

11. mérés

Programozható perifériák mérése

Bevezetés

A beágyazott rendszerek, egyszerű vezérlők, különböző felhasználói és adatkapcsolati perifériákon keresztül kommunikálnak környezetükkel. A *felhasználói interfészek* tipikusan az adatbeviteli (kapcsolók, nyomógombok, billentyűzet, egér) és adatkiviteli (LED-ek, kijelzők, megjelenítők) funkciókat biztosítják, míg az *adatkapcsolati interfészek* az adatátvitelt valósítják meg az egyes modulok és berendezések között különböző szabványok szerint. Az interfészek használata speciális *perifériaegységeken* keresztül történik, amelyek a *vezérlővonalakat* működtetik, megvalósítva az *átviteli protokoll fizikai*, esetleg *adatkapcsolati* szintjét. Ezeket a periféria egységeket közvetlenül hardverben, vagy egy programozható vezérlőegységet használva szoftverben is megvalósíthatjuk. A mérés – illetve az arra történő felkészülés – során megismerkedünk két elterjedten használt kommunikációs protokollal, majd ezeket (a megfelelő konfigurációval) hardverben implementálva egy perifériát (hőmérőt) illesztünk a beágyazott egységünkhöz, illetve a PC-hez.

A mérés célja

A mérés célja, hogy (1) a hallgatók behatóan megismerjék az UART és SPI kommunikációs protokollt, majd ezeket hardverben implementálják, (2) ezek segítségével egy tipikus periféria interfészt valósítsanak meg, (3) mindeközben elmélyítsék a CAD rendszer segítségével történő hardver tervezéshez kapcsolódó ismereteiket.

Felkészülés a mérésre

Ez a mérés épít a korábbi „*Sorrendi hálózat vizsgálata*” mérésben tanultakra, illetve az ott átvett korábbi anyagokra, így ezeket ismételve át!

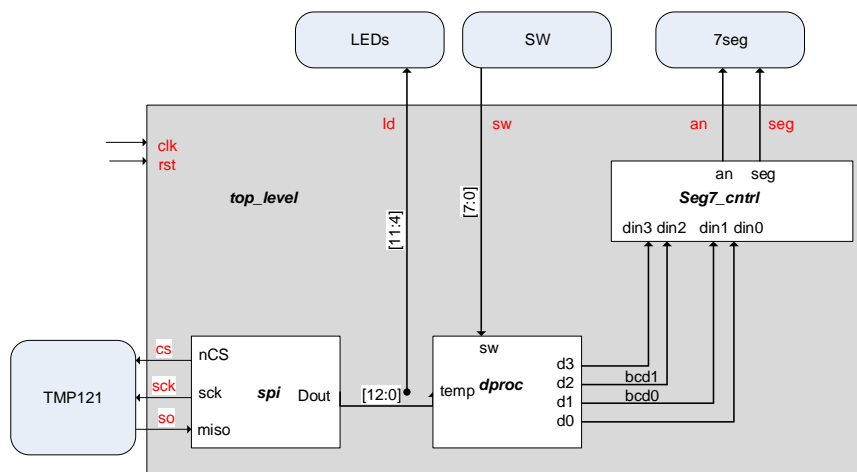
Figyelmesen olvassa el a tárgy honlapján elektronikusan elérhető dokumentumokat!

- A „*Soros átviteli protokollok ismertetése*” dokumentum SPI interfészt taglaló része a méréshez nélkülözhetetlen információkat tartalmaz, az UART ismertetése csak érdekesség.
- „*SPI vevő tervezése Verilog nyelven*”: a mérésen megvalósítandó SPI vevő tervezési lépéseit mutatja be, illetve ismerteti az illesztendő hőmérő legfontosabb adatait.

Tekintse át a *mérési feladatokat*, az *ellenőrző kérdéseket*!

Mérési feladatok

A mérésen megvalósítandó teljes hardver struktúráját az alábbi ábra tükrözi. Minden modul használja a rendszer reszet és órajel bemenetét, ezt az átláthatóság érdekében külön nem jelöltük.

