

Project AC

Besoi David, Bolat Tayfun si Avram Sorin

November 23, 2025

1 SPI Bridge

1.1 Descriere generala

Modulul `spi_bridge` implementeaza o interfata de comunicatie seriala sincrona de tip SPI (Serial Peripheral Interface) in configuratia Slave.

Acest modul serveste drept "pod" (bridge) intre un Master extern (ex: un microcontroler) si logica interna a perifericului FPGA. Scopul principal este de a serializa datele trimise de sistem catre exterior si de a deserializa datele primite de la Master, asigurand totodata sincronizarea semnalelor intre domeniile de ceas.

1.2 Specificatii protocol

Modulul este configurat conform urmatoarelor specificatii standard SPI:

- **Rol:** Slave (Receptioneaza ceasul si semnalul de selectie).
- **Mod de operare:** Mod 0 (CPOL = 0, CPHA = 0).
 - CPOL = 0: Linia de ceas (`sclk`) este in starea LOW (0) cand este inactiva.
 - CPHA = 0: Datele sunt esantionate (citite) pe primul front (crescator) si schimbate pe al doilea front (descrescator).
- **Ordinea bitilor:** MSB First (Cel mai semnificativ bit este transmis primul).
- **Dimensiune date:** Transferuri pe 8 biti (Octet).

1.3 Interfata Modulului (Porturi)

1.3.1 Semnale de Sistem (Clock Domain Intern)

Nume Port	Tip	Descriere
<code>Clk</code>	Input	Ceasul sistemului intern (10 MHz). Toata logica interna este sincrona cu acest ceas.
<code>Rst_n</code>	Input	Reset asincron, activ pe nivel jos (Active Low).

1.3.2 Interfata SPI (Externa)

Nume Port	Tip	Descriere
Sclk	Input	Serial Clock. Ceasul generat de Master.
Cs_n	Input	Chip Select (Active Low). Activeaza interfata slave.
Mosi	Input	Master Out Slave In. Linia de date de intrare (scriere in periferic).
miso	Output	Master In Slave Out. Linia de date de iesire (citire din periferic). Iesire tri-state (Hi-Z cand CS e inactiv).

1.3.3 Interfata Interna (Catre Periferic)

Nume Port	Tip	Descriere
Data_out	Input	Byte-ul de date pregatit de periferic pentru a fi trimis catre Master.
Data_in	Output	Byte-ul de date receptionat complet de la Master.
Byte_sync	Output	Semnal de tip "strobe" (activ un ciclu de ceas clk) care indica finalizarea receptiei unui nou octet valid.

1.4 Descrierea Functionarii

1.4.1 Sincronizarea Semnalelor (CDC)

Desi specificatia mentioneaza ca frecventa `sclk` este identica cu `clk` (10MHz), faza semnalelor poate diferi. Pentru a evita metastabilitatea si pentru a detecta corect fronturile, toate semnalele de intrare SPI (`sclk`, `cs_n`, `mosi`) sunt trecute prin un sincronizator cu doi bistabili.

Logica interna se bazeaza pe detectia fronturilor semnalelor sincronizate:

- `sclk_rising_edge`: Momentul esantionarii datelor de intrare (MOSI).
- `sclk_falling_edge`: Momentul schimbarii datelor de iesire (MISO).

1.4.2 Procesul de Receptie (MOSI)

1. Cat timp `cs_n` este activ (LOW), la fiecare detectare a frontului crescator al `sclk`, bitul prezent pe linia `mosi` este shiftat in registrul `data_in_reg` (LSB).
2. Numaratorul `bit_count` se incrementeaza.
3. Cand `bit_count` ajunge la valoarea 7 (s-au primit 8 biti), se genereaza pulsul `byte_sync`, semnaland sistemului intern ca `data_in` contine un octet valid.

