

# Proiectarea si Implementarea unui Periferic PWM Controlat prin Interfata SPI

Avarvarei Andrei-Alexandru  
Ilie Eduard Cristian  
Echipa 3

14 decembrie 2025

## Cuprins

<b>1</b>	<b>Introducere</b>	<b>2</b>
<b>2</b>	<b>Arhitectura generala a sistemului</b>	<b>2</b>
<b>3</b>	<b>Analiza detaliata a implementarii</b>	<b>2</b>
3.1	Interfata fizica si sincronizarea semnalelor (spi_bridge.v) . . . . .	2
3.2	Decodificarea instructiunilor (instr_dcd.v) . . . . .	2
3.3	Banca de registre si logica de auto-clear (regs.v) . . . . .	2
3.4	Numaratorul si prescaler-ul (counter.v) . . . . .	3
3.5	Generatorul PWM (pwm_gen.v) . . . . .	3
<b>4</b>	<b>Corectarea modulului top.v</b>	<b>3</b>
<b>5</b>	<b>Simulare si validare</b>	<b>3</b>
<b>6</b>	<b>Concluzii</b>	<b>4</b>

# 1 Introducere

Acest document descrie soluțiile adoptate pentru proiectarea și implementarea unui periferic hardware PWM (Pulse Width Modulation), care poate fi configurat printr-o interfață SPI. Proiectul a fost realizat conform specificațiilor temei, având ca scop principal crearea unui sistem modular, configurabil și fiabil.

Perifericul permite controlul complet al semnalului PWM, inclusiv perioada, factorul de umplere, modul de aliniere și direcția de numărare, prin scrierea în registre interne accesibile unui master SPI extern. Documentația se concentrează pe detaliile implementării și pe provocările apărute în timpul integrării sistemului.

## 2 Arhitectura generala a sistemului

Sistemul este organizat modular și este compus din mai multe blocuri functionale, fiecare având o responsabilitate bine definită. Aceasta abordare a permis dezvoltarea și testarea individuală a modulelor, urmată de integrarea acestora în modulul de top.

Fluxul de date este următorul: semnalele seriale sunt recepționate prin interfața SPI, sunt decodificate în comenzi interne, stocate în banca de registre și utilizate ulterior de numărator și de generatorul PWM pentru sinteza semnalului de ieșire.

## 3 Analiza detaliata a implementarii

### 3.1 Interfata fizica si sincronizarea semnalelor (`spi_bridge.v`)

Modulul `spi_bridge` reprezintă interfața directă cu exteriorul și implementează protocolul SPI în modul `CPOL=0, CPHA=0`. Datele sunt esantionate pe frontul crescător al semnalului `sclk`, iar ieșirea `miso` este actualizată pe frontul descrescător.

Semnalele `sclk` și `cs_n` sunt asincrone față de ceasul intern al sistemului, motiv pentru care sunt tratate cu atenție pentru a evita probleme de metastabilitate. Recepția unui byte complet este semnalizată prin impulsul `byte_sync`, utilizat de etajele superioare pentru delimitarea comenzilor SPI.

### 3.2 Decodificarea instructiunilor (`instr_dcd.v`)

Modulul `instr_dcd` este responsabil de interpretarea fluxului de date provenit de la `spi_bridge`. Decodificarea se face folosind o structură secvențială cu două etape: recepționarea header-ului, care conține tipul operației și adresa, urmată de recepționarea datelor în cazul unei operații de scriere.

Semnalele de control `read` și `write` sunt generate sub formă de impulsuri de un singur ciclu de ceas, pentru a preveni scrieri multiple neintenționate în banca de registre.

### 3.3 Banca de registre si logica de auto-clear (`regs.v`)

Modulul `regs` gestionează stocarea tuturor parametrilor de configurare ai perifericului. Registrele de 16 biți sunt accesate prin două operații succesive pe magistrala SPI, una pentru octetul inferior și una pentru octetul superior.

Un registru special este cel de resetare software a număratorului. Acesta este implementat cu o logică de tip auto-clear, astfel încât, după activarea resetului printr-o scriere

SPI, semnalul este dezactivat automat de hardware dupa un numar redus de cicluri de ceas. Aceasta solutie reduce incarcarea magistralei si simplifica controlul din software.

### 3.4 Numaratorul si prescaler-ul (counter.v)

Modulul counter reprezinta baza de timp a sistemului. Acesta suporta numarare crescatoare sau descrescatoare, resetare la atingerea valorii period si utilizarea unui prescaler configurabil.

Prescaler-ul este implementat eficient folosind deplasari logice pe biti, conform relatiei:

$$Limit = 2^{prescale}$$

Aceasta metoda este optima din punct de vedere al resurselor hardware utilizate.

### 3.5 Generatorul PWM (pwm\_gen.v)

Modulul pwm\_gen utilizeaza valoarea curenta a numaratorului si registrele de comparatie pentru a genera semnalul PWM. Sunt implementate mai multe moduri de functionare, incluzand alinierea la stanga, alinierea la dreapta si modul nealiniat intre doua valori de comparatie.

Pentru siguranta sistemului, iesirea PWM este conditionata de semnalul pwm\_en. Atunci cand acest semnal este dezactivat, iesirea este fortata la nivel logic zero, prevenind aparitia de impulsuri nedorite in timpul configurarii.

## 4 Corectarea modulului top.v

Fisierul top.v primit initial continea mai multe probleme structurale care impiedicau functionarea corecta a sistemului.

Prima problema a fost redeclararea porturilor de intrare si iesire ca semnale interne de tip wire. In Verilog, porturile sunt implicit de tip wire, iar redeclararea lor poate conduce la erori de compilare sau la conectari incorecte intre module.

A doua problema majora a fost conectarea incompleta a modulului spi\_bridge. Semnalele esentiale pentru comunicarea SPI, respectiv byte\_sync, data\_in si data\_out, nu erau conectate catre modulele instr\_dcd si regs. Din aceasta cauza, comenzile SPI nu erau decodificate, registrele nu erau programate, iar perifericul ramanea in stare inactiva.

In implementarea finala, aceste probleme au fost corectate prin eliminarea redeclararilor ilegale ale porturilor si prin conectarea completa a interfetei SPI intre module, fara a modifica interfata externa a modulului top. Astfel, compatibilitatea cu testbench-ul primit a fost pastrata.

## 5 Simulare si validare

Validarea functionala a fost realizata prin simulare, folosind testbench-ul primit. Valorile necunoscute observate la inceputul simularii sunt normale si apar din cauza initializarii semnalelor inainte de aplicarea resetului global.

Dupa configurarea corecta a registrelor prin SPI, numaratorul incepe sa functioneze conform parametrilor setati, iar semnalul PWM generat respecta caracteristicile asteptate.

## 6 Concluzii

Proiectul a demonstrat implementarea completa a unui periferic PWM configurabil prin interfata SPI. Principalele dificultati intampinate au fost legate de integrarea corecta a modulelor si de conectivitatea semnalelor, nu de logica interna a acestora.

Solutiile adoptate au permis obtinerea unui sistem functional, robust si extensibil, care respecta specificatiile temei.