

Opbouw implementatie

Stef Sijben

26 april 2007

1 Algemeen

In dit bestand wordt beschreven welke subroutines er nodig zijn in het programma. Sommige van deze subroutines hebben parameters of geven een waarde terug. In het geval van parameters wordt A of, in het geval van meerdere parameters, R0..Rn gebruikt om deze door te geven. Return waarden worden afhankelijk van hun type verschillend teruggegeven: booleans komen in de carry bit, andere types in A.

2 Low-level I/O

De subroutines die hierbij horen worden doen heel simpele dingen, als het in- of uitschakelen van een enkele motor, lamp of ventiel of het uitlezen van een enkele sensor. Hier komt dus gewoon voor elk hardwareapparaat een of meerdere subroutines.

3 Belangrijke acties

3.1 Wafer verplaatsen

Hierin komen een paar subroutines die zorgen dat de tweede lopende band voor een bepaald aantal seconden of totdat een sensor iets registreert in een bepaalde richting beweegt.

Daarnaast is er een subroutine die probeert een wafer van de eerste band te halen.

- MoveForwardT(s): laat de tweede lopende band voor $s * \text{factor}$ seconden voorwaarts bewegen en stop dan.
- MoveBackwardT(s): idem, maar dan achteruit.
- MoveForwardS1, MoveForwardS2: Laat de tweede lopende band voorwaarts bewegen totdat lichtsensoren 1 resp. 2 iets registreert en return true, of als de sensor na een vooraf vastgestelde tijd nog niets geregistreerd heeft, return false.
- MoveBackwardS1, MoveBackwardS2: idem, maar dan achteruit.
- LoadWafer: Probeer een wafer te laden van de eerste band. Als een wafer klaar ligt, return true. Als dit na een vastgestelde tijd nog niet gelukt is, return false.

3.2 Luchtsluis

Hierin komen subroutines voor het openen en sluiten van de deuren.

- OpenDoor1, OpenDoor2: Controleer of de andere deur dicht is en zo ja, open de desbetreffende deur en controleer of hij open is. Return of dit gelukt is.

- CloseDoor1, CloseDoor2: Sluit een deur en controleer of ze echt dicht is. Return of dit gelukt is.

Doe een paar pogingen om een deur open of dicht te krijgen en return pas false als het dan nog steeds niet gelukt is. In dat geval lijkt de deur dus vast te zitten en moet het programma stoppen.

3.3 UV lamp

Hierin zit enkel de subroutine die het daadwerkelijk branden van de wafer afhandelt.

- Burn: Verander de Interrupt vector van de noodknop naar EMBurn, schakel de lamp in, wacht 2 seconden, schakel de lamp weer uit en zet de interrupt vector weer terug naar EMStandard.

3.4 onderbrekingen

Hierin komen interrupt handlers voor de noodstop en voor het verliezen van wafers.

- EMStandard: De normale handler voor het indrukken van de noodstop. Sla de toestand van alle uitvoerpoorten ergens op en schakel vervolgens alles uit. Wacht dan tot de noodstop opnieuw wordt ingedrukt en herstel dan alle uitvoerpoorten naar hun oude toestand.
- EMBurn: De handler voor het indrukken van de noodstop tijdens het branden van een wafer. In dat geval is de wafer mislukt en wordt de lamp uitgeschakeld. Bij het opnieuw indrukken van de knop wordt de wafer naar de afvalbak getransporteerd en vervolgens keren de deuren terug in hun uitgangspositie, waarna de volgende wafer geladen wordt.
- WaferLost: Registreer een verloren wafer, als er hierna nog steeds minder dan 5 verloren zijn, gaat het systeem naar de uitgangspositie en probeert de volgende wafer te laden. Als er wel 5 verloren zijn crasht het systeem.