1.4.3 Procesul de Transmisie (MISO)

1. **Initializare:** La activarea Chip Select (`cs_n_falling_edge`), octetul prezent la intrarea `data_out` este salvat in `data_out_latch`.
2. **Nota:** Primul bit (MSB, bitul 7) este plasat imediat pe linia `miso` pentru a fi gata de citire de catre Master la primul front al ceasului.
3. **Deplasare:** La fiecare detectare a frontului descrescator al `sclk`, linia `miso` este actualizata cu urmatorul bit din `data_out_latch`, conform indexului dat de `bit_count`.
4. Daca `cs_n` devine inactiv (HIGH), linia `miso` intra in stare de inalta impedanta (`1'b1`).

2 Instruction Decoder (instr_dcd)

2.1 Descriere Generala

Modulul `instr_dcd` (Instruction Decoder) reprezinta "creierul" logicii de comunicatie a perifericului. Acesta functioneaza ca un Automat Finit (FSM - Finite State Machine) care interpreteaza sevenetele de biti receptionate prin interfata SPI si le traduce in semnale de control pentru bancul de registri interni.

Rolul sau principal este de a decodifica pachetele de date in doua etape (Setup si Data), determinand tipul operatiei (Citire/Scriere), adresa tinta si gestionand fluxul de date bidirectional.

2.2 Structura Protocolului de Comanda

Decodorul interpreteaza primul octet al fiecarei tranzactii ca fiind un **Octet de Instruc-tiune**. Semnificatia bitilor este descrisa mai jos:

Bit	Denumire	Descriere
7	R/W	Selecteaza tipul operatiei: 1 = Write (Scriere in registru) 0 = Read (Citire din registru)
6	High/Low	Selecteaza octetul din registrul de 16 biti tinta: 1 = Partea High (bitii [15:8]) 0 = Partea Low (bitii [7:0]) (Actioneaza ca offset de adresa +1 sau +0).
5:0	Base Addr	Adresa de baza a registrului (0 - 63).

Table 1: Structura Octetului de Instructiune

2.3 Interfata Modulului (Porturi)

2.3.1 Semnale de Sistem

Nume	Tip	Descriere
Clk	Input	Ceasul sistemului (10 MHz). Logica FSM este sincrona cu acest ceas.
Rst_n	Input	Reset asincron (Active Low). Aduce automatul in starea IDLE.

2.3.2 Interfata catre SPI Bridge (Slave)

Nume	Tip	Descriere
Byte_sync	Input	Semnal puls (strobe). Indica faptul ca un nou octet a fost primit complet de la <code>spi_bridge</code> sau ca un octet a fost transmis.
Data_in	Input	Octetul de date primit de la Master (prin SPI Bridge).
Data_out	Output	Octetul de date ce va fi trimis catre Master (rezultatul unei citiri).

2.3.3 Interfata catre Bancul de Registri

Nume	Tip	Descriere
Read	Output	Semnal de control (Active High, puls). Initiaza o citire la adresa <code>addr</code> .
Write	Output	Semnal de control (Active High, puls). Initiaza o scriere la adresa <code>addr</code> .
Addr	Output	Adresa finala calculata (6 biti) pentru accesul la memorie/registri.
Data_read	Input	Datele citite din registri (raspuns la comanda read).
Data_write	Output	Datele ce trebuie scrise in registri (asociate cu comanda write).

2.4 Automatul de Stari (FSM)

Logica de control este implementata printr-un FSM cu doua stari principale:

2.4.1 Starea S_IDLE (Faza de Setup)

- Comportament:** Starea implicita. Asteapta semnalul `byte_sync` de la SPI Bridge.
- Actiune:** Cand `byte_sync` devine activ, octetul prezent pe `data_in` este salvat in registrul intern `instr_reg`.
- Tranzitie:** Trece automat in starea S_SETUP.

2.4.2 Starea S_SETUP (Faza de Date)

- **Comportament:** Asteapta al doilea `byte_sync`, care indica sosirea octetului de date (pentru scriere) sau finalizarea transmiterii (pentru citire).
- **Decodificare Adresa:** Se calculeaza adresa fizica:

$$Adresa_Finala = Adresa_Instructiune[5 : 0] + Bit_Selectie[6]$$

Aceasta permite adresarea individuala a jumatatilor din registrii de 16 biti.

