



Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Méréstechnika és Információs Rendszerek Tanszék

Rendszerarchitektúrák

Házi feladat

AXI - SPI perifériaillesztő

Kardos Bálint, ZI84PX
Murányi Péter, A74MW9
Konzulens: Raikovich Tamás

2018. május 12.

Tartalomjegyzék

1.	1
2.	1
2.1.	1
2.2.	1
2.3.	1
2.3.1.	1
2.3.2.	1
3. SPI modul működése	1
3.1.	1
3.2.	1
4. Szimuláció	1
4.1. BFM	1
4.2. EEPROM	1
4.3. Szimuláció eredménye	2
5. Forráskód	2
5.1. SPI és busz illesztő	2
5.1.1.	2
5.1.2.	2
5.2.	2
5.2.1.	2
5.3.	2
5.3.1.	2
6.	2
Hivatkozások	3

1.

2.

2.1.

2.2.

2.3.

2.3.1.

2.3.2.

3. SPI modul működése

3.1.

3.2.

4. Szimuláció

A létrehozott periféria működése szimulációval lett ellenőrizve. A szimulációhoz a Vivado beépített szimulátorát használtuk. A perifériához az AXI interfész felől egy AXI-LITE Master-t szimuláló BFM lett illesztve, míg az SPI oldalról egy Microchip EEPROM funkcionális verilog modellje lett illesztve. A szimuláció egy az EEPROM-ba való írást, majd onnan a beírt adatok kiolvasását valósítja meg.

A szimuláció 100MHz-s órajelet használ, a reset vonal induláskor 20ns ideig aktív alacsony állapotban van, majd visszatér magas szintre, ami elindítja az összes eszközt.

Az SPI modul /16 órejelosztást használ, amivel az SCK 6,25 MHz lesz.

4.1. BFM

4.2. EEPROM

Az eszközhöz egy Microchip 25AA010A EEPROM szimulációs verilog modellje van illesztve, hogy a valós működés is tesztelve legyen. Ez a modul egy 1kbit méretű flash EEPROM-ot valósít meg, valós időzítési értékekkel. Az eszköz leírása [ezen a linken](#) érhető el.

Az eszköz valós időzítéseket is szimulál, ezért például az SCK órajel nem lehet gyorsabb 10MHz-nél, ugyanis az eszköz nem lesz képes elég gyorsan reagálni az órajel változására. Ezt az SPI órajelosztójának megfelelő beállításával lehet elérni.

Hogy az eszközt írhasuk, először egy instrukciót kell neki küldeni ami engedélyezi az írást az eszközön belül. Ehhez egy külön írási ciklusra van szükségünk ahol \overline{CS} jelet aktív alacsonyra kell húzni, majd az írásengedélyező (WREN) parancsot kiküldve a \overline{CS} jelet egy órajelciklus idejére vissza kell engedni magas szintre. Ha ezt nem tesszük meg, nem fogunk tudni írni az eszközbe. Ezek után lehet ténylegesen írni az eszközt. Íráskor először egy 8-bites írás instrukciót kell kiküldeni, ez után jöhet a cím, szintén 8-biten, majd ezt követően kezdődik meg a tényleges adat kiküldése. Lehet bájtos, illetve burst módban írni, viszont burst módban maximum 16 bájtot lehet kiküldeni, ugyanis az EEPROM csak ekkora bemeneti bufferrel rendelkezik. Az írást a \overline{CS} magasra húzásával lehet befejezni, ekkor az írásengedélyező belső flag automatikusan 0-ra áll vissza. Az EEPROM verilog modelljében apró változtatásokat kellett eszközölni, ugyanis a valóságot szimulálva 5ms-ig késleltet mielőtt az adatot ténylegesen beírja az adattároló regisztereibe. Ezt hogy a szimuláció rövidebb legyen, és a kapott hullámformák átláthatóbbak legyenek 100ns-re csökkentettük.

4.3. Szimuláció eredménye

5. Forráskód

5.1. SPI és busz illesztő

5.1.1.

5.1.2.

5.2.

5.2.1.

5.3.

5.3.1.

6.

Hivatkozások

- [1] *SPI leírása* https://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus
2018. május 12.
- [2] *AXI Lite leírása és időzítési diagramja*
https://www.xilinx.com/support/documentation/ip_documentation/axi_lite_ipif/v3_0/pg155-axi-lite-ipif.pdf
2018. május 12.
- [3]
2018. május 12.
- [4]
2018. május 12.
- [5]
- [6]
2018. május 12.