

2

(B L O K 2)

Opbouw van een PLC

Om een PLC te gebruiken, moet men geen elektronica- of informatica-specialist zijn. Dit boek is geen handleiding voor het zelfstandig bouwen van een PLC. Het is echter wel noodzakelijk enige basiskennis over de opbouw van de PLC te bezitten, om deze foutloos te kunnen implementeren in sturingsschakelingen. In dit blok wordt een beschrijving gegeven van de verschillende functionele delen waaruit een PLC is samengesteld en wordt de werking nader toegelicht.

Doelstellingen

- Aan de hand van een blokschema, de verschillende delen van een PLC benoemen
- Verklaar hoe de ingangen en uitgangen van de PLC zijn geconstrueerd
- De constructie verantwoorden, gebaseerd op technologische principes
- Het begrip "cyclische verwerking" verklaren en het verband leggen met het PII- en PIQ-geheugen
- De verschillende soorten geheugens benoemen en het verschil verklaren
- Verklaar hoe de adressering van ingangen en uitgangen wordt uitgevoerd
- Het toepassingsgebied omschrijven van remanente en niet-remanente merkers
- De verschillende elementen van een instructie opsommen
- Het onderscheid duidelijk maken tussen de programmeertalen
- De voor- en nadelen van de verschillende programmeertalen opsommen
- Het verschil tussen lineair en gestructureerd programmeren verklaren



1 Bestanddelen van een PLC

Het blokschema van fig. 2.1 geeft schematisch weer hoe een PLC is opgebouwd uit verschillende functionele delen.

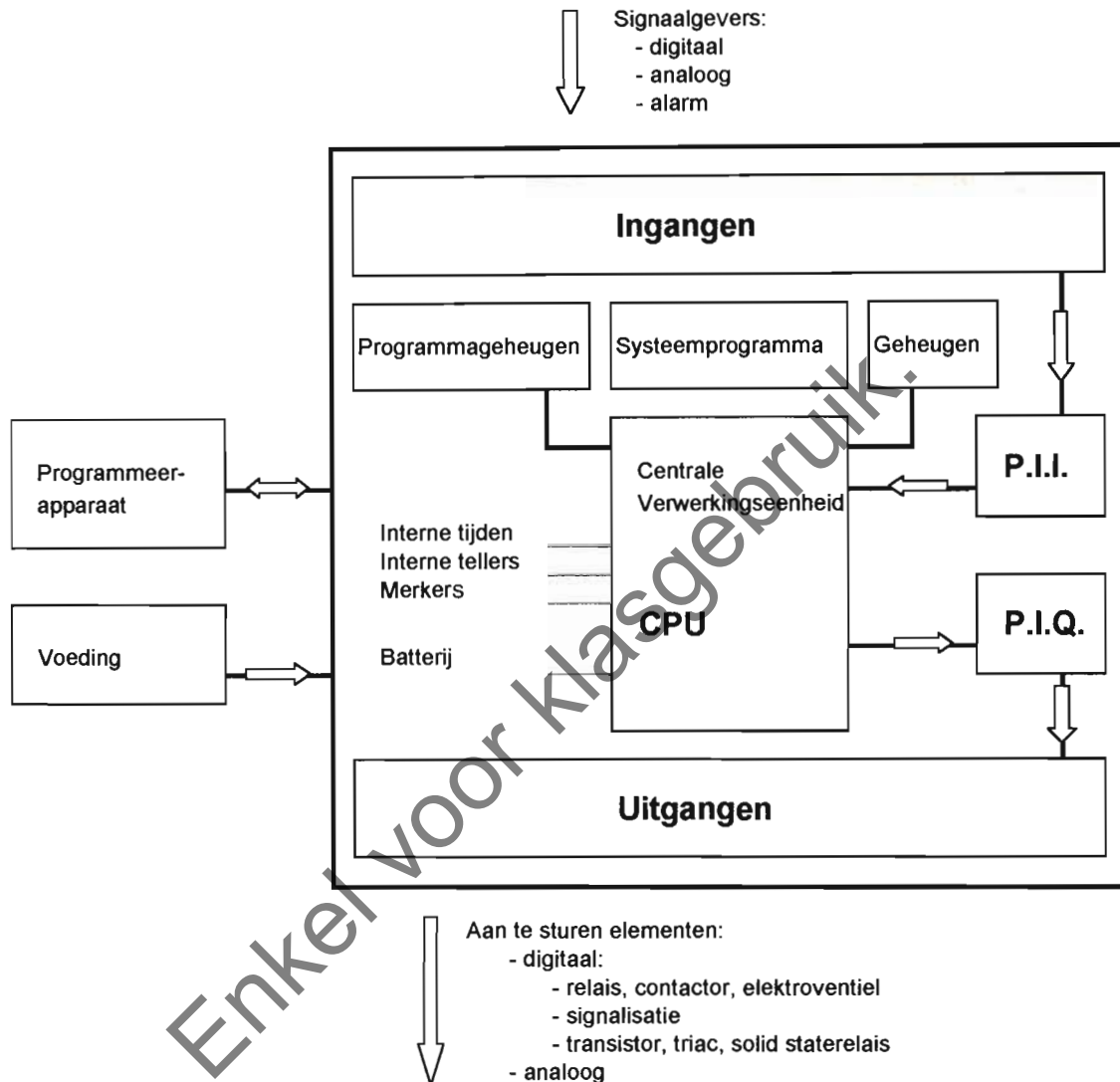


Fig. 2.1 Blokschematische voorstelling van een PLC

Een PLC bestaat in principe uit:

- een **centrale verwerkingseenheid** met stuurorgaan en programmageheugen;
- een aantal **in- en uitgangsbouwgroepen** (periferie);
- een **voedingseenheid**.

De signaalgevers zijn aangesloten aan de ingangen van de PLC. In de centrale eenheid wordt, door het besturingsorgaan, het in het geheugen opgeslagen programma verwerkt. Instructies in dit programma kunnen de toestand van de ingangen afvragen. Er wordt gedetecteerd of er al dan niet een spanning aan de opgeroepen ingang aanwezig is. Afhankelijk van deze toestand kunnen beslissingen worden genomen die in het aansturen van een uitgang kunnen resulteren.

In de volgende alinea wordt uitvoerig de functie van de samenstellende onderdelen van een PLC besproken. Er wordt aangeduid hoe de signaalverwerking binnen de PLC gebeurt.

2 Ingangen

Wordt een proces geautomatiseerd, dan worden een aantal schakelstanden van detectoren verwerkt. De toestand van deze detectoren, die opgesteld staan in de installatie, worden via de ingangseenheid aan de PLC gelezen. De CPU tast de toestand van de signalen van de sensoren af. De toestanden worden naar het ingangsprocesbeeld **PII** (PII = **P**rocess-**I**mage **I**nterface), bij elk cyclusbegint, overgedragen.

Met behulp van een LED, zichtbaar in de PLC gemonteerd, wordt de toestand van een ingang weergegeven. De LED licht op bij een gesloten stroomkring. Zo'n stroomkring bestaat uit de serieschakeling van de voeding, het schakelcontact van de detector en de aansluitklem van de ingang. Deze indicatie is zeer efficiënt bij het testen van een nieuwe installatie of bij het opsporen van fouten.

De ingangen zijn meestal zo geconstrueerd dat ze volledig galvanisch van het centrale verwerkingsgedeelte zijn gescheiden. Uitwendige fouten kunnen geen invloed op het processorgedeelte uitoefenen. Om deze **galvanische scheiding** te realiseren tussen de PLC en de overige delen van de installatie, worden de ingangselementen via een optische koppeling aangesloten. Deze bestaat uit een LED en een transistor, in een hermetisch gesloten behuizing ondergebracht. De transistor wordt als open of gesloten schakelaar door de lichtstraal van de LED gestuurd. Deze schakeltechniek wordt in een afzonderlijk hoofdstuk besproken.

Bij de kleinere PLC's wordt de voeding, noodzakelijk om de ingangen te sturen, ingebouwd. Bij grote PLC-systemen moet men de voeding als een afzonderlijke component naast de PLC monteren. Bij de meeste PLC's wordt 24V DC toegepast omdat dit een veilige spanning is.

De ingangselementen hebben ook tot taak de stoorsignalen buiten de PLC te houden. Dikwijls gebruikt men daarom een RC-Network met een tijdconstante van 0,2 à 0,3 ms. De correcte tijden zijn terug te vinden in de technische specificaties die bij de PLC horen. Dit netwerk heeft als nadeel dat de **responsietijd** vergroot. Anderzijds heeft dit het voordeel dat de **contactdender** van de detectoren geen invloed heeft op het schakelproces. Hier moeten fabrikanten tegenstrijdige belangen verzoenen. Hoe beter de PLC de contactdender elimineert, hoe trager hij het proces opvolgt. Voor de meeste processen speelt deze vertraging echter geen rol omdat de optredende tijdsvertragingen klein zijn.

Praktisch wordt de storingsgevoeligheid door een selectief systeem verbeterd. Door de PLC worden de ingangen als een logisch nulsignaal gedetecteerd, ook al loopt de spanning op van 0 V tot 5 V. Pas vanaf een bepaalde drempelwaarde, bijvoorbeeld 13 V, verkrijgt men een logisch één signaal. Dit wordt voorgesteld in figuur 2.2.

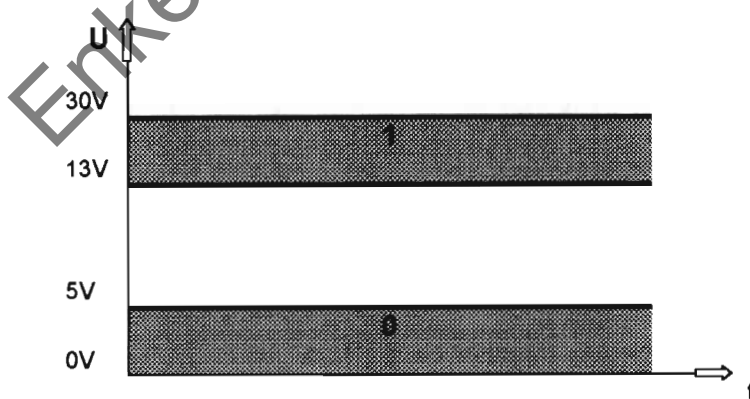


Fig. 2.2 Logisch "0" en "1"-signaal bij detectie van ingangssignalen

Bij een gesloten contact van de detector bedraagt de stroom die door het schakelement vloeit, ongeveer 5 mA. Het is noodzakelijk dat er een stroom door het schakelement vloeit. Wordt de stroomsterkte verlaagd, dan kan een kleinere voeding gebruikt worden. Dat beïnvloedt de kostprijs van de installatie gunstig. Wordt de detector in een industriële omgeving geplaatst en is de stroomsterkte kleiner dan 5 mA, dan is de kans groot dat er **oxidatie** gaat optreden van de schakelcontacten. Is de stroom groter dan deze drempelwaarde, dan ontstaat er bij gebruik een kleine vonkvorming; die zorgt ervoor dat de contactpunten van het contact zuiver blijven.

