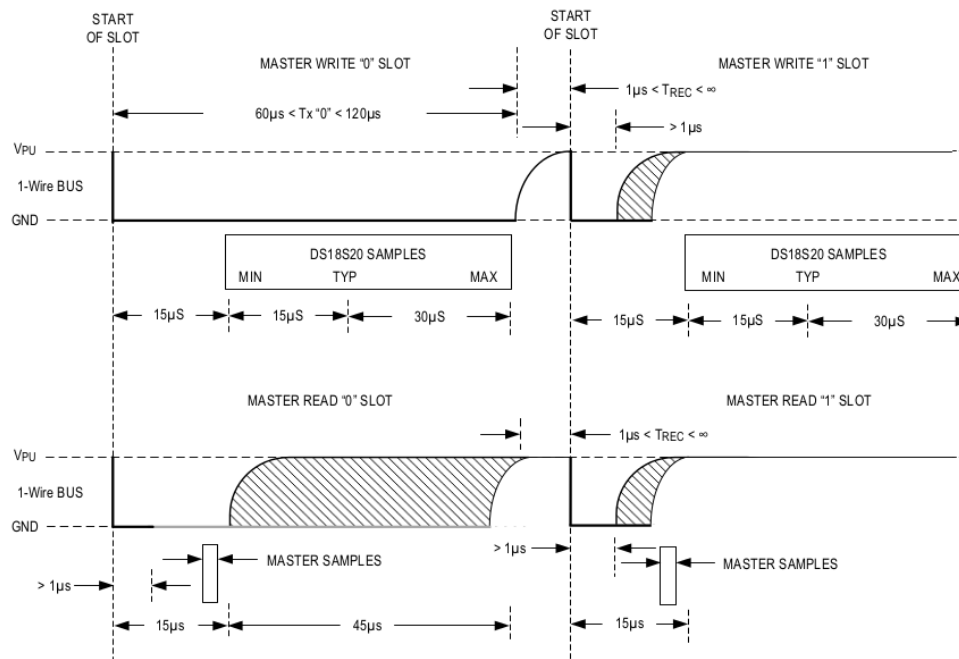
3.2.1 Zapis bitu

Operacje zapisu bitu realizowane są w ściśle określonych slotach czasowych. Długość jednego slotu wynosić zwykle $60\mu s$. Próbkowanie dokonywane jest mniej więcej w środku slotu celem uodpornienia na błędy. Pomiedzy kolejnymi operacjami wymagana jest przynajmniej $1\mu s$ odstępu.

Zapis rozpoczyna się wysterowaniem linii danych przez master na poziom niski. Zapis 0 wymaga utrzymania jej w tym stanie przez cały slot. Zapis 1 jest nieco bardziej skomplikowany. Master musi w czasie nie dłuższym niż $15\mu s$, ale nie krótszym niż $1\mu s$ zwolnić magistralę tak aby w momencie próbkowania (po $30\mu s$ od zbocza opadającego) była w stanie wysokim.

3.2.2 Odczyt bitu

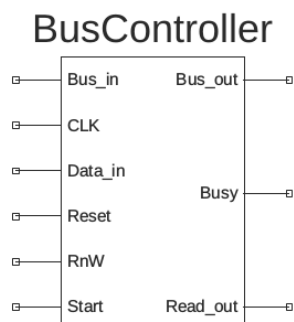
Odczyt bitu również wymaga $60\mu\text{s}$ slotu oraz $1\mu\text{s}$ przerwy. Odczyt rozpoczyna się wymuszeniem przez master stanu niskiego na linii danych na czas nie krótszy niż $1\mu\text{s}$ i zwolnienie jej (powrót do stanu wysokiego). Po tym sygnale sterowanie linią przejmuje urządzenie slave, wysyłające bit 0 lub 1. Slave po wykryciu zbocza opadającego wymusza stan niski (dla 0) lub utrzymuje wysoki (dla 1) linii danych. Sygnał musi być wtedy spróbkowany przez master. Przed upłynięciem czasu końca slotu magistrala zostaje zwolniona przez slave, co powoduje jej powrót do stanu wysokiego.