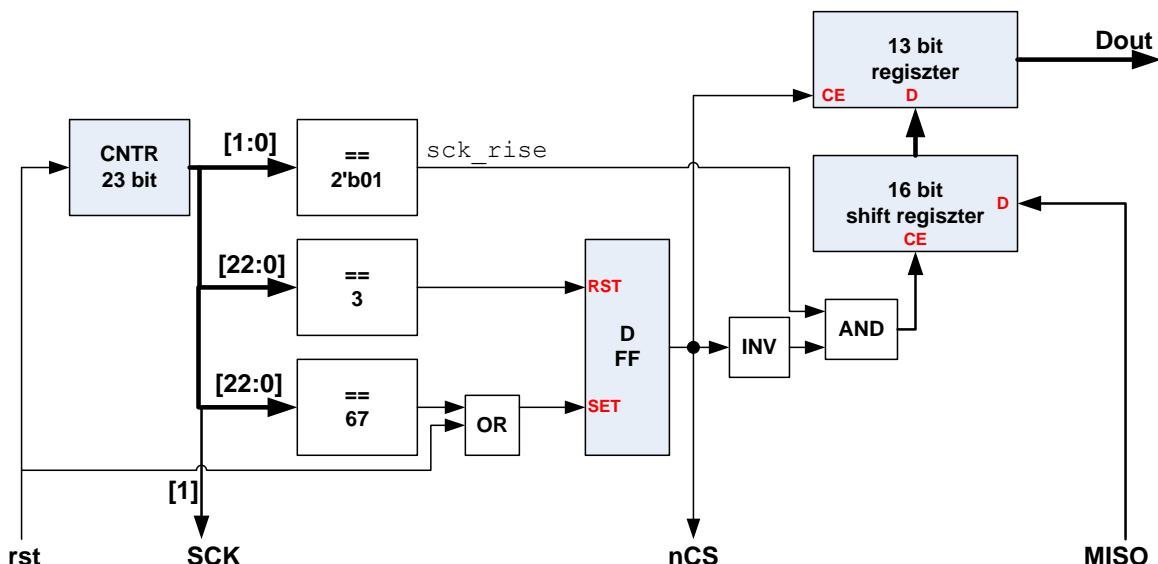


A tárgy honlapról letölthető vázfilmek a következők:

- A *top_level* a házi feladatként elkészítendő *uart* modul kivételével az összes almodult példányosítja.
- *Seg7_ctrl*: a korábbi mérésen elkészített hétszegmenses kijelző vezérlő modul kódja azzal a kiegészítéssel, hogy a nem BCD kódokat is hasznosítjuk: decimális 13 bemenet hatására mínuszjel jelenik meg, minden más nem BCD bemenet esetén pedig kikapcsol az érintett szegmens. A tizedespont a *din1* szegmensnél folyamatosan ég, míg a többinél kikapcsolt állapotban van. Az engedélyező jel generálásának a korábban tanultnál hatékonyabb módját mutatjuk be. Keresse meg a kódban az ezeket az „újításokat” megvalósító sorokat!
- *spi*: a mérés során elkészítendő SPI vevő váza. Kiinduló állapotban *nCS* = 1, *sck* = 0, *Dout* = 13'b0. A modul kimenetére a hőmérő adatának 13 értékes bitje kerüljön! Az egész részt reprezentáló 8 bit értéke jelenik meg a LED-eken.
- A *tb_spi_temp* test bench fájl az elkészítendő modul tesztelését szolgálja. A mérés során ezt értelmezni és használni kell.
- A *dproc* modul állítja elő a hétszegmenses dekóder 4 digitjének megfelelő 4 db négy bites BCD kódot. A modul a hőmérséklet mellett a kapcsolók állapotát olvassa be: az alsó 7 bit értékét a 8. bit (*sw[7]*) állapotától függően vagy közvetlenül a kimeneti bináris-BCD átalakítóra kapcsolja, vagy kivonja a hőmérséklet értékből. Utóbbi esetben az eredő hőmérséklet kerül a bináris-BCD átalakítás után a kimenetre. A mérés során e modulban kis módosítást kell végezni. Fontos a teljes kód áttekintése, megértése. Ellenőrizze, hogy adódnak a kimenetek, melyik digit mit reprezentál! (Pl.: melyik digiten jelenik meg a mért hőmérséklet előjele?)
- A *pins.ucf* fájl tartalmazza a blokkvázlatnak megfelelő lábkiosztást. Módosításra nincs szükség. (Az érdekesség kedvéért tekintse át a fájlt!)