Voor het nemen van beslissingen in het programma wordt door de PLC de “toestand” van de ingangssignalen bekeken.

De toestand is afgeleid van een elektrische spanning:

- Toestand “0” komt overeen met “spanning niet aanwezig”.
- Toestand “1” komt overeen met “spanning wel aanwezig”.

De elektrische spanning stelt een procesgegeven voor. In de elektronische besturingstechniek wordt veel met binaire signalen gewerkt. Dit zijn signalen waarbij slechts twee toestanden mogelijk zijn, namelijk “0” of “1”. Ook procesgegevens worden binair (=tweewaardig) geïnterpreteerd. Het procesgegeven is een bewering dat met JA of NEE beantwoord kan worden.

In het programma worden beslissingen genomen afhankelijk van het wel of niet aanwezig zijn van spanningen op de ingangen van de PLC. Daarbij wordt niet gekeken of de detectors als maak- of verbreekcontacten zijn uitgevoerd. Vooraleer men het programma kan opstellen, moeten deze gegevens wel gekend zijn.

Is de detector met een maakcontact uitgevoerd, dan heeft de ingang toestand “1” wanneer aan de voorwaarde behorende bij de detector voldaan is. Wordt de detector niet bediend, dan wordt toestand “0” gedetecteerd. Dit noemt men het **arbeidsstroomprincipe**.

Is de detector als verbreekcontact uitgevoerd en werd aan de procesvoorwaarde voldaan, dan wordt het contact geopend. Aan de ingang van de PLC komt dit overeen met toestand “0”. Dit is het **ruststroomprincipe**. Het ruststroomprincipe wordt meestal uit veiligheidsoverwegingen gebruikt. Een installatiedeel moet men kunnen afschakelen, ook bij draadbreek. Bij het programmeren van stopfuncties zal er dieper op deze methode van programmeren worden ingegaan, rekening houdend met het arbeidsstroom- en ruststroomprincipe.

3 Uitgangen

De uitgangsmodule zet resultaten uit het gebruikersprogramma om in signalen. De resultaten van de verwerking door de CPU worden bewaard in het uitgangsprocesbeeld **PIQ** (PIQ = **P**rocess-**I**mage **O**utput table). Aan het einde van elke cyclus worden de toestand, aanwezig in het PIQ-geheugen, gekopieerd naar de uitgangen. De uitgangen sturen deze signalen naar de aangesloten verbruikers (contactoren, ventielen, signaallampen, e.a.).

Bij PLC-systemen worden drie verschillende soorten uitgangen onderscheiden. Bij elk van de drie systemen kan een volledige galvanische scheiding, tussen de uitgang en het centrale verwerkingsgedeelte van de PLC, worden gerealiseerd.

Bij elke uitgang is, net zoals bij de ingangen, een LED-indicatie voorzien die de toestand weergeeft van de betreffende uitgang.

Bij de keuze van een bepaald systeem zijn het schakelvermogen en de schakelfrequentie meestal doorslaggevend.

Relais

Een relais is een robuust element. Men kan als uitgangsrelais uitvoeringen verkrijgen tot 230 V. De toegelaten stromen kunnen variëren van 100 mA tot 2 à 3 A bij een Ohmse belasting. Men moet er rekening mee houden dat kleine toegelaten stroomsterktes de prijs van de PLC gunstig beïnvloeden. Daartegenover staat dat men bij iedere uitgang van een dergelijke PLC, zelf een extra relais moet monteren en aansluiten. Er is uiteraard ook een relatie tussen het schakelvermogen van de uitgangen en de compactheid van de PLC. Er wordt zeer veel gebruik gemaakt van PLC's met relaisuitgangen, voornamelijk omwille van de kostprijs. Alleen als de schakelfrequentie te groot wordt, is het gebruik van een PLC-uitvoering met relais niet aangewezen. Bij dergelijke toepassingen is het maximum aantal schakelingen van het relais bepalend voor de levensduur van de PLC.

Vele PLC's beschikken over potentiaalvrije contacten. De beide contactpunten van de verschillende uitgangsrelais zijn verbonden met klemmen, voorzien op de PLC. Bij dergelijke PLC's kan men aan de uitgangen toestellen aansluiten, geschikt voor verschillende spanningen.

Het komt ook veel voor dat één aansluitdraad gemeenschappelijk is. Er is dan slechts één aansluitdraad per uitgang aan de aansluitklemmen verbonden. Bij deze PLC-uitvoering moeten alle aan te sturen toestellen geschikt zijn voor dezelfde spanning. Anderzijds hebben deze PLC's het voordeel dat er minder bedrading nodig is.

Transistor

Een transistor is nuttig als de schakelingen op een gelijkspanning zijn aangesloten. De toegelaten schakelfrequentie kan veel hoger liggen dan bij relaisuitgangen.

Solid-state uitgangen

Deze uitvoering wordt toegepast bij die schakelingen die een grote schakelfrequentie vereisen en bij schakelingen aangesloten op een wisselspanning.

De solid-state uitvoering heeft twee grote nadelen. De kostprijs van een PLC met solid-state uitvoering is merkbaar hoger dan die met een relaisuitvoering. Bovendien kan een foutieve aansluiting tot gevolg hebben dat een solid-state blijvend defect raakt. Daarbij kunnen de aansluitingen niet met eenvoudige apparatuur worden gecontroleerd.

De ingangen en uitgangen zijn zowat de belangrijkste delen van de PLC. Dit niet alleen door hun prijs, maar ook door de invloed van hun keuze op het succes van het project. Voor een middelgrote automaat situeert de kostprijs van de in- en uitgangsmodule zich meestal boven de helft van de totale kostprijs.

Men kan de in- en uitgangsmodule als volgt rangschikken:

Binaire modules

Binaire modules werken met eenvoudige 0 en 1 signalen; ze worden daarom zeer veel gebruikt. De ingangen worden gebruikt om de stand van de detectoren te verwerken. De uitgangen functioneren als schakelaars om de stroomkringen te sluiten.

Snelle telleringangen

Deze ingangen verwerken de signalen eveneens binair, maar door hun specifieke functie zijn ze afzonderlijk vermeld.

Pulsen met een duurtijd of intervaltijd kleiner dan de cyclustijd van de centrale verwerkingseenheid kunnen alleen door speciale ingangen worden verwerkt. In de technische specificaties staat vermeld welke ingang(en) men voor deze functie kan gebruiken. Bij grotere systemen bouwt men een afzonderlijke kaart in met een aantal snelle telleringangen. Een afzonderlijke verwerking, onafhankelijk van de normale cyclus, zorgt voor detectie en verwerking. Pas na het bereiken van de geprogrammeerde voorwaarden wordt dit gegeven in het normale procesverloop verwerkt.

De vertragingstijd beperkt zich tot 10 μ s. De telfrequentie kan 1, 2, 5 of zelfs 750 kHz bedragen.

In de technische specificaties is het mogelijk de signaalduur af te lezen.

Bijvoorbeeld : signaalduur > 100 μ s.

Alarmingangen

Deze ingangen hebben als specifieke functie het cyclisch programma te onderbreken en een alarmprogramma te bewerken. Wanneer een signaal aan deze ingang wordt gedetecteerd, zorgt een afzonderlijk programmeel ervoor dat het proces op een snelle manier wordt beïnvloed. In dit geval spreekt men van een alarminterrupt.

Conversie modules of registers

De meest toegepaste code is een BCD-code. De ingangen worden gebruikt om de stand van duimwiel-schakelaars te lezen. De uitgangen sturen displays aan.

Analoge ingangen of uitgangen

Deze zijn bestemd voor een continu signaal, zoals het meetsignaal van een transmitter of het stuursignaal van een regelklep.

4 De centrale verwerkingseenheid

De centrale verwerkingseenheid wordt in het Engels “**C**entral **P**rocessing **U**nit” genoemd, afgekort **CPU**. Deze module bestaat uit drie delen:

- De **logische eenheid** of in het Engels “Logic Unit”
- De **accumulator** of kortweg de accu genoemd
- De **tijd- en controle-eenheid**

De centrale verwerkingseenheid is het brein van de PLC. De ingebouwde microcomputer voert allerlei bewerkingen uit aan de hand van opdrachten die hem worden gegeven. Deze reeks van opdrachten vormt samen een programma. Elke opdracht wordt voorafgegaan door een adres.

4.1 Logische eenheid

In dit deel van de CPU gebeurt de verwerking van de gegevens, gebaseerd op het programma dat in het programmeergeheugen aanwezig is.

De verwerking is “logisch”, omdat het programma uit regels bestaat met instructies van logische bewerkingen zoals EN, OF en NIET. Ook inhouden van geheugens die alleen de waarden “0” of “1” kunnen aannemen, komen erin voor.

De programmalijnen worden door de centrale verwerkingseenheid in stijgende volgorde verwerkt. Dit gebeurt aan de hand van twee soorten informatie: deze die in de vorm van ingangssignalen worden aangeboden en deze die in de vorm van instructies in het programmeergeheugen staan.

Eén van de essentiële verschillen tussen de klassieke sturingen en de besturingen die gebruik maken van een PLC, is de manier hoe de aangeboden signalen worden verwerkt:

- *Parallelverwerking*

In een besturing, opgebouwd met contactloze elementen of relais, kunnen meerdere logische functies gelijktijdig worden uitgevoerd.

- *Serieverwerking*

Bij een PLC-sturing worden de logische functies na elkaar gelezen, berekend en uitgevoerd.

