

# Software-styrt verifikasjon med UVVM gjennom co-simulering

Moderne FPGA-systemer inngår ofte som en del av et større systemer der program- og maskinvare samspiller tett. I praksis er det ofte programvare som styrer systemets oppførsel, mens FPGA håndterer tidskritisk funksjonalitet.

Verifikasjon av slike systemer foregår derimot ofte delt, hvor maskinvaren verifiseres i en simulator med en testbenk, og programvaren kjøres separat eller først testes på ferdig hardware. Denne fremgangsmåten gjøre det vanskelig å verifisere realistiske systemscenarier tidlig i utviklingsløpet.

UVVM (Universal VHDL Verification Methodology) gir en strukturert metodikk for verifikasjon med BFMer (Bus Functional Model) og VVCer (VHDL Verification Components) på transaksjonsnivå, og som styres fra VHDL-testbenken. Denne oppgaven undersøker hvordan et software-miljø kan integreres med UVVM slik at programvaren kan styre verifikasjonen.

## Problemstilling

Hvordan kan man koble et software-miljø til en UVVM-basert testbenk slik at programvare kan styre og samhandle med VVCer, BFMer og andre verifikasjonskomponenter under simulering?

Oppgaven skal undersøke:

- Hvordan kommunikasjon mellom software og UVVM kan etableres.
- Hvordan software kan sende kommandoer til og utveksle data med VVCer.
- Hvordan man håndterer synkronisering mellom software og simulator.

## Målsetting

Målet er å utvikle et oppsett der software kan styre et UVVM-basert verifikasjonsmiljø. Oppgaven vil fokusere på:

- Etablere et co-simuleringsmiljø med software og simulator.
- Definere et grensesnitt (API) mellom software og UVVM testbenk.
- Gjøre det mulig å styre VVCer fra software.
- Evaluere brukbarhet og ytelse.

## Oppgaver

Studenten skal

- Sette seg inn i UVVM-arkitekturen (VVCer, BFMer, kommandoekeer etc).

- Undersøke eksisterende metoder for co-simulering.
- Designe et grensesnitt mellom software og UVVM.
- Løse utfordringer knyttet til synkronisering mellom programvare- og maskinvaredomener, og implementere mekanismer for å håndtere tidsdomener og sikre korrekt datautveksling.
- Utføre eksperimenter for å måle ytelse, responstid og nøyaktighet i co-simuleringsoppsettet, og sammenligne resultatene med alternative metoder eller eksisterende løsninger.
- Vurdere effektiviteten av det utviklede oppsettet, og identifisere potensielle forbedringer og fremtidige forskningsmuligheter.
- Dokumentere alle trinn i prosessen, inkludert designbeslutninger, implementeringsdetaljer og eksperimentelle resultater, oppsummerer funnene og gi anbefalinger.

## Relevante forskningsspørsmål

- Hvordan eksponere VVC-funksjonalitet til software på en robust måte?
- Hvordan håndtere synkronisering mellom software og simulator?
- Hva er de største flaskehalsene i co-simulering mellom software og FPGA, og hvordan kan de overvinnes?
- Hvordan påvirker co-simulering ytelsen og nøyaktigheten sammenlignet med individuelle simuleringer?