

SPI vevő tervezése Verilog nyelven

(Szántó Péter, Csordás Péter – BME MIT)

Az alábbiakban egy általános SPI vevő modul blokkvázlatát tervezzük meg, ami később a konkrét hardverek (FPGA fejlesztő kártya, illesztendő SPI eszköz) ismeretében testre szabható. E dokumentum megértéséhez szükséges az SPI protokoll ismerete.

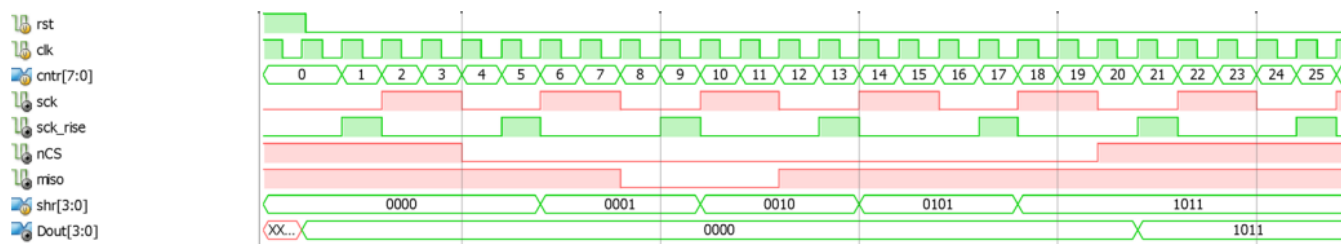
Általános blokkvázlat készítése

Egy SPI vevő modul feladatai és interfésze:

- Az **SCK** (SPI clock) *kimeneten* megfelelő frekvenciájú órajel biztosítása
- A **nCS** (Chip Select) *kimenet* megfelelő vezérlésével az SPI eszköz (slave) kiválasztása. Itt 'n' a jel negálására utal: a slave nCS=0 esetén van kiválasztva.
- Az **MISO** (Master In Slave Out) *bemeneten* az órajelhez szinkronizáltan az adat beolvasása.
- A **Dout** *busz kimeneten* a beolvasott adat továbbítása feldolgozásra
- A modul természetesen megkapja a rendszer **clk** órajelét és **rst** reset jelét.

Egyetlen adat vételi ciklus alatt a modulnak folyamatosan biztosítani kell az órajelet, a nCS=0 jellel kiválasztani a Slave eszközt, a MISO bemeneten a Slave eszköz specifikációjában megadott számú adatbitet fogadni, nCS=1 jellel az eszközt „elengedni”, végül a következő adat olvasás előtt legalább a specifikációban adott ideig várni. A Dout kimenetet az egyes olvasások végén kell frissíteni, így mindig konzisztens adat áll rendelkezésre. Ha az olvasás során „értéktelen” biteket is küld a slave eszköz, ezeket a vevő kimenetén nem kell megjeleníteni.

A tervezési szempontok megértését segítően itt közöljük egy SPI vevő hullámformáit. Az alább definiált fontos paraméterek értéke a példa esetében p=1, r=2=4.