Door het fundamenteel verschil in besturing, zal men er rekening mee moeten houden op het ogenblik dat men zijn programma opstelt.

4.2 Cyclische programmaverwerking

Cyclische programmaverwerking betekent dat de processor de verschillende instructies na elkaar uitvoert. Bij het einde van het programma gekomen, herhaalt de processor onafgebroken deze verwerking.

De tijd die verloopt tussen de eerste en de laatste programmastap wordt de cyclustijd of scantijd genoemd. De cyclustijd is een maat voor de snelheid van de PLC. De cyclustijd wordt beïnvloed door de soort van de gebruikte instructies en de omvang van het programma. Omdat de cyclustijd kort is, wordt elke toestandswijziging van een ingangssignaal onmiddellijk onmiddellijk verwerkt.

De werking van de meeste moderne PLC's is op een cyclische programmaverwerking gebaseerd. Een groot voordeel van cyclische programmaverwerking is dat de ingangen aan het begin van elke programmacycclus opgevraagd worden. Zo is er altijd een actueel beeld van de toestand van de ingangen beschikbaar. De resultaten van de verwerking van het programma worden in een geheugen bewaard. Aan het einde van elke programmacycclus worden de toestanden van de uitgangen aangepast.

Het programma loopt dus continu door: als er een tijdfunctie wordt gestart, dan loopt het programma verder en stopt niet zoals in een Pascal programma.

Tijdens één cyclus worden achtereenvolgens volgende acties uitgevoerd:

- Lezen van de ingangen en bewaren van de toestanden in het PII-geheugen
- Bewerken van het opgemaakte programma en de resultaten sturen naar het PIQ-geheugen
- Kopiëren van toestanden van het PIQ-geheugen naar de uitgangen

Deze volgorde wordt behouden en steeds herhaald tijdens de cyclische bewerking van het programma in de CPU.

Afhankelijk van het type PLC onderscheidt men verschillende methodes om het opgestelde programma te verwerken.

- **Vaste cyclustijd**

Onafhankelijk van de lengte van het programma worden al de programmalijnen doorlopen, ook de niet-gebruikte programmalijnnummers. Wanneer de laatste programmalijn gelezen is, gaat het programma verder met de eerste programmalijn.

- **Variabele cyclustijd**

Na de laatste instructie en onafhankelijk van de geheugenlengte wordt het programma verder gezet op de eerste programmalijn. De niet-gebruikte programmalijnen worden niet gelezen. Bij deze PLC's zal men het programma moeten beëindigen met een specifieke code, bijvoorbeeld END of MEND (main end).

- **Variabele cyclustijd, afhankelijk van de sprongfuncties**

Deze werkwijze heeft dezelfde kenmerken als de variabele cyclustijd, maar er wordt ook met spronginstructies rekening gehouden.

PLC's beschikken over een instructieset die het mogelijk maakt dat de PLC, onder bepaalde voorwaarden, het programma vanaf een aangewezen programmalijnnummer verder leest en verwerkt.

Men kan stellen dat de cyclustijd bij de meeste moderne PLC's zeer kort is. Bij de meeste sturingen lijkt het alsof de PLC ogenblikkelijk reageert. Het is dikwijls onmogelijk met het blote oog de vertragingen waar te nemen.

4.3 Accumulator

De accumulator is een register of geheugen waarin het tussenresultaat van de bewerkingen wordt bewaard tot een volgende bewerking wordt uitgevoerd. De inhoud van de accumulator wordt dan overschreven door het nieuwe resultaat.

Aan het begin van elke cyclus vraagt het stuurorgaan de signaaltoestanden van alle ingangen op. De toestand van alle ingangen wordt gekopieerd naar een geheugen: het PII-register (**PII** = **P**rocess-**I**mage **I**nterface). Daarna begint de eigenlijke verwerking van het programma; instructie na instructie wordt afgewerkt.

De tussenresultaten van de bewerkingen in de logische eenheid worden bewaard tot er nieuwe tussenresultaten voorkomen. De oude toestanden worden overschreven. De resultaten van de verwerkingen, die op de uitgangen betrekking hebben, worden weggeschreven in het PIQ-register (**PIQ** = **P**rocess-**I**mage **O**utput table). Na de laatste instructie wordt de inhoud van het PIQ-register gekopieerd naar de uitgangen.

4.4 Geheugens

Men kan de verschillende soorten geheugens onderverdelen in de volgende groepen:

- **RAM**

Random Access-Memory

Het RAM-geheugen is een vrij programmeerbaar lees- en schrijfgeheugen. Valt de voedingsspanning weg dan verliest het geheugen zijn inhoud.

Het volledige gebruikersprogramma wordt voor de bewerking van het inwendige EEPROM geheugen naar dit geheugen geschreven. De verwerkingssnelheid van een RAM geheugen is veel hoger dan van een EEPROM.

- **ROM**

Read Only Memory

Het ROM-geheugen kan alleen uitgelezen worden. De informatie wordt tijdens de fabricage in de ROM vastgelegd. Men kan de informatie niet wijzigen.

- **PROM**

Programmable Read Only Memory

Het PROM-geheugen is een soort van ROM geheugen, dus alleen uitleesbaar. De gebruiker kan, met een programmeerapparaat, een eigen gemaakt programma in het geheugen invoeren. Dit gebeurt door verbindingen blijvend te onderbreken. De onderbrekingen kunnen niet worden opgeheven.

- **EPROM**

Erasable Programmable Read Only Memory

Het EPROM-geheugen wordt elektrisch, door spanningspulsen, geprogrammeerd. De geheugeninhoud kan door bestraling met ultraviolet licht worden gewist. Is het geheugen volledig gewist, dan kan men het opnieuw programmeren.

- **EEPROM**

Electrically Erasable Programmable Read Only Memory

Het EEPROM-geheugen kan elektrisch, zowel geprogrammeerd als gewist worden. De mogelijkheid bestaat om slechts een deel uit het geheugen te wissen. Bij een EPROM wordt door de UV-bestraling de hele inhoud gewist. Het EEPROM wordt soms voorgesteld als E²PROM.

Het gebruikersprogramma, de systeemparemeters en gegevens die als remanent worden geparemetreerd worden, nadat ze via het programmeerapparaat naar de PLC werden overgedragen, in een intern EEPROM-geheugen bewaard.

Een externe EEPROM-module mag men niet beschouwen als een geheugenuitbreiding. De module wordt in de CPU geplugd. Ze is hoofdzakelijk bedoeld als hulp voor de gebruikerssoftware of om eenvoudig een programmawisseling in de PLC uit te voeren, zonder gebruik te maken van een programmeerapparaat.

4.5 Bussysteem

Om de verschillende delen van de PLC onderling te laten samenwerken, zijn er verbindingen nodig: het bussysteem.

Het bussysteem is een verzameling van leidingen die zorgen voor de overdracht van signalen. Via deze bus gebeurt, bijvoorbeeld, de uitwisseling van gegevens tussen de processor en de in- en uitgangskarten. De bus bestaat uit meerdere parallelle signaalleidingen:

- De adresbus, waarmee de adressen op de verschillende kaarten aangesproken worden
- De databus, waarmee gegevens afkomstig van ingangskarten of vertrekkende naar uitgangskarten overgedragen worden
- De sturingsbus waarmee de stuur- en bewakingssignalen worden overgedragen.

Is de PLC opgebouwd als één geheel (monoblok), dan zijn deze verbindingen intern gerealiseerd. Heeft men te maken met een modulair opgebouwde PLC, dan zijn deze verbindingen op de met steekplaatsen voorziene moederplaat van het rack aangebracht.

4.6 Voedingseenheid

Men moet een onderscheid maken tussen twee soorten voedingen:

De **externe** voeding

Deze voeding brengt de gebruikelijke spanning tot een veilige spanningswaarde terug. Deze spanning wordt ook gebruikt om de signaalgevers aan de ingangen van de PLC aan te sluiten. Bij sommige PLC's, vooral bij kleinere systemen, is de voeding ingebouwd.

De **interne** voeding

Deze voedingseenheid transformeert de netspanning. De nodige gelijkrichters, filters en ontstoorapparatuur zijn ingebouwd.

4.7 Bufferbatterij

Door een bufferbatterij kan het RAM-geheugen of een gedeelte hiervan onder spanning worden gehouden. Is de PLC met een bufferbatterij uitgerust, dan gaat de informatie bij een eventuele spanningsonderbreking van het net niet verloren. Op die manier is het mogelijk dat men, bijvoorbeeld, de inhoud van een teller onthoudt.

4.8 Merkers

Merkers dienen in PLC-programma's als hulpgeheugens. Er is geen fysische koppeling met de periferie. Zo is het mogelijk dat bij programma's met complexe functies een tussenresultaat in een merker wordt opgeslagen en elders in het programma terug wordt opgevraagd.

Merkers zijn een aantal bits in het RAM-geheugen van de PLC. Elke bit (merker) kan op dezelfde manier als de uitgangen worden geprogrammeerd.

Een PLC kan over remanente en niet-remanente merkers beschikken.

Remanente merkers onthouden hun waarde omdat ze door een bufferbatterij gebufferd zijn. Bij een eventuele spanningsonderbreking blijven ze hun toestand bewaren. Niet alle merkers zijn remanent. Meestal zijn ze voor de helft als remanente en voor de andere helft als niet-remanente merkers uitgevoerd. Bij moderne PLC's kan men softwarematig instellen welke merkers remanent worden gebruikt.

Men moet bij het toewijzen van de merkers en het opstellen van het programma rekening houden met het doel dat men wenst te bereiken. Om dit te verduidelijken twee voorbeelden:

Voorbeeld 1:

Bij een draaiende machine zal het zeker niet aangewezen zijn dat, na een spanningsonderbreking, de machine zonder menselijke ingreep automatisch terug start. In dergelijke gevallen zal men gebruik maken van niet-remanente merkers.

Voorbeeld 2:

Bij een verpakkingsmachine heeft men een aantal eenheden aangevoerd. Op een bepaald ogenblik treedt er een spanningsonderbreking op. Op dat ogenblik is het noodzakelijk dat in een geheugen de toestanden bewaard blijven, zodat het proces nadien zijn normale werking kan verder zetten.

Men zal remanente merkers altijd toepassen in combinatie met bistabiele ventielen of contactoren. Daarover vindt u meer in blok 3.