1. SPI vevő implementációja, szimulációja, letöltése

Az „*SPI vevő tervezése Verilog nyelven*” dokumentumban leírtak alapján, válasszuk az SCK órajelet 4 MHz-re, vegyük figyelembe, hogy a rendszer órajele 16 MHz, az illesztendő periféria konverziós ideje 130 ms, és a 16 továbbított adatbitből a felső 13 bit értékes! Mindezek alapján a megvalósítandó hardver blokkvázlata az alábbi:



- 1.1. Értelmezze, hogyan adódott a fenti blokkvázlat! Teljesül a TMP121 specifikációja? Pontosan milyen hosszú a rendszer egy teljes vételi ciklusa? Mi a szerepe a 3 komparátornak? (Miért épp a megadott komparálási értékek szerepelnek?) Mi biztosítja a konzisztens kimenetet?
- 1.2. Készítse el az *spi* modult, majd szimulációval ellenőrizze a működést! **A kapott test bench fájlt nézze át, értse meg!** A jegyzőkönyvben válaszoljon a következő kérdésekre: Egy általános tesztfájl esetén mit mondhatunk a gerjesztő jelek, illetve a kimenetek típusáról? Teljesül ez a jelen fájlnál is? Írja le röviden, hogy valósul meg a hőmérő szimulációja! Megengedett –e, hogy egy jelet több always blokkban írjunk? Milyen frekvenciájú a szimuláció órajele? Mi a szimulált hőmérő által küldött adat? Változatlanul megjelenik ez az *spi* modul kimenetén?

Természetesen futtassa a szimulációt, értelmezze szövegesen is a kapott hullámformát!

- 1.3. Letöltés után ellenőrizze, valóban működik-e az elkészült modul! A kapcsolók segítségével állítson be 42 fok ofszet hőmérsékletet! Magyarázza el tapasztalatait!

2. Kiegészítő feladatok

- 2.1. A *dproc* modulban keresse meg és értelmezze a bináris-BCD átalakítást végző kódrészt! Mi a konverzió algoritmus? Hány órajel szükséges egy konverzióhoz? Milyen esemény hatására indul a konverzió? Írja le, mi a szerepe a *data_old*, *data_out*, *data_conv* és *data_high* regisztereknek! Milyen értékeket tárolnak ezek a konverzió elején illetve végén?
- 2.2. Nézze meg, hogyan történik a tizedes jegy számítása! Miért kapunk a szintézisnél figyelmeztető üzenetet? (Segítség: írja fel az elvégzett műveleteket binárisan “páron”!)

Ellenőrző kérdések

1. Mik az SPI interfész jellemzői? Milyen jelek szükségesek a kommunikációhoz és mik ezen jelek szerepe?
2. Rajzolja fel a TMP121 kommunikáció idődiagramját!
3. Hány vezeték kell 4 periféria SPI buszon történő használatához, mik ezek?
4. Mit jelent az előjeles, fixpontos számábrázolás? Mennyi az egyes bitek helyiértéke?

5. Milyen formátumban küldi a TMP121 a hőmérséklet adatokat? Milyen hőmérsékletnek felel meg a 0x0C88 hexadecimális érték?
6. Írja fel a -42h értéket 2-es komplement bináris formában 8 bites egész számformátumban! Mekkora decimális értéknek felel meg a bitminta, ha előjeles, fixpontos, két törtrész bittel rendelkező értéknek tekintjük?
7. Adja meg egy 16 bites, engedélyezhető balra léptető shift regiszter Verilog kódját. A modul bemenetei: órajel (clk), engedélyezés (en); a kimenete a shift regiszter aktuális állapota (shr).
8. Adja meg egy 8 bites, engedélyezhető jobbra léptető shift regiszter Verilog kódját. A modul bemenetei: órajel (clk), engedélyezés (en); a kimenete a shift regiszter aktuális állapota (shr).
9. Adja meg egy 20 bites felfele számláló számláló Verilog kódját. A modul bemenetei: órajel (clk), reset (rst); a kimenete a számláló aktuális állapota (cntr).
10. Adja meg (assign értékadással) azon Verilog kódsort, amely képes egy 14 bites kettes komplement érték abszolút értékének kiszámítására. A bemeneti jel neve legyen data_2comp, az abszolútérték jel neve pedig data_abs. Ez utóbbi jelet deklarálja is.