- **Executie Comanda:**

- Daca Bit 7 = 1 (Scriere): Se activeaza `write` si se pun datele pe `data_write`.
- Daca Bit 7 = 0 (Citire): Se activeaza `read`. Bancul de registri furnizeaza datele pe `data_read`, care sunt salvate in `data_out_reg`.

3 REGS (Memoria de Configurare)

3.1 Descriere Generala

Modulul `regs` reprezinta memoria de configurare a perifericului. Este compus dintr-o serie de celule de memorie (flip-flops) organizate sub forma unor registri adresabili. Desi datele sunt transferate pe magistrala de 8 biti, registrii pot avea dimensiuni variabile (1, 8 sau 16 biti).

3.2 Harta Registrilor

Spatiul de memorie este adresabil pe 6 biti (0x00 - 0x3F). Registrii mai mari de 8 biti ocupă două adrese consecutive (Little Endian).

Nume	Addr	Acc	Bit	Descriere
PERIOD	0x00	R/W	16	Defineste perioada semnalului PWM (nr. cicli ceas). 0x00: LSB, 0x01: MSB.
CNT_EN	0x02	R/W	1	Activare numarator intern (1=Pornit).
COMPARE1	0x03	R/W	16	Valoare prag pentru primul front PWM. 0x03: LSB, 0x04: MSB.
COMPARE2	0x05	R/W	16	Valoare prag secundara (pentru mod nealiniat). 0x05: LSB, 0x06: MSB.
CNT_RST	0x07	W	1	Reset sincron numarator (auto-curatare).
CNT_VAL	0x08	R	16	Read-Only. Valoarea curenta a numaratorului.
PRESCALE	0x0A	R/W	8	Divizorul de ceas ($2^{PRESCALE}$).
UP/DOWN	0x0B	R/W	1	1 = Incrementare (Up), 0 = Decrementare.
PWM_EN	0x0C	R/W	1	Activeaza iesirea fizica a semnalului PWM.
FUNC	0x0D	R/W	2	Bit 0: Stanga/Dreapta, Bit 1: Aliniat/Nealiniat.

Table 2: Harta Registrilor Perifericului

3.3 Detalii Implementare

- **Scrierea (Write):** Este sincrona cu frontul crescator al ceasului. Scrierea unui registru de 16 biti necesita două tranzactii separate (LSB și MSB).
- **Citirea (Read):** Implementata folosind logica combinationala. Cand semnalul `read` este activ, multiplexorul selecteaza datele corespunzatoare adresei.
- **Auto-Clear:** Registrul `counter_reset` se reseteaza automat pe 0 in urmatorul ciclu de ceas dupa ce a fost scris cu 1.

4 Counter (Unitatea de Baza de Timp)

4.1 Descriere Generala

Modulul `counter` reprezinta "inima" perifericului generator de PWM, oferind baza de timp necesara functionarii acestuia.

Acet modul este compus din două elemente functionale distincte conectate in cascada:

1. **Prescaler-ul:** Un divizor de frecventa care incetineste ritmul de numarare.
2. **Numaratorul Principal:** Un contor pe 16 biti care incrementeaza sau deacrementeaza valoarea curenta.

4.2 Specificatii Functionale

- **Rezolutie:** 16 biti (Valori intre 0 si 65535).
- **Scalare (Prescaling):** Divizor programabil pe 8 biti.
- **Directie de numarare:** Configurabila, Crescator (Up) sau Descrescator (Down).
- **Controlul Perioadei:** Limita superioara de numarare este setabila prin registrul `period`.

4.3 Interfata Modulului (Porturi)

4.3.1 Semnale de Sistem

Nume	Tip	Descriere
<code>Clk</code>	Input	Ceasul sistemului. Toate tranzitiile sunt sincrone cu acest ceas.
<code>Rst_n</code>	Input	Reset asincron global (Active Low). Aduce numaratorul la 0.

