

# Documentatie PWM generator

November 22, 2025

## 1 Motivarea alegerii modului de functionare pentru modulele FPGA

### 1.1 counter

Pentru modulul `counter` am ales o structura cu registru intern si prescaler, deoarece aceasta este metoda cea mai simpla si sigura pentru implementarea unui contor flexibil. Prescalerul permite reducerea frecventei efective de numarare, ceea ce este util atunci cand perioada dorita este mai mare decat frecventa ceasului.

Contorul poate numara atat crescator (UP), cat si descrescator (DOWN), iar overflow-ul sau underflow-ul este tratat prin resetarea valorii la zero sau la valoarea period-1. Resetul explicit `count_reset` permite control extern rapid, fara a afecta alte functii ale contorului. Aceasta structura simpla este predictibila si usor de testat.

### 1.2 instr\_dcd

Pentru modulul `instr_dcd` am ales o schema cu doua stari, **SETUP** si **DATA**, deoarece este cea mai clara si sigura metoda de a lucra cu instructiuni pe 8 biti. Ideea principala a fost sa separ complet momentul in care instructiunea este citita de momentul in care se executa operatia, pentru a evita orice posibilitate de citire sau scriere accidentală.

In starea SETUP modulul primeste byte-ul de instructiune si il decodeaza. Aici se extrag informatii importante: tipul operatiei (read sau write), adresa si partea de date (high sau low). Punand aceasta logica in SETUP, modulul stie deja ce trebuie sa faca atunci cand intra in starea DATA, eliminand confuziile.

In starea DATA se realizeaza efectiv operatia. Daca instructiunea indica o scriere, modulul preia byte-ul urmator si il trimit catre registrul corespunzator. Daca instructiunea indica citire, modulul scoate valoarea din registru. Totul este sincronizat cu semnalul `byte_sync`, pentru a evita operatii premature sau tardive.

In plus, am pastrat semnalul `high_low` ca iesire separata pentru debugging si test-bench, astfel incat se poate vedea direct ce parte de date a fost selectata. Aceasta structura simpla reduce riscul de erori si permite extinderea usoara in viitor.

### 1.3 regs

Pentru modulul `regs` am ales o organizare pe adrese fixe, astfel incat fiecare registru sa fie accesat printr-o adresa clara. Aceasta metoda este simpla, usor de folosit si evita ambiguitatile in citiri sau scrieri din partea altor module sau procesor.

Registrul de perioada, precum si registrele de comparatie, sunt pe 16 biti, astfel ca le-am impartit in doua adrese: una pentru partea joasa si una pentru partea inalta. Aceasta structura este standard, intuitiva si permite actualizarea valorilor fara logica complicata. Fiecare scriere actualizeaza direct registrul vizat, iar la citire se trimite exact byte-ul cerut de adresa.

Am introdus un regisztr special pentru resetarea numaratorului: o simpla scriere genereaza un puls scurt de reset. Aceasta permite resetarea rapida a contorului fara a afecta alte functii ale modulului.

Pentru restul regiszrelor, cum ar fi prescalerul, enable-ul, directia de numarare sau enable-ul PWM, am folosit adrese simple si valori pe 1 sau 8 biti, astfel incat sa fie usor de controlat si intelese. La citire, modulul intoarce intotdeauna continutul real al regiszrelui, ceea ce face sistemul usor de monitorizat. Aceasta organizare este clara, usor de extins si potrivita pentru orice modul configurabil prin adrese.

## 1.4 spi\_bridge

Modulul `spi_bridge` a fost conceput pentru a realiza interfata SPI master. Am folosit shift register atat pentru intrare (MOSI), cat si pentru iesire (MISO), deoarece este metoda standard si sigura de a transforma date seriale in byte-uri paralele si invers.

Datele intrate sunt preluate bit cu bit, MSB first, si stocate intr-un regiszru temporar. Datele de iesire sunt scoase dintr-un alt regiszru si trimise pe MISO. Un contor de biti (`bit_cnt`) detecteaza cand un byte este complet, iar semnalul `byte_sync` indica decoderului ca datele sunt gata de procesare.

Aceasta abordare modulara face ca SPI bridge sa fie compatibil cu decoderul de instructiuni si modulele de regiszre, oferind un flux clar de date si sincronizare sigura.