5 Adressering

Door gebruik te maken van een voorbeeld wordt het systeem van adressering verduidelijkt. Wanneer de schakelaars S1 en S2 gesloten zijn, moet de lamp H1 branden. We hebben hier te maken met een gewone serieschakeling. In het volgende blok wordt verklaard hoe men het programma opstelt. De schakelaars zijn aangesloten aan de ingangen van de PLC, de lamp is aangesloten aan de uitgangen. Om de schakelaars S1 en S2 in het programma te laden, moet er een duidelijk onderscheid zijn tussen het adres van schakelaar S1 en dat van schakelaar S2. De adressen van de schakelaars worden bepaald door de aangebrachte bedrading. Schakelaars worden logisch genummerd op basis van de door de gebruiker geprogrammeerde schakelvolgorde. Schakelaar S1 heeft als adres I0.1 in de PLC, schakelaar S2 heeft als adres I0.2. De letter I betekent Input, 0.1 en 0.2 voor byteadres en bitadres. De lamp wordt aangesloten aan de uitgang Q0.1. Men gebruikt de letter Q i.p.v. de letter O om geen verwarring te veroorzaken tussen de letter O en het cijfer 0.

5.1 Bit-, byte- en woordadres

Een bit is de kleinste eenheid van het binaire systeem.

Een eenheid van 8 op elkaar volgende bits noemt men een byte. De PLC vat bijvoorbeeld de toestand van 8 in- of uitgangen samen in een ingangsbyte (IB), respectievelijk een uitgangsbyte (QB). Elke bit van een byte kan de waarde 0 of 1 aannemen. De PLC is eveneens in staat een volledige byte als geheel te verwerken.

Worden 16 opeenvolgende bits tot een eenheid gebracht, dan spreekt men van een woord. Een woord bestaat uit 16 bits of 2 bytes. In de PLC wordt bijvoorbeeld de toestand van 16 in- of uitgangen samengevat in een ingangswoord (IW), respectievelijk een uitgangswoord (QW).

Worden 32 opeenvolgende bits tot een eenheid gebracht, dan spreekt men van een dubbelwoord. In de technische documentatie wordt ID gebruikt om een ingangsdubbelwoord aan te duiden. Voor een uitgangsdubbelwoord gebruikt men QD. In dit boek wordt er verder geen aandacht geschonken aan deze mogelijkheden omdat deze instructies de doelstellingen van dit boek overstijgen.

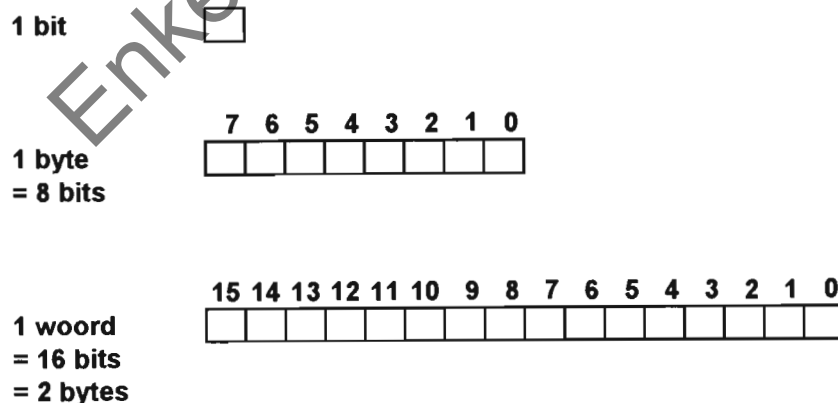


Fig. 2.3 Voorstelling van bit, byte en woord

Om een bit te kunnen aanspreken, wordt aan elke bit binnen een byte een nummer toegekend. Men noemt dit nummer een bitadres. In elke byte krijgt de uiterst rechtse bit het bitadres 0 en de uiterst linkse bit het bitadres 7.

Als men het zogenaamde woordadres gebruikt, dan kunnen 16 informatieplaatsen in één woord worden samengevat. Zo kenmerkt IW2 het ingangswoord dat bestaat uit ingangsbytes IB2 en IB3. Elke byte is van een nummer voorzien, dat men het byteadres noemt. Bijkomend wordt de operand nog nader geïdentificeerd, zodat bijvoorbeeld IB2 voor ingangsbyte 2 en QB4 voor uitgangsbite 4 staat. Een combinatie van het bitadres en het byteadres maakt het mogelijk een specifieke, enkelvoudige bit in een specifieke byte aan te spreken. Hierbij wordt het bitadres van het byteadres door een punt gescheiden. Links van het punt staat het byteadres; rechts van het punt het bitadres.

Bij modulaire PLC's komt het zeer frequent voor dat de byteadressen van in- en uitgangen afhankelijk zijn van het steekplaatsnummer van de overeenkomstige in- of uitgangskaart. Aan elke steekplaats in het rack is een vast nummer verbonden. Dit nummer legt het beginadres van de op deze plaats geïnstalleerde kaart vast. De nummering loopt door, zelfs wanneer niet alle steekplaatsen gebruikt zijn.

Bij vaste steekplaatsadressering kunnen in- en uitgangen niet hetzelfde adres hebben. Een adres gekoppeld aan een ingangskaart kan niet opnieuw als adres voor een uitgangskaart worden gebruikt. Grote PLC's maken gebruik van variabele adressering van in- en uitgangskaarten. Het adresbereik ligt niet vast maar kan door de gebruiker worden ingesteld.

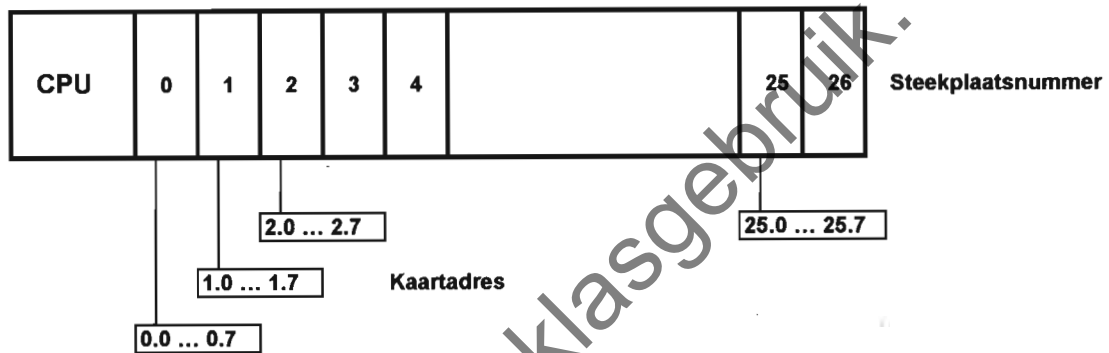


Fig. 2.4 Steekplaats en adressering

Figuur 2.4 geeft een voorbeeld van een adressering van een PLC, maar is zeker niet algemeen geldend.

5.2 Instructieopbouw

In het programma worden d.m.v. instructies ingangen afgevraagd en uitgangen aangestuurd. Een instructie is de kleinste mogelijke zelfstandige eenheid van een programma. Een instructie is een voorschrift voor het besturingsorgaan in de centrale eenheid van de PLC.

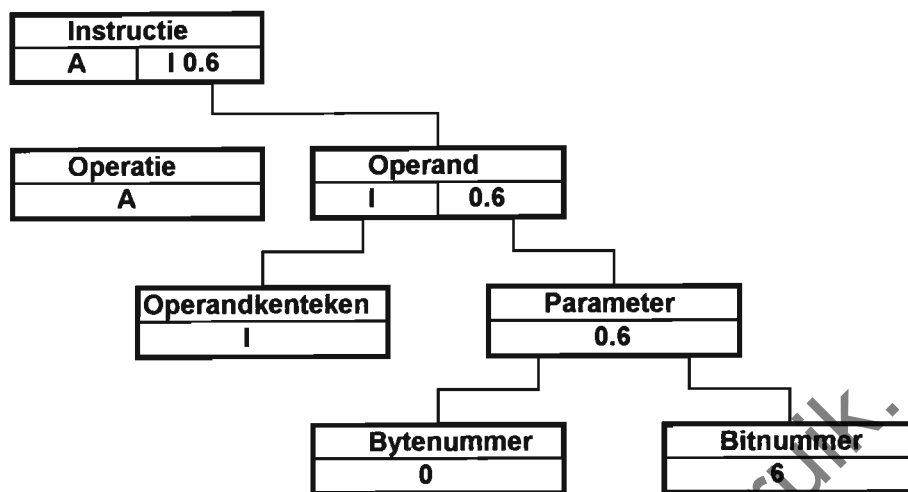


Fig. 2.5 Instructieopbouw

Operatie

Het operatiegedeelte geeft een omschrijving van de functionele verwerking van de operand. Men maakt een onderscheid in de volgende soorten operaties:

Binaire operaties

Daarmee kan men binaire signalen opvragen en verbinden. De belangrijkste binaire operaties zijn beschreven in de volgende tabel.

Benaming	Operatie	Omschrijving
Load	LD	Laad een bitwaarde
EN	A	EN-instructie (AND)
OF	O	OF-instructie (OR)
NIET	N	Geïnverteerde bitwaarde (NOT)
Toewijzen	=	Toestandtoewijzing voor een bit
Setten (instellen)	S	Een bitwaarde activeren
Resetten (terugstellen)	R	Een bitwaarde deactiveren

Digitale operaties

Hieronder vallen rekenkundige operaties, vergelijkingsinstructies, operaties voor code-omzetting ed. De digitale operaties worden ook als woordverwerkende operaties gezien, daar de operaties uit "woorden" bestaan.

Operaties voor programmaorganisatie

Tot deze operaties behoren programmavertakkingen, dus sprongaanwijzingen, zoals bouwsteenfuncties. (spronginstructies, activeren van subroutines, bouwsteeneinde).

Operand

Het operandgedeelte is dat gedeelte van een instructie dat de noodzakelijke data voor de uitvoering van de operatie specificeert. Om de operand te benoemen, is het noodzakelijk dat een operandkenteken en een parameter worden benoemd.