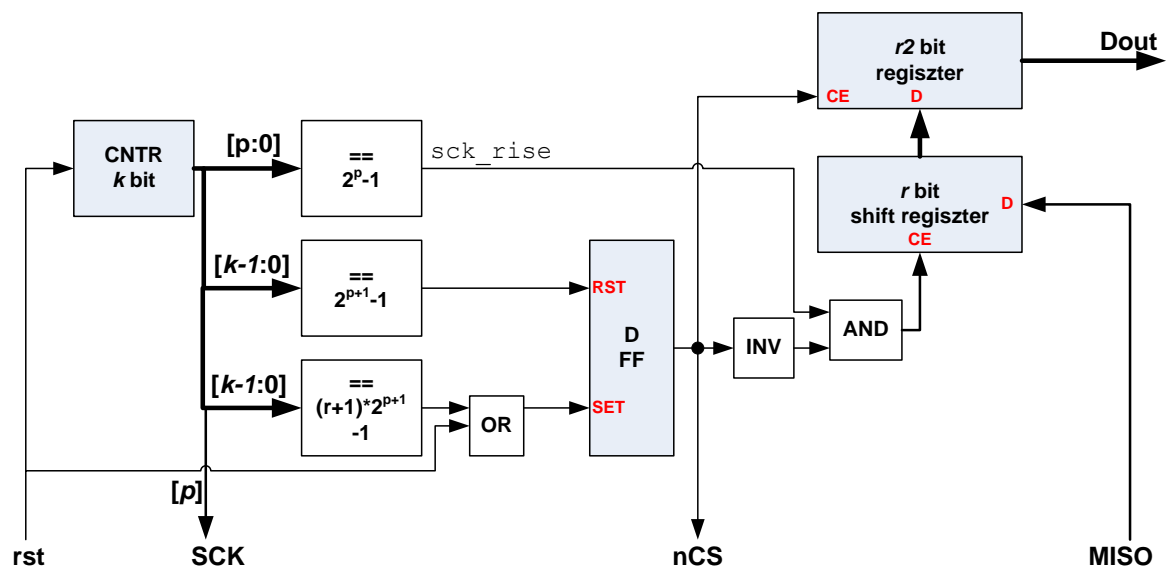


Tervezési megfontolások:

- A vevő modul folyamatosan fogadja az adatokat, **egy belső számláló (cntr) segítségével a teljes vételi ciklust ütemezzük**. Nem célunk a lehető leggyakoribb olvasás, így megengedhető, hogy egy olvasási ciklus hossza épp a számláló periódusideje legyen.
- A vételi ciklus elején történjen a tényleges kommunikáció, majd pedig az előírt két kommunikáció közötti várakozás! Ha utóbbi teszi ki a ciklus jelentős részét, **a cntr számláló bit szélességét (k)** a rendszer órajel frekvencia (f_{clk}) és a szükséges várakozási idő (t_w) viszonya szabja meg: $k = \lceil \log_2(f_{clk} * t_w) \rceil$
- Az SPI órajel periódusidejére általában széles határok közt mozoghat, így mindig beállítható olyan p paraméter, amire $f_{SCK} = f_{clk} / 2^{(p+1)}$ a slave eszköz specifikációját teljesíti. Így módon **SCK generálható cntr p. bitjének kivezetésével**. A megadott példában a rendszer órajelet négyvel osztjuk (p=1).

- A továbbiakban feltételezzük, hogy a slave egység az SCK lefutó élére írja az adatot a MISO vonalra, tehát az általunk írt modulnak felfutó élre célszerű azt mintavételeznie. A **mintavételezést engedélyező sck_rise jel** előállításához a *cntr* utolsó $p+1$ bitjét érdemes vizsgálni. A példában akkor engedélyezzük a mintavételezést, ha a számláló utolsó két bitje 2'b01, hisz a következő clk órajellel együtt fog SCK is felfutni.
- A slave adatlapja specifikálja, hogy engedélyezés után mennyi időnek kell eltelni, az első érvényes adatbit beolvasása előtt, azaz esetünkben SCK első felfutó éléig. Ha ez az idő rövidebb, mint SCK félperiódusa, akkor a **kiválasztó jel lefutását célszerűen szinkronizálhatjuk az átviteli ciklus első SCK lefutó éléhez**, ami $cntr = 2^{p+1} - 1$ után következik be.
- A slave által egy kommunikáció során továbbított bitek számát jelölje r . Ennek megfelelően a tervünkben egy r bites shift regiszternek kell szerepelnie, a **slave „elengedését”** ($nCS=1$) pedig $cntr (2^{p+1}) * (r + 1) - 1$ értékével vezérelhetjük, hisz az eszközt a ciklus kezdete után 1 SCK periódussal választottuk ki, és az engedélyezést r darab SCK periódusig kell fenntartani.
- Annak érdekében, hogy a modul kimenete mindig „érvényes” legyen, a shift regiszter értékét (illetve annak $r2$ értékes bitjeit) egy külön kimeneti regiszterbe kell menteni akkor, amikor az adat konzisztens, vagyis épp amikor nincs adatforgalom, azaz $nCS=1$.

