

Systemy wbudowane'18

14-18 maja 2018

Tematem tej listy jest projekt prostego procesora, komunikującego się za pomocą *magistrali trójstanowej* z innymi procesorami. Od urządzenia typu **master** procesory otrzymują polecenia, po czym po ich wykonaniu przetrzymują wynik do momentu otrzymania polecenia `DATA_REQ`, kiedy wynik jest umieszczany na magistrali – a zatem udostępniany wszystkim urządzeniom.

Z tego prostego opisu wynika, że magistrala jest sygnałem wspólnym dla wszystkich urządzeń, zatem należy dać możliwość ustawiania jej stanu przez wszystkie urządzenia. Rodzi to problem, gdyż w dwuwartościowej logice “0” oraz “1” nie ma możliwości, by jedno urządzenie “odłączyło się” od magistrali, pozwalając na jej wysterowanie jakiemś innemu. Stąd wprowadza się pojęcie *wysokiej impedancji* (oznaczane przez 'Z') jako trzecią wartość, jaką dane urządzenie może ustawić *po swojej stronie* magistrali. Jest to stan logicznie nieokreślony – zatem urządzenie ustawiające swój sygnał wyjściowy w stan wysokiej impedancji nie ustawiana magistrali w żadnym konkretnym stanie. Pozostałe urządzenia mogą zatem wysterować magistralę tak, jak to wynika z ich aktualnej potrzeby.

Oczywiście, jeśli protokół komunikacji, kontrolujący momenty, w którym urządzenia mogą sterować magistralą, jest błędny, doprowadzić to może do przypadku, gdy więcej niż jedno urządzenie ustawia “0” lub “1” na magistrali – co może prowadzić do konfliktu.

Rozwiąż poniższe zadania, identyfikując problem w nich zawarty i proponując rozwiązanie.

Zadanie 1 Przenalizuj działanie załączonego do tej listy kodu procesora. Dopisz funkcjonalności:

ACC (akumulacja kolejnych danych) – wysłanie tej komendy powoduje, że urządzenie *slave* będzie zliczało sumę kolejnych bajtów wysłanych do niego. Suma będzie przechowywana po zakończeniu przesyłania sekwencji bajtów i zwracana po otrzymaniu polecenia `DATA_REQ`.

CRC – obliczanie sumy CRC kolejnych bajtów. Po wysłaniu każdego bajtu urządzenie *slave* natychmiast odpowiada aktualną sumą CRC. Do wyliczania sumy CRC użyj jednostki z listy 4.

Zadanie 2 Uruchom kilka procesorów jednocześnie, podłącz je do wspólnej magistrali. W programie testowym zasymuluj urządzenie *master*, które tak będzie sterowało komunikacją, aby nie dochodziło do konfliktów na magistrali. Stwórz sytuację, w której dane pochodzące z jednego procesora są odbierane i przetwarzane przez drugi, z pominięciem jednostki *master*.