Operandkenteken

Volgende letters worden gebruikt om een operand te benoemen:

Benaming	Kenteken	Omschrijving
Ingangen	I	Input
Uitgangen	Q	Output
Hulpgeheugen	M	Merker
Geheugenplaats met specifieke functie	SM	Speciale Merker
Tijdfuncties	T	Timer
Tellers	C	Counter
Variabele	V	Willekeurig getal

Men gebruikt de letter I voor ingang en de letter Q voor uitgang. Deze letters zijn afgeleid van de Engelse woorden "Input", respectievelijk "Output". Met de letter Q wil men vergissingen voorkomen tussen de letter O en het cijfer 0.

Bij PLC van Duitse fabrikanten werden de aanduidingen E (Eingang) en A (Ausgang) gebruikt. Deze methode wordt nu minder gebruikt omdat ze niet in overeenstemming is met de norm IEC 1131.

Parameter

De parameter bestaat uit een combinatie van het byteadres en het bitadres. De in- en uitgangen van de PLC zijn gegroepeerd in groepen van 8 (0 tot en met 7). De parameters werden behandeld bij het adresseren.

5.3 Toekennen van ingangen en uitgangen aan componenten

Wordt gebruik gemaakt van een elektrisch schema, dan wordt voor het opstellen van het PLC-programma een lijst gemaakt. Men heeft een duidelijke overzichtstabel nodig waarbij wordt aangegeven welke schakelaar aan welke ingang van de PLC wordt verbonden. Deze lijst wordt eveneens gebruikt bij het aansluiten van de verschillende componenten aan de PLC.

Voorbeeld:

Symbool	PLC Adres	Omschrijving
S1	I0.1	Hoofdschakelaar
S3	I0.3	Startdrukknop
S4	I0.4	Stopdrukknop
K1	Q0.1	Contactoor voor motor
H3	Q0.3	Controlelamp werking
K5	M0.5	Hulpelais gesloten toestand

Om de transfer tussen schema en PLC-programma zo duidelijk mogelijk te maken, heeft men er alle belang bij dat er enige overeenkomst bestaat tussen de nummering van de toestellen en de adressering in de PLC.

6 Programmeertalen

De belangrijkste eigenschap van een ontwerpmethode is zijn "toegankelijkheid". Deze eigenschap moet het mogelijk maken dat:

- de werking snel kan begrepen worden door personen die geen voorkennis hebben van het technisch dossier;
- in geval van storingen men snel in staat moet zijn de onregelmatigheid te lokaliseren en te herstellen. Deze ingrepen moeten door een onderhoudsman met een elementaire kennis van besturingstechnieken, kunnen uitgevoerd worden;
- bij veranderingen of uitbreidingen in het proces de besturing gemakkelijk moet aan te passen zijn, zonder afbreuk te doen aan het geheel van de sturing.

De beschrijving, uitdrukkingen, tekens en symbolen die in een ontwerpmethode worden gebruikt, moeten bij voorkeur op een eenvoudige wijze vertaald worden in de "taal" van de PLC.

Voor de introductie van programmeerbare automaten, werden automatische sturingen verwezenlijkt door relais, stappenschakelaars, e.d., met andere woorden door contacten. Heel frequent wordt tijdens het ontwerpen van automatische sturingen met PLC's nog in contactlogica geredeneerd. Men maakt vooral gebruik van de LAD-programmeerwijze.

Momenteel gebruikt men verschillende programmeermethoden. De meest voorkomende programmeertalen worden als volgt voorgesteld:

Nederlandse benaming	Benaming volgens IEC	Afkorting	Siemens S5
Ladderdiagram	Ladder Diagram	LAD (LD)	KOP
Instructielijst	Statement List (Instruction List)	STL (IL)	AWL
Logisch functiesymbool	Function Block Diagram	FBD	FUP
Functiediagram	Sequential Function Chart	SFC	Graf

Men kan een programma ingeven op verschillende wijzen, LAD en STL. Het programmeertoestel zet het programma om in een machinecode die voor de voorstellingswijzen dezelfde is. Het is mogelijk om een programma dat in LAD werd ingegeven om te zetten in STL, of omgekeerd.

6.1 Ladderdiagram

Het ladderdiagram is een navolging van het gewone stroomkring- of principeschema dat gebruikt wordt in het elektrisch schematekenen. Wegens de eenvoudige weergave op beeldschermen en printers lopen de stroomwegen niet verticaal, maar horizontaal van links naar rechts.

Men tekent de beide stroombanen, respectievelijk de fase- en de nulgeleider, verticaal. De verschillende schakelkringen worden horizontaal voorgesteld.

De symbolen die men gebruikt, zijn aanwezig in om het even welke teksteditor, zodat beeldschermen en printers ze zonder meer kunnen weergeven. De voornaamste symbolen zijn:

Symbolen	Betekenis
	Afvragen naar toestand "1"
	Afvragen naar toestand "0"
	Uitgang of merker
	Setten van uitgang of merker
	Resetten van uitgang of merker

Let op:

In het ladderdiagram wordt het PLC-programma voorgesteld en niet de fysische toestand van de detectoren die aan de ingangen aangesloten zijn.

De voordelen van deze ontwerpmethode zijn:

- Het ladderdiagram is gebaseerd op de relaistechniek, een techniek die diep ingeworteld zit bij alle besturingstechnici en daarom vlot verstaanbaar is voor eenvoudige processen.
- Veel programmeerbare automaten zijn voor relaisvervangende installaties bestemd. Het ontwerp kan tegelijk het programma zelf zijn.
- Vele PLC's kunnen rechtstreeks met ladderdiagramma's worden geprogrammeerd, waardoor het ontwerp gelijktijdig het PLC-programma is.

De belangrijkste nadelen zijn:

- De geavanceerde programmeertechnieken wijken sterk van de gekende contactlogica af. Deze instructies overstijgen de klassieke mogelijkheden. Voor de geavanceerde instructies worden specifieke symbolen gebruikt die enerzijds merkgebonden zijn en anderzijds afwijken van bestaande normen.
- Het ontwerpen is hoofdzakelijk op ervaring gebaseerd en verloopt weinig gestructureerd. Daardoor is het bereikte resultaat voor derden dikwijls weinig overzichtelijk.
- Verduidelijkingen, aan de hand van beschrijvingen, blijken noodzakelijk te zijn.

6.2 Logische functiebouwstenen FBD

De aanduiding FBD is het letterwoord dat verwijst naar "Function Block Diagram". Het is een grafische programmeermogelijkheid.

De symbolen van de schakeltechniek volgens IEC normen worden gebruikt.

Elke poort stelt een elektronische schakelfunctie voor met één of meerdere ingangen en één of meerdere uitgangen. De logische toestand van de uitgang(en) wordt enkel en alleen door de logische toestand van de ingangen bepaald.

Deze methode wordt vooral door personen met een elektronische vooropleiding gebruikt. De transfer naar de logische poorten in de IC's is gemakkelijk te maken.

Deze methode is bijzonder goed geschikt voor het oplossen van combinatorische problemen.

De voornaamste symbolen zijn:

Symbolen	Betekenis
	Identiteit
	Inversie
	EN functie Serieschakeling
	OF functie Parallelschakeling
	Niet - EN
	Niet - OF
	EXOF

6.3 Instructielijst STL

In de instructielijst worden alle functies met gemakkelijk te onthouden mnemotechnische afkortingen geprogrammeerd. Deze methode is de krachtigste en kan de meest complexe functies verwerken.

Doordat de instructies beknopt worden weergegeven, krijgt men veel informatie op een beeldscherm en wordt daardoor de overzichtelijkheid van het programma vergroot.

Bv. LD I0.0
A I0.1
LD I0.4
A I0.5
OLD
= Q0.0
MEND

6.4 Functiediagram SFC

In blok 5 komen we hier uitvoerig op terug omdat deze programmeerwijze duidelijk van de drie voorgaande verschilt. Het is ook niet mogelijk om zonder meer de overstap tussen een functieprogramma en een andere programmeertaal te maken. Bij de vorige is een druk op een toets voldoende om een PLC-programma op de andere manier voor te stellen.

Om een programma op het scherm te tekenen volgens deze methode, heeft men meestal een ander softwarepakket nodig.

7. Programmaverwerking

7.1 Cyclus

De werking van de meeste PLC's is gebaseerd op een cyclische programmaverwerking. Instructies worden in het geheugen opgeslagen. Ze worden één voor één in het besturingsorgaan "gelezen" en uitgevoerd. Na het uitvoeren van alle instructies, een programmacyclus, begint de procedure opnieuw bij de eerste instructie. Omdat het uitvoeren van alle instructies zich steeds herhaalt, spreekt men van de cyclische programmaverwerking.

Indien men de PLC start, worden alle uitgangen eerst op nul gezet.

Aan het begin van elke cyclus vraagt het stuurorgaan de signaaltoestanden van alle ingangen op en maakt er een kopie van in het PII-register. Daarna begint de eigenlijke verwerking van het programma. De instructies worden één voor één afgewerkt in stijgende volgorde van de programmalijnnummers.

In het programma houdt de logica enkel rekening met de inhoud van het PII-register en dus niet met de reële toestand van de ingangen. Dit heeft het voordeel dat tijdens één programmacyclus dezelfde toestand wordt gebruikt voor een ingang. Het gevaar zou er anders in bestaan dat een ingang in het begin van de cyclus logisch "1" zou zijn en op het einde logisch "0". In realiteit zou dit overeenkomen met een signaal dat tegelijkertijd nul en één is. Vermits een cyclustijd zeer kort is (maximum cyclustijd is 300 ms) kan men veronderstellen dat voor "trage" processen het PII-register overeenkomt met de reële toestand van de ingangen.

De resultaten van de logische bewerkingen worden weggeschreven in het PIQ-register. Wanneer het programma de laatste instructie tegenkomt (MEND, master end), worden de toestanden in het PIQ-register gekopieerd naar de uitgangen. Een nieuwe programmacyclus kan beginnen.

Met behulp van de aangepaste instructies (immediate contacts) kan men het in- en uitgangsregister voorbijgaan en de periferie direct aanspreken.

Bij de communicatie tussen PII-register en periferie wordt alleen die periferie betrokken, die daadwerkelijk in het systeem aanwezig is (in- en uitgangskaarten). De aanwezige periferie wordt door het systeem bij elke nieuwe start opnieuw gedetecteerd. Hierdoor wordt een tijdoptimale communicatie mogelijk en bovendien wordt het mogelijk fouten in die periferiebouwgroepen te ontdekken.