A fenti szempontok alapján az alábbi blokkvázlat adódik. Fontos, hogy ennek minden elemét megértse!



Jelölések:

- Kéesszürke doboz: sorrendi hálózat. Minden FF a rendszer órajelről működik (ami az ábrán nincs külön feltüntetve).
- Fehér doboz: kombinációs logika.
- Piros feliratok: az adott funkcionális elem portja:
 - RST: FF reset bemenet, 0-ba állítja a FF-t.
 - SET: FF set bemenet, 1-be állítja a FF-t.
 - CE: FF clock enable bemenet.
 - D: FF adat bemenet.
- SCK: SPI órajel kimenet.

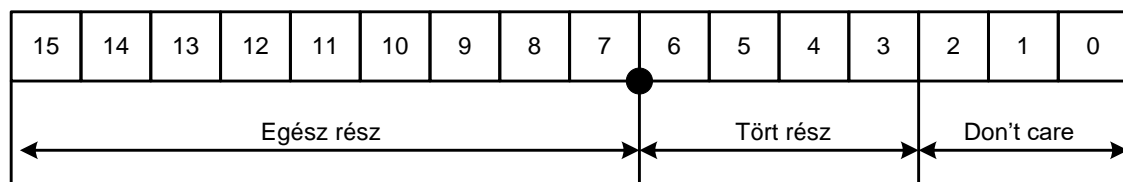
- nCS: SPI kiválasztójel kimenet.
- MISO: SPI soros adatbemenet.
- p , k , r és $r2$ a fent részletezett paraméterek

A TMP121 hőmérő SPI interfésze

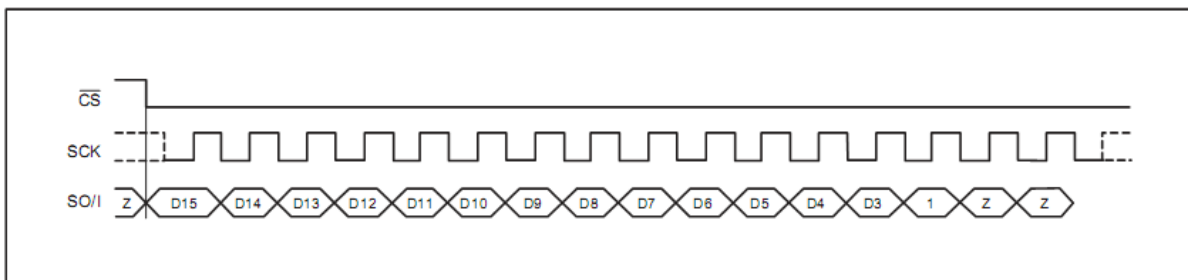
A TMP121 a **mért hőmérsékletet 16 biten, kettes komplementes formában** továbbítja úgy, hogy **az alsó 3 bit nem tartalmaz érdemi információt** (az alábbi táblázatban ezen bitek értéke 3'b000). Az adatformátumot az alábbi táblázat mutatja.

TEMPERATURE (°C)	DIGITAL OUTPUT ⁽¹⁾ (BINARY)	HEX
150	0100 1011 0000 0000	4B00
125	0011 1110 1000 0000	3E80
25	0000 1100 1000 0000	0C80
0.0625	0000 0000 0000 1000	0008
0	0000 0000 0000 0000	0000
-0.0625	1111 1111 1111 1000	FFF8
-25	1111 0011 1000 0000	F380
-55	1110 0100 1000 0000	E480

A táblázat 5. és 6. sorából láthatóan a legelső értékes bit változása 0,0625 °C-os hőmérsékletváltozásnak felel meg. Mivel $1/0,0625=16$, a 13 értékes bitből tehát az alsó 4 a hőmérséklet törtrésze, a maradék – felső 9 – az egészrésze.



