**VHDL Ontwikkeljournaal**

**Projectnaam: Motorbesturing met FSM**

**Auteur: Jesse Ernste**

**Datum: 04-03-2025 & 06-03-2025**

**1. Inleiding**

Dit project betreft de implementatie van een finite state machine (FSM) voor de besturing van een motor of centrifuge. De FSM regelt verschillende snelheden en wachttijden op basis van inputsignalen en systeemstatussen.

**2. Specificaties**

* **Doelhardware:** FPGA
* **Klokfrequentie**
* **Interfaces:** Geen externe interfaces, alleen digitale besturing
* **Functionaliteit:**
  + Start en stop van de motor
  + Meerdere snelheidsniveaus (bijv. 200, 800, 1200 tpm)
  + Wachttijden tussen snelheidsveranderingen
  + Verwerking van configuratie-invoer (centrifugekeuze)

**3. Ontwerpbeslissingen**

* **FSM-gebaseerde besturing**: De motor wordt bestuurd met een toestandsmachine, zoals geïllustreerd in het toestandsdiagram.
* **Snelheidsoverdracht**: De FSM verhoogt de snelheid stapsgewijs en wacht een bepaalde tijd voordat de volgende overgang plaatsvindt.
* **Gebruik van std\_logic en std\_logic\_vector**: Dit zorgt voor compatibiliteit met FPGA-hardware.
* **Asynchrone reset**: De FSM wordt gereset wanneer aan\_uit = '0'.

**4. Implementatie**

* **Bestandsstructuur:**
  + fsm\_motor.vhd: Hoofdmodule met de FSM
  + fsm\_motor\_tb.vhd: Testbench voor simulatie
* **Belangrijkste signalen:**
  + aan\_uit: Start/stop van het systeem
  + centrifugekeuz: Configuratie van de motor
  + clk: Kloksignaal
  + go\_trommel: Startsignaal voor de trommel
  + motor: 3-bits uitgangssignaal voor snelheid
  + trommel\_ready: Statussignaal voor gereedheid

**5. Simulatieresultaten**

* **Testbenches**:
  + Stapsgewijze verhoging van de snelheid
  + Wachttijden worden correct nageleefd
  + Reactie op inputsignalen
* **Tools**:
  + ModelSim / Vivado voor simulatie en synthese
* **Resultaten**:
  + FSM doorloopt de juiste toestanden
  + Uitgangssignalen reageren zoals verwacht

**6. Problemen en Oplossingen**

* **Glitches in snelheidsoverdracht**: Opgelost door extra synchronisatie met klokranden.
* **Resettiming**: Asynchrone reset correct geïmplementeerd om racecondities te voorkomen.

**7. Verbeteringen en Toekomstige Werkzaamheden**

* Optimalisatie van wachttijden met preciezere timing-regeling.
* Toevoegen van foutdetectie en beveiligingsmechanismen.
* Implementatie van een PWM-besturing voor fijnere snelheidsoverdracht.

**8. Conclusie**

Dit project heeft met succes een FSM geïmplementeerd voor de besturing van een motor met meerdere snelheden en wachttijden. De FSM functioneert correct volgens simulaties, en verdere optimalisaties kunnen worden toegepast voor een robuuster systeem.