De cyclustijd (bv. 350 ms) wordt bewaakt. Wordt een vooraf bepaalde, maximale tijd overschreden, dan gaat de PLC in de STOP-toestand.

De meeste PLC's bezitten de mogelijkheid om naast de cyclische programmaverwerking ook nog aan tijdgestuurde en alarmgestuurde programmaverwerking te doen. Het grote voordeel van alarmgestuurde programmaverwerking situeert zich in de onmiddellijke respons op een binnenkomend alarm en het verlaten van de cyclische programmastructuur. (interrupt programma's)

De verschillende bewerkingen die de PLC achtereenvolgens bij een cyclische programmaverwerking doorloopt, staan hieronder schematisch weergegeven.

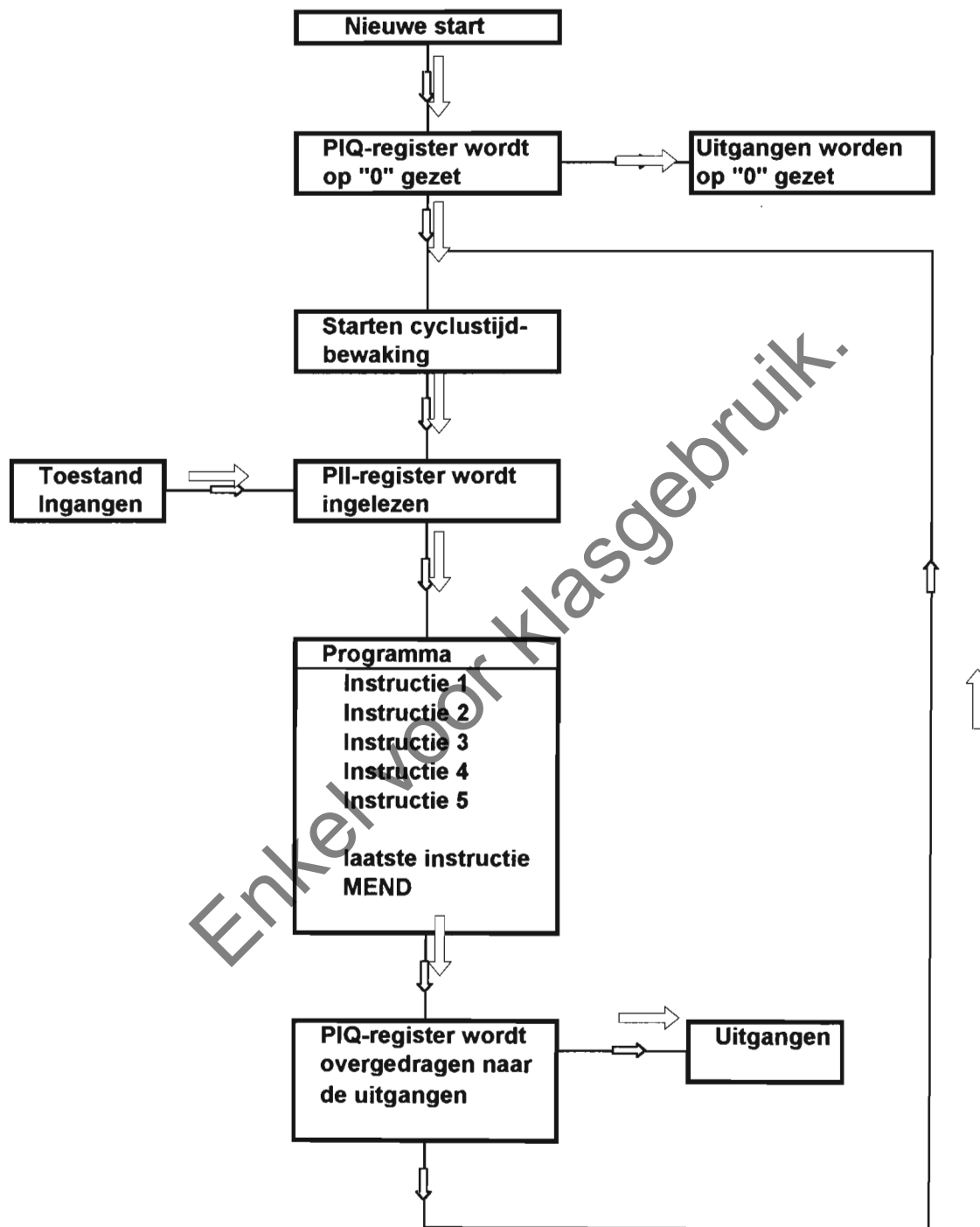


Fig. 2.6 Programmaverwerking

De schakelaar met de schakelstanden "RUN", "TERM" en "STOP" die voorkomt op het voorpaneel van de PLC, werkt direct in op het cyclusverloop. Met het oplichten van een LED op het voorpaneel van de PLC ziet men permanent in welke grondtoestand de PLC zich bevindt.

Werkingstoestand "RUN"

- Het programma wordt cyclisch bewerkt.
- De in het programma gestarte tijden beginnen te lopen.
- De signaaltoestanden van de ingangen worden ingelezen.
- De uitgangen nemen de toestanden aan die aangegeven worden door de PIQ.
- Het is niet mogelijk een programma van het programma van het programmeerapparaat over te dragen in het geheugen van de PLC

Werkingstoestand "STOP"

- Het programma wordt niet verwerkt. Alleen in deze toestand is het mogelijk een nieuw programma in het geheugen van de PLC te laden.
- De waarden van tijden, tellers en merkers die bij het intreden van de STOP-toestand actief waren, worden vastgehouden.
- De uitgangen zijn gesperd (toestand 0).
- De STOP-toestand treedt op om de volgende redenen:
 - de schakelaar "STOP-RUN" wordt op "STOP" geplaatst;
 - de STOP-toestand wordt door een commando van het programmeerapparaat gestuurd;
 - fouten in de programmering beletten een normale afloop van het programma;
 - de cyclustijd is overschreden;
 - bij het terugkeren van het net na een spanningsonderbreking en de voorwaarden voor een automatische NIEUW-start zijn niet vervuld;
 - de stuureenheid heeft een "STOP"-instructie in het programma gevonden en uitgevoerd;
 - hardware fout.

Werkingstoestand "THERM"

Als de schakelaar op het frontpaneel van de PLC in deze stand staat, kan men via het programmeerapparaat de werkingstoestand instellen. Softwarematig kiest men de stand RUN of STOP. De LED-indicatie zal de werkingstoestand aangeven.

7.2 Netwerken

Een PLC-programma wordt opgedeeld in kleine, overzichtelijke delen. De netwerken worden voorzien van een volgnummer. Volgens deze volgorde wordt het PLC-programma verwerkt.

Elk Network kan voorzien worden van een titel. Daardoor wordt het programma beter verstaanbaar en kan men bepaalde acties snel terugvinden in het geheel van het PLC-programma. De gebruiker heeft er echter alle belang bij zijn programma zo duidelijk en overzichtelijk mogelijk op te stellen.

Bij het lezen, foutzoeken en testen kunnen de verschillende delen veel eenvoudiger gelokaliseerd worden.

In één netwerk wordt meestal slechts één merker, uitgang, tijdfunctie, teller, enz. gedefinieerd. Het laatste netwerk wordt gebruikt om het einde van het programma aan te geven.

Door een deel van een programma als een geheel te beschouwen, is het mogelijk een netwerk te kopiëren, te wissen, een nieuw netwerk tussen te voegen, e.d..... De gebruikelijke handelingen die men in een tekstverwerkingsprogramma kan uitvoeren, kunnen toegepast worden op een netwerk.

7.3 Programma documenteren

Men beschikt over verschillende mogelijkheden om het programma te documenteren. Al deze mogelijkheden zijn uitsluitend toegevoegd om het comfort van de gebruiker te verhogen. Wanneer men een PLC-programma opstelt in STL zullen alle lijnen die beginnen met “//” erkend worden als commentaar. Deze lijnen, die men toevoegt aan het PLC-programma, beïnvloeden de werking niet. Ze worden zelfs niet overgedragen naar het geheugen van de PLC. Ze blijven in het geheugen van het programmeerapparaat. Ze worden wel bewaard als men het PLC-programma kopieert op diskette of harde schijf. Bij het afdrukken van een programma worden de titels en commentaarlijnen toegevoegd. Men kan gebruik maken van volgende mogelijkheden:

Titel

Het toevoegen van een titel heeft een aantal voordelen. Wanneer men een programma van de diskette terug in het geheugen van het programmeerapparaat brengt, wordt de titel een referentie om aan te geven dat het juiste programma geladen is. De titel zal dezelfde functie hebben bij het gebruik van een afdruk.

Titelcommentaar

Men beschikt over de mogelijkheid om na de titel één of meerdere lijnen tussen te voegen. In deze rubriek worden opgesomd: de ontwerper, de datum, het project, gegevens van de klant, ed.

Netwerktitel

Deze tussentitel wordt voornamelijk gebruikt om de verschillende delen van de sturing te lokaliseren.

Commentaar per netwerk

Alle hints en bijzonderheden die vermeld moeten worden om de programmeerwijze te verklaren, worden toegevoegd aan het netwerk.