Rysunek 3: Diagramy czasowe dla procedury zapisu i odczytu

4 Implementacja podstawowych operacji

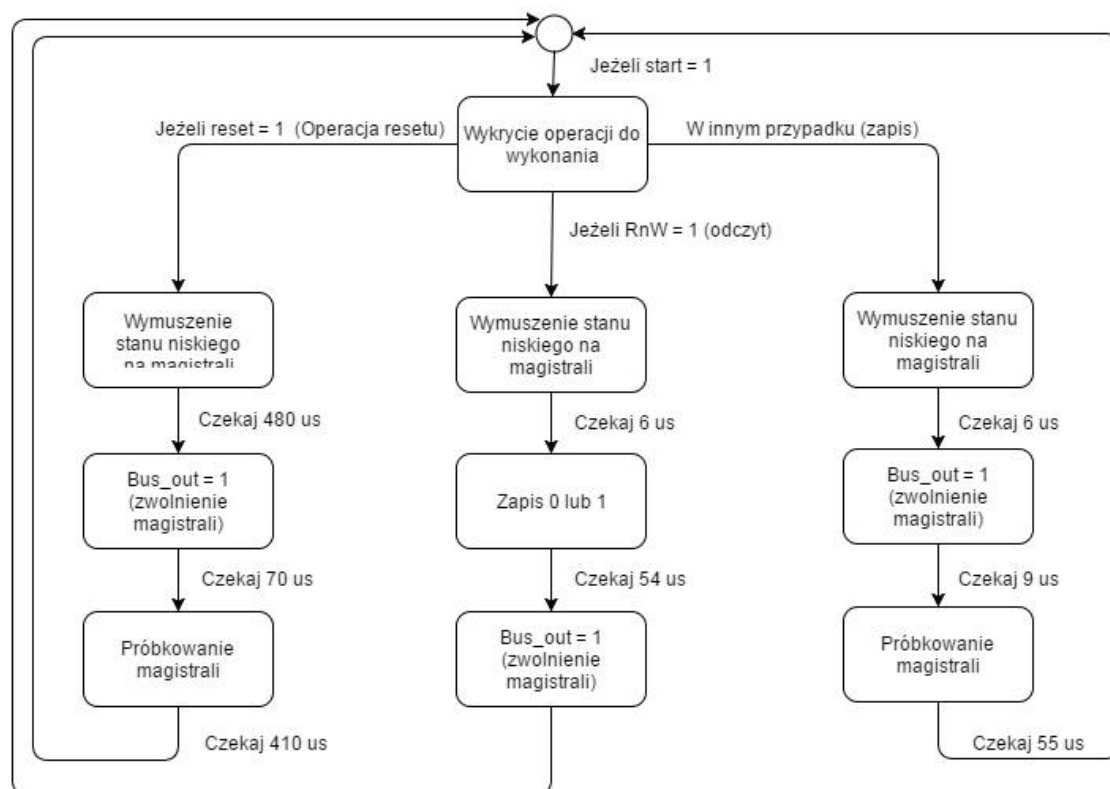
Za realizację podstawowych operacji odpowiada moduł BusController.



Rysunek 4: Moduł obsługi podstawowych operacji bitowych

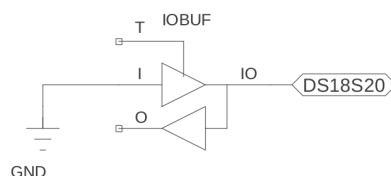
Poniżej przedstawiono schemat blokowy tego automatu.

Poprawna obsługa magistrali wymaga jej zwalniania poprzez ustawienie w stan wysokiej impedancji (podciągnięcie do V_{cc} przez rezystor pull-up). Służy do tego element IOBuf. Podanie logicznego zera na



Rysunek 5: Maszyna stanów BusController

wejście T otwiera bufor wyjściowy i przekazuje sygnał podawany na pin I (GND - Logiczne 0). Logiczne 1 na wejściu T ustawia linie w stanie wysokiej impedancji (zwalnia magistralę). Pin O służy do odczytu stanu magistrali.