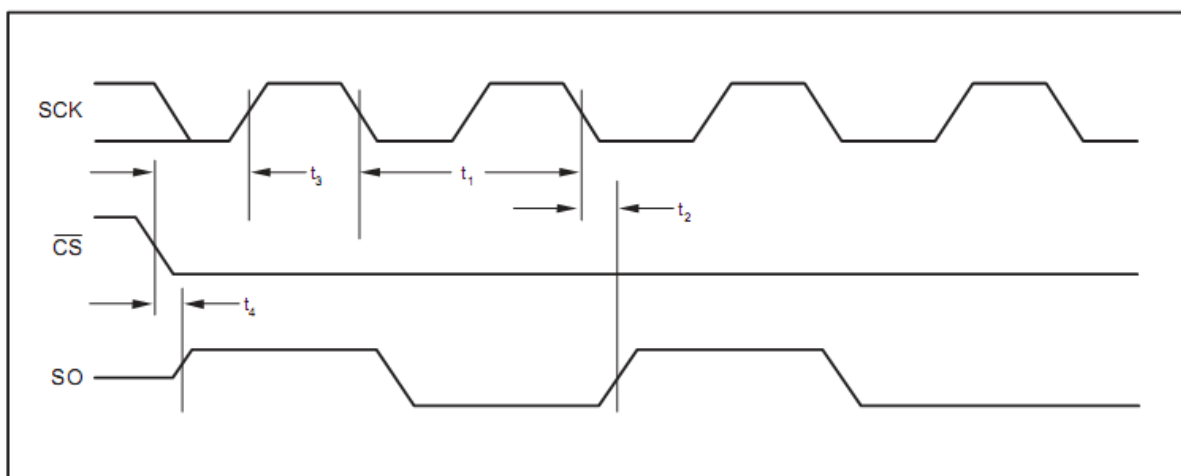
A mért értéket a TMP121-ből SPI interfészen lehet kiolvasni. Az olvasás hullámformáját az alábbi ábra mutatja.



Az SPI kiválasztó jel (nCS) aktív ideje (0 szint) alatt 16 órajel (SCK) ütemben az előző pontban ismertetett 16 bit jelenik meg az IC soros adatkimenetén (SO), az alsó 3 bit értéke érdektelen. Az IC

adatlapja szerint (és az ábrából láthatóan) az SPI kommunikáció során a TMP121 az SCK órajel lefutó élére shifteli ki a 16 bitet.

Az SPI interfész időzítési követelményeit ugyancsak az adatlapból tudjuk megállapítani, az alábbi információk segítségével.



PARAMETER		MIN	MAX	UNITS
SCK Period	t_1	100		ns
SCK Falling Edge to Output Data Delay	t_2		30	ns
CS to Rising Edge SCK Set-Up Time	t_3	40		ns
CS to Output Data Delay	t_4		30	ns
CS Rising Edge to Output High Impedance	t_5		30	ns

Összefoglalva a diagram és a táblázat adatait:

- **t1: Az SPI órajel (SCK) periódusideje minimum 100 ns** (azaz frekvenciája max. 10 MHz).
- **t2: Az SPI órajel (SCK) lefutó éle után az adat legfeljebb 30 ns múlva jelenik meg a kimeneten.** Azaz ennél korábban nem mintavételezhetjük az FPGA-ban a TMP121 IC kimenetét.
- **t3: Az SPI kiválasztó jel lefutó éle és az ezt követő első SPI órajel (SCK) felfutó él között legalább 40 ns-nak el kell telnie.**
- **t4: Az SPI kiválasztó jel lefutó éle után az első adat legfeljebb 30 ns múlva jelenik meg a kimeneten.**
- **t5: Az SPI kiválasztó jel felfutó éle után legfeljebb 30 ns ideig hajtja meg az IC az SPI adatvonalat, ez után már biztosan nagy impedanciás állapotba kerül a kimenet.**

Az adatlapból az is kiderül, hogy az IC a hőmérséklet értéket akkor konvertálja (azaz állítja elő a digitális értéket) amikor az SPI kiválasztó jel 1 értékű. **Egy konverzió ideje** – ez ugyancsak az adatlapból deríthető ki – **320 ms**, a hőmérséklet folyamatos frissítéséhez tehát garantálnunk kell, hogy a nCS jel legalább ennyi ideig magas állapotú.