Commentaar per programmalijn

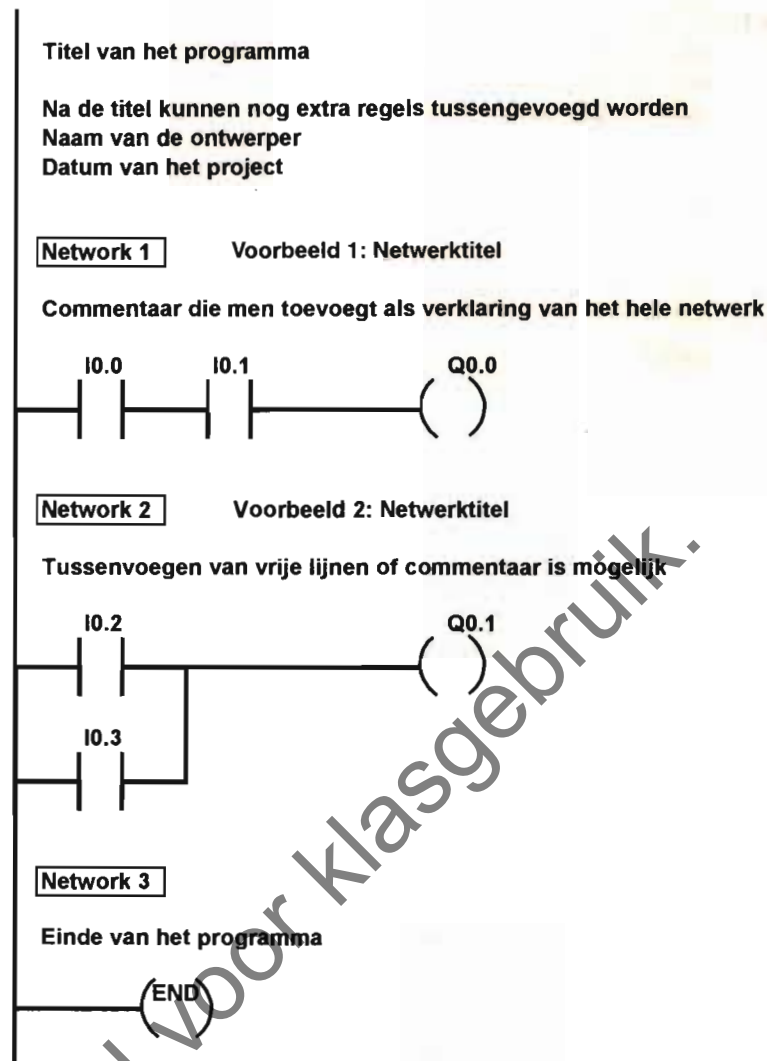
Als men gebruik maakt van de programmeertaal STL, is het mogelijk om elke programmalijn te voorzien van commentaar. Op deze manier wordt het programma duidelijk leesbaar voor derden die niet vertrouwd zijn met het project of het ontwerp.

Hierna wordt een programmavoorbeeld weergegeven dat voorzien is van commentaar. Het programma is opgesteld in STL.

```

1 //
2 // Titel van het programma
3 //
4 // Na de titel kunnen nog extra regels tussengevoegd worden.
5 // Naam van ontwerper
6 // Datum
7 //
8
9 Network 1 // Voorbeeld 1 van een netwerktitel
10 //
11 // Commentaar die men toevoegt als verklaring van het hele netwerk
12 //
13 LD I0.0 // Als de hoofdschakelaar ingeschakeld is
14 A I0.1 // en het thermostaatcontact heeft toestand “1”
15 = Q0.0 // zal de brander geactiveerd zijn.
16
17 Network 2 // Voorbeeld 2 van een netwerktitel
18 //
19 // Tussenvoegen van vrije lijnen of commentaar is mogelijk.
20 LD I0.2 // Als de schakelaar van de benedenverdieping is ingeschakeld
21 O I0.3 // of de schakelaar van de bovenverdieping is ingeschakeld
22 = Q0.1 // dan zal de lamp branden.
23
24 Network 3
25 //
26 // Einde van het programma
27 MEND
```

Als men het hoger vermelde PLC-programma in STL vertaalt tot een LAD-programma dan worden de commentaarlijnen bij de programmalijnen niet toegevoegd. Men verkrijgt:



7.4 Symbolische adressering

Om het programma nog beter verstaanbaar te maken, kan men gebruik maken van symbolische adressering. Zoals hoger reeds aangehaald, wordt een tabel gemaakt waarin de verschillende schakelementen met hun respectievelijke in- en uitgangen worden vermeld.

Het is mogelijk om deze tabel in het geheugen van het programmeerapparaat te brengen. Deze vertaalt dan zelf de gekozen aanduidingen (symbolische adressering) tot absolute adressen. Zowel bij de programmeertaal STL als LAD kunnen symbolische adressen gebruikt worden in plaats van de absolute adressen.

Bij de instellingen, op het scherm van het programmeerapparaat, wordt de keuze "Symbolic Addressing" geactiveerd. Als voorbeeld wordt onderstaande tabel weergegeven:

Symbol Name	Address	Comment
Hoofdsch	I0.0	Hoofdschakelaar
Therm	I0.1	Thermostaat in de woonkamer
Brander	Q0.0	Brander van de centrale verwarming
Sch.ben	I0.2	Schakelaar op de benedenverdieping
Sch.bov	I0.3	Schakelaar op de bovenverdieping
Lamp	Q0.1	Verlichting in de traphal

Bij onderstaand programma, uitgevoerd in STL, worden de commentaarlijnen niet toegevoegd.
Het programma wordt:

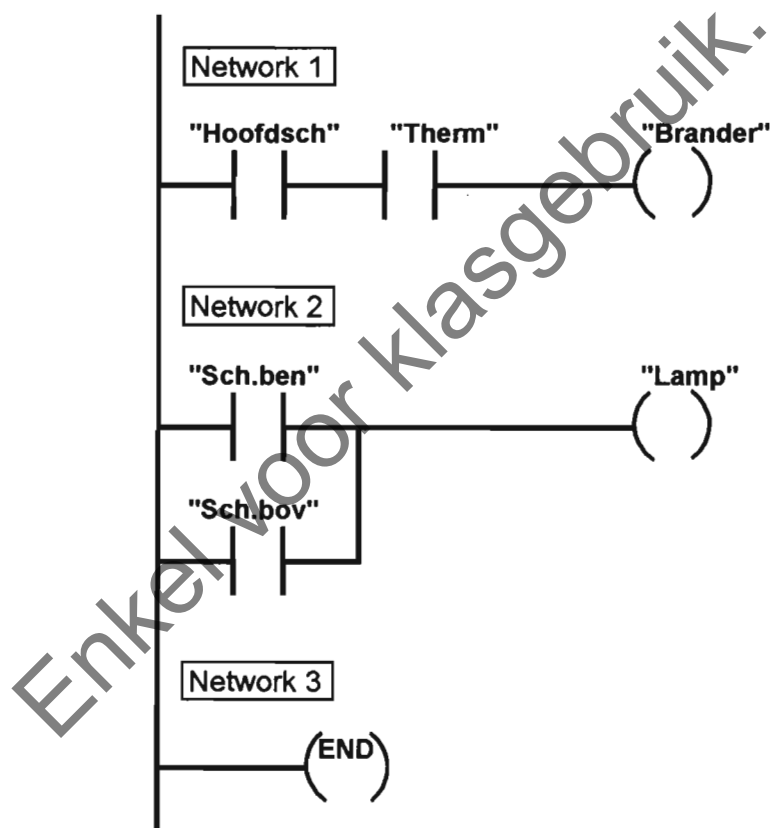
```

Network 1:
LD    "Hoofdsch"
A      "Therm"
=      "Brander"

Network 2:
LD    "Sch.ben"
O      "Sch.bov"
=      "Lamp"

Network 3:
MEND
    
```

Hetzelfde programma, vertaald in LAD, geeft:

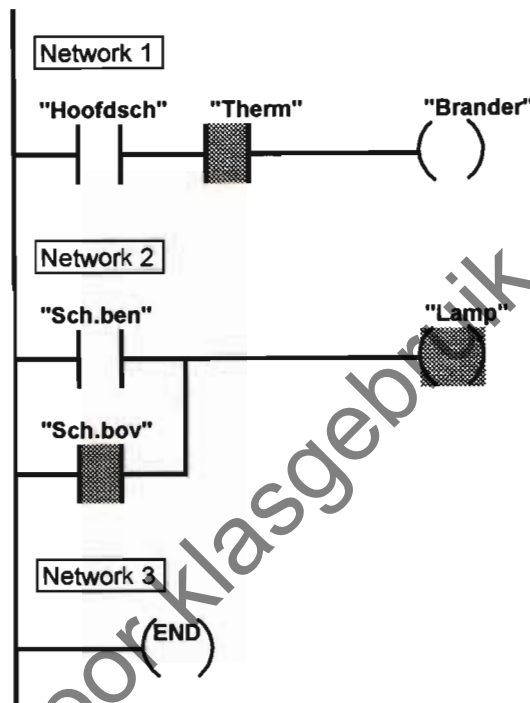


7.5 Testmogelijkheden van PLC-programma's

Het is noodzakelijk dat men absoluut zeker is van een feilloze werking van het PLC-programma, voordat de PLC de kostbare installaties zal sturen.

Na het opstellen van het programma, wordt het overgedragen naar het geheugen van de PLC. Men gebruikt daarvoor het commando: "Download". Bij de keuzemogelijkheden van het scherm wordt de keuze "Debug" geactiveerd. In dit submenu wordt de keuze "Ladder Status On" geactiveerd.

Men gebruikt de programmeertaal LAD om het programma voor te stellen op het scherm. Als de ingangen, uitgangen of merkers de toestand "1" aannemen, dan worden de betrokken velden grijs opgevuld, zoals weergegeven in onderstaand voorbeeld.



Maakt men gebruik van tellers of tijdfuncties dan wordt de momentele waarde weergegeven bij het betrokken symbool.

Als het programma veel langer is dan op één beeldscherm kan worden weergegeven, dan kan men het beeld laten doorschuiven. Op deze manier worden de verbanden niet altijd duidelijk weergegeven. Daarom heeft de fabrikant een extra mogelijkheid voorzien. In een tabelvorm geeft men de verschillende elementen op. Men kiest alleen die elementen waarvan men op dat ogenblik de toestanden wil controleren. Deze tabel wordt zichtbaar als gekozen wordt voor: "Status Chart". De volgende tabel geeft een voorbeeld van een "Status Chart".

Address	Format	Current Value	Change Value
T37	Signed	+ 632	
"Lamp"	Bit	2#0	
"Hoofdsch"	Bit	2#1	
Q0.2	Bit	2#1	

Bij de adressen kan men door elkaar de symbolische of de absolute adressen gebruiken.

Het programmeerapparaat zal zelf aangeven of we te maken hebben met één bit informatie of met een groter informatiegegeven.

Bij tijdfunctie T37 werden in het voorbeeld 632 tijdseenheden geteld. De lamp is niet actief want na het symbool # staat de status "0" weergegeven. De hoofdschakelaar en de uitgang Q0.2 zijn geactiveerd want daar wordt, na het hekje, de toestand "1" aangegeven.