Rysunek 6: Symbol IOBuf

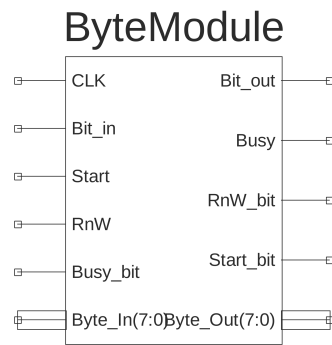
5 Transmisja bajtu

Komunikacja master - slave jest dwukierunkowa. Dane przesyłane są w formie bajtów, kolejność bitów określa zasada najpierw najmłodszy.

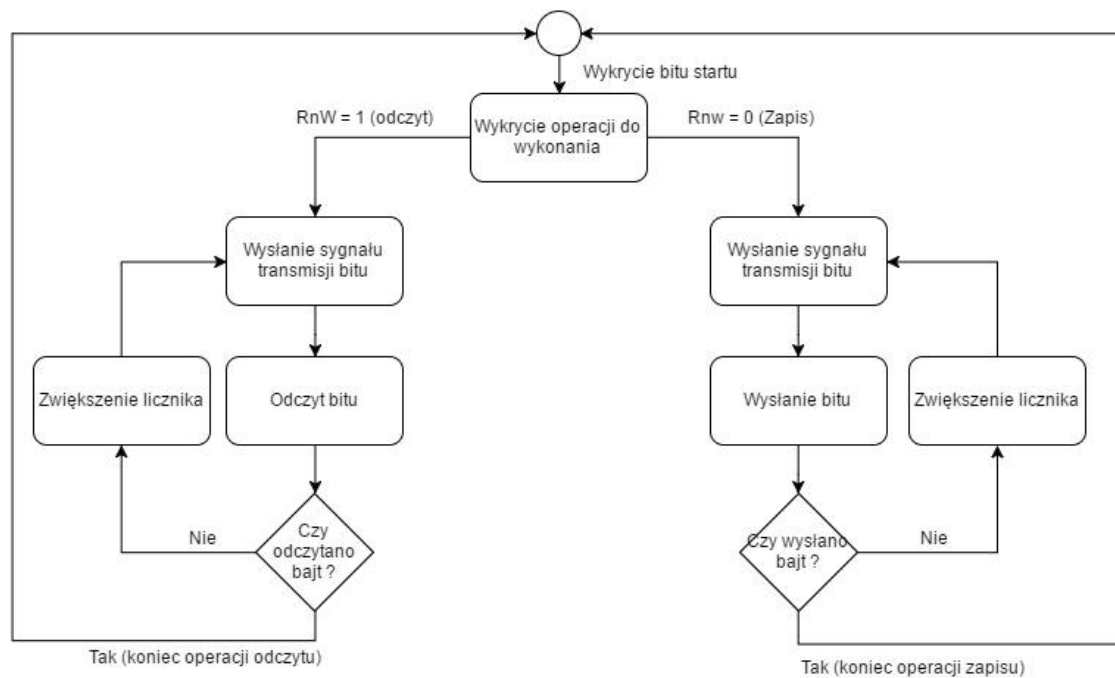
6 Implementacja transmisji bajtu

Transmisja bajtu realizowana jest poprzez następny moduł - ByteModule.

Automat zawiera po 4 stany na odczyt i zapis oraz jeden stan oczekiwania na sygnał startu.



Rysunek 7: Moduł transmisji bajtu



Rysunek 8: Maszyna stanów ByteModule

7 Sekwencja konwersji i odczytu temperatury

Aktualna odczyt teperatury zapisany jest w dwóch pierwszych bajtach pamięci Scratchpad. Po poprawnej inicjalizacji czujnika DS18S20 ich zawartość odpowiada temperaturze $+85^{\circ}\text{C}$. Sekwencję poleceń

8 Implementacja konwersji i odczytu temperatury

9 Algorytm double dabble

10 Implementacja double dabble