4.3.2 Semnale de Control (Dinspre Registri)

Nume	Tip	Descriere
En	Input	Enable. Daca este 0, contorul ingheata la valoarea curenta.
count_reset	Input	Synchronous Reset. Reseteaza fortat valoarea la 0 la urmatorul front de ceas.
upnotdown	Input	1 = Numarare Crescatoare ($0 \rightarrow Period$) 0 = Numarare Descrescatoare ($Period \rightarrow 0$)
prescale	Input	Valoarea divizorului de ceas (8 biti).
period	Input	Valoarea maxima de numarare (TOP).

4.4 Descrierea Functionarii Detaliate

4.4.1 Etapa de Scalare (Prescaler)

Pentru a permite controlul fin al duratei, exista un contor secundar (`prescale_cnt_reg`) care numara de la 0 pana la valoarea setata pe intrarea `prescale`. Cand acest contor atinge valoarea `prescale`, el se reseteaza si genereaza un semnal `prescale_tick`.

Frecventa de actualizare a numaratorului principal este:

$$F_{counter} = \frac{F_{clk}}{prescale + 1}$$

4.4.2 Logica Numaratorului Principal

Numaratorul principal evolueaza doar la aparitia semnalului `prescale_tick`:

- **Modul Crescator (`upnotdown = 1`):** Secventa este $0 \rightarrow 1 \rightarrow \dots \rightarrow period \rightarrow 0$.
- **Modul Descrescator (`upnotdown = 0`):** Secventa este $\dots \rightarrow 2 \rightarrow 1 \rightarrow 0 \rightarrow period$.

5 PWM_GEN (Generatorul de Semnal)

5.1 Descriere Generala

Modulul `pwm_gen` reprezinta etajul final al perifericului, responsabil pentru generarea efectiva a semnalului digital modulat in durata (PWM). Acesta primeste starea curenta a numaratorului si configuratia din registri, comparandu-le continuu pentru a decide starea logica a iesirii `pwm_out`.

5.2 Moduri de Operare

Comportamentul este dictat de registrul `FUNCTIONS`:

1. Aliniat la Stanga (Left Aligned):

- Semnalul porneste in starea HIGH (1) la inceputul perioadei (`count = 0`).

- Comuta in LOW (0) cand numaratorul atinge valoarea `compare1`.

2. Aliniat la Dreapta (Right Aligned):

- Semnalul porneste in starea LOW (0).
- Comuta in HIGH (1) cand numaratorul atinge valoarea `compare1` si ramane asa pana la final.

3. Nealiniat (Centrat/Definit intre limite):

- Semnalul este HIGH (1) doar in intervalul definit de `compare1` si `compare2`.
- Relatie: $compare1 \leq count_val < compare2$.

5.3 Interfata Modulului

Nume	Tip	Descriere
<code>pwm_en</code>	Input	Bit de activare a iesirii PWM (Master Switch).
<code>Period</code>	Input	Perioada semnalului (limita <code>count_val</code>).
<code>Functions</code>	Input	Registru de configurare a modului.
<code>Compare1</code>	Input	Valoare de prag principala (Duty Cycle/Start).
<code>Compare2</code>	Input	Valoare de prag secundara (Stop pentru modul Nealiniat).
<code>Count_val</code>	Input	Valoarea curenta a numaratorului.
<code>pwm_out</code>	Output	Semnalul PWM final, sintetizat si sincronizat.

5.4 Sincronizarea Iesirii (Glitch-Free)

Pentru a asigura un semnal curat, fara oscilatii tranzitorii (glitches) care pot aparea in circuitele combinacionale pure, rezultatul comparatiei interne este trecut printr-un **bistabil D (Flip-Flop)**. Iesirea `pwm_out` se actualizeaza doar pe frontul crescator al ceasului `clk`, introducand o intarziere de exact 1 ciclu de ceas, dar garantand stabilitatea semnalului.