Op deze manier kan men het opgestelde programma controleren op een foutloze werking. Vooraleer men de PLC zal inschakelen in het proces zal een volledige testprocedure doorlopen worden, waarbij onder andere ook de bedrading gecontroleerd wordt. Hoger aangehaalde testmogelijkheden betreffen uitsluitend het testen van het PLC-programma.

7.6 Lineair programmeren

Bij het starten van de PLC is alleen de organisatiebouwsteen OB1 actief. Als men het programma zuiver lineair opstelt zal men alle instructies en de verschillende programmadelen na elkaar plaatsen.

Het volledige programma wordt volledig cyclisch verwerkt zoals uiteengezet in de vorige paragraaf.

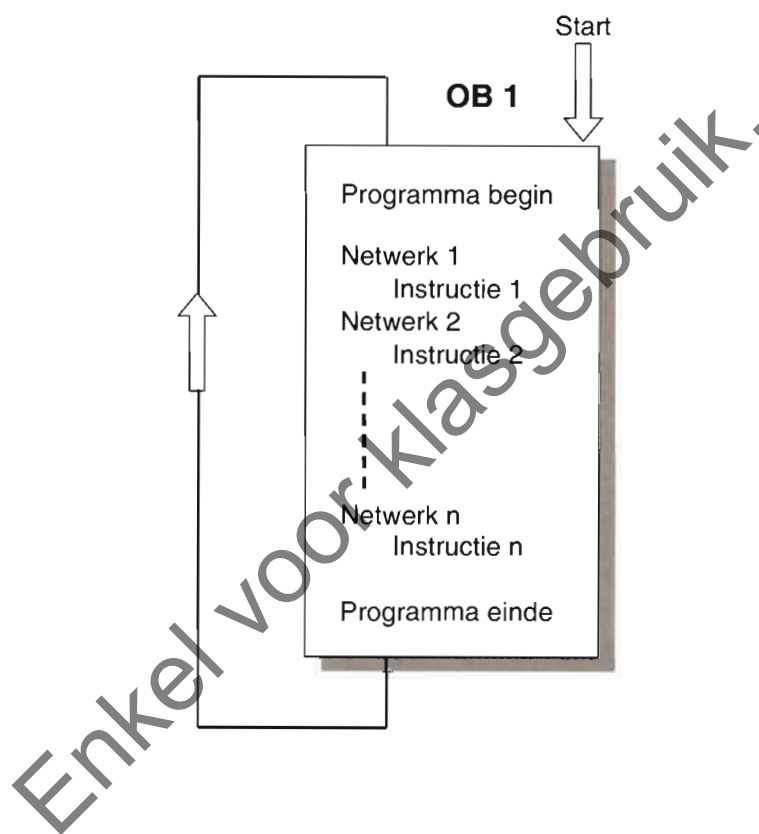


Fig. 2.7 Lineair programma

7.7 Gestructureerd programmeren

Het programma wordt in modules geschreven. Modulair programmeren is bedoeld om programma's gestructureerd te ontwerpen zodat men een overzichtelijk geheel krijgt.

Als een installatie zowel automatisch als manueel moet kunnen functioneren, zal men beide delen in afzonderlijke subroutines programmeren. Deze zullen nooit samen moeten werken. Door het oproepen van de gewenste subroutine, zal de PLC alleen springen naar de geactiveerde subroutine. De overige programmalijnen worden niet verwerkt, zodat de PLC sneller zal reageren. Bovendien zal het testen gemakkelijker worden omdat het programma overzichtelijker wordt.

De volgorde waarin de afzonderlijke bouwstenen opgeroepen en bewerkt moeten worden, worden in het programma vastgelegd. Er moet een onderscheid gemaakt worden tussen subroutines en interrupt programma's. Wanneer aan de geprogrammeerde voorwaarden voldaan is zal het subroutine elke cyclus opnieuw, samen met het hoofdprogramma, worden verwerkt. Interruptprogramma's worden alleen actief als aan speciale voorwaarden voldaan is. Het is mogelijk dat een ingang (noodstop) zal zorgen dat het normale programmaverloop onderbroken wordt en doorgeschoven wordt naar het interruptprogramma. Andere mogelijkheden zijn real-time-clock, snelle telleringang, tijdgestuurde interrupts, e.a.

Na de instructie END in het hoofdprogramma (MEND), worden de nodige subroutines en eventuele onderbrekingsprogramma's toegevoegd. Het is mogelijk om terug te keren naar het hoofdprogramma, afhankelijk van geprogrammeerde voorwaarden. Het is ook mogelijk om aan het einde van een subroutine onvoorwaardelijk terug te keren naar het hoofdprogramma.

Het hoofdprogramma OB 1 wordt cyclisch bewerkt.

Met oproepen kan een hoofdprogramma verlaten worden en kan er naar onderprogramma's gesprongen worden. Men kan tot maximaal 8 niveaus van vernesting gaan, alhoewel de vernestingsdiepte afhankelijk is van het type PLC.

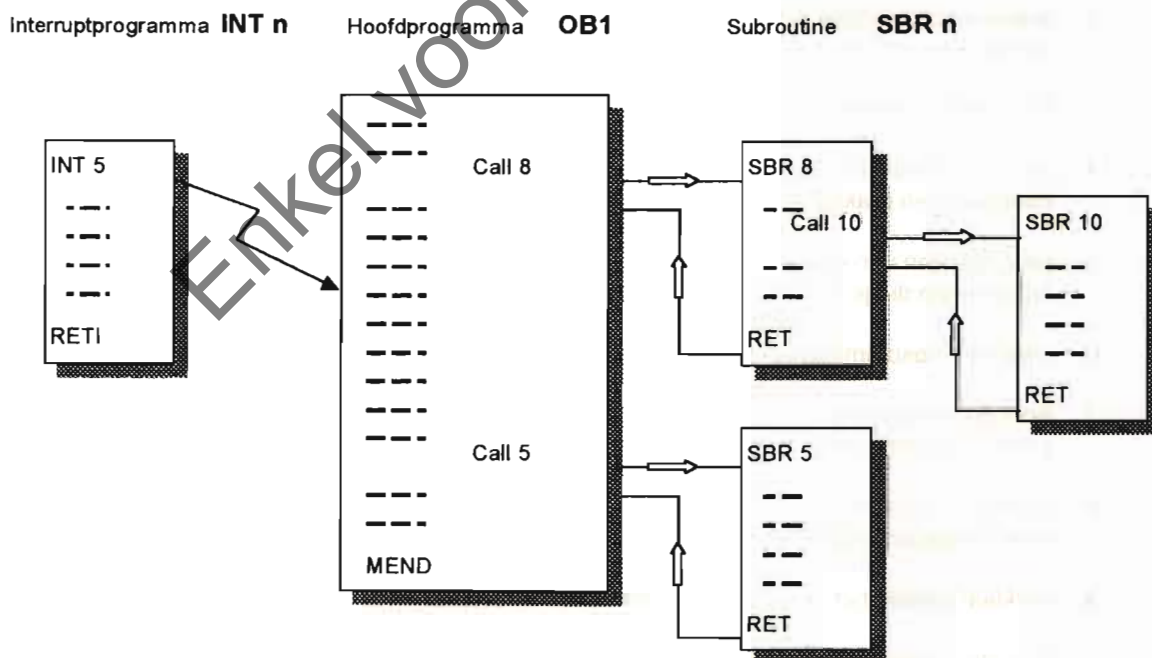


Fig. 2.8 Gestructureerd programmeren

8 Samenvatting

- 1 Een PLC is met een microprocessor vergelijkbaar. Naast de noodzakelijke in- en uitgangen bevat de PLC een centrale verwerkingseenheid.
- 2 De centrale verwerkingseenheid bevat de logische verwerkingseenheid, de accumulator en de tijd- en controle-eenheid.
- 3 De schakelaars en sensoren worden aangesloten aan de ingangen van de PLC.
- 4 De aan te sturen vermogenselementen worden aan de uitgangen van de PLC aangesloten. Afhankelijk van het te realiseren project wordt gekozen uit relais-, transistor- of solid-state-uitgangen.
- 5 Het opgestelde PLC-programma wordt cyclisch verwerkt. De toestanden van de ingangen worden bij het begin van een cyclus in het PII-geheugen bewaard. De programmalijnen worden achtereenvolgens verwerkt en de resultaten in een PIQ-geheugen bewaard. Aan het einde van een cyclus worden de toestanden van het PIQ-geheugen naar de periferie getransfereerd.
- 6 Men kan een programma opstellen en via het programmeerapparaat in het geheugen van de PLC op een verschillende programmeerwijze ingeven. De volgende programmeerwijzen worden veel gebruikt:
LAD: contactlogica of ladderdiagram
STL: instructielijst in mnemotechnische code
FB : logische symbolen

9 Opdrachten

- 1 Teken het blokschema van een PLC.
- 2 Welke voorzieningen zijn er in een PLC aanwezig, die ervoor zorgen dat de PLC zo correct mogelijk de signalen van de detectoren opvolgt?
- 3 Bespreek in verband met de ingangen van de PLC de uitdrukking "toestand 0" en "toestand 1".
- 4 Geef het verschil tussen het arbeidsstroom- en ruststroomprincipe. Verduidelijk je antwoord aan de hand van een praktisch voorbeeld.
- 5 De uitgangen van een PLC kunnen uitgevoerd zijn met relais, transistor of solid-state. Bespreek de argumenten die je keuze zullen bepalen.
- 6 Geef een opsomming van de acties die de PLC uitvoert binnen één cyclus.
- 7 Som de verschillende geheugens op en verklaar het letterwoord. Geef voor elke soort een praktische toepassing.
- 8 Verklaar het verschil tussen een remanente en niet-remanente merker. Geef voor deze soorten typische toepassingsvoorbeelden.
- 9 Verklaar bondig het verschil tussen bit-, byte- en woordadres.
- 10 Som de verschillende delen van een instructie op en benoem de delen.
- 11 Op de voorzijde van de PLC staat een keuzeschakelaar met de aanduidingen: "STOP", "RUN" en "THERM". Bespreek bondig het verschil tussen beide toestanden.
- 12 Geef enkele redenen op waarom het beter is om een PLC-programma gestructureerd op te stellen.
- 13 Bespreek bondig het verschil tussen LAD, STL en FBD.